

Libro de Actas



4to Congreso Virtual de Microcontroladores y sus Aplicaciones

30/09 al 03/10 de 2024

Facultad Regional Paraná



Universidad Tecnológica Nacional
Facultad Regional Paraná
Departamento de Ingeniería Electrónica
Laboratorio de Técnicas Digitales

Cuarto congreso virtual de microcontroladores y sus aplicaciones / Agustín Avendano

...

[et al.] ; Contribuciones de Enrique Sergio Burgos ... [et al.] ; Compilación de Raúl Manuel Caballero ; Coordinación general de Raúl Manuel Caballero. - 1a ed. - Ciudad Autónoma de Buenos Aires : Universidad Tecnológica Nacional, 2024.

Libro digital, PDF

Archivo Digital: descarga y online

ISBN 978-950-42-0246-2

1. Microcontroladores. 2. Electrónica Digital. I. Avendano, Agustín II. Burgos, Enrique Sergio, colab. III. Caballero, Raúl Manuel, comp. IV. Caballero, Raúl Manuel, coord.

CDD 620

Libro de Actas

4to Congreso Virtual de Microcontroladores y sus Aplicaciones

Fecha del Congreso: 30/09 al 03/10 de 2024

Lugar de las Jornadas: Facultad Regional Paraná, Entre Ríos, Argentina

Formato: Virtual

DOI: <https://doi.org/10.33414/ajea.1757.2024>

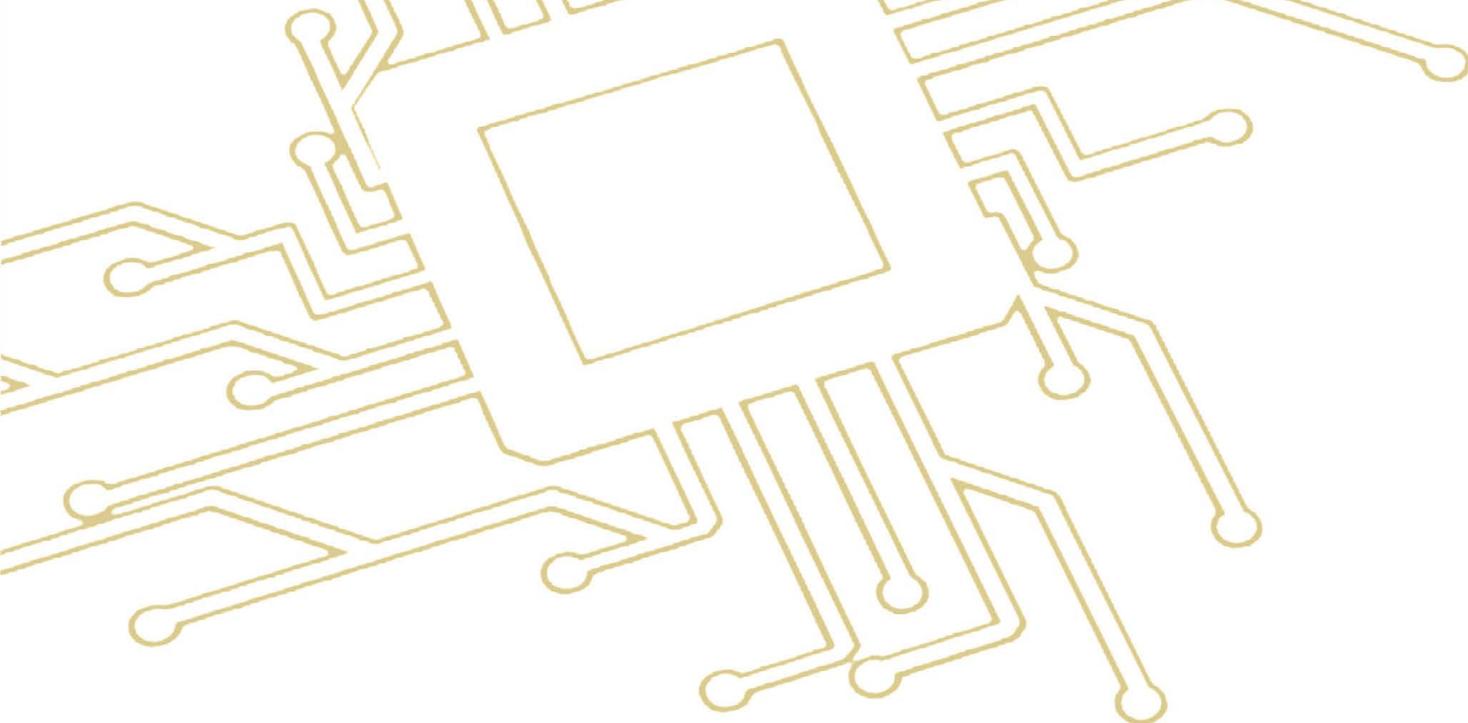
ISBN: 978-950-42-0246-2

Fecha de Publicación: 12/2024

ISBN 978-950-42-0246-2



Esta obra está bajo una licencia de Creative Commons Reconocimiento–NoComercial 4.0 Internacional.



4to Congreso Virtual
de Microcontroladores y sus
Aplicaciones

30/09 al 03/10 de 2024
Facultad Regional Paraná
www.frp.utn.edu.ar

Actas



Índice de Contenidos

Importancia	Pág.10
Un Congreso Virtual	Pág.11
Comité Organizador	Pág.11
Avales al Congreso.....	Pág.11
Los Objetivos.....	Pág.12
Ejes Temáticos para Trabajos.....	Pág.12
Metodología.....	Pág.14
Las Ponencias.....	Pág.15
El Material.....	Pág.17
Ventajas para el asistente	Pág.17
Sobre los asistentes al Congreso	Pág.19
Política de Datos	Pág.23
Seminarios Virtuales	Pág.24
Seminarios realizados durante el Congreso.....	Pág.25
Costos.....	Pág.28
El Cronograma.....	Pág.29
La Encuesta Final	Pág.29
Sobre las valoraciones de los asistentes al Congreso.....	Pág.29
Sobre el Congreso y la Organización.....	Pág.30
Sobre los Resúmenes de Trabajo y las Exposiciones	Pág.32
Afiche.....	Pág.36
Resumen de Congresos anteriores	Pág.37
El Yaguareté.....	Pág.39

Trabajos

Eje Temático: Aplicaciones en Bioelectrónica

Sistema Electrónico de Dosificación de Medicamentos para Adultos Mayores Basado en ArduinoPág.42
Edwin Sánchez Uriza; Ilber Ruge Ruge; Ingrid Ortíz Álvarez

Avances preliminares en el diseño y desarrollo de un implante coclear con fines didácticos mediante procesamiento embebido de señales Pág.47
Edgardo Porral; Gerardo Battaglia; Iván Alejandro Reyes; Martín Enríquez; Axel Gómez

Implementación de placas de desarrollo ESP32 y Raspberry Pi pico para un Sistema Mecatrónico en una Cama hospitalaria Pág.54
Glen Alejandro Guerrero; Andres Felipe Arteaga; Tito Manuel Piamba

Análisis Dinámico de Eventos Transitorios en Señales ECG utilizando Transformada Wavelet.....Pág.62
Juan Vorobioff; Magdalena Cecilia Noé; Martín Vainer; Nahuel Orlando; Javier Horacio Luder; Hernan Manuel Garcia Blesa

Aplicación de bloques de entrada/salida programables a la adquisición de biopotenciales orientada a HMI Pág.69
Paloma Domínguez; Federico N. Guerrero; Enrique M. Spinelli



Posturógrafo basado en el empleo de Microcontrolador, Instrumentación Virtual e Imágenes...Pág.75
Yamel Jesús Almeida Pichardo; Alejandro Rivas Gamallo; Luis Antonio Noa Llorens; Carmen Busoch Morlán;
Carmenchu Regueiro Busoch; Angel Regueiro Gómez

Eje Temático: Aplicaciones en Conectividad y Comunicaciones

Reprogramación de firmware de Arduino a través de transceptor XBEE Pág.80
Rodrigo E. Russo; Melisa G. Kuzman; Walter A. Gemin; Juan M. Lopez; Esteban A. Zapirain

Análisis de protocolo MQTT con broker mosquitto aplicado en dispositivos iot mediante sistemas de control y telemetría de invernaderos Pág.84
Daniel Eduardo Oneddú; Rubén Ricardo Urquijo; Nancy Beatriz Ganz; Marcelo Julio Marinelli

Uso de un contenedor inteligente en la logística del traslado de mercadería Pág.89
Ciro Edgardo Romero

Transmisión on line de datos RIOMETRICOS..... Pág.93
Leonardo J. Navarria; Ricardo E. García; Guillermo D. Rodríguez

AutoTicket: Sistema de cobro por QR con plataforma ESP32 Pág.97
Santiago Rotger; Lucas I. Calderon; Roberto M. Murdocca; Sergio F. Hernandez Velazquez

Aplicaciones de microcontroladores en sistemas de comunicación por luz visible bidireccional led a led Pág.104
Andres Felipe Isaza Piedrahita; Francisco Eugenio Lopez; Roger Alexander Martinez

Módem para telelectura de medidores eléctricos Pág.110
Leonardo Daniel Del Sancio

Sistema de medición y registro multiplexado de termocuplas vía WiFi..... Pág.114
Matias Herrera; Gerardo Imbrioscia; Jonathan Angiano; Agustín Avendano; Pablo Caron

Eje Temático: Aplicaciones en Ecología

Monitorización de variables mediante microcontrolador PIC para el análisis de la eficiencia energética en un prototipo de edificio inteligente Pág.120
Cesar Alejandro Beltran Ospina; Angie Viviana Sandoval Castillo; Ilber Adonayt Ruge Ruge

Análisis de la Linealidad del ADC del ESP32 en la Medición de Radiación UV..... Pág.125
Mario Blas Lavorato; Nicolás Urbano Pintos; Nehuen Oscár Pocovi

Prototipo de vehículo no tripulado de superficie para aplicaciones ambientales..... Pág.130
Ignacio Zaradnik; Oreste Daniel Lupi; Luciano Salvatore; Diego Caccaviello

Evaluación de aguas superficiales en el norte argentino mediante imágenes satelitales..... Pág.136
Juan Vorobioff; Norberto Boggio; Federico Checozzi



Eje Temático: Aplicaciones en Educación y Enseñanza

Validación de sistemas de control en planta de corte al vuelo para el entorno académico Pág.142
Diego Fernando Ramírez Jiménez; Diego Camilo Guisao Herrera; Sebastián Pulido Catillo

Fortalecimiento de habilidades de programación en ingeniería electrónica mediante sistemas microcontrolados..... Pág.153
Ramiro Alejandro Plazas Rosas

Software-in-the-loop para la implementación de una red vehicular con hardware de código abierto Pág.157
Ing. Sergio J. Ortíz-Hernández; Dr. Luis A. Sánchez-Gaspariano; Dr. José E.M. Gutiérrez-Arias

Mejoras en uModelFactory: recursos para el diseño didáctico de sistemas embebidos Pág.165
Nicolás Almaraz; Nahuel González

Automatización y control de redes de distribución de agua Pág.171
Luis Mariano Campos

Implementación de un sistema embebido para la digitalización del registrador de trazos batitermográficos. Pág.184
Walter Gabriel Correa; Christian Galasso

MS-MEGA: Sintetizador monofónico de 2 osciladores de onda cuadrada..... Pág.187
Perez Cerutti, Valentín

Generador de señales en HF para el desarrollo de una ionosonda Pág.192
Leonardo J. Navarria; Ricardo E. García; Guillermo D. Rodríguez

Diseño e implementación de una placa de desarrollo como herramienta educativa..... Pág.197
Wilberth López; Gabriel Fernández; Anthony Quesada

Desarrollo de una plataforma de laboratorios remotos con microcontroladores Pág.202
Marcelo Zabaljauregui; Héctor Hugo Mazzeo; Omar E. Rodríguez; Alejandro M. Rapallini; José A. Rapallini

Diseño de un laboratorio remoto para la enseñanza de sistemas embebidos Pág.208
Juan Piñeiro; Jorge R. Osio; Constantino Palacio; Noemí Arce; Marcelo Cappelletti; Martín Morales

Diseño de Consolas para el Aprendizaje Lúdico hacia la Programación de Microcontroladores PIC y Algunos Periféricos de Interfaz con el Usuario Pág.223
Helen Zambrano; Nury Moreno; Robin Blanco

Aproximaciones sobre el diseño curricular para la formación en sistemas embebidos Pág.231
Marcelo Giura; Marcelo Trujillo; Nahuel González

Avance de adaptador ISA a USB para adaptación tecnológica Pág.242
Sebastian M. Heredia; Roberto M. Murdocca; Sergio F. Hernandez Velazquez

Implementación de un core RISC-V RV32I Pág.247
Roberto G. Simone; Alejandro J. Furfaro; Luciano P. Ferreyro



Prácticas de Programación de sistemas embebidos simples en modo remoto..... Pág.255
Paola I. Beltramini; Pedro Valentín Nieva; Jesús E. Cano

Contador de vueltas reversible para bobinadora de núcleos toroidales Pág.262
Juan Manuel Conde Garrido; Edgardo Vinson; Alejandro Jurado; Josefina María Silveyra

Comunicación entre sistemas embebidos y GNU Octave con fines didácticos..... Pág.267
E. Sergio Burgos

Eje Temático: Aplicaciones de uso Hogareño

Uso de ESP32-CAM para seguridad y vigilancia controlado por Telegram..... Pág.273
Julio Dario Solis Leyes; Roberto M. Murdocca; Sergio F. Hernandez Velazquez

Uso de un Chat Bot de Telegram como plataforma de IOT..... Pág.278
Luis Miguel Echevarría Rodríguez

Eje Temático: Aplicaciones en Inteligencia Artificial

Modelos de inteligencia artificial en microcontroladores de bajo costo (TinyML): aplicación en identificación de colores y reconocimiento de patrones Pág.287
Ilber Adonayt Ruge Ruge; Ingrid Carolina Ortiz Álvarez; Edwin Javier Sánchez Uriza

Rendimiento de algoritmos de reconocimiento facial para autenticación biométrica en sistemas de monitoreo doméstico con Edge Computing..... Pág.292
Vanessa Luna Rosero; Alejandro Muñoz Lopez

Detección de armas sobre video Pág.298
Florencia Priscilla Vela

Diseño de sistemas difusos para FPGAs desde Labview Pág.309
Eduardo García; Yazmin Maldonado

Eje Temático: Aplicaciones en la Industria y Control de Potencia

Modulador Trifásico de Salida Sinusoidal Implementado sobre ESP32 para Accionamiento de Motores y Generación Distribuida Pág.321
Santiago A. Verne

Incubadora de Huevos con "Tecnología Apropriada", Software y Hardware Libre Pág.327
Javier Jorge; Fabricio Puzio; José Amado

Regulador de Potencia para Láseres Pág.333
Lucas Adriel Cusit

Sistemas de control para microreactores nucleares Pág.338
Ruben Roberto Navarro; Juan Carlos Colombo; Viviana Nahid



Sistema de telemedición para una subestación transformadora..... Pág.344
Leonado Daniel Del Sancio

Eje Temático: Aplicaciones de Internet de las Cosas (IoT)

HEPPIE: un algoritmo simple para la estimación de tono, en un amplio espectro, con exactitud y costo computacional ajustables Pág.348
Gerardo A. Laguna-Sánchez; Diana V. Ortiz-Martínez; Víctor M. Pichardo-Infante

Herramientas de Machine Learning para IoT Pág.355
Marcos Darío Aranda; Eduardo Enrique Piray; Paola Inés Beltramini; Ricardo Elian González; Gustavo Andrés Futo; Martin Gianotti

Inclusión de tecnologías IoT y Ciencia de datos en la producción ganadera..... Pág.359
Paola I. Beltramini; Marcos D. Aranda; Jesús E. Cano; Luis D. Villagrán; Sergio H. Gallina; Ivanna M. Lazarte

Uso de redes descentralizadas en sistemas ciber-físicos basados en la placa ESP32 Pág.364
Ciro Edgardo Romero; Alejandro Elustondo

Desarrollo Abierto para Internet de las Cosas y Automatización Industrial: Evaluación de Rendimiento en Entornos Críticos Pág.369
Ricardo Antonio López

Aplicaciones de Internet de las Cosas (IoT) para la conservación y gestión eficiente de la energía... Pág.381
Guillermo Friedrich; Guillermo Reggiani

Desarrollo de un Sistema de Adquisición de Datos Basado en CIAA-Safety Pág.388
Pablo N. SOLIVELLAS; Juan P. RUMIE VITTAR; Darío W. DIAZ

Creación de un sistema IoT embebido en el microcontrolador Esp32 para la gestión de sensores y actuadores en modo AP (Punto de acceso) y STA (Estación)..... Pág.394
Luis Carlos Triviño Torres

Propuesta de Diseño e Implementación de Arquitecturas IoT para el Monitoreo Remoto de Pacientes..... Pág.398
Carlos Torres; Salvador Nuñez; Raul Montiel; Jorge Roa; Sergio Gramajo

Implementación de un Sistema de Gestión de Instrumental y Control de Acceso de Personal. (SiGICAP)..... Pág.401
Alejandro Aldo Lorenzut

Sistema de Monitoreo y Control con tecnología BT e IoT, utilizando la placa de desarrollo ESP32.... Pág.408
Fernando J. Carranza; Francisco M. Cortez Nachar; Roberto M. Murdocca; Sergio F. Hernandez Velazquez

Eje Temático: Aplicaciones en Robótica

Algoritmo de control y observador hiperbólico..... Pág.415
José Guadalupe Chapa Huerta; Fernando Reyes Cortes

Modelo dinámico de un robot RPP..... Pág.424
Jesús Misael González Teco; Moisés Gizeh Jiménez Hernández; José Fernando Reyes Cortés

Control de un robot RPP Pág.432
Moisés Gizeh Jiménez Hernández; Jesús Misael González Teco; José Fernando Reyes Cortés

Diseño de un controlador digital de la velocidad de un motor de corriente directa basado en microcontrolador DSPic30f4012 Pág.440
Ing. Bárbaro Rodríguez Cárdenas; Ing. Arlet Montiel Ramos; Ing. Frank E. Aties Rojas

Interfaz háptica para el control remoto de robots móviles Pág.447
Mateo N. Medina Morales; Pressello Lucas; Roberto M. Murdocca; Sergio F. Hernandez Velazquez

Diseño e Implementación de una banda transportadora automática con el microcontrolador PIC18f45k50 y una interfaz de control local y remota Pág.452
Jeisson Avila; Juan Pablo Blanco; Robin Blanco

Diseño y fabricación de un Robot Delta..... Pág.457
Elio Ruben Ogas; Gabriel Iglesias; Oscar Daniel Moran

Aplicación de Visión Artificial para Robótica Educativa: Reconocimiento y Manipulación de Objetos por Robots Antropomorfo y Delta Pág.465
Elio Ruben Ogas; Gabriel Iglesias; Funes Matias; Oscar Daniel Moran

Desarrollo de Aplicaciones Interactivas para la Robótica Educativa con Godot Engine y ESP32..... Pág.471
Andrés Tapari

Diseño de Robot de Exploración para Entornos Peligrosos con Sistema de Monitoreo y Control Basado en el microcontrolador ESP32 Pág.473
Edwin Yair Osorio Cuatrecatl; Emanuel Ojeda Perez; Alma S. Lizet Salazar Valdez

Eje Temático: Aplicaciones en Seguridad

Sensor de Inclinación y Vibración para Monitoreo de Apuntalamientos en Estructuras Colapsadas Pág.481
Juan M. Souto; Claudio J. Paz

Eje Temático: Aplicaciones en Software

telBot, una biblioteca para la implementación de clientes de mensajería Telegram Pág.487
E. Sergio Burgos

Implementación de Modelos de Software en la Formación de Sistemas Embebidos: Un enfoque Metodológico..... Pág.492
Roberto M. Murdocca; Sergio F. Hernandez Velazquez; Claudio Caballero

Aporte a la confiabilidad de software embebido escrito en C mediante contratos y análisis estático con VeriFast Pág.498
Jordán Francisco Insfrán; Javier Eduardo Diaz Zamboni



Importancia

Las presentaciones de trabajos, publicación de papers, exposición de proyectos, reuniones científicas, seminarios y congresos, entre otros, son actos que fortalecen nuestro sistema científico y de conocimiento.

En la actualidad, resulta imposible que un profesional logre dominar todos los campos del saber de su especialidad. De ahí que los congresos científicos, jornadas, seminarios y eventos de divulgación vienen a ser los forjadores de la continuidad en la capacidad y el avance profesional. Por ello, la asistencia del profesional a estos eventos le permiten madurar en el campo de su especialidad.

Los congresos no sólo son la continuidad de la universidad, también son una oportunidad de profesionales, estudiantes e idóneos para fortalecer sus conocimientos e incursionar en otros. Es el lugar donde se consolidan los conocimientos adquiridos y se aprenden otros nuevos. Los eventos científicos, nacionales o internacionales, nos presentan estas oportunidades.

El contacto con profesionales; nacionales o extranjeros; con mayor experiencia en un determinado campo, el contacto social con colegas de diferentes especialidades, el encuentro inesperado con un investigador experimentado, producen resultados invaluable que van a beneficiar al individuo y a la institución que hace posible la asistencia de los profesionales a estos eventos científicos y técnicos.

Lógicamente, para poder asistir a un congreso hay gastos y erogaciones inevitables. Pero el costo económico de la asistencia y participación en un congreso científico es simplemente secundario, cuando se toman en cuenta los beneficios que se obtienen. Es más caro no disponer de ellos.

Afortunadamente; este aspecto; no debió considerarse en este caso por las características propias de nuestro Congreso Virtual que como en oportunidades pasadas fue totalmente gratuito para todos aquellos que participaron.

En nuestra propuesta utilizamos múltiples sistemas de interacción y comunicación, para expresar el estado del arte de la ingeniería y los nuevos roles de los profesionales.

Profesionales, expertos, desarrolladores, docentes y alumnos exigen repensar la forma de realización de eventos de estas características y asegurar la calidad del proceso.

Este método de difusión nos ha permitido constatar que el intercambio de conocimientos y la numerosa participación nacional; unida a la participación internacional; demuestran que se dispone de un capital humano y técnico extraordinario que hay que cultivarlo, protegerlo e incentivarlo para proyectar nuevos logros.

En lo específico, con este, hemos organizado cuatro Congresos Virtuales de Microcontroladores y sus Aplicaciones, durante los años 2009, 2010, 2014 y ahora en 2024 con nutrida concurrencia. La procedencia de asistentes a los mismos fue de todo Iberoamérica y las expresiones de evaluación vertidas han sido siempre positivas, lo cual nos alienta a continuar en el camino que nos hemos trazado.



Un Congreso Virtual

Si bien se realizan otros congresos sobre tecnología, proyectos de investigación y microcontroladores, muchas veces estos están fuera del alcance de los estudiantes, pequeños desarrolladores y profesionales en general.

Gastos de alojamientos y pasajes hasta estos congresos, son factores excluyentes a la hora de participar en ellos. En otros casos la imposibilidad de asistir se debe a compromisos y responsabilidades laborales y/o personales.

Intentamos por ello, buscar equidad e incorporar a la comunidad entera al debate sobre estos temas, abriendo canales de participación para la comunicación de las numerosísimas experiencias realizadas en las distintas instituciones educativas, empresas y usuarios particulares.

Ya hemos realizado parte del camino en ediciones pasadas y con este Cuarto Congreso se ha buscado mejorar el intercambio buscando una sinergia más armoniosa que nos permita crecer y fortalecernos en el lugar donde nos encontremos beneficiando y mejorando la sociedad que nos rodea.

Comité Organizador

Coordinador

Ing. Raúl Manuel Caballero

Colaboradores

Ing. Enrique Sergio Burgos

Ing. Alejandro Raúl Dachary

Ing. Pablo Javier Suculini

Ing. Rubén Adrián Britos

Ing. Gustavo Daniel Maggiolo

Sr. Joaquín Leikam

Sr. Darío Hernán Klug

Sobre los avales al Congreso

Apoyaron y avalaron

- » Consejo Superior de la Universidad Tecnológica Nacional RESOLUCION N°300/2024
- » Consejo Directivo de la Facultad Regional Paraná RESOLUCIÓN N°614/2023
- » Consejo departamental Ingeniería Electrónica Facultad Regional Paraná Universidad Tecnológica Nacional. DISPOSICIÓN N°78/2023
- » Consejo Directivo de la Facultad de Ciencia y Tecnología de la Universidad Autónoma de Entre Ríos RESOLUCION N°235/2024

- » Consejo Directivo de la Facultad de Ingeniería de la Universidad Nacional de Entre Ríos RESOLUCIÓN N° 12.520/24
- » Facultad de Tecnología y Ciencias Aplicadas de la Universidad de Catamarca RESOLUCIÓN N° 233/24
- » Consejo Directivo de la Facultad de Mecatrónica de la Universidad Mariana – Colombia
- » Secretaría de Modernización de la Provincia de Entre Ríos RESOLUCION N°10/2024

Los Objetivos

- Abrir un espacio de reflexión y debate sobre el uso de microcontroladores como solución a problemas y desarrollo de productos.
- Fomentar el desarrollo tecnológico y la investigación.
- Promover mediante este tipo de encuentro la calidad y actualización académica.
- Divulgar modelos y tendencias tecnológicas.
- Promover el intercambio institucional, profesional y personal entre todos los participantes respecto de experiencias, métodos y desarrollos de su trabajo con microcontroladores y sistemas embebidos.
- Generar un espacio de discusión de amplia cobertura sobre la temática.
- Difundir en el ámbito profesional y académico las tecnologías asociadas a los sistemas microcontrolados.
- Fomentar la interacción industria-academia en temas asociados a los sistemas microcontrolados.
- Incentivar entre los estudiantes universitarios y los jóvenes profesionales el interés por los sistemas de control digital microcontrolado.
- Permitir un mayor acceso de los interesados a este conocimiento, sin importar la ubicación geográfica que los mismos tengan.

Ejes Temáticos para Trabajos

El desarrollo del congreso estará enfocado por las siguientes líneas temáticas:

Aplicaciones Académicas



Aplicaciones y publicaciones sobre herramientas de estudio y desarrollo.

Programas educativos, aspectos metodológicos y de enseñanza, actividades, etc

Aplicaciones en Bioelectrónica



Todas aquellas aplicaciones que se utilicen en las áreas de las ciencias médicas, biológicas, kinésicas o de tratamiento, de laboratorio y afines al diagnóstico y tratamiento de las personas.

Aplicaciones en Comunicaciones/Conectividad



Todas aquellas aplicaciones que incluyan en su diseño aspectos de la comunicación tanto cableada como inalámbrica. En el primer caso implementaciones con USB, Ethernet, CAN, protocolos series diversos, etc. Mientras que en las segundas aplicaciones sobre Bluetooth, RF, IrDA, ZigBee, Wíreless, IEEE 802.1x, RFID, etc.

Aplicaciones de Concepto Ecológico



Aquellas aplicaciones que buscan reducir el impacto ambiental, manejo eficiente de la energía, energías renovables, y protección del medio ambiente. Se incluye además todas aquellas aplicaciones para el correspondiente sensado de las variaciones del ambiente y los sistemas como así también las que permiten el tratamiento de los elementos para reducir el impacto en el ambiente.

Aplicaciones en Dispositivos Hogareños



Todas aquellas aplicaciones posibles de utilizar en la electrónica de consumo aplicada a electrodomésticos, aplicaciones domóticas y de ayuda al confort hogareño.

Aplicaciones en Control de la Industria y Potencia



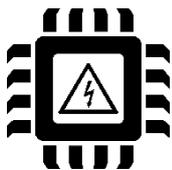
Todas aquellas aplicaciones para el control de motores, de iluminación, soldadoras, compactadoras, empacadoras, dobladoras, etc. Se incluyen aplicaciones de control de altas cargas de energía eléctrica. Control de fuentes de energía. Máquinas herramientas. Aplicaciones para uso comercial. Automatizaciones para producción.

Aplicaciones en Robótica



Todas aquellas aplicaciones para el diseño de robots, control de sus motores de desplazamiento y movimiento, sensores, etc.

Aplicaciones en Seguridad



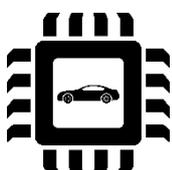
Todas aquellas aplicaciones que estén referidas al uso de la electrónica para la seguridad personal, del hogar, la industria y otras.

Aplicaciones de uso de Software



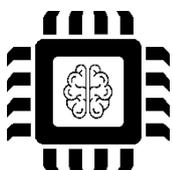
Todas aquellas aplicaciones que estén referidas al uso normal del software para el diseño de aplicaciones con los microcontroladores. Tipo de software a utilizar, uso de métricas, plantillas, algoritmos de uso específico, análisis de uso de recursos, etc. Trabajo con Sistemas Operativos de Tiempo Real. Software Embebido

Aplicaciones en el Transporte



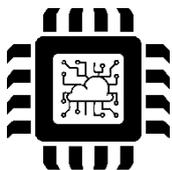
Todas aquellas aplicaciones que se utilicen en la industria del transporte, sea terrestre, marítimo o aéreo. Aplicaciones diseñadas para la ayuda a la navegación, el control de los diferentes sistemas de tracción y energía, los avisos de funcionamiento, etc. También se incluyen aquellas de ayuda al transporte y su control

Aplicaciones con Inteligencia Artificial



Todas aquellas aplicaciones que integren microcontroladores con técnicas de inteligencia artificial, como sistemas de aprendizaje automático, visión por computadora, procesamiento de lenguaje natural, toma de decisiones autónomas, etc.

Aplicaciones Internet de las Cosas



Todas aquellas aplicaciones que permiten la conectividad de dispositivos a través de la Internet de las Cosas, abarcando sensores, actuadores y sistemas embebidos.

Metodología

Se propuso reunirnos virtualmente por un período de cuatro días, durante el mes de setiembre/octubre para la realización de este evento. Esto ocurrió a partir del 30 de setiembre de 2024 fecha en que se realizó la apertura del Congreso.

El Congreso se sustanció sobre los entornos que se presentan en las siguientes URLs



<http://www.cvm.utn.edu.ar> <http://www.congresovirtual.utn.edu.ar>

Luego de la apertura quedaron abiertos los foros de debate, uno por cada línea temática y también uno por ponencia. Cada ponencia forma un tópico específico, en el que se sustanció el debate moderado por los propios ponentes.

Todos los asistentes y ponentes del congreso dispusieron de la posibilidad de participar en todos los espacios de debate así creados.

Las instituciones que presentaron ofertas educativas, contaron con un espacio para la difusión de la propuesta educativa, un foro de debates y una cuenta de correo interna para el planteamiento y resolución de las dudas.

Las instituciones o empresas que presentaron productos, contaron con un espacio para la difusión de su producto, un foro de debates y una cuenta de correo interna de la plataforma para la resolución de dudas.

Durante el desarrollo del Congreso, se dictaron una serie de seminarios web (24) sobre temas específicos, relacionados directamente con el desarrollo de aplicaciones con microcontroladores. Estos seminarios fueron de acceso libre para los participantes y ponentes del Congreso.

Para el desarrollo de estos seminarios web se utilizó una plataforma específica, por esta razón los usuarios debieron inscribirse a cada seminario en que estaban interesados en participar. El calendario de seminarios se publicó oportunamente en el sitio Web del Congreso indicando la temática, el o los expositores, la fecha y el horario de exposición en los husos horarios de los participantes del Congreso.

Finalmente, al cuarto día se produjo el cierre del Congreso.

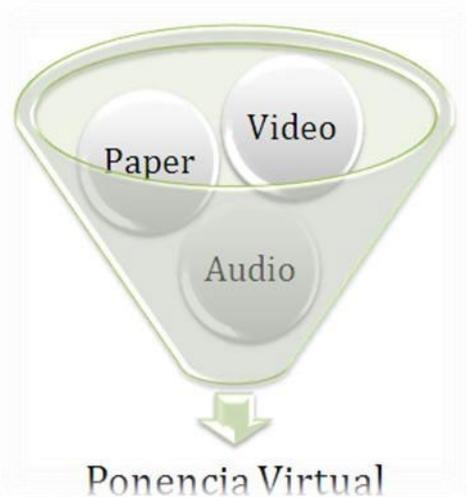
Luego de este cierre, por un período de 14 días, se mantuvo la plataforma disponible en su acceso para extender la continuidad de consultas e intercambio entre asistentes y ponentes sobre los trabajos publicados.

Con posterioridad y según el Cronograma se puso a disposición los certificados de asistencia y de participación a expositores, asistentes y autores.

Las Ponencias

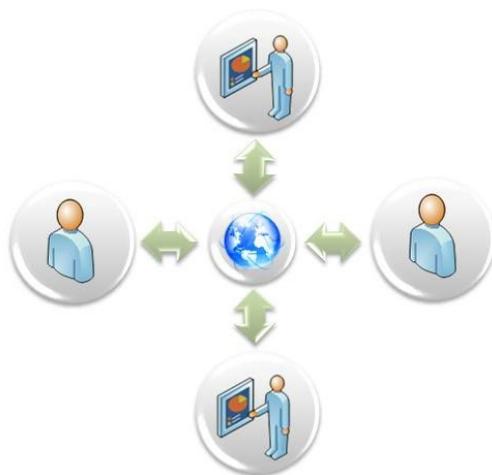
¿Qué entendemos por ponencia virtual?

Es una ponencia presentada en formato digital, publicada en la web del congreso y sobre la cual, los congresistas pudieron leer y navegar interactivamente desde la red mencionada, (una ponencia virtual admite información textual, gráfica, vídeo, audio, etc. y la posibilidad de interactuar sobre ella o acceder a las webs sugeridas en la ponencia).



Cada **PONENCIA** tiene asociada una **TEMÁTICA** concreta. Para cada temática se ofreció un **FORO** activo durante el Congreso y un enlace al **CORREO ELECTRÓNICO** del ponente o ponentes. Con esta metodología, una ponencia virtual ofreció una gran interacción a todos los niveles.

Una ponencia a través de Internet ofrece las mismas facilidades para los asistentes que una presencial, gracias al conjunto de herramientas adicionales asociadas: foros, chat, seminario web y correo electrónico directo con los ponentes. En realidad, la principal diferencia reside en que toda la información presentada está en formato digital, pero a la vez, este formato permite un dinamismo de contenido y duración imposible de conseguir en una presentación convencional.



- **Interacción entre ponentes:** Mediante el chat, foros de discusión, seminarios web y correo electrónico
- **Interacción entre congresistas:** Mediante el chat, seminarios web, foros y correo electrónico.
- **Interacción entre ponente y congresista:** Mediante la exposición y lectura de la ponencia, la visualización del material gráfico anexo a la ponencia (vídeo, fotografías, gráficos, etc.), acceso directo al correo electrónico del ponente, foros de discusión y chats del área temática.



El Material

Las ponencias estuvieron disponibles toda la duración del Congreso y finalizado el mismo por catorce días más. Todas ellas pudieron ser accedidas desde Internet siendo la única condición de acceso por parte de los usuarios de estar registrados. Con la presente edición y publicación que realizará AJEA (Actas de Jornadas y Eventos Académicos) de la Universidad Tecnológica Nacional en la República Argentina también se encuentran disponibles en su repositorio

Además de los Trabajos remitidos por las/los autoras/es, se contó con un conjunto de seminarios web de reconocidos especialistas y empresas líderes en la temática. Estos se han publicado a una plataforma de streaming de acceso libre en el canal del Congreso https://www.youtube.com/@CVM_UTN

Los trabajos se mantendrán durante cierto tiempo en el mismo sitio y finalmente se compilarán junto con todos los Seminarios Web y contarán con su correspondiente ISBN. El mismo es gestionado por la Organización a través de la Secretaría de Extensión Universitaria de la Universidad Tecnológica Nacional en RTyC disponiendo además de su correspondiente DOI o Identificador de Objeto Digital.

Ventajas para el asistente

Disponer de libre acceso para su lectura de todos los trabajos y ponencias presentados.

En cualquier momento puede consultar la información tantas veces como lo requiera y enviar tantas preguntas al ponente como sean necesarias.

Consultar todas las preguntas y respuestas generadas durante la publicación de cualquier ponencia por todos los participantes en el foro de discusión.

Intercambiar vivencias y opiniones en directo con ponentes y otros congresistas en el chat temático

Contactar en forma directa al autor o ponente

Acceder a las exposiciones programadas a través de Seminarios Web habiéndose registrado previamente

Disponer al final del Congreso de un Certificado Digital de Asistencia si ha cumplido con los requerimientos de asistencia y participación del mismo



Compromiso de los Ponentes*

Los autores se debían comprometer a:

- Responder diariamente las consultas por correo electrónico de los participantes en un tiempo de respuesta inferior a las 24 hs. durante el Congreso. Sin mérito que con posterioridad mantenga la comunicación con quienes lo contactan.
- Participar en el foro temático correspondiente, respondiendo a las preguntas referidas a su trabajo y ponencia
- Responder a la Encuesta de Fin de Congreso

Compromiso de los Asistentes*

Los asistentes debían comprometerse a:

- Acceder diariamente durante el período de exposición de trabajos y ponencias del Congreso.
- Participar en forma activa, a través de consultas en los foros correspondientes de al menos cuatro (4) ponencias.
- Asistir en forma virtual al menos a dos (2) seminarios web
- Responder a la Encuesta de Fin de Congreso

***Estas condiciones debieron cumplirse a efectos de poder acceder a la certificación como asistente o como ponente según corresponda**

Los criterios de evaluación tenidos en cuenta por parte del Comité Científico para la evaluación de Trabajos fueron los siguientes:

- La congruencia del Tema del Trabajo presentado con alguna de las líneas temáticas establecidas.
- La estructuración del Trabajo de manera clara, concisa y organizada.
- La especificación de manera clara si se trata de una experiencia institucional, o de un ensayo producto de una investigación.
- El aporte neto que produce el Trabajo al conocimiento de la comunidad académica.

El dictamen se envió vía correo electrónico conforme al calendario previsto.

La Organización del Congreso generó digitalmente las certificaciones correspondientes a quienes cumplieron los requisitos de participación, siendo debidamente notificados de la disponibilidad de las mismas.

Sobre los asistentes al Congreso

Un indicador de sumo interés es la cantidad de asistentes al Congreso.

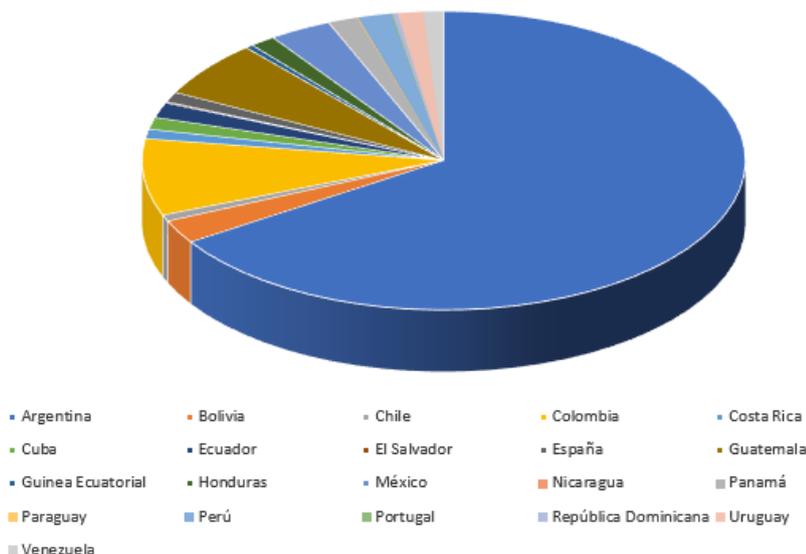
Al mismo se pre inscribieron un mil seiscientos un (1601) personas si bien sólo se tomó el registro de 1459 personas.

Los orígenes son diversos, teniendo asistentes de Argentina, Bolivia, Chile, Colombia, Costa Rica, Cuba, Ecuador, El Salvador, España, Guatemala, Guinea Ecuatorial, Honduras, México, Nicaragua, Panamá, Paraguay, Perú, Portugal, República Dominicana, Uruguay y Venezuela.

La distribución en fue la siguiente:

PAIS	Porcentaje sobre el Total	CANTIDAD
Argentina	65,80%	960
Bolivia	2,54%	37
Chile	0,69%	10
Colombia	8,29%	121
Costa Rica	1,03%	15
Cuba	1,30%	19
Ecuador	1,64%	24
El Salvador	0,14%	2
España	1,10%	16
Guatemala	6,10%	89
Guinea Ecuatorial	0,41%	6
Honduras	1,37%	20
México	3,36%	49
Nicaragua	0,07%	1
Panamá	1,51%	22
Paraguay	0,07%	1
Perú	1,78%	26
Portugal	0,07%	1
República Dominicana	0,27%	4
Uruguay	1,37%	20
Venezuela	1,10%	16

Origen de los Inscriptos



Para el debate, se presentaron 72 Ponencias o Resúmenes de Trabajo con los siguientes títulos



Aplicaciones en Bioelectrónica

Sistema Electrónico de Dosificación de Medicamentos para Adultos Mayores Basado en Arduino

Avances preliminares en el diseño y desarrollo de un implante coclear con fines didácticos mediante procesamiento embebido de señales

Implementación de placas de desarrollo ESP32 y Raspberry Pi pico para un Sistema Mecatrónico en una Cama hospitalaria

Análisis Dinámico de Eventos Transitorios en Señales ECG utilizando Transformada Wavelet

Aplicación de bloques de entrada/salida programables a la adquisición de biopotenciales orientada a HMI

Posturógrafo basado en el empleo de Microcontrolador, Instrumentación Virtual e Imágenes

Aplicaciones en Conectividad y Comunicaciones

Reprogramación de firmware de Arduino a través de transceptor XBEE

Análisis de Protocolo MQTT Con Broker Mosquitto Aplicado en dispositivos IoT Mediante Sistemas de Control y Telemetría de Invernaderos

Uso de un contenedor inteligente en la logística del traslado de mercadería

Transmisión on line de datos RIOMETRICOS

AutoTicket: Sistema de cobro por QR con plataforma ESP32



Aplicaciones de Microcontroladores en Sistemas de Comunicación por Luz Visible Bidireccional Led a Led

Módem para telelectura de medidores eléctricos

Sistema de medición y registro multiplexado de termocuplas vía WiFi

Aplicaciones en Ecología



Monitorización de variables mediante microcontrolador PIC para el análisis de la eficiencia energética en un prototipo de edificio inteligente

Análisis de la Linealidad del ADC del ESP32 en la Medición de Radiación UV

Prototipo de vehículo no tripulado de superficie para aplicaciones ambientales

Evaluación de aguas superficiales en el norte argentino mediante imágenes satelitales

Aplicaciones en Educación y Enseñanza



Validación de sistemas de control en planta de corte al vuelo para el entorno académico

Fortalecimiento de habilidades de programación en ingeniería electrónica mediante sistemas microcontrolados

Software-in-the-loop para la implementación de una red vehicular con hardware de código abierto

Mejoras en uModelFactory: recursos para el diseño didáctico de sistemas embebidos

Automatización y control de redes de distribución de agua.

Implementación de un sistema embebido para la digitalización del registrador de trazos batitermográficos

MS-MEGA: Sintetizador monofónico de 2 osciladores de onda cuadrada

Generador de señales en HF para el desarrollo de una ionosonda

Diseño e implementación de una placa de desarrollo como herramienta educativa

Desarrollo de una plataforma de laboratorios remotos con microcontroladores

Diseño de un laboratorio remoto para la enseñanza de sistemas embebidos

Diseño de Consolas para el Aprendizaje Lúdico hacia la Programación de Microcontroladores PIC y Algunos Periféricos de Interfaz con el Usuario

Aproximaciones sobre el diseño curricular para la formación en sistemas embebidos

Avance de adaptador ISA a USB para adaptación tecnológica

Implementación de un core RISC-V RV32I

Prácticas de Programación de sistemas embebidos simples en modo remoto

Contador de vueltas reversible para bobinadora de núcleos toroidales

Comunicación entre sistemas embebidos y GNU Octave con fines didácticos

Aplicaciones de uso Hogareño



Uso de ESP32-CAM para seguridad y vigilancia controlado por Telegram

Uso de un Chat Bot de Telegram como plataforma de IOT

Aplicaciones en Inteligencia Artificial



Modelos de inteligencia artificial en microcontroladores de bajo costo (TinyML): aplicación en identificación de colores y reconocimiento de patrones

Rendimiento de algoritmos de reconocimiento facial para autenticación biométrica en sistemas de monitoreo doméstico con Edge Computing

Detección de armas sobre video

Diseño de sistemas difusos para FPGAs desde Labview

Aplicaciones en la Industria y Control de Potencia



Modulador Trifásico de Salida Sinusoidal Implementado sobre ESP32 para Accionamiento de Motores y Generación Distribuida

Incubadora de huevos con "Tecnología Apropriada", software y hardware libre

Regulador de Potencia para Láseres

Sistemas de control para micro reactores nucleares

Sistema de telemedición para una subestación transformadora

Aplicaciones de Internet de las Cosas (IoT)



HEPPIE: un algoritmo simple para la estimación de tono, en un amplio espectro, con exactitud y costo computacional ajustables

Herramientas de Machine Learning para IoT

Inclusión de tecnologías IoT y Ciencia de datos en la producción ganadera

Uso de redes descentralizadas en sistemas ciber-físicos basados en la placa ESP32

Desarrollo Abierto para Internet de las Cosas y Automatización Industrial: Evaluación de Rendimiento en Entornos Críticos.

Aplicaciones de Internet de las Cosas (IoT) para la conservación y gestión eficiente de la energía

Desarrollo de un Sistema de Adquisición de Datos Basado en CIAA-Safety

Creación de un sistema IoT embebido en el ESP32 para la gestión de sensores y actuadores en modo AP (Punto de Acceso) y STA (Estación)

Propuesta de Diseño e Implementación de Arquitecturas IoT para el Monitoreo Remoto de Pacientes

Implementación de un Sistema de Gestión de Instrumental y Control de Acceso de Personal. (SiGICAP)

Sistema de Monitoreo y Control con tecnología BT e IoT, utilizando la placa de desarrollo ESP32

Aplicaciones en Robótica

Algoritmo de control y observador hiperbólico

Modelo dinámico de un robot RPP

Control de un robot RPP

Diseño de un controlador digital de la velocidad de un motor de corriente directa basado en microcontrolador DSPic30f4012.

Interfaz háptica para el control remoto de robots móviles

Diseño e Implementación de una banda transportadora automática con el microcontrolador PIC18f45k50 y una interfaz de control local y remota

Diseño y fabricación de un Robot Delta

Aplicación de Visión Artificial para Robótica Educativa: Reconocimiento y Manipulación de Objetos por Robots Antropomorfo y Delta.

Desarrollo de Aplicaciones Interactivas para la Robótica Educativa con Godot Engine y ESP32

Diseño de Robot de Exploración para Entornos Peligrosos con Sistema de Monitoreo y Control Basado en el microcontrolador ESP32

Aplicaciones en Seguridad

Sensor de Inclinación y Vibración para Monitoreo de Apuntalamientos en Estructuras Colapsadas

Aplicaciones en Software

telBot, una biblioteca para la implementación de clientes de mensajería Telegram

Implementación de Modelos de Software en la Formación de Sistemas Embebidos: Un enfoque Metodológico

Aporte a la confiabilidad de software embebido escrito en C mediante contratos y análisis estático con VeriFast



Política de Datos

Exención de responsabilidad

La organización del Congreso publicó y mantiene su página web con el fin de optimizar el acceso público a información sobre sus actividades y conferencias.

La página web se actualiza y mejora con frecuencia. Los nuevos contenidos se añaden en la medida de su disponibilidad. La Organización mantiene el objetivo de establecer



un grado de exactitud y actualidad máximos. Si se verificara algún error se corrige de inmediato.

Sin embargo, La Organización no puede garantizar y no asume responsabilidad alguna en cuanto a la exactitud, integridad y actualidad de la información suministrada por las/los autores en sus trabajos. La Organización no aceptará responsabilidades por errores u omisiones en los contenidos de esta página y rechaza la responsabilidad de cualquier naturaleza derivada del uso de la información contenida en la página web.

La Organización podrá modificar el contenido de la página web en cualquier momento sin previo aviso. Los hipervínculos páginas web o fuentes externas se proporcionan únicamente a efectos de conveniencia. La Organización no asume la responsabilidad, ni necesariamente aprueba los contenidos de las páginas web externas.

Fue nuestro objetivo el minimizar las molestias por causas técnicas. Sin embargo, no podemos garantizar que nuestro servicio no quede interrumpido o afectado de cualquier otra forma por problemas técnicos. La Organización no acepta responsabilidad con respecto a estos problemas derivados del uso de esta página u otras páginas externas vinculadas. Esta exención no pretende contravenir ninguno de los requisitos contemplados en la ley, ni la exclusión de responsabilidades por acciones vinculantes bajo esta ley.

Protección de datos

Todos los datos que se nos ha proporcionado quedan bajo la protección de La Organización, con el compromiso de no facilitar los mismos a terceros. Los datos se utilizarán únicamente con fines administrativos de certificación de identidad y para realizar comunicaciones.

Las ponencias, aceptadas son propiedad de sus autores, autorizando al Congreso para incluirlas y compilar su publicación en la Web, debidamente identificadas como material de congreso, siendo estos los únicos derechos sobre las ponencias presentadas.

Seminarios Virtuales

Durante la realización del Congreso, de igual manera que en los Congresos anteriores se realizaron un conjunto de Seminarios Web sobre las temáticas propuestas. Los mismos incluyeron experiencias de diseño, presentación de trabajos seleccionados, presentación de productos, etc.

Los accesos a estos eventos fueron para aquellos inscriptos al Congreso sin restricción y de manera gratuita. Para cada caso el asistente debió registrarse previamente en virtud de la existencia de un límite de personas conectadas con la posibilidad de participar activamente con preguntas y opiniones con el/los disertantes.

En el caso de haberse cubierto el cupo o no poder haber seguido la disertación en línea se dispuso de las exposiciones en una plataforma de streaming aunque no en forma sincrónica.



El asistente pudo seguir este evento desde cualquier lugar del mundo a través de un equipo con acceso a internet, pudiendo participar activamente del mismo a través de consultas y comunicación directa con el orador.

Seminarios realizados durante el Congreso

Además del espacio de debates sobre las ponencias del Congreso y para disponer de un elemento adicional de difusión y conocimiento sobre las temáticas propias del Congreso se dispuso de un sistema de generación de Seminarios Web que se utilizó diariamente.

Los expositores realizaron las mismas desde diferentes puntos geográficos tales como las ciudades de Buenos Aires, Santa Fe, Paraná, Bahía Blanca, Bariloche, Neuquén, Córdoba, San Juan, Madrid (España) y Lund (Suecia)

Los seminarios que se dictaron fueron los siguientes:

Lunes 30/09/24 10:00 hs. (GMT -3)

RISC-V en Microcontroladores

Por Martín Ribelotta

Lunes 30/09/24 12:00 hs. (GMT -3)

¿El microcontrolador elegido cuenta con recursos suficientes para una aplicación determinada?

Por Juan Manuel Cruz

Lunes 30/09/24 14:00 hs. (GMT -3)

Teoría y Aplicación de la Radio Definida por Software (SDR)

Por Carlos Zerbini y Guillermo Riva

Lunes 30/09/24 15:50 hs. (GMT -3)

Principios Básicos de Soldadura

Por Sergio Guberman

Lunes 30/09/24 17:45 hs. (GMT -3)

Tecnología SMT, Principios Básicos y Retrabajo

Por Sergio Guberman

Lunes 30/09/24 19:10 hs. (GMT -3)

Aprender haciendo: Una experiencia con la Metodología en un Curso Virtual

Por Ariel Lutemberg, Pablo Gomez y Eduardo Filomena

Lunes 30/09/24 20:15 hs. (GMT -3)

ARMando el rompecabezas

Por Sergio Caprile



Martes 01/10/24 10:00 hs. (GMT -3)

Sistema Operativo para IoT RT-Threads
Por Martín Ribelotta

Martes 01/10/24 12:00 hs. (GMT -3)

Internet de las Cosas en la era de la Inteligencia Artificial y la Edge Computing
Por Marcelo Romeo

Martes 01/10/24 14:00 hs. (GMT -3)

Criptografía y seguridad en sistemas embebidos
Por Jaime Jorge

Martes 01/10/24 16:00 hs. (GMT -3)

uModelFactory, Software didáctico profesional para el desarrollo de máquinas de estados
Por Marcelo Trujillo y Nahuel Gonzalez

Martes 01/10/24 18:00 hs. (GMT -3)

Tecnología BGA, Principios Básicos y Retrabajo
Por Sergio Guberman

Martes 01/10/24 20:00 hs. (GMT -3)

Procesadores ARM Cortex-M: DECLASSIFIED
Por Sergio Caprile

Miércoles 02/10/24 10:00 hs. (GMT -3)

Explotando el potencial de Raspberry Pi con cargas Intensas en el desarrollo de aplicaciones HMI para IoT
Por Daniel Avale

Miércoles 02/10/24 12:00 hs. (GMT -3)

Innovación en Movimiento Humano: Desarrollo de Productos de Vanguardia en Robótica para Rehabilitación desde la óptica de una PYME Tecnológica Española
Por Elkyn Belalcazar

Miércoles 02/10/24 14:00 hs. (GMT -3)

Diseño de Sistemas Digitales usando Sistemas en Chip Programmables (SoPC)
Por Cristian Sisterna

Miércoles 02/10/24 16:00 hs. (GMT -3)

Arquitectura de procesadores para DSP: optimización y métricas de rendimiento
Por Daniel Rosso

Miércoles 02/10/24 18:00 hs. (GMT -3)



Microcontroladores al Límite: Un Viaje con Boron y Photon2 de Particle.io Por Alejandro Cabral

Miércoles 02/10/24 20:00 hs. (GMT -3)

Comunicación de aplicaciones mediante TCP/IP utilizando microcontroladores y Mongoose

Por Sergio Caprile

Jueves 03/10/24 10:00 hs. (GMT -3)

De LoRa a LoRaWAN

Por Edwin Barragán y Emilio Bascary

Jueves 03/10/24 12:00 hs. (GMT -3)

Direcciones de la Tecnología de Comunicaciones Ópticas

Por Ariel Pola

Jueves 03/10/24 14:00 hs. (GMT -3)

Análisis de Costos para Determinación de Honorarios

Por Carlos Centeno

Jueves 03/10/24 16:00 hs. (GMT -3)

Introducción a los Sistemas Operativos en Tiempo Real – RTOS

Por Carlos Centeno

Jueves 03/10/24 18:00 hs. (GMT -3)

Diseño de Sistemas Embebidos aplicados al Registro de Biopotenciales

Por Eduardo Filomena

La distribución temporal propuesta y utilizada fue la siguiente

Horario	LUNES 30/9	Horario	MARTES 01/10
10:00 hs	RISC-V en Microcontroladores Por Martin Ribelotta	10:00 hs	Sistema Operativo para IoT RT-Threads Por Martin Ribelotta
12:00 hs	¿El microcontrolador elegido cuenta con recursos suficientes para una aplicación determinada? Por Juan Manuel Cruz	12:00 hs	Internet de las Cosas en la era de la Inteligencia Artificial y la Edge Computing Por Marcelo Romeo
14:00 hs	Principios de la radio definida por software (SDR) Por Carlos Zerbini y Guillermo Riva	14:00 hs	Criptografía y seguridad en microcontroladores Por Jaime Jorge
15:50 hs	Principios Básicos de Soldadura Por Sergio Guberman	16:00 hs	uModelFactory, Software didáctico profesional para el desarrollo de máquinas de estados Por Marcelo Trujillo y Nahuel Gonzalez
17:45 hs	Tecnología SMT, Principios Básicos y Retrabajo Por Sergio Guberman	18:00 hs	Tecnología BGA, Principios Básicos y Retrabajo Por Sergio Guberman
19:10 hs	Aprender haciendo: Una experiencia con la Metodología en un Curso Virtual Por Ariel Lutemberg, Pablo Gomez y Eduardo Filomena		
20:15 hs	ARMando el rompecabezas Por Sergio Caprile	20:00 hs	Procesadores ARM Cortex-M: DECLASSIFIED Por Sergio Caprile

Horario	MIÉRCOLES 02/10	Horario	JUEVES 03/10
10:00 hs	Explotando el potencial de Raspberry Pi con cargas intensas en el desarrollo de aplicaciones HMI para IoT Por Daniel Avale	10:00 hs	De LoRa a LoRaWAN Por Edwin Barragán y Emilio Bascary
12:00 hs	Innovación en Movimiento Humano: Desarrollo de Productos de Vanguardia en Robótica para Rehabilitación desde la óptica de una PYME Tecnológica Española Por Elkyn Belalcázar	12:00 hs	Direcciones de la Tecnología de Comunicaciones Ópticas Por Ariel Pola
14:00 hs	Diseño de Sistemas Digitales usando Sistemas en Chip Programmables (SoPC) Por Cristian Sisterna	14:00 hs	Análisis de Costos para Determinación de Honorarios Por Carlos Centeno
16:00 hs	Arquitectura de procesadores para DSP: optimización y métricas de rendimiento Por Daniel Rosso y Facundo Larosa	16:00 hs	Introducción a los Sistemas Operativos en Tiempo Real – RTOS Por Carlos Centeno
18:00 hs	Microcontroladores al Limite: Un Viaje con Boron y Photon2 de Particle.io Por Alejandro Cabral	18 hs	Diseño de Sistemas Embebidos aplicados al Registro de Biopotenciales Por Eduardo Filomena
20:00 hs	Comunicación de aplicaciones mediante TCP/IP utilizando microcontroladores y Mongoose (blue's) Por Sergio Caprile		

Puede observarse en el Cronograma lo apretado en el tiempo para la realización de las exposiciones que exigió de una precisa coordinación de los organizadores con cada uno de los expositores, no sólo participando en cada charla como anfitriones y auxiliares sino también coordinando las fases de prueba y preparación para cada caso. Considerando que fueron 24 exposiciones en un promedio de 45 horas en cuatro días (12 horas diarias).

Los seminarios dictados contaron con un promedio de asistencia de 231 personas. Se informa además que este sistema utilizado, plataforma Zoom, permite la emisión y acceso hasta un máximo de 300 personas en forma concurrente.

El sistema suministra a los organizadores información sobre la cantidad de asistentes, el grado de atención al seminario, la cantidad de minutos efectivamente conectados, las preguntas que se realizaron y quien la realizó, los tiempos de ingreso y egreso de cada participante entre otras informaciones de estadística propias de cada Seminario Web. Por otro lado, durante la conferencia los asistentes pudieron realizar preguntas al orador en forma escrita o a través del micrófono de su equipo informático o de comunicación.

Estos seminarios si bien en primera instancia se programaron para los asistentes al Congreso, se tomó determinación luego, que el acceso a los mismos podía ser libre invitándose a alumnos y docentes de nuestra facultad y cualquier otra institución

Tenemos conocimiento que en diferentes instituciones se replicaron en forma simultánea los seminarios para que puedan ser vistos por sus propias comunidades académicas.

Costos

La participación fue totalmente gratuita y abierta al público. Ponentes y congresistas asistentes sólo debieron realizar su inscripción vía web como tales, previo al inicio del evento.

El Cronograma

El cronograma fue respetado, cumpliendo con las fechas correspondientes y sólo permitiendo una extensión temporal en la recepción de ponencias o sus correcciones a las observaciones propuestas

Fecha	Actividad	Observaciones
Hasta el 02 de Setiembre de 2024	Recepción de ponencias	
Notificación de aceptación	Hasta 20 días después de recibida	El comité revisor estará integrado por personalidades de prestigio en el ámbito afin a cada línea temática.
30 de Setiembre de 2024	Cierre de inscripciones para Ponentes, Congresistas y Asistentes Libres.	
30 de Setiembre de 2024	Evento de inauguración Inicio del Congreso	
30 de Setiembre de 2024	Apertura de foros de debate y publicación simultánea de ponencias	
30 de Setiembre al 03 de Octubre de 2024	Participación en foros de debate y realización de Seminarios Web	
03 de Octubre de 2024	Fin del Congreso	
03 al 17 de Octubre de 2024	Período de tiempo con acceso disponible al espacio del Congreso para permitir continuidad de comunicación entre Ponentes y Asistentes	
21 de Octubre de 2024	Disponibilidad de Certificados para Asistentes y Ponentes	

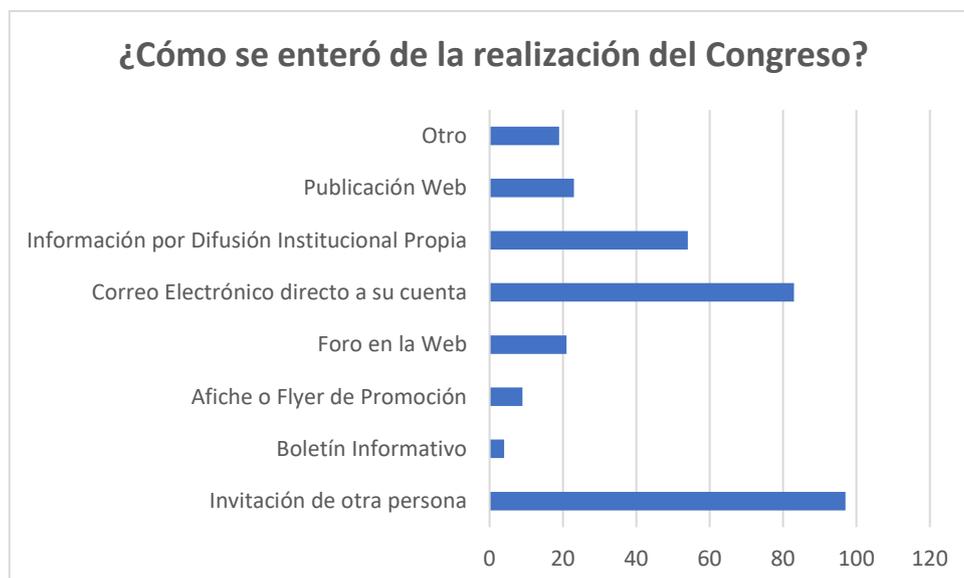
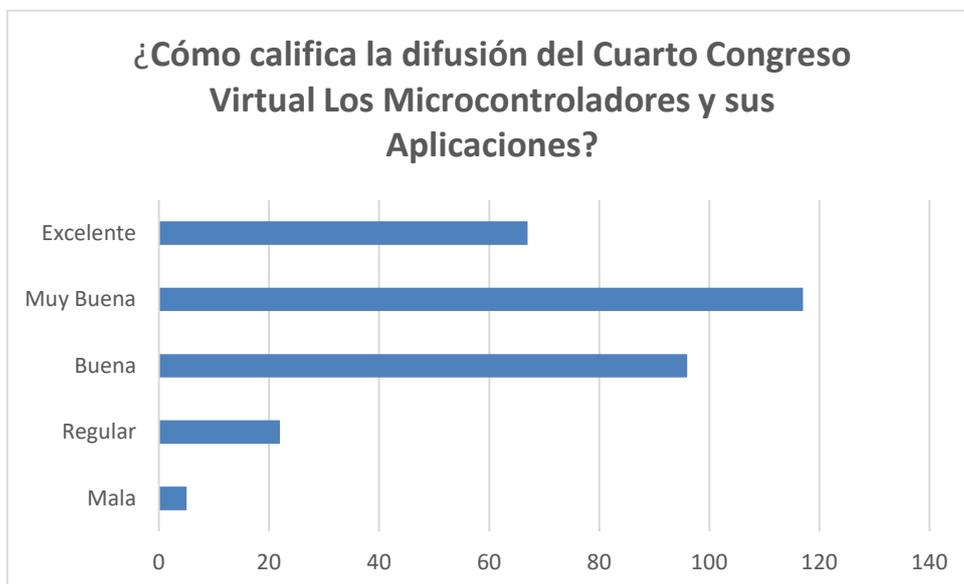
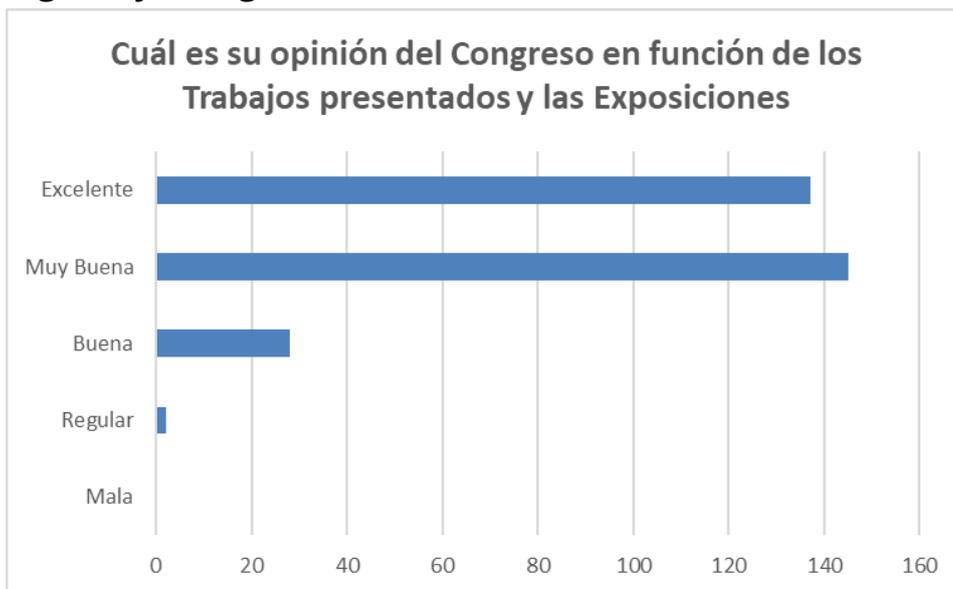
La Encuesta Final

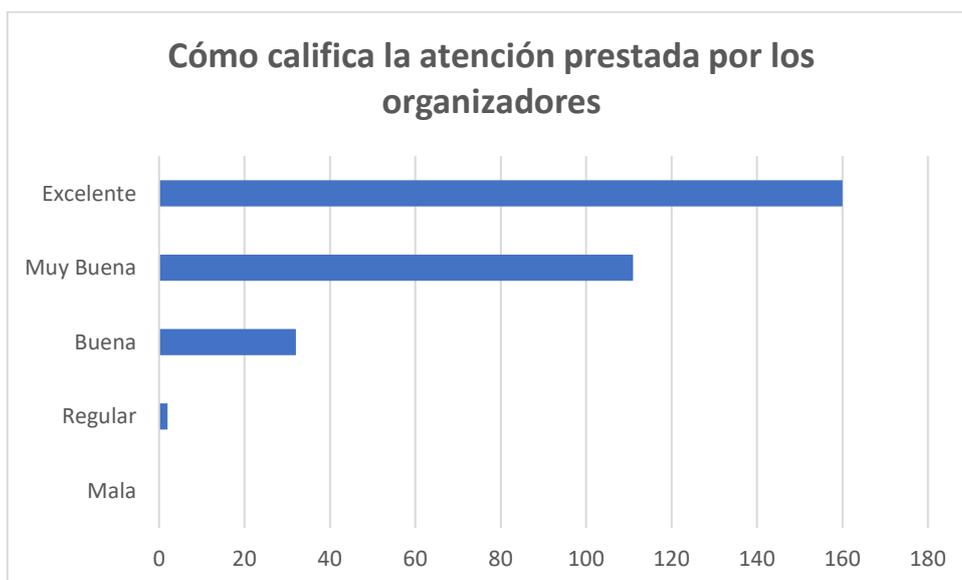
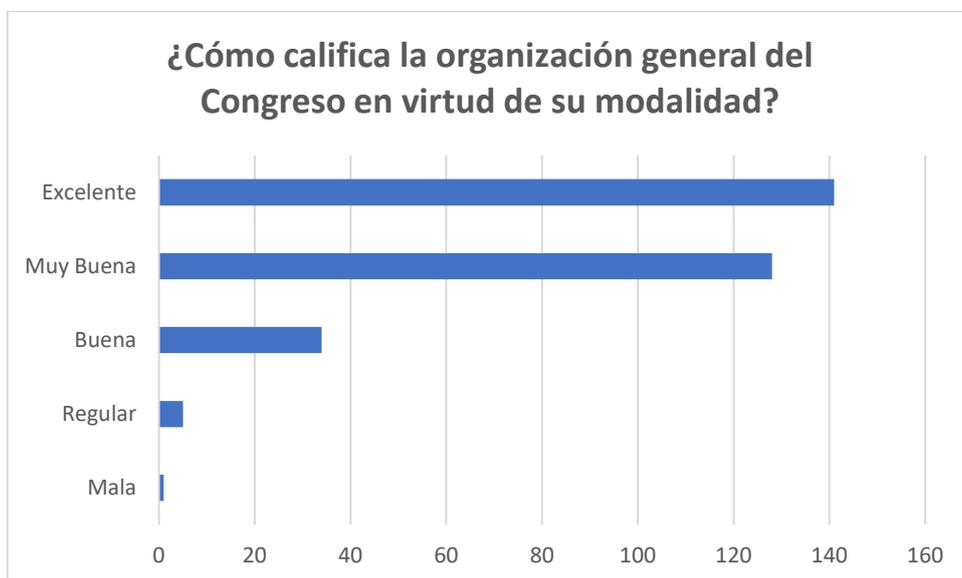
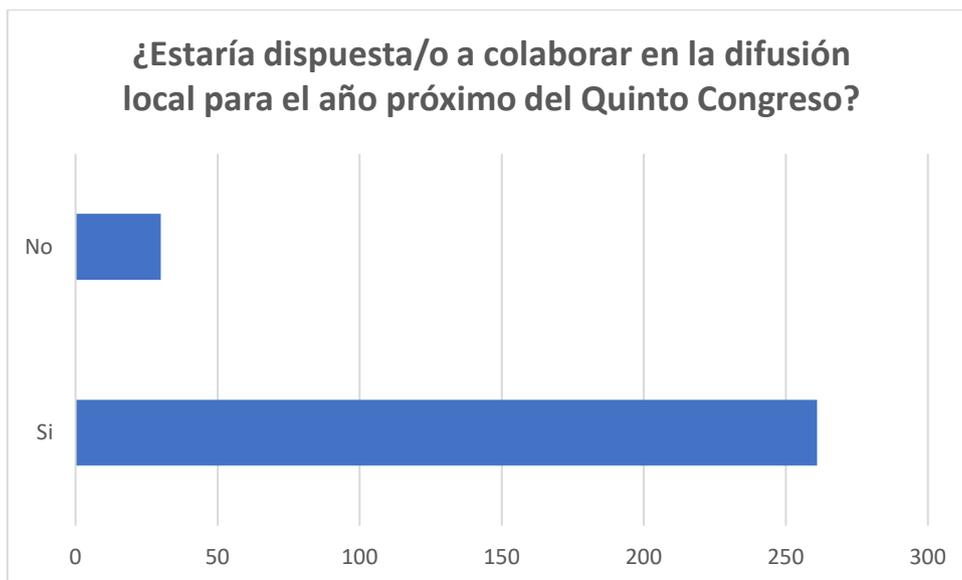
Sobre las valoraciones de los asistentes al Congreso

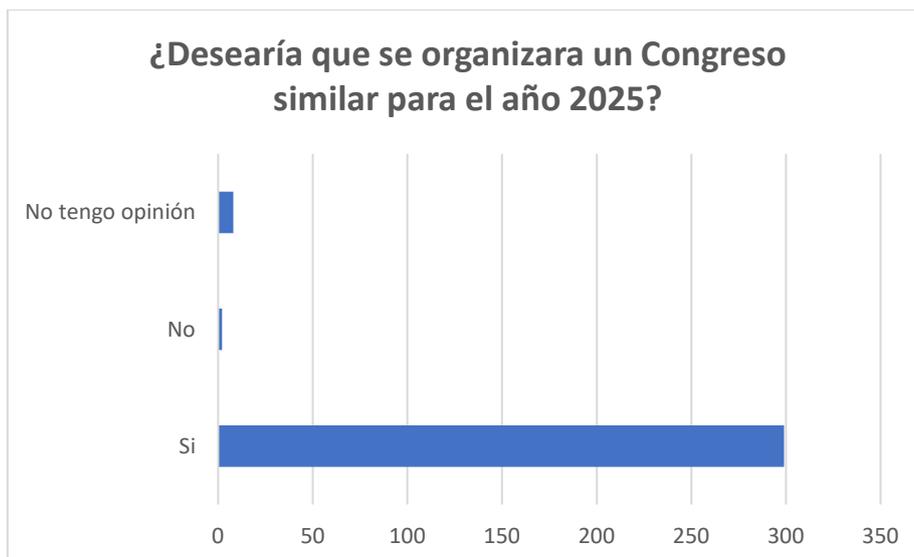
La visión y percepción de los asistentes sobre el Congreso fue recabada en los foros pero además se realizó una encuesta para detectar el grado de satisfacción respecto de la atención brindada, la calidad del sistema, las ponencias, seminarios y demás. Las estadísticas y los resultados arrojados son los siguientes:



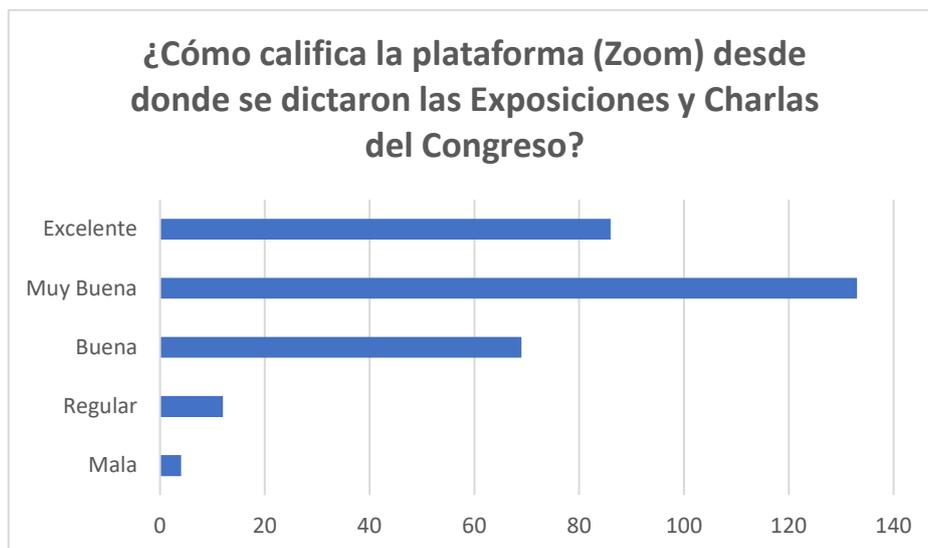
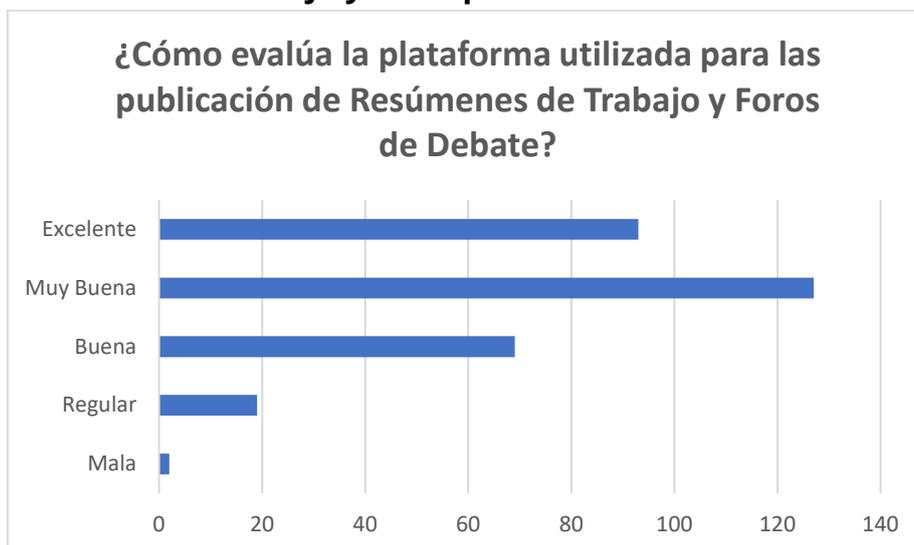
Sobre el Congreso y la Organización

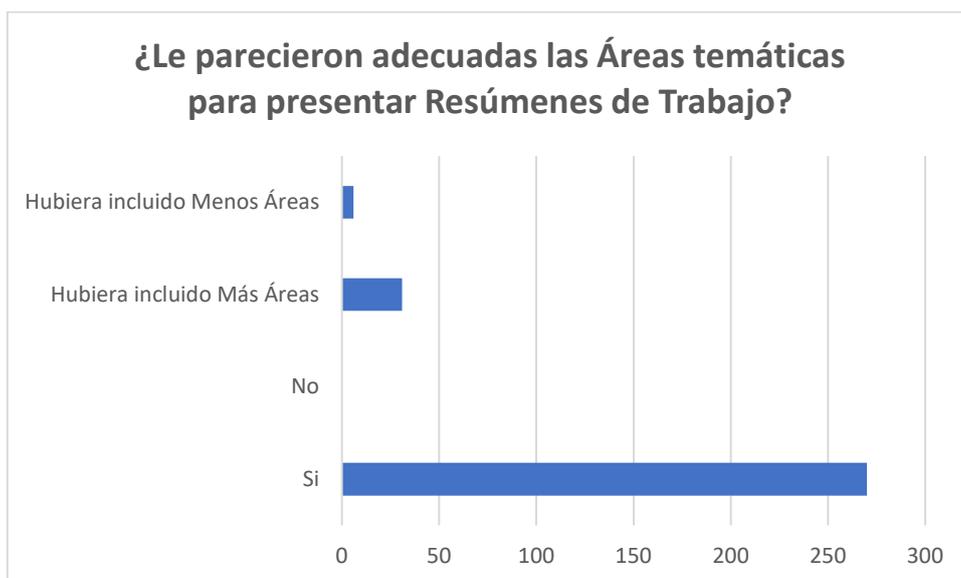
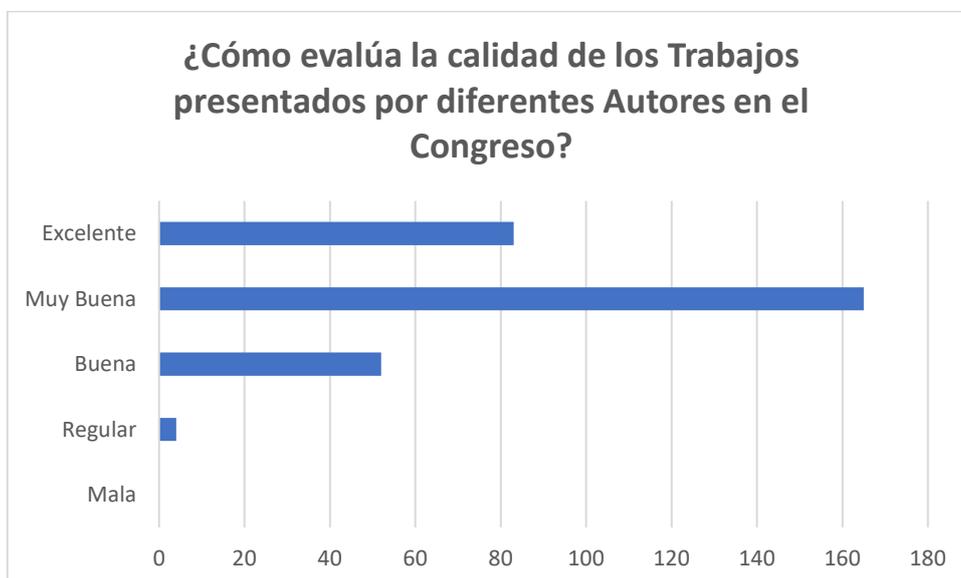
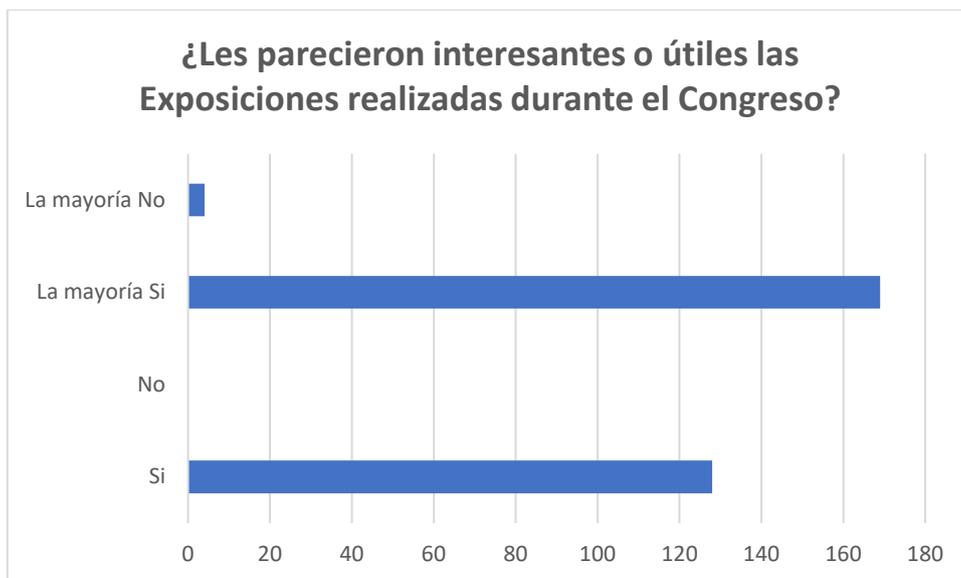


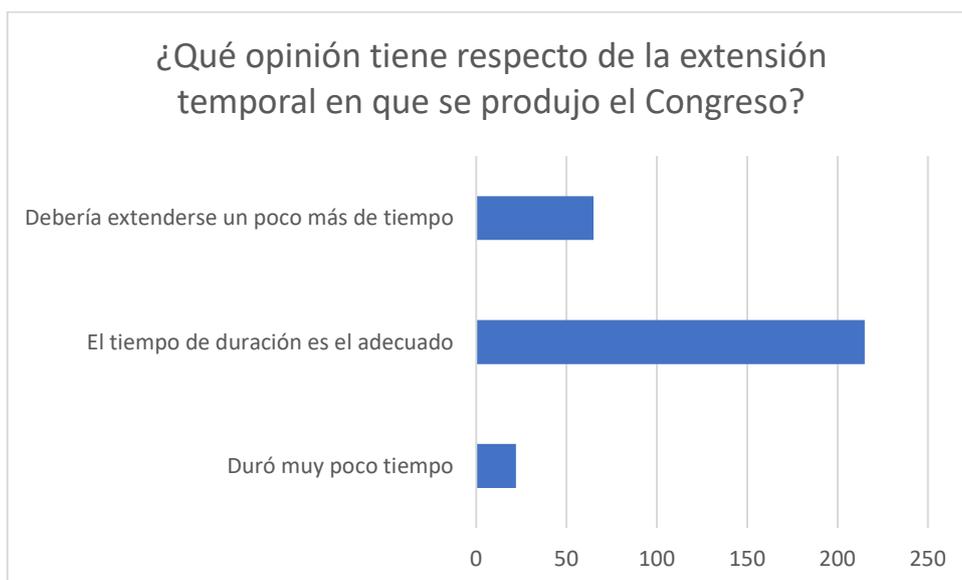
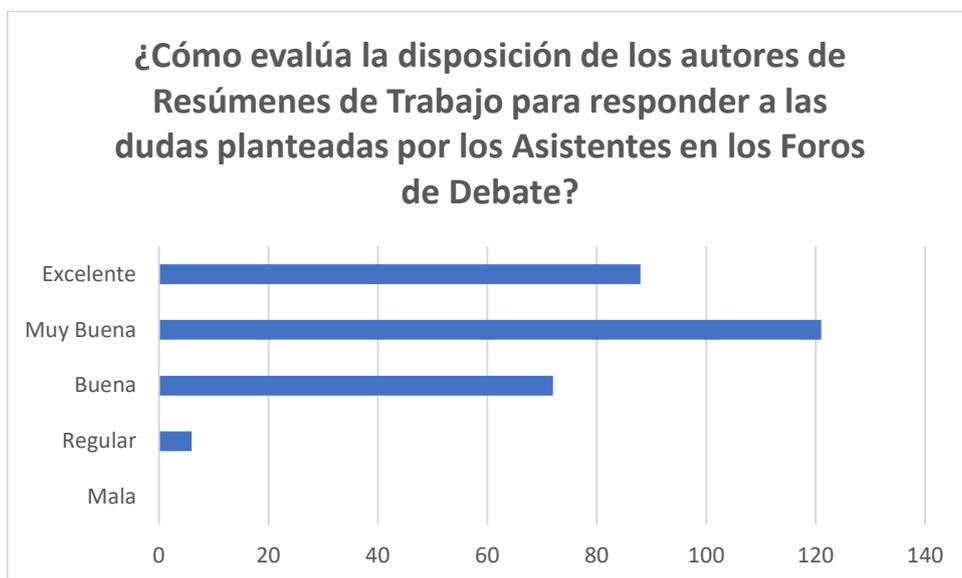
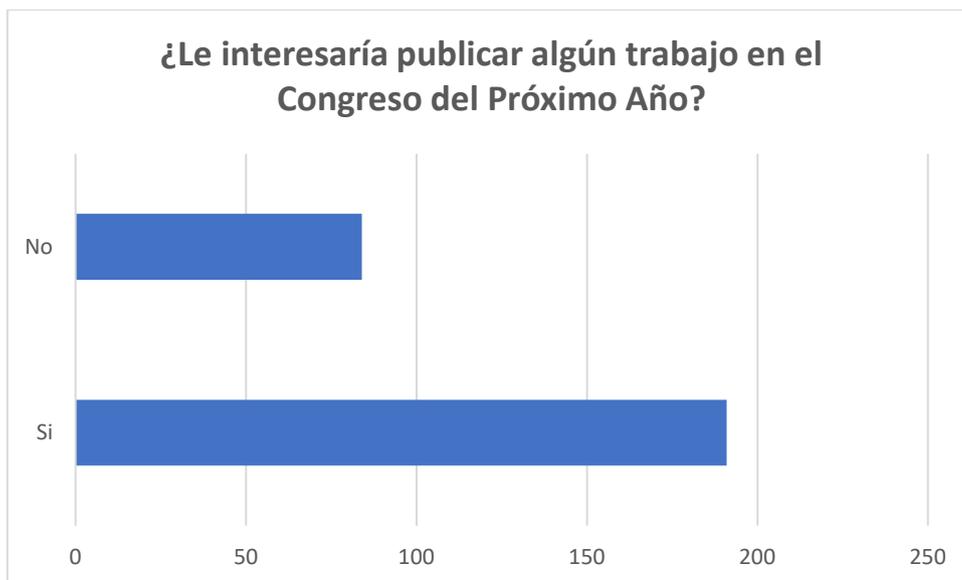




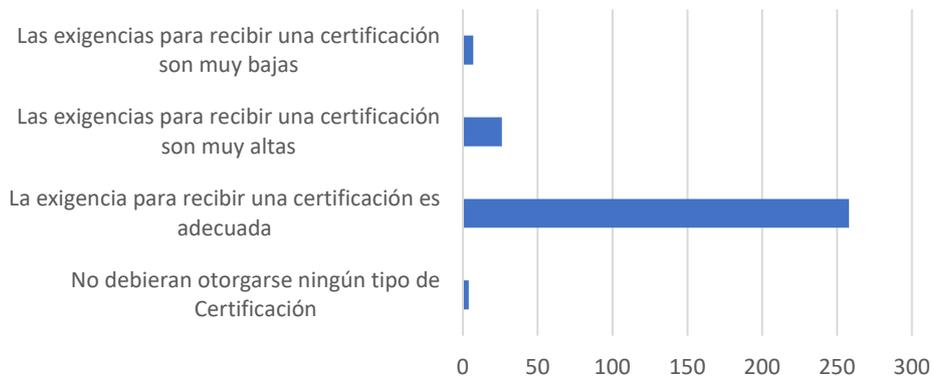
Sobre los Resúmenes de Trabajo y las Exposiciones



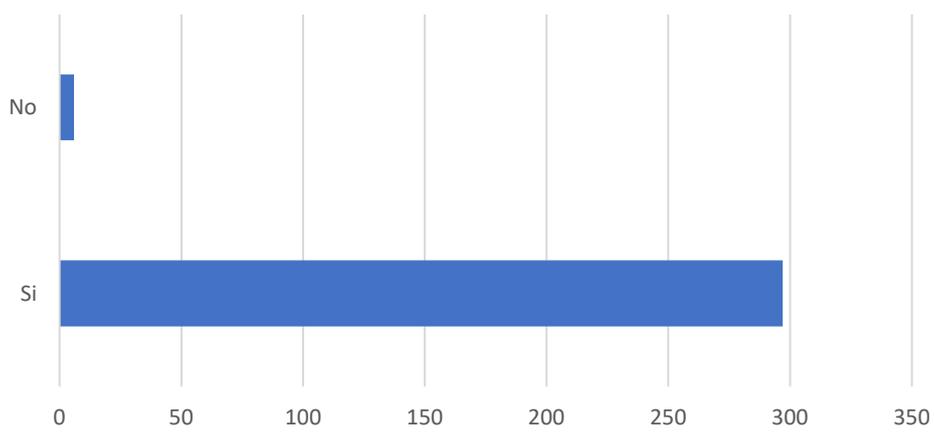




¿Qué opinión tiene respecto de las exigencias del Congreso para otorgar Certificaciones de Asistentes y Autores?



¿Le parece correcto que el Congreso sea Libre y Gratuito?



Afiche

El afiche utilizado se presenta sin logos de instituciones que avalan el Congreso.

Siendo este afiche, junto con la página del sitio Web elementos que se utilizarán para la difusión del evento.



4^{to} CONGRESO VIRTUAL DE MICROCONTROLADORES Y SUS APLICACIONES

PRESENTACIÓN CONVOCATORIA OBJETIVOS TEMÁTICAS CONSULTAS **INSCRIPCIÓN**

PARANÁ
INICIA
30
SEPTIEMBRE
2024

Las nuevas tendencias tecnológicas.

ELECTRÓNICA
UTN
PARANÁ

30 de septiembre de 2024 al 3 de octubre 2024

Sin arancel.
Con inscripción previa.

Virtual desde:
UTN Facultad Regional Paraná

Seminarios WEB en vivo



Resumen de Congresos anteriores

Primer Congreso Virtual
Los Microcontroladores y sus Aplicaciones
14 al 30 de setiembre de 2009

Publique en este Congreso sus Aplicaciones realizadas con Microcontroladores
Participe del Congreso que unira a Universidades, Empresas y Desarrolladores Independientes

Áreas Temáticas

Conectividad
Transporte
Hogar
Industria
Educación
Medio Ambiente

Aplicaciones

Informes e Inscripciones
www.areacapacitacion.com.ar
e-mail: congreso.microcontroladores@gmail.com

Organiza:
Cátedras de Técnicas Digitales
FRP UTN

Logos: MICROCHIP, freescale semiconductor, AMEL, RABBIT, ARM

Logos: UTN, FICH - UNL, Facultad de Ciencia y Tecnología Universidad Autónoma de Entre Ríos

Primer Congreso Virtual de Microcontroladores y sus Aplicaciones

Fecha de inicio:

14 de setiembre de 2009

Duración: Diecisiete (17) días

Asistentes:

Ochocientos cuarenta (840)

Número de países de origen:

Todo Latinoamérica y España

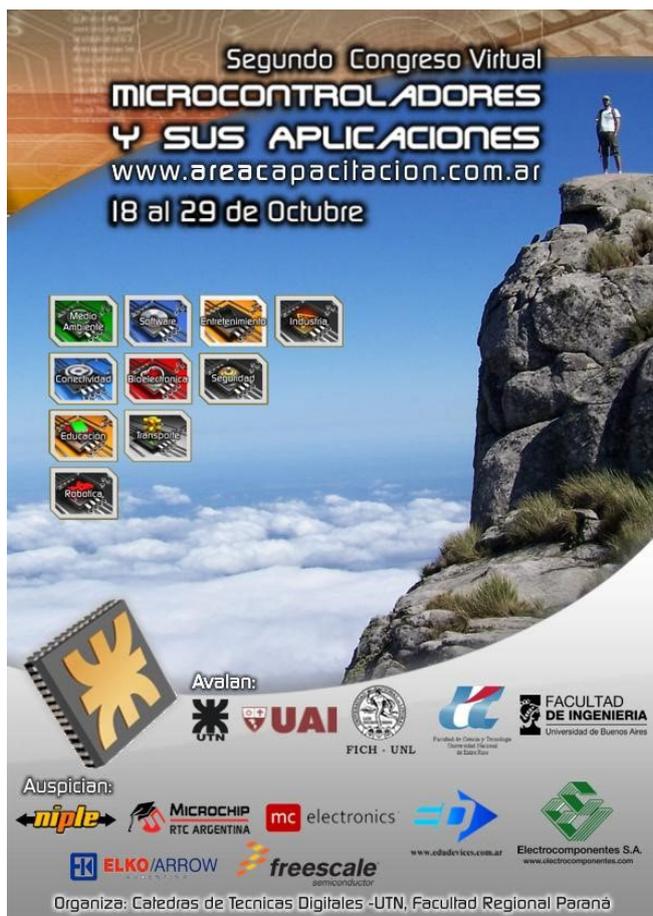
Seminarios Web: Trece (13)

Ponencias: Sesenta y cinco (65) **ISBN:**

978-987-25360-4-6.

Avales institucionales:

- Consejo Superior de la Universidad Tecnológica Nacional.
- Consejo Académico de la Facultad Regional Paraná.
- Consejo Académico de la Facultad de Ingeniería y Ciencias Hídricas de la Universidad Nacional del Litoral.
- Universidad Abierta Interamericana.
- Decanato de la Facultad de Ciencia y Tecnología de la Universidad Autónoma de Entre Ríos.



Segundo Congreso Virtual de Microcontroladores y sus Aplicaciones

Fecha de inicio:

18 de octubre de 2010

Duración: Doce (12) días

Asistentes:

Mil novecientos cuarenta y siete
(1947)

Número de países de origen:

Todo Latinoamérica y España

Seminarios Web: Veintisiete (27)

Ponencias: Cuarenta y dos (42)

ISBN: 978-987-25855-5-6

Avales institucionales:

- Consejo Superior de la Universidad Tecnológica Nacional.
- Consejo Académico de la Facultad Regional Paraná.
- Consejo Académico de la Facultad de Ingeniería y Ciencias Hídricas de la Universidad Nacional del Litoral.
- Universidad Abierta Interamericana.
- Decanato de la Facultad de Ciencia y Tecnología de la Universidad Autónoma de Entre Ríos.
- Rectorado Universidad Autónoma de Entre Ríos
- Facultad de Ingeniería de la Universidad de Buenos Aires
- Facultad de Ingeniería de la Universidad Nacional de Entre Ríos
- Facultad de Ingeniería de la Universidad de Belgrano
- Colegio Oficial de Ingenieros Técnicos Industriales de La Coruña España (COETICOR)
- Instituto Universitario de Tecnología de Valencia - Venezuela



Tercer Congreso Virtual de Microcontroladores y sus Aplicaciones

Fecha de inicio:

26 de Mayo de 2014

Duración: Seis (6) días

Asistentes:

Mil seiscientos treinta y dos (1632)

Número de países de origen:

Todo Latinoamérica y España

Seminarios Web: Veinte (20)

Ponencias: Cincuenta y dos (52)

Avales institucionales:

- Consejo Superior de la Universidad Tecnológica Nacional.
- Consejo Académico de la Facultad Regional Paraná.
- Consejo Académico de la Facultad de Ingeniería y Ciencias Hídricas de la Universidad Nacional del Litoral.
- Universidad Abierta Interamericana.
- Decanato de la Facultad de Ciencia y Tecnología de la Universidad Autónoma de Entre Ríos.
- Rectorado Universidad Autónoma de Entre Ríos
- Facultad de Ingeniería de la Universidad de Buenos Aires
- Facultad de Ingeniería de la Universidad Nacional de Entre Ríos
- Facultad de Ingeniería de la Universidad de Belgrano
- Colegio Oficial de Ingenieros Técnicos Industriales de La Coruña España (COETICOR)
- Municipalidad de la Ciudad de Paraná

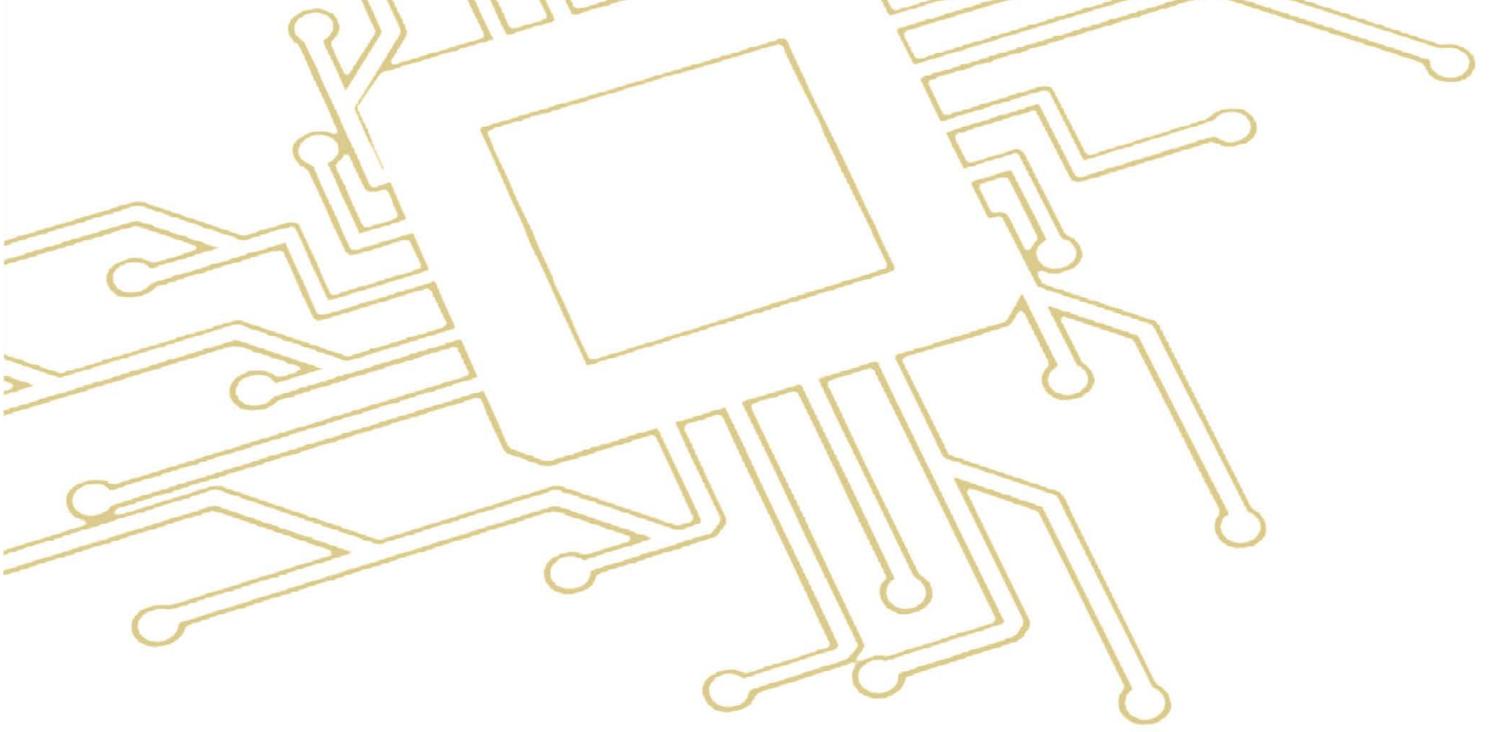
El Yaguareté



El Yaguareté, custodio de nuestras barrancas en el río Paraná

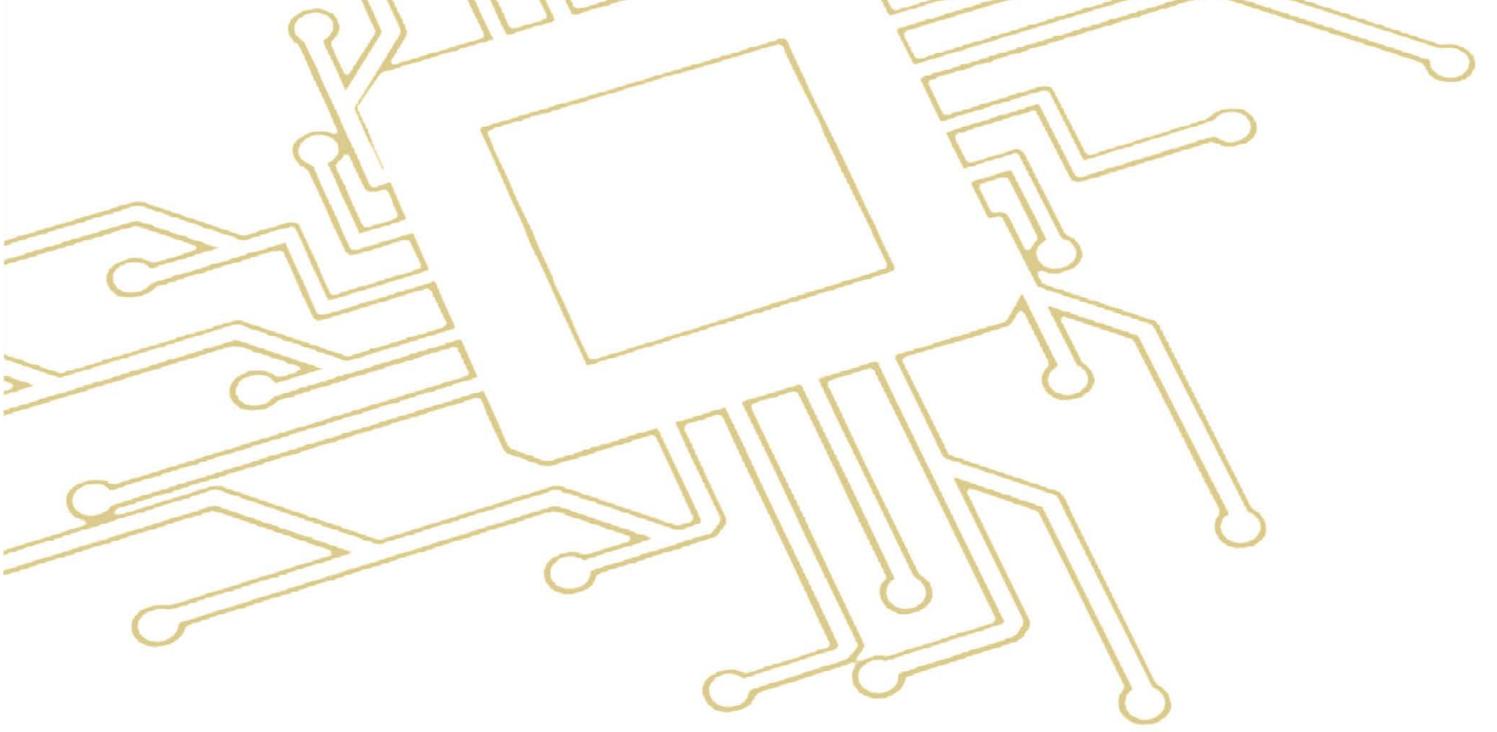
Ubicado sobre la barranca del Parque Urquiza que fue diseñado por el paisajista francés Charles Thays

La obra Yaguareté fue realizada en 1935. Su autor, Emilio Jacinto Sarniguet. Es un elemento emblemático en nuestro Parque Urquiza y en nuestra Paraná, Ciudad Capital



Trabajos





Eje Temático

Aplicaciones en Bioelectrónica





Sistema Electrónico de Dosificación de Medicamentos para Adultos Mayores Basado en Arduino

Edwin Sánchez Uriza

Ilber Ruge Ruge

Ingrid Ortíz Álvarez

Universidad Pedagógica y
Tecnológica de Colombia

Universidad Pedagógica y
Tecnológica de Colombia

Universidad Pedagógica y
Tecnológica de Colombia

Escuela de Ingeniería Electrónica,
grupo de investigación I2E

Escuela de Ingeniería Electrónica,
grupo de investigación I2E

Escuela de Ingeniería Electrónica,
grupo de investigación I2E

Tunja, Colombia

Tunja, Colombia

Tunja, Colombia

edwinjavier.sanchez@uptc.edu.co

ilber.ruge@uptc.edu.co

ingrid.ortiz05@uptc.edu.co

Resumen— La propuesta surge de la consideración de que la población de adultos mayores está aumentando rápidamente en la actualidad, lo que significa que deben tomar varios medicamentos al día, que son obligatorios e imprescindibles para mantener su salud estable. El sistema electrónico satisface esta necesidad teniendo en cuenta tres datos: la hora de toma de la medicación, la dosis que se debe tomar y el nombre del medicamento. El sistema contiene seis cajones y dos subdivisiones (teniendo en cuenta que el pastillero será para dos pacientes). Dispone de elementos de entrada como un sensor de entrada, almacenamiento de información y un reloj en Tiempo Real RTC para la administración de los horarios de toma de medicación, una alarma sonora, pantalla LCD entre otros. Como procesador principal se utiliza un Arduino MEGA basado en el microcontrolador ATMEGA2560.

Palabras clave: Dosificación de medicamentos, adulto mayor, Arduino MEGA.

I. INTRODUCCIÓN

En la actualidad los sistemas de asistencia electrónica basados en Arduino para administrar medicamentos a personas mayores utilizan tecnología IoT que incorpora nuevos terminales de dispensación de fármacos que mejoran el procedimiento de ingesta y la automatización de la dosificación. Estos sistemas suelen integrar placas Arduino con otras piezas como sensores, actuadores, una

cámara y módulos de comunicación para ofrecer sistemas eficaces de administración de medicamentos. Por ejemplo, los sistemas explicados en [1] y [2] abarcan la administración de fármacos, el reconocimiento facial y la dispensación de pastillas utilizando Arduino y Raspberry Pi para administrar el medicamento en el momento y la cantidad adecuados.

Además, soluciones como la propuesta en [3] que emplea Arduino y NodeMCU, que implican detalles de dosificación, sensores IR para captar al usuario y GSM para monitorizar el uso.

Por otro lado, el dispensador de píldoras inteligente y vaso inteligente alimentados por Arduino Nano RP2040 [4], ofrece funciones como dispensación precisa, verificación del consumo e interfaces de gestión de la medicación fáciles de usar. Además, sistemas como el Dispensador Automático de Medicamentos con Raspberry Pi y reconocimiento facial [4] y el botiquín inteligente con Arduino y módulo RTC [5] aseguran el consumo puntual de la medicación mediante la automatización de la dispensación, el envío de notificaciones en tiempo real y la monitorización remota.

En los últimos años los sistemas basados en Arduino se han aplicado en la creación de muchos dispositivos para la administración de medicamentos dentro del campo médico y se basan principalmente en componentes que incluyen RTC, pantalla LCD, zumbadores,



motores, drives especializados, módulos de comunicaciones, circuitos moduladores de voz entre otros para desarrollar una sólida solución de recordatorio de medicación que se incorpora en dispositivos médicos [6][7][8]. Estos dispositivos emplean la placa Arduino Uno como microcontrolador para proporcionar un control eficiente de la administración de medicamentos, el seguimiento de la temperatura corporal y el cumplimiento de los horarios de dosificación.

La principal contribución es la creación de un sistema electrónico que recuerda la toma de medicamentos de bajo coste y fácil de usar para personas mayores, con llamadas desde teléfonos móviles y algunas funciones adicionales de seguimiento de la salud. Además, hemos creado una aplicación que permite a los usuarios introducir el calendario de medicación del kit sugerido. Los errores de medicación sólo se producirán cuando un médico prescriba un medicamento equivocado; por lo demás, nuestra técnica sugerida elimina por completo la posibilidad de que se produzcan errores de medicación. Dado que no existe la posibilidad de abrir la cámara de medicamentos incorrecta, sólo se desbloquea una cámara de medicamentos adecuada a la vez según la entrada de la prescripción del médico.

Las pruebas prácticas del sistema se realizaron principalmente en entornos domésticos; sin embargo, puede extenderse a la asistencia a médicos y enfermeros en entornos hospitalarios. Cuando un paciente de urgencias deba permanecer en el hospital durante varios días y necesite que se le administre la medicación con frecuencia y a su debido tiempo, el sistema propuesto puede desplegarse para el paciente y, cada vez que se le indique, el médico o la enfermera asignados recibirán una alarma, es decir, un recordatorio, lo que reduce el estrés de tener que seguir la agenda de un gran número de pacientes.

II. MATERIALES Y MÉTODOS

Este proyecto es de carácter cuantitativo, experimental y se basa en la implementación de un dispositivo electrónico para la dosificación de medicamentos, dirigido

especialmente a la población de adultos mayores, donde surge la necesidad de la tecnología asistida [9], promoviendo la independencia de la persona y una mejora en su calidad de vida.

El dispositivo cuenta con alarma sonora para anunciar al paciente que es momento de tomar el medicamento. También cuenta con elementos indicadores de tipo lumínico (diodos emisores de luz tipo LED) para indicar al usuario el cajón donde se aloja el medicamento de interés, y también una pantalla LCD 16x2, donde se visualiza el nombre del medicamento y la dosis respectiva.

Como es un sistema de alarma se debe contar con un reloj para programar la hora y fecha exacta de suministro del medicamento. Para esto se utiliza un calendario digital RTC (Real Time Clock o reloj de tiempo real), que lleva la cuenta de los segundos, minutos, hora y además de los días, meses y años. Este reloj tiene memoria RAM de 56 bytes, además de contar con una batería de litio que sirve para mantener la hora y fecha actualizada cuando sufre corte de energía eléctrica. El consumo de energía es bajo y la duración de batería oscila entre 5 y 8 años según datos suministrados por fabricante.

Se cuenta también con un sensor de movimiento piroeléctrico pasivo PIR, que detecta los cambios infrarrojos emitidos por cualquier objeto o persona. Su alcance es de aproximadamente 6 metros a su alrededor, y cuenta con la opción de poder ajustar el alcance de detección. Este sensor se utilizar para detener la alarma en el sistema electrónico, una vez detecte la presencia del asistente/paciente [10].

Finalmente, el cerebro del proyecto se basa en la plataforma de desarrollo Arduino, placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra donde van conectados los elementos. En el proyecto se usó el Arduino MEGA basado en microcontrolador ATMEGA 2560, es alimentado con fuente de 5Vdc y se puede alimentar directamente del puerto USB de una computadora. Para el desarrollo se

alimenta con una batería de 9Vdc.

III. DESARROLLO

El algoritmo programado en el microcontrolador Arduino se describe en el diagrama de flujo mostrado en la Figura 1. Este código se hizo en base a que existieran dos pacientes (paciente 1, paciente 2), donde cada paciente tiene que tomar tres medicamentos, tres veces al día (se repartió entre mañana, tarde y noche). El funcionamiento del sistema se describe en el diagrama:

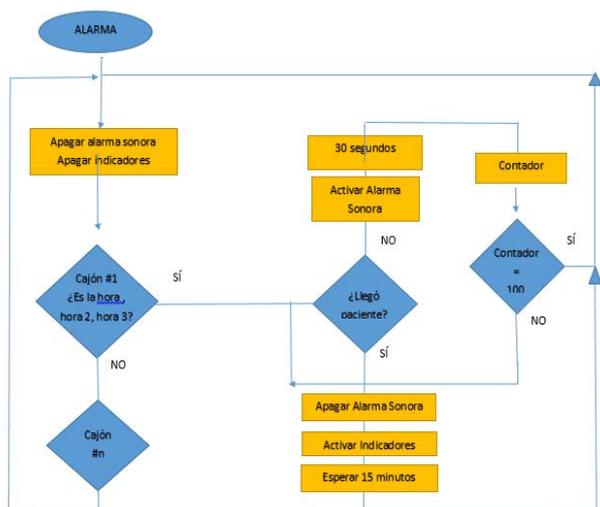


Figura 1. Diagrama de flujo de pastillero electrónico.

A. MATERIALES UTILIZADOS

Los principales componentes electrónicos utilizados para el desarrollo del dosificador:

- RTC DS1307: reloj en tiempo real.
- Sensor PIR: detección de movimiento (presencia del usuario).
- Servomotores: Para suministrar la medicación.
- Leds: Para indicar la dosificación de cada paciente.
- Buzzer o bocina: Alarma para alertar al usuario del momento de tomar la medicación.
- LCD: Visualización de la información.

A continuación, se presenta la tabla 1, la cual sintetiza el código en Arduino desarrollado:

Tabla 1. Síntesis del código desarrollado.

Sección	Descripción
Librerías Incluidas	Wire.h, RTCLib.h, LiquidCrystal_I2C.h, Servo.h - Librerías para comunicación I2C, RTC, control de servos, y LCD.
Definición de Pines	Define los pines para el sensor PIR, servos, LEDs, y buzzer, necesarios para la operación del sistema.
Objetos RTC y LCD	Creación de objetos RTC_DS1307 para el reloj en tiempo real y LiquidCrystal_I2C para la pantalla LCD.
Objetos Servo	Declaración de los objetos servo1 y servo2 para controlar la dispensación de medicación a dos pacientes.
Constantes de Horario	Define las horas y minutos para la dosificación de medicación de los pacientes 1 y 2.
Función setup()	Configura los pines, inicializa el RTC, servos, y LCD, y verifica la conexión del RTC. Muestra un mensaje inicial en la pantalla LCD.
Función loop()	Ciclo principal que monitorea el sensor PIR, verifica la hora para la medicación, y activa los LEDs, buzzer, y LCD para cada paciente.
Detección de Movimiento	Verifica si hay movimiento usando el sensor PIR, y controla los LEDs y mensajes en la pantalla LCD en consecuencia.
Suministro de Medicación	Función suministroMedicacion que mueve el servo a 90° para dispensar la medicación y luego regresa a la posición inicial.
Indicaciones de Estado	Muestra mensajes de estado ("Todo normal", "No hay usuario") en la pantalla LCD dependiendo de la presencia de movimiento detectado.

B. Uso de memoria del Código

1. Memoria Flash (ROM): El Arduino Mega 2560 tiene 256 KB de memoria flash. El tamaño del código se puede comprobar en el IDE de Arduino después de cargarlo. Las librerías y el código combinados normalmente utilizarán alrededor de 10-20 KB, dejando mucho espacio para funciones adicionales.

2. SRAM (RAM): El Mega tiene 8 KB de SRAM. Las variables usadas en el código (como la hora del RTC, las posiciones de los servos, etc.) son mínimas, normalmente usan menos de 1 KB.

3. EEPROM: el Mega tiene 4 KB de EEPROM, que no se utiliza en este código.

C. Tiempo de ejecución del ciclo:

Para calcular el tiempo de ejecución del ciclo, se estimó el tiempo que tarda cada operación en el bucle:

- Comprobación del RTC: La lectura de la hora del RTC tarda unos milisegundos.
- Comprobación del sensor PIR: La lectura del sensor PIR suele tardar menos de 1 ms.
- Control de LEDs: El encendido y apagado de los LEDs es instantáneo.
- Alarma: La activación del zumbador también es instantánea.
- Pantalla LCD: La visualización de mensajes en la pantalla LCD puede tardar unos 2-5 ms dependiendo de la longitud del mensaje.
- Control del servo: Mover el servo tarda unos 20-30 ms dependiendo de la velocidad del servo.

Tiempo de ejecución del bucle:

- Lectura RTC: 5 ms
- Lectura PIR: 1 ms
- Control LED: 1 ms
- Control del buzzer: 1 ms
- Actualización LCD: 5 ms
- Servocontrol: 30 ms (si se dispensa medicación)

Suponiendo que la medicación se dispensa una vez en el bucle:

Total = 5 + 1 + 1 + 1 + 5 + 30 = 43 ms (para la dispensación)

Si no se produce ninguna dispensación Total = 5 + 1 + 1 + 1 + 5 = 13 ms

IV. RESULTADO Y DISCUSIÓN

Todos los elementos nombrados anteriormente se instalaron y organizaron en una caja de plástico, con 6 compartimentos enumerados de 1 a 6, cada cajón con sus respectivos leds, dos servomotores como mecanismos de movimiento, un sensor PIR y un RTC (Figura 2).

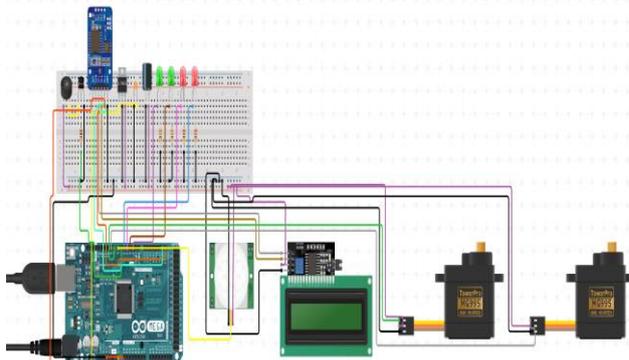


Figura 2. Diagrama electrónico de conexiones.

Adicionalmente en la parte de arriba se encuentra la pantalla y el sensor para fácil acceso y mayor eficacia, debajo de ellos, en un fondo falso se encuentra la PCB conectada al Arduino Mega que se dispuso afuera para mayor comodidad al momento de conectar y orden en todo el circuito, como se evidencia en la Figura 3.

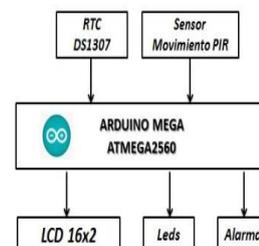


Figura 3. Pastillero electrónico.

El proyecto logró un impacto positivo en la vida de los adultos mayores que lo utilizaron, teniendo en cuenta las características de la máquina, su alcance llega hasta los 6 medicamentos y puede ser usado por dos pacientes. Su tamaño y material permite al usuario transportarlo de lugar con facilidad, además de tener una interfaz amigable con el

usuario, cumpliendo con uno de los objetivos el cual era lograr una fácil interacción persona-máquina, donde los adultos mayores no se sintieran aturdido por muchos botones o información.

Por medio de este sistema se logra cumplir con los horarios e indicaciones dadas por el personal médico y así controlar la toma de medicamentos sin problemas.

Los puntos a mejorar sería el perfeccionamiento del diseño y la inclusión de personas con discapacidades auditivas y visuales, teniendo en cuanto que se fabricó con materiales al alcance, adaptando ciertos elementos a la conveniencia del sistema de alarma, los elementos electrónicos fueron soldados manualmente y hubo ciertos problemas a la hora de hacer la PCB, sin embargo se utilizó una universal.

CONCLUSIONES

En general, las aplicaciones de sistemas basados en Arduino en dispositivos médicos ayudan a suministrar medicamentos a los pacientes y, al mismo tiempo, garantizan que el paciente los utilice correctamente, controlan las actividades de dispensación y, en general, la forma en que se trata a un paciente.

Este tipo de sistemas de dosificación electrónicos no sólo mejoran el destino del paciente acelerando el cumplimiento de la medicación prescrita, sino que también disminuyen la carga que supone la gestión de los medicamentos para los cuidadores y, en consecuencia, mejoran la calidad de vida de las personas mayores.

REFERENCIAS

- [1] Hassan, N., Gaur, A., Jadaun, Y., & Bhasney, A. (2024). IoT-Enabled Medicine Dispenser for Pills and Liquid Medication. 2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT), 905-910.
- [2] R. Achammal.S et al., "Development of Smart Automatic Drug Dispenser for Elderly and Disabled People," 2024 International Conference on Science Technology Engineering and Management (ICSTEM), Coimbatore, India, 2024, pp. 1-5, doi: 10.1109/ICSTEM61137.2024.10560887.
- [3] Souza Junior, S et al., (2023). IoT and smartphone app for elderly pill management. In Anais Estendidos do XIII Simpósio Brasileiro de Engenharia de Sistemas Computacionais, (pp. 55-58). Porto Alegre: SBC. doi:10.5753/sbesc_estendido.2023.235372
- [4] V. Peddisetti, P et al., "Smart Medication Management: Enhancing Medication Adherence with an IoT-Based Pill

- Dispenser and Smart Cup," 2024 IEEE First International Conference on Artificial Intelligence for Medicine, Health and Care (AIMHC), Laguna Hills, CA, USA, 2024, pp. 137-144, doi: 10.1109/AIMHC59811.2024.00032.
- [5] R. Kiruba, M et al., "Design and Implementation of Internet of Things - Based Smart Medicine Box for Patients," 2023 Third International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS), Gobichettipalayam, India, 2023, pp. 36-41, doi: 10.1109/ICUIS60567.2023.00014.
- [6] Prayogo, M. F. D., Aji, W. S. (2020). Design and Build Reminder for Arduino Based Drug Schedule Consumption System. Buletin Ilmiah Sarjana Teknik Elektro, 2(2), 75-82. <https://doi.org/10.12928/biste.v2i2.1031>
- [7] Rodolfo, Echandi-Pacheco. (2022). Design and prototyping of an automated didactic medicine dispenser. Journal of Technology and Innovation, Vol.9 No.24 12-18, doi: 10.35429/jti.2022.24.9.12.18
- [8] V, S., S. Bose et al "Ocular drug delivery system using open-source syringe pump". Asian Journal of Pharmaceutical and Clinical Research, vol. 11, no. 6, June 2018, pp. 152-7, doi:10.22159/ajpcr.2018.v11i6.24151
- [9] Parra Ullauri, J. M., & Valdez Solís, W. F. (2017). Diseño e implementación de un prototipo de pastillero digital para la Toma de Medicación de Pacientes Adultos Mayores, mediante la integración de nuevas tecnologías de hardware Y Software Libre. Repositorio Institucional Universidad de Cuenca: <https://dspace.ucuenca.edu.ec/handle/123456789/27377>
- [10] Torres, D. (2016) Arduino Curso Practico de Formación, Academia.edu. Disponible en: https://www.academia.edu/29198652/Arduino_Curso_Practico_de_Formacion.



Avances preliminares en el diseño y desarrollo de un implante coclear con fines didácticos mediante procesamiento embebido de señales

1st Edgardo Porral

Laboratorio de tecnología biomédica
UTN Avellaneda

Buenos Aires, Argentina
eporral@fra.utn.edu.ar

2nd Gerardo Battaglia

Laboratorio de tecnología biomédica
UTN Avellaneda

Buenos Aires, Argentina
gbattaglia@fra.utn.edu.ar

3rd Iván Alejandro Reyes

Laboratorio de tecnología biomédica
UTN Avellaneda

Buenos Aires, Argentina
ireyes@fra.utn.edu.ar

4th Martín Enriquez

Laboratorio de tecnología biomédica
UTN Avellaneda

Buenos Aires, Argentina
menriquez@fra.utn.edu.ar

5th Axel Gomez

Laboratorio de tecnología biomédica
UTN Avellaneda

Buenos Aires, Argentina
gomezaxellucas@gmail.com

Abstract—Este artículo presenta los avances realizados hasta el momento en el PID titulado "Diseño y Desarrollo de un Implante Coclear con Fines Didácticos mediante Procesamiento Embebido de Señales". Un implante coclear es un dispositivo médico que proporciona audición parcial a personas con discapacidad auditiva severa mediante la estimulación eléctrica directa en el oído interno. Este dispositivo consta de un procesador de habla externo, un enlace de transmisión de radiofrecuencia, y un receptor interno con un estimulador y matrices de electrodos que transmiten señales al oído interno. En este proyecto, adoptamos un enfoque sistémico para abordar cuestiones críticas en el diseño y desarrollo del implante coclear didáctico, garantizando la seguridad y fiabilidad de los componentes y anticipando los desafíos futuros que enfrentarán los diseñadores y usuarios de implantes cocleares. Este trabajo no solo se centra en la creación del implante coclear didáctico, sino también en utilizarlo como modelo para mejorar las técnicas de enseñanza tanto a profesionales de la salud como del área de la ingeniería biomédica. Hasta la fecha, se han completado el diseño de los filtros digitales, la definición de las especificaciones generales de los bloques constitutivos y el diseño de las tramas de envío de datos que se utilizarán en la comunicación entre los microcontroladores externo e interno.

Index Terms—Implante Coclear, Filtros Digitales, Bioelectrónica, Educación en Ingeniería Biomédica, Estimulación Eléctrica

I. INTRODUCCIÓN

La pérdida auditiva profunda bilateral afecta significativamente la calidad de vida de las personas, limitando su capacidad de comunicación e interacción social [1]. Los implantes cocleares han surgido como una solución

efectiva para este problema, permitiendo a las personas con discapacidad auditiva grave recuperar la audición y participar plenamente en la sociedad [2]. Estos dispositivos transforman las señales acústicas en impulsos eléctricos que estimulan directamente el nervio auditivo, evitando las áreas dañadas del oído interno [3]. El avance tecnológico en el campo de los implantes cocleares ha sido notable en las últimas décadas, con mejoras significativas en la miniaturización de componentes, la calidad del sonido y la comprensión del habla en entornos ruidosos [4], [5]. Sin embargo, la complejidad de estos dispositivos y los principios subyacentes a su funcionamiento pueden ser desafiantes de comprender para estudiantes e investigadores que se inician en el campo de la ingeniería biomédica y la audiología.

En este contexto, el presente trabajo se enfoca en el diseño y desarrollo de un implante coclear con fines didácticos. Este dispositivo tiene como objetivo principal simular la funcionalidad de un implante coclear real, proporcionando una herramienta educativa práctica para mejorar la comprensión de los estudiantes sobre el funcionamiento de los implantes cocleares y el procesamiento de señales acústicas. La motivación detrás de este proyecto radica en la necesidad de mejorar la formación de recursos humanos en el campo de la ingeniería biomédica y la audiología [6], así como fomentar la investigación y el desarrollo de nuevas tecnologías en el área de los implantes cocleares. Al proporcionar una herramienta de enseñanza efectiva, se espera que los estudiantes puedan adquirir una comprensión más profunda de los principios fundamentales de la bioelectrónica, los

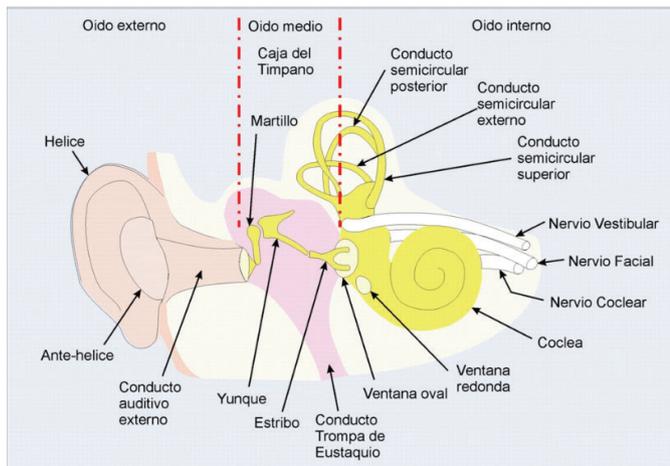


Fig. 1. Diagrama anatómico del oído humano, mostrando las tres secciones principales: oído externo, oído medio y oído interno. Se destacan las estructuras clave involucradas en la audición y el equilibrio.

microsistemas, la ingeniería biomédica, la acústica y la neurociencia, así como desarrollar habilidades prácticas en el diseño, desarrollo e implementación de dispositivos biomédicos complejos [7].

II. METODOLOGÍA

A. Marco Teórico

El diseño y desarrollo de un implante coclear con fines didácticos se basa en la comprensión profunda de la anatomía y fisiología del oído como se muestra en la Fig. 1, así como en los principios de la bioelectrónica y el procesamiento de señales. El presente trabajo se fundamenta en la teoría tonotópica, que describe la distribución de frecuencias a lo largo de la membrana basilar en la cóclea [8]–[10] como puede observarse de manera simplificada en las 2 y 3. Esta teoría establece que cada punto de la membrana basilar responde de manera óptima a una frecuencia específica, lo que permite la discriminación de diferentes tonos.

El implante coclear didáctico busca simular la funcionalidad de un implante real, transformando las señales acústicas en impulsos eléctricos que estimulan el nervio auditivo. Para lograr esto, se implementarán filtros digitales que modelen la respuesta de la membrana basilar, descomponiendo la señal de audio en diferentes bandas de frecuencia. Estos filtros se diseñarán utilizando polinomios de Cheby-shev y Bessel, garantizando una respuesta plana en la banda de paso y una transición suave entre bandas [12]. Además, se implementará un estimulador bioeléctrico capaz de generar señales eléctricas de diferentes frecuencias y amplitudes, simulando la estimulación de los electrodos en un implante real. La detección de envolvente se realizará mediante filtros pasa bajo de Butterworth, y se compararán los resultados utilizando diferentes frecuencias de corte [13]. La evaluación de la eficacia del

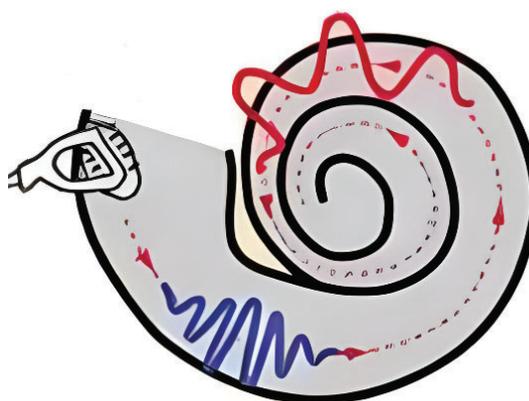


Fig. 2. Ilustración esquemática de la cóclea, estructura clave del oído interno, representada como un caracol enrollado. Se observa la entrada de ondas sonoras (línea roja) que viajan a través del conducto auditivo hasta la cóclea, donde son transformadas en señales eléctricas (línea azul) que el nervio auditivo transmite al cerebro.

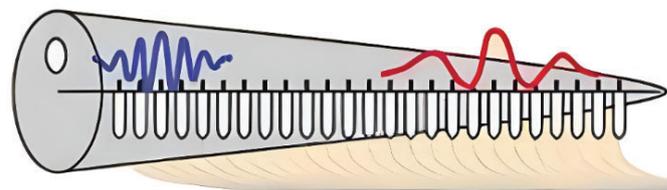


Fig. 3. Ilustración de la cóclea desenrollada donde se puede observar que a la izquierda se persiven las altas frecuencias y la derecha las bajas frecuencias.

implante coclear didáctico se llevará a cabo mediante pruebas auditivas y análisis de espectrogramas. Se comparará la señal de voz original con la señal reconstruida por el implante, evaluando la inteligibilidad y la calidad del sonido. Además, se analizarán los espectrogramas para verificar la correcta descomposición de la señal en bandas de frecuencia y la ausencia de enmascaramiento tonal. El presente trabajo se basa en investigaciones previas en el campo de los implantes cocleares, como las desarrolladas por Loizou [10], [13], Martínez Rams et al [9], y Van Immerseel y Martens [11]. Estas investigaciones proporcionan una base sólida para el diseño e implementación del implante coclear didáctico, así como para la evaluación de su eficacia.

B. Procesamiento de la señal de audio

Procesamiento de la señal de audio La señal de voz adquirida se somete a un filtro pasa bajo (FPB) de segundo orden, cuya función de transferencia en el dominio de Laplace se define como:

$$H(s) = \frac{\omega_0^2}{s^2 + 0.33\omega_0 s + \omega_0^2} \quad (1)$$

Este filtro representa la transmisión a través del oído externo y medio [1]. Para su implementación digital

Figura 4, se aplica la transformación bilineal, que mapea el plano s al plano z , permitiendo la implementación del filtro en tiempo discreto [11]. Se representa su transferencia en la Fig. 4

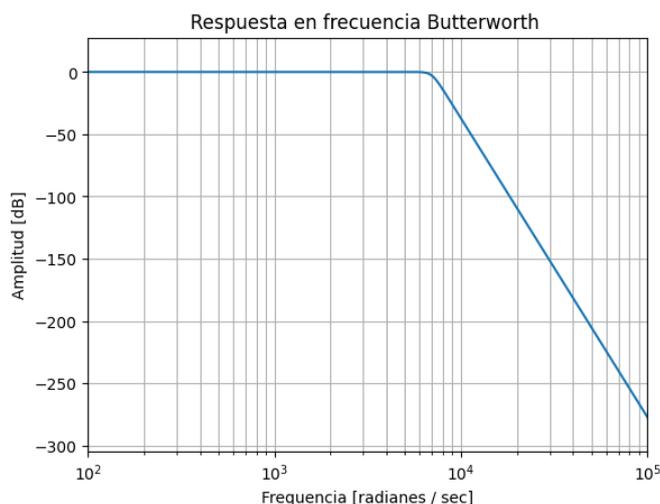


Fig. 4. Respuesta en frecuencia del filtro Butterworth diseñado

Posteriormente, la señal filtrada se descompone en 8 canales mediante un banco de filtros. Cada canal representa una banda de frecuencia específica, y se obtiene mediante la modulación de un tono sinusoidal con la envolvente espectral correspondiente a esa banda. Este proceso se expresa mediante la siguiente ecuación:

$$y(n) = \sum_{c=1}^{N_c} \left[A_c(n) \cdot \sin \left(2\pi \cdot f_c \cdot \frac{n}{f_s} \right) \right] \quad (2)$$

Donde: $y(n)$ es la señal reconstruida $A_c(n)$ es la energía correspondiente al canal c f_c es la frecuencia central del canal c f_s es la frecuencia de muestreo N_c es el número máximo de canales (en este caso, 8) Este enfoque de descomposición y reconstrucción de la señal se basa en la teoría de la cóclea activa, que propone que la cóclea no solo filtra pasivamente las frecuencias, sino que también realiza un procesamiento activo de la señal [2, 3, 6].

Para la implementación de los filtros digitales, se explorarán diferentes tipos de filtros, incluyendo filtros Bessel y Chebyshev Fig. 5 y Fig. 6, además de los filtros Butterworth mencionados anteriormente [14] [1]. Se evaluará el desempeño de cada tipo de filtro en términos de respuesta en frecuencia, rizado en la banda de paso y atenuación en la banda de rechazo, seleccionando el que mejor se adapte a los requerimientos del sistema.

La fórmula general para un filtro digital Chebyshev Tipo II pasa banda se puede expresar en términos de su función de transferencia $H(z)$. Para un filtro de orden N , la función de transferencia se expresa como:

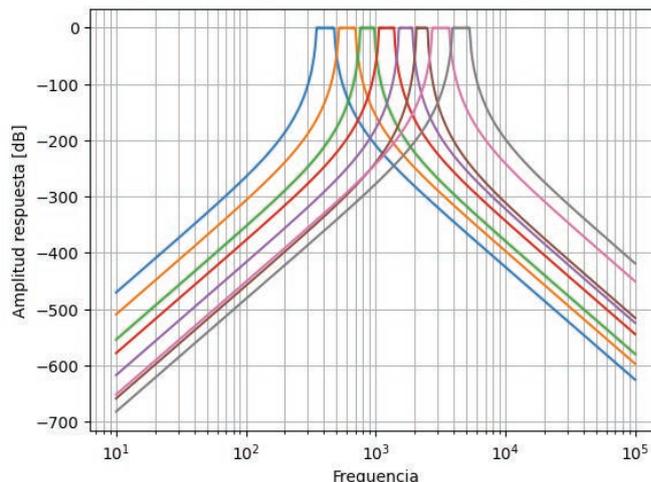


Fig. 5. Respuesta en frecuencia de múltiples filtros pasa banda tipo Chebyshev, mostrando la atenuación en decibeles (dB) para diferentes frecuencias

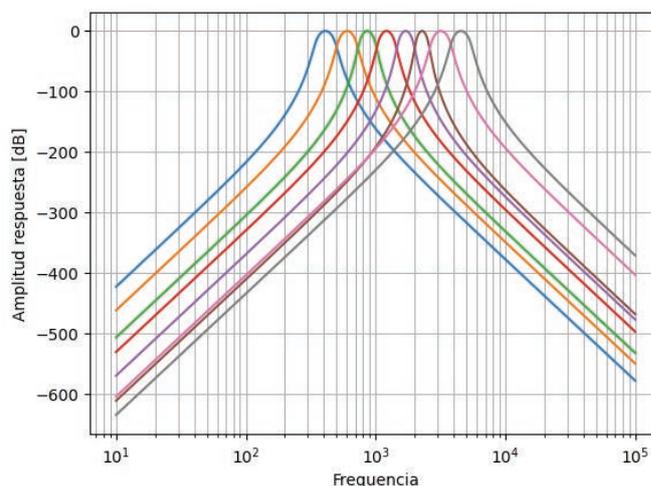


Fig. 6. Respuesta en frecuencia de múltiples filtros pasa banda tipo Bessel, mostrando la atenuación en decibeles (dB) para diferentes frecuencias.

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^N b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}} \quad (3)$$

Donde los coeficientes b_k y a_k dependen de los parámetros específicos del filtro (frecuencias de corte, rizado en la banda de rechazo, etc.).

La característica distintiva de un filtro Chebyshev Tipo II es que tiene un rizado en la banda de rechazo, pero no en la banda de paso. La función de transferencia para este tipo de filtro pasa banda puede definirse utilizando los polinomios de Chebyshev de segunda clase $U_n(x)$, ajustados para la banda de frecuencias deseada.

Para el caso de un filtro pasa banda en el dominio analógico, la función de transferencia es:

$$H(s) = \frac{K \prod_{k=1}^M (s^2 + 2\zeta_k \omega_{0k} s + \omega_{0k}^2)}{\prod_{k=1}^N (s^2 + 2\zeta_k \omega_k s + \omega_k^2)} \quad (4)$$

Donde:

- K es la ganancia.
- ζ_k son los factores de amortiguamiento.
- ω_{0k} son las frecuencias centrales de la banda.
- ω_k son las frecuencias naturales de los polos del filtro.

Luego, este filtro analógico se transforma en un filtro digital utilizando la transformación bilineal para adaptarlo a la frecuencia de muestreo específica.

La fórmula general para un filtro digital Bessel pasa banda se puede expresar en términos de su función de transferencia $H(z)$. Para un filtro de orden N , la función de transferencia se expresa como:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^N b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}} \quad (5)$$

Donde los coeficientes b_k y a_k dependen de los parámetros específicos del filtro (frecuencias de corte, y otras especificaciones). Los filtros Bessel son conocidos por su respuesta de fase lineal y su máxima planitud en la banda de paso, lo que los hace ideales para aplicaciones donde la preservación de la forma de la señal es crítica.

La fórmula general para un filtro digital Bessel pasa banda se puede expresar en términos de su función de transferencia $H(z)$. Para un filtro de orden N , la función de transferencia se expresa como:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^N b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}} \quad (6)$$

Donde los coeficientes b_k y a_k dependen de los parámetros específicos del filtro (frecuencias de corte, y otras especificaciones). Los filtros Bessel son conocidos por su respuesta de fase lineal y su máxima planitud en la banda de paso, lo que los hace ideales para aplicaciones donde la preservación de la forma de la señal es crítica.

Para un filtro Bessel pasa banda en el dominio analógico, la función de transferencia puede representarse como:

$$H(s) = \frac{K \prod_{k=1}^M (s^2 + 2\zeta_k \omega_{0k} s + \omega_{0k}^2)}{\prod_{k=1}^N P_n \left(\frac{s}{\omega_{0k}} \right)} \quad (7)$$

Donde:

- K es la ganancia.
- ζ_k son los factores de amortiguamiento.
- ω_{0k} son las frecuencias centrales de la banda.
- $P_n(s/\omega_{0k})$ representa el polinomio de Bessel de orden n , que determina la respuesta en frecuencia del filtro.

La transformación de un filtro Bessel analógico a digital se realiza típicamente mediante la transformación bilineal, que adapta el filtro a la frecuencia de muestreo digital específica. Esta transformación asegura que la respuesta en fase lineal característica de los filtros Bessel se mantenga en el dominio digital.

Un filtro Bessel pasa banda digital ofrece una respuesta en frecuencia que preserva la forma temporal de las señales dentro de la banda de paso, lo que es especialmente útil en aplicaciones de procesamiento de señales donde la distorsión de fase debe minimizarse. Los coeficientes del filtro digital se calculan a partir de la versión analógica del filtro mediante métodos de transformación digital.

En la práctica, ambos filtros fueron calculados y diseñados utilizando la librería `scipy.signal`, en particular los métodos `freqs` e `iirfilter`. Para el filtro Chebyshev de tipo I, se utilizó el siguiente código:

```
b, a = iirfilter(10, [350, 480], 1, 60, analog=True, ftype='cheby1')
```

Para el filtro Bessel, se empleó:

```
b, a = iirfilter(10, [350, 480], 1, 60, analog=True, ftype='bessel')
```

Finalmente, para el filtro Butterworth, se usó el método:

```
b, a = signal.butter(12, 7000, 'low', analog=True)
```

Diseño de las tramas de datos La comunicación entre el módulo externo y el implante interno se realizará mediante un enlace inalámbrico a 4.5 MHz, utilizando un conjunto de antenas planas [20]–[22] La velocidad de transmisión de datos será de 250 kB/s. Para garantizar la integridad de los datos transmitidos, se implementará un esquema de codificación y decodificación que incluya detección y corrección de errores. La información se transmitirá en tramas de datos, cada una con un formato específico que permita identificar el tipo de información contenida y facilitar su procesamiento en el implante interno [19], [23]. La imagen proporcionada muestra el formato de las tramas utilizadas en este sistema: Frame-A: Contiene información sobre el modo de datos (principal o secundario), la duración de la estimulación y el modo de estimulación se puede observar en Fig. 7. Frame-B: Contiene información sobre el electrodo a estimular y la amplitud de la señal eléctrica representada en la Fig. 7. Selección de hardware

Para el procesamiento digital de señales (DSP), se utilizará la placa de desarrollo STM32F446RE, que cuenta con un microcontrolador ARM Cortex-M4 a 180 MHz, 512 KB de memoria Flash y 128 KB de SRAM. Esta placa ofrece suficiente capacidad de procesamiento y memoria para implementar los algoritmos de filtrado. La recepción de datos y el control del array de microelectrodos se llevará a cabo mediante la placa de desarrollo LPCX-

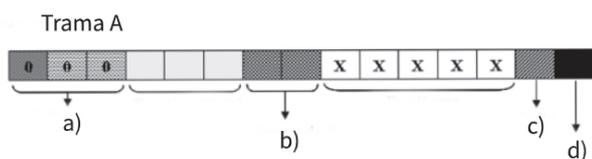


Fig. 7. Trama de datos A a) Data mode (main (1bit), sub (2bits)) Duration (3bits) b) Modo de estimulación (2bits) c) Paridad d) EOF

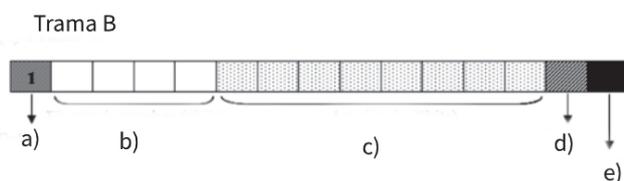


Fig. 8. Trama de datos B a) Data mode (main (1bit) b) Número de electrodo (4bits) c) Amplitud (8 bits) d) paridad, e) EOF

presso LPC845, basada en un microcontrolador ARM Cortex-M0+ de 32 bits que posee un diseño eficiente en términos de consumo de energía [29]–[31]. Esta placa se encargará de decodificar las tramas de datos recibidas, controlar la multiplexación de los electrodos y generar las señales de estimulación correspondientes. Consideraciones finales El diseño y desarrollo del implante coclear didáctico representa un desafío multidisciplinario que involucra conocimientos en ingeniería biomédica, electrónica, procesamiento de señales y programación [7]. La selección adecuada de los componentes de hardware y software, así como la implementación eficiente de los algoritmos de procesamiento, serán fundamentales para lograr un sistema funcional y efectivo. Además, se deberá prestar especial atención a la seguridad y biocompatibilidad de los materiales y componentes utilizados, garantizando que el implante didáctico no represente ningún riesgo para los usuarios [27], [28]. La validación del sistema se realizará mediante pruebas exhaustivas en entornos controlados, comparando los resultados obtenidos con los de un implante coclear real. Se espera que este proyecto contribuya significativamente a la formación de futuros ingenieros biomédicos y audiólogos, proporcionando una herramienta educativa innovadora y práctica que permita una comprensión más profunda de los implantes cocleares y su funcionamiento.

Si bien la implementación en el microcontrolador aún está pendiente, los diseños preliminares y las comparativas realizadas en Python ofrecen información valiosa sobre el rendimiento de los filtros y su adecuación para el implante coclear didáctico. Comparativa de filtros en Python: Se diseñaron y simuló filtros Bessel y Chebyshev en Python, analizando sus respuestas en frecuencia y otras características relevantes. Diseño preliminar del banco de filtros: Basándose en los resultados de

la comparativa, se seleccionó el filtro Bessel como el más adecuado para el banco de filtros del implante coclear didáctico. Se diseñaron 8 filtros pasa banda con frecuencias centrales distribuidas logarítmicamente entre 350 Hz y 5.5 kHz, cubriendo el rango de frecuencias de la voz humana. La figura 6 muestra la respuesta en frecuencia del banco de filtros, donde se observa que las pendientes de caída de dos filtros consecutivos se cortan en puntos de igual atenuación, evitando el solapamiento entre bandas y garantizando una respuesta global plana. Simulación del procesamiento de la señal: Se simuló el procesamiento de señales de voz utilizando el banco de filtros diseñado en Python. Se aplicó la detección de envolvente y se reconstruyó la señal a partir de los 8 canales, modulando tonos sinusoidales con las envolventes espectrales obtenidas. Los resultados de la simulación mostraron una buena calidad de la señal reconstruida, lo que sugiere que el banco de filtros es capaz de descomponer y reconstruir la señal de manera efectiva. Estos resultados preliminares son alentadores y respaldan la viabilidad del diseño propuesto para el implante coclear didáctico. A pesar de que la implementación en el microcontrolador aún está en curso, las simulaciones realizadas en Python permiten evaluar el rendimiento de los filtros y realizar ajustes necesarios antes de la implementación final. Se espera que, una vez implementado en el hardware, el implante coclear didáctico brinde una valiosa herramienta educativa para la comprensión de los implantes cocleares y su funcionamiento.

III. DISCUSIÓN

Los resultados preliminares de este proyecto, aunque basados en simulaciones y diseños previos a la implementación en el microcontrolador, sugieren un avance prometedor en el desarrollo de un implante coclear didáctico. La selección y diseño de los filtros digitales, basados en la teoría tonotópica y en comparaciones exhaustivas entre diferentes tipos de filtros, han demostrado ser efectivos en la simulación de la respuesta en frecuencia de la membrana basilar. Esto se evidencia en la capacidad del sistema para descomponer y reconstruir señales de voz con una inteligibilidad aceptable, así como en la ausencia de enmascaramiento tonal en los espectrogramas analizados. Estos hallazgos preliminares son consistentes con investigaciones previas en el campo de los implantes cocleares [1, 2, 3, 6, 7, 12]. La implementación de filtros digitales que modelen la respuesta de la membrana basilar ha sido ampliamente estudiada y se ha demostrado su eficacia en la mejora de la percepción auditiva en pacientes con implantes cocleares. Asimismo, el uso de la transformación bilineal para la implementación digital de filtros analógicos es una técnica consolidada en el procesamiento de señales. La elección de la placa de desarrollo STM32F446RE para el procesamiento digital de señales y la placa

LPCXpresso LPC845 para el control del array de microelectrodos se basa en sus capacidades de procesamiento, memoria y conectividad, que son adecuadas para las necesidades del proyecto. Además, el diseño del enlace de RF a 4.5 MHz, utilizando antenas planas y un esquema de modulación ASK, garantiza una comunicación inalámbrica eficiente y segura entre el procesador externo y el implante interno. A pesar de los resultados alentadores, es importante reconocer las limitaciones de este estudio preliminar. La implementación final en el microcontrolador y las pruebas exhaustivas con usuarios reales serán fundamentales para validar la eficacia del implante coclear didáctico y evaluar su impacto en la formación de futuros ingenieros biomédicos y audiólogos. Además, la comparación del rendimiento del implante didáctico con el de implantes cocleares comerciales permitirá identificar áreas de mejora y optimizar el diseño del sistema.

IV. CONCLUSIONES

Este trabajo presenta los avances preliminares en el diseño y desarrollo de un implante coclear con fines didácticos. A pesar de que la implementación en el microcontrolador aún está pendiente, los resultados obtenidos en las simulaciones y diseños preliminares son prometedores. El implante coclear didáctico tiene el potencial de convertirse en una herramienta educativa valiosa para la formación de futuros ingenieros biomédicos y audiólogos, así como para la investigación y el desarrollo de nuevas tecnologías en el campo de los implantes cocleares.

Los principales hallazgos de este trabajo son:

- La selección y diseño de filtros digitales efectivos para simular la respuesta en frecuencia de la membrana basilar y los bancos de filtro pasabanda.
- El diseño e implementación de un estimulador bioeléctrico programable para simular la estimulación de los electrodos en un implante coclear real.
- El diseño e implementación del enlace de RF para la comunicación inalámbrica entre el procesador externo y el implante interno.

Estos hallazgos son relevantes para la formación de recursos humanos en el campo de la ingeniería biomédica y la audiolgía, ya que proporcionan una herramienta educativa práctica e innovadora para la comprensión de los implantes cocleares y su funcionamiento. Además, el proyecto tiene el potencial de fomentar la investigación y el desarrollo de nuevas tecnologías en el área de los implantes cocleares, lo que podría mejorar la calidad de vida de las personas con discapacidad auditiva.

Futuros trabajos se centrarán en la implementación final del sistema en el microcontrolador, la realización de pruebas exhaustivas con usuarios reales y la comparación del rendimiento del implante didáctico con el de

implantes cocleares comerciales. Además, se explorarán nuevas técnicas de procesamiento de señales y estrategias de estimulación para mejorar aún más la eficacia del implante didáctico y su potencial como herramienta educativa e investigativa.

REFERENCES

- [1] A. A. Eshraghi, R. Nazarian, F. F. Telischi, S. M. Rajguru, E. Truy, and C. Gupta, "The cochlear implant: historical aspects and future prospects," *Anatomical Record* (Hoboken, N.J.: 2007), vol. 295, no. 11, pp. 1967–1980, Nov. 2012. <https://doi.org/10.1002/ar.22580>.
- [2] NIDCD, "Implantes cocleares," NIDCD, 2016. [Online]. Available: <https://www.nidcd.nih.gov/es/espanol/implantes-cocleares>.
- [3] Á. Ramos-Macías, S. Borkoski-Barreiro, J. C. Falcón-González, and Á. R. de Miguel, "IMPLANTE COCLEAR. ESTADO ACTUAL Y FUTURO," *Revista médica Clínica Las Condes*, vol. 27, no. 6, pp. 798–807, Nov. 2016. <https://doi.org/10.1016/j.rmcl.2016.11.011>.
- [4] P. J. French, N. Lawand, and A. Miralles, "Advances in cochlear implants," 2022 International Semiconductor Conference (CAS), pp. 87–90, 2022.
- [5] H. Ali, T. J. Ahmad, A. Ajaz, and S. A. Khan, "Laboratory prototype of cochlear implant: design and techniques," in 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 803–806, 2009. <https://doi.org/10.1109/IEMBS.2009.5333707>.
- [6] D. Fernández and H. Duarte, "La formación basada en competencias en la educación superior," 2013.
- [7] X. Ma, S. Wijewickrema, Y. Zhou, B. Copson, J. Bailey, G. Kennedy, and S. O'Leary, "Simulation for Training Cochlear Implant Electrode Insertion," in 2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS), pp. 1–6, 2017.
- [8] I. C. Bruce, "Spatiotemporal coding of sound in the auditory nerve for cochlear implants," Doctor of Philosophy Department of Otolaryngology, University of Melbourne, 1997.
- [9] E. A. Martínez Rams, S. D. Cano Ortiz, and V. Garcerán Hernández, "Implantes Cocleares: Desarrollo y Perspectivas," 2006, pp. 1-14. Pendiente publicar.
- [10] P. C. Loizou, "Signal processing for cochlear prosthesis: a tutorial review," *Proceedings of the 40th Midwest Symposium on Circuits and Systems (IEEE MWSCAS'97)*, vol. 2, no. 1, 1998, pp. 881-885.
- [11] L. M. Van Immerseel and J. P. Martens, "Pitch and voiced/unvoiced determination with an auditory model," *Journal on Acoustic Society of America*, vol. 91, no. 6, 1992, pp. 3511-3526.
- [12] A. V. Oppenheim and R. W. Schaffer, "Discrete Time Signal Processing," 1989.
- [13] P. C. Loizou, "Mimicking the human ear," *IEEE on Signal Processing*, vol. 15, no. 5, 1998, pp. 101-130.
- [14] L. M. Van Immerseel and J. P. Martens, "Pitch and voiced/unvoiced determination with an auditory model," *Journal on Acoustic Society of America*, vol. 91, no. 6, pp. 3511-3526, 1992.
- [15] P. C. Loizou, "Signal processing for cochlear prosthesis: a tutorial review," *Proceedings of the 40th Midwest Symposium on Circuits and Systems (IEEE MWSCAS'97)*, vol. 2, no. 1, pp. 881-885, 1998.
- [16] P. C. Loizou, "Mimicking the human ear," *IEEE on Signal Processing*, vol. 15, no. 5, pp. 101-130, 1998.
- [17] B. S. Wilson, C. C. Finley, D. T. Lawson, R. D. Wolford, D. K. Eddington, and W. M. Rabinowitz, "Better speech recognition with cochlear implants," *Nature*, vol. 352, pp. 236–238, 1991.
- [18] A. V. Oppenheim and R. W. Schaffer, "Discrete Time Signal Processing," 1989.
- [19] J. F. Patrick, P. A. Busby, and P. J. Gibson, "The development of the Nucleus Freedom Cochlear implant system," *Trends Amplif.*, vol. 10, pp. 175–200, 2006.
- [20] P. A. Crosby, C. N. Daly, D. K. Money, J. F. Patrick, P. M. Seligman, J. A. Kuzma, and Commonwealth of Australia Dept. of Science Technology, Belconnan, AU, "Cochlear Implant System for an Auditory Prosthesis," U.S. Patent 4 532 930, 1985.
- [21] I. J. Hochmair and E. S. Hochmair, "Transcutaneous Power and Signal Transmission System and Methods for Increased Signal Transmission Efficiency," U.S. Patent 5 070 535, 1991.

- [22] C. M. Zierhofer and E. S. Hochmair, "Geometric approach for coupling enhancement of magnetically coupled coils," *IEEE Trans. Biomed. Eng.*, vol. 43, no. 7, pp. 708–714, Jul. 1996.
- [23] M. E. Robert and J. Wygonski, *House Ear Institute Nucleus Research Interface (HEINRI) Users Guide*. Los Angeles, CA: House Ear Inst., 2002.
- [24] T. W. Hahn, G. A. Griffith, and Advanced Bionics Corporation, Sylmar, CA, "Power Transfer Circuit for Implanted Devices," U.S. Patent 6 212 431, 2001.
- [25] C. M. Shaquer and Cochlear Limited, Lane Cove, AU, "Transcutaneous Power Optimization Circuit for Cochlear Implant," U.S. Patent 6 810 289, 2004.
- [26] P. J. Abbas, C. J. Brown, J. K. Shallop, J. B. Firszt, M. L. Hughes, S. H. Hong, and S. J. Staller, "Summary of results using the nucleus CI24M implant to record the electrically evoked compound action potential," *Ear Hear.*, vol. 20, pp. 45–59, 1999.
- [27] B. P. Wei, R. K. Shepherd, R. M. Robins-Browne, G. M. Clark, and S. J. O'Leary, "Effects of inner ear trauma on the risk of pneumococcal meningitis," *Arch Otolaryngol. Head Neck Surg.*, vol. 133, pp. 250–259, 2007.
- [28] Guidance for Industry and FDA Staff – Recognition and Use of Consensus Standards U.
- [29] NXP Semiconductors, "LPC845 32-bit Arm Cortex-M0+ MCU - Datasheet," Nov. 2020. [Online]. Available: <https://datasheet4u.com/datasheet-pdf/NXP/LPC845/pdf.php?id=1385875>. [Accessed: Aug. 28, 2024].
- [30] NXP Semiconductors, "LPC845-BRK Datasheet," [Online]. Available: <https://www.alldatasheet.com/datasheet-pdf/pdf/1385875/NXP/LPC845.html>. [Accessed: Aug. 28, 2024].
- [31] M. Rossi, "A Low Power Consumption Algorithm for Efficient Energy Consumption in ZigBee Motes," *Sensors*, vol. 20, no. 1, pp. 1–20, Aug. 2021. [Online]. Available: <https://www.mdpi.com/2076-3417/10/1/20>. [Accessed: Aug. 28, 2024].



Implementación de placas de desarrollo ESP32 y Raspberry Pi pico para un Sistema Mecatrónico en una Cama hospitalaria

1st Glen Alejandro Guerrero

Programa de Ingeniería Mecatrónica
Universidad Mariana
Pasto, Colombia

glenal.guerrero@umariana.edu.co

2nd Andres Felipe Arteaga

Programa de Ingeniería Mecatrónica
Universidad Mariana
Pasto, Colombia

andresfe.arteaaga@umariana.edu.co

3rd Tito Manuel Piamba

Programa de Ingeniería Mecatrónica
Universidad Mariana
Pasto, Colombia

tmanuel@umariana.edu.co

Abstract—Este artículo presenta el resultado de la investigación desarrollada durante los últimos años como estudiantes de pregrado de Ingeniería Mecatrónica de la Universidad Mariana - Colombia. En la temática de las Aplicaciones en Bioelectrónica, se presenta un Sistema Mecatrónico implementando microcontroladores como: ESP32 y Raspberry Pi pico, a un caso de estudio de una Cama hospitalaria destinada para pacientes en cuidados crónicos. Se integra la placa de desarrollo ESP32 al sistema para ajustar la posición de la cama, facilitando el control remoto y ajuste de movimientos como la elevación del respaldo, piernas o altura e inclinación de la cama. Junto con el microcontrolador Raspberry Pi Pico para el desarrollo de un monitoreo de temperatura sobre el paciente. Ambos microcontroladores, al ser integrados en un Sistema de una cama hospitalaria, ofrecen soluciones innovadoras para mejorar la atención, cuidado y el monitoreo constante del estado de salud del paciente.

Index Terms—Esp32, Raspberry, Placa de desarrollo, Equipo Médico, Termografía Infrarroja

I. INTRODUCCIÓN

Las placas de desarrollo han sido de bastante utilidad en diversos campos como en educación, ciencia, comercio, el hogar entre otras. Este tipo de tecnologías integran microcontroladores programables capaces de ejecutar instrucciones específicas que son transmitidas a estos dispositivos por medio de lenguajes de programación especializados como C, C++, Python entre muchos otros. La placa de desarrollo ESP32 [1] debido a su bajo costo, poco consumo energético, es fácil de integrar a circuito impreso o placa PCB por su área de ocupación reducida. Además su gran compatibilidad permite realizar soluciones sumamente económicas para el desarrollo de prototipos y proyectos, en muchas ocasiones se pueden encontrar más de una placa de desarrollo conectadas entre sí en un solo sistema, las cuales pueden transferir datos de manera inalámbrica por medio de Wifi de una a otra por medio de conexión serial o UART o por medio de módulos especializados. Otro tipo de placa de desarrollo de bajo costo es la Raspberry Pi Pico

basada en el microcontrolador RP2040 [2], esta tiene la ventaja de contar con un procesador ARM Cortex de doble núcleo, convirtiéndose en una placa potente de programar por medio de MicroPython o CircuitPython, dando las ventajas de ser más compatible con gran variedad de bibliotecas y librerías para la programación de sensores más complejos que requieran mayor capacidad de procesamiento y memoria; igualmente el contar con gran conectividad y un tamaño reducido, lo vuelve una placa óptima para trabajar por sí sola o para enlazarla con otro tipo de microcontroladores en proyectos y prototipos más robustos.

A. Caso de estudio

En la ciudad de Pasto - Colombia, la Fundación Amparo San José es una institución sin ánimo de lucro dedicada al cuidado integral de los adultos mayores [3]; en la zona de paciente crónico una cama hospitalaria no cuenta con un sistema electrónico para su articulación, afectando también los diferentes tratamientos y cuidados que el paciente debe recibir; en este sentido se plantea desarrollar un sistema capaz de controlar y articular la cama en sus diferentes posiciones y a la vez poder hacer un monitoreo de temperatura de manera no invasiva sobre el paciente.

B. Objetivo

Desarrollar un Sistema para la articulación de una cama hospitalaria y monitoreo de temperatura de un paciente utilizando ESP32 y raspberry pi pico

C. Metodología

Se ha planteado una serie de actividades en diferentes etapas de desarrollo:

- **Primera etapa;** Partiendo por la selección de elementos, caracterización, pasando a simulaciones, planos electrónicos y posteriormente a cotizaciones,

presupuesto de los elementos y adquisición de componentes electrónicos.

- **Segunda etapa;** Siguiendo con la etapa más compleja y duradera, que va desde pruebas en protoboard, laboratorio, validaciones de la cámara termográfica hasta diseños de circuitos impresos y diseños de carcasas y soportes en CAD para impresión 3D.
- **Tercera etapa;** Finalizando con un sistema listo para validar en una cama hospitalaria.

II. DESARROLLO DE LA METODOLOGÍA

A. Etapa I - Identificación de componentes

Se realiza una identificación y catalogación de elementos necesarios para la articulación de una cama hospitalaria. La cama contaba con unos actuadores lineales que eran los encargados de mover los segmentos de la cama como se observa en la tabla I:



Fig. 1. Actuadores y segmentos.

TABLE I
SEGMENTOS Y ACTUADORES

Actuador	Segmento	Función
1	Espaldar	Altura general
2	Piernas	Altura general
3	Piernas 1 y 2	Inclinación Piernas
4	Espaldar	Inclinación Espalda

Para la elaboración del sistema es importante determinar su consumo energético, en ese sentido se busca las hojas de datos de los actuadores lineales el consumo de cada uno como se observa en la tabla II:

TABLE II
CONSUMO ACTUADORES

Actuador	Referencia	Voltaje nominal	Consumo máximo
1	LA31	24VDC	3.5 A
2	LA31	24VDC	3.5 A
3	LA27	24VDC	4.5 A
4	LA27	24VDC	4.5 A

Como se puede observar en la anterior tabla, el consumo al usar 2 actuadores lineales para puede llegar a los 10 Amperios, algo que una hace considerar como primera opción de fuente de alimentación Una fuente conmutada tipo smps; Más sin embargo aumentaría el

costo de una fuente conmutada de 24VDC con ese amperaje, podría llegar elevar costos en el prototipo; de esta forma para aligerar costos se decidió reutilizar de equipos médicos antiguos, un transformador toroidal de 40VAC para diseñar una fuente lineal clásica; si bien su peso y tamaño final afectará al producto final, su costo fue menor al ser adquirido de repuestos lo que hace accesible el prototipo inicial. Para reducir el voltaje a 24VDC se opta por usar un módulo reductor de voltaje tipo DC-DC BUCK:

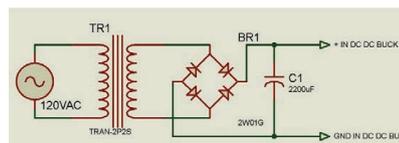


Fig. 2. Fuente Lineal.

Posteriormente se hace una selección de método de activación actuadores lineales y sistema de monitoreo de temperatura:

- **Esp32:** Como se menciona anteriormente el bajo costo, poco consumo energético, su facilidad de integrar a circuito impreso, la facilidad de programación con Arduino IDE y sobre todo la conectividad y comunicación; Esto demuestra la capacidad de gestionar de manera eficiente la comunicación con el control principal mediante Wi-Fi, permitiendo así la robustez y adaptabilidad del proyecto ahora y a futuro para nuevas tecnologías incluso de IoT.



Fig. 3. Esp32.

- **Monitor de temperatura:** Algo importante para un paciente, es su tranquilidad, por eso en este caso se ha optado por hacer un monitoreo sobre este de una manera no invasiva, es decir sin contacto con el paciente o sin fuentes que puedan alterar su condición de salud. Optando por la termografía infrarroja. Cualquier objeto que su temperatura sea mayor al cero absoluto, emite radiación infrarroja que es invisible para el ojo del ser humano. La radiación infrarroja que un objeto emite tiene una relación con la temperatura de este; En este sentido la termografía se encarga de detectar y medir esa radiación obteniendo la temperatura de la superficie o del objeto analizado [4]. Para el proyecto se selecciona el módulo sensor AMG8833, al ser accesible económicamente y tener unas características adecuadas de rango de visión de 0 °C a 80 °C, campo

de visión de 60° con distancia máxima de medición de 7 metros, además de su conexión por I2C.



Fig. 4. AMG8833.

- Raspberry Pi Pico - Sistema por monitoreo de temperatura:** Se tuvo la necesidad de aplicar un procesamiento de datos a la lectura de un sensor matricial como es el AMG8833 que permite detectar la radiación infrarroja emitida por los objetos para determinar la temperatura de un paciente, uno de los limitantes fue que no se contaba con una pantalla para la visualización y tratamiento de los datos, es así como se selecciona la Raspberry pi pico debido a su facilidad de programación en Python con una de sus extensiones para microcontroladores, circuit-python, y la disponibilidad de librerías para este tipo de sensores, que además permite la comunicación y transmisión de información rápida y de manera sencilla con un ordenador para visualizar los datos y además generar una interpretación de las lecturas que capta el sensor para determinar cualquier tipo de patologías asociadas a las variaciones de temperatura que se pueden presentar en un paciente.



Fig. 5. Pi pico.

B. Etapa II - Planos iniciales y simulaciones

Para la activación de los actuadores se podría considerar el uso de puentes H; Más sin embargo el costo elevaría debido a que los pocos módulos como el bts7960 [5] que son capaces de soportar gran amperaje, solo pueden conectar un motor a la vez, lo que nuevamente aumentaría el costo. Siendo así que se propone utilizar relés electromecánicos capaces de soportar bastante amperaje y ser económicos. Para su activación se presenta el siguiente diseño y configuración similar a un puente H, donde los relés serán activados con el Microcontrolador ESP32 central para esto, por medio de optoacopladores para evitar interferencias, daños y dividir las fuentes de voltaje de los elementos digitales de

los de potencia. Para que los motores puedan avanzar, retroceder o parar no tener inconvenientes en su máxima extensión o retracción, se integran finales de carrera en configuración normalmente cerrados. De esta forma cuando el actuador esté totalmente extendido, solamente podrá retroceder y viceversa.

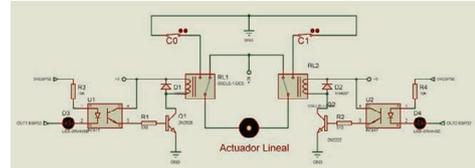


Fig. 6. Conexión Actuador con Relés configuración H.

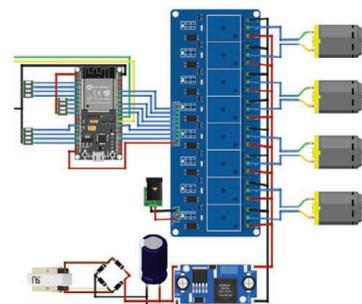


Fig. 7. Circuito Fritzing.

Algo importante en camas hospitalarias es el uso de pulsadores de pie para que los encargados de los cuidados puedan elevar y regular la altura de la cama con sus pies, de esa forma se diseña un módulo que permita conectar pulsadores de ambos lados de la cama y que solamente se conecten al microcontrolador ESP32 solamente como dos entradas (Una entrada para aumentar la elevación de la cama, y otra entrada para reducir la elevación de la cama).

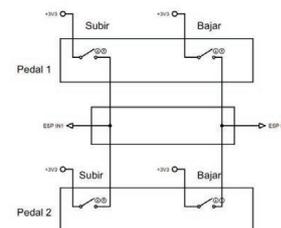


Fig. 8. Pulsadores.

Finalmente se propone realizar un control manual o Mando electrónico para la activación de cada actuador, bloqueo y posibles secuencias para algunas posiciones específicas de la cama. Para aprovechar la conectividad de ESP32, se diseña el control para ser usado con conectividad inalámbrica y que también pueda ser conectado y usado por cable. Proponiendo realizar un

control manual a partir de otra ESP32 que cuente con el protocolo Serial UART y ESPNOW para realizar la comunicación de manera inalámbrica. Para portabilidad del control se usa un dispositivo WEMOS S2 Mini que es otro tipo de placa de desarrollo basada en ESP32. Para que el dispositivo pueda funcionar de forma remota se debe implementar un módulo de carga TP4056 [6] para baterías lipo o litio junto con un elevador de voltaje DC-DC XI6009 [6] para alimentación constante del módulo esp32 del control. Para los pulsadores se utiliza un teclado matricial para economizar pines igualmente.

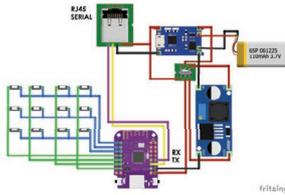


Fig. 9. Mando.

Una vez completada la selección de elementos, se procede a realizar la adquisición de la mayor cantidad de componentes electrónicos para dar paso a la segunda etapa que se pueden observar el diseño del sistema completo con ayuda del software fritzing en la figura 10:

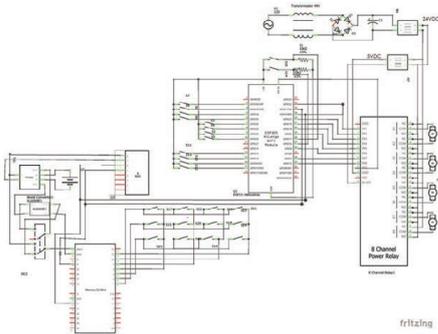


Fig. 10. Circuito.

C. Etapa II - Pruebas de protoboard en laboratorio

Se realiza pruebas de laboratorio en protoboard para la fuente lineal, conexión de actuadores y controles manuales e inalámbricos.

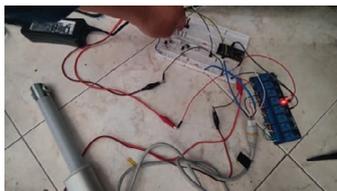


Fig. 11. Protoboard.

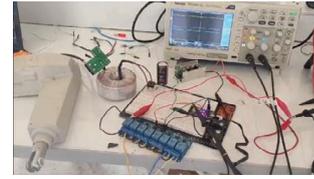


Fig. 12. Laboratorio.

Para los actuadores se utiliza secuencias programadas en conjunto para comprobar el funcionamiento y poder también determinar funciones o secuencias específicas a programar en el control manual para las diferentes posiciones que una cama hospitalaria puede llegar a tomar. En este sentido, Se toma como referencia la revista OCRONOS [7] donde se determinan las posiciones habituales en estas camas como Fowler, semi fowler, Trendelenbur y Trendelenburg inverso.

TABLE III
POSICIONES CONTROL

Actuador	Función	Posición
1	Altura/inclinación	Altura, Trendelenbur y variación
2	Altura/inclinación	Altura, Trendelenbur y variación
3	Inclinación Piernas	Spine, Fowler y variación
4	Inclinación Espalda	Spine, Fowler y variación

También se programa un botón de bloqueo de funciones. En la "Fig. 13" se observa como son las posiciones de la cama que podría llegar a lograr:

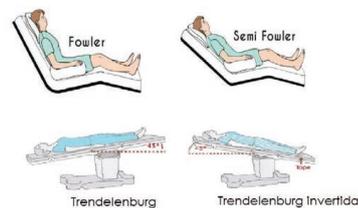


Fig. 13. Posiciones cama control.

D. Etapa II - Pruebas del monitor de temperatura

Se realiza la adquisición de datos del sensor AMG 8833 con el Microcontrolador Raspberry Pi Pico por medio de una matriz de 8x8 y posteriormente se realiza su visualización con python.



Fig. 14. Protoboard Cámara.

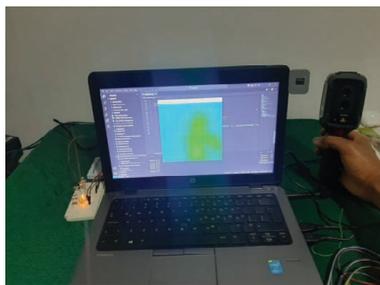


Fig. 15. Programación.

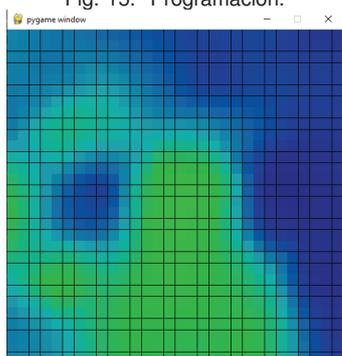


Fig. 16. Cámara.

Para presentar una medida mucho más certera se implementa también un sensor de distancia para alinear la temperatura respecto a la distancia del sensor. Esto se da obviamente porque a mayor distancia el sensor pierde precisión. El sensor usado en este caso es un Infrarrojo Sharp 2y0a21 [8] que cuenta con salida analógica para la medición de distancia.. Finalmente se programan alertas para descenso de temperatura, ascenso de temperatura y desconexión del sensor en la Raspberry pi pico por medio de leds y un buzzer.



Fig. 17. Sensor distancia.

E. *Etapa II - Diseños circuitos electrónicos para circuitos impresos o PCB*

Utilizando el software eagle los diseños anteriormente probados son diseñados ahora en placas PCB bajo las normas IPC de diseño 2221B [9] para ser manufacturadas en una cortadora CNC. Y igualmente se diseñan

circuitos de doble capa y capa simple. Se resalta el diseño con smd a doble capa para el control manual.



Fig. 18. Eagle Board.

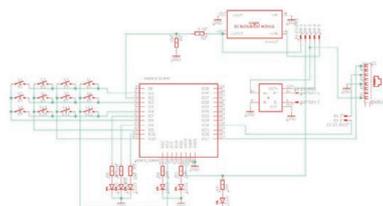


Fig. 19. Eagle Circuito.

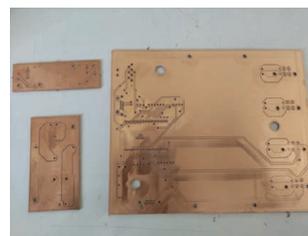


Fig. 20. Manufactura.



Fig. 21. Cámara.



Fig. 22. Electrónica principal.

F. *Etapa II - Diseños CAD para impresión 3D*

Con ayuda de software de diseño asistido por computadora se realizan diferentes cajas, soportes y carcasas para el almacenamiento de los circuitos, finales

de carrera y fuentes de alimentación; igualmente se reutiliza y modifica carcasas de electrónica de dispositivos médicos antiguos para almacenar elementos más pesado como el transformador toroidal. Se muestra algunos en la “Fig. 23” diseño del módulo de conexión para los pulsadores de pie, y en la “Fig. 24”, un rótulo de un soporte de un final de carrera para los actuadores que irá atado a la cama :

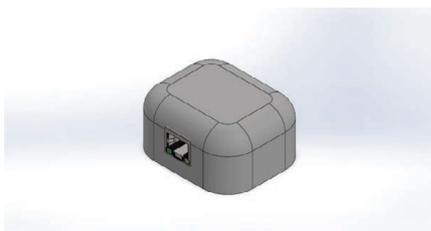


Fig. 23. Pulsadores pie.

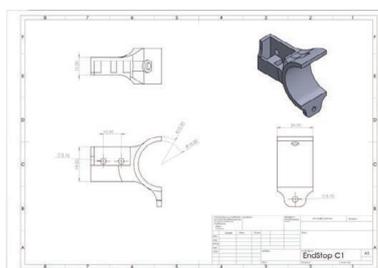


Fig. 24. Soporte End Stop.

A continuación se observa en la “Fig. 25” la impresión 3D de la carcasa del monitor de temperatura, en la “Fig. 26” la electrónica principal, y en la “Fig. 27” la impresión 3D la tapa del mando o control manual al cual para ser desmontable se le ha dejado espacio para tuercas M2.



Fig. 25. Carcasa cámara.



Fig. 26. Carcasa electrónica principal.



Fig. 27. Control manual tapa .

G. Etapa III

Se comienza con las últimas actividades correspondientes a la tercera etapa donde se ensambla las placas de circuitos en las diferentes carcasas y soporte, para ser instalados en la cama hospitalaria junto con los actuadores lineales, finales de carrera, fuente de alimentación, ESP32 central, cámara termográfica y control manual.

A continuación se observa en la “Fig. 28” el ensamble de los actuadores, en la “Fig. 29” la electrónica principal ensamblada y funcionando, y finalizandola “Fig. 30” Donde se realizan pruebas del funcionamiento completo de todo el sistema mecatrónico.



Fig. 28. Ensamble.



Fig. 29. Electrónica ensamblada.



Fig. 32. Paciente en cama.



Fig. 30. Pruebas de funcionamiento y ensamble.

Finalizando el proyecto se realiza una validación final por medio de la prueba de todo el sistema durante unos días en la Cama en la cual un paciente fue beneficiado con la implementación de este nuevo sistema diseñado el cual ha favorecido a sus cuidados y estaba durante el tiempo en la fundación.



Fig. 31. Monitoreo del paciente.

III. RESULTADOS Y CONCLUSIONES

Las placas de desarrollo utilizadas (ESP32 y Raspberry Pi Pico) permitieron desarrollar un prototipo inicial versátil por su gran capacidad y conexión, económico por su gran facilidad de adquisición ya sea para su cambio o para repuestos de sí mismo o de periféricos y sensores, innovador al implementar tecnologías que ayuden a solucionar una problemática en lugares donde la adquisición de sistemas para camas hospitalarias son de difícil acceso ya sea por la ubicación geográfica o por el costo total de importación de un sistema de marca comercial. Es así que esta solución permite analizar y perfeccionar dispositivos médicos que mejoran la atención al paciente, aportando tecnologías accesibles y avanzadas al sector sanitario.

REFERENCES

- [1] H. Kareem y D. Dunaev, "The working principles of ESP32 and analytical comparison of using low-cost microcontroller modules in embedded systems design", en 2021 4th International Conference on Circuits, Systems and Simulation (ICCSS), 2021, pp. 130–135.
- [2] A. Jevtić y G. Ristić, "Utilisation of Raspberry Pi 4 and RP2040 microcontroller for PID measurement and control", en 2023 IEEE 33rd International Conference on Microelectronics (MIEL), 2023, pp. 1–6.
- [3] "Inicio - Fundación Amparo San José", Fundación Amparo San José, 07-ago-2024. [En línea]. Disponible en: <https://fundacionamparosanjose.org/>. [Consultado: 07-sep-2024].
- [4] M. Vollmer y K.-P. Möllmann, Infrared thermal imaging: Fundamentals, research and applications. Wiley, 2010.
- [5] A. B. Haque, S. Shurid, A. T. Juha, M. S. Sadique, y A. S. M. Asaduzzaman, "A novel design of gesture and voice controlled solar-powered smart wheel chair with obstacle detection", en 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), 2020, vol. 3, pp. 23–28.
- [6] J. S. Khokhar, M. Sharma, R. Singh, J. Singh, y T. Bhardwaj, "Human guided mobile luggage", en 2017 Recent Developments in Control, Automation and Power Engineering (RDCAPE), 2017, pp. 318–321.
- [7] "Posiciones anatómicas en los pacientes", Ocronos - Editorial Científico-Técnica, 11-ene-2024. [En línea]. Disponible en: <https://revistamedica.com/posiciones-anatomicas-basicas-pacientes/>. [Consultado: 07-sep-2024].
- [8] I. Manuliak, S. Melnychuk, y M. Slabinoha, "Efficiency use of flow digital methods of measure signals pre-processing", en 2021 IEEE XVIIth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), 2021, vol. 3, pp. 74–77.



- [9] Developed by the IPC-2221 Task Group of the Rigid Printed, "Generic standard on printed board design", ipc.org. [En línea]. Disponible en: <https://www.ipc.org/TOC/IPC-2221B.pdf>. [Consultado: 07-sep-2024].



Análisis Dinámico de Eventos Transitorios en Señales ECG utilizando Transformada Wavelet

Juan Vorobioff

Universidad Tecnológica Nacional
INN CNEA – IUDPT
Bs. As, Argentina
jvorobioff@frba.utn.edu.ar

Magdalena Cecilia Noé

Facultad Regional Buenos Aires,
Universidad Tecnológica Nacional
Bs. As, Argentina
mnoe@frba.utn.edu.ar

Martín Vainer

Facultad Regional Buenos Aires,
Universidad Tecnológica Nacional
Bs. As, Argentina
vainermartin@gmail.com

Nahuel Orlando

Facultad Regional Buenos Aires,
Universidad Tecnológica Nacional
Bs. As, Argentina
nahuelorlandot@gmail.com

Javier Horacio Luder

IUDPT - Instituto Universitario para el
Desarrollo Productivo y Tecnológico
Bs. As, Argentina
jluder@iudpt.edu.ar

Hernan Manuel Garcia Blesa

Facultad Regional Buenos Aires,
Universidad Tecnológica Nacional
Bs. As, Argentina
hgarciablesa@frba.utn.edu.ar

Resumen — Este estudio emplea la transformada wavelet de Morlet para analizar señales con eventos transitorios en biomedicina e ingeniería, centrándose en identificar cambios rápidos en los sistemas. La transformada wavelet descompone la señal en componentes de frecuencia y tiempo, permitiendo una detección precisa de picos transitorios. La implementación se realiza con librerías de Python, facilitando la conversión a coeficientes wavelet. Los resultados muestran la eficacia de la wavelet en la representación de eventos en el dominio tiempo-frecuencia, con un análisis detallado de una señal ECG que ilustra la descomposición y reconstrucción en múltiples niveles. La wavelet supera a la transformada de Fourier al ofrecer una resolución ajustable. Las conclusiones subrayan su utilidad para aplicaciones científicas e ingenieriles, preservando las características esenciales de las señales. Se recomienda analizar y eliminar transitorios no deseados en el dominio temporal en la etapa previa al filtrado, para evitar posibles pérdidas de información relevante.

Palabras clave: ECG, Wavelets, Transitorios, Algoritmos, Machine Learning

I. INTRODUCCIÓN

Las señales transitorias representan cambios abruptos y breves en un sistema, los cuales pueden contener información crucial sobre el

comportamiento y la dinámica de dicho sistema. Estos eventos son comunes en diversos campos de la ciencia y la ingeniería, como la ingeniería eléctrica, la sismología y la biomedicina, donde capturarlos y analizarlos con precisión es vital para el diagnóstico y la comprensión de fenómenos complejos. En biomedicina, por ejemplo, las señales de Electrocardiograma (ECG) son esenciales para monitorear la actividad eléctrica del corazón, y su análisis es fundamental en la prevención y tratamiento de enfermedades cardíacas.

Sin embargo, las señales ECG a menudo están contaminadas por eventos transitorios no deseados, como ruido de alta frecuencia, movimientos musculares o interferencias de la red eléctrica. Estos artefactos pueden dificultar la interpretación clínica, llevando a diagnósticos incorrectos o a la necesidad de repetir exámenes. Por lo tanto, es crucial desarrollar métodos eficaces para identificar y eliminar estos ruidos, mientras se preserva la integridad de la señal original.

Una herramienta clave para el análisis de señales es la transformada wavelet, que ha emergido como una alternativa poderosa frente a métodos tradicionales como la transformada de Fourier. La transformada wavelet se basa en la descomposición de una señal en componentes que varían en escala y tiempo, lo que permite analizar tanto las características de baja frecuencia (tendencias generales) como



las de alta frecuencia (detalles finos) en una misma señal [1].

A diferencia de la transformada de Fourier, que convierte una señal del dominio temporal al dominio de la frecuencia sin retener información temporal, la transformada wavelet mantiene la capacidad de localizar eventos en el tiempo. Esto es especialmente importante para señales no estacionarias, donde las características de la señal pueden cambiar rápidamente. Por ejemplo, en una señal ECG, los picos de QRS, las ondas P y T, y los segmentos entre estos, contienen información esencial que puede perderse o distorsionarse si se aplica un análisis puramente frecuencial.

La transformada wavelet utiliza "onditas" o "wavelets," que son funciones base de corta duración que se desplazan y escalan para analizar la señal en diferentes niveles de detalle. Una característica importante de las wavelets es que pueden ajustarse tanto en tiempo como en frecuencia, permitiendo un análisis multiresolución [2,3]. Esto significa que se puede utilizar una wavelet de mayor escala (baja frecuencia) para capturar tendencias generales de la señal y una wavelet de menor escala (alta frecuencia) para detectar cambios rápidos o eventos transitorios [4].

Entre las wavelets más utilizadas está la wavelet de Morlet, que combina una onda sinusoidal modulada por una función gaussiana. Esta wavelet es particularmente útil para analizar señales que contienen información en diferentes bandas de frecuencia, como el ECG, donde los detalles finos y las tendencias deben ser analizados conjuntamente. Otra wavelet comúnmente utilizada es la Daubechies 4, conocida por su capacidad para representar señales de manera compacta y eficiente, especialmente en aplicaciones que requieren descomposición en múltiples niveles de detalle [5-7].

En este trabajo, se propone el uso de la transformada wavelet continua de Morlet para el análisis de señales ECG, aprovechando su capacidad para proporcionar una resolución ajustable tanto en el dominio temporal como en el frecuencial. Además, se implementa una descomposición de la señal utilizando la ondita Daubechies 4 con un nivel de descomposición de 5, lo que permite una reconstrucción precisa de la señal filtrada y la eliminación efectiva de

artefactos.

Este enfoque no solo facilita la eliminación de artefactos, sino que también preserva las características esenciales de la señal ECG, mejorando la precisión en la detección de anomalías cardíacas y optimizando el proceso de diagnóstico. La capacidad de la transformada wavelet para adaptarse a diferentes escalas y su resolución ajustable la convierten en una herramienta indispensable en la ingeniería biomédica moderna, donde la calidad de las señales y la precisión en su análisis son de máxima importancia.

En conclusión, la aplicación de la transformada wavelet, y en particular de la ondita Morlet y Daubechies, ofrece una solución avanzada para la gestión de señales transitorias en ECG. Este estudio aporta una metodología robusta para mejorar la calidad del análisis de señales biomédicas, contribuyendo a la precisión y fiabilidad del diagnóstico clínico, y abriendo nuevas posibilidades para el desarrollo de tecnologías de monitoreo y análisis de señales más sofisticadas [8].

II. METODOLOGÍA

1. Preparación de la Señal ECG

Se comienza con una señal ECG base obtenida de un registro estándar, la cual se modifica para simular diversas perturbaciones que pueden ocurrir en un entorno clínico real. Los eventos transitorios simulados son:

Simulación de Desconexión de Electrodo: Se introduce un intervalo en el que la señal se pierde completamente, simulando un escenario en el que los electrodos se desconectan temporalmente. Esto se realiza reemplazando una porción de la señal con valores nulos, generando un vacío de datos que puede afectar el análisis.

Desplazamiento de la Señal: Se desplaza temporalmente la señal a lo largo del eje temporal, simulando errores en la sincronización de la adquisición de datos. Este desplazamiento puede ser constante o variable, dependiendo de la magnitud del desplazamiento simulado.

Ruido de Baja Frecuencia: Se añade un componente de ruido de baja frecuencia, que puede representar interferencias ambientales o

fluctuaciones lentas en el sistema de adquisición. Este ruido se introduce como una onda senoidal o un polinomio de bajo orden, ajustando su amplitud para simular diferentes niveles de interferencia.

Ruido de Alta Frecuencia: Se incorpora un componente de ruido de alta frecuencia, típicamente generado por una señal aleatoria o una función de ruido blanco, para simular interferencias rápidas y transitorias que afectan la calidad de la señal ECG.

El segmento de señal seleccionado para el análisis incluye varios ciclos cardíacos, lo que proporciona una visión detallada de cómo los eventos transitorios afectan la señal en un contexto más amplio. La visualización se realiza utilizando gráficos temporales y espectrogramas para observar claramente los efectos de los eventos transitorios.

2. Señal Original:

La señal original, antes de la introducción de eventos transitorios, es examinada para identificar las características fundamentales de la señal ECG, como los complejos QRS, las ondas P y T, y los intervalos PR y QT. Los picos y los componentes del ciclo cardíaco se marcan en la gráfica para establecer una referencia clara para el análisis posterior.

3. Análisis Wavelet:

La transformada wavelet continua de Morlet se aplica a la señal para descomponerla en componentes de frecuencia y tiempo, permitiendo un análisis detallado de los eventos transitorios.

Descomposición de la Señal:

Transformada Wavelet Continua (CWT): La señal es descompuesta en diferentes escalas utilizando la wavelet de Morlet. Esta wavelet es elegida por su capacidad para proporcionar una buena resolución en ambas dimensiones (tiempo y frecuencia). La CWT genera una representación en el dominio tiempo-frecuencia, donde cada coeficiente representa la correlación entre la señal y la wavelet en una escala y posición temporal específicas [9-10].

Análisis de Eventos Transitorios:

Localización y Caracterización: Se analizan los coeficientes de la wavelet para localizar eventos transitorios y caracterizar su duración y frecuencia. Las variaciones en la energía de la señal en diferentes escalas permiten identificar los componentes no deseados, como el ruido y

los artefactos.

Visualización de Componentes: Se utilizan espectrogramas y mapas de calor para visualizar cómo los eventos transitorios afectan la señal a lo largo del tiempo y en distintas frecuencias. Esto facilita la identificación de patrones y la evaluación de la severidad de los eventos transitorios.

Filtrado y Eliminación de Artefactos:

Aplicación de Filtros: Se pueden aplicar filtros en el dominio wavelet para eliminar componentes no deseados. Los coeficientes asociados a los eventos transitorios se ajustan o eliminan, y la señal filtrada se reconstruye para mantener las características importantes mientras se eliminan los artefactos.

Reconstrucción de la Señal: La señal se reconstruye a partir de los coeficientes wavelet ajustados, utilizando la inversa de la transformada wavelet para obtener una señal limpia que preserve los detalles clínicamente relevantes.

4. Evaluación de Resultados:

Comparación con la Señal Original: Se comparan las señales procesadas con la señal original para evaluar la eficacia del método en la eliminación de artefactos. Se analizan métricas de rendimiento, como la preservación de las características cardíacas (picos, ondas, intervalos) y la reducción de ruido.

Validación Clínica: Se realizan pruebas adicionales para validar la precisión del método en un contexto clínico simulado. Esto incluye la evaluación de la capacidad del método para detectar anomalías cardíacas y la verificación de la integridad de la información clínica después de la eliminación de artefactos.

5. Implementación Práctica:

Uso de Herramientas y Software: La implementación del análisis se realiza utilizando librerías de Python específicas para el procesamiento de señales y la transformada wavelet, como PyWavelets y SciPy. Estos paquetes permiten una ejecución eficiente y flexible del análisis wavelet.

Optimización del Proceso: Se exploran diferentes configuraciones de parámetros de la wavelet y niveles de descomposición para optimizar la detección y eliminación de eventos transitorios, ajustando los parámetros para obtener los mejores resultados en diferentes condiciones de señal.

Este enfoque integral proporciona un marco robusto para el análisis y procesamiento de señales ECG, mejorando la calidad del diagnóstico al reducir el impacto de eventos transitorios y preservar la información esencial para la evaluación clínica.

III. RESULTADOS

1. Representación Temporal:

En la Fig. 1(arriba), se muestra la señal ECG temporal con la adición de los cuatro eventos transitorios descritos anteriormente. La señal presenta las siguientes características:

De 0 a 1 seg.: Se observan dos pulsos cardíacos normales, que aparecen claramente en la señal.

De 1 a 2 seg.: Se identifica una desconexión de los electrodos, evidenciada por un intervalo en el que la señal es nula o anómala.

De 2 a 3 seg.: Se observa un pulso desplazado temporalmente, lo que indica un desfase en la adquisición de datos.

De 3 a 4 seg.: Se introduce ruido de baja frecuencia, que aparece como una variación lenta y regular en la señal.

De 4 a 5 seg.: Se observa ruido de alta frecuencia, que se manifiesta como fluctuaciones rápidas y erráticas en la señal.

El gráfico superior muestra la señal original, en la cual los picos transitorios son claramente visibles. Durante los eventos transitorios, se nota un aumento significativo en la amplitud, lo que puede dificultar la interpretación y análisis preciso de la señal.

2. Espectrograma Wavelet:

En la Fig. 1(abajo), se presenta el espectrograma generado mediante la transformada wavelet de Morlet. Esta representación en el dominio tiempo-frecuencia proporciona una visión detallada de cómo se distribuyen los eventos transitorios:

Representación Temporal-Frecuencial: El espectrograma revela claramente los cuatro eventos transitorios en el dominio tiempo-frecuencia. Las áreas más oscuras en el espectrograma indican concentraciones más altas de energía en ciertas escalas y tiempos, correspondientemente a los eventos transitorios en la señal original.

Análisis de Eventos Transitorios: Los eventos transitorios se identifican en el espectrograma

por sus características distintivas en la energía y la distribución en el tiempo y la frecuencia. Esto permite observar cómo cada tipo de evento (desconexión de electrodos, desplazamiento de señal, ruido de baja frecuencia y ruido de alta frecuencia) afecta la señal en diferentes escalas de tiempo y frecuencia.

3. Impacto en la Frecuencia Cardíaca:

En la señal ECG temporal, se buscan las amplitudes máximas para calcular la frecuencia cardíaca. Se observa que los picos causados por los eventos transitorios interfieren con el cálculo preciso de la frecuencia cardíaca. Las perturbaciones introducidas por estos eventos transitorios afectan la precisión del análisis, ya que alteran las mediciones de los picos cardíacos.

Para garantizar un cálculo preciso de la frecuencia cardíaca, es fundamental eliminar los eventos transitorios de la señal. El espectrograma wavelet se utiliza para identificar y recortar estas perturbaciones, permitiendo una evaluación más precisa de la frecuencia cardíaca. La eliminación de los eventos transitorios mejora la calidad del análisis al eliminar las interferencias que podrían distorsionar los resultados.

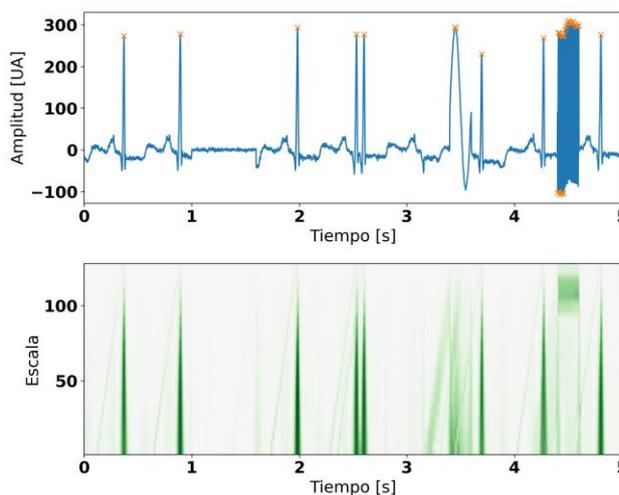


Fig. 1. (arriba) Señal ECG temporal con eventos transitorios. (abajo) Espectrograma wavelet (onda Morlet)

Descomposición y Reconstrucción de la Señal ECG: La Fig. 2 proporciona un análisis detallado de una señal ECG mediante la descomposición wavelet en varios niveles. Se utiliza la wavelet Daubechies 4 con un nivel de descomposición de 5, permitiendo una

evaluación exhaustiva de la señal en diferentes escalas.

Señal Original: En la parte superior de la Fig. 2, se presenta la señal ECG original, donde se pueden observar claramente los picos correspondientes a los latidos cardíacos, así como los eventos transitorios que afectan la señal. Estos eventos incluyen picos de ruido y distorsiones que se introdujeron durante la simulación.

Coefficientes de Detalle y Aproximación: La señal se descompone en coeficientes de detalle y aproximación desde el nivel 0 hasta el nivel 5. Estos coeficientes permiten analizar cómo la energía de la señal se distribuye en diferentes escalas de frecuencia.

Coefficientes de Aproximación (Nivel 0 a Nivel 5): Representan las componentes de baja frecuencia de la señal, capturando la tendencia general y las características de baja frecuencia.

Coefficientes de Detalle (Nivel 1 a Nivel 5): Representan las componentes de alta frecuencia, proporcionando información sobre los detalles finos y las variaciones rápidas en la señal.

Análisis de los Coeficientes:

Nivel 0: En este nivel se observa claramente el ruido de baja frecuencia, que afecta la señal general.

Niveles 4 y 5: Estos niveles muestran los ruidos de alta frecuencia, que aparecen como fluctuaciones rápidas y erráticas en la señal.

Niveles 1 a 4: Los coeficientes en estos niveles revelan la desconexión de los electrodos, evidenciada por interrupciones o variaciones en la señal.

Niveles 2 y 3: Estos niveles muestran el desfase temporal, con una representación de la señal que se ha desplazado temporalmente en relación con su posición original.

Señal Reconstruida: La señal ECG reconstruida a partir de los coeficientes wavelet muestra una buena preservación de las características principales de la señal original. La reconstrucción mantiene los picos cardíacos y las características esenciales de la señal mientras se eliminan los artefactos. Este resultado destaca la capacidad de la transformada wavelet para preservar la información crítica de la señal y mejorar la claridad del análisis al filtrar efectivamente los eventos transitorios.

La descomposición wavelet y la posterior reconstrucción demuestran la eficacia de la transformada wavelet para separar y manejar diferentes componentes de la señal ECG, facilitando la identificación y eliminación de eventos transitorios y mejorando la calidad del análisis.

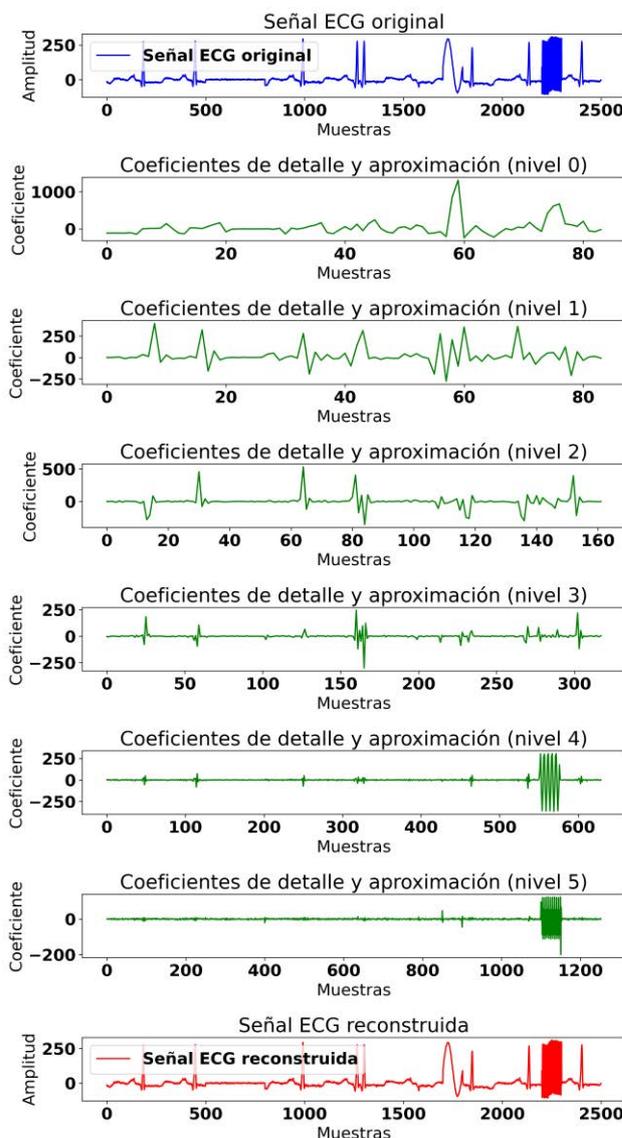


Fig. 2. Descomposición y Reconstrucción de la Señal ECG

IV. DISCUSIÓN

El análisis wavelet ofrece ventajas significativas en comparación con las técnicas tradicionales de análisis de señales, principalmente debido a su capacidad para proporcionar una resolución temporal y frecuencial simultánea. Esta característica es especialmente útil para analizar señales no estacionarias, como las que contienen eventos



transitorios, que presentan variaciones rápidas en diferentes momentos.

1. Ventajas de la Transformada Wavelet:

Resolución Temporal y Frecuencial Ajustable: La transformada wavelet permite ajustar la escala de análisis, proporcionando una resolución variable en el tiempo y en la frecuencia. Esta flexibilidad es crucial para identificar y analizar eventos transitorios de diferentes duraciones y frecuencias, ya que permite examinar tanto los detalles finos como las tendencias generales de la señal en distintas escalas.

Comparación con la Transformada de Fourier: A diferencia de la transformada de Fourier, que ofrece una resolución constante en el dominio frecuencial pero fija en el tiempo, la transformada wavelet proporciona una visión más dinámica de las señales no estacionarias. La transformada de Fourier no es adecuada para capturar la evolución temporal de las frecuencias en señales que cambian con el tiempo, ya que no distingue entre componentes que ocurren en diferentes momentos. En contraste, la transformada wavelet, al descomponer la señal en componentes de diferentes escalas, facilita el análisis de eventos que ocurren en periodos cortos y varían con el tiempo.

2. Aplicaciones Prácticas:

Análisis de Eventos Transitorios: La capacidad de la transformada wavelet para adaptar su resolución según la escala de análisis resulta en una herramienta poderosa para detectar y analizar eventos transitorios. En el contexto de señales ECG, por ejemplo, esto permite identificar artefactos y perturbaciones que podrían pasar desapercibidos con métodos de análisis menos flexibles.

Filtrado y Reconstrucción de Señales: El análisis wavelet facilita la separación de componentes significativos de la señal de los eventos transitorios no deseados. Esto es especialmente relevante para aplicaciones clínicas donde la precisión en la medición de parámetros como la frecuencia cardíaca es crucial. La capacidad de reconstruir la señal después de eliminar los artefactos garantiza que las características fundamentales se mantengan intactas mientras se mejora la calidad del análisis.

3. Limitaciones y Consideraciones:

Selección de la Wavelet y Parámetros de Análisis: La efectividad del análisis wavelet depende de la elección de la wavelet y los parámetros de descomposición utilizados. Aunque la wavelet de Morlet y Daubechies son comúnmente empleadas, su desempeño puede variar según la naturaleza de la señal y los eventos transitorios. La selección adecuada de estos parámetros es esencial para optimizar el análisis y obtener resultados precisos.

Complejidad Computacional: El análisis wavelet, especialmente en múltiples niveles de descomposición, puede ser computacionalmente intensivo. Aunque las herramientas modernas y el software especializado han facilitado este proceso, es importante considerar los recursos necesarios para realizar análisis extensivos y en tiempo real.

En resumen, la transformada wavelet ofrece una ventaja clara sobre métodos tradicionales como la transformada de Fourier al proporcionar una resolución ajustable en el tiempo y la frecuencia. Esta capacidad es esencial para el análisis efectivo de señales no estacionarias y la detección precisa de eventos transitorios. Aunque el método presenta ciertas limitaciones y desafíos, su flexibilidad y capacidad para mejorar la calidad del análisis lo convierten en una herramienta valiosa en el procesamiento de señales complejas.

V. CONCLUSIONES

La aplicación de la transformada wavelet de Morlet se revela como una herramienta altamente efectiva para el análisis de señales que contienen eventos transitorios. Este método proporciona una comprensión detallada de la dinámica de la señal, facilitando la identificación y caracterización precisa de eventos que ocurren en diferentes escalas de tiempo y frecuencia.

1. Efectividad en el Análisis de Señales Transitorias: La transformada wavelet de Morlet destaca por su capacidad para representar señales en el dominio tiempo-frecuencia, lo que permite un análisis más profundo y flexible de las señales no estacionarias. Esta característica es especialmente valiosa en aplicaciones donde los eventos transitorios pueden afectar la calidad y precisión del análisis, como en la monitorización de señales biomédicas y en



sistemas de ingeniería.

2. Aplicaciones en Diversas Áreas: La flexibilidad y precisión de la transformada wavelet de Morlet la convierten en una herramienta valiosa para una amplia gama de aplicaciones científicas y de ingeniería. Su capacidad para ajustar la resolución temporal y frecuencial permite la adaptación a diferentes contextos y requisitos analíticos, mejorando la capacidad de detección y análisis de eventos transitorios en diversas disciplinas.

3. Preservación de la Información Relevante: La transformada wavelet proporciona un enfoque eficaz para detectar y eliminar eventos transitorios sin comprometer la información relevante de las señales. Este proceso asegura que las características esenciales de la señal se mantengan intactas, mientras se minimizan los artefactos y se mejora la calidad del análisis. La capacidad para reconstruir la señal con alta fidelidad tras la eliminación de transitorios subraya la potencia de la técnica en la preservación de datos críticos.

En resumen, la transformada wavelet de Morlet ofrece una solución robusta para el análisis de señales con eventos transitorios, proporcionando una herramienta potente para mejorar la precisión y calidad en diversas aplicaciones científicas y de ingeniería.

REFERENCIAS

- [1] I. Daubechies, Ten Lectures on Wavelets. Philadelphia, PA: SIAM, 1992.
- [2] C. Torrence and G. P. Compo, "A Practical Guide to Wavelet Analysis," Bulletin of the American Meteorological Society, vol. 79, no. 1, pp. 61-78, 1998.
- [3] S. Mallat, A Wavelet Tour of Signal Processing. San Diego, CA: Elsevier, 1999.
- [4] M. Vetterli and J. Kovacevic, Wavelets and Subband Coding. Englewood Cliffs, NJ: Prentice Hall, 1995.
- [5] R. Polikar, "The Story of Wavelets," The Internet Encyclopaedia of Philosophy, 1996.
- [6] D. Donoho, "De-Noising by Soft-Thresholding," IEEE Transactions on Information Theory, vol. 41, no. 3, pp. 613-627, 1995.
- [7] A. Cohen, W. Dahmen, and R. DeVore, "High-dimensional Adaptive Approximation," Acta Numerica, vol. 22, pp. 1-57, 2013.
- [8] S. G. Mallat, "Understanding Deep Convolutional Networks," Foundations and Trends in Machine Learning, vol. 10, no. 1, pp. 1-75, 2017.
- [9] P. J. Flandrin, "Wavelet Analysis and Synthesis of Fractional Brownian Motion," IEEE Transactions on Information Theory, vol. 38, no. 2, pp. 910-916, 1992.
- [10] R. K. Mallat and J. F. Cai, "Signal Decomposition and Reconstruction Using Wavelets," Journal of Computational Physics, vol. 148, no. 1, pp. 30-55, 1999.



Aplicación de bloques de entrada/salida programables a la adquisición de biopotenciales orientada a HMI

Paloma Domínguez

Federico N. Guerrero

Enrique M. Spinelli

Facultad de Ingeniería, Universidad
Nacional de La Plata

LEICI (UNLP-CONICET-CIC)

LEICI (UNLP-CONICET-CIC)

LEICI (UNLP-CONICET-CIC)

La Plata, Argentina

La Plata, Argentina

La Plata, Argentina

paloma.dominguez@alu.ing.unlp.edu
.ar

federico.guerrero@ing.unlp.edu.ar

spinelli@ing.unlp.edu.ar

Resumen— Las interfaces persona-máquina basadas en biopotenciales miden señales eléctricas del cuerpo y las procesan para interpretar la voluntad de la persona que la utiliza, con el fin de comandar otros equipos, usualmente computadoras. Muchos de estos dispositivos se basan en microcontroladores que administran la adquisición de datos, el procesamiento de señales, y la transmisión de comandos. Las alternativas de implementación buscan aumentar la capacidad de cómputo, disminuir el consumo, y el tamaño. En este trabajo se realiza un estudio de factibilidad sobre el uso de una nueva clase de periférico incorporado en el microcontrolador RP2040 que permite el procesamiento de entrada-salida programable independiente de los núcleos del microcontrolador. Este periférico permite recuperar señales de alta resolución (24 bits) al mismo tiempo que proveer un preprocesamiento básico, sin intervención del procesador. Este trabajo demuestra su utilización por primera vez en la aplicación objetivo. Se presenta la implementación del algoritmo que controla estos periféricos, logrando muestrear y procesar señales de electromiografía y electrocardiografía en tiempo real en un prototipo compatible con la aplicación propuesta.

Palabras clave: biopotenciales, interfaz persona-máquina, EMG, ECG, periférico de entrada-salida, PIO.

I. INTRODUCCIÓN

Las interfaces persona-máquina (HMIs por sus siglas en inglés) basadas en biopotenciales miden señales eléctricas del cuerpo y las procesan para interpretar la voluntad de la persona que la utiliza. De esta manera, proveen al usuario un camino alternativo para interactuar con la tecnología [1, 2]. Uno de los desafíos en este campo es lograr dispositivos de altas prestaciones respecto a la calidad de adquisición, que al mismo tiempo puedan ser utilizados dentro del paradigma de los sistemas vestibles: compactos, de bajo consumo y simples de utilizar comparados con equipos clínicos [3, 4, 5].

En la medición de biopotenciales, como las señales de ECG (electrocardiograma) o EMG (electromiograma), se pretende capturar señales de muy baja amplitud y frecuencia con alta precisión y bajo nivel de ruido. Cuando se realizan estas medidas en un entorno clínico, profesionales de la medicina pueden controlar el posicionamiento de electrodos y su instalación, aportando al buen funcionamiento del equipo. En una HMI vestible se prescinde de la asistencia de personal especializado y se busca que la operación sea intrínsecamente robusta.

Una estrategia de adquisición de biopotenciales que permite gran flexibilidad y tolerancia a las condiciones cambiantes de las medidas es el uso de convertidores de alto rango dinámico que hace innecesarias etapas

complejas de procesamiento analógico y permite adaptar en forma más rápida y dinámica al procesamiento a las condiciones de medida; por supuesto, esto se logra a costa de trasladar esa responsabilidad al dominio digital [6].

La implementación de esta estrategia se basa en el uso de convertidores analógico-digitales (ADCs) de tecnología Sigma-Delta que combinan un rango dinámico de 24 bits con un bajo ruido referido a la entrada [7], pudiendo adquirir sin filtrar y con un bajo nivel de amplificación las señales de biopotencial [8]. Sin embargo, el filtrado pasa-altos y el acondicionamiento en general se realiza con filtros y operaciones numéricas en el procesador; adicionalmente, el tratamiento de los datos insume mayores recursos de memoria y cómputo por el simple hecho de que cada muestra es de mayor tamaño [9]. En sistemas de bajo costo y bajo consumo se hace deseable economizar recursos y por lo tanto interesa buscar herramientas para mitigar la carga de procesamiento extra que implica utilizar convertidores sigma-delta.

En este trabajo se presenta un dispositivo que integra un convertidor sigma-delta de altas prestaciones y bajo costo relativo [10] en una plataforma portátil haciendo uso de un periférico especializado de entrada/salida que es capaz de asistir a la captura de muestras de alta resolución sin intervención del procesador. El aporte original se basa en la demostración experimental, por primera vez, del uso de este tipo de bloque (denominado PIO en la literatura del fabricante) en la aplicación de adquisición de biopotenciales orientada a interfaces persona-máquina. Es las secciones siguientes se describe la implementación del dispositivo incluyendo medidas de ECG y EMG.

II. MATERIALES Y MÉTODOS

A. Microcontrolador con bloques de entrada/salida programables

El sistema de adquisición se basa en el microcontrolador RP2040 y la característica que lo hace de especial interés para este trabajo: posee dos bloques de Entrada/Salida Programable (PIO), que permiten implementar interfaces especializadas. Los bloques PIO son programables en el mismo sentido que un procesador y están compuestos por cuatro máquinas de estado, capaces de ejecutar programas secuenciales de manera

independiente para manipular GPIOs y transferir datos.

Cada máquina de estados de los PIOs cuenta con 4 registros propios (2 de desplazamiento y 2 de trabajo) y un FIFO de 8 palabras de 32bits (4 para transmitir y 4 para recibir) que puede utilizarse para intercambiar datos con la memoria del microcontrolador utilizando DMA. Además, es capaz de desplazar datos de sus registros a los pines de entrada/salida de propósito general (GPIO) y responder a varias fuentes de interrupción.

Cada PIO cuenta con una memoria compartida entre las 4 máquinas de estados, que es capaz de almacenar hasta 32 instrucciones en total, por lo que los programas que se desean correr en cada máquina están limitados a esa cantidad de instrucciones. El set de instrucciones es simple, cuenta con 9 instrucciones de 16 bits de longitud, las cuales permiten realizar saltos condicionales y no condicionales, mover valores entre los registros, configurar valores de salida en los GPIOs, leer las entradas, realizar desplazamientos y permanecer en espera hasta que se cumpla una determina condición, como la activación de un pin. Estas características presentes en los bloques y sus prestaciones resultan adecuadas en la implementación de protocolos de comunicación personalizados.

B. Adquisición de biopotenciales Sigma-Delta

La etapa de adquisición se basó en el convertidor analógico-digital ADS13M04 de Texas Instruments [10]. Convertidores basados en la misma tecnología como el ampliamente utilizado ADS1299 se orientan específicamente a la adquisición de biopotenciales; sin embargo, variantes como el mencionado ADS13M04 presenta prestaciones similares a un costo un orden de magnitud inferior. Este dispositivo presenta cuatro canales diferenciales independientes, un amplificador de ganancia programable (PGA) de bajo ruido por cada canal, y una referencia de tensión interna de 1.2V. El ADC no cuenta con un oscilador interno y debe suministrarse una señal de clock externa.

La comunicación de los datos se realiza mediante el protocolo SPI sumada a una señal de salida adicional que se activa cuando los 4 valores de los canales están listos. La señal muestreada es enviada al microcontrolador, en particular al módulo de *Comunicación*, el cual se implementa con 2 máquinas de estados de

los bloques PIO. Luego de la recepción, se encuentran los bloques de *Procesamiento* y *Transmisión inalámbrica*.

El diseño del módulo de *Comunicación* se centró en la sincronización entre el ADS131M04 y el microcontrolador, utilizando dos bloques de Entrada/Salida Programable (PIO 0 y PIO 1):

C. Esquema de adquisición basada en periféricos PIO

El ADC se administra enteramente utilizando recursos independientes del núcleo del procesador: los periféricos PIO y canales de DMA. En particular se implementan estas tareas según el esquema descrito a continuación:

1) Generación del clock para el ADC (PIO 0):

Una máquina de estados del PIO 0 se utiliza para generar el clock de 8.192 MHz necesario para el funcionamiento del ADC. Esta señal de reloj sincroniza las operaciones de muestreo y transmisión de datos del ADS131M04. La máquina de estados que genera este clock está en funcionamiento continuo a una frecuencia de 125MHz. El pin de salida por el que se genera el clock permanece 8 ciclos en nivel alto y 7 ciclos en nivel bajo, dando como resultado un periodo de 120ns y una frecuencia de clock de 8.33MHz. Esta tasa representa una desviación del 1,68% respecto de la tasa nominal a la que debe operar el ADC y debe tenerse en cuenta al momento de procesar las señales.

2) Recepción de datos SPI (PIO 1):

La segunda máquina de estados, perteneciente al PIO 1, se encarga de recibir los datos digitalizados que el ADC envía mediante SPI. El proceso comienza cuando el pin del ADC que indica que los datos están listos se activa realizando una transición de alto a bajo, lo que enciende la máquina de estados para iniciar la recepción de datos. Los datos recibidos son palabras de 24 bits que representan las señales muestreadas.

Durante la recepción, la máquina de estados realiza una extensión de signo para adaptar estos datos de 24 bits a variables de 32 bits, las cuales son más adecuadas para su almacenamiento y procesamiento posterior en el microcontrolador.

Los datos extendidos se colocan en el FIFO RX de la máquina de estados. Para optimizar el proceso y reducir la carga en el procesador, se utiliza un canal de DMA que transfiere

automáticamente estos datos desde el FIFO a una ubicación en la memoria del microcontrolador.

D. Procesamiento y transmisión de datos

Los datos entregados por el periférico PIO se administran en un buffer de tipo ping-pong de tamaño variable que intenta maximizar el tamaño de paquete disponible en un envío UDP, y al mismo tiempo garantizar un envío cada 20 ms con el fin de lograr una percepción de tiempo real en la actualización de las señales. De acuerdo a la tasa de adquisición configurada, se modifica el tamaño del buffer según estos objetivos. Cuando se ha llenado el buffer, el DMA genera una interrupción que activa el núcleo del microcontrolador para iniciar el procesamiento de las señales y su posterior envío a una computadora mediante Wi-Fi.

El módulo de *Procesamiento* se configura para el tipo de señal, como ECG o EMG, y se implementan dos etapas: primero el filtrado Pasa-Altos elimina el offset de continua y reduce el ruido de baja frecuencia, estabilizando la línea base de la señal y centrando las señales en 0. Esto facilita la interpretación y reduce el rango dinámico necesario de etapas posteriores. Luego, se realiza el cálculo de la Envolvente. La envolvente es una representación más suave y lenta de la señal, que refleja de manera clara y sencilla fenómenos como el esfuerzo muscular. Esto facilita la interpretación y el uso de la señal, especialmente en contextos educativos o artísticos. Esta característica se debe al uso futuro esperado como HMI en el contexto del proyecto WIMUMO [11].

Finalmente, El módulo de *Transmisión inalámbrica* se emplea para enviar los datos desde el microcontrolador hacia la computadora por Wi-Fi mediante sockets con el protocolo UDP. Para establecer la comunicación, es necesario conocer el nombre de la red Wi-Fi, la clave de acceso, y la dirección IP del dispositivo receptor.

Si bien el flujo del sistema completo es secuencial, la adquisición de los datos se realiza de forma concurrente junto con el procesamiento y envío, esto se debe al aporte que realizan los bloques programables PIO.

E. Prototipo y set-up experimental

Se construyó un prototipo para realizar medidas con la estrategia de adquisición

propuesta en una plataforma compatible con el paradigma vestible. Se utilizó un factor de forma relativamente compacto, alimentado con una batería de litio tipo 14500 de 800 mAh.

Se cortocircuitaron los canales de medida para realizar medidas de ruido y se añadieron electrodos activos construidos con amplificadores operacionales en configuración de ganancia unitaria en uno de los canales el cual opera a una frecuencia de 4KHz. Las medidas se realizaron asimismo con configuración de ganancia unitaria en el amplificador de ganancia programable interno al ADS131 y se incluyó un filtro pasa-bajos en una frecuencia de 10 kHz suficiente para actuar de filtro anti-aliasing para la frecuencia de adquisición del ADC Sigma-Delta.

III. RESULTADOS

La Fig. 1 muestra una fotografía del prototipo implementado, en un PCB de simple faz

Utilizando los bloques de entrada/salida PIO se logró capturar la señal proveniente del ADS131 con intervención mínima del procesador. La Tabla I resume los elementos utilizados y su función. El uso de los periféricos PIO permite implementar en forma independiente del núcleo del microcontrolador la generación de la frecuencia de referencia para el ADC, y la lectura SPI además de la conversión de datos de 24 bits a 32 bits con extensión de signo en complemento a 2, lo cual marca un diferencial con respecto al uso del DMA y facilita el procesamiento posterior de los datos.

En la Figura 2 se observa el ruido medido del dispositivo, el cual tiene un valor de $8.79\mu\text{Vrms}$. Este valor se encuentra por encima del dato que brinda la documentación del adquirente, que es de $5.35\mu\text{Vrms}$ para una ganancia de 1 y frecuencia de 4KHz, mismas condiciones en las que se obtuvo la medida. Este exceso de ruido puede deberse a ruido de alta frecuencia presente en la alimentación analógica del ADS131, debido a que se utilizó la salida regulada de la placa Raspberry Pico W. Se tomó esta decisión debido a la intención de utilizar el dispositivo como HMI de bajo costo y puede concluirse que la performance no se degrada de manera inaceptable.

Es posible aumentar la ganancia del amplificador de ganancia programable para reducir este ruido referido a la entrada; sin embargo, en la hoja de datos se observa que sólo se lograría reducir hasta aproximadamente

1/3 de su valor. Para lograr medidas de alta calidad sería necesario reducirlo al menos en un orden de magnitud, por lo que se concluye que sería necesario incorporar electrodos activos con ganancia en el orden de 10 veces.

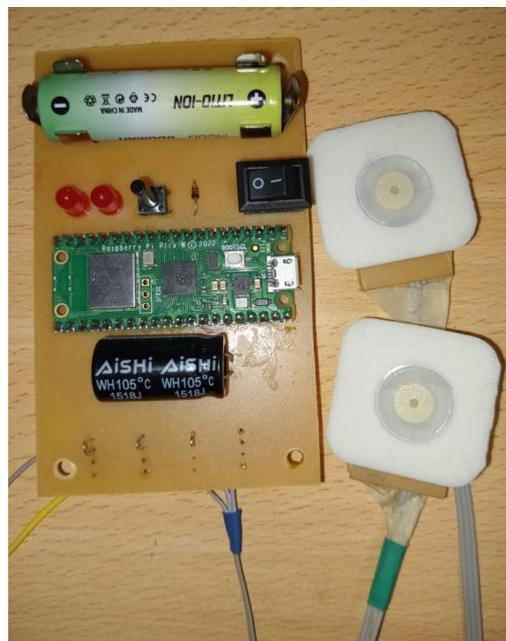


Fig. 1. Fotografía del prototipo implementado, con dimensiones de 10 cm x 6 cm y un canal diferencias con electrodos activos de ganancia unitaria.

Aún en condiciones de ruido levemente excesivo, se tomaron medidas de biopotenciales para verificar el funcionamiento de la cadena de procesamiento y transmisión de datos digitales. El dispositivo implementado logró adquirir exitosamente señales de biopotencial tanto de ECG como EMG. En la figura 3 se muestra una captura de las señales medidas de ECG a las cuales se les aplicó un filtro pasa-banda Butterworth de 2 polos, entre 0.05 Hz y 150 Hz. En la figura 4 se muestra una captura de EMG con la frecuencia ajustada entre 30 y 450 Hz. En ambos casos, se utilizó un electrodo de referencia conectado a $V_{dd}/2$, donde V_{dd} es la tensión de alimentación. A pesar de esto (es decir, de la falta de circuito de DRL), gracias a las características de tamaño y uso alimentado a baterías no hubo presencia de interferencia electromagnética [12].

Se incluyó material adicional sobre el desarrollo del dispositivo en un sitio web del grupo de trabajo [13].

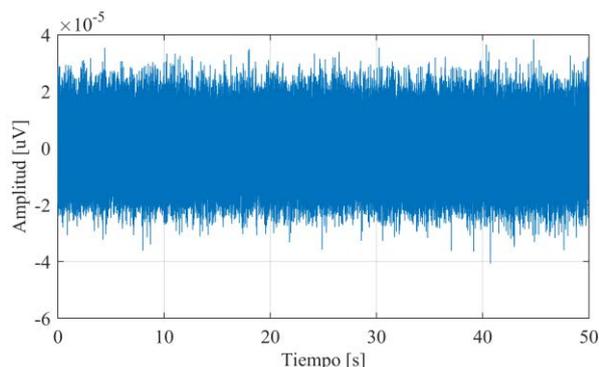


Fig. 2. Medición de ruido obtenido del dispositivo.

TABLA I
RESUMEN DE DISTRIBUCIÓN DE RECURSOS PARA ADQUISICIÓN
DE BIOPOTENCIALES

Periférico	Función
PIO0-SM0	Generación de clock de ADC
PIO1-SM0	Captura de dato por SPI y extensión a 32 bits con signo de datos
DMA1	Transferencia de dato convertido a memoria

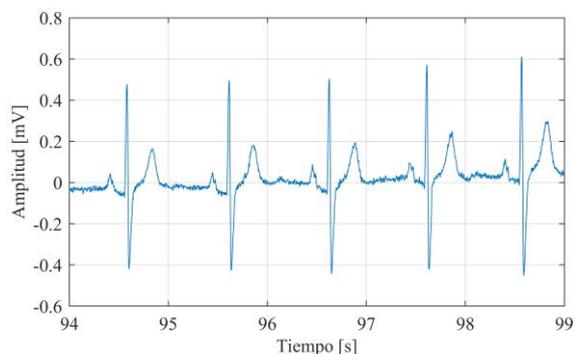


Fig. 3. Segmento de señal de ECG capturada en derivación no-estándar (entre los hombros, con referencia en la cintura).

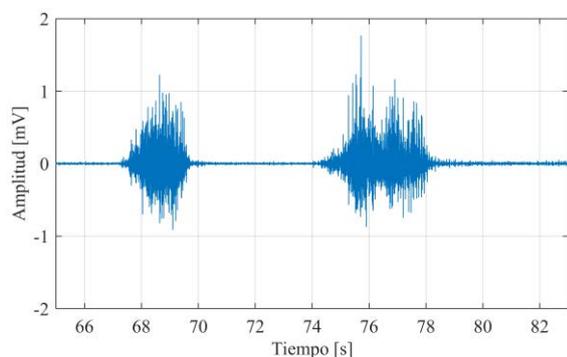


Fig. 4. Segmento de señal de EMG capturada en el antebrazo, incluyendo dos movimientos de flexión de la muñeca.

IV. CONCLUSIONES

En este trabajo, se demostró la viabilidad de utilizar los bloques de Entrada/Salida Programables (PIO) del microcontrolador RP2040 para la adquisición y procesamiento de biopotenciales en tiempo real. Se implementó un prototipo que captura señales con una alta resolución, logrando un desempeño satisfactorio en la medición de señales de electromiografía (EMG) y electrocardiografía (ECG) en diferentes condiciones de prueba.

La implementación evidencia que los bloques PIO pueden gestionar la comunicación entre diferentes dispositivos y realizar operaciones de preprocesamiento sin intervención significativa del procesador principal. Este enfoque no solo optimiza el uso de los recursos del microcontrolador, sino que también permite la adquisición y procesamiento de señales de forma eficiente y con bajo consumo de energía, alineándose con los requerimientos de dispositivos vestibles.

V. RECONOCIMIENTOS

Este trabajo fue financiado por la Beca de Estimulación de Vocaciones Científicas del Consejo Interuniversitario Nacional (EVC-CIN) 2023, el proyecto UNLP PID-I254 y CONICET PIP0323.

VI. REFERENCIAS

- [1] M. Y. M. Naser y S. Bhattacharya, «Towards Practical BCI-Driven Wheelchairs: A Systematic Review Study», *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 1030-1044, 2023, doi: 10.1109/TNSRE.2023.3236251.
- [2] J. Minguillon, I. Volosyak, C. Guger, M. Tangermann, y M. A. Lopez, «Editorial: Brain-Computer Interfaces: Novel Applications and Interactive Technologies», *Front. Comput. Neurosci.*, vol. 16, jun. 2022, doi: 10.3389/fncom.2022.939202.
- [3] L. Cheng, J. Li, A. Guo, y J. Zhang, «Recent advances in flexible noninvasive electrodes for surface electromyography acquisition», *npj Flex Electron*, vol. 7, n.º 1, pp. 1-26, ago. 2023, doi: 10.1038/s41528-023-00273-0.
- [4] M. Al-Ayyad, H. A. Owida, R. De Fazio, B. Al-Naami, y P. Visconti, «Electromyography Monitoring Systems in Rehabilitation: A Review of Clinical Applications, Wearable Devices and Signal Acquisition Methodologies», *Electronics*, vol. 12, n.º 7, Art. n.º 7, ene. 2023, doi: 10.3390/electronics12071520.
- [5] S. Ban et al., «Soft Wireless Headband Bioelectronics and Electrooculography for Persistent Human-Machine Interfaces», *ACS Appl. Electron. Mater.*, vol. 5, n.º 2, pp. 877-886, feb. 2023, doi: 10.1021/acsaem.2c01436.
- [6] D. Berry, F. Duignan, y R. Hayes, «An Investigation of the use of a High Resolution ADC as a Digital Biopotential Amplifier», *4th European Conference of the International Federation for Medical and Biological Engineering*, pp. 0-6, 2009.



- [7] Scott D. Kulchycki, «Continuous-Time Sigma-Delta ADCs», Texas Instruments, SNAA098, 2008.
- [8] E. M. Spinelli y F. N. Guerrero, «Chapter 12 - The Biological Amplifier», en Further understanding of the human machine: The road to Bioengineering, M. E. Valentinuzzi, Ed., New Jersey: World Scientific Publishing Co. Pte. Ltd, 2017, pp. 463-500.
- [9] F. N. Guerrero y E. M. Spinelli, «A simple encoding method for Sigma-Delta ADC based biopotential acquisition systems», Journal of Medical Engineering & Technology, vol. 41, n.º 7, pp. 546-552, oct. 2017, doi: 10.1080/03091902.2017.1366562.
- [10] «ADS130B04-Q1 Automotive, 4-Channel, 32-kSPS, Simultaneous-Sampling, 16-Bit, Delta-Sigma ADC», Texas Instruments, SBASAD2, nov. 2021. [En línea]. Disponible en: <https://www.ti.com/lit/ds/sbasad2/sbasad2.pdf>
- [11] F. N. Guerrero et al., «WIMUMO project: A wearable open device for physiological signals acquisition», jun. 2020, Accedido: 2 de septiembre de 2024. [En línea]. Disponible en: <https://ri.conicet.gov.ar/handle/11336/140757>
- [12] V. A. Catacora, F. N. Guerrero, y E. M. Spinelli, «Size Constraint to Limit Interference in DRL-Free Single-Ended Biopotential Measurements», J. Med. Biol. Eng., vol. 42, n.º 3, pp. 332-340, jun. 2022, doi: 10.1007/s40846-022-00720-9.
- [13] «Aplicación de bloques de entrada/salida programables a la adquisición de biopotenciales orientada a HMI – GIBIC». Accedido: 2 de septiembre de 2024. [En línea]. Disponible en: <https://gibic.ing.unlp.edu.ar/2024/09/02/aplicacion-de-bloques-de-entrada-salida-programables-a-la-adquisicion-de-biopotenciales-orientada-a-hmi/>



Posturógrafo basado en el empleo de Microcontrolador, Instrumentación Virtual e Imágenes

1st Yamel Jesús Almeida Pichardo

Universidad Tecnológica de La Habana “José Antonio Echeverría”

Departamento de Bioingeniería
(CEBIO)

La Habana, Cuba

0000-0003-1356-1766

4th Carmen Busoch Morlán

Universidad Tecnológica de La Habana “José Antonio Echeverría”

Departamento de Automática

La Habana, Cuba

0000-0002-1209-669x

2nd Alejandro Rivas Gamallo

Universidad Tecnológica de La Habana “José Antonio Echeverría”

Departamento de Bioingeniería
(CEBIO)

La Habana, Cuba

0000-0002-2166-0396

5th Carmenchu Regueiro Busoch

Universidad Tecnológica de La Habana “José Antonio Echeverría”

Departamento de Automática

La Habana, Cuba

0000-0001-5101-4005

3rd Luis Antonio Noa Llorens

Universidad Tecnológica de La Habana “José Antonio Echeverría”

Departamento de Bioingeniería
(CEBIO)

La Habana, Cuba

0000-0001-7150-217x

6th Angel Regueiro Gómez

Universidad Tecnológica de La Habana “José Antonio Echeverría”

Departamento de Bioingeniería
(CEBIO)

La Habana, Cuba

0000-0003-2239-1324

Resumen—El análisis del movimiento humano en diferentes posiciones presenta una serie de parámetros que permiten definir su correcta postura; entre ellos la variabilidad del Centro de Masa (CM).

La posturografía estática o dinámica proporciona datos del grado de estabilidad del sujeto y sirve para diferenciar el impacto de cada sistema: vestibular, visual o propioceptivo, en el mantenimiento del equilibrio espacial.

En el siguiente trabajo se incluye la caracterización del diseño de un nuevo posturógrafo donde los registros de la variabilidad del centro de masa y la imagen plantar del sujeto son integrados para permitir un mejor análisis durante estudios diagnósticos.

Palabras clave: posturografía, centro de masa, imagen plantar, diagnóstico

I. INTRODUCCIÓN

A. Posturografía

Diferentes estudios encontraron correlaciones fuertes entre los resultados arrojados por las pruebas de equilibrio clínico y estáticas [1]. La posturografía ha resultado ser valiosa, en casos

como el diagnóstico de disfunciones motoras relacionadas con caídas.

La simplicidad y seguridad de las pruebas posturográficas estáticas las hace atractivas para su uso en la clínica neurológica. Además, parece ser muy sensible a los cambios dentro del sistema sensoriomotor, lo cual es clave en el diagnóstico y seguimiento de muchas enfermedades relacionadas con el sistema nervioso central [2].

Sin embargo, la mayoría de los exámenes posturográficos se limitan a la recolección de datos relacionados con la variabilidad del CM. La distribución del peso de un individuo refleja las características de su estructura ósea, y las características plantares, es decir, la superficie de sus pies puede manifestar irregularidades que pueden arrojar resultados confusos. Los equipos que proporcionan información adicional respecto al peso que aporta cada sección del pie suelen ser muy costosos [3].

El objetivo del nuevo sistema diseñado está relacionado con proporcionar información relevante respecto al contacto de la planta de los pies en la superficie usando procesamiento de imágenes con un menor costo.

II. MATERIALES Y MÉTODOS

A. Posturografía e Imagen plantar

La figura 1 muestra el esquema más general del diseño del nuevo posturógrafo donde se combina el procedimiento convencional con la imagen plantar.



Fig. 1. Combinación de la posturografía convencional con procesamiento de imagen en tiempo real.

B. Arquitectura general del sistema diseñado

El diseño consiste en una plataforma compuesta por los soportes metálicos, el cristal templado donde se ubican las plantas de los pies del sujeto, los sensores de presión y la cámara web (Fig. 2).

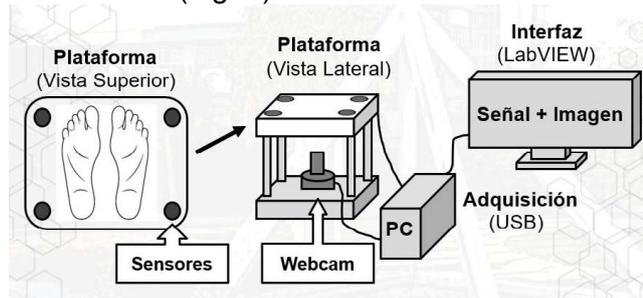


Fig. 2. Arquitectura general del sistema diseñado.

La figura 3 muestra el diagrama en bloques del diseño eléctrico desarrollado.

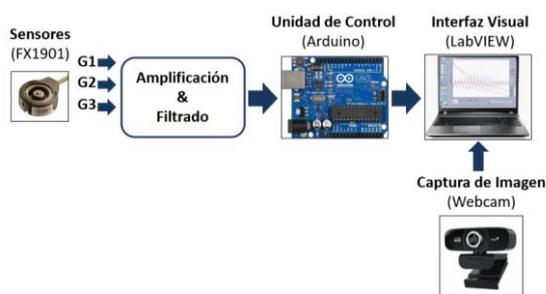


Fig. 3 Diagrama en bloques general del nuevo posturógrafo desarrollado.

- Los sensores

Se utilizaron celdas de carga FX1901 [4], que soportan hasta 200 libras cada una, distribuidas como se muestran en la figura 2, las cuáles

adquieren el peso distribuido del sujeto bajo estudio durante su posicionamiento en la base del sistema.

- Webcam

Para esta aplicación se utiliza una cámara Web Logitech [5] que permite la adquisición de imagen y su posterior procesamiento. La cámara web envía los datos vía puerto USB a PC donde son capturados y registrados a una frecuencia suficiente como para detectar los cambios en el contacto de la planta de los pies con la superficie del soporte de cristal. Esta se encuentra iluminada desde un costado para incrementar el contraste de colores, lo que permite mejor resolución en el procesamiento desarrollado.

- Tarjeta de adquisición

Como tarjeta de adquisición se utilizó el ATmega328P ubicado en un kit de desarrollo Arduino UNO [6]. A través de una API llamada NI-VISA (v5.0.3), se comunica a través del puerto serie con la interfaz de usuario desarrollada de manera bidireccional.

- Interfaz de usuario

La interfaz fue diseñada con formato genérico en LabVIEW 2022 Q3 [7] con la ayuda del kit de adquisición y el manejo de imágenes que brinda *National Instruments*.

Esta interfaz se divide en dos: interfaz principal para adquisición y registro de las variables, e interfaz de revisión (Figuras 4 y 5 respectivamente).



Fig. 4. Interfaz principal de adquisición y registro simultáneo de sensores e imagen.

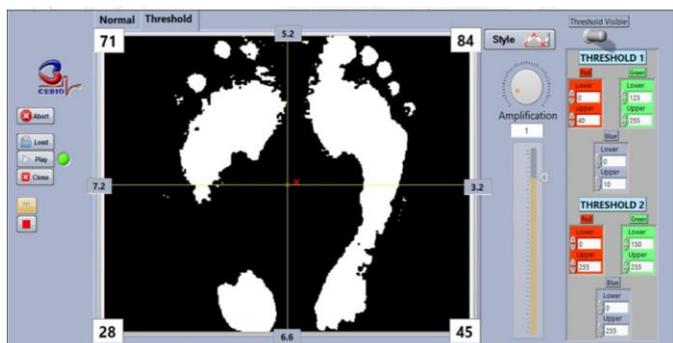


Fig. 5. Interfaz de revisión de registros y análisis de imagen.

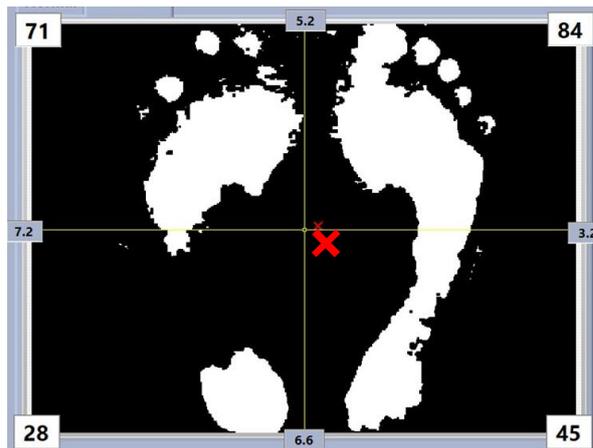


Fig. 6. Resultado del procesamiento de la imagen plantar del sujeto en posición de equilibrio.

El sistema permite la representación visual de las variabilidades del CM del sujeto y la variabilidad de la superficie del pie que está en contacto con la plataforma.

- Procesamiento de Imagen

Como se observa en la figura 5, la interfaz de revisión procesa la imagen adquirida por la cámara y la hace pasar por un filtro que tiene un umbral doble configurable donde quedan de un color aquellos píxeles que cumplan con el criterio de ambos umbrales mientras que los otros quedan de otro color (blanco y negro).

Luego, se divide la imagen resultante en cuatro cuadrantes y se evalúa la cantidad de píxeles blancos (pasaron el umbral de contacto), se cuantifican y se muestran el total en cada una de las esquinas de la figura representada. Esto permite un análisis comparativo posterior.

III. RESULTADOS Y DISCUSIÓN

A partir del registro del CM y la imagen plantar se puede implementar un protocolo para detectar deformaciones que vincule e integre ambos tipos de datos.

Experimentalmente se le pidió a un grupo de 10 sujetos que realizaran movimientos de equilibrio estático e inclinación hacia los cuatro puntos cardinales. Luego se procedió a hacer un análisis comparativo y correlacionar el centro de masa con la superficie en contacto.

La figura 6 muestra el resultado final luego de llevar a cabo el protocolo en un sujeto supuestamente sano en la posición estándar de equilibrio, sin ninguna orden de inclinación.

Desde este momento se puede observar que el sujeto presenta cierta malformación en el pie derecho, la diferencia cuantitativa y cualitativa entre los píxeles del cuadrante inferior derecho y el izquierdo es notable.

El CM se encuentra ligeramente desplazado hacia la izquierda, lo cual se corresponde con el resultado de la imagen.

Las figuras 7-9 muestran los resultados obtenidos cuando se le solicita al sujeto que se incline hacia diferentes posiciones.

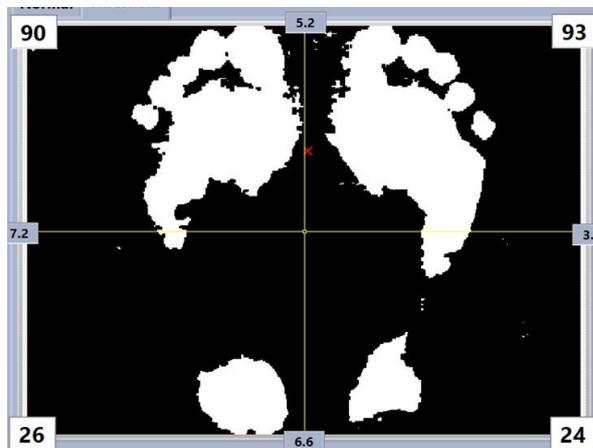


Fig. 7. Resultado del procesamiento de la imagen plantar del sujeto en posición de inclinación hacia adelante.

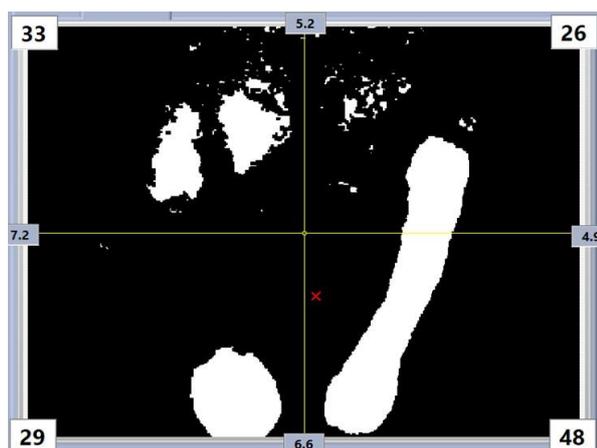


Fig. 8. Resultado del procesamiento de la imagen plantar del sujeto en posición de inclinación hacia atrás.

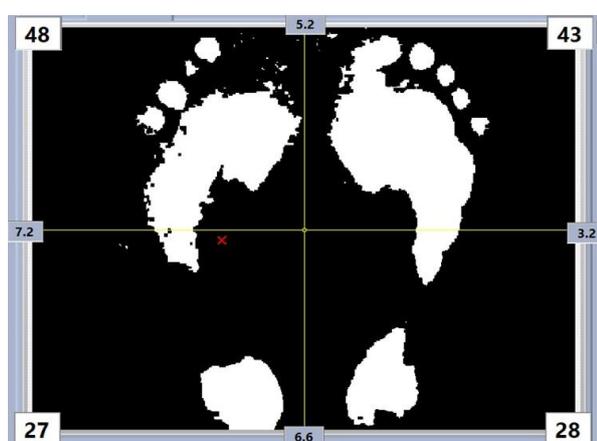


Fig. 9. Resultado del procesamiento de la imagen plantar del sujeto en posición de inclinación hacia la derecha.

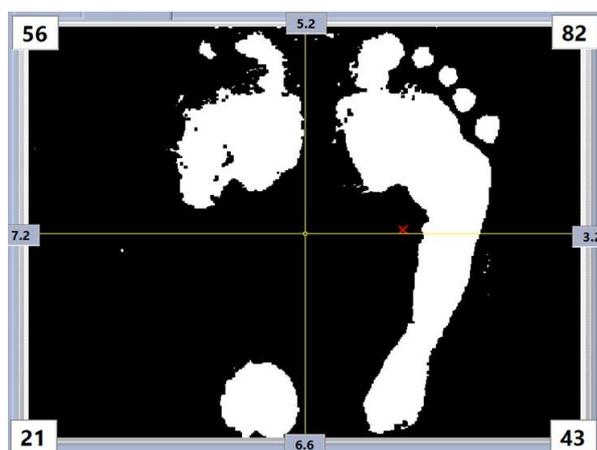


Fig. 6. Resultado del procesamiento de la imagen plantar del sujeto en posición de inclinación hacia la izquierda.

Es notable observar en la figura 9 que, incluso cuando la inclinación es hacia el pie que presenta la deformidad, la cantidad de superficie en la zona inferior del pie (en píxeles) es similar a la del otro pie. Este efecto es recurrente en el resto de las imágenes donde

siempre existe superioridad cuantitativa del pie izquierdo en la zona inferior respecto al pie derecho en la misma zona.

La figura 8 arroja un resultado interesante, debido a la deformidad del pie afectado (derecho), para mantener el equilibrio, el sujeto recurre a un apoyo mayor en la zona superior del pie, incluso mayor a la misma zona del otro pie cuando el sujeto está inclinado hacia el pie izquierdo.

El nuevo sistema permite identificar malas posturas y problemas osteo-musculares en los individuos estudiados cuyas extremidades supuestamente son normales, a través del registro de la variabilidad del CM y la distribución de su peso obtenida por análisis digital de imágenes de las plantas de los pies.

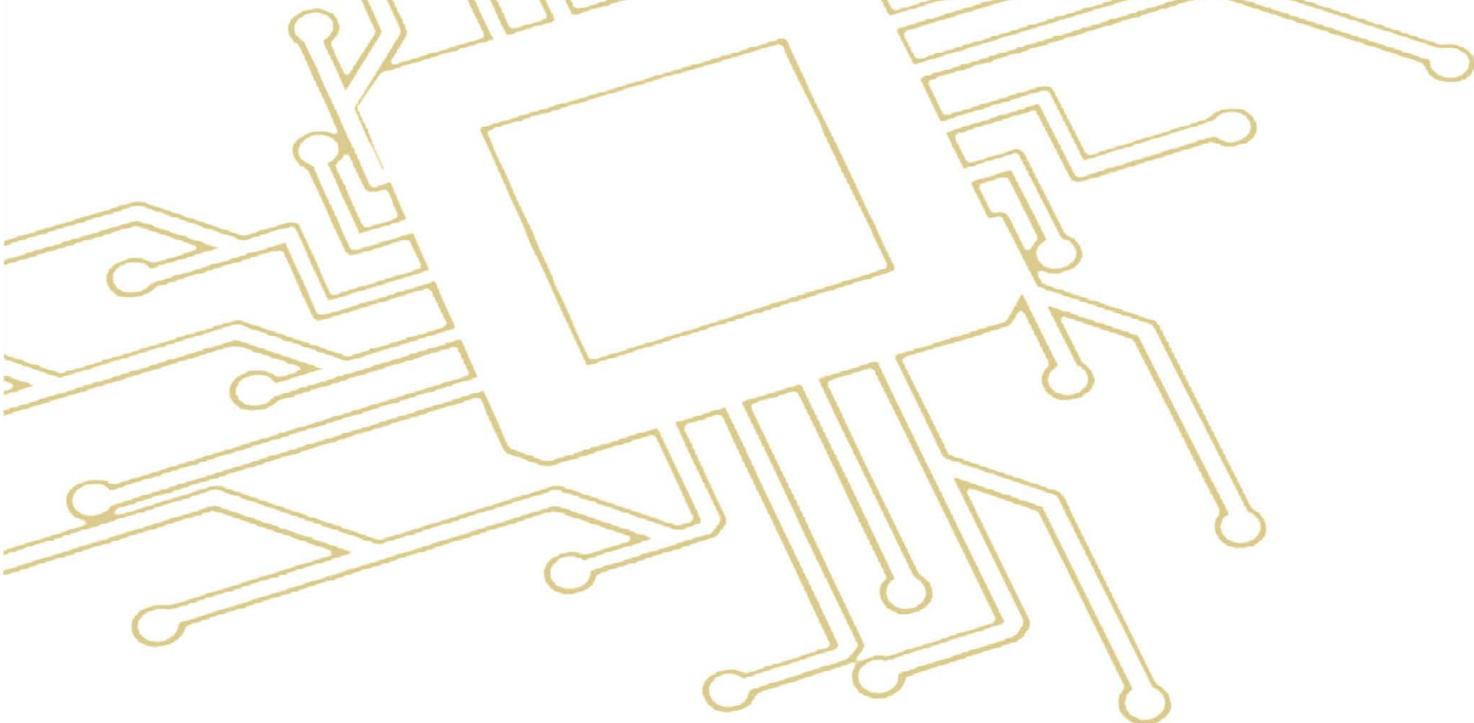
Además, esta nueva propuesta permite ajustar la altura de la colocación geométrica de las prótesis en un individuo, de forma que quede estable y balanceado en su correcto posicionamiento minimizando posibles afectaciones a estos individuos con pérdida de extremidades inferiores.

CONCLUSIONES

El sistema, luego de que se realizaran varios experimentos, arrojó resultados cuantitativos y cualitativamente congruente en los sujetos analizados. Así, se puede comprobar que el sistema es útil para el análisis de patologías relacionadas con el equilibrio y anomalías anatómicas asociadas a la planta de los pies.

REFERENCIAS

- [1] S. Sebastia-Amat, J. Tortosa-Martínez, and B. Pueo, "The use of the static posturography to assess balance performance in a Parkinson's disease population," *International journal of environmental research and public health*, vol. 20, no. 2, p. 981, 2023.
- [2] J. W. Błaszczyk, J. Cieślińska-Świder, and R. Orawiec, "New methods of posturographic data analysis may improve the diagnostic value of static posturography in multiple sclerosis," *Heliyon*, vol. 7, no. 2, 2021.
- [3] "Kinefis Podia baropodometric platform." fisaude. <https://www.fisaude.eu/kinefis-podia-baropodometric-platform-pressure-platform-for-static-dynamic-and-posturological-analysis-includes-walkway-and-camera-p-49103.html> (accessed 16/09/2024, 2024).
- [4] *Compression Load Cell*. [Online]. Available: <https://docs.rs-online.com/1060/A700000006772478.pdf>
- [5] *Logitech StreamCam*, Amazon, 2024. [Online]. Available: <https://www.amazon.es/Logitech-StreamCam-Streaming-Contenido-Versatilidad/dp/B07W4DHNBF>.
- [6] *Arduino UNO R3*, Amazon. [Online]. Available: <https://www.amazon.com/-/es/A000066-Arduino-Uno-R3-Microcontrolador/dp/B008GRTSV6>.
- [7] *LabVIEW 2022 Q3*. (2022). NI. [Online]. Available: <https://www.ni.com/en/support/downloads/software-products/download.labview.html?srsltid=AfmBOopbZEehIZIaeWLM8Rkz1w-KffVNRH3WVjLb4EqL2ZJqGh1oVamP#>



Eje Temático

Aplicaciones en Conectividad y Comunicaciones



Reprogramación de firmware de Arduino a través de transceptor XBEE

Rodrigo E. Russo

Departamento de Ingeniería
Electrónica y Computación - ICyTE
UNMdp

Mar del Plata, Argentina
rodrigo.e.russo@fi.mdp.edu.ar

Melisa G. Kuzman

Departamento de Ingeniería
Electrónica y Computación - ICyTE
UNMdp

Mar del Plata, Argentina
melisakuzman@fi.mdp.edu.ar

Walter A. Gemin

Departamento de Ingeniería
Electrónica y Computación - ICyTE
UNMdp

Mar del Plata, Argentina
agemin@fi.mdp.edu.ar

Juan M. López

Departamento de Ingeniería
Electrónica y Computación - ICyTE
UNMdp

Mar del Plata, Argentina
juanml@fi.mdp.edu.ar

Esteban A. Zapirain

Departamento de Ingeniería
Electrónica y Computación - ICyTE
UNMdp

Mar del Plata, Argentina
estebanzapirain@fi.mdp.edu.ar

Resumen—Este trabajo muestra el proceso de actualización de firmware de placas de desarrollo Arduino Nano, conectadas a través de una red inalámbrica XBEE con topología BUS. Se detallan las configuraciones necesarias en los módulos XBEE, las modificaciones en el bootloader y código que es necesario agregar al fuente de Arduino, para su correcto funcionamiento.

Index Terms—Arduino, XBEE, IoT, RF, bootloader

I. INTRODUCCIÓN

Muchas veces es necesario actualizar el firmware de microcontroladores que están ubicados en lugares de difícil acceso y como en este caso particular, pertenecen a una red de dispositivos que se comunican a través de mensajes en formato ASCII. Este trabajo presenta una solución a este problema, a través de una red inalámbrica conformada por módulos comerciales XBEE PRO S2C como el de la Fig. 1. Estos operan en la banda de 2.4 GHz y poseen un alcance de 3200 metros en espacios abiertos [1].

El esquema utilizado en este trabajo consiste en varias placas de desarrollo Arduino Nano con microcontrolador ATmega328p (Fig. 2), conectadas a un módulo XBEE por medio de su interface serie, posibilitando la comunicación inalámbrica entre ellos y una computadora conectada a un módulo XBEE que funciona como programador. Cada conjunto conformado por un módulo XBEE y un Arduino Nano o una computadora lo denominamos Endpoint. Esto se observa en la Fig. 3.

La idea es que se pueda reprogramar a cada uno de los Arduino Nano reemplazando el cable USB utilizado tradicionalmente, por un enlace inalámbrico realizado



Fig. 1. Módulo XBEE PRO S2C

por los módulos XBEE como se observa en la Fig. 4. También es necesario una manera de direccionar el módulo Arduino a reprogramar. Esto se analizará en detalle más adelante en este artículo.

II. CONFIGURACIÓN DE MÓDULOS XBEE

En este caso se utilizaron módulos XBEE PRO S2C XBP24CZTSIT-004 y por medio de la aplicación Digi XCTU [2] se carga el firmware 802.15.4 TH PRO, Fig. 5, que permite armar una red tipo BUS para enviar y recibir mensajes serie tipo caracteres ASCII.

Además, permite encriptar los mensajes para evitar la intromisión de terceros, que con un módulo XBEE, podrían enviar mensajes con intenciones maliciosas.

Los parámetros mínimos de configuración necesarios para este trabajo se muestran en la Tabla I.

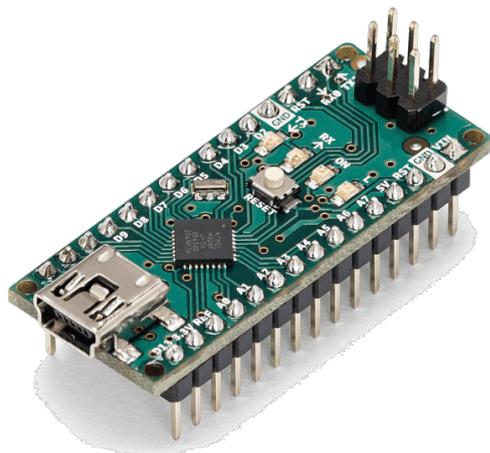


Fig. 2. Módulo Arduino Nano

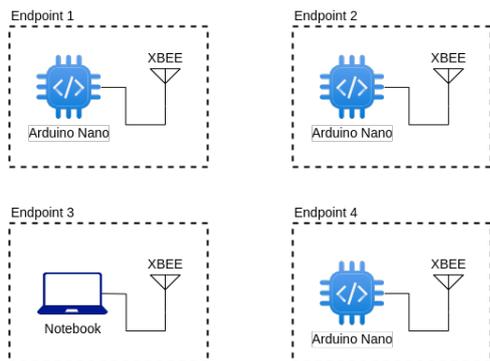


Fig. 3. Topología de la red

Aplicada esta configuración a cada uno de los módulos que van a intervenir en la red, se obtiene una topología BUS inalámbrica. Los caracteres que se envían a través de uno de ellos, llegan al resto de los módulos que conforman la red, dentro de su alcance.

De las pruebas realizadas surge que la configuración mostrada en la Fig. 3 para reprogramación de los Arduino Nano, solo funciona cuando hay únicamente dos módulos XBEE activos. Es decir que al momento de reprogramar, la red inalámbrica debe estar compuesta por el programador con su correspondiente módulo XBEE y el Arduino Nano conectado a otro módulo XBEE.

III. CONFIGURACIÓN DE MÓDULOS ARDUINO

III-A. Bootloader

Las placas de desarrollo Arduino Nano cuentan con un bootloader que permite la carga del firmware a través de un puerto UART utilizando un protocolo denominado stk500 [3].

El proceso de carga del firmware consiste en reiniciar el microcontrolador y enviar el código compilado palabra por palabra. Para este proceso se cuenta con una ventana de tiempo después del reinicio para comenzar la transferencia.

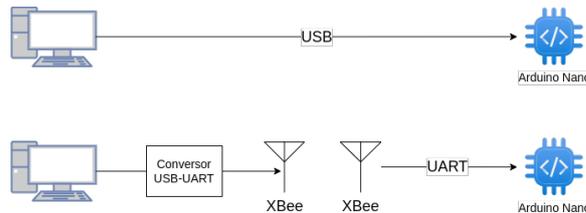


Fig. 4. Reemplazo de cable USB por enlace inalámbrico

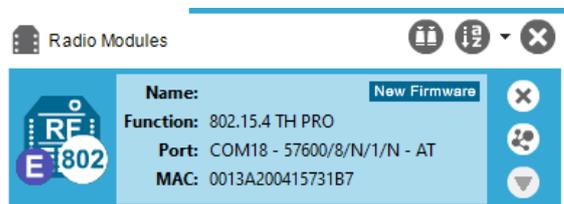


Fig. 5. Aplicación XCTU – Configuración de firmware

El bootloader utilizado debido a su simplicidad y disponibilidad de código fuente es el Optiboot [4]. En el mismo es posible cambiar la velocidad de carga del firmware a través de la interface serie, donde se debe hacer coincidir con la velocidad de puerto serie del XBEE de 57600 baudios, para que la comunicación entre ambos sea exitosa.

Una vez compilado el bootloader, el archivo con extensión hex generado, debe ser grabado en el microcontrolador por medio de un programador como el USBasp [5].

La velocidad de 57600 baudios resultó ser la más exitosa logrando un balance entre máxima velocidad y confiabilidad de transmisión, a diferencia de la velocidad por defecto que es de 115200 baudios.

III-B. Hardware

Las conexiones entre los módulos XBEE y Arduino Nano que conforman un Endpoint, se observan en la Fig. 6.

El microcontrolador ATmega328P, no tiene una instrucción para hacer un RESET por software, por lo que se implementa a través de una salida digital que mantiene por un tiempo en bajo la entrada de RESET por hardware. Con el objetivo de generar este tiempo es que se incorpora la red RC entre la salida D2 y RESET.

El Dip switch conectado entre D9 y D12 proporciona la dirección en binario del Endpoint.

III-C. Firmware

Para reprogramar el microcontrolador, se implementan cuatro comandos que se describen en la Tabla II. En primer lugar, los comandos PROG y NORM consisten en alternar entre los modos de programación y operación normal del equipo. En el modo PROG, el XBEE



Fig. 7. Diagrama de flujo de reconfiguración de PAN ID de XBEE

Tabla II
COMANDOS NECESARIOS PARA INICIAR LA REPROGRAMACIÓN DE ARDUINO

Comando	Descripción
PROG<ID>	Cambia el PAN ID del XBEE a uno específico para la reprogramación (en este caso, PAN ID 9999)
NORM<ID>	Vuelve al PAN ID original para volver al funcionamiento normal del equipo
RESE<ID>	Reinicia el Arduino para comenzar el proceso de reprogramación
STRT<ID>	Comando que envía el Arduino al inicializarse para comunicarse correctamente

Tabla I
PARÁMETROS DE CONFIGURACIÓN DEL FIRMWARE DE XBEE

Parámetro	Valor	Descripción
+++		Ingreso al modo programación (comandos AT)
ID	0x2345	Identificador de Red de 4 caracteres hexadecimales
EE	Enable [1]	Habilitar encriptación AES
KY	0x1234...	Clave de encriptación AES de 32 caracteres hexadecimales
BD	57600	Velocidad en baudios de interfaz de serie.

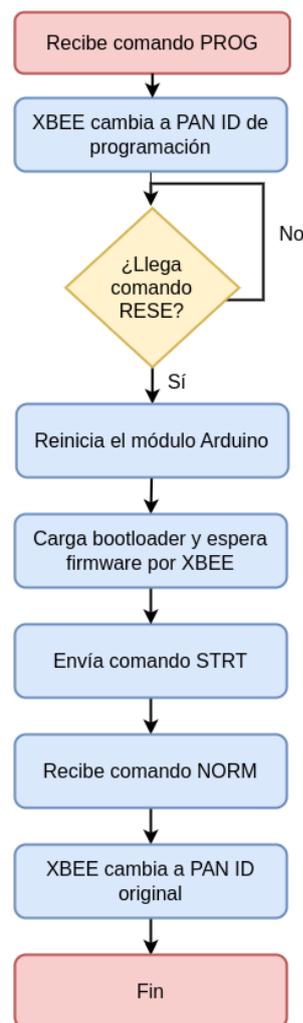


Fig. 8. Diagrama de flujo de reconfiguración de PAN ID de XBEE

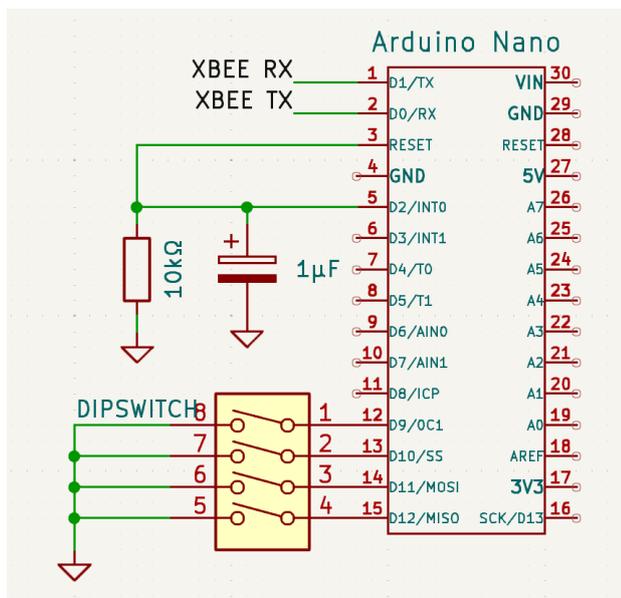


Fig. 6. Conexiones entre módulo XBEE y Arduino Nano

correspondiente al equipo indicado con ID, debe ser reconfigurado para comunicarse en otro PAN ID para evitar interferencias con el resto de los equipos. El proceso de reconfiguración, se realiza utilizando comandos AT implementados en el firmware de XBEE y consiste en el proceso mostrado en la Fig. 7. Luego, el comando NORM, lo vuelve a su PAN ID original para volver a su funcionamiento normal. El comando RESE, dispara el reinicio del microcontrolador para habilitar la carga del nuevo firmware como se describe en la sección IV. Finalmente, el comando STRT se envía cada vez que se inicia el dispositivo y esto es utilizado para detectar el final de la reprogramación.

El firmware del módulo Arduino, además, debe contemplar el proceso de reconfiguraciones de las diferentes PAN ID y los comandos mencionados en la Tabla II. Esto se puede observar en el diagrama de flujo de la Fig. 8.

IV. SOFTWARE DEL PROGRAMADOR

Para que la actualización del firmware sea ágil, se utiliza un script para reprogramar los módulos Arduino. El XBEE del programador debe configurarse en el mismo PAN ID que el resto de los dispositivos y enviar el comando PROG. A continuación, se reconfigura el XBEE del programador para que pueda comunicarse en el PAN ID que el dispositivo a reprogramar. Luego, envía el comando RESE y, seguido, el programa en formato hex a través del protocolo stk500. Finalmente, una vez completada la reprogramación, se envía el comando NORM para que el dispositivo vuelva a su funcionamiento normal. Este procedimiento se ilustra en la Fig. 9.

El programador, para cada comando enviado a través de XBEE, espera una confirmación proveniente del

módulo Arduino. En cada caso, si esta confirmación no llega, se reintenta el comando o, finalmente, se reinicia el programa.

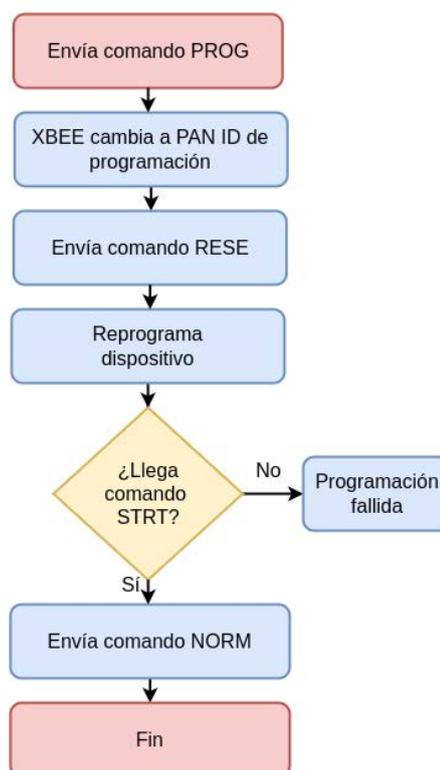


Fig. 9. Diagrama de flujo de reconfiguración de PAN ID de XBEE

V. CONCLUSIONES

Se logró implementar una red inalámbrica que permite reprogramar módulos Arduino con microcontroladores ATmega328p, aunque también es posible con otros módulos Arduino como los que tienen ATmega2560, siguiendo los mismos pasos que se describen en este artículo, utilizando el bootloader que corresponde a este microcontrolador. Esto es particularmente útil para cuando los nodos de una red se encuentran distribuidos y en lugares de difícil acceso.

REFERENCIAS

- [1] Digi. (2024) Sitio oficial módulo xbee. [Online]. Available: <http://digi.com/products/embedded-systems/digi-xbee/rf-modules/2-4-ghz-rf-modules/xbee-802-15-4>
- [2] Digi. (2024) Sitio oficial software xctu. [Online]. Available: <https://www.digi.com/products/embedded-systems/digi-xbee/digi-xbee-tools/xctu>
- [3] Arduino. (2024) Sitio oficial bootloader arduino. [Online]. Available: <https://docs.arduino.cc/retired/hacking/software/Bootloader/>
- [4] Optiboot. (2024) Repositorio de github de bootloader optiboot. [Online]. Available: <https://github.com/Optiboot/optiboot>
- [5] T. Fischl. (2024) Sitio oficial de módulo programador usbasp. [Online]. Available: <https://www.fischl.de/usbasp/>



ANÁLISIS DE PROTOCOLO MQTT CON BROKER MOSQUITTO APLICADO EN DISPOSITIVOS IOT MEDIANTE SISTEMAS DE CONTROL Y TELEMETRÍA DE INVERNADEROS

Daniel Eduardo Oneddu

Rubén Ricardo Urquijo

Nancy Beatriz Ganz

Marcelo Julio Marinelli

Facultad de Ciencias
Exactas, Químicas y
Naturales; Universidad
Nacional de Misiones

Departamento de
Informática; Facultad de
Ciencias Exactas, Químicas
y Naturales; Universidad
Nacional de Misiones

Facultad de Ingeniería;
Universidad Nacional de
Misiones

Departamento de Informática;
Facultad de Ciencias Exactas,
Químicas y Naturales;
Universidad Nacional de
Misiones

Posadas, Misiones,
Argentina

Posadas, Misiones,
Argentina

Oberá, Misiones,
Argentina

Posadas, Misiones,
Argentina

oneddu@fceqyn.unam.e
du.ar

rubenurquijo@fceqyn.unam.
edu.ar

nancyganz@fceqyn.unam.
edu.ar

marcelomarinelli@fceqyn.unam
.edu.ar

Resumen— En los últimos años, IoT ha revolucionado la agricultura, especialmente en invernaderos, mejorando la gestión y eficiencia de producción. El protocolo MQTT es destacado por su eficacia y fiabilidad en la comunicación entre dispositivos IoT. Este trabajo de investigación se centró en analizar el protocolo MQTT, su interacción con el Broker Mosquitto y su aplicación en sistemas de telemetría y control de invernaderos. Se evaluó la eficiencia en el consumo de energía y ancho de banda del protocolo en la optimización de invernaderos. Se desarrolló un analizador de tráfico MQTT con el Broker Mosquitto en dispositivos IoT, revisando trabajos previos, identificando requerimientos y diseñando un prototipo en Python ejecutado en un servidor Raspberry Pi. Las pruebas en entornos reales evaluaron su rendimiento, comparándolo con soluciones existentes. Los resultados subrayaron los beneficios del uso de MQTT en IoT y ofrecieron recomendaciones para su implementación.

Palabras clave: IoT, MQTT, Mosquitto, Node Red, Invernadero.

I. INTRODUCCIÓN

En los últimos años, el Internet de las Cosas (IoT) revolucionó diversos sectores, incluyendo la agricultura. La aplicación de tecnologías IoT

en invernaderos permitió optimizar los procesos de control y monitoreo, mejorando la eficiencia en la producción agrícola. En este contexto, el protocolo MQTT (Message Queuing Telemetry Transport) se destacó como una solución eficiente y confiable para la comunicación entre dispositivos IoT.

El trabajo final de investigación presentado tuvo como objetivo explorar en profundidad el análisis del protocolo MQTT, en combinación con el uso del Broker Mosquitto, para su aplicación en sistemas de control y telemetría de invernaderos. Se buscó comprender cómo este protocolo, conocido por su bajo consumo de energía y ancho de banda, pudo ser implementado de manera efectiva en un entorno agrícola para optimizar la gestión de los invernaderos y mejorar la producción de cultivos.

La investigación se centró en examinar los aspectos técnicos y funcionales del protocolo MQTT, así como en analizar las características y ventajas del Broker Mosquitto como plataforma de mensajería. También se exploraron casos de estudio previos y proyectos relacionados para evaluar los resultados obtenidos en la aplicación de MQTT en sistemas de control y telemetría en invernaderos.

Mediante la recolección y análisis de datos

provenientes de sensores y dispositivos IoT distribuidos en el invernadero, se esperaba obtener información precisa y en tiempo real sobre diversos parámetros ambientales y condiciones de cultivo. Esto permitió una toma de decisiones más informada, facilitando la implementación de estrategias de riego, iluminación y climatización más eficientes y ajustadas a las necesidades específicas de los cultivos.

El trabajo de investigación propuso una metodología integral que abarcó desde la investigación teórica y el diseño de la arquitectura del sistema, hasta la implementación práctica y la evaluación de resultados. Asimismo, se estableció un cronograma detallado y se identificaron los recursos necesarios para llevar a cabo el proyecto en su totalidad.

El análisis del protocolo MQTT con el Broker Mosquitto aplicado en dispositivos IoT para sistemas de control y telemetría de invernaderos presentó un potencial significativo para optimizar la producción agrícola, maximizar el uso de recursos y contribuir a un enfoque más sostenible y eficiente en la agricultura moderna. Los resultados obtenidos a partir de este estudio sirvieron como base para futuras investigaciones y aplicaciones prácticas en el ámbito de la agricultura inteligente.

A. Protocolo MQTT

El protocolo MQTT es ampliamente utilizado en aplicaciones IoT debido a su bajo consumo de ancho de banda y la facilidad de implementación en dispositivos con recursos limitados. En este contexto, el presente trabajo de investigación busca establecer una base de conocimientos para el análisis de protocolo MQTT utilizando el broker Mosquitto aplicado en dispositivos IoT mediante sistemas de control y telemetría de invernaderos[11].

El protocolo MQTT es un protocolo de mensajería ligero diseñado para aplicaciones IoT. MQTT se utiliza para intercambiar datos entre dispositivos IoT y un broker de mensajes, que actúa como intermediario entre los dispositivos. Los mensajes se envían a través de un canal de comunicación denominado topic, que funciona como un canal de publicación y suscripción de mensajes[11], [12].

B. Arquitectura de publicación/suscripción MQTT

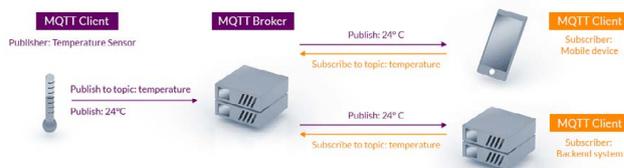


Figura 1: Arquitectura MQTT[13]

Como se puede apreciar en la figura 1, la arquitectura de publicación/suscripción MQTT es un modelo de comunicación ampliamente utilizado en IoT. A continuación, se describen sus claves y cómo funcionan:

1. **Broker MQTT:** Este es el núcleo del modelo MQTT. Actúa como un intermediario que recibe mensajes de los publicadores y los distribuye a los suscriptores. El broker se encarga de gestionar la red y asegurar que los mensajes lleguen de manera eficiente a los destinos correctos.
2. **Publicadores (Publishers):** Son los clientes o dispositivos que envían mensajes al broker. Estos mensajes pueden ser datos de sensores, comandos de control, o cualquier otra información. Los publicadores envían mensajes a un 'topic' específico.
3. **Suscriptores (Subscribers):** Son clientes o dispositivos que se registran en el broker para recibir mensajes de uno o más 'topics'. El broker envía los mensajes correspondientes a estos suscriptores cuando hay datos nuevos en los topics a los que están suscritos.
4. **Topics:** Son las etiquetas o categorías bajo las cuales se clasifican los mensajes. Los publicadores envían mensajes a un topic específico y los suscriptores reciben mensajes de los topics a los que están suscritos.
5. **Protocolo de Comunicación:** MQTT utiliza un modelo basado en TCP/IP para la transmisión de datos. Es conocido por su ligereza y eficiencia, lo que lo hace ideal para dispositivos con recursos limitados o conexiones de red inestables.
6. **Calidad de Servicio (QoS):** MQTT ofrece diferentes niveles de QoS para garantizar la

entrega de mensajes de acuerdo con las necesidades de la aplicación. Estos niveles van desde el envío de un mensaje sin confirmación hasta garantizar la entrega con confirmaciones múltiples.

7. Seguridad: Aunque MQTT es un protocolo ligero, puede integrarse con mecanismos de seguridad como SSL/TLS para la encriptación de mensajes y autenticación de clientes.

Este modelo es altamente escalable y flexible, lo que permite su uso en una amplia gama de aplicaciones, desde sistemas domésticos inteligentes hasta redes industriales complejas. La eficiencia en el uso del ancho de banda y la capacidad de funcionar en redes con latencia alta o intermitente son algunas de las razones de su popularidad en el mundo del IoT[13], [14].

C. Broker Mosquito

Mosquito es uno de los brokers MQTT más populares y ampliamente utilizados en aplicaciones IoT. Es una implementación de código abierto del protocolo MQTT y está diseñado para ser altamente escalable y eficiente en términos de recursos. El broker Mosquito es capaz de manejar grandes volúmenes de mensajes MQTT con una baja latencia y un alto rendimiento[15].

Sistemas de control y telemetría de invernaderos: En el Jardín Botánico de la Ciudad de Posadas, se utilizan invernaderos para el cultivo de plantas. Los sistemas de control y telemetría se utilizan para monitorear y controlar las condiciones ambientales en el invernadero, como la temperatura, la humedad, la iluminación y la irrigación. Estos sistemas pueden ser monitoreados y controlados a distancia a través de una red IoT utilizando el protocolo MQTT y el broker Mosquito[16].

Análisis de protocolo MQTT y Broker Mosquito en dispositivos IoT: El análisis del protocolo MQTT y el broker Mosquito en dispositivos IoT es fundamental para comprender cómo funcionan los sistemas de control y telemetría en un invernadero. El análisis de tráfico de red puede proporcionar información valiosa sobre el

rendimiento del sistema, identificar cuellos de botella y posibles vulnerabilidades de seguridad. Además, el análisis de la estructura de los temas y la calidad de servicio puede mejorar la eficiencia y la confiabilidad del sistema[15], [17].

D. Sistemas de telemetría

Los sistemas de telemetría son tecnologías que permiten la medición y transmisión remota de datos. Estos datos son recogidos en un lugar y luego enviados a otro para su monitorización, análisis y almacenamiento. Los sistemas de telemetría se utilizan en una variedad de aplicaciones, que van desde el monitoreo del estado de salud de pacientes en hospitales hasta el rastreo de vehículos en movimiento y el seguimiento de sondas espaciales[18].

Las aplicaciones de los sistemas de telemetría son innumerables, algunos de los cuales son los siguientes: Aeroespacial, Sanitaria, Industrial, Automovilística, Ambiental, Deportiva, etc[16], [18], [19].

II. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

Este trabajo tiene como principal línea de investigación el desarrollo y evaluación de un analizador de tráfico de paquetes, del protocolo MQTT utilizando broker Mosquito aplicado en dispositivos IoT para sistemas de control y telemetría de invernaderos. El estudio se llevará a cabo en el centro de investigación Jardín Botánico de la ciudad de Posadas, Provincia de Misiones, República Argentina.

III. PRUEBAS REALIZADAS

- 1) Tipo de herramienta:
 - a. Tcpdump es una herramienta de línea de comandos utilizada para la captura y análisis básico de paquetes.
 - b. Wireshark es una herramienta gráfica avanzada para captura y análisis detallado de paquetes.
 - c. El archivo analizador.py es un Prototipo de una aplicación gráfica para monitorear

- y graficar datos de mensajes MQTT en tiempo real.
- 2) Captura de paquetes y Análisis en tiempo real:
 - a. Tcpcdump y Wireshark permiten la captura de paquetes de red.
 - b. El Prototipo analizador.py no captura paquetes de red, sino que se enfoca en la captura y visualización de datos MQTT.
 - 3) Filtros:
 - a. Tcpcdump usa filtros BPF para capturar paquetes específicos.
 - b. Wireshark ofrece filtros avanzados para captura y visualización.
 - c. El Prototipo analizador.py no incluye filtros de captura ni visualización, ya que se enfoca en datos MQTT.
 - 4) Facilidad de uso:
 - a. Tcpcdump requiere conocimiento de comandos específicos.
 - b. Wireshark y el Prototipo analizador.py proporcionan una interfaz gráfica intuitiva, siendo el Prototipo fácil de usar con botones y menús desplegables.
 - 5) Desempeño:
 - a. Tcpcdump es muy eficiente en términos de recursos.
 - b. Wireshark consume más recursos debido a su interfaz gráfica.
 - c. El Prototipo analizador.py también utiliza recursos debido a su interfaz gráfica y la visualización en tiempo real.
 - 6) Exportación de datos:
 - a. Tcpcdump y Wireshark permiten la exportación de datos capturados.
 - b. El Prototipo analizador.py no está diseñado para exportar datos, sino para visualizarlos en tiempo real.
 - 7) Compatibilidad:
 - a. Todas las herramientas son multiplataformas, funcionando en varios sistemas operativos.
 - 8) Visualización de datos:
 - a. Tcpcdump presenta los datos en formato de texto.
 - b. Wireshark y el Prototipo analizador.py proporcionan visualización avanzada con gráficos y colores, siendo el Prototipo especializado en gráficos de datos MQTT.
 - 9) Decodificación de protocolos:
 - a. Wireshark soporta una amplia variedad de protocolos.
 - b. El Prototipo analizador.py se limita a manejar datos de mensajes MQTT.
 - 10) Análisis posterior:
 - a. Wireshark permite análisis detallado post-captura.
 - b. El Prototipo analizador.py se enfoca en monitoreo en tiempo real sin opciones de análisis posterior.
 - 11) Funcionalidades extra:
 - a. Tcpcdump puede ser utilizado en scripts para automatización.
 - b. Wireshark tiene soporte para extensiones y plugins.
 - c. El Prototipo analizador.py incluye funcionalidades para conectar a múltiples brokers y sensores MQTT, y graficar datos en tiempo real.
 - 12) Uso típico:
 - a. Tcpcdump y Wireshark son utilizados por administradores de red y analistas de seguridad.
 - b. El Prototipo analizador.py es adecuado para monitorear y analizar en tiempo real datos de sistemas basados en MQTT.

IV. RESULTADOS OBTENIDOS

El Prototipo desarrollado para el estudio de caso del Instituto de Investigación Jardín Botánico prioriza la durabilidad y coste-efectividad, adaptándose a las restricciones de hardware, software e infraestructura. En lugar de usar herramientas más pesadas como Wireshark, que es ideal para análisis intensivos pero imprácticos en un Raspberry Pi con limitaciones de hardware y expuestos a daños físicos, o Tcpcdump, que solo posee capacidad de líneas de comando; se optó por un código Python personalizado en Raspbian. Este enfoque minimiza riesgos y maximiza la funcionalidad a pesar de las caídas de red, la humedad, y

amenazas de robo, usando dispositivos económicos como NodeMCU para reemplazar los Raspberry Pi cuando sea necesario. Esta solución permite una monitorización efectiva y confiable en condiciones desafiantes, manteniendo la viabilidad económica y técnica del proyecto. Proporciona una monitorización efectiva sin abrumar el sistema del Raspberry Pi, garantizando así la continuidad operativa incluso en condiciones de conectividad fluctuante.

V. FORMACIÓN DE RECURSOS HUMANOS

Este proyecto es parte de un plan de tesis de maestría en Tecnologías de la Información, en el cual se prevén incorporar becarios, tesistas, ayudantes de cátedra para Sistemas Operativos, Comunicación y Redes I y Comunicación y Redes II de las Carreras de Analista en Sistemas de Computación, Licenciatura en Sistemas de Información y Profesorado Universitario en Computación, de la Facultad de Ciencias Exactas, Químicas y Naturales (FCEQyN), de la Universidad Nacional de Misiones (UNaM).

VI. REFERENCIAS

- [1] M. N. Patel y O. P. Wilson, "Challenges and Opportunities in IoT Data Analytics", *Journal of Computational Intelligence and Systems*, vol. 6, núm. 4, pp. 256–263, 2021.
- [2] J. K. Adams, "The Evolution of IoT: From Concept to Real-World Applications", en *Proceedings of the International Conference on Digital Transformation and Innovation*, Boston, MA, USA, 2020, pp. 410–415.
- [3] F. G. Lee y H. I. Walker, "Adaptive Algorithms for Smart Climate Control in IoT Environments", *Journal of Internet and Things Research*, vol. 3, núm. 1, pp. 12–20, 2019.
- [4] S. T. Brown, "Real-Time Data Collection and Processing in IoT Networks", en *Proceedings of the Annual Symposium on Networked Systems*, San Francisco, CA, USA, 2018, pp. 320–327.
- [5] L. M. White y P. Q. Johnson, "Smart Home Devices: A Deep Dive into Intelligent Thermostats", *Journal of Modern Electronics and Communication*, vol. 5, núm. 2, pp. 78–85, 2020.
- [6] A. B. Smith, J. L. Clark, y D. E. Rodriguez, "Internet of Things (IoT): Principles and Applications", en *Proceedings of the International Conference on Advanced Computing Technologies*, New York, NY, USA, 2019, pp. 225–230.
- [7] J. K. López y M. N. Pérez, "Aplicaciones del IoT en la industria y agricultura: Potencial y desafíos", en *Actas del Simposio Internacional sobre Innovación Tecnológica*, Barcelona, España, 2018, pp. 402–409.
- [8] F. G. Gómez y H. I. Castillo, "Privacidad y IoT: Problemas emergentes en la recopilación de datos personales", *Revista Iberoamericana de Privacidad y Datos Personales*, vol. 4, núm. 1, pp. 15–23, 2021.
- [9] L. M. García y P. Q. Fernández, "El impacto transformador del Internet de las Cosas en la sociedad moderna", *Revista de Tecnologías Avanzadas*, vol. 7, núm. 2, pp. 45–52, 2020.
- [10] S. T. Rodríguez y A. B. Morales, "Desafíos de seguridad en dispositivos IoT: Un análisis profundo", en *Actas de la Conferencia Iberoamericana sobre Ciberseguridad*, Madrid, España, 2019, pp. 318–325.
- [11] J. López y M. Pérez, "Análisis del protocolo MQTT para aplicaciones de Internet de las Cosas", *Revista de Investigación en Tecnología de la Información y Comunicación*, vol. 8, núm. 1, pp. 45–56, 2019.
- [12] A. Martínez y L. González, "Implementación y análisis de protocolos de comunicación en Internet de las Cosas", *Revista Cubana de Ciencias Informáticas*, vol. 14, núm. 2, pp. 135–149, 2020.
- [13] "MQTT - The Standard for IoT Messaging". Consultado: el 13 de noviembre de 2023. [En línea]. Disponible en: <https://mqtt.org/>
- [14] A. M. García y A. J. García, "MQTT: Un protocolo para la internet de las cosas", *Revista Tecnológica-EI*, vol. 6, núm. 1, pp. 1–11, 2017.
- [15] "Eclipse Mosquitto". Consultado: el 13 de noviembre de 2023. [En línea]. Disponible en: <https://mosquitto.org/>
- [16] J. Torres y A. López, "Implementación y análisis de sistemas de control y telemetría en invernaderos utilizando MQTT y el broker Mosquitto", *Revista de Investigación en Ingeniería Eléctrica, Electrónica y de Sistemas*, vol. 5, núm. 2, pp. 45–56, 2018.
- [17] R. Pérez y M. Gómez, "Análisis de rendimiento del broker Mosquitto en aplicaciones IoT", *Revista de Investigación en Tecnología de la Información y Comunicación*, vol. 9, núm. 2, pp. 78–89, 2020.
- [18] A. López y M. García, "Sistemas de telemetría: Principios y aplicaciones", en *Tecnologías de la Información y Comunicaciones en Ingeniería*, 2020, pp. 45–62.
- [19] R. Singh y S. Gupta, "Review of Message Queuing Telemetry Transport Protocol for Internet of Things", *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 4, núm. 2, pp. 1009–1013.

Uso de un contenedor inteligente en la logística del traslado de mercadería

Ciro Edgardo Romero

Departamento de Investigación, Desarrollo e Innovación

C & S Informática S. A

Buenos Aires, Argentina

cromero@cys.com.ar

Abstract—En varios sistemas de seguimiento de envíos, se utilizan diferentes dispositivos para agregar precisión. Es común utilizar GPS integrados a vehículos o aplicaciones de celular, donde se envíe información relevante. En este esquema de funcionamiento, se realiza el seguimiento de medio de transporte, o de las personas ocupadas en realizar los traslados. El objetivo del presente artículo, es describir un sistema de seguimiento, que realice la supervisión de bienes, de informa independiente del medio por el cual es trasladado.

Palabras Clave—Internet de las Cosas, IOT, Logística

I. INTRODUCCION

En un sentido clásico, el propósito de la trazabilidad es el seguimiento activo desde el lugar de producción hasta el destino. El objetivo de este seguimiento es identificar los orígenes y estudiar las distintas etapas de todo el proceso de distribución hasta llegar al consumidor final [3]. Los avances en la tecnología IoT han ayudado a las empresas involucradas en actividades logísticas a comenzar a utilizar dispositivos inteligentes para transportar mercancías. Estos dispositivos permiten recopilar, procesar y enviar datos relacionados con el traslado de mercancías a un software especializado, creando así un sistema de seguimiento inteligente. Gracias a estos avances tecnológicos, el prototipo presentado en este artículo, logra ser un sistema de seguimiento de productos y no del vehículo que lo traslada. El mismo, está basado en el proyecto final de posgrado del autor y el departamento de I+D+i de C&S [14].

A. Posibles aplicaciones

Se pueden monitorear las variables ambientales internas que afectan el estado sensorial de los productos de envío biodegradables. Condiciones similares fuera del contenedor pueden afectar negativamente al propio contenedor y dañar su contenido [11]. Además, la exposición a la luz puede cambiar ciertos productos, como los medicamentos. La luz es una forma de energía, por lo que se convierte en calor, lo que puede afectar negativamente al recipiente y/o a su contenido [5].

Otra variable que se puede monitorear es la posición del contenedor con respecto a un plano de referencia que

representa el suelo sobre el cual se mueve el contenedor. Recordar las diferentes posiciones del envase puede ayudarte a detectar errores a la hora de manipular el envase, ya que una inclinación excesiva o movimientos muy bruscos pueden dañar el producto [16]. Finalmente, se debe conocer la ubicación geográfica del contenedor para garantizar un seguimiento detallado y preciso del seguimiento. La seguridad y la confianza se crean al poder encontrar productos en sistemas de referencia de forma transparente e identificable [3].

II. ESTRUCTURA GENERAL DEL PROTOTIPO

El prototipo cuenta con contenedor térmico que envía constantemente información de las condiciones internas y externas, recopiladas por sensores. El envío de esta información se realiza a través de Internet y es recibida por un servidor que persiste todo en una base de datos. Esto se suma a una aplicación, donde se visualizan los detalles de las mediciones realizadas durante cada etapa de cada viaje.

A. Contenedor térmico

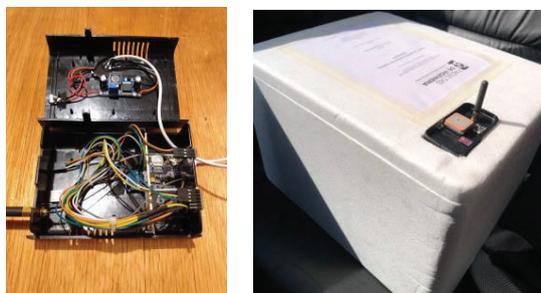
El contenedor del sistema, posee como componente principal la placa de desarrollo TTGO T-Call. Este permite implementar conectividad Wifi e Internet móvil desde una tarjeta SIM, por medio de tecnología GSM [10]. La placa como base el SoC (*System on Chip*) ESP32-wroom-32 de la empresa Espressif [19]. Se seleccionaron diferentes sensores y módulos para obtener las variables que serán monitorizadas durante el traslado del contenedor:

- Temperatura, presión y humedad (interna y externa): las variables en cuestión son obtenidas por medio de los módulos BMP180 y BME280, basado en los sensores homónimos, ubicados de forma externa e interna respectivamente. Por un lado, el sensor BMP180 permite medir presión barométrica y temperatura [1]. por otro lado, el sensor BME280 mide presión barométrica, humedad relativa y temperatura [2]. En ambos casos, se utiliza el protocolo de comunicación I2C para conectar los módulos con la placa de desarrollo.
- Luz (interna y externa): la medición de esta variable se realiza por medio del módulo-sensor TEMT6000

[15]. Este mismo, transmite la medición por medio de una señal analógica que la placa procesa posteriormente.

- Posición relativa: la posición se mide utilizando el módulo MPU6050, el cual es una unidad de medición inercial o IMU (del inglés *inertial measurement unit*) [6] de seis grados de libertad, que combina un acelerómetro de 3 ejes y un giroscopio de 3 ejes, fabricado por la empresa Invensense [7]. Este módulo, también se encuentra comunicado por medio de protocolo I2C.
- Ubicación: para ello se utilizó el módulo NEO-6M, parte de una familia de receptores GPS con alto rendimiento y bajo costo de la marca Ublox [18]. Este módulo, se comunica con el microcontrolador por medio de protocolo UART.

La conexión de los componentes se realiza mediante cableado simple utilizando cables prototipo de 0,5 mm de espesor y el hardware está protegido en un gabinete de plástico. Este último, se puede modificar para agregar antenas de comunicación, botones y luces indicadoras al dispositivo. Lleva incorporadas dos baterías recargables, lo que garantiza la autonomía del prototipo. Al mismo tiempo, el gabinete esta equipado con una caja aislada para mantener la temperatura interna cercana a la de un prototipo comercial. Esto puede ilustrarse en la Fig. 1 donde se puede observar el prototipo terminado, con las antenas GSM y GPS sobresaliendo de la tapa de la caja térmica.



(a) Gabinete (b) Contenedor

Fig. 1. Prototipo de contenedor inteligente

El microcontrolador incluido en el contenedor es el encargado de orquestar el funcionamiento de los diferentes módulos del sistema. El código del programa para ejecutar el funcionamiento fue escrito en MicroPython, una implementación sencilla y eficiente de Python 3 [9]. Este lenguaje incluye un pequeño subconjunto de la biblioteca estándar, reimplementada y optimizada para ejecutarse en microcontroladores.

El programa implementa las librerías de los módulos antes mencionados, utilizando dos archivos importantes

que MicroPython busca en la raíz de su sistema de archivos: *boot.py* y *main.py*. El primero se ejecuta al encender, o reiniciar, el microprocesador y contiene la configuración de la placa en el arranque. El segundo, contiene el programa de los servicios que implementan las librerías de los sensores, la inicialización de los puertos, la conexión a Internet y la lógica de envío de los datos. En la Fig. 2 se muestra el diagrama de flujo que describe el comportamiento del programa que se ejecuta dentro del contenedor.

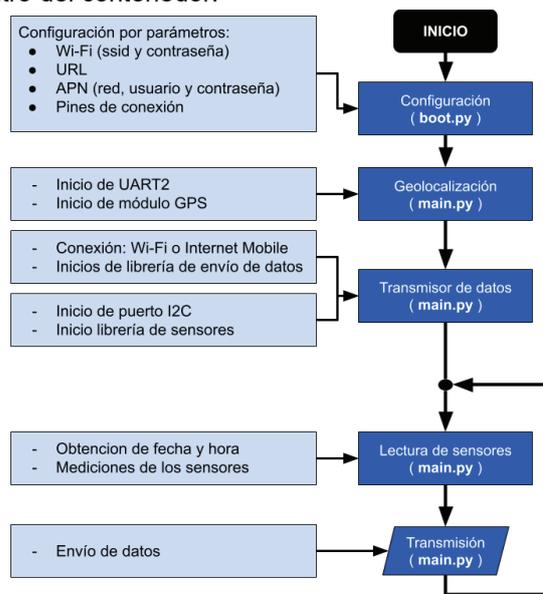


Fig. 2. Diagrama de flujo que describe el programa del contenedor

B. Recepción de datos

La lógica para enviar mediciones se construye a partir de un contenedor. Las especificaciones de esta placa son para el funcionamiento del módulo y la conexión a Internet. Utiliza métodos para la lectura de cada sensor, los guarda y agrupa en un objeto JSON declarado en el programa. Este objeto se envía como una *request* al servidor web que procesa y almacena la información. A partir de ahí, se expone una interfaz llamada API REST para recibir y almacenar datos [8]. Esta interfaz verifica los contenedores habilitados en el sistema para limitar los dispositivos que pueden interactuar con la aplicación.

C. Plataforma Mobile

La plataforma *mobile* proporciona a los usuarios la capacidad de visualizar gráficamente la información brindada por los contenedores. Dicha plataforma se desarrolló utilizando la arquitectura basada en *Context API* en React Native[13] es una opción poderosa y simplificada para gestionar el estado global de una aplicación sin necesidad de instalar librerías externas como Redux. La *Context API* de React permite compartir

datos a través de componentes sin necesidad de pasar datos manualmente en cada nivel del árbol de componentes generado en React. Esto resulta especialmente útil en aplicaciones de tamaño pequeño a mediano.

Al mismo tiempo, se implementó una base de datos para persistir la información. Por último, se utilizó la plataforma abierta conocida como Docker para facilitar la ejecución de todos los componentes [4].

- **Back-end:** desarrollado en NestJS [12], un *framework* moderno para aplicaciones *backend* de Node.js que permite a los desarrolladores crear aplicaciones robustas, eficientes y escalables. Fue diseñado para aprovechar las características de TypeScript [17] y proporcionar una arquitectura modular y fácilmente extensible.
- **Mobile:** desarrollado en React Native, *framework* diseñado para construir aplicaciones compatibles con dispositivos móviles.
- **Base de datos:** se utilizó una base de datos relacional *Postgres*.

En la Fig. 3 se observa un diagrama con las diferentes tecnologías utilizadas en la plataforma.

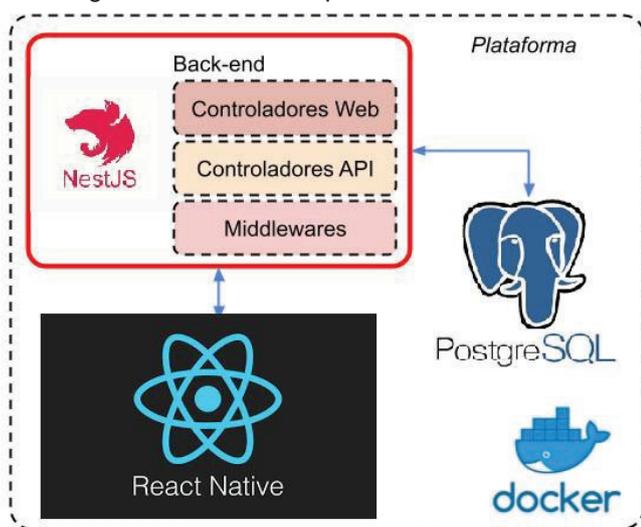


Fig. 3. Componentes de la plataforma.

Existe una capa de servicios, desarrollada para manejar la lógica de la plataforma. Se utiliza para el almacenamiento de datos y proporcionar canales de comunicación (conocidos como *endpoints*), que la aplicación *mobile* utiliza tanto para crear un nuevo envío como para consultar el estado del mismo. La comunicación entre la aplicación *mobile* y los servicios, se realizan a través de solicitudes HTTP.

III. CONCLUSIONES Y TRABAJO FUTURO

El conocimiento adquirido durante el desarrollo e implementación del presente prototipo, aportó valor a todos los *stakeholders*. Se ha alcanzado un cierto nivel de innovación en relación con los sistemas de logística y trazabilidad. Todos los requisitos se analizan deliberadamente para que el proyecto pueda ser útil a la empresa y sus colaboradores. La propuesta sirve como prueba de conceptos y, utilizando las lecciones aprendidas de este trabajo, se obtuvo suficiente conocimiento para comenzar a refinar varios aspectos del prototipo para garantizar que madure.

REFERENCES

- [1] Componentes 101. *BMP180 - Atmospheric Pressure Sensor*. <https://components101.com/sensors/bmp180atmospheric-pressure-sensor>. Mar. 2021. (Visited on 03/02/2021).
- [2] Bosch. *BME280*. <https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors/bme280/>. Mar. 2021. (Visited on 03/02/2021).
- [3] Miguel Gaston Cedillo Campos y Cuauhtemoc Sanchez Ramirez. *Análisis dinámico de sistemas industriales*. Editorial Trillas, Mexico, 2008.
- [4] Docker. *Our Company*. Visitado: 2021-03-24. Mar. 2021. URL: <https://www.docker.com/company>.
- [5] L Dominguez and M Parzanese. "Luz ultravioleta en la conservación de alimentos". In: *Alimentos argentinos* 52.2 (2011), pp. 70–76.
- [6] DS Fedorov et al. "Using of measuring system MPU6050 for the determination of the angular velocities and linear accelerations". In: *Automatics & Software Engineering* 11.1 (2015), pp. 75–80.
- [7] InvenSense. *MPU-6000 and MPU-6050 Product Specification - Revision 3.4*. <https://invensense.tdk.com/wpcontent/uploads/2015/02/MPU-6000-Datasheet1.pdf>. Mar. 2021. (Visited on 03/02/2021).
- [8] Rafael Luis Granados La Paz. *Desarrollo de aplicaciones web en el entorno servidor. IFCD0210*. IC Editorial, 2015.
- [9] Micropython. *Micropython (Home)*. Visitado: 2021-03-10. Mar. 2021. URL: <https://micropython.org/>.
- [10] Michael Mouly and Marie-Bernadette Pautet. "GSM System for Mobile Communications, The". In: *Revista Zona Logística* (1992).
- [11] HECTOR Navarro. "Logística en la cadena de frío". In: *ISBN 2950719007* (2013), pp. 34–37.
- [12] NestJS. *NestJS*. Visitado: 2024-07-24. July 2024. URL: <https://nestjs.com/>.
- [13] React. *React*. Visitado: 2024-03-24. Apr. 2024. URL: <https://react.dev/reference/react/apis>.



- [14] TS. Ciro Edgardo Romero. "Smart traceability, sistema de logistica inteligente". Carrera de Especializacion en Internet de las Cosas, Feb. 2021. URL: [http : / / laboratorios.fi.uba.ar/lse/tesis/LSE-FIUBA-TrabajoFinal-CEIoT-Ciro-Edgardo-Romero-2021.pdf](http://laboratorios.fi.uba.ar/lse/tesis/LSE-FIUBA-TrabajoFinal-CEIoT-Ciro-Edgardo-Romero-2021.pdf).
- [15] Vishay Semiconductors. *Ambient Light Sensor*. <https://www.vishay.com/docs/81579/temt6000.pdf>. Mar. 2021. (Visited on 03/02/2021).
- [16] GS1 Global Traceability Standard. *Business Process and System Requirements for Full Supply Chain Traceability*. [https : / / www . gs1 . org / sites / default / files / docs / traceability / Global Traceability Standard . pdf](https://www.gs1.org/sites/default/files/docs/traceability/GlobalTraceabilityStandard.pdf). Mar. 2021. (Visited on 03/01/2021).
- [17] TypeScript. *TypeScript*. Visitado: 2024-07-24. July 2024. URL: <https://www.typescriptlang.org/>.
- [18] U-Blox. *NEO-6 series. Versatile u-blox 6 GPS modules*. <https://www.u-blox.com/en/product/neo-6-series>. Mar. 2021. (Visited on 03/17/2021).
- [19] Xinyuan-LilyGO. *LilyGo T-Call SIM800 Series*. Visitado: 2021-03-10. Mar. 2021. URL: [https://github.com/ Xinyuan-LilyGO/LilyGo-T-Call-SIM800](https://github.com/Xinyuan-LilyGO/LilyGo-T-Call-SIM800).



Transmisión on line de datos RIOMETRICOS

Leonardo J. Navarria
*Grupo de Investigación y Desarrollo
en Radiofrecuencia e
Instrumentación*
*Facultad de Ciencias Astronómicas
y Geofísicas*
Universidad Nacional de La Plata
Paseo del Bosque s/n B1900FWA
La Plata
lnavarria@fcaglp.unlp.edu.ar
ORCID 0000-0002-1061-4896

Ricardo E. García
*Grupo de Investigación y Desarrollo
en Radiofrecuencia e
Instrumentación*
*Facultad de Ciencias Astronómicas
y Geofísicas*
Universidad Nacional de La Plata
Paseo del Bosque s/n B1900FWA
La Plata
regarcia@fcaglp.unlp.edu.ar

Guillermo D. Rodríguez
*Grupo de Investigación y Desarrollo
en Radiofrecuencia e
Instrumentación*
*Facultad de Ciencias Astronómicas
y Geofísicas*
Universidad Nacional de La Plata
Paseo del Bosque s/n B1900FWA
La Plata
grodriguez@fcaglp.unlp.edu.ar

Resumen— El objetivo del trabajo consistió en el desarrollo de un sistema de adquisición y transmisión de datos en tiempo real para un instrumento Riometro, (de sus siglas en inglés, **Relative Ionospheric Opacity Meter for Extra-Terrestrial Emissions of Radio**). La adquisición de los datos en tiempo real permite observar en todo momento el comportamiento de las capas inferiores de la ionósfera monitoreando los niveles de ionización en estas. El equipo original empleaba un multímetro, el cuál permitía la conexión con una PC con Windows XP, resultando actualmente obsoleto e impráctico. En este trabajo se detalla una actualización de la adquisición y transmisión de datos, además de una mejora a los efectos de un ruido electromagnético que interfiere en la adquisición correcta de los datos.

Palabras Claves—**RIOMETRO, adquisición, transmisión**

I. INTRODUCCIÓN

La red SARINET [1] (South America Riometer Network) era una red compuesta de riómetros de la Universidad de Takushoku (Tokio, Japón) instalada en Sudamérica para investigar la ionósfera y principalmente los efectos que sobre ella producen la precipitación de partículas energéticas, las cuales se ven incrementadas por la presencia de la Anomalía Magnética del Atlántico Sur (AMAS). La red poseía 12 riómetros operativos [2], cinco de los cuales se habían instalado por gestión y con participación del Departamento de Geomagnetismo y Aeronomía de la Facultad de Cs. Astronómicas y Geofísicas de la UNLP, en distintas instituciones de la Argentina.

La discontinuidad de la red SARINET ha generado que los equipos queden al cuidado de

las Unidades Académicas que recolectaban los datos. En el caso en estudio, quienes quedaron a cargo del mantenimiento del instrumental es el Grupo de Investigación y Desarrollo en Radiofrecuencia e Instrumentación. (GIDRI) de la Facultad de Ciencias Astronómicas de la Universidad Nacional de La Plata (FCALP-UNLP).

Los riómetros son instrumentos pasivos que miden en forma continua la recepción de emisiones electromagnéticas muy estables de ciertas galaxias cercanas. Llamado ruido cósmico. Dichas ondas de radio sufren normalmente absorción en la parte inferior de la ionosfera (particularmente en capa D), y este fenómeno se incrementa en caso de eventos ionizantes extraordinarios (precipitación de partículas, fulguraciones solares, etc). El instrumento permite la detección y caracterización de tales eventos recibiendo señales de radio a frecuencias del orden de los 30 MHz [2].

II. FUNCIONAMIENTO DE LOS RIOMETROS

Cuando una onda electromagnética atraviesa la ionósfera sufre una atenuación, principalmente en la capa D, la cual es posible determinar aplicando la fórmula de Appleton-Hartree (Davies, 1990) (1).

$$Att[dB] = 4,6 * 10^{-5} \int \frac{N_e * v}{v^2 + (\omega \pm \omega_H \cos \Theta)^2} dl \quad (1)$$

En esta ecuación Att es la atenuación en decibeles, Ne es la densidad electrónica, v es la frecuencia de colisión entre electrones y partículas neutras, ω_H la pulsación de la girofrecuencia del electrón, ω la pulsación de la onda electromagnética, Θ el ángulo entre la perpendicular del campo magnético y la

dirección de propagación y, por último, de la diferencial de camino recorrido por la onda.

Analizando las diferencias de absorción de un día cualquiera en comparación con el nivel de señal recibida en días sin eventos ionizantes extraordinarios (Día Quieto), es factible determinar fenómenos anómalos de variación en la densidad electrónica como lo expresa la ecuación 1.[3]

La observación del ruido cósmico en la superficie terrestre por los riómetros es efectuada en el rango de 20 a 200 MHz, empleando nuestros equipos, 38,2 MHz.

Los RIOMETROS requieren para su instalación emplazamientos especiales, donde las perturbaciones electromagnéticas sean mínimas. En la actualidad, estas ubicaciones son difíciles de encontrar tanto en los grandes centros urbanos como en ciudades más pequeñas debido a la gran diversidad de dispositivos inalámbricos empleados en la vida actual, por lo que se suma la necesidad de disponer de un sistema autónomo independiente de provisión de energía eléctrica, para operar el mismo en zonas remotas. Cabe destacar que para la riometría es fundamental medir de forma continua los valores durante un período mensual, pudiendo así poder determinar la Curva de Día Quieto [2] que permita determinar las anomalías de absorción

III. SISTEMA DE MEDICIÓN

A. Riómetro original

El riómetro original opera con un arreglo de dos dipolos con un diagrama de radiación fijo, midiendo la radiación recibida desde el cenit. En la fig. 1 se muestra la placa del amplificador y convertidor potencia-tensión del RIOMETRO.



Figura 1: Tarjeta Amplificadora

Este equipo es alimentado por una fuente de alimentación de 110 VAC. Esta tensión es rectificadora por una fuente lineal para obtener las tensiones de 12V, 0V y -12V necesarias para el funcionamiento. Dentro de la tarjeta

amplificadora/medidora de potencia existen controles que permiten modificar el valor de Tensión de salida y el nivel de mínima detección, es decir, ajuste de ganancia y ajuste de cero, además de compensaciones de ganancia por temperatura fundamental en este tipo de sistemas radiométricos.

B. Sistema de almacenamiento de datos.

Los instrumentos originales realizaban la adquisición de datos conectando la salida de la señal del convertidor potencia -tensión a voltímetros con conexión a PC. Para aumentar la autonomía por duración de la batería de los voltímetros, estos eran alimentados con una fuente de 110 VAC, similar a la alimentación del RIOMETRO. El voltímetro obtenía la medición en tiempo real y luego mediante una conexión vía USB y su SOFTWARE se almacenaban en una PC con Windows XP. El almacenamiento de estas muestras debía ser guardado diariamente de forma manual por un operador para no perder los valores obtenidos, generando así una tarea engorrosa y monótona para obtener las mediciones. Fig. 2

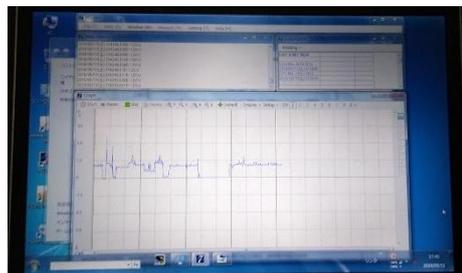


Figura 2: Captura de la medición en tiempo real

Los valores obtenidos eran mensualmente enviados a la red SARINET para el procesamiento de forma conjunta con el resto de los RIOMETROS.

IV. IMPLEMENTACIÓN DEL NUEVO SISTEMA

Para poder actualizar el sistema y realizar el muestreo de las tensiones y la adquisición de datos se evaluaron diferentes tecnologías disponibles optando por emplear una placa Arduino® [4]. Esta plataforma de desarrollo está basada en una placa electrónica de hardware libre que incorpora un microcontrolador re-programable. Este dispositivo posee además un convertidor analógico digital interno de 10 bits que provee una resolución adecuada para realizar la adquisición de datos deseada. Además, el equipo de recepción fue calibrado

con un generador de frecuencias y analizador espectral.

Para poder transmitir, almacenar y acceder a los datos de forma remota se seleccionó el dispositivo Arduino modelo UNO junto con una tarjeta de expansión modelo W5100 con la finalidad de agregar un puerto de red Ethernet y un slot para conexión de tarjetas de memoria SD.

Para almacenar los datos se utilizó la plataforma ThingSpeak (fig. 4), la cual se basa en Internet of Things (IoT). Esta plataforma pertenece a la empresa MATLAB® y permite almacenar datos obtenidos de sensores en la nube y desarrollar aplicaciones IoT. Thingspeak también ofrece aplicaciones que permiten analizar, visualizar los datos con la herramienta MATLAB así como también modificarlos con una rutina semi- personalizada. Los datos de los sensores pueden ser enviados desde Arduino, Raspberry Pi, BeagleBone Black u otras plataformas similares.

El Observatorio Magnético de “Las Acacias” (LAS, Lat.: -35°.0; Long.: 302°.3) (fig. 3) perteneciente a la Facultad de Ciencias Astronómicas y Geofísicas de la Universidad Nacional de La Plata ha sido el lugar elegido para instalar el instrumento, sitio en el cual desde hace unas seis décadas se viene registrando el campo magnético terrestre y realizando trabajos de actualización [5].



Figura 3: Ubicación del predio “Las Acacias”

C. Inconvenientes encontrados

Para la conexión con el router, por condiciones del emplazamiento se optó por Wi-Fi pero debido a que el equipo se encuentra instalado a una distancia aproximada de 100 metros del modem de transmisión de señal, hubo que instalar dos routers y configurarlos con dirección IP fija para que funcionen en modo repetidor y así obtener el alcance necesario .

El sistema de alimentación original se realizaba a través de una fuente convencional de transformador con rectificador. Tanto la placa de procesamiento de señal como la fuente

se encontraban en el mismo gabinete provocando esto interferencias a la señal de interés, no pudiendo así obtener correctamente el dato. Como solución, se aisló la fuente instalándose en un gabinete externo empleando cable mallado para alimentar al sistema.

D. .Medición en Campo

Con las actualizaciones realizadas el equipo está funcionando en forma estable desde hace aproximadamente un semestre. En la siguiente figura puede observarse la pantalla de adquisición.

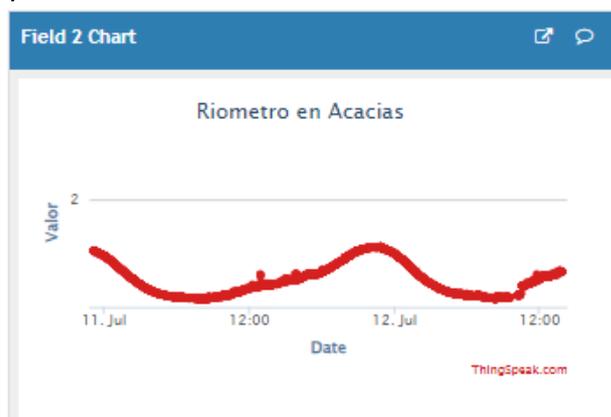


Figura 4: Lectura Remota

V. CONCLUSIONES Y TRABAJO A FUTURO

Con la realización de este proyecto se logró obtener una adquisición de los datos riométricos sin interferencia, en forma estable y pudiendo colocar la información en forma on-line. Esto último, permite actualmente el monitoreo en tiempo real del sistema, pudiendo detectar y caracterizar la recepción de ruido cósmico. Para finalizar la instalación del riómetro en el predio de “Las Acacias” fortalece el funcionamiento del Observatorio Geomagnético ya que posibilita observaciones de eventos en conjunto

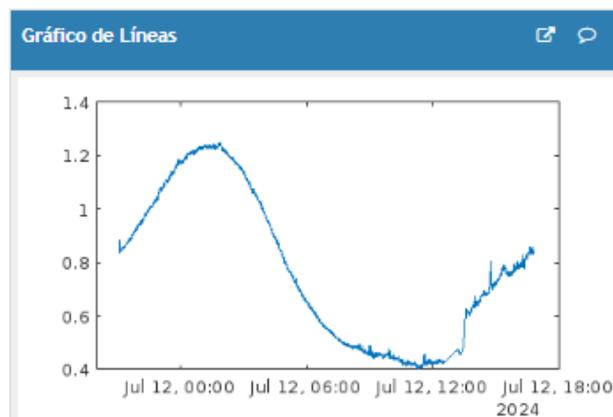


Figura 5: Procesado en tiempo real de la lectura remota



Como trabajo futuro se instalará un equipo de recepción de Very Low Frequency (VLF) para con el equipamiento ya instalado realizar observación en Climatología Espacial (Space Weather).

REFERENCIAS

- [1] <https://polaris.nipr.ac.jp/~ytanaka/riometer.html>
- [2] Riómetros: red SARiNet en Argentina, observaciones y procesamiento de datos R. E. García, J. C. Gianibelli, G. D. Rodríguez, P. Sallago, K. Makita y N. Quaglino. La Plata AAGG 2017
- [3] García, R. E., Rodríguez, G. D., & Navarra, L. J. (2023). Quiet day curve for RIOMETERS: Analysis and comparison of methods. In Space Weather Observations Throughout Latino America: Filling the Southern Gaps 2023 (SWOL) (Ushuaia, Tierra del Fuego, 2 al 6 de octubre de 2023).
- [4] <https://www.arduino.cc/>
- [5] Navarra, L. J., García, R. E., & Rodríguez, G. D. (2023). Lectura remota del magnetómetro Geometrics G856. IV Jornadas de Geociencias para la Ingeniería (Buenos Aires, 1° y 2 de noviembre de 2023). <http://sedici.unlp.edu.ar/handle/10915/160368>

AutoTicket: Sistema de cobro por QR con plataforma ESP32

Santiago Rotger, Lucas I. Calderon, Roberto M. Murdocca, Sergio F. Hernandez Velazquez

Departamento de Electrónica
Universidad Nacional de San Luis
San Luis, Argentina

e-mail: {santi.rotger, calderonlucasignacio, mmurdocc, sergio.sfhv}@gmail.com

Resumen - En el presente proyecto se desarrolla un sistema automático de cobro haciendo uso del microcontrolador ESP32. El sistema se aprovecha de las ventajas ofrecidas por las herramientas de pago digital de Mercado Pago: generando, cobrando y validando de forma automática y sin intermediarios un pago realizado a través de QR. Se implementa un servidor elaborado con JavaScript para realizar la autenticación del usuario de modo tal que pueda operar con el sistema. Mediante las capacidades de conectividad Wi-Fi de la ESP32, se implementaron solicitudes HTTP para interactuar con la base de datos y la API de Mercado Pago.

Palabras clave: ESP32, Mercado Pago, QR, Node.js, HTTP.

I. INTRODUCCIÓN

La idea inicial que impulsó este proyecto surge de la observación directa de una problemática presente en el comedor universitario de la Universidad Nacional de San Luis (UNSL), en muchas ocasiones los alumnos no cuentan con dinero en efectivo pero sí en las distintas billeteras digitales que no son aceptadas como medio de pago actualmente. A dicha observación se suma los tiempos de espera para realizar la compra del ticket.

A fin de dar respuesta a esta problemática y aprovechando el fácil acceso a las billeteras virtuales, surge el desarrollo de AutoTicket, que permite automatizar y mejorar la experiencia de pago.

El sistema se implementó utilizando una ESP32, esto ya que cuenta con funcionalidades de conectividad Wi-Fi lo que le permite conectarse a un servidor para mantener un control de los alumnos y comunicarse con la API de Mercado Pago.

II. DESCRIPCIÓN

El sistema integra un microcontrolador ESP32 con un servidor basado en Node.js que permiten la validación de los datos del alumno y del pago que se realice.

El usuario que realizará la compra deberá ingresar su DNI a través de un teclado matricial para así realizar la autenticación.

Tanto la ESP32 como el servidor interactúan con la API de Mercado Pago mediante peticiones HTTP para gestionar en tiempo real las transacciones, proporcionando un flujo continuo desde la generación del código QR hasta la confirmación del pago.

Si todas las comprobaciones son correctas se procede con la impresión del ticket. En la Fig.1 se muestra un diagrama en bloques completo del sistema Auto Ticket.

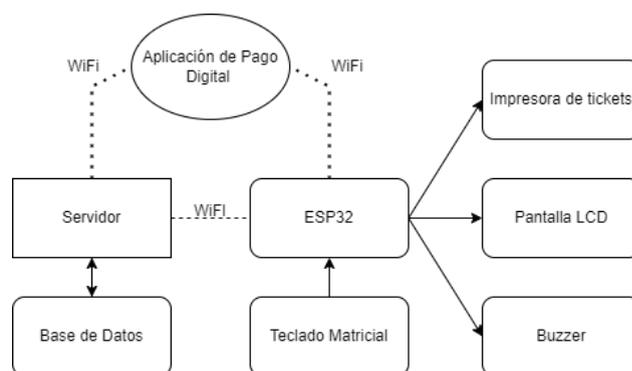


Fig. 1. Diagrama en bloques del sistema

Un aspecto a considerar en la implementación del sistema es la falta de una impresora térmica de tickets compatible con el microcontrolador, debido a su alto costo. Actualmente, la confirmación del pago se realiza visualmente a través del display y de forma auditiva con el buzzer. Aunque la integración de la impresora está pendiente, no afecta la validación del pago ni la operatividad

del sistema. La referencia a la impresión del ticket indica que el proceso de compra se ha completado con éxito.

Para contener el microcontrolador ESP32 junto con el teclado matricial, el buzzer, la pantalla LCD y la impresora de tickets con los que el usuario interactúa, se planeó el housing observable en la Fig.2, dicho gabinete se puede fijar a la pared a una altura apropiada o apoyarse sobre un mostrador.



Fig. 2: Housing Propuesto

III. ARQUITECTURA FÍSICA

La arquitectura está conformada por un microcontrolador ESP32 del fabricante Espressif (ESP32_DEVKIT_V1.1), un teclado matricial 4x4 (HC-543), un buzzer (TMB12A05), una pantalla LCD de dimensiones 16x2 (LCD-016N002B) y un Módulo que permite comunicación I2C (I2C-PCF8574).

La selección del microcontrolador ESP32 para este proyecto se debe a la capacidad de procesamiento que posee ya que tiene un procesador potente y eficiente. Esto es esencial para un proyecto que requiere la gestión de peticiones HTTP, procesamiento de datos, y el control de hardware [1]. Más específicamente las funcionalidades utilizadas del ESP32 en el sistema son: la conectividad WiFi ya que trabaja con el protocolo TCP-IP y además permite trabajar con el protocolo de comunicación I2C, usando únicamente el GPIO21 como línea serial de datos (SDA) y GPIO22 como línea de reloj serial (SCL) facilitando la comunicación con la pantalla LCD y permitiendo utilizar menos pines de salida. Además este microcontrolador cuenta con funcionalidades extras que aunque aquí no se

utilicen pueden integrarse en futuras versiones de AutoTicket.

El modelo utilizado “ESP32_DEVKIT_V1.1”, permite utilizar una fuente de corriente continua que suministre 5 (V), ya que contiene un regulador de tensión que la reduce a 3,3 (V), y que entregue por lo menos 500 (mA) ya que los demás componentes se alimentan a partir del microcontrolador.

La pantalla LCD 16x2 se llama así porque tiene 16 columnas y 2 filas. Por lo tanto, tendrá $(16 \times 2 = 32)$ 32 caracteres en total y cada carácter está formado por 5x8 píxeles [7].

El teclado está formado por una matriz de pulsadores que se encuentran dispuestos en filas (L1,L2,L3,L4) y columnas (C1,C2,C3,C4), esta organización es de acuerdo a reducir el número de pines requeridos para su conexión y programación [5]. En la Fig.3 se muestra el diagrama de conexiones del hardware. En la Fig.4 se observa el prototipo del sistema.

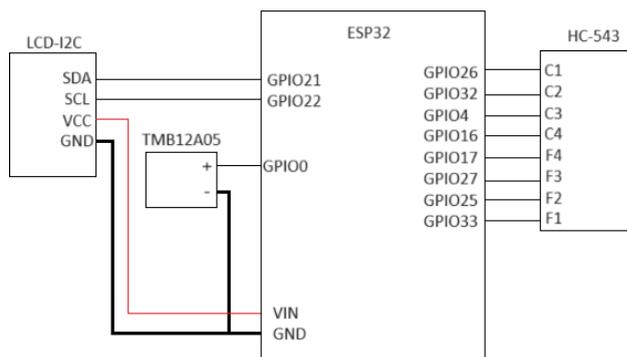


Fig. 3: Diagrama de conexiones

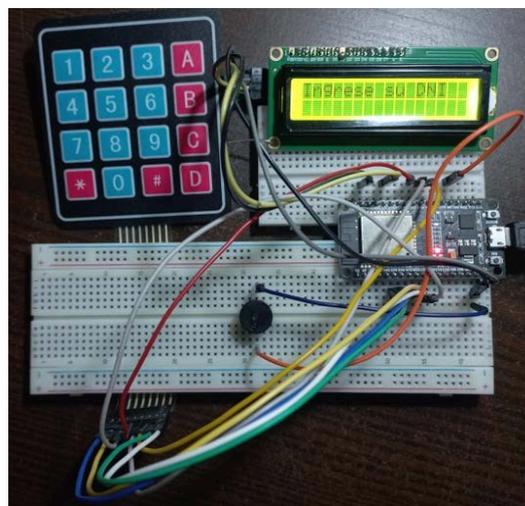


Fig. 4: Prototipo

IV. ARQUITECTURA DEL SISTEMA

A. ESP32

Para el funcionamiento en conjunto de las distintas partes del sistema, se hace uso de las funciones de las siguientes librerías:

- **Wifi.h**: proporciona funciones para iniciar la conexión WiFi, verificar el estado de la conexión, la dirección IP, etc [4].
- **HTTPClient.h**: permite al dispositivo realizar solicitudes HTTP a servidores [4].
- **Keypad.h**: se encarga de gestionar la entrada de datos desde el teclado matricial [15].
- **LiquidCrystal_I2C.h**: permite controlar pantallas LCD a través de I2C [16].

B. Servidor

El servidor está basado en Node.JS que se utiliza para crear aplicaciones de y permite gestionar muchas conexiones de forma simultánea [2], en él se integra las siguientes herramientas para su funcionamiento:

- **Express**: es un framework, que brinda las herramientas necesarias para la construcción del servidor [12].
- **body-parser**: es un middleware que analiza los cuerpos de las solicitudes HTTP y permite trabajar con datos en formato JSON [13].
- **Librería axios**: utilizada para interactuar con APIs externas permitiendo enviar y recibir datos de una forma sencilla [14].
- **Módulo sqlite3**: brinda las herramientas para la creación y trabajo con la base de datos SQL [8].

C. Base de Datos

La base de datos se construyó utilizando SQLite, se almacena en ella el DNI, el nombre y el apellido de los alumnos autorizados para comprar en el comedor universitario.

Se optó por SQLite ya que es fácil de integrar y brinda las herramientas suficientes para el número de datos que se están utilizando en este proyecto. También se tuvo en cuenta la gran cantidad de información disponible sobre esta herramienta debido a ser un proyecto de código abierto con una gran comunidad [8].

Aunque la base de datos almacena por el

momento la información justa y necesaria para la compra, es una herramienta que da flexibilidad al sistema ya que permitiría llevar un control sobre distintas variables como por ejemplo si un alumno ya compró su comida, cuántas veces utilizó el servicio en la semana, implementar una chequera, etc.

V. MERCADO PAGO DEVELOPERS

Una parte fundamental en el funcionamiento de AutoTicket es la conexión con la API de Mercado Pago Developers, esta brinda la oportunidad de trabajar en la creación de todas las etapas necesarias para una integración adecuada, segura y personalizable, a través de peticiones HTTP [3].

Mercado Pago no permite trabajar con su API si no se tienen los permisos necesarios en la cuenta, y estos son validados en cada petición, a través de lo que se denomina "Acces Token"[3].

La ESP32 está vinculada a una "Caja" que posee un QR fijo, dicha "Caja" pertenece a una Sucursal creada en Mercado Pago. Estas partes, como se mencionó anteriormente, fueron creadas a través de peticiones HTTP de URL's específicas brindadas por la API ya que este método permite un control y seguimiento mayor sobre las características e identificadores específicos de la Caja y de la Sucursal, que son necesarios para llevar un control y manejar las solicitudes.

La creación de la orden de compra se hace a través de una solicitud HTTP tipo PUT que utilizando los respectivos identificadores, carga la información en el QR de la Caja. Las variables que definen la compra, como lo son el nombre, la descripción, el código interno, el precio, etc. se definen en el body de la petición en formato JSON como se puede ver en la Fig.5.

También es importante indicar que a través de este mecanismo la orden de pago que se genera tiene el monto definido, sin la posibilidad que el usuario elija el valor, evitando dificultades a la hora de que el usuario efectúe el pago[3].

```
String body = "{"
  "\"cash_out\":{"
    "\"amount\":0"
  },"
  "\"description\": \"Menu del Dia\","
  "\"external_reference\": \"12345\","
  "\"items\":["
    "{"
      "\"sku_number\": \"██████████\","
      "\"category\": \"COMIDA\","
      "\"title\": \"AutoTicket\","
      "\"description\": \"Menu del Dia comprado a traves de AutoTicket\","
      "\"unit_price\":1,"
      "\"quantity\":1,"
      "\"unit_measure\": \"unit\","
      "\"total_amount\":1"
    }"
  ],"
  "\"notification_url\": \"\" + String(url) + \"/webhook/mercadopago\","
  "\"title\": \"Almuerzo\","
  "\"total_amount\":1"
}";
```

Fig. 5: body formato json

VI. FUNCIONALIDAD

Para explicar el funcionamiento se debe tener en cuenta que AutoTicket es un sistema compuesto por 3 partes, aquella contenida dentro de la Arquitectura Física, otra compuesta por el servidor y otra por la base de datos. Ambas se comunican con la API de Mercado Pago. A continuación se detalla la serie de tareas realizada por cada una de estas partes.

A. ESP32

Las acciones realizadas por el conjunto ESP32 e interfaz de usuario se pueden dividir en tres partes fundamentales:

1. *Ingreso y comprobación del DNI*
2. *Generación del QR de pago*
3. *Recepción de la validación del pago*

1. *Ingreso y comprobación del DNI*

Haciendo uso de la pantalla LCD se solicita que se ingrese el DNI del alumno que está por llevar a cabo la compra del ticket.

A través del teclado matricial, el alumno ingresa su DNI digito por digito y un zumbido generado por un buzzer indica la lectura de cada dígito ingresado. Por medio de la pantalla LCD se va mostrando el DNI a medida que se ingresa. En caso de que se ingrese un número incorrecto se puede borrar, mediante el botón "D" del teclado. El programa ignora el ingreso de los botones que no son necesarios como "A", "B", "C", "#", y "*".

Una vez ingresado completamente el DNI, se genera una petición HTTP POST, hacia el

servidor en Node.js. El "body" de esta petición contiene únicamente el DNI ingresado, para corroborar que el alumno se encuentre en la base de datos del sistema y evitar que una persona no registrada, pueda comprar el ticket. Luego el sistema queda a la espera de la validez del DNI ingresado y una vez que se comprobó, la respuesta de la petición indica si es válido o no.

En caso de que sea incorrecto o no se encuentre en la base de datos, el buzzer emite un sonido dando a entender que lo ingresado es invalido y por la pantalla LCD se solicita nuevamente que se ingrese un DNI.

2. *Generación del QR*

En el momento en que se certifica que el DNI es válido, se genera una nueva petición HTTP, pero ahora de tipo PUT, a la API de Mercado Pago Developers, que genera una orden de pago y la carga en el QR.

Una vez que está cargado el QR, la ESP32 espera la confirmación del servidor de que el usuario ha escaneado y realizado el pago. En esta instancia se contemplan dos situaciones particulares que hay que tener en cuenta, por si el alumno se arrepiente de la compra o simplemente se va.

En caso de que el alumno se arrepienta, puede cancelar la compra presionando la tecla "D", la ESP32 genera una petición HTTP DELETE y cancela el pago. En ausencia de la confirmación de pago del usuario, y transcurrido un intervalo de tiempo predefinido, el sistema procede a cancelar automáticamente la transacción, emitiendo una solicitud HTTP DELETE. En ambos casos se indica por pantalla que se canceló la compra y también se genera un zumbido a través del buzzer.

3. *Recepción de la validación del pago*

La ESP32 tiene en este caso funcionalidades propias de un servidor, esto primeramente podría parecer innecesario, pero su utilidad recae en la posibilidad de escuchar peticiones desde el servidor en Node.js, esto permite que la información de la confirmación del cobro sea mucho más fluida ya que no se requiere que la ESP32 espere un tiempo y pregunte al servidor si el pago se realizó, si no que el servidor le

avisará con una petición GET que el pago se completó.

La confirmación del pago genera que la ESP32, emita una melodía, encienda un led e imprima el ticket. El formato de impresión se puede observar en la Fig.6

```

-----
-UNSL-
-Auto-Ticket-
-----
-dd/mm/aa-
-Hora:minuto-
-----
Alumno:
Nombre-Apellido
-----
    
```

Fig. 6. Formato impresión de ticket

B. Servidor en Node.js

El servidor realiza las siguientes acciones:

1. Validación del DNI
2. Recepción y envío de la confirmación de pago.

1. Validación del DNI.

El servidor recibe a través de una petición HTTP POST la solicitud de la ESP32 para validar la existencia de un DNI que llega en el "body" de dicha petición en formato JSON.

El servidor revisará la existencia del DNI en la base de datos y a través de la respuesta de la petición enviará una confirmación o negación.

2. Recepción y envío de confirmación de pago.

Cuando la orden de pago es creada y cargada en el QR, el servidor empieza a recibir información del estado de dicha orden ya que se encuentra configurado como webhook.

Webhook es un mecanismo de comunicación que permite que una web envíe notificaciones o datos en tiempo real [11]. En lugar de tener que consultar constantemente para verificar si ha sucedido algún evento, este sistema permite que la API de Mercado Pago notifique

automáticamente al servidor cuando sucede algo.

Mediante la verificación de la existencia del ID de pago se comprueba que la transacción de dinero se ha completado correctamente ya que dicho ID existe únicamente al concretarse exitosamente el cobro.

Cuando el pago es confirmado, el servidor avisará mediante una petición HTTP GET a la ESP32, que se encuentra esperando.

En la Fig.7 se puede observar el funcionamiento del sistema AutoTicket a través de una Máquina de Estados Finitos (MEF).

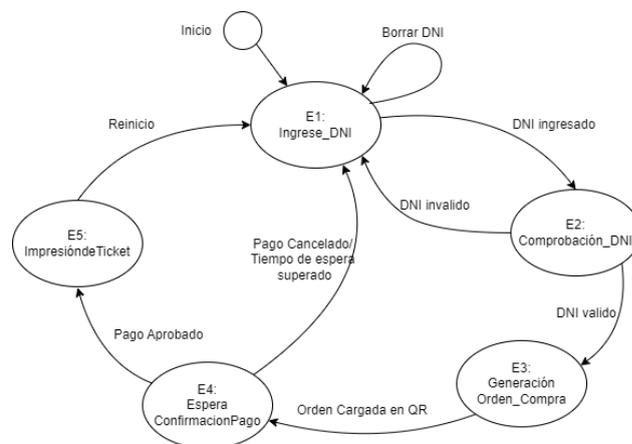


Fig. 7: Máquina de estados

VII. RESULTADOS

Las pruebas del servidor se realizaron utilizando una herramienta llamada Ngrock.

Esta permite crear un túnel entre un puerto local en el que se encuentra el servidor y una URL accesible a través de internet ofrecida por Ngrock gratuitamente [9], de esta forma se realizaron pruebas de las distintas peticiones y también el funcionamiento como webhook del servidor. Además cuenta con una interfaz que permite observar las respuestas a cada petición, y así validar si la comunicación entre las distintas partes, funciona correctamente. En la Fig.8 se puede observar la interfaz de Ngrock con algunas peticiones realizadas.

```
ngrok (Ctrl+C to q)
Help shape K8s Bindings https://ngrok.com/new-features-update?ref=k8s
Session Status online
Account Santiago (Plan: Free)
Update update available (version 3.15.1, Ctrl-U to update)
Version 3.14.0
Region South America (sa)
Latency 46ms
Web Interface http://127.0.0.1:4040
Forwarding https://9d82-45-178-1-13.ngrok-free.app -> http://localhost:3000

Connections
  ttl   opn   rt1   rt5   p50   p90
  10    0     0.00  0.01  5.02  5.03

HTTP Requests
-----
03:38:14.917 -03 POST /webhook/mercadopago 200 OK
03:38:00.216 -03 POST /comprobarDNI       200 OK
03:37:53.371 -03 POST /comprobarDNI       200 OK
03:37:35.016 -03 POST /comprobarDNI       200 OK
```

Fig. 8: Interfaz Ngrok

Inicialmente las pruebas de las peticiones a la API de Mercado Pago Developers para la creación de las sucursales, las cajas y las órdenes de pago se realizaron utilizando “Postman”, plataforma que permite realizar peticiones y ver su respuesta [10], dicha herramienta también se utilizó para probar las peticiones realizadas entre el servidor y el microcontrolador.

La integración de un buzzer probó tener gran utilidad para las pruebas ya que tener una señal sonora del correcto funcionamiento permitió mantener mayor fluidez en las pruebas, por lo que se optó por incluirlo definitivamente en el sistema [6].

Se ha elaborado una maqueta, Fig. 8 para representar y demostrar el funcionamiento del sistema, permitiendo visualizar de manera tangible cómo operan todos los componentes y cómo se integran en el sistema general. Se puede observar en la Fig.9.



Fig. 9. Maqueta Funcional AutoTicket

En el QR de la Fig.10, se puede visualizar algunas pruebas hechas y varias situaciones posibles que se pueden dar, las cuales fueron explicadas anteriormente.



Fig. 10. Pruebas de AutoTicket

VIII. ESCALABILIDAD

Inicialmente, el desarrollo de este proyecto se pensó con un único objetivo, tal como se menciona en la introducción. Sin embargo, a medida que el proyecto avanzaba, se hizo evidente su potencial para escalar, revelando su adaptabilidad a diversas aplicaciones relacionadas con la venta automatizada.

Esta adaptabilidad se manifiesta tanto en las implementaciones de hardware como de software. Por ejemplo, el sistema podría incorporar autenticación biométrica, como reconocimiento de huellas dactilares o tecnología RFID, como alternativas al teclado matricial. En cuanto al software, la flexibilidad del sistema permite su operación con cualquier tipo de servidor, facilitando la gestión de múltiples ofertas de venta y variaciones de precios.

La escalabilidad de este proyecto abre numerosas posibilidades. Podría ampliarse para coordinar múltiples sistemas de "AutoTicket", permitiendo que todas las transacciones se dirijan a la misma cuenta o a diferentes cuentas según sea necesario.

El proyecto ofrece la capacidad de satisfacer una amplia gama de necesidades, con el potencial de soportar aplicaciones mucho más amplias de lo inicialmente previsto.



IX. CONCLUSIONES

En este proyecto se desarrolló un sistema con las capacidades necesarias para poder llevar a cabo ventas automatizadas.

Se pudo implementar el sistema completo dentro de una ESP32 y un servidor en JavaScript dando la posibilidad de probar el sistema con pocos recursos.

Se han cumplido los objetivos inicialmente propuestos, dando solución a la problemática. Además en el proceso se reveló el potencial que brindan la combinación de las tecnologías utilizadas, por lo que se puede asegurar que AutoTicket tiene mucho potencial por explotar.

REFERENCIAS

- [1] Espressif Systems. (2024). ESP32 datasheet. Obtenido de https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [2] OpenJS Foundation. (2024). Node.js. Obtenido de <https://nodejs.org/en/about>
- [3] Mercado Libre. (2024, 8 de agosto). Mercado Pago Developers. Obtenido de <https://www.mercadopago.com.ar/developers/es/reference>
- [4] Librería WiFi.h y HTTPClient.h Obtenido de <https://github.com/espressif/arduino-esp32/blob/master/libraries/README.md#httpclient>
- [5] Parallax Inc. (2011, 16 de diciembre). Product documentation. Obtenido de <https://cdn.sparkfun.com/assets/fff/a/5/0/DS-16038.pdf>
- [6] (s.f.). Obtenido de <https://www.quick-teck.co.uk/Management/EEUploadFile/1420788438.pdf>
- [7] VISHAY. (2013, 18 de marzo). LCD016N002BCFHE. Obtenido de <https://www.vishay.com/docs/37484/lcd016n002bcfh.pdf>
- [8] SQL. (2024, 10 de agosto). SQLite. Obtenido de <https://www.sqlite.org/index.html>
- [9] Ngrok Docs. Obtenido de <https://ngrok.com/docs/>
- [10] What is Postman?. Obtenido de <https://www.postman.com/home>
- [11] Webhooks to revolutionize the web. Obtenido de <https://web.archive.org/web/20150826052314/http://progrium.com/blog/2007/05/03/web-hooks-to-revolutionize-the-web/>
- [12] OpenJs Foundation. (s.f.). Express. Obtenido de <https://expressjs.com/>
- [13] OpenJs Foundation. (s.f.). Express. Obtenido de <https://github.com/expressjs/body-parser>
- [14] The Axios Project: Copyright © 2014-present Matt Zabriskie and contributors. (s.f.). Axios. Obtenido de <https://axios-http.com/es/>
- [15] Librería Keypad.h. Obtenido de <https://www.arduino.cc/reference/en/libraries/keypad/>
- [16] Librería LiquidCrystal_I2C.h. Obtenido de <https://www.arduino.cc/reference/en/libraries/liquidcrystal-i2c/>



APLICACIONES DE MICROCONTROLADORES EN SISTEMAS DE COMUNICACIÓN POR LUZ VISIBLE BIDIRECCIONAL LED A LED

1st Andrés Felipe Isaza Piedrahita
Facultad de ingeniería
Institución Universitaria ITM
Medellín, Colombia
andresisaza34307@correo.itm.edu.co

2nd Francisco Eugenio Lopez
Facultad de ingeniería
Institución Universitaria ITM
Medellín, Colombia
franciscolopez@itm.edu.co

3rd Roger Alexander Martinez
Facultad de ingeniería
Institución Universitaria ITM
Medellín, Colombia
rogermartinez@itm.edu.co

Resumen—En el ámbito de la comunicación por luz visible (VLC), los diodos emisores de luz (LEDs) son comúnmente utilizados como transmisores de información, mientras que los receptores suelen ser fotodiodos, cámaras, paneles solares y fotoceldas, lo que típicamente resulta en enlaces unidireccionales. Sin embargo, la implementación de comunicación bidireccional LED a LED abre nuevas posibilidades, permitiendo que los LEDs actúen tanto como emisores como receptores. Esto es posible gracias a la integración de microcontroladores, que gestionan el comportamiento de los LEDs y optimizan el sistema.

En esta presentación, se describe un sistema innovador que utiliza un LED RGB de 5 mm como un dispositivo multifuncional para comunicación bidireccional y recolección de energía. Mediante el uso de un microcontrolador Arduino Nano, se controla la polarización del LED, lo que permite su operación tanto en modos de transmisión como de recepción. Además, el microcontrolador facilita la recolección de energía mediante técnicas de recolección de energía, aprovechando la capacidad fotovoltaica del LED RGB para cargar un capacitor. Este trabajo destaca cómo los microcontroladores no solo permiten una comunicación efectiva entre LEDs, sino que también facilitan la integración de sistemas de recolección de energía, demostrando su potencial en aplicaciones avanzadas de VLC y gestión energética. La presentación abordará la innovación de usar el LED RGB como divisor de potencia, la implementación del sistema con microcontroladores y la aplicación a un sistema de acceso.

Index Terms—Arduino Nano, LED a LED, acumulación de energía, circuito conmutador.

I. INTRODUCCIÓN

La comunicación por luz visible (VLC) está emergiendo como una solución innovadora en el campo de las telecomunicaciones, ofreciendo una alternativa a las tecnologías basadas en radiofrecuencia. Utilizando diodos emisores de luz (LEDs) para transmitir datos, VLC permite una comunicación rápida y eficiente a través de la luz visible, aprovechando la infraestructura de iluminación existente [1].

Tradicionalmente, los sistemas VLC han dependido de fotodetectores especializados, como fotodiodos o paneles solares, para recibir la información emitida por los LEDs. Sin embargo, estos enfoques presentan limitaciones, principalmente en términos de unidireccionalidad [2]. La capacidad de los LEDs para actuar simultáneamente como emisores y receptores transforma este paradigma, abriendo nuevas posibilidades para la comunicación de datos y la integración de funciones adicionales, como la recolección de energía.

En este trabajo, se presenta un sistema de comunicación bidireccional LED a LED, el cual emplea un Arduino Nano para gestionar el proceso de transmisión de datos, recepción y elevación de tensión en el modo de generación de energía.

Este sistema es particularmente relevante en aplicaciones donde la energía disponible es limitada o donde se busca reducir el consumo energético, como en dispositivos de Internet of Things (IoT). El uso de LEDs como transmisores y receptores, combinado con la capacidad de almacenar y reutilizar energía, propone una solución

innovadora para la comunicación en ambientes de baja potencia. El presente artículo detalla el diseño, implementación y evaluación de este sistema, destacando las ventajas de integrar microcontroladores como el Arduino Nano en la gestión de comunicación y energía en sistemas embebidos.

II. MODELO DEL SISTEMA

El modelo del sistema diseñado y construido se muestra en la fig 1. El sistema consiste en 2 PCs, desde los cuales por medio del monitor serial del **Arduino IDE** se pueden enviar y recibir mensajes en forma bidireccional, utilizando LEDs RGB. La condición para la transmisión de datos es que mientras un LED esta enviando el dato el otro debe estar en modo receptor, para esto se utiliza el Circuito conmutador el cual permiten la actuación del LED como sensor y como transmisor.

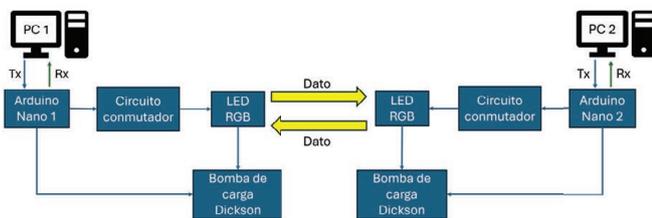


Fig. 1: Esquema de comunicación y acumulación de energía en el receptor

Cuando el LED esta en modo receptor, debe tener la capacidad de decodificar datos (actuar como sensor) y acumular energía (funcionar como panel solar), esto se conoce como divisor de potencia donde en algunos casos es un concepto [3] o se requieren elementos de hardware como amplificadores de transimpedancia, elevadores DC-DC y gestionar la programación [5] para que el LED pueda funcionar en sus tres regiones [4].

A. Innovación: Uso del LED RGB como divisor de potencia

Debido a la complejidad inherente tanto en hardware como en software para el control simultáneo del LED en su función de emisor-receptor y generador de energía, surge la consideración de emplear el LED RGB. Este componente incorpora tres LEDs distintos (rojo, verde y azul) en su estructura. La utilización de este tipo de LED ofrece la ventaja de ejecutar la división de potencia de manera simplificada con las operaciones de comunicación y acumulación de voltaje, tal como se ilustra en la fig. 2.



Fig. 2: LED RGB como generador de energía y comunicación bidireccional por luz visible

Se utiliza el canal con longitud de onda de color rojo debido a que su responsividad es mas alta con respecto a los otros canales y los canales verde y azul son utilizados para generar energía.

B. Circuito conmutador

Se procede a replicar el experimento previamente realizado en [6], enfocado en la transmisión unidireccional de señales de audio de un LED a otro. Con base en la valiosa experiencia adquirida durante la ejecución de este experimento, se emprende el diseño y construcción de un circuito conmutador basado en transistores BJT que permite polarizar el LED en la region inversa para que funcione como sensor y en región directa para que funcione como transmisor. Como resultado, se logra la elaboración del circuito ilustrado en la fig. 3.

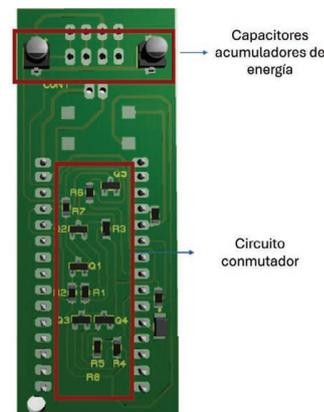


Fig. 3: Prototipo del circuito bidireccional y acumulador de energía

C. Funcionamiento del circuito

Cuando el pulsador está presionado mantiene un estado de "uno lógico", el LED cambia a modo de emisión lo que hace que el LED se encienda y envíe el dato "123" como se muestra en la fig. 4a.

Cuando el pulsador no está presionado, el circuito mantiene un estado de "cero lógico", lo que hace que el LED entre en modo sensor y cuando recibe el dato "123" se enciende un LED indicador confirmando que el dato es seguro como se muestra en la fig. 4b .

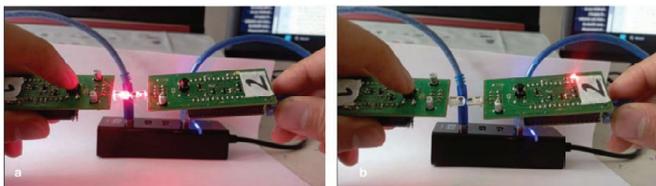


Fig. 4: Comunicación entre tarjetas

D. Comunicación bidireccional LED a LED utilizando modulación OOK

En comunicación LED a LED se han utilizado algunas técnicas de modulación binaria como OFDM en un LED RGB como sensor [7]. En [8] comparan la modulación OOK y PPM en un LED rojo como sensor y eligieron la modulación PPM debido a que tienen una tasa de bits más alta que la modulación OOK en condiciones de SNR. Pero la modulación PPM requiere sincronización entre los sistemas de transmisión y recepción. La modulación OOK es una forma de modulación de amplitud, y es mucho más simple que las técnicas de modulación de frecuencia. Tanto [9], [10] optaron por implementar OOK por su simplicidad, buen rendimiento promedio y componentes de hardware menos costosos.

E. Implementación de OOK en comunicación bidireccional LED a LED

La comunicación bidireccional LED a LED en este sistema se logra mediante el uso de la modulación OOK, complementada por un circuito con transistores BJT que permite que el LED funcione como emisor y receptor. A continuación, se describen los componentes y el flujo de operación del sistema:

1) Componentes del sistema

- LED RGB como Transmisor/Receptor: En este sistema, el LED RGB actúa tanto como emisor como receptor. Esto es posible gracias al Circuito conmutador con transistores BJT (2N3904 y 2N3906), que conmutan el LED entre los modos de transmisión y recepción. En modo receptor, el LED RGB no solo detecta señales de luz sino que también puede generar una pequeña corriente eléctrica acumulando tensión.
- Circuito de Control con Transistores BJT: Los transistores BJT se utilizan para conmutar la funcionalidad del LED entre transmisor y receptor. El transistor NPN (2N3904) controla la activación del LED como emisor, mientras que el transistor PNP (2N3906) permite la operación inversa para recibir señales luminosas.

- Modo conmutador: Un pin de control del microcontrolador alterna entre los modos de transmisión y recepción, permitiendo la comunicación bidireccional. Este conmutador maneja los transistores para ajustar la funcionalidad del LED.

En la fig 5 se muestra la arquitectura de las señales que envía el microcontrolador al circuito bidireccional y la señal analoga que recibe del acondicionamiento del LED como sensor a partir de un transistor NPN, también se puede apreciar como se genera una señal de reloj para la acumulación de tensión.

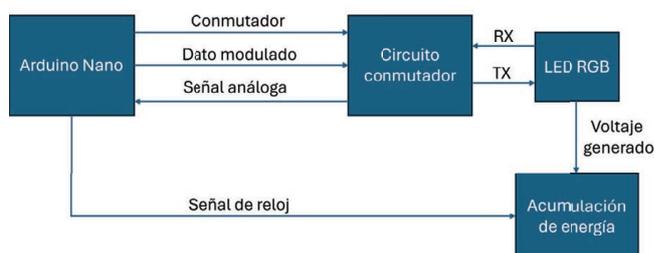


Fig. 5: Arquitectura del microcontrolador y el circuito conmutador

2) Proceso de Transmisión

En el modo de transmisión, el LED RGB envía datos utilizando la modulación OOK, modulando su luz para representar los datos binarios:

- Inicio de la Transmisión: El microcontrolador activa el transistor NPN para permitir que el LED RGB emita luz. El LED se apaga brevemente para indicar el inicio de la transmisión.
- Envío de Bits: Utilizando la modulación OOK, se transmiten pulsos de luz para representar cada bit del byte. Si el bit es '1', el LED se enciende; si es '0', el LED permanece apagado. El control preciso del LED es manejado por los transistores BJT.
- Finalización de la Transmisión: Después de enviar todos los bits, el LED se mantiene encendido para indicar que la transmisión ha terminado. Los transistores se ajustan para detener el flujo de corriente en modo de transmisión.

En la fig. 6 se muestra el diagrama de flujo del modo transmisor.

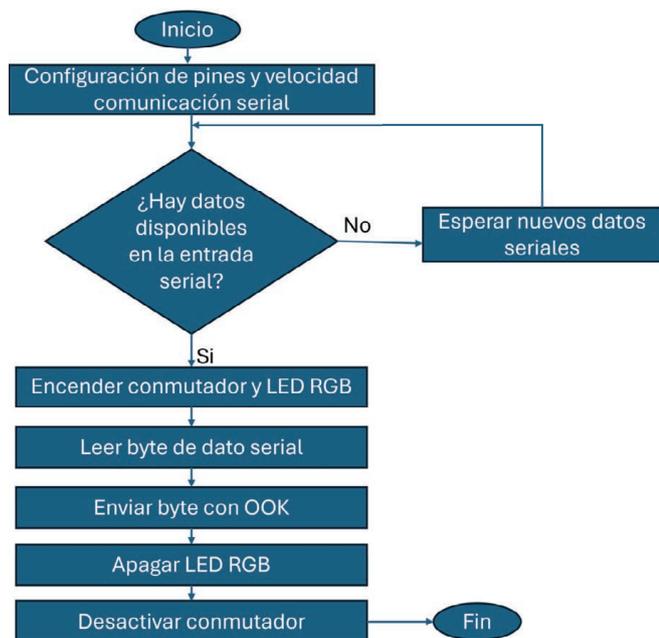


Fig. 6: Diagrama de flujo transmisor

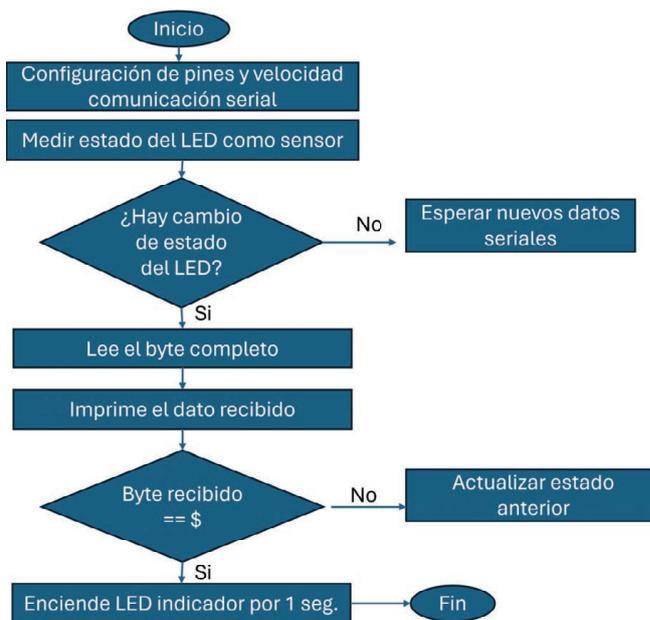


Fig. 7: Diagrama de flujo receptor

3) Proceso de recepción

En el modo de recepción, el sistema cambia el estado de los transistores para aprovechar la capacidad del LED de recibir luz y generar una corriente eléctrica:

- **Detección de Señal Lumínica:** El LED RGB actúa como un fotodetector. Los cambios en la luz recibida por el LED se convierten en señales eléctricas. El transistor PNP se activa para facilitar la detección de estas señales.
- **Acumulación de Tensión:** En el modo receptor, el LED RGB puede acumular tensión cuando recibe luz, aprovechando su capacidad de generar corriente cuando se expone a la luz. Esta característica es clave para aplicaciones donde la recuperación de energía es necesaria.
- **Reconstrucción de Bytes:** Los bits de datos se reconstruyen en el microcontrolador al medir la señal recibida por el LED RGB. Se sigue un ciclo de temporización para asegurar que los datos se lean de manera sincronizada.
- **Impresión de Datos Recibidos:** Una vez que se han reconstruido los bytes, se envían a través del puerto serie para su visualización. Si se detecta un carácter específico (como '\$'), un indicador visual se activa para confirmar la recepción exitosa.

En la fig. 7 se muestra el diagrama de flujo del modo receptor.

F. Configuración de la bomba de carga Dickson

Con la finalidad de aumentar la tensión acumulada en los capacitores, se opta por aplicar la técnica de bomba de carga Dickson mostrada en la fig. 8 la cual se compone de diodos tipo schotky y una serie de capacitores.

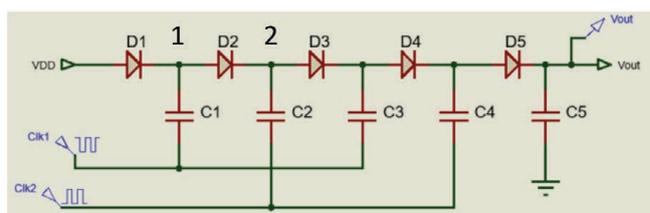


Fig. 8: Circuito bomba de carga Dickson

Esta técnica permite elevar el voltaje generado por los LEDs y acumularlo en el capacitor C5. Se genera una señal oscilante con el Arduino Nano, con capacidad adicional para recibir y transmitir mensajes. A través de un transistor, esta señal se desfasa 90° con respecto a la original, lo que acelera la carga de los capacitores. En la fig. 9. se muestra la señal de reloj en color amarillo que hace funcionar la bomba de carga Dickson y el mensaje enviado de color azul, cada vez que se envía el mensaje el conmutador esta en un uno logico.

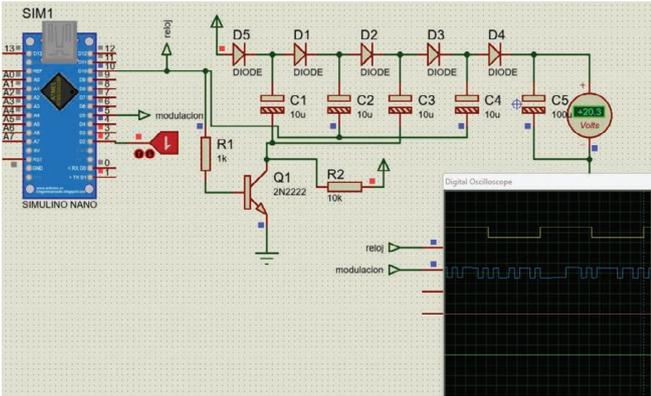


Fig. 9: Señales para bomba de carga Dickson

III. RESULTADOS

Se diseño y construyo con exito un modelo para demostrar la comunicación bidireccional entre LEDs y la acumulación de energía. El objetivo era establecer la comunicación de dos dispositivos por medio del espectro visible usando el mismo LED. En la etapa inicial se utilizo el microcontrolador Arduino Nano y Arduino IDE para recibir y transmitir un texto fijo. Posteriormente se utilizo un proceso paso a paso para escribir texto en el terminal serial de Arduino IDE y enviarlo a una tasa de velocidad de 250 kbps. La distancia entre ambos LEDs se fijo en 2 cms, logrando acumular 19 V con la bomba de carga Dickson. En las siguientes figuras se muestran los resultados de transmisión recepción de dato con la generación de energía.

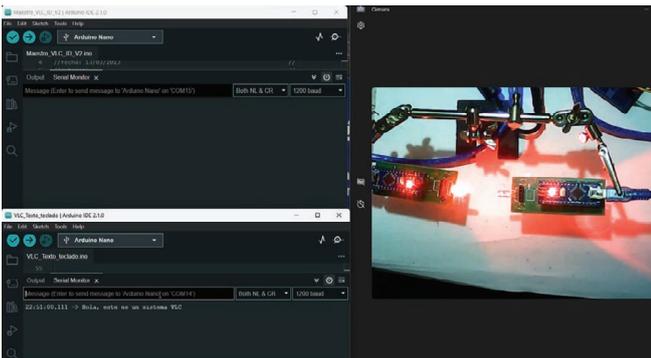


Fig. 10: Envío de dato y recepción.

En la fig. 10 se muestra como se envía el mensaje "Hola, este es un sistema VLC" de la tarjeta de izquierda a derecha.

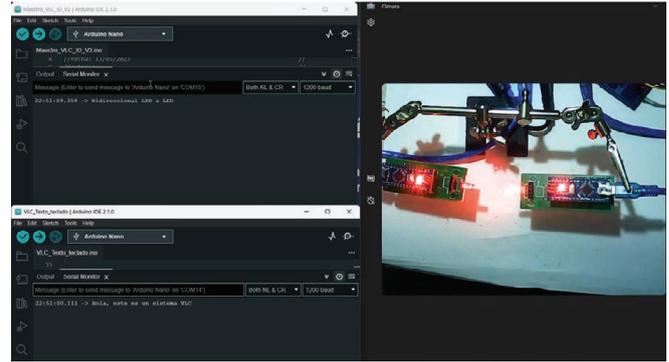


Fig. 11: Reenvío de dato

En la fig. 11 se puede ver la comunicación bidireccional enviando de derecha a izquierda el mensaje "Bidireccional LED a LED".

Finalmente en la fig. 12 se puede apreciar como se consigue la recepción de datos y acumulación de energía.

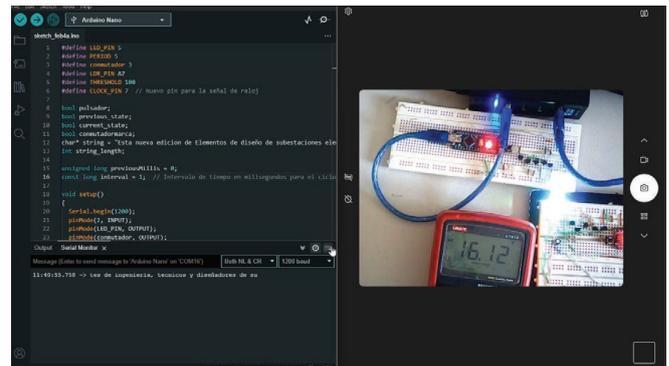


Fig. 12: Comunicación y acumulación de energía

IV. CONCLUSIÓN

Este estudio demuestra la viabilidad de la transmisión y recepción de datos a través de LEDs utilizando un circuito basado en transistores BJT, mientras se aprovecha la capacidad de generación de energía del mismo sistema. Se ha logrado una velocidad de transmisión de datos de 250 kbps, lo que indica una eficiente capacidad de comunicación para un sistema VLC-ID.

La implementación y análisis de la bomba de carga Dickson demostraron su eficacia para elevar y acumular la tensión generada por el LED RGB utilizado como generador de energía. Esta técnica se adaptó específicamente a los niveles de tensión alcanzados, permitiendo mejorar la eficiencia de la recolección de energía. Además, se observó que, a pesar de las limitaciones iniciales en otras configuraciones de convertidores, la bomba de carga Dickson fue una solución viable y eficiente para



este propósito, mostrando un potencial significativo en la generación y acumulación de energía.

RECONOCIMIENTO

Esta investigación y la APC fueron financiadas por el Instituto Metropolitano de Tecnología [beca P23202], Medellín, Colombia

REFERENCIAS

- [1] Matheus, L. E. M., Vieira, A. B., Vieira, L. F., Vieira, M. A., & Gnawali, O. (2019). Visible light communication: concepts, applications and challenges. *IEEE Communications Surveys & Tutorials*, 21(4), 3204-3237. 01
- [2] Mir, M. S., Majlesein, B., Guzman, B. G., Rufo, J., & Giustiniano, D. (2021, June). LED-to-LED based VLC Systems: Developments and open problems. In *Proceedings of the Workshop on Internet of Lights* (pp. 1-6).
- [3] Ndjiongue, A. R., & Ngatched, T. M. (2020, October). Led-based energy harvesting systems for modern mobile terminals. In *2020 International Symposium on Networks, Computers and Communications (ISNCC)* (pp. 1-6). IEEE.
- [4] Fan, X., Leon-Salas, W. D., Fischer, T., & Perez-Olvera, A. (2016, October). An LED-based image sensor with energy harvesting and projection capabilities. In *2016 IEEE SENSORS* (pp. 1-3). IEEE.
- [5] Haydaroglu, I., & Mutlu, S. (2014). Optical power delivery and data transmission in a wireless and batteryless microsystem using a single light emitting diode. *Journal of Microelectromechanical Systems*, 24(1), 155-165.
- [6] Hasegawa, M. (2019, May). Satisfactory role of LEDs as a light receiving component and their uses in science demonstration experiments for educational purposes. In *Education and Training in Optics and Photonics* (p. 11143_79). Optica Publishing Group.
- [7] Sabne, A., Panda, A., & More, V. (2019, July). Simplified wavelength division multiplexing in visible light communication by using RGB LED as frequency selective receiver. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-5). IEEE.
- [8] Liang, S., Li, S., Pan, Q., & Xu, Z. (2019, July). Self-powered Weak Light LED-LED Communications. In *2019 15th International Conference on Telecommunications (ConTEL)* (pp. 1-5). IEEE.
- [9] Kowalczyk, M., & Siuzdak, J. (2015, July). VLC link with LEDs used as both transmitters and photo-detectors. In *2015 Seventh International Conference on Ubiquitous and Future Networks* (pp. 893-897). IEEE.
- [10] Milovančev, D., Vokić, N., & Schrenk, B. (2021, June). VLC using a red SMD LED as 2.48 Gb/s multi-carrier receiver and wide field-of-view 25 Mb/s OOK transmitter. In *Optical Fiber Communication Conference* (pp. F1A-6). Optica Publishing Group.



Módem para telelectura de medidores eléctricos

Leonardo Daniel Del Sancio

Noanet S.A.

Universidad Nacional de Tucumán

Tucumán - Argentina

ldelsancio@herrera.unt.edu.ar

Resumen—El trabajo aborda la problemática de la lectura de medidores eléctricos en la provincia de Tucumán, Argentina, donde las lecturas se realizan manualmente, lo que implica altos costos operativos, errores humanos, y problemas de fraude. Para mitigar estos desafíos, el autor propuso el diseño de un módem que permite la telelectura automática de medidores eléctricos, enviando los datos recopilados a un servidor web mediante tecnología 4G [1]. Este dispositivo está destinado a la empresa Noanet [2] y busca reducir la necesidad de despliegue de personal en campo, mejorando la eficiencia y precisión del proceso de lectura de medidores.

Palabras clave--Módem, telelectura, medidores eléctricos, IoT.

I. INTRODUCCIÓN

Actualmente las lecturas de medidores en la provincia de Tucumán se realizan de manera presencial, in-situ. Esta metodología requiere un gran despliegue de unidades recorriendo muchos kilómetros diariamente. En muchos casos los tableros de mediciones de grandes clientes no tienen acceso público y se debe solicitar un permiso especial para poder ingresar a las instalaciones. Esta problemática genera demoras en las lecturas y en varias ocasiones el lectorista debe regresar reiteradas veces para poder efectuar la medición. En caso de no lograrla, la empresa debe estimar los valores de consumo con todas las dificultades que esto conlleva. Todas las compañías distribuidoras de energía a lo largo y ancho de Argentina enfrentan problemas en común que acaecen en malas facturaciones y degradación en la calidad de servicio. Entre estos problemas, son destacables los citados a continuación:

- **Fraude en medición por parte de los consumidores:** El fraude da lugar a una facturación irreal del servicio prestado por las distribuidoras. Esto se pronuncia cuando es

realizado por grandes consumidores y a lo largo del tiempo se vuelve muy difícil de sobrellevar, detectar e impugnar, a fin de lograr una medición justa y equitativa. Sólo en actividades de fraude y conexiones clandestinas se pierden alrededor de 20 mil millones de pesos en Argentina. Así, estas prácticas fraudulentas pueden arremeter contra la calidad de servicio prestado y poner a los usuarios en una situación peligrosa que puede incluso costar vidas.

- **Costo y error humano:** Los empleados de estas empresas, conocidos como lectoristas, realizan ejercicios rutinarios y de riesgo eléctrico, que pueden estar sujetos a error simplemente por el factor de la naturaleza humana. Uno de ellos es el error de observación en la medición, que se puede corregir con una mejora tecnológica. Otro defecto en el sistema actual son las faltas de mediciones y/o la estimación de las mismas, debido a la mala logística a la hora de trazar rutas (dejando así de consultar los medidores de ciertos sectores o zonas) y/o pasar por alto algún tablero de difícil acceso.
- **Escasa cantidad de datos de los consumidores:** Debido a la naturaleza de la toma de lectura por parte del personal en las distribuidoras, los datos obtenidos por cada lectorista, en el mejor de los casos, son mensuales dando lugar a una baja tasa de actualización. Esto lleva a la empresa a realizar trabajos como refacturación o inferencias en los consumos de los usuarios. Potenciando dichos problemas se encuentran además el error humano y las prácticas fraudulentas, llevando a grandes pérdidas en el cobro, sobre todo en grandes consumidores.

II. ESTRUCTURA GENERAL DEL SISTEMA

En la figura 1 se ilustra el diagrama en bloques general del sistema. Se observa que el elemento principal es el módem de comunicación encargado de interactuar con el medidor eléctrico y reportar los datos a un servidor privado de la empresa Noanet.

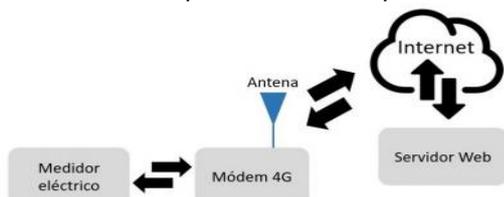


Fig. 1. Diagrama simplificado del sistema.

El software de gestión de datos llamado telecenter, es el encargado de iniciar la comunicación con el módem para proceder a la lectura del medidor eléctrico. Para la comunicación entre el módem y el servidor web se utiliza el protocolo ANSI 12.21 [3], que es el estándar nacional estadounidense para transportar datos de tablas a través de redes, con el propósito de interoperabilidad entre módulos de comunicaciones y medidores. Este estándar utiliza el cifrado AES [4] para permitir comunicaciones sólidas y seguras, incluidas la confidencialidad y la integridad de los datos.

A. Diagrama de flujo general

A continuación, se detallan las funciones y tareas principales del firmware y sus criterios de programación. En la figura 2 se ilustra el diagrama de flujo del programa principal.

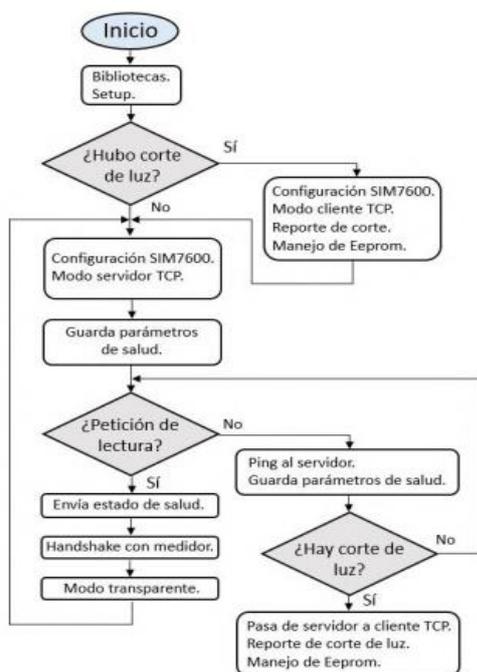


Fig. 2. Diagrama de flujo del firmware

III. DISEÑO E IMPLEMENTACIÓN

A. Firmware de control

Para el desarrollo del firmware se evaluaron todos los requerimientos y las tareas que debía realizar el procesador, se probaron diferentes bibliotecas, sistemas operativos y la API de la empresa Espressif-ESP-IDF[5]. Dada la simplicidad del manejo de esta API, sumado a la variedad de drivers y bibliotecas disponibles en internet, se optó por programar el firmware en bare-metal. En el diseño del firmware se organizó y se empleó una estructura de capas para poder explicar las partes que componen el sistema. En la figura 3, se observa la estructura de capas que se detallan a continuación:



Fig. 3. Arquitectura IOT.

- Capa de aplicación: es la capa de más alto nivel, aquí es donde se implementa toda la lógica del sistema. Está compuesta por módulos como máquinas de estado, interrupciones, empaquetamiento de datos, funciones, etc.
- Capa intermedia: está compuesta por un conjunto de bibliotecas y funciones que se encargan de controlar las comunicaciones UART, Wi-Fi, Bluetooth, 4G entre otras.
- Capa de bajo nivel: en esta capa es donde ocurre la comunicación con el hardware mediante la API ESP-IDF. En esta API se incluyen los controladores y drivers para el manejo de los puertos GPIO, UART, PWM, SPI, I2C entre otros.

B. Trama de datos

A continuación, se describen los paquetes de datos enviados por el canal de comunicación entre el módem y el software telecenter. Como se mencionó anteriormente estos paquetes de datos deben cumplir con el estándar ANSI C12.19[6].

En la figura 4 se ilustra una trama enviada desde telecenter al módem de comunicación.

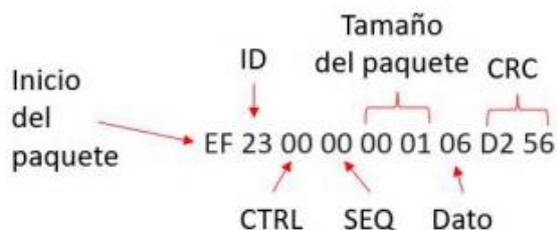


Fig. 4. Trama enviada desde el módem al servidor.

Se observa en la trama que el inicio del paquete de datos va encabezado por el byte 0xEF, luego el ID asignado para el módem de comunicación es 0x23. Para este ejemplo la trama que devuelve el módem es un byte de ACK con el valor 0x06. Todos los paquetes de datos deben cumplir con los estándares mencionados con anterioridad.

C. Arquitectura IOT empleada

Existen diferentes sistemas o arquitecturas que nos permiten establecer una comunicación entre el módem con el software telecenter ubicado en el servidor web. Aunque solo veamos la interfaz de hardware, detrás hay una compleja arquitectura donde la información fluye de un punto a otro. El sistema global, desde el medidor eléctrico hasta el telecenter, transforma una variable física (tensión y corriente) a una variable digital cuya información viaja por un medio físico a un medio virtual. A este proceso se lo divide en cuatro capas o fases que se ilustran en la figura 5.



Fig. 5. Arquitectura IoT empleada.

- Cosas / dispositivos: es la parte visible a los usuarios. Dentro de este bloque se encuentra el medidor eléctrico y los puertos de comunicación del módem.
- Puntos de acceso: los puntos de acceso permiten la conectividad de las cosas, objetos o dispositivos al servidor web. En este trabajo se utilizan dos tecnologías, la comunicación WiFi y la comunicación celular 4G LTE.
- Procesamiento de datos: el eje central del IoT es la estructura de datos que se controlan en la etapa de procesamiento del módem. El buen funcionamiento del sistema de telelectura dependerá de las capacidades en

la gestión de estos datos y el uso inteligente que se haga de ellos.

- Aplicaciones: en este bloque se encuentran las dos aplicaciones destinadas para este trabajo: telecenter cuya función es conectar y mostrar los datos finales a los usuarios, y la aplicación Android que se utiliza para realizar los test y las pruebas en terreno.

D. Manufactura

Si bien una empresa del grupo de Noanet tiene capacidad para importar, los constantes cambios en las políticas aduaneras del país llevaron a minimizar la dependencia de proveedores extranjeros, motivo por el cual se optó por realizar la manufactura con la empresa Nubiot S.A.[6]. Esta empresa, ubicada en la provincia de Tucumán, se encarga de la fabricación del circuito impreso, de la compra y montaje de los componentes entregando un producto final listo para ser montado en el gabinete. La primera producción fue de 1500 unidades. En la figura 6 se ilustra el circuito impreso terminado.



Fig. 6. Manufactura del producto.

IV. CONCLUSIONES Y TRABAJO A FUTURO

El proyecto concluye con éxito en el desarrollo de un dispositivo que optimiza la telelectura de medidores eléctricos, reduciendo costos operativos y mejorando la precisión en la facturación de energía. Se destacan como posibles trabajos futuros la integración del módem con otros tipos de medidores y la mejora del software de gestión de datos para hacerlo más versátil.

En resumen, este trabajo ofrece una solución

tecnológica innovadora para la telelectura de medidores eléctricos, alineada con las necesidades actuales de las empresas distribuidoras de energía en Argentina.

- En: (2012). URL:
"<https://webstore.ansi.org/Standards/NEMA/ansic12222012>".
[7] Nubiot S.A. Ruta 9 KM 1308 – Barrio: Country la Herencia
Tafí Viejo - Tucumán. eduardo@nubiot.io

V. RESULTADOS

A continuación, se ilustran algunas imágenes de la instalación de módem de comunicación en diferentes tipos de medidores. Este proceso está a cargo de la empresa de distribución eléctrica de la provincia. En la figura 7 se observa la instalación de un medidor con protocolo ANSI.



Fig. 7. Imagen de la instalación final.

El despliegue de las instalaciones en terreno comenzó en el mes de abril del año 2022. Hasta el día de la fecha se instalaron más de 5000 módems fabricados por la empresa Noanet.

REFERENCIAS

- [1] «LRTC to Launch Lithuania's First Mobile WiMAX 4G Internet Service». WiMAX Forum. 3 de marzo de 2009. Archivado desde el original el 12 de junio de 2010. Consultado el 26 de noviembre de 2010.
- [2] Noanet Soluciones T.I. <http://www.noanet.com.ar>. (Visitado 07-07-2022).
- [3] ANSI C12.22 (also known as IEEE Std 1703) is an application layer protocol designed for use on electric utility meter networks of the Smart Grid URL: <https://wiki.wireshark.org/C12.22>
- [4] SocSecNumber SK LastName FK DepartmentNum. «El estándar de cifrado avanzado». En: (2001). URL: "https://techlandia.com/funciona-aes-info_215975/" (visitado 11-07-2022).K. Elissa, "Title of paper if known," no publicado.
- [5] Espressif. «El estándar de cifrado avanzado». En: (2010). URL: "https://docs.espressif.com/projects/espressif/en/latest/esp32/api_reference/index.html" (visitado 12-07-2022).
- [6] National Electrical Manufacturers Association. «Protocol Specification For Interfacing To Data Communication Networks».



Sistema de medición y registro multiplexado de termocuplas vía WiFi

1st Matias Herrera

Laboratorio de Aerodinámica y Fluidos
Universidad Tecnológica Nacional Regional Haedo
Buenos Aires, Argentina
gmf-layf@frg.utn.edu.ar

2nd Gerardo Imbrioscia

Laboratorio de Aerodinámica y Fluidos
Universidad Tecnológica Nacional Regional Haedo
Buenos Aires, Argentina
gmf-layf@frg.utn.edu.ar

3rd Jonathan Angiano

Laboratorio de Aerodinámica y Fluidos
Universidad Tecnológica Nacional Regional Haedo
Buenos Aires, Argentina
gmf-layf@frg.utn.edu.ar

4th Agustín Avendano

Laboratorio de Aerodinámica y Fluidos
Universidad Tecnológica Nacional Regional Haedo
Buenos Aires, Argentina
gmf-layf@frg.utn.edu.ar

5th Pablo Caron

Laboratorio de Aerodinámica y Fluidos
Universidad Tecnológica Nacional Regional Haedo
Buenos Aires, Argentina
gmf-layf@frg.utn.edu.ar

Abstract—Las estufas de tipo *rocket* son conocidas por su diseño simple, bajo costo y sencillez de operación. A fin de poder parametrizar su funcionamiento en lo que respecta a la distribución de temperaturas, se diseñó e implementó un sistema de medición de temperaturas, a través de un conjunto de termocuplas, y posterior registro. Debido a la cantidad de termocuplas a instrumentar, las cuales exceden las entradas digitales disponibles, se decidió utilizar multiplexores para unificar la trama de datos entrante.

La electrónica de medición y adecuación fue separada de la de resguardo, las que se comunican vía WiFi. De esta forma se redujo el nivel de exposición de electrónica a las fuentes de calor, reduciendo el tamaño del equipo que se debe proteger a altas temperaturas.

Index Terms—Temperatura, termocuplas, ESP, WiFi, Adquisición

I. INTRODUCCIÓN

Las estufas denominadas *rocket* son una alternativa económica de sistema de cocción a leña/carbón. Estos sistemas constan de un tubo central, generalmente de sección cuadrada, a modo de chimenea, un tubo lateral para la carga de combustible y una pieza móvil inferior que sirve de recolector de cenizas y entrada de aire, como se observa en la figura 1.

A fin de poder parametrizar el funcionamiento del diseño propuesto, se decidió la medición de temperatura en diferentes puntos estratégicos. Debido a la asimetría

geométrica de la estufa, se debe medir en una gran cantidad de puntos, lo que conlleva a utilizar varias termocuplas simultáneamente, superando la cantidad de entradas digitales disponibles. Para dar una solución a dicho problema, se diseñó y construyó un sistema de medición de termocuplas, adecuación haciendo uso de multiplexores, y posterior registro en memoria de los datos obtenidos.

II. DESCRIPCIÓN GENERAL

El objetivo general fue diseñar un sistema que permita tomar las mediciones de las termocuplas y registrarlas en una tarjeta de memoria junto con la fecha y hora de la medición. Dada la alta temperatura a registrar, se separó la electrónica en dos módulos: el de medición y de almacenamiento. Esto dio la ventaja de alejar la electrónica de adquisición de la fuente de calor, reduciendo el tamaño del modulo aislante térmico.

El modulo de medición es el encargado de tomar los datos de las termocuplas conectadas, acondicionar su señal y enviarlos mediante una comunicación WiFi al módulo de almacenamiento. Dicho módulo se encargará de recibir los datos mediante WiFi y almacenarlos en una tarjeta de memoria tipo SD junto con la fecha y hora de la medición. Un esquema de conexión se muestra en la figura 2).

III. DESCRIPCIÓN DE MÓDULOS

A. Módulo de medición

1) *Placa acondicionamiento de señal:* En base a la necesidad de medición de termocuplas tipo K, se desarrolló y construyó una placa con la capacidad de conectar hasta cuatro sensores, con sus correspondientes conversores ADC [1]. A fin de verificar que el módulo de termocuplas se encuentra con alimentación, se colocó un led, el cual enciende solamente al conectarse el sistema de transmisión de datos. La conexión a esta se realiza mediante el conector tipo IDC de diez terminales. La placa se puede ver en la figura 3.

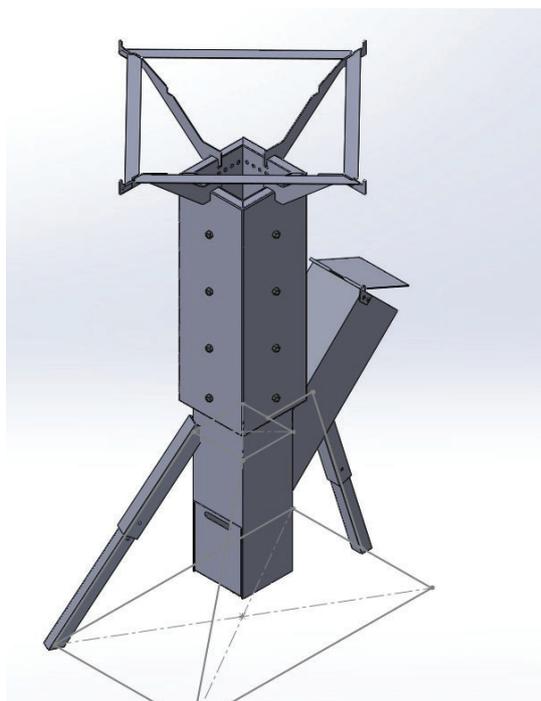


Fig. 1. Diseño de las estufas rocket.

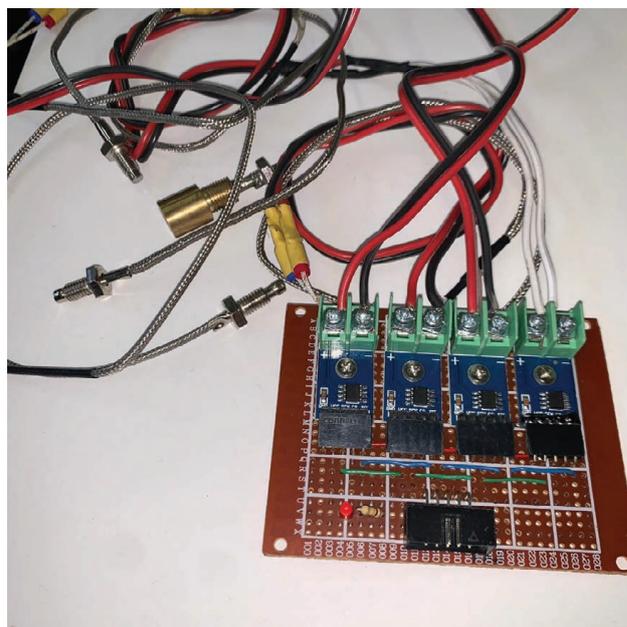


Fig. 3. Placa con termocuplas conectadas

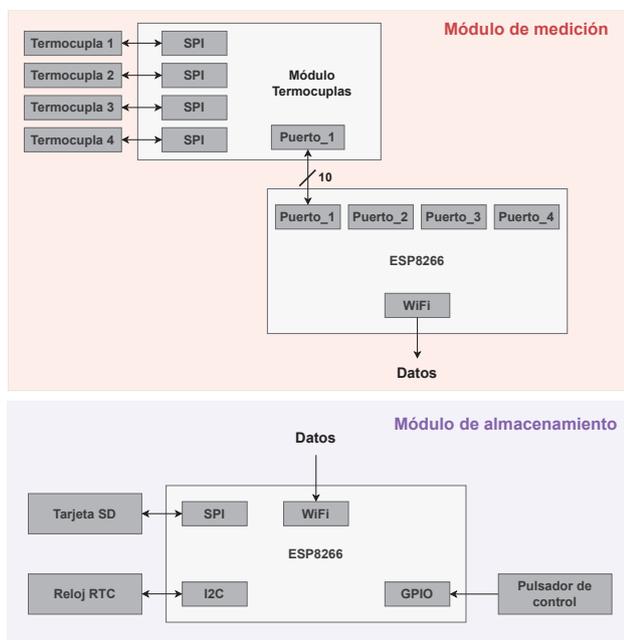


Fig. 2. Descripción general del sistema electrónico.

2) *Placa de transmisión de datos:* Este módulo está basado en la placa NodeMCU ESP8266 [2] y es el encargado de tomar las mediciones de las termocuplas y enviar los datos al módulo de almacenamiento (receptor).

Dado que la cantidad de pines del ESP8266 no son suficientes para realizar la lectura de todas las termocuplas, se utilizaron multiplexores para poder ampliar la cantidad máxima de termocuplas conectadas a ocho, pudiendo medir el sistema hasta un máximo de dieciséis. En la placa se encuentran disponibles cuatro puertos de conexión para establecer la comunicación con las placas de termocuplas como se observa en la figura 4. Este módulo se alimenta con una tensión de 5VDC en la bornera dedicada para ello con la polaridad indicada.

Al iniciar, la placa se conecta a la red WiFi creada por el módulo de almacenamiento, realiza la lectura de las termocuplas conectadas y envía los datos mediante el protocolo 'UDP' al 'receptor'.

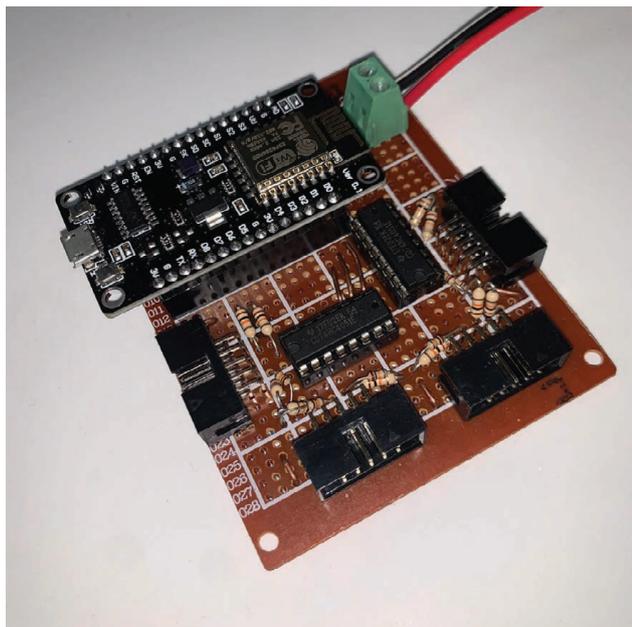


Fig. 4. Módulo de medición y sus puertos de conexión.

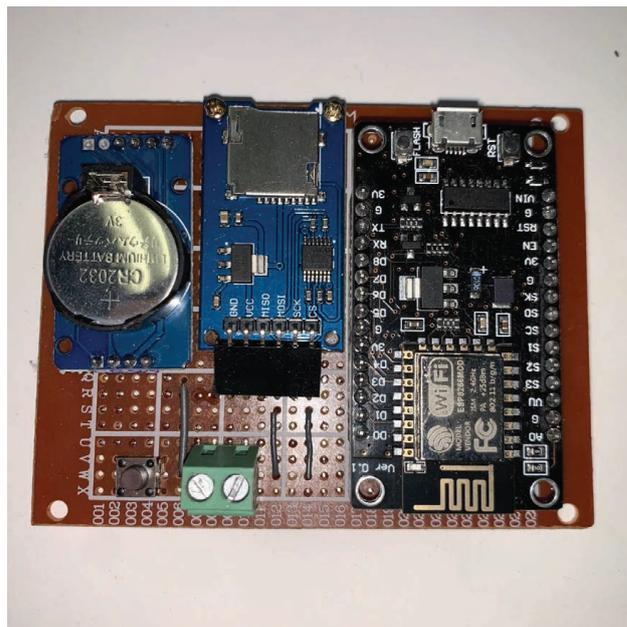


Fig. 5. Módulo de almacenamiento.

B. Módulo de almacenamiento

El módulo de almacenamiento, o receptor, es el encargado de recibir los datos y almacenarlos en la tarjeta de memoria SD junto con la fecha y hora de la medición.

Este modulo también basó en la tecnología ESP8266, el cual se alimenta con 5VDC desde la bornera o también desde el puerto micro USB. Se incorporó un reloj de tiempo real (RTC) [3] que permite tener la fecha y hora actualizada aunque se desconecte el dispositivo y así, evitar corrimientos en el tiempo. Para almacenar los datos se utilizó un dispositivo de lectura y escritura de tarjeta microSD [4].

Esta placa crea una red WiFi (Access Point) donde se conectará el Módulo de Medición para enviar los datos. La placa posee un led indicador para conocer el estado de la tarjeta de memoria SD y si se esta realizando el registro de datos. El inicio y finalización del ensayo se controla con un pulsador. El módulo se puede observar en la figura 5.

IV. DESCRIPCIÓN DEL HARDWARE

A. NodeMCU ESP8266

Se utilizó este dispositivo como núcleo del módulo de medición y el de almacenamiento. La NodeMCU es una placa de desarrollo que posee nueve pines digitales para propósitos generales. Permite utilizar protocolos de comunicación tipo I2C, SPI y UART, y además controlar los multiplexores. Su característica mas importante es que tiene integrado el módulo WiFi, en el receptor opera en modo AP (Access Point) creando la red para la comunicación y en el transmisor en modo STA (Station) conectándose a esa red.

B. Termocuplas

En el sistema se utilizaron termocuplas tipo K con el conversor analógico digital MAX6675. La temperatura máxima que se puede medir con estas termocuplas es 800°C con una resolución de 0.25°C. La comunicación con el MAX6675 se realizó mediante el protocolo SPI. Los conversores y las termocuplas se observan en la figura 3.

C. Multiplexores

Dado el requerimiento de medir un mínimo de ocho termocuplas, la cantidad de pines del ESP8266 no eran suficientes para esto. Para solucionar este limitante se agregaron dos multiplexores. Estos circuitos permiten que utilizando solo cinco pines digitales, el sistema elija con cual de las dieciséis termocuplas comunicarse. Se utilizó el 74HC4051 [5], multiplexor analógico de 8 canales con tres entradas digitales de selección y

un pin para habilitar o deshabilitar el dispositivo. El primer multiplexor se encarga de controlar las primeras ocho termocupas (Puerto 1 y 2) y el segundo de las ocho restantes (Puerto 3 y 4). Los módulos de acondicionamiento de termocupas utilizan una comunicación del tipo SPI, lo que permite a través del pin 'chip select' (CS) elegir de que termocupla tomará los datos el sistema. La placa de muestra en la figura 7.

V. COMUNICACIÓN ENTRE PLACAS

Para poder establecer la comunicación entre las placas se utilizó la tecnología de redes WiFi. El receptor al iniciarse crea una red WiFi, con un nombre y contraseña específicos. El transmisor se conecta a ésta red a través del protocolo 'User Datagram Protocol' (UDP). En nuestra aplicación particular, solo se envían datos desde el módulo de medición hacia el de almacenamiento, pero el protocolo UDP permite que el envío de información sea bidireccional.

Como corolario de la sección, se observa en la figura 7 el diagrama de conexión del módulo de medición, las conexiones del módulo de termocupas en la figura 8 y el diagrama del módulo de almacenamiento en la figura 6.

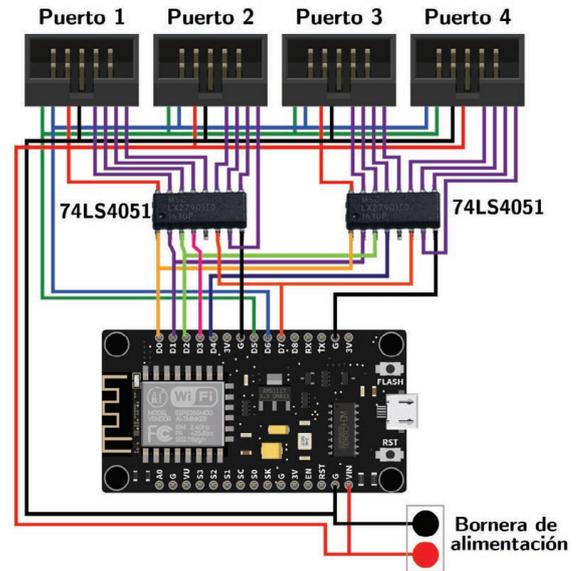


Fig. 7. Transmisor: diagrama de conexiones

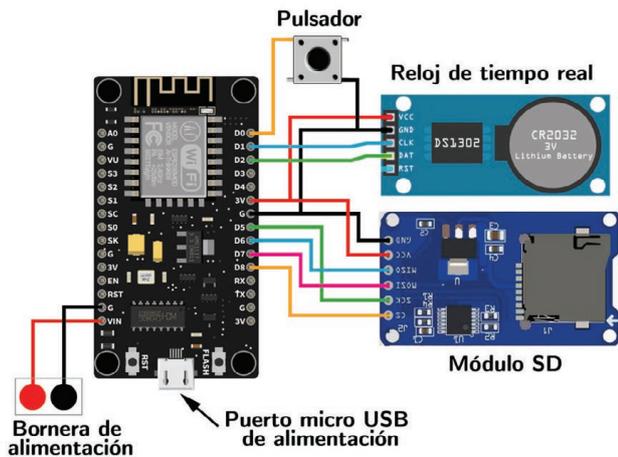


Fig. 6. Receptor: diagrama de conexiones

VI. FIRMWARE

Se desarrollaron dos firmwares para el sistema: uno correspondiente al módulo de almacenamiento y otro para el módulo de medición.

A. Módulo de medición

El módulo se inicia y establece la conexión a la red WiFi creada por el receptor. Si existe algún error de conexión, el led indicador comenzará a parpadear y el módulo reintentará conectarse. Cuando la conexión se establezca correctamente, se inicia la comunicación

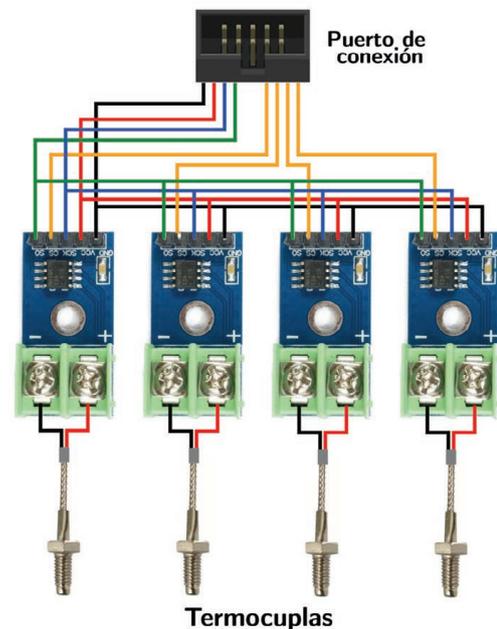


Fig. 8. Módulo de termocupas: diagrama de conexiones



con las termocuplas. El microcontrolador activa el primer multiplexor y apaga el segundo, efectúa la lectura de las primeras ocho (8) termocuplas y las almacena en un buffer. Seguidamente se deshabilita el primer multiplexor y se activa el segundo, realizando la misma operatoria descrita anteriormente. A continuación, se forma una trama con las mediciones de todas las termocuplas y se envía al módulo de almacenamiento mediante WiFi.

B. Módulo de almacenamiento

Al encenderse el dispositivo este verifica la conexión de la tarjeta SD. En caso de que ésta no esté insertada o tenga alguna falla el sistema no iniciará y el led indicador de la placa comenzará a parpadear. Si la tarjeta de memoria funciona correctamente, se crea la red WiFi con la que se realizará la comunicación entre los módulos.

Una vez finalizada la inicialización de la placa, el usuario puede operar el sistema. El led permanecerá apagado indicando que no se están almacenando datos en la tarjeta SD. Para comenzar el ensayo y registro de datos el usuario debe presionar el pulsador de la placa, luego el led se encenderá y permanecerá encendido durante todo el ensayo. Al presionar nuevamente el botón se finalizará el ensayo y el led indicador se apagará.

Cada vez que se comience un ensayo se creará un nuevo archivo en la tarjeta de memoria donde se guardarán las mediciones. El nombre del archivo sera la fecha y la hora del comienzo del ensayo.

VII. ENSAYOS DE PRUEBA

Se realizaron una serie de ensayos de pruebas a temperatura ambiente para verificar el funcionamiento de todo el sistema. Dichos resultados se observan en la tabla I.

TABLE I
RESULTADOS OBTENIDOS

Fecha	Hora	Temp_1	Temp_2	Temp_3	Temp_4
15/08/2024	17:59:24	19.50	18.75	19.25	19.00
15/08/2024	17:59:44	20.00	18.50	19.00	19.50
15/08/2024	18:00:04	19.75	18.50	19.00	19.00
15/08/2024	18:00:24	19.75	18.75	19.25	19.25
15/08/2024	18:00:44	20.00	18.75	19.50	19.25
15/08/2024	18:01:04	20.25	18.50	19.50	19.25
15/08/2024	18:01:24	20.00	18.50	19.25	19.00
15/08/2024	18:01:44	20.00	18.25	19.50	19.25

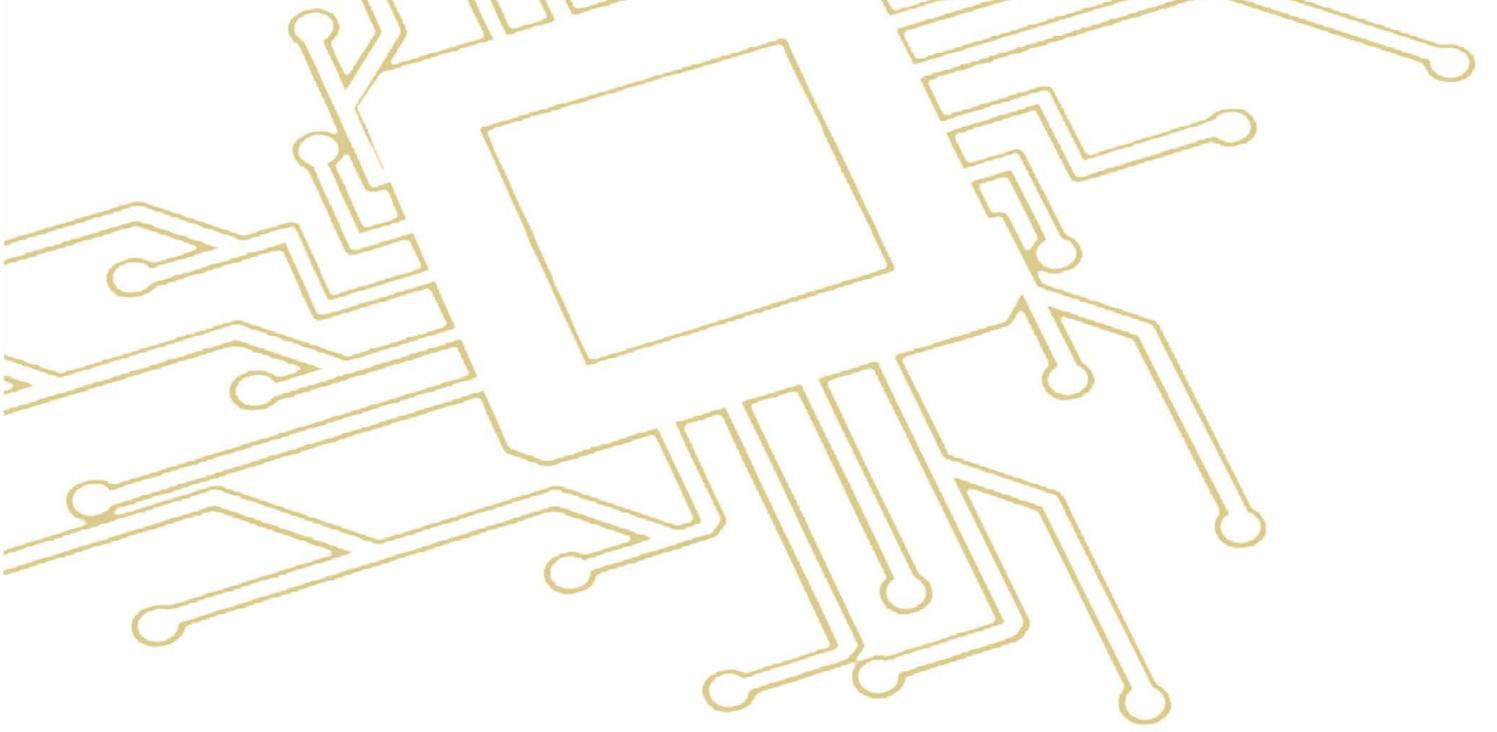
Fecha	Hora	Temp_5	Temp_6	Temp_7	Temp_8
15/08/2024	17:59:24	19.00	20.00	18.75	19.50
15/08/2024	17:59:44	19.25	20.00	18.50	19.50
15/08/2024	18:00:04	19.25	20.25	18.75	19.75
15/08/2024	18:00:24	19.00	20.00	19.25	19.25
15/08/2024	18:00:44	19.00	20.25	19.00	19.75
15/08/2024	18:01:04	19.25	20.00	19.00	19.50
15/08/2024	18:01:24	19.25	20.00	18.75	19.75
15/08/2024	18:01:44	19.25	20.25	18.75	19.75

VIII. CONCLUSIONES

En el presente trabajo se diseñó y construyó un sistema de medición y registro de temperaturas mediante termocuplas, de forma desacoplada vía transmisión WiFi. Este dispositivo resolvió una problemática puntual a la hora de estudiar el comportamiento y distribución de temperaturas en estufas tipo *rocket*, manejando una cantidad de sensores considerables.

REFERENCES

- [1] R. N. Tutorials. (2024) K-type thermocouple with max6675 amplifier. [Online]. Available: <https://randomnerdtutorials.com/arduino-k-type-thermocouple-max6675/>
- [2] L. Llamas. (2024) Technical reference. [Online]. Available: <https://www.luisllamas.es/esp8266-nodemcu/>
- [3] A. Devices. (2024) Data sheet. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/DS3231.pdf>
- [4] R. N. Tutorials. (2024) Guide to sd card module with arduino. [Online]. Available: <https://randomnerdtutorials.com/guide-to-sd-card-module-with-arduino/>
- [5] Nexperia. (2024) 74hc4051. [Online]. Available: <https://www.nexperia.com/products/analog-logic-ics/analog-switches-and-multiplexers/analog-switches/series/74HC4051-74HCT4051.html>



Eje Temático

Aplicaciones en Ecología





Monitorización de variables mediante microcontrolador PIC para el análisis de la eficiencia energética en un prototipo de edificio inteligente

Cesar Alejandro Beltran Ospina

Universidad Pedagógica y
Tecnológica de Colombia

Facultad de Ingeniería

Tunja, Colombia

cesar.beltran@upte.edu.co

Angie Viviana Sandoval Castillo

Universidad Pedagógica y
Tecnológica de Colombia

Facultad de Ingeniería

Tunja, Colombia

Sandoval.angie@uptc.edu.co

Ilber Adonayt Ruge Ruge

Universidad Pedagógica y
Tecnológica de Colombia

Facultad de Ingeniería

Tunja, Colombia

Ilber.ruge@uptc.edu.co

Resumen— Este artículo presenta un prototipo basado en el microcontrolador PIC16F877A, integrado por un sistema de seguridad, lectura de sensores y visualización de datos en una pantalla LCD. Utiliza un sensor de temperatura y humedad DHT11, un reloj en tiempo real DS3231, y sensores de medición de voltaje y corriente. Incluye la interfaz gráfica EcoMetrics v1.0 para supervisar, almacenar y analizar datos. El sistema calcula y comunica valores de potencia, mostrando un rendimiento fiable con errores de 2% a 6% en mediciones comparativas. Esta flexibilidad permite futuras mejoras, haciéndolo adecuado para el desarrollo y aprendizaje en electrónica y programación de microcontroladores.

Palabras clave: Microcontrolador PIC, gestión energética, ahorro de energía.

I. INTRODUCCIÓN

Los edificios a nivel mundial representan aproximadamente el 30% del consumo total de energía y son responsables de una parte significativa de las emisiones de carbono, lo que plantea desafíos críticos en términos de sostenibilidad y eficiencia energética global [1]. Los sistemas de calefacción, ventilación, aire acondicionado y los sistemas de iluminación son los principales consumidores de energía en edificios residenciales, comerciales y oficinas, llegando a consumir hasta el 60% de la energía total utilizada [2]. Según el estudio mostrado por Neirotti [3], otro de los factores de mayor consumo energético en los edificios es la hora

del día; los edificios residenciales tienen un consumo menor durante el día, mientras que los edificios comerciales y administrativos tienen un consumo alto durante el día y menor en la noche. El consumo de energía en edificios inteligentes se puede reducir mediante la gestión adecuada de recursos energéticos distribuidos y la programación óptima de tareas domésticas, ya que estas pueden representar un ahorro del 10% hasta el 30% en el consumo de energía [4].

La integración de dispositivos programables como los microcontroladores para el desarrollo de prototipos tiene un papel importante para hacer seguimiento, control y medición eficiente del consumo de energía. Algunos estudios han propuesto sistemas microcontrolados para la gestión de la energía, como Biswas S. [5] quien describe una red inteligente basada en un microcontrolador PIC18F452 para optimizar el uso de la energía e incorporar fuentes de energía renovables como un panel solar. Islam M. [6] desarrolla un sistema de control de iluminación para ahorrar energía que reduce el uso analizando la intensidad de la luz en distintas zonas de una habitación, mostrando resultados positivos en la reducción de consumo de la energía de red hasta en un 44%. También, Roslina R. [7] desarrolla un sistema inteligente que utiliza sensores y microcontroladores para ahorrar energía eléctrica en función de la presencia humana en un campus público, donde obtuvieron resultados de un promedio de energía ahorrada

de 12.51kWh y un porcentaje de energía ahorrada con respecto a lo normal de 11.81%. Otros estudios implementan estos sistemas prototipados usando placas de desarrollo Arduino y Raspberry, adicionando sistemas de comunicación GSM. Por ejemplo, Al-sehail D. [8] utiliza tecnología GSM inalámbrica (Global System for Mobile Comunication) con servidores públicos para evitar perdida de electricidad y mejora en la red eléctrica. Chekired F. [9] desarrolla un prototipo con Arduino para gestión energética domestica de bajo costo que contiene controladores de carga PWM (Pulse Width Modulation) y sensores, demostrando una eficiencia en la optimización y el ahorro de energía. Otro caso es el expuesto por Choudhery R. [10], quienes desarrollan un sistema de control de ahorro energético de lámparas con sensores IR y Arduino, para ajustar la intensidad del alumbrado en función de la presencia de vehículos, obteniendo una reducción en el consumo de energía y una ayuda en la mejora de la vida útil de las lámparas.

Este artículo plantea el desarrollo de un prototipo de casa de un piso con cuatro compartimentos, los cuales disponen cada uno de una bombilla como elemento consumidor de energía. El prototipo dispone de sensores de corriente, voltaje, temperatura y humedad, cuyos valores son medidos mediante conversor análogo a digital (ADC) y módulo de comunicación serial USART (Universal Synchronous Asynchronous Receiver Transmitter) de un microcontrolador PIC16F877A de la familia Microchip®, para calcular el consumo de potencia y energía. El resultado de los cálculos son visualizados en una pantalla de cristal líquido LCD 16x2, y además enviados mediante protocolo RS-232 configurado a 9600 baudios a una interface gráfica de usuario (GUI) desarrollada en lenguaje Python; la interface gráfica visualiza los datos, almacena la información en una base datos, realiza operaciones básicas de promedio en ventanas de tiempo diario, semanal y mensual para cada variable medida, y finalmente genera un reporte en archivo Excel de la información procesada.

II. MATERIALES Y MÉTODO

El prototipo a escala se construye para simular el comportamiento dinámico de consumo de energía en una casa de un piso con cuartos, pasillos, comedor y cocina. El

consumo de energía está representado por el accionamiento independiente de una bombilla de 12 Vdc dispuesta en cada compartimento, las cuales van a simular el consumo de energía cada vez que alguna de ellas sea encendida por usuario: Por ejemplo, la bombilla de la cocina podría representar el funcionamiento de un electrodoméstico como la nevera, y la bombilla dispuesta en los cuartos podría representar la activación de un sistema de aire acondicionado, ventilador o iluminación. Se dispone de sensores para la medición de corriente y voltaje, cuyos valores permiten el cálculo de potencia y energía en el prototipo, usando las ecuaciones (1) y (2).

$$v * i = P. \quad (1)$$

$$P * t = E. \quad (2)$$

Una temperatura ambiente y una ventilación correcta son fundamentales para la salud y el confort. Por tanto, se dispone de un sensor de temperatura y humedad para considerar estos parámetros en los análisis de consumo energético. La Tabla 1 muestra una descripción de los sensores instalados en el prototipo.

Como dispositivo procesador se dispone de un microcontrolador PIC16F877A de la familia Microchip®. Es un procesador digital de 8 bits, 14kB de memoria Flash de programa, 256 bytes de memoria de datos EEPROM, velocidad de operación hasta 5MIPS, periférico digital de comunicación USART y conversor ADC de 10 bits.

Tabla 1
Descripción de sensores implementados en prototipo.

Referencia	Variable(s)	Descripción
FZ0430	Voltaje	Sensor de voltaje AC y DC, Proporciona una salida de voltaje analógica, Consiste de un divisor de tensión conformado por dos resistores de 7.5KΩ y 30KΩ, Tiene un voltaje de alimentación de 3.3V a 5V
ACS712	Corriente	Sensor de corriente basado en efecto Hall. Mide corrientes AC y DC. Voltaje de alimentación de 5V y sensibilidad de 66mV/A.
DHT11	Temperatura y Humedad	Sensor digital con interface Single-bus (bidireccional). Conformado por dos sensores resistivos (NTC y humedad), Precisión de ±2°C en temperatura y precisión de ±5% en humedad relativa. Voltaje de alimentación de 3.3V a 5V.

Para la visualización de los datos medidos se tiene una pantalla de cristal líquido LCD 16x2. Un teclado matricial 4x4 permite al

usuario ingresar clave de acceso y variación de parámetros en el sistema (cambio de clave). El prototipo también contiene un módulo de calendario digital (RTC DS3231), que permite etiquetar los datos con fecha y hora de registro, y conversor micro USB a serial TTL de referencia CP2102, para realizar la comunicación serial con la interfaz gráfica de usuario. La Fig.1 presenta el diagrama de bloques del sistema.

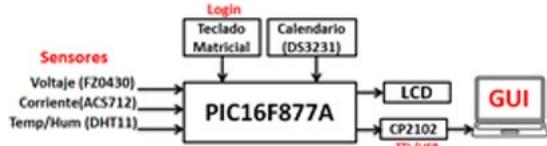


Fig. 1, Diagrama de bloques del prototipo.

El software de programación usado es CCS Compiler for PIC (PIC C Compiler v5.0) basado en lenguaje de alto nivel C/C++. Las funcionalidades programadas en el microcontrolador son:

- Control de acceso: El usuario debe disponer de una clave para ingresar al sistema. Debe ser digitada a través del teclado matricial.
- Configuración: Opción para cambiar clave, una vez se tenga acceso autorizado al sistema.
- Registro de datos: Calculo de Potencia (W) y Energía (Wm) en función de la medición de voltaje y corriente. Medición de temperatura y humedad.
- Envío de datos: Envío de valores de variables medidas a interface gráfica de usuario.

Las magnitudes de Potencia (P) y Energía (E) calculados son Vatios (W) y Vatios/hora (Wh), considerando los valores pequeños de las variables de voltaje y corriente (12Vdc y 0.4A por bombilla). El diagrama de flujo mostrado en la Fig. 2 detalla el algoritmo desarrollado en el microcontrolador PIC.

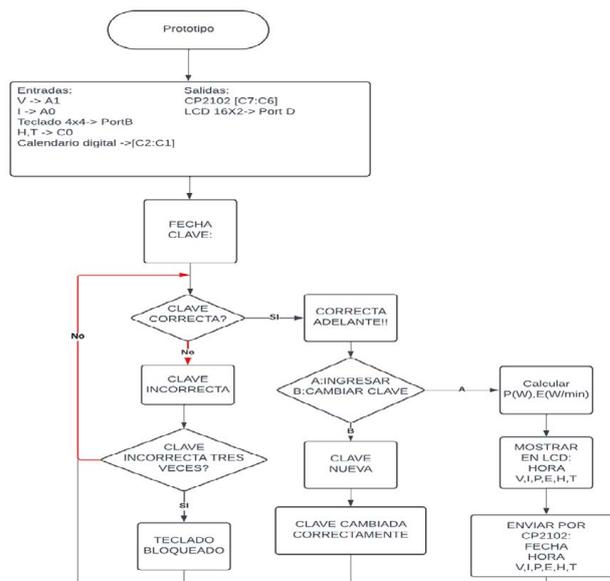


Fig. 2, Algoritmo desarrollado en microcontrolador PIC.

La Fig. 3 muestra el aspecto físico del sistema electrónico desarrollado para la medición de las variables para el cálculo de potencia, energía, temperatura y humedad.

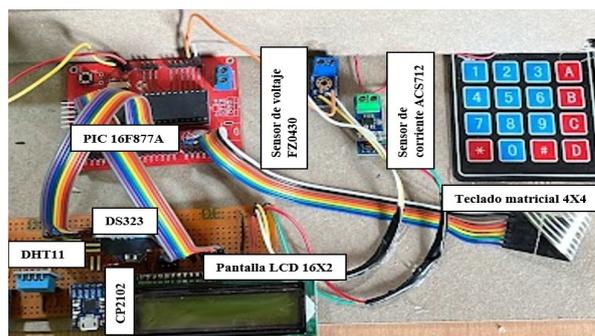


Fig. 3, Prototipo para medición de consumo de energía.

Una interface gráfica de usuario en lenguaje Python ha sido desarrollada como herramienta para la monitorización y análisis de datos del consumo de energía en el prototipo y almacenamiento en una base de datos. La comunicación del prototipo con la interface gráfica se realiza mediante un módulo de conversión TTL a USB de referencia CP2102, que corresponde a una comunicación serial con protocolo RS232 configurado a una velocidad de transmisión de 9600 baudios. La interface gráfica permite hacer seguimiento en tiempo real de las variables medidas en el prototipo con 3 horas de periodo de muestreo. Además, calcula promedios para rangos de supervisión diario, semanal y mensual. Se dispone de la posibilidad de almacenar estos datos en un

archivo Excel, los cuales pueden ser usados para interpretaciones y análisis.

III. ANÁLISIS DE RESULTADOS

Para validar el sistema electrónico de medición, se simula en ISIS Proteus 8.9. Se verifican los valores medidos de voltaje, corriente, temperatura y humedad según los parámetros de los sensores. Se utiliza el monitor serial de ISIS Proteus y el Hércules Setup Utility para la comunicación. La Figura 4 ilustra el circuito electrónico y el prototipo físico.

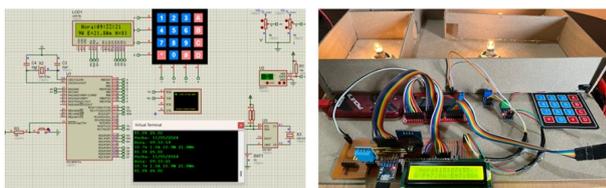


Fig. 4, Simulación de sistema de medición y prototipo.

Se registran algunos datos proporcionados por el prototipo en diferentes situaciones de consumo de energía. Estas situaciones corresponden al encender las bombillas que representan el consumo de energía de forma secuencial hasta tener todas las bombillas encendidas. La Tabla 2 muestra los resultados obtenidos y mostrados tanto en la LCD como en la interface gráfica de usuario para un periodo de prueba de 10 minutos para cada situación.

Tabla II.
Medición de datos en prototipo y registro en GUI

Situación	Voltaje (V)	Corriente (A)	Potencia (W)	Energía (Wm)
Cuarto 1	12.3	0.3	4.0	40
+ Cuarto 2	11.5	0.5	5.7	57
+ Cocina	11.0	0.9	9.8	98
+ Comedor	10.6	1	10.7	107

Ahora, en la Tabla 3 se muestra los resultados de corriente y voltaje medidos con multímetro digital Fluke 115, bajo las mismas condiciones y dinámica que el caso anterior.

Tabla 3.
Medición de datos en prototipo con instrumento patrón.

Situación	Voltaje (V)	Corriente (A)	Potencia (W)
Cuarto 1	11.75	0.29	3.40
+ Cuarto 2	11.19	0.49	5.48
+ Cocina	10.86	0.88	9.33
+ Comedor	10.57	0.98	10.36

La Tabla 4 muestra el cálculo de los errores porcentuales de cada medida, como estimativo

del grado de confiabilidad del sistema electrónico de medición desarrollado. Los resultados muestran un error promedio total de 5.04% en el cálculo de Potencia, lo cual es un valor aceptable para los propósitos de estimación de consumo de energía.

Tabla 4.
Cálculo de error porcentual.

Situación	Error % voltaje	Error % corriente	Error % Potencia
Cuarto 1	2.12%	3.44%	5.8%
+ Cuarto 2	1.87%	2.04%	4.01%
+ Cocina	2.20%	2.27%	6.1%
+ Comedor	2.17%	2.04%	4.24%
Promedio	2.10%	2.45%	5.04%

Las variaciones en los datos medidos en el prototipo pueden obedecer a factores como la precisión de los sensores, la fuente de alimentación del circuito eléctrico o variaciones durante la conversión análoga a digital de los datos.

Para un análisis más detallado del comportamiento de las variables medidas, la interface gráfica de usuario dispone de la opción "Datos del Prototipo", que visualiza en tiempo real con periodo de muestreo igual a 5s. La Fig. 5 muestra un ejemplo del detalle de la medición realizada.

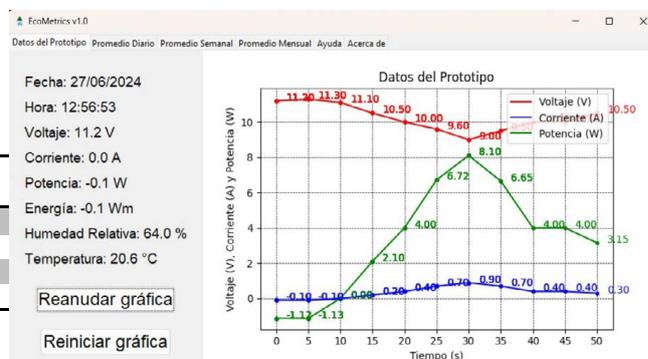


Fig. 5, Registro de medición en tiempo real mediante GUI.

La interfaz de usuario puede configurarse para hacer registro de estos datos con periodo de muestreo igual a 3 horas, y los valores se almacenan en una base de datos. Esta funcionalidad de almacenamiento permite al usuario visualizar la dinámica en ventanas de tiempo "Diario", "Semanal" o "Mensual". La Fig. 6 muestra un ejemplo del resultado de visualización correspondiente al comportamiento diario de la Energía. El día es seleccionado por usuario mediante Calendario.

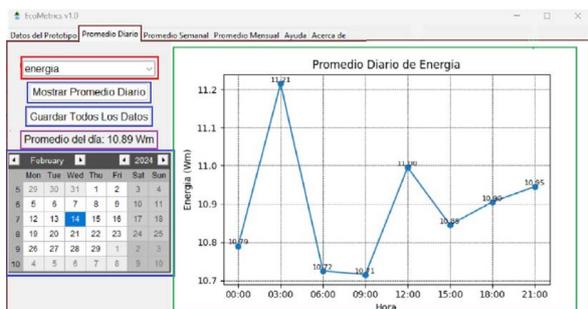


Fig. 6, Análisis de consumo energético opción “Diario” mediante GUI.

De igual manera, la Fig. 7 muestra un ejemplo del resultado de visualización correspondiente al comportamiento semanal de la Energía. La semana de interés es seleccionada por usuario mediante Calendario. Así mismo, la interface gráfica de usuario dispone de la opción de análisis “Mensual”. La Fig. 8 muestra un ejemplo de este tipo de resultado.

La flexibilidad del código Python sobre la cual se desarrolló la Interface Gráfica de Usuario, permite adaptaciones y mejoras, como agregar nuevos sensores o mejorar la interfaz de usuario, lo que lo hace adecuado para el desarrollo y evolución del prototipo, así como para proyectos de desarrollo y aprendizaje en electrónica y programación de microcontroladores.

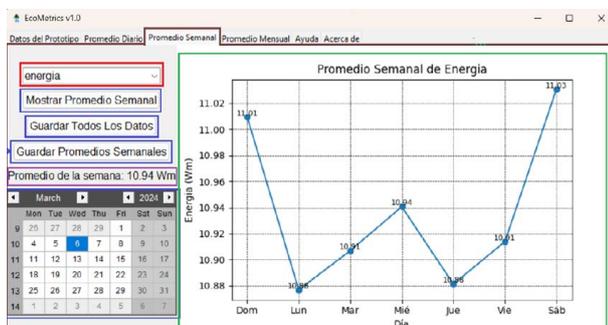


Fig. 7, Análisis de consumo energético opción “Semanal” mediante GUI.

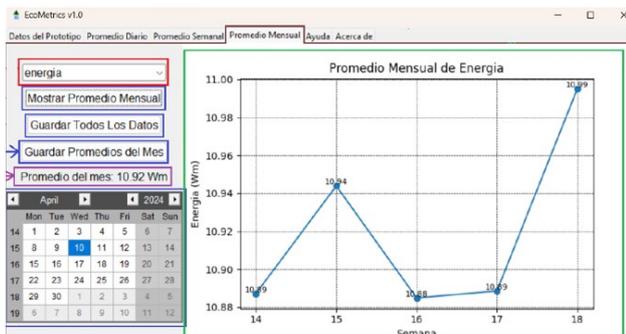


Fig. 8, Análisis de consumo energético opción “Mensual” mediante GUI.

IV. CONCLUSIONES

La recolección y análisis de datos son efectivos, mostrando cómo varían la corriente, voltaje y potencia en diferentes situaciones de consumo energético. La interfaz gráfica permite un monitoreo continuo y preciso, facilitando la configuración y visualización de datos del prototipo, así como su almacenamiento en una base de datos, lo que permite análisis profundos y a largo plazo.

Los resultados obtenidos del prototipo físico y la interfaz gráfica son consistentes, validando la precisión y fiabilidad del sistema. La capacidad de registrar datos en intervalos de tiempo (diarios, semanales, mensuales) es útil para el análisis energético. Comparando los valores del prototipo y la maqueta, los errores porcentuales son inferiores al 3.44% para voltaje y corriente, y entre 4.01% y 6.1% para potencia, lo que confirma la robustez del sistema para aplicaciones energéticas.

REFERENCIAS

- [1] L. Yu, S. Qin, M. Zhang, C. Shen, T. Jiang, and X. Guan, “A Review of Deep Reinforcement Learning for Smart Building Energy Management,” 2020, doi: 10.1109/JIOT.2021.3078462.
- [2] N. M. Abdelhamid, B. Lejdel, and E. Clementini, “Optimizing of Consumption Energy in Smart building,” WSEAS Trans. Syst. Control, vol. 17, pp. 428–438, 2022, doi: 10.37394/23203.2022.17.47.
- [3] P. Neirotti, A. De Marco, A. C. Cagliano, G. Mangano, and F. Scorrano, “Current trends in smart city initiatives: Some stylised facts,” Cities, vol. 38, no. September 2018, pp. 25–36, 2014, doi: 10.1016/j.cities.2013.12.010.
- [4] D. Zhang, N. Shah, and L. G. Papageorgiou, “Efficient energy consumption and operation management in a smart building with microgrid,” Energy Convers. Manag., vol. 74, pp. 209–222, 2013, doi: 10.1016/j.enconman.2013.04.038.
- [5] S. Biswas, “Development of Microcontroller Based Smart Grid Framework,” Int. J. Sci. Manag. Stud., no. August, pp. 315–324, 2021, doi: 10.51386/25815946/ijms-v4i4p129.
- [6] M. S. Islam et al., “An Energy-Efficient Microcontroller-Based Smart Light Controlling System,” J. Electr. Electron. Eng., vol. 14, no. 2, pp. 25–30, 2021, doi: 10.20944/preprints202107.0251.v1.
- [7] R. Roslina, A. Amelia, H. Pranoto, and B. V. Sundawa, “System of smart detection and control to electrical energy for saving of electrical energy consumption,” Bull. Electr. Eng. Informatics, vol. 10, no. 5, pp. 2454–2465, 2021, doi: 10.11591/eei.v10i5.2826.
- [8] D. Al-sehail, O. Al-Atbee, and A. Marhoon, “Smart Energy Metering Based on Arduino, GSM, and Raspberry pi3 as Server,” 2022, doi: 10.4108/eai.7-9-2021.2314803.
- [9] F. Chekired, O. Taabli, Z. M. Khellili, A. Tilmatine, A. T. de Almeida, and L. Canale, “Near-Zero-Energy Building Management Based on Arduino Microcontroller—On-Site Lighting Management Application,” Energies, vol. 15, no. 23, 2022, doi: 10.3390/en15239064.
- [10] R. B. Choudhery and A. M. Nikam, “Arduino based smart street light system with accident avoidance in U-turns,” Int. J. Adv. Res., vol. 5, no. 9, pp. 214–218, 2019, [Online]. Available: www.ijsr.net2015.

Análisis de la Linealidad del ADC del ESP32 en la Medición de Radiación UV

Mario Blas Lavorato*, Nicolás Urbano Pintos[†], Nehuen Oscár Pocovi[‡]

Grupo de Tecnología Aplicada al Medio Ambiente

Facultad Regional Haedo- Universidad Tecnológica Nacional

Haedo, Argentina

Correo electrónico: *mlavorato@frh.utn.edu.ar, [†]nurbano@frh.utn.edu.ar, [‡]npocovi128@alumnos.frh.utn.edu.ar

Resumen—Este trabajo aborda la corrección de la no linealidad en la conversión analógica-digital (ADC) del microcontrolador ESP32, aplicado a la medición de radiación UV. Se implementó un polinomio de cuarto grado para corregir las distorsiones del ADC, logrando una mejora significativa en los niveles altos de señal, con un error reducido al 3%. Sin embargo, persisten problemas en los niveles bajos de tensión, donde la distorsión sigue siendo considerable, afectando la precisión de las mediciones. A pesar de estas limitaciones, se demostró que el ESP32 es adecuado para aplicaciones donde el rango operativo se encuentra en niveles más altos. Se discuten alternativas para mejorar la precisión, como el uso de un ADC externo o versiones avanzadas del ESP32. Los resultados subrayan la importancia de un análisis detallado en aplicaciones críticas para asegurar la fiabilidad en diversas condiciones operativas.

Palabras Claves—Linealidad, ADC, ESP32, Radiación UV.

I. Introducción

El Sol es la principal fuente de radiación que llega a la Tierra, jugando un papel crucial en los procesos climáticos. Dentro del espectro solar, la radiación ultravioleta (UV) destaca por sus significativos efectos sobre la vida en nuestro planeta. Aunque la radiación UV representa apenas el 5-6% de la radiación solar total, su impacto es notable. Se divide en tres bandas: UVC (180-280 nm), que es letal pero es completamente absorbida por la atmósfera; UVB (280-320 nm), capaz de dañar el ADN y causar cáncer; y UVA (320-400 nm), menos dañina y fundamental para la vida [1].

La necesidad de monitorear en tiempo real los índices de radiación UV es evidente, especialmente considerando los altos costos de los radiómetros comerciales. En respuesta, se ha desarrollado un radiómetro UV de bajo costo utilizando fotodetectores específicos. Tras un proceso exhaustivo de selección, se eligieron dos sensores disponibles en el mercado local: el ML8511 [2], que mide la radiación UV total (UVB + UVA), y el UVM-30A [3], que mide el Índice UV (IUV) y es apto para evaluar la peligrosidad de la radiación UV. Estos dispositivos permiten una monitorización accesible y precisa, esencial para la protección y el estudio del clima y la salud humana.

Para implementar este monitoreo en tiempo real, se requiere un sistema embebido capaz de adquirir y registrar los datos de radiación, y además, facilitar el acceso a esta información para el público general a través de un sitio web. El ESP32 (NODEMCU32 ESP-WROOM-32s) [4], también denominado ESP32-S1 es un dispositivo de bajo costo, se presenta como una solución ideal debido a su capacidad para realizar procesamiento in situ, gracias a su microcontrolador de 32 bits, y su conectividad tanto Wi-Fi como Bluetooth.

Sin embargo, el ESP32 S1 disponible en Argentina presenta desafíos significativos con su conversor analógico-digital (ADC - Analog Digital Converter), que muestra una no linealidad considerable en la conversión, además de dificultades para discriminar valores cercanos a 0 V. Aunque este problema es conocido en la comunidad de usuarios del ESP32 y se han propuesto soluciones [5] [6] [7], la literatura carece de material claro que describa detalladamente el comportamiento del ADC.

Este trabajo presenta una experimentación detallada para evaluar la linealidad del ADC del ESP32, y evalúa una corrección basada en un polinomio de cuarto grado. Finalmente, se analizan señales de radiación UV registradas por el ADC, aplicando la corrección matemática desarrollada y se discute sobre posibles soluciones. Este estudio contribuye a la caracterización del ADC del ESP32 y explora diversas opciones para mitigar sus limitaciones.

II. Antecedentes

Cit et al. [5] emplearon un conversor digital-analógico (DAC) externo de 12 bits de resolución (MCP4725) para generar una rampa lineal de 0 a 3.3 V, la cual fue registrada por el ADC del ESP32. Para corregir la no linealidad, utilizaron una red neuronal artificial de tipo perceptrón multicapa, entrenada con los valores medidos y los valores generados por el DAC. Aunque lograron una transferencia lineal a partir de 0.1 V, no pudieron obtener datos precisos por debajo de este umbral.

Anshori et al. [6] implementaron una calibración utilizando el DAC integrado del ESP32, generando una tabla

de consulta para corregir las mediciones. Sin embargo, tampoco pudieron registrar valores inferiores a 0.1 V.

Maier et al. [7] aplicaron una rampa de 0 a 3.5 V en la entrada del ADC del ESP32, con incrementos de 0.1V. Evaluaron la medición bajo diferentes configuraciones de atenuación del ADC, obteniendo los mejores resultados con -6 dB, pero sin poder medir valores por debajo de 0.17 V. Como solución, ajustaron la señal a un rango de 0.2 V a 1.9 V.

Cheheung [8], basándose en el trabajo de Helmut Weber [9], desarrolló un repositorio que genera una rampa con el DAC del ESP32 y crea una tabla de consulta para corregir la linealidad. Este trabajo destaca que cada dispositivo ESP32 posee una tensión de referencia única, lo que requiere una tabla específica para cada unidad.

Finalmente, Bird [10] creó un repositorio que utiliza una función polinómica de cuarto grado para corregir la linealidad del ADC del ESP32, logrando una precisión cercana al 1%.

III. Experimentación

A. Caracterización de la transferencia del ADC

La primera etapa del experimento fue para conocer a grandes rasgos los niveles de distorsión y/o linealidad del ADC del ESP32, para ello se realizó una experiencia ingresando valores de tensión con una Fuente de Alimentación variable y medida con un multímetro de 4 dígitos y medio. Durante la experiencia, para la adquisición se utilizó la instrucción *analogRead* que da como resultado cuentas del ADC que van de 0 (0 V) a 4095 (3.3 V); estos valores, luego, son convertidos en valores de tensión. Luego se repitió la misma experiencia utilizando la instrucción *analogReadMilliVolts* que entrega directamente valores de tensión (en mV) equivalentes a la instrucción anterior. En ambos casos y debido a que el ADC entregaba valores con alta dispersión, se optó por agregar en el programa una instrucción para promediar 128 valores instantáneos adquiridos en forma consecutiva. En la Tabla I se muestran los resultados obtenidos, donde se observa la dispersión entre los valores reales y los medidos por el ADC.

Multímetro	Calculado	Lectura ADC	Valor ADC
3.285 V	4095	4095	3.300 V
2.513 V	3132	2925	2.346 V
1.520 V	1894	1759	1.411 V

TABLA I
Dispersión de Valores del ADC del ESP32

En la segunda etapa del experimento y para conocer los niveles de linealidad del ADC del ESP32 se optó por generar una rampa lineal con una salida de Modulación por ancho de pulsos (PWM- Pulse-width modulation) (D5) filtrada de un Arduino NANO mientras que los valores eran registrados en una tarjeta SD (Figura 1 y 2). Con

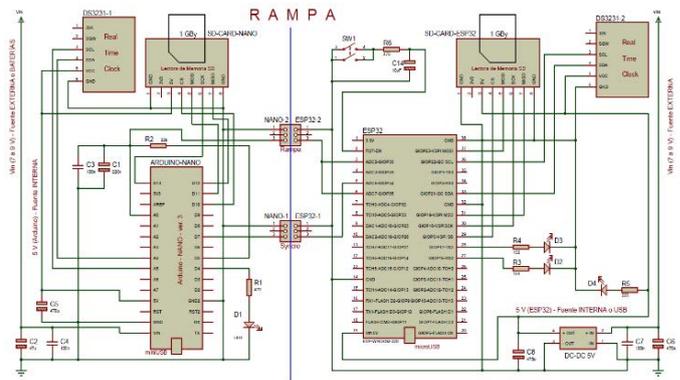


Fig. 1. Esquema circuital del sistema Rampa-ADC (NANO – ESP32)

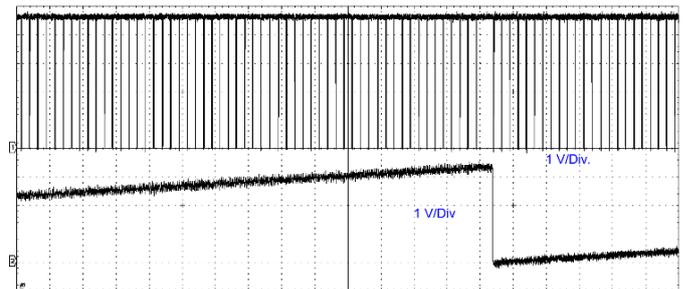


Fig. 2. Rampa y señal de sincronismo de Arduino Nano

esa señal, como se ve en el esquemático de la Figura 1 se ingresa el ESP32 a dos de sus entradas analógicas para verificar que la no linealidad depende del ADC y no de la entrada analógica seleccionada; por ello se utilizan los pines: PIN 3 ADC0 GPIO 36 y el PIN 6 ADC7 GPIO 35, finalmente se registra el resultado en un la memoria SD. Ambos dispositivos cuentan con Relojes (Real Time Clock – DS3231). Además de la rampa, el Arduino NANO genera una señal de sincronismo (PIN D7 – lógica negativa que se observa en la Figura 2) para indicarle al ESP32 que haga la adquisición (PIN 5 ADC6 GIOP34). Los niveles de la rampa fueron incrementados cada 5 segundos iniciando en 0 (0 V) hasta 2678 (~ 3.2V) mientras que la señal de sincronismo se emitía a los 2 segundos del incremento, esto se hizo para evitar distorsiones y efectos no deseados provocados por los flancos de la rampa durante la adquisición del ESP32.

En la Figura 3 se muestran dos curvas, una representada como una línea recta corresponde a la Rampa generada por el Arduino NANO y que se ingresa a los canales analógicos ADC0 y ADC7 del ESP32; las otras dos curvas representan los datos grabados en la memoria SD de cada una de las entradas; se puede observar que casi no hay diferencia entre los resultados de la adquisición de ambos registros; por el contrario, se puede apreciar la enorme distorsión y falta de linealidad que tiene el conversor ADC del Microcontrolador.

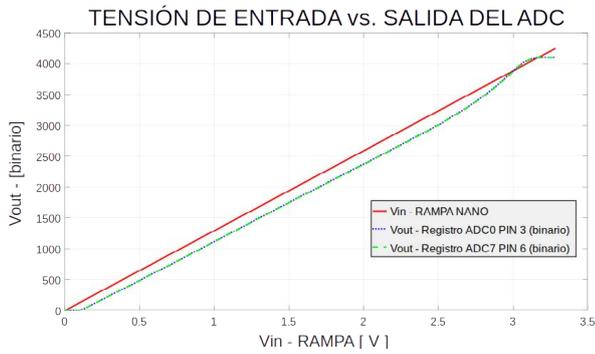


Fig. 3. Rampa adquirida por el ADC del ESP32 y registrada en la SD



Fig. 4. Sensores de UV conectados a ESP32 con batería

B. Corrección de la transferencia del ADC

Para paliar el problema de no linealidad del ADC del ESP32, y basados en el trabajo de Bird [10] se llevó adelante una corrección por software aplicando a la señal adquirida; un polinomio de grado cuatro. El mismo fue obtenido de una regresión polinómica a partir del promedio de más de 1000 valores medidos de la de la rampa generada por el Arduino Nano. En la Ecuación 1 se detalla el polinomio utilizando. El ajuste del polinomio se hizo a través del método de mínimos cuadrados.

$$In_{adc}^*(V) = a.In_{adc}^4 + b.In_{adc}^3 + c.In_{adc}^2 + d.In_{adc} + e \quad (1)$$

Donde In_{adc}^* es la lectura del ADC corregida por el polinomio expresada volts, In_{adc} la lectura del ESP32 expresada en cuentas, y a, b, c, d y e los coeficientes del polinomio de grado cuatro. En la tabla II se detallan los coeficientes obtenidos en la regresión polinómica.

Coefficiente	Valor
a	-1.07976×10^{-14}
b	7.09764×10^{-11}
c	-1.63250×10^{-7}
d	9.51995×10^{-4}
e	8.88719×10^{-2}

TABLA II
Coeficientes del polinomio de cuarto grado

C. Medición de radiación UV

Para evaluar la corrección con datos de sensores UV, se conectó el sensor ML8511 y el sensor UVM-30A a un sistema portátil con un microcontrolador ESP32, con baterías y una tarjeta SD. Ambos sensores entregan señales analógicas que son adquiridas por las entradas AD del microcontrolador. El sistema fue montado en un trípode (Figura 4) y se registraron mediciones diarias de radiación UV en el mes de marzo de 2024, en la localidad de Merlo, provincia de San Luis, Argentina.

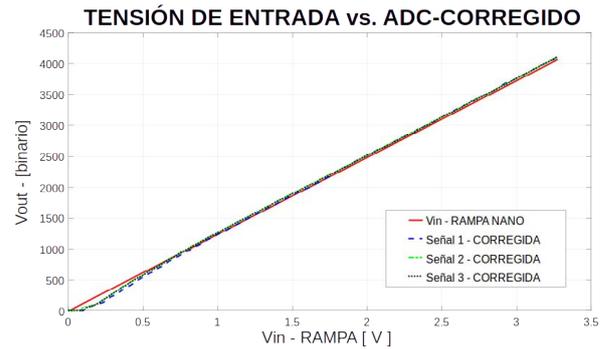


Fig. 5. Rampa y medición corregida

IV. Resultados

Se realizaron las correcciones de no linealidad del ADC del ESP32 utilizando el polinomio de cuarto grado descrito en la sección anterior. Nuevamente, se utilizó como medición de referencia a la rampa generada por el Arduino Nano. Se puede observar que este polinomio corrige muy bien los niveles altos de la señal (Figura 5) mientras que, como se puede visualizar en la Figura 6 en los niveles bajos de la señal, la conversión falla dando una apreciable distorsión.

En las Figuras 5 y 6 se observa que a partir de los $0.15V$ se comienza a realizar la conversión con una con-

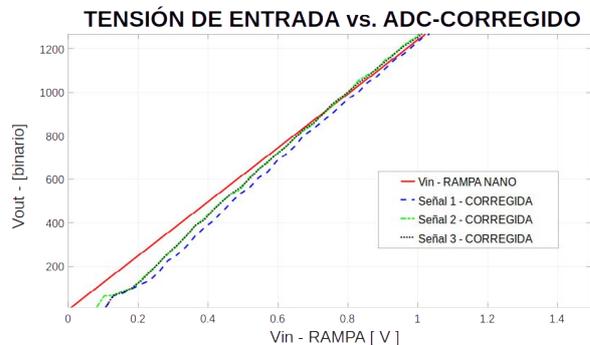


Fig. 6. Rampa y medición corregida. Ampliación zona de baja tensión

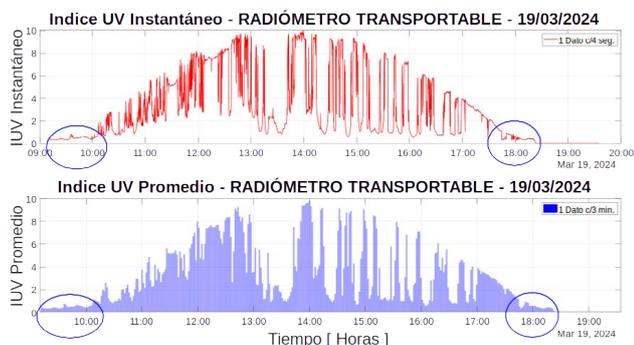


Fig. 7. Evolución temporal del IUV medido el 19/03/2024 con el cielo nublado en forma intermitente. La zona circulada corresponde a los valores con mayor distorsión y poco fiables

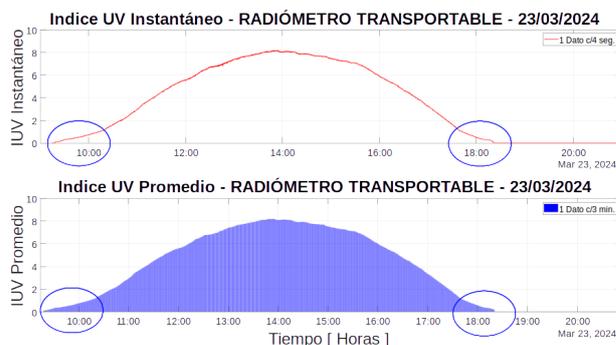


Fig. 8. Evolución temporal del IUV medido el 23/03/2024 con el cielo completamente despejado. La zona circulada corresponde a los valores con mayor distorsión y poco fiables

siderable distorsión y a partir de los $0.5V$ la conversión comienza a ser aceptable. Cabe aclarar que, para los sensores que estamos utilizando en nuestro proyecto, en particular el ML8511 no presenta dificultad alguna, ya que la salida del mismo opera entre $1V$ y los $2.6V$, justo en la zona donde el ADC y su corrección polinómica trabajan muy bien, sus resultados presentan un error aproximado del 3%. Mientras que el otro dispositivo, el UVM-30A opera entre $0V$ y $1.25V$ teniendo entre $0V$ y $0.35V$ el mayor error por distorsión y linealidad.

Como ejemplo de operación se muestran dos curvas diarias adquiridas con ambos sensores UV (ML8511 y UVM-30A) donde la mayor parte de las señales adquiridas fueron corregidas correctamente por el polinomio de 4to. Grado, y puede apreciarse muy bien la distorsión presente en los niveles bajos de tensión, en la Figura 6 se muestran 3 de más de 200 mediciones realizadas donde se observa que en la zona de bajo nivel de tensión hay una gran dispersión de señales pese a ser adquiridas en las mismas condiciones no sigue un patrón, la distorsión en los niveles bajos de la señal corregida por el polinomio resulta ser muy variable. En las Figuras 7 y 8 se muestran los resultados de mediciones de IUV realizadas en una zona montañosa de San Luís; ambas figuras presentan la evolución temporal diaria del IUV en forma de adquisición instantánea y promediada. La Figura 7 corresponde a un día nublado con alternancias de cielo despejado mientras que en la Figura 8 el cielo se mantuvo completamente despejado durante todo el día.

En ambas figuras se encuentran marcadas con círculos las zonas donde el ADC del ESP32 tiene mayor distorsión y error, eso significa que los valores obtenidos durante la adquisición no son fiables. En nuestro caso corresponde a los valores de IUV que van entre 0 y 2. Cabe aclarar que esos son valores muy bajos y que no resultan críticos ante la exposición de la piel ni los ojos; entonces, aunque el ADC opere con un alto error de conversión ($\sim 30\%$) los valores obtenidos del IUV no tienen mayor efecto.

V. Discusión

Se presentan a continuación diversas soluciones y modificaciones propuestas para abordar el problema de conversión en la zona de baja tensión del ADC integrado en el ESP32, un microcontrolador con amplias capacidades operativas. A pesar de su potencia, el defecto conocido en la conversión ADC puede ser mitigado mediante los siguientes enfoques:

- Acondicionamiento de señales de sensores: Una posible solución, es adecuar las señales de los sensores a la zona lineal del ADC. En este caso, el sensor ML8511 no presenta inconveniente, ya que funciona directamente en ese rango. Pero en el caso del sensor UVM-30A, sería necesario modificar el rango, y para ello es necesario de circuitería adicional, por ejemplo un sumador y una tensión de referencia, aunque esto trae aparejado un mayor costo, y mayor consumo energético.
- Corrección Polinómica: En nuestro caso particular, el sensor UVM-30A puede operar adecuadamente utilizando una corrección polinómica. Aunque el error y la distorsión son más pronunciados en niveles bajos de tensión, estos corresponden a valores del Índice UV menores a 2, donde las medidas de precaución no son estrictamente necesarias. Dado que la exposición en estos niveles es segura durante periodos prolongados, la precisión en esta zona resulta menos crítica.
- Uso de un Conversor ADC Externo: Otra solución es incorporar un conversor analógico-digital externo, como el ADS1015 de 12 bits con interfaz I2C [11]. Aunque implica un costo adicional y la complejidad de integrar un nuevo componente, esta opción mejora significativamente la calidad y precisión de la conversión AD, ofreciendo una alternativa robusta para aplicaciones que requieren alta exactitud.
- Migración a ESP32-S2 o ESP32-S3: Se podría considerar la sustitución del ESP32 por la versión ESP32-S2 o S3, ya que el ESP32-S3 incluye un



circuito de pre-calibración del ADC, lo que corrige el problema de no linealidad. Con este dispositivo, la respuesta a una rampa de voltaje es una rampa lineal desde 0 V hasta un máximo de 3.2 V. No obstante, una limitación actual es la disponibilidad en el mercado local de estos modelos en placas de desarrollo, lo cual puede retrasar su implementación.

VI. Conclusiones

Los resultados obtenidos demuestran que el uso de un polinomio de cuarto grado para corregir la no linealidad del ADC del ESP32 mejora significativamente la precisión en los niveles altos de señal, reduciendo el error a aproximadamente un 3% en estos rangos. Sin embargo, en los niveles bajos de tensión, la conversión sigue siendo problemática, con una distorsión considerable que afecta la fiabilidad de las mediciones, especialmente por debajo de los 0.15V.

Para aplicaciones específicas, como la medición de la radiación UV utilizando sensores como el ML8511 y el UVM-30A, estos errores en los niveles bajos de señal no resultan críticos, ya que los niveles de Índice UV (IUV) asociados con estas tensiones son bajos y no representan un riesgo significativo para la salud. No obstante, es esencial considerar estas limitaciones en diseños futuros donde la precisión en todo el rango de operación del ADC sea crítica.

A pesar de las limitaciones encontradas, el ESP32 sigue siendo una opción viable para aplicaciones donde el rango operativo se encuentra en niveles de tensión más altos. Para mejorar la precisión en futuras implementaciones, se recomienda evaluar alternativas como el uso de un ADC externo o considerar versiones actualizadas del ESP32, como el ESP32-S3, que incorpora una calibración interna más avanzada del ADC.

Finalmente, los resultados presentados aquí subrayan la importancia de realizar un análisis detallado de las características del ADC en aplicaciones críticas, y se prevé continuar con el desarrollo y mejora de estos sistemas para asegurar la fiabilidad en entornos operativos diversos.

Agradecimientos

Se agradece a las autoridades de la Universidad Tecnológica Nacional Facultad Regional Haedo por el apoyo económico y humano. Este trabajo fue financiado por el Proyecto de Investigación y Desarrollo PID 10164, denominado Red de medición de Índice UVE con equipos fijos y portátiles, que comenzó en el año 2024.

References

- [1] United States Environmental Protection Agency, *UV Radiation*, June 2010, accessed: 27 Apr. 2024. [Online]. Available: <https://www.epa.gov/sites/default/files/documents/uvradiation.pdf>
- [2] L. Semiconductor, *ML8511 UV Sensor with Voltage Output*, March 2013, accessed: 27 Apr. 2024. [Online]. Available: <https://datasheetspdf.com/pdf-down/ML/8/ML8511-LAPIS.pdf>
- [3] A. M. Shojaei. (2019) Interfacing uvm30a uv sensor module with arduino. Accessed: 27 Apr. 2024. [Online]. Available: <https://electropeak.com/learn/interfacing-uvm30a-uv-light-sensor-module-with-arduino/>
- [4] E. Systems, *ESP-WROOM-32s*, 2023, accessed: 27 Apr. 2024. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
- [5] G. Cit, J. Monteiro, T. Quirino, and J. Quirino, "Development and calibration of a low-cost electrical measurement instrument," *International Journal of Electrical Engineering and Computer Science*, vol. 5, pp. 129–134, 08 2023.
- [6] I. Anshori, G. F. Mufiddin, I. F. Ramadhan, E. Ariasena, S. Harimurti, H. Yunkins, and C. Kurniawan, "Design of smartphone-controlled low-cost potentiostat for cyclic voltammetry analysis based on esp32 microcontroller," *Sensing and Bio-Sensing Research*, vol. 36, p. 100490, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2214180422000198>
- [7] A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the esp32 microcontroller module for the internet of things," in *2017 Internet Technologies and Applications (ITA)*, 2017, pp. 143–148.
- [8] H. Cheung. (2020) Esp32-adc-calibrate. Accessed: 27 Apr. 2024. [Online]. Available: <https://github.com/e-tinkers/esp32-adc-calibrate>
- [9] H. Weber. (2019, March) Esp32 adc non-linear issues - how do i fix / change attenuation or width? Accessed: 27 Apr. 2024. [Online]. Available: <https://esp32.com/viewtopic.php?f=19&t=2881&start=30#p47663>
- [10] D. Bird. (2020) Esp32-adc-accuracy-improvement-function. Accessed: 27 Apr. 2024. [Online]. Available: <https://github.com/G6EJD/ESP32-ADC-Accuracy-Improvement-function>
- [11] Texas Instruments, *ADS1015*, May 2019, accessed: 27 Apr. 2024. [Online]. Available: <https://www.ti.com/product/ADS1015>



Prototipo de vehículo no tripulado de superficie para aplicaciones ambientales

Ignacio Zaradnik, Oreste Daniel Lupi, Luciano Salvatore, Diego Caccaviello

Departamento de Ingeniería Universidad Nacional de la Matanza

Buenos Aires, Argentina

izaradnik@unlam.edu.ar

El desarrollo de vehículos no tripulados presenta un gran número de áreas de aplicación. Una de estas áreas es el monitoreo de la calidad del agua de ecosistemas acuáticos, y dentro de ella la determinación del contenido de micro y nano plástico. A continuación, se presenta la implementación mecánica y electrónica del vehículo, esta última se realiza a través de una arquitectura distribuida, en donde un procesador corriendo Linux funciona como nodo central y múltiples microcontroladores se encargan de tareas concretas (manejo de motores y recepción de comandos). Esto permitió la navegación a través de control remoto, generando las bases para el uso de otras tecnologías que optimicen el desempeño del vehículo.

Palabras clave: vehículo no tripulado, microcontrolador, microprocesador, ecosistema acuático.

I. INTRODUCCIÓN

Tal como se ha mencionado en trabajos previos [1][2][3], el agua representa un recurso de vital importancia para el desarrollo socioeconómico, la energía, la producción de alimentos, los ecosistemas y la supervivencia de los seres humanos. Por lo tanto, la medición de su calidad es fundamental. Un parámetro que está tomando relevancia es el contenido de micro/nano plásticos en el agua, el cual tiene un efecto nocivo sobre los seres vivos, ya sea por ingestión o por toxicidad. En el caso puntual de la concentración de estos en ambientes acuáticos, si bien es posible su detección, tal como se detalla en [4] o en [5-7], dichos

métodos tienen la dificultad de no poder ser realizados en tiempo real y de ser muy costosas las campañas de muestreo y análisis. Un dispositivo adecuado para la toma de las muestras antes mencionado puede ser un pequeño vehículo de superficie no tripulado (USV por sus siglas en inglés), manejado a través de control remoto o parcialmente autónomo. En la actualidad existen diversos productos comerciales como los detallados en [8], así como diversos trabajos que presentan este tipo de solución [9-11], inclusive con la toma de muestras [12]. Los primeros poseen un costo elevado, mientras los segundos están pensados para aplicaciones que pueden no adecuarse a las necesidades propias y la documentación disponible no es adecuada para su reproducción. Por lo tanto, se planteó desarrollar una pequeña embarcación experimental autónoma y/o por control remoto para la medición de parámetros ambientales y la toma de muestras en ecosistemas acuáticos.

II. DESARROLLO

A. Sistema mecánico

Del estudio realizado sobre las embarcaciones se concluyó que una estructura del tipo catamarán (dos cascos paralelos de igual tamaño) resultaría la más adecuada para el proyecto, ya que esta permitiría poner una superficie entre ambos cascos que albergaría el sistema de control y de toma de muestras (teniendo este un mejor acceso al agua). Así mismo, se optó por una propulsión basada en dos motores, ya que esta ofrece una mejor movilidad que la basada en un solo motor con

timón. En función de lo detallado se elaboró un diseño de la embarcación en el software Autodesk Inventor, el cual se empleó como base para el cálculo de perfiles de aluminio para la estructura y el material para los cascos (Telgopor de alta densidad preformado). En lo que respecta al sistema de propulsión se optó por motores de corriente continua, de bajo costo, para el cual se desarrolló un sistema de reducción a través de impresión 3D. En la figura N°1 se puede ver el diseño en Autodesk mientras que en la figura N°2 el DIIT-A (nombre del prototipo) implementado.

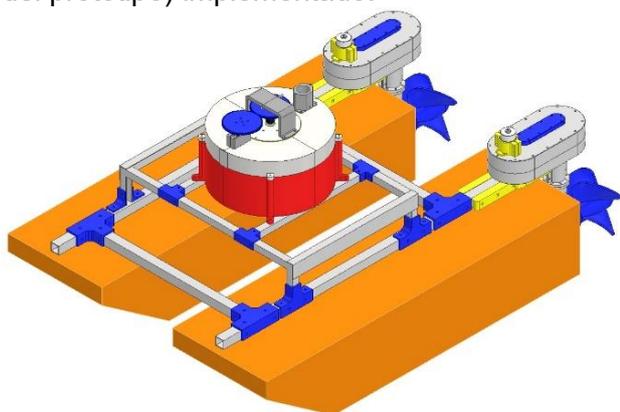


Fig. 1. Diseño en Autodesk Inventor de USV.

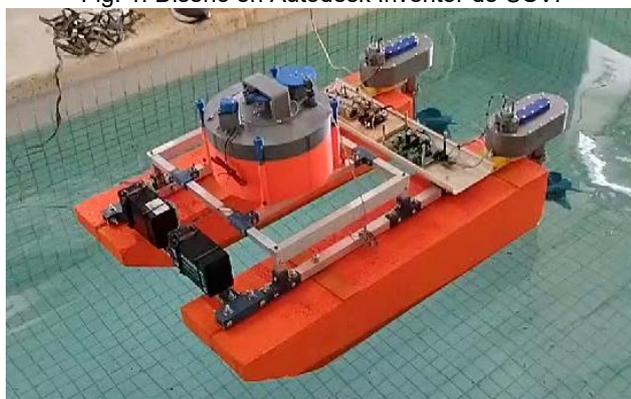


Fig. 2. Implementación del USV.

El sistema de toma de muestras, también desarrollado a través de impresión 3D, tiene la capacidad para 8 muestras, en la figura N°3 se puede apreciar el interior del sistema.

B. Arquitectura del sistema

En la figura N°4 se puede ver la arquitectura planteada para el USV, en donde se tiene un sistema central y subsistemas para las distintas tareas a desarrollar. El subsistema de alimentación brindará la energía necesaria a cada uno de los elementos del dispositivo. El subsistema de radiocontrol recibirá las órdenes de movimiento para el posicionamiento inicial y

rescate del dispositivo. Con la posición destino, determinada previamente o informada por el subsistema de telemetría, el sistema central de procesamiento determinará la trayectoria a seguir en función de su posición actual, obtenida por un módulo de posicionamiento global (GNSS) incluido en el subsistema de posición. Este último, además, cuenta con sensores inerciales que brindarán información para los ajustes de la trayectoria. Con la trayectoria determinada, se enviarán comandos al subsistema de propulsión para el manejo de los motores. Una vez que se llega a destino, se activará el subsistema de toma de muestra.



Fig. 3. Sistema de toma de muestras.

De dicha arquitectura, en el presente trabajo se presenta la implementación parcial del subsistema central de procesamiento, el subsistema de radio control y el subsistema de propulsión. Implementación mínima para lograr la navegación.

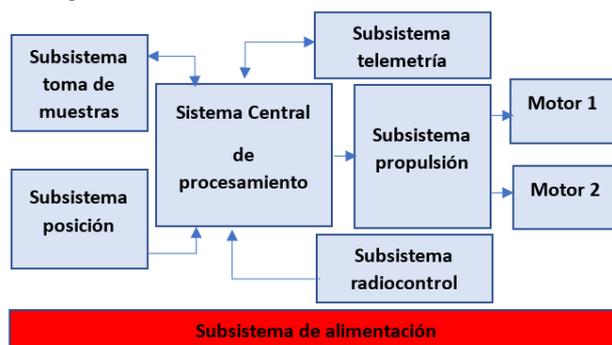


Fig. 4. Arquitectura del USV.

C. Hardware

En lo que respecta al sistema central de procesamiento se empleó la placa Orange Pi PC2 es una SBC (Single-Board Computer), la

cual integra un procesador Quad-core de 64 bits ARM-Cortex A53, 1GB de memoria RAM (DDR3), 2GB de memoria flash (NOR), interfaz para SD y múltiples interfaces de comunicación [13]. El sistema operativo embebido en la plataforma seleccionada es un Armbian [14], que no es otra cosa que un Debian (una de las distribuciones de Linux más populares) compilado con los requerimientos necesarios para ejecutarse sobre procesadores con arquitectura ARM. Las rutinas fueron implementadas en C sobre un entorno Eclipse en PC, y luego se realizó una compilación cruzada (Cross-compilation) con ayuda de Linaro Toolchain ARMv7 [15] para embeberlas en la placa.

En lo que respecta a los subsistemas actualmente implementados (radio control y propulsión), los mismos se basaron en la plataforma Arduino Uno, la cual integra el microcontrolador ATMEGA328 [16], la elección de esto se fundamentó en su bajo costo, confiabilidad y disponibilidad. La programación de estos fue realizada a través del lenguaje propio de Arduino (lenguaje de programación de alto nivel Processing, similar a C++).

El subsistema de radio control se complementa con un módulo Lora RFM95CW de la firma HopeRF [17], el cual se comunica con el microcontrolador de la placa Arduino UNO a través de la interfaz SPI y con el sistema central de proceso a través de una interfaz serial con niveles RS232. En la figura N°5 se aprecia la placa realizada con el módulo de comunicaciones.

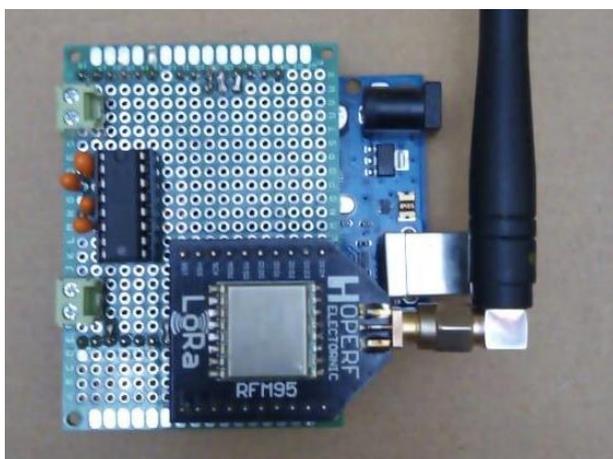


Fig. 5. Hardware asociado al subsistema de radio control.

En lo que respecta al subsistema de propulsión, a la plataforma Arduino se le suman

dos circuitos del tipo puente H para el manejo de los motores de continua, encargados del movimiento. Los mismos están basados en transistores discretos (IRL540 e IRF9540), permitiendo una implementación económica y fácil de reparar. En la figura N°6 se puede observar, en un primer plano, el subsistema de propulsión, mientras en el segundo plano se encuentra el sistema central de procesamiento, plataforma Orange Pi junto a placas conversores de niveles RS232, sensores inerciales y módulo de posicionamiento (estas últimas dos no utilizadas para la implementación del trabajo que se describe).

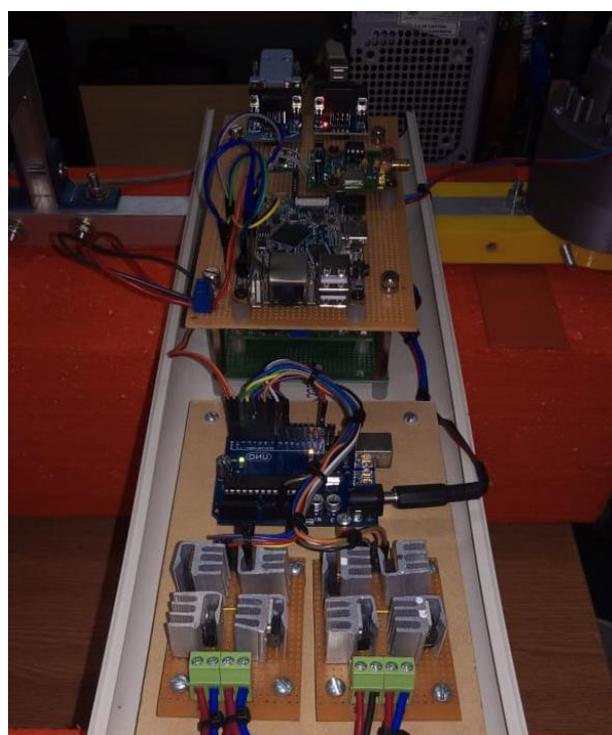


Fig. 6. Hardware asociado al subsistema de radio control.

D. Firmware

La puesta en marcha del prototipo del vehículo no tripulado de superficie implicó la definición de un protocolo de comunicación entre los distintos subsistemas. En esta etapa se definieron tres comandos: \$AT, \$MA y \$CM. El primero de ellos para indicar el modo de funcionamiento autónomo del vehículo, el segundo para indicar el modo de funcionamiento remoto y el último para enviar instrucciones al subsistema de propulsión, ya sea recibidas a través del subsistema de radio control o generadas por el sistema de procesamiento, como se prevé en un futuro. A continuación, se describe el comando \$CM:

$\$CMw,xxx,yyy,zzz,<checksum>,<CR>$

Donde:

- \$CM es el encabezado de la trama.
- w es el número de motor (1 o 2)
- xxx es el estado del motor (ONN u OFF)
- yyy es el sentido de giro (DER o IZQ)
- zzz es el nivel de la señal PWM (000-255).
- <checksum> es la suma de verificación.
- <CR> carácter 0x13 - fin de la trama.

El firmware asociado al subsistema de procesamiento, por estar implementado sobre un sistema operativa Linux, consta de un programa principal y de múltiples hilos de tareas ejecutándose en paralelo. El programa principal inicializará dichos hilos y luego permanecerá en un bucle infinito. A continuación, se puede ver el código asociado a la definición de los hilos y su inicialización.

```
pthread_t thread_rc, thread_timer;
```

```
if (pthread_create(&thread_rc, NULL,
tarea_RC, NULL) != 0) {
    perror("Error al crear el hilo de RC.");
    return 1;
}
```

```
if(pthread_create(&thread_timer, NULL,
tarea_timer, NULL) != 0) {
    perror("Error al crear el hilo del timer.");
    return 1;
}
```

En la figura N°7 se puede apreciar el diagrama de flujo del hilo asociado a la comunicación serial (entre el subsistema de radio control y el de procesamiento y entre este último y el subsistema de propulsión). Dicho programa consta de dos estados. Se permanece en el primero de ellos mientras el carácter recibido no sea "\$", una vez recibido se pasa al siguiente estado en donde se almacenan los caracteres recibidos hasta la llegada de la indicación de fin de trama. Recibido dicho carácter, se analiza si el comando corresponde a un comando de configuración (modo automático o modo manual) o si corresponde a un comando de manejo remoto. En el primer caso se configura

el sistema en el modo indicado, mientras en el segundo se retransmite el comando recibido al subsistema de propulsión. Tal como se ha mencionado previamente, el presente trabajo detalla una implementación parcial, es por este motivo que en el bloque pendiente actualmente no se ejecuta ninguna acción.

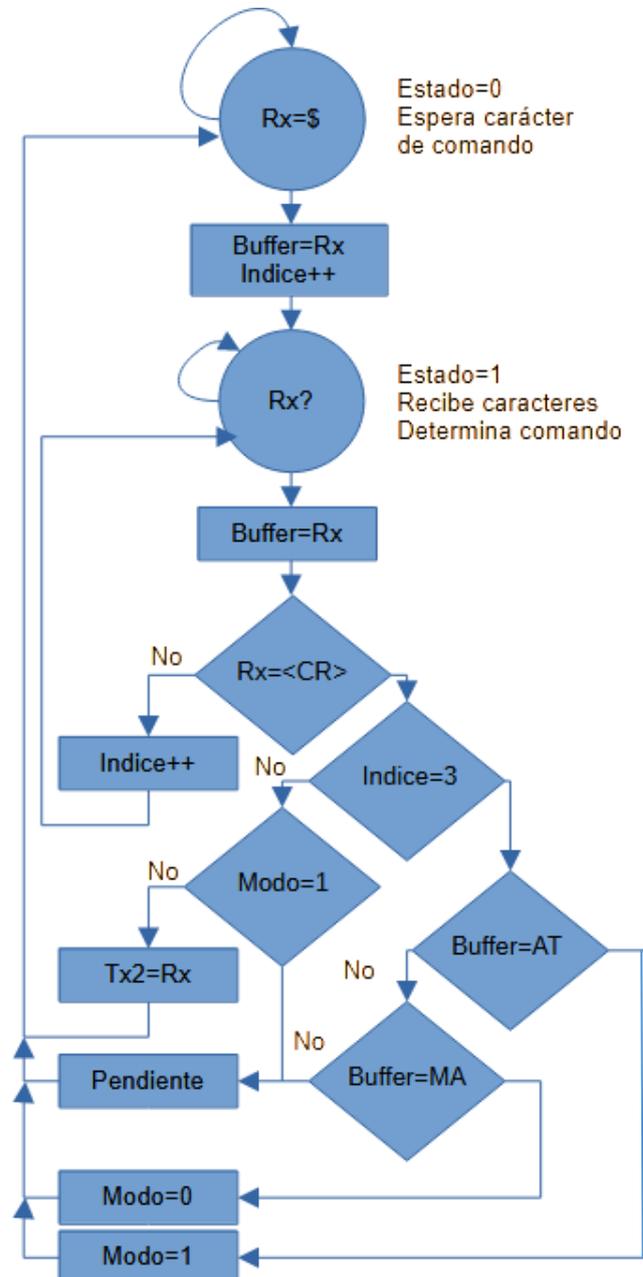


Fig. 7. Diagrama de flujo del hilo de recepción del subsistema de procesamiento.

La implementación del firmware del subsistema de propulsión no requirió la inclusión de librerías adicionales a las que vienen por defecto con el entorno. Se emplearon las siguientes funciones: *Serial.read* y *Serial.write* para la recepción de los

comandos y su confirmación; *Serial.println* para brindar información de consola; *digitalWrite* para el encendido y el apagado de los motores y para indicar el sentido de giro de estos, y *analogWrite* para manejar su velocidad.

En la figura N°7 se puede ver el diagrama de flujo del programa implementado (por cuestiones de espacio y claridad la opción de que el comando \$CM2 sea detectado se omitió, siendo su desarrollo igual al del comando \$CM1). Una vez configuradas las distintas funcionalidades de los pines, el procesador permanece esperando la recepción de un carácter. Los caracteres recibidos son almacenados hasta la recepción del fin de línea, momento en que se analiza la trama. Se verifica si el comando está asociado al motor 1 o al motor 2, si se le indica que se encienda o se apague, el sentido de giro (derecho o izquierdo) y a que velocidad debe girar. En el caso de que los parámetros no sean correctos se informa un error y no se ejecuta ningún comando.

En lo que respecta al subsistema de Radio Control, se emplearon las librerías de Sandeep Mistry versión 0.8.0, disponibles en: <https://github.com/sandeepmistry/arduino-LoRa>.

El código de la aplicación se basó en el ejemplo "LoRaReceiver" disponible en la documentación de la librería. A continuación, se presenta el bucle principal del programa, el cual corresponde al del ejemplo con el simple agregado del envío de los datos recibidos (solo los datos) por el puerto serie.

```
void loop() {
  // try to parse packet
  int packetSize = LoRa.parsePacket();
  if (packetSize) {
    // received a packet
    Serial.print("Received packet ");
    // read packet
    while (LoRa.available()) {
      Serial.print((char)LoRa.read());
      Serial.write((char)LoRa.read());
    }
    // print RSSI of packet
    Serial.print(" with RSSI ");
    Serial.println(LoRa.packetRssi());
  }
}
```

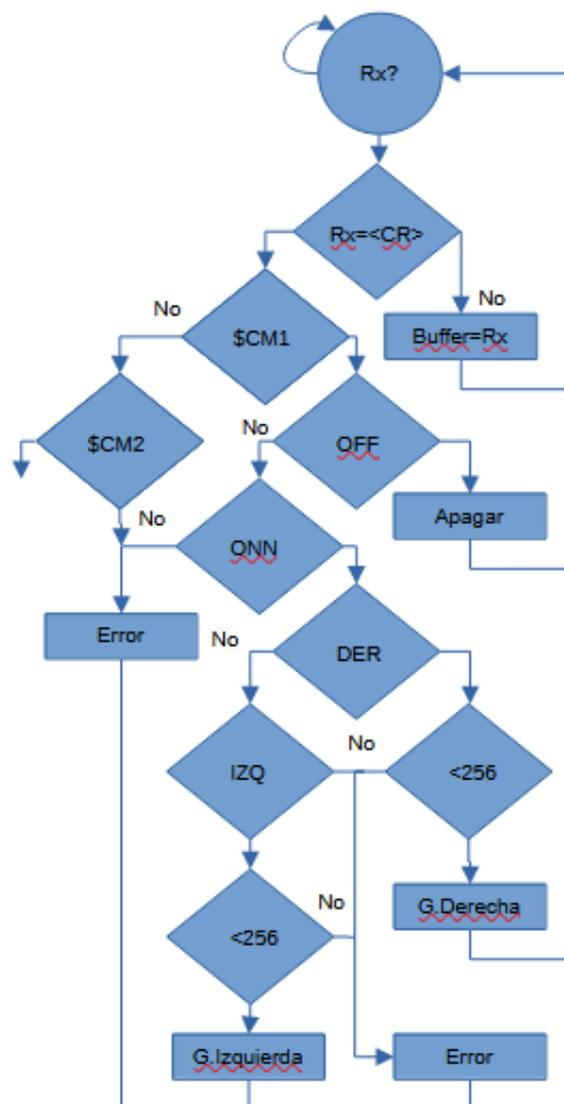


Fig. 7. Diagrama de flujo del hilo de recepción del subsistema de propulsión.

III. PRUEBAS Y RESULTADOS

Las pruebas de la implementación realizada se llevaron adelante en las piscinas de natación de la Universidad Nacional de la Matanza, en la figura N°2 se puede ver la embarcación DIIT-A navegando con el sistema de muestreo instalado y la electrónica de control previamente mencionada. Con estas se verificó el funcionamiento de sistema de propulsión, navegación y maniobrabilidad, el funcionamiento de electrónica de control de motores, la operatividad del sistema embebido central y la comunicación con los distintos subsistemas, el enlace de comunicación Lora para control remoto y el funcionamiento general de todos los sistemas en conjunto. Todas estas

pruebas resultaron satisfactorias, aunque se detectaron aspectos de mejora en la implementación del sistema de propulsión realizado con impresión 3D.

- [15] Linaro. [Online]. Disponible: <https://www.linaro.org/about/>
[16] Arduino Uno Rev3. [Online]. Disponible: <https://store.arduino.cc/products/arduino-uno-rev3>
[17] HopeRf. [Online]. Disponible: <https://www.hoperf.com/modules/lora/RFM95W-Special.html>

RECONOCIMIENTOS

Se agradece la colaboración del Departamento de Humanidades y Ciencias Sociales de la Universidad Nacional de la Matanza por brindar acceso a las instalaciones de las piscinas.

REFERENCIAS

- [1] Oreste D. Lupi, Ignacio J. Zaradnik, Monica B. Canziani. "Estado del arte de los sistemas de monitoreo de calidad de agua". Revista Digital de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de la Matanza, vol. 5, nro. 2, (2020).
- [2] Oreste D. Lupi, Diego H. Turconi, Javier M. Slawiski. "Monitoreo de ecosistemas acuáticos". Revista Digital de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de la Matanza, vol. 6, nro. 2, (2021).
- [3] Ignacio Zaradnik, Facundo Dominguez, Augusto Kumvich, Oreste D. Lupi, Diego Caccaviello. "Desarrollo de middleware y aplicación cliente para sistema de miniboyas ambientales". Libro de actas XXIV Workshop de Investigadores en Ciencias de la Computacion, pp.438-442, (2022).
- [4] Sergio Calvo Anglada.: Detección y Caracterización por Tamaños de Micro y Nanoplasticos en Muestras de Interes Ambiental. Universidad Zaragoza (2020)
- [5] M. C. Evans and C. S. Ruf.: Toward the Detection and Imaging of Ocean Microplastics With a Spaceborne Radar. En IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-9, (2022)
- [6] D. G. Kotsifaki, C. Ripken and S. N. Chormaic.: Detection and analysis of microplastics in the subtropical ocean of Okinawa using micro-Raman Optical Tweezers. En 2021 International Workshop on Metrology for the Sea; Learning to Measure Sea Health Parameters (MetroSea), Reggio Calabria, Italy, pp. 158-162, (2021)
- [7] O. Malyuskin.: Microplastic Detection in Soil and Water Using Resonance Microwave Spectroscopy: A Feasibility Study. En in IEEE Sensors Journal, vol. 20, no. 24, pp. 14817-14826, (2020)
- [8] NauticExpo, <https://www.nauticexpo.es/fabricante-barco/vehiculo-no-tripulado-marino-23029.html>
- [9] Alexander T. Demetillo, Evelyn B. Taboada.: Real-Time Water Quality Monitoring For Small Aquatic Area Using Unmanned Surface Vehicle. En Engineering, Technology & Applied Science Research Vol. 9, No. 2, pp. 3959-3964 (2019)
- [10] Wonse Jo, Yuta Hoashi, Lizbeth Leonor Paredes Aguilar, Mauricio Postigo-Malaga, José M. Garcia-Bravo, Byung-Cheol Min.: A low-cost and small USV platform for water quality monitoring. En HardwareX, Volume 6, (2019)
- [11] G. Ferri *et al.*: Design, fabrication and first sea trials of a small-sized autonomous catamaran for heavy metals monitoring in coastal waters: En 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, pp. 2406-2411, (2011)
- [12] G. Ferri, A. Manzi, F. Fornai, F. Ciuchi and C. Laschi.: The HydroNet ASV, a Small-Sized Autonomous Catamaran for Real-Time Monitoring of Water Quality: From Design to Missions at Sea. En IEEE Journal of Oceanic Engineering, vol. 40, no. 3, pp. 710-726, (2015)
- [13] Orange_Pi. [Online]. Disponible: <http://www.orange-pi.org/html/hardWare/computerAndMicrocontrollers/details/Orange-Pi-PC-2.html>
- [14] Arbian. [Online]. Disponible: <https://www.arbian.com/>



Evaluación de aguas superficiales en el norte argentino mediante imágenes satelitales

Juan Vorobioff

Norberto Boggio

Federico Checozzi

Facultad Regional Buenos Aires,
Universidad Tecnológica Nacional,

Facultad Regional Buenos Aires,
Universidad Tecnológica Nacional,

Facultad Regional Buenos Aires,
Universidad Tecnológica Nacional,

Instituto de Nanociencia y
Nanotecnología CNEA - CONICET,

Instituto de Nanociencia y
Nanotecnología CNEA - CONICET,

Instituto de Nanociencia y
Nanotecnología CNEA - CONICET,

Bs. As. Argentina

Bs. As. Argentina

Bs. As. Argentina

jvorobioff@frba.utn.edu.ar

norbertoboggio@cnea.gob.ar

federicochecoizzi@cnea.gov.ar

Resumen. La región noreste de Argentina, especialmente la ciudad de Formosa y sus alrededores, posee una rica ecodiversidad que está siendo gravemente impactada por el cambio climático, la deforestación y la contaminación humana. Este estudio analiza las variaciones en las aguas superficiales de esta área utilizando datos satelitales globales obtenidos a través de la plataforma Google Earth Engine y dispositivos Raspberry Pi para la recolección de datos en campo. Se examinan la presencia de agua y las estadísticas relacionadas para identificar los cambios hídricos en la región, clasificando diferentes zonas según los tipos de transiciones observadas desde 1984 hasta 2015. Se proponen diversos enfoques para el análisis de áreas de interés, destacando los cambios estacionales provocados por la deforestación. Además, la región de estudio puede ajustarse fácilmente para investigar otras áreas de interés.

Palabras clave: Imágenes satelitales, Agua superficial, Ríos, Raspberry Pi, Algoritmos IA.

I. INTRODUCCIÓN

La biodiversidad en la provincia de Formosa ha sido impactada por diversas actividades humanas, como la deforestación para la obtención de durmientes, tanino y madera para aserraderos, la expansión de la agricultura con la introducción de ganado bovino, equino y caprino (lo que ha llevado al sobrepastoreo), el

monocultivo de algodón en la zona central, y la caza y captura comercial de fauna silvestre. El cambio climático también ha tenido repercusiones ambientales significativas. Además, el río Pilcomayo está siendo contaminado por la actividad minera en Bolivia [1]. Una consecuencia conocida de la deforestación es la alteración de los ciclos del agua, debido al papel moderador que desempeñan los bosques y selvas (ya sea absorbiendo o transpirando agua). Esto provoca una prolongación de la estación seca y un aumento en la frecuencia de inundaciones [2]. La eliminación de la vegetación afecta a muchas especies, creando zonas con diferentes niveles de humedad, lo que resulta en variaciones en la fauna y la vegetación. Formosa cuenta con diversos recursos hídricos, y la inclinación del terreno influye en el curso de los ríos Pilcomayo, Bermejo y varios arroyos que corren en paralelo a estos ríos. El río Pilcomayo se origina en los deshielos de Bolivia, mientras que el río Bermejo nace en el oeste de Argentina. Ambos ríos desembocan en el río Paraguay, que en su paso por territorio argentino tiene un caudal ancho que permite la navegación de buques [3].

El clima de la provincia es tropical. En la región oriental, las lluvias son constantes durante todo el año, superando los 1200 mm anuales cerca del río Paraguay. El clima de Formosa se clasifica en subtropical sin estación



seca en el este y subtropical con invierno seco en el oeste. Durante el verano, las temperaturas pueden superar los 44 °C en toda la provincia, mientras que en invierno, el viento pampero puede hacer que las temperaturas desciendan hasta los 6 °C. El viento del norte, cálido y húmedo, sopla con fuerza desde finales de agosto hasta principios de abril, afectando el estado de ánimo de la población [4].

Este estudio se centra en la ciudad de Formosa y sus alrededores. Es crucial analizar las propiedades del agua superficial en el noreste de Argentina para medir los impactos ambientales causados por la actividad humana. Para ello, se desarrollaron programas en JavaScript para la plataforma Google Earth Engine, los cuales pueden adaptarse fácilmente para investigar diferentes regiones del país [5].

II. METODOLOGÍA

Para realizar el análisis, se utiliza una placa Raspberry Pi para procesar y visualizar los datos en tiempo real, facilitando la integración de la información satelital con los datos locales y proporcionando una visión más completa de las variaciones en las aguas superficiales. Se utiliza el sistema operativo Linux que brinda muchas facilidades para este propósito.

Se utiliza el modelo 5 de Raspberry Pi, cuenta con un procesador Broadcom BCM2712 de cuatro núcleos ARM Cortex-A76 a 2.4 GHz, acompañado de 8 GB de RAM LPDDR4x. La placa ofrece almacenamiento a través de una tarjeta microSD hasta 2TBytes. Se conecta mediante Ethernet Gigabit, WiFi dual-band y Bluetooth 5.0. Incluye dos puertos USB 3.0, dos puertos USB 2.0, y dos salidas micro-HDMI. También tiene un conector de 40 pines GPIO, y utiliza un puerto de alimentación USB-C. Todo en un formato compacto.

La Raspberry Pi se seleccionó por su bajo costo, capacidad de procesamiento eficiente y compatibilidad con conexiones a internet, lo que permite la adquisición y procesamiento de datos satelitales en tiempo real directamente desde Google Earth Engine. En esta primera etapa del proyecto, se usa solamente internet para acceder a Google Earth Engine, sin sensores adicionales. Aunque actualmente no se utilizan sensores, la Raspberry Pi ofrece la flexibilidad de agregar opcionalmente cámaras

de video USB estándar y sensores ambientales de temperatura y humedad, como el modelo DHT22, que se conecta directamente al puerto GPIO de la placa. Esta versatilidad la convierte en una plataforma adecuada para nuestros proyectos, donde en forma opcional se pueden verificar los datos satelitales de internet con los registrados por los sensores conectados a la placa.

Además, se implementó un punto de acceso WiFi, lo que permite acceder y controlar la Raspberry Pi de manera remota desde un celular u otro dispositivo sin necesidad de un monitor físico. Esto se facilita gracias al uso de VNC (Virtual Network Computing), que permite la gestión remota del sistema con una interfaz gráfica completa.

Se está utilizando una sola placa Raspberry Pi, pero se puede expandir fácilmente mediante una red cableada LAN o inalámbrica WiFi para distribuir la información hacia otras redes de internet.

En resumen, la Raspberry Pi no solo proporciona una solución accesible y eficaz para manejar la información satelital a través de internet, sino que también ofrece opciones de expansión, conectividad remota y control sin monitor físico. Esto aumenta su utilidad en diversos contextos, todo a un costo significativamente bajo, con un requerimiento de espacio muy reducido y un bajo consumo de energía.

Mediante la placa en entorno Linux, se analizan las variaciones en las aguas superficiales en el noreste de Argentina, específicamente en la ciudad de Formosa y sus alrededores. Utilizando software desarrollado en la plataforma Google Earth Engine, se accede a los datos satelitales del Global Surface Water (GSW). Se examina la intensidad de los cambios en la detección del agua y se clasifican las regiones según sus variaciones desde 1984 hasta 2015. Se selecciona una región de interés y se proponen diferentes enfoques para detectar la presencia de agua y los cambios temporales entre dos períodos. Además, se clasifican las regiones de acuerdo con estos enfoques [6].

Se analiza la ocurrencia de agua, se indica las áreas donde hubo agua superficial entre 1984 y 2015, proporcionando una visión general de la dinámica del agua. Esta métrica

captura tanto la variabilidad como los cambios intra e interanuales [7]. La ocurrencia se calcula mensualmente mediante la proporción de detecciones de agua sobre el total de observaciones válidas, y luego se promedian las ocurrencias mensuales. Esto permite normalizar las variaciones causadas por el agua estacional, que podrían quedar ocultas con un promedio directo a lo largo de un período prolongado.

Se identificó la presencia de agua y se representó su ocurrencia en un mapa de la región. Primero, se carga la imagen que contiene información sobre las aguas superficiales a nivel mundial y se selecciona la banda de ocurrencia, que se expresa como un valor porcentual. Finalmente, se añade una capa de ocurrencia al mapa generado por Earth Engine, utilizando una paleta de colores que varía gradualmente del rojo al azul, y se centra el mapa en la ciudad de Formosa, ver Fig. 1.



Fig. 1. Presencia de agua entre 1984 y 2015. Rojo (mínimo), azul (máximo)

Se analiza una medida de cómo ha variado el agua superficial entre dos periodos: 1984-1999 (periodo 1) y 2000-2015 (periodo 2). La capa de datos satelitales denominada "Cambios de Intensidad" ofrece una medida de estos cambios en el agua superficial entre los periodos mencionados. Esta capa muestra el cambio en la ocurrencia mediante el promedio de la diferencia de ocurrencias entre pares de meses equivalentes de los dos periodos.

Se eligió la banda de cambio de intensidad

absoluta de los datos globales de agua superficial, utilizando una paleta de colores que varía gradualmente de rojo a negro y de negro a verde para valores entre -50 y 50. El mapa se centró de la misma manera que en el script de la sección anterior, ver Fig. 2.

Además, se utiliza un histograma para resumir los cambios en una región poligonal de interés (ROI) específica de 240 km² (como se muestra en la Fig. 2), ver Fig. 3.

Posteriormente, se analiza la capa de transición de agua permitiendo diferenciar varias clases de zonas según su ocurrencia: sin presencia de agua, agua estacional, agua permanente, y agua efímera (permanente y estacional). En resumen, se observa una disminución en los niveles de agua en el segundo periodo. Además, se detectó agua en el 34 % de las zonas, de las cuales el 31,6 % corresponde a agua permanente, el 47,4 % a agua efímera estacional, y el resto a otras categorías. Con Earth Engine, se añadió una capa al mapa y se centró en la ciudad para visualizar la transición del agua.

Se selecciona la banda que contiene la información de transición y se genera un gráfico similar a los de las secciones anteriores para visualizar las diferentes clases (agua estacional, efímera permanente, etc.), ver Fig. 4. Luego, se calcula el área correspondiente a cada clase y se crea un gráfico de pastel para facilitar la visualización de la información. Se genera una imagen con dos bandas: clase de transición y área real de cada píxel en metros cuadrados. Se aplica un reductor sobre una región de interés para calcular el área por clase, sumando todas las áreas de los píxeles pertenecientes a cada clase, y se guardan los resultados en una lista.

Se genera una colección de características para cada clase, que incluye el número de clase, el nombre de la clase, el color (extraído de la imagen global de agua superficial) y el área calculada previamente. Por último, se crea el gráfico de torta, el cual requiere la colección de características obtenida previamente y una lista de colores, ver Fig. 5.

III. RESULTADOS Y DISCUSIÓN

La Fig. 2 ilustra los cambios de intensidad en

la ocurrencia de agua para una región poligonal de interés (ROI) de 240 km². Un histograma resume estos cambios de intensidad para la región de interés, ver Fig. 3.

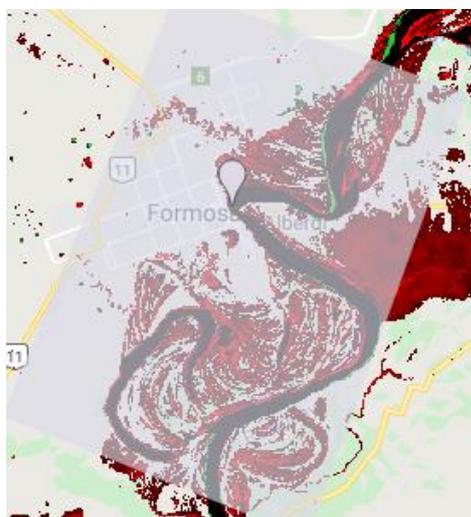


Fig. 2. Variaciones en la intensidad de la ocurrencia de agua. Rojo (mínimo), negro (medio), verde lima (máximo)

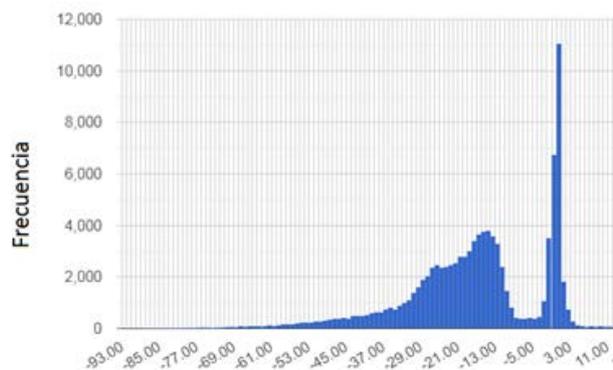


Fig. 3. Histograma de variaciones en la intensidad de la ocurrencia de agua

En la Fig. 4 se muestra un mapa indicando diferentes áreas con distintas clases de transición. En la Fig. 5 se muestra un gráfico de torta, indicando los porcentajes de áreas correspondientes a las distintas clases

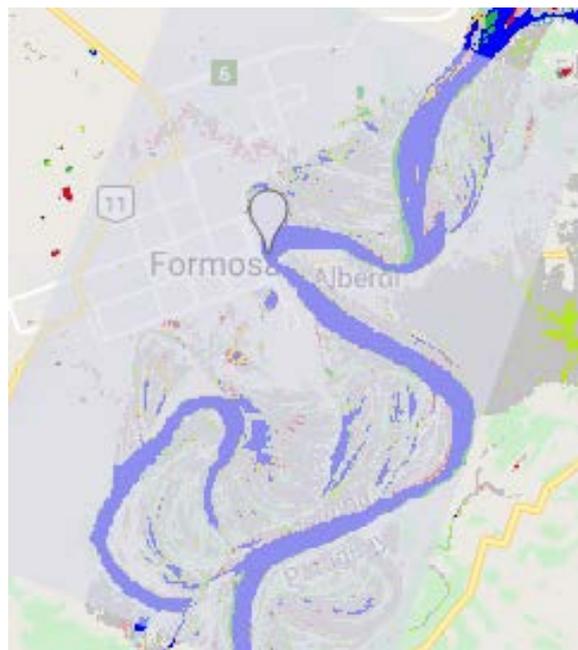


Fig. 4. Mapa de clases de transición

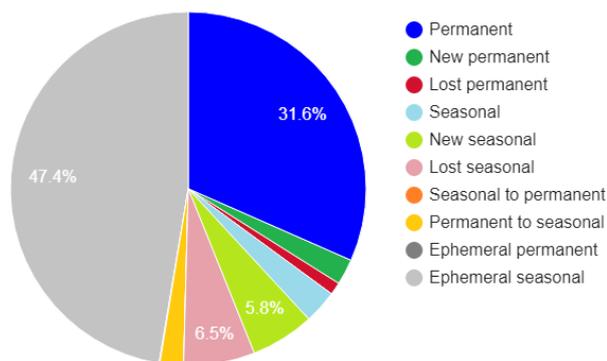


Fig. 5. Resumen de áreas de clases de transición

Se estima el porcentaje de agua en la región contando el área de los píxeles cuya clase de transición corresponde a agua y luego calculando el porcentaje respecto al área total. Primero, se crea una imagen cuyos elementos representan el área del píxel correspondiente a la banda de transición si pertenece a una clase permanente (clases 1, 2 o 7, que corresponden a permanente, nueva permanente y estacional a permanente) y un valor nulo en caso contrario. Esto se logra multiplicando elemento a elemento una matriz booleana (1 si la clase es 1, 2 o 7, 0 en caso contrario) por una matriz que contiene las áreas de cada píxel de la banda. Luego, se suma el área de los píxeles mediante un reductor. Finalmente, se obtiene el área de la región poligonal y, con base en eso,

se calcula el porcentaje. Se estima que la masa de agua según este criterio es del 11,65%. Es importante visualizar los cambios en la estacionalidad provocados por la deforestación en la región. Para ello, se utiliza el historial anual del agua superficial, que proporciona información sobre el agua permanente y estacional desde 1984 hasta 2015. Debido a la falta de información, se realiza una comparación entre los años 1986 y 2015.

- [5] J.F. Pekel, A. Cottam, N. Gorelick, A.S. Belward, "High-resolution mapping of global surface water and its long-term changes," *Nature*, 2016. <https://doi.org/10.1038/nature20584>
- [6] A.C. Fassoni-Andrade, R.C.D. de Paiva, C.M. Rudorff, C.C.F. Barbosa, E.M.L. de M. Novo, "High-resolution mapping of floodplain topography from space: A case study in the Amazon," *Remote Sensing of Environment*, 2020. <https://doi.org/10.1016/j.rse.2020.112065>
- [7] R. C Menni., Miquelarena, A. M., Lopez, H. L., Casciotta, J. R., Almiron, A. E., & Protogino, L. C. (1992). Fish fauna and environments of the Pilcomayo-Paraguay basins in Formosa, Argentina. *Hydrobiologia*. <https://doi.org/10.1007/BF00006154>

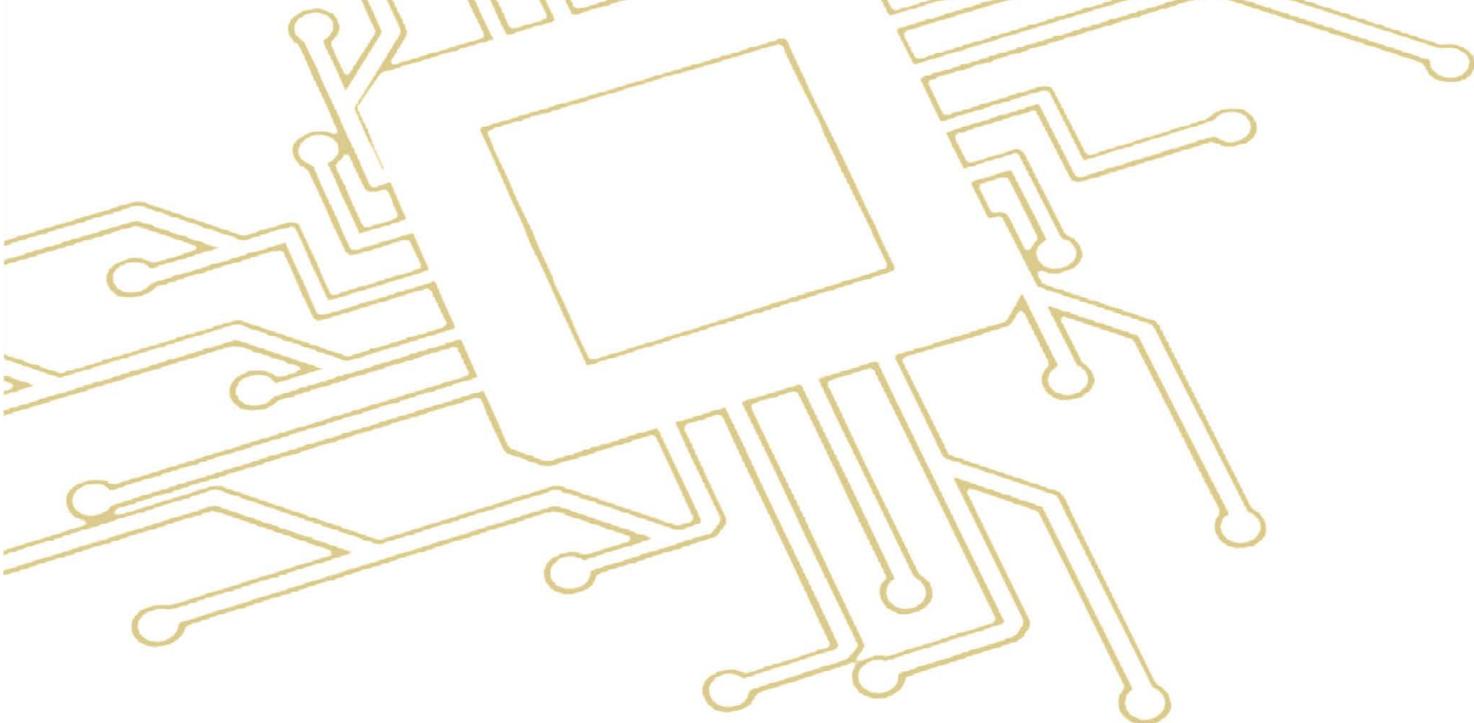
IV. CONCLUSIONES

Este estudio ha cuantificado las variaciones en las aguas superficiales de la ciudad de Formosa utilizando imágenes satelitales en el periodo de 1984 a 2015. Se destaca la eficacia de los datos satelitales del Global Surface Water (GSW) y el uso de dispositivos Raspberry Pi para la recolección de datos en campo.

El análisis del histograma revela que muchas zonas presentan una tendencia negativa en la ocurrencia de agua. Además, el historial anual de agua muestra un aumento en la estacionalidad, atribuible a la deforestación. Con las modificaciones adecuadas, esta metodología puede aplicarse para estudiar diferentes regiones de interés. La herramienta desarrollada, que combina imágenes satelitales y datos recolectados con Raspberry Pi, es extremadamente potente para el análisis de aguas superficiales y el estudio de la modificación de los cauces de los ríos, y puede ser utilizada en otras regiones del país.

REFERENCIAS

- [1] F.C. Gorleri, "Diversidad y variación estacional del ensamble de Aves en distintas," 2018.
- [2] D. Ellison, C.E. Morris, B. Locatelli, D. Sheil, J. Cohen, D. Murdiyarto, V. Gutierrez, M. van Noordwijk, I.F. Creed, J. Pokorny, D. Gaveau, D.V. Spracklen, A. Bargaes Tobella, U. Istedt, A.J. Teuling, S.G. Gebrehiwot, D.C. Sands, B. Muys, B. Verbist, E. Springgay, Y. Sugandi, C.A. Sullivan, "Trees, forests and water: Cool insights for a hot world," *Global Environmental Change*, vol. 43, pp. 51-61, 2017. <https://doi.org/10.1016/j.gloenvcha.2017.01.002>
- [3] S. Gavilan, J.I. Pastore, J. Uranga, A. Ferral, A. Lighezzolo, P. Aceñolaza, "Metodología operativa para la obtención de datos históricos de precipitación a partir de la misión satelital Tropical Rainfall Measuring Mission. Validación de resultados con datos de pluviómetros," *Revista de La Facultad de Agronomía*, 2019. <https://doi.org/10.24215/16699513e011>
- [4] M. Mendoza, "Range area and seasonal campsites of Toba bands in western Chaco, Argentina," *Before Farming*, 2003. <https://doi.org/10.3828/bfarm.2003.4.2>



Eje Temático

Aplicaciones en Educación y Enseñanza





Validación de sistemas de control en planta de corte al vuelo para el entorno académico*

Diego Fernando Ramírez Jiménez

Diego Camilo Guisao Herrera

Sebastián Pulido Catillo

Programa de Ingeniería Electrónica

Programa de Ingeniería Electrónica

Programa de Ingeniería Electrónica

Universidad del Quindío

Universidad del Quindío

Universidad del Quindío

Armenia – Quindío, Colombia

Armenia – Quindío, Colombia

Armenia – Quindío, Colombia

dframirez@uniquindio.edu.co

dcguisao@uqvirtual.edu.co

spulidoc@uqvirtual.edu.co

Resumen— Este trabajo presenta el diseño e implementación de un controlador PID para una planta de corte al vuelo, empleando el kit de desarrollo LAUNCHXL-F28377S de Texas Instruments. La planta fue desarrollada en el área de Automatización y Control del programa de Ingeniería Electrónica de la Universidad del Quindío e incluye una banda transportadora y un mecanismo de corte emulado, garantizando seguridad y funcionalidad para fines educativos. El sistema integra sensores, actuadores y un algoritmo de control que gestiona la operación en tiempo real. La simulación de la planta y el diseño del controlador PID se llevaron a cabo en MATLAB y Simulink, y la programación del controlador PID en la LAUNCHXL-F28377S se realizó mediante Code Composer Studio – CCS. Como resultados se destacan la respuesta del prototipo experimental usado y la capacidad de ofrecer soluciones económicas y robustas, que faciliten el aprendizaje práctico y la investigación aplicada en el uso de los microcontroladores y sus aplicaciones en la industria.

Palabras clave: corte al vuelo, LAUNCHXL-F28377S, MATLAB, microcontroladores c2000, Texas Instruments.

I. INTRODUCCIÓN

A. Problema de investigación

En la ingeniería se han presentado diferentes retos para la enseñanza en las diferentes áreas de formación y la automatización y el control de procesos es una de ellas. Lo anterior, cobra importancia cuando la formación pretende que un ingeniero en formación esté en contacto

permanente con los desafíos cotidianos que tienen lugar en la industria. En este contexto, es necesaria la preparación en áreas como la electrónica analógica, electrónica digital, modelado dinámico y la introducción al control. Este último, es uno de los más relevantes en la automatización de procesos ya que se pueden aprovechar diferentes montajes para facilitar su aprendizaje.

En los laboratorios del programa de Ingeniería Electrónica de la Universidad del Quindío, se han identificado deficiencias de plantas de prueba de corte continuo relacionadas con los diferentes contextos y posibles escenarios que se encuentran fuera del entorno académico.

Las plantas de prueba han tenido especial interés en los últimos años debido a la importancia de la automatización de procesos y el control en las empresas y la alineación de la academia con estos [1]. Por lo anterior, se presenta una propuesta de planta de corte al vuelo y su posible uso en los espacios formativos del área de automatización y control del programa de Ingeniería Electrónica de la Universidad del Quindío.

B. Trabajos previos

La importancia de los procesos de enseñanza radica en el uso de estrategias que conlleven a un aprendizaje efectivo. Por tal motivo, se hace necesario precisar el valor que tienen las labores académicas en los procesos de automatización y control que ayudan a los estudiantes a potenciar habilidades como la gestión del tiempo y la optimización de recursos en el desarrollo de

actividades que requieren precisión y robustez, tal como se realiza en la industria [2]. Por lo anterior, en la Tabla I se presentan algunos trabajos realizados en plantas de corte al vuelo, los cuales se desarrollaron utilizando diferentes métodos.

TABLA I
Trabajos previos sobre plantas de corte al vuelo/cizalla volante

Nombre del Artículo	Resumen
Flying Shear Control System [3]	Se diseñó un sistema de control para un servo en cizallas volantes utilizando un controlador servo universal con función de leva electrónica. El sistema se integró con un motor de engranaje y se evaluó mediante simulaciones y pruebas experimentales. El sistema mejoró la precisión del corte y la respuesta dinámica del sistema.
Dynamic Simulation and Control Strategy of Centrifugal Flying Shear [4]	Se desarrolló un modelo tridimensional de una cizalla volante centrífuga, utilizando Solidworks/COSMOS Motion y MATLAB/SimMechanics. Se realizó un análisis estructural y una simulación dinámica del proceso de corte. Los resultados proporcionan una base teórica para el diseño del sistema de control electrónico y optimización de parámetros en cizallas volantes similares.
Improving the Operation of a Flying Dividing Shear by Using Direct Torque Control [5]	Este artículo describe la aplicación de la tecnología de control de par directo (DTC) para mejorar el control del par en un sistema de cizalla volante divisoria en un laminador de acero. La tecnología DTC mejoró significativamente el control del par y la precisión del corte en aplicaciones de alta velocidad.
FPGA implementation of general purpose real time controller-applied to flying shears in hot rolling mills [6]	Este artículo presenta un controlador en tiempo real basado en FPGA para su aplicación en cizallas volantes en molinos de laminación en caliente. Se utilizan herramientas de diseño como VHDL y Xilinx ISE. El proceso incluye la implementación del controlador en una FPGA Spartan-3 y la integración con el sistema de corte existente. El método se centra en la posición y control de velocidad para minimizar la energía y el desperdicio durante el corte. Los resultados muestran que el controlador basado en FPGA mejora el rendimiento en términos de precisión y velocidad de corte en comparación con los controladores tradicionales.
Metallurgical Investigation of Flying Shear Blades at Hot Rolling Rebar Mills [7]	Se utilizaron herramientas como microscopía electrónica de barrido (SEM), espectrometría de energía dispersiva (EDS) y ensayos de dureza. El proceso incluyó la evaluación de la microestructura y la composición química de las cuchillas, así como pruebas de resistencia al desgaste. El análisis reveló que las cuchillas estaban compuestas de acero AISI-D2

	y presentaban carburo de cromo, lo que contribuía a su fragilidad. Se recomendó una modificación en el tratamiento térmico para mejorar la resistencia al desgaste, prolongando así la vida útil de las cuchillas y reduciendo el tiempo de inactividad en la producción.
Diseño de una cizalla industrial para el corte de láminas de acero al silicio utilizadas en transformadores secos de energía eléctrica [8]	Se realizó el diseño mecánico de una cizalla al vuelo para el corte de láminas de acero al silicio en la construcción de núcleos de transformadores, enfocando la atención en las pérdidas que se generan en los núcleos por el corte irregular en estas piezas. Se minimiza el esfuerzo que normalmente realiza un operario, para aumentar la productividad en las empresas.
Modernización del sistema de control de la cizalla volante del proceso de laminación de la planta siderúrgica de Aceros Arequipa [9]	Se desarrolló un proceso continuo de rolado para archivadores en una empresa de la región de Laguneros en Perú. En este proyecto se centraron en la realización de tableros para el control por PLC del corte al vuelo, con el fin de hacer más competitiva la producción.
Solid Modeling and Optimization Design of Crank Rocking Flying Shear [10]	Se realizó un trabajo sobre la cizalla volante de cigüeñal, lo cual es un proceso más específico de ingeniería mecánica donde modelaron el comportamiento de la misma a partir del software Solid Works, con el fin de tener un análisis cinético a partir del análisis tridimensional con el ensamble y la animación.
Laboratorio Distribuido con Acceso Remoto para la Enseñanza de la Robótica [11]	En este proyecto se desarrolló un laboratorio de plantas de prueba las cuales consisten en un robot industrial y un robot móvil. El enfoque está orientado a la conexión remota de los mismos, propiciando un primer avance a nivel local en laboratorios remotos, basados en el paquete Servlets Java.

Los trabajos presentados en la Tabla I muestran que las aplicaciones realizadas en plantas de corte al vuelo han estado orientadas al control de velocidad de corte y modelamiento dinámico de la misma, en la mayoría de los casos a nivel de simulación como sucede en [8], donde el trabajo realizado se centró en el diseño de las piezas mecánicas para la precisión de corte, dejando de lado aspectos importantes en el proceso de implementación y uso de dispositivos electrónicos que permitan la implementación de sistemas de control.

C. Contribuciones

En este trabajo se presenta el diseño y la implementación de una planta de corte al vuelo para el entorno académico, la cual es controlada a través de un microcontrolador C2000 usando el kit de desarrollo LAUNCHXL-F28377S de Texas Instruments para validar su funcionamiento. Adicionalmente, otras

contribuciones importantes con el desarrollo de este trabajo son:

- Sistemas de control digital en tiempo real que puedan ser replicables en otros prototipos de prueba.
- Uso de microcontroladores desarrollados especialmente para aplicaciones de control en tiempo real como por ejemplo: industria automotriz, medicina, control de motores automatización, entre otras [12].
- Diseño e implementación de controladores 100% digitales debido a que las variables, los lazos de control y el procesamiento de señales se realizan dentro del microcontrolador.
- Planta de prueba a escala real, modular, compacta, robusta y de bajo costo, a través de la cual los estudiantes del área de automatización y control del programa de Ingeniería Electrónica de la Universidad del Quindío pueden realizar prácticas y validación en el entorno académico de proceso industriales reales.
- Desarrollo de líneas y temas de investigación que integren al sector industrial en la implementación de soluciones eficientes para el mejoramiento de los procesos.
- Desarrollo de prototipos centrados en aspectos técnicos de su construcción como la mecánica y no solo en el desarrollo de estrategias de control o modelamiento a nivel de simulación.

D. Estructura del trabajo

La estructura del trabajo está organizada en diferentes secciones o capítulos como sigue: el capítulo II presenta el diseño del sistema, una pequeña descripción del diseño y la implementación de la planta de corte al vuelo, proceso de modelamiento de la dinámica del sistema y el controlador utilizado para validar su funcionamiento. El capítulo III presenta los resultados y pruebas de funcionamiento de la planta de corte al vuelo tanto a nivel de simulación como de implementación. Finalmente, el capítulo IV presenta las principales conclusiones sobre los aspectos desarrollados dentro del proceso.

II. DISEÑO DEL SISTEMA

A. Planta de corte al vuelo

Las máquinas de corte al vuelo son mecanismos de corte continuo que pueden operar en sincronismo entre la cuchilla y el material en movimiento, o a través del ajuste de velocidades con ayuda de dispositivos electrónicos adicionales. Existen diferentes tipos de máquinas de corte al vuelo, definidos en algunos casos por quienes fabrican este tipo de máquinas como por ejemplo la empresa de origen chino RISHBIN, quien dispone en su catálogo de máquinas de corte al vuelo para placas metálicas y las clasifican, según el tipo de corte, en máquinas rotativas, de palanca giratoria de manivela, de manivela excéntrica y de péndulo [13]. En las Fig. 1 y 2 se presentan algunos esquemas de los diferentes tipos de planta de corte al vuelo que existen a nivel industrial.

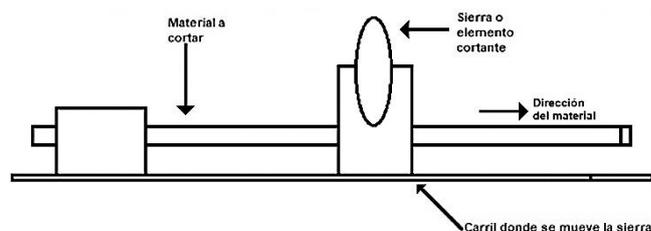


Fig. 1. Planta de corte al vuelo T1.

Fuente: los autores.

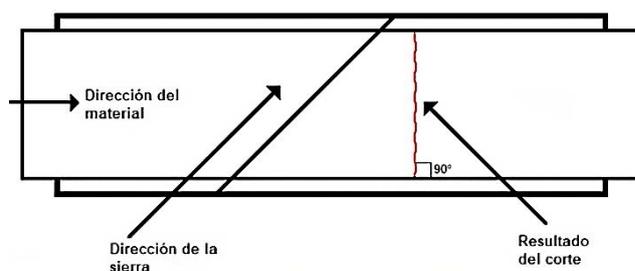


Fig. 1. Planta de corte al vuelo T2.

Fuente: los autores.

B. Estructura de la planta (sistema mecánico)

La estructura utilizada para el diseño y la implementación de la planta de corte al vuelo se presenta en la Fig. 3.

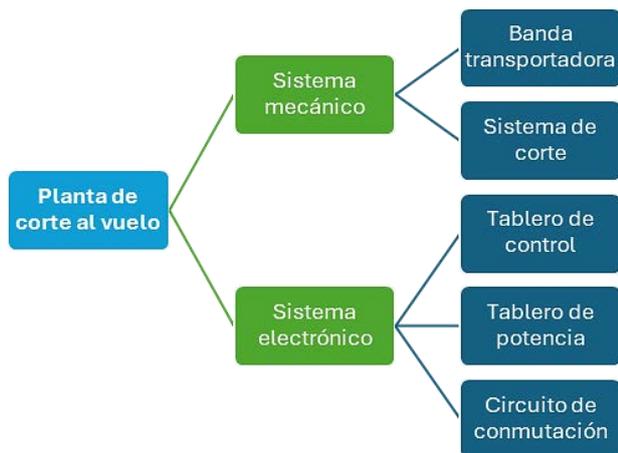


Fig. 3. Diagrama de bloques que representa la estructura de la planta de corte al vuelo.

Fuente: los autores.

En la Fig. 3 se aprecia el prototipo propuesto, conformado por dos sistemas principales. Por un lado, el sistema mecánico que cuenta una banda transportadora y un sistema de corte con movimiento en los ejes X, Y y Z. Por otra parte, el sistema electrónico posee un tablero de control donde se concentran periféricos, un circuito de potencia que concentra la tarjeta principal y un circuito de conmutación que posibilita el uso del sistema con las tarjetas internas o un dispositivo de control externo.

Con base en los tipos de plantas presentados en las Fig. 1 y 2, y teniendo en cuenta las características descritas en el diagrama de bloques de la Fig. 3, el modelo en 3D para la planta de corte al vuelo propuesta se presenta en la Fig. 4.

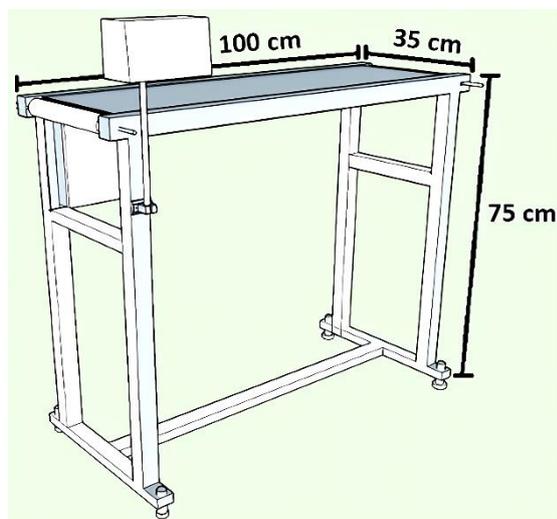


Fig. 4. Prototipo de planta de corte al vuelo propuesta.

Fuente: los autores.

C. Sistema electrónico

El diseño de los circuitos electrónicos se realizó pensando en la escalabilidad de la máquina y la interconexión eficiente entre las diferentes tarjetas como se muestra en los diagramas de bloques de las Fig. 5 y 6, donde se describe la conexión general del sistema destacando tres grupos: el sistema mecánico, donde se encuentran finales de carrera, encoders, motores y servo, tablero de potencia en el cual se encuentran los drivers y acondicionamientos necesarios para el manejo de sensores, actuadores y kit de desarrollo LAUNCHXL-F28377S de Texas Instruments. Por último, el tablero de control donde el usuario interactúa con la planta de corte al vuelo.



Fig. 5. Diagrama general de conexiones.

Fuente: los autores.



Fig. 6. Diagrama interno de conexiones para los módulos de la Fig. 5.

Fuente: los autores.

D. Construcción de tableros

Para la protección de circuitos electrónicos, cableado de conexiones y fuente de alimentación, se diseñaron dos tableros que permitieron organizar y gestionar los diferentes sensores y actuadores del sistema. En la Fig. 7, se presenta el diagrama de bloques del tablero de potencia, en el cual se concentran las conexiones de los actuadores y sensores provenientes del sistema mecánico. La alimentación de todo el sistema está a cargo de una fuente de poder conmutada con salidas de 5 y 12 voltios (v) que llegan al circuito de potencia directamente y de allí se reparten a los demás dispositivos.

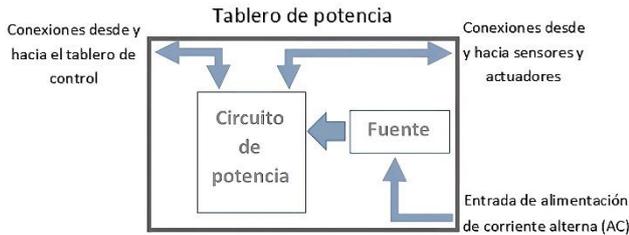


Fig. 7. Diagrama de bloques del tablero de potencia.
Fuente: los autores.

En la Fig. 8 se presenta el diagrama de bloques del tablero de control, en el cual se tienen las conexiones provenientes del tablero de potencia y conexión de dispositivos que hacen parte de la interfaz gráfica de usuario.

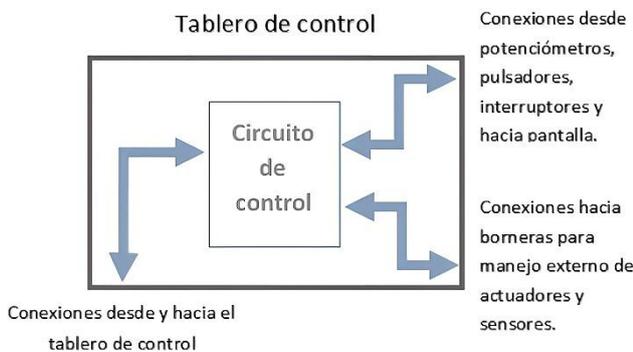


Fig. 8. Diagrama de bloques del tablero de control.
Fuente: los autores

E. Entorno y estructura de programación

Para la integración de todas las funcionalidades del sistema se requiere de programación en cada una de las tarjetas de desarrollo utilizadas. En el caso del kit de desarrollo LAUNCHXL-F28377S, hay varias opciones para programar sus funcionalidades. Por un lado, se encuentra el software Energia como entorno de desarrollo integrado (IDE) de código abierto basado en el marco de referencia Arduino que como se muestra en [12], facilita la programación de las tarjetas y kits de Texas Instruments. Otra herramienta es el Code Composer Studio (CCS) que, al ser desarrollado por Texas Instruments, tiene soporte para todos los kits de desarrollo que fabrican.

CCS cuenta con las herramientas necesarias para la programación de microcontroladores y su interfaz gráfica de usuario se presenta en la Fig. 9.

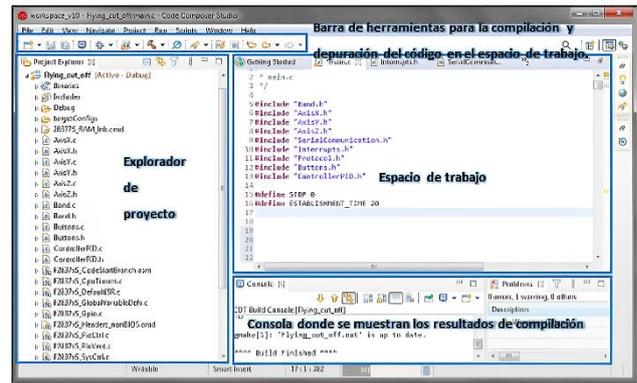


Fig. 9. Interfaz gráfica de usuario del CCS.
Fuente: los autores.

Para el desarrollo de los algoritmos necesarios para el funcionamiento y control de la planta de corte al vuelo, se definió una estructura de programación en el CCS. El lenguaje de programación utilizado fue C y en la Fig. 10 se presenta la estructura de programación definida.

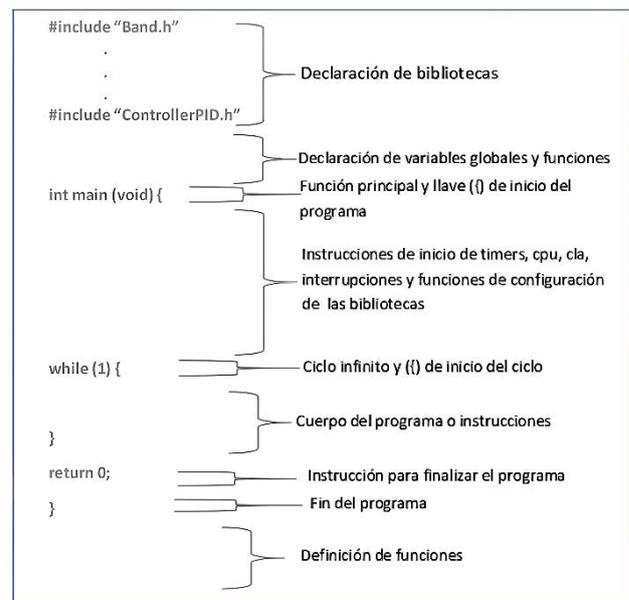


Fig. 10. Estructura de programación definida para el CCS.
Fuente: los autores.

F. Estrategia de control

El sistema de control de la planta busca realizar un trazo perpendicular sobre la cinta transportadora mientras esta se encuentre en movimiento, si la velocidad de la cinta varía el sistema de corte debe variar e igualar su velocidad, de lo contrario el trazo sería irregular. Parte fundamental de la estrategia de control es garantizar que ambas velocidades (eje X y banda transportadora) sean equivalentes y el trazado quede formando un ángulo de 90° con respecto a la horizontal o eje X. Por lo anterior, y

como prueba de funcionamiento se propone un controlador PI, el cual se puede implementar en diferentes formas de acuerdo a lo establecido en [14], [15] y [16]. En el controlador que se realizó se usó el lazo de control presentado en la Fig. 11.

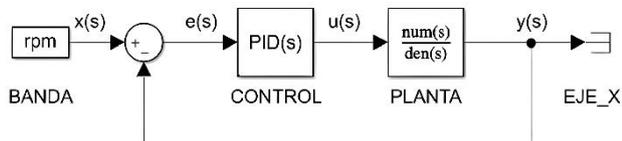


Fig. 11. Lazo de control implementado.
Fuente: los autores

III. PRUEBAS Y RESULTADOS

A. Estructura de la banda transportadora

La estructura de la banda se realizó con tubo de hierro rectangular de 4×2 cm con calibre 0.8 mm. Otro aspecto importante en la banda es el sistema de tracción, el cual se construyó a partir de una relación de poleas de sincronismo donde la unión entre ambas se dio por medio de una correa 2GT 6 mm. En la Fig. 12 se presenta la estructura de la banda transportadora y el sistema de poleas desarrollado para la planta de corte al vuelo.



Fig. 12. Estructura de banda transportadora.
Fuente: los autores

Luego de tener la estructura de la banda transportadora, se eligió la cinta de la misma, la cual se construyó en lamina microporosa con calibre de 2 mm recubierto en una de las caras con vinilo adhesivo blanco. Al trabajar con este tipo de material no se podía dejar la cinta transportadora con broche, pues con el trabajo continuo tendría ruptura en los puntos de unión, por lo cual fue necesario construirla cerrada. En la Fig. 13 se presenta la cinta transportadora puesta sobre la estructura.

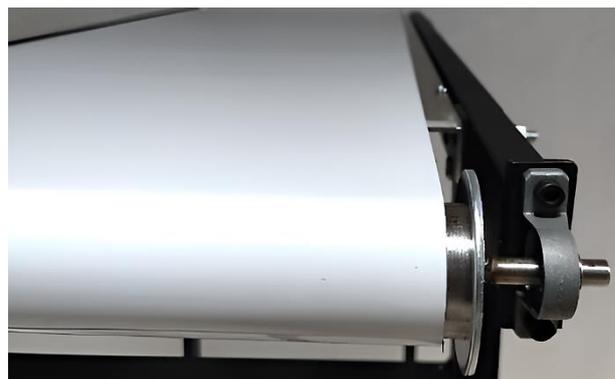


Fig. 13. Cinta transportadora puesta en la estructura.
Fuente: los autores

B. Sistema de corte

La validación del funcionamiento de la planta se comprueba, entre otras cosas, si ésta realiza un trazado correcto, es decir, uniforme y sin zigzagueo; para lo cual el conjunto de piezas que componen el sistema de corte debe estar correctamente acomodadas. Por tal motivo, se concibió un sistema de coordenadas X, Y y Z que generan el movimiento a partir de los actuadores como el mostrado en la Fig. 14.

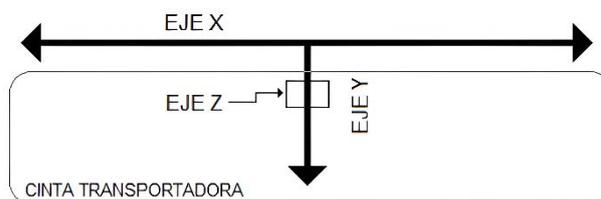


Fig. 14. Coordenadas definidas para el sistema de corte.
Fuente: los autores

C. Circuito de potencia

El circuito de potencia presentado en la Fig. 15 usa un L298N para los motorreductores de la cinta en la banda transportadora, control del sistema de corte, pines para conectar el kit de desarrollo LAUNCHXL-F28377S [17] y el circuito de conmutación. Este circuito también contiene borneras, conectores molex, USB y RJ-45 para la interconexión con otros componentes de la planta de corte al vuelo como la fuente de alimentación, sensores, actuadores, circuito de control, interfaz de programación, entre otros. Otro complemento del circuito de potencia es presentado en la Fig. 16, donde se observa el sistema de aislamiento por optoacoplador entre los pines digitales del kit de desarrollo de Texas Instruments y a su vez sirve como regulador de voltaje entre 3.3 v a 5 v y viceversa.

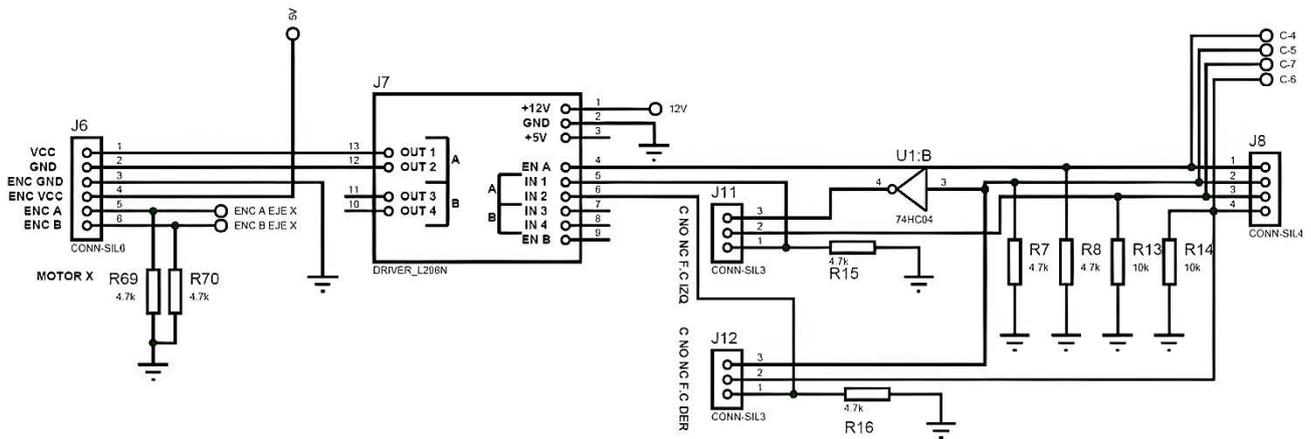


Fig. 15. Circuito de potencia para la planta de corte al vuelo.

Fuente: los autores

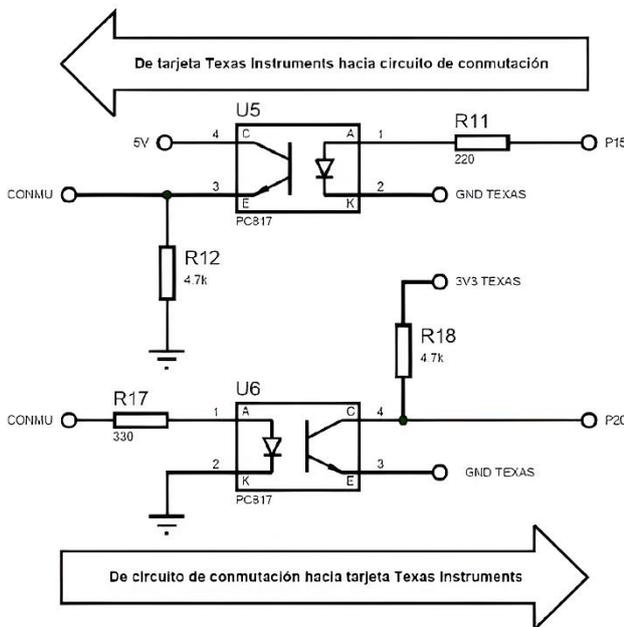


Fig. 16. Circuito optoacoplador para entradas y salidas digitales.

Fuente: los autores

D. Circuito de control

El circuito de control sirve de puente entre la interfaz de usuario y la tarjeta de potencia. Para tal fin, este circuito permite la conexión de tres puertos RJ-45 mediante los cuales se intercambian señales para el manejo de sensores, actuadores, comunicación serial, elementos de propósito general y switch de cambio de modo; este último para definir el modo de funcionamiento de la máquina. Un Arduino Nano es el encargado de la visualización de la información del menú por LCD y del

procesamiento de las señales de interacción con el usuario y el envío de sus respectivos valores vía serial a la tarjeta principal del circuito de potencia. En la Fig. 17 se presenta el circuito de control desarrollado.

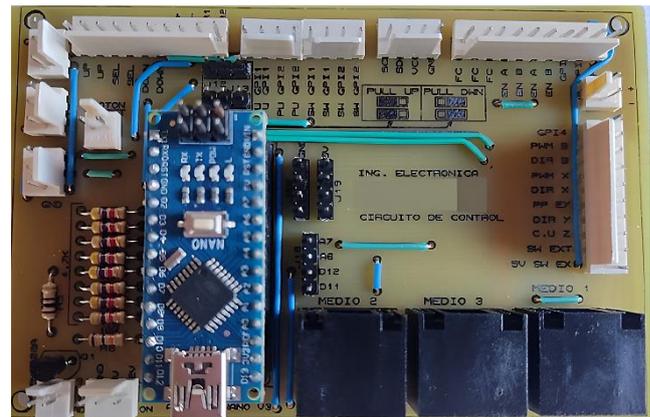


Fig. 17. Circuito de control para la interfaz de usuario y tablero de potencia.

Fuente: los autores

E. Tableros de potencia y control

Luego de baquelar los circuitos de potencia, control y conmutación presentados en las Fig. 15, 16 y 17, se realizó la implementación de los tableros de potencia y control a través de los cuales se opera la planta de corte al vuelo tal y como se observa en las Fig. 18 y 19.



Fig. 18. Vista interna de los tableros de control y potencia para la planta de corte al vuelo.

Fuente: los autores

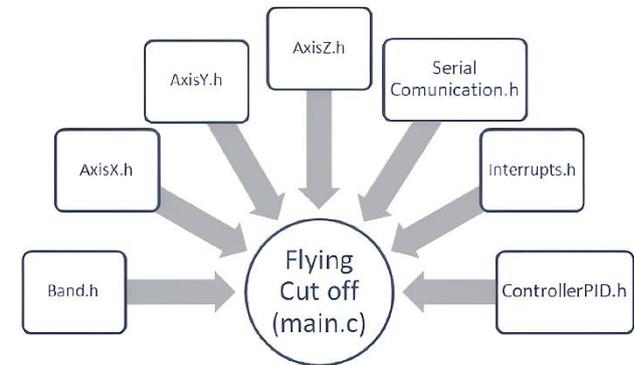


Fig. 20. Librerías desarrolladas en CCS para la programación del microcontrolador C2000.

Fuente: los autores



Fig. 19. Vista externa de los tableros de control y potencia para la planta de corte al vuelo.

Fuente: los autores

F. Programación del sistema de control

Como el desarrollo de software en este caso se realiza para un sistema embebido, se creó un archivo general a través del cual se compilan diferentes archivos que contienen las funciones necesarias para el manejo de sensores y actuadores de la planta de corte al vuelo. En la Fig. 20 se muestran las bibliotecas que se desarrollaron en CCS para cargar y ejecutar en el microcontrolador C2000 del kit de desarrollo LAUNCHXL-F28377S de Texas Instruments, para manejar cada uno de los dispositivos electrónicos de la planta. Por facilidad y buenas

prácticas de programación, se desarrollaron librerías para cada objeto del sistema, en cada una de ellas existe la configuración requerida para el manejo del actuador o sensor respectivo. La programación realizada en CCS para obtener la curva de reacción del actuador en el eje X, consistió en configurar y aplicar una señal PWM con ciclo útil de 0 % al 100% y realizar la lectura de su encoder. Posteriormente, los datos de la señal PWM generada y los datos leídos del encoder fueron enviados a MATLAB desde el puerto serial para realizar modelamiento del sistema. Los datos generados tuvieron una respuesta como la que se observa en Fig. 21.

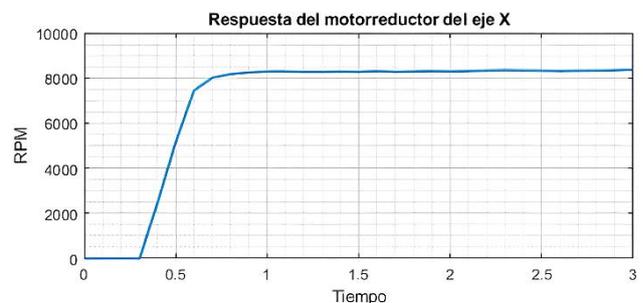


Fig. 21. Respuesta transitoria del motorreductor del eje X.

Fuente: los autores

Con base a la información presentada en la Fig. 21, se obtuvo un modelo matemático

aproximado para el sistema representado por (1).

$$G(s) = \frac{8368.1}{0.182s + 1} \quad (1)$$

Posteriormente, se uso el PID Tuner de MATLAB para realizar la sintonización del controlador PI para el sistema descrito en (1), obteniendo como parámetros para el controlador los presentados en la Fig. 22.

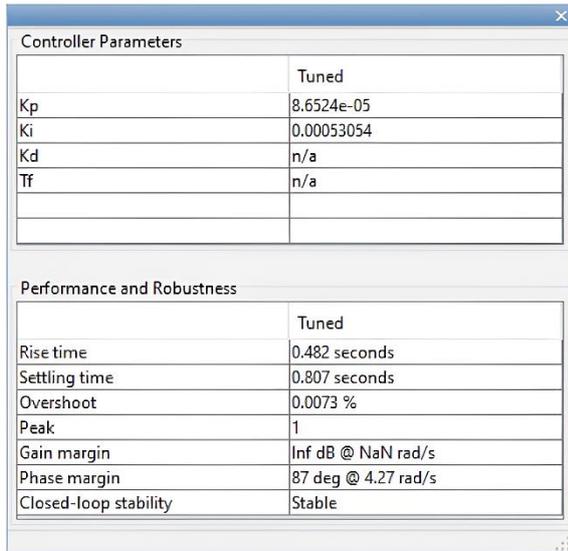


Fig. 22. Parámetros obtenidos para el controlador PI.
Fuente: los autores

Al implementar el controlador PI en el lazo de control presentado en la Fig. 11 se obtuvo la respuesta presentada en la Fig. 23.

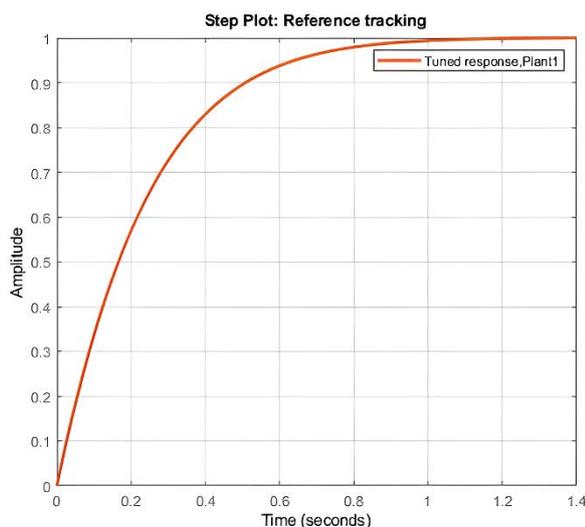


Fig. 23. Respuesta del sistema controlado.
Fuente: los autores

Como se puede apreciar en la Fig. 23, el

controlador PI diseñado funciona de manera apropiada, puesto que sigue la referencia establecida. En este punto es importante recordar que los valores de las variables están normalizados y que una amplitud de 1 corresponde a una señal PWM con ciclo útil de 100% (valor máximo).

Discretizando el sistema con un tiempo de muestreo $T_m = 0.1 \text{ seg}$, se obtuvieron las ecuaciones en diferencia para el controlador digital presentadas en la Fig. 24.

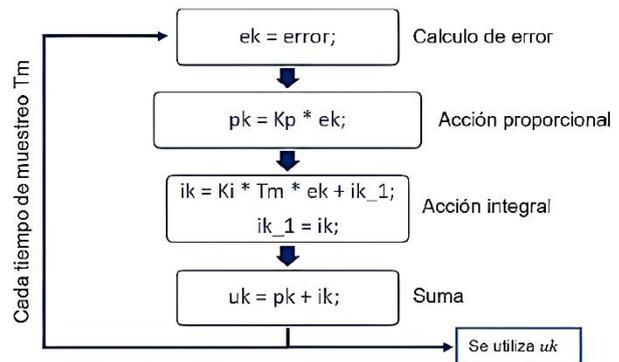


Fig. 24. Representación del controlador PI mediante ecuaciones en diferencia.

Fuente: los autores

Luego de implementar el controlador PI en el kit de desarrollo LAUNCHXL-F28377S de Texas Instruments, se aplicaron señales PWM con ciclo útil del 100% a los motores de la banda transportadora y del eje X para validar su respuesta y la velocidad máxima alcanzada. En la Fig. 25 se presentan las respuestas para cada uno de los motores usando una caja reductora con relación 100: 1.

Es importante destacar que el movimiento del mecanismo en el eje X debe tener la misma velocidad que la banda transportadora para que se puedan realizar los cortes de manera regular, por lo que en este caso se debe introducir un factor de compensación en el lazo de control del microcontrolador, a fin de garantizar que el mecanismo del eje X siga la referencia establecida por la banda transportadora.

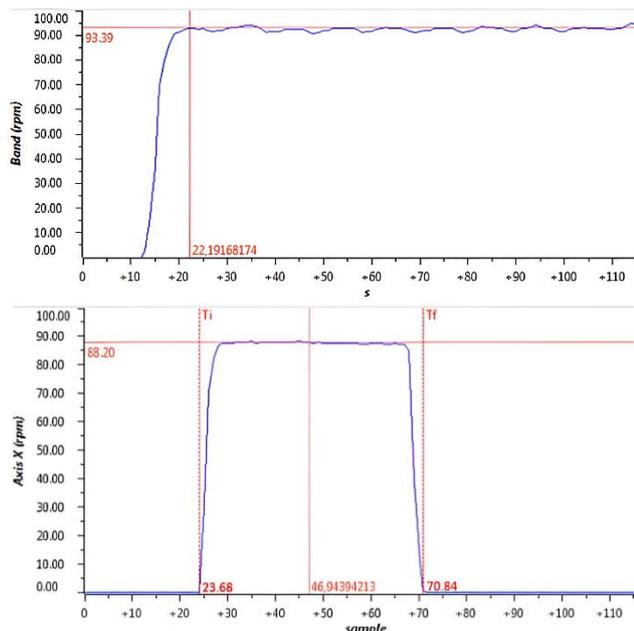


Fig. 25. Velocidad máxima alcanzada por los motores. Parte superior motor de la banda transportadora y parte inferior mecanismo del eje X.

Fuente: los autores

Una vez ajustada la velocidad de los motores se realizó prueba de validación y funcionamiento de la planta de corte al vuelo en un proceso de corte. En la Fig. 26 se presenta la respuesta del sistema. A los 6 segundos el sistema de corte se encuentra en la posición de inicio. En 6.8 segundos se establece un ciclo útil del 50% en la banda y transcurren 1.8 segundos para que la velocidad de ésta se establezca en 6330 rpm aproximadamente. El eje X a su vez detecta una variación en el set-point debido al movimiento de la banda y al instante se produce un esfuerzo de control para activar el motorreductor.

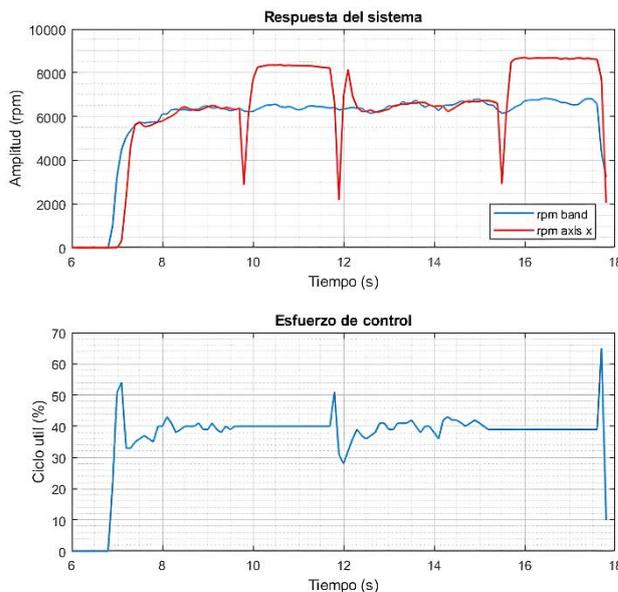


Fig. 26. Respuesta de la planta durante un proceso de corte. Parte superior azul banda transportadora, parte superior roja mecanismo del eje X y parte inferior esfuerzo de control.

Fuente: los autores

IV. CONCLUSIONES

El controlador PI implementado demostró eficiencia al mantener una sincronización precisa entre la velocidad de la banda transportadora y el mecanismo de corte. Adicionalmente, los resultados experimentales mostraron que el sistema puede ajustar su operación para realizar cortes consistentes, reduciendo significativamente los errores de corte en pro de un mejor producto final.

El uso del kit de desarrollo LAUNCHXL-F28377S de Texas Instruments facilitó la implementación de un sistema de control en tiempo real, aprovechando las capacidades avanzadas de los microcontroladores C2000. El uso de dispositivos como este un prototipo contribuyen a un sistema robusto que no solo es replicable, sino también escalable en la emulación de diferentes aplicaciones industriales.

La planta de corte al vuelo desarrollada ofrece un recurso valioso para la formación académica en el área de automatización y control del programa de Ingeniería Electrónica de la Universidad del Quindío, ya que permite a los estudiantes experimentar y validar conceptos teóricos mediante prácticas reales, fomentando un aprendizaje profundo y aplicado. Asimismo, el proyecto abre las puertas a nuevas líneas de

investigación aplicada, integrando necesidades industriales en el entorno académico y fortaleciendo los procesos entre la universidad y la industria para el desarrollo de soluciones tecnológicas avanzadas.

El diseño del prototipo consideró aspectos de seguridad y funcionalidad, asegurando que las prácticas educativas se realicen en un entorno controlado y seguro. La implementación de sensores y actuadores, junto con un sistema de monitoreo en tiempo real, garantiza la operatividad segura del sistema, que además de modular y compacto podría ser controlado a través de aplicaciones IoT por ejemplo.

REFERENCIAS

- [1] A. P. Lorandi Medina, G. Hermida Saba, J. Hernandez Silva, and E. L. De Guevara Durán, "Los Laboratorios Virtuales y Laboratorios Remotos en la Enseñanza de la Ingeniería," *Rev. Int. Educ. En Ing.*, vol. 4, pp. 24–30, 2011.
- [2] J. G. Castro Lugo, J. J. Padilla Ybarra, and E. Romero A., "Metodología para realizar una automatización utilizando PLC," *Rev. Ing. Eléctrica Electrónica Comput.*, vol. 1, pp. 18–21, Dec. 2005.
- [3] N. Peric and I. Petrovic, "Flying shear control system," *IEEE Trans. Ind. Appl.*, vol. 26, no. 6, pp. 1049–1056, Dec. 1990, doi: 10.1109/28.62386.
- [4] J. C. Song, C. Z. Wang, and D. Xu, "Dynamic Simulation and Control Strategy of Centrifugal Flying Shear," *Appl. Mech. Mater.*, vol. 16–19, pp. 278–282, Oct. 2009, doi: 10.4028/www.scientific.net/AMM.16-19.278.
- [5] G. Ng, S. Deleanu, J.-P. Prevost, and D. Carpenter, "Improving the operation of a flying dividing shear by using Direct Torque Control," in *2017 International Conference on Modern Power Systems (MPS)*, Cluj-Napoca, Romania: IEEE, Jun. 2017, pp. 1–8. doi: 10.1109/MPS.2017.7974419.
- [6] N. Madhivanan and K. Narayanan, "FPGA implementation of general purpose real time controller-applied to flying shears in hot rolling mills," in *2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET)*, Nagercoil, Tamil Nadu, India: IEEE, Mar. 2012, pp. 521–525. doi: 10.1109/ICCEET.2012.6203844.
- [7] S. Das *et al.*, "Metallurgical Investigation of Flying Shear Blades at Hot Rolling Rebar Mills," *J. Fail. Anal. Prev.*, vol. 23, no. 5, pp. 1851–1859, Oct. 2023, doi: 10.1007/s11668-023-01737-1.
- [8] A. Serrano Huertas and O. A. López Madrid, "Diseño de una cizalla industrial para el corte de láminas de acero al silicio utilizadas en transformadores secos de energía eléctrica.," Universidad Tecnológica de Pereira, Facultad de Tecnologías, 2015.
- [9] P. E. Ravichagua Fuentes, "Modernización del sistema de control de la cizalla volante del proceso de laminación de la planta siderúrgica de Aceros Arequipa."
- [10] T. Chen, Y. P. Hu, and J. P. Zhu, "Solid Modeling and Optimization Design of Crank Rocking Flying Shear," *Appl. Mech. Mater.*, vol. 608–609, pp. 7–13, Oct. 2014, doi: 10.4028/www.scientific.net/AMM.608-609.7.
- [11] E. Caicedo Bravo, E. B. Bacca, B. A. Calvache, J. E. Cardona Aristizabal, and J. A. Buitrago, "Laboratorio distribuido con acceso remoto para la enseñanza de la robótica," *Rev. Educ. En Ing.*, vol. 7, pp. 51–61, Jun. 2009.
- [12] D. F. Ramirez-Jimenez, A. L. Parrado, and J. V. Medina, "Overview of a framework for Implementation of digital controllers in Energia IDE using Texas Instruments microcontrollers," in *2021 IEEE 5th Colombian Conference on Automatic Control (CCAC)*, Ibagué, Colombia: IEEE, Oct. 2021, pp. 13–18. doi: 10.1109/CCAC51819.2021.9633305.
- [13] "Productos," RISHBIN WUXI CO., LTD. Accessed: Jul. 05, 2024. [Online]. Available: <http://es.rishbin.com>
- [14] J. Hoyos-Gutiérrez, J. Cardona-Aristizabal, P. Muñoz-Gutiérrez, and D. Ramirez-Jimenez, "A Systematic Literature Review on Rapid Control Prototyping Applications," *IEEE Rev. Iberoam. Tecnol. Aprendiz.*, vol. 18, no. 1, pp. 76–85, Feb. 2023, doi: 10.1109/RITA.2023.3250559.
- [15] D. F. Ramirez Jiménez, "Sistema de medición y control de temperatura para un prototipo de planta de tratamiento de aguas residuales," *Investig. E Innov. En Ing.*, vol. 9, no. 1, pp. 100–113, Apr. 2021, doi: 10.17081/invinno.9.1.4305.
- [16] D. F. Ramirez-Jiménez, P. A. Bedoya-Benítez, and P. A. Munoz-Gutierrez, "Visualization and Control System for a Wastewater Laboratory Plant with Biological Treatment," *Cienc. E Ing. Neogranadina*, vol. 34, no. 1, Art. no. 1, Jun. 2024, doi: 10.18359/rcin.7270.
- [17] T. Instruments, "LAUNCHXL-F28377S Overview User's Guide." Texas Instruments, Jun. 2015. [Online]. Available: https://www.ti.com/lit/ug/sprui25d/sprui25d.pdf?ts=1720282454678&ref_url=https%253A%252F%252Fwww.google.com%252F

Fortalecimiento de habilidades de programación en ingeniería electrónica mediante sistemas microcontrolados

Ramiro Alejandro Plazas Rosas

Facultad de Ingeniería

Institución Universitaria Antonio José Camacho

Santiago de Cali, Colombia

ORCID: 0000-0001-6595-3839

Abstract—Este estudio pretende demostrar cómo la utilización de herramientas tecnológicas, como los sistemas microcontrolados, permite a los estudiantes de ingeniería electrónica en los primeros semestres consolidar los conceptos de programación. Entre estos conceptos se incluyen condicionales, definición de variables, uso de funciones y apropiación de librerías, entre otros. El desarrollo de proyectos por parte de los estudiantes también contribuyó al fortalecimiento de habilidades técnicas en programación, así como al desarrollo de habilidades blandas.

Index Terms—Habilidades, programación, sistemas microcontrolados, proyectos

I. INTRODUCCIÓN

El mundo de la programación requiere habilidades fundamentales que a menudo no son visibles para los nuevos estudiantes. Además, la aparición de herramientas como la inteligencia artificial puede generar cierta aprehensión entre algunos profesores [1]. Sin embargo, estas herramientas pueden ser utilizadas para potenciar las capacidades de los estudiantes, especialmente en el ámbito de la programación [2].

Tradicionalmente, el aprendizaje de la programación se centraba en compilar código como primer paso y luego completar tareas específicas o como lo menciona [3] donde la regulación de esfuerzo implica mantener la atención y el empeño frente a distractores o actividades que no se consideran interesantes por parte de los estudiantes. No obstante, cuando se inicia en la ingeniería electrónica, el estudiante busca una conexión más cercana con las aplicaciones físicas.

Un aspecto que inquieta en la enseñanza de la programación es qué lenguajes deben conocer los nuevos estudiantes, especialmente aquellos de ingeniería electrónica. Lenguajes como C, C++, ensamblador, *Processing*, *Arduino*, *Java*, entre otros, han sido utilizados de forma tradicional para este propósito [4], [5]. Además, durante el desarrollo de su carrera, los estudiantes de ingeniería electrónica también se encontrarán con diferentes lenguajes, como *Matlab*, *LabView* e incluso *Ladder*.

La conceptualización es clave para migrar entre los diversos lenguajes de programación que enfrentará un ingeniero. Algunos conceptos son transversales en varios lenguajes, como la definición de variables y el uso de estructuras condicionales (*IF*, *FOR* y *WHILE*). Asimismo, la habilidad para crear y utilizar funciones personalizadas es fundamental para resolver problemas de manera eficiente.

Sin embargo, se presentan dos características con respecto a los cursos de formación. Los estudiantes de primeros semestres, o aquellos que retoman el ciclo profesional de la ingeniería después de haber dejado sus estudios por algún tiempo, carecen de la capacidad para afianzar los temas vistos en la primera asignatura de programación. Además, el cambio del lenguaje de programación con el que aprendieron, como C o C++, no se articula muy bien con el lenguaje de programación que se aborda actualmente en el curso, como Python. Este diagnóstico académico se realiza en la primera clase de todas las asignaturas de la Institución (UNI-AJC), convirtiéndose así en el primer acercamiento a los conocimientos previos de los estudiantes con respecto a la asignatura a iniciar.

Con el propósito de abordar los diferentes retos de aprendizaje de la programación, además de las características de aprendizaje autónomo, transferencia de conocimiento que ofrece el aula invertida [6], y utilizando la metodológica de aprendizaje basado en proyectos (ABL) como en [7], se plantearon varios proyectos al inicio de semestre a los estudiantes del curso de programación II -Grupo 702- del primer semestre del 2024 del programa de ingeniería electrónica de la *Institución Universitaria Antonio José Camacho*. Los requerimientos generales consistían en:

- Utilizar un sistema microcontrolado.
- Establecer y seleccionar los sensores para cada proyecto.
- Realizar diagrama de flujo, diagrama de bloques, diagrama de conexiones.
- Elaborar manual de usuario e informe final.

Por otra parte, los principales resultados de aprendizajes de la asignatura con el desarrollo del proyecto son:

- Aprender a utilizar sensores.
- Crear interfaces de usuario para monitoreo y control.
- Combinar diferentes tecnologías.
- Implementar sistemas de comunicación por WiFi o Bluetooth.

II. SISTEMAS DESARROLLADOS

Los proyectos planteados para todo el curso de Programación II (Grupo 702) consisten en una estación meteorológica, control de nivel, reconocimiento de voz, control de pesaje, monitoreo de temperatura y sistema de riego. Además, debían realizar la implementación del prototipo, trabajar en equipo y realizar la sustentación grupal e individual del código propuesto para el proyecto. La selección del sistema microcontrolado *ESP32*, corresponde a sus características y la posibilidad de utilizarlo en otras asignaturas de la carrera.

A. Estación meteorológica inteligente con ESP32

Para el desarrollo de este proyecto, se utilizó el lenguaje de programación MicroPython junto con el entorno de desarrollo integrado *Thonny*. El sensor DHT22, de bajo costo y fácil de usar, proporciona mediciones precisas de temperatura y humedad. En la Fig. 1 se muestra la visualización de la información, la cual se realiza mediante la configuración del Servidor Web (*ESP WEBSERVER*). En esta visualización, se presentan los valores de temperatura en grados Celsius y el porcentaje de humedad relativa. Para el manejo del sensor, se empleó la librería *dht*.



Fig. 1. Interfaz de usuario desarrollada en ESP32 Servidor Web

B. Sistema de control de nivel

Este proyecto se enfoca en el control de nivel de un tanque mediante un sistema basado en ESP32 programado en MicroPython. Para medir el nivel del tanque, se utiliza el sensor HC-SR04. El sistema de medición se comunica a través de WiFi con el software Matlab, donde se ha desarrollado la interfaz de usuario. En esta interfaz, se visualizan las alertas de nivel bajo y alto, así como el valor del nivel en tiempo real, como se muestra en la Fig. 2.

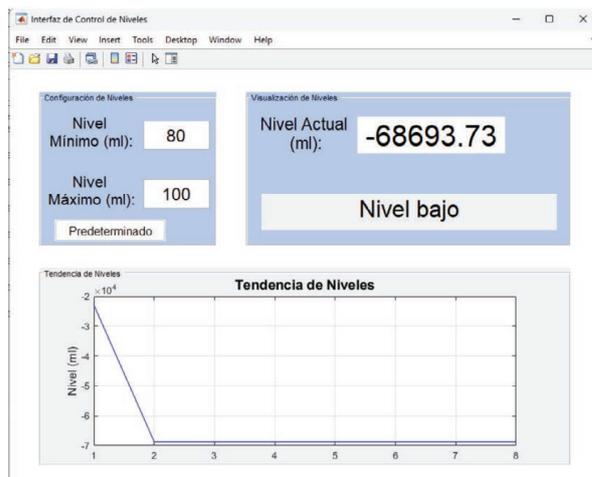


Fig. 2. Interfaz de usuario desarrollada en Matlab

C. Sistema de reconocimiento de voz para control de dispositivos domésticos

En este proyecto, se implementa un sistema de reconocimiento de voz utilizando la plataforma *Microsoft Azure Speech Service* en su versión de prueba. Se aprovechan todos los recursos disponibles, incluida la optimización de voz mediante inteligencia artificial y la conversión precisa, que reduce el ruido externo y mejora la precisión del proceso de conversión. Además, se definen comandos y estados que permiten la activación de diferentes áreas del hogar. Para verificar el funcionamiento, se desarrolló una maqueta con LEDs que indican la activación y un servomotor para abrir una puerta (ver Fig. 3). Asimismo, en la ESP32 se generará un punto de acceso (access point) con una vista HTML para proporcionar las credenciales de conexión, siempre que no sea posible conectarse a Internet.

D. Control de cobro de pesaje

El desarrollo de un sistema de peaje inteligente implica la utilización de sensores para detectar vehículos en la zona de llegada o lugar de pesaje. Además, se incluye una barrera para controlar el paso. En la Fig. 4 se muestra el sistema, que cuenta con sensores de longitud y peso con un error menor al 10%, una pantalla LCD para la visualización de datos y un teclado para operar la máquina. También se han incorporado indicadores sonoros y visuales para alertar al conductor del vehículo o indicarle el siguiente paso. Las librerías se utilizan para el manejo de la pantalla LCD. El valor del peaje se determina en función de la longitud y el peso del vehículo.

E. Monitoreo de la temperatura

En este proyecto, se monitorea la temperatura de una habitación utilizando el sensor de temperatura MAX6675 y se comunica con el sistema ESP32. Los datos del

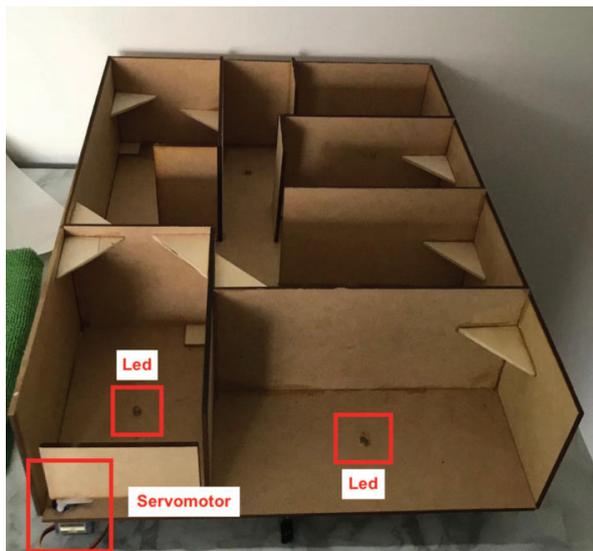


Fig. 3. Prototipo para control de activación por voz

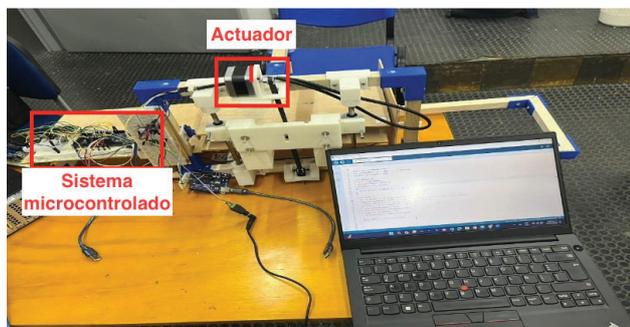


Fig. 4. Prototipo del sistema de pesaje

sensor se envían al panel de control de *ThingSpeak*, lo que permite el monitoreo en tiempo real de la temperatura medida por el sensor, como se muestra en la Fig. 5. Además, se tuvieron en cuenta aspectos como la configuración de la red WiFi en la ESP32, utilizando la librería *WiFiMulti*.



Fig. 5. Visualización de datos de temperatura en ThingSpeak

F. Sistema de riego inteligente

El sistema de riego inteligente desarrollado combina una interfaz de usuario creada en Python utilizando la biblioteca *Tkinter* y un firmware ejecutado en un ESP32. Este sistema está diseñado para monitorear las condiciones ambientales, como la temperatura y la humedad. De hecho, ofrece una solución integral para la gestión eficiente del riego en entornos agrícolas o de jardinería, como se muestra en la Fig. 6. Para la medición de la temperatura, se utilizó el sensor DS18B20 y un sensor de humedad del suelo de bajo costo, junto con una minibomba.



Fig. 6. Prototipo de sistema de riego

III. CONCLUSIONES

Los sistemas desarrollados por todos los estudiantes cumplieron con los objetivos del curso, que buscaban mejorar la apropiación de las diferentes librerías requeridas, los condicionales y la utilización de variables, así como la realización de diagramas de flujo. De hecho, según el diagnóstico académico, los cuales son los datos iniciales de los profesores, realizado el primer día de clase y la realimentación por parte de los estudiantes, se evidenciaron mejoras significativas en los conceptos de definición de variables, condicionales y la utilización de funciones y librerías.

La elección del entorno de desarrollo integrado (IDE) para elaborar el código fue un aspecto interesante, ya que se abordó en clase a través de videotutoriales sobre un IDE específico. Sin embargo, muchos estudiantes optaron por otros entornos de desarrollo, explicando sus elecciones y destacando las ventajas y desventajas de cada uno. Además, las diferentes plataformas y entornos de trabajo utilizados (por ejemplo, MicroPython, Matlab, ThingSpeak) contribuyeron al aprendizaje, según lo expresado por los estudiantes. Durante la sustentación del proyecto, los estudiantes evidenciaron un crecimiento en sus habilidades blandas, como hablar en público, redactar el documento final y trabajar en equipo.

Finalmente, los indicadores de logro de los resultados de aprendizaje planteados fueron:

- Funcionalidad del sistema: El sistema debe cumplir con los objetivos y requisitos especificados, como la



detección de vehículos, el control de acceso, y la transmisión de datos en tiempo real.

- Calidad del código: El código fuente debe ser claro, bien documentado y seguir las mejores prácticas de programación.
- Interfaz de usuario: La interfaz debe ser intuitiva, fácil de usar y permitir el monitoreo y control eficiente del sistema.

AGRADECIMIENTOS

A los estudiantes de programación II, grupo 702 del programa de ingeniería electrónica de la Institución Universitaria Antonio Jose Camacho, por su compromiso y dedicación para el desarrollo de la asignatura. Así mismo, a la Facultad de Ingeniería y al programa de Ingeniería Electrónica.

REFERENCES

- [1] J. M. Llanos Mosquera, C. G. Hidalgo Suarez, and V. A. Bucheli Guerrero, "Una revisión sistemática sobre aula invertida y aprendizaje colaborativo apoyados en inteligencia artificial para el aprendizaje de programación," *Tecnura*, vol. 25, no. 69, pp. 196–214, Jul. 2021.
- [2] O. Revelo, C. Collazos, and J. Jiménez, "El trabajo colaborativo como estrategia didáctica para la enseñanza/aprendizaje de la programación: una revisión sistemática de la literatura," *Tecnológicas*, vol. 21, no. 41, pp. 123–7799, 2017.
- [3] J. J. Ramírez Echeverry, F. A. Olarte Dussan, and A. García Carrillo, "Estrategias de aprendizaje usadas por estudiantes de ingeniería eléctrica e ingeniería electrónica de primer semestre," *Educ. en Ing.*, vol. 9, no. 18, pp. 216–227, 2014.
- [4] F. Ramírez, "Introducción a la programación. Algoritmos y su implementación en Visual Basic.NET, C#, Java y C++." 2007.
- [5] F. Ben and C. Reas, "Processing," 2024. [Online]. Disponible: <https://processing.org/>. [Accessed: 01-Aug-2024].
- [6] N. M. Arellano, J. F. Aguirre, and M. V. Rosas, "Clase invertida: una experiencia en la enseñanza de la programación," en *X Congreso sobre Tecnología en Educación & en Tecnología (TE & ET) (Corrientes, 2015)*, 2015, pp. 538–546.
- [7] R. A. Plazas-Rosas and E. J. Avella-Rodríguez, "Sistema de suministro energético en cercas eléctricas para ganadería utilizando paneles solares," en *Primer Congreso Virtual Los microcontroladores y sus aplicaciones*, 2009, p. 14.

Software-in-the-loop para la implementación de una red vehicular con hardware de código abierto

1st Ing. Sergio J. Ortíz-Hernández

Facultad de Ciencias de la Electrónica
Benemérita Universidad Autónoma de Puebla
Puebla, México
ORCID: 0009-0000-8412-9538

2nd Dr. Luis A. Sánchez-Gaspariano

Facultad de Ciencias de la Electrónica
Benemérita Universidad Autónoma de Puebla
Puebla, México
ORCID: 0000-0002-3899-0746

3rd Dr. José E.M. Gutiérrez-Arias

Facultad de Ciencias de la Electrónica
Benemérita Universidad Autónoma de Puebla
Puebla, México
ORCID: 0000-0002-8224-1099

Resumen—Este artículo presenta el diseño e implementación de un prototipo de red vehicular el cual incluye una interfaz gráfica que emula el cuadro de instrumentos automotriz, así como su integración con algunas unidades de control electrónico (ECUs, por sus siglas en inglés) mediante hardware de código abierto. Además, se presentan los resultados de su aplicación en un entorno educativo y se discuten las ventajas y limitaciones observadas.

Index Terms—CANBUS, Microcontrolador ESP32, Hardware de código abierto, Software-in-the-Loop.

I. Introducción

El desarrollo de redes vehiculares ha evolucionado significativamente en los últimos años, impulsado por la necesidad de mejorar la eficiencia, seguridad y comodidad en el sector automotriz. Estas redes, comúnmente conocidas como CANBus (Controller Area Network Bus), permiten la comunicación eficiente entre los distintos sistemas y componentes de un vehículo [1], [2]. La enseñanza de estas tecnologías en el ámbito académico resulta crucial para la formación de ingenieros y técnicos capacitados que puedan enfrentar los desafíos actuales y futuros de la industria automotriz [3], [4].

En este contexto, presentamos el desarrollo de un prototipo de red vehicular diseñado específicamente como herramienta didáctica. Este prototipo incluye una interfaz gráfica que emula el cuadro de instrumentos de un Kia Sorento 2019 [5], con el objetivo de proporcionar una alternativa económica y accesible a la compra de cuadros de instrumentos automotrices reales, cuyo costo puede ser prohibitivo para muchas instituciones educativas.

El prototipo se basa en unidades de control electrónico (ECU, por sus siglas en inglés) que utilizan microcontro-

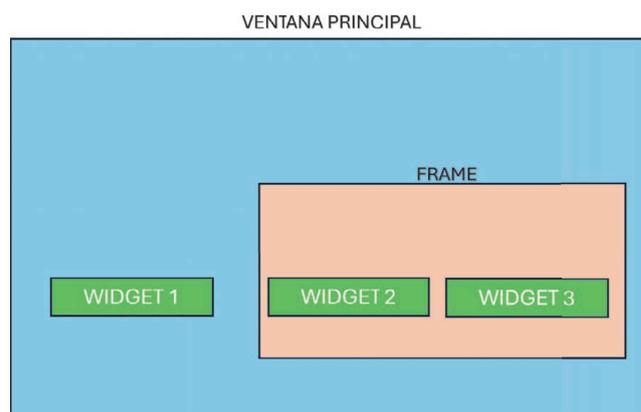


Figura 1. Ventana a generar en Tkinter.

ladores ESP32 y módulos MCP2515 para la comunicación CANBus. La ESP32 es un microcontrolador altamente versátil y potente, equipado con capacidades de conectividad Wi-Fi y Bluetooth, lo que lo convierte en una opción ideal para proyectos de IoT y automatización [6], [7]. Por otro lado, el MCP2515 es un controlador de bus CAN externo [8] que permite a la ESP32 comunicarse con otros dispositivos en la red vehicular [9].

Además, este desarrollo integra métodos de simulación de software in the loop (SIL, por sus siglas en inglés), que son fundamentales para el diseño, prueba y validación de sistemas electrónicos complejos [10]. La simulación SIL se centra en la validación del software de control en condiciones de operación simuladas. Estos métodos no solo reducen costos, sino que también me-



Figura 2. Tarjeta ESP 32 de 38 pines.

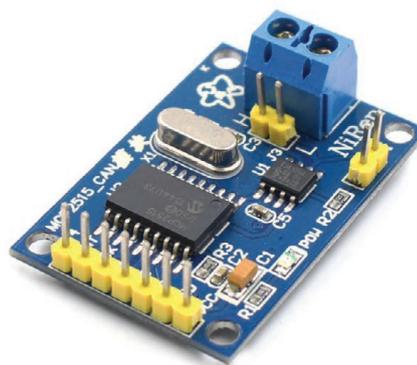


Figura 3. Tarjeta MCP2515.

joran la seguridad y eficiencia del proceso de desarrollo [11].

La implementación de este prototipo educativo no solo facilita la comprensión teórica de las redes vehiculares y sus componentes, sino que también proporciona una plataforma práctica para que los estudiantes experimenten y apliquen sus conocimientos en un entorno controlado y seguro. La interfaz gráfica del cuadro de instrumentos ofrece una representación visual intuitiva y familiar para los estudiantes, mejorando su interacción y comprensión de los datos y señales que se transmiten a través del CANBus.

El trabajo está organizado de la siguiente forma: en la sección dos se presentan los métodos y materiales empleados en la realización del proyecto descrito; posteriormente, en la sección tres se discuten los aspectos más relevantes del prototipo propuesto; finalmente, los resultados, conclusiones y las directrices más importantes del trabajo futuro se presentan en la sección cuatro.

II. Metodos y Materiales

II-A. Tkinter

Tkinter es la librería estándar para la creación de interfaces gráficas de usuario, la cual ya forma parte del entorno de programación de **Python** [12]. Dicha librería está basada en el lenguaje de programación *tc*. El funcionamiento básico de dicha librería, implica la creación de una ventana principal (o *root window*) utilizando la clase **Tk()**. Esta ventana actúa como el contenedor principal para todos los elementos de la Interfaz. A partir de ahí, se pueden agregar *widgets* (como botones, etiquetas, cuadros de texto, etc.) a la ventana principal [13], como se puede observar en la Figura 1.

II-B. ESP 32

La ESP 32 de 38 pines es una tarjeta de desarrollo altamente versátil y poderosa basada en el microcontrolador ESP32 de Espressif Systems (Figura 2). Esta placa cuenta con un procesador dual-core Tensilica LX6

a 240 MHz, Wi-Fi y Bluetooth integrados, lo que la hace ideal para aplicaciones de Internet de las Cosas (IoT), automatización y proyectos de electrónica avanzada [6]. Sus 38 pines proporcionan una amplia variedad de interfaces, incluyendo GPIO, ADC, DAC, SPI, I2C, UART y PWM, facilitando la conexión con sensores, actuadores y otros periféricos. Además, la ESP32 ofrece un amplio soporte de software, compatible con el entorno de desarrollo Arduino IDE y el framework de Espressif, ESP-IDF, permitiendo a los desarrolladores aprovechar al máximo sus capacidades [14].

II-C. MCP2515

La MCP2515 es una tarjeta de controlador de bus CAN (Controller Area Network) diseñada por Microchip Technology. Este controlador se utiliza ampliamente en aplicaciones automotrices e industriales para permitir la comunicación eficiente entre diferentes dispositivos en un sistema. La MCP2515 integra un controlador CAN completo, compatible con el estándar CAN 2.0B, y proporciona una interfaz SPI para la comunicación con microcontroladores. Entre sus características destacadas se encuentran la capacidad de operar a velocidades de hasta 1 Mb/s, los filtros de aceptación para mensajes recibidos, y los buffers de transmisión y recepción que mejoran la eficiencia del manejo de datos.

La MCP2515 es especialmente valorada por su capacidad para reducir la carga de trabajo del microcontrolador principal al gestionar el tráfico de mensajes CAN, permitiendo así un rendimiento más eficiente del sistema en aplicaciones críticas. Este módulo, mostrado en la Figura 3, es esencial en el desarrollo de aplicaciones como sistemas de control de motores, sistemas de diagnóstico a bordo (OBD), y otras implementaciones automotrices e industriales donde la comunicación robusta y confiable es crucial [15].

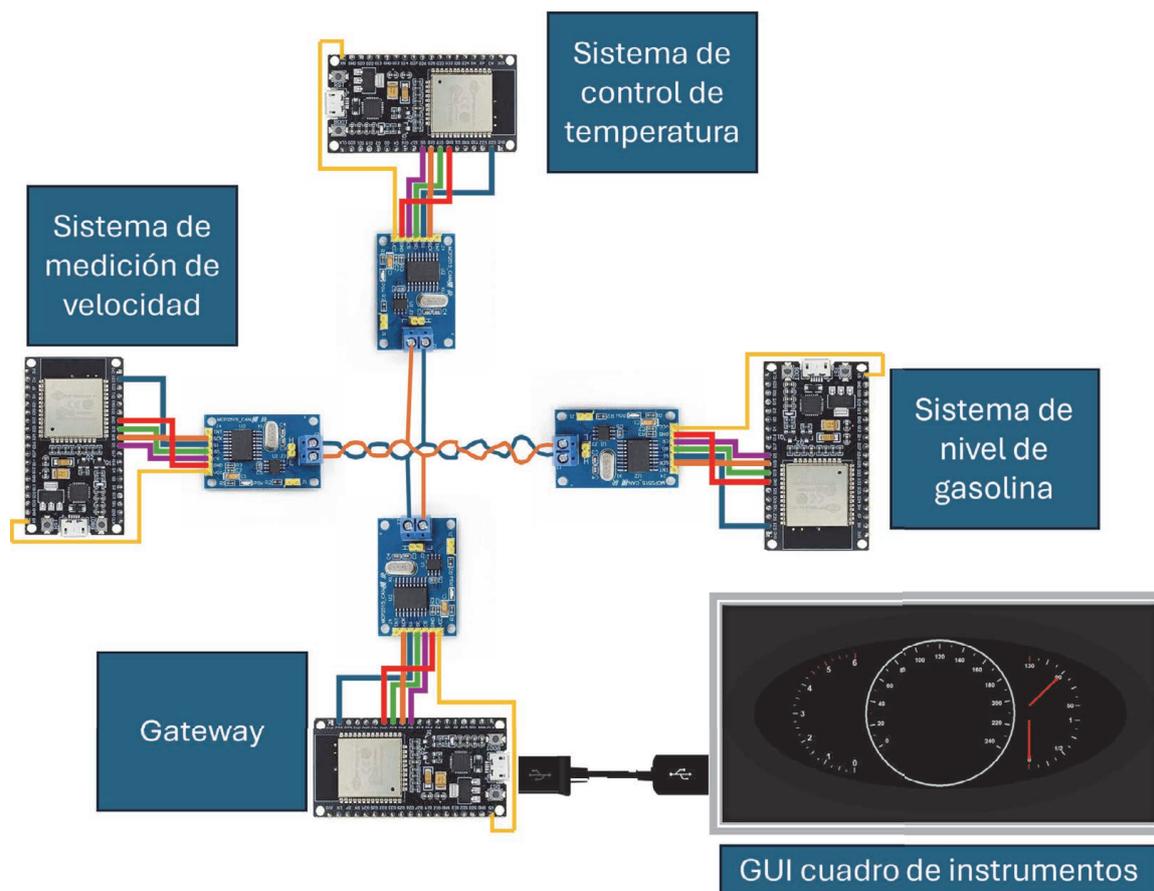


Figura 4. Esquema del Prototipo Propuesto.

II-D. Diseño del prototipo

El esquema del prototipo propuesto se muestra en la Figura 4. Como puede apreciarse, consta de 4 módulos de control; los cuales están conformados por una unidad de control electrónico (ECU) programada para alguna tarea en específico. Estas ECUs se comunican mediante el protocolo CANBus. Las ECUs emuladas para el desarrollo de la red son:

- Sistema de nivel de gasolina.
- Sistema de control de temperatura.
- Sistema de medición de velocidad.
- Gateway.

Cabe mencionar que es posible agregar otros módulos para abarcar más sistemas que se pueden encontrar dentro de la red de un vehículo real. El límite dependerá de los dispositivos empleados para la construcción de la red, pero en teoría el protocolo CANBus ha sido desarrollado para conectar en red poco más de 100 nodos [1].

Para este artículo, se trabajó únicamente con el nodo del gateway. Enfocándonos en realizar la interfaz gráfica del cuadro de instrumentos mediante **TKinter** de **Python**. De igual manera, realizamos la conexión entre

la tarjeta ESP 32 y la interfaz mediante comunicación serial. Por último, en este trabajo se muestra la visualización de los datos provenientes del gateway en una página web de una red local. Este primer nodo nos ayudara a comprender de mejor manera como podemos implementar la tarjeta ESP 32 como una unidad de control electrónico.

III. Implementación

III-A. Software

Para reducir el costo que implica comprar un cuadro de instrumentos automotriz, se optó por realizar la interfaz gráfica del mismo. Se utilizó la herramienta de **Tkinter**, la cual se menciona con anterioridad. La interfaz del cuadro de instrumentos, se basa en el que se encuentra dentro de un Kia Sorento del año 2019, como se observa en la Figura 5. Para poder realizar dicha interfaz mediante la librería ya mencionada, se ocupó la clase conocida como **canvas**; la cual permite colocar diferentes elementos dentro de una ventana.

Mediante esta clase es posible colocar mediante coordenadas cada uno de los elementos a implementar dentro de la interfaz que se desarrolló. Gracias a esto,



Figura 7. Captura de pantalla desde una laptop donde se observan los datos de monitoreo del gateway implementado.

instrumentos con los tres sistemas de medición de tipo aguja con escala de medición en despliegue esférico, similar a los sistemas de medición analógicos basados en galvanómetros, tal y como se encuentran en el cuadro de instrumentos físico del Kia Sorento.

Por otro lado, para que nuestra interfaz pueda leer los datos enviados por el puerto serial, se requiere instalar la librería **pyserial**. Posteriormente, se configura la conexión de la ESP 32 con nuestra interfaz con el siguiente código:

```
#Configuracion del puerto serial
esp32 = serial.Serial('COM11',115200)
```

Hay que considerar que dependiendo del equipo de cómputo que se esté ocupando hay que cambiar el puerto COM que ocupe dicho equipo, de igual manera debemos configurar la velocidad a la que se transmite la información. En este caso, la tasa de transmisión está dada en baudios. En las pruebas realizadas se transmiten 3 datos diferentes los cuales se envían en una sola línea, separados por una coma. Para poder leer estos datos y posteriormente usarlos para cambiar la posición de las flechas, se creó una función la cual se puede observar en el siguiente código:

```
def actualizar_tablero():
    try:
        # Leer datos del puerto serial
        data = esp32.readline().decode().strip()
        velocidad, nivel, temp = data.split(',')

        # Actualizar los elementos del tablero
        modifica_texto(velocidad)
        modifica_nivel(nivel)
        modifica_temperatura(temp)
    except Exception as e:
        print(f"Error al leer del puerto serial: {e}")

# Llamar a esta función de nuevo después de 100 ms
cluster.after(10, actualizar_tablero)
```

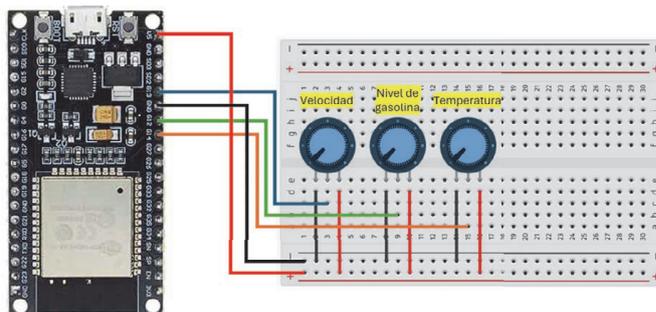


Figura 8. Diagrama de conexión de prueba de comunicación de módulo gateway con interfaz de cuadro de instrumentos.

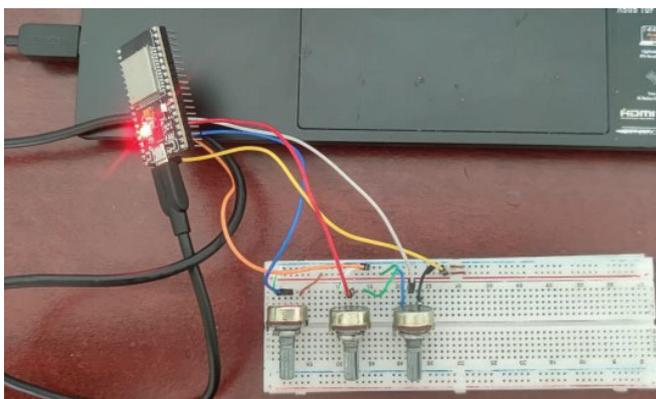


Figura 9. Conexión física para prueba de comunicación de módulo gateway con interfaz de cuadro de instrumentos.

Mediante **data.split** separamos por como los datos recibidos y les asigna una variable la cual se usó para mandar a otras funciones las cuales nos ayudan a cambiar la posición de las flechas de cada válvula, como se observa en el siguiente código:

```
def modifica_nivel(nivel):
    niv = float(nivel)
    gasolina = radians(180 - (0.9*niv))
    contenedor.delete('dato')
```

Además de la interfaz gráfica, se generó una página web mediante código html a través del programa de Arduino IDE, con el cual se programa la tarjeta ESP32 para leer los datos de los potenciómetros y posteriormente generar una dirección IP la cual contiene la página web en donde se pueden visualizar dichos datos. Cabe mencionar que como únicamente trabajaremos con una red local, solo dispositivos conectados a la misma pueden observar la página diseñada. En la figura 7 se muestran los datos de los 3 potenciómetros y algunas alertas que aparecen dependiendo del dato que se muestre en nivel de gasolina y temperatura; por ejemplo, si el nivel de gasolina es menor al 10%, aparece el anuncio **falta de gasolina**.

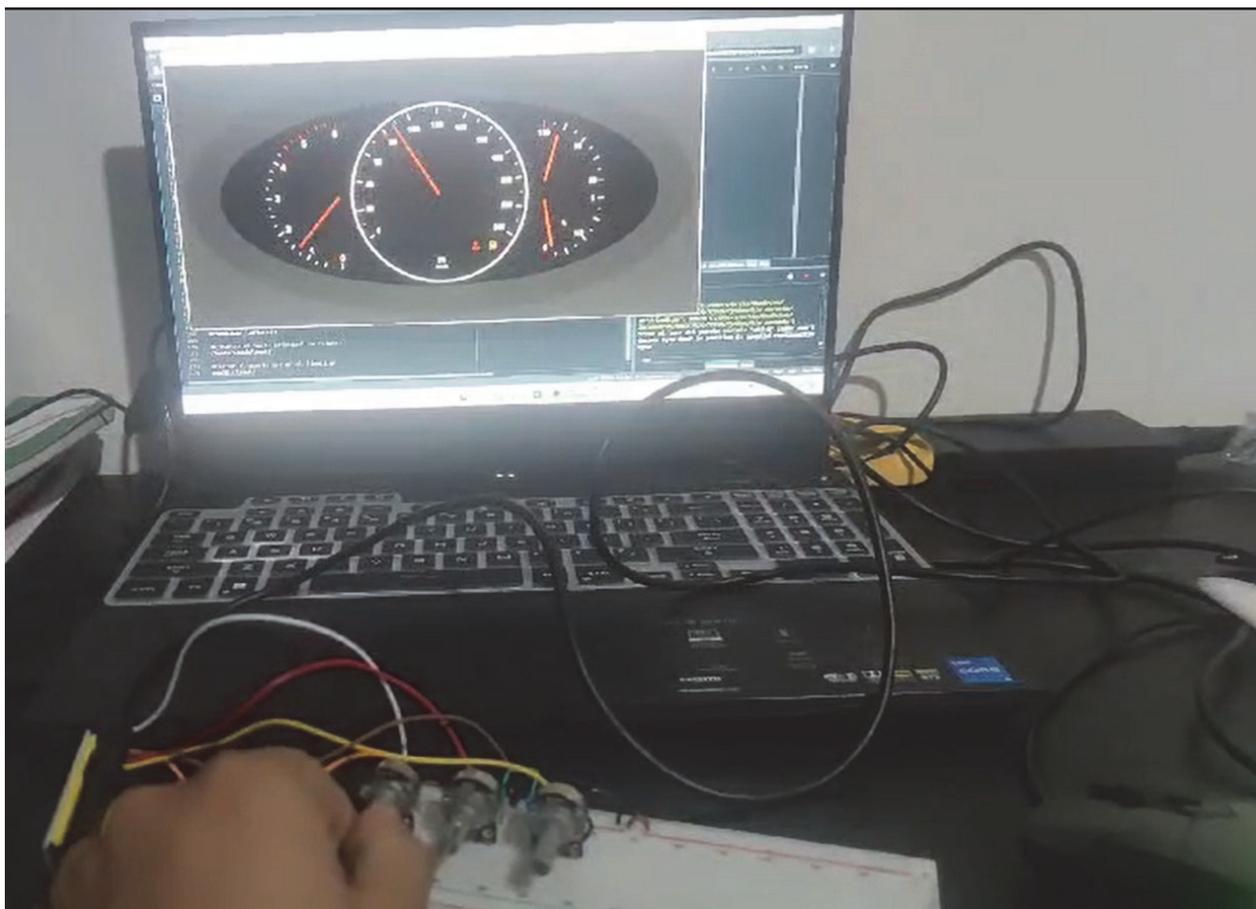


Figura 10. Diagrama de conexión de prueba de comunicación de módulo *gateway* con interfaz de cuadro de instrumentos.

III-B. Hardware

La primera prueba para implementar la parte física de nuestro prototipo fue corroborar la comunicación serial entre la ESP 32 y la interfaz creada. Para ello, se conectaron 3 potenciómetros a la tarjeta y mediante la comunicación del puerto serial del ordenador se mandaban los valores obtenidos de los potenciómetros en tiempo real para su correspondiente despliegue en la interfaz. El diagrama de conexión se puede observar en la Figura 8, donde se observa a que pines de la tarjeta están conectados cada uno de los elementos. A continuación se listan los pines involucrados y su conexión física:

- 5V - Positivo de los potenciómetros
- GND - Tierra de los potenciómetros
- G13 - Salida potenciómetro de velocidad
- G12 - salida potenciómetro de nivel de gasolina
- G14 - Salida potenciómetro de Temperatura

En la Figura 9, se puede observar la conexión física del Esp 32 y los 3 potenciómetros. Por otra parte, en la Figura 10 se muestra la interacción directa entre el microcontrolador conectado por una parte a los tres

sensores resistivos y por otra al equipo de cómputo a través de la comunicación serial provista por el puerto USB de la laptop.

Otro elemento que se ocupó de la tarjeta ESP 32 y por el cual se eligió para el desarrollo de nuestro proyecto, es su módulo WiFi, con el que nos conectamos a la red local, en donde se realizaron las pruebas. El módulo WiFi de igual manera se puede programar en Arduino IDE, mediante el uso de las librerías **Wifi.h** y **WebServer.h** y utilizando el siguiente código:

```
#include <WiFi.h>
#include <WebServer.h>

const char* ssid = "NombredeRed";
const char* password = "Contraseña";

WebServer server(80);

const int potPin1 = 13; // GPIO 13
const int potPin2 = 12; // GPIO 12
const int potPin3 = 14; // GPIO 14

void setup() {
  Serial.begin(115200);

  pinMode(potPin1, INPUT);
  pinMode(potPin2, INPUT);
}
```

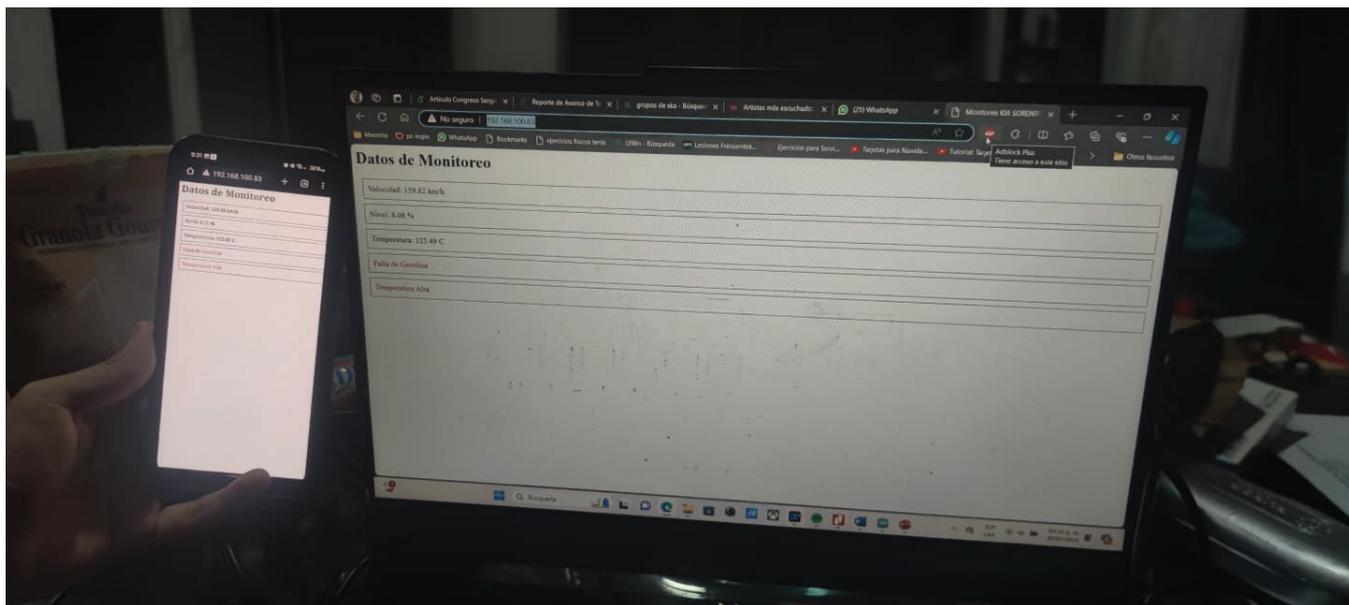


Figura 11. Dos dispositivos diferentes en donde se puede observar la pagina web conectadas a la misma red.

```
pinMode(potPin3, INPUT);

WiFi.begin(ssid, password);
Serial.print("Connecting to WiFi");
while (WiFi.status() != WL_CONNECTED)
{
    delay(1000);
    Serial.print(".");
}
Serial.println("\nConnected to WiFi");
Serial.println(WiFi.localIP());

server.on("/", handleRoot);
server.on("/data", handleData);

server.begin();
Serial.println("HTTP server started");
}

void loop() {
    server.handleClient();
}

void handleRoot() {
    server.send(200, "text/html", ...);
}

void handleData() {
    int potValor1 = analogRead(potPin1);
    int potValor2 = analogRead(potPin2);
    int potValor3 = analogRead(potPin3);

    float velocidad = (240.0 * potValor1)...;
    float nivel = (100 * potValor2) ...;
    float temperatura = (130 * potValor3)...;

    Serial.print(velocidad);
    Serial.print(",");
    Serial.print(nivel);
    Serial.print(",");
    Serial.println(temperatura);

    String jsonResponse = "{\"velocidad\": ...}";
}
```

De esta manera, ocupamos tanto la comunicación serial entre nuestra interfaz y la tarjeta ESP32, además de su módulo WiFi, con lo que se completa la primera etapa de nuestro prototipo.

IV. Resultados, conclusiones y trabajo futuro

Hasta el momento se cuenta con un módulo funcional, el *gateway*, el cual cuenta con tres sensores resistivos (potenciómetros) que en este caso generan los valores que enviarán los otros 3 módulos que conforman la red vehicular completa. Por el momento, hemos asegurado la comunicación entre nuestra interfaz gráfica de usuario del cuadro de instrumentos con el *gateway*. De igual manera, tenemos dentro de una red local el envío de los datos. Dichos datos los podemos observar mediante una IP. Dicha IP nos abre una página en donde podemos observar la información que llega a nuestro cuadro de instrumentos. En la figura 11 se puede observar dos dispositivos que muestran la página web donde se observan los datos.

En conclusión, se cuenta con un avance interesante en el desarrollo de nuestra red vehicular. Gracias a las tarjetas seleccionadas podemos tener un proyecto que no genera un costo tan elevado para su implementación. Por lo cual, es posible diseñar una red con más nodos interconectados. Se espera que esta red sea de ayuda para comprender de mejor manera el funcionamiento del protocolo CAN y como este protocolo ayuda al conductor a estar más seguro y conocer el estado del vehículo.

El trabajo a futuro consiste en completar la red vehicular, incorporando los módulos restantes. De igual mane-

ra, se utilizarán sensores convencionales que tomarán el papel de los sensores automotrices que forman parte del sistema que estamos implementando. Entre los sensores que ocuparemos se encuentran el sensor de nivel, termistor, sensor de efecto *hall*, entre otros. Dichos sensores se caracterizarán y se linealizarán en caso de ser necesario para su posterior incorporación a nuestra red vehicular.

De igual manera, se buscará la forma de poder enviar los datos a un servidor y poder observar los datos en una página de internet o aplicación celular, sin tener la necesidad que la información se encuentre en una red local.

Referencias

- [1] Pimentel Cruz, Jesús , Pérez Romero Patricia . Introducción a las Redes CAN 1a. Parte. Polibits. 2007, (35), 18-21.
- [2] Martínez Requena, Adrián. Introducción a CAN bus: Descripción, ejemplos y aplicaciones de tiempo real. Tesis de Maestría. Universidad Politécnica de Madrid. 2017.
- [3] N. Yee, P. Chand and S. Foehst, "Student Designed CANBus Simulator Used as Teaching Aid in Autotronics Course,"2017 4th Asia-Pacific World Congress on Computer Science and Engineering (APWC on CSE), Mana Island, Fiji, 2017, pp. 82-87.
- [4] López-López, F., Gomis-Bellmunt, O., Teixidó-Casas, M., Rafecas-Sabaté, J. and Muñoz-Gazquez, J.P. (2011), A novel educational platform to teach CANopen field bus. *Comput. Appl. Eng. Educ.*, 19: 377-384.
- [5] Kia Motors México, Conoce el tablero de tu Kia", YouTube, 2019 [Video en línea]. Disponible en <https://youtu.be/QvukdWwHLA?si=cXm2x9AcK0HAvDbP>.
- [6] M. Babiuch, P. Foltýnek and P. Smutný, "Using the ESP32 Microcontroller for Data Processing,"2019 20th International Carpathian Control Conference (ICCC), Krakow-Wieliczka, Poland, 2019, pp. 1-6.
- [7] T. Cherifi, A. Salag, and S. Kerrouchi, Development of an educational low-cost and ESP32-based platform for fundamental physics, *Comput. Appl. Eng. Educ.* 2023; 31: 1796–1807.
- [8] Liu, J. M., Wang, J. H., & Sun, D. H. (2014). The Design of CAN Bus Communication System Based on MCP2515 and S3C2440. *Advanced Materials Research*, 933, 516–520.
- [9] J. Palomino, E. Cuty and A. Huanachin, "Development of a CAN Bus datalogger for recording sensor data from an internal combustion ECU,"2021 IEEE International Workshop of Electronics, Control, Measurement, Signals and their application to Mechatronics (ECMSM), Liberec, Czech Republic, 2021, pp. 1-4.
- [10] Kiesbye, J.; Messmann, D.; Preisinger, M.; Reina, G.; Nagy, D.; Schummer, F.; Mostad, M.; Kale, T.; Langer, M. Hardware-In-The-Loop and Software-In-The-Loop Testing of the MOVE-II CubeSat. *Aerospace* 2019, 6, 130.
- [11] S. Sharma, J. Lúßmann and J. So, Controller Independent Software-in-the-Loop Approach to Evaluate Rule-Based Traffic Signal Retiming Strategy by Utilizing Floating Car Data, in *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3585-3594, Sept. 2019.
- [12] Alan D. Moore, Python GUI Programming with Tkinter: Design and build functional and user-friendly GUI applications , Packt Publishing, 2021.
- [13] Tomás. Domínguez Mínguez, Desarrollo de interfaces gráficas en Python 3 con Tkinter, 1ª ed. Barcelona: Marcombo, 2022.
- [14] Espressif Systems, ESP 32 Series Datasheet, 2020.
- [15] S. Karnik, et al., Design of CAN Bus Controller and Implementation for Automotive Application, *International Journal of Computer Applications*, vol. 97, no. 13, pp. 20-25, 2014.



Mejoras en uModelFactory: recursos para el diseño didáctico de sistemas embebidos

Nicolás Almaraz
Departamento de Ing. Electrónica
UTN BA
Argentina
nalmaraz@frba.utn.edu.ar

Nahuel González
Departamento de Ing. Electrónica
UTN BA
Argentina
ngonzalez@frba.utn.edu.ar

Resumen— En el marco del proyecto PID ASPPBBA0008444 del Departamento de Ingeniería Electrónica de la UTN FRBA, se desarrolló el software uModelFactory, destinado al diseño, depuración y simulación de máquinas de estados finitas tipo Moore. Este software, pensado tanto para la industria como el ámbito educativo, fue implementado en la cátedra de Informática II como herramienta didáctica para sistemas embebidos. Sin embargo, la interfaz inicial resultó poco intuitiva para los desarrolladores novatos. Para mejorar la experiencia del usuario, se añadió una sección de ayuda dinámica con tres niveles, proporcionando información contextual sobre los elementos del software. Además, se implementó una funcionalidad para documentar proyectos, permitiendo la generación de documentos editables (Docx) que incluyen descripciones, diagramas, listas de variables, eventos e implementaciones en C. Estas mejoras favorecieron la usabilidad del software para los estudiantes.

Palabras clave: Máquinas de Estados, Sistemas Embebidos, Documentación de Proyectos, Software de Diseño, Educación.

I. INTRODUCCIÓN

El software uModelFactory ^[1] ^[A1] se presenta como una herramienta para asistir en la elaboración de diagramas y máquinas de estado finitas (MEF) tipo Moore gobernados por eventos ^[2]. Su uso ha sido analizado en el ámbito educativo, específicamente en la cátedra de Informática II de la Universidad Tecnológica Nacional - Facultad Regional Buenos Aires, donde se ha valorado como una tecnología útil para el diseño y diagramado de MEF con creciente complejidad. Sin embargo, la realimentación de los estudiantes y equipos

docentes ha identificado deficiencias en la simplicidad y facilidad de uso del software, lo que ha llevado a aumentar su curva de aprendizaje, generando algunas dificultades en la apropiación del conocimiento.

Además, las versiones anteriores del software permitían la documentación automática mediante la generación de archivos HTML, pero estos no eran editables y no cumplían con el objetivo de generar informes personalizables para ser aprovechados por los estudiantes como parte de su proyecto integrador. Por ello, se ha desarrollado una mejora en la funcionalidad de documentación, que ahora permite la generación de archivos editables en formatos Docx u ODT. Esta actualización facilita la interacción con el software y la personalización de los documentos, mejorando así la experiencia del usuario y el proceso de aprendizaje.

II. MARCO TEÓRICO

A. Concepción de las máquinas de estado

Las máquinas de estado son una técnica fundamental para modelar el comportamiento de sistemas diversos, desde electrodomésticos como una cafetera, hasta procesos industriales complejos. Este modelo permite estructurar el comportamiento del sistema dividiéndolo en distintos "estados", cada uno representando una fase o momento del proceso. Los estados están conectados por "transiciones", que se activan mediante estímulos externos o eventos

de entrada, y determinan cómo el sistema cambia de un estado a otro. Estas transiciones también definen las "salidas" o acciones que el sistema debe ejecutar en respuesta a dichos estímulos.

B. Componentes de una máquina de estado

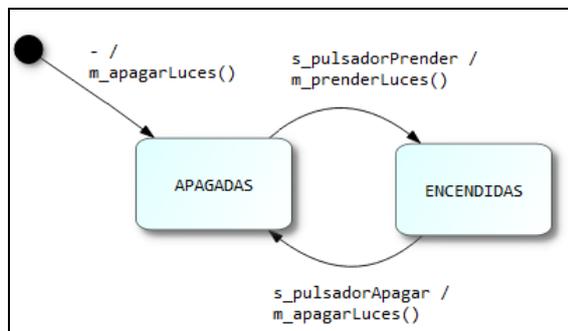


Figura 1: Diagrama con pilares de una máquina de estados elaborada en uModelFactory

Las máquinas de estado se componen de cuatro elementos esenciales (Figura 1):

Estados: representan las diferentes fases de un proceso dentro del sistema. Cada estado se transita a través de condiciones específicas representadas en diagramas de globo (por ejemplo, "ENCENDIDAS" y "APAGADAS").

Transiciones: son los cambios de un estado a otro, impulsados por eventos específicos. En los diagramas, las transiciones se representan mediante flechas que conectan los estados. Cada transición está asociada con un evento que la desencadena y puede incluir acciones que deben llevarse a cabo. Por ejemplo, una transición de "ENCENDIDAS" a "APAGADAS" podría ser activada por el evento "s_pulsadorApagar", que desencadenaría la acción de apagar las luces.

Eventos: son estímulos, ya sean internos o externos, que provocan cambios en el estado del sistema. Un evento puede ser la presión de un pulsador, que genera un evento como "s_pulsadorApagar".

Acciones: son las operaciones que se ejecutan cuando se produce una transición entre estados. Estas acciones determinan el comportamiento del sistema durante la transición, como encender o apagar luces.

C. Estados compuestos y máquinas de estado concurrentes

Los sistemas complejos a menudo requieren estructuras más avanzadas, como estados compuestos y máquinas de estado concurrentes. Los estados compuestos permiten la inclusión de máquinas de estados adicionales dentro de un estado, creando una jerarquía de estados que facilita la representación de sistemas más complejos. Por otro lado, las máquinas de estado concurrentes gestionan múltiples procesos simultáneamente, permitiendo la ejecución paralela de eventos. uModelFactory soporta estas configuraciones avanzadas, ofreciendo herramientas para simular y gestionar estos aspectos complejos, ampliando así su aplicabilidad.

D. Áreas de aplicación

Las máquinas de estados son particularmente útiles en sistemas embebidos, donde se necesita un enfoque sistemático para manejar estímulos y ejecutar tareas específicas. Este método organiza las tareas de manera estructurada, lo que facilita la implementación y reduce la probabilidad de errores.

El software uModelFactory está diseñado específicamente para este campo, permitiendo no solo la simulación de sistemas embebidos sino también la generación de código necesario para su implementación en microcontroladores. Esta capacidad lo convierte en una herramienta valiosa para el diseño y desarrollo de sistemas embebidos complejos.

III. MATERIALES Y MÉTODOS

Las opiniones de los estudiantes se recopilaron mediante encuestas anónimas, observaciones directas de los docentes y consultas frecuentes durante las clases.

Las encuestas incluían preguntas sobre aspectos personales, como el tiempo dedicado al estudio y la disponibilidad del hardware necesario para ejecutar el software, así como sobre la facilidad de uso en función de la comprensión de los conceptos abordados

Algunos ejemplos de las preguntas incluidas en la encuesta fueron:



- ¿Cómo calificarías el uso del software para el modelo de MEF?
- ¿Cómo calificarías su interfaz gráfica y experiencia de usuario?
- ¿Cómo calificarías la generación de código proporcionada?

Con base en esta información, se diseñó un sistema de ayuda para explicar conceptos clave del marco teórico relacionado con los diagramas y las MEF. La información se filtró para enfocarse exclusivamente en críticas sobre las características y presentación del software.

Se desarrolló el sistema de ayuda segmentándolo en tres áreas principales: categorización cualitativa de los temas en niveles definidos, fácil traducción y modificación, y una interfaz intuitiva y efectiva.

Para el desarrollo del software, se utilizó el framework Qt debido a su amplia gama de bibliotecas adaptables y su documentación detallada. No obstante, dado que Qt no incluye clases para la escritura directa de archivos Docx, se optó por implementar esta funcionalidad en Python utilizando el módulo `python-docx` [3].

La estrategia consistió en generar un documento XML auxiliar desde la aplicación, que luego se procesó con un script en Python para crear el archivo Docx final. La implementación dentro del bloque uModelFactory se organizó en tres niveles jerárquicos:

-Nivel alto: define el template del informe y recibe el contenido del mismo.

-Nivel intermedio: incluye métodos para configurar el template, como el formato de encabezados, títulos y subtítulos.

-Nivel bajo: se encarga de las funciones básicas, como la escritura de texto con formato,

imágenes, tablas, listas, encabezados y pies de página.

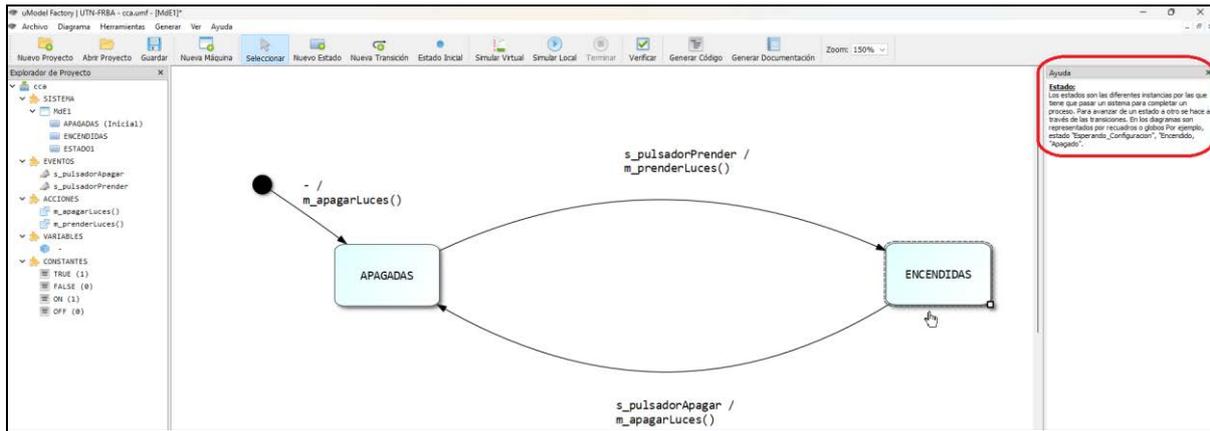
IV. RESULTADOS

A. División cualitativa de la ayuda

La implementación de la ayuda dinámica en uModelFactory se realizó con el objetivo de facilitar un aprendizaje [4] progresivo que pueda contar con apoyos flexibles donde el estudiante puede elegir el tipo de ayuda. Se introdujeron tres niveles de ayuda: Básico, Medio y Avanzado. El nivel Básico es el más detallado y está destinado a los usuarios principiantes, proporcionando información extensa sobre los elementos del software. El nivel Medio ofrece una reducción en la cantidad de detalles, adecuado para usuarios con experiencia intermedia. El nivel Avanzado proporciona una referencia rápida, útil para usuarios con conocimientos previos. Además, se incluyó una opción para desactivar completamente la ayuda, permitiendo una personalización del entorno de trabajo. La configuración de estos niveles se puede realizar desde la barra de herramientas en la opción de ayuda.

B. Fácil traducción y modificación del contenido

Para facilitar la modificación y traducción del contenido de ayuda, se empleó un archivo en formato XML [5] (eXtensible Markup Language). Esta elección se debe a la facilidad de implementación y edición que ofrece XML, así como su capacidad para organizar la información de manera clara y accesible. Los archivos XML permiten almacenar tanto los textos de ayuda como metadatos sobre el versionado y el idioma, y son legibles por cualquier persona sin necesidad de software especializado. Esta estructura facilita la adaptación del contenido a diferentes idiomas y la actualización continua del sistema de ayuda.



<p>Ayuda</p> <p>Estado: Los estados son las diferentes instancias por las que tiene que pasar un sistema para completar un proceso. Para avanzar de un estado a otro se hace a través de las transiciones. En los diagramas son representados por recuadros o globos. Por ejemplo, estado "Esperando_Configuración", "Encendido", "Apagado".</p>	<p>Ayuda</p> <p>Estado de reset: Se le llama estado de reset al estado en el que se inicia la máquina de estados. En el diagrama se puede identificar como una transición que no parte propiamente de ningún estado de la máquina y conduce al estado inicial. En esta se indican las condiciones iniciales de la máquina. Por ejemplo, los motores y la luz roja comienzan apagados, la luz verde comienza encendida.</p>	<p>Ayuda</p> <p>Transición: Es el paso entre un estado y otro. Estas se ejecutan si se cumplen ciertas condiciones de entrada. Se dice que una transición se dispara frente a un evento determinado. En estas transiciones se accionan las salidas o alguna variable intermedia en caso de ser necesario. Son representadas por las flechas donde se indican "eventos/ acciones". Por ejemplo, apagar luces cuando se presiona el pulsador de parada.</p>
--	--	---

Figura 2: Captura de pantalla de la aplicación con el panel de ayuda desplegado y resaltado en rojo (con diferentes ejemplos de diálogo)

C. Interfaz intuitiva

Se optó por integrar la ayuda dinámica en un panel móvil y de tamaño editable dentro de la ventana principal del software, para maximizar la comodidad del usuario. El panel muestra información contextual sobre el objeto bajo el cursor del mouse, adaptándose al nivel de ayuda seleccionado. Esta interfaz permite al usuario acceder fácilmente a información sobre estados, transiciones y otros elementos del diagrama, mejorando así la experiencia de uso. En la figura 2 se muestra un ejemplo de la implementación

D. Devolución por parte de los estudiantes

La retroalimentación de los cursantes actuales y estudiantes que aprobaron la materia sobre la nueva funcionalidad de ayuda dinámica ha sido positiva. Los estudiantes que utilizaron esta versión de uModelFactory reportaron que la ayuda dinámica mejoró significativamente la interacción en el proceso de diseño. Los estudiantes actuales no manifestaron problemas ni sugerencias adicionales relacionadas con la accesibilidad de la información, indicando una aceptación generalizada de las mejoras implementadas.

E. Diseño de la propuesta de documentación

El nuevo sistema de documentación permite a los usuarios generar informes configurables en formatos Docx o HTML. Un botón agregado en la barra de herramientas abre un menú de configuración que permite seleccionar qué secciones incluir en el informe, como autor, fecha, descripción del proyecto, introducción teórica, máquinas de estados y sus implementaciones en C. Al configurar el informe y presionar "Ok", se genera una ventana de confirmación que muestra la ruta del archivo exportado (Figura 3).

Figura 3: Pantalla de configuración de documentación

Los informes generados (Figura 4) incluyen una página de portada con detalles de la universidad y del autor, una descripción del proyecto, y listas de acciones, eventos y variables. La segunda página presenta la máquina de estados, incluyendo nombre, diagrama y tabla de estados y transiciones. Finalmente, el informe incluye una sección para

las implementaciones en C, cargando el contenido de los archivos ubicados en el directorio especificado por el usuario. Esta funcionalidad permite la integración de código no exportado directamente desde uModelFactory, ofreciendo flexibilidad en la documentación del proyecto.

The image shows two pages from a generated document. The left page contains project metadata (UTN.BA logo, authors: Nicolas, date: 27-08-2024 16:38:50), a title 'Ejemplo', a project description, lists of events (s_pulsadorApagar, s_pulsadorPrender), actions (m_apagarLuces(), m_prenderLuces()), and variables (No hay). It also features a state machine diagram with states 'APAGADAS' and 'ENCENDIDAS' and transitions labeled with events and actions. The right page shows a table of states and transitions, followed by a C code implementation for 'AP FuncionesMDE.c' which includes file paths, comments, and preprocessor directives.

Figura 4: Ejemplo de archivo Docx generado con uModelFactory

V. CONCLUSIONES

El impacto positivo de las nuevas funcionalidades implementadas en uModelFactory demuestra que un mayor número de herramientas a disposición del alumno puede mejorar significativamente su rendimiento. Esto subraya la importancia de continuar desarrollando y perfeccionando herramientas que faciliten el proceso de aprendizaje. Actualmente, se está trabajando en la creación de un menú de ayuda con un buscador integrado que permitirá a los usuarios acceder rápidamente a descripciones de funcionalidades y aspectos teóricos de la aplicación.

Esta nueva herramienta también estará configurada con archivos XML para permitir la traducción a diferentes idiomas y contará con un sistema de hipervínculos que ofrecerá información adicional sobre herramientas útiles, adaptándose a las necesidades del usuario y evitando la sobrecarga de información no deseada.

El desarrollo del módulo de documentación ha cumplido con las necesidades identificadas, ofreciendo una solución para la generación automática de documentación en un formato editable. Esto permite a los usuarios, especialmente a los estudiantes de la asignatura Informática II, complementar y extender la documentación generada con otros



módulos o información adicional. La documentación automática proporciona una base sólida para la realización de trabajos prácticos integradores, apoyando la continuidad y desarrollo de proyectos.

Para futuras mejoras, se planea refinar la funcionalidad de inclusión de código en los informes. Una posible mejora sería permitir la selección específica de funciones o fragmentos de código para incluir en el informe, en lugar de incorporar el código completo. Este avance implicará la capacidad de interpretar el código de archivos fuente seleccionados, brindando a los usuarios la opción de incluir solo las partes relevantes de su código, optimizando así la documentación generada. La evaluación de estas mejoras se realizará con la cohorte de estudiantes de la asignatura en el año 2024, con el fin de ajustar y perfeccionar las funcionalidades según las necesidades de los usuarios.

REFERENCIAS

- [1] L. Sugezky, M. Prieto, N. González, M. Giura, Y. Kuo, M. Trujillo, J.M. Cruz. "Desarrollo e implementación de herramientas de simulación de modelos para sistemas embebidos". Congreso Argentino de Sistemas Embebidos, 2016.
- [2] N. Gonzalez, J. Cruz, L. Sugezky, M. Giura, M. Trujillo, M. Prieto. "Analysis of a UML-based embedded system modeling software application". Congreso Argentino de Sistemas Embebidos, 2014.
- [3] python-docx, "python-docx documentation", Versión 0.8.11, python-docx, 2024. Disponible en: https://www.oasis-open.org/committees/xliff/documents/translation_wxml_tools.pdf [Consultado el: 27/08/2024]
- [4] A. Zydron, (2004). "Translating XML Documents with XML". Disponible en: <https://www.xml.com/pub/a/2004/01/07/xlmtm.html> [Consultado el: 27/08/2024]
- [5] N. González, L. Sugezky, M. Prieto, M. Giura, Y. Kuo, M. Trujillo, J.M. Cruz, "Evaluación del software uModelFactory como herramienta didáctica". IEEE Argencon, 2016.

ANEXOS

- [A1] Enlace al sitio oficial de uModelFactory:
<https://www.frba.utn.edu.ar/electronica/umodelfactory-2/>

Automatización y control de redes de distribución de agua

1st Luis Mariano Campos
Profesional Adjunto - Docente
Conicet - UNT
San Miguel de Tucumán, Argentina
lmarianocampos@gmail.com

Resumen—El presente documento aborda el diseño y la construcción de un prototipo a escala para el control automático del caudal de agua en canales abiertos. El objetivo principal del sistema desarrollado es gestionar de manera eficiente una red de canales. Lo que permite resolver problemas de disponibilidad en regiones con sequía y mejorar el suministro a cada usuario. Como consecuencia de esto, se optimiza la gestión por parte del proveedor. Se diseñó una válvula de control servoalimentada, que implementa un motor paso a paso como elemento actuador. Para la función de control, se aplica un algoritmo PI. La medición de la variable que se controla se realiza mediante un caudalímetro, diseñado e implementado con una placa de aforo triangular cuyas características dimensionales se determinaron de manera empírica. El componente principal de la etapa de control es un microcontrolador con arquitectura ARM Cortex-M4, en el que se implementó una aplicación y se ejecuta mediante un sistema operativo de tiempo real. Además, se desarrollaron interfaces de programación de aplicaciones (API) necesarias para la modularización del software.

Palabras clave—Automatización, control de caudal, canales abiertos, PID, microcontroladores, sistema de tiempo real, electroválvulas

I. Introducción general

Este análisis se centra en la conformación de una red de canales de agua para riego a cielo abierto. Se examinan las dificultades que estas enfrentan en la actualidad y se destaca la necesidad de desarrollar un sistema tecnológico que actúe como solución para gestionar el recurso hídrico de forma eficiente.

I-A. Descripción del estado actual sobre el manejo de agua en canales abiertos

Una red de canales de riego [1] es una extensa área geográfica formada por un conjunto de canales que comparten una o más fuentes de abastecimiento. Su función principal es transportar el agua desde el punto de captación hasta los campos agrícolas donde se aplica a los cultivos. La gestión de estas redes abarca no solo el manejo físico del agua y la infraestructura, sino también aspectos legales, administrativos, socioeconómicos y productivos. Por decreto del poder ejecutivo, se

otorgan títulos de concesión a los usuarios, quienes se organizan en asociaciones civiles responsables del empleo, administración, operación, conservación del agua e infraestructura hidrológica. Para obtener una comprensión detallada, en la Fig. 1 se ilustra una configuración típica de una red de canales de riego. Una vez comprendida

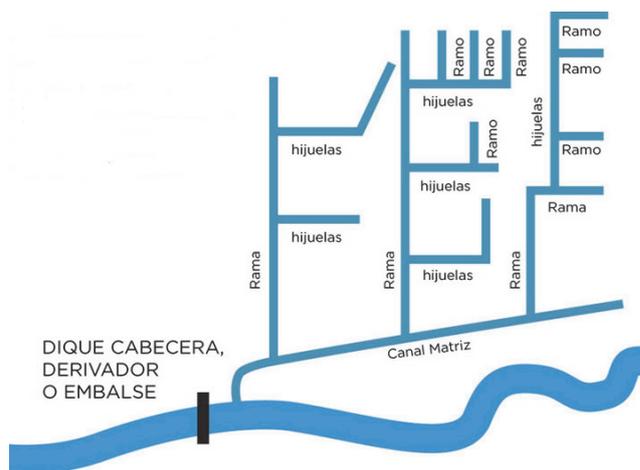


Figura 1. Configuración típica de una red de canales de riego a cielo abierto, que ilustra la distribución del agua para la irrigación agrícola.

la configuración general de una red de canales de riego, es posible analizar el caso específico en la provincia de Jujuy. Este sistema abarca 128 km de canales de tierra, 64 km de canales revestidos con hormigón y 400 km de acequias de tierra, diseñados principalmente para caudales inferiores a 150 L s^{-1} . Los operarios recorren diariamente estos puntos, para cubrir desde pocos hasta varios kilómetros, a veces en múltiples ocasiones al día, para verificar, ajustar caudales, medir niveles de agua y realizar aforos manualmente. Los datos obtenidos se utilizan para cálculos precisos y fijar caudales específicos. La red incluye canales de diversas jerarquías (principales, secundarios, terciarios, cuaternarios y acequias) que suministran agua a diferentes grupos de usuarios. La distribución del agua a un usuario depende, generalmente,

de varios movimientos de compuertas en cadena. La 'operación de canales de riego' se refiere al conjunto de actividades necesarias para planificar, programar, distribuir y entregar el agua de riego a los productores de manera eficiente y oportuna. De este modo, se asegura que el agua llegue a los cultivos en el momento y cantidad precisos para optimizar la productividad agrícola.

I-B. Problemáticas actuales en la gestión del recurso hídrico y posible solución tecnológica

En diversas regiones del país, la gestión del agua es fundamental para sostener los sistemas socioproductivos, pero enfrenta desafíos significativos debido a la escasez, el empleo ineficiente y las dificultades de acceso al recurso. Aunque tanto organismos públicos como privados reconocen estos problemas. La falta de tecnologías adecuadas y los obstáculos en la organización y gestión limitan la capacidad de resolverlos. Actualmente, la operación de canales y redes de distribución de agua se realiza de forma manual, esto genera una lenta capacidad de respuesta ante la demanda variable, lo que aumenta el costo de operación. Una solución tecnológica que automatice estos procesos podría mejorar la eficiencia y reducir gastos.

Las pérdidas de agua en canales abiertos se clasifican en intrínsecas y operacionales. Las intrínsecas, que incluyen la evaporación, infiltración y fugas, dependen de factores ambientales y del estado de los canales, mientras que las operacionales están vinculadas al manejo del caudal, la experiencia del personal y la supervisión. Aunque la infiltración puede reducirse con revestimientos costosos, las pérdidas operacionales aceptables son inferiores al 5 %, considerándose adecuada una operación si se mantienen entre el 5 % y el 10 %. En la práctica, las pérdidas a menudo superan estos valores.

Para mejorar la gestión y reducir las pérdidas operacionales, se propone un esquema que automatice las compuertas y establezca un centro de monitoreo central. La comunicación entre las compuertas y el sistema principal se realizará mediante LoRaWAN, una tecnología de bajo consumo y largo alcance ideal para áreas sin cobertura telefónica, internet o energía eléctrica, donde se utilizará energía solar fotovoltaica. Además, se desarrollará una aplicación móvil que permitirá a los operarios gestionar el sistema de forma remota.

Cada compuerta que se automatice estará equipada con un motor eléctrico, un microcontrolador, un caudalímetro, un panel solar y un dispositivo LoRaWAN para la comunicación. Este conjunto de componentes forma parte de un sistema global de gestión de una red de canales, que permitirá tanto el control remoto desde el centro de monitoreo como la operación autónoma mediante preprogramación. Dentro de este sistema, la unidad básica de control en cada punto de la red, responsable de gestionar localmente el movimiento de las compuertas y ajustar el caudal de agua según las necesidades se

denomina 'Celda Primaria'. Este documento presenta la construcción de un prototipo a escala que demuestra cómo una celda primaria desempeña su función dentro de la red automatizada de compuertas. Esto se puede observar en la Fig. 2.

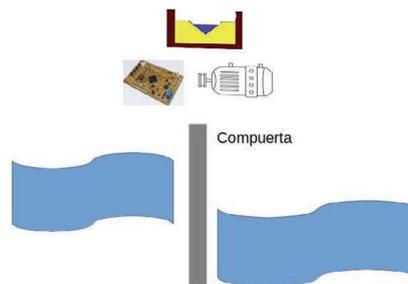


Figura 2. Celda primaria.

I-C. Motivación

En diversas provincias del país, las redes de canales de riego enfrentan serias deficiencias en el mantenimiento y control de la gestión del agua, lo que genera desafíos significativos para las asociaciones que administran el recurso. En la provincia de Jujuy, específicamente en el Consorcio de Riego del Valle de Perico, aunque no se presentan problemas graves de escasez, la falta de control preciso sobre el caudal suministrado a los usuarios es evidente. Los administradores han reportado dificultades logísticas debido a la distancia entre compuertas, lo que incrementa los costos operativos por las necesidades de visitas frecuentes.

En La Rioja, los pequeños productores sufren la escasez de agua superficial, particularmente entre octubre y diciembre, esto afecta gravemente sus cultivos. Las principales causas de este problema incluyen la ineficiencia en la recolección de agua, la deficiente supervisión de las compuertas y la falta de instrumentos de medición de caudal. Además, el manejo manual de las compuertas facilita el aprovechamiento no autorizado del recurso por parte de algunos usuarios.

Estas problemáticas impulsaron el desarrollo de este trabajo, que busca ofrecer una solución competitiva y de fabricación nacional, tanto en costo como en prestaciones, frente a los productos importados. Además, se pretende proporcionar soporte local y adaptar el sistema a las necesidades específicas de los clientes.

I-D. Objetivos y alcances

Este trabajo tiene como objetivo general diseñar y desarrollar un sistema que, con el empleo de un algoritmo de control PID, regule el caudal de agua en canales abiertos. El sistema está diseñado para ofrecer una alta adaptabilidad, capaz de integrarse tanto en redes de canales complejas como en configuraciones más simples, mediante la incorporación de tecnologías de última generación. A continuación, se detallan los subobjetivos específicos considerados en el trabajo.

- Diseño y construcción de un sistema de control que permita al usuario establecer un caudal de agua por medio de una aplicación externa.
- Diseño y fabricación de un circuito de control de señales cuya finalidad es cumplir la función de interfaz entre la etapa de control y la de potencia correspondiente al driver del servomotor paso a paso.
- Incluir como parte del firmware un sistema de tiempo real.
- Para regular el caudal de agua, aplicar un algoritmo de control PID.
- Empleo de un sensor de presión para detectar el nivel de la superficie del agua.
- La apertura, control y cierre de las compuertas, se realizará mediante un sistema de servomotor, paso a paso, cuyo ángulo de actuación se medirá y controlará a través de un sensor potenciométrico resistivo.
- Proporcionar una gestión de agua de riego de forma eficiente.
- Desarrollar un firmware escalable que se ejecute en cada celda primaria y permita la comunicación entre ellas.

II. Introducción específica

En esta sección se abordan temas relacionados con el análisis de la estructura del sistema desarrollado, la gestión y planificación del trabajo, así como las técnicas empleadas en la utilización del sensor.

II-A. Estructura general del sistema

Durante el diseño del prototipo a escala, se identificó un problema con la utilización de una compuerta colocada perpendicularmente al canal y con desplazamiento vertical. Esta configuración provocaría filtraciones de agua en las guías de desplazamiento. Estas afectarían las mediciones de la altura de la superficie de agua y los cálculos necesarios para determinar el caudal mediante el método de aforo por compuerta [2]. En consecuencia, el sistema no proporcionaría resultados precisos, ya que el caudal que circularía por debajo de la compuerta sería menor que el que se registraría en el caudalímetro. Esto produciría dos valores de caudal diferentes. Para solucionar este problema, se decidió reemplazar en el

prototipo la compuerta por una válvula, así eliminar los errores causados por dichas filtraciones. El control de la válvula se realiza mediante un servomotor, que se elaboró con el empleo de un motor paso a paso y su controlador correspondiente. Para medir el caudal, se fabricó un caudalímetro fundamentado en una placa de aforo triangular, también conocido como vertedero [3]. Este instrumento, que funciona como un dique o pared, intercepta el flujo de agua en un punto específico del canal.

En general, los vertederos se utilizan para mantener un nivel de agua constante aguas arriba o para medir el caudal del agua que circula por el canal. Este tipo de medidor es sencillo, pero efectivo para su empleo en canales abiertos. El principio de funcionamiento de la celda primaria consiste en ajustar el caudal de agua a un valor establecido por el usuario. Para lograr esto, se desarrolló un firmware que controla un servomotor conectado a una válvula. Esta puede ser una llave esférica o un sistema de cremallera, y regula el flujo de agua según sea necesario. En el diagrama de control de la Fig. 3, se puede apreciar que el servomotor actúa directamente sobre la válvula, y el flujo controlado de agua se mide con un caudalímetro. Los resultados de este instrumento se comparan con el valor deseado (set point) fijado por el usuario. La diferencia entre el valor deseado y el valor medido genera una señal de error, que se procesa mediante un algoritmo de control PID para ajustar el caudal hasta que la señal de error sea cero.

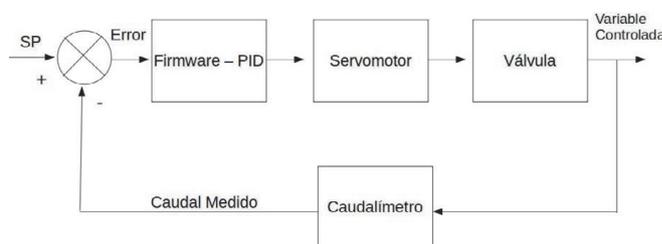


Figura 3. Diagrama en bloque de control - celda primaria.

II-B. Requerimientos a nivel de software y hardware

En esta sección se presentan los requerimientos específicos de software y hardware del sistema. Estos se tuvieron en cuenta al momento de inicio del desarrollo de este trabajo, cuyo fin es, mediante el control de movimiento de una válvula, regular el caudal de agua según las necesidades del usuario.

II-B1. Requisitos específicos del software: Los requisitos específicos de software fueron:

1. El firmware deberá generar como señal, trenes de pulsos para controlar el movimiento del eje perteneciente al servomotor.

2. El firmware deberá determinar, mediante el sensor de presión en conjunto con la placa de aforo triangular, el caudal de agua que fluye a través de este instrumento de medición.
3. El firmware deberá incluir un algoritmo de control PID, que mediante un lazo de retroalimentación regule la variable a controlar.
4. El firmware deberá ser capaz, a través de un pin configurado como salida, establecer el sentido de giro del eje del servomotor, horario - antihorario.
5. El firmware deberá habilitar o deshabilitar el servomotor mediante un pin configurado como salida.
6. El firmware deberá interactuar, mediante el empleo del puerto serie del microcontrolador, con otra aplicación.
7. Con un protocolo de comunicación definido entre la aplicación externa y el firmware, este último deberá ser capaz de recibir y enviar datos.
8. El firmware deberá enviar una notificación de correcta recepción de cualquier instrucción mediante el envío de un mensaje específico hacia la aplicación externa.
9. El firmware deberá reportar a la aplicación externa la recepción de una instrucción inválida mediante un comando específico.
10. En caso de recepción de comandos válidos, el firmware deberá informar internamente que hay datos a procesar.

II-B2. Requisitos específicos del hardware: Los requisitos específicos principales de hardware fueron:

1. Se hará empleo de la placa EDU-CIAA-NXP [4] como computadora principal para el prototipo a escala.
2. Se construirá un mecanizado de válvula de control estándar mediante un servomotor energizado paso a paso, y un elemento medidor de ángulo tipo potenciométrico resistivo.
3. Se construirá un medidor de caudal con placa de aforo, que empleará un sensor de presión MPX5010DP para medir la altura del nivel de agua.
4. Se diseñará y fabricará un circuito como interfaz que controlará las señales enviadas desde la placa EDU-CIAA-NXP al controlador del servomotor paso a paso.

II-C. Características de hardware propias del equipo

II-C1. Motor paso a paso y controlador: Para construir la válvula de control, se utilizó un motor paso a paso y un controlador, específicamente los modelos 86HS85 y MA860H. El motor paso a paso, con un torque nominal de 8,5 Nm, es adecuado para realizar movimientos precisos de apertura y cierre de la válvula según las necesidades de caudal. El controlador MA860H es compatible con este motor y permite su funcionamiento eficiente.

II-C2. Sensor de presión: Para monitorizar el nivel de la columna de agua, se utilizó el sensor de presión MPX5010DP debido a su elevada precisión. Este transductor piezo-resistivo de silicio es compatible con microcontroladores o microprocesadores con entradas ADC. El MPX5010DP [5] entrega un rango de voltaje de salida entre 0 V y 5 V, lo que permite estimar la presión hasta 10 kPa, equivalente a una columna de agua de hasta 100 cm de altura. El dispositivo proporciona una salida lineal, como se muestra en la Fig. 4 de su hoja de datos.

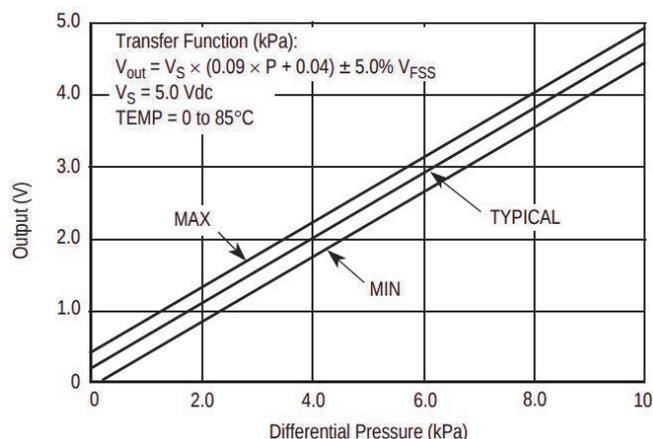


Figura 4. Función de salida del sensor de presión MPX5010DP.

De acuerdo con la función de la Fig. 4, la ecuación para obtener la presión del sensor viene dada por (1):

$$P = \frac{V_{out} - 0,04 \cdot V_s \pm Tol}{0,09 \cdot V_s} \quad (1)$$

Donde:

V_{out} : Es la tensión de salida del sensor.

V_s : Es la tensión de suministro al sensor.

Tol : Es la tolerancia del sensor.

La ecuación (2) de presión diferencial, que relaciona la altura, se utiliza como base para determinar que (recordando que a la presión atmosférica normal es cero):

$$P = P_H - P_L = \rho gh = \frac{V_{out} - 0,04 \cdot V_s \pm Tol}{0,09 \cdot V_s} \quad (2)$$

Donde:

P : Es la presión.

ρ : Es la densidad del agua, con un valor de $\rho = 1000 \text{ kg m}^{-3}$.

g : Es la aceleración debida a la gravedad, con un valor de $g = 10 \text{ m s}^{-2}$.

h : Es el nivel de la columna líquida.

La ecuación (3) resultante para obtener h es:

$$h = \frac{P}{g \cdot \rho} \quad (3)$$

La disposición de los pines del dispositivo se ilustra en la Fig. 5:

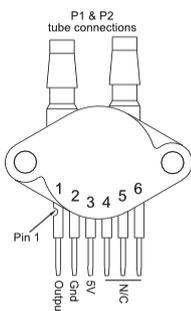


Figura 5. Disposición de pines del sensor de presión MPX5010DP.

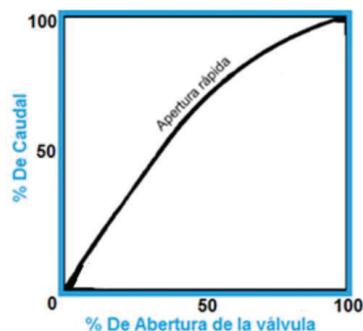


Figura 6. Gráfica del caudal en función de la apertura de la válvula.

III. Diseño e implementación

Este análisis abarca las decisiones clave tomadas durante el diseño e implementación del hardware y firmware. Asimismo, se destacan los aspectos más relevantes del desarrollo.

III-A. Hardware

III-A1. Construcción de la válvula de control: En la elaboración del prototipo, se creó una caja desmultiplicadora de fuerza con la finalidad de dirigir el recorrido de apertura y cierre de una válvula. Para generar estos movimientos se utilizó un servomotor paso a paso. La válvula es un ensamblaje compuesto de un cuerpo con conexión a una tubería y de un obturador operado por accionamiento. Su función principal consiste en variar el caudal del fluido que circula a través de ella, y se comporta como un orificio cuya área varía continuamente. Cuando el obturador se desplaza, genera un área de pasaje con una relación específica entre la fracción de carrera y el caudal que atraviesa dicha área. Esta relación recibe el nombre de característica inherente de caudal de válvula. Particularmente, para este trabajo se utilizó una válvula cuya característica inherente es tipo de apertura rápida. Se trata de una propiedad que produce una variación grande de caudal con una carrera pequeña. Esto posibilita el pasaje de casi la totalidad del caudal nominal con apenas una apertura de 25 % de la carrera total. De esta forma, genera una ganancia muy alta a bajas aperturas de carrera y una ganancia muy baja en aperturas por encima de 60 % de carrera total. La Fig. 6 muestra la curva típica de una válvula de apertura rápida.

III-A2. Servomotor: Los componentes principales para la construcción del servomotor son un motor paso a paso, un controlador, un sensor resistivo de ángulo y una entrada ADC del microcontrolador. En cuanto a su funcionamiento, se debe aclarar que está compuesto por una fracción del firmware que comprende la parte inteligente de este dispositivo fabricado. Asimismo, es el encargado de establecer al obturador de la válvula en una determinada posición, lo que permite proporcionar el caudal de agua requerido por el usuario. Para esto

fue indispensable obtener una estimación del porcentaje de apertura de la válvula. Por ello, se empleó un sensor resistivo de ángulo cuya señal se inyecta de forma retroalimentada a una de las entradas ADC del microcontrolador para su procesamiento. Este dispositivo adosado al eje de la válvula es un potenciómetro de una vuelta de 50 k Ω cuyo modelo es 91A-503, Bourns Cermet. En la Fig. 7, se puede apreciar un diagrama en bloque general que indica entre otras partes del trabajo, cómo está constituido el servomotor. Cada componente se encuentra delimitado en líneas punteadas, lo que permite identificar claramente qué partes pertenecen al servomotor y al microcontrolador.

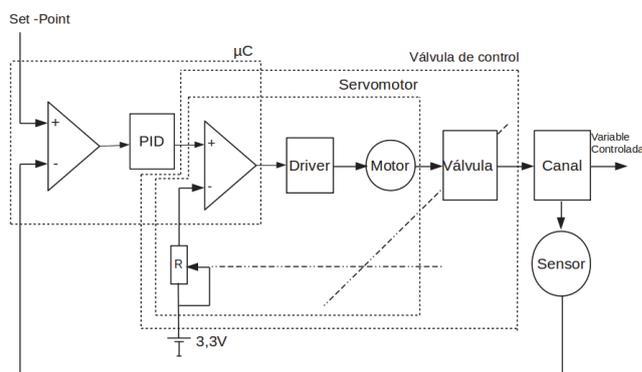


Figura 7. Diagrama en bloque de la celda primaria.

III-A3. Medidor de caudal: La medición del agua, también llamada aforo, es la cuantificación del caudal de agua que pasa por la sección transversal de un río, canal o tubería. La herramienta de medición utilizada en este trabajo para verificar el caudal de agua que se suministra, recibe el nombre de medidor de caudal de canal abierto tipo aforo. Para efectuar el presente trabajo se utilizó el tipo de aforo con vertedero triangular con escotadura en V. La medición del caudal en corrientes naturales nunca es exacta, debido a la irregularidad del canal. Esto provoca una relación variable entre el nivel y el caudal. Los canales de corrientes naturales están también sometidos a cambios debidos a erosión o depósitos. Se

pueden obtener cálculos más confiables cuando el caudal pasa a través de una sección donde esos problemas se han limitado. Los vertederos pueden ser definidos como simples aberturas, sobre las que un líquido fluye. El término se aplica también a obstáculos en el paso de la corriente y a las excedencias de los embalses. Los vertederos son orificios sin el borde superior y ofrecen las siguientes ventajas en la medición del agua:

- Se logra con ellos precisión en los aforos.
- La construcción de la estructura es sencilla.
- No son obstruidos por materiales que flotan en el agua.
- La duración del dispositivo es relativamente larga.

Para la construcción del vertedero triangular se precisó de una chapa de hierro plana de 1 mm de espesor. En ella se realizó una hendidura de sección triangular, cuyo valor de ángulo es de 18° . Esta chapa supera los límites del ancho del canal de forma que, ubicándola verticalmente y en forma transversal al canal, opera como un embalse que por su hendidura circula un flujo de agua. De esta forma se puede medir, en distintos lugares del recorrido del canal, el caudal real en un punto particular. La Fig. 8, muestra la vista frontal de una estructura de un vertedero con escotadura en V.

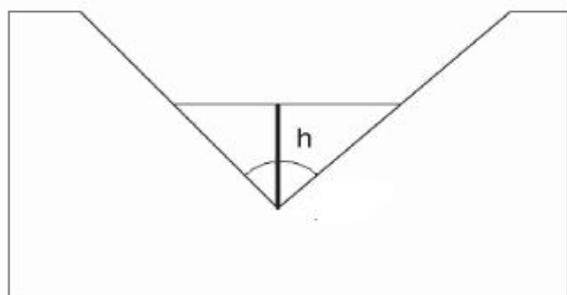


Figura 8. Placa de aforo: vista frontal.

La Fig. 9, ilustra la vista lateral del vertedero y cómo fluye el líquido por su abertura.

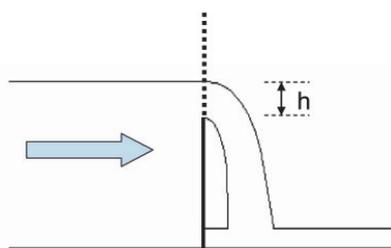


Figura 9. Placa de aforo: vista lateral.

La ecuación (4) se utiliza para calcular el caudal de la siguiente manera:

$$Q = \frac{8}{15} \cdot \sqrt{2gc} \cdot \tan(\alpha) \cdot H^{\frac{2}{3}} \quad (4)$$

Donde:

- Q : Caudal en $m^3 s^{-1}$.
- g : Coeficiente de gravedad.
- c : Coeficiente de descarga.
- α : Ángulo del vertedero en grados.
- H : Nivel de agua en m.

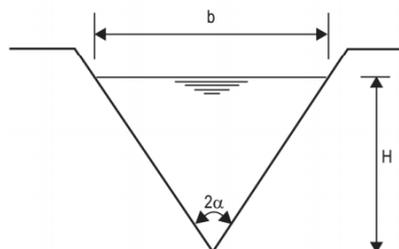


Figura 10. Placa de aforo: dimensiones.

La Fig. 11 presenta un vertedero triangular con una escotadura en V, que ejemplifica el diseño del dispositivo de control de flujo de agua.



Figura 11. Esquema de un vertedero triangular con escotadura en V, utilizado para controlar el flujo de agua en aplicaciones hidráulicas.

Los medidores de caudal con aforo triangular como el que se emplea en el presente trabajo, responden a una ecuación del modelo (4) obtenida empíricamente. El medidor empleado en este trabajo se utilizó una sola vez, por lo que la tabla de calibración del caudalímetro se obtuvo de manera experimental. En la Fig. 12, se detalla la vista de frente del canal con el vertedero triangular y los diversos niveles de altura que se consideraron al momento de realizar las mediciones pertinentes.

A partir de la Fig. 12 se simboliza una manguera, conectada por un extremo a la base del canal y, en el otro, a una de las dos espigas del sensor. Además, se puede advertir los niveles de altura que se deben tener en cuenta. H_{total} corresponde a la altura entre el nivel de agua en un instante dado y la manguera conectada al sensor, $H_{constante}$ se encuentra asociada al nivel de altura que está comprendido entre dicha manguera y el vértice del triángulo del vertedero, y finalmente $H_{variable}$ es la altura que está contenida entre el nivel de agua y el vértice del triángulo de dicho vertedero. Por lo tanto, este

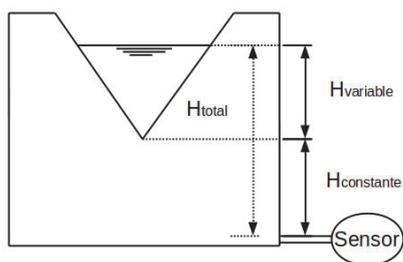


Figura 12. Placa de aforo: dimensiones.

caudalímetro de canal abierto por sistema de aforo utiliza como medidor de nivel de columna de agua un sensor de presión modelo MPX5010DP. La señal proveniente del sensor en conjunto con un algoritmo de control PID, permite regular el caudal de agua en un determinado valor preestablecido. Una vez construido el prototipo, se realizó una calibración de caudal obteniéndose una tabla de Caudal vs H_{total} (altura). Esta es utilizada en el firmware para obtener el valor de caudal en lugar de la fórmula presentada.

III-B. Software

III-B1. Arquitectura de software: Durante la definición de la arquitectura de software se seleccionaron e integraron dos para confeccionar una híbrida, de modo tal que se adapte a las necesidades concretas de este trabajo. Por un lado, se eligió una arquitectura que está conformada por tres capas claramente definidas. En la capa HAL, se encuentran los drivers encargados de abstraer los detalles de acceso al hardware. De esta forma, facilitará la migración a otro microcontrolador en caso de que en el futuro hiciera falta. Adicionalmente, permite una mayor claridad en la implementación, al separar los drivers de la lógica de la aplicación. Debido a que en la producción se utilizará como hardware la CIAA-NXP y durante el desarrollo la EDU-CIAA-NXP, se utiliza el firmware de la CIAA versión 3.0 como capa de abstracción de hardware. En la capa de aplicación se encuentran todos los componentes que encapsulan la lógica de aplicación, y finalmente, la tercera corresponde al sistema operativo de tiempo real FreeRTOS [6]. El sistema incluye sensores que proveen información relacionada al ámbito a controlar y un actuador para operar sobre el sistema. Por lo tanto, en respuesta a las alteraciones identificadas por los sensores, se envían señales de control a los actuadores. Entonces, al identificar que el sistema debe poseer este tipo de comportamiento se definió emplear una arquitectura de tiempo real inherente al "Control Ambiental" [7]. Este patrón de arquitectura de software, brinda la posibilidad de recopilar datos del entorno por medio de sensores, como así también el estado en el que se encuentran los actuadores conectados al sistema. Con

base en datos reunidos de sensores y del actuador, se envían señales de control hacia el actuador para producir cambios en el entorno controlado. En la Fig. 13, se puede observar el diagrama del patrón arquitectónico que es la base del diseño del sistema de control. Este se utiliza en la capa aplicación.

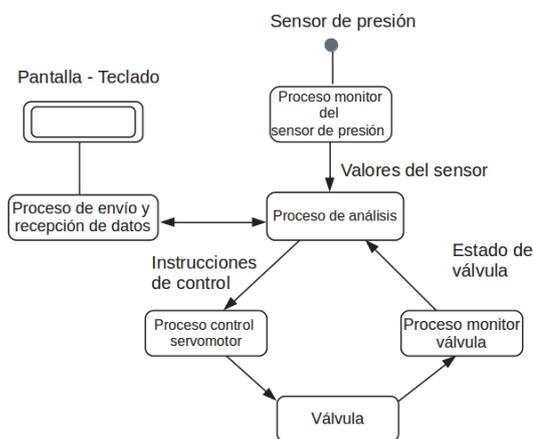


Figura 13. Arquitectura de sistema de control de caudal de agua en canal a cielo abierto.

Al aplicar ambos patrones se constituyó un patrón híbrido de tres capas. La capa inferior es la capa de abstracción de hardware. La capa intermedia corresponde al sistema operativo de tiempo real. Finalmente, la capa superior pertenece a la aplicación que incluye los componentes de un patrón de control ambiental.

III-B2. Componentes de software: Cada capa de software es considerada un componente de software. Con lo que se poseen los siguientes componentes:

- HAL.
- Sistema Operativo.
- Aplicación.

En la Fig. 14, se puede apreciar el nivel de jerarquía de cada una de las capas donde la HAL es la más baja:

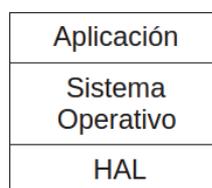


Figura 14. Jerarquía de componentes de software.

III-B3. Diseño detallado: Esta subsección incluye dos de los diseños detallados de los componentes de software presentados en la Fig. 13, con sus respectivos diagramas de actividades [8] Fig. 15 y Fig. 16.

III-B4. Casos de uso: En la etapa de captura de requerimientos funcionales que el sistema debía satisfacer, se emplearon casos de uso correspondientes al Lenguaje

Unificado Modelado (UML) [9]. En la Tabla I se presenta un caso de uso que se documentó como parte integral de la especificación de requisitos del software del equipo.

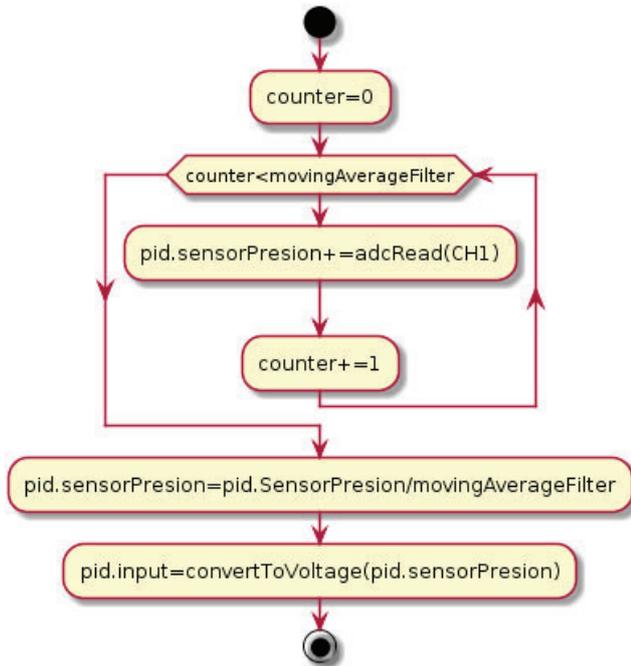


Figura 15. Proceso monitor del sensor de presión.

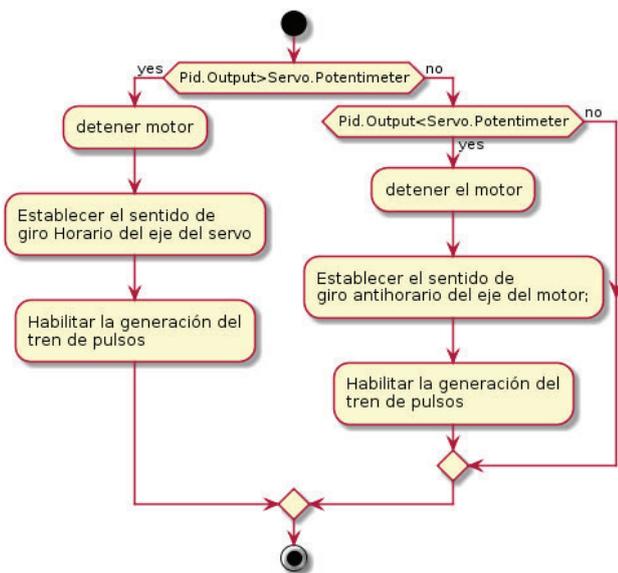


Figura 16. Proceso de control del servomotor.

Tabla I

Caso de Uso: Establecer un Determinado Valor de Caudal.

Nombre	Establecer un determinado valor de caudal.
Breve descripción	El firmware debe establecer un caudal mediante la manipulación de la posición de la válvula de control.
Actor Principal	La aplicación móvil.
Disparadores	La recepción de un comando.
Flujo de Eventos	<ul style="list-style-type: none"> El firmware recibe el comando desde un smartphone o desde una aplicación de comunicación serial desde una PC. El firmware desencapsula el comando recibido y extrae los diferentes campos de información. El firmware interpreta el tipo de operación. El firmware lee un pin configurado como ADC asociado al sensor de presión para obtener un valor digital. Se repite el paso anterior hasta 5 veces, para acumular los valores obtenidos. El firmware realiza un filtro de promedio móvil con los valores acumulados. El firmware convierte el valor digital promediado a voltaje. El firmware obtiene el caudal utilizando el dato obtenido mediante una tabla experimental. El firmware compara el resultado del caudal obtenido con el set point. Si los valores son diferentes, el firmware genera una señal de tren de pulsos para ajustar la posición de la válvula de control. Se repiten los pasos anteriores hasta que el caudal obtenido sea igual al set point.
Flujo Alternativo	<ul style="list-style-type: none"> Si los valores son iguales, el firmware detiene la generación del tren de pulsos.
Requisitos Especiales	<ul style="list-style-type: none"> El firmware debe estar en estado de correcto funcionamiento. La comunicación serial debe encontrarse en condiciones óptimas.
Post-Condiciones	El firmware deberá quedar operativo para recibir futuros comandos.

III-B5. Diagrama de clases del firmware: Previo al desarrollo del firmware, se realizó un diagrama de clases [10], que se puede apreciar en la Fig. 17.

- Uart: Este módulo recibe y envía los comandos desde y hacia la aplicación de comunicación serial

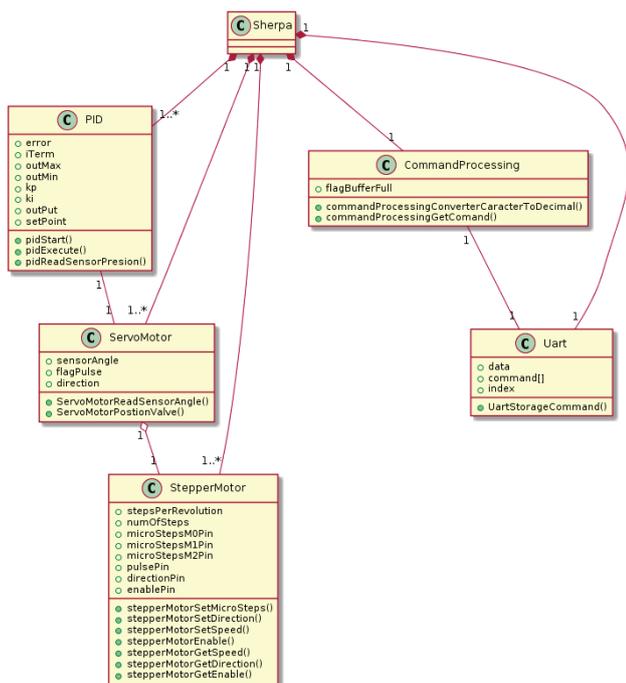


Figura 17. Diagrama de clases.

que se ejecuta en la PC.

- **CommandProcessing:** Acumula y convierte de carácter a decimal los comandos recibidos y los devuelve cuando es solicitado.
- **StepperMotor:** Este módulo posee todas las funciones inherentes al control del motor paso a paso. Interactúa con el módulo Servomotor para integrar las características de un servomotor convencional.
- **ServoMotor:** Este módulo tiene la misión de establecer el obturador en una posición determinada a través de la generación de una señal de tren de pulsos. Esto se realiza mediante una comparación entre la salida del algoritmo de control PID y la señal del sensor resistivo de ángulo. Cuando la comparación resulta igual a cero implica que alcanzó situar el obturador en la posición deseada
- **PID:** Implementa un algoritmo de control PID que se ejecuta cada un segundo para cuantificar el error o desviación que existe entre el valor de caudal medido a través del sensor de presión y el valor deseado.
- **Sherpa:** Es la clase central, que al momento de la ejecución está compuesta por al menos una o más instancias de las clases PID y StepperMotor y, además, por sola una instancia de commandProcessing.

III-B6. Protocolo de comunicación: Se definió un paquete de datos para que una aplicación externa y el firmware puedan interactuar para el funcionamiento del sistema. En la Tabla II se observan los comandos con su respectiva descripción.

Tabla II
Descripción de comandos.

Comandos	Descripción
ME	Deshabilita el motor.
MD	Habilita el motor.
MMSF	Establece el avance a pasos completos.
MMSH	Establece el avance a medio pasos.
MMS08	Establece el avance en fracciones de micropasos de 8.
MMS16	Establece el avance en fracciones de micropasos de 16.
MMS32	Establece el avance en fracciones de micropasos de 32.
MTA	Establece el sentido de giro antihorario del eje.
MTH	Establece el sentido del motor horario.
MS1201	Establece la cantidad de pasos. Cuatro dígitos para especificar el número de pasos, 1201.
MA0360	Establece el ángulo de giro del eje del motor. Cuatro dígitos para especificar el ángulo en grados, 360.
MFT0160	Establece la cantidad de vueltas completas. Cuatro dígitos para especificar el número de vueltas, 160.
SP100	Establece el valor del set point en porcentajes. Tres dígitos para especificar el valor del set point, 100 %.
NA	Devuelve el valor de altura del nivel del agua.

Al final de cada una de las tramas, se incorpora el carácter de salto de línea, que indica el límite final del comando. Una vez recibido el sistema lo procesa, de forma que si presenta un error, el firmware envía por el mismo medio una cadena de notificación “comando no válido”, caso contrario “comando válido”.

III-B7. Algoritmo PID: Durante el diseño del proyecto se identificó la necesidad de implementar un control de proceso con realimentación. La tarea específica del sistema de control es determinar y actualizar la posición de la válvula a medida que cambian las condiciones de carga. De esta manera, la variable controlada (caudal) alcanzará el valor deseado y se mantendrá en su entorno, incluso frente a perturbaciones externas. Con base en la necesidad expuesta, se incorporó al firmware un algoritmo de control PID [11] como una herramienta tecnológica que permite cuantificar el error o desviación que existe entre un valor medido y un valor deseado de caudal. La ecuación (5) describe la salida:

$$\text{Salida} = \frac{100}{BP} \left[e_0 + \frac{1}{I_t} \int e_0 dt - D_t \frac{dm}{dt} \right] \quad (5)$$

Donde:

- $\frac{100}{BP}$: Ganancia del sistema (en %).
- e_0 : Error o diferencia entre el set point y el valor medido.
- $\frac{1}{I_t}$: Tiempo de integración.
- D_t : Tiempo de derivación.

Este algoritmo está basado en el publicado por Brett Beauregard [12] y se modificó para satisfacer las necesidades concretas de este trabajo. Es relevante destacar que se seleccionó un modo de control que combina la acción integral con la acción proporcional, y elimina la

acción derivativa. La ecuación (5) queda de la siguiente forma:

$$\text{Salida} = \frac{100}{BP} \left[e_0 + \frac{1}{I_t} \int e_0 dt \right] \quad (6)$$

Cuando el objeto control es una válvula, la acción derivativa no se utiliza, dado que su propiedad es la de actuar en porcentajes elevados de salida frente a variaciones muy pequeñas o rápidas de error. Es la parte del algoritmo que, en el desarrollo teórico, se añadió al final de todo, para acelerar la respuesta. Por la naturaleza matemática del algoritmo, la acción de control de la parte derivativa es muy alta en porcentaje, cuando la velocidad del error es elevada, aunque en magnitud sea pequeña. Esto hace que ante cualquier variación rápida del error, la válvula reaccione abruptamente, abriéndose o cerrándose. Estos movimientos repetitivos y violentos, lógicamente pueden provocar daños a esta pieza. Por esta razón, la acción derivativa no se utiliza cuando el actuador del sistema de control es una válvula. Entonces, ante una rápida variación (por ejemplo ruido electromagnético) se establece un porcentaje de salida muy grande.

III-B8. Descripción general del firmware: El firmware emplea un algoritmo de control PI con el objeto de corregir el error que existe entre el set point y el valor de caudal medido. El resultado de procesar la señal realimentada del sensor de presión contra el set point preestablecido por el usuario, es la salida o acción de control. Luego, este valor de salida expresado en voltaje, se compara con la señal del sensor resistivo. De esta forma, se habilita y deshabilita la señal de tren de pulsos que es enviada al controlador con el fin de girar el eje del motor paso a paso en uno u otro sentido. Cuando ambas señales se anulan ante un segundo comparador, significa que el obturador alcanzó la posición requerida para suministrar el caudal deseado. En esta única ocurrencia se inhabilita la señal de tren de pulsos que genera el microcontrolador. En caso que la salida del algoritmo PI sea mayor que la señal del sensor resistivo, se habilita el giro del eje en sentido horario. En caso contrario se habilita el giro en sentido antihorario. La señal proveniente del sensor resistivo de ángulo, que también es retroalimentada, indica la posición actual del obturador conforme este varía, como se observa en la Fig. 7. Entonces, los tres casos fundamentales a considerar son:

- Si la salida del algoritmo de control PI es igual a la señal del sensor resistivo, se inhabilita la señal de tren de pulsos.
- Si la salida del algoritmo de control PI es mayor que la señal del sensor resistivo, se habilita la señal de tren de pulsos y el giro del eje del servomotor en sentido horario.
- Si la salida del algoritmo de control PI es menor que la señal del sensor resistivo, se habilita la señal de tren de pulsos y el giro del eje del servomotor en sentido antihorario.

Para el modelado del firmware se utilizó una metodología orientada a objetos. El inconveniente que se presenta al aplicar este paradigma de programación, es que el lenguaje C, comúnmente empleado en el desarrollo de sistemas embebidos, no es un lenguaje orientado a objetos. Como solución, se utilizó una técnica que aprovecha el uso de punteros a estructuras y permite emular de manera efectiva la programación orientada a objetos, utilizando el lenguaje C estándar [13]. Esta técnica considera el archivo de cabeceras (.h) como la interfaz pública de la clase y el archivo de código fuente (.c) como la implementación privada. La ventaja que ofrece implementar esta técnica es que permite generar un código que puede escalar muy fácilmente.

IV. Ensayos y resultados

En esta sección se detallan las pruebas realizadas durante el desarrollo del trabajo y finalmente se analizan los resultados obtenidos.

IV-A. Ensayo de funcionamiento y calibración del sensor de presión

Se llevó a cabo una prueba acerca del correcto funcionamiento y calibración del sensor de presión MPX5010DP. Para ello, se utilizó un osciloscopio como instrumento de medición de la señal derivada del sensor, una fuente de alimentación de 5 V para la energización del dispositivo y una manguera flexible de cristal con un diámetro externo de 5 mm, un diámetro interno de 3 mm y una longitud de 30 mm. Un extremo de ella se conecta a una de las espigas del sensor, y el otro se utiliza para introducir y evacuar agua mediante el empleo de una jeringa. En la Fig. 18, se ilustra la conexión del dispositivo transductor con la manguera flexible y la herramienta de medición. La

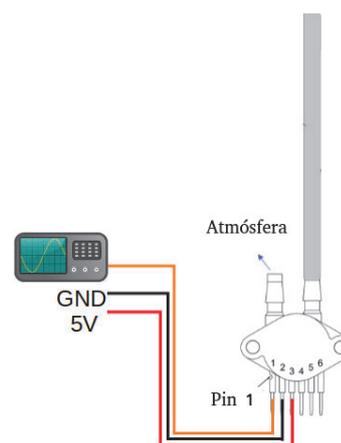


Figura 18. Herramientas utilizadas para la verificación del funcionamiento del sensor y obtención de una función patrón.

finalidad de este ensayo fue la adquisición de datos para comprobar que el sensor presente un comportamiento lineal, **Voltaje (V)** vs. **Presión (kPa)**, como lo especifica

su respectiva hoja de datos. A medida que se variaba la altura de la columna de agua de la manguera, se obtenía a la salida del sensor su correspondiente valor de voltaje. De esta forma, se generó una tabla con valores de **Voltaje (V)** vs. **Altura (cm)**, que podemos corresponder con la presión, mediante la (2). En la Tabla III, se puede apreciar que para cada valor de altura de columna de agua tiene asociado un valor de voltaje. A partir de los resultados obtenidos, como se puede observar en la Tabla III, se proyectó una gráfica que responde a una función lineal como se aprecia en la Fig. 19. Es importante notar en la gráfica, la presencia de un desplazamiento hacia arriba de la función en 0,250 V respecto al eje de las abscisas, tal como especifica la hoja de datos del sensor. Por lo tanto, para una altura de columna de agua de 0 cm el sensor proporciona a su salida 0,250 V. Tras este ensayo, se concluyó que el sensor presenta un correcto desempeño.

Tabla III
Adquisición de datos por la calibración del sensor.

Nº	Altura (cm)	Voltios (V)
1	29.1	1.52
2	28.1	1.48
3	25.25	1.40
4	23.6	1.34
5	21.9	1.26
6	20.1	1.18
7	17.82	1.06
8	16.5	1.03
9	15.4	0.96
10	14.6	0.90
11	12.7	0.84
12	10.7	0.72
13	8.3	0.66
14	5.55	0.51
15	3.5	0.46
16	1.1	0.36
17	0.7	0.30
18	0.65	0.26
19	0	0.25

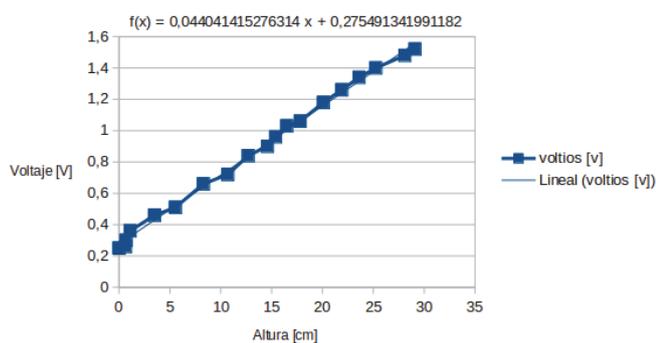


Figura 19. función patrón del sensor de presión calibrado: voltaje vs. altura.

IV-B. Calibración del sensor en el prototipo

Se realizó un ensayo del prototipo tras confirmar el correcto funcionamiento del sensor, en él se utilizó el

firmware para adquirir datos de presión, voltaje y altura del nivel del agua. Además, se midió el nivel real de agua con una regla. Esto permitió ajustar la altura de la columna de agua mediante una regresión lineal. La Tabla IV refleja los resultados obtenidos durante el experimento. Se cubrió el rango de altura comprendido entre el suelo del canal y el vértice del caudalímetro. Según los valores de la Tabla IV, la altura entre el suelo del canal y el vértice del caudalímetro es de 12,5 cm. Por lo tanto, si el nivel de agua no supera esta altura, el caudal es cero.

En la Tabla V se muestra que, cuando la altura del agua excede los 12,5 cm, se empieza a registrar un flujo de agua a través de la hendidura de la placa de aforo. Esto permite calcular el caudal en litros por minuto en función de la altura de la columna de agua. De la Tabla V, se puede deducir que los valores máximo y mínimo de caudal obtenidos son de 5,375 L min⁻¹ y 0,1 L min⁻¹ respectivamente. A partir de la información obtenida a través de las mediciones que componen la Tabla V, se extrajeron los valores correspondientes a las variables caudal y voltaje para construir la Tabla VI.

Tabla IV
Adquisición de datos del sistema I.

Nº	Presión (Pa)	Voltios (V)	Altura del sist. (cm)	Altura real (cm)
1	0.065	0.229	0.66	1
2	0.1	0.245	1.03	1.5
3	0.136	0.261	1.39	2
4	0.179	0.281	1.83	2.6
5	0.229	0.303	2.35	3
6	0.272	0.323	2.79	3.5
7	0.315	0.342	3.23	4.1
8	0.351	0.358	3.6	4.5
9	0.401	0.381	4.11	5.1
10	0.444	0.4	4.55	5.5
11	0.487	0.419	4.99	6.1
12	0.523	0.435	5.36	6.5
13	0.566	0.455	5.8	7
14	0.616	0.477	6.31	7.5
15	0.667	0.5	6.82	8
16	0.703	0.516	7.19	8.5
17	0.753	0.539	7.7	9.05
18	0.81	0.565	8.29	9.55
19	0.846	0.581	8.66	10
20	0.896	0.603	9.17	10.5
21	0.939	0.623	9.61	11
22	0.996	0.648	10.2	11.5
23	1.039	0.668	10.64	12
24	1.09	0.69	11.15	12.5

Estos datos se utilizaron para trazar una gráfica y verificar que responde a una función polinómica, como se describe en la (4). La gráfica obtenida, que se ilustra en la Fig. 20, confirma que el comportamiento del caudal en función del voltaje corresponde a una función polinómica de segundo grado. El diagrama del prototipo, que se muestra en la Fig. 21, incluye un depósito de agua conectado a una llave siempre abierta, seguido de una válvula de control con su motor paso a paso, controlador y parte del firmware, junto con un algoritmo de control PI. El canal, hecho de placas de madera impermeabilizadas,

Tabla V
Datos de adquisición del sistema II.

Nº	Presión (Pa)	Volt. (V)	Altura sist. (cm)	Altura real (cm)	Q (l/m)
25	1	0.706	11.52	13	0.1
26	1.19	0.735	12.18	13.6	0.3
27	1.269	0.771	12.91	14.4	1
28	1.348	0.806	13.79	15.4	2.25
29	1.419	0.839	14.53	16.1	3.74
30	1.455	0.855	14.89	16.5	4.25
31	1.47	0.861	15.04	16.6	4.75
32	1.491	0.871	15.26	17.1	5.375

Tabla VI
Datos obtenidos para construir función polinómica de segundo grado.

Nº	Voltios (V)	Q (l/m)
1	0.706	0.1
2	0.735	0.3
3	0.771	1
4	0.806	2.25
5	0.839	3.74
6	0.855	4.25
7	0.861	4.75
8	0.871	5.375

dirige el agua hacia un medidor de caudal compuesto por un vertedero con hendidura en V y un sensor de presión.

IV-C. Resultados de las pruebas

El sistema tiene un comportamiento naturalmente lento, ya que tarda aproximadamente 120 s en alcanzar el caudal máximo al abrir la válvula del 0% al 100% de forma manual e instantánea.

Esto se verificó en un ensayo específico. En la Fig. 22 se muestra que el sistema en lazo cerrado alcanza el estado de salida deseado.

En el ensayo, se fijó un set point del 35% del caudal máximo. El sistema demostró su capacidad para reducir errores, por lo que tardó unos 150 s en alcanzar el caudal deseado desde 0%. Además, el sistema respondió correctamente en pruebas realizadas con caudales al 25%, 50%, 75% y 100%.

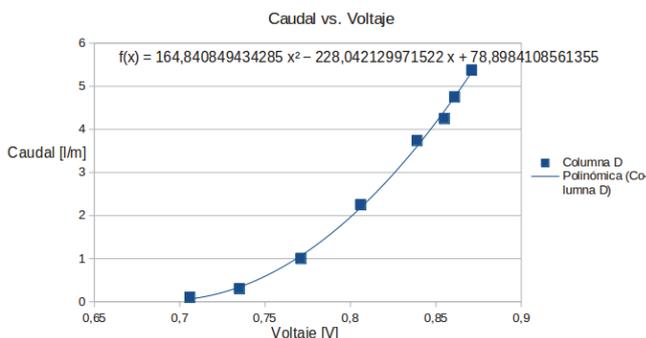


Figura 20. Función polinómica de segundo grado: caudal vs. voltios.

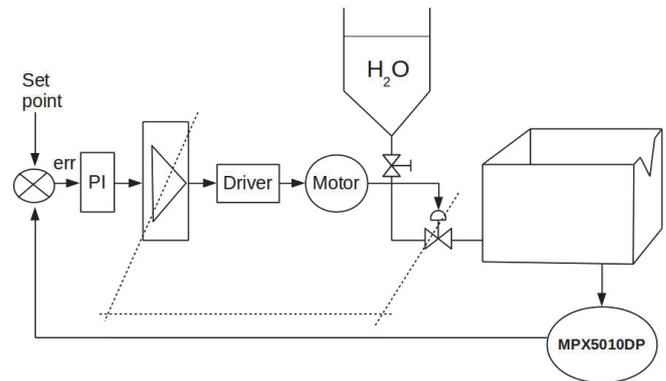


Figura 21. Diagrama en bloque del prototipo.

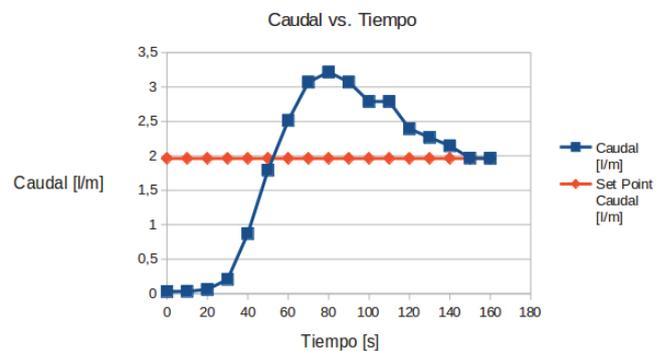


Figura 22. Función salida del sistema.

Las pruebas realizadas están documentadas en el video disponible en YouTube [14].

V. Conclusiones

En esta última sección se presentan las conclusiones generales sobre el rendimiento del sistema en su conjunto. Además, se destacan las posibles mejoras y los trabajos pendientes por realizar.

V-A. Resultados obtenidos

Se implementó exitosamente un prototipo de sistema de control de caudal en un canal a cielo abierto, validado mediante pruebas que confirmaron su correcto funcionamiento y permitieron extraer conclusiones importantes:

1. Se logró construir un medidor de caudal con mayor precisión.
2. Se desarrolló una electroválvula que opera el obturador de manera adecuada.
3. Se implementó un algoritmo de control PI para regular eficientemente el caudal a un valor predefinido.
4. El diseño y la codificación empleados en el desarrollo del firmware optimizan la escalabilidad de manera sencilla sin pérdida de calidad del sistema.

V-B. Próximos pasos

A continuación, se identifican los trabajos futuros necesarios para desarrollar un sistema integral capaz de controlar el caudal en múltiples compuertas de una red de canales y asegure un desempeño autónomo y eficiente. Dado que se trata de un prototipo es crucial comenzar con la adaptación del sistema a una compuerta a escala real en una red de canales existente. A continuación, se deberían realizar pruebas de campo para evaluar su desempeño general y, validar que cumple con los requisitos funcionales establecidos. El siguiente paso consistiría en desarrollar un sistema de software principal escalable, que permita integrar un gran número de 'celdas primarias', dado que la mayoría de las redes de canales de distribución de agua cuentan con numerosas compuertas. Este sistema también debería gestionar la comunicación entre las celdas primarias instaladas en cada compuerta mediante la tecnología LoRaWAN, con el fin de asegurar una operación autónoma. Asimismo, se debería desarrollar una aplicación móvil que permita gestionar el flujo de agua a través del sistema principal, así permitiría abastecer los canales y cumplir con las dotaciones de riego previamente establecidas.

Para una comprensión más profunda del tema y detalles adicionales sobre la metodología, resultados y análisis, se invita a los lectores a consultar la memoria completa del trabajo final de posgrado, disponible en [15].

- [13] B. P. Douglass, *Design Patterns for Embedded Systems in C*. Cambridge, MA, USA: Cambridge University Press, 2011.
- [14] L. M. Campos, "Demostración: Automatización y control de redes de distribución de agua," YouTube, 26 abr. 2021. [En línea]. Disponible en: <https://www.youtube.com/watch?v=MhEnYA7zCKA>. [Accedido: 30-ago.-2024].
- [15] L. M. Campos, "Automatización y control de redes de distribución de agua," Memoria de Trabajo Final de la Carrera de Especialización en Sistemas Embebidos, Facultad de Ingeniería, Universidad de Buenos Aires, Buenos Aires, Argentina, 2021. Disponible en: <https://lse-posgrados-files.fi.uba.ar/tesis/LSE-FIUBA-Trabajo-Final-CESE-Luis-Mariano-Campos-2021.pdf>.

VI. Bibliografía

Referencias

- [1] "Wikipedia: Canal de riego," Wikipedia, [En línea]. Disponible en: https://es.wikipedia.org/wiki/Canal_de_riego. [Accedido: 02-abr.-2024].
- [2] M. Basán Nickisch, "Aforadores de Corrientes de Agua," Ing. en Rec. Hidr. (M.Sc.), disponible en: https://www.produccion-animal.com.ar/agua_bebida/127-curso_aforadores_agua.pdf. [Accedido: 25-ago-2024].
- [3] Wikipedia, "Vertedero de Pared Delgada," disponible en: https://es.wikipedia.org/wiki/Vertedero_de_pared_delgada. [Accedido: 25-feb.-2020].
- [4] Proyecto CIAA. "Placa EDU-CIAA NXP," disponible en: <https://www.proyecto-ciaa.com.ar/devwiki/doku.php?id=desarrollo:edu-ciaa:edu-ciaa-nxp.html>. [Accedido: 25-jun-2019].
- [5] NXP Semiconductors, "MPX5010DP," disponible en: <https://www.nxp.com/docs/en/data-sheet/MPX5010.pdf>. [Accedido: 14-ago.-2020].
- [6] FreeRTOS™, "Real-time operating system for microcontrollers and small microprocessors," disponible en: <https://www.freertos.org/>. [Accedido: 26-jun.-2019]. visitado el 26 de junio de 2019.
- [7] I. Sommerville, *Ingeniería de Software*, 9ª ed. México: Pearson Educación, 2011, pp. 537–564.
- [8] Wikipedia, "Diagrama de flujo," disponible en: https://es.wikipedia.org/wiki/Diagrama_de_flujo. [Accedido: 02-abr-2019].
- [9] "Diagramando UML con PlantUML," 2019. [En línea]. Disponible en: https://pdf.plantuml.net/PlantUML_Language_Reference_Guide_es.pdf. [Accedido: 02-abr.-2019].
- [10] Wikipedia, "Diagrama de clases," disponible en: https://es.wikipedia.org/wiki/Diagrama_de_clases. [Accedido: 02-abr.-2019].
- [11] "Conceptos, terminología y Técnicas para el Control de Procesos," disponible en: http://infopl.net/files/documentacion/control_procesos/infoPLC_net_ControlProcesos.pdf. [Accedido: 15-may.-2020].
- [12] B. Beauregard, "Improving the Beginner's PID – Introduction," <http://brettbeauregard.com/blog/2011/04/improving-the-beginners-pid-introduction/>, marzo de 2011. [Accedido: 08-ago.-2020].

Implementación de un sistema embebido para la digitalización del registrador de trazos batitermográficos.

1st Walter Gabriel Correa

Facultad Regional Bahía Blanca
Universidad Tecnológica Nacional
Bahía Blanca, Argentina
Waltercorrea86@gmail.com

2nd Christian Galasso

Servicio de Análisis Operativo, Armas y Guerra
Base Naval Puerto Belgrano
Argentina
Christian_galasso81@yahoo.com.ar

Un registrador XBT (eXpendable BathyThermograph) por sus siglas en inglés, es un equipo capaz de medir el perfil de temperatura de una columna de agua de mar desde un barco en navegación sin que este afecte sus condiciones de operatividad. El presente trabajo tiene como objetivo presentar la implementación de un sistema embebido para la digitalización de las señales, que provienen de la sonda batitermográfica, basado en la placa de desarrollo STM32 Núcleo-144, además se implementa una etapa de acondicionamiento de las señales que provienen de la probeta para un posterior almacenamiento y análisis.

I. INTRODUCCIÓN

Un batitermógrafo descartable, XBT por sus siglas en inglés (eXpendable BathyThermograph), es un equipo analógico utilizado para medir el perfil de temperatura de una columna de agua en navegación sin afectar las condiciones de operación del buque. El sistema fue desarrollado en la década del setenta por la firma Sippican Inc. [1], hoy subsidiaria de Lockheed Martin y se sigue utilizando en la actualidad tanto en el área civil/científica como militar. El sistema consta de tres componentes principales:

- Sonda Descartable.
- Un lanzador que puede ser fijo o portátil.
- Registrador analógico.

La sonda se deja caer por la banda del buque, a medida que desciende a través de la columna de agua, mide y registra la temperatura. Pequeños cables transmiten los datos de temperatura al registrador analógico, donde se registran en papel para su análisis. Debido a que la sonda cae a una velocidad conocida, la profundidad de la misma se puede deducir desde el momento del lanzamiento con una ecuación provista por el fabricante.

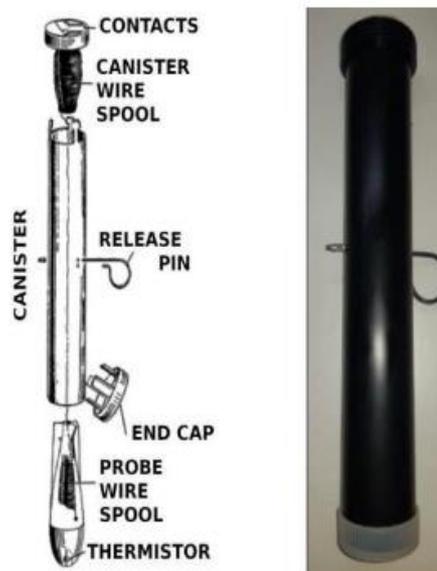


Fig. 1: Izq.: Descripción de cada una de las partes de un XBT. Der.: Canister XBT.

II. DISEÑO E IMPLEMENTACIÓN

El proyecto consiste en digitalizar las señales de tensión que provienen de la sonda XBT, esta digitalización incluye un enlace con otros sistemas de navegación, almacenamiento en memoria externa de la información de temperatura vs profundidad y un display para realizar ajustes y adiestramiento. Para ello se implementa un sistema embebido basado en el kit Núcleo de *STMicroelectronics* [2]. La digitalización se consigue censando las variaciones de tensión que proviene del XBT, producidas por los cambios de valores resistivos del termistor, por medio de un “circuito adaptador de niveles de tensión” que basa su funcionamiento en dos amplificadores operacionales, este último circuito adapta los niveles de tensión censado en un rango de tensión útil para ser leídos por un pin ADC (*Analog to digital Conversion*)

Luego de la conversión AD (*analógico a digital*) la placa Núcleo procesa esta información donde se obtienen los valores de temperatura y, además, se realiza el cálculo de la profundidad a la que se encuentra la sonda lanzada al agua. La información de profundidad y temperatura se almacenan en una memoria externa para su posterior análisis. También se cuenta con capacidad de integración a otros sistemas navales a través de un enlace asíncrono UART que permite el traslado de información obtenida *in-situ* a otro software de usuario para el cálculo de parámetros acústicos.

A. Etapa de acondicionamiento de la señal del XBT

Se diseñó el circuito mostrado en la fig. número 2 para lograr el acondicionamiento de las señales que provienen del circuito de conexión de señales al que se conecta la sonda XBT. Estos niveles de tensión deben adaptarse a un rango de tensión útil para un conversor AD de la placa Núcleo STM32F429zi. En este diseño la entrada principal son los niveles de tensión que provienen del circuito de conexión de señales, basa su funcionamiento en dos amplificadores operacionales realimentados. El primer amplificador es un seguidor de tensión y se utiliza para no cargar la etapa anterior, el segundo amplificador es quien fija, mediante

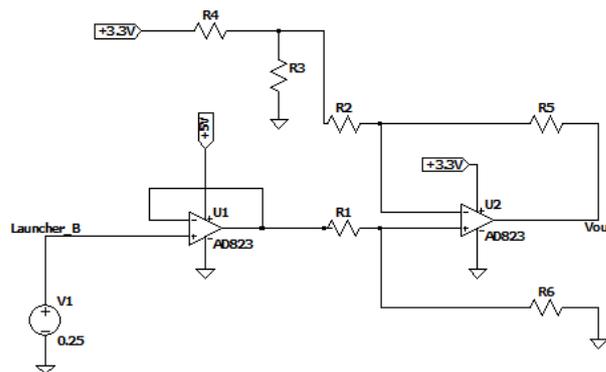


Fig. 2. Magnetización como una función del campo aplicado.

una red de resistencias, las señales de salida *Vout* que se dirigen al ADC.

B. Firmware

La arquitectura del firmware de control del registrador digital se estructuró mediante HAL (*Hardware Abstraction Layer*) del paquete STM32. Se implementó una máquina de estados finitos (MEF) para realizar la adquisición y procesamiento de la información y para manejar la comunicación hacia otros sistemas. En la fig. número 3 se muestra el ciclo de trabajo del XBT digital que se inicia en el estado **INIT** donde se realizan todas las inicializaciones de hardware y software necesario para establecer un canal de comunicación asíncrono y verificar la existencia de una memoria para realizar el registro cuando se esté realizando una medición de temperatura, luego la MEF pasa al estado **INACTIVO** donde permanecerá a la espera de un evento que será el lanzamiento de una sonda, cuando se quite el pin de seguridad de que asegura la sonda al canister, esta tocará el agua de mar cerrando un circuito masa en conjunto con el agua de mar y el casco del buque, en ese momento se produce una transición al estado de **PROCESAMIENTO** donde se convierten niveles de tensión analógicos en valores digitales representativos de temperatura y tiempo en profundidad mediante un timer asociado. Dentro de este estado se realiza el cálculo de la profundidad que posee el canister en el agua, esto se logra gracias a que se conoce la velocidad de caída del mismo porque lo provee el fabricante y se encuentra en función del tiempo.

XBT_Digital

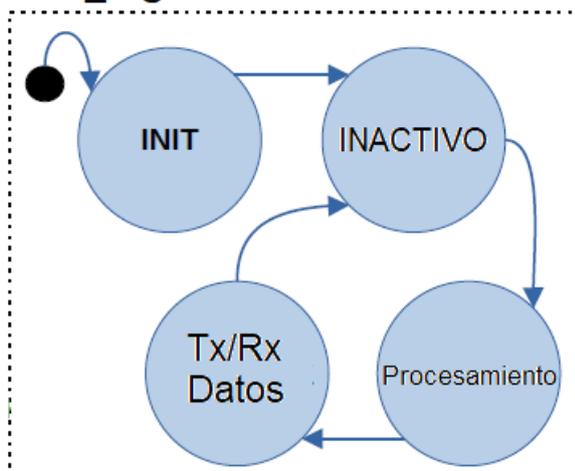


Fig. 3. Máquina de estados finitos del registrador digital

Finalmente, la MEF pasa al estado de **Tx/Rx Datos**, este último estado envía la información de temperatura vs profundidad por un puerto UART y almacena la información en memoria externa, cuando finaliza estas dos tareas pasa nuevamente al estado de INACTIVO a la espera de una nueva carga de canister y un nuevo lanzamiento de la sonda.

III. RESULTADOS

Los ensayos del prototipo permitieron generar un archivo en formato *.txt* que se almacenan en memoria externa. Se utilizó el software Matlab para graficar la curva de temperatura vs profundidad, estas curvas se muestran en la fig. número 4.

Para un microcontrolador que cuenta con un conversor AD de 12 bit la resolución es una parte en 4096 valores, la señal recibida por el ADC varía en un rango de 0 a 3 volts por lo que se obtiene un incremento detectable más pequeño de 732 μV . En la fig. número 4 se muestra el gráfico obtenido con el software Matlab donde se muestra el perfil de temperatura obtenido en un lanzamiento.

IV. CONCLUSIÓN Y DISCUSIÓN

Se logró diseñar e implementar un sistema embebido basado en un microcontrolador Cortex M4, este prototipo permite recuperar la capacidad de operativa de unidades de superficie que, por el paso de tiempo y limitación en la fabricación de repuestos, se ha ido perdiendo. Además, permite incrementar la capacidad de análisis de aquellas unidades que aún cuentan con el sistema analógico. La

digitalización del sistema para un posterior análisis y la interconexión con otros sistemas para el pasaje de información “*in-situ*” implica una mejora significativa en las condiciones de trabajo del usuario ya que este pasaje se realizaba en forma manual. El sistema también permite reducir el tiempo en que es procesada la información y mejora la confiabilidad dado que mejora la resolución respecto del registrador analógico.

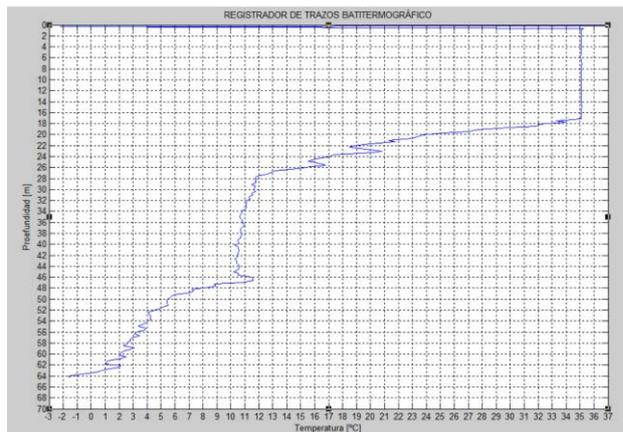


Fig. 4. Gráfico de profundidad vs temperatura

REFERENCIAS

- [1] <https://www.lockheedmartin.com/>
- [2] https://www.st.com/content/st_com/en.html
- [3] The Sippican Corporation, “INSTRUCTION MANUAL for the EXPENDABLE BATHY THERMOGRAPH SYSTEM”, R-603G, 1972.
- [4] STMicroelectronics, “Description of STM32F4 HAL and low-layer drivers”, User manual, UM1725, Rev 7 - junio 2021
- [5] Instrucciones hidrográficas #4. Empleo del batitermógrafo desechable (XBT). 1979

MS-MEGA: Sintetizador monofónico de 2 osciladores de onda cuadrada

Basado en el ATmega328p, con entrada MIDI y salida por 2 osciladores de onda cuadrada con afinación independiente y *gate*.

Pérez Cerutti, Valentín

Ingeniería Electrónica, Facultad de Ingeniería UNLP

La Plata, Argentina

pc.valen03@gmail.com

Abstract—El presente proyecto presenta la implementación de un sintetizador monofónico con un ATmega 328p (microcontrolador asignado por la cátedra Arquitectura de Computadoras FI UNLP) resuelto en los tiempos de la asignatura (2 meses), haciendo uso de todos los recursos disponibles y sentando las bases para un proyecto abierto y colaborativo.

I. INTRODUCCIÓN

A. Un poco de historia

Los sintetizadores son instrumentos musicales surgidos en la década del 70' cuya novedad fue producir sonidos a partir de un circuito electrónico en lugar de amplificar un sonido producido analógicamente como es el caso de la guitarra eléctrica. Sin embargo, el mayor problema que estos instrumentos presentaban era su estabilidad, ya que para controlar los osciladores se usaba una técnica denominada *control voltaje* (CV), la cual requería un teclado (piano) especial donde cada tecla debía ser manualmente calibrada para entregar la tensión correspondiente a la nota correspondiente. Como los osciladores analógicos (generalmente circuitos biestables) y el CV requerían mantenimiento constante y tenían una importante deriva térmica, lentamente se fue migrando a una tecnología híbrida: *computer controlled analog synthesis* [1] (síntesis analógica controlada por computadora).

En esta tecnología, el oscilador analógico es reemplazado por un microcontrolador con *timers*, y el protocolo de CV por el protocolo **MIDI** (Musical Instrument Digital Interface), en el cual un controlador MIDI (cuya interfaz física puede ser un piano o lo que el usuario elija) transmite por protocolo serial a 31.250 baudios mensajes de

- *Note On* (nota presionada) con el número de nota y la velocidad con la que fue pulsada,
- *Note Off* (nota soltada) con el número de nota, y
- otros mensajes de control como *pitch bend* y *mod wheel*.

Un sintetizador que resulta representativo de esta tecnología y que se utilizó como modelo para el desarrollo de este proyecto es el **Korg MS-20** (Fig. 1).



Fig. 1. Sintetizador monofónico Korg MS-20.

B. El proyecto **MS-MEGA**

De todo el sistema complejo que representa un sintetizador de esta magnitud, este proyecto se limitará únicamente a los 2 osciladores principales con sus respectivos modificadores, la señal de *gate* (que será utilizada por los siguientes módulos) y la interfaz MIDI, como se muestra en la Fig. 2.



Fig. 2. Banco de osciladores del Korg MS-20.

En esta primera instancia los osciladores serán únicamente de onda cuadrada, ya que se aprovechan las salidas de los timers del ATmega 328p. De todas maneras, se podrían conseguir fácilmente otras formas

de onda a la salida usando diversos filtros, integradores, o incluso sintetizando formas de onda con sub-armónicos de la onda cuadrada generada.

C. Limitaciones

Siendo este un proyecto de la cátedra Arquitectura de Computadoras de la Facultad de Ingeniería UNLP, el microcontrolador utilizado es el **ATmega 328p** [2], ya que así se solicita en la asignatura. ¿Es este el microcontrolador ideal para esta aplicación? Para nada. La aplicación requiere de una precisión mayor al 1% en sus 2 osciladores en el rango utilizable, mientras que el ATmega 328p cuenta con

- 1 timer de 16 bits que cumple con esta precisión en casi todo el rango
- 2 timers de 8 bits que cumplen con la precisión en sólo la primer octava.

II. IMPLEMENTACIÓN

A. Protocolo MIDI

En primer lugar, se escribió una biblioteca MIDI que, haciendo uso de la interfaz USART del ATmega 328p permite recibir los mensajes enviados desde un controlador MIDI externo. Para esto se consultó la página del protocolo [3] y se siguieron las especificaciones de la versión 1.0.

B. Osciladores

Para evitar el posible *jittering* que presentaría un oscilador virtual con los escasos recursos del ATmega 328p, se decidió hacer la síntesis exclusivamente vía hardware haciendo uso de los 3 timers que el microcontrolador dispone. Esto trae consigo un problema gigante: la resolución.

La relación N° de nota - frecuencia no es lineal, sino que responde a la fórmula

$$f(n) = 440 \cdot 2^{(n-69)/12}$$

lo cual hace que la exactitud de los osciladores vaya cayendo de *octava a octava* (octava: 12 notas - doble de frecuencia). En concreto, para un timer de 8 bits como el timer 0 y timer 2, el error pasa de $(2^7)^{-1} = 0,8\%$ para la primera octava, lo cual para el oído suena *afinado*, a $(2^6)^{-1} = 1,5\%$ para la segunda, lo cual ya se empieza a percibir *desafinado*. Considerando que el protocolo MIDI soporta más de 10 octavas, esto se vuelve inviable.

Todavía queda un recurso en los timers del ATmega, el cual permite dividir la frecuencia en múltiplos de 2 y así mantenerse siempre en el rango donde $e_f < 1\%$: ¡el prescaler!

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	clk _{ICP} (no prescaling)
0	1	0	clk _{ICP} /8 (from prescaler)
0	1	1	clk _{ICP} /64 (from prescaler)
1	0	0	clk _{ICP} /256 (from prescaler)
1	0	1	clk _{ICP} /1024 (from prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	clk _{T2S} (no prescaling)
0	1	0	clk _{T2S} /8 (from prescaler)
0	1	1	clk _{T2S} /32 (from prescaler)
1	0	0	clk _{T2S} /64 (from prescaler)
1	0	1	clk _{T2S} /128 (from prescaler)
1	1	0	clk _{T2S} /256 (from prescaler)
1	1	1	clk _{T2S} /1024 (from prescaler)

Fig. 3. Prescalers para el timer 0 y timer 2.

Pero todavía deja qué desear: como se ve en la Fig. 3, el timer 0 tendría la posibilidad de cambiar de prescaler cada 2 octavas en el mejor de los casos (de $clk_{ICP}/64$ a $clk_{ICP}/256$ y a $clk_{ICP}/1024$) por lo que se lograría un rango de casi 6 octavas con error del 1,5%. El timer 2 por otra parte presenta un muy buen rango entre $clk_{T2S}/32$ y $clk_{T2S}/256$ ya que el prescaler permite reproducir esas cuatro octavas con un error menor al 0,8%, pero el oscilador queda acotado sólo a ese rango.

Finalmente se llegó a una decisión de compromiso: usar el timer 2 como prescaler del timer 0, pudiendo así alcanzar las divisiones del clk_{ICP} que sean necesarias. A continuación se presenta la tabla de frecuencias usadas y su correspondiente nota:

TABLA I
TABLA DE FRECUENCIAS

MIDI N°	Nota	Frecuencia	timer
59	Si 3	246,9 Hz	252
60	Do 4	261,6 Hz	238
61	Do# 4	277,2 Hz	224
62	Re 4	293,7 Hz	212
63	Re# 4	311,1 Hz	200
64	Mi 4	329,6 Hz	189
65	Fa 4	349,2 Hz	178
66	Fa# 4	370 Hz	168
67	Sol 4	392 Hz	158
68	Sol# 4	415,3 Hz	149
69	La 4	440 Hz	141
70	La# 4	466,2	133

de esta manera se consigue mantener para todo el rango de notas utilizable la mejor precisión que el timer de 8 bits permite:

$$\frac{1}{2^7} = 0,8\%$$

Para lograr esto se seteó el timer 2 en modo CTC y se conectó externamente la salida OC2A a la entrada del timer 0 que hace de clock, T0 (Fig. 4).

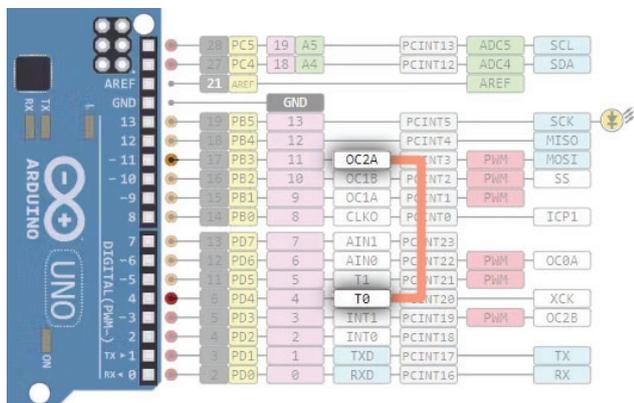


Fig. 4. Conexión externa del timer 2 al timer 0.

De esta manera queda conformado el oscilador 0, cuya afinación está fija a

$$f_{osc.0}(T_0, T_2) = \frac{16 MHz}{4 \cdot (T_0 + 1) \cdot (T_2 + 1)}$$

así por ejemplo se consigue

$$f_{osc.0}(141, 63) = 440, 14 Hz \text{ (La 4)}$$

con un error sistemático de 0,03% y no presenta deriva térmica medible (ver apartado de mediciones).

Por último el timer 1 constituye el oscilador 1, el cual se puede *desafinar* respecto del oscilador 0 y se le puede cambiar el ancho de pulso.

C. Efectos y agregados

Los controles que se ponen a disposición del usuario son:

- *Detune* del osc. 1
- Ancho de pulso del osc. 1
- *Glissando*
- Octava de ambos osciladores (-3 a +2)
- *Sync* de los 2 osciladores
- *Gate on-off* de los osciladores

El *glissando* se logra interpolando de manera dinámica la tabla de frecuencias anteriormente presentada, de acuerdo a la velocidad establecida por el usuario.

El *gate on-off* apaga los osciladores una vez que no se aprietan más teclas. Este efecto si bien sirve para tener un sintetizador básico sin componentes adicionales, no es deseado cuando se tiene una etapa de *VCA* (amplificador controlado por tensión) y/o *VCF* (filtro controlado por tensión) luego de los osciladores.

III. MEDICIONES

Se realizaron diversas mediciones sobre el proyecto, implementado con un Arduino UNO y los siguientes componentes:

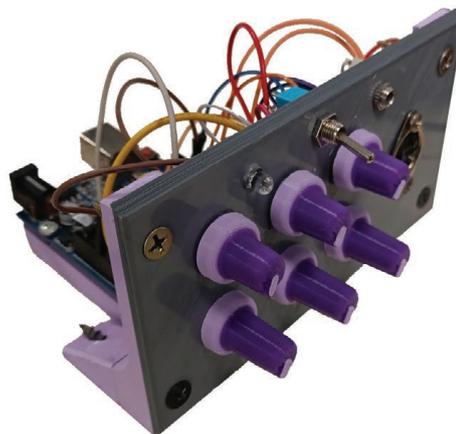


Fig. 5. Implementación del proyecto en un Arduino UNO.

- 6 potenciómetros: *detune*, *pw*, *oct 1 y 2*, *glissando* y un potenciómetro adicional que permite mezclar ambos osciladores a la salida,
- entrada MIDI con optoacoplador,
- LED indicador del *gate*,
- switch para la función de *sync*, y
- jack de 3,5 mm para salida de audio.

Sobre este set-up (Fig. 5) se realizaron mediciones de la frecuencia del oscilador 0, ya que es el que más error tiene respecto de la frecuencia real, y que va a ser copiado -a priori- por el oscilador 1 (luego se desafina respecto de ese punto). Para las mediciones se usó un multímetro de 30.000 cuentas Unigor 390.

Se determinó el error en frecuencia para cada una de las 12 notas de una octava, ya que luego el error relativo de cada nota va a ser el mismo en el resto de octavas; la variación de una única nota a medida que se aumenta la octava; y la deriva térmica.

A. Error relativo de las 12 notas

De esta manera se determinaron las siguientes frecuencias en el rango de una octava, indicando también el apartamiento de la frecuencia real de cada nota (Fig. 6):

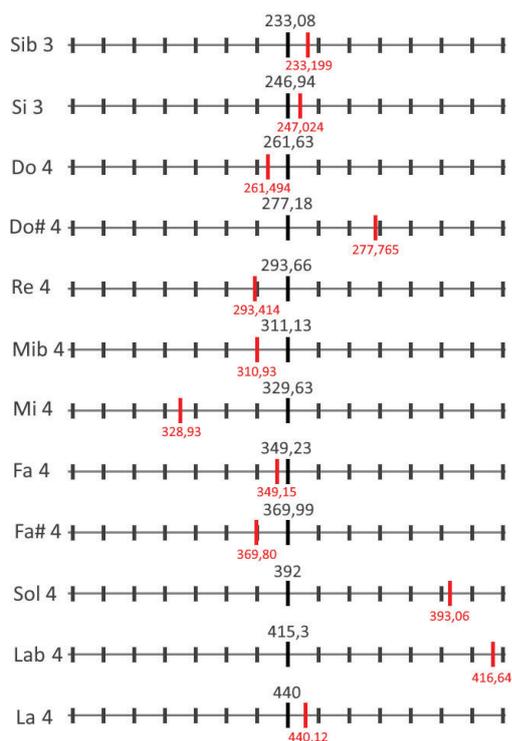


Fig. 6. Errores de frecuencia por nota, cada división equivale a 0,1 Hz.

El error relativo más grande se da para todos los La b y es de 0,32% (menor al 0,8% que el timer de 8 bits garantiza por su resolución).

B. Error por octava

A medida que el prescaler aumenta y el oscilador disminuye su frecuencia, los errores absolutos empiezan a cobrar mayor peso.

TABLA II
ERROR DE FRECUENCIA POR OCTAVA

Nota	La 1	La 2	La 3	La 4	La 5
f_{real} [Hz]	55	110	220	440	880
f_{medida} [Hz]	55,014	110,030	220,060	440,12	880,23

Por lo que se verifica que el error relativo se mantiene constante en las diferentes octavas, mientras que el error absoluto se duplica por octava porque se duplica la frecuencia.

C. Error por deriva térmica

Por último, tomando la frecuencia de La 5 (880,23 Hz) como referencia se procedió a calentar de manera controlada el cristal de cuarzo de 16 MHz que hace de reloj para el ATmega 328p. Sin embargo, no se observó deriva térmica apreciable en la frecuencia a una temperatura de $\sim 80^{\circ}C$.

IV. LLAMADO A LA COMUNIDAD

Este proyecto acerca enormemente el mundo de la síntesis analógica a los estudiantes, *hobbyistas* y amantes de la electrónica porque permite generar con una altísima exactitud algo que décadas atrás sólo era posible con circuitos integrados cuyo valor hoy superaría 50 veces el del microcontrolador utilizado. Sin embargo, este proyecto todavía tiene mucho potencial a futuro, y se hace un llamado a la comunidad para trabajar colectivamente y construir un recurso que cualquiera pueda acceder y utilizar.

A. Planes a futuro con el ATmega 328p

Si bien ya se exprimieron lo más que se pueden los 3 timers de este microcontrolador, todavía quedan un par de recursos y trucos para sacarle un poco más de jugo.

- *Daisy-chaining*: Siendo que el módulo de USART (usado para la recepción MIDI) dispone de una salida independiente de la entrada, una propuesta viable es encadenar varios de estos sintetizadores para lograr la polifonía (múltiples voces), haciendo que cada sintetizador reproduzca la primer nota que le llega hasta que esta sea soltada por el usuario y envíe por MIDI a el/los siguiente/s el resto de notas que le lleguen.
- *Secuenciador*: Usando un controlador MIDI que cumpla la función de maestro y proporcione un *clock* adicional (visto como pulsos el comando `0xf8` en el protocolo), se puede guardar una secuencia del largo especificado por el usuario y luego reproducirla nuevamente en el mismo sintetizador y/o enviarla nuevamente por MIDI
- *CV*: Con un circuito simple (diodo de protección, resistencia limitadora de corriente) se podía agregar la opción de que los potenciómetros sean reemplazados por una entrada de CV (control por tensión) para adicional módulos externos y así variar de forma dinámica los parámetros del sonido.

Es por esto que el proyecto puede ser encontrado en GitHub, junto con una configuración mínima para su testeo; y se publicarán allí futuras actualizaciones y adiciones. Por supuesto, también queda la puerta abierta para cualquier entusiasta de la música y la electrónica que desee contribuir con el proyecto.

github.com/pcvalen2003/MS-mega

B. Y si cambiamos de micro?

Otra propuesta viable es reemplazar el Arduino UNO R3 y su ATmega 328p por un Arduino UNO R4, el cual viene equipado con un microcontrolador RA4M1 (Cortex M4, ver Fig. 7).

RA4M1		48MHz 32-Bit Arm® Cortex®-M4 Core		FPU ARM MPU NVIC JTAG SWD ETB Boundary Scan	
Memory Code Flash (256KB) SRAM (16KB) Parity SRAM (16KB) ECC Data Flash (8KB)	Analogue 14-bit A/D (25ch) 15HS 12-bit DAC (1ch) OPAMP (4ch) Low Power Comparator (2ch) Temperature Sensor	Timers GPT 32-bit (2ch) GPT 16-bit (8ch) Low Power GPT (2ch) WDT RTC, Calendar, Vbat	HMI Capacitive Touch Sensing Unit (27ch) Segment LCD Controller 32 seg x 4 com 31 seg x 8 com	Communication USB2.0 FS x1 CAN x1 I2C x2 SCI x4 SPI x2 SSI x1	System DMA (6ch) DTC Clock Generation On-Chip Oscillator HCOO (04.32-48.5MHz) MOCO (8MHz) LOCO (32KHz) ILOCO (15KHz) Low Power Modes ELC Interrupt Controller
	Safety Memory Protection Unit SRAM Parity Check ECC in SRAM POE Clock Frequency Accuracy Measurement CRC Calculator WDT Data Operation Circuit Flash Area Protection ADC Self Test	Security AES (128/256) TRNG Key Management GHASH	Package LQFP 48, 64, 100 QFN 48, 64, 40, LGA 100		

Fig. 7. Periféricos del microcontrolador RA4M1.

V. CONCLUSIONES

Si con un microcontrolador tan austero con respecto a los periféricos como es el ATmega 328p se pudo resolver el problema planteado, queda demostrado que tomando cualquier otro micro con más recursos el potencial del proyecto es gigante.

BIBLIOGRAFÍA

- [1] P.J. Cheyne B.E.(Hons) (1978) *COMPUTER CONTROLLED MUSIC SYNTHESIS*, University of Canterbury.
- [2] Microchip Technology Inc. (2017). *ATmega328P Datasheet: 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash*. Recuperado de <https://www.microchip.com/wwwproducts/en/ATmega328P>
- [3] MIDI Association. (n.d.). *MIDI.org*. Disponible en <https://www.midi.org>

Generador de señales en HF para el desarrollo de una ionosonda

Leonardo J. Navarra
Grupo de Investigación y Desarrollo
en Radiofrecuencia e
Instrumentación
Facultad de Ciencias Astronómicas
y Geofísicas
Universidad Nacional de La Plata
Paseo del Bosque s/n B1900FWA
La Plata
lnavarria@fcaglp.unlp.edu.ar
ORCID 0000-0002-1061-4896

Ricardo E. García
Grupo de Investigación y Desarrollo
en Radiofrecuencia e
Instrumentación
Facultad de Ciencias Astronómicas
y Geofísicas
Universidad Nacional de La Plata
Paseo del Bosque s/n B1900FWA
La Plata
regarcia@fcaglp.unlp.edu.ar

Guillermo D. Rodríguez
Grupo de Investigación y Desarrollo
en Radiofrecuencia e
Instrumentación
Facultad de Ciencias Astronómicas
y Geofísicas
Universidad Nacional de La Plata
Paseo del Bosque s/n B1900FWA
La Plata
grodriguez@fcaglp.unlp.edu.ar

Resumen— El objetivo del trabajo fue desarrollar un sistema de generación de señales en la banda de HF (alta frecuencia) para ser empleado en una ionosonda de bajo costo (full custom). Se desarrollan aquí los conceptos fundamentales del sondeo remoto, como así también las soluciones de bajo costo implementadas para poder construir un transmisor de frecuencia variable.

Finalmente se mostrará el circuito terminado junto con la solución probada y medida en laboratorio.

Palabras Claves—radiosondeo, generación, diseño, simulación

I. INTRODUCCIÓN

La ionosfera es un plasma que se caracteriza por su perfil de densidad electrónica, el cual se encuentra estratificado debido a una combinación de efectos: la distribución de especies atómicas con la altura y la penetración de agentes ionizantes, tal como muestra la figura 1.

Se denomina plasma al cuarto estado de agregación de la materia, siendo este similar al gaseoso aunque con la particularidad de poseer partículas cargadas eléctricamente y no ende, sin equilibrio electromagnético.

Con estas características, la ionósfera actúa como una capa conductora para las ondas electromagnéticas que inciden en ella y poseen longitudes de onda muy superiores a la distancia media entre las moléculas que la conforman. En términos prácticos, se puede decir que las ondas electromagnéticas (OEM) que poseen frecuencia por debajo de los 30 MHz son reflejadas por la ionósfera, y la magnitud de la frecuencia a la cual suceden

tales reflexiones depende de la densidad electrónica en la misma

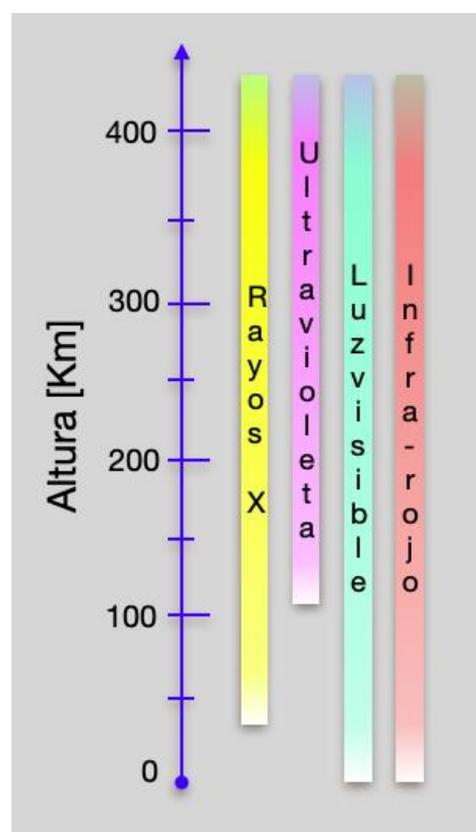


Figura 1: distribución de especies atómicas en función de la altura

II. FUNCIONAMIENTO DE UNA IONOSONDA.

La Ionosonda es un radar que envía pulsos electromagnéticos de diferentes frecuencias hacia la ionosfera, recibe el eco o reflexión de esta onda, midiendo el tiempo de viaje de la onda. A partir de este dato, conocida la frecuencia de los pulsos enviados y asumiendo

la velocidad de propagación, es posible determinar la distribución de la densidad electrónica con la altura

La técnica de sondeo ionosférico comenzó a utilizarse en 1925 con un sistema desarrollado por Breit y Tuve [1] aunque el conocimiento de la reflexión de las OEM en la ionosfera ya era conocido desde años antes.

El producto que se obtiene de un sondeo vertical es el ionograma, el cual se muestra en la figura 2, y dónde el eje vertical representa la altura a la cual se produjeron los ecos y el horizontal la frecuencia de la onda incidente (transmitida).

III. GENERACIÓN DE PULSOS

La posibilidad de tener un eco detectable depende de la energía del pulso emitido (el cuál es proporcional a la longitud del pulso). Por otro lado, pulsos más largos atentan contra la resolución del estudio. Como solución a este compromiso, se codificaron los pulsos con N bits empleando modulación BPSK, utilizando luego técnicas de correlación para la detección. Esto posibilita que la resolución solo sea la del tiempo de un bit y no la de todo el pulso

El objetivo de la ionosonda en desarrollo es obtener una resolución de 5 Km. Considerando (ec. 1) que la señal viaja a la velocidad de la luz © y recorre el trayecto (ΔR) dos veces (ida y retorno), podemos determinar un requerimiento de tiempo de bit (T_b) de 33 μ s.

$$2 * \Delta R = c * T_b \quad (1)$$

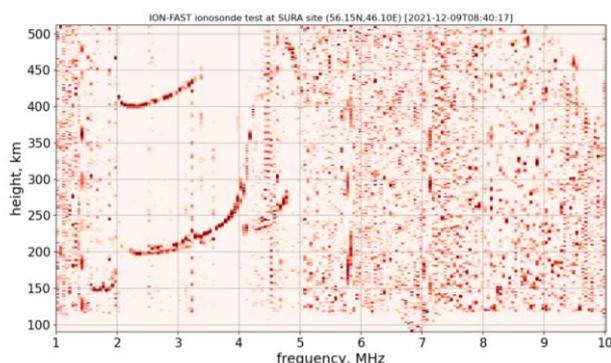


Figura 2: Resultado obtenido del sondeo vertical

Estos tiempos deben contemplar el tiempo del envío del pulso sumado al tiempo en que el sistema tarda en procesar las instrucciones, siendo esto el tiempo total de retardo.

IV. IMPLEMENTACIÓN DEL NUEVO SISTEMA

A. Desarrollo del generador

Para poder generar una señal de alta frecuencia, acorde a las necesidades se empleó el Sintetizador DDS AD9850 disponible en el mercado local. Este equipo posee tecnología DDS (síntesis digital directa), la cual es una técnica utilizada en los sistemas de comunicaciones actuales (inalámbricos, satelitales, etc). Los dispositivos DDS se han convertido en una alternativa cada vez más utilizada a las soluciones analógicas tradicionales de síntesis de frecuencias, debido que ofrecen ventajas de rendimiento, como facilidad de sincronización para obtener señales en cuadratura, otras relaciones de fase entre señales, y una transición de frecuencia más rápida, entre otras.

Los DDS de la firma Analog Devices [2] emplean en cada canal, un convertidor digital-analógico de 10 bits, para obtener un frecuencia estable y controlable. El AD9850 proporciona una resolución de 0.0291 Hz con un reloj de entrada de 125 MHz y un incremento de fase de 11.25°. Posee un comparador de alta velocidad junto con un registro de 40 bits utilizados para programar el control de la frecuencia, la fase y el encendido. Los 40 bits se componen por 32 bits para el control de la frecuencia, 5 bits para el control de fase y los bits restantes para el control del registro.

El AD9850 permite la programación en serie, a través de SDI o de forma paralela, cada una de las formas de programación posee tiempos de respuesta diferentes, lo que es requisito indispensable para los tiempos de sondeo necesarios. Sumado a las características anteriores se debe tener en cuenta el valor de impedancia de salida, que es de 120 k Ω , valor que debe tenerse en consideración para acoplar la salida al transmisor.

La tensión de salida del AD9850 [3] es del orden de los milivolts, por lo que se necesita amplificar de forma variable la señal generada para ser enviadas por el transmisor de alta potencia.

Para poder obtener un ajuste fino de la señal de entrada al transmisor de potencia y poder acoplar impedancias, se empleó un amplificador operacional de alta corriente de salida. En esta

aplicación el utilizado fue el AD815 [4] que se caracteriza por soportar hasta 40 V pico a pico de salida con 120 MHz de ancho de banda y 70 ns de retardo para las operaciones demandadas.

B. Simulación de circuitos

Como su nombre lo indica un simulador es una herramienta para efectuar en forma sistemática cálculos de circuitos eléctricos y electrónicos, de acuerdo con ciertos modelos que en muchos casos el usuario puede modificar o crear. El simulador no reemplaza los conocimientos de los circuitos analizados o de los temas tratados. El usuario del simulador, en este caso el estudiante, debe utilizar su sentido crítico que le permita juzgar sobre la racionalidad o viabilidad de los resultados obtenidos. Debe tenerse presente que el simulador no hace otra cosa que representar con modelos determinados a los componentes de un cierto circuito o sistema, y por ende las soluciones no pueden ir más allá de lo que dichos modelos permiten. [5]

Para verificar el funcionamiento del circuito se empleó el simulador de circuitos LTSPICE¹, provisto por la empresa de tecnología Analog Devices que se caracteriza por ser de distribución libre y gratuita. El simulador se encuentra basado en el software denominado SPICE (Simulation Program with Integrated Circuit Emphasis) y es el encargado de realizar el funcionamiento del circuito de interés antes de realizar el prototipo de este. Esta posibilidad permite reducir errores, costos y mejorar la performance de cualquier diseño. Este entorno de simulación permite realizar diversos tipos de análisis sobre el circuito y sus componentes: temporal, en frecuencia, determinación del punto de operación de reposo, de Fourier, evaluación de los efectos de ruido, optimización de parámetros, etc.

Para el caso en estudio se utilizará como fuente de señal al AD9850 generando una señal de hasta 10 MHz y que tiene como impedancia de salida en 120 kΩ. En la figura 3 se muestra el circuito construido para adaptar impedancias y niveles. El amplificador AD815 nombrado anteriormente es el encargado de amplificar la

señal proveniente del generador. El resistor Rvar es de valores variables (trimmer) y permite ajustar de forma fina la señal de salida al Resistor de carga Rc, en este caso siendo el modelo de la etapa de potencia de la ionosonda.

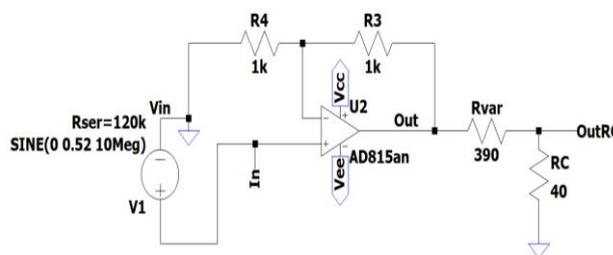


Figura 3: Circuito implementado en simulador

En la figura 4 se puede observar en color rojo la señal de entrada (salida del AD9850),

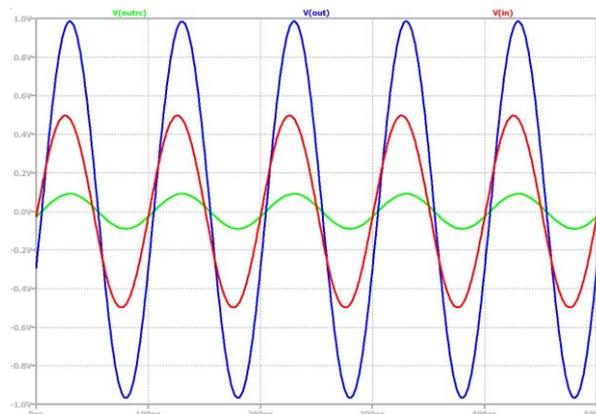


Figura 4: Señales simuladas

en color azul la salida del operacional (amplificación y adaptación de impedancias) y en verde la señal ajustada de entrada al transmisor de potencia.

C. Construcción

Como mencionamos anteriormente el oscilador AD9850 contiene un registro de 40 bits que permite programar el valor de la frecuencia y el valor de la fase. La carga del registro puede ser realizada enviando los datos en modo serie o paralelo. En la figura 5 se muestra la carga de datos en forma paralela, donde cada palabra se encuentra representada por W (0..4).

¹ <https://www.analog.com/en/resources/design-tools-and-calculators/ltspice-simulator.html>

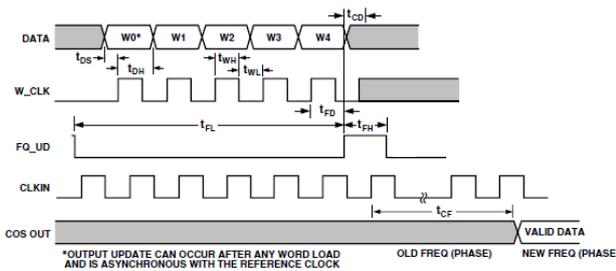


Figura 5: Envío de datos

En modo paralelo, se cargan las cinco palabras (32 bits) y son actualizados mediante la señal FQ_UD con un único pulso.

Para el caso del envío de datos en modo serie (fig. 6), los datos se actualizan con una subida de pulso en la señal FQ_UD (frequency update) con una demora de 40 ciclos de reloj para la actualización.

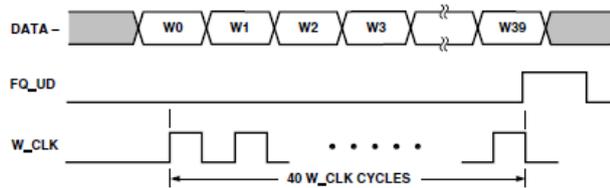


Figura 6: Carga de datos en modo serie

En la figura 7 se muestra la placa de desarrollo utilizada HCSR08 que contiene el AD9850 junto con un oscilador de 125 MHz, además de un potenciómetro para regular el ciclo de trabajo de la onda rectangular.

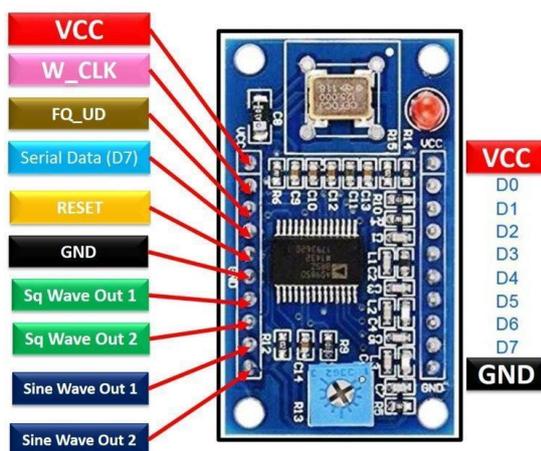


Figura 7: Pinout placa de desarrollo HC-SR08

Evaluando las opciones para poder realizar el control de la placa, se decidió utilizar un dispositivo de la plataforma Arduino®. En una primera etapa de desarrollo del proyecto se optó

por realizar la carga de datos en formato serie utilizando Arduino UNO, a través de un pin digital pero el tiempo logrado para la carga de bits no cumplía con la resolución deseada necesaria por el tiempo de generación de pulsos, ya que entre ciclos de carga se midió un tiempo de retraso de 700 microsegundos (μ s). Este resultado implicaba una resolución mínima de 105 Km, muy distante de la pretendida. Como segunda opción se probó utilizar el puerto I²C (SPI) que logró reducir los tiempos hasta 38 microsegundos (μ s) arrojando como resultado unos 6 kilómetros de resolución, tiempo que se encuentra al límite del objetivo buscado Finalmente se optó utilizar el Arduino MEGA que posee tres puertos paralelos disponibles de 8 bits para ser programados. Utilizando esta configuración, se posibilita llevar los tiempos de carga a tan solo 1,17 microsegundos (μ s), pudiendo de esta forma poder modificar los cambios de bit BPSK en los 33 (μ s de objetivo. En el apartado siguiente se muestra un extracto del código de programación para la carga de los datos.

```
PORTC = B10000000; // FQ_UD = high; w_clk = low
PORTC = B00000000; // FQ_UD = low; w_clk = low
```

```
PORTA = 0x00; // DDS W0
PORTC = B00000001; // w_clk
PORTC = B00000000; // w_clk
```

```
PORTA = 0x14; // DDS W1
PORTC = B00000001; // w_clk
PORTC = B00000000; // w_clk
```

```
PORTA = 0x71; // DDS W2
PORTC = B00000001; // w_clk
PORTC = B00000000; // w_clk
```

```
PORTA = 0xC3; // DDS W3
PORTC = B00000001; // w_clk
PORTC = B00000000; // w_clk
```

```
PORTA = 0xC0; // DDS W4
PORTC = B00000001; // w_clk
PORTC = B00000000; // w_clk
```

```
PORTC = B10000000; // FQ_UD High; w_clk = low
```

Figura 8: extracto de código de programación

Como se comentó anteriormente, para poder lograr la adaptación de impedancias y niveles se utilizó un AD815an con lo que se diseñó (fig 9) la tarjeta electrónica en el software Altium® y luego el montaje de componentes (fig. 10)

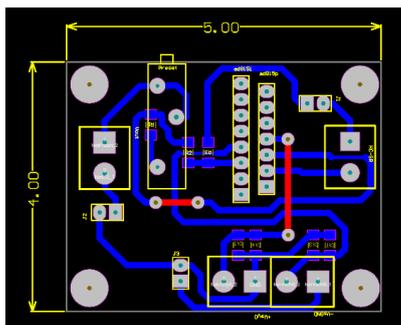


Figura 9: diseño de la tarjeta para amplificador

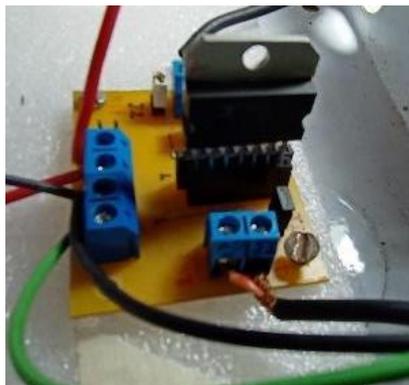


Figura 10: Montaje final de componentes

D. Prueba

Para realizar las pruebas de funcionamiento se realizaron varios barridos de frecuencia y se analizaron tanto las señales de entrada del oscilador como las señales de salida del amplificador operacional. De manera conjunta se controlaron los tiempos de los pulsos para mantener la resolución requerida.

En la figura 11 se muestran las señales generadas por la placa de desarrollo HC-SR08 comandada por Arduino. Las señales son valores de frecuencia distintos y poseen diferentes amplitudes.

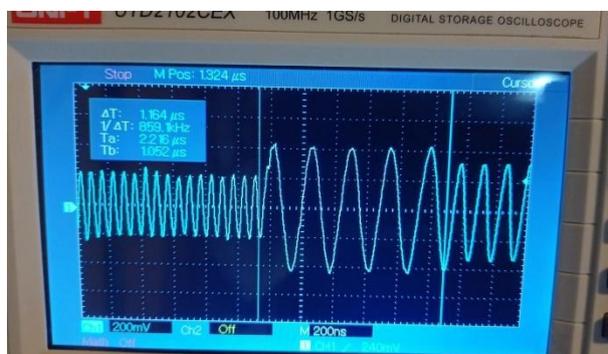


Figura 11: Generación de señales

En la figura 12 se observan los prototipos interconectados para el ensayo

V. CONCLUSIONES Y TRABAJO A FUTURO

En esta etapa del trabajo se ha logrado implementar un sistema que permite generar códigos con modulación BPSK, de largo y ancho de bit programable, para distintas frecuencias en la banda de HF. Los parámetros son seleccionables por el usuario, en forma sencilla y eficaz, para ser transmitidos por la etapa de potencia.

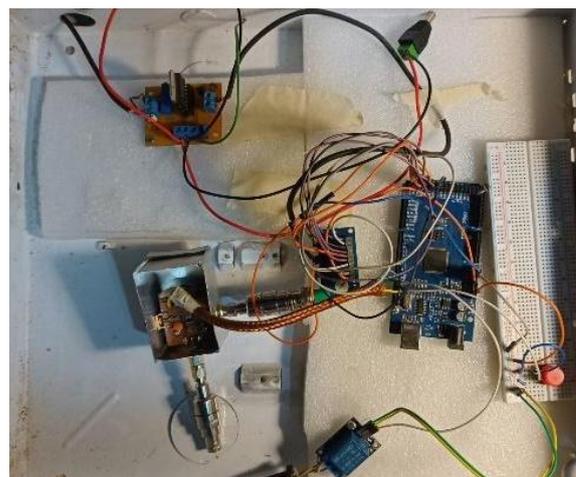


Figura 12: Montaje de componentes en la caja

Cómo línea de trabajo futura se puede mencionar que se está desarrollando el sistema de recepción para poder concluir con el proceso completo de radiosondeo de la ionósfera.

REFERENCIAS

[1] G. BREIT AND M. A. TUVE, "A TEST OF THE EXISTENCE OF THE CONDUCTING LAYER," *PHYS. REV.*, VOL. 28, NO. 3, PP. 554–575, SEP. 1926, DOI: 10.1103/PHYSREV.28.554.

[2] <https://www.analog.com/en/index.html>

[3] [HTTPS://WWW.ANALOG.COM/MEDIA/EN/TECHNICAL-DOCUMENTATION/DATA-SHEETS/AD9850.PDF](https://www.analog.com/media/en/technical-documentation/data-sheets/ad9850.pdf)

[4] [HTTPS://WWW.ANALOG.COM/MEDIA/EN/TECHNICAL-DOCUMENTATION/DATA-SHEETS/AD815.PDF](https://www.analog.com/media/en/technical-documentation/data-sheets/ad815.pdf)

[5] NAVARRIA, L. J. (2023). LABORATORIO VIRTUAL DE ELECTRÓNICA BÁSICA PARA ALUMNOS UNIVERSITARIOS DENTRO DE AULA EXTENDIDA (DOCTORAL DISSERTATION, UNIVERSIDAD NACIONAL DE LA PLATA).



Diseño e implementación de una placa de desarrollo como herramienta educativa

1st Wilberth López
Ingeniería Electrónica
Universidad Técnica Nacional
Alajuela, Costa Rica
0009-0001-9561-5918

2nd Gabriel Fernández
Ingeniería Electrónica
Universidad Técnica Nacional
Alajuela, Costa Rica
0009-0007-7285-9171

3rd Anthony Quesada
Ingeniería Electrónica
Universidad Técnica Nacional
Alajuela, Costa Rica
0009-0002-8827-0357

Abstract—Este informe presenta el diseño y la implementación de una placa de desarrollo basada en el microcontrolador PIC16F877A, que sirve como herramienta educativa para facilitar la enseñanza y la experimentación en electrónica y programación de microcontroladores. La placa incluye una variedad de componentes, como LEDs, interruptores, potenciómetros, entre otros, lo que permite a los estudiantes realizar diversas prácticas y experimentos. Asimismo, se detalla el proceso de diseño esquemático, la creación de la placa de circuito impreso y su implementación física.

Index Terms—PIC16F877A, microcontroladores, placa de desarrollo, herramienta educativa, innovación educativa

I. INTRODUCCIÓN

La supervisión de tareas mediante microcontroladores es fundamental tanto en la industria como en la vida cotidiana. En el mundo, muchas empresas buscan automatizar procesos de manera eficiente y económica, lo que ha generado una creciente demanda por soluciones basadas en microcontroladores. Esto ha convertido al diseño y la implementación de estos sistemas en un pilar esencial en la formación de ingenieros electrónicos, quienes deben estar capacitados para desarrollar y aplicar estas tecnologías en un mercado cada vez más competitivo.

En este informe, se realiza el diseño y el ensamble de una placa de desarrollo basada en el microcontrolador PIC16F877A, una herramienta esencial para la enseñanza y experimentación en electrónica y programación de microcontroladores. Esta placa permite programar y depurar el microcontrolador, además de incluir una variedad de componentes adicionales que facilitarán la realización de múltiples prácticas y experimentos.

La placa está equipada con LEDs, pulsadores, resistencias y otros componentes importantes, permitiendo a los estudiantes explorar diversas aplicaciones y configuraciones del PIC16F877A. El objetivo es crear una herramienta educativa integral que potencie las capacidades de aprendizajes y experimentación de los estudiantes, promoviendo el

desarrollo de habilidades técnicas y la innovación en el campo de la programación de microcontroladores.

II. OBJETIVOS

A. Objetivo General

Diseñar una placa de desarrollo basada en el microcontrolador PIC16F877A que sirva como herramienta educativa para la enseñanza y experimentación en microcontroladores.

B. Objetivos Específicos

- 1) Evaluar las características principales del PIC16F877A.
- 2) Desarrollar el diseño esquemático de la placa de desarrollo.
- 3) Diseñar la placa de desarrollo de acuerdo con el diseño esquemático.
- 4) Implementar físicamente la placa de desarrollo.
- 5) Evaluar la funcionalidad y rendimiento de la placa.

III. FUNDAMENTO TEÓRICO

La formación de ingenieros y técnicos competentes en el campo de la electrónica, específicamente en la programación de microcontroladores, requiere un enfoque práctico, donde las actividades en laboratorio juegan un papel esencial. Las prácticas permiten a los estudiantes aplicar conceptos teóricos en un entorno controlado, desarrollar habilidades teóricas y enfrentarse a la resolución de problemas de manera directa. Sin embargo, la eficiencia de estas prácticas depende en gran medida de las herramientas disponibles. En este sentido, la creación de una placa de desarrollo basada en un PIC se presenta como una solución eficaz para optimizar el tiempo de laboratorio, permitiendo así que los estudiantes realicen un mayor número de prácticas y profundicen en su aprendizaje.

Las placas de desarrollo ya hechas basadas en microcontroladores, como los PIC, han transformado la enseñanza de la electrónica al ofrecer una plataforma integrada que facilita la implementación de aplicaciones complejas sin la necesidad de configurar circuitos desde cero en cada sesión. Además, el uso de un PIC, conocido por su versatilidad y bajo costo, permite a los estudiantes enfocarse en la lógica de programación y el desarrollo de aplicaciones prácticas, en lugar de dedicar tiempo a tareas repetitivas como el montaje y configuración del hardware. Esto es particularmente importante en un entorno educativo, donde el tiempo es un recurso limitado y valioso.

Por lo tanto, uno de los beneficios más evidentes de utilizar una placa de desarrollo es la drástica reducción del tiempo necesario para preparar y ejecutar una práctica. En un escenario tradicional, la configuración de un circuito requiere tiempo para la búsqueda y montaje de los componentes, verificación de conexiones y solución de posibles errores. Este proceso, si bien es cierto tiene su nivel educativo, pero consume tiempo que podría ser empleado de una mejor manera en la exploración de conceptos más avanzados o en la realización de experimentos adicionales, y cabe señalar que los conceptos de conexiones y demás son abarcados en otros niveles inferiores de la carrera. Entonces, como se mencionó anteriormente, con una placa de desarrollo, los estudiantes pueden iniciar sus prácticas casi de inmediato, lo que reduce significativamente el tiempo total dedicado a cada sesión y permite realizar un mayor número de prácticas en el mismo periodo de tiempo.

A continuación se presenta una gráfica tomada de un artículo en la cual se puede visualizar un estudio que realizaron sobre el tiempo invertido en una práctica con tarjeta y sin ella:

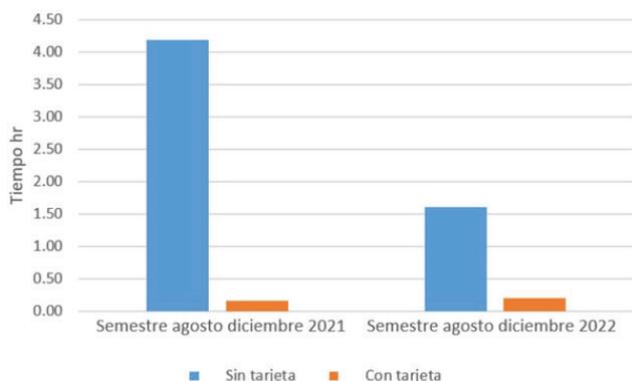


Fig. 1. Tiempo invertido en prácticas. [1]

La gráfica muestra una clara comparación entre el tiempo invertido en dos periodos semestrales, específicamente de agosto a diciembre de 2021 y de agosto a diciembre de 2022, bajo dos condiciones diferentes: con y sin placa. Durante el semestre de agosto a diciembre

de 2021, se observa que el tiempo invertido sin tarjeta es significativamente mayor, rondando las 4 horas, mientras que con tarjeta, el tiempo se reduce drásticamente a aproximadamente 0.25 horas. Este patrón de eficiencia se mantiene en el semestre siguiente, donde el tiempo sin tarjeta disminuye a cerca de 2 horas, lo que sugiere una mejora en el proceso. Sin embargo, el tiempo con tarjeta permanece casi constante, ligeramente por encima de 0.25 horas, lo que indica que la tarjeta es consistentemente eficiente. La idea de este análisis, es observar como el uso de una placa de desarrollo es efectiva para ahorrar tiempo que puede ser utilizado en prácticas más complejas.

Ahora, en la actualidad, tanto instituciones educativas como empresas de microcontroladores están intensificando sus esfuerzos para desarrollar placas de desarrollo basadas en este componente. El auge de estas iniciativas responde a la necesidad de contar con plataformas de desarrollo versátiles, que permitan experimentar y prototipar de manera rápida y eficiente. En un entorno donde la automatización y la conectividad son cada vez más relevantes, las placas de desarrollo basadas en microcontroladores se han convertido en un recurso indispensable para profundizar más el aprendizaje y optar por la creación de dispositivos inteligentes, sistemas embebidos y aplicaciones IoT (Internet de las Cosas).

[2] Las instituciones educativas están a la vanguardia en este campo, desarrollando placas que no solo cumplen con los requerimientos de enseñanza, sino que también se alinean con las necesidades reales de la industria, como por ejemplo, el Instituto Tecnológico de Celaya con la plataforma de entrenamiento para PIC, y otros. Estas colaboraciones de educación y sector productivo no solo facilitan la formación de profesionales más capacitados, sino que también contribuyen al fortalecimiento de la innovación tecnológica en el mundo.

[3] Por otro lado, las empresas tecnológicas como Microchip con la Curiosity también están invirtiendo considerablemente en el diseño y producción de estas placas, conscientes de su potencial para impulsar la competitividad y la eficiencia en distintos sectores. A través de la creación de plataformas accesibles y personalizables, fomentan un ecosistema de desarrollo que promueve la creatividad y la implementación de soluciones innovadoras.

IV. RESULTADOS OBTENIDOS

En primera instancia, se realizó el diseño esquemático de la PCB:

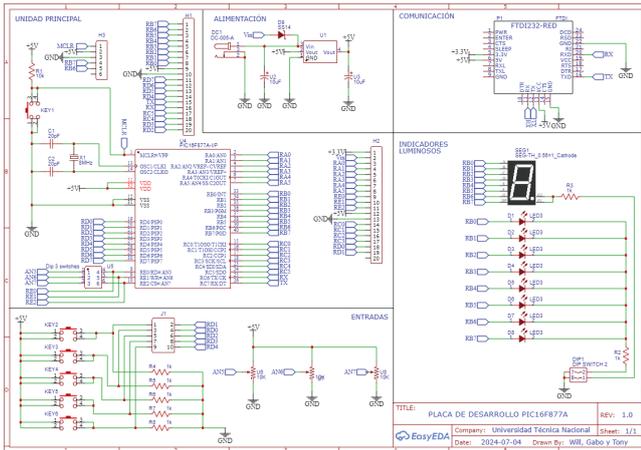


Fig. 2. Diseño esquemático. Elaboración propia.

Luego, se diseñó la PCB a partir del diseño esquemático:

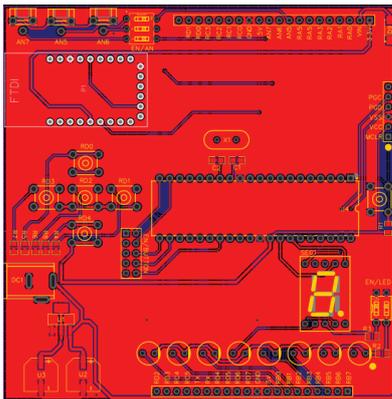


Fig. 3. Capa superior de la PCB. Elaboración propia.

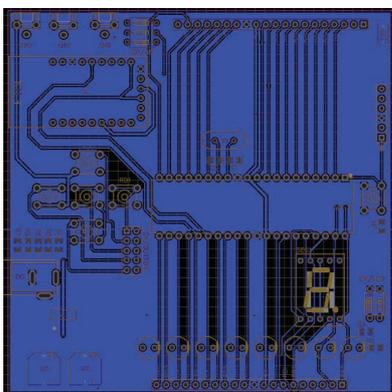


Fig. 4. Capa inferior de la PCB. Elaboración propia.

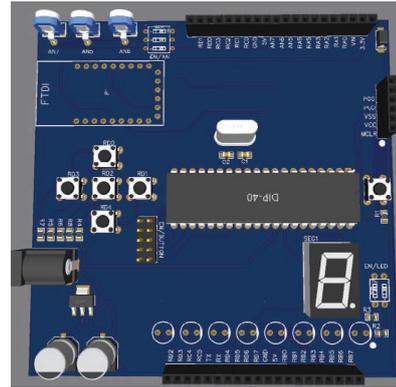


Fig. 5. Capa superior de la PCB en 3D. Elaboración propia.

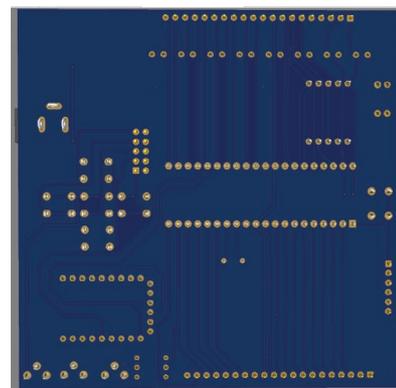


Fig. 6. Capa inferior de la PCB en 3D. Elaboración propia.

Y finalmente, se ensambló en físico. Cabe señalar que para este proceso, la tarjeta fue producida por un proveedor especializado y luego se le colocaron todos los componentes:

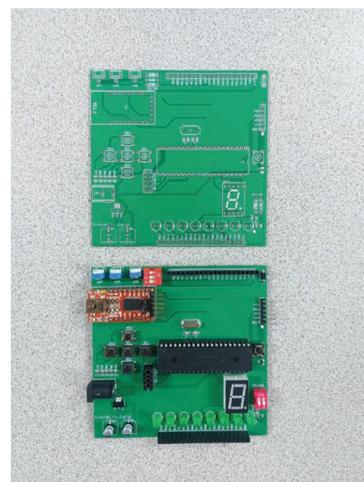


Fig. 7. Placa de desarrollo en físico. Elaboración propia.

V. ANÁLISIS DE RESULTADOS

El diseño esquemático se compone de varias partes:

A. Unidad principal

En la unidad principal del circuito se encuentra el microcontrolador PIC16F877A, el componente principal de la placa. Este es el cerebro del sistema y cuenta con múltiples puertos con diferentes funciones, que se utilizan para interactuar con otros componentes del circuito.

Adicionalmente, se encuentra el oscilador de cristal de 20MHz junto con los capacitores C1 y C2 que proporcionan la señal de reloj necesaria para el funcionamiento del microcontrolador. Asimismo, está el circuito de reset que incluye un pulsador S1 y una resistencia R1 conectada a VCC para el pull-up. Esto en conjunto permite reiniciar el microcontrolador manualmente.

B. Alimentación

Este módulo está compuesto por un regulador de voltaje AMS1117 que convierte la entrada de voltaje no regulada del jack DC-005 a una salida estable de 5V. Este voltaje es utilizado para alimentar el microcontrolador y otros componentes del circuito.

Asimismo, cuenta con capacitores de desacoplo que ayudan a estabilizar la alimentación eliminando el ruido eléctrico, y un diodo a la entrada que proporciona protección contra la polaridad inversa.

C. Comunicación

La sección de comunicación incluye un módulo FTDI232 que permite la correspondiente comunicación serial entre el microcontrolador y un ordenador u otro dispositivo serie, lo cual facilita la conexión con un PC para la programación y depuración del microcontrolador.

De igual manera, se colocaron los pines correspondientes a la programación mediante Pic kit, para obtener una mayor flexibilidad en este proceso de comunicación con el microcontrolador.

D. Indicadores Luminosos

En esta sección, los indicadores luminosos están compuestos por varios LEDs y un display de 7 segmentos que proporcionan retroalimentación visual de las operaciones del microcontrolador, tanto para visualizar caracteres como código binario.

Ambos recursos se encuentran debidamente conectados al puerto B del microcontrolador, y se colocó un dip switch para realizar la habilitación o deshabilitación de cada parte.

E. Entradas

Finalmente, se encuentra la sección de entradas, esta incluye varios pulsadores y resistencias pull-down que se conectan a los pines de entrada del microcontrolador, específicamente al puerto D, para permitir la interacción manual del usuario con el sistema en caso de necesitar entradas momentáneas.

De igual manera, se incluyeron potenciómetros en el puerto E para la obtención de señales analógicas. Para el caso de las entradas digitales, se colocaron conectores que actúan como puentes para habilitar o deshabilitar estas entradas, y con respecto a las entradas analógicas, se utilizó un dip switch al igual que para los indicadores luminosos.

Entre otras características de la placa, se colocaron conectores de expansión para obtener todas las funciones del PIC16F877A mediante cables o los llamados "jumpers".

Posteriormente, se realizó la PCB, para ello se utilizó el software Easy EDA. En este caso, se abarcaron varios aspectos fundamentales para garantizar la funcionalidad y eficiencia de la placa.

- Se revisó que los componentes estuvieran colocados de manera lógica y eficiente, facilitando el ruteo de las pistas y minimizando la longitud de las conexiones críticas.
- Asimismo, se evaluó si las pistas tenían los anchos apropiados para las corrientes que debían soportar
- Se comprobó que existieran pistas de tierra y potencia sólidas, y que estuvieran bien conectadas a todos los componentes que los requirieran.
- Se aseguró que la PCB cumpliera con las dimensiones previamente establecidas.
- En cuanto a tolerancias y espaciamentos, se aseguró de que todas las pistas, espacios, y agujeros cumplieran con las tolerancias y reglas mínimas que se propusieron.
- Se evaluó si el uso de pistas estaba optimizado, evitando el uso excesivo que podría incrementar los costos y potencialmente afectar la fiabilidad.

Y finalmente se ensambló la placa. Con esta ya finalizada por completo, se le realizaron pruebas funcionales para confirmar que el diseño cumpliera con las especificaciones eléctricas y mecánicas. En primera instancia, se inició con pruebas de voltaje y continuidad, para posteriormente, cargarle un código y verificar la interacción de algunos componentes de la placa.

VI. CONCLUSIONES

- 1) La evaluación de las características del PIC16F877A ha confirmado su idoneidad para aplicaciones educativas. Con su arquitectura de 8 bits, capacidades

de I/O versátiles y soporte para comunicación serial, y ADCs, el PIC16F877A proporciona una plataforma robusta para el aprendizaje y la experimentación en el campo de los microcontroladores.

- 2) El diseño esquemático de la placa de desarrollo ha sido fundamental para garantizar la funcionalidad, así como también, para formar la iniciativa del resultado esperado. La integración de LEDs, interruptores, resistencias y otros componentes se realizó de manera que se permita una amplia variedad de experimentos y prácticas.
- 3) El diseño de la placa se llevó a cabo con un enfoque en la organización lógica y la optimización del espacio. La disposición estratégica de los componentes y el enrutamiento eficiente de las pistas aseguran una placa funcional y fácil de utilizar.
- 4) El ensamble físico de la placa se realizó siguiendo el diseño de la placa establecido. Después de la fabricación de la PCB por un proveedor especializado, los componentes fueron ensamblados de manera cuidadosa y precisa, utilizando materiales que aseguran durabilidad y funcionalidad.
- 5) La evaluación final de la funcionalidad y rendimiento de la placa de desarrollo ha mostrado resultados positivos. La placa cumple con los objetivos establecidos, proporcionando una herramienta educativa eficaz para la enseñanza y experimentación con microcontroladores.

- [2] M. López, M. A. Ruiz y M. López, *Plataforma de entrenamiento para PIC*, Instituto Tecnológico de Celaya, México: 2015.
- [3] Microchip Technology Inc., *Curiosity Development Board User's Guide*, DS40001804B, 2016, [Online]. Available: <https://ww1.microchip.com/downloads/en/DeviceDoc/40001804B.pdf>

RECONOCIMIENTOS

Queremos expresar nuestro más sincero agradecimiento a varias personas que han sido fundamentales para la realización de este proyecto. Sin su apoyo y colaboración, este trabajo no habría sido posible.

En primer lugar, agradecemos al profesor Esteban Alonso Rojas Jiménez por su orientación y su constante apoyo a lo largo de todo el proceso. Su conocimiento profundo y su entusiasmo por la enseñanza han sido una fuente de inspiración y han guiado nuestro trabajo desde la concepción inicial hasta la finalización del proyecto.

Finalmente, extendemos nuestro agradecimiento a nuestras familias y amigos por su comprensión y apoyo durante el transcurso de este proyecto. Su paciencia y aliento han sido fundamentales para mantenernos motivados y enfocados en nuestros objetivos.

Este proyecto es el resultado de un esfuerzo conjunto y el apoyo de todos los mencionados ha sido invaluable. A cada uno de ellos, nuestro más profundo agradecimiento.

BIBLIOGRAFÍA

- [1] J. Silvestre y N. E. Caudillo, *Diseño, desarrollo y aplicación de una tarjeta electrónica didáctica para la enseñanza aprendizaje de sistemas embebidos*, Tecnológico de Nacional de México, México: 2023.



Desarrollo de una plataforma de laboratorios remotos con microcontroladores

<i>Marcelo Zabaljauregui</i>	<i>Héctor Hugo Mazzeo</i>	<i>Omar E. Rodríguez</i>	<i>Alejandro M. Rapallini</i>	<i>José A. Rapallini</i>
Centro CODAPLI	Centro CODAPLI	Centro CODAPLI	Programa PNT- Esquel	Centro CODAPLI
Universidad Tecnológica Nacional	Universidad Tecnológica Nacional	Universidad Tecnológica Nacional	Universidad del Chubut	Universidad Tecnológica Nacional
La Plata, Buenos Aires, Argentina	La Plata, Buenos Aires, Argentina	La Plata, Buenos Aires, Argentina	Esquel, Chubut, Argentina	La Plata, Buenos Aires, Argentina
<i>mzabaljauregui@frlp. utn.edu.ar</i>	<i>hugo.maz@frlp.utn.e du.ar</i>	<i>orodriguez@frlp.utn.e du.ar</i>	<i>amrapallini@udc.edu. ar</i>	<i>rapallini@frlp.utn.edu. ar</i>

Resumen—En este trabajo se describe un proyecto de automatización y acceso remoto para la realización de prácticas de laboratorio. La idea de la utilización de laboratorios remotos no es nueva, se potenció durante la pandemia, y se estableció en función de la posibilidad de realizar laboratorios disponibles en forma continua y con equipamiento de costos elevados.

Con el objetivo principal de brindar a los usuarios la posibilidad de realizar prácticas en entornos controlados, con elementos de trabajo reales y donde se pueda interactuar en forma remota con equipamiento, dispositivos o sistemas físicos, químicos, electrónicos, etc., se desarrollaron las actividades necesarias para su implementación.

Algunas experiencias realizadas por el Centro de Codiseño Aplicado (CODAPLI) sobre este tema demuestran la necesidad del uso de elementos físicos para su desarrollo y sobre todo el conocimiento de protocolos de comunicación capaz de ofrecer funcionalidades para acceso remoto a servidores y otros equipos remotos desde cualquier lugar solo con la ayuda de una conexión a Internet y un navegador web.

Palabras clave: laboratorios, remotos, microcontroladores, redes, educación.

I. INTRODUCCIÓN

Actualmente la educación a distancia se presenta como la solución ideal para un conjunto de trabajos prácticos que exigen el

disponer de sistemas de enseñanza mucho más flexibles, accesibles y adaptativos (sin limitaciones espaciales ni temporales) que los tradicionales.

Por consiguiente, trasladando este entorno práctico a la enseñanza a distancia, el elemento necesario para abordar la realización de prácticas sobre diversas materias de Ingeniería es la existencia de un sistema de apoyo a la enseñanza consistente en un laboratorio virtual y de telepresencia accesible a través de una red interna o Internet que permita al estudiante practicar de una forma lo más similar posible a como si estuviese en las dependencias del laboratorio, dándole la posibilidad de manejar las simulaciones o interactuar con los sistemas reales.

Es precisamente de la mano de las TICs desde donde se pueden aportar algunas soluciones para ampliar el acceso a la experimentación en forma de laboratorios remotos. Así se conseguirán simultáneamente dos objetivos didácticos: (a) realizar prácticas relacionadas con la asignatura ampliando la disponibilidad de los laboratorios y (b) formar a los estudiantes en el uso de las TICs.

La aplicación que se describe en este artículo fue pensada originalmente como un complemento a las clases prácticas de asignaturas de Ingeniería en Sistemas. Sin embargo, nada impide adaptarla para su utilización en otras áreas. Para cada caso particular, sólo es necesario modificar el software y hardware propio de la aplicación a

automatizar, adaptar sus entradas y salidas y reconfigurar algunos parámetros del sistema para acondicionarlo y vincularlo apropiadamente con el sistema principal.

La idea de realizar prácticas de laboratorio en forma remota tiene varios objetivos:

- Ampliar la ventana de tiempo en que los usuarios acceden al equipamiento, es decir, maximizar el uso del equipamiento disponible.
- Desarrollar competencias en los usuarios al familiarizarse con nuevas formas de comunicación con el equipamiento, posiblemente más cercanas a muchas aplicaciones en la industria.
- Independizar el desarrollo de trabajos de laboratorio de lo presencial, cuestión que cobró inesperada relevancia a partir de la pandemia.

En nuestra facultad, el CODAPLI desarrolla experiencias tendiendo a la remotización de laboratorios desde 2015, concretando logros significativos:

- Investigación y desarrollo de proyectos relacionados con los sistemas de tiempo real: sistemas centralizados y distribuidos constituidos por computadoras personales, sistemas embebidos basados en placas con microcontrolador, etc.
- Implementación de un sistema de desarrollo de laboratorios remotos con software libre, adaptable a la realización de prácticas de distinta índole [3, 4].
- Laboratorio remoto para la realización de experiencias de química para el Departamento de Ciencias Básicas [5-7].

Al consolidarse la implementación de laboratorios remotos, se ampliaron los objetivos iniciales, incluyendo nuevas estrategias en docencia:

- Formación de recursos humanos: desarrollo de un ámbito para la generación de recursos humanos altamente especializados. Estudio y resolución de problemáticas industriales (diseño innovador de sistemas). Transferencia de los conocimientos al sistema académico (alumnos, profesionales, cátedras y grupos de trabajo).
- Transferencia de conocimientos a los alumnos que requieran una formación adicional a la recibida en las cátedras de la

especialidad.

II. IMPLEMENTACIÓN DEL ACCESO REMOTO

Uno de los objetivos que se planteó en el proyecto fue que el usuario del laboratorio disponga de una plataforma remota con la cual poner en práctica los conocimientos adquiridos en las materias que cursa sin necesidad de recurrir a la adquisición de material extracurricular [1, 2].

Como se observa en la Fig. 1, el sistema está compuesto por un conjunto de máquinas conectadas a un servidor que se encarga de gestionar las reservas de los laboratorios y permite a los usuarios acceder al mismo por medio de un navegador web con soporte Java instalado.

De esta manera, distintos usuarios pueden acceder a diferentes aplicaciones remotas en cualquier momento, sólo será necesario seleccionar y reservar la práctica de laboratorio correspondiente mediante el sistema de reservas.

En cuanto a las máquinas locales, cada una de ellas dispone de los siguientes elementos:

- Una cámara web y una placa microcontrolada con entradas y salidas analógicas y digitales, ambas conectadas a la computadora local mediante un puerto USB. La placa cuenta con un entorno de desarrollo que permite implementar un sistema embebido con escaso o nulo hardware adicional, facilitando el uso de la electrónica en proyectos multidisciplinarios.

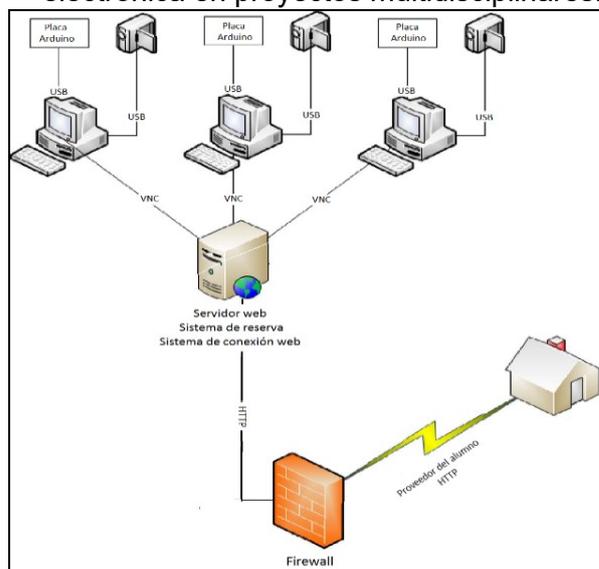


Fig. 1. Diagrama esquemático del laboratorio remoto

- Hardware y software específico para el desarrollo de la práctica de laboratorio, dependiendo de las necesidades de cada caso particular. Además de la placa con microcontrolador adecuada, se requerirán sensores y/o actuadores, interfaces o drivers de potencia para comandar los dispositivos electromecánicos a controlar y una aplicación en software encargada de controlar localmente el sistema físico a experimentar y que a su vez se comunicará con el servidor.
- La aplicación de software principal encargada de automatizar las tareas de la práctica, gestionar las conexiones de datos, turnos y horarios de utilización, etc.

Con respecto al servidor que corre la aplicación de acceso remoto, la utilización de Guacamole demostró que requiere importantes recursos en ciclos de cómputo y memoria RAM. Dado que Guacamole corre en una plataforma Java y permite concentrar muchos accesos de usuarios, hace necesario usar procesadores Intel I5 de segunda generación para cubrir esa demanda.

El proyecto se dividió en tres etapas:

1. Puesta en funcionamiento

Se inicia la puesta en funcionamiento de las PC locales con las cámaras IP y las placas conectadas, configuradas adecuadamente, como también del servidor que permite conectarse a las mismas desde la web. Para tal fin se instalaron y configuraron las siguientes aplicaciones:

1.1. Para las máquinas locales:

- Sistema operativo: Debian Stretch, optimizado para que funcione con la menor cantidad de recursos de hardware, CPU y RAM.
- Aplicación propia para el desarrollo de la práctica de laboratorio: en cada caso habrá que definirla dependiendo del tipo de práctica a realizar, el hardware y software de control, cantidad de sensores y actuadores necesarios, etc.
- Servidor VNC: necesario para la visualización de la experimentación remota.

1.2. Para el servidor de acceso:

- Sistema Operativo: Debian Stretch

- HTML: se instaló un servidor Apache.
- Bootstrap: se usa para darle un aspecto más moderno a los portales de acceso.
- JQuery: sirve de apoyo al Bootstrap.
- MySQL: se utiliza como base de datos para los usuarios y laboratorios.
- PHP: para poder interactuar con MySQL.
- Tomcat: requerimiento para poder instalar Guacamole.
- Guacamole: servicio de conexión remota para HTML5.

1.3. Diseño de un sistema de reservas

Permite gestionar los laboratorios y adecuación del sistema de acceso web para que pueda convivir con un sistema de reservas.

Ya que el sistema permite seleccionar distintos laboratorios para realizar, el usuario deberá ingresar primero en el laboratorio correspondiente y luego ir a la opción de reserva. Una vez allí podrá seleccionar el día y horario de realización de la práctica, siempre y cuando estén disponibles. Luego de verificar los datos que aparecen en la pantalla e ingresar la dirección de email del usuario (a la cual se le enviará un mensaje con los datos necesarios para ingresar al sistema) y la contraseña, aparecerá la confirmación validando la operación.

Se utilizaron distintas tecnologías: HTML, PHP, JQuery y MySQL. Su conjunción permitió diseñar un portal ágil para la gestión de los laboratorios, tanto para los usuarios como para los administradores de la plataforma. Además, el uso combinado de Tomcat y Guacamole provee el acceso remoto a través de HTML 5.

2. Apertura de los protocolos necesarios a Internet

Para que el laboratorio pueda ser accedido desde cualquier lugar por el usuario remoto.

Se cuenta con dos tipos de accesos, uno dedicado a la reserva y activación del laboratorio y otro para el acceso de los administradores a fin de visualizar las reservas hechas y realizar estadísticas de su uso o algún 'deslogueo' forzado en el caso de que surja algún problema o situación imprevista.

Existe además un "deslogueo" automático en el caso que, después de una reserva se genere otra. Como una posibilidad adicional, está contemplado el caso en el cuál si un usuario

está usando el sistema y agota su tiempo (no teniendo una reserva posterior para su uso) se le permita seguir utilizándolo.

3. Ingreso a la URL asignada para el laboratorio.

En primer lugar, se requiere un procedimiento para la reserva de laboratorios. Tomaremos como ejemplo el caso de una práctica para la materia Aplicaciones en Tiempo Real. En la pantalla de ingreso seleccionamos la solapa "Laboratorios" en la barra de opciones de la parte superior (Fig. 2).

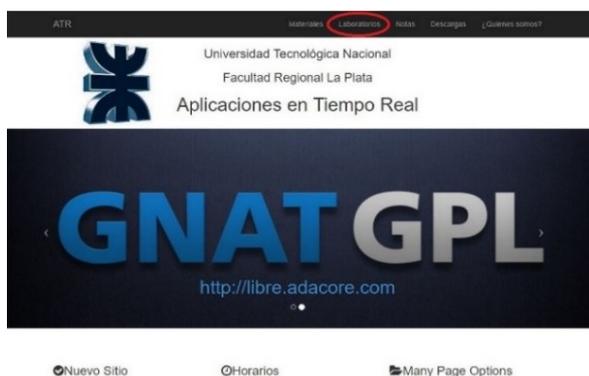


Fig. 2. Pantalla de ingreso al laboratorio remoto

3.1. Reserva de laboratorio

Una vez seleccionado el laboratorio a realizar, habrá que ir a la opción "Reserva de laboratorios", donde se podrá reservar día y horario de realización (Fig. 3).



Fig. 3. Pantalla de selección y reserva de horario

3.2. Ingreso al laboratorio remoto

Una vez realizada la reserva, habrá que verificar los datos que aparecen en la pantalla e ingresar una dirección de e-mail válida (que será utilizada para el envío de la confirmación de la reserva, usuario y contraseña para ingresar al sistema. Fig. 4).



Fig. 4. Confirmación de reserva recibida por e-mail

Con el usuario y contraseña recibidos, ya será posible acceder al sistema. El procedimiento para el ingreso al laboratorio se realizará de la siguiente manera:

Ingresar al enlace enviado por email: <https://codapli.frlp.utn.edu.ar/remotelab/#/> (Fig. 5).



Fig. 5. Pantalla de ingreso al sistema

Ingresar el nombre de usuario y contraseña recibidos en el e-mail de confirmación de la reserva. Al ingresar se verá el login del sistema operativo remoto (al cual se accede con user: codaplilab01, password: lab01 -Fig. 6-):



Fig. 6. Pantalla de acceso al laboratorio remoto

Ya se está en condiciones de operar en el laboratorio remoto. Una vez ingresado al entorno de la aplicación, el usuario dispondrá de un menú de aplicaciones que podrá seleccionar. Por ejemplo, dentro del menú Aplicaciones/ATR se encontrarán las distintas aplicaciones para la cátedra correspondiente, en este caso ATR (Aplicaciones en Tiempo Real). La aplicación “Cheese” del menú permite abrir la cámara que apunta al proyecto de laboratorio con el cual se experimentará (Fig. 7).

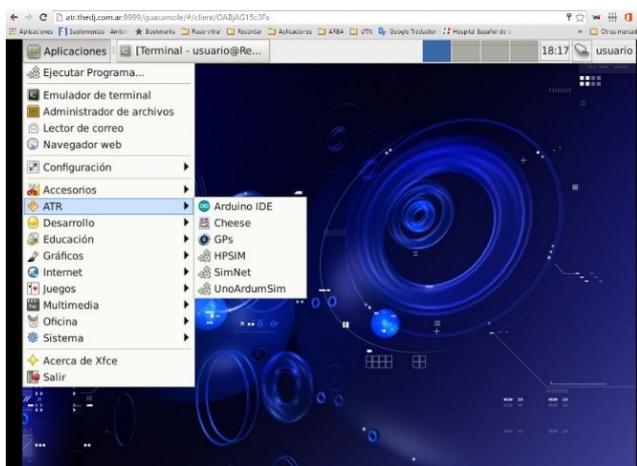


Fig. 7. Aplicaciones disponibles ya dentro del entorno

III. LABORATORIOS DESARROLLADOS

En la primera etapa de utilización de esta herramienta se logró la implementación de un sistema genérico, basado en las experiencias realizadas con laboratorios remotos de cátedras de Ciencias Básicas, en particular de Física y Química, contando con el funcionamiento de los sistemas de adquisición, visualización y programación de acceso remoto [6]. Para ello se automatizó entre otros, un laboratorio remoto para la “medición de presión de gas al variar volumen del recipiente”. Se implementó el sistema de adquisición con sensores marca Vernier, con una jeringa a modo de émbolo y un sensor de presión que mide la presión del gas en función del volumen al desplazar el pistón. El accionamiento del émbolo se hace a través de un motor paso a paso o “stepper” y un sensor que mide su desplazamiento mediante un tornillo controlado por software (Fig. 8).

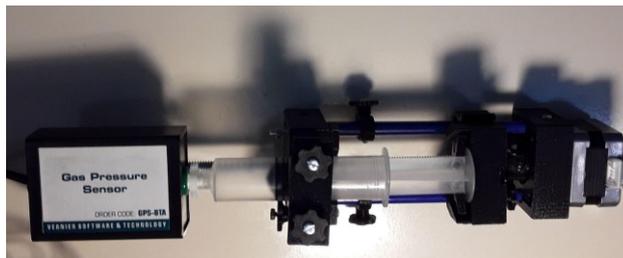


Fig. 8. Dispositivo para medir la presión de un gas

Actualmente se desarrollan prácticas remotas con la plataforma Arduino. Para ello, se ha conectado una placa Arduino Uno y una ‘protoboard’ a las que se han conectado diodos leds de colores, un servomotor y un sensor de temperatura y humedad (DHT11). De esta manera se pueden ejecutar diversos ‘sketchs’ utilizando estos componentes. En la Fig. 9 se muestra un ‘sketch’ que toma valores de temperatura y humedad del laboratorio a través del DHT11 y los muestra en el monitor serie del IDE de Arduino [10].

Se observa la vista de la cámara web de la placa Arduino Uno y la ‘protoboard’ con los sensores y actuadores conectados. En otra de las ventanas abiertas se muestra el sketch para la medición de temperatura y humedad y abajo la salida del monitor serie con la impresión de los valores cada 1 segundo.

Con el fin de evitar utilizar una PC en forma permanente, la máquina local fue reemplazada con éxito por una placa ‘Raspberry Pi’ con OS Linux instalado, que además permite bajar costos del equipamiento.

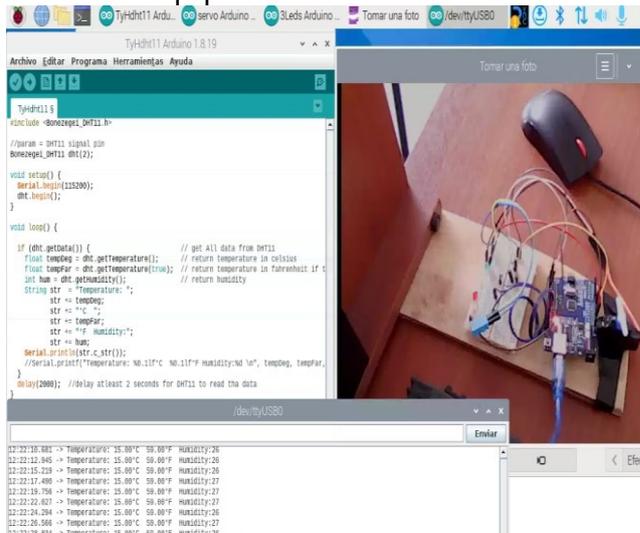


Fig. 9. Medición de temperatura y humedad con DHT11



IV. TRABAJOS PREVISTOS A REALIZAR

Con otras Universidades:

A partir de las experiencias realizadas en el Centro CODAPLI, se prevé un desarrollo conjunto con la Universidad del Chubut, dentro del Programa Nuevas Tecnologías (PNT) en la sede académica de Esquel, lo que permitirá que los alumnos dispongan de laboratorios en otras sedes muy distantes de la UDC como Rawson, Gaiman, Trelew, Puerto Madryn y Sarmiento. En la primera etapa se pondrán en funcionamiento los módulos de entrada salida con placas Arduino [8] [9] conectadas a sistemas de cómputo Raspberry Pi encargados de interactuar entre la experiencia y el servidor que provee el acceso remoto (Fig. 1) a los laboratorios previstos a remotizar.

En la Universidad Tecnológica Nacional, Facultad Regional La Plata, continuar con el desarrollo de los siguientes proyectos:

1. Junto al Grupo de Investigación en Enseñanza de las Ciencias (IEC), en el desarrollo de nuevas experiencias de laboratorios remotos de Física [10].
2. Con el área de Energía Eléctrica del Centro CODAPLI, trabajando en los Laboratorios de Medidas Eléctricas y Laboratorio de Máquinas Eléctricas y Accionamientos, [11][12], en banco de pruebas con generador eléctrico que permite controlar el giro de motores a distintas velocidades a través de un control con PLC, experiencias que una vez consolidadas permitirán ser ofrecidas a industrias, para su utilización en cursos de formación del personal.

- Congreso nacional CoNallSI, 3 y 4 de noviembre de 2022, Concepción del Uruguay, Entre Ríos, Argentina.
- [6] S. Juanto, S. Pastorino, L. Mardones. "Experiencia de laboratorio de Gases Ideales: un método para la integración de contenidos en Ingeniería", https://www.memoria.fahce.unlp.edu.ar/trab_eventos/ev.8099/ev.8099.pdf, Universidad Nacional de La Plata, 2015.
 - [7] S. Juanto, D. Alustiza, F. Prodanoff, C. Quintero, N. Cristofoli. "Desarrollo propio de un sistema de adquisición de datos para trabajos de laboratorio". Las ciencias de la naturaleza y la matemática en el aula: nuevos desafíos y paradigmas -1ª ed.- Tandil 2018. Libro digital, pdf. Archivo Digital: descarga y online ISBN 978-950-658-449-8.
 - [8] Laboratorio remoto para prácticas con Arduino, <https://codapli.frlp.utn.edu.ar/remotelab/#/>
 - [9] Computers & Education. Revista de Elsevier dedicada principalmente a aspectos pedagógicos del uso del computador y las comunicaciones en la enseñanza, incluyendo la enseñanza a distancia.2023
 - [10] Laboratorios de Acceso Remoto 2023 (Homologación-N°133523300-APN-SECPU#ME). Secretaria de Políticas Universitarias. Ministerio de Educación.
 - [11] IEEE Transactions on Education. "Métodos y tecnologías de educación, materiales para la enseñanza, y programas de desarrollo educacionales y profesionales sobre disciplinas de ingeniería eléctrica".. 2022
 - [12] Homologación (ENECLP0010186TC) del Proyecto de Investigación y Desarrollo. Convocatoria PID UTN 2023 Disposición SCyT N°32/2023 Rectorado UTN.

REFERENCIAS

- [1] Isidro Calvo, Ekaitz Zulueta, Unai Gangoiti, José Manuel López, "Laboratorios remotos y virtuales en enseñanzas técnicas y científicas", Dpto. Ingeniería de Sistemas y Automática Escuela Universitaria de Ingeniería de Vitoria-Gasteiz UPV/EHU.
- [2] Domínguez, M., Fuertes, J.J., Reguera P. "Desarrollo de un laboratorio remoto para la formación vía Internet en Automática". Año 2009.
- [3] H. Mazzeo, J. Rapallini, M. Zabalgauregui, O. Rodríguez. "Desarrollo de laboratorios virtuales y remotos para enseñanza universitaria". II Jornada Argentina de Tecnología, Innovación y Creatividad. Año 2016.
- [4] H. Mazzeo, J. Rapallini, M. Zabalgauregui, O. Rodríguez. "Diseño de una plataforma remota para desarrollo de prácticas de laboratorio". LVR – CACIC 2018.
- [5] S. Juanto, M. Zabaljauregui, D. Alustiza, H. Mazzeo, O. Rodríguez, J. Rapallini, J. Carrillo Pavón. "Laboratorios Remotos Aplicados a la Enseñanza Universitaria". X

Diseño de un laboratorio remoto para la enseñanza de sistemas embebidos

Juan Piñeiro	Jorge R. Osio	Constantino Palacio	Noemí Arce	Marcelo Cappelletti	Martín Morales
TICAPPS, IlyA	TICAPPS, Universidad Nacional Arturo Jauretche	Grupo GCA, Instituto	Grupo GCA, Instituto	Grupo GCA, Instituto	TICAPPS, IlyA,
UNAJ	Grupo GCA, LEICI (UNLP-CONICET)	LEICI (UNLP-CONICET)	LEICI (UNLP-CONICET)	LEICI (UNLP-CONICET)	UNAJ
Florencio Varela, Argentina	La Plata, Argentina	La Plata, Argentina	La Plata, Argentina	La Plata, Argentina	La Plata, Argentina
juanignacio.ing7@gmail.com	Josio@unaj.edu.ar	constantinopalacio@gmail.com	noemiarceperez@gmail.com	mcappelletti@unaj.edu.ar	Martin.morales@unaj.edu.ar

Resumen— Un Laboratorio Remoto se basa en un recurso de acceso remoto de naturaleza real. Este representa el acceso a un laboratorio real, es decir, a un laboratorio con equipos reales, pero de manera no presencial, el mismo se efectiviza a través de un medio de comunicación como, por ejemplo, Internet. En estos laboratorios el usuario opera y controla en forma remota el laboratorio real ubicado en los laboratorios de informática de la universidad a través de una interfaz de experimentación, desarrollada e implementada desde la máquina servidor.

El trabajo consiste en el diseño de un Laboratorio Remoto en un Entorno Virtual de Enseñanza y Aprendizaje que favorezca la experiencia del alumno respecto al tema “Programación de aplicaciones en sistemas embebidos”, mediante el control de la placa “Arduino uno” por medio de una interfaz de usuario de acceso remoto. Además, el entorno provee turnos para evitar superposiciones en el uso del Laboratorio.

La parte de implementación consiste en desarrollar el código y las configuraciones necesarias para poder manejar el hardware, el streaming/cámara y los equipos (placa sobre Arduino) que se encuentran disponibles en el laboratorio de forma remota.

Palabras clave: Laboratorios remotos, sistemas Embebidos, Internet de las cosas

I. INTRODUCCIÓN

El objetivo principal del trabajo consiste en desarrollar una plataforma de control y acceso al Laboratorio físico de la universidad, conformado por el Microcontrolador Atmega328p (de arduino) y una placa de periféricos, desde una máquina remota. Específicamente se implementará una plataforma de acceso para los/as alumnos/as donde puedan interactuar con el Laboratorio físico y de manera remota. El usuario podrá compilar el código remotamente y el código binario resultante se cargará en el kit arduino uno conectado mediante la interfaz USB al servidor.

En la siguiente figura (figura 1), se puede observar cómo es la arquitectura general del proyecto.

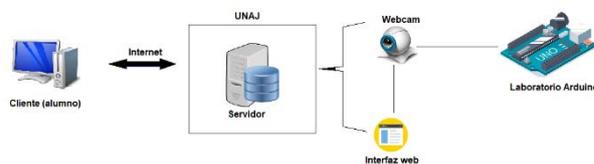


Figura 1. Diagrama del proyecto general

Los objetivos específicos son:

- Configurar la herramienta para reconocer el Hardware específico utilizado en el Laboratorio Real
- Programar y configurar un servidor web en donde se visualiza la plataforma, con



sus datos, el código del programa y parámetros ingresados por los estudiantes.

- Resolver las solicitudes de asignación de turnos y autenticación de los usuarios.
- Desarrollar la interfaz web que use la API REST y muestre los datos de ejecución de la aplicación cargada en la placa

A continuación, se detallan los conceptos abordados para la realización del laboratorio remoto y su contribución como herramienta de aprendizaje en el contexto académico.

A) Educación a distancia

La educación a distancia es una forma de enseñanza en la cual los estudiantes no requieren asistir físicamente al lugar de estudios. En este sistema de enseñanza, el alumno recibe el material de estudio (personalmente, por correo postal, correo electrónico u otras posibilidades que ofrece Internet), permitiendo que, en el acto educativo, se empleen nuevas técnicas y estrategias de aprendizaje centradas en el propio estudiante, fomentando así el autodidactismo y la autogestión. Es decir, se trata de una forma de educación flexible, autónoma y autodirigida, cuyas principales herramientas son las tecnologías de la comunicación y la información. Al aprendizaje desarrollado con las nuevas tecnologías de la comunicación se le llama "aprendizaje electrónico" ("*E-learning*", en inglés). La plataforma más utilizada actualmente para esta modalidad es Moodle.

Beneficios de la educación a distancia:

- La educación a distancia satisface las necesidades de los estudiantes que de otro modo no podrían asistir a clases presenciales, debido a las restricciones de distancia o de tiempo. Uno de los mayores beneficios de la educación a distancia es la flexibilidad.
- Los Programas de educación a distancia permiten un mayor acceso al aprendizaje y fomentan el aprendizaje permanente. El aprendizaje a distancia permite a los estudiantes elegir entre un conjunto más amplio de instituciones académicas para su aprendizaje permanente.

- Otro beneficio de la educación a distancia para los estudiantes es la capacidad de hacer el trabajo en equipo en grupos interactivos. Los estudiantes tienen la oportunidad de comunicarse con otras personas de diferentes orígenes y de escuchar a una gran variedad de expertos de todo el mundo.
- Elimina las barreras geográficas, la población puede acceder a este tipo de educación independientemente de donde resida.
- Reduce costos al evitar gastos de traslados o residencia en un lugar diferente.
- Incorpora herramientas tecnológicas para el manejo de la información, las cuales son necesarias para desempeñarse profesionalmente en la sociedad en constante cambio, tales como las plataformas virtuales.
- El alumno desarrolla una alta capacidad para autorregular su propio aprendizaje favoreciendo así sus actitudes y valores de responsabilidad, disciplina y compromiso para lograr ser autónomo.
- El rol del estudiante es activo pues desarrolla estrategias intelectuales importantes para realizar tareas colaborativas, comunicarse efectivamente, ser creativo e innovador.
- El asesor lleva un seguimiento riguroso del estudiante empleando diversos instrumentos para evaluarlo respetando el ritmo de trabajo del alumno.
- Facilita a las personas con capacidades diferentes el acceso a cursar una carrera.
- Tienen la misma validez los papeles que se obtienen al término de la carrera, que el que se cursa de manera escolarizada.
- Debido a su comodidad, el alumno tiene la posibilidad de manejar el tiempo dedicado a cada actividad de acuerdo a sus otras actividades o a la rapidez con la que avance flexibilizando el tiempo de finalización de la carrera.

B) Modelo de aprendizaje flexible inteligente 2005 – Actualidad

Este modelo aún está en desarrollo, pero, la quinta generación se ha caracterizado por la

aplicación de sistemas de respuesta inteligente que permiten hacer más efectivos los sistemas de tutoría y favorecer economías de escala y costo de efectividad asociados a ellas en grupos numerosos de estudiantes. Esta generación se encuentra en pleno desarrollo.

Por otra parte, en el año 2006 nace el concepto de M-Learning que es la abreviatura de Mobile E-learning, término en inglés que en términos de enseñanza virtual se traduce como "Aprendizaje Electrónico Móvil". Es una evolución del e-learning, que a su vez es una evolución de la formación a distancia. Está asociada al uso de tecnologías móviles en la educación. Es una metodología de enseñanza y aprendizaje que se basa en el uso pedagógico a través de pequeños dispositivos móviles como laptops, teléfonos móviles, tablets y todo dispositivo de mano que tenga alguna forma de conectividad inalámbrica. Todas estas tecnologías proveen facilidades para el acceso remoto a la información, pero hay actividades presenciales que son complejas de reemplazar a distancia. Si bien existen muy buenos simuladores que arrojan resultados similares a la experiencia real, no hay nada más práctico que realizar la experiencia con los instrumentos reales. Esta es la motivación para el desarrollo e implementación de los laboratorios remotos.

C) Laboratorios remotos

Los Laboratorios Remotos (LR), son herramientas tecnológicas compuestas por software y hardware que les permite a los estudiantes de manera remota realizar sus prácticas como si estuvieran en un laboratorio tradicional, generalmente el acceso se realiza a través de Internet o mediante una red académica.

Los LR son normalmente utilizados cuando se requiere que los estudiantes pongan en práctica de manera autónoma lo aprendido las veces que lo requieran, esto les permite confrontar sus conocimientos previos con los nuevos conocimientos adquiridos hasta llegar a construir un nuevo conocimiento, se podría decir entonces que en este nuevo escenario prevalece el modelo constructivista que les permite a los estudiantes adquirir aprendizajes significativos para la vida profesional. Los LR son producto del desarrollo de las Tecnologías de la información y la comunicación (TIC) que contribuyen con el mejoramiento de los

procesos educativos, son considerados como sistemas que no están basados en prácticas simuladas, si no en acciones que permiten al estudiante realizar actividades de laboratorio con dispositivos o instrumentación real, transfiriendo información entre el procedimiento práctico y el estudiante. El alumno maneja y controla los recursos disponibles en el laboratorio.

El diseño y desarrollo de los LR, está basado en un enfoque en la web, buscando utilizar Internet como un marco genérico de tecnología estándar ampliamente utilizada a nivel mundial, para que el desarrollo sea adaptable a diversas situaciones. Dicho enfoque puede ilustrarse en la Figura 2, donde se muestran diferentes elementos involucrados en los LR.

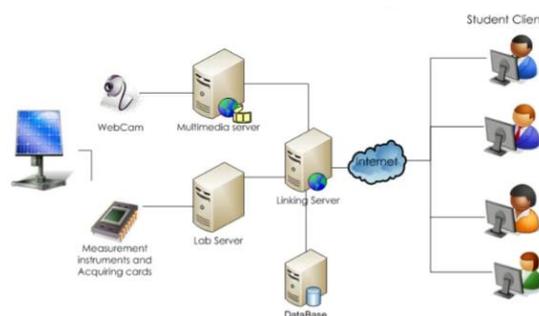


Figura 2. Elementos de los LR.

En la Figura 2 se muestran los elementos de un laboratorio remoto:

- Alumno, cliente o usuario: representa al individuo que va a efectuar el laboratorio remoto mediante una computadora. La computadora está conectada a internet y mediante un navegador web establecerá una conexión con el servidor que gestiona el servicio de LR.
- Servidor de enlace: es el que gestiona y administra las conexiones de los usuarios que quieren efectuar el LR.
- Servidor multimedia: es el que provee al servidor de enlace las imágenes y el audio en tiempo real del ambiente experimental.
- Servidor de laboratorio: es el que da soporte a una aplicación en particular, en este caso, una placa de adquisición e instrumentación y está conectado al servidor de enlace.
- Objetivo experimental: objeto que está bajo el control del servidor de laboratorio y que permite adquirir o efectuar

accionamientos, que se pueden medir, observar o escuchar gracias al servidor multimedia.

D) Sistemas embebidos

La propuesta de este laboratorio se enfoca en la enseñanza de los sistemas embebidos, también conocidos como computadoras integradas, son computadoras de factor de forma pequeña que impulsan tareas específicas. Pueden funcionar como dispositivos independientes o como parte de sistemas más grandes, de ahí el término "embebido", y a menudo se usan en aplicaciones con restricciones de tamaño, peso, energía y costo.

Como la mayoría de las computadoras, los sistemas embebidos son una combinación de hardware y software, por lo general:

- Microprocesadores o microcontroladores
- Unidades de procesamiento de gráficos (GPU)
- Memoria volátil y no volátil
- Puertos e interfaces de comunicación de entrada/salida
- Código del sistema y de la aplicación
- Fuentes de alimentación

II. IMPLEMENTACIÓN

A) Descripción de las Herramientas de HW y SW seleccionadas

Hardware

Las herramientas de Hardware utilizadas fueron seleccionadas al inicio del proyecto con el fin del desarrollo del mismo, teniendo en cuenta la popularidad y soporte que proveen. Se utilizó un Microcontrolador Atmega328p (de Arduino) y se diseñó una placa de periféricos a medida.

Kit ATmega328P (de Arduino UNO)

A continuación, se describirán las características del kit basado en el ATmega328p denominado Arduino uno. En la figura 3 se presenta el kit de Arduino con los pines I/O disponibles en los conectores laterales con las distintas funcionalidades que

se le pueden configurar a cada pin. En este kit los pines A0-A5 se pueden configurar como entradas analógicas o entradas/salidas digitales, según se requiera. Los pines precedidos con ~ se pueden configurar como salida PWM o entradas/salidas digitales (por ejemplo el pin ~6). El resto de los pines se puede configurar como entradas/salidas digitales. Los pines A4 y A5 (SDA y SCL respectivamente) se pueden configurar para implementar el protocolo I2C. Los pines 11, 12 y 13 (MOSI, MISO y SCK respectivamente) se pueden configurar para implementar el protocolo SPI. Por último, los pines 0 y 1 (RXD y TXD respectivamente) se pueden configurar para implementar el protocolo UART (protocolo serie asincrónico).

Label	Port
A0	PC0{ADC0}
A1	PC1{ADC1}
A2	PC2{ADC2}
A3	PC3{ADC3}
A4	PC4{ADC4}
A5	PC5{ADC5}
SCL	PC5{ADC5/SCL}
SDA	PC4{ADC4/SDA}
AREF	AREF
GND	GND
13	PB5{SCK}
12	PB4{MISO}
11	PB3{MOSI}
10	PB2{OC1B}
9	PB1{OC1A}
8	PB0
7	PD7
6	PD6
5	PD5
4	PD4
3	PD3{INT1}
2	PD2{INT0}
1	PD1{TXD}
0	PD0{RXD}

Figura 3. Pinout del Arduino UNO

Descripción del kit de periféricos del ATmega328p

El kit de periféricos para el microcontrolador ATmega328p fue diseñado con la idea de disponer de un kit genérico con suficientes periféricos para realizar las actividades prácticas en las diferentes materias del área de digitales de la carrera.

En la figura 4 se muestra el kit de periféricos visto de arriba, en donde se pueden apreciar una serie de conectores y componentes que conforman los periféricos. Además, se dispone de 9 jumpers que se incluyeron para compartir algunos pines I/O entre los periféricos (cabe destacar que en un laboratorio remoto estos jumpers se mantienen fijos).

Entre los periféricos principales contemplados se tiene un display LCD (puede ser de 2x16 mediante el conector LCD_2x16 o de 2x8 mediante el conector LCD_2x8), un teclado matricial de 4x4, un sensor de



iluminación LDR, un potenciómetro, un led RGB ánodo común, un sensor de temperatura y humedad DHT11/DHT22 y un parlante.

Respecto a los periféricos secundarios, se provee conexión para un acelerómetro, un reloj de tiempo real, un módulo bluetooth y se dejó disponible un conector UART para conectar cualquier dispositivo con interfaz UART (por ejemplo, módulo GSM/GPRS, módulo Wifi, módulo inalámbrico GPS, etc.).

Cámara web

Una cámara web es un dispositivo de entrada que captura imágenes digitales. Estas se transfieren al servidor, en donde pueden ser accedidas remotamente desde la página de alojamiento.

Software

Al momento de seleccionar el software a utilizar, se tuvo en cuenta que este fuera de Código Abierto (*Open Source*) y que contara con documentación de respaldo o foros de debate para posibles dificultades al momento de la implementación.

Cómo sistema operativo para el servidor se usó Ubuntu, que es una distribución Linux basada en Debian GNU/Linux, que incluye principalmente software libre y de código abierto.

La transmisión en vivo de streaming en el que se transmite audio o video a través de Internet, se realizó mediante MJPG-streamer. Aquí los medios se transmiten mientras se graban, lo que permite a los espectadores verlos o escucharlos en tiempo real. Dicha aplicación es ideal para la distribución web de vídeo desde cámaras compatibles con Linux. Esta aplicación genera datos JPEG directamente desde la cámara y emite una transmisión MJPG de alta velocidad en tiempo real.

Visual Studio Code

Visual Studio Code es un editor de código fuente gratuito, ligero pero potente que se ejecuta en su escritorio y en la web y está disponible para Windows, macOS, Linux y Raspberry Pi OS. Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene

un rico ecosistema de extensiones para otros lenguajes de programación (como C++, C#, Java, Python, PHP y Go), tiempos de ejecución (como .NET y Unity), entornos (como Docker y Kubernetes) y nubes (como Amazon Web Services, Microsoft Azure y Google Cloud Platform).

PlatformIO IDE

PlatformIO es una herramienta profesional multiplataforma, multiarquitectura y marco múltiple para ingenieros de sistemas embebidos y para desarrolladores de software que escriben aplicaciones para productos integrados. Visual Studio Code otorga una extensión llamada PlatformIO IDE. Sin entrar en demasiados detalles acerca de su implementación interna, PlatformIO cuenta con los siguientes puntos clave que describen su funcionamiento:

- La plataforma de desarrollo, así como la placa física empleada son configuraciones del proyecto y como tal van en el fichero "platformio.ini."
- En dicho fichero pueden existir diferentes configuraciones para diferentes entornos, todas ellas compartiendo el mismo código fuente.
- Relacionado con la anterior, el código fuente es independiente de la plataforma de desarrollo empleada y va localizado bajo la carpeta "src".
- Basándose en el fichero platformio.ini, el sistema se encarga de descargar e instalar todos los requerimientos necesarios.
- La carpeta "lib" almacena las diferentes librerías requeridas por la aplicación que se esté desarrollando.
- El usuario dispone en una única herramienta y con una API sencilla de toda la funcionalidad que requiere para crear, compilar y subir código.

PlatformIO Core (CLI)

PlatformIO Core (herramienta CLI) es el corazón de todo el ecosistema PlatformIO y consta de:

- Sistema de construcción multiplataforma
- Administrador de paquetes unificados



- Gestión de bibliotecas
- Buscador de dependencias de biblioteca (LDF)
- Monitoreo de puerto serie
- Componentes de integración (IDEs de escritorio y nube e integración continua).

Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh, entre otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual, según la normativa RFC 2616.

phpMyAdmin

phpMyAdmin es una herramienta de software libre escrita en PHP, destinada a manejar la administración de MySQL en la Web. phpMyAdmin admite una amplia gama de operaciones en MySQL y MariaDB. Las operaciones de uso frecuente (administración de bases de datos, tablas, columnas, relaciones, índices, usuarios, permisos, etc.) se pueden realizar a través de la interfaz de usuario, mientras aún tiene la capacidad de ejecutar directamente cualquier instrucción SQL.

Tecnologías web

Las tecnologías web se refieren a las diversas herramientas y técnicas que se utilizan en el proceso de comunicación entre diferentes tipos de dispositivos a través de Internet. Se utiliza un navegador web para acceder a las páginas web. Los navegadores web se pueden definir como programas que muestran texto, datos, imágenes, animaciones y videos en Internet. Se puede acceder a los recursos con hipervínculos en la World Wide Web utilizando las interfaces de software proporcionadas por los navegadores web.

El desarrollo web se puede clasificar en dos formas:

- **Desarrollo Frontend:** La parte de un sitio web con la que el usuario interactúa

directamente se denomina front-end. También se le conoce como el 'lado del cliente' de la aplicación.

- **Desarrollo de backend:** backend es el lado del servidor de un sitio web. Es la parte del sitio web que los usuarios no pueden ver e interactuar. Es la parte del software que no entra en contacto directo con los usuarios. Se utiliza para almacenar y organizar datos.

MYSQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web.

NO-IP

No-IP es un servicio DNS dinámico que toma una dirección IP dinámica y la hace actuar como si fuera estática al señalarse un nombre de host estático y verificar cada 5 minutos los cambios en la dirección IP.

B) Descripción de la implementación del sistema:

En este apartado, se detallarán todas las tareas realizadas durante la implementación. Cabe aclarar que el proyecto se desarrolló en una distribución de Linux, llamada Ubuntu 20.10. Pero, a través de configuraciones personalizadas, se pudo lograr migrar el proyecto a la distribución más actualizada que hay (Ubuntu 22.04.1 LTS).

Extensión PlatformIO IDE

Antes de instalar la extensión de PlatformIO IDE dentro de Visual Studio Code, fue importante instalar las dependencias de Python necesarias.

En primer lugar, se actualizaron lista de repositorios mediante el siguiente comando:

sudo apt-get update

En segundo lugar, se actualizan los paquetes mediante el siguiente comando:

sudo apt-get upgrade

En tercer lugar, instala python3 con el siguiente comando:

sudo apt install python3-pip

Finalmente, se instaló la extensión PlatformIO IDE. Para ello, se buscó la extensión en el buscador de VSC y se instaló como muestra la figura 5.



Figura 4. Instalación de PlatformIO IDE

Primera aproximación

En la extensión de PlatformIO IDE se logró crear un proyecto para Arduino a través de los siguientes pasos:

1. Se entra a la plataforma a través de la extensión instalada
2. Se hace click en New Project.
3. Se crea el proyecto
4. Se llenan los siguientes campos:
 - Name: Nombre de proyecto.
 - Board: La placa. En este caso, como se tiene Arduino Uno, esta fue la elegida.
 - Framework: Elegimos Arduino en este caso.
5. Se hace click en el botón "Finish" y crea el primer proyecto.

Directorio de proyectos PlatformIO

A continuación, se verá cómo se organiza un proyecto creado a través de la extensión de PlatformIO IDE en la figura 5.

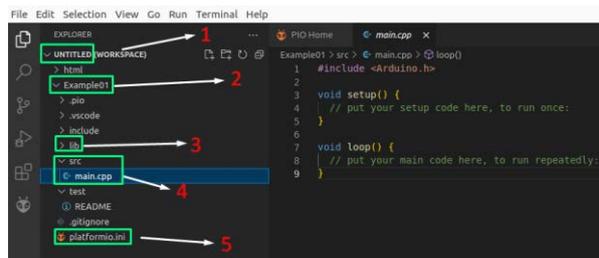


Figura 5. Organización de proyecto en PlatformIO IDE.
Fuente: Producción propia

1. Espacio de trabajo: «workspace» contiene el proyecto en el que se esté trabajando actualmente, se puede agregar más de uno y eliminarlos sin problemas.
2. Directorio del proyecto: es la carpeta principal que contiene los archivos y subcarpetas que PlatformIO crea automáticamente. Puede haber más de uno según la cantidad de proyectos que se estén gestionando.
3. Librerías: en esta carpeta el gestor de librerías de PlatformIO incluye las librerías privadas de cada proyecto. Las librerías son muy importantes, cuando se compila un código y se ejecuta en la placa, debemos tener en cuenta que funcionalidades tiene el código para ver si es necesario instalar una librería externa.
4. Código principal: la carpeta SCR (simboliza la abreviatura de "Source", fuente en inglés) aquí se incluye el código principal o código fuente del proyecto. Por otro lado, PlatformIO crea un fichero con el nombre "main.cpp". Este es el equivalente al archivo compilado por el IDE Arduino llamado: "codigo.ino".
5. platformio.ini: es un archivo de configuración que dota de funcionalidades extras al entorno. Es capaz de agilizar el trabajo cuando se trata de grandes proyectos.

En sistemas Unix y similares a Unix, se pueden crear *enlaces simbólicos* en el directorio: \$HOME/.local/bin/ a los ejecutables PlatformIO necesarios.

Esto permitirá ejecutar comandos de PlatformIO desde cualquier emulador de terminal.

Teniendo en cuenta lo mencionado anteriormente, los pasos a seguir son los siguientes:

En primer lugar, se debe exportar el directorio \$HOME/.local/bin/ a la variable de entorno PATH con el siguiente comando:

```
- export PATH=$PATH:$HOME/.local/bin
```

Por último, se crean los enlaces simbólicos con los siguientes comandos:

```
cd /
sudo ln -s ~/.platformio/penv/bin/platformio
usr/local/bin/platformio
sudo ln -s ~/.platformio/penv/bin/pio usr/local/bin/pio
sudo ln -s ~/.platformio/penv/bin/piodebuggdb
usr/local/bin/piodebuggdb
```

Primer proyecto desde línea de comando

Para inicializar un proyecto mediante comandos en la terminal se hizo lo siguiente:

En primer lugar, se creó una carpeta para alojar el primer proyecto mediante los siguientes comandos:

```
mkdir proyecto1
cd proyecto1
```

En segundo lugar, se inicializó PlatformIO con el siguiente comando:

```
platformio init --board=uno
```

Por último, luego de ejecutar el comando `ls`, se puede observar que la estructura del proyecto en la figura 6 es similar a la estructura generada cuando se creó un proyecto desde la extensión de PlatformIO IDE de VSC.

```
juanp@juanp-VirtualBox: ~/Escritorio/proyecto1$ ls
include lib platformio.ini src test
```

Figura 6. Estructura del proyecto de PlatformIO generada con comandos.

El programa se debe alojar en la carpeta src. Como ejemplo, se creó y abrió un archivo con extensión .ino para realizar el programa deseado mediante el siguiente comando:

```
# sudo nano hola.ino
```

Para ejecutar comandos de PlatformIO, se debe posicionar en el directorio de proyecto fuera de src, por lo tanto se ejecutó el siguiente comando:

```
# cd ..
```

Luego, para compilar se ejecutó:

```
# platformio run
```

Finalmente, se subió el programa a la placa de Arduino UNO mediante el comando:

```
# platformio run --target upload
```

1) Instalación y configuración MJPG-Streamer

En esta sección, se detallan los pasos de instalación y configuración de MJPG-Streamer.

- Se debe comprobar si estaba instalado git en el sistema operativo. Para ello, primero se debe actualizar el sistema.
`sudo apt update`
`sudo apt upgrade -y`
- Luego se puede comprobar si está instalado:
`git --version`
- En caso de no estar instalado, se debe instalar ejecutando el comando:
`sudo apt install git`
- Luego se debe instalar cmake, libjpeg, g++ porque son dependencias necesarias antes de instalar mjpg-streamer.
`sudo apt-get install cmake libjpeg8-dev`
`sudo apt-get install gcc g++`
- Finalmente se debe clonar el repositorio de la aplicación mjpg-streamer mediante los siguientes comandos.

```
cd ~/git clone
https://github.com/jacksonliam/mjpg-streamer.git
```

- Por último, se debe instalar mjpg-streamer mediante los siguientes comandos:

```
cd ~/mjpg-streamer/mjpg-streamer-experimental
make
sudo make install
```

Configuración e inicialización

MJPEG-Streamer no tiene un archivo de configuración, pero todos los elementos de configuración deben pasarse al programa con argumento de comando. El siguiente comando sirvió para inicializar el programa y configurarlo al mismo tiempo.

```
export LD_LIBRARY_PATH=.
```

```
./mjpg-streamer -o "output_http.so -w ./www -p 8080" -i "input_uvc.so -d /dev/video2 -r 1280x720 -fps 30 -q 10"
```

Complemento de salida (output_http)

[w- | --www]...: Carpeta que contiene páginas web en jerarquía plana (sin subcarpetas)

[-p | --port]...: Puerto TCP para el servidor HTTP. En este caso es el 8080.

Complemento de entrada (input_uvc)

[-d | --device]...: Dispositivo de video para abrir.

Adicional: Se ejecutó el siguiente comando para ver si el sistema operativo reconocía la cámara conectada.

```
lsusb
```

Se obtuvo la siguiente salida (donde el sistema reconoce la cámara utilizada):

```
Bus 001 Device 007: ID 80ee:0030 VirtualBox  
VirtualBox Webcam - eFace 1300
```

[-r | --resolution]...: Es la resolución del dispositivo de video que se está empleando. En este caso, el valor correspondiente es 1280x720.

[-f | --fps]...: Fotogramas por segundo. El valor elegido en base a los requerimientos del sistema es 30.

2) Creación del sitio web inicial

Se fue desarrollando el sitio web en base a las necesidades del laboratorio remoto desarrollado.

Inicialmente, se realizó una página web simple donde el usuario subía un archivo (extensión. ino, cpp) al servidor. Del lado servidor, el desafío era que este archivo se compile mediante la herramienta de PlatformIO. La primera aproximación se realizó mediante un script de bash que se puede observar en la figura 7.

```
$ compile.sh x
html > static > bash > $ compile.sh
1  #!/bin/sh
2  cd /var/www/html/static/function/project/src
3  chmod -R 777 ./
4  cd ..
5  sudo platformio run
6
7
```

Figura 7. Script de Bash para compilar programas a través de PlatformIO.

En la figura 8 se muestra la parte de la interfaz web que permite subir un archivo al servidor y compilar dicho archivo a la placa de Arduino

UNO, conectada al servidor.

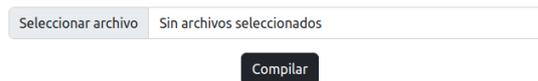


Figura 8. Botón para seleccionar un archivo (programa) y compilarlo.

Una vez que el usuario seleccione el archivo y haga clic en compilar, el formulario se enviará, ya que el botón compilar es de tipo submit (parámetro de la etiqueta HTML). En la figura 9, se observa una petición de ajax, donde se envía el archivo a través del método POST, al archivo llamado "compile.php".

```
$.ajax({
  url: 'static/function/compile.php', //RUTA A LA QUE SE DIRIGE
  type: 'POST',
  enctype: 'multipart/form-data',
  data: formData, //MANDA LOS DATOS DEL FORMULARIO.
  contentType: false,
  processData: false,
  cache: false,
  success: function (response) {
```

Figura 9. Petición a través de AJAX.

La ejecución del script se guarda en la variable "\$comando", esto es muy importante, ya que esta variable contendrá una cadena de caracteres, los cuales son el resultado de lo que devuelve cada comando ejecutado en el script de Bash. Esto sirve para saber si la validación de la compilación fue exitosa, por lo tanto, fue resuelto, simplemente con verificar si la cadena contiene "SUCESS". En caso de ser verdadero (Se ha compilado sin errores), se le devuelve una alerta al usuario para que sepa que la compilación fue exitosa. En caso de ser falso (Error en la compilación), se devuelve una alerta al usuario para que sepa que hubo un error en la compilación. En la figura 10 se observa la alerta que indica cuál es el resultado de la compilación.



Figura 10. Resultado de la compilación del programa.

En el caso de requerir subir el archivo a la placa de Arduino UNO, se debe seleccionar un botón de tipo "switch" como se observa en la figura 11.

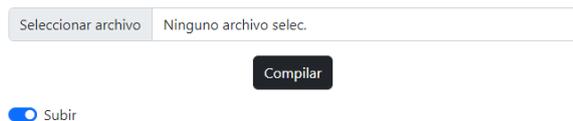


Figura 11. Botón switch.

Este botón tendrá valor 'on' en caso de encendido y 'off' en caso de apagado. Por lo tanto, se agregó al formulario que se envía mediante el método POST al archivo script de PHP (*compile.php*). Además, se creó una variable en dicho script de PHP, donde esta variable contendrá el valor del botón del switch: $\$check = \$_POST['checkupdate']$

Incorporación del Streaming a la página web

Teniendo en cuenta que se instaló y configuró MJPG-Streamer, la idea es poder incrustar mediante una etiqueta simple de html, la dirección del Streaming. En la línea siguiente, se encuentra la etiqueta `` de HTML; en esta, hay varios atributos. El más importante es el atributo *src*, ya que éste indica la URL de la imagen y, además, es obligatorio para este tipo de etiquetas. En este caso, la URL es la dirección donde se encuentra funcionando el MJPG-Streamer.

```

```

Como se tiene funcionando mediante un cliente No-IP, se accede a la cámara a través del dominio, sin olvidar indicar el puerto por el cual está transmitiendo la imagen:

“<http://laboremotounaj.ddns.net:8080/?action=stream>”
DOMINIO - **PUERTO** - **RUTA**
STREAMING

Finalmente, se puede observar en la figura 12 el funcionamiento de la cámara web transmitiendo en tiempo real.



Figura 12. Funcionamiento del Streaming.

Inclusión de un monitor serie en la página web

El monitor serial permite enviar y recibir mensajes de texto entre el kit arduino y la máquina servidor, estos son útiles para la depuración del código y también para el control de Arduino. Por ejemplo, es posible enviar comandos desde el ordenador para encender LEDs.

PHP dispone de una clase (<https://www.phpclasses.org/>) que se puede utilizar para comunicarse con un puerto serie en Linux o Windows.

Toma la ruta (como `"/dev/ttyS0"` para Linux o `"COM1"` para Windows) del dispositivo serie y verifica si es válido antes de abrir una conexión con él.

Una vez que se abre la conexión, puede enviar datos al puerto y leer las respuestas (la lectura solo se implementa para Linux).

La clase también puede cambiar los parámetros de conexión para el dispositivo serie dado.

Para hacer uso de la clase, en primer lugar, se creó un script PHP. En segundo lugar, se incluyó la clase `'PhpSerial.php'` en dicho script. En tercer lugar, se inicializo y se especificó el dispositivo serial. En cuarto lugar, se configuró los parámetros de tasa de baudios, la paridad, la longitud, los bits de parada y el control de flujo. Por último, se abrió el puerto serial y se comenzó con el envío y recepción de datos.

Teniendo en cuenta el funcionamiento de la clase *phpSerial.php*, para disponer de esta funcionalidad, se debe crear una ventana en la página web donde se muestran los mensajes en la consola serie de la placa Arduino. Para lograr esto, se planificó este módulo de tal forma que el usuario seleccione la tasa de baudios, (un dato importante a tener en cuenta a la hora de configurar el puerto serie), y los segundos, cuyo dato es relevante, ya que, a través de una función llamada **sleep**, propia de PHP, se puede retrasar la ejecución del programa durante un número de segundos para

finalmente capturar la mayor cantidad de datos provenientes del puerto serie.

En la figura 13 se muestra la ventana de configuración de la consola serie. Respecto a la tasa de baudios, se pueden seleccionar los siguientes valores: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200. Respecto a los segundos, es posible seleccionar: 5, 10, 20, 40, 60, 120.



Figura 13. Ventana del monitor serie.

Se tienen dos botones: Iniciar y Limpiar. La función del botón iniciar, técnicamente, lo que hace, es activar un evento (click) de JavaScript donde se dispara una petición POST a través del Motor de Ajax al servidor. El servidor toma en cuenta dos datos que se enviarán a través de la petición POST, uno es la tasa de baudios y otro son los segundos. El servidor maneja estos datos de la siguiente manera:

- La tasa de baudios se pasará por parámetro a la función `confBaudRate` proveniente de la clase `PhpSerial.php`
- Los segundos se pasarán a la función `sleep()` de PHP

Inclusión de un teclado matricial

Se agregó un teclado matricial al laboratorio remoto (ver figura 14). Este mismo se desarrolló utilizando la clase de "PhpSerial" de PHP. Así como se pueden leer datos del puerto serie, también se pueden enviar, mediante la siguiente línea de código:

```
$serial->sendMessage();
```

En la anterior línea de código, entre paréntesis, se coloca el mensaje a enviar a la placa de arduino que se tenga conectada mediante usb a la máquina. Teniendo en cuenta esto, cuando se presione una tecla del teclado matricial observado de la página web, se ejecutará una acción mediante JavaScript y el dato que corresponda a dicha tecla se enviará mediante

una petición de AJAX al servidor. En el lado servidor, se recibirá un dato (`$key=$_POST['key']`) en un script de PHP creado y este mismo dato se agregará entre paréntesis a la línea de código mencionada anteriormente.

Para emular el teclado, se dispone de un kit destinado exclusivamente a recibir los comandos y generar acciones sobre las entradas de la placa de arduino que se correspondan con el teclado.

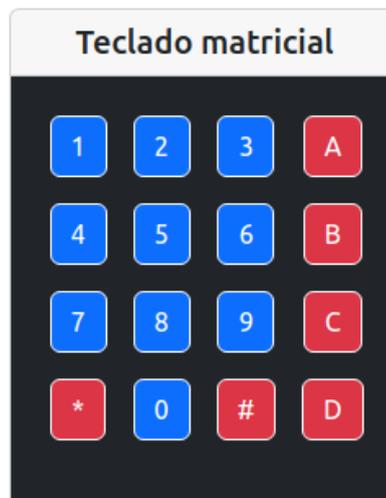


Figura 14 . Teclado matricial

III. RESULTADOS

En esta sección se describirá la validación realizada al laboratorio completo, desde el registro de usuario, el login, la selección de un turno y el acceso a la programación y carga de códigos en el kit de arduino.

En la figura 15, se muestra la interfaz de login que provee el laboratorio, con las opciones de ingresar usuario y contraseña o de crear un nuevo usuario.

Para la validación de contraseña, cuando se ingresa una contraseña nueva, se debe confirmar. En caso de que la confirmación sea distinta a la contraseña elegida como nueva, el sistema desactiva el botón de "Crear cuenta". Cuando se ingresa una contraseña nueva, se debe confirmar. En caso de que la confirmación sea distinta a la contraseña elegida como nueva, no se podrá crear la nueva cuenta.

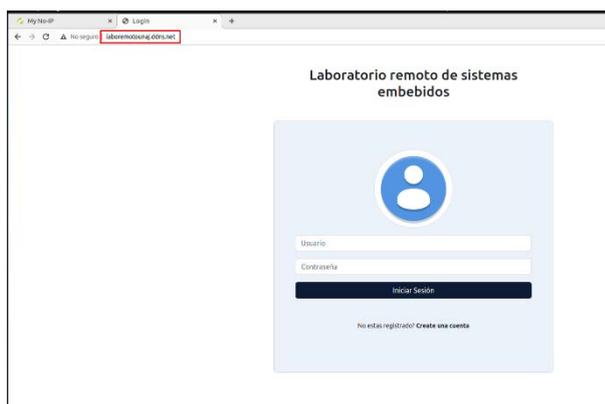


Figura 15. Funcionamiento de la página.

Finalmente, si se registra el usuario, por ejemplo, con las credenciales que se ven en la figura 16, la creación se concretará correctamente.

- Usuario: juan
- Contraseña: 1234
- Nombre: Juan
- Apellido: Piñeiro

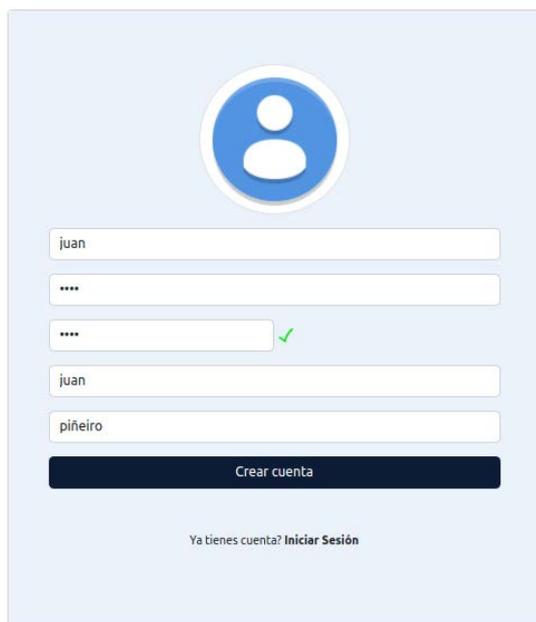


Figura 16. Registro de usuario.

El siguiente paso es validar el inicio de sesión, esto se realizó con el nuevo usuario. Se verificó la respuesta del sistema al ingresar usuario no registrado y contraseña incorrecta para comprobar que el sistema responde con un mensaje de acceso fallido. Por último, se ingresaron las credenciales correctas para el inicio de sesión y se pudo acceder a la pantalla de inicio que se ve en la figura 17.

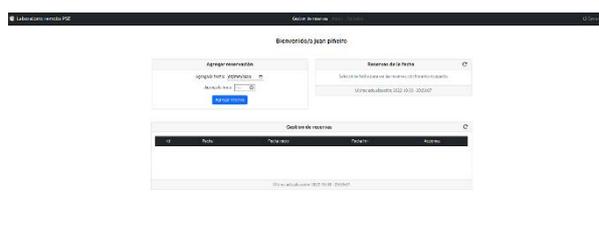


Figura 17. Pantalla de inicio.

El siguiente paso consistió en agregar una reserva ingresando la fecha y la hora, como se observa en la figura 18.



Figura 18. Agregar una reserva.

La reserva se agregará, luego de presionar el botón “aceptar”, a la tabla de gestión de reservas como se ve en la figura 19. En este caso, al haber sacado la reserva en la fecha y hora actuales, se desbloqueará la web del laboratorio remoto para el usuario, que podrá ingresar dando click al icono de candado verde.

Gestión de reservas				
id	Fecha	Fecha inicio	Fecha fin	Acciones
240	2022-10-30	21:00:00	21:30:00	

Última actualización: 2022-10-30 - 21:07:37

Figura 19. Tabla de gestión de reservas.

El último paso de validación de reserva consiste en seleccionar una fecha y hora ya ocupada. Supongamos que entra otro usuario llamado “test”. Este usuario agrega una reserva con la fecha 2022-11-06, con la hora de inicio 22:00 y por supuesto con la hora final 22:30. Luego, entra a su cuenta el usuario “juan” e intenta agregar la misma reserva que agregó el usuario “test”. Luego de esto sucederá lo siguiente:

Cuando el usuario “juan” seleccione fecha en el campo, en la ventana de “reservas de fecha” le saldrá todos los horarios ocupados para tal fecha, incluido el horario de la reserva hecha por el usuario “test”.

En caso de que el usuario “juan” intente agregar la reserva a la misma hora, le aparecerá una advertencia de que ya existe una reserva para esa fecha y hora, impidiendo agregar la reserva.



Figura 24. Funcionamiento del monitor serie.

IV. CONCLUSIONES

En el proceso de desarrollo del proyecto, se tuvo que realizar una investigación sobre las generaciones de la educación a distancia, hasta llegar a la actualidad donde se ve un fuerte impacto de la tecnología con modelos que se basan en conceptos como E-Learning o M-Learning. Fue importante tener en cuenta estos conceptos para entender la relación que se sostiene con la realización de un laboratorio remoto para la programación de sistemas embebidos.

Se vive en un mundo de constantes cambios tecnológicos, que dan como resultado nuevas formas de aprendizaje (especialmente a causa de la pandemia). Estas formas de aprendizaje fueron tenidas en cuenta a la hora de diseñar el sistema desarrollado. Además, se realizó una investigación sobre sistemas embebidos, microcontroladores y la plataforma de Arduino. Siendo esto crucial para poder comprender los conceptos que se abordan en la enseñanza de dichos sistemas.

Para cumplir con los requerimientos del sistema, fue fundamental el lenguaje de programación del lado servidor (PHP) ya que ofrece en funciones simples la capacidad de poder ejecutar Scripts de bash (lo que permitió implementar la mayoría de las funcionalidades del sistema).

Uno de los mayores retos presentados consistió en solucionar la cuestión de la superposición al momento que los alumnos ingresan al laboratorio remoto, algo que no tiene que ver con la funcionalidad del laboratorio, pero que es indispensable para proveer un acceso correcto y ordenado. Para esto se consideraron las siguientes alternativas:

- Realizar la gestión de los turnos desde Moodle y agregar un link de acceso al servidor del laboratorio.
- Plataforma VCL (significa Laboratorio de Computación Virtual). Es una plataforma informática en la nube gratuita y de código abierto con el objetivo principal de ofrecer

entornos informáticos personalizados y dedicados a los usuarios.

Luego de analizar en detalle cada una de las alternativas, ninguna fue convincente, principalmente porque el objetivo final es que este laboratorio forme parte de la red de laboratorios remotos financiada por el confedi y porque las plataformas utilizaban versiones de tecnologías obsoletas. Por lo tanto, se optó por diseñar e implementar un Sistema de Gestión de Reservas propio que funcione en el mismo servidor.

En cuanto a los resultados obtenidos, estos fueron los esperados respecto a los objetivos planteados, ya que todos los módulos en conjunto funcionaron correctamente. Se logró crear un sistema de gestión de turnos, simple e intuitivo para el usuario, subir varios ejemplos a la placa de Arduino UNO, visualizar los resultados a través de la webcam y utilizar el monitor serie que lee datos de la Placa Arduino UNO. Por otro lado, respecto al teclado matricial (para enviar datos), se implementó el código, pero falta el hardware para hacer pruebas futuras.

Respecto a la propuesta, se pudo cumplir con los objetivos propuestos y afrontar los problemas que se fueron presentando para finalmente obtener resultados satisfactorios. El presente laboratorio se considera de gran utilidad para la enseñanza de sistemas digitales, principalmente en relación con la programación de sistemas embebidos que incluye la lectura de sensores, accionamiento de actuadores y la visualización de mensajes mediante consola serie, display LCD o leds indicadores.

En cuanto a los laboratorios remotos, se debe destacar que actualmente hay una amplia variedad de carreras y cursos que se dictan de forma virtual, para este tipo de carreras es indispensable que las prácticas se puedan realizar a distancia sobre equipamiento real. Este trabajo busca cubrir esa necesidad, siendo el puntapié inicial para la creación de una gran variedad de laboratorios con características similares.

REFERENCIAS

- [1] Luis Joyanes Aguilar, "Industria 4.0, la cuarta revolución industrial, Alfaomega, 2016. ISBN: 978-607-622-942-2
- [2] Leila Fatmasari Rahman, "Choosing your IoT Programming Framework: Architectural Aspects", 2016 IEEE 4th



International Conference on Future Internet of Things and Cloud (IF 10.11/JCR 2016)

- [3] "MQTT: The Standard for IoT Messaging" mqtt.org (accedida en junio de 2022)
- [4] Cabado Leonel, Osio Jorge, David Mársico, Lucas Olivera, Cappelletti Marcelo, Morales Martín, "Sistema IoT para el monitoreo y seguimiento de la calidad del agua en arroyos", VII conaiisi, 2020
- [5] Jorge R. Osio, Marcelo Cappelletti, Mauro Salina, Mauro Gomez, Leonel Navarro, Juan Salvatore, Daniel Alonso, Diego Encinas, Martín Morales. "Tecnologías de la información y las comunicaciones mediante IoT aplicadas a soluciones en el medio productivo y medioambiental", UNPA, El Calafate, WICC 2020.
- [6] J. Osio, M. Cappelletti, G. Suárez, J. Salvatore "Diseño de aplicaciones de IoT para la solución de problemas en el medio socio productivo", UNSJ, San Juan, WICC 2019.



Diseño de Consolas para el Aprendizaje Lúdico hacia la Programación de Microcontroladores PIC y Algunos Periféricos de Interfaz con el Usuario

1st Helen Zambrano

Universidad ECCI

Bogotá, Colombia

helenan.zambranona@ecc.edu.co
ORCID iD: 0009-0006-8714-4956

2nd Nury Moreno

Universidad ECCI

Bogotá, Colombia

nuryl.morenor@ecc.edu.co
ORCID iD: 0009-0008-1071-4476

3rd Robin Blanco

Universidad ECCI

Bogotá, Colombia

rblancoc@ecc.edu.co
ORCID iD: 0000-0002-6105-3689

Resumen— Este documento describe el desarrollo de un proyecto que integra conceptos de programación, como lo es la escritura de algoritmos básicos y avanzados que permitan controlar los componentes dentro de una consola de videojuegos. Conocimientos de electrónica, en el diseño y elaboración del circuito, el cual se componen principalmente de un microcontrolador PIC, pantalla LCD caracteres y teclado matricial. Por último, el diseño mecánico, el cual consiste en la construcción de una caja tipo arcade para darle un aspecto más atractivo al usuario. Su implementación como herramienta para un aprendizaje lúdico, permitirá que resulte más agradable y motivador para el estudiante adquirir conocimientos sobre el funcionamiento del microcontrolador PIC, el manejo de la pantalla LCD caracteres, el teclado matricial, y adicionalmente, el diseño de tarjetas y algoritmos.

Palabras clave: Microcontroladores PIC, pantalla LCD, teclado matricial, juegos.

I. INTRODUCCIÓN

En la década de los años '70', mientras Intel desarrollaba el primer microprocesador estable de 8 bits [1], Gary Boone y Michael Cochran dos ingenieros de Texas Instruments, crearon el primer microcontrolador, el TMS 1000. Este dispositivo comenzó como un

microprocesador con una función de memoria RAM y ROM que requería de una programación especializada, por lo que era necesario quitar el dispositivo del circuito [2].

Con el propósito de orientar este dispositivo al control de entradas y salidas (E/S), la compañía General Instruments crea el microcontrolador PIC (Peripheral Interface Controller). Diez años más tarde se integraría una memoria EEPROM o flash, inventada por Toshiba en 1980 [3], lo que permitió que los microcontroladores pudieran ser programados y reprogramados eléctricamente, sin necesidad de ser removidos del circuito. Esto dio lugar a lo que hoy conocemos como el microcontrolador PIC, siendo Microchip Technology uno de sus principales fabricantes [4].

Estos dispositivos son utilizados actualmente con mayor frecuencia, debido a la facilidad de su uso, versatilidad, bajo costo y eficiencia. La evolución que ha tenido esta tecnología, ha permitido la aplicación en diversos sectores, simplificando procesos y proporcionando mayor precisión y eficacia en la realización de tareas complejas.

Este trabajo se propone como una guía inicial para el aprendizaje lúdico de programación de microcontroladores PIC en lenguaje C, utilizando en su desarrollo periféricos de interfaz con el usuario. Tiene la

finalidad de generar interés, fomentar la motivación y facilitar el aprendizaje de conocimientos teóricos y prácticos mediante la creación de algoritmos de videojuegos.

II. MARCO TEÓRICO

A. Lenguaje de programación C

Es considerado uno de los lenguajes más importantes en la actualidad [6], ya que es esencial en el desarrollo de aplicaciones y es la base de muchos de los actuales lenguajes, como lo son Java, C++ o C#. Gracias a su alta flexibilidad, se permite migrar a diferentes sistemas operativos y ser usado para la creación de aplicaciones, juegos, hojas de cálculo, etc.

B. Microcontrolador PIC

Son circuitos integrados que actúan como computadores digitales, y que permiten ser programados para ejecutar instrucciones automáticamente[7]. En su interior cuentan con un microprocesador o una CPU, una memoria para almacenar instrucciones, una memoria para almacenar datos y puertos de entrada/salida[8].

- Microcontrolador PIC18F4550

Cuenta con arquitectura Harvard, en la cual, se leen y ejecutan simultáneamente las instrucciones y datos, lo que lo hace ideal para aplicaciones que requieran de alto rendimiento, eficiencia y capacidad de respuesta inmediata.

Puertos de entrada y salida: El PIC18F4550 cuenta con 40 pines en su empaquetado PDIP, de los cuales 34 pines pueden ser utilizados como entradas y 31 como salida. Inicialmente, no todos los pines empiezan como entradas y salidas digitales (13 de ellos empiezan como análogos)[9].

Los puertos RB y RD cuentan con 8 pines, por lo que se les considera puertos completos. Mientras que los puertos RA, RC y RE son considerados incompletos, debido a que los dos primeros cuentan con 7 pines, y el último únicamente, véase figura 1.

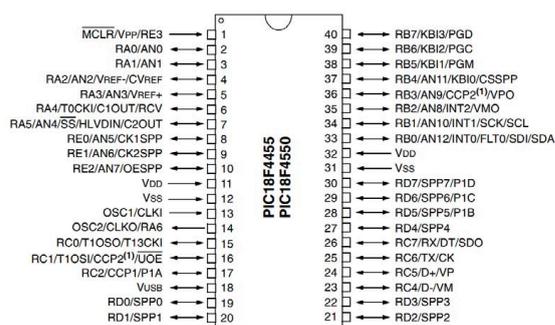


Fig. 1. Distribución de pines PDIP.

Es importante mencionar que, cuenta con un oscilador interno, el cual se puede configurar según se requiera, siendo su capacidad máxima de operación de 8MHz. Además, cuenta con dos pines destinados para el uso de un oscilador externo, el cual tiene una capacidad máxima de 48Mhz.

C. Videojuegos

Son programas diseñados para entretener al usuario a través de periféricos que simulan situaciones en una pantalla[12]. Los videojuegos integran elementos gráficos, sonoros e interactivos que sumergen al jugador en experiencias dinámicas[13].

- Ahorcado: Es un juego clásico en el que el objetivo es adivinar una palabra antes de un número determinado de intentos, cada vez que se falla, se irá dibujando una parte del ahorcado en la pantalla.
- Memoria: Es un juego de concentración, en el que el usuario debe recordar y repetir una secuencia de colores y/o formas en un orden específico. A medida que avanza el juego, la dificultad va incrementando, debido a que la secuencia se hace más larga.
- Picas y fijas: Se trata de un juego de adivinanzas, en el que el jugador debe encontrar el número de 4 dígitos en la menor cantidad de intentos posible, recibiendo en cada intento el número de dígitos que se encuentran en la posición correcta (fijas) y la cantidad de dígitos que se encuentran en el número a adivinar, pero no en la posición correcta (picas)[10].
- Parejas: Se trata de un juego de

memoria, en el que el objetivo es encontrar todos los pares de cartas idénticas. Las cartas se encuentran boca abajo hasta que el usuario selecciona dos, si estas son iguales, se retiran del tablero, si no, se vuelven a voltear.

- **Buscaminas:** El objetivo es despejar todas las casillas de un tablero que no oculten una mina. El usuario recibirá pistas al pulsar sobre las celdas, se despejará una zona y números que indican cuántas minas hay en celdas cercanas[11].

D. Aprendizaje Lúdico

Se trata de un método de enseñanza [14], donde el objetivo es generar un ambiente agradable y llamativo al estudiante al momento de aprender temas que podrían resultar difíciles, mejorando no solo su capacidad de concentración, sino también su compromiso y persistencia.

III. METODOLOGÍA

Inicialmente, se desarrollaron diagramas de flujo para representar gráficamente los algoritmos de los juegos a crear, véase figura 2,3,4,5 y 6.

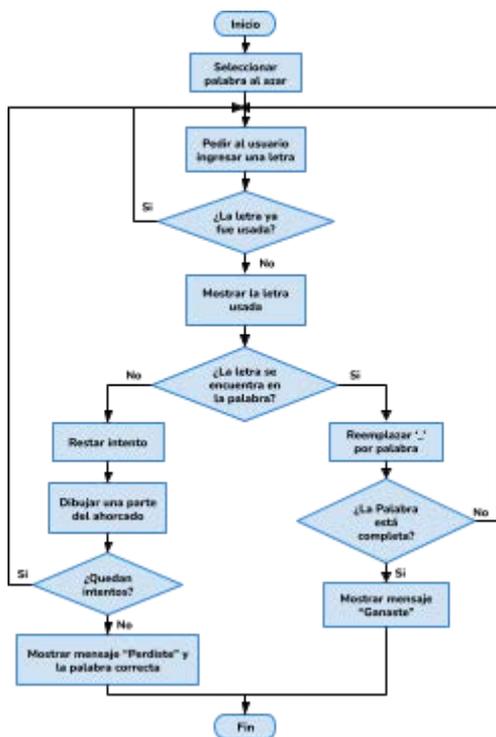


Fig. 2. Diagrama de flujo 'Ahorcado'.

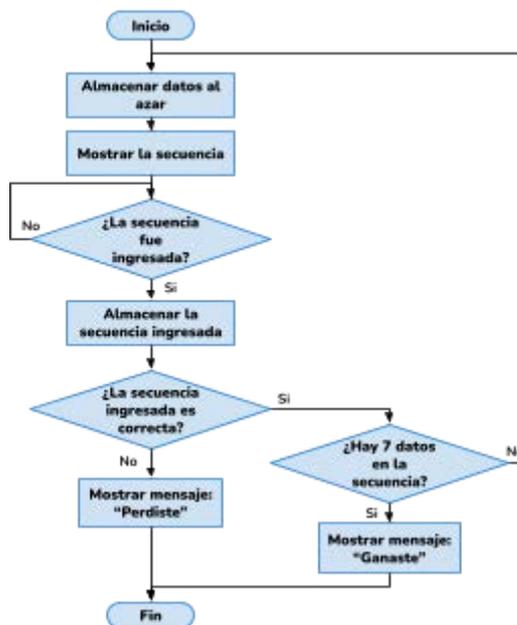


Fig. 3. Diagrama de flujo 'Memoria'.

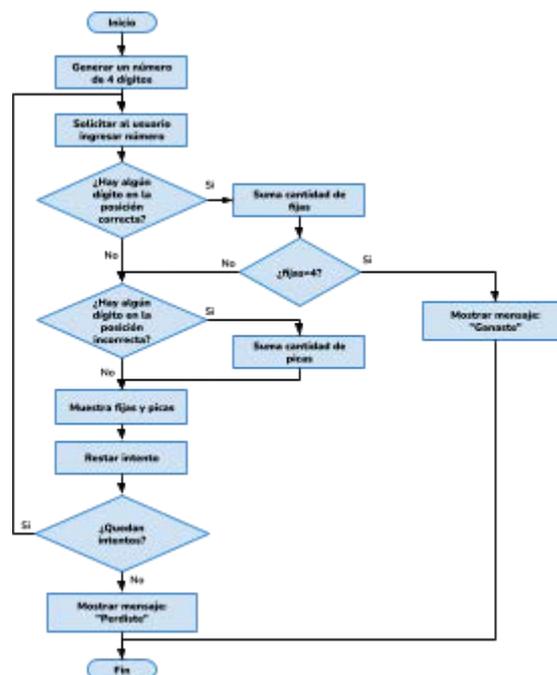


Fig. 4. Diagrama de flujo 'Picas y Fijas'.

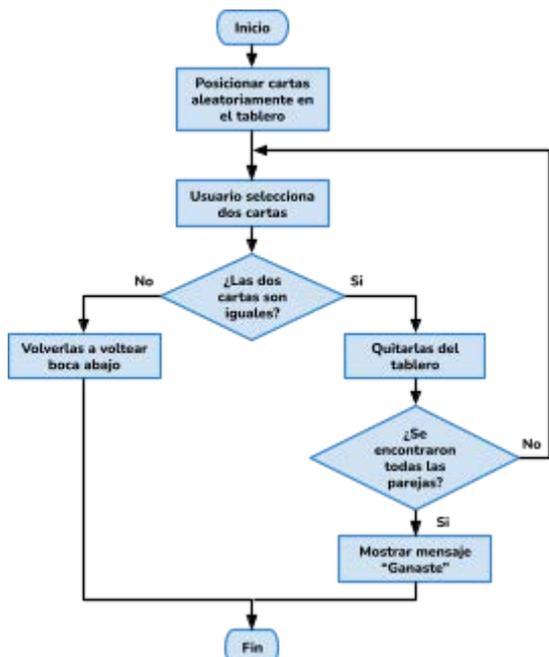


Fig. 5. Diagrama de flujo 'Parejas'.

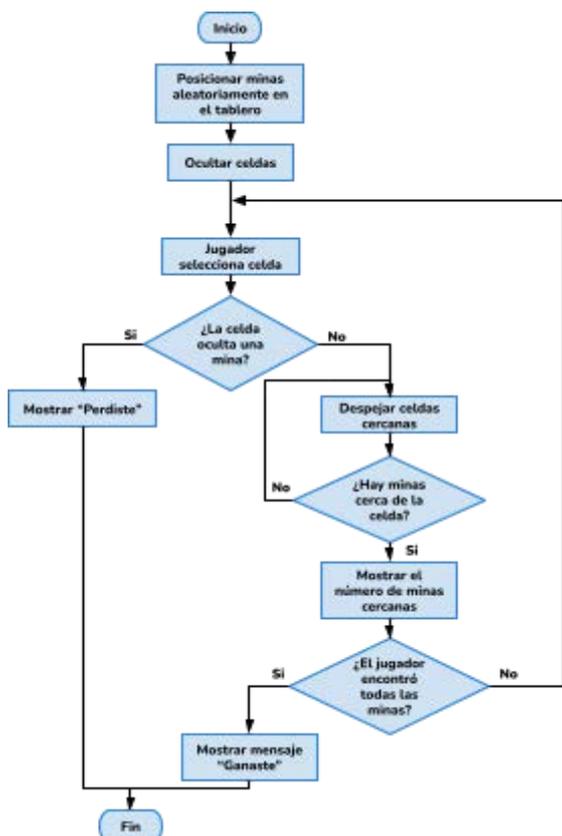


Fig. 6. Diagrama de flujo 'Buscaminas'.

Posteriormente, se realizó la implementación del sistema electrónico usando Proteus [15], lo que permite una correcta conexión y visualización de los componentes del circuito. Este proceso facilita su simulación, y asegura la fabricación de un diseño de PCB preciso,

véase figura 7 y 8.

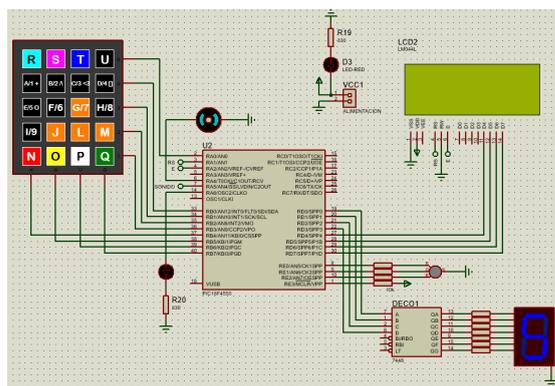


Fig. 7. Diagrama Circuitual.

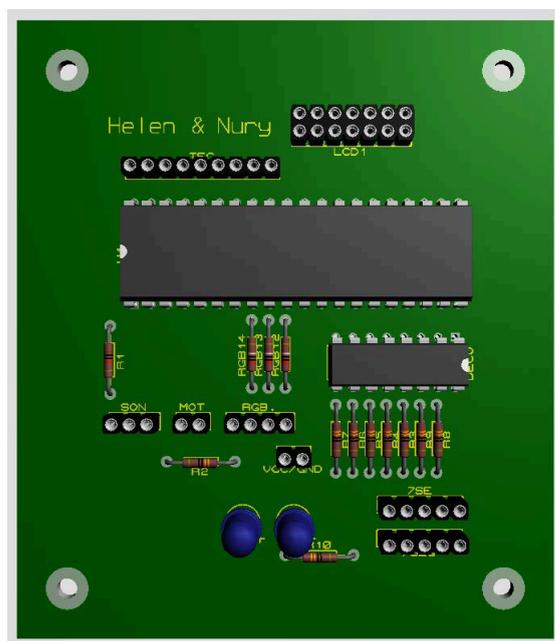


Fig. 8. Visualización PCB 3D (vista 1).

A continuación, se presenta un listado de los principales componentes:

- Pantalla LCD caracteres (4x20)
- Teclado matricial (5x4)
- Display 7 segmentos
- Motor vibrador
- Módulo reproductor de sonido ISD1820
- Led RGB

Paralelamente, se realizó la programación de cada uno de los juegos utilizando el entorno de desarrollo MPLAB X, con la ayuda de la simulación previamente mencionada. Esta simulación permitió verificar el correcto funcionamiento de los juegos antes de implementarlos en un circuito real.

A través de las herramientas proporcionadas por MPLAB X, se configuraron las entradas y salidas del microcontrolador, permitiendo así la interacción directa entre el jugador y la consola. Además, se manejaron interrupciones para gestionar la lógica del juego y garantizar una respuesta rápida a las entradas del usuario.

Finalmente, se diseñó el prototipo de la consola haciendo uso del programa SolidWorks y librerías descargadas en la web [18], véase figura 9, 10, 11 y 12.

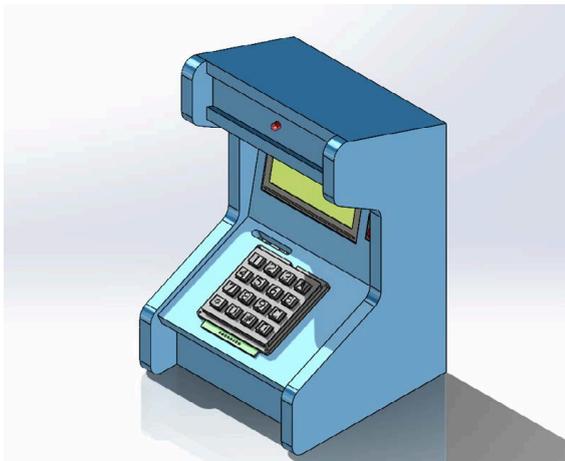


Fig. 9. Diseño Mecánico (vista 1).

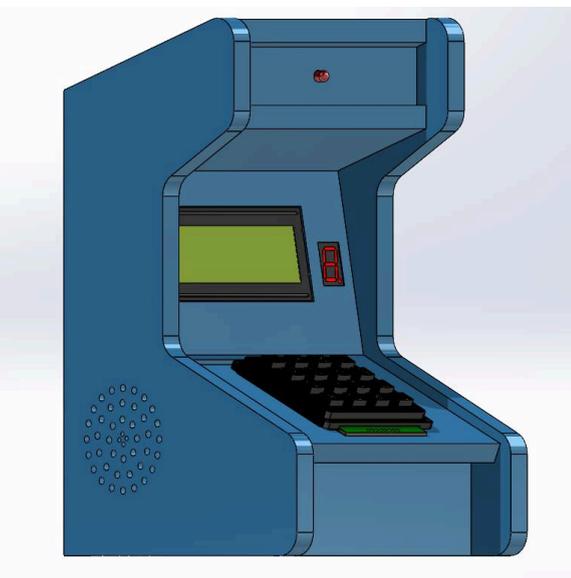


Fig. 10. Diseño Mecánico (vista 2).

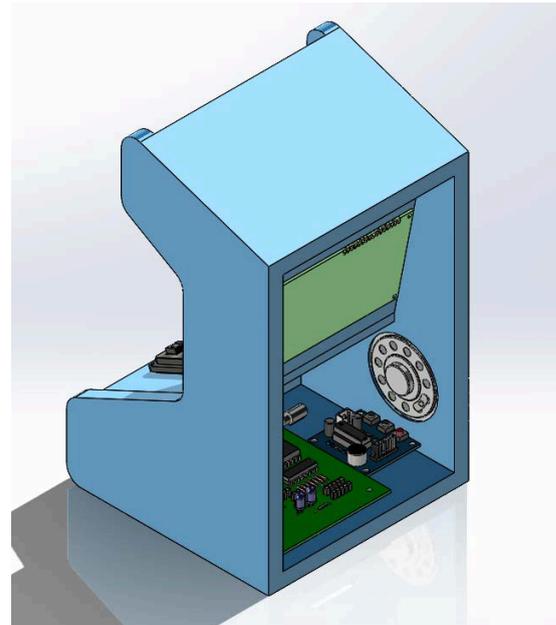


Fig. 11. Diseño Mecánico (vista 3).

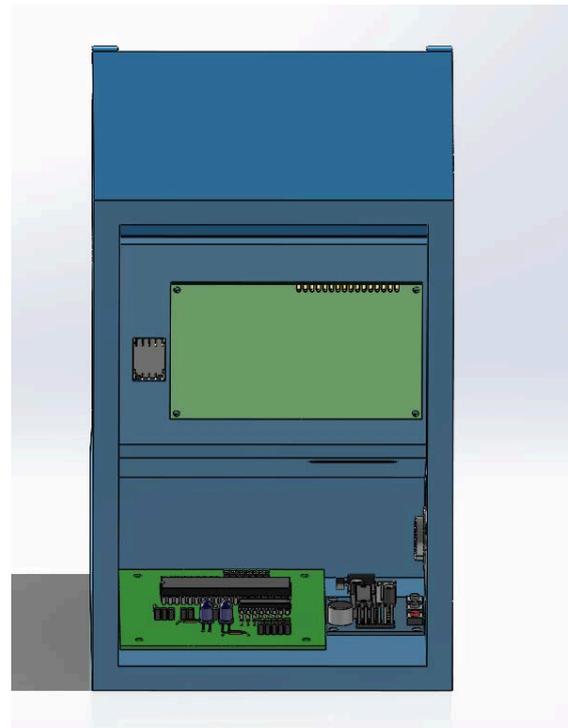


Fig. 12. Diseño Mecánico (vista 4)

IV. RESULTADOS

Como resultado, se ha desarrollado el diseño de una consola de videojuegos, partiendo desde la programación hasta su fabricación mecánica y electrónica. A continuación se muestra el funcionamiento del programa haciendo uso de la simulación de proteus.

En primer lugar, se muestra un mensaje de bienvenida, véase figura 13



Fig. 13. Mensaje de bienvenida.



Fig. 16. Ahorcado.

A continuación, se desplegará un menú en donde el usuario podrá escoger el juego que deseé, véase figura 14.

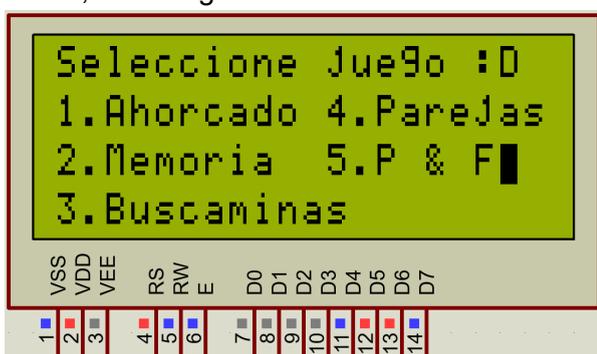


Fig. 14. Menú principal.

Al seleccionar el juego, el usuario podrá elegir entre jugar, observar las instrucciones o salir de este menú, véase figura 15.

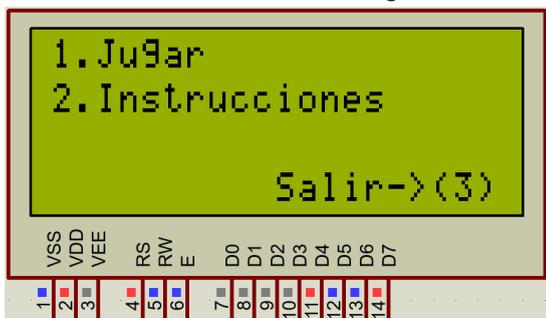


Fig. 15. Menú secundario.

Si el usuario desea jugar al ahorcado, se mostrará en la pantalla los espacios de la palabra que tendrá que completar para ganar el juego. Al equivocarse se irá completando el dibujo del ahorcado a la derecha, véase figura 16.

Los intentos restantes se mostrarán en un display de 7 segmentos. Si el usuario no logra completar la palabra antes de terminar todos los intentos, se mostrará en la pantalla el mensaje "Perdiste" y el led se encenderá de color rojo, véase figuras 17 y 18.

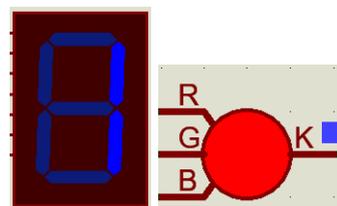


Fig. 17. Número de intentos y RGB.

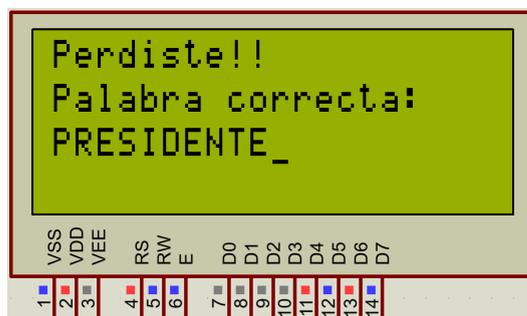


Fig. 18. Mensaje "Perdiste".

Después de mostrarse el mensaje, se desplegarán dos opciones, las cuales permiten jugar nuevamente o salir del juego, véase figura 19.

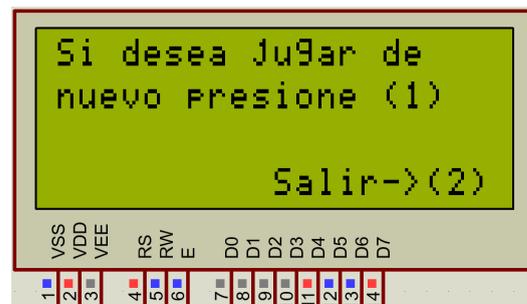


Fig. 19. Menú jugar nuevamente.

En el caso del que el usuario desee jugar memoria, se mostrará tres opciones de dificultad, véase figura 20.



Fig. 20. Selección de dificultad.

Cada una de las opciones mostrará una secuencia de colores y/o formas, véase figura 21.

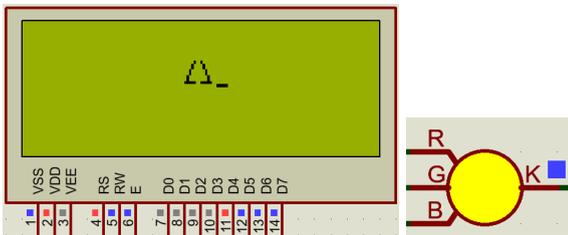


Fig. 21. Formas y colores.

Después de visualizarse la secuencia, se pide al usuario que ingrese las formas y/o colores en el orden indicado. El usuario podrá ingresar su secuencia, mientras se muestre en la pantalla "Tu turno", véase figura 22.



Fig. 22. Mensaje "Tu turno".

Si el jugador consigue ingresar la secuencia de 7 colores y/o figuras correctamente, se mostrará en la pantalla el mensaje "Ganaste", véase figura 23.



Fig. 23. Mensaje "Ganaste".

Si el usuario desea jugar Picas y fijas, se mostrará en la pantalla el número ingresado, seguido de la cantidad de picas y fijas encontradas, véase figura 24.



Fig. 24. Picas y fijas.

En este juego, la cantidad de intentos corresponde a nueve. Si el usuario logra encontrar el número correcto, el led se encenderá de color verde, véase figura 25.

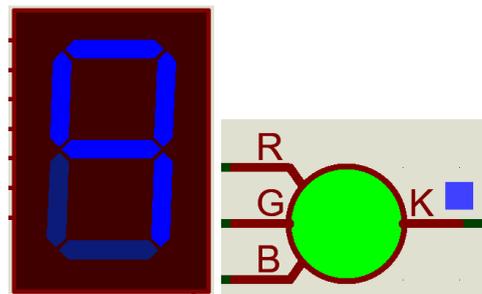


Fig. 25. Cantidad de intentos y RGB.

- CONCLUSIONES

A partir de lo anteriormente mencionado, es posible concluir que, generar un ambiente tranquilo e inspirador, depende no solamente del entorno físico, sino también de las herramientas y métodos empleados en el proceso de aprendizaje. Gracias a esto, recurrir a sistemas atractivos para el estudiante, como los juegos clásicos, pueden desarrollar habilidades beneficiosas como lo

son la concentración, el compromiso y la persistencia. Este proyecto contará con una guía, que puede ser utilizada como una estrategia pedagógica en las aulas, además de fomentar el autoaprendizaje en los estudiantes.

REFERENCIAS

- [1] Instituto Nacional de Educación Tecnológica. (s.f.). *Microprocesadores y microcontroladores*. <http://bnm.me.gov.ar/giga1/documentos/EL007282.pdf>
- [2] *Historia de los Microcontroladores y sus Fabricantes*. (s.f.). VISTRÓNICA SAS. <https://www.vistronica.com/blog/post/historia-de-los-micro-controladores-y-sus-fabricantes.html>
- [3] *Recordando la historia de los dispositivos de memoria*. (s.f.). Rochester Electronics. <https://www.rocelec.mx/noticias/recordando-la-historia-de-los-dispositivos-de-memoria>
- [4] *microcontroladores pic*. (s.f.). MIKROE. https://www.mikroe.com/ebooks/microcontroladores-pic-programacion-en-c-con-ejemplos/microcontroladores-pic?srsltid=AfmBOoo03L5m37TJkaO7HL5ioxI7NTOyONwf32TN70uPMSA_LnMV-f2A
- [5] López, G., & Margni, S. (2003). *Funcionamiento de microcontroladores*. https://www.academia.edu/36851516/Introducción_al_funcionamiento_básico_de_microcontroladores_PIC
- [6] López, M. (2024, 17 abril). Lenguaje de programación C cumple 50 años - Immune Institute. Immune Technology Institute. <https://immune.institute/blog/lenguaje-de-programacion-c/>
- [7] TECmikro. (s.f.). Introducción a la programación de microcontroladores PIC. TECmikro Ecuador. <https://tecmikro.com/content/8-programacion-microcontroladores-pic>
- [8] Torres Torriti, M. (2007, 26 de abril). TUTORIAL MICROCONTROLADORES PIC. [sitios.ing. https://mtorrest.sitios.ing.uc.cl/downloads/pic/tutorial_pic.pdf](https://mtorrest.sitios.ing.uc.cl/downloads/pic/tutorial_pic.pdf)
- [9] MICROCHIP. (2006). *PIC18F2455/2550/4455/4550 Data Sheet*. <https://www1.microchip.com/downloads/en/devicedoc/39632c.pdf>
- [10] Picas y Fijas. (s.f.). Sphere Online Judge (SPOJ). <https://www.spoj.com/PA6610/problems/PIYFI.pdf>
- [11] Cómo se juega al Buscaminas: aprende de una vez por todas. (s.f.). Alfa Beta Juega. <https://www.mundodeportivo.com/alfabeta/guia/como-se-juega-buscaminas#:~:text=El%20objetivo%20del%20Buscaminas%20es,la%20proximidad%20de%20una%20mina>
- [12] Carricay, G. (2017, 9 de octubre). ¿Qué son los videojuegos? Medium. <https://medium.com/grupo-carricay/qué-son-los-videojuegos-os-d640dcb6aa84>
- [13] Escudero, G., Hiegeo, D., Tena, N., & Moreda, I. (s.f.). Videojuegos. Repositori UJI. <https://repositori.uji.es/xmlui/bitstream/handle/10234/5993/Videojuegos.pdf?sequence=1>
- [14] Pontificia Universidad Católica de Chile. (2022, 16 de junio). Aportes del Juego para el Aprendizaje en Educación Superior - Centro de Desarrollo Docente UC. Centro de Desarrollo Docente UC. <https://desarrollodocente.uc.cl/2022/06/16/aportes-del-juego-para-el-aprendizaje-en-educacion-superior/>
- [15] Labcenter Electronics. (s.f.). PCB Design and Circuit Simulator Software - Proteus. <https://www.labcenter.com/>
- [16] Inicio. (s.f.-b). SOLIDWORKS. <https://www.solidworks.com/>

- [17] GrabCAD. (s.f.). GrabCAD Making Additive Manufacturing at Scale Possible. <https://grabcad.com/dashboard>
- [18] My SolidWorks. (s.f.). MySolidWorks - Official SOLIDWORKS Community. <https://my.solidworks.com>



Aproximaciones sobre el diseño curricular para la formación en sistemas embebidos

Marcelo Giura

Universidad Tecnológica Nacional
Buenos Aires

Departamento de Ing. electrónica

Buenos Aires, Argentina

mgiura@frba.utn.edu.ar

Marcelo Trujillo

Universidad Tecnológica Nacional
Buenos Aires

Departamento de Ing. electrónica

Buenos Aires, Argentina

mtrujillo@frba.utn.edu.ar

Nahuel González

Universidad Tecnológica Nacional
Buenos Aires

Departamento de Ing. electrónica

Buenos Aires, Argentina

ngonzalez@frba.utn.edu.ar

Resumen— Se propone examinar el diseño curricular de la asignatura Informática II en la carrera de Ingeniería Electrónica en la Universidad Tecnológica Nacional, con foco en la formación en sistemas embebidos. La asignatura en cuestión integra contenidos de programación avanzada, acceso a hardware y desarrollo de sistemas embebidos mediante el uso de C y C++. La metodología de enseñanza se basa en enfoques constructivistas y cognitivistas, promoviendo el aprendizaje significativo a través de proyectos grupales y herramientas de modelado como uModelFactory. La evaluación incluye trabajos de laboratorio, presentaciones y exámenes parciales. A su vez, la propuesta se adapta a la virtualidad sincrónica, incorporando métodos como el aula invertida para favorecer el aprendizaje. Se destaca la importancia de la articulación entre asignaturas y la necesidad de una visión integral que prepare a los estudiantes para los desafíos profesionales, alineándose con el perfil profesional esperado en la Ordenanza 1849.

Palabras clave: diseño curricular, ingeniería electrónica, enseñanza de sistemas embebidos.

I. INTRODUCCIÓN

Se propone llevar adelante un trabajo de exploración de contenidos y competencias requeridas por parte de los estudiantes de ingeniería electrónica en el abordaje de los procesos de enseñanza y aprendizaje de sistemas embebidos situado en la materia Informática II. De forma de comenzar con el análisis de la asignatura, resulta conveniente contextualizarla en el diseño curricular de la

carrera.

Informática II es la asignatura integradora del segundo nivel de la carrera de Ingeniería Electrónica. Se trata de una asignatura anual con una carga horaria de 160 horas-cátedra organizada en 32 clases de 5 horas-cátedra cada una.

Esta asignatura pertenece al área de Técnicas Digitales, la cual se encuentra compuesta por:

- Informática I (primer año)
- Informática II (segundo año)
- Técnicas digitales I (tercer año)
- Técnicas digitales II (cuarto año)
- Técnicas digitales III (quinto año)

El diseño curricular en la Universidad Tecnológica Nacional (UTN) plantea que las materias integradoras en sus diferentes niveles permiten acercar al estudiante desde el comienzo de la carrera a la profesión y a la metodología del trabajo profesional.

En ese sentido, propone para las materias integradoras de los primeros niveles (primero, segundo y tercero) los siguientes objetivos:

- o Conocer los problemas básicos de la ingeniería abordando problemas desde el inicio.
- o Introducir a los estudiantes en conceptos generales de la ingeniería.
- o Valorar la competencia de los profesionales de la ingeniería elegida.



Los procesos de enseñanza y aprendizaje son procesos complejos donde los diferentes actores y su interacción tienen un rol fundamental. Las materias integradoras posibilitan la integración de contenidos propiciando la incorporación de nuevos saberes a partir del conocimiento previo de los estudiantes.

A su vez, favorecen el trabajo grupal, el análisis de situaciones problemáticas de la vida real y fomentan la capacidad de reflexión de los estudiantes.

El análisis de la asignatura Informática II toma de referencia el nuevo plan de estudio aprobado. El plan 2023 corresponde al nuevo diseño curricular de Ingeniería Electrónica (ordenanza 1849), el cual toma de base la ordenanza 1753 la cual incorpora el nuevo enfoque sobre las actividades reservadas y alcances.

En dicho plan se establecen los objetivos de la carrera y en particular el perfil profesional de la misma, como así también las competencias de egreso, tanto genéricas como específicas.

II. ANÁLISIS

Análisis de los objetivos

La formación en competencias puede potenciar y enriquecer los procesos de enseñanza y aprendizaje en el ámbito universitario. Dicha conceptualización se apoya en una mirada constructivista, donde las competencias se construyen en interacción con otros y en contextos específicos.

Es así como las competencias brindan mejores posibilidades para articular la teoría y la práctica, promueven el uso de estrategias colaborativas y ponen en el centro de la escena los avances y logros de los estudiantes en su proceso de aprendizaje.

Competencias de egreso específicas a las que contribuye:

Competencia	Competencias de Actividades reservadas			Competencias de Alcances
	Baja	Media	Alta	
CE 1.1: Diseñar, proyectar y calcular sistemas, equipos y dispositivos de generación, transmisión y/o procesamiento de campos y señales analógicos y digitales; circuitos integrados ; hardware de sistemas de cómputo de propósito general y/o específico y el software a él asociado; hardware y software de sistemas embebidos y dispositivos lógicos programables; sistemas de automatización y control; sistemas de procesamiento y de comunicación de datos y sistemas irradiantes, para brindar soluciones óptimas de acuerdo a las condiciones técnicas, legales, económicas, humanas y ambientales.			X	
CE 1.2: Plantear, interpretar, modelar y resolver los problemas de ingeniería descritos en CE 1.1.		X		
CE 2.1: Proyectar, dirigir y controlar la construcción, implementación, mantenimiento y operación de lo mencionado en CE 1.7 *.		X		

Fig. 1. Competencias de egreso específicas

Competencias de egreso genéricas a las que contribuye:

Competencia	Baja	Media	Alta
Identificar, Formular y resolver problemas de Ingeniería Electrónica		X	
Utilizar de manera efectiva las técnicas y herramientas de aplicación de la Ingeniería Electrónica.		X	
Desempeñarse de manera efectiva en equipos de trabajo.		X	
Comunicarse con efectividad.	X		
Aprender en forma continua y autónoma.	X		

Fig. 2. Competencias de egreso genéricas

Los objetivos de la asignatura están plasmados de forma de dar cuenta y evidenciar su relación con las competencias desde la asignatura en cuestión y sus contenidos específicos.

Dentro de la asignatura en análisis se encuentran sus objetivos definidos a partir de las instancias de aprendizaje y los logros a alcanzar.

- Diseñar y desarrollar aplicaciones para resolución de problemas complejos aplicados a ingeniería electrónica.
- Utilizar fluidamente herramientas de documentación, de control de versiones, y de automatización de la construcción de un programa o biblioteca a partir de las fuentes.
- Adquirir y aplicar los principios de la programación orientada a objetos utilizando el lenguaje C++ para resolver problemas.
- Evaluar la pertinencia del uso de un lenguaje estructurado o uno orientado a objetos para la resolución de problemas.
- Desarrollar interfaces gráficas sobre



framework Qt para realizar aplicaciones en donde se necesite la interacción mediante puerto serie entre la PC o un dispositivo móvil y un microcontrolador.

- Conocer las estructuras avanzadas de datos utilizadas más frecuentemente para el desarrollo de programas en C y C++.
- Adquirir las herramientas necesarias para modelar en forma adecuada Sistemas de Control Reactivos.
- Adquirir habilidades de programación de sistemas embebidos en bajo nivel para el desarrollo de programas que utilicen acceso a registros de periféricos, mediante lenguaje C y C++.
- Conocer y aplicar técnicas para el manejo de periféricos de uso común, como displays, conversores AD y DA y puertos serie, para resolver aplicaciones comunes en sistemas embebidos.

El Consejo Federal de Decanos de Ingeniería (CONFEDI) define competencias generales de ingreso y egreso [1], dividiéndolas en competencias básicas, transversales y específicas.

Al analizar los objetivos planteados, los mismos orientan acerca de cuáles son las competencias que debe desarrollar en esta etapa un estudiante de ingeniería electrónica, profundizando luego las mismas a lo largo de su trayecto de formación.

El desarrollo de competencias en el proceso de formación requiere repensar las estrategias de enseñanza y aprendizaje, de forma que el estudiante pueda alcanzar los objetivos propuestos, tal es así que también debemos considerar si la forma de evaluación resulta consistente con nuestra metodología de enseñanza.

A partir de este punto, se comenzará a realizar el análisis correspondiente a los contenidos, metodología y evaluación con relación a la asignatura Informática II.

Análisis de los contenidos

A continuación, se presentan los contenidos mínimos de la asignatura:

Contenidos mínimos

- Programación avanzada utilizando el lenguaje de programación introducido en Informática I
- Introducción a las estructuras dinámicas de datos.
- Herramientas de construcción de software
- Control de periféricos y/o comunicación con microcontroladores.
- Desarrollo de aplicaciones usando interfaces de usuario gráficas
- Desarrollo de aplicaciones utilizando cálculo numérico.
- Introducción a un lenguaje de programación avanzado.
- Introducción a los fundamentos de los sistemas operativos avanzados.

Se observa dentro de los contenidos la existencia de puntos en común con otras asignaturas tanto del mismo nivel (articulación horizontal) como de otros niveles (articulación vertical). Este análisis se profundizará en la sección "Articulación con la asignatura".

A lo largo de los últimos años, la asignatura ha evolucionado productos de concursos, cambios internos en la cátedra y una Dirección de Cátedra abierta y proactiva, posibilitando así un programa analítico acorde a las necesidades de formación de nuestros estudiantes. En particular, en el presente año y con la ordenanza 1849, se formaliza y consolida el trabajo continuo que se viene desarrollando desde la cátedra.

A partir de los conocimientos en lenguaje C que los alumnos traen consigo al cursar la asignatura Informática II, adquiridos previamente en Informática I, se decidió profundizar los paradigmas desarrollados en esta asignatura. En particular, el objetivo fue avanzar más allá del paradigma de programación orientada a procedimientos (POP) y crecer en los paradigmas de programación dirigida por eventos (PDE) y programación orientada a objetos (POO).

Aunque la transición no presentó grandes dificultades y permitió un aumento en la



cantidad y complejidad de los ejercicios, la programación orientada a objetos (POO) se incrementó en una serie de nuevos conceptos que inicialmente limitaron el alcance de la práctica y la ejercitación. Para superar este desafío, se optó por abandonar la programación en lenguaje C para sistemas embebidos y adoptar C++, que facilita la implementación de la POO.

La transición a C++ no solo permitió la incorporación temprana y efectiva de los conceptos de POO, sino que también favoreció una práctica más significativa. Este cambio ha aportado beneficios adicionales, como un refuerzo continuo de los conceptos fundamentales necesarios para el desarrollo de software en sistemas embebidos. Además, el uso de C++ ha permitido una mejor alineación con las prácticas actuales de la industria, proporcionando a los estudiantes una experiencia más relevante y actualizada.

Dado el extenso contenido y la profundidad de los temas abordados, es importante que los alumnos se familiaricen con las herramientas y técnicas modernas desde etapas tempranas de su formación. Esto asegura una comprensión sólida y una preparación adecuada para los desafíos profesionales en el campo de la ingeniería electrónica.

Contenidos analíticos

-Unidad temática 1: El Lenguaje C++ y las Estructuras Avanzadas de Datos. Introducción a la programación orientada a Objetos - Lenguaje C++ como evolución respecto del lenguaje C - Tipos abstractos de datos: Clases - Clases y Objetos - Miembros públicos y privados - constructores y destructores - Concepto de encapsulamiento. Entrada/Salida en C++. Operador visibilidad - Operadores para gestión dinámica de memoria: new y delete - Sobrecarga de funciones y operadores. Funciones y clases friend - uso del apuntador this - calificador const - especificador de clase de almacenamiento: extern y static. Herencia: tipos (hincapié en herencia pública) - clase base y derivada - miembro protected - constructores y destructores en la clase derivada. Contenedores de datos - Aplicaciones

con Estructuras Avanzadas de Datos: Colas, Pilas, Listas. Funciones Virtuales y Virtuales puras - Clases Abstractas - Concepto de Interface.

-Unidad temática 2: Entornos Gráficos

Introducción a la programación en entornos gráficos - Bibliotecas y entornos de desarrollo: Caso particular de QT y el QT Creator. - Introducción a la programación gobernada por eventos y las señales (Signals and Slots) de QT. - Formularios Básicos: Cuadros de Diálogo y aplicaciones de formularios sencillas. - Widgets mas comunes: Button, Check box, Radio button, Menu bar, Toolbar, Scrollbar, Text box, Combo box, Label y Otros. Ejemplos de aplicación: Cientes de sistemas computacionales (PC , bases de datos, Sistemas embebidos, etc.) comunicación serie, representaciones gráficas (data logger).

-Unidad temática 3: Introducción a los microcontroladores y los Sistemas Embebidos.

Concepto de Sistema Embebido. Presentación de periféricos elementales: Puertos de entrada/salida, Contador, Timer, UART, etc. Presentación del Kit de la Cátedra. IDE para microcontroladores - Compilación modular - Metodología de proyectos - Diagrama de capas: drivers, primitivas, aplicación - portabilidad. Concepto de Sistema Operativo en Tiempo Real. Aplicaciones elementales sobre el Kit de la Cátedra. Diagrama de capas.

-Unidad temática 4: Introducción a la arquitectura de un Microcontrolador. Caso de uso: Cortex M0+ y su programación en C++

Microcontrolador: Análisis de un circuito inteligente básico. Integración de periféricos en un solo chip. Familia CórteX M0+. Presentación del mapa de memoria y periféricos. Técnicas de acceso. Identificación de Registros del Core y de sus periféricos. Especificadores de memoria. La Switch Matrix (SWM), registros y operación.

-Unidad temática 5: Introducción a GPIO.

GPIO (General Purpose Input Output): Configuración de los puertos de entrada/salida de propósito general: tratamiento como entrada, tratamiento como salida, tratamiento bidireccional. Registros asociados. Lectura de entradas digitales simples - Acción sobre



salidas digitales. Consideraciones teóricas de las entradas y las salidas de los microcontroladores. Desarrollo de funciones utilitarias en C++ para el manejo de GPIO.

-Unidad temática 6: Polling (encuesta) e Interrupciones

Idea general - Estrategias de atención - Funciones de interrupción - Poling vs. Interrupciones - Registros asociados - Vector de interrupciones - Prioridades - Distribución de tiempos de ejecución para el programa principal y las funciones de interrupción - Mínimo tiempo admisible entre interrupciones - Pasaje de información entre el programa principal y las funciones de interrupción. Fuentes interrupción externas (en pines).

-Unidad temática 7: Contadores / Temporizadores

Contador - Características principales - Temporizador: Caso particular de contador - Utilización como Contador (acumuladores de eventos) - Utilización como Temporizador (Base de Tiempo) - Multiplicadores de la base de Tiempo (ticks). Implementación - Registros asociados - Modos de funcionamiento. scheduler: Principios de funcionamiento. Caso de estudio: El systick del core del Cortex M.

-Unidad temática 8: GPIO avanzado.

Desarrollo del firmware y primitivas asociadas a Display de 7 segmentos y a la lectura y depuración mediante técnicas de absorción de transitorios de teclados lineales, teclados matriciales y entradas digitales de microswitchs. Escritura de salidas digitales. Display LCD.

-Unidad temática 9: Programación Gobernada por Eventos

Introducción a los diagramas secuenciales con múltiples acciones temporizadas. Desarrollo de funciones facilitadoras: TimerStart(), TimerStop(), TimerClose(), etc. Máquina de Estados: Diagrama de globos. Implementación con switch-case. Implementación con múltiples if. Implementación con punteros a función. Modelización y Síntesis de problemas mediante la utilización de máquinas de estado - Casos de automatismos independientes dentro de un mismo equipo - Utilización de máquinas de estados en paralelo: ventajas. Análisis de casos

de aplicación típicos. Resolución de problemas mediante la combinación de Máquinas de estado y Diagramas secuenciales.

-Unidad temática 10: Comunicación Serie.

Necesidad de la comunicación serie - Serializadores y paralelizadores - Comunicación serie asincrónica: Conceptos. Velocidad de transmisión - Registros asociados en el microcontrolador - Modos de operación - Buffers de Rx y Tx: Pilas y Colas circulares - Implementación de protocolos punto a punto y multipunto - Estrategias de programación por polling e interrupciones. Desarrollo del firmware y primitivas asociadas a la comunicación serie.

-Unidad temática 11: Introducción a la Medición y Generación de Señales Analógicas

Introducción a la medición de magnitudes analógicas (temperatura, presión, humedad, etc.) - Conversores ADC: y DAC características principales - Registros asociados. Estrategias de programación por pooling y por interrupciones - Interpretación de los valores obtenidos (Tablas, filtros, etc.) - Eliminación de valores espurios: Filtros de media móvil y de mediana. Desarrollo del firmware y primitivas asociadas a la medición y generación de señales analógicas. Aplicaciones algorítmicas haciendo uso de métodos numéricos tradicionales. Soluciones aproximadas. Precisión. Errores.

Este análisis se compone por tres (3) grandes bloques:

- o Programación orientada a objetos
- o Metodologías de acceso a hardware
- o Programación avanzada en C y C++

Programación orientada a objetos

Se trata de un bloque transversal a toda la asignatura ya que los temas abordados en las primeras cuatro clases se trabajarán desde una mirada diferente, donde su utilización no solo radicará en una aplicación, basada en clases, que correrá en una PC sino que se resignificará este paradigma de programación aplicándolo al diseño de sistemas embebidos. Esta mirada parte de la idea de aprendizaje significativo planteada por Ausubel [2,3] y en concordancia con la conceptualización de las competencias



plasmadas en el nuevo plan de estudio.

Metodologías de acceso a hardware

A lo largo del primer cuatrimestre, el estudiante enfrenta por primera vez la necesidad de acceder/manejar/utilizar un hardware independiente. En Informática I los estudiantes aprendieron los rudimentos básicos del lenguaje C utilizando como plataforma una PC y un sistema operativo de base (Linux).

El estudiante se encuentra en esta nueva etapa, donde no hay un sistema operativo que oficie de administrador y proteja de los equívocos de un programador inexperto, al menos en este contexto de uso.

Se enseñan y se ponen en práctica diferentes formas de acceso al hardware, manipulando registros a través del uso de punteros, complementando el trabajo mediante macros o la utilización de uniones y campos de bits.

A su vez, se proponen diferentes formas de comunicación con los periféricos evaluando su uso (poling, interrupciones e introducción a DMA).

Programación avanzada en C y C++

Una vez fijadas las bases sobre el acceso a hardware se propone utilizar un modelo de programación que permita la organización de nuestro código y maximice su portabilidad. Se define en este bloque un modelo de capas donde se trabajará en primitivas y aplicación, introduciendo e implementando diferentes metodologías de trabajo como por ejemplo maquinaria de timers (por software). El núcleo de este bloque termina de definirse a través de la programación orientada a eventos permitiendo así la introducción del uso de diagramas de estado para el modelado de un sistema embebido.

Distribución de la carga horaria entre actividades teóricas y prácticas

Modalidad organizativa de las clases	Horas reloj presenciales	Horas reloj virtuales	Horas totales
Teórica	72	0	72
Formación práctica	48	0	48
Total de horas			120

Fig. 3. Distribución de la teoría y práctica

Tipo de prácticas	Horas Relej totales presenciales	Horas reloj totales virtuales
Formación experimental	24	0
Problemas abiertos de Ingeniería (ABP)	15	0
Proyecto y diseño	9	0
Otras:	-	-
Práctica supervisada	0	0
Total de horas	48	0

Fig. 4. Distribución por tipo de práctica

III. METODOLOGÍA DE ENSEÑANZA Y EVALUACIÓN

La aproximación a la enseñanza toma los aportes de los enfoques pedagógicos constructivistas y cognitivistas. Desde estas concepciones se trata de generar situaciones problemáticas para que los estudiantes pongan en juego sus ideas previas e interactúen con los nuevos saberes de forma tal que puedan construir nuevos conocimientos producto de esta interacción.

Se trata de promover procesos cognitivos que permitan un abordaje integral para poder ir avanzando hacia la construcción de un proyecto integrador cuya presentación se realizará al final del curso.

Se privilegia un enfoque grupal desde el punto de vista de la interacción de los estudiantes. Como estrategia metodológica complementaria, se utiliza las presentaciones como disparador temático en soportes de diapositivas (PPT) para poner en disponibilidad ideas previas por parte de los estudiantes a modo fomentar aprendizajes significativos [2,3].

La utilización de herramientas de modelado y simulación como **uModel Factory** desarrollado en el marco del Proyecto de Investigación y Desarrollo del Departamento de Electrónica, permiten introducir el tema de Programación Gobernada por Eventos favoreciendo el rol activo del estudiante y permitiendo la autorregulación del aprendizaje.

En el marco de la cátedra de Informática II de la carrera de Electrónica de la UTN FRBA, se formó un equipo dedicado al desarrollo y depuración de software orientado a sistemas embebidos. Este esfuerzo culminó en la creación de una herramienta llamada **uModelFactory**.

uModelFactory es un software diseñado para

asistir en la elaboración de diagramas y Máquinas de Estado Finitas (MEF) tipo Moore, las cuales están gobernadas por eventos. A pesar de que en el estado actual del arte existen diversas herramientas similares, ninguna ofrece una perspectiva didáctica que subraye la importancia de las máquinas de estado en la resolución de problemas relacionados con la implementación de sistemas de control de máquinas y equipos. Esta dimensión didáctica es esencial para el proceso de enseñanza-aprendizaje.

Por ello, hemos implementado un sistema con tres niveles de ayuda que se adaptan a la curva de aprendizaje del usuario. Las principales características de uModelFactory son:

- Descripción mediante Modelado Dirigido por Modelos (MDE) de sistemas que responden a estímulos y ejecutan tareas específicas.
- Posibilidad de dividir el sistema en múltiples MEF concurrentes.
- Simulación del funcionamiento del sistema.
- Análisis intrusivo del comportamiento del sistema.
- Niveles de ayuda diferenciados.
- Generación de código en C para integración en proyectos de sistemas embebidos.
- Generación de documentación del proyecto en diversos formatos (PDF, DOC, HTML).

El enfoque actual sobre la metodología de enseñanza de la asignatura permite al estudiante introducirse a los sistemas embebidos, aprender diferentes metodologías de desarrollo y gestionar su auto-aprendizaje, favoreciendo así el aprendizaje significativo.

Repensando nuestras prácticas docentes luego de la pandemia

A partir de la experiencia de virtualidad sincrónica realizada durante los dos años de Pandemia, y considerando las posibilidades tecnológicas, a partir del ciclo lectivo, la asignatura incorpora la modalidad de dictado donde se contempla la virtualidad sincrónica.

En este contexto, se propone incorporar en la programación didáctica de la asignatura la

estrategia de clases dictadas en virtualidad sincrónica, apoyándose en los recursos institucionales y tecnológicos con los que ya contamos donde por supuesto se destaca la plataforma Moodle, punto de encuentro asincrónico entre estudiantes y docentes a partir del Aula Virtual.

A través de esa estrategia, se busca alcanzar el objetivo de implementar la metodología de Aula Invertida, la cual obtuvo buenos resultados durante el período de Pandemia, especialmente en 2021 dado lo aprendido durante el dictado en ASPO durante 2020.

La metodología de aula invertida [4] se basa en considerar diferentes momentos para el aprendizaje que no quedan circunscriptos exclusivamente al momento de la clase sincrónica.



Imagen 1. Clase invertida

La gran generación de materiales didácticos en video, infografías interactivas y presentaciones realizados durante 2020 y 2021 que, ahora editados, constituyen verdaderos objetos de aprendizaje, serán la base para el momento **ANTES**. La posibilidad de incorporar en la instancia previa estos materiales, permite, entre otras cosas, que el estudiante pueda asimilar los conceptos a su ritmo. El video se puede ver, parar, volver a revisar, en el caso de las infografías las mismas se pueden recorrer de forma autónoma, se puede tomar nota y analizar los materiales. Esto colabora con privilegiar el ritmo del propio estudiante. Además, los estudiantes deberán realizar actividades específicas de evaluación formativa (autoevaluaciones) previstas en el Aula Virtual a tal efecto.

El momento **DURANTE** se constituirá en el



espacio de reflexión y consolidación de los temas, así como también el del pasaje a la práctica de aquellos temas que así lo requieran. Este momento, siempre sincrónico, podrá estar mediado por tecnologías de video comunicación, tipo Zoom. En particular, la elección de la plataforma se encuentra alineada con la política institucional en términos de acuerdos y convenios.

Específicamente los encuentros planificados para realizar actividad práctica y las evaluaciones (aproximadamente un 60% de la carga horaria), se realizan de manera presencial física en los laboratorios del Departamento de Ing. Electrónica, utilizando los recursos adecuados, a saber: el stick LPC845BRK, el kit didáctico de la cátedra en conjunto con el IDE MCUXpresso o QtCreator y el software uModel Factory según corresponda. El 40% restante se realizará, también de manera sincrónica, por medio de una plataforma de video comunicación (Zoom) cumpliendo con las condiciones requeridas para ser considerada una presencialidad remota.

El momento **DESPUÉS** será, como en la modalidad tradicional, de mayor responsabilidad por parte de quien aprende. Se incluye allí la realización nuevamente de la evaluación formativa prevista para cada módulo a los efectos de poder valorar el aprendizaje realizado. Las métricas del resultado de las evaluaciones citadas son generadas automáticamente por Moodle y de fácil revisión por parte del equipo docente.

El dictado de clases con modalidad de virtualidad sincrónica será anunciada y explicada el primer día de clase (contrato pedagógico) a los y las estudiantes, a quienes se les informará que en caso de no disponer de los medios para cumplimentar con los requisitos tecnológicos para participar con sesión de video y audio en los encuentros mediados por tecnologías, podrán dirigirse a la sede del Departamento de Electrónica donde se le facilitarán los equipos para poder participar en igualdad de condiciones.

La evaluación

El sistema de evaluación actual es presentado al inicio del ciclo lectivo y se compone por:

- Realización y aprobación del 75% de los trabajos prácticos de laboratorio.
- Presentación, defensa y aprobación del Trabajo Práctico Obligatorio.
- Aprobación de los 2 parciales en primera instancia o recuperación.

En el caso de la presente asignatura, si el estudiante cumple con las condiciones administrativas necesarias puede llegar a la condición de promoción. En ese sentido, aquellos estudiantes que han logrado alcanzar los objetivos propuestos en Informática I y han aprobado el final, poseen una gran ventaja al comenzar la cursada de Informática II.

A lo largo de la cursada los y las estudiantes deben realizar cuatro trabajos prácticos de laboratorio que se detallan a continuación:

- TPL1: Uso de IDE - Acceso a hardware - interrupciones – Systick
- TPL2: Teclado, display, entradas digitales
- TPL3: Ejercitación Máquina de Estados
- TPL4: Práctica comunicación serie asincrónica y ADC

La aprobación del TPO consistirá en:

- Presentación de un informe del TP FORMAL según formato de cátedra.
- Demostración del funcionamiento del proyecto y defensa INDIVIDUAL de cada integrante del trabajo realizado.

Requisitos de regularidad:

- Los dos parciales aprobados con 6 o más puntos.
- Cumplimentar las exigencias de laboratorio (TPL), u actividad supletoria requerida por el auxiliar docente en caso de ausencia del alumno a algún TPL.
- TPO aprobado
- Cumplir con los requisitos de asistencia establecidos por la Universidad

<p>CRITERIO 3A la Res. 276/17 del CD FRBA</p>	<p><u>Característica principal:</u> 2 (dos) instancias de evaluación con nivel de exigencia creciente.</p> <p><u>Instrumento de evaluación para la promoción:</u> las evaluaciones mencionadas.</p> <p>La suma de sus calificaciones para promocionar debe ser de 15 (quince) puntos o más, siempre y cuando la primera instancia de evaluación esté aprobada y la calificación de la segunda instancia de evaluación sea de 8 (ocho) puntos o más.</p> <p>Se puede alcanzar la promoción hasta incluso en la recuperación de las evaluaciones parciales (ver configuración RPAD)</p> <p style="text-align: center;">3 opcionales APLICAN: AN – PC y RPAD</p>
---	---

Fig. 5. Requisitos de aprobación directa

Notas:

RPAD (Cantidad de Recuperatorios permitidos para la Aprobación Directa): 1(uno) solo durante diciembre del año de cursada.

PC (Pruebas Complementarias): Es una nueva evaluación para indagar exclusivamente sobre los temas que en algún examen parcial no alcanzaron el nivel suficiente. Se aplica exclusivamente a alumnos que ya tienen aprobados los parciales (notas 6 o 7) pero no les alcanza para promocionar. Se admite el complemento de una sola de las evaluaciones parciales.

AN (Actualización de notas): Si el estudiante se presenta a la recuperación de un parcial con el objeto de levantar la nota aspirando a promocionar, pone en juego la calificación obtenida previamente (en caso de que tuviera el parcial aprobado (notas 6 o 7) en primera instancia.

IV. ARTICULACIÓN DE LA ASIGNATURA

Con el área

Informática II es la asignatura integradora del segundo nivel de la carrera de Ingeniería Electrónica. Se trata de una asignatura anual con una carga horaria de 160 horas-cátedra organizada en 32 clases de 5 horas-cátedra cada una.

Esta asignatura pertenece al área de Técnicas digitales, la cual se encuentra compuesta por:

- Informática I (primer año)
- Informática II (segundo año)
- Técnicas digitales I (tercer año)
- Técnicas digitales II (cuarto año)
- Técnicas digitales III (quinto año)

Entre Informática I e Informática II existe una transición coherente de primero a segundo año donde los estudiantes comienzan a transitar los primeros pasos en el lenguaje C y la

arquitectura de una computadora para luego reconocer las similitudes y diferencias al trabajar sobre sistemas embebidos.

Respecto a la articulación con las asignaturas de los niveles superiores se encuentra como punto en común el desarrollo de modelos y de metodologías de desarrollo que permiten afianzar los saberes del estudiante. En la actualidad, se observa una continuidad de los temas que abordados en Técnicas Digitales II y Técnicas Digitales III.

Aunque existe un salto abrupto con Técnicas Digitales I, donde si bien pertenecen a la misma área, los estudiantes encuentran una discontinuidad.

Articulación con el nivel

Tal como se describió previamente, dentro del segundo nivel de la carrera de Ingeniería Electrónica, Informática II es la materia integradora. En función del rol establecido en el plan de estudios, resulta conveniente articular los contenidos con asignaturas del mismo nivel.

A partir de los puntos en común evaluados tras la lectura de los programas analíticos de las asignaturas del mismo nivel, se proponen a continuación diferentes disparadores que permitan la integración y aprehensión del conocimiento.

Sobre el inicio de segundo cuatrimestre se aborda la metodología de la programación gobernada por eventos, en dicho tema se plantea el uso de modelos.

Decimos que un modelo es una representación de la realidad, capaz de describir lo que un observador establezca dentro del alcance de ese modelo.

Es posible utilizar el concepto de modelo como disparador e invitar a los estudiantes a reflexionar sobre el alcance de los modelos que han conocido a lo largo de las primeras materias, es decir, modelos físicos, modelos químicos, aproximaciones matemáticas, tal es así que en la asignatura Análisis de Señales y Sistemas aprenderán a modelar sistemas físicos desde diferentes enfoques, variando el



“lente” a utilizar en función de la respuesta a evaluar (tiempo, frecuencia).

Con relación a aplicaciones de otras materias dentro de Informática II, es posible al presentar la utilización del conversor analógico / digital (ADC), introducir el concepto de filtro, diferenciar un filtro analógico de uno digital, evaluar la implementación de un filtro simple (filtro de media móvil) en código C para luego reflexionar sobre el alcance y funcionamiento de ese filtro, como afecta el orden del conversor, etc.

Se propone dados los coeficientes de un filtro digital (Análisis de señales y sistemas), realizar la implementación de este en C o C++ y discutir su implementación en una PC y en un sistema embebido en función de los tipos de datos utilizados (tipos de variables) y el error cometido por redondeo.

Como actividad complementaria, resulta interesante realizar la propagación de error sobre una medición (articulación con Física II / Probabilidad y Estadística), reflexionando sobre el tipo de señal (fuente) y su orden de magnitud, en comparación con el piso de ruido.

Articulación con el diseño curricular

Los estudiantes en sus primeros años de carrera encuentran dificultades sobre la visión de esta, es por ello que las materias integradoras cumplen un papel fundamental [5,6].

- Los estudiantes requieren de ejemplos concretos de aplicación de los temas abordados en ciencias básicas.
- Los estudiantes que no son técnicos se encuentran desorientados con relación a la práctica.
- La práctica llega al recibirse, uno aprende a ser ingeniero cuando ya se recibió.

Es de destacar, el proceso que se viene desarrollando desde el Departamento de Ing. Electrónica desde el año 2006 donde se han realizado cambios en las asignaturas, a su vez, se ha incorporado nuevo instrumental y kits de desarrollo que posibilitan más y mejores

prácticas de laboratorio y la realización de talleres específicos

En la actualidad, los contenidos de Informática II resultan atractivos y motivadores para los estudiantes, son de aplicación directa y encuentran una materia que realmente cumple con su designación de integradora.

V. REFLEXIONES FINALES

De la Ordenanza 1849 que regula la carrera Ing. Electrónica Plan 2023 se desprenden no solo los objetivos de la carrera, las competencias de egreso (genéricas y específicas) sino en particular el perfil esperado del Ingeniero en Electrónica.

Perfil del Ingeniero y la Ingeniera de la UTN:

Su formación está orientada al manejo, aprovechamiento, cuidado y conocimiento de los recursos, en base a las expectativas y necesidades de la región iberoamericana.

La competencia de autoformación y la flexibilidad para aceptar la naturaleza permanente de los cambios son parte de su formación profesional con capacidad de innovación para atender el impacto que tienen en la región los dinámicos cambios del conocimiento, la obsolescencia de las tareas profesionales, los virajes en la orientación geoeconómica, los acuerdos sobre protección del ambiente y las crecientes demandas de participación democrática y desarrollo sostenido.

Se caracterizan por enfocarse en la producción sostenible preservando los recursos naturales para las generaciones futuras y la responsabilidad de mantener el equilibrio entre la protección de estos recursos y la satisfacción de las necesidades básicas de la población. Asumen la responsabilidad de resolver los problemas de las comunidades y de las regiones o territorios a las que pertenecen.

El perfil profesional

La carrera forma profesionales con capacidad para desarrollar sistemas de ingeniería electrónica y paralelamente aplicar la tecnología existente, comprometida con el medio, lo que le permite ser promotora del



cambio, con capacidad de innovación, al servicio de un conocimiento productivo, generando empleos y posibilitando el desarrollo social. Cuenta con las competencias para integrar la información proveniente de distintos campos disciplinarios concurrentes a un proyecto común. Posee la capacidad para abordar proyectos de investigación y desarrollo, integrando a tal efecto equipos interdisciplinarios, en cooperación, o asumiendo el liderazgo efectivo en la coordinación técnica y metodológica de los mismos. Tiene la preparación para generar nuevas tecnologías y/o producir innovación sobre tecnologías existentes, para resolver problemas inéditos en la industria, la sociedad y/o soluciones a problemas de ingeniería tomando en consideración aspectos científicos, técnicos, sociales y éticos, así como de responsabilidad profesional.

Por último, la preparación integral en materias técnicas, humanísticas, seguridad, higiene y medio ambiente le ubican en una posición relevante en la sociedad que demandará cada vez más del ingeniero y la ingeniera un compromiso ético y responsabilidad social en su quehacer profesional.

Al analizar los perfiles encontramos un profesional con conocimientos sólidos en ciencias básicas, capaz de resolver problemas a través del uso de la tecnología, administrando para ello diferentes recursos siempre manteniendo una mirada social y reflexiva sobre la incidencia de la actividad profesional en la sociedad en la que vivimos.

Para lograr profesionales con estas habilidades, debemos retomar el concepto de competencias de egreso establecido por el CONFEDI y desde la comunidad educativa bregar por la formación de estudiantes donde fomentemos:

- La pasión por lo que emprenden
- La capacidad del auto-aprendizaje
- La capacidad de comunicar a pares y a terceros
- La capacidad de reflexionar y tener una mirada crítica sobre lo que realizan.

La articulación del área es indispensable en función de lo comentado previamente.

Las asignaturas del área no deben ser compartimentos estancos sino favorecer la

continuidad de contenidos de forma de brindar a los estudiantes un contexto que favorezca su aprendizaje.

REFERENCIAS

- [1] Confedi. Competencias genéricas de egreso del Ingeniero Argentino.
https://confedi.org.ar/download/documentos_confedi/Cuaderno-de-Competencias-del-CONFEDI.pdf
- [2] Ausubel, D. P. (1963). The psychology of meaningful verbal learning. New York: Grune & Stratton
- [3] Ausubel, D.P. (2000). The acquisition and retention of knowledge: a cognitive view. Dordrecht: Kluwer Academic Publishers.
- [4] González, N (2021). Podcast Aula invertida.
<https://soundcloud.com/nahuel-gonzalez-973018730/podcast-aula-invertida>.
- [5] Paez, O (2013). Influencia de las materias integradoras en la formación del estudiante de Ingeniería de la UTN.
<http://www.frbp.utn.edu.ar/frbb/images/secretarias/planeamiento/influenciadelamateria.pdf>
- [6] Paez, O. (2012). Hacia la construcción de una didáctica de la profesión ingeniería y el rol de las materias integradoras en la UTN. Cataldi, Z. y Lage, F. (Comp.) Libro de Actas II JEIN. San Nicolás, UTN FRSN.

Avance de adaptador ISA a USB para adaptación tecnológica

Sebastian M. Heredia, Roberto M. Murdocca, Sergio F. Hernandez Velazquez

Departamento de Electrónica
Universidad Nacional de San Luis
San Luis, Argentina

e-mail: {sebastianmheredia, mmurdocc, sergio.sfhv}@gmail.com

Resumen - Este trabajo presenta los avances y pruebas iniciales de una interfaz de bus ISA a USB, con el fin de actualizar una placa de adquisición de datos basada en tecnología de bus ISA. El objetivo es reutilizar estas placas como herramientas didácticas en los laboratorios de la carrera de Ingeniería Electrónica en la Universidad Nacional de San Luis. Se ha logrado establecer comunicación con la placa, y el próximo paso será desarrollar una librería con las funciones necesarias para que permita a los usuarios en Windows acceder a la configuración de la placa de manera transparente.

Palabras clave: Bus ISA, I2C, Adquisición de datos, ESP32, adaptador.

I. INTRODUCCIÓN

El presente proyecto surge de la necesidad de poder reutilizar placas de adquisición de datos que a la fecha son obsoletas debido a que su interfaz física se realiza a través de bus ISA. Es importante recordar que este bus se utilizaba en las PC de hace un par de décadas por lo menos y en las motherboards actuales solo se dispone de bus PCI, PCI-Express y puertos USB como medio de incorporación de placas o módulos externos a la PC.

En la materia Interfaces de la carrera Ing. Electrónica con OSD, se trabaja con sensores y se realiza adquisición de datos para lo cual actualmente se cuenta con solo una placa de adquisición de tipo USB y tres obsoletas con interfaz ISA. Esto motivó a analizar las características técnicas de las placas y se observó que, salvo por la interfaz, el resto de las especificaciones son a la actualidad por demás suficientes en lo referente a adquisición de señales en el ámbito académico donde no

se requiere alta velocidad. Por estos motivos en un principio se buscó una opción de adaptación comercial de ISA a USB y se encontraron algunos adaptadores afines, pero el precio de los mismos además de las restricciones de importación obligó a descartar esta opción. A modo de ejemplo en la Fig. 1 se muestra uno de estos adaptadores el cual posee un costo aproximado de US\$150 más gastos de envío. En la Fig.2 podemos ver el bus ISA y sus señales correspondientes [1]. Es por ello que se investigaron otras opciones que permitieron llevar a cabo la interfaz adaptadora para la reutilización de dicho hardware. Las opciones inicialmente analizadas fueron: por una lado realizar toda una interfaz lógica con transceiver, registros lath, registros de desplazamiento, es decir tecnología pasiva. Por otro lado, utilizar dispositivos reconfigurables, es así que surge como primera opción una PPI (Interfaz Periférica Programable) 8255 [2] y [3]. Esta interfaz era utilizada en arquitecturas x86 para ampliación de líneas de E/S. En la Fig. 3 se visualiza esta interfaz. Otra opción que surgió es utilizar pines GPIO de un microcontrolador en forma directa, lo cual requería utilizar un microcontrolador con gran cantidad de líneas GPIO. Por último, se incorporó al análisis el uso de un chip adaptador de I2C a GPIO, MCP23017. Cada una de estas opciones presentó sus fortalezas y debilidades que fueron analizadas.



Fig 1. Placa adaptadora de bus ISA a USB comercial.

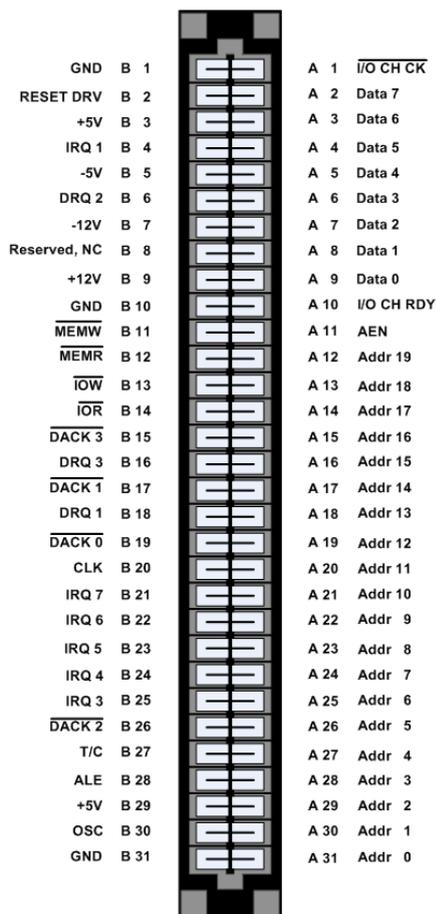


Fig 2. Bus ISA y sus señales asociadas.

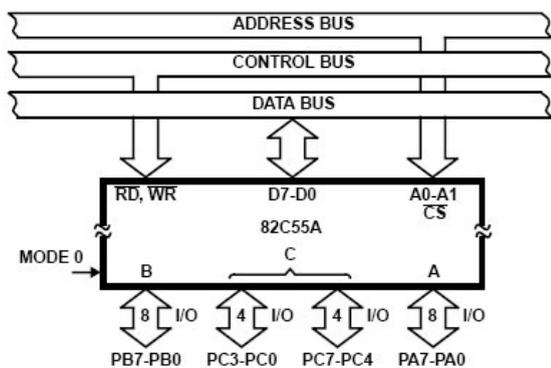


Fig 3. Interfaz PPI8255

II. ESPECIFICACIONES Y REQUERIMIENTOS

En base a las características del bus ISA, surgen entonces las especificaciones del adaptador a desarrollar. El mismo debía poseer un conector ISA de 8 bits para poder realizar la conexión a la placa de adquisición. A su vez, fue necesario un análisis exhaustivo de las interfaces de conexión y de la especificación del bus ISA a nivel de hardware como de software para poder establecer la comunicación con la placa. Los niveles lógicos deben ser de 5V ya que este bus paralelo emplea lógica TTL.

Se debía proveer además alimentación a la placa de adquisición, estas tensiones son 5, 12 y -12V.

III. DISEÑO Y METODOLOGÍA

De las alternativas mencionadas anteriormente para la adaptación y luego de realizar un análisis en cuanto a complejidad de la interfaz, forma de programación, mercado actual de componentes, entre otras, se decidió utilizar el dispositivo MCP23017 [4] de Microchip, mostrado en la Fig. 4. Este componente consiste en un expansor de E/S paralelo de propósito general.

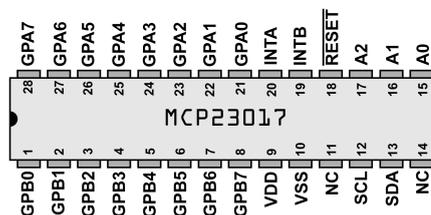


Fig 4. Adaptador GPIO a I2C MCP23017.

En base a esta selección se implementa la interfaz. En la Fig. 5 se observa el diagrama en bloques del sistema adaptador, donde se detallan los distintos componentes utilizados.

Como gestor de este adaptador se escogió una placa ESP32 [5] por su bajo costo, potencialidad, interfaz USB y conectividad con la que cuenta para futuras ampliaciones. La ESP32 se encarga de controlar al MCP23017 [6] a través de I2C para configurar la dirección de los pines GPIO y así poder generar los comandos necesarios para emular las señales ISA, teniendo presente que este último está constituido por señales de datos, direcciones y control. Por lo cual debe direccionar la placa, generar las señales de lectura y escritura en periférico (/IOR - IORW) y la señal de habilitación de dirección (AEN). En la Fig. 6 se muestran las señales necesarias para la implementación de la interfaz de la placa de adquisición de datos modelo ADQ12-B [7].

Para realizar las pruebas se utiliza una PC con un terminal serie.

La programación del firmware de control y adquisición se llevó a cabo a través del soporte en lenguaje C/C++ para ESP32 basado en el framework de Arduino, utilizando el software IDE Visual Studio Code.

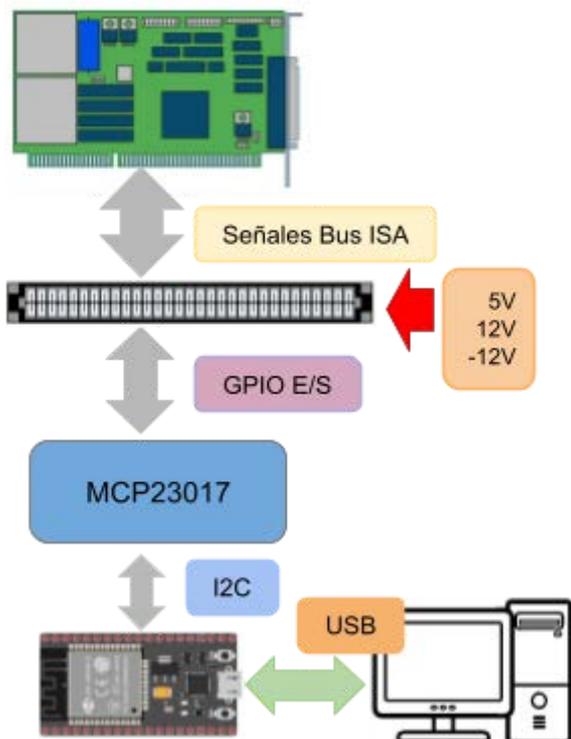


Fig 5. Diagrama en bloques del adaptador.

III. IMPLEMENTACION

A. HARDWARE

Para el prototipo se utilizó un PCB universal de 9x15 para montar el zócalo ISA, el cual fue reciclado de una vieja motherboard en desuso, el MCP23017, conexiones, bornes de alimentación y conexión al ESP32. En la Fig. 7 podemos ver como fueron ubicados los componentes en dicha placa. Este es el primer prototipo para realizar las pruebas de comunicación con la placa de adquisición ADQ12-B.

La ESP32 se montó sobre un protoboard y se realizaron las conexiones a través del bus I2C. La adaptación de las señales de 5V, correspondientes a las líneas SDA y SCL (I2C), a 3.3V del microcontrolador se implementó usando transistores MOS 2N7000 [8]. Para las pruebas de funcionamiento se utiliza una fuente de PC ATX, la cual cuenta con todas las tensiones necesarias mencionadas anteriormente.

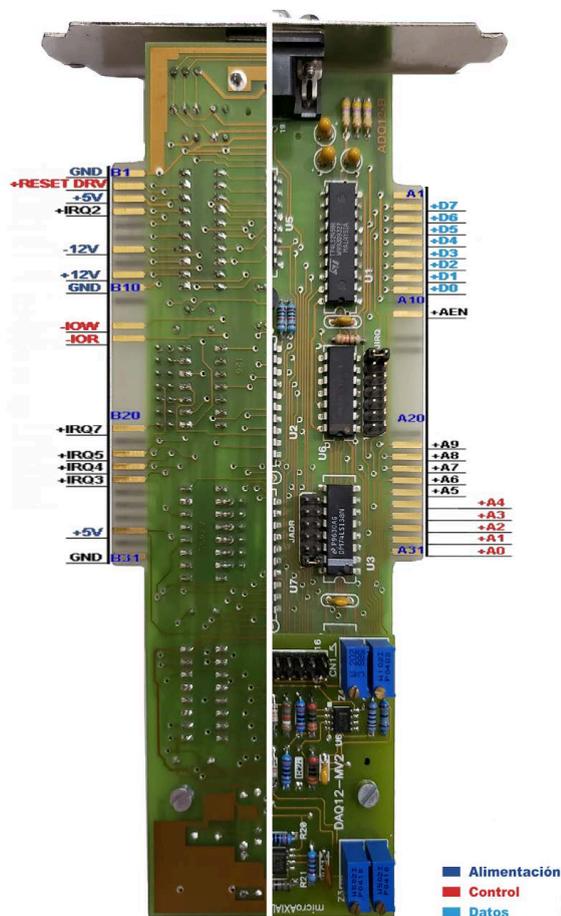


Fig 6. Señales usadas de la placa.

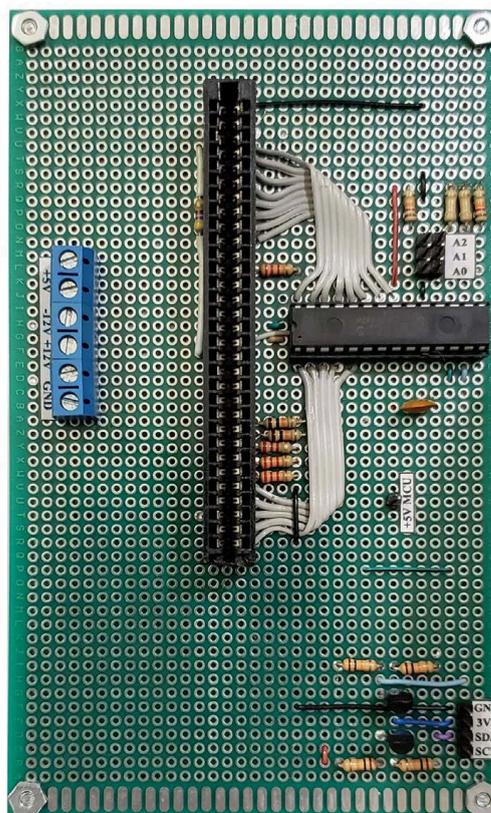


Fig 7. Prototipo del adaptador.

B. FIRMWARE DE LA ESP32

El programa de control desarrollado en lenguaje C++ para ESP32 está basado en la emulación de un puerto ISA para replicar el comportamiento de conexión y comunicación con el hardware de adquisición de datos, permitiendo que los sistemas interactúen como si estuvieran utilizando el protocolo original.

Para ello, se recopiló la documentación técnica y las especificaciones sobre el protocolo de bus ISA de 8 bits y se identificaron las señales específicas que utiliza la ADQ-12B. Los usuarios que empleaban hardware ISA tenían que disponer de información especial sobre los dispositivos que iban a añadir al sistema, ya que generalmente la configuración era manual.

Esto posibilita examinar el funcionamiento interno de la comunicación analizando códigos de programas en lenguaje QBASIC, PASCAL, C, ASSEMBLY y de funciones de librería para el usuario, además de registros propios de la placa de adquisición de datos. Por lo tanto, se reprodujeron las operaciones de escritura y lectura, y se llevaron a cabo pruebas preliminares a nivel de bits.

Con los procesos de bajo nivel establecidos, se procedió a implementar las funciones de control para la adquisición de datos, las cuales pueden ser utilizadas a través de una comunicación UART con una computadora para visualizar los datos en un software de terminal serie. Las pruebas realizadas permitieron establecer la comunicación con los siguientes módulos de la placa de adquisición: GPIO, conversor AD, selección de canal de conversión AD, cambio de ganancia, escritura en los DAC y acceso al timer 8253 [9].

VI. Resultados

A través de la realización de este prototipo se logró una interfaz operativa para establecer la comunicación con la placa de adquisición de datos.

La implementación de esta etapa posibilitó el control del dispositivo de adquisición de datos mediante el envío de comandos que permiten la programación bit a bit de sus puertos digitales de entrada y salida, leer y escribir en registros de configuración asociados a la selección de canales, fondo de escala, parámetros de conversión, entre otros, y en registros de E/S para la conversión A/D y D/A, y uso general de su temporizador interno. En la

Fig. 8 podemos ver el prototipo bajo pruebas de lectura y escritura. En la Fig.9 se muestra el envío de datos a la ESP32 desde la placa de adquisición de datos.

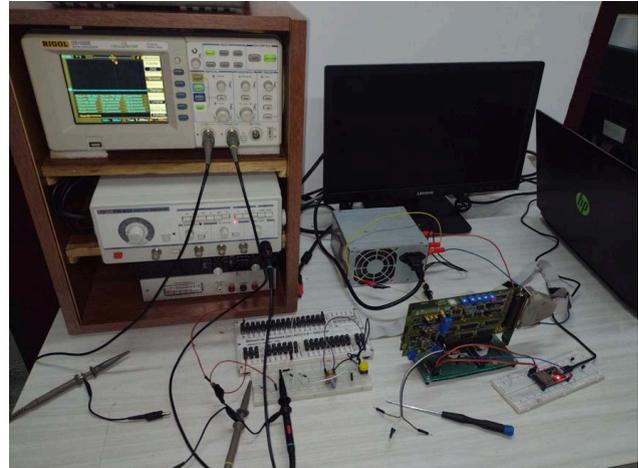


Fig 8. Prototipo bajo pruebas.



Fig 9. Captura de prueba de ADC.

VI. Conclusiones

Se lograron los resultados esperados en esta primera etapa del trabajo. Se pudo extender la vida útil de las placas y presentar una herramienta útil para diversas aplicaciones. Se seguirá avanzando para desarrollar un software de computadora para el uso en forma directa a través de funciones básicas relacionadas con la placa, con el objeto de que el usuario solo las utilice independientemente de la forma en que esto se realice, y puedan ser empleadas para la recopilación, procesamiento y análisis de datos en programas afines, como por ejemplo, MATLAB, SciLab o Python. Además, una vez terminadas las pruebas y realizada la librería se diseñará un PCB para montar todo el adaptador y placa ESP32 en un único circuito dentro de un gabinete. A su vez, se estudiará la posibilidad de incorporar un modo de trabajo inalámbrico ya sea por Bluetooth y/o WiFi para el acceso a



la placa de adquisición de datos, y se diseñará una fuente dedicada para reducir posibles problemas de ruido de la fuente ATX y facilitar su manipulación y mantenimiento.

VIII. REFERENCIAS

- [1] Intel. (1989). ISA Bus Specification and Application Notes. Intel.
https://ia904701.us.archive.org/32/items/bitsavers_intelbusS pep89_3342148/Intel_ISA_Spec2.01_Sep89_text.pdf
- [2] CPCWiki. (2024). 8255. CPCWiki.
<https://www.cpcwiki.eu/index.php/8255>.
- [3] Intel. (2024). Intel 8255A datasheet. CPCWiki.
https://www.cpcwiki.eu/imgs/b/bd/Intel8255A_datasheet.pdf
- [4] Microchip Technology Inc. (2018). MCP23017 16-Bit I/O Expander with Serial Interface Datasheet.
<https://ww1.microchip.com/downloads/aemDocuments/documents/APID/ProductDocuments/DataSheets/MCP23017-Datasheet-DS20001952.pdf>.
- [5] Espressif Systems. (2024). ESP32 datasheet. Obtenido de https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
- [6] Lemasle, B. (n.d.). Support of MCP23017. GitHub.
<https://github.com/blemasle/arduino-mcp23017>.
- [7] Microaxial. (s.f.). Manual ADQ12.
<http://www.microaxial.com.ar/historicos/Adquisidores%20ISA/ManualADQ12.pdf>.
- [8] Microchip (2024). N-Channel Enhancement-Mode Vertical DMOS FET
<https://ww1.microchip.com/downloads/aemDocuments/documents/APID/ProductDocuments/DataSheets/2N7000-N-Channel-Enhancement-Mode-Vertical-DMOS-FET-Data-Sheet-20005695A.pdf>.
- [9] Intel. (n.d.). Intel 8253 Programmable Interval Timer Datasheet. <https://www.cpcwiki.eu/imgs/e/e3/8253.pdf>

Implementación de un core RISC-V RV32I

1st Roberto G. Simone

Lab. de Microelectrónica (Departamento de Electrónica)
Facultad Regional Buenos Aires (UTN)
Buenos Aires, Argentina
rsimone@frba.utn.edu.ar

2st Alejandro J. Furfaro

Lab. de Microelectrónica (Departamento de Electrónica)
Facultad Regional Buenos Aires (UTN)
Buenos Aires, Argentina
afurfaro@frba.utn.edu.ar

3st Luciano P. Ferreyro

Lab. de Microelectrónica (Departamento de Electrónica)
Facultad Regional Buenos Aires (UTN)
Buenos Aires, Argentina
lferreyro@frba.utn.edu.ar

Abstract—Presentamos la implementación en una fpga de un core RISC-V RV32I sincrónico para usos en investigación y académicos. El requerimiento del uso académico nos llevo a desarrollar versiones de cpu de Ejecución en un Único Ciclo de Reloj y con Pipeline, así como realizar el desarrollo en forma incremental donde primero se implementaron las instrucciones aritméticas y lógicas, luego las de salto y finalmente las de acceso a memoria de datos. De esta manera mostramos el impacto de cada una en la microarquitectura. Con la incorporación del Pipeline pudimos mostrar la aparición de Hazards, como detectarlos y evitarlos mediante el hardware adecuado. También se desarrollaron útiles de software para establecer una interface firmware-hardware. También hacemos una breve mención de los útiles desarrollados para validar la corrección del diseño del procesador.

Index Terms—RISC-V, RV32I, Pipeline, Hazards

I. INTRODUCCIÓN

RISC-V es un Repertorio de Instrucciones (ISA) abierto para desarrollar procesadores personalizados y Extensiones para usos específicos [1]. Si bien las especificaciones no exigen una implementación con pipeline, es la mayormente adoptada. Esto lo volvió atractivo para nuestro grupo de investigación sobre circuitos asincrónicos ya que empleamos la metodología de Relative Timing [2] que es particularmente eficiente en pipelines.

Antes de abordar una implementación asincrónica era conveniente disponer de su contraparte sincrónica lo que permitiría en un futuro comparar métricas de consumo de energía. Escogimos la arquitectura de enteros RV32I que es la más sencilla. De esta forma nos pareció natural implementar este core en una fpga, concretamente en uno de la firma Intel-Altera dado que disponemos de kits de entrenamiento [3] y experiencia en el uso de sus herramientas.

II. LA ARQUITECTURA RV32I

La especificación da cuenta de 47 instrucciones a saber

- 1) Carga de inmediatos
- 2) Saltos y Llamados a subrutinas
- 3) Acceso a memorias de datos
- 4) Aritméticas y Lógicas
- 5) Sincronización
- 6) Entorno
- 7) Acceso a los registros de estado y control

El uso de las instrucciones de Sincronización (fence y fence.i) se usan mayormente en sistemas multiprocesador. Las instrucciones de Entorno (break y ecall) se necesitan para dotar al procesador de capacidades de debugging, y de modalidades kernel (Machine) y usuario. Los registros de estado y control dan soporte a estas funciones.

Puesto que nos impusimos mantener sencilla nuestra implementación dejamos de lado las instrucciones mencionadas, que corresponden a las categorías 5, 6 y 7 del listado anterior.

Además de estas instrucciones, el core RV32I debe contar con 32 registros de datos identificados con los nombre xn (ej: registro x1). En particular el registro x0 está hardcodedo con la constante 0 lo cual resulta útil en determinados escenarios. El contador de programa pc también es de 32 bits.

Las instrucciones están codificadas de manera que cada una ocupe un sola posición de memoria de programa de 32 bits. Existen 6 tipos de codificaciones que establecen el formato de los campos de bits dentro de la palabra de instrucción [4]

- 1) R - Operaciones entre registros
- 2) I - Operaciones registro e inmediato
- 3) U - Carga de inmediatos
- 4) J - Llamado a subrutinas
- 5) B - Comparacion y salto
- 6) S - Acceso a memoria de datos

III. MATERIALES Y MÉTODOS

Hicimos un diseño en forma incremental dividido en 6 etapas en cada una de las cuales desarrollamos una implementación del procesador RISC-V. La mitad de ellas ejecutan instrucciones en un ciclo de reloj (acrónimo SC de Single Cycle Execution) y la otra mitad utilizan un pipeline (acrónimo PL). En cuanto a las funcionalidades, en la primera fase de trabajo implementamos las instrucciones tipo I y U, la arquitectura más sencilla posible. En la segunda fase incorporamos las tipo J y B (acrónimo J) y en la última fase las tipo S (acrónimo JD por memoria de Datos).

- 1) riscv_SC
- 2) riscv_PL
- 3) riscv_SC_J
- 4) riscv_PL_J
- 5) riscv_SC_JD
- 6) riscv_PL_JD

Las implementaciones con pipeline tienen a su vez 2 sub-versiones, una con hazards y otra libre de hazards. De esta forma pudimos documentar adecuadamente las modificaciones microarquitectónicas necesarias para suprimir los hazards.

El punto de partida fue la implementación **riscv_SC** que nos permitió establecer la estructura de trabajo que se replicaría en las siguientes. Además de su descripción Verilog HDL se lo dotó de un testbench genérico para ensayarlo con una batería de programas en lenguaje ensamblador. La generación de estos la automatizamos con unos scripts a fin de cubrir el mayor número de escenarios con cada una de las instrucciones implementadas.

La Fig. 1 muestra un diagrama de bloques de la implementación. Las divisorias en líneas de puntos anticipan la planificación del pipeline.

La pieza clave de riscv_SC es el *Decodificador de Instrucciones* ya que genera todas las señales de control de la alu y del banco de registros. En la Fig. 2 vemos las microoperaciones que genera, de los que destacamos a **rs1**, **rs2** y **rd** que seleccionan los operandos origen y destino y, **funct3** y **funct7** que determinan la operación que realizará la alu.

IV. PIPELINE Y HAZARDS

El procesador riscv_SC reconvertido a una versión con pipeline (riscv_PL) requiere de la inserción de registros interetapa. Adoptamos un enfoque convencional de 4 etapas [5]

- 1) Fetch
- 2) Decodificación
- 3) Ejecución
- 4) Write-Back

La etapa de **Fetch** agrupa al *contador de programa* y la *memoria de programa*, la de **Decode** al *decodificador de instrucciones* y el *File Register* y la de **Execute** a la Alu. La etapa de **Write-Back** devuelve los resultados de

la Alu al File Register para su almacenamiento. La Fig. 3 da cuenta de ello.

Podemos comprender como aparecen Hazards en este pipeline analizando la ejecución de las siguientes instrucciones

```
addi x1, x1, 0x00001
addi x1, x1, 0x00001
```

cuyo resultado es sumarle 2 al contenido del registro x1. La Fig. 4 describe la secuencia de ejecución asumiendo que el contenido del registro x1 previo a la ejecutar las instrucciones era 0. Vemos que los operandos son suministrados por el banco de registros, procesados por la alu y el resultado regresa al banco de registros. Ello demanda 2 ciclos de reloj ya que deben atravesarse 2 registros interetapas. De esta manera los registros modificados por una instrucción se actualizarán con una demora de 2 ciclos de reloj.

En la secuencia de instrucciones propuesta el registro destino (x1) de la primera es usado como operando (x1) en la siguiente. Eso desencadena el Hazard ya que el registro x1 aún no fue actualizado en el banco de registro y sin embargo se lo emplea para realizar otra operación.

La eliminación de los hazards puede hacerse por software, introduciendo instrucciones **nop** que den tiempo al pipeline a actualizar los registros antes de volver a operar con ellos o bien detectando por hardware el uso en instrucciones consecutivas de un mismo registro como operando origen y destino. Implementamos 2 arquitecturas que responden a uno y otro esquema.

Para el primer caso debimos generar una batería de firmware de prueba que incluyeran las instrucciones nop que aseguren que no se produzcan hazards. La arquitectura libre de hazards requirió de hardware para detectar esta condición como muestra el siguiente código Verilog

```
assign rs1_zero = (i_ex_rs1 == 0);
assign rs2_zero = (i_ex_rs2 == 0);

assign o_lookahead_op1 = (i_ex_rs1 ==
    i_wb_rd) & ~rs1_zero;
assign o_lookahead_op2 = (i_ex_rs2 ==
    i_wb_rd) & ~rs2_zero;
```

La activación de la señal **o_lookahead_op1** ó de **o_lookahead_op2** indican un hazard en el operando 1 ó 2 respectivamente. En tal caso un multiplexor se encarga de suministrar a la alu el valor correcto del operando como ilustra el código de la Fig. 5.

V. SALTOS Y LLAMADOS A SUBROUTINAS

La ISA de RV32I especifica 6 instrucciones de salto condicional

- beq
- bne

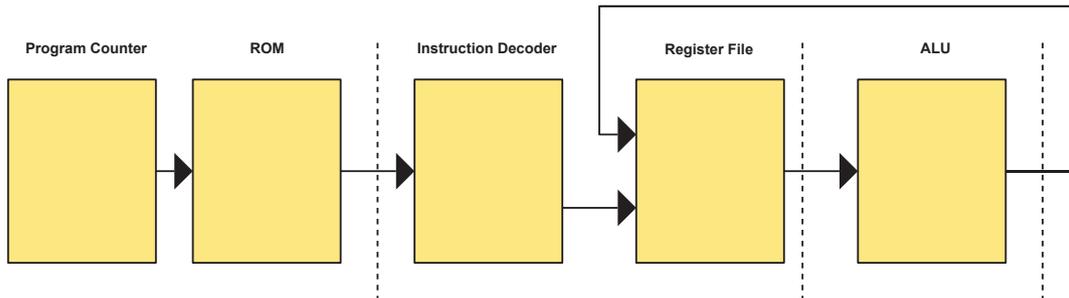


Fig. 1. Diagrama de riscv_SC

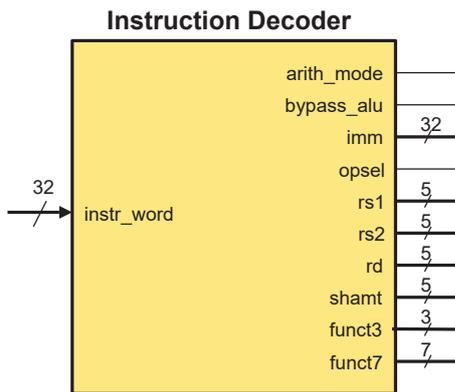


Fig. 2. Decodificador de instrucciones

- blt
- bltu
- bge
- bgeu

Para lo cual fue necesario introducir un módulo de hardware que realice comparaciones y la carga paralelo de una palabra binaria en el contador de programa. La Fig. 6 muestra el diagrama de bloques de la arquitectura que ejecuta en un ciclo de reloj (**riscv_SC_J**).

Estas instrucciones implicaron incorporar nuevas microinstrucciones al Decodificador de Instrucciones, la primera de ellas **reg_file_wr**. Dado que las instrucciones de salto condicional no alteran el contenido de los registros de datos, su escritura debe deshabilitarse cuando se está ejecutando este tipo de saltos.

Otro cambio importante ocurrió en el contador de programa donde permitimos una carga paralelo de un *valor inmediato* que permite realizar los saltos de programa. La dirección de salto se obtiene de la salida de la alu como vemos en el bus de color verde de la Fig. 6. La realización del salto es ordenada por la microinstrucción de carga paralelo del contador de programa, dibujada como la señal roja. Este puede provenir de 2 fuentes

- saltos incondicionales
- saltos condicionales

Para el primer caso el Decodificador de Instrucciones

proporciona una microinstrucción específica (**jp**) mientras que en el segundo se emplea directamente el resultado de la comparación (**cp**). Un multiplexor selecciona una u otra dependiendo del tipo de salto, lo cual está gobernado por la señal **branch**, de color azul en el esquema, también provista por el Decodificador de Instrucciones, y tal como expresa el código Verilog

```
assign pc_ld = (branch) ? cp :
                jp;
```

No debemos olvidar que también existe un tipo de instrucciones para el llamado a subrutinas

- jal
- jalr

que saltan y guardan la dirección de retorno en un registro de datos. Estas instrucciones no hacen más que activar la señal de salto incondicional (**jp**) a la vez que la escritura en el banco de registros (**reg_file_wr** y **rd**). Recordemos que RISC-V no maneja por hardware a *la pila*, útil para subrutinas anidadas, o *un frame stack de lenguaje C*. Debemos hacerlo por programa usando varias instrucciones.

El procesador **riscv_PL_J** nace de introducir el pipeline a la microarquitectura recién analizada resultando el esquema de la Fig. 7. No hay grandes cambios globales respecto del pipeline anteriormente visto. De hecho se pueden producir los mismos hazards en el camino de datos de no mediar los circuitos de lookahead. Sin embargo aparece una nueva fuente de hazards producto de la instrucciones de salto.

Es fácil apreciar el origen de este nuevo hazard con el siguiente extracto de código ensamblador

```
beq    x1, x2, skip
or     x3, x1, x2
and    x4, x1, x2
```

```
skip:
addi   x1, x1, 0x00001
```

El salto condicional **beq** se producirá si la condición de igualdad es verdadera y en tal caso las instrucciones **or** y **and** no se ejecutarán. La situación experimentada en el pipeline es bien diferente ya que las instrucciones son

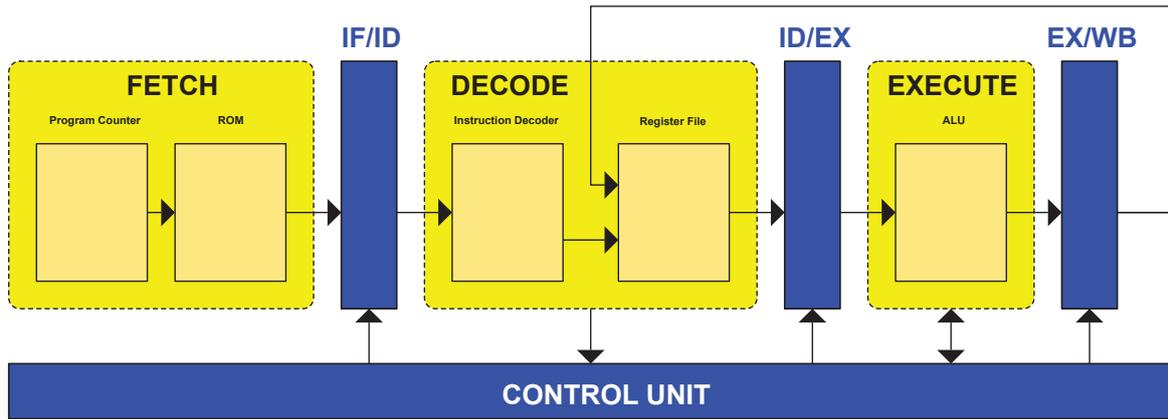


Fig. 3. Diagrama de RISC_V_PL

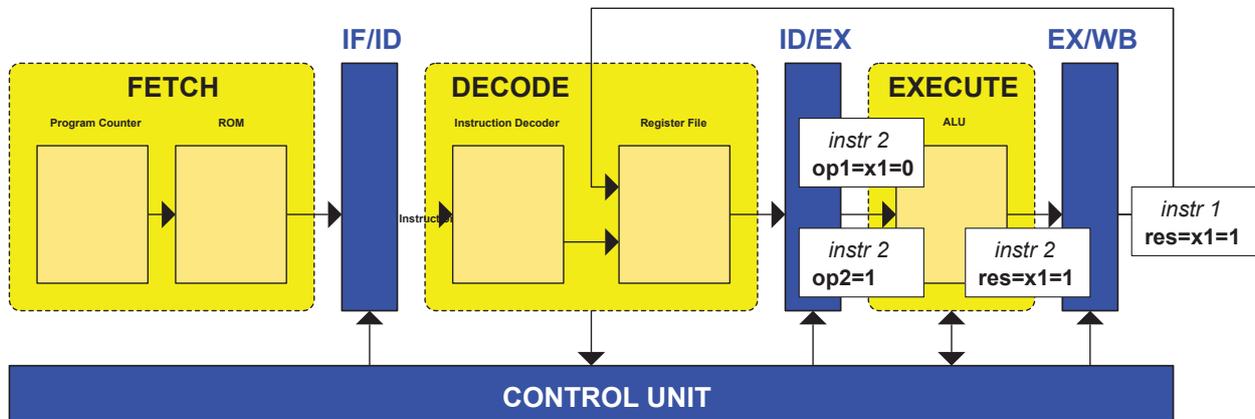


Fig. 4. Hazard en el riscv_PL

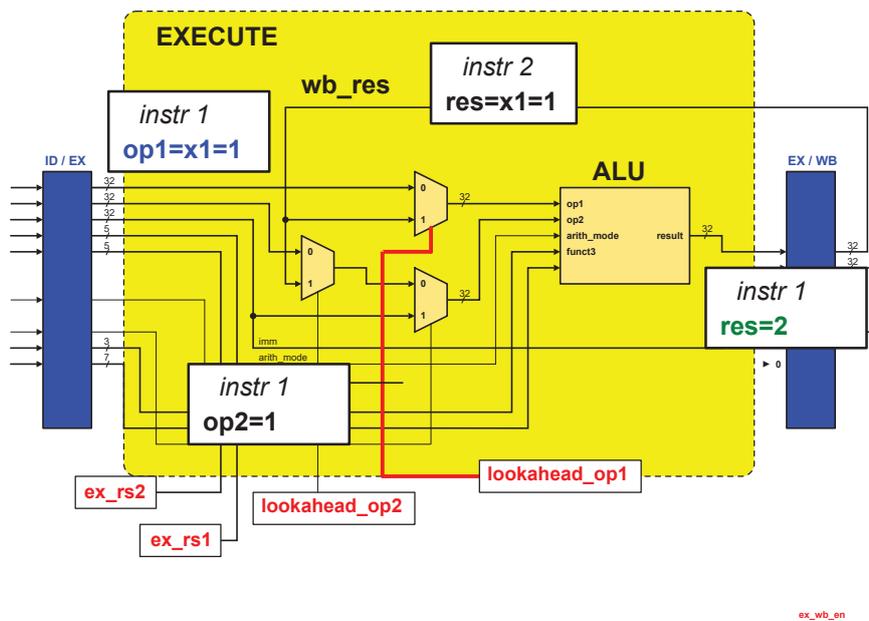


Fig. 5. Circuito de Lookahead o Forwarding

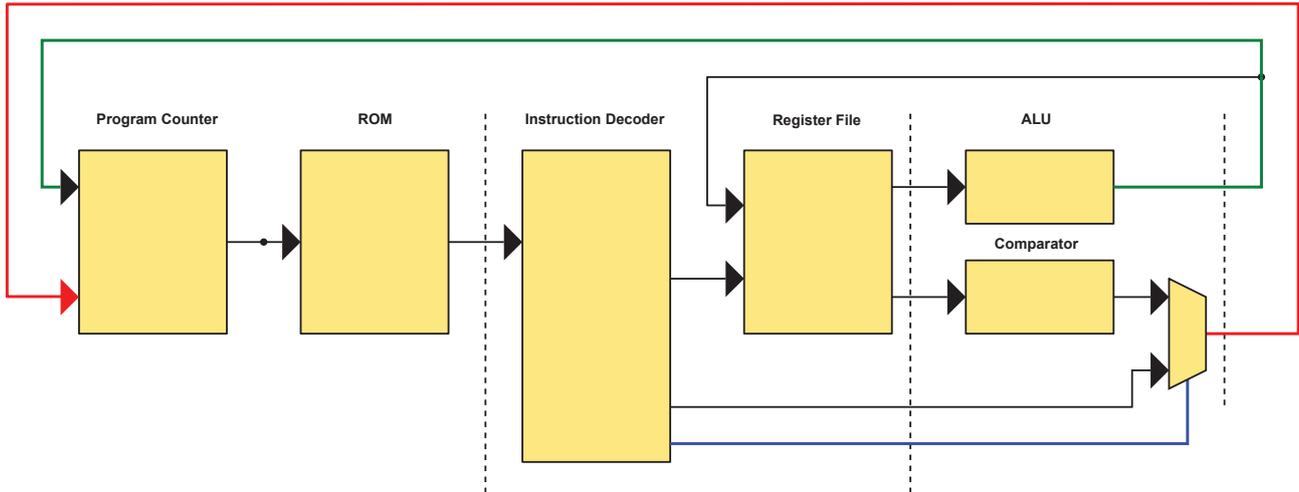


Fig. 6. Diagrama de riscv_SC_J

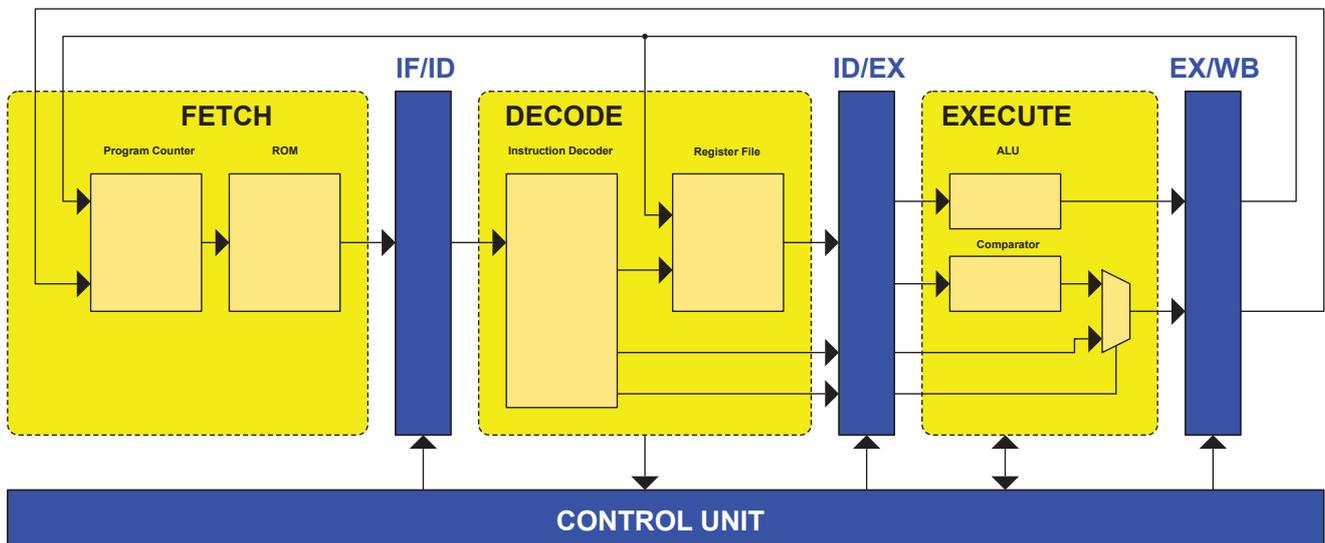


Fig. 7. Diagrama de riscv_PL_J

leídas en la etapa de Fetch y la comparación recién se efectúa en la de Execute. La etapa de Fetch no tiene forma de saber si el salto debe realizarse. Tampoco tiene medios para calcular la dirección de salto con la microarquitectura planteada hasta el momento.

Como ilustra la Fig. 8 el pipeline procede a una ejecución lineal del programa tal como si el salto nunca tuviera lugar. Como consecuencia se encolarán indebidamente 2 instrucciones en las etapas previas a Execute (Fetch y Decode).

El remedio para este hazard es anular las microoperaciones que intenten realizar estas 2 instrucciones incorrectamente ingresadas al pipeline, en concreto *impedir modificaciones en los registros de datos o en el contador de programa*. Esta acción se conoce como **stall** y se efectúa toda vez que se realice un salto [6]. La unidad de

control recibe la microoperación de salto y envía órdenes de stall a los registros interetapa (IF/ID y ID/EX). La Fig. 9 da cuenta de esta situación para el ejemplo que desarrollamos.

Luego del stall el funcionamiento del pipeline se recompone naturalmente ya que el Contador de Programa se carga con la dirección de salto y las próximas instrucciones son leídas de la memoria de programa a partir de esta nueva dirección.

VI. ACCESO A MEMORIA DE DATOS

La incorporación de un RAM síncrona [7] como Memoria de datos al procesador RISC-V introdujo algunas peculiaridades al diseño, siendo la de mayor implicancia el registro interno que posee para latchedar la dirección de memoria. Podemos cargar con la penalidad

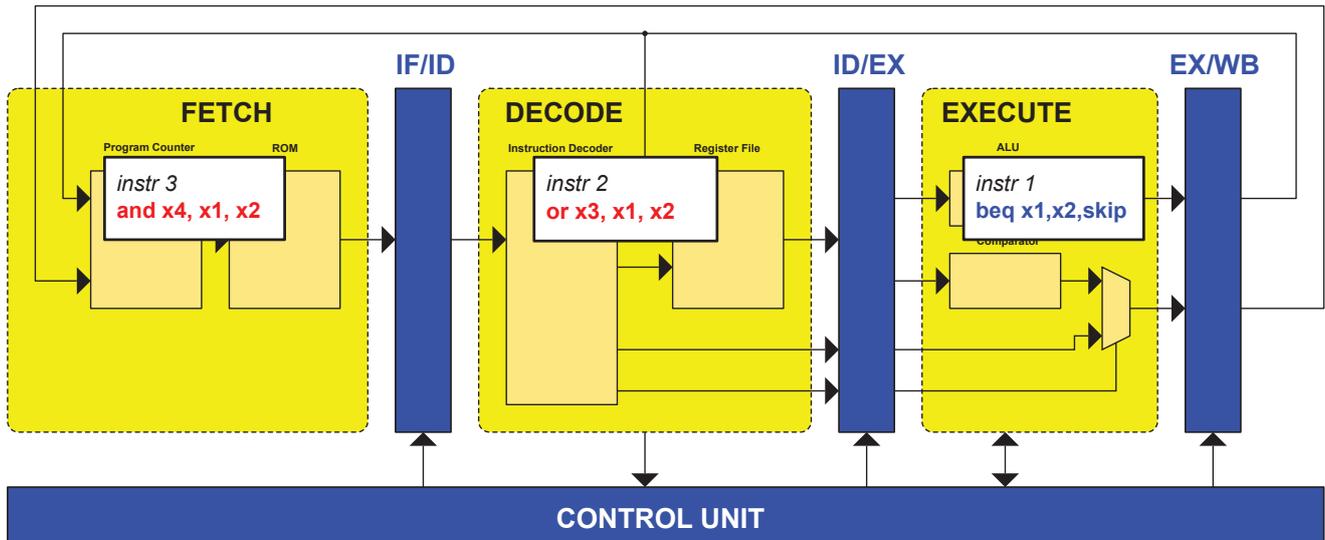


Fig. 8. Hazard en una instrucción de salto

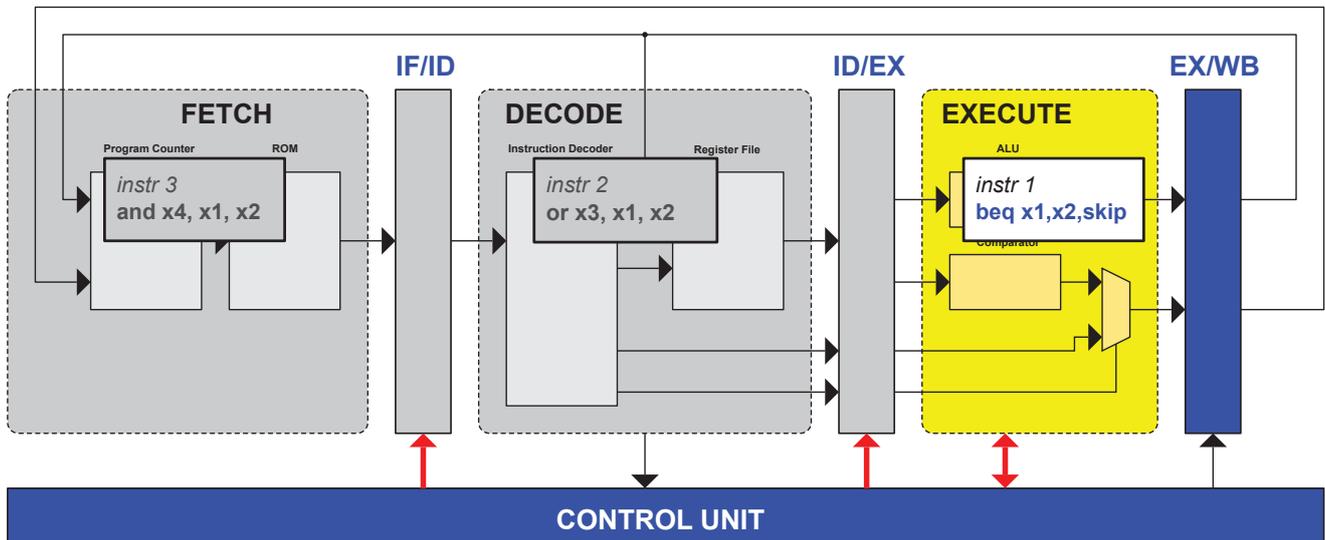


Fig. 9. Stall producido por un salto

de un ciclo de reloj adicional en la ejecución si no hacemos las previsiones arquitectónicas necesarias.

Para el caso de la versión de Ejecución en un Único Ciclo de Reloj esta penalidad resultó inevitable. En realidad pasó de ser un procesador de ejecución en un único ciclo a uno de ejecución en 2 fases (2 ciclos de reloj). Los ciclos de reloj impares (fase 2) están dedicados al acceso a la ram sincrónica , en tanto que los ciclos de reloj pares (fase 1) al resto de las funciones del procesador, que ya explicamos como ser fetch, lectura de operandos, aritmética, comparación, etc. La denominación **riscv_SC_JD** se mantuvo por convención y la Fig. 10 muestra su diagrama de bloques.

La especificación de la arquitectura RV32I contempla las siguientes instrucciones para acceder a memoria de

datos

- sb
- sh
- sw
- lb
- lh
- lw
- lbu
- lhu

Tanto para la escritura como para la lectura podemos operar a nivel de byte y half word (16 bits) además de los 32 bits nominales. Las *Unidades de Store* y *Load* se encargan de reducir o extender la longitud de palabra y el acceso a memoria de acuerdo a longitud de bits

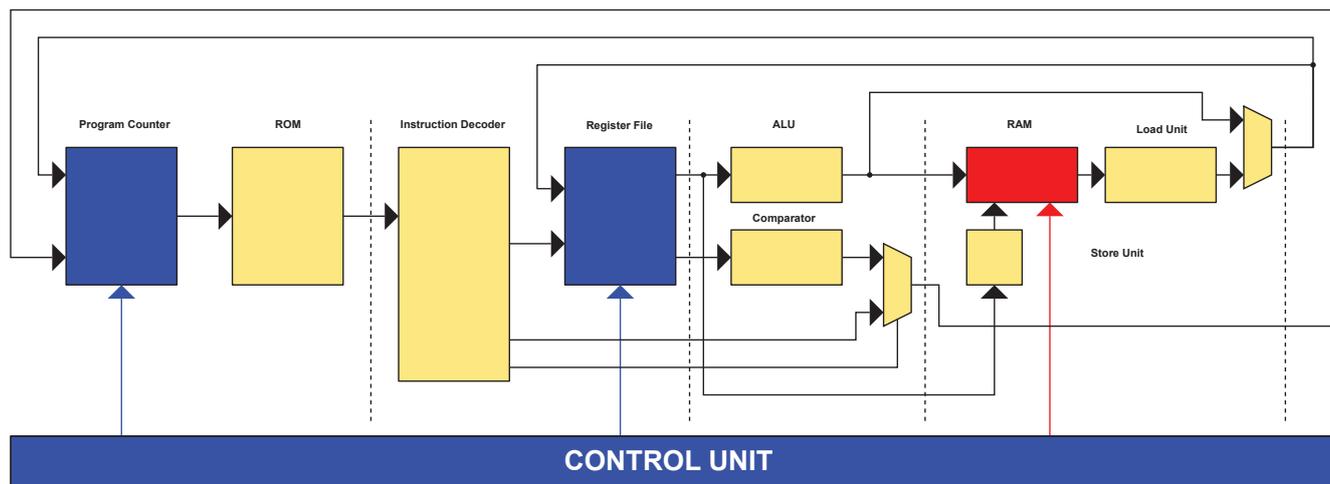


Fig. 10. Diagrama del riscv_SC_JD

requerida por la instrucción. Respecto a la posibilidad de utilizar accesos desalineados en memoria (la dirección no es múltiplo de 4) la especificación RV32I no se pronuncia categóricamente pero en nuestro diseño forzamos accesos alineados para simplificar el hardware.

La operación en 2 fases de esta arquitectura podemos verla reflejada en la Fig. 10, donde operan alternadamente los elementos sincrónicos del procesador. En la fase par, en color azul, actúan el Contador de Programa y el Banco de Registros. Por otra parte, en la fase impar en color rojo, la memoria RAM sincrónica.

Gracias a un bypass de la dirección de memoria por sobre el registro interetapa (**EX/ME**) no pagamos una penalidad de un ciclo de reloj adicional en la ejecución en la cual podría incurrirse debido al latch de la RAM sincrónica. La Fig. 11 muestra parte del detalle microarquitectónico.

Los hazards citados con anterioridad siguen potencialmente presentes en esta arquitectura y la forma de evitarlos o remediarlos sigue siendo la misma. Con la introducción de la memoria de datos aparece una nueva fuente de hazards dado que la lectura de un dato y su consecuente almacenamiento en un registro puede implicar la invalidación de un operando en el pipeline, de forma muy similar a lo ocurrido con las operaciones aritméticas y lógicas. Y la solución es idéntica , con un circuito de lookahead.

En este pipeline puede ocurrir la situación que se detecten simultáneamente hazards en la etapa de Execute y la de Memory. En tal caso debe emplearse como contenido más actual del registro destino el que está atravesando la etapa Memory y no el que se halla en la de Write-back. El siguiente código Verilog muestra como se resuelve este conflicto

```
assign rs1_zero = (i_ex_rs1 == 0);
assign rs2_zero = (i_ex_rs2 == 0);
```

```
assign lookahead_ex_me_op1 =
    (i_ex_rs1 == i_me_rd) & ~rs1_zero;
assign lookahead_ex_me_op2 =
    (i_ex_rs2 == i_me_rd) & ~rs2_zero;
assign lookahead_ex_wb_op1 =
    (i_ex_rs1 == i_wb_rd) & ~rs1_zero;
assign lookahead_ex_wb_op2 =
    (i_ex_rs2 == i_wb_rd) & ~rs2_zero;
```

```
assign o_lookahead_op1 =
    (lookahead_ex_me_op1) ?
    i_me_lookahead_reg :
    i_wb_lookahead_reg;
```

```
assign o_lookahead_op2 =
    (lookahead_ex_me_op2) ?
    i_me_lookahead_reg :
    i_wb_lookahead_reg;
```

```
assign o_lookahead_op1_sel =
    lookahead_ex_me_op1 |
    lookahead_ex_wb_op1;
```

```
assign o_lookahead_op2_sel =
    lookahead_ex_me_op2 |
    lookahead_ex_wb_op2;
```

VII. DISCUSIÓN Y RESULTADOS

Todo el desarrollo se realizó a nivel simulación empleando el software *ModelSim* [8] de Mentor Graphics (Siemens) en su versión gratuita que acompaña al software *Quartus* [9] de Intel-Altera. A través de un toolchain de verificación que desarrollamos para este trabajo, y que escapa al alcance de este documento, aseguramos que el diseño se ajustara a la especificación RV32I.

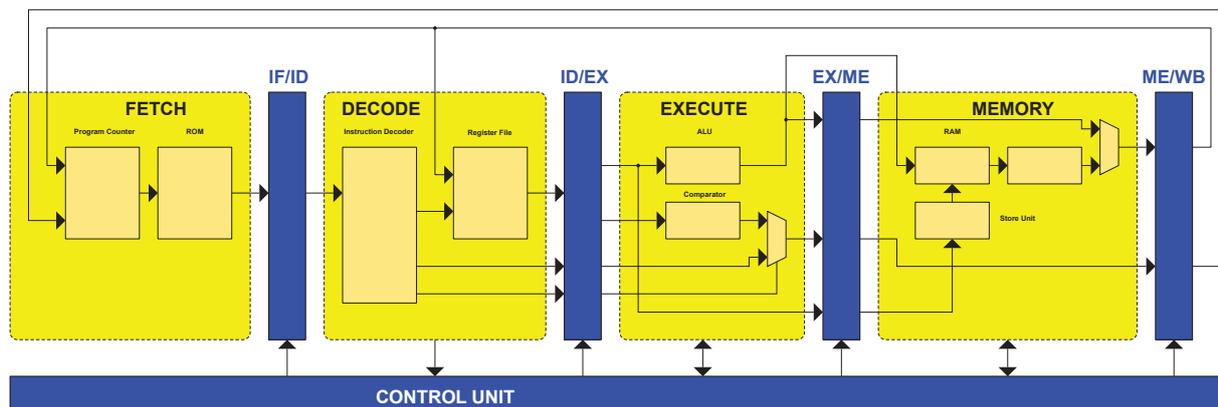


Fig. 11. Diagrama del riscv_PL_JD

Con el diseño adecuadamente validado hicimos una experiencia en una fpga. Realizamos la Síntesis y Place and Route del diseño utilizando el programa Quartus antes citado para un dispositivo *Cyclone IV* contenido en el **kit DE0 Nano**. Ruteamos los 8 bits menos significativos del registro x1 a unos puertos de salida de la fpga conectados a sendos leds y mediante un firmware de RISC-V adecuado producimos el destello de uno de ellos cada 1s. No se implementó un periférico *gpio* para no alterar el diseño que ya estaba validado.

A futuro están previstas varias líneas de trabajo a partir del procesador obtenido. La primera es abordar la Arquitectura Privilegiada [10] que permite el manejo de interrupciones, en particular la del temporizador. Una etapa posterior deberá incorporar los Mecanismos de Protección de Memoria que allanan el camino a la ejecución multitarea. Concomitante con esto pensamos en el desarrollo de periféricos estándares como *gpio*, *uart*, etc.

Más importante aún es la **Implementación Asíncrona** de esta arquitectura que fue la motivación inicial de este trabajo, para lo cual deberá sintetizarse la versión sincrónica con útiles de software para diseño VLSI.

- [8] <https://www.intel.com/content/www/us/en/software-kit/750368/modelsim-intel-fpgas-standard-edition-software-version-18-1.html>
- [9] <https://www.intel.la/content/www/xl/es/products/details/fpga/development-tools/quartus-prime/resource.html>
- [10] D. Patterson y A. Waterman., "RISC-V Guía Práctica," pp. 104–123, Julio 2018.

REFERENCES

- [1] RISC-V International, "RISC-V Specifications", <https://riscv.org/technical/specifications>.
- [2] K. Stevens, S. Rotem, and R. Ginosar. 1999. Relative Timing. In Proceedings of the 5th International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC '99). IEEE Computer Society, USA, 208.
- [3] Terasic, "DE0-Nano Development and Education Board", <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=593#contents>
- [4] D. Patterson y A. Waterman., "RISC-V Guía Práctica," pp. 17–18, Julio 2018.
- [5] D. Patterson y J. Hennesy., "Computer Organization and Design RISC-V Edition," pp. 518–526, 2018.
- [6] D. Patterson y J. Hennesy., "Computer Organization and Design RISC-V Edition," pp. 588–591, 2018.
- [7] Intel-Altera., "Cyclone IV Device Handbook Vol. 1," Chapter 3.



Prácticas de Programación de sistemas embebidos simples en modo remoto

Paola I. Beltramini
Facultad de Tecnología y Cs.Apl.
UNCA
Catamarca, Argentina
pbeltramini@tecno.unca.edu.ar

Pedro Valentín Nieva
Facultad de Tecnología y Cs.Apl.
UNCA
Catamarca, Argentina
valentin9702@gmail.com

Jesús E. Cano
Facultad de Tecnología y Cs.Apl.
UNCA
Catamarca, Argentina
jecano@tecno.unca.edu.ar

Resumen— En los tiempos que corren las tecnologías que denominamos el Internet de las Cosas (IoT) comienzan a ser usadas también en los ámbitos académicos, configurando un nuevo contexto conocido como la Educación 4.0, donde estas tecnologías se destacan por su aporte innovador para mejorar el proceso de enseñanza aprendizaje. Nuestro proyecto de investigación denominado “Estudio de factibilidad de Laboratorio Remoto para la formación experimental en las asignaturas de ciencias y tecnologías básicas de la Facultad de Tecnología y Ciencias Aplicadas de la UNCA” pretende indagar la posibilidades de desarrollar laboratorios remotos de bajo costo a partir de la utilización de la gran variedad de placas de microcontroladores y de sensores que se disponen actualmente, escenario que conduce a un sinfín de posibilidades de construcción de estos laboratorios. En este artículo se presentan los fundamentos de montaje de un laboratorio para el desarrollo de un experimento de acceso remoto para la programación y experimentación de un sistema embebido de bajo costo en el marco de la enseñanza práctica de asignaturas específicas de la carrera de Ingeniería Electrónica de la UNCA.

Palabras clave: IoT, Laboratorio Remoto, Programación, Arquitectura de red

I. INTRODUCCIÓN

La pasada pandemia de Covid-19 aceleró el proceso de transformación de una educación completamente presencial a una educación a distancia, o mejor aún, a una modalidad híbrida entre los dos modelos. En particular la formación práctica con su preeminencia experimental en cuanto a las habilidades a desarrollar por los estudiantes, representa un desafío para la educación a distancia, ya que descontando los múltiples trabajos de investigación que

demuestran la potencialidad de los laboratorios remotos (LR) en cuanto a participación y motivación, el desarrollar las plataformas y asociarlas a un contexto de aprendizaje representa una asociación aún por establecer.

Nuestro equipo de investigación ha analizado los LR disponibles en Internet e indagando en la bibliografía, presentamos los fundamentos de una arquitectura de red para un LR genérico para asociarlo al desarrollo propio consistente en un experimento de acceso remoto para la programación y experimentación de un sistema embebido de bajo costo en el marco de la asignatura Sistemas Lógicos II de la carrera de Ingeniería Electrónica de la Facultad de Tecnología y Ciencias Aplicadas de la UNCA.

Particularmente el desafío de formar a los estudiantes, principalmente en los cursos de las áreas de ciencia, ingeniería y tecnología, solo se adquieren a través del uso práctico en los laboratorios, la alternativa viable sería el uso de laboratorios remotos. Según [1] un laboratorio remoto permite realizar experimentos reales, desde cualquier lugar, en cualquier momento, a través de una interfaz de control remoto, permitiendo a un usuario con acceso a Internet configurar equipos experimentales e iniciar un experimento, que puede ser observado en tiempo real en la pantalla de una computadora que recibe las imágenes captadas por las cámaras locales.

Al brindar un laboratorio remoto, una institución educativa puede optimizar el uso de este recurso educativo: tenerlo disponible las 24 horas del día, cualquier día de la semana, atendiendo a una mayor cantidad de estudiantes con posibilidad de acceder a un laboratorio y realizar sus experimentos en horarios alternativos y en cualquier lugar, con el único requisito de tener acceso a una computadora o celular conectado a Internet.



Adicionalmente se facilita el acceso de aquellos alumnos que pretendan rehacer los experimentos en tiempos alternativos; los conceptos aprendidos en las clases teóricas son fijados e implementados en la práctica de forma autónoma por los alumnos; se realizan experimentos básicos sin requerir grandes recursos computacionales.

En la última década se han desarrollado varias plataformas experimentales con acceso remoto. Sin embargo, la realización de nuestro abordaje se justificará por sus características innovadoras en cuanto a: arquitectura simplificada, uso de dispositivos de bajo costo y menor necesidad de apoyo local o remoto para llevar a cabo actividades experimentales.

II. CONTEXTO DE LA INVESTIGACIÓN

La investigación se inserta dentro de la línea prioritaria "Innovación y Tecnologías Educativas" y el ODS (objetivo de desarrollo sostenible) N° 4: Garantizar una educación inclusiva, equitativa y de calidad y promover oportunidades de aprendizaje durante toda la vida de todos en el marco de proyectos presentado para la convocatoria 2022 de Proyectos de Investigación y Desarrollo Interdisciplinarios 2022 (PIDI 2022) de la Secretaría de Investigación y Posgrado de la UNCA en el marco de su Programa de Desarrollo Científico y Tecnológico.

El equipo de trabajo es multidisciplinar y está conformado por docentes investigadores de las carreras de Ingeniería Electrónica, Ingeniería Informática, y del departamento de Formación Básicas de la UNCA, que vienen trabajando y desarrollando actividades de docencia, investigación y vinculación tecnológica, en temáticas afines al proyecto. Integrantes del Laboratorio de Tecnologías de la Información y las Comunicaciones (LaTICs) y del grupo de investigación en Internet de las Cosas (IIoT) de la Facultad de Tecnología y Ciencias Aplicadas (FTyCA) de la UNCA, con experiencia en proyectos de investigación relacionados a los sistemas embebidos, la Domótica y al desarrollo e implementación de nodos y redes inteligentes para propósitos generales y aplicaciones educativas y ganaderas.

Respecto a las actividades que anteceden a esta investigación, debemos mencionar que en los años 2016 al 2019 hemos elaborado

prácticas de electrónica básica utilizando un laboratorio portátil de escritorio, es decir, con acceso local, pero de prestaciones adecuadas para llevar las prácticas a otros ámbitos (la casa o en aulas comunes dentro de la institución, etc.) sin tener que ocupar el siempre restringido espacio de los laboratorios.

Este proyecto, a través del cual se propone implementar un nodo de acceso remoto de un laboratorio de formación experimental, surge de proyectos anteriores: la investigación iniciada en las convocatorias 2016 y 2018 de Proyectos de Investigación y Desarrollo de la Secretaría de Ciencia y Tecnología de la Universidad Nacional de Catamarca (UNCA). Además, contó con financiamiento en el marco de la convocatoria de Proyectos Catamarqueños de Instrumentación Científica y Tecnológica ProCalCyT 2016 de la Subsecretaría de Ciencia y Tecnología de la Provincia de Catamarca.

El primer proyecto (2016) se enfocó en el diseño del prototipo, el PROCAYCIT en la construcción de las partes y las pruebas de funcionamiento, y en el 2018 se enfocaron el diseño de las actividades de enseñanza.

Teniendo el aporte de investigadores de las dos carreras de ingeniería afines se han publicado artículos relacionados a la investigación del proyecto en varios eventos científicos: CONAIISI (4º Congreso Nacional de Ingeniería en Informática/ Sistemas de Información); IV JUCEN - Jornada Educación en la Ciencia y en la Tecnología; WICC 2017 (Workshop de Investigadores en Ciencias de la Computación); CASE 2017 (Congreso Argentino de Sistemas Embebidos 2017 - Facultad de Ingeniería de la UBA); Libro: VI Producción Científica de la Facultad de Tecnología y Ciencias Aplicadas - UNCA - 2017; XII Jornadas de Ciencias y Tecnología de las Facultades de Ingeniería del NOA - Catamarca 2017; 1º Congreso Latinoamericano de Ingeniería - CONFEDI - Paraná - Entre Ríos 2017; Revista Argentina de Ingeniería (Vol.5); IV Congreso Argentino de Ingeniería (CADI) y X Congreso Argentino de Educación en Ingeniería (CAEDI) Córdoba 2018; entre otros.

Los integrantes del grupo, en su mayoría, han cursado y aprobado los cursos básicos y avanzados que se imparten año a año por parte de la Red CONFEDI de Sistemas Embebidos (RUSE).

La investigación hasta 2018 se ha fundado en aprovechar la disponibilidad de la mencionada CIAA y la experiencia previa que integrantes del grupo de investigación tenían con esa unidad de procesamiento. En 2019 el desafío fue encontrar un sistema embebido diferente con mayor potencialidad, pero a su vez sencillo de programar. Los resultados fueron no sólo en esa tendencia sino también en la de agregar instrumentación virtual, haciendo que la portabilidad del software nos bajará los costos de la interfaz gráfica del prototipo completamente portátil. Los resultados fueron publicados en diferentes eventos.

Dentro de las características más importantes de los laboratorios remotos está la posibilidad de operar a distancia un equipo real con todas las perturbaciones que el entorno pueda tener, lo que permite enriquecer el proceso de aprendizaje y la consolidación de conceptos a través de la observación, interpretación, comparación y análisis de los resultados [6], igualmente, cuentan con esquemas de seguridad que impiden el daño de equipos y puesto que son operados de manera remota no representan ningún riesgo para los usuarios, por otro lado, dan la libertad de experimentar a un ritmo propio, con pocas restricciones de horario y sin la necesidad de desplazarse hasta las instalaciones del laboratorio.

El reto en este proyecto consiste en diagramar y planificar las necesidades básicas para que nuestra institución tenga la posibilidad de proponer un nodo de acceso remoto a una experiencia de formación experimental o actividad de laboratorio, para las carreras de Ingeniería, ya sea en las asignaturas que manipulan dispositivos programables, circuitos de electrónica e incluso de formación básica como ser Física, es decir, en las que conocemos como asignaturas de las ciencias y las tecnologías básicas.

III. DESCRIPCIÓN DE LA PROPUESTA

En resumen, la propuesta consiste de un laboratorio remoto que consta de una página web que presenta un entorno de programación, donde el usuario debería ser capaz de escribir código, compilarlo y ejecutarlo en un microcontrolador real, permitiendo observar la placa en tiempo real la concreción de su programa y actuación sobre las salidas que

eventualmente programe.

El laboratorio deberá ser capaz de programar diversas placas, tanto de la familia de los microcontroladores de placa única o SBM (Arduino, Adafruit, STM, NodeMCU, LPC, etc.) como de las FPGA.

La arquitectura desarrollada consiste en usuarios conectados a Internet acceden a nuestra red local en la que alojamos un servidor que hace el procesamiento y supervisión de datos (puede ser un PC o un computador de placa única como raspberry pi) con interfaces adecuadas para conectarse con una o varias placas de microcontrolador que actúan y visualizan el experimento.

En la Fig. 1 se esquematiza el sistema realizado, consistente en una arquitectura de red de laboratorio remoto de bajo costo.

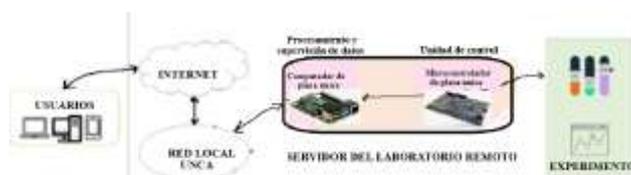


Fig. 1. Descripción de la propuesta

En una primera etapa en lugar de trabajar sobre un computador de placa única se utilizó un PC disponible en nuestro grupo de investigación. Allí se alojaron y configuraron el software de servidor y además de los programas para controlar y visualizar el experimento. Explicado esto en los siguientes ítems se presenta la descripción de las etapas de programación del PC y el hardware, es decir, a) la Comunicación con el cliente/usuario, b) el control del experimento y c) el hardware o experimento en sí. Finalmente, la implementación del LR se describe en dos ítems: d) diseño de la página web y e) programación de los archivos ejecutables por lote.

A. Comunicación con el cliente

Se implementó con una página web, que servirá de soporte para interfaz del usuario y de la programación, mediante XAMPP que es un paquete de software libre creado por la organización "Apache Friends" [2]. Así dispusimos de un servidor web Apache con protocolo HTTP (Hypertext Transfer Protocol, 'protocolo de transferencia de hipertextos') de código abierto, que nos permite crear una página

web, a través de la cual se transfieren archivos mediante las solicitudes que se hacen al servidor. XAMPP es de fácil utilización para crear servidores con Apache, cuenta con las herramientas que necesitamos, entre ellas PHP, que es un lenguaje de programación “Backend” con el cual podemos manejar las solicitudes con el servidor y los archivos que necesitamos. La aplicación web fue programada en HTML, PHP, CSS y JavaScript. Con HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) se da una estructura a la página web, con PHP (Hypertext Preprocessor) se programan las funcionalidades que se encargan de enviar y recibir la programación hecha por el usuario, y la ejecución de los archivos “batch” o por lotes (para que se realicen las tareas automatizadas de cargar, compilar y ejecutar la programación del usuario). Con JavaScript utilizamos lo que se conoce como Framework, es decir, una librería en la cual existen funciones ya creadas por sus desarrolladores que nos permiten simplemente utilizarlas en vez de crear funcionalidades desde cero. Para el entorno de codificación del usuario recurrimos a ACE (disponible en <https://ace.c9.io/>) un editor de códigos para páginas web que nos otorga las herramientas para hacer un editor de textos en el lenguaje de programación que necesitemos, incluye corrector de sintaxis, numeración de líneas de código, que nos permite poder brindar al usuario un lugar claro de programación y detectar dónde están sus errores a la hora de codificar. Con CSS (del inglés Cascading Style Sheets) conseguimos darle un “estilo a la página web”, es lo que nos permite modificar tamaños, colores, animaciones de cualquier objeto en nuestra página web, para hacer un entorno visual más amigable e intuitivo para el usuario.

B. Control del experimento

La herramienta que utilizamos es el MCUXPresso IDE, disponible para descarga en la página del fabricante [3], que es un entorno de programación que nos otorga todas las herramientas necesarias para programar la placa que utilizaremos para nuestro prototipo, que es la LPC1769 (NXP, 2013). Esta es la primera placa con la que dotamos al laboratorio como prototipo, pero el objetivo es que el LR sea capaz de programar cualquier placa que se

programme con MCUXPresso IDE fácilmente. Dentro del IDE, existen aplicaciones que nos permiten cargar, compilar y ejecutar código sin necesidad de ingresar al programa.

Con el MCUXPresso IDE mediante el “headless mode” podemos compilar y cargar el código a la placa a través de comandos de terminal. Para automatizar esta tarea se creó un archivo por lotes (conocidos como *.bat); de manera tal que baste con ejecutar estos archivos para que automáticamente se generen los archivos de compilación, compile y en caso de no tener errores se ejecute el programa en la placa. De manera similar, se crearon archivos *.bat adicionales para enviar mensajes a través del puerto serial a la placa WeMos D1 mini, de esta forma, según el dato recibido, la placa actuará enviando un nivel de voltaje a los pines que correspondan, simulando la presión de un botón, que se verá reflejado en el o los pines correspondientes de la placa LPC1769.

Por último, utilizamos un programa conocido como YawCam (disponible para descarga en <https://www.yawcam.com/>), que se encarga de capturar mediante webcam la ejecución de nuestro programa en la placa, lo que le permitirá al usuario observar la ejecución de su código en la placa en tiempo real.

C. Hardware del Experimento

El experimento físicamente consiste en actuar sobre dos placas que se usan para la ejecución de los programas que los estudiantes deben crear para cumplir las consignas de trabajo. Estas placas son las siguientes: LPC1769 [4]: Esta es la placa que se busca programar mediante la aplicación, que utiliza MCUXPresso IDE para cargar, compilar y ejecutar el código que recibirá desde el servidor. Esta placa estará conectada mediante un puerto USB al PC de manera tal que permanecerá alimentada y se podrán cargar el programa de manera sencilla. WeMos D1 mini: Esta pequeña placa que se programa mediante el IDE de Arduino, la utilizaremos como “botonera”. En pocas palabras, es la placa que se encarga que desde la aplicación web, podamos simular una interacción con la LPC1769, como lo es el simular la pulsación de un botón, la detección de un sensor, etc. Esta placa estará conectada mediante otro puerto USB al HOST, y se comunicará mediante el puerto serial, para

enviar y recibir los comandos necesarios para “accionar” el “botón” correspondiente, que luego recibirá la LPC1769. Otra parte del Hardware en la cual no entraremos en detalles es la webcam, que es con la que visionaremos desde la aplicación web la ejecución de nuestro programa cargado.

D. Diseño de la página del LR

En primer lugar, desde la aplicación Xampp se habilita nuestro servidor apache, se prueban y ven los aspectos del entorno de programación web para el usuario utilizando las herramientas previamente mencionadas. Se puede observar nuestro entorno en la fig. 2 donde queda evidenciado que al ser los objetos “responsive” nos aseguramos de que todo entre en la pantalla correctamente sin recortes.

Podemos ver también cómo el primer archivo cargado, que es de programación C se llama “basic” y contiene la “plantilla” para empezar a programar, incluidas las librerías necesarias, y definido nuestro “main”, los pines de la placa LPC que están conectados, y el cómo están conectados. Con esta información el usuario sabrá qué pines de entrada y salida debe utilizar según su proyecto. En la parte inferior se ven los cuatro botones y sus acciones. Cada acción está realizada en PHP. El botón de guardar se encarga de almacenar el archivo en nuestro servidor, es necesario que se guarde el archivo antes de poder ser compilado y ejecutado. El nombre del archivo guardado tendrá el nombre que indique el usuario en la parte superior del entorno de programación. El botón de cargar se encarga de cargar y mostrar en el entorno de programación un archivo que previamente haya guardado el usuario, de esta manera puede recargar un código que haya hecho previamente para continuar con su trabajo, o si simplemente quiere hacer otra prueba con su código, sin necesidad que empiece a programar desde cero. Cabe recalcar que nunca se podrá guardar un archivo con el nombre de “basic.c” ya que esto sobrescribiría la plantilla base de nuestro programa. Si el usuario intenta guardar un archivo con ese nombre, se mostrará un mensaje de alerta en la pantalla, indicando que debe elegir otro nombre. Los botones de compilar y ejecutar, se encargan de ejecutar los archivos por lotes correspondientes.



Fig. 2. Captura de pantalla del entorno de programación web del LR

E. Programación de los archivos ejecutables por lote

El diseño de los archivos por lotes tiene el fin de automatizar la interacción de los usuarios con las placas. Para la placa LPC1769, se crearon dos archivos. Uno se encarga de la compilación, y otro se encarga de la ejecución. La codificación en ambos casos se hace según lo indicado en el manual de usuario de MCUXpresso IDE. Para la placa WeMos D1 mini, es sólo un archivo que se encarga de enviar al puerto serial donde está conectada la placa un número. Según el número que sea, la placa “seteará” los pines de salida en 1 o 0 según corresponda. De esta manera es

que podemos simular los botones con un simple archivo por lotes.

De acuerdo a todo esto, el funcionamiento será el siguiente: cuando se compila el programa, la página esperará a que el programa MCUXPresso IDE termine de compilar el programa, y devolverá los resultados de compilación, que serán mostrados en la parte inferior de la página al usuario, así podrá ver si hubo algún error en la compilación y corregir el código si hiciese falta. Cuando la compilación fue correcta, se permite ejecutar el programa, de manera tal que el archivo por lotes correspondiente se encargará de enviar el código compilado a la placa.

Para ver la ejecución del programa solamente basta con presionar el botón “ver” para que se despliegue y abra la página donde se visualiza la placa a través de la webcam (ver fig. 3). Nuevamente podemos observar que los tamaños de la página se modifican según la resolución de pantalla. Debajo de la imagen de la cámara donde se puede observar la placa, tenemos botones. Cada botón se encarga de ejecutar el archivo por lotes correspondiente, que se encarga de enviar por puerto serial la señal que simule una presión de un botón a la WeMos D1 mini. Con esto podemos hacer los experimentos que el usuario necesita para comprobar que el código funciona correctamente.



Fig. 3. Captura de pantalla del entorno web con la visualización mediante webcam del LR

IV. RESULTADOS OBTENIDOS

El trabajo realizado nos ha permitido demostrarnos que la experimentación remota es

posible con herramientas accesibles sin grandes costos de hardware o software comercial.

Para este año 2024, el planteamiento del LR fue primero volver a poner en marcha el mismo, recuperar documentación y hacer pruebas. Todo este proceso nos ha llevado a la conclusión de que, para poder brindar la mejor experiencia posible, una de esas cosas a mejorar es la transmisión de la cámara web.

Así surgió la idea de estudiar sobre la tecnología de WebRTC, que nos permite transmitir datos en vivo, como si de una cámara de seguridad se tratase, dicha tecnología además le quita una carga muy grande al servidor.

Por lo tanto, la implementación de WebRTC nos lleva a establecer una conexión P2P (peer to peer) entre el servidor con la webcam, y el usuario que quiere programar el sistema embebido. Esto lo hacemos mediante un servidor STUN de google, que es completamente gratuito. Este servidor se encarga de realizar lo que es un “handshake” (“apretón de manos”), que es un proceso donde el servidor y el usuario establecen una sesión y una vez conectados se intercambian datos directamente entre ellos con su IP. Así una vez conectado el servidor y el usuario, estos se empiezan a comunicar entre sí, sin necesidad de seguir utilizando el servidor STUN, haciéndolo en tiempo real, y sin límite de tiempo ni otra restricción.

Este método nos permite que el servidor sea capaz de recibir más cantidad de usuarios, gastando menos recursos, ideal para la segunda parte de lo que se buscaba trabajar este año; el añadido de más placas de programación.

Este año partimos con la intención de crear algo más amplio, extendiéndose a placas como la ESP32, la ESP8266, Arduino, etc. En la fig. 4 se observa como quedaría la página principal de nuestro LR con las placas a elegir. Todo esto culminará en crear nuestra propia red de entornos de programación y testeo remotos, dando la posibilidad de programar múltiples placas, y pudiendo abarcar a más usuarios a la vez.

Actualmente se inició el proceso de probar el servidor en una raspberry pi, donde el objetivo es que haya al menos un laboratorio por raspberry, pudiendo así crear una red de laboratorios ocupando la menor cantidad de espacio físico posible.

Luego de esto se pretende crear una base de datos donde se pueda dividir los usuarios ya sean profesores o alumnos, y que los alumnos sean capaces de guardar su código y los profesores de evaluarlos. Pudiendo finalmente así crear una implementación completa del laboratorio acorde a la educación a distancia.

del programa RLAB (Red de laboratorios remotos del CONFEDI – Consejo Federal de Decanos de Ingeniería)

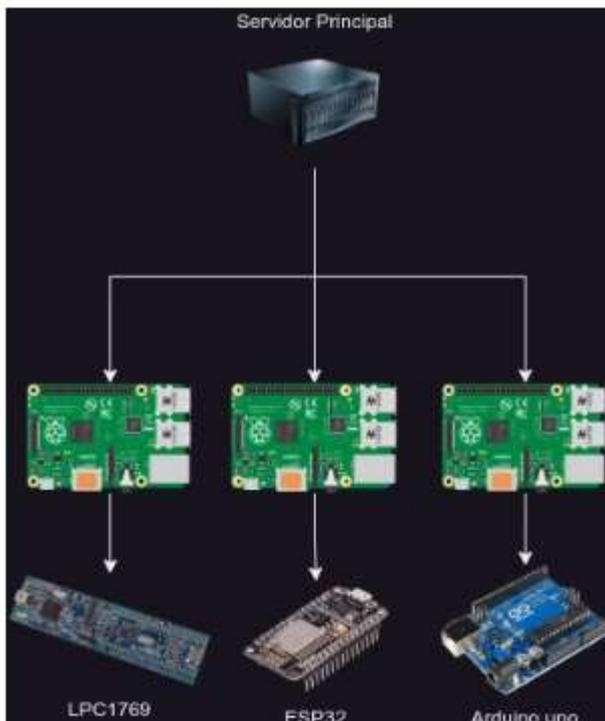


Fig. 4: Forma que tomará el servidor con las placas a programar

En paralelo a nuestro equipo, dentro del mismo grupo se están elaborando otros LR que remotizarán prácticas de asignaturas de las ciencias básicas, dos del área Física y una del área Química (ver fig. 5). Estas prácticas cuentan con financiamiento desde la SPU (Secretaría de Políticas Universitarias) a través

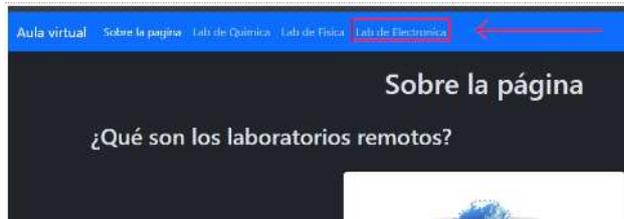


Fig. 5: Imagen de lo que será el LR de la UNCA

V. CONCLUSIONES

En el desarrollo de este proyecto se logró demostrar que el desarrollo de LR se puede alcanzar con herramientas accesibles sin grandes costos de hardware o software comercial, y con la intervención de nuestros estudiantes investigadores que realizaron gran parte de la programación aquí descrita, lo que nos ha dado la posibilidad de cumplir dos metas a la vez: el desarrollo en un bajo costo y el aprendizaje para los estudiantes y docentes investigadores.

REFERENCIAS

- [1] Vilela, D. C., Germano, J. S. E., Monteiro, M. A. A., & Carvalho, S. J. "Estudo comparativo de um experimento de eletrodinâmica: Laboratório Tradicional x Laboratório Remoto". *Revista Brasileira De Ensino De Física*, v.41, n. 4, e20190041. <https://doi.org/10.1590/1806-9126-RBEF-2019-0041>. 2019.
- [2] <https://www.apachefriends.org/es/index.html>.
- [3] <https://www.nxp.com/design/software/development-software/mcuxpresso-software-and-tools/mcuxpresso-integrated-developmentenvironment-ide:MCUXpresso-IDE>
- [4] NXP. LPC1769 Datasheet (PDF) – NXP Semiconductors (2013). Recuperado en Julio 2023 en: <https://pdf1.alldatasheet.es/datasheetpdf/view/328846/NXP/LPC1769.html>.



Contador de vueltas reversible para bobinadora de núcleos toroidales *

Juan Manuel Conde Garrido

Edgardo Vinson

Alejandro Jurado

Josefina María Silveyra

INTECIN, Facultad de
Ingeniería, Universidad de
Buenos Aires - CONICET

Dpto. de Energía, Facultad
de Ingeniería, Universidad
de Buenos Aires

Dpto. de Energía, Facultad
de Ingeniería, Universidad
de Buenos Aires

INTECIN, Facultad de
Ingeniería, Universidad de
Buenos Aires - CONICET

Buenos Aires, Argentina

Buenos Aires, Argentina

Buenos Aires, Argentina

Buenos Aires, Argentina

jmcondegarrido@fi.uba.ar
0000-0002-7689-3912

evinson@fi.uba.ar

ajurado@fi.uba.ar

jsilveyra@fi.uba.ar
0000-0003-0307-3419

Resumen— Las bobinas son componentes esenciales de diversos dispositivos eléctricos. Recientemente, presentamos una máquina bobinadora de núcleos toroidales de bajo costo, útil para prototipos y producciones de bajo volumen. La bobinadora utilizaba un contador de vueltas arrolladas basado en un smartphone y un imán, que sufría de efectos rebote y no podía detectar cambios del sentido de rotación. Pero el registro preciso del número de vueltas bobinadas es crucial para el desempeño adecuado del inductor a fabricar. Por ello, este trabajo tiene como objetivo implementar en la bobinadora desarrollada un contador de vueltas confiable y reversible. A través de un microcontrolador ATmega328P, el sistema propuesto filtra y decodifica la señal de un arreglo de 3 sensores ópticos infrarrojos y calcula el número de vueltas bobinadas o desbobinadas. Por medio de periféricos, brinda funciones de notificación, pausa y edición. Ensayos de sensibilidad garantizan la exactitud del contador desarrollado bajo distintas condiciones ambientales.

Palabras clave: contador reversible, bobinadora toroidal, sensor IR, ruido ambiental.

I. INTRODUCCIÓN

Las bobinas son componentes esenciales en una variedad de aplicaciones, como transformadores y motores eléctricos. De hecho, el bobinado del estator es a menudo considerado el paso más crítico en el proceso

de producción de motores eléctricos [1]. La técnica de bobinado sobre núcleos toroidales es rara vez discutida en la literatura y requiere una estrategia fundamentalmente distinta a la empleada en las máquinas bobinadoras convencionales: el alambre no puede ser alimentado de manera continua desde los carretes de cobre, ya que su extremo debe completar una revolución alrededor del eje de bobinado por cada vuelta a bobinar [2].

Recientemente, hemos presentado una prueba de concepto de una máquina bobinadora manual de núcleos toroidales de bajo costo (ver Fig. 1), diseñada para etapas de prototipado o en producciones de bajo volumen, donde las máquinas industriales no son una opción viable debido a su elevado costo [3]. En este sistema, el alambre es transferido desde un carrete de cobre primario a un anillo acanalado trasbordador durante la operación de carga (anillo de carga), para luego ser desenrollado sobre el núcleo toroidal con la ayuda de un segundo anillo durante la operación de bobinado (anillo bobinador).

Durante ambas operaciones, carga y bobinado, es importante llevar un conteo correcto de la cantidad de vueltas dadas. Por un lado, se utiliza el registro del número de vueltas del anillo de carga para calcular la cantidad de alambre a transferir durante la operación de carga. Por otro lado, el número de vueltas del anillo bobinador equivale al número de vueltas arrolladas sobre el núcleo toroidal en la operación de bobinado.

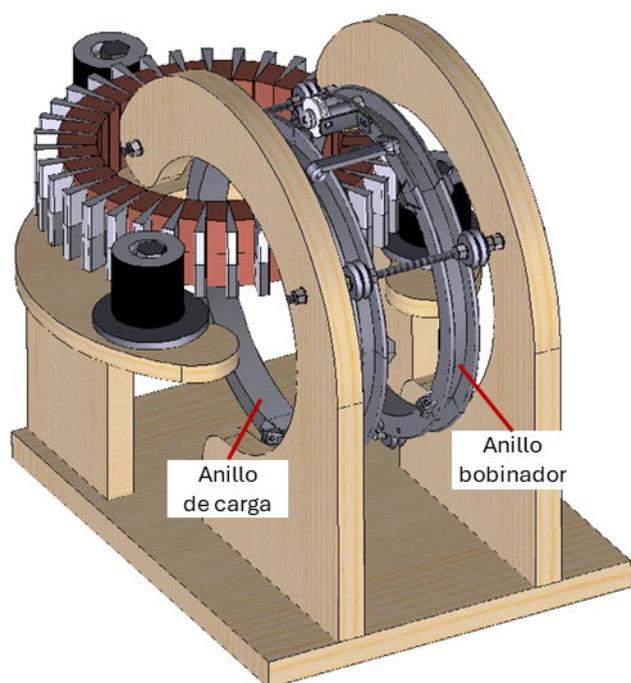


Fig. 1. a) Diseño de la máquina bobinadora de núcleos toroidales presentada en [3], tras mejoras que le otorgan mayor robsutez.

Para contar el número de vueltas de un anillo, la máquina original utilizaba un contador de revoluciones simple basado en la detección de campo magnético. El sistema consistía en un teléfono inteligente equipado con un sensor interno de campo magnético, la aplicación móvil "Magnetic Field Counter" [4], y un imán permanente adherido al anillo a controlar. Desafortunadamente, los sensores magnéticos son propensos a detectar múltiples transiciones o "rebotes" en sistemas mecánicos, lo que puede fácilmente provocar errores en el conteo de las vueltas arrolladas sobre el núcleo. Otra limitación es que, al igual que la mayoría de los contadores simples, el utilizado en el diseño original no es reversible; es decir, no puede contar hacia atrás, ya que solo suma y no resta la cantidad de vueltas en sentido inverso [5]. En lugar de registrar el conteo real, el contador incrementa su valor tanto si el sistema avanza como si retrocede, resultando en la pérdida del conteo preciso si el anillo rota en dirección inversa.

Como señaló Wilkinson, "el valor de un contador que 'descuenta' cuando se desenrolla alambre de una bobina es muy alto; (...) es frecuentemente necesario desenrollar un poco"

[6]. Las consecuencias de registrar un exceso en el número de vueltas durante la operación de carga pueden mitigarse cargando preventivamente un poco de alambre extra: de no cargar suficiente alambre en el anillo de carga, no podrá agregarse posteriormente más cantidad para completar la bobina, mientras que el alambre remanente en el anillo deberá ser descartado. Sin embargo, es inaceptable incluso una pequeña discrepancia en el número de vueltas durante la operación de bobinado: puede afectar significativamente el desempeño de la máquina eléctrica a construir [7], y ser especialmente crítico en el caso de bobinas conectadas en paralelo debido a posibles corrientes inducidas de retorno [8].

Con la simplicidad y el bajo costo como principios fundamentales de diseño, este estudio se enfoca en la implementación de un contador de vueltas económico, confiable y reversible.

II. DISEÑO Y DESARROLLO

Para contar el número de vueltas arrolladas sobre el núcleo toroidal, es necesario medir las revoluciones del anillo bobinador de la bobinadora. Es requisito que el contador sea capaz de operar tanto cuando se gira el anillo bobinador con la mano (modo manual), como con un motor a través de un sistema de transmisión con correas y poleas (modo semi-automático añadido al diseño de [3], para el que la mano aún debe ser utilizada para el movimiento trasversal del alambre sobre el núcleo toroidal). Debido a la necesidad de funcionar aún en modo manual, no es posible calcular el número de vueltas de los anillos de manera indirecta, registrando las revoluciones del eje del motor utilizado para la semi-automatización de la máquina. Para esta aplicación no se necesita una alta resolución angular, por lo que no se implementó una solución basada en *encoders* de cuadratura.

Siguiendo los principios de simplicidad y bajo costo, se diseñó un sistema controlado por un microcontrolador ATmega328P montado en una placa de desarrollo Arduino Nano [9]. El sistema incluye un arreglo de 3 (tres) sensores o transductores en la estructura de madera de la bobinadora, estratégicamente colocados alrededor del anillo bobinador a intervalos de

120° aproximadamente, y 1 (un) objetivo o *target* instalado en la superficie del anillo bobinador, frente a los sensores. Esta configuración permite diseñar un algoritmo simple que, dependiendo la dirección de la rotación, suma o resta tercios de revoluciones del anillo bobinador. La Fig. 2 muestra, a modo de ejemplo, el diagrama de flujo seguido por el algoritmo utilizado para el sensor ubicado a 120°; se utiliza uno análogo para los sensores ubicados a 0° y 240°. El conteo de las revoluciones del anillo de carga durante la operación de carga se realiza acoplando mecánicamente ambos anillos.

De las diversas tecnologías de sensores disponibles en el mercado, se optó por sensores ópticos infrarrojos (IR) debido a su bajo costo, fácil acceso y eficiencia en el consumo de energía. Esta última característica permite que la bobinadora funcione con baterías cuando se opera manualmente. Se utilizaron módulos TCRT5000, que combinan un emisor IR y un fototransistor en un solo paquete y, como objetivo reflectante, utilizamos una moneda brillante de 5 pesos argentinos. Así, el módulo emite una señal IR que, cuando es interceptado por la moneda en el anillo que gira, se refleja y es detectado por el fototransistor.

Si bien el módulo TCRT5000 dispone de una carcasa para filtrar luz visible, el fototransistor que mide la intensidad de luz IR es igualmente bastante sensible a variaciones de luz y temperatura en el ambiente. Si el sensor es calibrado por la mañana, no puede ser utilizado por la tarde. Lo mismo ocurre con cambios en las luces eléctricas de la sala. Por esta razón, se implementó un método de filtrado que permite evitar la calibración de los módulos cada vez que cambian las condiciones ambientales. Al encender el LED emisor IR, el fototransistor recibe la suma de la luz IR reflejada por el objetivo y el ruido ambiental de fondo, proveniente de otras fuentes de luz visible e IR. Al apagar el LED IR, el fototransistor recibe únicamente el ruido ambiental de fondo. La estrategia consiste entonces en realizar seguidamente dos lecturas del fototransistor para cada módulo TCRT5000, una con el LED IR encendido y otra con el LED IR apagado. La diferencia de señales brinda la lectura sin ruido.

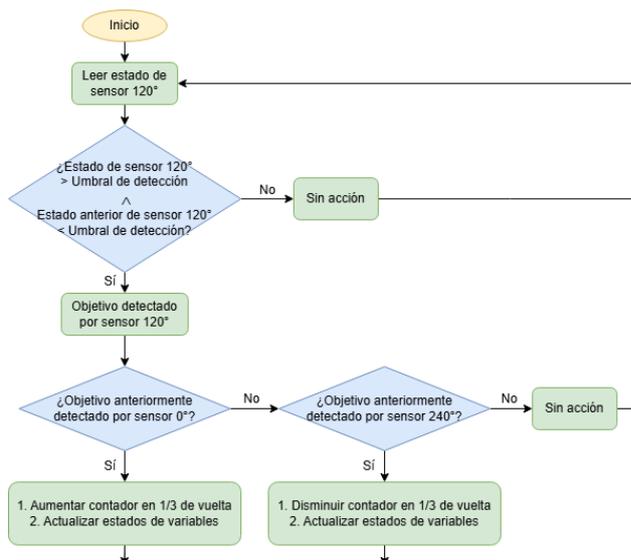


Fig. 2. Diagrama de flujo del algoritmo utilizado para el sensor ubicado a 120° sobre la estructura de la bobinadora y frente al anillo bobinador.

Con el fin de ensayar el método de filtrado, se realizaron pruebas con un módulo TCRT5000 bajo distintas condiciones ambientales. Los resultados se muestran la Fig. 3, donde se observan las señales obtenidas por el fototransistor con el LED IR encendido, con el LED IR apagado y la diferencia de ambas, demostrando la efectividad del método. Luego, se caracterizaron los escenarios de un módulo TCRT5000 frente al anillo negro y frente al anillo con la moneda. Así, se determinó una intensidad umbral de 500 para caracterizar el estado de detección del objetivo reflectante (Fig. 4).

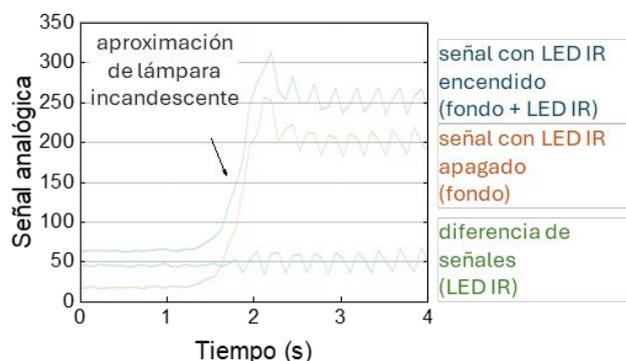


Fig. 3. Demostración de eliminación de ruido de fondo de la señal adquirida por el fototransistor del módulo sensor IR TCRT5000. La señal filtrada, en ausencia del objetivo reflectante, mantiene su valor promedio al aproximar a la bobinadora una lámpara incandescente.

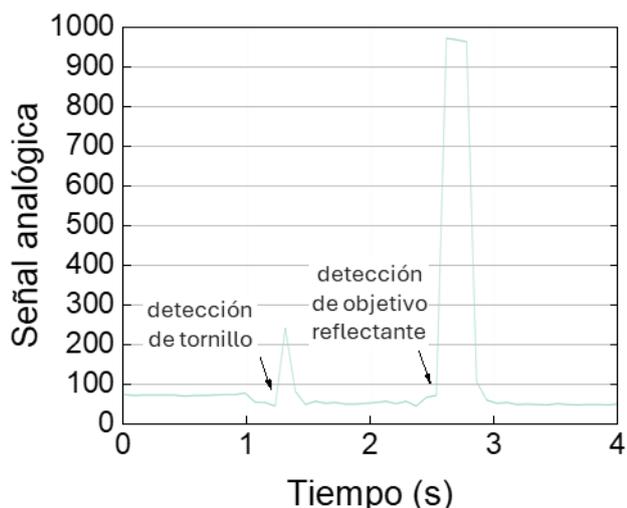


Fig. 4. Señal filtrada de sensor IR durante el giro de la rueda bobinadora. Se indican los instantes en los que pasa un tornillo del anillo y el objetivo reflectante frente al módulo sensor IR. Se estableció un valor umbral de 500 para caracterizar el estado de detección del objetivo.

El arreglo de sensores es controlado por el microcontrolador ATmega328P y un circuito simple basado en resistores y un transistor BJT (*Bipolar Junction Transistor*) (Fig. 5). Cada LED emisor IR es conectado en serie con un resistor de 330 Ω a 5 V (con una resistencia menor, como la de 180 Ω recomendada en [10], satura la señal del fototransistor con la luz IR reflejada por el anillo de ácido poliláctico o PLA negro brillante). El transistor controla el encendido de los tres LEDs IR (deben encenderse y apagarse para la operación de filtrado y también se pueden apagar en caso de pausar el contador para una mejor eficiencia energética y prolongar su vida útil). El fototransistor es conectado a Tierra a través de un resistor *pull-down* (como es aconsejado en [10]) de 10 k Ω . La señal del fototransistor es adquirida por un pin analógico del microcontrolador, devolviendo valores entre 0 y 1023.

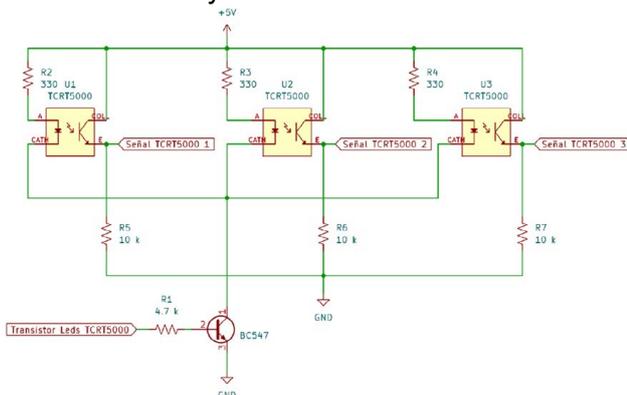


Fig. 5. Esquemático del circuito de sensores ópticos IR basado en resistores.

Además del arreglo de módulos de sensores, el microcontrolador controla una serie de periféricos que brindan funciones de notificación, pausa y edición (ver esquemático en Fig. 6).

Se implementó un método de notificación sonora para que el usuario pueda fácilmente detectar casos de falla en el registro de vueltas por parte del contador. Un zumbador o *buzzer* emite un sonido cada vez que uno de los tres fototransistores detecta el paso del objetivo reflectante; el sonido es de mayor frecuencia cada vez que se completa un número entero de vueltas. Si no suena por más que gire el anillo, existe una falla que debe ser analizada y corregida. Como se utilizó un *buzzer* económico del tipo inductivo, se le conectó en paralelo un diodo *flyback* para proteger al microcontrolador.

Un pulsador con retención permite entrar y salir del modo pausa, en el que se apagan los LEDs IR y no se detecta la rotación del anillo.

Un *encoder* rotativo con pulsador ofrece una función de edición que permite agregar o restar vueltas de manera deliberada mediante su rotación en sentido horario o antihorario, respectivamente (el pulsador debe presionarse para entrar o salir del modo de edición).

El contador utiliza una pantalla o *display* LCD 16x02 con controlador I2C (*Inter-Integrated Circuit*) en el que notifica continuamente el número de vueltas bobinadas mediante 3 dígitos enteros y una fracción de tercios. La pantalla parpadea cuando está en modo pausa o edición. Cuando está en modo edición, indica además el número de vueltas establecidas con el *encoder* rotativo.

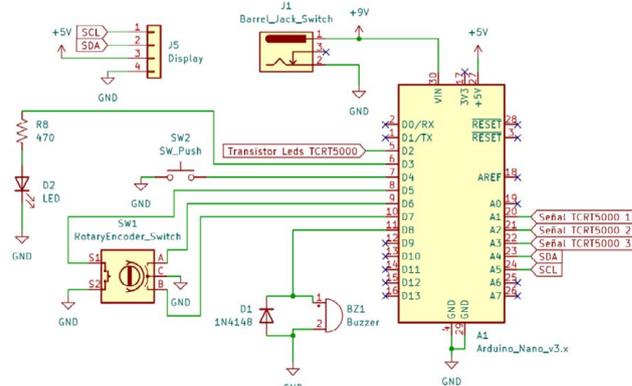


Fig. 6. Esquemático de conexiones del microcontrolador con sensores y periféricos.

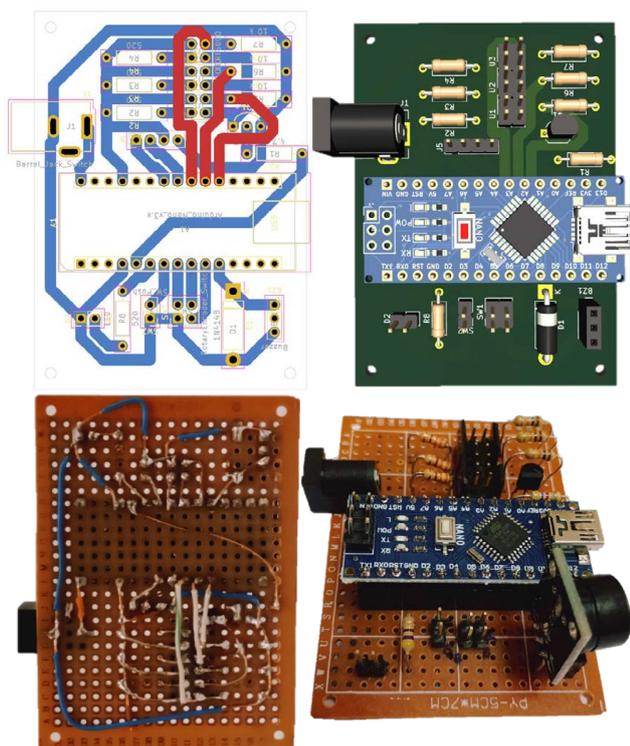


Fig. 7. Arriba: Diseño de circuito. Abajo: Circuito soldado en placa experimental perforada.

El circuito del contador fue diseñado en KiCad [11] y soldado en una placa experimental perforada (Fig. 7). El software fue desarrollado en el entorno de desarrollo integrado (IDE) de Arduino [9] e implementa una máquina de estados finitos que resuelve de manera eficiente la transición entre los estados contador, pausa y editor.

III. RESULTADOS

El contador de vueltas implementado fue utilizado para arrollar exitosamente bobinas de 109 vueltas sobre un núcleo toroidal ferromagnético con una matriz de ranuras polimérica fabricada por manufactura aditiva o impresión 3D (Fig. 8). Su funcionamiento demostró ser sencillo, eficaz y confiable, tanto al utilizar a la bobinadora en modo manual como el semiautomático.

RECONOCIMIENTOS

Este trabajo contó con el apoyo del CONICET, la Universidad de Buenos Aires (UBACyT 20020190200001BA), y la ANPCyT (PICT-2019-2019-02122).

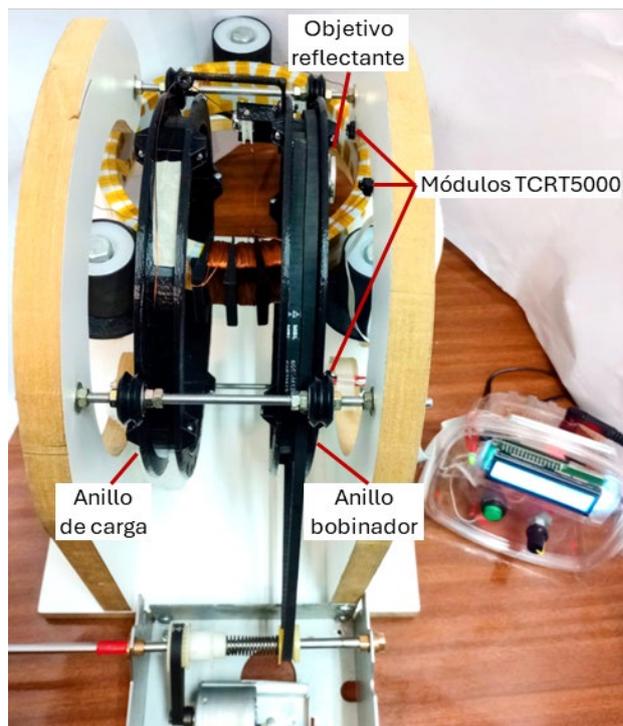


Fig. 8. Bobinas arrolladas sobre núcleo toroidal ferromagnético, recubierto con Mylar, Kapton y sector de ranuras poliméricas. El *display* del contador de vueltas indica la cantidad de vueltas.

REFERENCIAS

- [1] J. Fleischer, S. Haag, and J. Hofmann, "Quo vadis winding technology," *A study on state of the art and research on future trends in automotive engineering*, 2017 https://www.wbk.kit.edu/downloads/2017_02_21_Studie_Wickeltechnik_final_EN.pdf
- [2] F. Planer, "Toroidal Winding Machines," *Journal of Scientific Instruments*, vol. 20, p. 185, 1943. 10.1088/0950-7671/20/12/301
- [3] J. M. Conde Garrido and J. M. Silveyra, "Design and fabrication of a low-cost coil winding machine for slotless toroidal stators," *LACCEI*, vol. 1, 2023. 10.18687/LACCEI2023.1.1.1609
- [4] Keuwlsoft, "Magnetic Field Counter," <https://www.keuwl.com/apps/037magneticfieldcounter/> 03/28/2023
- [5] D. Roegel, "Carries Stripped to the Bone: Episodes in the History of Coaxial Modular Digital Counters," *IEEE Annals of the History of Computing*, vol. 39, pp. 55-64, 2017. 10.1109/mahc.2017.3481339
- [6] K. Wilkinson, *Rewinding Small Motors*: Elsevier, 1965 <https://www.sciencedirect.com/book/9780408003087/>
- [7] S. De Silva and G. Perera, "Design and fabrication of a digital turns counter for an armature winding machine," *Proc. Annual Symposium on Research & Industrial Training*, vol. 3, pp. 297-302, 2016 <http://repository.wyb.ac.lk/bitstream/handle/1/1516/ASRITE-2016-297.pdf>
- [8] J. M. Silveyra and J. M. Conde Garrido, "Electrically connected and magnetically coupled inductors: Aiding or opposing fluxes?," *American Journal of Physics*, vol. 90, pp. 365-372, 2022. 10.1119/5.0067939
- [9] "Arduino," <https://www.arduino.cc/>
- [10] Vishay Semiconductors, "Application of Optical Reflex Sensors TCRT1000, TCRT5000, CNY70," 2002 <https://www.vishay.com/docs/80107/80107.pdf> 08-09-24
- [11] KiCad Development Team, "KiCAD," <https://www.kicad.org/>

Comunicación entre sistemas embebidos y GNU Octave con fines didácticos

E. Sergio Burgos

Facultad Regional Paraná

Universidad Tecnológica Nacional

Paraná, Entre Ríos, Argentina

sergioburgos@frp.utn.edu.ar

Resumen—En este trabajo se propone un protocolo de comunicación para GNU Octave, con fines didácticos, que posibilita el intercambio de información con sistemas embebidos utilizando puertos serie como capa física. El desarrollo, fue realizado en lenguaje C y con scripts para GNU Octave. Se describen las aplicaciones de prueba realizadas en tres plataformas de hardware.

Palabras clave—sistemas embebidos, GNU Octave, guiEditor, Cortex M3, Arduino, ESP32

I. INTRODUCCIÓN

El uso de software de cálculo en el ámbito universitario, en carreras de grado y posgrado, permite obtener rápidos resultados al evaluar algoritmos y procesos complejos. Dentro de las alternativas disponibles en cursos relacionados al control de procesos y procesamiento de señales Matlab, GNU Octave [1] y Scilab son de uso común. Entre estas dos últimas alternativas, ambas gratuitas y de código abierto, GNU Octave suele preferirse en algunos ámbitos por su bajo consumo de recursos y compatibilidad con Matlab.

El uso de aplicaciones *script* gráficas a partir de GNU Octave ha presentado resultados satisfactorios en educación de diferentes niveles [2], [3] y con resultados igualmente favorables se observa la incorporación de sistemas embebidos. Ya sea comunicando estos sistemas desde las herramientas de cálculo [4], [5] o no [6]. Esto ha motivado a proponer un método de comunicación entre GNU Octave y sistemas embebidos que posibilite interactuar con sistemas físicos de diferente modo. A fin de mejorar la experiencia de usuario, el abordaje se realiza considerando el uso de interfaces gráficas de usuario.

II. ANTECEDENTES

GNU Octave es una herramienta de cálculo de código libre, lo que dentro del ámbito científico y académico constituye un valor agregado, permitiendo compartir aplicaciones y generar entornos de ejecución a fin de reproducir resultados experimentales [7]. El motor de cálculo es capaz de resolver un conjunto de operaciones mientras que herramientas para la resolución de problemas de áreas específicas se incorporan a través de

paquetes. En su origen, se diseñó como una herramienta para ser ejecutada en terminales de texto. Según evolucionó, se incorporaron características gráficas, tanto para interactuar con el motor de cálculo como para la representación de interfaces visuales. No obstante, no incorpora en su distribución un editor de interfaces gráficas para aplicaciones *script*.

Entre los paquetes existentes para GNU Octave `instrument-control` [8] permite la comunicación con sistemas externos utilizando diferentes protocolos y tecnologías. En este trabajo es de interés la comunicación a través de interfaces de comunicación serie (`serialport`) y este paquete incorpora funcionalidades para la identificación de dispositivos disponibles en el sistema, envío y recepción de datos. Las cuales son el punto de partida para la implementación de comunicaciones con dispositivos externos.

guiEditor [9] por su parte es un entorno para el desarrollo de aplicaciones *script* visuales, que es capaz de generar código fuente nativo para GNU Octave. Entre los controles incorporados en su interfaz, incluye uno para establecer comunicación con placas de desarrollo Arduino, pero la interacción está orientada a realizar operación sobre la placa, no previendo una comunicación de iguales características en ambas direcciones.

En este trabajo se aborda el problema de la comunicación entre sistemas embebidos y GNU Octave para el desarrollo de aplicaciones *script* gráficas utilizando guiEditor. El objetivo es lograr una forma de comunicación bidireccional que permita sacar provecho al mismo tiempo de las capacidades de los sistemas embebidos y del motor de cálculo.

III. PROTOCOLO PROPUESTO

Considerando que se propone un protocolo de aplicación general, ha sido diseñado a fin de permitir trabajar con una carga útil de información de longitud variable.

Se basa en comandos (identificadores numéricos definidos por el desarrollador) a fin de indicar la operación a realizar. Según el valor del comando, se utilizan parámetros adicionales, cuya cantidad es indicada como un campo. De este modo, en una aplicación particular, para

cada transacción siempre se tendrán paquetes de una longitud predeterminada.

A fin de asegurar una correcta interpretación de los paquetes se incorporan marcas de inicio, finalización y suma de comprobación implementada a través de un código de redundancia cíclico de 8 bits. Esto fija en cinco bytes el mínimo tamaño de un paquete.

Si bien la comunicación es bidireccional, se asume que el inicio de las transferencias se realizará desde el motor de cálculo, por lo que puede entenderse que obrará como maestro, y el sistema embebido generará una respuesta (esclavo). Se optó por esta forma de trabajo debido a que la recepción de la información en GNU Octave debe realizarse utilizando *polling*, de este modo, cuando se desea acceder a un valor externo se lo indica al sistema embebido enviando un paquete y la respuesta contendrá el valor requerido. En caso de que el comando tenga como propósito realizar una acción, se tendrá una confirmación sobre su ejecución como respuesta.

III-A. Implementación para microcontroladores

La implementación se realizó utilizando lenguaje C a fin de lograr compatibilidad con diferentes entornos de desarrollo. El procesamiento del protocolo se realiza utilizando una máquina de estados que, según se recibe cada byte, conmuta entre los diferentes estados. Cada estado se encuentra asociado a una función, estando almacenada la dirección de todas las funciones en un vector. Luego, cada estado es identificado con un valor numérico siendo coincidente con la posición de la función asociada en el arreglo.

Se utiliza una estructura llamada `protocolData` (Listado 1) para almacenar el estado del procesamiento, en ella se almacena la identificación del estado siguiente y los valores recibidos. Estos son interpretados según la máquina evoluciona por los diferentes estados. El campo más significativo es `dataVal` ya que almacena los argumentos que corresponden con cada comando (identificado por el campo `cmd`).

```
typedef struct
{
    uint8_t cmd;
    uint8_t dataVal[MAX_DATA_LEN];
    uint8_t dataLen;
    uint8_t dataInx;
    uint8_t crc;
    uint8_t nextState;
}protocolData;
```

Listado 1: Definición de la estructura `protocolData`

En el último estado, al recibir el indicador de finalización establecido en el protocolo, es posible que se tenga un paquete válido. Si es así, se invoca a una función que contendrá las acciones a ejecutar como consecuencia de la información recibida. Para esto se cuenta con un vector de punteros a funciones (`cmdActionList`) donde

la posición en el arreglo de cada función es coincidente con el valor del comando. Por esto, el valor de los comandos utilizados son correlativos iniciando en el valor cero.

Cada función recibe como argumento los datos recibidos a través de la estructura `protocolData` y que ya fueron validados durante la ejecución de la máquina de estados.

Con esta implementación, extender el protocolo a fin de implementar nuevas operaciones en el sistema embebido requiere solo realizar la implementación de la función deseada, agregarla al arreglo de acciones y modificar la cantidad de acciones reconocidas como válidas a través de la constante simbólica `LAST_CMD` definida en el *header* `serialProtocol.h`.

IV. IMPLEMENTACIÓN EN GNU OCTAVE

Se realizó utilizando `guiEditor` como entorno de desarrollo a fin de construir una aplicación gráfica. En estos casos, al requerir utilizar un recurso de comunicaciones tal como un puerto, suele solicitarse al usuario al inicio de la aplicación indicar el puerto a utilizar. En el desarrollo se optó por una forma alternativa, donde al requerir realizar la primera transferencia de información se solicita al usuario indicar el puerto a utilizar y todas las operaciones siguientes se realizan utilizando el mismo dispositivo. Esta forma de trabajo posibilita que, si la aplicación permite realizar operaciones que no requieren comunicarse con un sistema embebido, no sea necesario contar con un puerto de comunicaciones para ejecutar la aplicación.

La implementación de la selección del puerto a utilizar se realizó a través de una función (`getSerialPort`) que recibe el *handle* de la figura asociada a la ventana y un valor lógico como argumentos. El *handle* de la figura asociada a la ventana principal es utilizado a fin de resolver el paso de valores entre la ventana principal y la ventana secundaria donde se selecciona el puerto utilizando la función `guidata`. El segundo argumento tiene como valor predeterminado `false` e indica si se desea liberar el puerto, de este modo, al finalizar la aplicación es posible indicar de forma explícita la liberación del recurso para poder utilizarlo nuevamente. Cuando se invoca a la función `getSerialPort` por primera vez se hace visible una ventana con la lista de puertos disponibles en el sistema, para esto se utiliza la función `serialportlist` del paquete `instrument-control`. Una vez seleccionado un puerto, es inicializado almacenando la referencia al objeto creado en una variable de modo persistente. En las subsiguientes invocaciones a esta función, si esta variable contiene un valor diferente de cero se asumirá que se ha seleccionado previamente un puerto y retornará la referencia al objeto sin necesidad de realizar la selección del puerto nuevamente.

La configuración del puerto se realiza indicando un valor de `timeout` positivo (5 segundos). De este modo

al realizar una operación de lectura, posterior a enviar valores a través del puerto, se esperará este tiempo. En caso de no haber recibido ningún valor, esta situación será detectada indicando un error de comunicación. La demora debida al procesamiento del comando (acciones realizadas en el sistema embebido) está contemplada en el valor del `timeout` indicado.

Cada uno de los valores recibidos es procesado de modo similar a como se propone en la implementación de microcontroladores, a través de una función llamada `procResponse` que recibe como argumento el valor recibido. Utiliza una máquina de estados a fin de realizar el procesamiento, en este caso, los valores asociados a la misma son almacenados como variables locales persistentes. Si bien esto impide gestionar más de un flujo de comunicaciones a la vez, tiene como ventaja que simplifica la invocación a la función. Cada estado tiene asociado una función, y todas las funciones están referenciadas desde un arreglo de celdas (*cell array*).

La función `procResponse` retornará dos valores, un vector con los datos útiles contenidos en el paquete y una variable lógica que indicará si se ha recibido un paquete válido. De este modo, según se reciban nuevos valores, se deberá invocar a esta función hasta que se confirme obtener un paquete completo.

Cada operación realizada en el sistema embebido es implementada como una función en GNU Octave, que recibe como argumento el objeto que referencia al puerto de comunicaciones y los argumentos requeridos. Estas funciones construyen el paquete y lo envían para luego intentar recibir la respuesta, cada valor es utilizado como argumento de `procResponse` hasta obtener un paquete válido o finalizar por `timeout`. Finalmente se retornará un valor lógico a fin de indicar el éxito de la operación y, si corresponde, los valores resultantes de la operación.

V. PRUEBAS REALIZADAS

El protocolo propuesto fue evaluado en diferentes plataformas de hardware, entre ellas Arduino Uno R3 utilizando el IDE de Arduino, Espressif ESP32 construyendo la aplicación con Visual Studio Code y PlatformIO y la FPGA Tang Nano 4K desarrollando las aplicaciones con Gowin EDA y Gowin MCU Designer. En los dos primeros casos el objetivo fue controlar el estado del led presente en la placa y tomar una señal senoidal generada desde el sistema embebido.

El control del led permite observar la implementación de una operación simple sobre la placa y el procesamiento de su respuesta.

En lo que respecta al acceso a una señal generada desde el microcontrolador esta operación es significativa ya que el procesamiento se realiza utilizando valores de tipo real, obteniendo desde el sistema embebido el valor de la señal de interés y la marca temporal en la que se obtuvo.

Los valores reales, específicamente el tipo `float` en lenguaje C, es codificados a fin de lograr una representación aritmética de punto flotante. Los compiladores considerados utilizan la representación IEEE754 [10] y un espacio de almacenamiento de 32 bits. Al intercambiar valores de este tipo entre un microcontrolador y GNU Octave es necesario realizar conversiones desde ambos espacios de ejecución a fin de lograr interpretarlos adecuadamente.

Desde los microcontroladores los valores de tipo real fueron procesados utilizando la unión `singleParts` (Listado 2). De este modo al asignar un valor real al campo `value` es posible dividirlo en sus partes constitutivas a través del arreglo `parts`, y a la inversa. Resolviendo la conversión en ambos sentidos.

```
typedef union{
    float value;
    uint8_t parts[4];
}singleParts;
```

Listado 2: Definición de la estructura `singleParts`

En GNU Octave el procesamiento se realiza a través de una dos transformaciones, primero a valores reales de tipo `single` (reales de 32 bits con formato IEEE754) para luego realizar una conversión de tipo (`typecast`) a `uint32`. De este modo es posible manipular el valor codificado como si se tratase de un valor entero. Dependiendo del sentido de la comunicación es la manera en la que se realiza el proceso. En los listados 3 y 4 se presentan fragmentos de código con el proceso de conversión. En el primero se indica la construcción de un paquete utilizando un arreglo que serán enviado al microcontrolador conteniendo un valor de tipo real. El microcontrolador implementa una función para procesar esta información y reconstruir el valor utilizando la unión `singleParts`. En el segundo listado se procesan los datos recibidos desde un microcontrolador y almacenados en el vector `data`, a fin de conformar un valor real que pueda ser interpretado por GNU Octave.

```
vR = single(vR);
vR = typecast(vR, "uint32");
toSend = [ ...
    uint8(HEADER()), ...
    uint8(0x02), ... % Command number
    uint8(0x04), ... % Package length
    uint8(bitand(0x000000FF, bitshift(vR, 0))), ...
    uint8(bitand(0x000000FF, bitshift(vR, -8))), ...
    uint8(bitand(0x000000FF, bitshift(vR, -16))), ...
    uint8(bitand(0x000000FF, bitshift(vR, -24))), ...
    uint8(0x00), ... % CRC8
    uint8(FOOTER())];
for k=1:7
    toSend(8) = crc8(toSend(8), uint8(toSend(k)));
endfor
```

Listado 3: Construcción de un paquete a fin de enviar un valor real (`vR`) desde GNU Octave al microcontrolador.

```
vR = uint32(0);
vR = bitor(uint32(data(1)), vR);
vR = bitor(bitshift(uint32(data(2)), 8), vR);
vR = bitor(bitshift(uint32(data(3)), 16), vR);
vR = bitor(bitshift(uint32(data(4)), 24), vR);
vR = typecast(vR, "single");
```

Listado 4: Construcción de un valor real (vR) a partir de la información recibida desde el microcontrolador y almacenada en el vector `data`

La señal generada en los sistemas embebidos utilizados fue de la forma:

$$f(t) = A * \sin(2 * \pi * f * t)$$

Aquí la amplitud (A) y la frecuencia (f) son configurados desde GNU Octave enviándolos al microcontrolador. Como valor de tiempo se utilizó la cantidad de mili segundos transcurridos desde que se inició el microcontrolador.

Al realizar estas operaciones se verifica el correcto funcionamiento de la implementación en ambos extremos de la comunicación.

En la figura 1 puede observarse la aplicación script ejecutándose y presentando la señal recibida. En la figura 2 se presentan las opciones de configuración. Es interesante notar que esta aplicación permite observar la relación entre las frecuencias de muestreo y de la señal a la hora de representar una señal.

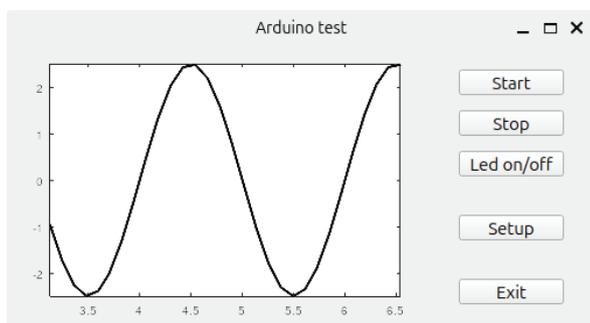


Figura 1: Aplicación gráfica representando la señal obtenida desde el microcontrolador

Las pruebas en la placa de desarrollo Tang Nano 4K se realizaron utilizando el microcontrolador ARM Cortex M3 embebido en la FPGA. A través de este microcontrolador se inicializó una cámara de video OV2640 incorporada en el kit, luego la señal de video es tratada a fin de generar una señal de video digital accesible a través del puerto HDMI.

Además de las funciones indicadas anteriormente, en este caso se agregaron dos opciones que permiten interactuar con la cámara de video, alternar entre la señal de video de la cámara y una señal de prueba así cómo invertir o no la imagen de video.

De este modo es posible controlar la cámara de video, a través del microcontrolador desde GNU Octave.

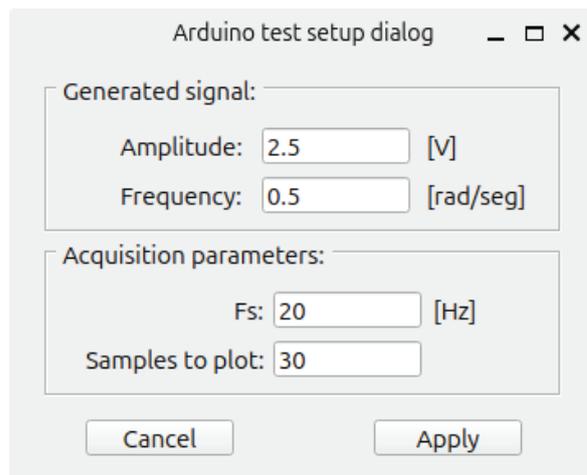


Figura 2: Configuración utilizada

	Arduino UNO	ESP32	Cortex M3
RAM	18 %	3.4 %	38.5 %
Flash	13 %	18.6 %	45.9 %

Tabla I: Consumo de memoria aproximado en cada sistema

VI. CONCLUSIONES

Desde el punto de vista del consumo de recursos, el protocolo propuesto utiliza poca cantidad de memoria RAM y FLASH (Véase tabla I). En el caso donde se da un mayor consumo de recursos porcentual es en la FPGA, que fue configurada con 32 KB de FLASH y 8 KB de RAM. De cualquier modo, los recursos libres permiten implementar algoritmos adicionales en cada dispositivo.

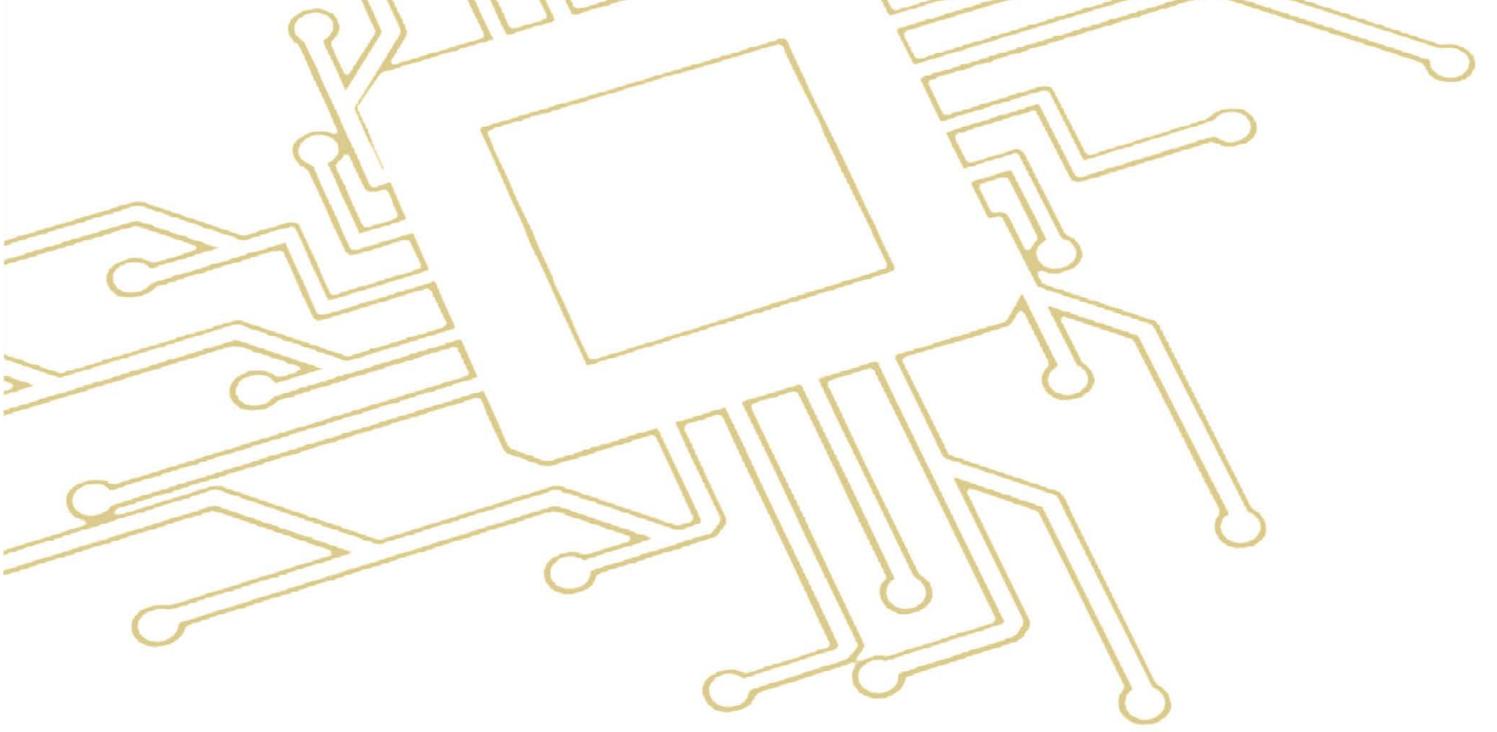
Las pruebas realizadas, cuyas implementaciones se encuentran disponibles en el repositorio del proyecto [11], han demostrado un funcionamiento correcto. La forma en la que se ha realizado la implementación permite agregar nuevas acciones sobre el sistema embebido sin requerir modificaciones en el procesamiento del protocolo, simplificando el desarrollo.

Se encontró que, en las aplicaciones utilizadas como caso de prueba, el tiempo de refresco de la gráfica hace que solo se pueda trabajar con bajas frecuencias a la hora de representar los datos. Igualmente, más allá de ese caso, las implementaciones posibilitan la comunicación, haciendo posibles compartir información entre el sistema embebido y GNU Octave. A modo de ejemplo, esto posibilitaría obtener mediciones de un sistema, identificarlo, diseñar y simular un controlador, para finalmente transmitir los parámetros del mismo nuevamente hacia el sistema embebido a fin de comprobar su respuesta. Ideas similares pueden aplicarse al diseño, simulación e implementación de filtros digitales.



REFERENCIAS

- [1] J. W. Eaton. (2024) Gnu octave website. [Online]. Disponible en: <https://octave.org/>
- [2] E. S. Burgos y E. J. Adam, "Graphical user interface editor for octave applications," *Engineering Reports*, vol. 2, no. 11, p. e12269, 2020.
- [3] E. J. Adam y E. S. Burgos, "Improvements implemented in guieditor and its use in teaching of process control systems," *LACCEI*, vol. 1, no. 8, 2023.
- [4] J. Guzmán, F. García-Mañas, Á. Hoyo, J. Ramos-Teodoro, y J. Donaire, "Use of tclab kits for control engineering curricula at the university of almería," *IFAC-PapersOnLine*, vol. 55, no. 17, pp. 362–367, 2022.
- [5] P. de Moura Oliveira, J. D. Hedengren, y J. Rossiter, "Introducing digital controllers to undergraduate students using the tclab arduino kit," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 17 524–17 529, 2020, 21st IFAC World Congress. [Online]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S2405896320334224>
- [6] L. G. de Souza, D. R. S. da Costa, N. F. de Andrade Esmeraldo, y P. C. L. da Silva, "O uso do arduino para o estudo de circuitos do tipo rc," *Conexões-Ciência e Tecnologia*, vol. 15, pp. e021 009–e021 009, 2021.
- [7] J. W. Eaton, "Gnu octave and reproducible research," *Journal of Process Control*, vol. 22, no. 8, pp. 1433–1438, 2012.
- [8] J. D. Andrius Sutas, Stefan Mahr. (2024) instrument-control website. [Online]. Disponible en: <https://gnu-octave.github.io/packages/instrument-control/>
- [9] E. S. Burgos. (2024) guieditor website. [Online]. Disponible en: <https://guieditor.net.ar/>
- [10] "ieee standard for floating-point arithmetic," *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pp. 1–84, 2019.
- [11] E. S. Burgos. (2024) Repositorio del proyecto. [Online]. Disponible en: <https://gitlab.com/labinformatica/octavecommunication>



Eje Temático

Aplicaciones de uso Hogareño



Uso de ESP32-CAM para seguridad y vigilancia controlado por Telegram

Julio Dario Solis Leyes, Roberto M. Murdocca, Sergio F. Hernandez Velazquez
Departamento de Electrónica
Universidad Nacional de San Luis
San Luis, Argentina
e-mail: {leyessolis, mmurdocc, sergio.sfhv}@gmail.com

Resumen - En este trabajo se presenta la implementación de un sistema de seguridad y vigilancia controlado remotamente mediante la plataforma Telegram. El sistema permite gestionar funciones como la captura y envío de imágenes ante la detección de movimiento, la rotación de la cámara, el encendido y apagado de luces y la activación de una sirena utilizando un ESP32-CAM.

Palabras clave: Seguridad y Vigilancia, Telegram, MEF, ESP32-CAM.

I. INTRODUCCIÓN

En un mundo cada vez más digitalizado, la seguridad es una preocupación clave para hogares y negocios. Los métodos tradicionales de vigilancia tienen limitaciones en cuanto a accesibilidad remota y control en tiempo real. Si bien existen sistemas de monitoreo remoto para seguridad y vigilancia modernos con muchas prestaciones, el costo es excesivo para muchos usuarios. Por este motivo se propone el diseño de un sistema modesto, de bajo costo y con posibilidades de ampliaciones futuras.

La demanda de soluciones de seguridad accesibles y avanzadas ha impulsado el desarrollo de sistemas que combinan hardware eficiente con plataformas de comunicación modernas. Los avances tecnológicos en comunicación inalámbrica, sensores inteligentes y procesamiento de datos han permitido la creación de sistemas integrados que operan en tiempo real, ofreciendo vigilancia dinámica y respuestas automáticas a eventos, como la detección de movimiento. Asimismo, la popularización de plataformas de comunicación como Telegram ha abierto nuevas posibilidades en el control remoto de sistemas de seguridad.

Utilizar aplicaciones móviles para gestionar dispositivos conectados ha demostrado ser un enfoque eficiente y conveniente, alineándose con la tendencia hacia la domótica y el Internet de las Cosas (IoT). Estas herramientas ofrecen una interfaz básica e intuitiva, que permite a los usuarios monitorear y controlar sus sistemas de seguridad desde cualquier parte del mundo, reforzando la sensación de control y seguridad.

Este proyecto aborda esa necesidad mediante el desarrollo de un sistema de seguridad y vigilancia remoto, controlado íntegramente a través de la popular plataforma de mensajería Telegram. La solución presentada se enfoca en ofrecer a los usuarios un control total sobre la seguridad de sus entornos, permitiéndoles gestionar de manera remota diversas funcionalidades como el encendido y apagado de luces, la captura de imágenes en tiempo real, y la rotación precisa de una cámara de vigilancia. Además, se incorporan dos modos de operación: el modo PIR, que utiliza un sensor de infrarrojos pasivo para la detección de movimiento y la captura inmediata de imágenes, y el modo alarma, que refuerza la seguridad activando una sirena y luces intermitentes.

II. DESCRIPCIÓN

El sistema propuesto está conformado por una placa de desarrollo ESP32-CAM, un motor paso a paso para posicionar la cámara, un sensor PIR para la detección de movimiento, una sirena, un módulo de luces LED, un módulo de potencia con relé, un modulo DC-DC Step Down y demás componentes auxiliares. En la Fig. 1 podemos ver un diagrama en bloques del sistema y en la Fig. 2 se muestra el circuito completo del sistema.

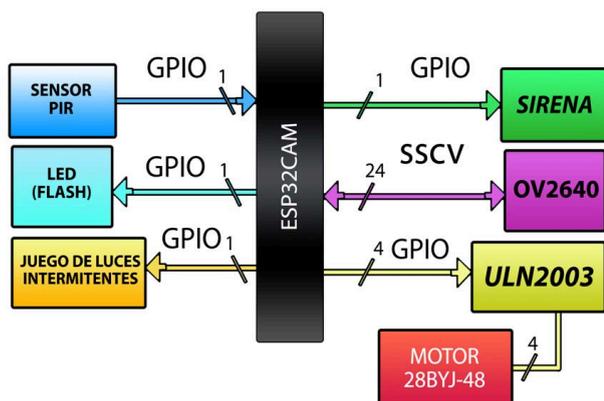


Fig. 1. Diagrama en bloques del sistema

A. HARDWARE

El núcleo del sistema está formado por una placa ESP32-CAM la cual consiste en una placa de desarrollo compacta que integra un microcontrolador ESP32 doble núcleo con Wi-Fi y Bluetooth, una cámara OV2640 capaz de capturar imágenes de alta resolución. Cuenta además, con una ranura para tarjeta microSD, 9 pines GPIO disponibles, y es programable con Arduino IDE y otras plataformas.

Para el movimiento de la cámara se utiliza un motor paso a paso modelos 28BYJ-48 y como driver para este último un ULN2003. La sirena se excita desde un módulo rele. Las luces se implementaron con un módulo LED. Todo el sistema se energiza con 12V y se utiliza un convertidor DC-DC Step Down para regular los 5V para la ESP32, motor paso a paso y sensor PIR. Todo este conjunto se montó en una caja impresa en 3D para este trabajo. En la Fig. 3 podemos ver el housing diseñado para albergar todo el hardware.

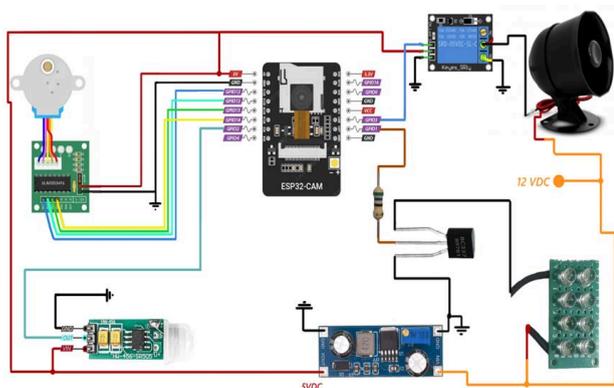


Fig. 2. Diagrama en bloques del sistema

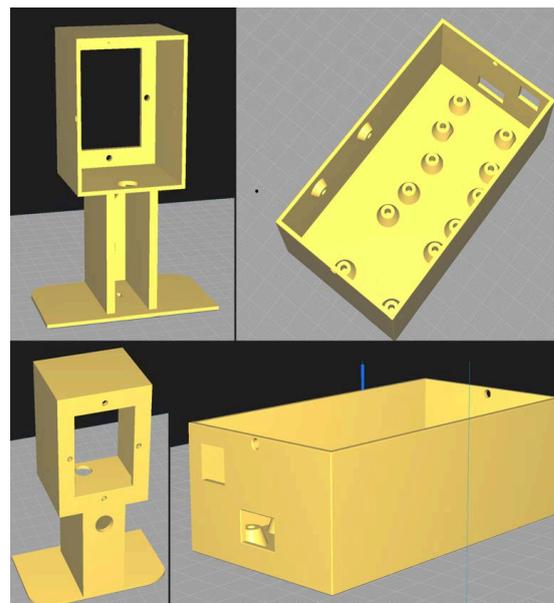


Fig. 3. Housing del proyecto.

B. SOFTWARE

La gestión del software del sistema se modela mediante una Máquina de Estados Finitos (MEF). La MEF detalla cómo se gestionan las tareas como la captura de imágenes, la rotación de la cámara y el cambio de modos de funcionamiento en respuesta a las condiciones y comandos recibidos. En la Fig. 4 podemos ver dicha MEF.

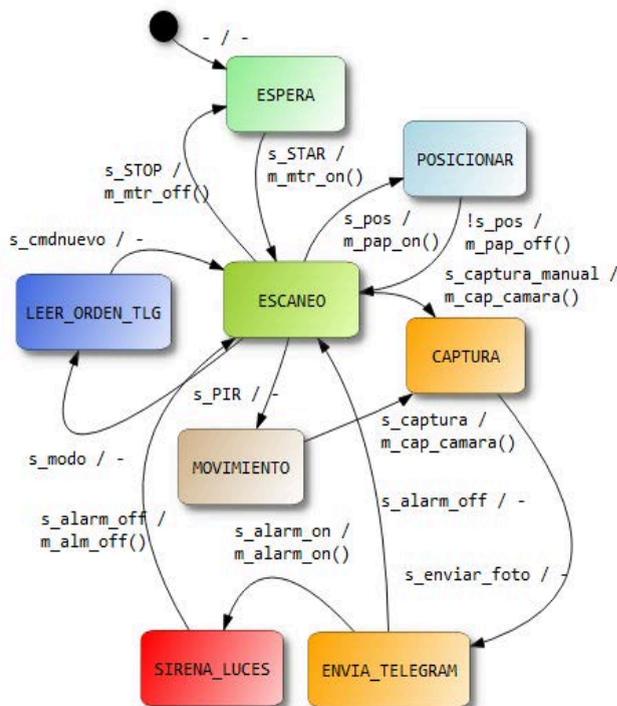


Fig. 4. Máquina de estados finitos (MEF)



En la MEF tenemos los siguientes estados:

- **Escaneo:** El estado principal donde el sistema monitorea continuamente en busca de eventos.
- **Captura:** Cuando se solicita una foto, el sistema toma la captura.
- **Posicionar:** En este estado se ajusta la posición de la cámara y luego vuelve a **Escaneo**.
- **Movimiento:** El sistema detecta movimiento con el sensor PIR y realiza la captura de imagen.
- **Sirena y Luces:** En este estado se activan la sirena y el módulo LED para alertar sobre amenazas detectadas.

III. FUNCIONAMIENTO

Inicialmente, las pruebas de funcionamiento del sistema se centraron en evaluar la eficiencia de la captura de imágenes y su posterior envío a través del servidor de Telegram. Se pudieron identificar y corregir problemas, como los retrasos en la transmisión de imágenes y los envíos erróneos que ocurrían al encender el sistema.

El sistema posee un modo de trabajo denominado PIR. En este modo se combina la funcionalidad de captura de imágenes con la lógica de detección basada en un sensor de infrarrojos pasivo (PIR). Las pruebas en este contexto se orientaron a validar que las imágenes solo se capturaron y enviaron cuando el modo PIR estuviera activado por el usuario y el sensor detectara movimiento de manera simultánea.

Para ello, se energizó el sistema con el modo PIR activo y se activó el sensor PIR en intervalos controlados mediante la detección de presencia, garantizando así la correcta sincronización entre la detección de movimiento y el envío de imágenes.

Para el posicionamiento de la cámara se utilizó un motor paso a paso 28BYJ-48, gestionado a través del controlador ULN2003. Se probó al principio un servomotor ya que se disponía de uno, pero él mismo hacía bastante ruido y esto podría alertar a los intrusos.

Se tuvo que tener en cuenta los límites de giro del motor y se realizó el almacenamiento en memoria no volátil de la última posición del motor para que en caso de corte de energía, al

retomar la misma el motor retome su posición previa, evitando errores derivados de la suposición de una posición inicial neutral o incorrecta, que podrían comprometer la precisión del control de los límites de giro.

La capacidad del sistema para identificar y mantener la posición de giro del motor es esencial para prevenir daños físicos, como la tensión excesiva o el corte de los cables que conectan la placa de desarrollo situada en la carcasa superior con los módulos alojados en la carcasa inferior. Este control preciso asegura la integridad de las conexiones y el correcto funcionamiento del sistema en su conjunto, garantizando la fiabilidad y la longevidad del equipo.

El sistema requiere la conexión con un servidor de Telegram, enlazado con un bot que utiliza encriptación para garantizar la máxima seguridad. Solo los usuarios autorizados, que posean el token correspondiente, tendrán acceso al sistema de seguridad. Una vez que tanto el sistema como el dispositivo del usuario estén conectados a internet, podrán interactuar de manera remota. El usuario tendrá acceso a una mini guía de uso y podrá enviar comandos al sistema desde cualquier lugar, asegurando un control eficiente y seguro.

IV. RESULTADOS

La implementación del sistema se llevó a cabo en una caja diseñada para este fin a través de impresión 3D como se muestra en la Fig. 6. La impresión fue realizada en dos partes. En la parte inferior de esta se montaron los circuitos de alimentación y el driver del motor. En la parte superior se colocó el ESP32, la cámara, el motor PAP y el sensor PIR. Dado que el microcontrolador se encuentra en la parte superior de la carcasa, fue necesario que todos los módulos ubicados en el gabinete inferior estén conectados a este a través de cables de longitud adecuada. Esto es crucial para garantizar que dichos cables no estén sometidos a tensiones mecánicas excesivas durante la rotación de hasta 180 grados que puede efectuar el sistema, evitando así posibles fallas debido al desgaste o desconexión accidental.

En la Fig. 6 se puede ver una foto del prototipo en su versión final. Las pruebas realizadas en este primer prototipo estuvieron a

la altura de los objetivos planteados al principio del trabajo. A modo de ejemplo en la Fig. 7 podemos ver una captura enviada a Telegram ante la detección de intrusos. Además el usuario puede interactuar con el sistema a través de un menú implementado en Telegram mediante un bot. Este bot posee comandos que permiten hacer capturas, encender la sirena, encender las luces y posicionar la cámara.

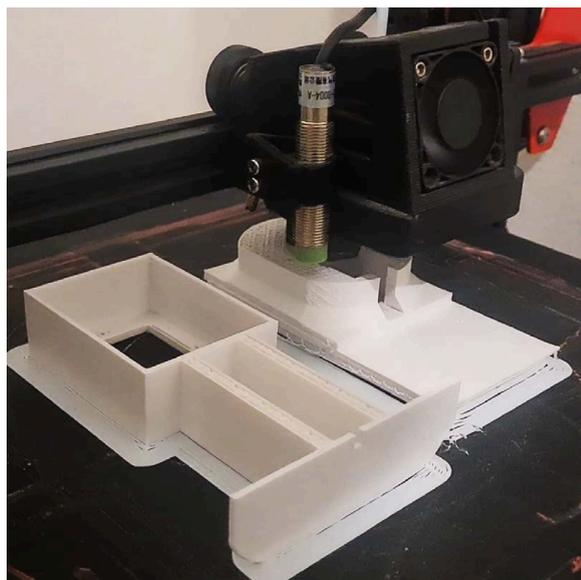


Fig. 5: Detalle de la carcasa.



Fig. 6 Dispositivo en su versión final.



Fig. 7: Detección de presencia.

V. CONCLUSIONES

En este proyecto se desarrolló con éxito un sistema integral de seguridad de bajo costo, ofreciendo un rendimiento sólido en términos de vigilancia y seguridad remota. Se logró implementar la detección de movimiento y la captura de imágenes para su envío a través de Telegram. Además, se integró un bot que permite al usuario interactuar con el sistema mediante un menú funcional de opciones.

Los objetivos propuestos al inicio del proyecto fueron alcanzados satisfactoriamente, resolviendo la problemática de seguridad planteada. Durante el proceso, se evidenció el potencial de la ESP32-CAM para realizar capturas y envíos de imágenes, lo cual sugiere nuevas posibilidades para explorar en futuras aplicaciones.

Entre las modificaciones futuras se considera el agregado de un sensor ranurado u óptico para tener una referencia de origen en caso de pérdida de pasos. Además, se prevé investigar sobre la incorporación de un pequeño algoritmo de inteligencia artificial para el seguimiento automático del objeto detectado. También se están realizando pruebas para habilitar la transmisión de video en tiempo real hacia un servidor, con el objetivo de grabar secuencias en caso de detección de presencia no deseada.



REFERENCIAS

- [1] Espressif Systems. (2024). *ESP32 datasheet*. Espressif Systems. Obtenido de https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [2] Manuals+. (s. f.). *ESP32-CAM Module Manual*. Manuals+. Recuperado el 15 de julio de 2024, de <https://manuals.plus/es/electronic-hub/esp32-cam-module-manual>
- [3] STMicroelectronics. (s. f.). *ULN2001–ULN2004: High voltage, high current Darlington transistor arrays [Datasheet]*. STMicroelectronics. Recuperado el 8 de mayo de 2024, de <https://www.st.com/resource/en/datasheet/uln2001.pdf>
- [4] Arduino. (s. f.). *Stepper* [Documentación de la biblioteca]. Arduino. Recuperado el 8 de mayo de 2024, de <https://www.arduino.cc/reference/en/libraries/stepper/stepper/>
- [5] Rapid Electronics. (s. f.). *28BYJ-48 Stepper Motor Datasheet*. Rapid Electronics. Recuperado el 8 de mayo de 2024, de https://static.rapidonline.com/pdf/78-4110_v1.pdf
- [6] SendPulse. (s. f.). *Cómo crear un chatbot para Telegram*. SendPulse. Recuperado el 2 de agosto de 2024, de <https://sendpulse.com/latam/knowledge-base/chatbot/telegram/create-telegram-chatbot>
- [7] Programar Fácil. (s. f.). *ESP32-CAM: Guía completa de uso y programación*. Programar Fácil. Recuperado el 8 de mayo de 2024, de <https://programarfácil.com/esp32/esp32-cam/>
- [8] Witnessmenow. (s. f.). *Universal Arduino Telegram Bot* [Repositorio en GitHub]. GitHub. Recuperado el 2 de agosto de 2024, de <https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot/tree/master>



Uso de un Chat Bot de Telegram como plataforma de IOT

Luis Miguel Echevarría Rodríguez

Universidad Tecnológica de La Habana José Antonio Echeverría
(CUJAE), Facultad de Ingeniería Automática y Biomédica

La Habana, Cuba

luismiguelechevarriarodriguez@gmail.com

Resumen—El uso del internet y el desarrollo del Internet de las Cosas (IoT) crece a paso acelerado todos los años, una muestra de ese desarrollo es la incorporación, perfeccionamiento y desarrollo de múltiples plataformas y soluciones para el uso del IoT. Este proyecto es un sistema integrado donde se combina un Bot de Telegram, un microcontrolador ESP32, un sensor DHT11 de temperatura y humedad, y actuadores para ofrecer un enfoque versátil en entornos IoT. El sistema proporciona monitorización remota de la temperatura y humedad, alertas personalizadas y automatización de respuestas, contribuyendo a la prevención de situaciones de riesgo y la respuesta eficiente ante distintas situaciones. En cuanto al desarrollo, la facilidad de integración y uso, el rápido prototipado y la escalabilidad del sistema ofrecen una plataforma robusta para el desarrollo ágil de aplicaciones IoT. En conjunto, este sistema fusiona la conectividad inteligente, la eficacia operativa y la innovación tecnológica para crear soluciones seguras y escalables en entornos domésticos y comerciales, destacando un enfoque integral y creativo para abordar desafíos en el desarrollo y en el contexto de la IoT.

Palabras clave: : Internet de las Cosas, Bot de Telegram, aplicaciones IoT, monitorización remota, microcontrolador ESP32.

I. INTRODUCCIÓN

En un mundo cada vez más interconectado y digitalizado, la integración de la Internet de las Cosas (IoT) en nuestras vidas cotidianas ha cobrado una relevancia sin precedentes. En

este contexto de innovación tecnológica, se ha demostrado de manera fascinante cómo la plataforma de mensajería Telegram puede convertirse en un vehículo efectivo para el control y monitoreo de dispositivos IoT.

En este proyecto se utilizó un Bot de Telegram en combinación con un microcontrolador ESP32 y un sensor de temperatura y humedad DHT11, junto con componentes electrónicos fácilmente accesibles, para ilustrar la viabilidad de Telegram como plataforma IoT. Lo que es aún más notable es que, además de la sofisticada infraestructura tecnológica, la comunicación y operación de este sistema innovador se llevó a cabo de manera sencilla y eficiente a través de un dispositivo común: un teléfono móvil.

En la experimentación, se demostró la capacidad de actuar conscientemente sobre actuadores específicos, en este caso, un motor y un deshumidificador, utilizando el Bot de Telegram como intermediario comunicativo. Estos actuadores, seleccionados estratégicamente como ejemplos representativos, destacan la versatilidad y potencialidad de la plataforma Telegram como un canal efectivo para implementar soluciones de IoT en entornos domésticos y aplicativos. Este proyecto no solo abre nuevas posibilidades en el campo del Internet de las Cosas, ni en el uso de la plataforma de mensajería de Telegram, sino que también promueve una visión creativa e integradora de la tecnología en nuestra vida diaria. La interacción fluida entre humanos, dispositivos y la inteligencia artificial encarnada en un Bot de

Telegram, revela un futuro prometedor donde la innovación y la accesibilidad convergen para transformar nuestra manera de habitar y relacionarnos con el entorno tecnológico que nos rodea.

Se busca destacar en esta iniciativa enfatizar la importancia de explorar y aprovechar al máximo las herramientas tecnológicas disponibles, con un enfoque creativo, preciso y efectivo. A través de la unión entre Telegram, la IoT y la operatoria consciente sobre dispositivos domésticos, se revela un horizonte fascinante de posibilidades que invitan a imaginar un futuro donde la conectividad y la inteligencia se fusionan para mejorar nuestra calidad de vida y experiencia tecnológica. Para lograr este objetivo se tuvo que tener en cuenta que se necesitaban varias configuraciones importantes como la confección de un Bot de Telegram, la programación del microcontrolador, la selección y calibración de los sensores y no por último es el menos importante, la explotación del sistema.

Uno de los objetivos clave de este sistema es mejorar la eficiencia energética al permitir un control preciso y programado de los actuadores, lo que puede resultar en un consumo energético más inteligente y sostenible. Brindar conveniencia y comodidad al usuario al posibilitar el control remoto de los dispositivos, facilitando la automatización de tareas. Proporcionar la capacidad de monitoreo en tiempo real y personalizado brindando información actualizada sobre el estado de los actuadores y asegurar una supervisión efectiva y proactiva. El sistema busca fomentar la innovación tecnológica al explorar nuevas formas de integrar plataformas de mensajería como Telegram con dispositivos IoT accesibles, demostrando nuevas posibilidades de interacción intuitiva y fácil para usuarios de diferentes niveles de experiencia tecnológica.

Si vemos este sistema como un sistema de control tendría las siguientes características:

- Interacción bidireccional
- Comandos personalizados
- Retroalimentación en tiempo real
- Automatización programada
- Notificaciones personalizadas
- Acceso remoto
- Fácil uso

MATERIALES Y MÉTODOS

Diseño del Bot de Telegram

Es muy importante el diseño de este Bot ya que con el interactuaremos con nuestros dispositivos, a continuación, se enumeran los pasos a seguir para crearlo.

1. Abre Telegram y busca a BotFather:

- Abre la aplicación Telegram en tu teléfono móvil.
- En la barra de búsqueda, escribe "BotFather" y selecciona el perfil de BotFather, la siguiente figura se observa el perfil a seleccionar.

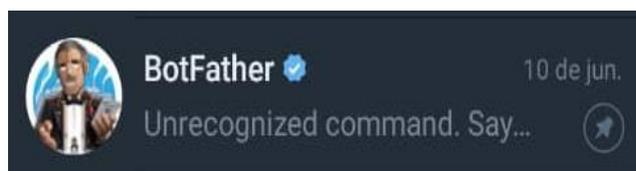


Fig.1 Perfil de BotFather

1. Inicia la conversación con BotFather:

- Haz clic en "Iniciar" para comenzar una conversación con BotFather.

2. Crea un nuevo Bot:

- Escribe el comando /newbot para crear un nuevo Bot.
- Sigue las instrucciones y proporciona un nombre único para tu Bot.
- Después, asigna un nombre de usuario para tu Bot que termine en "Bot" (por ejemplo, MiBotEjemplo_bot).

3. Obtén el token de acceso:

- Una vez creado con éxito, BotFather te proporcionará un mensaje con el token de acceso que necesitarás para interactuar con tu Bot a través de la API de Telegram. Guarda este token de forma segura y no lo compartas con nadie, ya lo utilizaremos en la programación y configuración de nuestro sistema microcontrolado.

En la siguiente figura se observa la interacción con BotFather para la configuración del Bot.

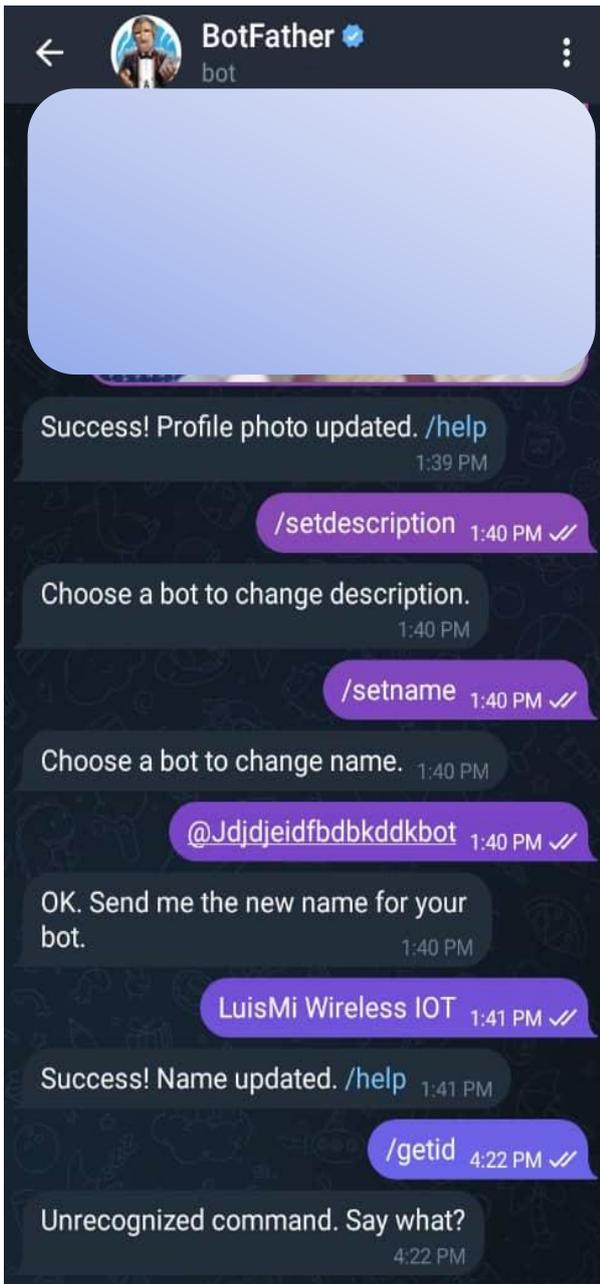


Fig.2 Interacción con BotFather para el diseño del Bot y configuración del mismo

¿Pero? Que tengamos el Bot creado y personalizado no quiere decir que estamos en condiciones de usarlo, aun nos falta un paso importante.

Obtención de nuestra identidad de chat

Telegram nos brinda una identidad de chat única para poder identificarnos, es como si nos estuviese asignando un número de teléfono, pero este número es solo para interactuar directamente con sus servidores. A

continuación, se muestran una serie de pasos a seguir para obtener nuestro Chat ID.

1. Abre Telegram y busca a IDBot:
 - Abre la aplicación Telegram en tu teléfono móvil.
 - En la barra de búsqueda, escribe "IDBot" y selecciona el perfil de IDBot, la siguiente figura se observa el perfil a seleccionar.

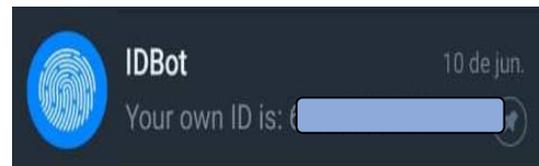


Fig.3 Perfil de IDBot

1. Inicia la conversación con IDBot:

- Haz clic en "/getid" para comenzar una conversación con IDBot y obtendrás tu identidad como usuario activo de los servicios de Telegram, en la siguiente figura se muestra la interacción con IDBot.



Fig.4 Interacción con IDBot para obtener el Chat ID

El chat ID es importante, al igual que el Token, que no lo compartamos con nadie más ya que

es un identificador único asignado a cada conversación en Telegram que permite al Bot enviar mensajes de manera precisa y segura a usuarios específicos. Este identificador asegura la identificación correcta del destinatario, controla el envío de mensajes, garantiza la seguridad y privacidad de las comunicaciones, personaliza la interacción con los usuarios y facilita la gestión efectiva de conversaciones grupales, siendo esencial para una operación eficiente y personalizada del Bot en la plataforma de Telegram.

Componentes principales

Microcontrolador ESP32 WROOM 32

- Conectividad Wi-Fi de doble banda (2.4 GHz y 5 GHz) y Bluetooth integrado, que permite la comunicación inalámbrica con dispositivos y redes.[1]
- Soporta BLE (Bluetooth de Baja Energía), ideal para aplicaciones de bajo consumo energético y comunicación eficiente con dispositivos cercanos.[1]
- Capacidades de red avanzadas que le permiten conectarse a Internet, enviar y recibir datos, y comunicarse con otros dispositivos a través de diferentes protocolos de red.[1]
- Cuenta con dos núcleos de procesamiento (Dual-Core) que pueden operar de manera simultánea o en paralelo, lo que mejora el rendimiento y la capacidad de ejecutar múltiples tareas de forma eficiente.[1]
- Puede alcanzar velocidades de reloj de hasta 240 MHz, lo que proporciona un alto rendimiento para aplicaciones que requieren un procesamiento rápido de datos.[1]

En la siguiente figura se observa el microcontrolador utilizado.



Fig.5 Microcontrolador ESP32-WROOM-32

1. Sensor Capacitivo de Temperatura y Humedad

- Tiene un tiempo de respuesta relativamente lento en comparación con otros sensores más avanzados, típicamente oscilando entre 1 y 2 segundos. Esto significa que la lectura de la temperatura y humedad puede tener un ligero retardo.[2]
- Rango de medición de temperatura suele ser de 0°C a 50°C (32°F a 122°F), lo que lo hace adecuado para aplicaciones en entornos de temperatura ambiente con una precisión de +- 1°. [2]
- Rango abarca del 20% al 95% en la medición de la humedad, lo que permite monitorear la humedad relativa en espacios interiores de manera efectiva con una precisión de +- 5%. [2]

En la siguiente figura se muestra el sensor utilizado.

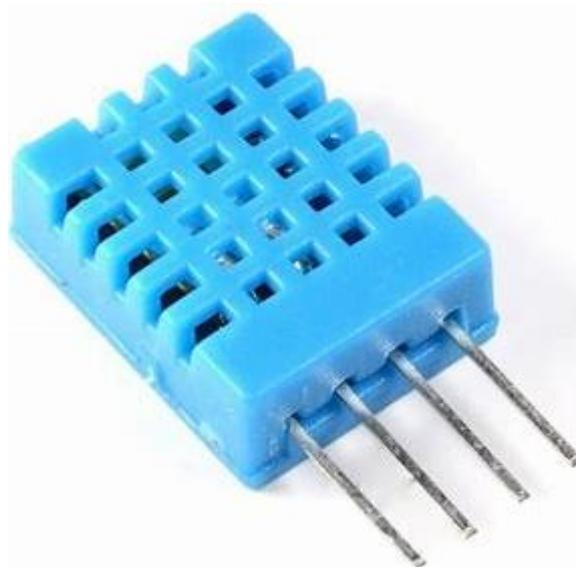


Fig.6 Sensor capacitivo de temperatura y humedad DHT11

1. Led Indicador

Usamos leds indicadores para mostrar cuando el sistema está en funcionamiento y para ilustrar cuando se activan los determinados actuadores.

Principio de funcionamiento

Adentrándonos en el funcionamiento general del sistema, una vez obtenidos el Token y nuestro Chat ID tenemos que tener en cuenta que necesitamos internet vía Wi-Fi ya que nuestro microcontrolador posee este medio inalámbrico de comunicación incorporado. Esta red será proporcionada por un Punto de Acceso creado desde nuestro teléfono móvil o desde un Router Wi-Fi. La red a utilizar tendrá que estar certificada en nuestro código con el nombre de la red (SSID) y la contraseña a utilizar la cual es obligatoria (PASS).

El sistema contará con dos modos de trabajo, un modo automático que se estará ejecutando solamente en el microcontrolador según la lógica programada en él, y el modo a distancia que será desde nuestro dispositivo móvil interactuando con el Chat Bot. Estos modos de trabajo no son elegibles por el usuario ya que dependerán de la conectividad a internet que presente la red Wi-Fi utilizada.

Una vez seleccionado el modo de trabajo por medio de la conectividad de la red Wi-Fi a utilizar entra en juego la lógica de programación descrita por cada método. A continuación, se describe como se ejecutaría el modo de trabajo automático.

El microcontrolador estará recibiendo de manera constante el valor de la temperatura y humedad censados por el DHT11 y con esos valores, a gusto del usuario y según las condiciones que se deseen, se ejecutarán las acciones pertinentes que pueden ser por ejemplo: Si la temperatura alcanza un umbral no seguro o deseado se puede actuar sobre distintos actuadores para lograr la condición deseada, al igual con la variable humedad en caso de que haga falta actuar de manera consciente sobre ella, puede ser actuando sobre un deshumidificador o en ambos casos sobre una alarma.

Si en este caso se selecciona una red con acceso a internet entraría en juego lo interesante e innovador de este proyecto que sería la interacción con el Chat Bot de Telegram. Cuando estamos en presencia de una red con conectividad a internet, el Bot automáticamente nos envía un mensaje de que toda la conexión se ha establecido con éxito y que todo está listo para empezar a usarse, en la siguiente figura se muestra como ocurre este proceso y como se interactúa con el Bot.

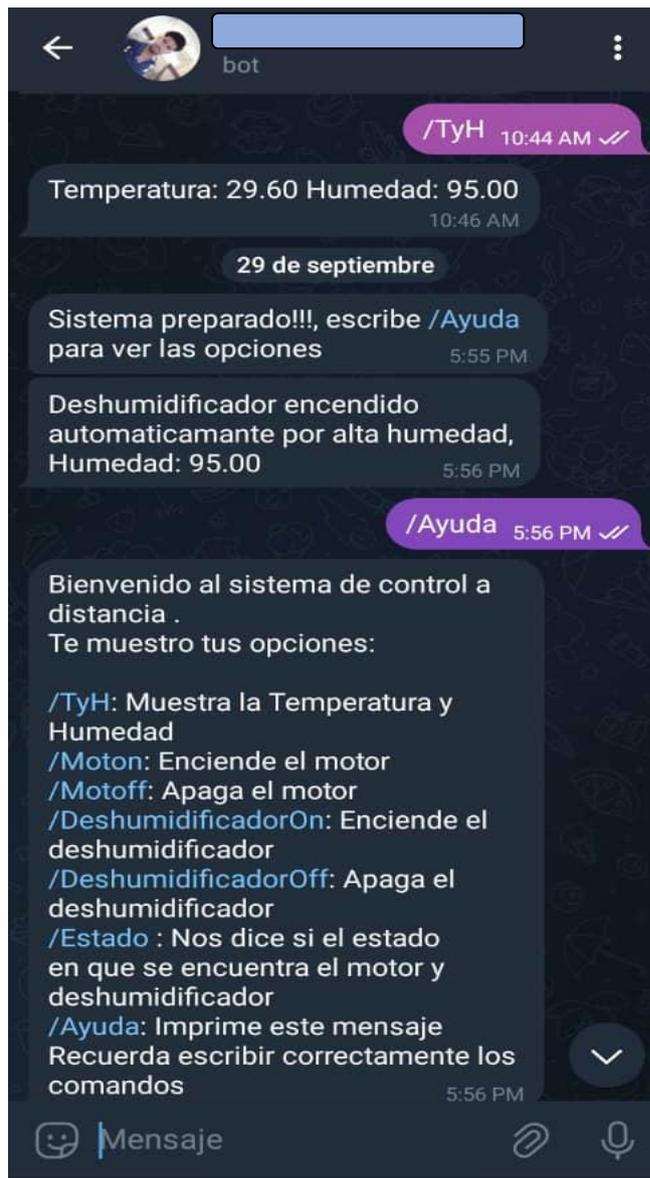


Fig.7 Mensaje que nos muestra el Bot al iniciarse.

La lista de comandos preestablecidos por el usuario en la programación y configuración de este microcontrolador y Bot se muestran en un mensaje al solicitar la ayuda del mismo. A continuación, se explica brevemente que hace cada comando.

- `/TyH`: Muestra el valor de la temperatura y humedad en la que se encuentra el local o donde se encuentre instalado el sensor DHT11.
- `/Moton`: Activa el motor a distancia con un nivel de prioridad alto.
- `/Motoff`: Desactiva el motor a distancia con un nivel de prioridad alto.

- /DeshumidificadorOn: Activa el deshumidificador a distancia con un nivel de prioridad alto.
- /DeshumidificadorOff: Desactiva el deshumidificador a distancia con un nivel de prioridad alto.
- /Estado: Nos dice el estado en que se encuentran cada uno de los actuadores por si desconocemos su estado de operación o/y activación.

Cuando nos referimos a un nivel de prioridad alto nos referimos a que, si se tiene una situación en que, según las condiciones programadas, por ejemplo, se activó automáticamente el motor por alta temperatura y no queremos que se active en ese momento, podemos escribir el comando /Motoff y se desactivará inmediatamente, es decir, el modo de funcionamiento a distancia tiene prioridad alta con respecto a la lógica de programación fija que se le fue programada al microcontrolador. En la siguiente figura se muestra como funcionaria la interacción con estos comandos en el Bot.

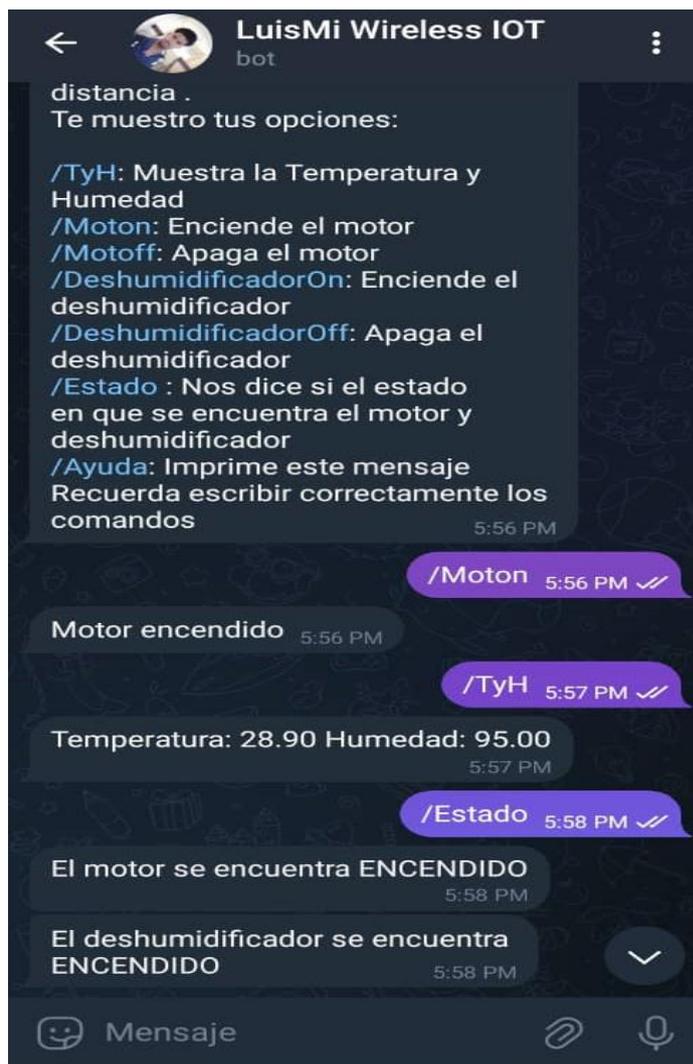


Fig.8 Interacción con el Bot en algunos comandos.

Si después de interactuar con nuestro móvil, Telegram y el Bot en nuestro sistema, podemos desconectarlo sin ningún problema de la red de internet, lo que ocurriría es que entraría en modo de funcionamiento automático. Y si queremos volver a actuar sobre él a distancia podemos sin ningún problema volver a conectarlo a internet y actual sobre él.

Tener en cuenta que para mejorar el tiempo de respuesta una vez dentro del modo de trabajo a distancia debemos percatarnos de tener una conexión a internet razonable para así evitar retardos incómodos.

Para acoplar el sistema a una carga se utilizó el siguiente circuito mostrado en la figura siguiente (Fig.9) ya que la salida del sistema a través de los pines de entrada y salida de propósito general (GPIO) del microcontrolador proporcionan una salida máxima de 2.64 V con

28 mA [1], por lo que se necesita un circuito para acoplar esas dos diferencias de carga y corriente [3].

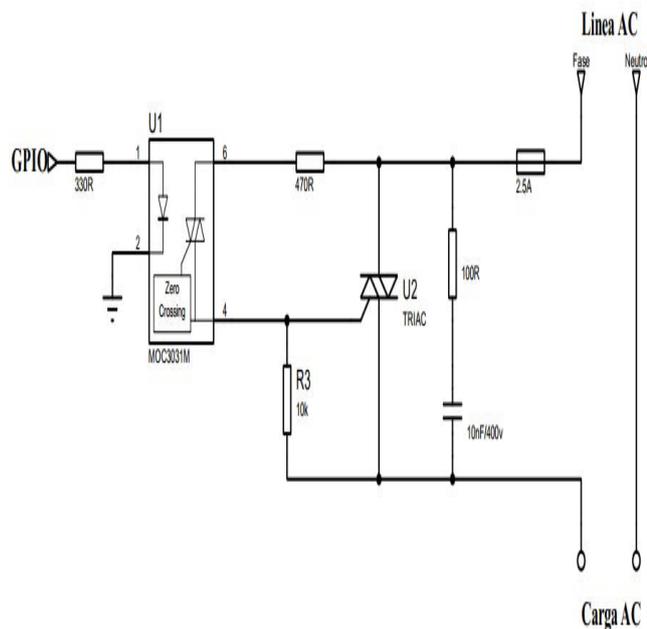


Fig.9 Circuito de acople.

RESULTADOS Y DISCUSIÓN

Los resultados obtenidos confirman el cumplimiento exitoso de los objetivos planteados, donde la monitorización precisa de la temperatura y humedad, la emisión oportuna de alertas personalizadas y la correcta activación de actuadores como el motor y deshumidificador han demostrado la plena operatividad y utilidad del sistema en entornos reales. La integración fluida entre los componentes dígame teléfono móvil con el Bot de Telegram, el hardware del sistema, etc. La respuesta inmediata a condiciones ambientales cambiantes y la capacidad de control remoto a través de Telegram son evidencia clara de la funcionalidad y eficacia de la solución implementada. Los resultados prácticos y aplicables obtenidos confirman el éxito del sistema en su implementación, destacando su potencial para mejorar la seguridad, eficiencia y comodidad en entornos IoT de manera efectiva y confiable donde el uso del móvil no sea una cuestión problemática.

En la siguiente tabla (Tabla 1) se muestra la cantidad de componentes a utilizar por tipo, en

caso de una implementación real del sistema, los componentes son genéricos y pueden ser reutilizados de cualquier equipo electrónico ya se trata de componentes electrónicos comunes en cualquier equipo, en esta lista hemos excluido el microcontrolador ESP32-WROOM32 y el sensor DHT11.

Tabla 1. Componentes a utilizar

Componente	Cantidad
Optotriac MOC30XX	1
Triac	1
Resistor 330Ω ¼ W	1
Resistor 470Ω ¼ W	1
Resistor 10KΩ ¼ W	1
Resistor 100Ω ¼ W	1
Capacitor 10nF 400V	1
Fusible 2.5 A	1

CONCLUSIONES

La integración de la Internet de las Cosas (IoT) con la plataforma de mensajería Telegram ha demostrado ser viable y efectiva, utilizando un Bot de Telegram junto con un microcontrolador ESP32 y un sensor de temperatura y humedad. A pesar de la sofisticación tecnológica involucrada, la comunicación y operación del sistema se realizan de manera sencilla a través de dispositivos comunes como teléfonos móviles. Esta integración resalta la versatilidad de Telegram como canal para implementar soluciones de IoT, permitiendo el control de actuadores y destacando el potencial para mejorar la eficiencia energética y brindar comodidad al usuario, mientras fomenta la innovación tecnológica en la interacción entre humanos y dispositivos. Además, para mejorar aún más la experiencia y abrir nuevas posibilidades para el futuro, se persigue perfeccionar el sistema utilizando la API de Google Assistant para permitir el control de los actuadores a través de comandos de voz, tan solo con decir "Ok Google".

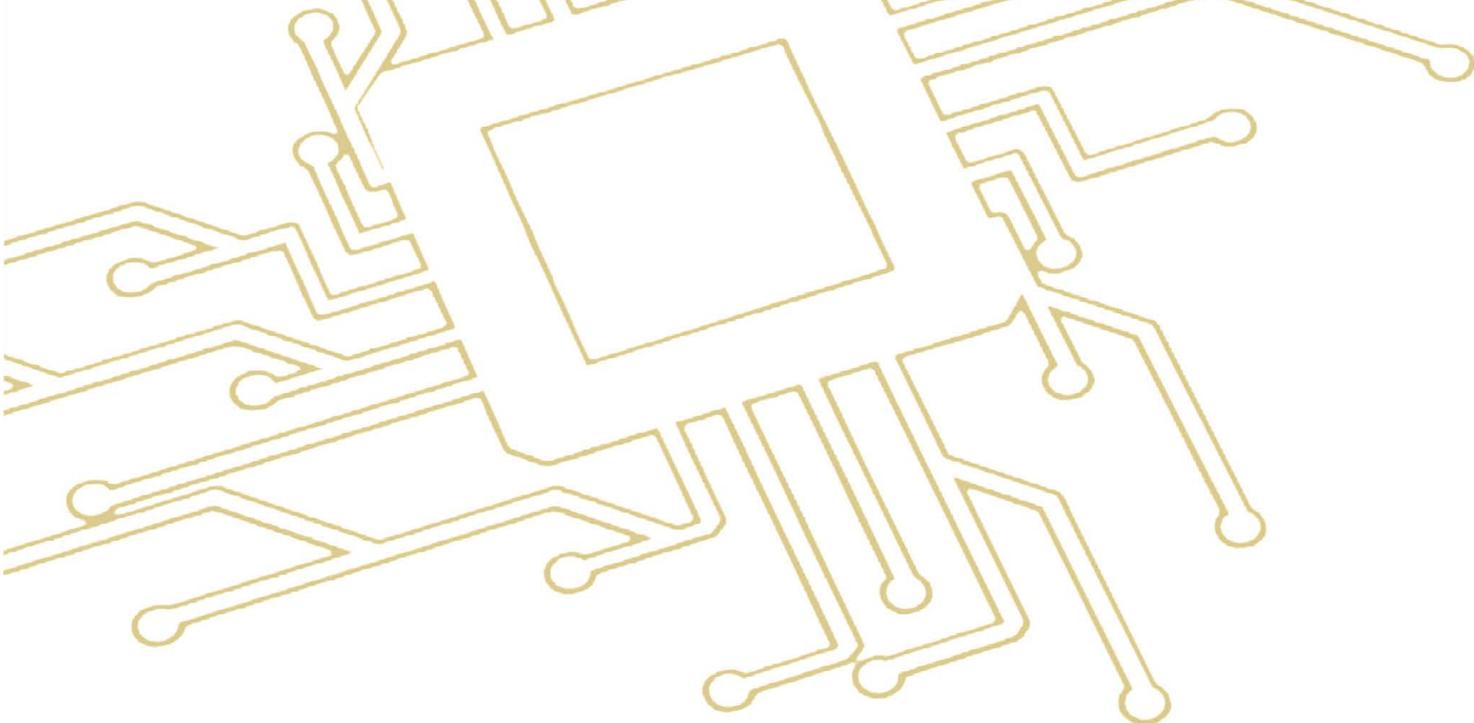


REFERENCIAS

[1] Colectivo de Autores, *ESP32 Series Datasheet*, 2023. Espressif Systems (Shanghai) Co., Ltd.

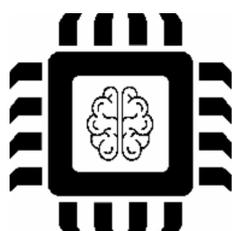
[2] DHT11 Temperature and Humidity Sensor Datasheet. Texas Instruments.

[3] Timothy J . Maloney, *Industrial Solid State Electronics: Devices and Systems* ,1980.ISBN:0-13-250225-9.



Eje Temático

Aplicaciones en Inteligencia Artificial





Modelos de inteligencia artificial en microcontroladores de bajo costo (TinyML): aplicación en identificación de colores y reconocimiento de patrones

Ilber Adonayt Ruge Ruge

Ingrid Carolina Ortiz Álvarez

Edwin Javier Sánchez Uriza

*Escuela de Ingeniería Electrónica
Tunja*

*Escuela de Ingeniería Electrónica
Tunja*

*Escuela de Ingeniería Electrónica
Tunja*

*Universidad Pedagógica y
Tecnológica de Colombia*

*Universidad Pedagógica y
Tecnológica de Colombia*

*Universidad Pedagógica y
Tecnológica de Colombia*

Tunja, Colombia

Tunja, Colombia

Tunja, Colombia

ilber.ruge@uptc.edu.co

ingrid.ortiz05@uptc.edu.co

edwinjavier.sanchez@uptc.edu.co

Resumen— Este artículo muestra la implementación y evaluación de modelos TinyML (Tiny Machine Learning) en microcontroladores económicos como Arduino UNO y MEGA2560 para dos aplicaciones prácticas: identificación de colores y reconocimiento de patrones. En el primer caso, se explora el uso de un sensor LDR y un modelo de clasificación de colores optimizado para microcontroladores utilizando redes neuronales del tipo Backpropagation. El segundo caso se centra en el reconocimiento de patrones para una aplicación didáctica utilizando una red neuronal multicapa que actúa como caperucita roja ante la identificación de patrones del lobo, la abuela y leñador. El entrenamiento de los modelos es realizado en MLP Topology Workbench, que es una plataforma de entorno interactivo para el diseño, análisis y optimización de redes neuronales multicapa, y ayuda a los usuarios a construir modelos más efectivos para tareas de aprendizaje automático e inteligencia artificial.

Palabras clave: TinyML, clasificación de colores, reconocimiento de patrones, máquinas de aprendizaje, Microcontrolador Arduino.

I. INTRODUCCIÓN

El avance de la tecnología de microcontroladores ha permitido la implementación de sistemas inteligentes en una

variedad de dispositivos con recursos limitados. Entre estas innovaciones, la integración de modelos de inteligencia artificial (IA) en microcontroladores de bajo costo, facilitada por la tecnología TinyML, ha emergido como una solución clave para habilitar capacidades avanzadas en aplicaciones de bajo consumo energético y bajo costo [1]. TinyML, o Machine Learning en dispositivos embebidos, se refiere a la ejecución de modelos de aprendizaje automático en dispositivos con limitaciones de memoria y procesamiento, haciendo posible que la IA se despliegue directamente en hardware con recursos reducidos [2].

La identificación de colores y el reconocimiento de patrones son dos aplicaciones que pueden ilustrar el beneficio de la tecnología TinyML. La identificación de colores se realiza utilizando sensores ópticos y algoritmos de clasificación, crucial en aplicaciones que van desde la automatización del hogar hasta la manufactura y la robótica [3]. Esta capacidad permite a los dispositivos distinguir entre diferentes colores en tiempo real, lo que resulta en sistemas más inteligentes y adaptativos [4]. Por otro lado, el reconocimiento de patrones emplea técnicas avanzadas como redes neuronales convolucionales (CNN), y permite a los dispositivos identificar y clasificar patrones complejos de datos obtenidos en la etapa de

sensado [5]. Esta habilidad es esencial para aplicaciones como la visión por computadora en dispositivos de bajo costo, donde se requiere una clasificación precisa de patrones a partir de datos visuales o de otro tipo [6-7].

Los casos de estudio proporcionan una visión integral sobre cómo los modelos de IA pueden ser optimizados y desplegados en microcontroladores económicos como Arduino Uno, Arduino MEGA, ESP32 entre otros, ampliando así el alcance y la accesibilidad de la inteligencia artificial en aplicaciones prácticas y de bajo costo.

II. METODOLOGIA

Para este estudio, se seleccionaron microcontroladores de bajo costo como el Arduino Uno basado en microcontrolador Atmega328P de la familia Microchip® de 8 bits, memoria Flash de 32KB, velocidad de operación hasta 20MHz, 6 canales ADC de 10 bits de resolución, voltaje de operación entre 1.8v y 5.5v, bajo consumo de energía de 0.2mA en modo activo y precio de compra de hasta USD30. Para el diseño de los modelos e aprendizaje automático para la identificación de colores y reconocimiento de patrones se utiliza la herramienta MLP Topology Workbench [8], desarrollado en JavaScript, HTML y CCS y comprende la implementación del clásico algoritmo de entrenamiento por retropropagación (backpropagation).

A. Implementación del modelo para identificación de colores

Se usa como dispositivo sensor de color una fotocelda (LDR: Resistencia dependiente de Luz) y una fuente constante de iluminación en disposición horizontal. Con este dispositivo se convierte la luz reflejada del objeto medido en corriente eléctrica: el voltaje y sus variaciones debidas al entorno (ruido) se entregan al conversor ADC del Arduino Uno. Se realizan pruebas experimentales para la conformación del conjunto de datos (dataset) en formato de texto plano (plain text) de entrada/salida según indicaciones del apartado "Dataset" de la plataforma MLP Workbench. El ajuste de parámetros en la interface de entrenamiento "Topology" y "Training" se establecen así: Inputs (1), 1st Hidden Layer (5), Outputs Layer

(5) y funciones de activación (sigmoid). Como criterios de parada: tasa de aprendizaje η (0.1) y precisión ϵ (0.01). La única entrada corresponde al voltaje del color sensado, y las cinco salidas a los colores azul, verde, amarillo, rojo y ambiente para los cuales se entrena la red neuronal. La plataforma genera el vector de pesos que conforman el modelo: el orden de los pesos en relación a los nodos es descrito en la guía de la plataforma para su uso y síntesis. En la Fig. 1 se muestra la asignación del pin analógico en el Arduino Uno para medir el voltaje del sensor de color, la definición del vector de pesos y un apartado de la síntesis de los dos primeros nodos del modelo de aprendizaje automático del sector de capa oculta y sus respectivas funciones de activación sigmoidal en el IDE Arduino®.

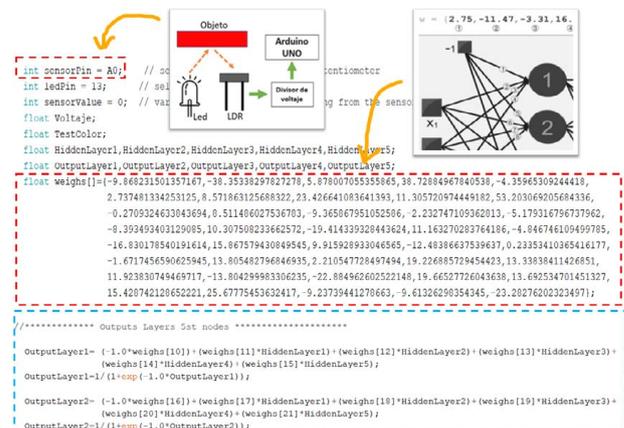


Fig. 1. Declaración del vector de pesos para síntesis del modelo de red neuronal en microcontrolador.

B. Modelo de Reconocimiento de Patrones

Una de las aplicaciones más frecuentes de las redes neuronales es el reconocimiento de patrones. De manera didáctica, se entrena una red multicapa para que actúe como caperucita roja y reconozca al lobo, al leñador y la abuela [9]. Las reacciones de caperucita roja frente a cada personaje se ilustran en la Tabla 1.

Tabla 1
RECONOCIMIENTO DE PATRONES PARA ACTUACIÓN DE CAPERUCITA ROJA.

Patrón	Reacción
Lobo	Corre, grita y busca al leñador
Abuela	Besar en la mejilla, ofrecer alimento
Leñador	Besar en la mejilla, ofrecer alimento, coquetear

Las características que se desean reconocer en cada uno de los sujetos, se ilustran en la Fig. 2.

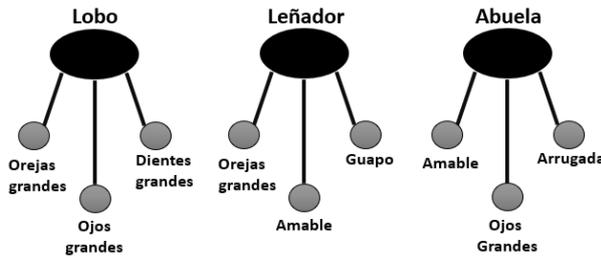


Fig. 2. Patrones a reconocer con una red neuronal.

Se plantea el diseño de una red neuronal multicapa 6-3-6. Los tres patrones (3 neuronas ocultas) tienen en total 6 características (orejas grandes, ojos grandes, dientes grandes, amable, arrugado y guapo) y caperucita roja presenta 6 reacciones (Correr, gritar, buscar al leñador, besar en la mejilla, ofrecer alimento y coquetear). La codificación binaria para los patrones se muestra en la Tabla II.

Tabla II
CODIFICACIÓN BINARIA DE PATRONES DE ENTRADA / SALIDA.

	Entradas	Salidas
Lobo	0.9 0.9 0.9 0.1 0.1 0.1	0.9 0.9 0.9 0.1 0.1 0.1
Abuela	0.1 0.9 0.1 0.9 0.9 0.1	0.1 0.1 0.1 0.9 0.9 0.1
Leñador	0.9 0.1 0.1 0.9 0.1 0.9	0.1 0.1 0.1 0.9 0.9 0.9

Donde 0.1 significa NO y 0.9 SI. Por ejemplo, el lobo SI tiene orejas grandes, SI tiene ojos grandes, SI tiene dientes grandes, NO es amable, NO esta arrugado, y NO es guapo. Caperucita cuando encuentra al lobo SI debe correr, SI debe gritar, SI debe buscar al leñador, NO lo besa en la mejilla, NO le ofrece alimento, y NO le coquetea.

Los valores para los patrones de entrada son generados mediante divisores de voltaje normalizados en software. Estos voltajes se entregan a 6 canales del conversor ADC de un Arduino MEGA2560 [10]. Para generar el modelo de aprendizaje automático se usa nuevamente la plataforma MLP Workbench, y el dataset se establece en formato plano de igual manera que la aplicación anterior, y bajo las consideraciones descritas en la Tabla I. El ajuste de parámetros en la interface de entrenamiento "Topology" y "Training" se establecen así: Inputs (6), 1st Hidden Layer (3), Outputs Layer (6) y funciones de activación (sigmoid). Como criterios de parada: tasa de aprendizaje η (0.1) y precisión ϵ (0.01). La plataforma genera el vector de pesos que conforman el modelo, y el orden de los pesos

en relación a los nodos es descrito en la guía de la plataforma. La Fig. 3 y 4 muestra la asignación de variables, vector de pesos y un apartado de la síntesis del modelo de aprendizaje automático del sector de capa oculta y sus respectivas funciones de activación sigmoidal en el IDE Arduino®.

```

Patrones Entrada float OrejasGrandes, OjosGrandes, DientesGrandes, Amable, Arrugado, BienParecido;
Vector de Pesos float weights[] = {1.095053380, 2.214673917, -0.074729968, 1.9114098028, -0.512105176,
-1.941247701, 1.195756167, 0.431952078, -0.393466104, 1.2987894684,
-2.285854631, 2.542483978, 1.042120807, 2.028979316, 1.275110376,
-0.20502598, -1.449629297, -1.149570785, 2.253286635, 0.008535666,
2.248936077, -0.119699761, 1.985189968, -2.284152709, -1.610472032,
-0.32645434, 1.794142374, -2.472490063, -1.52005636, -0.382208341,
1.705972287, -2.682136735, -1.255566068, 0.960828369, -1.158836224,
2.493635593, 2.022143355, 0.896701513, -1.278900291, 2.540052295,
1.95623485, 3.74034037, 2.107040615, 1.482219699, 2.251236598 };
Reacciones float HiddenLayer1, HiddenLayer2, HiddenLayer3;
float OutputLayer1, OutputLayer2, OutputLayer3, OutputLayer4, OutputLayer5, OutputLayer6;
float Correr, Gritar, BuscarLeñador, BesarMejilla, OfrecerAlimento, Coquetear;
Data_ADC=analogRead(A6);
OrejasGrandes=(Data_ADC*1.0)/1023.0;
Data_ADC=analogRead(A7);
OjosGrandes=(Data_ADC*1.0)/1023.0;
    
```

Fig. 3. Definición de variables y vector de pesos para RNA "Reacción caperucita roja".

```

//***** Evaluar RNA *****
//***** Hidden Layers 3st nodes *****
//***** Activation Function Sigmoidal g(u)=1/(1+e^-u) *****
HiddenLayer1= (-1.0*weights[0])+(weights[1]*OrejasGrandes)+(weights[2]*OjosGrandes)+
(weights[3]*DientesGrandes)+(weights[4]*Amable)+(weights[5]*Arrugado)+
(weights[6]*BienParecido);
HiddenLayer1= 1/(1+exp(-1.0*HiddenLayer1));
HiddenLayer2= (-1.0*weights[7])+(weights[8]*OrejasGrandes)+(weights[9]*OjosGrandes)+
(weights[10]*DientesGrandes)+(weights[11]*Amable)+(weights[12]*Arrugado)+
(weights[13]*BienParecido);
HiddenLayer2= 1/(1+exp(-1.0*HiddenLayer2));
//***** Outputs Layers 6st nodes *****
OutputLayer1= (-1.0*weights[21])+(weights[22]*HiddenLayer1)+(weights[23]*HiddenLayer2)+(weights[24]*HiddenLayer3);
OutputLayer1= 1/(1+exp(-1.0*OutputLayer1));
OutputLayer2= (-1.0*weights[25])+(weights[26]*HiddenLayer1)+(weights[27]*HiddenLayer2)+(weights[28]*HiddenLayer3);
OutputLayer2= 1/(1+exp(-1.0*OutputLayer2));
    
```

Fig. 4. Apartados de síntesis de capas ocultas y salidas de la RNA "Reacción caperucita roja"

III. ANÁLISIS DE RESULTADOS

Se presenta un análisis detallado de los resultados obtenidos de la implementación de los modelos TinyML en microcontroladores de bajo costo, con enfoque en dos aplicaciones específicas: identificación de colores y reconocimiento de patrones. Se analizan la precisión, velocidad de procesamiento y eficiencia energética de los modelos implementados, así como su rendimiento en diferentes condiciones de prueba.

El modelo de identificación de colores implementado en el Arduino Uno mostró una precisión promedio del 92% en la clasificación de colores. Esta precisión fue consistente en condiciones normales de iluminación, pero presentó una ligera disminución al enfrentarse a variaciones extremas en la iluminación (baja o alta intensidad). La precisión en estas

condiciones extremas se redujo a aproximadamente 85% debido a la sensibilidad del sensor de color a las variaciones de luz.

El tiempo promedio de procesamiento por entrada fue de 28.9 ms. Este tiempo incluye la captura del dato del sensor y el procesamiento por parte del modelo TinyML. La técnica utilizada para medir la latencia de la aplicación es el uso de la función `micros()` de Arduino al inicio y fin del código: esta función devuelve el tiempo transcurrido en Microsegundos.

```
void loop() {
  unsigned long startTime = micros(); // Tiempo de inicio

  sensorValue = analogRead(sensorPin);
  Voltaje=(sensorValue*5.0)/1023.0;
  TestColor= Voltaje/5.0;
  //TestColor=0.56;

  ANN_Evaluation(TestColor);

  unsigned long endTime = micros(); // Tiempo de finalización
  unsigned long latency = endTime - startTime; // Calcula la latencia
  Serial.print("Latencia: ");
  Serial.print(latency);
  Serial.println(" microsegundos");
}
```

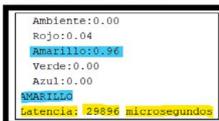


Fig. 5. Latencia de la RNA "Clasificación de Colores".

El consumo de energía durante la ejecución del modelo en el Arduino Uno fue de aproximadamente 15 mAh, lo que indica una alta eficiencia energética. La medida fue realizada con el probador digital USB con pantalla LCD de referencia UNI-T UT658. El porcentaje de uso de espacio de almacenamiento de memoria fue de 19%, correspondiente a 6174 bytes de los 32256 bytes (32K Flash Memory Program) de su capacidad total, y 16% de la memoria dinámica correspondiente a 328 bytes de los 2048 bytes (2K RAM).

El modelo de reconocimiento de patrones implementado en el Arduino MEGA2560, logró una precisión promedio del 88% en la clasificación de patrones. Por ejemplo, un leñador grosero tiene un patrón (0.9 0.1 0.1 0.6 0.1 0.9) y la red responde (0.4 0.4 0.4 0.6 0.6 0.6), es decir, Caperucita Roja no está segura de besar al leñador (0.6). Por tanto, la precisión fue relativamente alta para patrones simples, pero mostró una disminución en patrones más complejos. Esta precisión para patrones complejos fue de aproximadamente 80%.

El tiempo promedio de procesamiento fue de 34.8 ms. La técnica utilizada para medir la latencia de la aplicación es igual a la descrita en la aplicación de clasificación de colores. El consumo de energía durante la ejecución del

modelo en el Arduino MEGA2560 fue de aproximadamente 24 mAh, medido con el probador digital USB UNI-T UT658. El porcentaje de uso de espacio de almacenamiento de memoria fue de 11%, correspondiente a 27372 bytes de los 262144 bytes (256K Flash Memory program) de su capacidad total, y 16% de la memoria dinámica correspondiente a 1383 bytes de los 8192 bytes (8K RAM).

El Arduino MEGA2560 consumió aproximadamente 24 mAh durante la ejecución del modelo de reconocimiento de patrones. Este consumo es mayor que el del Arduino Uno debido a la mayor complejidad del modelo. Sin embargo, el consumo de energía sigue siendo manejable para aplicaciones de bajo costo con suministro adecuado. Los microcontroladores de bajo costo demuestran un buen equilibrio entre velocidad y eficiencia energética. Aunque el tiempo de procesamiento puede llegar a ser mayor en comparación con sistemas más potentes como Raspberry Pi, BeagleBone, Arduino Yun, Odroid-C4, Orange Pi, entre otros, la eficiencia energética es significativamente mejor, lo que es beneficioso para aplicaciones que requieren operación continua con batería.

Los modelos mostraron variabilidad en el desempeño bajo diferentes condiciones ambientales y de entrada. En el caso de la identificación de colores, la variabilidad en la iluminación afectó la precisión mientras que, en el reconocimiento de patrones la complejidad de los valores de las características de los sujetos influyó en los resultados. Para mejorar el desempeño de los modelos TinyML, se recomienda explorar técnicas adicionales de optimización y compresión, así como el uso de sensores de mejor desempeño y precisión. Además, la implementación de algoritmos de corrección de errores y técnicas de preprocesamiento de datos podría mejorar la robustez de los sistemas en condiciones variadas.

IV. CONCLUSIONES

El uso de TinyML en microcontroladores de bajo costo para aplicaciones de identificación de colores y reconocimiento de patrones es viable y ofrece varias ventajas, como la reducción de costos y la mejora en la eficiencia energética. Los resultados obtenidos demuestran que estos sistemas pueden ser



implementados en una variedad de aplicaciones prácticas. Sin embargo, se deben considerar limitaciones como el tamaño del modelo y la capacidad de procesamiento para aplicaciones más complejas.

La investigación confirma que los modelos de inteligencia artificial basados en TinyML son viables y efectivos cuando se implementan en microcontroladores de bajo costo. Los microcontroladores seleccionados, como el Arduino Uno y Arduino MEGA2560, demostraron ser capaces de ejecutar modelos de aprendizaje automático complejos con un rendimiento aceptable en términos de precisión y velocidad. Estos resultados abren la oportunidad de integrar algoritmos inteligentes en dispositivos con limitaciones de recursos, extendiendo su aplicabilidad en diversas áreas.

El reconocimiento de patrones e identificación utilizando redes neuronales en microcontroladores de bajo costo mostró resultados prometedores. Los modelos fueron capaces de identificar patrones con una tasa de éxito entre el 80% y 92%, aunque se observaron limitaciones relacionadas con la complejidad de los patrones. Estos desafíos destacan la necesidad de continuar desarrollando técnicas de optimización para mejorar el desempeño en aplicaciones más complejas. De igual manera, la eficiencia energética y la reducción de costos son dos de las principales ventajas al utilizar TinyML en microcontroladores de bajo costo. Los modelos implementados demostraron un consumo energético de hasta 15mAh, lo que es crucial para aplicaciones que requieren operación continua y autónoma. Además, el bajo costo de los microcontroladores permite la expansión de soluciones inteligentes a una mayor variedad de aplicaciones y mercados.

La capacidad de implementar modelos de IA en microcontroladores de bajo costo tiene el potencial de revolucionar diversas industrias, incluyendo la automatización del hogar, la robótica, la manufactura y la vigilancia. Las soluciones inteligentes basadas en TinyML pueden mejorar la eficiencia operativa, reducir costos y permitir la creación de productos y servicios innovadores con capacidades de procesamiento en el borde (Edge Computing).

REFERENCIAS

- [1] P. Warden, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. Sebastopol, CA: O'Reilly Media, 2019.
- [2] C. A. L. Martin, *Hands-On Machine Learning for Embedded Systems*. Sebastopol, CA, USA: O'Reilly Media, 2021.
- [3] J. A. Stankovic, *Machine Learning for Embedded Systems*. Cambridge, UK: Cambridge University Press, 2024.
- [4] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. Senhaji Hafid, A Comprehensive Survey on TinyML, *IEEE Access*, vol. 1, pp. 1-35, Mar. 2024.
- [5] N. Tekin, A. Aris, A. Acar, S. Uluogac, and V. C. Gungor, A Review of On-Device Machine Learning for IoT: An Energy Perspective, *Ad Hoc Networks*, vol. 153, pp. 256-278, Feb. 2024.
- [6] Q. Zhang, L. Zhuo, J. Li, J. Zhang, H. Zhang, and X. Li, Vehicle color recognition using multiple-layer feature representations of lightweight convolutional neural network, *Signal Processing*, vol. 147, pp. 146-153, Aug. 2018.
- [7] F. Sabry, T. Eltaras, W. Labda, K. Alzoubi, and Q. Malluhi, *Machine Learning for Healthcare Wearable Devices: The Big Picture*, *Journal of Healthcare Engineering*, vol. 2022, 25 pages, 2022.
- [8] Benati, C. MLP Topology Workbench – A background for Multi-layer Perceptrons, [Online]. Available: <http://www.moretticb.com/blog/mlp-topology-workbench-a-playground-for-multilayer-perceptrons/>. [Accessed: 30-Jul-2024].
- [9] Delgado, A. *Inteligencia Artificial y Minirobots*. ECOE Ediciones. pp. 68-70. Bogotá D.C. 1998.
- [10] Atmel ATmega2560 Datasheet. 8-bit Atmel microcontroller with 16/32/64KB In-System Programmable Flash. [Online]. Available: https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf. [Accessed: 30-Jul-2024]



Rendimiento de algoritmos de reconocimiento facial para autenticación biométrica en sistemas de monitoreo doméstico con Edge Computing

Vanessa Luna Rosero

Alejandro Muñoz Lopez

Universidad Mariana

Universidad Mariana

facultad de ingeniería mecatronica

facultad de ingeniería mecatronica

Pasto, Colombia

Pasto, Colombia

vanesalunarosero@gmail.com

Gamljc97@gmail.com

Resumen— Este trabajo presenta un análisis comparativo de técnicas de reconocimiento facial aplicadas en una plataforma de Edge Computing, utilizando la tarjeta Jetson Nano. Se evaluaron los algoritmos YOLOv8, VGG y ResNet50 en métricas como precisión, confianza, recall y matriz de confusión. Se entrenaron los modelos con una base de datos de 893 imágenes, divididas en una proporción de 74/26 para entrenamiento y validación. Para optimizar el rendimiento, se emplearon técnicas de aumento de datos. Los resultados muestran que el modelo más robusto logró una precisión superior al 90% y tiempos de ejecución más rápidos, demostrando su viabilidad para sistemas de monitoreo residencial.

Palabras clave: Reconocimiento facial, Edge Computing, Jetson Nano, YOLOv8, biometría.

I. INTRODUCCIÓN

Este artículo explora el campo en expansión del reconocimiento facial, enfocándose específicamente en algoritmos que aprovechan técnicas de aprendizaje automático supervisado en el contexto de plataformas de Edge Computing. Con una variedad cada vez mayor de métodos, algoritmos y técnicas disponibles para el reconocimiento facial, cada uno presenta un conjunto único de características y métricas de rendimiento. Sin embargo, estos

métodos no están exentos de desafíos. Los problemas comunes incluyen dificultades en la transmisión de imágenes en videos en vivo o en reproducción debido a factores como la mala iluminación, sesgos raciales o étnicos, cambios en la apariencia facial a lo largo del tiempo y obstrucciones causadas por accesorios faciales. Esta investigación tiene como objetivo profundizar en estos desafíos, realizando un análisis comparativo para identificar y evaluar algoritmos que exhiben un rendimiento superior en el reconocimiento facial. El objetivo final es determinar el algoritmo más efectivo para la implementación en sistemas de monitoreo del hogar, aprovechando el poder del Aprendizaje Automático y Edge Computing.[1].

En la actualidad, se emplean una amplia variedad de algoritmos para el procesamiento de imágenes y el reconocimiento facial en diversos campos, como seguridad, medicina diagnóstica, entrenamiento y aplicaciones militares. El objetivo principal es lograr una inferencia más eficiente y precisa con menor costo computacional, empleando plataformas móviles o de Edge Computing en la identificación de un rostro (Diego A, 2002) [2]. Además, el uso de sistemas informáticos en conjunto con el aprendizaje automático, especialmente mediante el empleo de redes

neuronales profundas, ha permitido avances significativos en el procesamiento de imágenes faciales y en la parametrización de un espacio euclidiano para medir las distancias entre característica faciales.

II. METODOLOGÍA

En la Tabla 1 se presentan un análisis de los modelos más influyentes en cuanto clasificación y detección en objetos y reconocimiento facial cada uno con sus diferentes arquitecturas entre ellos se encuentran YoloV8 y su versión V5, LBP (Local Binary Pattern), HAAR, ResNet50Y VGG.

TABLA I
ANÁLISIS DE MODELOS

Modelo	Precisión	Técnica de detección	Velocidad de procesamiento	Entrenamiento
YOLO V8	Muy alta	Redes Neuronales Convolucionales profundas	Muy rápida	Requiere gran cantidad de datos y recursos computacionales
LBP	baja	Patrones binarios locales	Moderada	Requiere menos datos y es menos intensivo en recursos
VGG	Alta	Red profunda	Moderada	Requiere gran cantidad de datos, recursos computacionales,
Haar	Moderada	Análisis de características de Haar para identificación de rasgos faciales	Rápida	Necesita un conjunto de datos Moderado y ajustes manuales
ResNet 50	Alta	Red neuronal convolucional (CNN)	Rápido	Requiere una cantidad significativa de recursos
YOLO	no es el	- FCOS	La	YOLOv5

modelo (Fully Convolutional One-Stage) p más preciso en términos absolutos velocidad de procesamiento varía según el tamaño del modelo puede entrenarse con una variedad de tamaños de datos,

A lo largo de las distintas fases del proyecto, se ha seguido un enfoque estructurado y metódico, dividido en varias fases clave. Cada etapa ha sido abordada meticulosamente, desde la selección de sistemas y algoritmos hasta la implementación y evaluación de rendimiento en entornos de Edge Computing. El proyecto se desarrollará en tres fases claves, presentadas a continuación:

A. Fase I

Para establecer los criterios de un método computacional basado en IA es importante realizar una revisión amplia y exhaustiva de conceptos, tener apropiación conceptual y haber experimentado con los diferentes tipos de algoritmos bases encontrados.

- Base de datos y criterios de búsqueda: Con la finalidad de lograr los resultados deseados, se presenta una metodología de mapeo sistemático, que permite identificar y comprender los lineamientos para abordar el desarrollo del proyecto.

- Análisis comparativo: Investigar y evaluar diferentes sistemas de autenticación biométrica basados en reconocimiento facial disponibles en el mercado. escogiendo los mejores y Comparando sus características, precisión, velocidad.

- Selección: seleccionar el sistema embebido que mejor se adapte a las necesidades del proyecto se ajuste a los criterios de precisión, velocidad y seguridad, asociados con la implementación y el mantenimiento del sistema.

Finalizando la Fase 1, obtendremos o elegiremos un sistema el cual contara con las características idóneas para un método computacional y para la aplicación de la inteligencia artificial mejorando así el proceso de reconocimiento de rostros.

B. Fase II

En la Fase 2, se llevó a cabo una evaluación exhaustiva y una comparación de algoritmos de identificación y clasificación de características faciales previamente seleccionados en la Fase I debido a su potencial. El objetivo principal consistió en analizar minuciosamente el rendimiento de estos algoritmos en condiciones variables de iluminación y poses. Esta evaluación permitió identificar los algoritmos más robustos y precisos, adecuados para entornos reales y desafiantes en el reconocimiento facial. Estos factores se consideran esenciales para mejorar la precisión y confiabilidad de los sistemas de autenticación biométrica.

Implementación y entrenamiento: de los modelos resultantes de la primera fase, se utilizaron versiones de YOLO (Ultralytics YOLOv8 y YOLOv5), VGG (Visual Geometry Group) y ResNet50 (Red Residual de 50 capas). Se creó una base de datos con 893 imágenes capturadas por la cámara de un dispositivo móvil, etiquetadas como "Vanessa" y "Alejandro". Además, se aplicó la técnica de aumento de datos para optimizar el entrenamiento.

Evaluación de rendimiento: Durante la evaluación de los seis modelos, se consideraron métricas de rendimiento como el tiempo de ejecución de cada uno, y la versión óptima en términos de tiempo de ejecución de cada modelo, con el objetivo de identificar los más eficientes, tal como se muestra en la Tabla 2. Además, emplea el optimizador *Adam* a los modelos para ajustar los pesos durante el entrenamiento, para mejorar la precisión general.

TABLA II
Tiempo de Ejecución de los modelos

Modelo	Tiempo de ejecución
VGG	3,20h
YOLOV8	0,229h
RESNET50	2,30h
YOLO V5	0,540h
HAAR	9,34h
LBP	10,45h

Teniendo en cuenta las especificaciones anteriores, se seleccionaron los tres modelos que cumplían con los requisitos: VGG, YOLOv8 y ResNet50. Estos modelos fueron sometidos a la etapa de entrenamiento bajo las mismas condiciones, utilizando 60 épocas, el optimizador "Adam", la misma base de datos y se usó la función *Early Stopping* la cual detiene el entrenamiento cuando el rendimiento en un conjunto de validación comienza a empeorar, evitando así el sobreajuste.

C. Fase III

Configuración de la plataforma de Edge Computing. Tras obtener resultados favorables, el entrenamiento se implementa en un sistema de Edge Computing, utilizando en la primera fase un kit de desarrollo Jetson Nano. Para configurarlo, se utiliza una tarjeta microSD con capacidades de lectura y escritura superiores a 100 Mb/s para garantizar una velocidad adecuada. Se descarga la imagen ISO de Jetpack 4.6 desde el sitio oficial de NVIDIA, y se siguen las instrucciones proporcionadas por la compañía para completar la configuración. Posteriormente, se instalan los requisitos necesarios para cada modelo. En este proceso, se emplea la herramienta Docker para crear contenedores, lo que evita conflictos entre versiones de software y facilita la ejecución de los modelos.



Fig 1. Configuración del kit de desarrollo Jetson nano

Análisis de resultados. Las pruebas finales son cruciales para validar el rendimiento del dispositivo en condiciones reales y garantizar que cualquier inconveniente sea resuelto antes de su implementación en un entorno práctico. Finalmente, se verifica la funcionalidad de los modelos usando el entorno de desarrollo Visual Studio Code, usando una versión de Python 3.10 y cargando los pesos entrenados que se guardan en un formato .pt (Pytorch) en la plataforma Jetson Nano, con el objetivo de visualizar en tiempo real la correcta detección de las clases. Estos resultados son fundamentales para evaluar la efectividad de una herramienta de control de acceso doméstico basada en inteligencia artificial y aprendizaje automático supervisado.

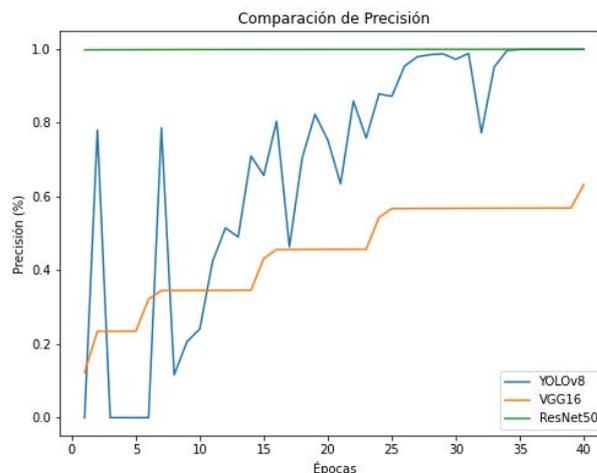


Fig 3. Comparativa precisión

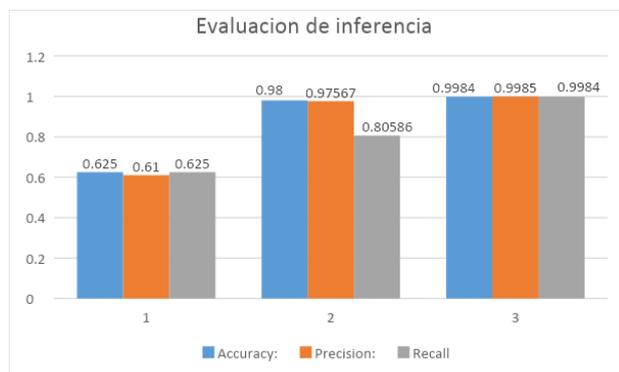


Fig. 2 Resultados de los modelos.

Basándonos en los resultados presentados en la Figura 23, YOLOv8 supera a los modelos VGG y RESNET50 en términos de Accuracy, precisión y Recall, mientras también es más eficiente en tiempo de inferencia.

Las Figuras 3, 4 y 5 indican un resultado gráfico de la comparación de los tres algoritmos presentados anteriormente en función de las métricas, precisión, Recall y Accuracy. Donde se puede apreciar de mejor manera el comportamiento de la curva de cada modelo.

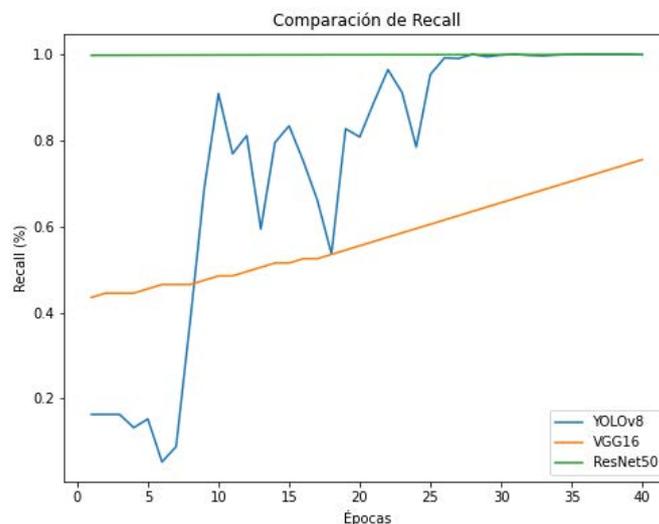


Fig. 4 Comparativa Recall

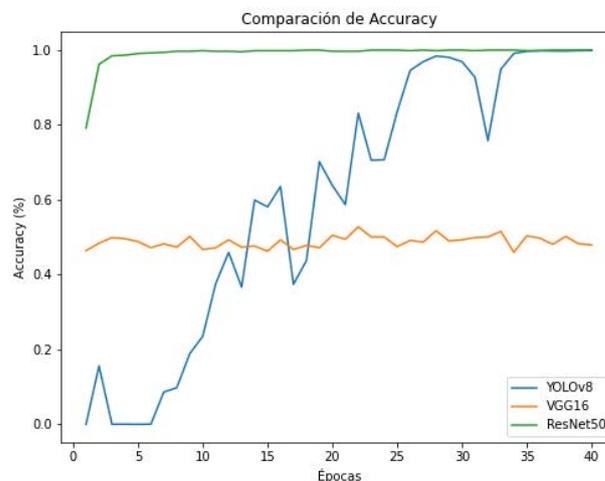


Fig. 5 Comparativa Accuracy

III. RESULTADOS

Este proyecto ha demostrado que, mediante la selección cuidadosa de algoritmos, el entrenamiento especializado y la optimización en plataformas de Edge Computing, es posible desarrollar un sistema de reconocimiento facial altamente eficiente y seguro. Al evaluar los algoritmos VGG, YOLOv8 y ResNet50 en un kit de desarrollo de Jetson Nano, se comprobó que cada modelo ofrece ventajas específicas en términos de precisión, Recall, matriz de confusión, Accuracy y tiempo de ejecución. El impacto potencial del proyecto a corto plazo, los resultados de este proyecto podrían acelerar la adopción de tecnologías de Edge Computing en aplicaciones de seguridad y reconocimiento facial, ofreciendo soluciones más rápidas y seguras que aquellas basadas en procesamiento en la nube. A largo plazo, el impacto potencial incluye la integración de sistemas de reconocimiento facial más confiables y eficientes en una variedad de contextos, desde la seguridad en el hogar hasta la vigilancia en entornos comerciales.

Factores críticos como la iluminación, la pose y las variaciones en los datos de entrada (como peinados y expresiones faciales) fueron identificados como elementos que pueden influir en el rendimiento de los algoritmos. La incorporación de un análisis detallado de estos factores y su mitigación mediante técnicas de preprocesamiento y ajustes en los modelos ha demostrado ser esencial para mejorar el sistema.

En la tarjeta Jetson nano el uso de Docker para la gestión de entornos permitió evitar conflictos de versiones y facilitó la implementación de los modelos en la plataforma, lo cual es necesario para garantizar la estabilidad del sistema en un entorno real. Este proyecto genera las bases para futuras investigaciones y mejoras en el campo de la seguridad digital y el monitoreo en entornos domésticos y empresariales. El impacto social, ético y económico de este proyecto también es significativo. Desde un punto de vista social, una mayor precisión en el reconocimiento facial puede mejorar la seguridad personal y comunitaria, reduciendo el riesgo de falsos positivos y errores de identificación. Sin embargo, también es necesario considerar los aspectos éticos, como la privacidad y el sesgo algorítmico, que deben

ser abordados para evitar desigualdades y proteger los derechos individuales.

REFERENCIAS

- [1] Antona-Cortés, C. (2007). *Herramientas modernas en redes neuronales: la librería Keras*. Editorial UAM. Departamento de Ingeniería Informática
- [2] Gortázar-Bellas, F., Martínez-Unanue, R., y Fresno-Fernández, V. (2016). *Lenguajes de programación y procesadores*. Scribd. <https://es.scribd.com/document/470290483/Lenguajes-de-Programacion-y-Procesadores-Francisco-Gortazar-Bellas-Raquel-Martinez-Unanue-Victor-Fresno-Fernandez>
- [3] Graveto, V., Cruz, T., y Simões, P. (2022). Security of Building Automation and Control Systems: Survey and future research directions. *Rev. Computers & Security*, 112.
- [4] Github. (2023). *Ultralytics. YOLOv8: Real-Time Object Detection*. Github. <https://github.com/ultralytics/ultralytics>
- [5] Han, H., Shan, S., Chen, X., y Gao, W. (2013). A comparative study on illumination preprocessing in face recognition. *Rev. Pattern Recognition*, 46(6), 1691-1699.
- [6] Igual, R., y Medrano, C. (2008). *Tutorial de OpenCV*. Academia.edu. https://www.academia.edu/29801907/Tutorial_de_OpenCV
- [7] Keras. (s.f.). *ResNet and ResNetV2*. Keras documentation. <https://keras.io/api/applications/resnet/>
- [8] Kowalski, M. Ł., y Grudzień, A. (2018). High-resolution thermal face dataset for face and expression recognition. *Rev. Metrology and Measurement Systems*, 25, 403–415. <https://doi.org/10.24425/119566>
- [9] Lindner, T., Wyrwał, D., Białek, M., y Nowak, P. (2020). *Face recognition system based on a single-board computer* (Presentacion de la conferencia). International Conference Mechatronic Systems and Materials (MSM).
- [10] Lluç-Crespo, J. (2022). *Introducción a la librería Pandas*. Universitat Politècnica de Valencia. <https://riunet.upv.es/handle/10251/183074?show=full>
- [11] Lundberg, S. (2018). *Welcome to the SHAP documentation*. Shap. <https://shap.readthedocs.io/en/latest/>
- [12] Molina-Garrido, D. A. (2019). *Clasificación de género con análisis racial en imágenes de espectro visual mediante técnicas de Deep Learning*. S.n.
- [13] Perez-Prieto, J. A. (2019). *La librería científica Scipy*. Reserach,iac.es. <https://research.iac.es/sieinvens/python-course/scipy.html>
- [14] Pertuz, C. M. P. (2022). *Aprendizaje automático y profundo en python*. Editorial Ra-Ma.
- [15] Policía Nacional de Colombia-PONAL. (2022). *Estadística delictiva*. PONAL.
- [16] Rojas, E. M. (2020). Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo. *Rev. Ibérica de Sistemas e Tecnologías de Informação*, (E28), 586-599.
- [17] Rouhiainen, L. (2018). *Inteligencia artificial 101 cosas que debes saber hoy sobre nuestro futuro*. Editorial Alienta.
- [18] Sánchez-Alberca, A. (2016a). La librería Matplotlib. *Blog Aprende con Alf*. <https://aprendeconalf.es/docencia/python/manual/capitulo-matplotlib/>
- [19] Sánchez-Alberca, A. (2016b). La librería Numpy. *Blog Aprende con Alf*. <https://aprendeconalf.es/docencia/python/manual/capitulo-numpy/>
- [20] Santana, M. A. G., Díaz-Sánchez, L. E., Paz, I. T., y Huertas, M. R. (2017). Estado del arte en reconocimiento facial. *Rev. Res. Comput. Sci.*, 140, 19-27.
- [21] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Rev. Neural Networks*, 61, 85-117. doi: 10.1016/j.neunet.2014.09.003
- [22] Shetty, A., Bhoomika, E., Deeksha, B., Rebeiro, J., y Ramyashree, B. (2021). Facial recognition using Haar cascade and LBP classifiers. *Rev. Global Transitions Proceedings*, 2(2), 330-335.



- [23] Socolinsky, D. A., y Selinger, A. (2002). A comparative analysis of face recognition performance with visible and thermal infrared imagery. *Rev. International Conference on Pattern Recognition*, 4, 217-222.
- [24] Tahir, M. F., y Saqib, M. A. (2016). Optimal scheduling of electrical power in energy-deficient scenarios using artificial neural network and Bootstrap aggregating. *Rev. International Journal of Electrical Power & Energy Systems*, 83, 49-57.
- [25] UniPython. (2021). Detección de rostros, caras y ojos con haar cascad. *Blog UniPython*. <https://unipython.com/deteccion-rostros-caras-ojos-haar-cascad/>
- [26] Universidad De Alcalá. (2022). *Scikit-learn, herramienta básica para el data science en python*. Universidad de Alcalá. <https://www.master-data-scientist.com/scikit-learn-data-science/>
- [27] Rozada Raneros, S. (2021). Estudio de la arquitectura YOLO para la detección de objetos mediante deep learning.
- [28] Villalba, M. (2020, October 26). Arquitectura VGG16 y VGG19 en Deep Learning. <https://keepcoding.io/blog/arquitectura-vgg16-vgg19-deep-learning/>
- [29] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90. keywords: {Training;Degradation;Complexity theory;Image recognition;Neural networks;Visualization;Image segmentation},
- [30] Valle-Barrio, A. (2018). *Aplicación de Tensorflow en deep learning* (Tesis de magister, Universidad Politecnica de Madrid).
- [31] Zou, J., Ji, Q., y Nagy, G. (2007). Un estudio comparativo del enfoque de coincidencia local para el reconocimiento facial. *Rev. IEEE Transactions on image processing*, 16(10), 2617-2628.

Detección de armas sobre video

Florencia Priscilla Vela

Facultad de Ingeniería

Universidad de Buenos Aires

Ciudad Autónoma de Buenos Aires, Argentina

flor.p.vela@gmail.com

Resumen—Este trabajo presenta el diseño e implementación de un sistema en tiempo real para la detección de armas y el seguimiento de personas en videos mediante técnicas de visión por computadora. Se entrenaron y optimizaron modelos YOLO con conjuntos de datos variados en cuanto a escenarios, iluminación y entornos. Además, se exploran métodos de seguimiento de objetos múltiples, como ByteTrack, para el seguimiento de personas de interés en escenas complejas. También se aplican optimizaciones con OpenVINO y técnicas de cuantización para acelerar la inferencia, haciendo el sistema adecuado para entornos con limitaciones de hardware.

Palabras clave—Visión por computadora, Detección de armas, OpenVINO, ByteTrack, YOLO.

I. Introduction

Los sistemas de videovigilancia se han convertido en una herramienta esencial para la protección de bienes y personas, desempeñando un papel crucial en la detección y prevención de situaciones de riesgo. Estos sistemas permiten una respuesta rápida ante incidentes, lo que los hace indispensables en la prevención y control de delitos, así como en el monitoreo de áreas sensibles.

Se proyecta que el crecimiento del mercado de la videovigilancia se vea impulsado por la introducción de nuevas tecnologías digitales basadas en inteligencia artificial, enfocadas en la detección y prevención de comportamientos indeseables, como el robo, el vandalismo y los ataques terroristas. Según un informe publicado por Mordor Intelligence [1], el mercado global de sistemas de videovigilancia se valoró en 52.450 millones de dólares en el año 2020 y se espera que alcance un valor de 90.370 millones de dólares para el año 2026, con una tasa de crecimiento anual estimada del 9,31% entre 2021 y 2026.

La inteligencia artificial y el aprendizaje automático han revolucionado la videovigilancia al permitir el desarrollo de sistemas capaces de aprender y adaptarse a diferentes escenarios. Si bien estas técnicas fueron propuestas hace décadas, recientemente lograron revolucionar el estado del arte gracias a dos factores clave. En primer lugar, la disponibilidad de grandes volúmenes de datos que permitió entrenar arquitecturas con millones de parámetros, lo que ha mejorado significativamente la capacidad de aprendizaje y generalización de las redes neuronales. En segundo lugar, el avance tecnológico y

la reducción de costos del hardware de cálculo intensivo, como las Unidades de Procesamiento Gráfico (GPU), permitieron alcanzar la potencia necesaria para realizar cálculos masivos y acelerar el entrenamiento y la inferencia de las redes neuronales. Estos desarrollos han potenciado la detección de eventos y la identificación de objetos y personas en tiempo real, con aplicaciones en áreas críticas como la detección de armas.

En los últimos años, se comprobó la eficacia de las técnicas de aprendizaje profundo y las Redes Neuronales Convolucionales (CNN, por sus siglas en inglés) en detección, clasificación y segmentación de imágenes. Para resolver estos problemas, se utilizan modelos como YOLO [2], Faster R-CNN [3] y VGG-16 [4] con resultados satisfactorios.

A pesar de estos avances, la detección de armas en videos presenta desafíos significativos, como la oclusión parcial o total de los objetos de interés, la variabilidad en las condiciones de iluminación y la pérdida de información durante la transmisión de imágenes. Tiwari et al. [5] subrayan estos desafíos específicos, que son ilustrados en la Fig. 1. Otro problema común en estos sistemas es la tasa de falsos negativos y falsos positivos. Dada la criticidad de estos sistemas, se requiere un procesamiento en tiempo real y una respuesta rápida. Por lo tanto, la investigación en este ámbito se enfoca en identificar y aplicar técnicas que aceleren el procesamiento de los modelos.

En este trabajo, se propone un sistema en tiempo real para la detección de armas y el seguimiento de personas en videos, utilizando técnicas de procesamiento de imágenes y aprendizaje automático y algoritmos de optimización de modelos. Este sistema busca mejorar la seguridad y optimizar las tareas de videovigilancia mediante la implementación de modelos que no solo sean precisos, sino también capaces de operar eficientemente en entornos de hardware limitados.

II. Trabajos relacionados

La detección de armas en videos ha sido ampliamente estudiada en la última década, con avances significativos en la precisión y la velocidad de los modelos de visión por computadora. En 2017, Olmos et al. [6] realizaron un



Fig. 1: Desafíos en la detección de armas. (a) Oclusión parcial. (b) Diferencia en formas. (c) Cambio de la vista. (d) Cambio en la iluminación. (e) Variación en la escala y la rotación. (f) Imagen con ruido. Imagen tomada de [5].

estudio utilizando redes neuronales convolucionales profundas (CNN) para la detección de objetos, desarrollando un conjunto de datos con más de 9.000 imágenes de diversas armas. Su enfoque basado en regiones, utilizando Faster R-CNN, mostró un rendimiento sobresaliente con 0 falsos positivos, 100% de *recall* y una precisión del 84,21%. Este método demostró ser útil para aplicaciones en tiempo real, como la detección de armas en áreas vigiladas y el control de contenido en redes sociales.

Uno de los primeros trabajos sobre detección de armas en videos de vigilancia fue propuesto en el año 2015 en [5], donde utilizaban segmentación basada en colores y detección de puntos de interés de Harris. La segmentación basada en colores se utiliza para eliminar colores u objetos no relacionados que no son de interés, quedándose con los colores de las armas que se buscaban identificar (negros y grises). Luego, las características de puntos de interés de Harris se utilizan para descubrir la similitud de cada objeto segmentado con el descriptor del arma. Si más del 50% de las características del descriptor del arma coinciden con las características del objeto, ese objeto se etiqueta como arma. En 2015 este enfoque era novedoso porque era resistente a la oclusión parcial, escala, rotación, variación afín y era capaz de detectar la presencia de múltiples armas en una imagen.

En trabajos como [7] y [8] se abarca la problemática desde una perspectiva de videovigilancia. Los autores de [7] entrenaron un modelo de detección de objetos llamado M2Det, que utiliza una arquitectura de red neuronal llamada MLFPN (*Multi-Level Feature Pyramid Network*). Esta arquitectura utiliza una pirámide de características de varios niveles para detectar objetos en una sola etapa. Los resultados obtenidos con su conjunto de datos fueron superiores a otros conjuntos de datos existentes, ya que tuvieron en cuenta la iluminación y los diferentes ambientes en los que se podrían encontrar las armas. Por otro lado, en [8] se destaca la necesidad de un sistema que pueda detectar armas y enviar alertas al personal de seguridad. Los autores presentaron un conjunto de datos diseñado específicamente para resolver esta tarea. En ambos trabajos, se hizo hincapié en la importancia de armar conjuntos de datos que incluyeran armas en diferentes contextos, lo cual contribuye a lograr

una mayor precisión en las detecciones.

Además de la detección de armas, la visión por computadora ha sido utilizada en diversas aplicaciones para el cumplimiento de la ley, como el reconocimiento facial [9] y la detección de matrículas de vehículos [10]. También se han explorado técnicas para detectar anomalías en el tráfico [11], demostrando cómo la visión por computadora puede reemplazar la supervisión humana en tareas como la detección y seguimiento de vehículos y peatones.

Estos trabajos muestran avances significativos en la detección de armas en videos de vigilancia, pero también revelan que aún existen desafíos importantes en cuanto a la precisión y la velocidad, lo que sugiere un considerable margen para mejoras futuras.

III. Conjunto de datos utilizado

Se llevó a cabo la recolección de múltiples conjuntos de datos con el objetivo de entrenar y posteriormente evaluar el sistema de detección de armas propuesto. Estos fueron elegidos por su diversidad y amplitud, lo que permitió abordar una amplia gama de situaciones y escenarios posibles. En este trabajo, se emplearon conjuntos de datos disponibles públicamente proporcionados por RoBoFlow [13].

En cuanto a la elección de un conjunto de datos, Ultralytics [14] ofrece recomendaciones que contribuyen a obtener los mejores resultados durante el entrenamiento de modelos YOLO (por sus siglas en inglés, *You Only Look Once*), una familia de modelos de detección de objetos en tiempo real. Algunas de estas sugerencias son:

- 1) Se recomienda tener un mínimo de 1500 imágenes por clase para asegurar una representación adecuada de los objetos de interés. Se aconseja contar con al menos 10000 instancias (objetos etiquetados) por clase.
- 2) La variedad de imágenes debe reflejar fielmente el entorno en el que se desplegará el modelo. Se sugiere utilizar imágenes de diferentes momentos del día, estaciones, condiciones climáticas, iluminaciones, ángulos y fuentes (obtenidas en línea, recolectadas localmente, cámaras diversas), entre otros.

3) Las imágenes de fondo, que no contienen objetos de interés, se pueden agregar al conjunto de datos para reducir Falsos Positivos (FP). Se sugiere incluir alrededor del 0-10% de imágenes de fondo (por ejemplo, COCO tiene 1000 imágenes de fondo, lo que representa el 1% del total). No es necesario etiquetar las imágenes de fondo.

Para los experimentos iniciales, se utilizó el conjunto de datos "Weapons" [15], el cual incluye etiquetas para dos categorías: "pistolas" y "cuchillos". Este conjunto está compuesto por más de 4.000 imágenes divididas en conjuntos de entrenamiento, validación y pruebas. Cuenta con 2.211 instancias de la clase "pistolas" y 2.091 de la clase "cuchillo". Es relevante mencionar que este conjunto de datos presenta algunas limitaciones, como la presencia de imágenes de pistolas en fondos blancos que no reflejan su contexto real. Un ejemplo de esto se puede ver en la primera imagen de la primera fila de la Fig. 2. A pesar de su tamaño y las limitaciones, se decidió usar este conjunto de datos para poder tener una primera impresión del rendimiento de los modelos.

En una fase posterior de experimentación, se tomaron en consideración las recomendaciones anteriores y se optó por utilizar un conjunto de datos más extenso [16]. Este conjunto consta de más de 14.000 imágenes, todas ellas pertenecientes a una única categoría: "gun" ("arma de fuego" en inglés). Se destaca la diversidad de resoluciones y formatos presentes en las imágenes, así como la variabilidad en escenarios, iluminación y poses, con el propósito de capacitar al modelo para manejar una amplia gama de situaciones y contextos. Además, se incluyeron imágenes que representan teléfonos celulares (como se muestra en la segunda imagen de la segunda fila de la Fig. 3) para evitar falsos positivos en la detección de armas de fuego.

IV. Detección de objetos con YOLO

Para el entrenamiento de modelos YOLO, se optó por utilizar Google Colaboratory como plataforma principal.



Fig. 2: Muestra de imágenes del conjunto de datos "Weapons".

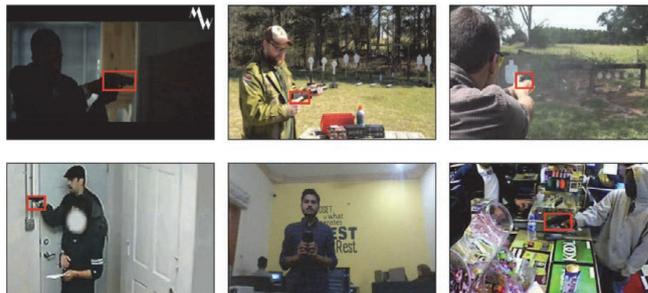


Fig. 3: Muestra de imágenes del conjunto de datos más extenso.

Esta elección se basó en la conveniencia de acceder a recursos computacionales significativos en la nube sin incurrir en costos excesivos.

Se llevaron a cabo experimentos utilizando tres versiones de YOLO (YOLOv5, YOLOv7 y YOLOv8), los cuales exploraron diferentes tamaños de imagen, cantidad de épocas y tasas de aprendizaje (*learning rate* en inglés). Se hicieron pruebas con y sin transferencia de aprendizaje (*transfer learning* en inglés).

Los primeros experimentos se entrenaron durante 50 épocas. Los resultados obtenidos se detallan en la Tabla I. A partir de estos experimentos, se pueden destacar las siguientes observaciones iniciales:

- En general, se observa que la transferencia de aprendizaje mejora el rendimiento en términos de precisión, *recall* y *F1-score*.
- El optimizador ADAM parece tener una mayor precisión en comparación con SGD.
- YOLOv8 tiende a superar a YOLOv5 y YOLOv7 en precisión, aunque YOLOv7 logra mejores resultados en *recall* y *mAP*.

Luego, se incrementó el número de épocas de entrenamiento de 50 a 100. Un mayor número de épocas podría permitir que los modelos aprendan patrones más complejos, lo que lleva a mejores resultados. Los resultados de estos experimentos se detallan en la Tabla II. De estos experimentos se realizaron las siguientes observaciones:

- No parece haber una diferencia significativa en las métricas con respecto a los experimentos realizados en 50 épocas.
- La tabla sugiere que tener un modelo más grande no garantiza necesariamente un mejor rendimiento. Por ejemplo, el modelo YOLOv8I, que tiene más parámetros que YOLOv8s, no presenta un mejor rendimiento.
- El tamaño de la imagen de entrada varió entre 640x640 y 800x800 píxeles en estos experimentos. No hubo una diferencia significativa en el rendimiento entre los dos tamaños de imagen, lo que sugiere que ambos son adecuados para estos

TABLA I: Tabla comparativa de experimentos realizados en 50 épocas.

Modelo	Dimensiones de la imagen (ancho x alto en pixeles)	Transferencia de aprendizaje	Tasa de aprendizaje	Optimizador	Precisión	Recall	F1-score	mAP@.5	mAP@.5:95
yolov5	800 x 800	no	0.010	SGD	0.737	0.625	0.338	0.701	0.358
yolov5	640 x 640	yolov5s	0.010	SGD	0.773	0.712	0.371	0.764	0.429
yolov5	800 x 800	yolov5s	0.010	SGD	0.796	0.716	0.377	0.755	0.428
yolov7	640 x 640	yolov7training	0.001	SGD	0.839	0.902	0.435	0.907	0.626
yolov8	800 x 800	no	0.010	SGD	0.842	0.744	0.395	0.841	0.579
yolov8	800 x 800	yolov8s	0.010	SGD	0.864	0.776	0.409	0.852	0.583
yolov8	640 x 640	yolov8s	0.001	Adam	0.868	0.782	0.411	0.857	0.595

modelos en particular.

- Diferentes versiones de YOLO presentan variaciones en el rendimiento según las métricas evaluadas. En términos de *recall*, YOLOv7 tiene un mejor rendimiento que YOLOv5 y YOLOv8. En términos de precisión, el mejor modelo se logró con YOLOv8.

En una segunda etapa de experimentación, se tenía que elegir un modelo para entrenar con el conjunto de datos más extenso, con una diversidad de escenarios, iluminación y poses. Idealmente, considerando las evaluaciones anteriores, la elección óptima hubiera sido YOLOv7. Sin embargo, debido a limitaciones de recursos en la plataforma de entrenamiento (Google Colaboratory), específicamente en términos de asignación de memoria, resultó inviable. Los intentos de entrenar los modelos en un conjunto de datos más grande resultaron en desbordamientos de memoria, lo que terminaba abruptamente el proceso de entrenamiento antes de que el modelo pudiera terminar la primera iteración del entrenamiento. Por lo tanto, la siguiente opción fue YOLOv8.

Con el segundo conjunto de datos, se llevaron a cabo una serie de experimentos variando algunos aspectos, incluyendo el tamaño de la imagen de entrada, que se probó con 640 y 800 pixeles; la utilización de *transfer learning* con los modelos *yolov8s*, *yolov8m* y *yolov8l* como base; las tasas de aprendizaje, oscilando entre 0,001 con el optimizador Adam y 0,01 con SGD; y finalmente, la duración del entrenamiento, con un rango de épocas que abarcó desde 40 hasta 300, explorando si un entrenamiento prolongado conduce a una mejora significativa en los resultados. Los resultados de la validación de estos modelos se ilustran en la Tabla III.

En la Tabla III, los modelos seleccionados para la optimización se distinguen por las filas resaltadas en gris. Se optó por elegir un conjunto diverso de modelos, abarcando diferentes tamaños y cantidades de épocas de entrenamiento. En particular, se incluyeron tres modelos de menor cantidad de épocas, representando cada uno de los tamaños disponibles (*yolov8s*, *yolov8m* y *yolov8l*) y dos modelos entrenados durante 100 épocas. La elección de estos modelos fue en base a las métricas precisión, *recall* y *mAP*.

V. Optimización de modelos con OpenVINO

Con el objetivo de mejorar la eficiencia del sistema, se procedió a la optimización de los modelos YOLO seleccionados mediante el uso de la herramienta OpenVINO (*Open Visual Inference and Neural Network Optimization*) [18]. OpenVINO es una plataforma de desarrollo de código abierto desarrollada por Intel que se utiliza para optimizar y acelerar la inferencia de modelos de redes neuronales en una variedad de hardware, incluidas las CPU de Intel, las GPU, los FPGAs y los dispositivos de visión de Intel. Ofrece una serie de herramientas y bibliotecas que permiten a los desarrolladores implementar modelos de aprendizaje automático y visión por computadora de manera eficiente en diferentes dispositivos y arquitecturas.

Una de las principales ventajas de OpenVINO es su capacidad para convertir modelos de aprendizaje automático entrenados en marcos como TensorFlow, PyTorch o ONNX a formatos optimizados, permitiendo su ejecución eficiente en hardware con arquitecturas X86, como las de Intel y AMD. Además, ofrece opciones de optimización para Apple Silicon y otras plataformas, lo que maximiza el rendimiento del hardware subyacente y se traduce en una inferencia más rápida y eficiente.

A. Exportar un modelo YOLO a OpenVINO

YOLOv8 cuenta con una funcionalidad de exportación en formato OpenVINO, que ofrece la posibilidad de lograr hasta un 3x de aceleración en la CPU, además de habilitar la aceleración en otros componentes de hardware de Intel, como iGPU, dGPU y VPU, entre otros. La exportación se realiza con el objetivo de optimizar los modelos y hacer que sean más compatibles y eficientes en el hardware subyacente, lo que resulta en una mejora significativa en el rendimiento de las inferencias.

Una de las ventajas más destacadas de esta optimización radica en la capacidad de alcanzar velocidades de inferencia similares a las obtenidas con GPU, pero utilizando la CPU. Esto se traduce en un ahorro sustancial de costos, especialmente en entornos en la nube como AWS, donde las instancias sin GPU suelen ser considerablemente más económicas. Esta optimización provee la flexibilidad de aprovechar al máximo la infraestructura

TABLA II: Tabla comparativa de experimentos realizados en 100 épocas.

Modelo	Dimensiones de la imagen (ancho x alto en pixeles)	Transferencia de aprendizaje	Tasa de aprendizaje	Optimizador	Precisión	Recall	F1-score	mAP@.5	mAP@.5:.95
yolov5	640 x 640	yolov5s	0.001	SGD	0.692	0.603	0.322	0.665	0.333
yolov5	800 x 800	no	0.010	SGD	0.793	0.689	0.369	0.747	0.422
yolov5	640 x 640	yolov5s	0.010	SGD	0.811	0.750	0.390	0.807	0.476
yolov8	640 x 640	yolov8s	0.001	Adam	0.852	0.796	0.412	0.849	0.616
yolov5	800 x 800	yolov5s	0.010	SGD	0.854	0.745	0.398	0.807	0.490
yolov7	800 x 800	yolov7training	0.010	SGD	0.856	0.808	0.416	0.858	0.583
yolov8	800 x 800	yolov8l	0.010	SGD	0.867	0.758	0.404	0.838	0.597
yolov7	640 x 640	yolov7training	0.001	SGD	0.870	0.904	0.443	0.914	0.640
yolov8	800 x 800	no	0.010	SGD	0.871	0.776	0.410	0.846	0.599
yolov7	640 x 640	yolov7training	0.010	SGD	0.874	0.849	0.431	0.881	0.624
yolov8	800 x 800	yolov8s	0.010	SGD	0.877	0.756	0.406	0.844	0.612
yolov8	640 x 640	yolov8s	0.010	SGD	0.888	0.777	0.414	0.860	0.623

TABLA III: Tabla comparativa de experimentos entrenados con el dataset más extenso.

Épocas	Dimensiones de la imagen (ancho x alto en pixeles)	Transferencia de aprendizaje	Tasa de aprendizaje	Optimizador	Precisión	Recall	F1-score	mAP@.5	mAP@.5:.95
40	800 x 800	yolov8s	0.010	SGD	0.910	0.862	0.443	0.928	0.705
50	640 x 640	yolov8l	0.001	Adam	0.916	0.831	0.436	0.909	0.704
50	640 x 640	yolov8m	0.010	SGD	0.925	0.874	0.449	0.938	0.731
50	640 x 640	yolov8s	0.001	Adam	0.925	0.836	0.439	0.915	0.705
50	640 x 640	yolov8l	0.010	SGD	0.927	0.880	0.451	0.944	0.733
60	800 x 800	yolov8s	0.010	SGD	0.931	0.879	0.452	0.940	0.723
80	800 x 800	yolov8s	0.010	SGD	0.931	0.879	0.452	0.938	0.727
100	640 x 640	yolov8s	0.001	Adam	0.916	0.873	0.447	0.927	0.717
100	800 x 800	yolov8s	0.010	SGD	0.922	0.883	0.451	0.938	0.728
100	640 x 640	yolov8m	0.010	SGD	0.926	0.891	0.454	0.941	0.743
300	640 x 640	yolov8s	0.010	SGD	0.933	0.898	0.458	0.946	0.731

existente o utilizar recursos de menor costo sin comprometer la velocidad y la eficiencia de las inferencias.

B. Optimización con cuantización de 8 bits

La cuantización consiste en representar números de punto flotante, que normalmente utilizan 32 o 64 bits, con tan solo 8 o 16 bits, redondeando los valores reales. Esto significa que los valores reales se redondean y se almacenan con menos bits de los que originalmente tenían.

La cuantización de 8 bits es especialmente útil en dispositivos con recursos limitados, como dispositivos móviles o sistemas embebidos, donde se busca reducir la sobrecarga computacional y de memoria. Aunque esta reducción de precisión puede causar una ligera pérdida de precisión en la inferencia de los modelos, en muchos casos, esta pérdida es aceptable y los modelos cuantizados de 8 bits aún pueden realizar tareas con un rendimiento satisfactorio.

NNCF (*Neural Network Compression Framework*) [19] ofrece un conjunto de algoritmos avanzados para la optimización de la inferencia de redes neuronales en OpenVINO con un mínimo de pérdida de precisión. Es posible utilizar la cuantización de 8 bits en el modo de

post-entrenamiento (sin necesidad del proceso de *fine-tuning*) para optimizar modelos YOLO

C. Resultados de la Optimización

Se exportaron los modelos YOLOv8 seleccionados en el formato OpenVINO con punto flotante y, posteriormente, estos modelos fueron sometidos a un proceso de optimización mediante la cuantización utilizando NNCF.

Para la implementación de la cuantización con NNCF se empleó la función `nncf.quantize()` con el modelo de PyTorch. En este proceso, se aplicó la configuración MIXED de NNCF, que se utiliza para aplicar cuantización de precisión mixta al modelo, utilizando tanto tipos de datos de 8 bits como de 16 bits. El resultado de esta función es un modelo cuantizado que puede utilizarse para realizar inferencias.

Para llevar a cabo las evaluaciones de las optimizaciones, se utilizó una computadora con las siguientes especificaciones de *hardware*:

- Arquitectura: x86_64.
- Sistema operativo: Ubuntu 22.04.2 LTS.
- Memoria: 16 GB.
- CPU: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz x 4.
- Capacidad de disco: 480 GB.

Los resultados de las optimizaciones con respecto al conjunto de datos extenso, se detallan en la Tabla IV. A pesar de que se observó que la precisión y el *recall* se mantuvieron en niveles similares a los obtenidos antes de la optimización, se notó un aumento en la precisión de los modelos *yolov8m* después de la optimización. Esto podría deberse a que los modelos originales estaban experimentando *overfitting*.

Uno de los aspectos más destacados fue la mejora significativa en el rendimiento en términos de velocidad. La velocidad de procesamiento de frames por segundo (FPS) experimentó un aumento notable con la técnica Quantized. Por otro lado, en el caso de Floating-Point, se observó una mejora en la velocidad para los modelos más pequeños, aunque los modelos *yolov8l* y *yolov8m* experimentaron una disminución en este aspecto.

VI. Seguimiento de objetos

El seguimiento de objetos (*object tracking* en inglés) es una técnica de visión por computador que implica detectar y seguir objetos en imágenes fijas o videos. Esta tarea se utiliza en diversas aplicaciones como vigilancia, robótica, vehículos autónomos y realidad aumentada. El objetivo es estimar o predecir el movimiento de los objetos a lo largo del tiempo.

El proceso de seguimiento de objetos generalmente consta de dos sub-tareas principales: detección de objetos y reidentificación de objetos entre cuadros. Comienza con la detección de los objetos de interés en el cuadro inicial utilizando algoritmos de detección de objetos. Una vez que los objetos son detectados, el algoritmo realiza el seguimiento de sus movimientos en cuadros sucesivos, manteniendo sus identidades y actualizando sus posiciones. Esto permite crear una trayectoria o ruta para cada objeto a lo largo del tiempo.

A. Seguimiento de personas con ByteTrack

Los métodos de seguimiento de objetos múltiples (MOT, por sus siglas en inglés) emplean el filtro de Kalman para anticipar la posición de los "tracklets" en el siguiente fotograma. Luego, fusionan las cajas de predicción con las cajas de detección para mejorar los resultados de detección. En su mayoría, estos métodos conservan únicamente las cajas de detección con puntuaciones elevadas, superando un umbral específico, como 0.5, y las utilizan como base para asociar los "tracklets" a las cajas de detección, emparejándolos según su similitud.

Sin embargo, el seguimiento de objetos presenta desafíos, como las ocultaciones que pueden causar la pérdida de detección, el desenfoque de movimiento que conduce a trayectorias fragmentadas y cambios en la identificación debido a objetos que entran y salen del encuadre, y los cambios de tamaño que también pueden provocar pérdida de trayectorias y cambios de identidad. La Fig. 4 (a) y (b) muestran este problema. Primero se

inician tres "tracklets" diferentes ya que sus puntuaciones son todas superiores a 0.5. Sin embargo, cuando ocurre una ocultación, la puntuación de detección correspondiente al "tracklet" rojo disminuye de 0.8 a 0.4 y luego de 0.4 a 0.1. Estas cajas de detección son eliminadas por que no pasan umbral y, en consecuencia, el "tracklet" rojo desaparece.

Para abordar estos desafíos, se ha desarrollado ByteTrack [17]. ByteTrack conserva todas las cajas de detección y las separa en dos grupos: las de alta puntuación y las de baja puntuación. Este enfoque funciona en dos etapas. En la primera, se emparejan las cajas de detección de alta puntuación con los "tracklets" basándose en la similitud de movimiento o de apariencia. Esto se logra mediante el uso del filtro de Kalman para predecir la ubicación de los "tracklets" en el siguiente fotograma, y se calcula la similitud utilizando la intersección sobre unión (IoU) o la distancia de características de re-identificación entre las cajas predichas y las de detección. En la segunda etapa, se realiza un segundo emparejamiento entre los "tracklets" no emparejados y las cajas de detección de baja puntuación, nuevamente utilizando similitud de movimiento como criterio. En la Fig. 4, en (c), se ilustra como dos cajas de detección de baja puntuación se emparejan con los "tracklets" mediante las cajas predichas por el modelo de movimiento y los objetos se recuperan correctamente.

B. Implementación del seguimiento de personas.

Para el seguimiento de personas, se utilizó el repositorio de ByteTrack. Tras clonar el repositorio, se configuró

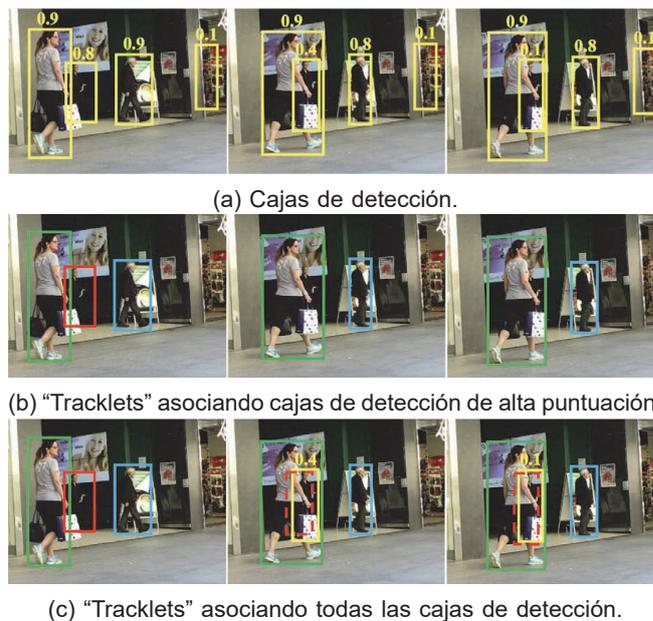


Fig. 4: Seguimiento de objetos con ByteTrack. Imagen tomada de [17].

TABLA IV: Resultados de las optimizaciones con OpenVINO.

Transferencia de aprendizaje	Dimensiones de la imagen (ancho x alto en pixeles)	Épocas	Precisión	Recall	mAP@.5	mAP@.5:.95	Frames por segundo	Optimización
yolov8s	800 x 800	100	0.924	0.868	0.926	0.691	8.350	Cuantización
yolov8s	800 x 800	100	0.927	0.867	0.925	0.702	4.850	Floating-point
yolov8s	800 x 800	100	0.922	0.883	0.938	0.728	3.167	Sin Optimización
yolov8s	800 x 800	60	0.927	0.873	0.929	0.690	8.180	Cuantización
yolov8s	800 x 800	60	0.922	0.873	0.931	0.704	4.760	Floating-point
yolov8s	800 x 800	60	0.931	0.879	0.940	0.723	3.249	Sin Optimización
yolov8m	640 x 640	100	0.939	0.871	0.932	0.713	3.180	Cuantización
yolov8m	640 x 640	100	0.941	0.873	0.934	0.728	1.790	Floating-point
yolov8m	640 x 640	100	0.926	0.891	0.941	0.743	2.151	Sin Optimización
yolov8m	640 x 640	50	0.939	0.854	0.933	0.710	3.140	Cuantización
yolov8m	640 x 640	50	0.936	0.860	0.932	0.721	1.760	Floating-point
yolov8m	640 x 640	50	0.925	0.874	0.938	0.731	2.131	Sin Optimización
yolov8l	640 x 640	50	0.920	0.883	0.937	0.712	1.540	Cuantización
yolov8l	640 x 640	50	0.921	0.889	0.941	0.723	0.800	Floating-point
yolov8l	640 x 640	50	0.927	0.880	0.944	0.733	1.145	Sin Optimización

el entorno de Python e instaló ByteTrack en modo de desarrollo, permitiendo su fácil personalización e importación en scripts de Python.

El flujo de seguimiento de objetos se desarrolla de manera sencilla. Primero, las detecciones se convierten en cajas delimitadoras que rodean los objetos detectados. Luego, utilizando estas cajas delimitadoras y el tamaño de la imagen, se actualizan las pistas de seguimiento mediante la función "byte_tracker.update()". Posteriormente, se asocian las detecciones con las pistas mediante la función "match_detections_with_tracks()", asignando un ID único a cada detección en el video. Para seguir a las personas de interés, se utiliza una lista de IDs de interés, y finalmente, las detecciones se filtran y se conservan aquellas relacionadas con IDs de seguimiento que se encuentren en dicha lista. Este proceso permite un seguimiento específico de las personas detectadas en el video.

Para realizar el seguimiento de personas portando armas, se contempló filtrar la lista de IDs con dos enfoques diferentes: (1) el seguimiento de la persona más cercana al arma y (2) el seguimiento de la persona con la mano más cercana al arma. Ambos enfoques son ilustrados en la Fig. 5.

1) *Seguimiento de la persona más cercana al arma:* Inicialmente, se adoptó una estrategia que implicaba rastrear a la persona más cercana a la ubicación del arma detectada. Para lograr esto, se utilizó un modelo de detección de objetos YOLOv8, preentrenado para detectar personas, y se combinó con el flujo de ByteTrack armado previamente. Para identificar a las personas de interés, se realizaba un cálculo para determinar cuáles de ellas tenían cajas delimitadoras que intersectaban con la caja delimitadora del arma detectada. Esta metodología permitía la identificación y el seguimiento constante de la persona más próxima al arma a medida que avanzaba

el video.

Sin embargo, un desafío importante surgió al implementar este enfoque. Si el individuo portando el arma se acercaba peligrosamente a otra persona o, por ejemplo, apuntaba el arma hacia alguien más, la persona más cercana al arma no necesariamente era el perpetrador, sino potencialmente la víctima. Como resultado, tanto el agresor (quien inicialmente tenía el arma más cerca de él) como la víctima, se identificaban como personas de



(a) Seguimiento de personas cercanas al arma.



(b) Seguimiento de la persona con la mano más cercana al arma.

Fig. 5: Enfoques para seguimiento de personas.

interés y se realizaba un seguimiento continuo de ambas personas en la secuencia. En la imagen (a) de la Fig. 5 se ilustra cómo en esa situación se hace el seguimiento de la persona que porta el arma, pero también de otra que no, pero que se encuentra cercana a ella.

2) *Seguimiento de la persona con la mano más cercana al arma*: El segundo enfoque se centró en el seguimiento de la persona cuya mano estuviera más cercana al arma detectada. Para implementar esto, se utilizó un modelo para estimación de pose de Ultralytics, llamado "YOLOv8-pose". Este modelo se encarga de calcular 17 puntos de interés clave en el cuerpo de una persona, incluyendo 2 puntos de interés que representan las ubicaciones de las muñecas. Utilizando esta información, se identifica la muñeca más cercana a la caja delimitadora del arma detectada, y se identifica a esa persona como persona de interés para el seguimiento con ByteTrack.

Se llevaron a cabo pruebas con videos que presentaban la misma problemática, y se observó una mejora notable en la precisión del seguimiento. Este resultado condujo a una elección clara a favor del segundo enfoque como la opción más efectiva para llevar a cabo el seguimiento de personas. El esquema del sistema con este enfoque se muestra en detalle en la Fig. 6.

VII. Validación de los modelos

Para realizar pruebas que simularan situaciones de la vida real, se utilizaron imágenes del conjunto de datos "Randomized Clips" [20], el cual consta de fotogramas de videos de cámaras de seguridad obtenidos de diversas fuentes públicas en internet, como YouTube. En la Fig. 7 se ilustra una muestra de las imágenes de este conjunto de datos.

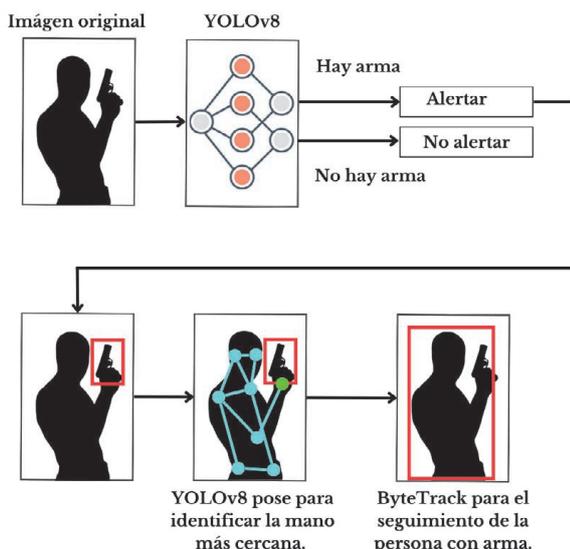


Fig. 6: Esquema del sistema de detección de personas.

Los modelos YOLOv8 previamente entrenados se sometieron a una evaluación utilizando el conjunto de datos para las pruebas, "Randomized Clips". Los resultados de estas evaluaciones se encuentran detallados en la Tabla V.

Inicialmente, se estableció el objetivo de lograr una precisión del 80% en la detección de objetos dentro del conjunto de datos seleccionado, un hito que se alcanzó con éxito durante la fase de entrenamiento. Sin embargo, al llevar a cabo pruebas en situaciones del mundo real, específicamente utilizando fotogramas de videos capturados por cámaras de seguridad, se observó una disminución significativa en el rendimiento del modelo. En este escenario más desafiante, el modelo solo logró una precisión del 58%, lo que puso de manifiesto la importancia de evaluar y adaptar los modelos en entornos que reflejen de manera más precisa las condiciones del mundo real.

Adicionalmente, se llevaron a cabo evaluaciones de los modelos optimizados con OpenVINO sobre el mismo conjunto de datos. Los resultados de estas evaluaciones se encuentran en la Tabla VI. Tras analizar los datos obtenidos en términos de precisión y tiempo de inferencia, se tomó una decisión para seleccionar el modelo final del sistema. Aunque se exploraron diversas opciones, se optó por utilizar el modelo *yolo8s*, entrenado con un tamaño de imagen de 800 píxeles durante 60 épocas (con tasa de aprendizaje 0.01 y SGD). Esta elección se basó en un equilibrio estratégico, ya que si bien existen modelos más rápidos, sacrifican precisión, y, por otro lado, el modelo *yolo8l*, aunque es más preciso, se desempeña considerablemente más lento.

VIII. Simulaciones y resultados

En el desarrollo de las simulaciones de este trabajo, se utilizaron videos públicos disponibles en la plataforma de YouTube. Estos videos proporcionaron una variedad de situaciones con armas y escenarios del mundo real que fueron fundamentales para evaluar el funcionamiento del sistema de detección de armas y seguimiento de personas propuesto. Es importante destacar que se utilizaron videos públicos, respetando la legalidad y ética en la recolección de datos.



Fig. 7: Muestra de imágenes del conjunto de datos "Randomized Clips".

TABLA V: Validación de los modelos sobre "Randomized Clips".

Épocas	Dimensiones de la imagen (ancho x alto en píxeles)	Transferencia de aprendizaje	Tasa de aprendizaje	Optimizador	Precisión	Recall	F1-score	mAP@.5	mAP@.5:.95
50	640 x 640	yolov8s	0.001	Adam	0.307	0.302	0.152	0.214	0.083
50	640 x 640	yolov8m	0.010	SGD	0.331	0.345	0.169	0.285	0.114
50	640 x 640	yolov8l	0.001	Adam	0.352	0.276	0.155	0.210	0.080
40	800 x 800	yolov8s	0.010	SGD	0.372	0.327	0.174	0.268	0.110
100	640 x 640	yolov8s	0.001	Adam	0.435	0.286	0.173	0.269	0.102
50	640 x 640	yolov8l	0.010	SGD	0.481	0.347	0.202	0.309	0.130
300	640 x 640	yolov8s	0.010	SGD	0.509	0.311	0.193	0.325	0.137
80	800 x 800	yolov8s	0.010	SGD	0.560	0.302	0.196	0.310	0.123
100	640 x 640	yolov8m	0.010	SGD	0.576	0.335	0.212	0.366	0.145
100	800 x 800	yolov8s	0.010	SGD	0.578	0.302	0.198	0.314	0.124
60	800 x 800	yolov8s	0.010	SGD	0.581	0.293	0.195	0.318	0.125

TABLA VI: Evaluación de los modelos optimizados con OpenVINO sobre el conjunto de datos "Randomized Clips".

Transferencia de aprendizaje	Dimensiones de la imagen (ancho x alto en píxeles)	Épocas	Precisión	Recall	mAP@.5	mAP@.5:.95	Frames por segundo	Optimización
yolov8s	800 x 800	100	0.274	0.273	0.181	0.072	8.390	Cuantización
yolov8s	800 x 800	100	0.293	0.283	0.201	0.081	4.920	Floating-point
yolov8s	800 x 800	100	0.578	0.302	0.314	0.124	3.168	Sin Optimización
yolov8s	800 x 800	60	0.311	0.227	0.183	0.074	8.230	Cuantización
yolov8s	800 x 800	60	0.308	0.216	0.196	0.079	4.810	Floating-point
yolov8s	800 x 800	60	0.581	0.293	0.318	0.125	3.250	Sin Optimización
yolov8m	640 x 640	100	0.423	0.343	0.281	0.108	3.190	Cuantización
yolov8m	640 x 640	100	0.439	0.338	0.279	0.110	1.800	Floating-point
yolov8m	640 x 640	100	0.576	0.335	0.366	0.145	2.151	Sin Optimización
yolov8m	640 x 640	50	0.362	0.325	0.251	0.100	3.150	Cuantización
yolov8m	640 x 640	50	0.344	0.354	0.251	0.100	1.780	Floating-point
yolov8m	640 x 640	50	0.331	0.345	0.285	0.114	2.132	Sin Optimización
yolov8l	640 x 640	50	0.570	0.276	0.318	0.133	1.550	Cuantización
yolov8l	640 x 640	50	0.470	0.326	0.323	0.136	0.880	Floating-point
yolov8l	640 x 640	50	0.481	0.347	0.309	0.130	1.145	Sin Optimización

Se pueden observar notables diferencias en las detecciones realizadas por los modelos, tanto aquellos sin optimizaciones como los optimizados con cuantización. Para los modelos sin optimizaciones, se destaca la buena precisión en muchas situaciones, lo que demuestra su capacidad para detectar armas de fuego en diversas condiciones.

Es importante señalar que en casos como las imágenes (c) y (g) de la Fig. 8, donde la calidad de la imagen es baja o los objetos de interés están parcialmente ocultos, se pueden encontrar limitaciones en la detección, y algunas armas pueden no ser detectadas. En el caso de las detecciones realizadas con el modelo optimizado mediante cuantización, se observan algunas diferencias en la precisión. Por ejemplo, en la imagen (d), se puede apreciar que parte del fondo es erróneamente detectado como un arma, lo que indica ciertas limitaciones en la segmentación de objetos. Además, en la imagen (b), donde se realiza el seguimiento de las víctimas en vez del seguimiento del perpetrador, es que el *tracker* se ve influenciado por detecciones previas de elementos cercanos que fueron interpretados como armas errónea-

mente. Esto sugiere que, si bien la optimización con cuantización puede mejorar la eficiencia del modelo, también puede introducir desafíos en la interpretación de las detecciones en ciertas situaciones.

Estos resultados resaltan la importancia de considerar el contexto y las condiciones en las que se utilizará el sistema.

Se puede ver en la Tabla VII que se logró una mejora significativa en la velocidad de inferencia mediante técnicas de optimización, siendo la optimización con cuantización la que obtuvo los mejores resultados.

IX. Conclusiones

Durante el desarrollo de este trabajo, se llevaron a cabo pruebas con diferentes modelos YOLO y diversas técnicas de optimización. El sistema desarrollado detecta armas de fuego en videos y proporciona un porcentaje de confianza para cada detección y las coordenadas correspondientes. Asimismo, realiza el seguimiento de la persona que porta el arma en el video.

A pesar de los avances logrados, el sistema alcanzó una precisión máxima de 57% en las pruebas realizadas, lo que indica la necesidad de mejorar la robustez de los

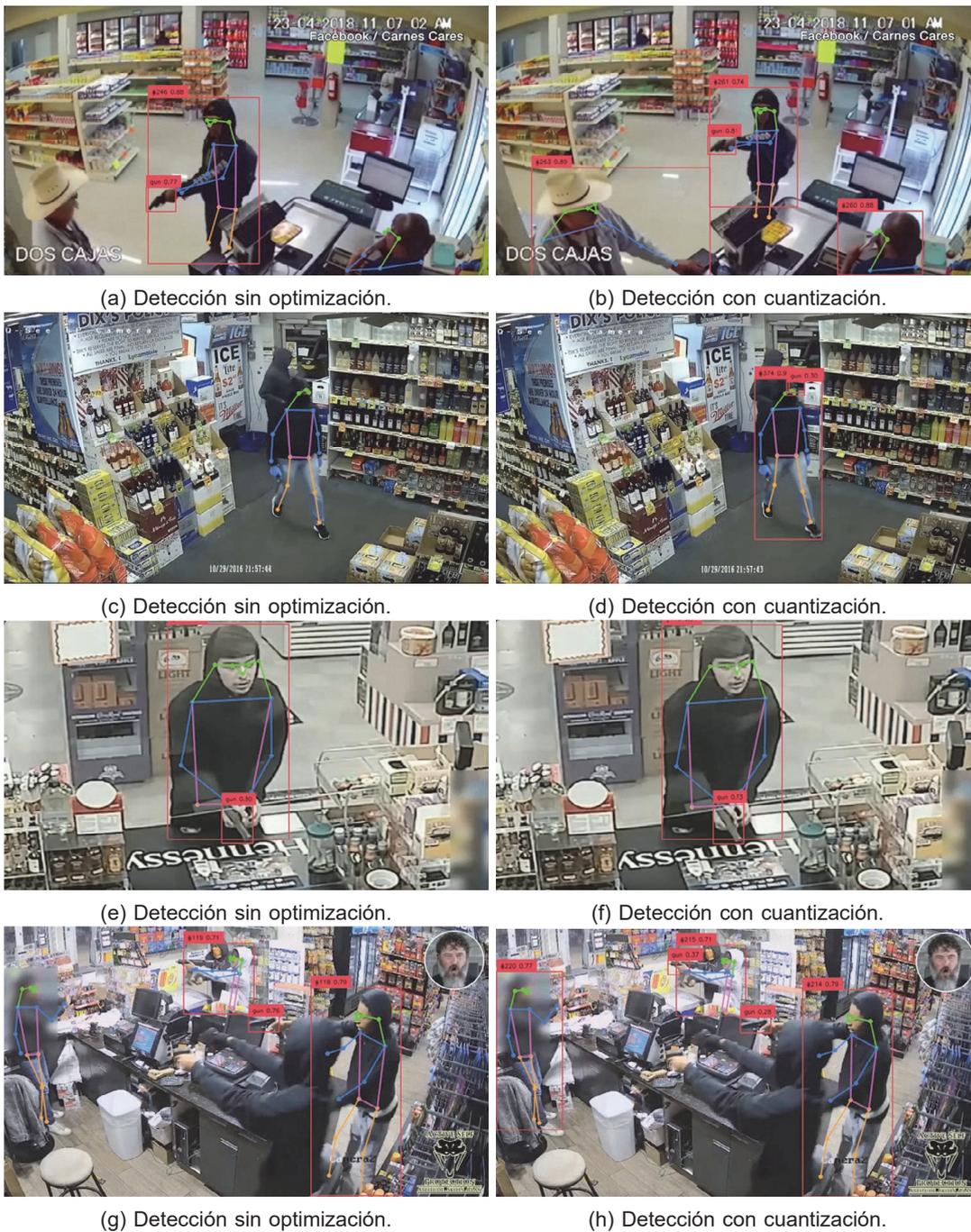


Fig. 8: Resultados de experimentos con y sin optimización sobre videos.

TABLA VII: Tiempos de detección de las simulaciones.

	Video 1	Video 2	Video 3	Video 4	Video 5	Video 6
Sin Optimización (milisegundos)	262.291562	270.323627	275.796037	268.530946	255.931583	247.127344
Sin Optimización (Frames por Segundo)	3.812551	3.699270	3.625868	3.723966	3.907294	4.046497
Con Floating-Point (milisegundos)	220.404815	221.152798	222.698712	219.912711	219.534242	220.103649
Con Floating-Point (Frames por Segundo)	4.537106	4.521761	4.490372	4.547259	4.555098	4.543314
Con Cuantización (milisegundos)	140.002001	140.353079	141.314379	138.411628	139.687376	140.171080
Con Cuantización (Frames por Segundo)	7.142755	7.124888	7.076421	7.224827	7.158843	7.134139

modelos en condiciones reales. Aunque se logró reducir el tiempo de inferencia por debajo de 300 milisegundos, se observó que algunos modelos sacrifican precisión por velocidad, subrayando la importancia de seguir optimizando este equilibrio.

Para futuros trabajos, es fundamental la adquisición de conjuntos de datos que representen mejor los escenarios reales en los que se desplegará el sistema. Esto permitirá obtener métricas más fiables y mejorar el rendimiento de los modelos en el conjunto de pruebas. Otro paso importante será explorar nuevas técnicas de optimización, como Apache TVM, con el objetivo de ver si se logra una mejora en reducir los tiempos de inferencia y lograr una respuesta más rápida del sistema en situaciones de detección en tiempo real.

Referencias

- [1] M. I. R. . Advisory., "Mercado de sistemas de videovigilancia: crecimiento, tendencias, impacto de covid-19 y pronósticos (2022 - 2027)." <https://www.mordorintelligence.com/es/industry-reports/global-video-surveillance-market-industry>, 06 2023.
- [2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.
- [3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.
- [4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.
- [5] R. K. Tiwari and G. K. Verma, "A computer vision based framework for visual gun detection using harris interest point detector," *Procedia Computer Science*, vol. 54, pp. 703–712, 2015, eleventh International Conference on Communication Networks, ICCN 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Data Mining and Warehousing, ICDMW 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Image and Signal Processing, ICISP 2015, August 21-23, 2015, Bangalore, India. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050915014076>
- [6] R. Olmos, S. Tabik, and F. Herrera, "Automatic handgun detection alarm in videos using deep learning," *Neurocomputing*, vol. 275, pp. 66–72, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231217308196>
- [7] J. Lim, M. I. Al Jobayer, V. M. Baskaran, J. M. Lim, K. Wong, and J. See, "Gun detection in surveillance videos using deep neural networks," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2019, pp. 1998–2002.
- [8] D. Qi, W. Tan, Z. Liu, Q. Yao, and J. Liu, "A dataset and system for real-time gun detection in surveillance video using deep learning," 2021.
- [9] K. J. P. Lakshmi, T. K. Kumar, and S. Warriar, "Automated face recognition by smart security system using ai & ml algorithms," *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 1363–1368, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:235616476>
- [10] K. Natalya and C. Sergei, "Software for car license plates recognition with minimal computing resources," *2022 24th International Conference on Digital Signal Processing and its Applications (DSPA)*, pp. 1–4, 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:249658567>
- [11] K. K. Santhosh, D. P. Dogra, and P. P. Roy, "Anomaly detection in road traffic using visual surveillance: A survey," *ACM Comput. Surv.*, vol. 53, no. 6, dec 2020. [Online]. Available: <https://doi.org/10.1145/3417989>
- [12] A. S. A. AL-SLEMANI and A. ZENGİN, "A new surveillance and security alert system based on real-time motion detection," *Journal of Smart Systems Research*, vol. 4, no. 1, pp. 31–47, 2023.
- [13] B. Dwyer and J. Nelson, "Roboflow (version 1.0)," 2022, computer vision. [Online]. Available: <https://roboflow.com>
- [14] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolov8," 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [15] ThreatDetection, "weapons dataset," <https://universe.roboflow.com/threatdetection/weapons-h9w1e>, jul 2022, visited on 2023-08-23. [Online]. Available: <https://universe.roboflow.com/threatdetection/weapons-h9w1e>
- [16] S. Salman, "proj3 dataset," <https://universe.roboflow.com/saad-salman/proj3-hbkyr>, mar 2022, visited on 2023-08-23. [Online]. Available: <https://universe.roboflow.com/saad-salman/proj3-hbkyr>
- [17] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," 2022.
- [18] Opencv™ toolkit. <https://github.com/opencv/opencv>.
- [19] A. Kozlov, I. Lazarevich, V. Shamporov, N. Lyalyushkin, and Y. Gorbachev, "Neural network compression framework for fast model inference," *CoRR*, vol. abs/2002.08679, 2020. [Online]. Available: <https://arxiv.org/abs/2002.08679>
- [20] P. Lavrador, "randomized clips dataset," <https://universe.roboflow.com/phillip-lavrador/randomized-clips>, mar 2022, visited on 2023-08-23. [Online]. Available: <https://universe.roboflow.com/phillip-lavrador/randomized-clips>



Diseño de sistemas difusos para FPGAs desde Labview*

Eduardo Garcia

Yazmin Maldonado

Tecnológico Nacional de México/I.T.
Tijuana

Tecnológico Nacional de México/I.T.
Tijuana

Departamento de Ingeniería Eléctrica
y Electrónica

Posgrado en Ciencias de la
Ingeniería, Maestría y Doctorado

Tijuana, B.C., México

Tijuana, B.C., México

eduardogarcia04@gmail.com

yaz.maldonado@tectijuana.edu.mx

Resumen— Los sistemas de inferencia difusos (FIS) son cada más utilizados en diferentes áreas del conocimiento como la inteligencia artificial, control, procesamiento de datos, cómputo en la nube, entre otros. Actualmente existen diversas herramientas para diseñarlos en software, mientras que existen muy pocas para diseñar sistemas difusos en hardware. En este trabajo de investigación se hace una propuesta de una herramienta para el diseño de FIS en FPGAs (Field Programmable Gate Array) mediante el software de Labview. En esta herramienta se eligen mediante una interfaz gráfica las variables y etiquetas lingüísticas del FIS, así como el método de inferencia y defuzzificación. La herramienta se diseñó y programó en entorno Labview, además fue validada mediante la programación del FPGA-Elvis, los resultados se compararon contra la versión del FIS diseñada en Matlab, queda como trabajo pendiente la integración a la interfaz gráfica de la elección de las reglas difusas.

Palabras clave: FPGA, sistemas difusos, hardware, Labview, Matlab.

I. INTRODUCCIÓN

La lógica difusa es parte de la inteligencia artificial que se ha convertido en una alternativa para abordar problemas en los que la toma de decisiones juega un papel crucial [1].

Lofti A. Zadeh es el padre de la lógica difusa, en esta teoría introdujo el concepto de conjunto difuso (*fuzzy set*) bajo la idea de que el pensamiento humano se construye con etiquetas lingüísticas y no con valores numéricos. La lógica difusa permite representar conocimiento en un lenguaje matemático formal a través de la teoría de conjuntos difusos [2].

La teoría de conjuntos difusos dio origen a los sistemas de inferencia difusos (FIS), los FIS permiten la toma de decisiones basados en información imprecisa, en este artículo se basa en los FIS tipo Mamdani. Los FIS están organizados en tres etapas, fuzificación, inferencia y defuzificación.

A. Fuzificación

En la fuzificación se transforman los valores de entrada numéricos o *crisp* en valores difusos mediante funciones de membresía (FM). La conversión de un valor de entrada numérico le corresponde un valor lingüístico asociado a un grado de membresía, los grados de membresía representan un valor de pertenencia a un conjunto difuso tomando un valor en el intervalo $[0,1]$.

Los subconjuntos difusos se representan mediante FM, los cuales se agrupan en conjunto de datos de entrada, cada variable de entrada (variable lingüística) al sistema difuso puede tener varias FM que definen a cada valor lingüístico (etiqueta lingüística). Existen

diferentes tipos de FM tales como gaussiana, trapezoidal, triangular, singleton, campana, sigmoideal, entre otras, siendo las FM más utilizadas en hardware las triangulares y trapezoidales [3].

B. Inferencia

En esta etapa se encuentra la base de reglas y el motor de inferencia, la base de reglas contiene el conocimiento del sistema en forma de reglas IF-THEN, las cuales relacionan las variables de entrada con las salidas, mientras que en el motor de inferencia evalúa cada regla de la base de conocimiento. Las reglas difusas se evalúan de acuerdo a los resultados de la etapa de fuzificación, en este trabajo se utiliza el método de inferencia mínimo para la operación de implicación, el cual elige el valor mínimo de cada fuerza de disparo, así como el método de inferencia máximo el cuál elige el valor máximo de la fuerza de disparo [4].

C. Defuzificación

En esta etapa se convierte el valor difuso en un *crisp* mediante diferentes métodos de defuzificación, tales como centroide, máximo, alturas, entre otros. En este trabajo se utilizará el método de las alturas, el cual realiza la sumatoria desde 1 hasta el *n-th* subconjunto difuso del valor máximo del consecuente multiplicado por la fuerza de disparo y el resultado se divide entre la sumatoria desde 1 hasta el *n-th* subconjunto difuso de la fuerza de disparo [5].

Los FIS ofrecen una forma práctica de modelar sistemas complejos para toma de decisiones con datos inciertos, los FIS tienen alta capacidad de manejar información imprecisa y sus aplicaciones cada vez son más como control [6], clasificación [7], energía [8], salud [9], entre otros, por lo que es necesario contar con herramientas que permitan el diseño, simulación e implementación de sistemas difusos en software o en dispositivos electrónicos.

Por otra parte, los FPGAs (Field Programmable Gate Array) son dispositivos electrónicos que tienen la capacidad de procesar información en paralelo además de ser dispositivos de uso dedicado, lo que significa que aceleran

significativamente la ejecución de algoritmos complejos. Diseñar lenguajes de descripción de hardware para modelar circuitos digitales en FPGAs requiere conocimiento especializado, por lo tanto, diseñar herramientas para utilizarlos y aprovechar sus ventajas es crucial.

El artículo se encuentra organizado de la siguiente manera. En la Sección II, se hace una revisión del estado del arte sobre herramientas para diseñar sistemas de inferencia difusos, en la Sección III, se detalla el diseño del FIS en Labview para FPGAs, mientras que en la Sección IV, se explica el diseño de la interfaz gráfica en Labview, la validación de la herramienta propuesta en este artículo se analiza en la Sección V, finalmente, en la Sección VI se dan las conclusiones y el trabajo pendiente.

II. HERRAMIENTAS PARA DISEÑAR FIS

Actualmente existen varias herramientas para el diseño de FIS en software tales como:

FISDeT. Es una herramienta de desarrollo de sistemas de inferencia difusos, es una herramienta para el diseño de FIS mediante código abierto basado en lenguaje Python [10]. Los aspectos clave de esta herramienta, son la facilidad de uso y la interactividad, que permiten al usuario definir y modificar rápidamente la base de conocimientos difusa mediante una interfaz gráfica sencilla e intuitiva.

FisPro. Es una herramienta para sistemas de inferencia difusos profesional, este software permite crear FIS, mediante código abierto el cual es dividido en una biblioteca de funciones escritas en C++, que se puede utilizar de forma independiente y la interfaz gráfica Java, que implementa la mayoría de las funcionalidades de la biblioteca C++ [11].

jFuzzyLogic. Es una biblioteca de lógica difusa de código abierto que implementa estándares de la industria para simplificar el desarrollo de sistemas difusos, está escrito en Java. jFuzzyLogic implementa lenguaje de control difuso, así como una biblioteca completa que simplificará en gran medida su trabajo de investigación o desarrollo de lógica difusa [12].

Fuzzycreator. Es un conjunto de herramientas de código abierto GNU (licencia publica general), multiplataforma y basado en Python

que facilita la construcción de conjuntos difusos de tipo 1 y tipo 2 automáticamente a partir de datos, y permite el análisis de estos conjuntos difusos a través de medidas, como la similitud y la distancia [13].

Juzzy Online. Es una herramienta para el desarrollo de sistemas de inferencia difuso de tipo 1 y tipo 2 en Java, disponible gratuitamente e independiente de la plataforma, de manera que permite que las aplicaciones se ejecuten en diferentes plataformas de hardware y sistemas operativos [14].

Fuzzy logic toolbox. Ofrece aplicaciones y funciones de Matlab, y un bloque de Simulink para analizar, diseñar y simular sistemas de lógica difusa. Permite especificar y configurar entradas, salidas, funciones de membresía y reglas de sistemas de inferencia difusa de tipo 1 y tipo 2 [15].

Diseñador de sistemas difusos. Labview cuenta con una herramienta para diseñar y controlar sistemas difusos de forma interactiva a través de su código gráfico. Se puede implementar el control basado en reglas para los sistemas que requieren control con varias entradas [16].

Por otro lado, se cuenta con herramientas para el diseño de FIS en hardware, tales como:

XFUZZY. Tiene un entorno en modo gráfico que facilita el diseño de FIS. Está desarrollado en Java y, por tanto, puede ser ejecutado en cualquier plataforma, además, su interfaz de usuario ofrece versatilidad, el FIS diseñado en esta herramienta se puede implementar en un FPGA [17].

Fuzzy hardware tool. Es una propuesta para principiantes o expertos en el área de sistemas difusos. Con esta propuesta, los diseñadores de FIS podrán interactuar a través de una pantalla multitáctil con un FIS implementado en un sistema embebido con FPGA. Fuzzy Hardware Tool es un entorno didáctico para diseñar, simular e implementar un FIS de tipo Mamdani. La herramienta le permite seleccionar y configurar entradas, salidas, funciones de pertenencia y reglas [18].

Las herramientas de Fuzzy logic toolbox de Matlab y Diseñador de sistemas difusos de

Labview fueron seleccionadas para comparar los resultados contra la propuesta que se hace en este artículo.

III. DISEÑO DEL SISTEMA DE INFERENCIA DIFUSO DESDE LABVIEW PARA FPGAS

La herramienta propuesta en este trabajo está diseñada en 8 bits en ambiente de programación de Labview. El FIS tiene dos entradas o variables lingüísticas (error E y cambio de error CE) y una salida, Cada variable lingüística tiene tres FM (dos trapezoidales a los extremos y una triangular en el centro), mientras que la salida tiene tres FM tipo singleton. Los valores binarios de entrada son convertidos a enteros sin signo, que están en el rango de 0-255 con el fin de facilitar las operaciones, este proceso de conversión se muestra en la Fig.1.

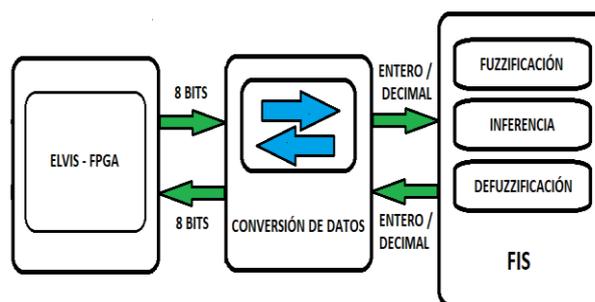


Fig. 1. Diagrama a bloques de la herramienta para diseñar FIS en FPGAs.

A. Diseño de la Fuzificación

Para la fuzificación hay dos variables lingüísticas E y CE, con 3 FM como se observa en la Fig. 2.

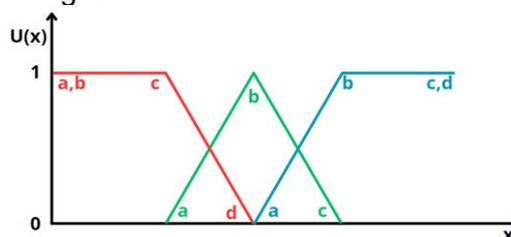


Fig. 2. Variable lingüística E y CE con sus respectivas FM.

En esta etapa se convierten los datos de entrada de binario a enteros sin signo para realizar las operaciones aritméticas que permitirán el cálculo de los grados de membresía, los cuales se llevan a cabo con la

ecuación de la recta y el cálculo de la pendiente. La Fig. 3, muestra el diagrama de flujo para el cálculo de la fuzificación. Los bloques de TRAP y TRIAN son funciones diseñadas en Labview para dibujar y evaluar la FM.

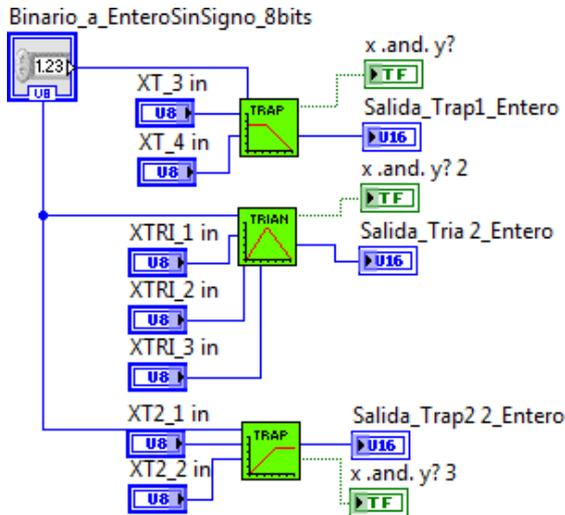


Fig. 3. Diagrama del cálculo de la fuzificación.

En la Fig. 4, se muestra el diagrama de la FM trapezoidal (TRAP) y en la Fig. 5, se muestra el diagrama para la FM triangular (TRIAN).

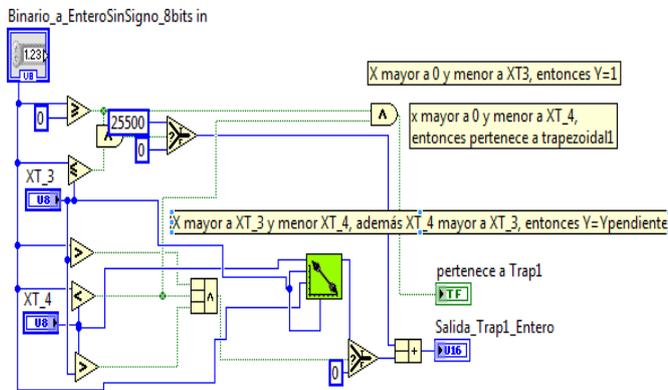


Fig. 4. Diagrama de bloques de la FM trapezoidal.

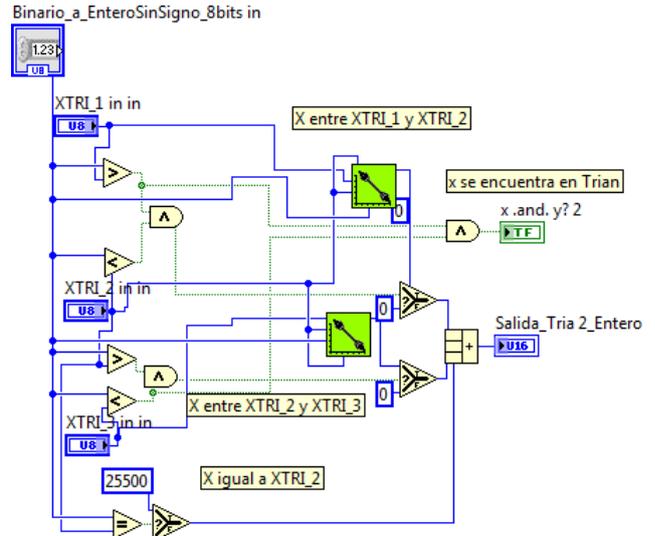


Fig. 5. Diagrama de bloques de la FM triangular.

Las FM tienen las restricciones mostradas en la Fig. 6, Fig. 7 y Fig. 8. En la Fig. 6, se muestra que los puntos x_1 y x_2 son fijos y los puntos x_3 y x_4 pueden ser modificados por el usuario mediante los controles de la interfaz gráfica.

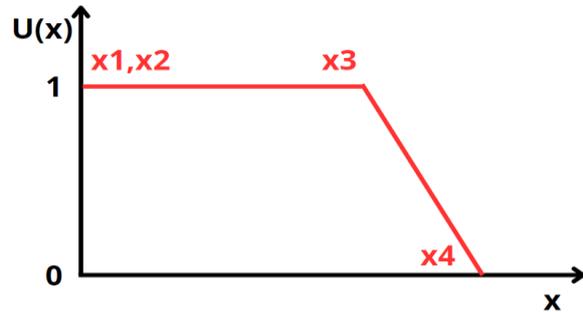


Fig. 6. FM 1 de la variable lingüística E y CE.

En la Fig. 7, se observa que todos los puntos (x_1 , x_2 y x_3) pueden ser modificados por el usuario mediante los controles de la interfaz gráfica.

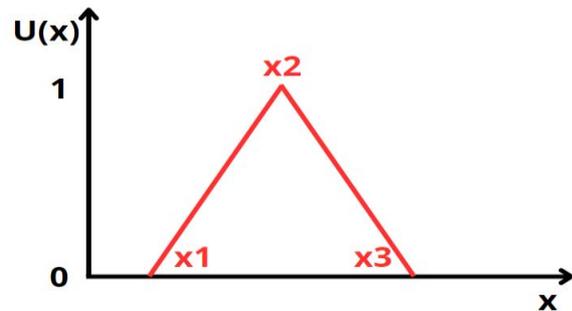


Fig. 7. FM 2 de la variable lingüística E y CE.

Finalmente, para la tercer FM, en la Fig. 8 se observa que los puntos x_3 y x_4 son fijos,

mientras que los puntos x_1 y x_2 son modificados por el usuario mediante la interfaz gráfica.

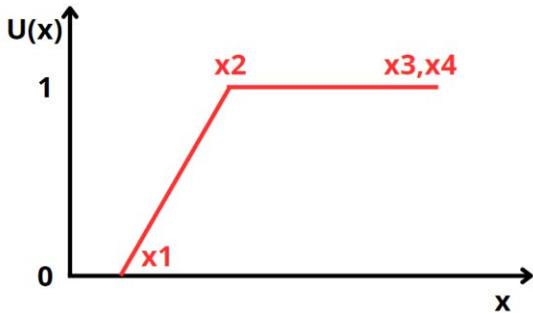


Fig. 8. FM 3 de la variable lingüística E y CE.

Para calcular el grado de membresía se diseñó la siguiente etapa, la cual considera los casos de pendientes positivas o negativas y un divisor con cociente con representación de punto fijo con dos decimales. En la Fig. 9, se muestra el diagrama a bloques del cálculo del grado de membresía.

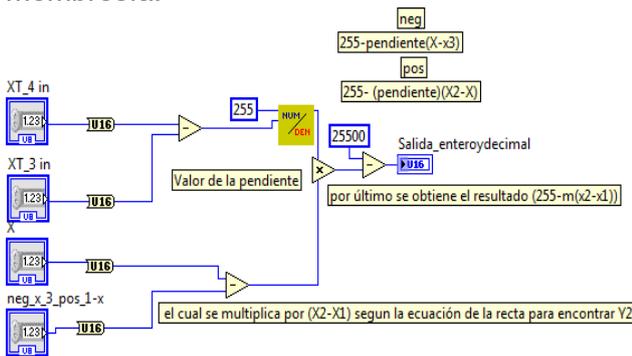


Fig. 9. Cálculo del grado de membresía.

B. Diseño de la Inferencia

El usuario elige mediante la interfaz gráfica el método de implicación mínimo o máximo, por otra parte, las reglas difusas se configuran desde el código de la interfaz gráfica, en la Tabla I, se muestra la base de reglas, en donde E y CE son las variables lingüísticas, mientras que Negativo, Cero y Positivo son las etiquetas lingüísticas, además, incrementa, mantiene y decrementa son los consecuentes.

TABLA I
BASE DE REGLAS

E/CE	Negativo	Cero	Positivo
Negativo	Incrementa	Incrementa	Mantiene
Cero	Mantiene	Mantiene	Mantiene
Positivo	Mantiene	Decrementa	Decrementa

La Fig. 10, muestra el diagrama con la base de reglas, los consecuentes se configuraron en: Decrementa=22, Mantiene=128 e Incrementa=156.

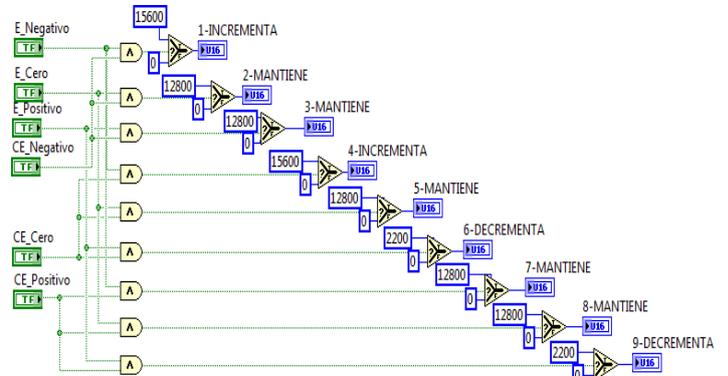


Fig. 10. Base de reglas difusas diseñada de acuerdo a la Tabla I.

El método de mínimos para el cálculo de las fuerzas de disparo es mostrado en la Fig. 11.

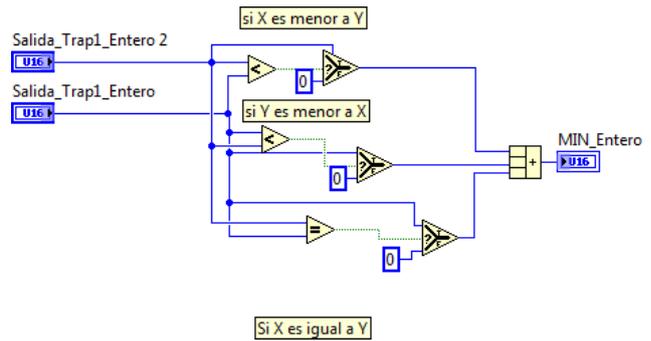


Fig. 11. Método de mínimos.

Mientras que el método de máximos es mostrado en la Fig. 12.

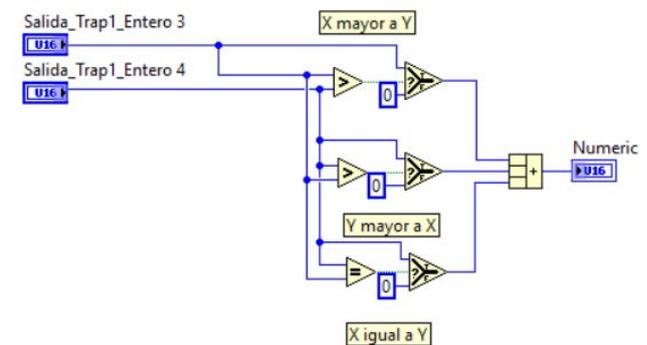


Fig. 12. Método de máximos.

C. Diseño de la Defuzificación

Como método de defuzificación se diseñó en Labview el método de las alturas. La fuerza de

disparo y su grado de membresía es calculado como se muestra en la Fig. 13.

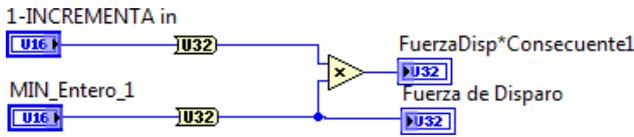


Fig. 13. Método de defuzificación de las alturas: cálculo de la fuerza de disparo y el grado del consecuente.

En la Fig. 14 se muestra el método de defuzificación de las alturas.

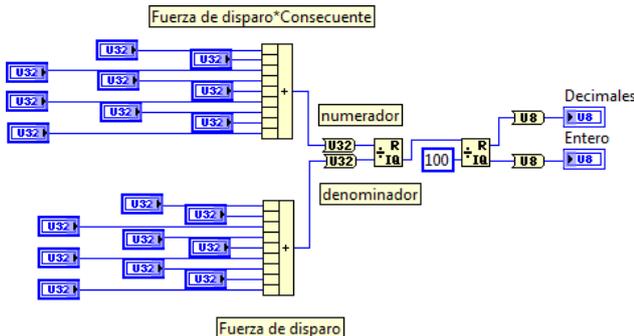


Fig. 14. Cálculo del método de defuzificación de las alturas.

D. Puertos de entradas y salidas

Una vez diseñada cada etapa del FIS en Labview, se continúa con la elección de los puertos de entrada y salida del FPGA, para la herramienta se utilizaron entradas y salidas de propósito general (GPIO), así como los interruptores (slide switch).

E. Sistema de Inferencia Difuso en Labview

El diagrama completo del FIS se muestra en la Fig.15. En la Figura se observa cada una de las etapas incluyendo las entradas y salidas, así como los bloques de conversión para trabajar de binario a decimal en entero sin signo, el FIS está diseñado en un ciclo *while* para ejecutarse de manera indefinida.

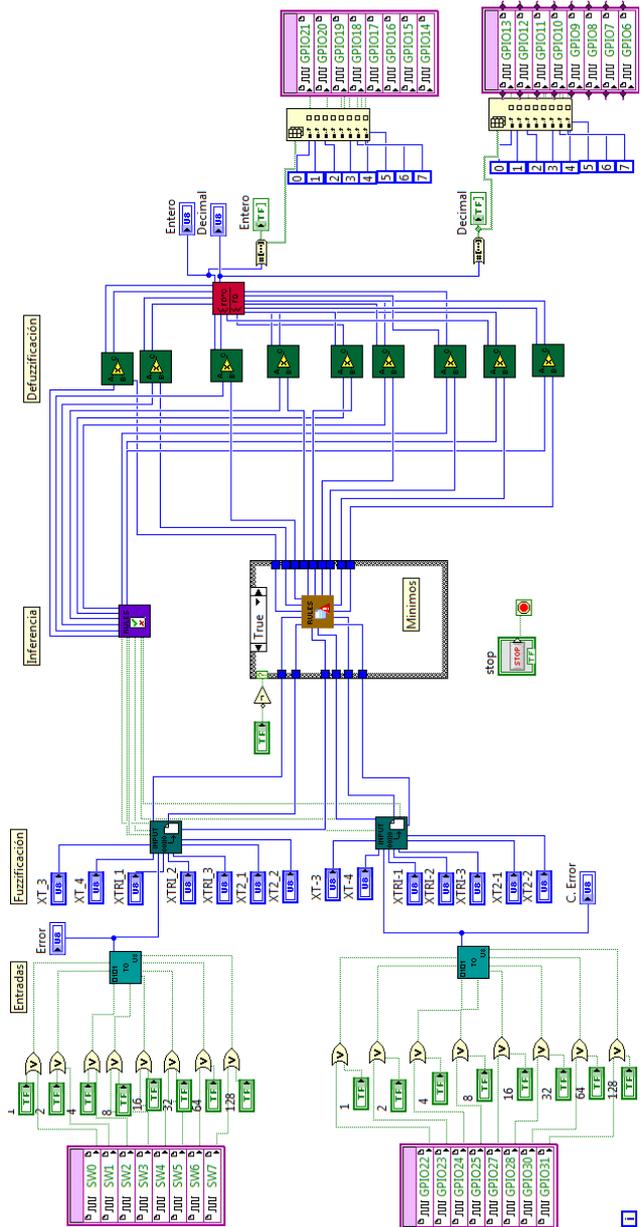


Fig. 15. Diagrama completo de la herramienta para diseñar FIS desde Labview para FPGAs.

IV. DESARROLLO DE LA INTERFAZ GRÁFICA

Al igual que el FIS, la interfaz gráfica fue diseñada en Labview. En la Fig. 16, se muestra el primer panel el cual se utiliza para elegir los valores de cada variable lingüística E y CE, para cada variable hay 8 botones los cuales representan 8 bits, también se tiene un display que muestra la conversión del número de entrada elegido mediante los interruptores.

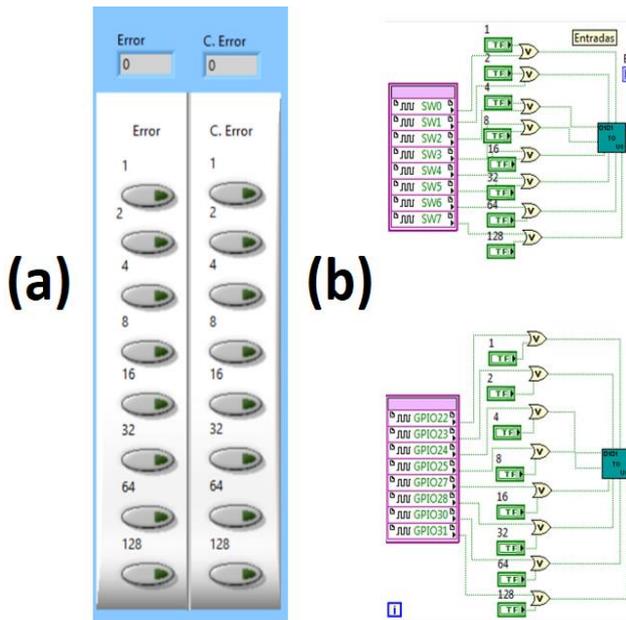


Fig. 16. Panel para parametrizar los valores de las variables lingüísticas E y CE, (a) 8 botones para la elección del valor de entrada y (b) conversión del valor binario a entero sin signo.

En la Fig. 17, se muestra el panel para el control de los parámetros de las FM, de acuerdo a la Sección III.A, el diseño del FIS tiene dos FM trapezoidales a los extremos en donde dos de sus parámetros están disponibles para que el usuario los configure, además de una FM triangular.

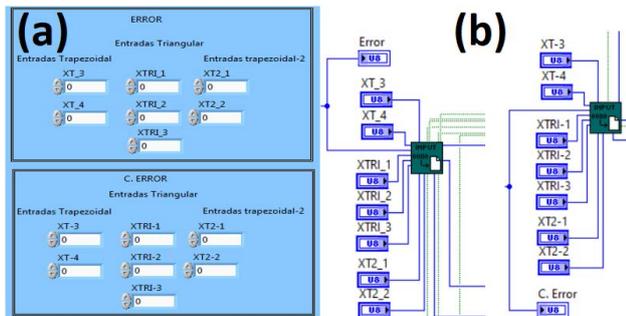


Fig. 17. Panel para parametrizar los valores de las FMs, (a) elección de los parámetros de cada FM y (b) diagrama de bloques de la parametrización de las FM.

El método de inferencia de máximos o mínimos se elige con un interruptor, el panel frontal y el diagrama a bloques se muestra en la Fig. 18.

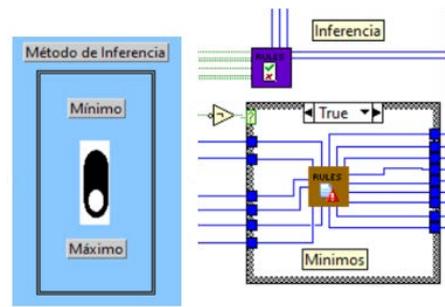


Fig. 18. Panel y el diagrama a bloques para elegir el método de inferencia.

En la Fig 19, se muestra el panel frontal en Labview de la interfaz gráfica de la herramienta para diseñar FIS en FPGAs.

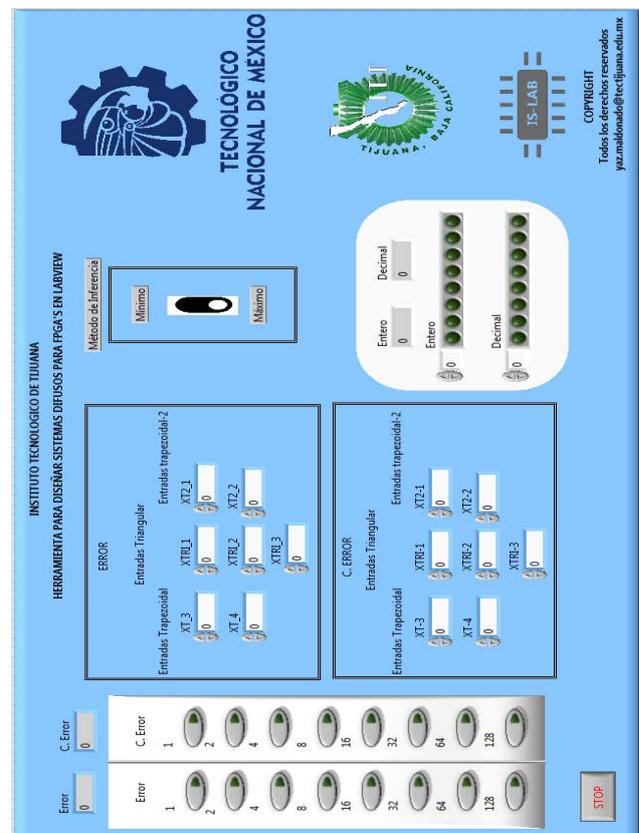


Fig. 19. Interfaz gráfica de la herramienta para diseñar FIS en Labview para FPGAs.

V. VALIDACIÓN DE LA HERRAMIENTA

Para la validación de la herramienta y la configuración de los experimentos se utilizó Labview 2012v, Labview FPGA module 2012v, NI digital electronics FPGA board software 2012-5.2v y NI FPGA Xilinx Tools 13.4v. Además del módulo NI Elvis II FPGA. Los resultados se comparan contra las herramientas de Fuzzy de Matlab y el diseñador

de sistemas difusos de Labview enfatizando que ambas herramientas son para simulación en software, mientras que la herramienta propuesta en este trabajo es también para implementación en hardware con FPGAs.

Los experimentos utilizan los parámetros mostrados en la Tabla II.

TABLA II
PARÁMETROS DEL FIS.

Variable ling.	Término lingüístico			Consecuente		
	Negativo	Cero	Positivo	D	M	I
Error (E)	a=b= 0 c= 40 d=60	a=40 b=60 c=80	a=60 b=80 c=255	22	128	156
Cambio de Error (CE)	a=b= 0 c= 40 d=60	a=40 b=60 c=80	a=60 b=80 c=255			

Donde, D= Decremental, M= Mantiene e I= Incrementa. En la Fig. 20 se muestran las FM para las variables lingüísticas E y CE, mientras que en la Fig. 21, se muestran las funciones de membresía del consecuente.

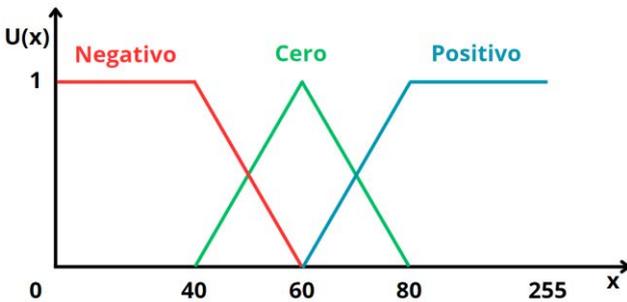


Fig. 20. Variable lingüística E y CE con las FM parametrizadas de acuerdo a la Tabla II.

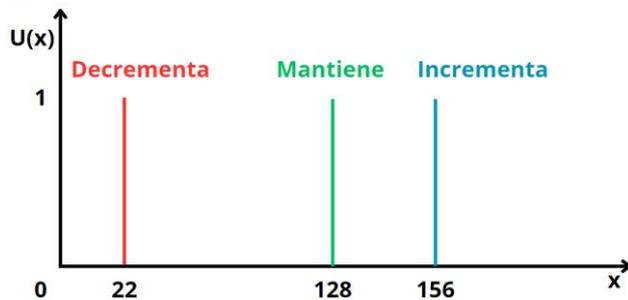


Fig. 21. Consecuente con las FM parametrizadas de acuerdo a la Tabla II.

En la Tabla III, se muestra la comparación entre la herramienta propuesta en este artículo, Fuzzy logic de Matlab y diseñador de sistemas difusos con Labview.

Los resultados del experimento No. 1 se muestran en la Fig. 22.

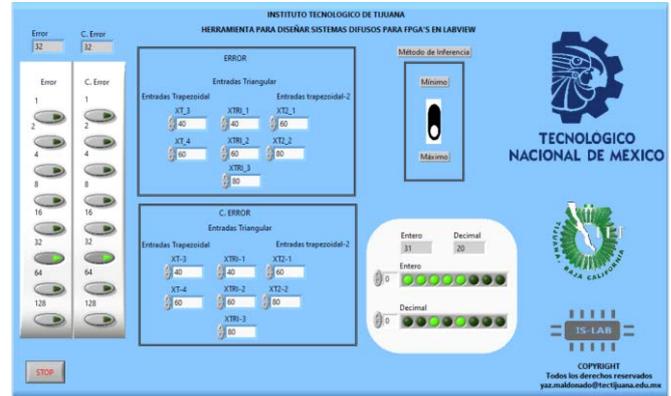


Fig. 22. Experimento No. 1 de la Tabla III en donde E=32, CE=32, con el método de máximos para la inferencia.

La herramienta para diseñar FIS desde Labview para FPGA propuesta en este artículo, utiliza el método de mínimos o máximos para la etapa de inferencia, métodos que también están disponibles en las otras dos herramientas, mientras que Fuzzy logic toolbox de Matlab y la herramienta de diseñador de sistemas difusos de Labview tienen métodos de defuzificación diferentes al método de las alturas que utiliza la herramienta propuesta, variación que se aprecia en los resultados comparativos del *crisp*. Esto se aprecia mejor en la Fig. 23, donde los resultados del desempeño de cada herramienta han sido graficados. Para comprender la variación entre los resultados del método de inferencia con mínimos y máximos se debe recordar que el método de defuzificación realiza la sumatoria del grado de membresía (mínimos o máximos, según el caso) y se multiplica por el consecuente (el cual depende de las reglas activas), esto se divide entre la sumatoria de los grados de membresía (mínimos o máximos dependiendo el caso). Por lo tanto, en el método de inferencia de los máximos, el denominador será mayor que en el método de los mínimos, lo que da por resultado valores menores en el *crisp*.

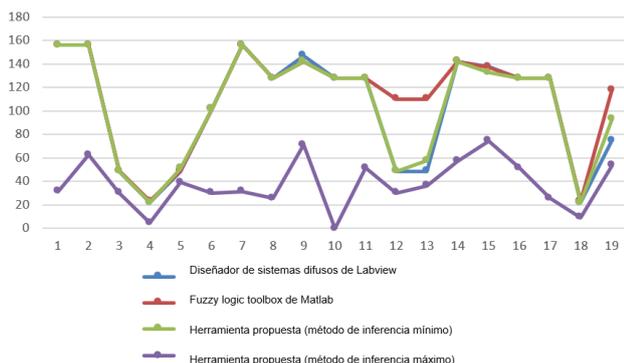


Fig. 23. Gráfica de desempeño de la herramienta propuesta contra las de la literatura, donde el eje x es el número del experimento y el eje y es el valor del *crisp* de la Tabla III.

VI. CONCLUSIONES

Los sistemas de inferencia difusos no están limitados a ciertas aplicaciones, por el contrario, los FIS por su buen desempeño con datos inciertos conquistan otros campos de estudio. Contar con herramientas versátiles que combinen el desarrollo de software con el hardware y que se lleven a la implementación son valiosas para buscar nuevas alternativas para aplicar FIS en problemas reales.

Con esta propuesta, se tiene una herramienta para diseñar FIS desde Labview para FPGAs, la herramienta cuenta con una interfaz gráfica fácil de usar para configurar un FIS de dos entradas y una salida, en donde cada entrada está conformada por 3 FM (2 trapezoidales y una triangular), también cuenta con dos métodos de implicación para la inferencia que son el mínimo y el máximo, el método de las alturas es utilizado para la defuzificación. Una vez que el usuario parametriza el FIS, el paso siguiente es programarlo en un FPGA.

Como trabajo futuro queda agregar más entradas al FIS, incluir en la interfaz gráfica la elección de las reglas difusas, así como agregar otros métodos de defuzificación como el centroide. Lo anterior hará que la herramienta sea robusta para competir con otras herramientas de la literatura como *Labview*, *Matlab* y *XFUZZY*, lo que motivará a expertos en el área de FIS o en el desarrollo de hardware a utilizarla por la facilidad para interactuar entre las dos áreas del conocimiento.

RECONOCIMIENTOS

Este trabajo fue financiado por el Tecnológico Nacional de México con el proyecto No. 19455.24-P.

REFERENCIAS

- [1] Carlos Gonzales Morcillo. Lógica difusa: Una introducción práctica. https://www.esi.uclm.es/www/cglez/downloads/docencia/2011_Softcomputing/LogicaDifusa.pdf.
- [2] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, en *Information Sciences*, 8 (1975), 199 - 249).
- [3] Zavala, A.; Nieto, O. Fuzzy hardware: A retrospective and analysis. *IEEE Trans. Fuzzy Syst.* 2011, 20, 623–635.
- [4] Mamdani, E.; Assilian, S. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J.-Man-Mach. Stud.* 1975, 7, 1–13.
- [5] Ross, T. *Fuzzy Logic with Engineering Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2005
- [6] Li, H.; Wu, C.; Shi, P.; Gao, Y. Control of nonlinear networked systems with packet dropouts: Interval type-2 fuzzy model-based approach. *IEEE Trans. Cybern.* 2014, 45, 2378–2389.
- [7] Narayan, V.; Mall, P.K.; Awasthi, S.; Srivastava, S.; Gupta, A. FuzzyNet: Medical Image Classification based on GLCM Texture Feature. In *Proceedings of the 2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*, Greater Noida, India, 27–29 January 2023; 769–773.
- [8] Hannan, M.A.; Ghani, Z.A.; Hoque, M.M.; Ker, P.J.; Hussain, A.; Mohamed, A. Fuzzy logic inverter controller in photovoltaic applications: Issues and recommendations. *IEEE Access* 2019, 7, 24934–24955.
- [9] Castellano, G.; Castiello, C.; Pasquardibisceglie, V.; Zaza, G. Fisdet: Fuzzy inference system development tool. *Int. J. Comput. Intell. Syst.* 2017, 10, 13–22.
- [10] Guillaume, S.; Charnomordic, B. Fuzzy inference systems: An integrated modeling environment for collaboration between expert knowledge and data using FisPro. *Expert Syst. Appl.* 2012, 39, 8744–8755.
- [11] Cingolani, P.; Alcalá-Fdez, J. jFuzzyLogic: A java library to design fuzzy logic controllers according to the standard for fuzzy control programming. *Int. J. Comput. Intell. Syst.* 2013, 6, 61–75.
- [12] McCulloch, J. Fuzzycreator: A python-based toolkit for automatically generating and analysing data-driven fuzzy sets. In *Proceedings of the 2017 IEEE International Conference on Fuzzy Systems (FUZZ—IEEE)*, Naples, Italy, 9–12 July 2017; 1–6.
- [13] McCulloch, J. Fuzzycreator: A python-based toolkit for automatically generating and analysing data-driven fuzzy sets. In *Proceedings of the 2017 IEEE International Conference on Fuzzy Systems (FUZZ—IEEE)*, Naples, Italy, 9–12 July 2017; 1–6.
- [14] Wagner, C.; Pierfitt, M.; McCulloch, J. Juzzy online: An online toolkit for the design, implementation, execution and sharing of type-1 and type-2 fuzzy logic systems. In *Proceedings of the 2014 IEEE International Conference on Fuzzy Systems (FUZZ—IEEE)*, Beijing, China, 6–11 July 2014; 2321–2328
- [15] The MathWorks, Inc. *Matlab Fuzzy Logic Tool Users Guide*. 2022. Available online: <https://www.mathworks.com/help/fuzzy/>
- [16] National Instruments. *Fuzzy Logic Vis Fuzzy Logic – NI*, 2024.
- [17] Moreno Velo, F.; Sánchez Solano, S.; Barriga Barros, Á.; Baturone Castillo, M.; López, D. Xf13: A new fuzzy system specification language. In *Proceedings of the 5th WSES /IEEE Multiconference on Circuits, Systems, Communications and Computers*, Rethymnon, Greece, 8–15 July 2001; 361–366.

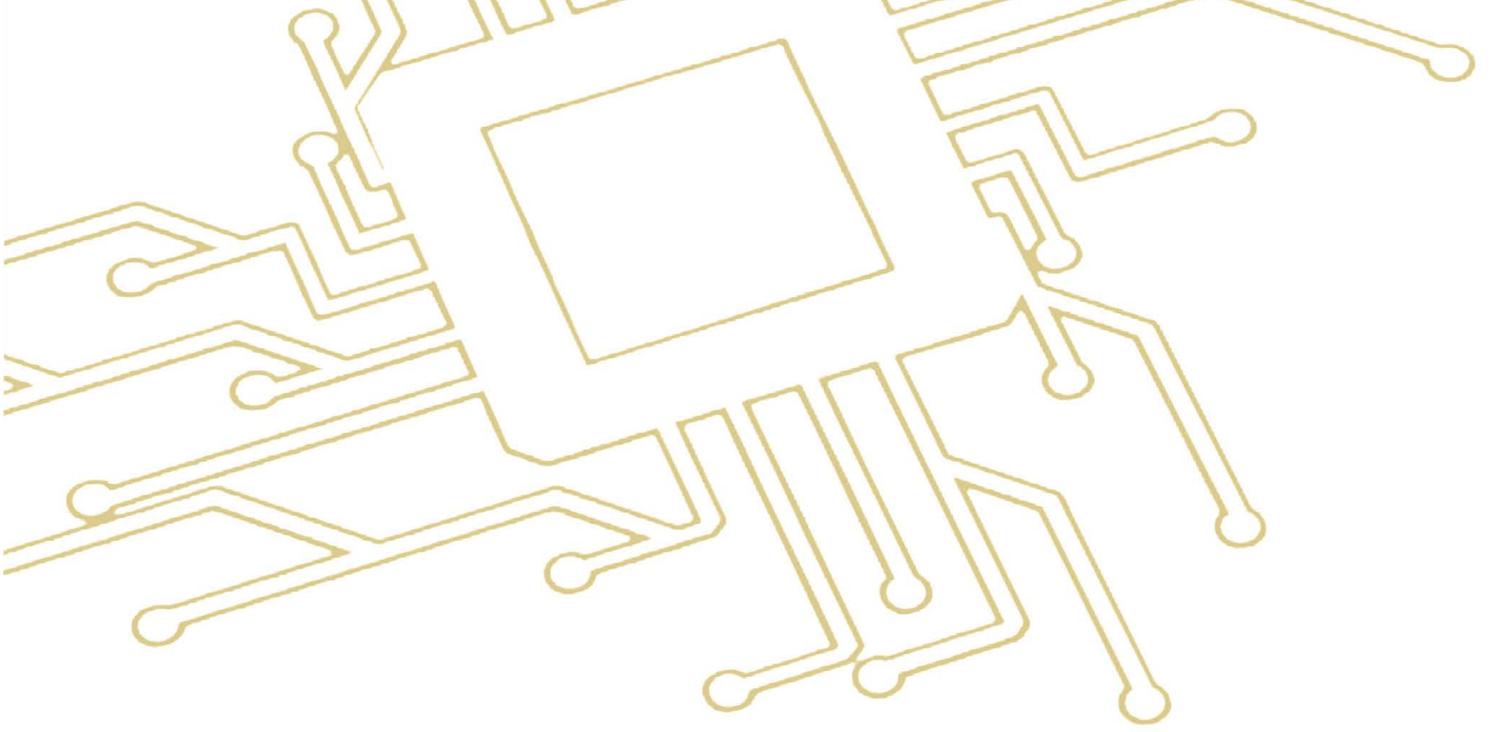
- [18] Valdez, R.; Maldonado, Y.; Quevedo, J.A. Fuzzy Hardware Tool: An Adaptable Tool to Facilitate the Implementation of Fuzzy Inference Systems in Hardware. *Electronics* 2023, 12, 2853.

TABLA III

COMPARACIÓN DE RESULTADOS UTILIZANDO LA HERRAMIENTA DE LABVIEW, MATLAB Y LA PROPUESTA EN ESTE ARTÍCULO.

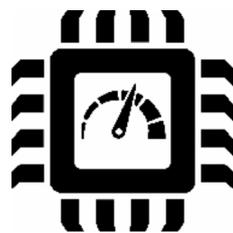


No.	Valor de entrada para E y CE	Crisp			
		Diseñador de sistemas difusos de Labview	Fuzzy logic de Matlab	Herramienta propuesta en este trabajo, método de inferencia de mínimos	Herramienta propuesta en este trabajo, método de inferencia de máximos
1	E=32, CE=32	156	156	156	31.20
2	E=32, CE=50	156	156	156	62.40
3	E=75, CE=96	48.5	49.1	48.5	30
4	E=96, CE=96	22	22.9	22	4.40
5	E=75, CE=79	48,5	49.1	50.9	38.97
6	E=96, CE=45	101.5	101	101.5	30
7	E=32, CE=60	156	156	156	31.20
8	E=32, CE=96	128	128	128	25.60
9	E=50, CE=50	146.66	142	142	71
10	E=60, CE=60	128	128	128	25.60
11	E=75, CE=32	128	128	128	51.20
12	E=75, CE=60	48.5	110	48.5	30
13	E=75, CE=75	48.5	110	57.33	35.77
14	E=50, CE=32	142	142	142	56.80
15	E=50, CE=75	137.33	137	132.66	74.22
16	E=50, CE=96	128	128	128	51.20
17	E=96, CE=32	128	128	128	25.60
18	E=96, CE=75	22	22.9	22	8.80
19	E=75, CE=50	75	118	92.66	53.44



Eje Temático

Aplicaciones en la Industria y Control de Potencia





Modulador Trifásico de Salida Sinusoidal Implementado sobre ESP32 para Accionamiento de Motores y Generación Distribuida

Santiago A. Verne

Instituto de Investigaciones en Electrónica, Control y Procesamiento de Señales
(LEICI)

La Plata, Argentina

santiago.verne@ing.unlp.edu.ar

Resumen—En este trabajo se presenta la implementación de un modulador trifásico PWM utilizando un microcontrolador ESP32. Por su bajo costo y conectividad Wi-Fi integrada, este microcontrolador se ha transformado en el caballo de batalla de la internet de las cosas (IoT). Sin embargo, el uso del módulo de control de motor (MCPWM) se ha restringido a aplicaciones con motores de corriente continua con escobillas. Este trabajo describe el uso de este periférico para realizar un modulador PWM trifásico con salida sinusoidal. El desarrollo puede utilizarse como base para el accionamiento de motores trifásicos o imanes permanentes e incluso sistemas de gestión de energía renovable. Dado que la mayor carga de trabajo recae sobre periféricos, los recursos utilizados del núcleo del microcontrolador son muy reducidos. Se describen los periféricos utilizados y sus funciones asociadas en la interfaz de programación de aplicaciones (API). La programación se realiza en el entorno Arduino IDE y los resultados se validan sobre un inversor prototipo.

Palabras clave: Inversor trifásico, ESP32, modulador por sub-oscilación, accionamiento de máquinas eléctricas.

I. INTRODUCCIÓN

La informática, los sistemas eléctricos y las comunicaciones se enfocan hacia el paradigma de la descentralización. Protocolos de consenso, contratos inteligentes, gestión y comercialización de la energía, monitoreo y accionamiento remoto son tópicos en plena

expansión [1][2], que sumados al modelo de código abierto abren el camino para innumerables aplicaciones y productos [3][4]. En todos estos casos cada agente de la red requiere un terminal de hardware específico. En el ámbito de las redes eléctricas se precisa convertir los niveles de tensión, o bidireccionalmente de alterna a continua o de mecánica a eléctrica, y aquí los convertidores electrónicos son un elemento clave. Dentro de este grupo, el inversor de tensión permite convertir energía eléctrica continua en alterna, y también eléctrica a mecánica a través de motores eléctricos [5][6].

El control del inversor contempla el algoritmo de modulación, el cual determina el patrón de apertura y cierre de las llaves del inversor, así como funciones auxiliares de bajo nivel como tiempos muertos y detección rápida de estados de falla, etc [7]. Si bien muchas de estas complicaciones se resuelven utilizando hardware específico para el control de convertidores, su costo suele ser elevado, siendo esto relevante en el desarrollo de productos de uso masivo. Si por otra parte, el sistema a implementar precisa conectividad para monitoreo o accionamiento remoto, la implementación se complejiza, aumentando la cantidad de componentes y degradando la confiabilidad del sistema. Esta problemática disminuye al utilizar soluciones *on-chip* y en aplicaciones *IoT* el microcontrolador ESP32 ha sido pionero desde su lanzamiento [8][9]. Aplicaciones y trabajos académicos de todo tipo lo han demostrado, [10]-[13], aunque en el contexto de inversores de tensión los

antecedentes son escasos. El reciente trabajo [14] presenta un inversor de tensión basado en ESP32, aunque no se especifican los recursos de hardware utilizados para ello. En [15] también se reporta un firmware correspondiente a un inversor de salida sinusoidal, aunque se realiza la modulación por software, y finalmente en [16] se demuestra el giro de un motor de imán permanente, aunque no se muestran formas de onda ni se documenta el método y los recursos utilizados.

En este trabajo se presenta un modulador PWM de onda sinusoidal para un inversor de tensión trifásico, implementado en un microcontrolador de bajo costo ESP32. El desarrollo tiene como objetivo explotar las cualidades del periférico de control de motor del ESP32, el cual, en conjunto con las prestaciones de conectividad de este microcontrolador, lo hacen ideal para proyectos relacionados con la conversión de energía renovable y el accionamiento de cargas mecánicas mediante motores trifásicos.

II. DESCRIPCIÓN DEL SISTEMA Y MÓDULO PARA CONTROL DE MOTOR DEL μ C ESP32

El esquema de un inversor trifásico se muestra en la Figura 1. El mismo se alimenta a partir de un bus de tensión continua de valor V_{cc} , al que se conectan las tres columnas del inversor (a, b y c) compuestas a su vez por pares de transistores Q1-Q4, Q2-Q5 y Q3-Q6 operando como interruptores complementarios. Las señales de activación de cada transistor son determinadas por el algoritmo de modulación del inversor. Entre la salida de corriente alterna del inversor (nodos V_a , V_b y V_c) y la carga, suele interponerse un filtro

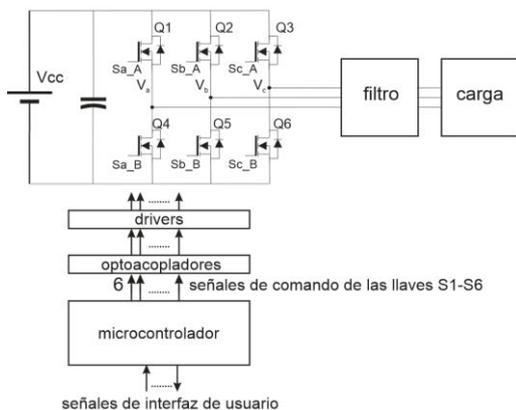


Figura 1. Esquema circuital del inversor trifásico

pasabajos cuya función es atenuar las componentes de alta frecuencia debidas a la conmutación.

A. Aspectos básicos de un modulador por el método de sub-oscilación

El objetivo de un modulador en el contexto de inversores consiste en codificar el contenido armónico una señal denominada modulante mediante un tren de pulsos. La señal resultante puede ser amplificada con bajas pérdidas mediante llaves electrónicas. En el método por sub-oscilación [17] esto se logra comparando la modulante con una señal triangular como se muestra en la Figura 2. Cuando la implementación de este algoritmo es digital, la señal triangular es un contador ascendente-descendente (A-D) cuyo registro se compara con el valor numérico de la modulante para dar como resultado un tren de pulsos cuyo valor medio replica a la modulante (Figura 2(b)). Extrapolando esto a tres comparadores y tres modulantes sinusoidales desfasadas 120 grados, la terna de trenes de pulsos pueden aplicarse a las tres columnas del inversor de la Figura 1.

B. Descripción del periférico para control de motor (MCPWM) del ESP32

El microcontrolador ESP32 posee dos módulos para control de motor (MCPWM0 y 1). Los mismos están orientados al control de máquinas de corriente continua sin escobillas (BLDC: brushless DC motor), aunque su uso habitual es para control de motores de CC con escobillas. Un diagrama del módulo se muestra en la Figura 3. El mismo posee seis salidas que son resultado de la comparación entre dos

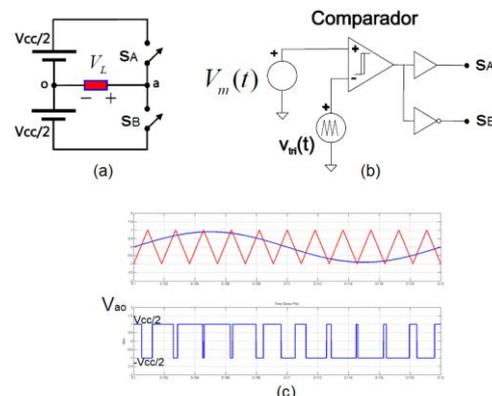


Figura 2. Aplicación del método por sub-oscilación a una columna de un inversor trifásico. a) Definición de la tensión de columna V_{ao} , (b) Proceso de comparación entre la modulante y la "portadora", (c) Forma de onda sobre la carga. Puede observarse que el valor medio móvil es proporcional a la señal modulante.

registros: uno asociado, en nuestro caso, a la señal modulante y el otro asociado al contador A-D proveniente de uno de los *timers* del módulo (Figura 3(a)). También se dispone de entradas al módulo, las que se dividen en tres conjuntos según su función: sincronismo externo, captura de señales de falla y captura y registro temporal de eventos en la carga. La Figura 3(b) muestra el detalle sobre cómo se conforma la señal PWM a partir de la comparación del registro del timer con el registro *OPERATOR* y su post procesamiento. Según la configuración, esto puede incluir agregar tiempos muertos, ejecución de acciones de emergencia e incluso modular la señal para poder transmitirla a través de un transformador *driver*.

C. Configuración del módulo MCPWM

En los párrafos siguientes se describe el uso y configuración del microcontrolador para sintetizar las señales de activación de los transistores Q1-6 de manera que, posterior al filtrado, el inversor entregue una terna sinusoidal de frecuencia y amplitud constantes. Este proceso se logra a partir de configurar el módulo MCPWM y un *timer* auxiliar asociado a una interrupción. Esto significa que una vez definida la amplitud y frecuencia deseadas, el modulador opera en forma autónoma sin ejecutar instrucciones en el programa principal.

Un ciclo completo de la sinusoide se genera *offline* y sus muestras son tomadas secuencialmente de una tabla a intervalos de T_{HOLD} segundos (Figura 4(a)) utilizando un *timer* auxiliar (*timer_aux*) externo al módulo MCPWM (Figura 4(b)). Cada T_{HOLD} , el *timer* produce una interrupción cuya rutina carga las muestras correspondientes a cada fase del inversor dentro de los registros *OPERATOR* 0,1,2) (Figura zm2(b)) del MCPWM. Dichas muestras son mantenidas hasta la próxima interrupción y durante ese tiempo son comparadas ciclo a ciclo del contador A-D que corre en *TIMER0* (por simplicidad los tres operadores se asocian al mismo timer). De esto surgen 3 trenes de pulsos y sus complementos (PWM_A y B) que se entregan a la salida de cada operador a fin de excitar a ambos transistores de cada columna del inversor.

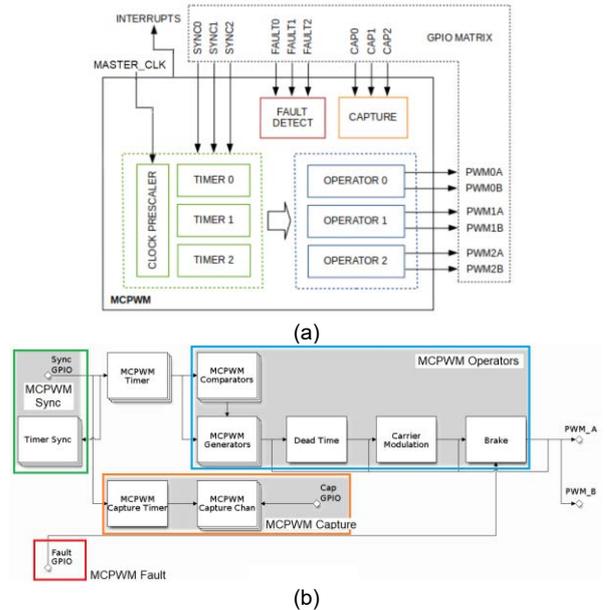


Figura 3. Diagrama de componentes del módulo MCPWM del ESP32, (a) Descripción general, (b) Detalle de las funcionalidades para el comando de una columna de la etapa de potencia.

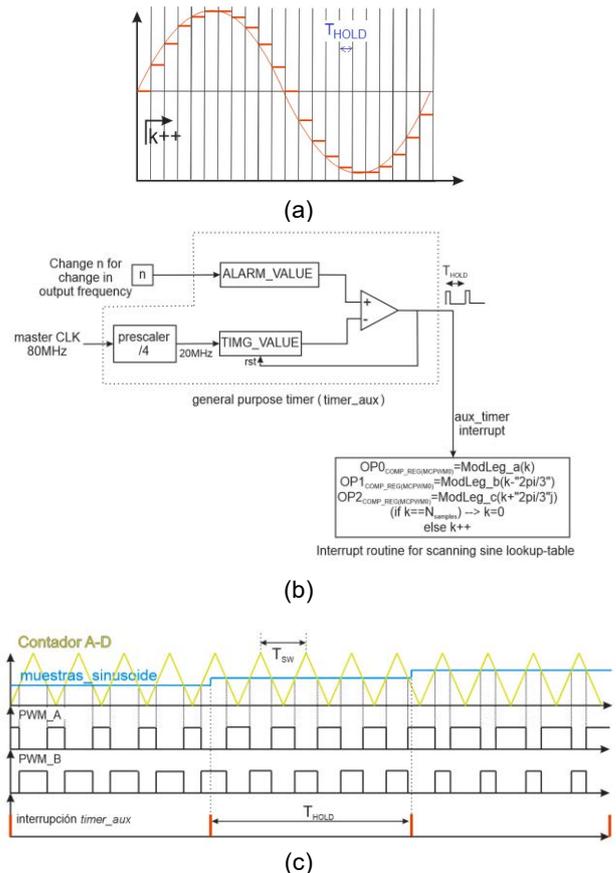


Figura 4. Esquema de muestreo de la modulante por el timer auxiliar, a) Registro de la señal modulante sinusoidal b) Muestreo mediante interrupción cada T_{HOLD} segundos, c) Esquema del timer auxiliar para la generación de la interrupción de refresco de la muestra.

La frecuencia del reloj del sistema (*MASTER_CLK*), el valor del *prescaler* del módulo utilizado (*CLOCK_PRESCALER*), el del timer (*TIMER0*) y el valor tope de cuenta del A-D determinan el tiempo que tarda dicho contador en llegar al valor máximo ($T_{SW}/2$), y por ende, el período de conmutación. Mediante el parámetro *counter_mode* se indica modo de conteo: triangular o diente de sierra, siendo el primero el utilizado en este caso debido a su mejor perfil espectral [18].

D. Funciones de la API para configuración del MCPWM

En el inicio, se configura el timer auxiliar a partir del cual, cada T_{HOLD} , se ejecuta la rutina de toma de muestras la sinusoide y su carga en los comparadores correspondientes a las columnas a, b y c (*OPERATOR0*, 1 y 2). A continuación se configura el módulo MCPWM (Figura 5) que se realiza en dos etapas.: la primera es de preconfiguración para una puesta en marcha sencilla y la siguiente para establecer con precisión los registros de prescaler, el de cuenta máxima, el modo de conteo, etc. Esto es necesario ya que la función provista por la API configura dichos registros de forma automática por lo que resulta una combinación arbitraria de los mismos. Para realizar estos ajustes finales, la API permite el acceso directo a los registros de hardware del dispositivo.

II. RESULTADOS EXPERIMENTALES

Utilizando como base el esquema descrito se implementó un variador de velocidad para un motor de inducción trifásico con control escalar. Este método se basa en modificar la frecuencia

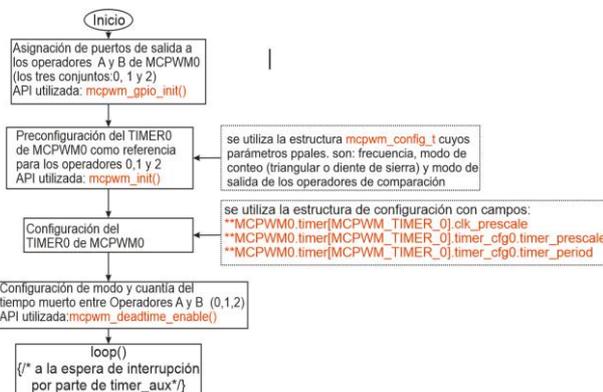


Figura 5. Flujo de configuración para el módulo MCPWM y funciones API asociadas (se omite la configuración del timer auxiliar).

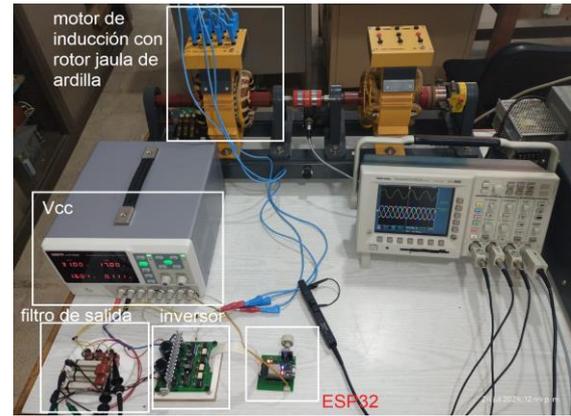


Figura 6. Implementación en el banco de pruebas.

de rotación del campo estático manteniendo la relación V/f constante e igual a V_{nom}/f_{nom} . Esto mantiene la característica par-velocidad a distintas frecuencias y permite accionar todo tipo de cargas: compresores, bombas, forzadores de aire, etc.

El diagrama del arreglo experimental se muestra en la Figura 7 y los parámetros del mismo se listan en la Tabla I.

Para excitar los transistores de potencia se utilizó el circuito *driver* integrado IR2110, el cual posee las dos salidas para excitar una columna completa. Por seguridad, se interponen optoacopladores tipo 6N137 entre el microcontrolador y el *driver*. A la salida del

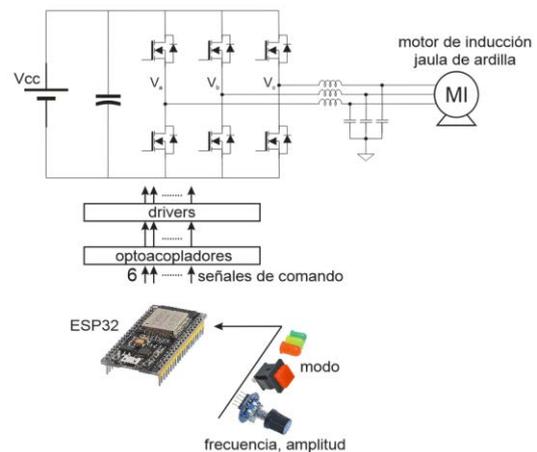


Figura 7. Diagrama de la implementación para pruebas del modulador trifásico.

TABLA I
PARÁMETROS DEL ARREGLO EXPERIMENTAL

Parámetro	Descripción
Vcc	31V
Motor de inducción	Tensión de línea 24V _{ef} @50Hz, conexión triángulo, 4 polos P _{omax} =200W, n _{nom} =1440RPM, rotor jaula de ardilla
Optoacopladores	6N137 con V _{ccout} =5V
Drivers	IR2110, con V _{ccin} =5V V _{ccout} =15V
Filtro de salida	C=2,2uF, L=2mH, R _{damp} =50Ω
f _{sw} =1/T _{sw}	20kHz

inversor se dispone un filtro LC pasabajos a fin de observar las tensiones filtradas de columna. El detalle del arreglo experimental puede observarse en la Figura 6.

A. Tensiones de salida del inversor

La Figura 8(a) muestra las tres tensiones de columna del inversor (V_a , V_b y V_c) respecto al negativo de la tensión del bus DC. Se observan los puntos correspondientes a los picos de sus correspondientes modulantes indicando, en este caso, un período de modulante de 20ms y una fase relativa entre fases consecutivas de 6,7ms, aproximadamente (120 grados eléctricos). En la Figura 8(b) se observan los trenes de pulsos para las tres fases cuyo período es 50us, lo que corresponde a $f_{sw}=20kHz$, frecuencia programada para la conmutación de las columnas (Sección II.C). Las tensiones filtradas, junto con la corriente en una de las fases del motor se muestran en la Figura 9 y lucen sinusoidales, reproduciendo con baja distorsión el contenido espectral de la modulante almacenada en la tabla de memoria. En trazo verde, la corriente también presenta forma sinusoidal.

B. Tiempos muertos

Una precaución al conmutar las columnas de



Figura 8. Terna de tensiones de columna, a la salida del inversor.

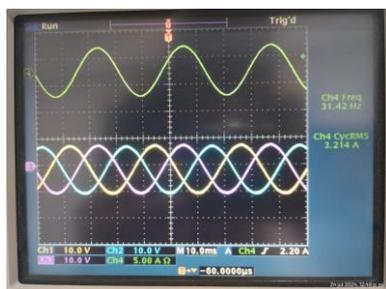


Figura 9. Señales de salida del inversor. Verde: Corriente de entrada al motor, Terna de tensiones de fase (con respecto al neutro del filtro).

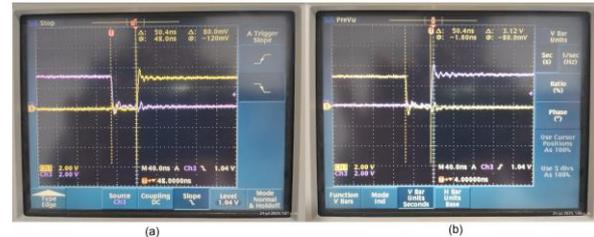


Figura 10. Señales de comando para una columna del inversor. Trazo violeta: para el transistor Q1, amarillo para Q4.

un inversor es apagar la llave activa y luego de un tiempo dado, denominado tiempo muerto (TM), encender su complementaria. Esto es necesario para evitar cortocircuitos transitorios de columna que ocurren debido al retardo de las llaves en su apagado. El módulo MCPWM del ESP32 tiene implementada esta funcionalidad y el TM se puede programar en múltiplos de 100ns con la función descrita en la Sección II.D. En la Figura 10 se muestran las señales de excitación para una columna del inversor. Entre el instante de apagado de una de ellas y el de encendido de la otra transcurre el tiempo muerto, durante el cual, la corriente de carga circula naturalmente por el diodo de rueda libre superior o el inferior dependiendo de la dirección de corriente de carga. Utilizando los cursores se pudo medir un tiempo de 50,4ns para cada transición (alto a bajo y viceversa sumando un total de 100,8ns siendo este resultado acorde a lo esperado, ya que el tiempo implementado en la configuración fue 100ns.

III. CONCLUSIONES

En este trabajo se presentó la implementación de un modulador trifásico de salida sinusoidal implementado con un microcontrolador de bajo costo ESP32. Se establecen los lineamientos para hacer uso de un poderoso periférico del dispositivo, el Módulo de Control de Motor (MCPWM), mediante el cual pudo obtenerse una terna sinusoidal de alta calidad utilizando escasos recursos computacionales. El desarrollo de dispositivos de conversión electrónica de potencia en relación con el accionamiento de máquinas eléctricas y las energías renovables puede potenciarse con el uso de un hardware tan popular como el microcontrolador ESP32. Su bajo costo y potencialidades de conectividad, lo hacen muy competitivo en este tipo de aplicaciones.



RECONOCIMIENTOS

El autor expresa su gratitud a las instituciones Universidad Nacional de La Plata, CONICET y ANPCyT, cuyo apoyo fue indispensable para la realización de este trabajo.

REFERENCIAS

- [1] Ying Wu, Yanpeng Wu, Josep M. Guerrero, Juan C. Vasquez, "Digitalization and decentralization driving transactive energy Internet: Key technologies and infrastructures", *Int. Jnl of Electrical Power & Energy Systems*, Vol_ 126, Part A, 2021.
- [2] Casquico, M., Mataloto. B.: Ferreira, J.C.; Monteiro, V.. Afonso, J.L.; Afonso. J A Blockchain and Internet of Things for Electrical Energy Decentralization: A Review and System Architecture, *Energies*, No. 14, 8043, 2021.
- [3] A. Samuel and C. Sipes, "Making Internet of Things Real.", *IEEE Internet of Things Magazine*, Vol. 2, no. 1, pp. 10-12. 2019.
- [4] A Moms, Nikolaos, Nikolaos Peladarinos. Vasileios Cheimaras, Panagiotis Papageorgas, and Dimitrios D. Piromalis, "A Thorough Review and Comparison of Commercial and Open-Source IoT Platforms for Smart City Applications", *Electronics*. Vol. 13, No 8. 1465. 2021.
- [5] Arfeen ZA, Kermadi M. Azam MK. et al.: Insights and trends of optimal voltage-frequency control DG-based inverter for autonomous microgrid: State-of-the-art review", *Int Trans Electr Energ Syst*. 30: e12555. 2020.
- [6] H. Lu, Q. Wang, J. Chai and Y. Li, "Review of Three-Phase Soft Switching Inverters and Challenges for Motor Drives" *CES Transactions on Electrical Machines and Systems*, Vol. 8, no 2, pp. 177-190, 2024.
- [7] E. R. C. da Silva. E. C. dos Santos and B. Jacobina, "Pulsewidth Modulation Strategies," *IEEE Industrial Electronics Magazine*, Vol. 5, no. 2, pp. 37-45, 2011.
- [8] R. Krishnamoorthy et al., "Systematic Approach for State-of-the-Art Architectures and System-on-Chip Selection for Heterogeneous IoT Applications," *IEEE Access*, Vol. 9, pp. 25594-25622, 2021.
- [9] Kareem and D. Dunaev, "The Working Principles of ESP32 and Analytical Comparison of using Low-Cost Microcontroller Modules in Embedded Systems Design" 2021 4th International Conference on Circuits, Systems and Simulation (ICCSS), Kuala Lumpur, Malaysia, 2021.
- [10] K. V. Sowmya, T. N. V. S. Pavan, V Teju, G. S. D. Prasad and P. V. Krishna, "IoT based Energy Monitoring in Smart Home using Mobile Application," 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida. India, 2021, pp. 771-776, 2021.
- [11] S. Damrongpreechar, N Boonyaudomsart and C. Charoenlarnnopparut, "Peer-to-Peer (P2P) Energy Trading Prototype Implementation with Private Ethereum Blockchain and WebSocket," 2024 International Conference on Smart Grid and Energy (ICSGE), Singapore, Singapore, pp. 31-37, 2024.
- [12] S. Bipasha Biswas and M. Tariq Iqbal, "Solar Water Pumping System Control Using a Low Cost ESP32 Microcontroller," 2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE), Quebec, QC, Canada, pp 1-5, 2018.
- [13] M. Babiuch, P. Foltyniek and P. Smutny, "Using the ESP32 Microcontroller for Data Processing," 2019 20th International Carpathian Control Conference (ICCC), Krakow-Wieliczka, Poland, pp. 1-6, 2019.
- [14] Putra, Zindhu & Nugraha, Anggara & Widiarti, Yuning & Safaroz, Wafiq & Sobhita, Rama. "Design of Unipolar Pure Sine Wave Inverter with Spwm Method Based On Esp32 Microcontroller As a Support of The Ebt System On Ship", *E3S Web of Conferences*, 473, 2024.
- [15] <https://github.com/cybernetic-research/esp32-sine-inverter/blob/master/mosfet-pushpull-300Hz.ino>

- [16] <https://www.youtube.com/watch?v=7nrxFbs7BE>, accessed 1st august 2024
- [17] A. laderosa, "Overmodulation and six-step mode for observer-based V/Hz control", Master thesis, Aalto University, School of Electrical Engineering, 2023
- [18] Holmes, D G and Lipo. T A (2003). *Pulse Width Modulation for Power Converters. Principles and Practice* IEEE Press, Piscataway, NJ.EUA.



Incubadora de Huevos con “Tecnología Apropriada”, Software y Hardware Libre

Javier Jorge
Instituto Nacional de Tecnología
Industrial (INTI)
Universidad Nacional de Córdoba
Córdoba, Argentina
jjorge@inti.gob.ar

Fabrizio Puzio
Altermundi
Comunidad Trabajo y Organización
(CTO)
Las calles, Argentina
fabrizio.puzio@altermundi.net

José Amado
Instituto Nacional de Tecnología
Industrial (INTI)
Universidad Nacional de Córdoba
Córdoba, Argentina
jamado@inti.gob.ar

Resumen— Los pequeños productores de la Agricultura Familiar Campesina e Indígena (AFCI) de gallinas y otras aves se encuentran en una encrucijada en la reproducción de genética avícola: las incubadoras, tanto importadas como nacionales, son demasiado costosas o bien no cuentan con prestaciones adecuadas y no pueden adaptarse a las necesidades requeridas (optimización de la producción, seguimiento de variables, alertas, etc.).

Por otro lado, las personas más jóvenes residentes en zonas rurales sufren desarraigo y existe la concepción de que no se puede desarrollar tecnología en estos sitios, alejados de las grandes urbes.

Para mitigar estas problemáticas, se desarrolla el presente proyecto, que consiste en diseñar una incubadora de huevos con tecnología (hardware y software) totalmente abierta y libre, con tecnología apropiada [1] que pueda adaptarse y producirse en estas zonas rurales. El grupo de trabajo está integrado por ingenieros y jóvenes recién egresados de escuelas secundarios de zonas rurales de las sierras de Córdoba.

Palabras clave: Tecnología Apropriada, LUA, ESP32, Incubadora.

I. INTRODUCCIÓN

Desde AlterMundi se viene acompañando procesos de acceso a conectividad en general y en particular en la ruralidad, no sólo como un fin en sí mismo, sino como un puente para que las comunidades se construyan como sujetos integrales, accediendo a derechos fundamentales como salud y educación. Se trata que las poblaciones rurales no sean meras

consumidoras sino productores protagonistas del mundo digital y de Internet en particular.

Uno de los objetivos de este proyecto es fomentar el arraigo rural y la generación de puestos de trabajo vinculando oficios clásicos, como herrería y carpintería, con tecnologías de actualidad, como electrónica y software.

Partiendo de la experiencia en el desarrollo de #LibreRouter [2], se establecieron vínculos con organizaciones de la Agricultura Familiar, Campesina e Indígena (AFCI) en la construcción de Redes Comunitarias de Internet, y a partir de allí se pudo diagnosticar, entre otros, problemas de acceso a agua, energía eléctrica y vías de comunicación adecuados. En particular se detectó la falta de acceso a Tecnología Apropriada e Innovación Tecnológica [1], tal que respondan a las necesidades situadas y reales de cada contexto. La tecnología adecuada (TA), también conocida como **tecnología apropiada** o tecnología intermedia, es aquella tecnología que está diseñada con especial atención a los aspectos medioambientales, éticos, culturales, sociales y económicos de la comunidad a la que se dirigen. Atendiendo a estas consideraciones, las TA emergen del medio local, y normalmente demandan menos recursos, son más fáciles de mantener, presentan un menor coste y un menor impacto sobre el medio ambiente respecto a otras tecnologías equiparables. [1]

En Argentina, como en muchos países de Latinoamérica, la producción de huevos y de carne de pollo es la base de la producción animal para las familias con poca extensión de tierra. Garantiza un autoconsumo saludable y constituye un bien fácil de comercializar e intercambiar (fundamentalmente por ser una

cadena de retorno corto, dos meses en el caso de la producción de carne y 5 meses en el caso de producción de huevos). Fortalecer la producción en Incubadoras pensadas para el sector de la AFCl impactará en la producción de genética propia adaptada a cada región. Tal es el caso de Comunidad, Trabajo y Organización (CTO), que seleccionó la genética de Ponedora Negra (desarrollada por el Instituto Nacional de Tecnología Agropecuaria) por su resistencia al frío y las condiciones regionales. Recuperar la genética de carne criolla, con sus razas pesadas específicas, es la base para generar diversidad y potenciar la producción propia avícola de este sector, que históricamente produce en condiciones ambientalmente sanas y sustentables, sin hacinamientos ni maltrato animal, sin sobre medicaciones ni enfermedades por confinamiento.

Para solucionar estas situaciones, se siguieron dos líneas de trabajo: armar un equipo de trabajo propio, con capacidades en Software y Hardware libre, y comenzar diálogos con otras organizaciones e instituciones para garantizar el acceso a las tecnologías y la utilización de avances tecnológicos de la población. Se busca disminuir la brecha digital que reproduce y amplía las desigualdades sociales.

En este trabajo se describen dos proyectos que se desarrollaron en AlterMundi para fortalecer la búsqueda de Soberanía Alimentaria en una cadena específica: la avícola. Ambos proyectos de implementaron completamente con herramientas de uso libre y gratuito, tanto en hardware como software. El proceso de producción y fabricación se diseñó para no requerir de maquinaria especializada. Toda la información está publicada en un repositorio de acceso libre.

II. OBJETIVOS DEL PROYECTO

Para enfocar el diseño se utilizó específicamente el territorio de la Provincia de Córdoba (Argentina) para caracterizar la demanda. En este territorio y en base a un relevamiento realizado en conjunto con el Ministerio de Agricultura de dicha provincia se pudieron determinar los siguientes aspectos:

- Existe una necesidad real de incubadoras accesibles y eficaces en todos los sectores

(familias agricultoras, cooperativas, pequeños y medianos productores).

- Las demandas son dispares y responden a las escalas productivas: las familias agricultoras necesitan pequeños dispositivos (80 a 150 huevos) para reposición intrafamiliar; las cooperativas necesitan incubadoras de entre 500 y 1200 huevos; los productores pequeños y medianos trabajan con 1200 huevos o más.

- En cada una de las 10 regiones de la provincia, se puede estimar una demanda aproximada de Incubación de 2500 huevos por mes para ponedoras, 2500 huevos por mes para carne (no hay genética propia de las organizaciones, están en proceso de desarrollo).

- Existen muchos emprendimientos productivos que se beneficiarían de un sistema de monitoreo genérico, accesible y adaptable.

Partiendo de estas caracterizaciones, se decidió desarrollar una Incubadora de 150 huevos de capacidad total. Este modelo propone en un ciclo de 21 días, una carga de 50 huevos semanal y un nacimiento de 50 huevos por semana a partir del día 21 (con un 80% estimado de eclosión). Para una Incubadora, una familia con apenas 10 gallinas (con una postura del 70%) y 2 gallos de buena genética son suficientes para realizar las cargas. La capacidad de la incubadora es fácilmente escalable, dado que está diseñada para armar "racks" y unir módulos, brindando así soluciones a un sector productivo muy amplio.

En paralelo y como derivado de este diseño se plantea el diseño de un sistema de monitoreo genérico con capacidad de sensor concentración de CO₂, temperatura y humedad para dar solución a algunas de las necesidades detectadas en la encuesta.

En base al diagnóstico descrito, se buscaron los siguientes objetivos específicos:

- A) Propiciar el acceso a Tecnología Apropia y Libre para los sectores más desfavorecidos de la sociedad, acorde a sus necesidades e involucrando a los usuarios como protagonista en el proceso de desarrollo y planificación.
- B) Garantizar el uso sustentable, el reciclado y el desarme, evitando la obsolescencia programada y el impacto ambiental.
- C) Innovar en sectores productivos específicos, acorde a las problemáticas

propias de cada sector o cadena, y mejorando las condiciones de los productores rurales,

- D) Reivindicar la cultura del Trabajo Rural y la Ruralidad, construyendo arraigo de jóvenes en el sector, especialmente de mujeres, generando condiciones de vida y de trabajo dignas, amables y modernas.
- E) Incluir jóvenes con perfiles de programadores o técnicos en el rol de estudiantes/aprendices, que puedan vivenciar experiencias ligadas a los sectores de la AFCEI.
- F) Articular con otras organizaciones e instituciones, públicas y comunitarias, tanto del sector de la AFCEI como del ámbito de Ciencias y Técnica, para potenciar, validar y mejorar las propuestas de desarrollo y brindar acompañamiento productivo específico.
- G) Acompañar a organizaciones en la implementación de procesos sistematizados de producción, o bien, mejorar los que ya existen.

III. DISEÑO, DESARROLLO Y VALIDACIÓN

En el presente proyecto se construyeron dos productos que comparten la misma electrónica: un sistema de control (la incubadora de huevos) y un sistema de monitoreo. El sistema de control tiene sensores y actuadores mientras que el sistema de monitoreo tiene solo sensores. El desarrollo de la incubadora tuvo en cuenta el diseño descrito en [3], introduciendo varias modificaciones.

Se utilizó un proceso de desarrollo iterativo incremental de prototipos, en los que se fueron validando diferentes elementos y subsistemas.

A. Prototipo 0

Sirvió de entrenamiento para los jóvenes, construido con placas MDF y uniones con tornillos, con un sistema de rotación de huevos por biela manivela.

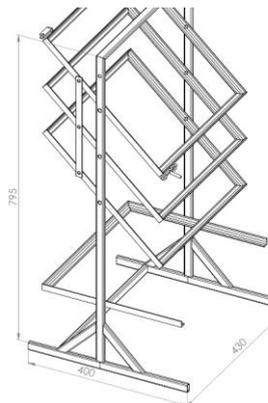


Fig. 1. Sistema de sujeción de bandejas.

En esta primera etapa se prototipo la electrónica en una placa multipropósito. Se utilizó un microcontrolador ESP32-CAM (se prevé incorporación de visión artificial en futuros proyectos), una placa de relay de cuatro canales, un sensor de temperatura y humedad BME280. Se compararon, teórica y empíricamente, el DHT22 y BME280, seleccionando el segundo debido a sus mejores prestaciones, menor dispersión y mayor exactitud. En las comparaciones se utilizó un termómetro con certificado de calibración (trazabilidad) como elemento de referencia.

El control de potencia es on-off y el calentamiento se realiza a través de una resistencia. El software presenta un esquema de módulos: módulo controlador, módulo incubador y módulo de envío de información (para visualización en servidor web).

El servidor elegido fue Grafana, ya que es de software libre y permite visualizar de manera ágil y clara el estado de cada incubadora.

El software del microcontrolador fue programado en LUA utilizando VSCodium [2]. Este es un lenguaje interpretado similar a Python, pero mucho más liviano y utiliza un concepto de tablas. Se empleó un intérprete de LUA para ESP32, el cual fue provisto por el proyecto Nodemcu [3]. LUA está orientado a tareas y las interrupciones están gestionadas por el intérprete como otra tarea.

El firmware cuenta con pruebas unitarias implementadas con el framework Busted [4]. Al utilizar Lua y Busted se puede correr el código en PC y allí validar gran parte del firmware antes de enviarlo al microcontrolador. El desarrollo fue registrado en un proyecto de Git [5].

Este primer prototipo sirvió para validar las dimensiones, la homogeneidad de la

temperatura en la totalidad del gabinete y el sistema de calefacción.

B. Prototipo 2

Se evolucionó el software, se cambiaron los repositorios e incluye una aplicación multiplataforma. A los módulos existentes se le agregaron un módulo de configuración y un módulo de servidor de Api Rest. Este último permite que la aplicación se comuniqué con la incubadora. La aplicación está escrita en Flutter, que presenta la ventaja de que la misma aplicación se puede desplegar sin mayor esfuerzo en forma de página web, aplicación Android o aplicación Linux. En este caso, se utiliza Android.

C. Prototipo 3

Se refinaron el sistema de control, los materiales de la carcasa y el sistema de movimiento de huevos. Este último era difícil de replicar por lo que se cambió por un sistema de correa con un motor que invierte su giro y un par de finales de carrera.

El sistema de control temperatura ahora es PID y la conmutación del elemento calefactor se realiza mediante Triac. Para esto, se requirió escribir parte del código en C++ y luego exportar a Lua.

El firmware tiene las siguientes funcionalidades:

- Control de temperatura y humedad con parámetros configurables.
- Registro de tiempos de incubación por bandejas.
- Envío de alertas al teléfono:
 - o Detección de no cambio de temperatura a pesar de tener encendido el calentador.
 - o Detección de falla de sensor.
 - o Falta de movimiento de los huevos.
 - o Temperaturas fuera de rango.
 - o Fecha en la que es necesario retirar los huevos y pasarlos a nacedora.
- Envío de información al servidor
- Configuración de WiFi en modo access point para uso del Api Rest.
- Configuración del access point para acceso a internet.

La cantidad de líneas de código de Lua se puede ver en la tabla 1.

TABLA I

TAMAÑO DE ARCHIVOS EN LUA EN LÍNEAS DE CÓDIGO

Loc	Nombre del archivo
300	Incubadora
180	Control
43	Alertas y envío de datos
100	Configuración
95	Servicio REST
189	Configuración WIFI

En este prototipo se diseñó una placa de circuito impreso (PCB), lo cual aportó robustez y confiabilidad.

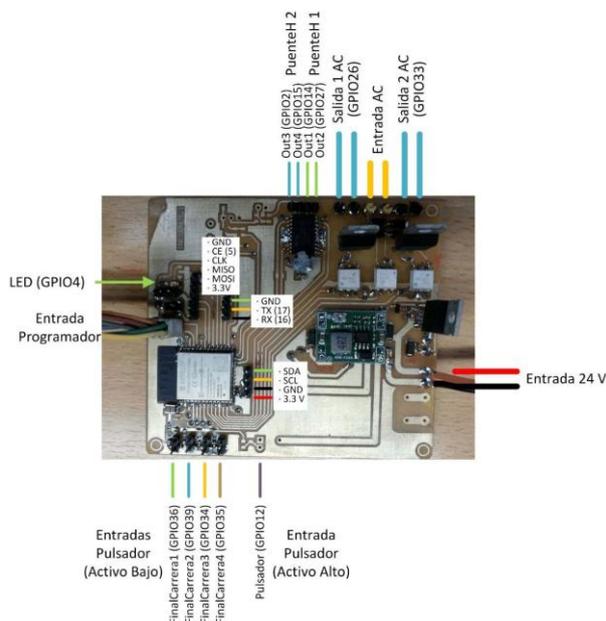


Fig. 2. Placa con indicaciones de uso de cada pin, cuatro salidas para controlar dos motores, dos salidas de potencia controladas por triac, entradas de alimentación, entrada para pulsador, 4 entradas para finales de carrera, conector para programador y para I2C.

D. Prototipo 4

La carcasa de este prototipo es de T-plac [8] un material construido a partir de cartones para bebidas usados.



Fig. 3. Placa de TPLAC con mejor aislamiento y además está fabricada con materiales reciclados.

Se modificó el PCB [9] para mejorar trazado y ancho de pistas, el recorrido y ancho de las pistas de GND y además se pusieron en práctica

las sugerencias del fabricante para mejorar el alcance de la antena WI-FI [10]. Por último estos cambios también aseguran y simplifican la fabricabilidad manual de estas placas mediante la técnica de ácido y revelado (esto permite que se pueda fabricar en cualquier lugar del país).

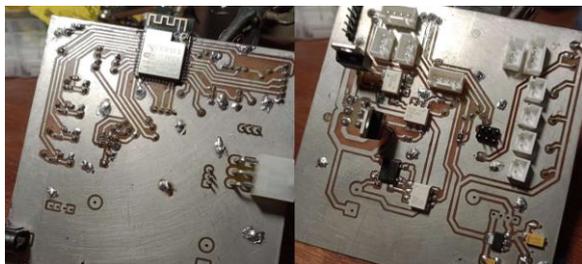


Fig. 4. Nuevas placas con componentes en ambas caras para disminuir el tamaño. Relocalización del microcontrolador para mejorar el rendimiento de la antena.

La composición final de la incubadora es la que se muestra en la fig. 5 y en la fig. 8. Esta cuenta con un ventilador para mantener la temperatura y la humedad homogéneas, tres bandejas porta huevos de 49 huevos cada una, un sensor de temperatura y humedad en el centro de la incubadora. Una bandeja nacedera donde se depositan los huevos para que nazcan los pollitos, un vaporizador para generar humedad y una resistencia para controlar la temperatura.

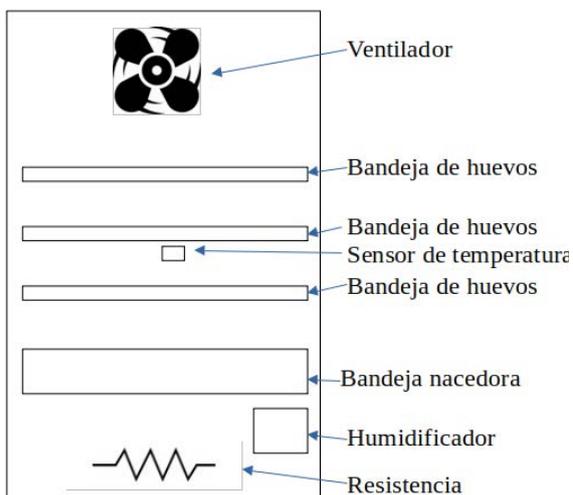


Fig. 5. Diagrama de implementación final de la incubadora

Actualmente los datos de la incubadora están siendo enviados a grafana Fig 6 y 7. Altermundi cuenta con una instancia propia de grafana que luego será utilizada para poder monitorear las incubadoras de manera remota.

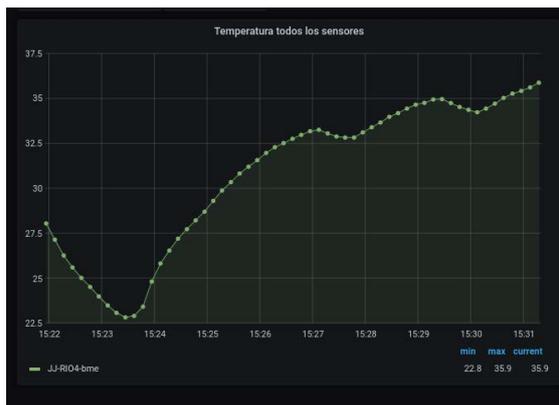


Fig. 6. curva de temperatura de una incubadora apagada y luego encendida a las 15:23



Fig. 7. Curva de humedad de una incubadora apagada y luego encendida a las 15:23

El equipo de trabajo puede verse en la fig 8. de izquierda a derecha: Fabricio, Pablo, Javier, Nico, Santi y Annie. En la foto faltan Jere y Fede.



Fig. 8. Incubadora y equipo de trabajo

IV. CONCLUSIONES

Se logró desarrollar una incubadora con herramientas libres fácil de producir, flexible y accesible a pequeños productores rurales y con un impacto ambiental reducido.

Se comprobó que LUA es un lenguaje adecuado y versátil para realizar desarrollos propios. No es necesario compilarlo y permite ejecutar comandos en tiempo real, similar a una consola en PC lo que permite acelerar la curva



de aprendizaje. Se observó la desventaja que en ocasiones se debe traducir código de C/C++ a LUA, puesto que no es muy difundido.

Se conformó un equipo de personas que realizará el ensamblado/producción y se ha implementado un lugar de trabajo para esta etapa, en conjunto con Cooperativa de Traslasierra (CTO).

Se lograron importantes mejoras en el equipo de desarrollo: coordinación, gestión, interacción. El proyecto se inició con un grupo de personas trabajando con escasa coordinación y se culminó con un grupo de trabajo totalmente coordinado, donde cada desarrollador cumple funciones bien determinadas y aporta lo mejor de sí para el proyecto, integrando jóvenes y adultos con distintos grados de formación.

Para que esta propuesta de innovación tecnológica funcione en el sector de la AFCE se requiere una organización comunitaria que lo sustente, prácticas productivas amigables con el medio ambiente y las personas.

Actualmente, se está finalizando el Prototipo 4 y se espera producir una primera serie de 10 incubadoras a finales de septiembre de 2024. En esta primera serie se intenta corregir algunos problemas de producción (manuales de armado, etc.). Con el primer sistema ensamblado comenzaran las capacitaciones a los usuarios de estas 10 incubadoras.

Se propuso a las comunidades encarar un proyecto técnico/productiva de realizar una mejora genética de gallinas criollas para la producción de carne, Si se logra esto, se estará dando importantes pasos en la búsqueda de la soberanía alimentaria y tecnológica brindando a nuestros pueblos alimentos sanos y de trabajo local.

Este desarrollo creado con herramientas libres sirve de base para pensar aplicaciones en otros procesos productivos, tales como el cultivo de gírgolas, plantineras, producción de alimentos (quesos y embutidos), producción y secado de tabaco, etc.

REFERENCIAS

- [1] "Tecnología apropiada," Wikipedia, última modificación el 15 de octubre de 2023. [Enlace: https://es.wikipedia.org/wiki/Tecnolog%C3%ADa_adeuada]
- [2] "LibreRouter: un enrutador inalámbrico multiradio para redes comunitarias," Internet Society, 12 de diciembre de 2018. [Enlace: <https://www.internetsociety.org/es/blog/2018/12/librerouter-un-enrutador-inalambrico-multiradio-para-redes-comunitarias/>]
- [3] L., Niranjan & Venkatesan, Cherappa & a R, & Satheeskumaran, Satheeskumaran & Nawaz, Sunwan.

- (2021). Design and implementation of chicken egg incubator for hatching using IoT. International Journal of Computational Science and Engineering. 24. 10.1504/IJCS.2021.10039967..
- [4] VSCodium, "VSCodium," VSCodium, 2023. [Online]. Available: <https://vscodium.com/>. [Accedido: 15 de agosto 2024]
- [5] NodeMCU, "NodeMCU Documentation," NodeMCU, n.d. [Online]. Available: <https://nodemcu.readthedocs.io/en/dev-esp32/>. [Accedido: 15 de agosto 2024]
- [6] Lunar Modules, "Busted," GitHub, 2023. [Online]. Available: <https://lunarmodules.github.io/busted/>. [Accedido: 15 de agosto 2024]
- [7] AlterMundi-MonitoreoyControl, "Proyecto-Incubadora," GitHub, n.d. [Online]. Available: <https://github.com/AlterMundi-MonitoreoyControl/Proyecto-Incubadora>. [Accesed: 15 de agosto 2024]
- [8] "Packaging Reciclado TPlak," CAUH, 07-Dec-2015. [Online]. Available: <https://www.cauh.co/single-post/2015/12/07/Packaging-Reciclado-TPlak>. [Accesed: 24-Apr-2023].
- [9] "Esquemáticos y PCB" GitHub. [Online]. Available: https://github.com/AlterMundi-MonitoreoyControl/Proyecto-Incubadora/tree/main/HardWare/Placas/Olivia_control/v0.1. [Accesed: 15 de agosto 2024].
- [10] "PCB Layout Design," Espressif Systems, [Online]. Available: <https://docs.espressif.com/projects/esp-hardware-design-guidelines/en/latest/esp32/pcb-layout-design.html#positioning-a-module-on-a-base-board>. [Accesed: 15 de agosto 2024].



Regulador de Potencia para Láseres

Lucas Adriel Cusit

Universidad Tecnológica Nacional

Facultad Regional Paraná

Paraná, Argentina

Cusitulucas2001@gmail.com

Resumen— El control preciso de la potencia de un láser es fundamental en aplicaciones críticas como la fabricación de piezas, la comunicación por fibra óptica y la investigación científica. Sin embargo, uno de los desafíos más persistentes es la alinealidad de la potencia lumínica emitida por el láser, lo que puede comprometer la precisión y la eficiencia de estos sistemas. Este artículo presenta el desarrollo de un regulador de potencia basado en modulación por ancho de pulso (PWM), controlado por un microcontrolador ESP32, diseñado específicamente para abordar este problema en láseres.

El dispositivo permite ajustes finos del ciclo de trabajo del PWM y muestra el porcentaje en tiempo real, utilizando una interfaz simple pero eficaz. Los resultados experimentales revelan una clara alinealidad en la respuesta del láser al ciclo de trabajo, especialmente a medida que se incrementa, lo que subraya la importancia de calibrar estos sistemas para garantizar su rendimiento óptimo.

Además, se identificó un fenómeno de histéresis relacionado con el calentamiento del láser durante la operación prolongada, lo que añade una capa de complejidad al control de la potencia. Los lectores interesados en mejorar la eficiencia y seguridad en aplicaciones láser encontrarán en este artículo un recurso valioso.

I. INTRODUCCIÓN

El control de la potencia de un láser es una tarea crítica en una amplia gama de aplicaciones industriales, científicas y médicas. Desde la fabricación y el corte de materiales, la comunicación por fibra óptica, hasta experimentos precisos en laboratorios, la capacidad de ajustar y mantener la potencia del láser con precisión es esencial para garantizar la calidad y la seguridad de los procesos. Sin

embargo, uno de los desafíos más importantes en el control de láseres es la alinealidad de la potencia emitida y la potencia lumínica real del láser.

Esta alinealidad puede tener un impacto significativo en el rendimiento de los sistemas que dependen de un control preciso de la potencia. Si la respuesta del láser no es lineal, los ajustes realizados no producirán los cambios esperados, lo que podría llevar a una pérdida de precisión en aplicaciones como el corte y grabado de materiales, así como en la transmisión de datos por fibra óptica. En el caso de las fibras ópticas, una potencia láser no lineal impediría alcanzar un índice de modulación óptimo, lo que afectaría negativamente características fundamentales del sistema de comunicaciones, como la distancia máxima de transmisión y la robustez frente al ruido generado por los circuitos electrónicos. Con una alinealidad controlada, se logran mayores amplitudes de modulación, lo que permite al sistema de comunicaciones alcanzar mayores distancias y una mejor calidad de la señal [3].

El proyecto descrito en este trabajo aborda estos desafíos mediante la creación de un regulador de potencia utilizando modulación por ancho de pulso (PWM) controlado por un microcontrolador. Este regulador está diseñado específicamente para láseres y tiene como objetivo controlar la intensidad del láser para poder medir su potencia lumínica, demostrando que, en la práctica, la relación entre la energía consumida y la potencia emitida por el láser no es lineal, a pesar de que teóricamente se podría esperar que lo fuera. Comprender y calibrar esta alinealidad no solo es crucial para mejorar la precisión y la eficiencia en las aplicaciones del láser, sino que también es vital para garantizar la seguridad y la caracterización

adecuada de las capacidades del láser en investigación y desarrollo.

II. PARTES DEL CONTROLADOR

Este dispositivo está equipado con cinco pulsadores que permiten ajustar el duty cycle (ciclo de trabajo) del PWM (Pulse Width Modulation) de las siguientes maneras: aumentar en 1% y 10%, disminuir en 1% y 10%, y reiniciar el ciclo de trabajo del PWM. (Ver Fig. 1).

El dispositivo cuenta con tres displays de 7 segmentos, cada uno con su respectivo codificador BCD (Binary Coded Decimal) a 7 segmentos, para mostrar el porcentaje de PWM (ver Fig. 2). Para controlar el láser, se utiliza un MOSFET (Metal-Oxide-Semiconductor Field-Effect Transistor) (ver Fig. 3).

Finalmente, el microcontrolador empleado es el NodeMCU ESP32 (ver Fig. 3).

III. CÓDIGO FUENTE

El microcontrolador se programa utilizando ESP-IDF (Espressif IoT Development Framework), el framework oficial de Espressif Systems para desarrollar software para los microcontroladores ESP32 de Espressif.

En la Fig. 4, se definen los pines y configuraciones necesarias para el control del láser y los botones. Esto incluye la configuración de temporizadores, modos de operación, resoluciones de duty cycle o ciclo de trabajo, y la declaración de funciones para escribir valores BCD y manejar la lógica de control de botones. Además, se configuran los pines GPIO utilizados para diferentes botones y la salida de control del láser.

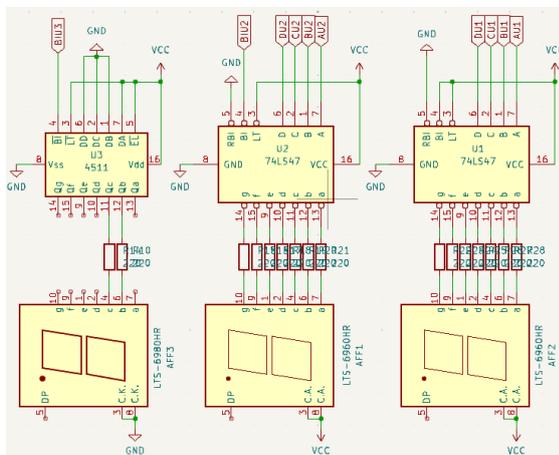


Fig.2.Displays y codificadores.

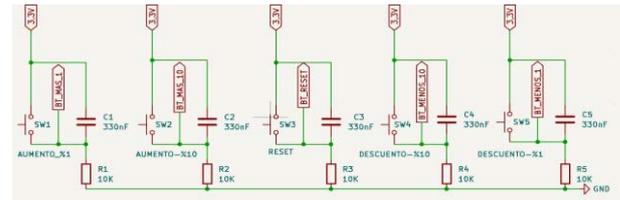


Fig.1.Pulsadores de control.

Los aspectos más relevantes de esta sección son las macros: **LEDC_TIMER_0**, que indica el uso del timer 0 del ESP32; **LEDC_LOW_SPEED_MODE**, que define un modo de velocidad baja para PWM, permitiendo una resolución más precisa en velocidades bajas; y **LEDC_TIMER_8_BIT**, que establece una resolución de 8 bits para el ciclo de trabajo en el timer configurado.

A continuación, se explicarán las funciones declaradas en la Fig.4.

A. Función "configuracion_bcd_pins"

En la Fig.5 se puede apreciar la función en la que se configuran los pines declarados en los vectores **bcd_pins_u** y **bcd_pins_d** como salidas. Estos pines proporcionan los números BCD a los codificadores de las unidades y decenas, respectivamente, del valor del ciclo de trabajo.

```

1  #ifndef FUNCIONES_H
2  #define FUNCIONES_H
3
4  #include <stdio.h>
5  #include "driver/ledc.h"
6  #include "esp_err.h"
7  #include "freertos/FreeRTOS.h"
8  #include "freertos/task.h"
9  #include "driver/gpio.h"
10
11 #define SALIDA_TIMER          LEDC_TIMER_0
12 #define SALIDA_MODE          LEDC_LOW_SPEED_MODE
13 #define SALIDA_DUTY_RES      LEDC_TIMER_8_BIT
14 #define BT_MAS_10            GPIO_NUM_33
15 #define BT_MENOS_10         GPIO_NUM_35
16 #define BT_MAS_1             GPIO_NUM_25
17 #define BT_MENOS_1          GPIO_NUM_34
18 #define BT_RESET             GPIO_NUM_32
19 #define SALIDA_CHANNEL       LEDC_CHANNEL_0
20 #define SALIDA                GPIO_NUM_26
21 #define NUM_DIGITS           4
22 #define bcd_pin_c             GPIO_NUM_22
23 #define activar_display      GPIO_NUM_18
24
25 extern int bcd_pins_u[NUM_DIGITS];
26 extern int bcd_pins_d[NUM_DIGITS];
27
28 void configuracion_bcd_pins();
29 void escribir_bcd(int value, int *v);
30 void BOTON(void);
31 void CONTROLADOR(void);
32 int control_duty(int sum_f, int sum_g);
33
34 #endif

```

Fig.4.Header donde se declaran funciones, variables globales, vectores y pines.

Además, se configura el pin **bcd_pin_c** como salida. Este pin activa o desactiva el pin 4 del codificador, el cual está conectado de tal manera que muestra el número uno en el display (ver Fig.2). Este pin indica la centena del ciclo de trabajo, que va del 0% al 100%. Por ello, el número BCD se introduce directamente en el circuito como se muestra en la Fig.2, en el codificador U3. Esto permite que, cuando el ciclo de trabajo es menor al 100%, el display correspondiente esté apagado.

Finalmente, se configura el pin **activar_display**, que enciende o apaga el display mediante el pin 4 del codificador U2 si el ciclo de trabajo es menor que el 10% (ver Fig.2).

B. Función "escribir_bcd"

En la Fig. 6 se muestra cómo se convierte un número entero en un número BCD y se asignan los valores a las salidas correspondientes mencionadas anteriormente.

C. Función "BOTON"

En esta función se configuran los pines como entrada, correspondientes a los pulsadores y se activa solamente la resistencia de pull_down, para tener una correcta lectura (ver Fig.7).

D. Función "CONTROLADOR"

Primero, esta función (ver Fig.8) configura el temporizador en modo de baja velocidad (LEDC_LOW_SPEED_MODE), utilizando el temporizador 0 (LEDC_TIMER_0) con una resolución de 8 bits (LEDC_TIMER_8_BIT) y una frecuencia de 10 kHz, permitiendo que el ESP32 seleccione automáticamente la fuente de reloj más adecuada (LEDC_AUTO_CLK).

Esta configuración se aplica con **ledc_timer_config(&LASER_timer)**, y **ESP_ERROR_CHECK** asegurando que no haya errores.

```

1 int bcd_pins_u[NUM_DIGITS] = { GPIO_NUM_16, GPIO_NUM_2, GPIO_NUM_0, GPIO_NUM_4 };
2 int bcd_pins_d[NUM_DIGITS] = { GPIO_NUM_17, GPIO_NUM_21, GPIO_NUM_19, GPIO_NUM_5 };
3
4 void configuracion_bcd_pins() {
5     gpio_set_direction(bcd_pin_e, GPIO_MODE_OUTPUT);
6     gpio_set_direction(activar_display, GPIO_MODE_OUTPUT);
7     for (int i = 0; i < NUM_DIGITS; i++) {
8         gpio_set_direction(bcd_pins_u[i], GPIO_MODE_OUTPUT);
9         gpio_set_direction(bcd_pins_d[i], GPIO_MODE_OUTPUT);
10    }
11 }
12
13

```

Fig.5.Funcion donde se configura la salida BCD.

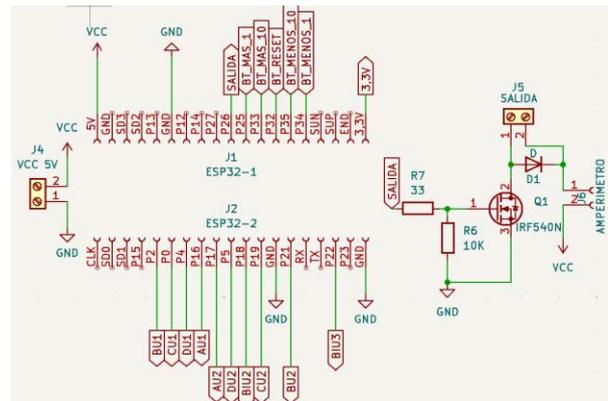


Fig.3.Entradas y salidas del ESP_32. Conexión de salida.

Luego, configura el canal para usar el temporizador, configurado previamente, asignando el GPIO especificado para la salida PWM (SALIDA), deshabilitando las interrupciones (LEDC_INTR_DISABLE), e inicializando el ciclo de trabajo a 0%. También se aplican las funciones **ledc_channel_config(&laser_channel)**, y **ESP_ERROR_CHECK** asegurando que no haya errores (ver Fig.8).

En resumen, esta función prepara el sistema para generar señales PWM que controlarán un dispositivo conectado al GPIO especificado.

E. Función "control_duty"

Esta función convierte los valores de los sumadores **sum_f** (que es el contador relacionado con el decremento o incremento del ciclo de trabajo en 1%) y **sum_g** (que se relaciona con el incremento o decremento del ciclo de trabajo en 10%) a valores porcentuales de 8 bits. Multiplica los valores por 256 y los divide por el porcentaje correspondiente para escalarlos adecuadamente. Luego, suma estos valores escalados para obtener un total, el cual se limita a 256 para asegurar que el ciclo de trabajo no exceda el 100%. Finalmente, devuelve el valor calculado (ver Fig.9).

```

15 void escribir_bcd(int value, int *v) {
16     if (value < 10) {
17         for (int i = 0; i < NUM_DIGITS; i++) {
18             int bit_value = (value >> i) & 1;
19             gpio_set_level(v[i], bit_value);
20         }
21     } else {
22         for (int i = 0; i < NUM_DIGITS; i++) {
23             gpio_set_level(v[i], 0);
24         }
25     }
26 }

```

Fig.6. Función donde se escriben valore a un formato BCD.

F. Función principal "main"

En la Fig.10 se observa la primera parte de la función principal, donde se inicializan las funciones CONTROLADOR, BOTON, y configuracion_bcd_pins. Se utilizan dos variables enteras, **sum_f** y **sum_g**, como contadores para representar las unidades y decenas respectivamente. Estos contadores se actualizan en tiempo real según la interacción del usuario con los botones físicos (BT_MAS_10, BT_MAS_1, BT_MENOS_10, BT_MENOS_1, BT_RESET). Cada vez que se presiona alguno de estos botones, se ejecuta una acción específica: incrementar o decrementar los contadores, o reiniciarlos, como se muestra en la Fig.10.

Se utilizan estructuras condicionales para garantizar que los valores permanezcan dentro de un rango válido (0 a 9 para cada dígito), y se actualiza la salida de un canal PWM utilizando las funciones **ledc_set_duty**, la cual tiene 3 argumentos, el primero es el modo de salida, el segundo el canal deseado y por ultimo el ciclo de trabajo deseado, que en este caso se utiliza la función **control_duty** (ver Fig.11).

Además del control de los contadores para la salida PWM, el código gestiona la activación de los display BCD y escribe los valores actuales de **sum_f** y **sum_g** en los pines correspondientes (ver Fig.11).

```

28 void BOTON(void) {
29     gpio_set_direction(BT_MAS_10, GPIO_MODE_INPUT);
30     gpio_set_direction(BT_MAS_10, GPIO_PULLDOWN_ONLY);
31     gpio_pulldown_en(BT_MAS_10);
32     gpio_set_direction(BT_MENOS_10, GPIO_MODE_INPUT);
33     gpio_set_direction(BT_MENOS_10, GPIO_PULLDOWN_ONLY);
34     gpio_pulldown_en(BT_MENOS_10);
35     gpio_set_direction(BT_MAS_1, GPIO_MODE_INPUT);
36     gpio_set_direction(BT_MAS_1, GPIO_PULLDOWN_ONLY);
37     gpio_pulldown_en(BT_MAS_1);
38     gpio_set_direction(BT_MENOS_1, GPIO_MODE_INPUT);
39     gpio_set_direction(BT_MENOS_1, GPIO_PULLDOWN_ONLY);
40     gpio_pulldown_en(BT_MENOS_1);
41     gpio_set_direction(BT_RESET, GPIO_MODE_INPUT);
42     gpio_set_direction(BT_RESET, GPIO_PULLDOWN_ONLY);
43     gpio_pulldown_en(BT_RESET);
44 }
    
```

Fig.7. Función donde se configuran los pines de pulsadores.

```

68 int control_duty(int sum_f, int sum_g) {
69     int aux_g = 0;
70     int aux_f = 0;
71     int total = 0;
72     aux_g = (sum_g * 256) / 10;
73     aux_f = (sum_f * 256) / 100;
74     total = aux_f + aux_g;
75     if (total > 256) {
76         total = 256;
77     }
78     return total;
79 }
    
```

Fig.9. Función donde se escala los valores enteros a valores porcentuales de 8 bit.

```

46 void CONTROLADOR(void) {
47     ledc_timer_config_t LASER_timer = {
48         .speed_mode = SALIDA_MODE,
49         .timer_num = SALIDA_TIMER,
50         .duty_resolution = SALIDA_DUTY_RES,
51         .freq_hz = 10000,
52         .clk_cfg = LEDC_AUTO_CLK
53     };
54     ESP_ERROR_CHECK(ledc_timer_config(&LASER_timer));
55
56     ledc_channel_config_t laser_channel = {
57         .speed_mode = SALIDA_MODE,
58         .channel = SALIDA_CHANNEL,
59         .timer_sel = SALIDA_TIMER,
60         .intr_type = LEDC_INTR_DISABLE,
61         .gpio_num = SALIDA,
62         .duty = 0,
63         .hpoint = 0
64     };
65     ESP_ERROR_CHECK(ledc_channel_config(&laser_channel));
66 }
    
```

Fig.8. Función donde se configura el canal y pin de salida.

IV. RESULTADOS DE LA IMPLEMENTACIÓN

El análisis de la gráfica (ver Fig.12) revela una clara alinealidad en la respuesta de la potencia del láser a las variaciones del ciclo de trabajo del PWM. Se observa que tanto en el incremento (gráfica naranja, 0%-100%) como en el decremento (gráfica azul, 100%-0%), la relación entre el %PWM y la potencia lumínica no es lineal. Pero también sucede que hay una diferencia en la potencia lumínica según se decrementa o incrementa el ciclo de trabajo.

Este comportamiento es particularmente interesante, ya que se repite en ambas situaciones: al incrementar el %PWM, la potencia lumínica es mayor, probablemente debido a que el láser opera a una temperatura más baja.

```

1 #include "funciones.h"
2
3 void app_main() {
4     CONTROLADOR();
5     BOTON();
6     configuracion_bcd_pins();
7     int sum_g = 0;
8     int sum_f = 0;
9
10     while (1) {
11
12         if ( gpio_get_level(BT_MAS_10) == 1) {
13             sum_g++;
14             if (sum_g >= 10) {
15                 sum_g = 10;
16                 sum_f = 0;
17             }
18
19             ESP_ERROR_CHECK(ledc_set_duty(SALIDA_MODE, SALIDA_CHANNEL, control_duty(sum_f, sum_g)));
20             ESP_ERROR_CHECK(ledc_update_duty(SALIDA_MODE, SALIDA_CHANNEL));
21         }
22         else if (gpio_get_level(BT_MAS_1) == 1) {
23             sum_f++;
24             if (sum_f >= 10) {
25                 sum_g++;
26                 sum_f = 0;
27             }
28             else if (sum_g == 10) {
29                 sum_f = 0;
30             }
31
32             ESP_ERROR_CHECK(ledc_set_duty(SALIDA_MODE, SALIDA_CHANNEL, control_duty(sum_f, sum_g)));
33             ESP_ERROR_CHECK(ledc_update_duty(SALIDA_MODE, SALIDA_CHANNEL));
34         }
35     }
    
```

Fig.10

Si el láser opera al 100% durante un período prolongado, su potencia caerá hasta el nivel máximo reflejado en la gráfica azul. Al disminuir el %PWM, la potencia seguirá el comportamiento mostrado en dicha gráfica, lo que sugiere un cambio en el comportamiento del láser con el tiempo. Si se incrementa nuevamente el ciclo de trabajo, el láser recuperará mayor potencia y seguirá el patrón de la gráfica naranja. Este ciclo se repetirá constantemente.

Entonces con estos resultados vemos que el laser no se comporta de una manera lineal y que según la temperatura de este puede aumentar o disminuir la potencia lumínica entregada por él, estas son cosas para tener en cuenta a la hora de construir algo que requiera precisión.

Para realizar estas mediciones, se utilizó el láser [1] conectado al controlador y el medidor de potencia lumínica [2].

```

35     else if (gpio_get_level(BT_MENOS_10) == 1) {
36         sum_g--;
37         if (sum_g < 0) {
38             sum_g = 0;
39         }
40         else if (sum_g==0 && sum_f>1){
41             sum_f=0;
42         }
43     }
44     ESP_ERROR_CHECK(ledc_set_duty(SALIDA_MODE, SALIDA_CHANNEL, control_duty(sum_f, sum_g)));
45     ESP_ERROR_CHECK(ledc_update_duty(SALIDA_MODE, SALIDA_CHANNEL));
46 }
47     else if (gpio_get_level(BT_MENOS_1) == 1) {
48         if (sum_f == 0 && sum_g > 0) {
49             sum_f = 9;
50             sum_g--;
51         } else if (sum_f > 0) {
52             sum_f--;
53         }
54     }
55     ESP_ERROR_CHECK(ledc_set_duty(SALIDA_MODE, SALIDA_CHANNEL, control_duty(sum_f, sum_g)));
56     ESP_ERROR_CHECK(ledc_update_duty(SALIDA_MODE, SALIDA_CHANNEL));
57 }
58     else if (gpio_get_level(BT_RESET) == 1) {
59         sum_f = 0;
60         sum_g = 0;
61     }
62     ESP_ERROR_CHECK(ledc_set_duty(SALIDA_MODE, SALIDA_CHANNEL, control_duty(sum_f, sum_g)));
63     ESP_ERROR_CHECK(ledc_update_duty(SALIDA_MODE, SALIDA_CHANNEL));
64 }
65 }
66     if (sum_g == 10) {
67         gpio_set_level(bcd_pin_c, 1);
68     }
69     else {
70         gpio_set_level(bcd_pin_c, 0);
71     }
72 }
73     if (sum_g<1 && sum_f<10){
74         gpio_set_level(activar_display, 0);
75     }
76     else{
77         gpio_set_level(activar_display, 1);
78     }
79 }
80     escribir_bcd(sum_f, bcd_pins_u);
81     escribir_bcd(sum_g, bcd_pins_d);
82 }
83     vTaskDelay(pdMS_TO_TICKS(200));
84 }
85 }

```

Fig.11

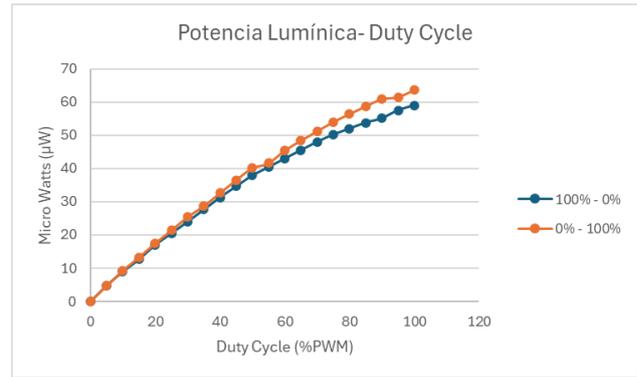


Fig.12. Grafica potencia según el ciclo de trabajo

V. OTROS USOS

Este controlador fue desarrollado específicamente para controlar un láser, pero también puede utilizarse para variar la intensidad de luz de LEDs, ajustar la velocidad de algunos motores de corriente continua, regular la temperatura de resistencias eléctricas y aplicarse en cualquier otra situación donde sea necesario modificar el ciclo de trabajo. Además, el circuito incluye una salida diseñada para conectar un amperímetro y medir la corriente eléctrica consumida por la carga (ver Fig.3).

REFERENCIAS

- [1] Semiconductor.es, "KY-008 Laser transmitter module," [Online]. Available: <https://semiconductors.es/datasheet/KY-008.html#ETC>. Accessed: July 5, 2024.
- [2] Apinex.com, "LP1ES Low Power CMOS Operational Amplifier," [Online]. Available: <https://www.apinex.com/es/lp1es.html>. Accessed: July 5, 2024.
- [3] Salueña, S. A. (2019). Caracterización experimental y modelado de un VCSEL para el diseño de un laser driver con elevada linealidad para aplicaciones de radio sobre fibra con tecnología CMOS nanométrica. Escuela de Ingeniería y Arquitectura de la Universidad Zaragoza.

Sistemas de control para microreactores nucleares

Ruben Roberto Navarro

Juan Carlos Colombo

Viviana nahid

Universidad Tecnológica Nacional

Universidad Tecnológica Nacional

Universidad Tecnológica Nacional

Facultad Regional Tucuman

Facultad Regional Tucuman

Facultad Regional Tucuman

Tucuman, Argentina

Tucuman, Argentina

Tucuman, Argentina

rubenrobertonavarro@gmail.com

colombojcc@gmail.com

img.vnaid@gmail.com

Resumen—En este trabajo se estudia la gran cantidad de desafíos técnicos que se presentan en el diseño de un sistema de instrumentación y control electrónico I&C para microreactores nucleares.

El principal punto para tener en cuenta es que, en este nuevo paradigma de la energía nuclear, los reactores son libres de mantenimiento. Por lo tanto, es deber del sistema de I&C el supervisar el dispositivo a lo largo de su vida útil reemplazando las decisiones humanas. Manteniendo en todo momento los estándares de la seguridad nuclear.

Siendo este trabajo parte del proyecto de investigación "Diseño y Simulación de Sistemas de Control Electrónicos en microgeneradores nucleares y baterías nucleares".

En el mismo se plantea como posible solución la utilización de un sistema basado en inteligencia artificial que mediante redes neuronales sea capaz de cumplir los requerimientos anteriormente citados.

Este sistema sería realizado utilizando controladores tolerantes a fallos, que usarían redes neuronales implementadas mediante hardware en una FPGA, estando en desarrollo las fases de preprocesamiento, entrenamiento y prueba de dichas redes.

Palabras clave: Microreactores Nucleares, Instrumentación y Control, Seguridad Nuclear, Inteligencia Artificial, FPGA.

I. INTRODUCCIÓN

La tecnología de los pequeños reactores

nucleares (SMR), microreactores nucleares (MR) y baterías nucleares se empezó a implementar en los últimos años [1]. El primer SMR avanzado del mundo se conectó a la red eléctrica en 2019 y comenzó a explotarse comercialmente en mayo de 2020.

Tenemos un ejemplo de este tipo de reactores en la central nuclear flotante Akademik Lomonosov que se muestra en la fig. 1, ubicada justo frente a la costa ártica de Rusia, aloja dos unidades SMR KLT40S de 35 MW(e) que están generando suficiente energía para abastecer a una ciudad de alrededor de 100 000 habitantes.



Fig. 1 Central Nuclear Akademik Lomonosov

Siendo esta una de las tecnologías con mayor investigación en la actualidad [2]. Si bien los diseños de SMR suelen basarse en familias de reactores conocidas, los MR utilizan principios de funcionamiento avanzados [3].

Esto es así porque tienen como premisa el ser lo suficientemente pequeños para que toda la central se construya en una fábrica y se puedan

transportar en un camión. Además, se busca que dispongan de sistemas de seguridad pasiva autorregulables.

Finalmente, otro importante objetivo de su diseño es que su operación y mantenimiento sean lo suficientemente simples como para no necesitar personal para su realización.

II. MICROREACTORES NUCLEARES

Estos constituyen junto con las baterías nucleares el nuevo paradigma de la energía nuclear este apareció con el resurgimiento la utilización este tipo de energía motivado por diversos factores geopolíticos tales como:

- Definición de la energía nuclear como energía verde en la Unión Europea en 2020
- Enfrentamiento de la Unión Europea y EE. UU con Rusia y China.

Si se examina el desarrollo a lo largo del tiempo de las centrales nucleares veremos que aparecen diferentes generaciones de reactores nucleares cada una de ellas con mejoras e incorporación de nuevas tecnologías.

La I generación desarrollada en la década del 50 nace como una derivación del desarrollo de las primeras armas nucleares siendo básicamente prototipos concebidos para experimentación.

La II generación es una versión mejorada de la I con fines comerciales, a esta generación pertenecen prácticamente todas las centrales que conocemos, Donde la fig. 2. Nos muestra un ejemplo de las mismas.



Fig. 2 Central Nuclear de II generación

La III generación apareció hace aproximadamente 20 años e incorporo mejoras de rendimiento y seguridad con respecto a la

anterior.

La IV generación a la que pertenecen los MR aparece a partir del año 2000 como una iniciativa del Foro Internacional Gen IV.

En el mismo se plantea un nuevo paradigma donde se busca diseñar reactores más pequeños, más baratos, más simples y versátiles [4].

Estos deben incluir conceptos como la capacidad de lograr “seguridad pasiva”, reducir o eliminar el personal necesario para su mantenimiento y operación y tener capacidades de 1 a 10 MW(e). En función de esto los principales objetivos en el diseño de un MR nuclear son [5]:

- Simplificar la producción, fabricación, transporte, instalación y operación.
- Ser pequeños, móviles y fácilmente desplegados.
- Permitir su utilización además de para producción de energía para funciones secundarias tales como el uso del calor generado para desalinización del agua.

A partir de allí se han desarrollado diversos proyectos en varios países tales como el reactor CAREM en la Argentina y el HTR-PM en China.

III. SEGURIDAD NUCLEAR

En un reactor nuclear existen diversos sistemas no orientados a la producción de energía eléctrica, sino enfocados únicamente a la seguridad de la instalación en caso de ocurrencia de transitorios y accidentes previstos en el diseño [6]. Siendo los principales objetivos de los sistemas de seguridad:

- Apagar el reactor.
- Mantener la situación de apagado.
- Prevenir la liberación de material radiactivo durante eventos o accidentes.

El control de la operación de un reactor se basa en monitorear la potencia generada y en regular su reactividad (coeficiente de aceleración de la reacción nuclear).

Para ello, el sistema de instrumentación y control del reactor determina el valor de todas las variables de la operación, como son el flujo neutrónico, la temperatura del refrigerante, caudal, presión, entre otras [7] [8] [9].

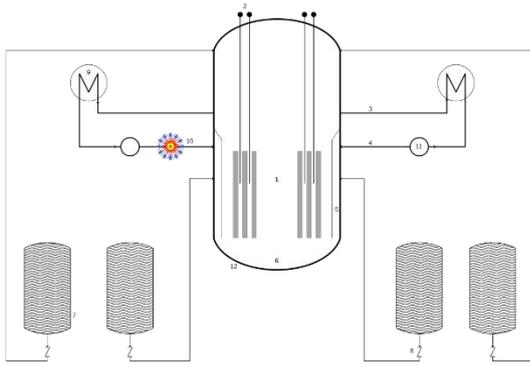


Fig. 3 Sistema de refrigeración pasivo de emergencia

El sistema de I&C limita y regula dichas variables mediante diferentes sistemas de protección como el que se observa en la fig. 3 [10].

Además, el sistema de I&C tiene preestablecidos valores de alarma de tal manera que, si se supera cualquiera de ellos, se genera de manera inmediata una serie de procedimientos de emergencia para evitar la pérdida del control del reactor [11].

IV. SISTEMA DE INSTRUMENTACIÓN Y CONTROL

Los sistemas de I&C de una central actual son extremadamente complejos, La fig. 4 muestra el diagrama en bloques típico de una central de II generación [12].

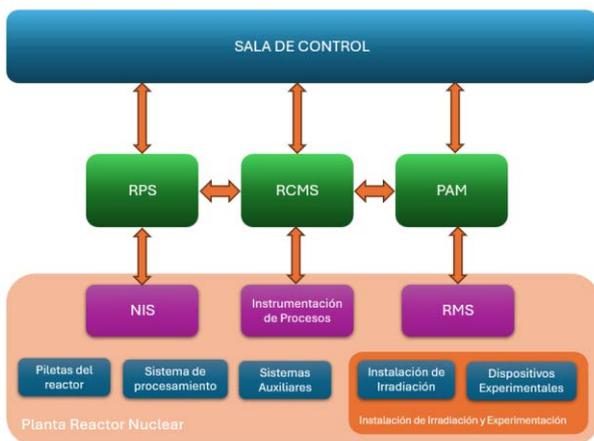


Fig. 4 Diagrama en bloques del sistema de I&C

La central dispone de diferentes sistemas encargados de realizar la obtención de los datos en campo para determinar el estado del reactor. Estos son el **NIS** Sistema de Instrumentación Nucleónica, el **RMS** Sistema de Instrumentación

de Procesos y en el Sistema de Monitoreo de Radiación.

La información recolectada es enviada al **RPS** Sistema de Protección del Reactor, que es el encargado de realizar las acciones de protección del núcleo.

Por otro lado, esta información también es enviada al **RCMS** Sistema de Control y Monitoreo del reactor que constituye es el elemento principal de la interfaz hombre máquina del reactor.



Fig. 5 Sala de control de una central nuclear

La información es proporcionada por el RCMS a los operadores en la sala de control como se muestra en la figura 5 [13]. Quienes tienen la responsabilidad de definir las operaciones a realizar en cada una de las situaciones que se presenten. Este punto es un cambio fundamental con respecto a los MR ya que estos deben ser capaces de funcionar en forma autónoma.

V. SISTEMA DE I&C PROPUESTO

Cuando se diseña un sistema de I&C se debe analizar la estructura del reactor categorizando cada uno de los sistemas según sus funciones de seguridad. Estas categorías establecidas por la OIEA (Organismo Internacional de Energía Atómica) clasifica los sistemas en importantes para la seguridad y Sistemas no importantes para la seguridad.

Posteriormente, estos se dividen en sistemas de seguridad y sistemas relacionados con la seguridad. Finalmente se comprueba la separación e independencia de estos. Ya que es un requisito obligatorio que los sistemas de seguridad se encuentren separados y sean independientes de los otros sistemas del reactor.

En nuestro caso utilizaremos el Simulador de Reactor Avanzado para el análisis del sistema, así como el entrenamiento de la red de

inteligencia artificial y la posterior validación del sistema.

Este simulador fue desarrollado en EE-UU y su uso está disponible a través de la OIEA, el mismo representa varios modelos de reactores.

Entre ellos podemos citar un PWR, un BWR y un HWR del orden de los 600 MWe. En el caso de los modelos de PWR, el simulador incluye centrales con generadores de vapor verticales de curva en U invertida de diseño occidental y centrales con generadores de vapor horizontales conforme a los diseños de la ex Unión Soviética.

También incorpora un PWR de VI generación con características de seguridad pasiva que se utilizara para las simulaciones en este proyecto.

Para cada uno de los bloques vistos en la fig. 4 se implementará un controlador tolerante a fallos basado en redes neuronales utilizando la placa ALINX AXU2CGA.

Esta placa de desarrollo FPGA que se muestra en la fig. 6 tiene un AMD Zynq UltraScale + MPSoC XCZU2CG, este integrado incluye un procesador ARM Cortex-A9 dual.

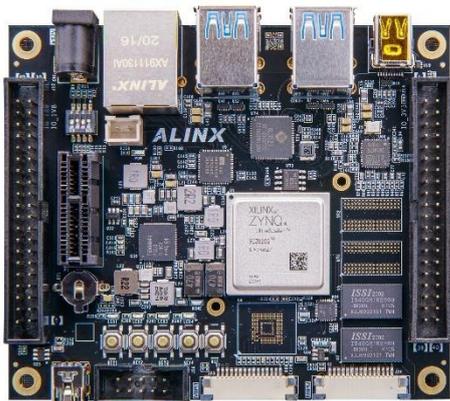


Fig. 6 Demoboard de IA ALINX AXU2CGA

Este integrado soporta el AMD Vitis – DPU que es un motor programable optimizado para redes neuronales profundas.

El mismo está constituido por un grupo de núcleos de IP prediseñados en hardware. Para la implementación de la DPU se utiliza un conjunto de instrucciones especializadas de Vitis AI, lo que le permite una gran eficiencia en la mayoría de las redes neuronales [14].

Esta placa tiene capacidad de comunicación en red, intercambio y almacenamiento de datos de alta velocidad, computación en la nube, control de motores industriales, monitoreo de sensores y aplicaciones de IoT industrial lo que la hace adecuada para esta aplicación.

La figura 7 muestra la arquitectura interna que se implementara en el controlador donde se observa que hay tres líneas de instrumentación y procesamiento aisladas y paralelas.

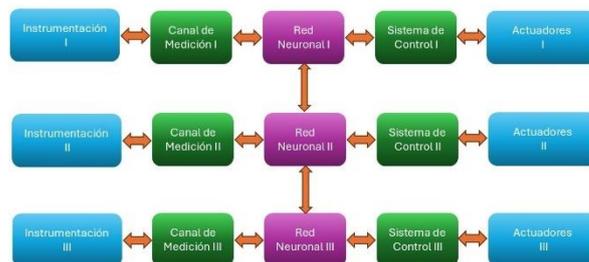


Fig. 7 Diagrama en bloques del controlador

Dentro de cada una de estas líneas de I&C hay una red neuronal que procesara independientemente la información recabada además de estar encargadas de extensos diagnósticos de las distintos componentes del sistema.

El controlador incluirá además mecanismos de votación tanto en hardware como en software especializados que calificarán y verifican todas las entradas y salidas digitales de campo, mientras que las entradas analógicas estarán sujetas a un proceso de selección de valor medio. Lo que proporcionara un funcionamiento ininterrumpido, sin errores y de alta integridad.

Para el diseño y entrenamiento de la red neuronal se utilizará el espacio de variables $S(u_1, u_2, u_3, \dots, u_{n-1}, u_n)$ en Vitis AI cuyos valores se obtendrán de las simulaciones realizadas en el Simulador de Reactor Avanzado.

Una vez que la red reciba la información del módulo de instrumentación, se selecciona una acción a realizar para pasar del estado siguiente. La función de fitness que puede verse en la ecuación 1.

$$f(t) = \frac{z(t)}{n} \quad (1)$$

La función de fitness se ha definido de forma tal que no sea lineal, ponderando las mejores soluciones y penalizando las mediocres como puede verse en la ecuación 2.

$$f(t) = \begin{cases} -10 & \text{si } z(t)/n = 1/n \\ (10 * z(t)/n) - 10 & \text{si } (z(t)/n) < 0.5 \\ 10 * z(t)/n & \text{si } (z(t)/n) \geq 0.5 \end{cases} \quad (2)$$

La política de entrenamiento utiliza una red neuronal de clasificación [15] para mapear la información de las entradas y salidas que se muestra en la Fig. 8. Esta será la que se implementará en la FPGA.

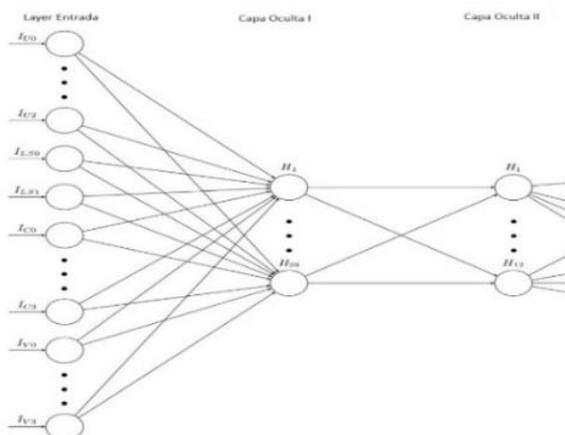


Figura 8. Red neuronal del controlador.

Esta red consta de 2 capas ocultas más las capas de entrada y salida, la función de activación de cada una de sus neuronas es tanh y obtiene la información del entorno discretizada para acelerar el proceso de aprendizaje.

La fig. 9 muestra el flujo de desarrollo de una aplicación en Vitis AI utilizando redes neuronales.

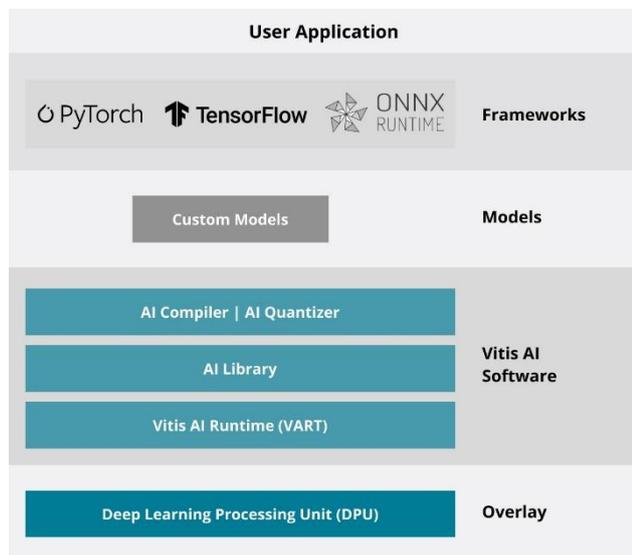


Figura 9. Flujo de desarrollo de una aplicación en Vitis AI.

Como se observa el estudio, implementación y entrenamiento de la red neuronal se puede realizar con varias herramientas disponibles en el mercado de las cuales utilizaremos

TensorFlow por adecuarse más a nuestra aplicación.

Una vez terminada esta etapa el Vitis AI realiza una compilación de la información sobre la red con sus librerías de modelos de I.A para obtener una red optimizada que cumpla con los parámetros definidos de manera tal que utilizando la DPU se implemente en la FPGA utilizando la menor cantidad de recursos posibles.

VI. CONCLUSIONES

Se ha realizado un somera análisis de la historia del arte de la generación nuclear poniendo el foco en los microreactores nucleares, Se han estudiado las características y dificultades de la implementación de un sistema de I&C para los mismos.

Planteándose como solución el desarrollo de controladores tolerantes a fallos utilizando como base la implementación en hardware de redes neuronales en una FPGA.

Para ello se utilizará la placa ALINX AXU2CGA donde para el análisis y entrenamiento de la red neuronal se utilizará el software TensorFlow.

Ingresando en el mismo los resultados de las simulaciones obtenidas del simulador de reactor avanzado, para en una etapa posterior ser compilado en el Vitis AI generando las DPU correspondientes para obtener el diseño final del controlador.

REFERENCIAS

- [1] TESTONI, Raffaella; BERSANO, Andrea; SEGANTIN, Stefano. Review of nuclear microreactors: Status, potentialities and challenges. *Progress in Nuclear Energy*, vol. 138, p. 103822, 2021.
- [2] INGERSOLL, D.T., Deliberately small reactors and the second nuclear era, *Prog. Nucl. Energy* 51 (2009) 589–603.
- [3] INTERNATIONAL ATOMIC ENERGY AGENCY, Status of Fast Reactor Research and Technology Development, IAEA-TECDOC-1691, IAEA, Vienna (2012).
- [4] UPADHYAYA, B.R., ZHAO, K., WOOD, R.T., INGERSOLL, D.T., Thermal-hydraulic analysis of a helical coil steam generator for level monitoring, *Trans. Am. Nucl. Soc.* 88 (2002) 283–284.
- [5] CARELLI, M.D., INGERSOLL, D.T., Handbook of Small Modular Nuclear Reactors, Woodhead Publishing, Cambridge (2014).
- [6] CLAYTON, D.A., WOOD, R.T., "The role of instrumentation and control technology in enabling deployment of small modular reactors", *Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies* (Proc. 7th American Nuclear Society Int. Topical Mtg, Las Vegas, 2010), ANS, LaGrange Park, IL (2010) CD-ROM.
- [7] WESTINGHOUSE ELECTRIC COMPANY, Nuclear Automation: Vibration Integrity Monitoring System, NA-0020, Westinghouse Electric Company, Cranberry Township, PA (2012).



- [8] MA, J., TRABIA, M., JIANG, Y., MOUJAES, S., Solid-phase Oxygen Control System, Mechanical Engineering Faculty Publications, Univ. Nevada, Las Vegas, NV (2005).
- [9] JEONG, S.H., BAHN, C.B., CHANG, S.H., CHOI, S.Y., HWANG, I.S., "Oxygen sensor development for LBE System", Proceedings of the Spring Meeting of the Korean Nuclear Society, 2005, KNS, Seoul (2005).
- [10] BALL, S.J., HOLCOMB, D.E., CETINER, S.M., HTGR Measurements and Instrumentation Systems, ORNL-TM-2012/107, Oak Ridge National Laboratory, Oak Ridge, TN (2012).
- [11] TALLACKSON, J.R., MOORE, R.L., DITTO, S.J., Instrumentation and Controls Development for Molten-Salt Breeder Reactors, ORNL-TM-1856, Oak Ridge National Laboratory, Oak Ridge, TN (1967).
- [12] INTERNATIONAL ATOMIC ENERGY AGENCY, Design of Instrumentation and Control Systems for Nuclear Power Plants, IAEA Safety Standards Series No. SSG-39, IAEA, Vienna (2016).
- [13] HOLCOMB, D.E., et al., Instrumentation, Controls, and Human-Machine Interface Technology Development Roadmap for Grid-appropriate Reactors, ORNL-GNEP-LTR-2008-041, Oak Ridge National Laboratory, Oak Ridge, TN (2008).
- [14] BOURZAC, KATHERINE, Neural Networks on the Go. IEEE Spectrum (2016)
- [15] GRAUDE, D., Principles of Artificial Neural Networks. Advanced Series on Circuits and Systems, vol. 6., 2nd. Edition. World Scientific Publishing Company (2007)

Sistema de telemedición para una subestación transformadora

Leonado Daniel Del Sancio

Noanet S.A.

Universidad Nacional de Tucumán

Tucumán - Argentina

ldelsancio@herrera.unt.edu.ar

Resumen—El presente trabajo describe el desarrollo e implementación de un equipo electrónico cuya función es medir parámetros físicos de una subestación transformadora, y enviarlos mediante una conexión remota TCP/IP con tecnología 4G. El trabajo fue realizado para la empresa NOANET [1], quien es la encargada de gestionar dichos datos. Los parámetros más importantes a medir son: tensiones, corrientes, potencias, temperatura externa del transformador y posicionamiento GPS entre otros. Tener el control de las variables puede ayudar a prevenir accidentes por sobrecalentamiento, baja tensión y sobrecarga. Además brinda una información valiosa del comportamiento de cada variable a lo largo del tiempo.

Palabras clave: Set, subestación transformadora, Iot, servidor Web, telemedición.

I. INTRODUCCIÓN

Una subestación eléctrica o subestación transformadora (SET) es una instalación que ajusta los niveles de tensión para la transmisión y distribución de energía eléctrica, siendo el transformador su componente principal. Las SET son necesarias para minimizar las pérdidas de potencia por efecto Joule [2] y para transportar la energía eficientemente. Hay dos tipos de SET: elevadoras, que aumentan la tensión para el transporte, y reductoras, que disminuyen la tensión para la distribución y el consumo final. El trabajo se enfoca en las SET reductoras, que ajustan la tensión a niveles comerciales para uso doméstico e industrial. Los parámetros físicos de la SET de interés para este proyecto son los que se pueden apreciar en la tabla 1.

TABLA I
VARIABLES FÍSICAS A MEDIR

Unidad	Variable
V	Tensión en 3 fases
A	Corriente en 3 fases
°C	Temperatura interna y externa
W	Potencia Activa por fase

VAR	Potencia Reactiva por fase
VA	Potencia Aparente por fase

II. INTRODUCCIÓN ESPECÍFICA

A. Esquema general del sistema

En la figura 1 se ilustra el esquema en bloques del sistema propuesto.

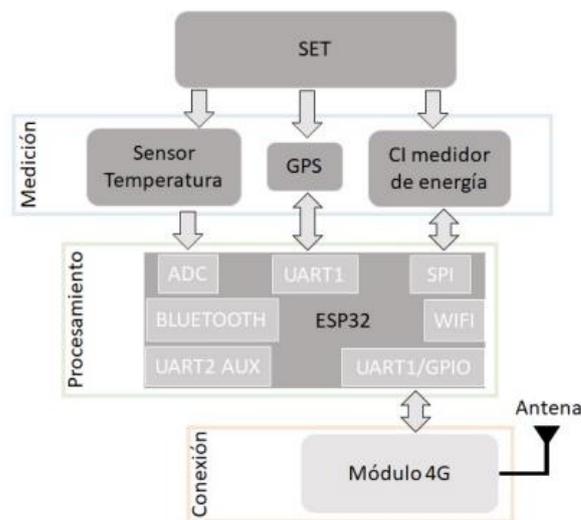


Fig. 1. Diagrama en bloques del sistema completo

Al sistema se lo puede separar en tres grupos: **Medición:** este bloque está compuesto por el sensor de temperatura LM35, el circuito integrado medidor de energía ATM90e36 y el módulo GPS que está embebido en el SIM7600SA. El sensor de temperatura LM35 es un sensor analógico que envía los datos al procesador ESP32 por un pin ADC0, se alimenta de 5 V y la tensión entregada es proporcional a la temperatura externa de la SET (chasis) en grados centígrados.

El circuito integrado medidor de energía ATM90E36 se encarga de medir la tensión y corriente en las tres fases de la SET, calcula la

potencia activa, reactiva y aparente, el coseno fi y la frecuencia de la red. Guarda todos los resultado en bloques de memoria interna para ser leídos por el microcontrolador mediante el protocolo SPI. El módulo GPS está embebido dentro del módulo SIM7600SA. Es el encargado de tomar los datos de posicionamiento de la SET y mediante el puerto serial UART1 se comunica con el bloque de procesamiento. **Procesamiento:** es el bloque principal, en donde se gestionan todos los datos recibidos y se configuran los dispositivos conectados. Envía los comandos de configuración, las consultas de posicionamiento GPS, lee los bloques del módulo medidor de energía mediante SPI, toma la lectura analógica del sensor de temperatura y guarda todos los datos en su memoria flash para luego armar la trama a ser enviada. **Conexión:** este bloque es el encargado de enviar los datos al servidor mediante la tecnología 4G [3]. Se conecta el módulo SIM7600SA a un servidor APN mediante una conexión TCP/IP [4]. Este dispositivo se configura por el bloque de procesamiento mediante el puerto serial UART1 y además, tiene destinados dos puertos GPIO para el encendido y reset del módulo.

III. DISEÑO E IMPLEMENTACIÓN

A. Diseño del Firmware

Para el desarrollo del firmware, se evaluaron los requerimientos y las tareas que debía realizar el procesador y se probaron diferentes bibliotecas y la API de la empresa Espressif-ESPIDF (Espressif IoT Development Framework) [5]. Dada la simplicidad del manejo de la API, la variedad de drivers y bibliotecas, sumado a la facilidad del manejo de interrupciones, se optó por programar el firmware en bare-metal. El diseño del firmware se organizó y se empleó una estructura de capas, esto permite la separación de las partes que componen el sistema. En la figura 2, se observa la estructura de capas:



Fig. 2. Diagrama de capas del software

- Capa de aplicación: es la capa más alta, en donde se implementó la lógica del equipo. Está compuesta por módulos como máquinas de estado, interrupciones, empaquetamiento de datos, etc.

- Capa intermedia: está compuesta por un conjunto de bibliotecas, las funciones que se encargan de la comunicación SPI, I2C, entre otras.
- Capa de bajo nivel: esta es la capa más baja, en donde ocurre la comunicación con el hardware mediante la API ESPIDF. En esta API, se incluyen los controladores para el manejo de los puertos GPIO, PWM, SPI y Wi-Fi entre otros.

B. Diagrama en bloques del sistema

En la figura 3, se observa el diagrama de flujo de control del sistema propuesto.

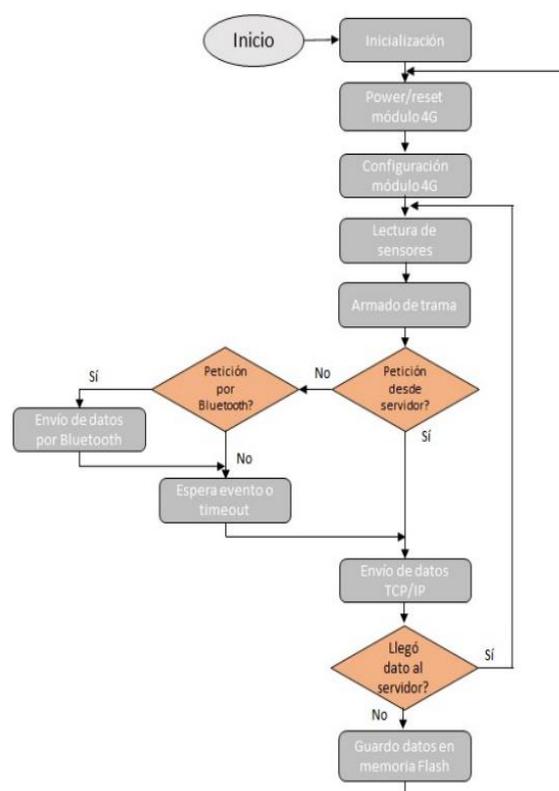


Fig. 2. Diagrama de flujo de control del sistema

El diagrama de flujo de control del sistema, mostrado en la figura anterior, ilustra los pasos principales que sigue el sistema para realizar sus funciones. Estos incluyen la inicialización del hardware, la configuración del módulo 4G, la lectura de los sensores de corriente, tensión, temperatura y GPS, la construcción de la trama de datos a transmitir, la espera de eventos o el cumplimiento de un tiempo límite, y finalmente, el envío de datos a través de Bluetooth o TCP/IP al servidor. Si la transmisión falla, los datos se guardan en la memoria flash para ser enviados posteriormente.

C. Empaquetado de datos

La codificación inicial de prueba adoptada para armar la trama a transmitir, es ilustrada en la figura 4.

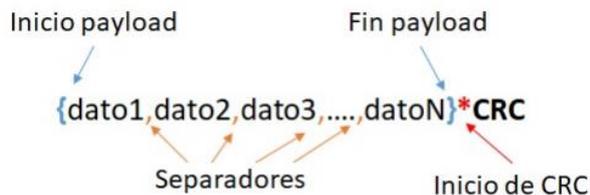


Fig. 4. Codificación de la trama a transmitir

El paquete está compuesto siempre por la misma cantidad de datos y se agregó al final de la trama un CRC [6], que es una verificación de redundancia cíclica. Esta herramienta permite identificar si hubo algún bit corrompido en la transmisión y de ser así, el servidor vuelve a realizar una petición de datos y descarta la trama anterior

D. Diseño del circuito impreso

A la hora de utilizar un criterio para la selección de los componentes del hardware, se tuvieron en cuenta aspectos importantes relacionados con la manufactura, disponibilidad y el uso de plataformas de desarrollo abiertas. Se diseñó un hardware compacto con componentes de proveedores nacionales.

En la figura 4 se ilustra el lado Top del circuito impreso en donde se observan los componentes principales.



Fig. 4. Fotografía lado TOP del PCB.

IV. CONCLUSIONES Y TRABAJO A FUTURO

El proyecto culminó exitosamente con el desarrollo de un sistema de telemedición para estaciones transformadoras, que incluye cinco prototipos electrónicos capaces de medir y transmitir parámetros físicos a un servidor mediante tecnología 4G. Los prototipos también permiten configuraciones in-situ a través de una aplicación móvil. A pesar de los desafíos, como la pandemia de COVID-19 que retrasó la certificación del sistema, y la necesidad de ajustar algunos requisitos de hardware y firmware,

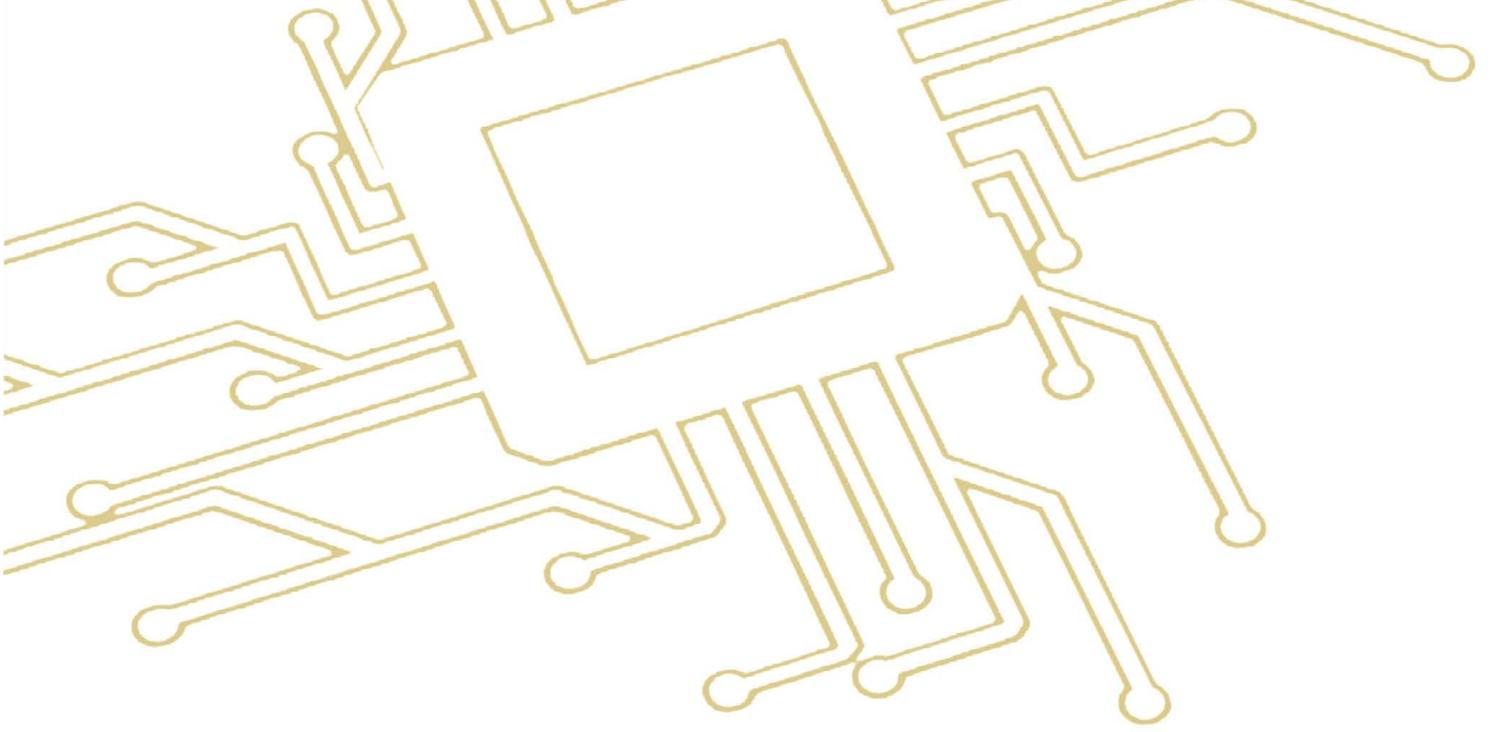
se logró un equipo confiable y compacto. Además, se aplicaron conocimientos adquiridos en la carrera de Especialización en Sistemas Embebidos, incluyendo gestión de proyectos, ingeniería de software, protocolos de comunicación, y diseño de PCB.

A pesar de cumplir con los objetivos iniciales, se identificaron áreas de mejora, como:

- Optimización de la aplicación web para la verificación en tiempo real.
- Rediseño del circuito impreso para mejorar la seguridad.
- Implementación de actualizaciones de firmware remotas.
- Inclusión de un mecanismo de corte de suministro eléctrico controlado remotamente.
- Desarrollo de un módulo externo con tecnología LoRa para ampliar la cobertura en áreas sin señal celular.

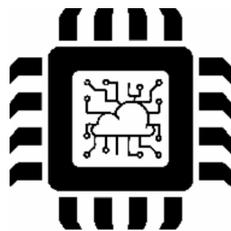
REFERENCIAS

- [1] Noanet Soluciones T.I. <http://www.noanet.com.ar>. (Visitado 07-07-2022).
- [2] trámites y servicios del Gobierno Vasco Información. Energía Eléctrica Y Efecto Joule. <https://www.hiru.eus/es/fisica/energia-electrica-y-efecto-joule>. Dic. de 2019. (Visitado 10-11-2020).
- [3] «LRTC to Launch Lithuania's First Mobile WiMAX 4G Internet Service». WiMAX Forum. 3 de marzo de 2009. Archivado desde el original el 12 de junio de 2010. Consultado el 26 de noviembre de 2010.
- [4] Protocolos TCP/IP ibm documentación AIX.7.2 <https://www.ibm.com/docs/es/aix/7.2?topic=protocol-tcpip-protocols>. Consultado el 2/5/2022.
- [5] Espressif. «El estándar de cifrado avanzado». En: (2010). URL: "<https://docs.espressif.com/projects/espressif/en/latest/esp32/api-reference/index.html>" (visitado 12-07-2022).
- [6] CRC (verificación de redundancia cíclica) link: <https://www.capterra.es/glossary/577/crc-cyclic-redundancy-check> visitado el 6/5/2022.



Eje Temático

Aplicaciones de Internet de las Cosas (IoT)



HEPPIE: un algoritmo simple para la estimación de tono, en un amplio espectro, con exactitud y costo computacional ajustables

Gerardo A. Laguna-Sánchez, Diana V. Ortiz-Martínez y Víctor M. Pichardo-Infante

Universidad Autónoma Metropolitana

CDMX, México

g.laguna@correo.ler.uam.mx

Resumen—El algoritmo denominado HEPPIE (por *HElpful and Practical Pitch Estimation*) permite la estimación de la frecuencia fundamental de señales sonoras, con un mínimo de parámetros y operando en el espectro audible que abarca de los 50 Hz a los 5 kHz, con una resolución ajustable que determina el costo computacional. Si algoritmo se configura para operar con una resolución de 1 Hz, reduce significativamente su costo computacional y se convierte en una alternativa tan competitiva como el popular algoritmo YIN, pero con una mayor robustez ante el ruido AWGN. El algoritmo ha sido diseñado para su implementación en sistemas con recursos limitados, como lo son los sistemas embebidos, el Internet de las Cosas. El algoritmo opera a partir de principios simples bien conocidos y aspira a ser una alternativa de fácil implementación y operación: en esencia, se calcula el periodograma y se procesa mediante algunos filtros FIR simples. Se comparó el desempeño del algoritmo, en cuanto a exactitud y costo computacional, contra diversos algoritmos bien conocidos de estimación de tono, incluyendo algunos muy recientes en el estado del arte, y la posición obtenida resultó ser muy competitiva.

Palabras clave—Frecuencia fundamental, Estimación de tono, Detección de tono, Notas musicales

I. Introducción

Las notas sonoras producidas por instrumentos musicales se conforman por la superposición de un cierto número de frecuencias armónicas correspondientes a una frecuencia fundamental característica. Las frecuencias armónicas presentan valores que son múltiplos enteros de su frecuencia fundamental. Dado que la intensidad de cada una de las frecuencias armónicas de una señal depende de las condiciones y características físicas del instrumento que la produce, el sonido resultante se distingue por un timbre específico. En este contexto, la estimación de la frecuencia fundamental de una señal audible resulta de gran interés, ya sea para la afinación de instrumentos musicales o para el procesamiento y análisis del audio.

El algoritmo HEPPIE (por *HElpful and Practical Pitch Estimation*) fue presentado recientemente [1] como una alternativa de fácil comprensión, implementación y uso, con un costo computacional que depende directamente

de la resolución requerida. A mayor resolución, mayores recursos computacionales son requeridos y mayor el tiempo consumido; a menor resolución, menores son los recursos computacionales requeridos y menor el tiempo consumido. El algoritmo fue implementado en Matlab y en este trabajo se presentan los resultados de la comparación de desempeño contra 8 algoritmos no supervisados de referencia que, o bien, se encuentran disponibles dentro del módulo de procesamiento de audio de Matlab [2], o sus códigos Matlab se encuentran disponibles en repositorios de código abierto.

Nuestra exposición considera las siguientes secciones: La sección II, refiere muy brevemente el trabajo relacionado y enlista los algoritmos empleados en la comparación de desempeño. La sección III, explica el algoritmo HEPPIE. La sección IV, presenta la metodología para la comparación de desempeño y discute los resultados obtenidos. Finalmente, en la sección V, compartimos nuestras conclusiones.

II. Trabajo relacionado y algoritmos empleados en la comparación de desempeño

En general, dentro de la literatura especializada, la estimación de la frecuencia fundamental se asocia con el proceso conocido como estimación/detección de tono (*pitch*). Una primera clasificación de los algoritmos de estimación/detección de tono es la que parte del dominio del procesamiento de la señal, a saber, 1) el dominio del tiempo, 2) el dominio de la frecuencia o 3) una combinación de ambos dominios, como en [3]. Otro enfoque, más amplio, es el que parte de la naturaleza supervisada o no supervisada del algoritmo, es decir, si el algoritmo requiere o no de un entrenamiento previo con datos de ejemplo, como en [4]. Finalmente, algunos autores, por ejemplo [5], también hacen énfasis en el hecho de que existen propuestas paramétricas, donde el objetivo es ajustar los parámetros del modelo hasta que éste reproduzca al comportamiento observado con un error mínimo. Las propuestas paramétricas llegan a ser significativamente más exactas, pero también son computacionalmente muy costosas. En este contexto,

podemos decir que algoritmo HEPPIE es una técnica no paramétrica, no supervisada y que opera en el dominio de la frecuencia.

Para comparar el desempeño del algoritmo HEPPIE, en principio, se contemplaron los 13 algoritmos de referencia que aparecen en la tabla I, incluyendo tres algoritmos rápidos bien conocidos: el algoritmo *Normalized Correlation Function* (NCF) [6], el algoritmo *Cepstrum* (CEP) [7] y el algoritmo YIN [8]. Esta lista también incluye un algoritmo paramétrico del estado del arte, el conocido como *Fast Nonlinear Least Squares* (FNLS) [4], [5] que teóricamente garantiza el mínimo error en entornos adicionados con ruido blanco. La lista también incluye propuestas de desempeño intermedio, incluyendo los algoritmos *Robust Algorithm for Pitch Tracking* (RAPT) [9] y YAAPT (*Yet Another Algorithm for Pitch Tracking*) [10], además de 7 algoritmos emparentados con el método *Harmonic Summation* (HS), con resultados que se aproximan al criterio *nonlinear least squares* pero con un menor costo computacional. Los algoritmos afines al tipo HS incluyen al *Log-Harmonic Summation* (LHS) [11], el algoritmo *Summation of Residual Harmonics* (SRH) [12], el algoritmo *Subharmonic-to-Harmonic Ratio* (SHR) [13], el algoritmo *Pitch Estimation Filter* (PEFAC/PEF) [14], el algoritmo *Sawtooth Waveform Inspired Pitch Estimator* (SWIPE) [15] y una variante del algoritmo SRH (SRHv) [16].

Sin embargo, debido a que no todos los algoritmos de la tabla I pudieron ajustarse para operar, con un error de estimación aceptable, en intervalo espectral de interés, que abarca de 50 Hz a los 5kHz, la lista final de algoritmos de referencia quedó constituida por el algoritmo FNLS, que provee del mínimo error teórico aunque con un costo computacional significativo; dos algoritmos que pueden considerarse, estrictamente, del tipo HS, los algoritmos LHS y SRH; dos algoritmos que, en sentido amplio, pueden considerarse como del tipo HS [16], los algoritmos PEF y SWIPE; y tres algoritmos soportados por autocorrelaciones en el dominio del tiempo, con tiempos de ejecución mínimos, los algoritmos NCF, RAPT y el muy popular algoritmo YIN. Ahora, debemos aclarar que el hecho de que un algoritmo se haya excluido de nuestra lista final no necesariamente implica que éste no se pueda desarrollar, con base en sus principios de operación, para funcionar bien en el intervalo de las frecuencias de interés, simplemente significa que los autores no encontraron la forma de configurarlo, con una misma configuración de sus parámetros, para que operara con un error aceptable en el referido intervalo del espectro.

Debido a que una de las métricas de desempeño es la exactitud en la estimación del valor de la frecuencia fundamental, se empleó una base de datos con señales sintéticas construidas con componentes espectrales de valores específicos, a partir de los patrones espectrales de los tonos de algunos instrumentos musicales reales

disponibles dentro de la base de datos TinySOL [24]. La selección de los instrumentos y tonos de referencia se realizó a partir del requerimiento de contar con las frecuencias de los tonos, sin incluir semitonos, de notas musicales que abarcaran el mayor intervalo posible dentro de la escala temperada. En particular, el conjunto de señales sintéticas comienza en la frecuencia de la nota A1 (55Hz) y termina en la frecuencia de la nota C8 (4186Hz), resultando en un total de 45 patrones espectrales representativos, incluyendo las notas producidas por el acordeón, la tuba bajo, el clarinete, el violín o la viola.

III. Operación del algoritmo HEPPIE

La idea básica del algoritmo HEPPIE consiste en obtener el periodograma de la señal sonora, para proceder con el procesamiento del perfil espectral en potencia y determinar los picos espectrales más significativos, correspondientes a las frecuencias de los armónicos, y concluir con la determinación de la distancia mínima entre estos picos espectrales que, en principio, correspondería con la frecuencia fundamental.

El sustento teórico para partir del periodograma es bien conocido, ya que corresponde al cálculo de la autocorrelación en el dominio de la frecuencia, lo que implica la minimización de los efectos del ruido gaussiano blanco y aditivo (AWGN, por sus siglas en inglés) [25]. En resumen, sea la autocorrelación de una señal muestreada, representada por la secuencia $x(n)$,

$$r(m) = E[x^*(n)x(n+m)] \quad (1)$$

con $E[\cdot]$ el operador de valor esperado y x^* el conjugado de x . Para secuencias finitas de tamaño N , un estimador sesgado de $r(m)$ sería:

$$\tilde{r}(m) = \frac{1}{N} \sum_{n=0}^{N-1-m} x^*(n)x(n+m) \quad (2)$$

para $0 \leq m < N$. El periodograma resulta de aplicar la transformada de Fourier discreta (DFT, por sus siglas en inglés) a 2, lo que resulta en

$$S(k) = \frac{1}{N} X^*(k)X(k) = \frac{1}{N} |X(k)|^2 \quad (3)$$

donde $X(k)$ es la DFT de $x(n)$ y se puede calcular, con un costo computacional mínimo, mediante el empleo del algoritmo de la transformada rápida de Fourier (FFT, por sus siglas en inglés).

La codificación del algoritmo propuesto resultó en la función `heppie()`, que devuelve la estimación de la frecuencia fundamental, f_0 , a partir de cinco parámetros: la secuencia de entrada, $s(n)$; la frecuencia de muestreo F_s , en Hz; el tamaño de la ventana de procesamiento $wsize$, en segundos; el paso para cada estimación hop , en segundos; y la mínima resolución requerida $mres$, en Hz. En el diagrama a bloques de la figura 1, se puede

TABLA I
Algoritmos de referencia

Acrónimo	Algoritmo	Dominio	Repositorio
CEP	Cepstrum	Frecuencia	[2]
FNLS	Fast Nonlinear Least Squares (modelo paramétrico)	Frecuencia	[17]
LHS	Log-Harmonic Summation	Frecuencia	[2]
NCF	Normalized Correlation Function	Tiempo	[2]
PEF	Pitch Estimation Filter without Amplitude Compression	Frecuencia	[2]
PEFAC	Pitch Estimation Filter with Amplitude Compression	Frecuencia	[18]
RAPT	Robust Algorithm for Pitch Tracking	Tiempo	[18]
SHR	Subharmonic-to-Harmonic Ratio	Frecuencia	[19]
SRH	Summation of Residual Harmonics	Frecuencia	[2]
SRHv	SRH Variant	Frecuencia	[20]
SWIPE	Sawtooth Waveform Inspired Pitch Estimator	Frecuencia	[21]
YAAPT	Yet Another Algorithm for Pitch Tracking	Híbrido	[22]
YIN	YIN	Tiempo	[23]

apreciar que lo primero que se hace es calcular el periodograma. Aunque el periodograma se calcula a partir de la información dentro de la ventana de procesamiento en turno, el número de muestras de la secuencia sobre la que realmente se calcula el periodograma se determina automáticamente, al interior de la función *heppie()*, y depende de la frecuencia de muestreo y de la mínima resolución requerida. De hecho, de ser necesario, se recurre a la técnica de *zero padding* para alcanzar la resolución requerida por el usuario. Específicamente, el tamaño de la secuencia sobre la que se debe calcular el periodograma está dada por

$$N = \frac{F_s}{mres} \quad (4)$$

Después del cálculo del periodograma, a fin de eliminar las irregularidades de los picos significativos en el perfil espectral, en ciertas circunstancias conviene realizar un procesamiento mínimo. En estos casos se emplea un filtro FIR pasa bajos (LP) de 17 coeficientes, con una frecuencia de corte, correspondiente a 0.1 de la frecuencia de Nyquist, a fin de suavizar el perfil de los picos de las componentes armónicas. A continuación del cálculo del periodograma, inicia la etapa donde se determinan cuáles son picos significativos y cuáles no. Para ello, el flujo del procesamiento continua con un filtro FIR diferenciador, es decir un filtro con coeficientes $b = [1 - 1]$, para contar con una estimación de la derivada del perfil espectral y poder detectar los picos significativos mediante la ubicación de los puntos con pendiente cero y, más específicamente, de los tránsitos de pendiente positiva a pendiente negativa, pasando por cero.

Así, la salida del filtro diferenciador permite, por una parte, estimar la longitud de la ondoleta que caracteriza el cambio de pendiente, de valor positivo a valor negativo, para el caso del pico de mayor potencia en el perfil espectral. Esta ondoleta típicamente tiene la forma aproximada de un ciclo senoidal y se pueden detectar fácilmente mediante una correlación a través de

un *matched filter* cuya respuesta al impulso, h , es precisamente la ondoleta de derivación caracterizada pero rotada sobre su eje vertical. Para esto, se determina la longitud de la ondoleta de derivación y la salida del filtro diferenciador se alimenta al *matched filter*. Entonces, a la salida del *matched filter*, se obtiene una secuencia con picos de correlación que señalan la posición de las componentes espectrales correspondientes a las frecuencias armónicas.

A continuación del *matched filter*, aparece el bloque que detecta los picos significativos, es decir aquellos picos de correlación que superan el umbral correspondiente al 0.1 de la magnitud del máximo valor de correlación. Finalmente, en el último bloque funcional, se realiza la estimación de la frecuencia fundamental, para lo que se recurre al criterio de la mínima distancia entre los picos de correlación, incluyendo, por defecto, a un pico de correlación en la frecuencia cero. Para conocer los detalles específicos de la implementación y operación práctica del algoritmo, en la función *heppie()*, el lector puede acceder al código fuente en Matlab, que se encuentra disponible en [26].

IV. Metodología y métricas para la comparación de desempeño

Durante la comparación de desempeño, se procuró emplear una misma configuración para todas las pruebas y para todo el intervalo [50Hz-5kHz]. Aunque todos los algoritmos probados permiten configurar del intervalo de frecuencias, no todos soportan los valores límite del intervalo de interés. Así mismo, la mayoría de los algoritmos probados, excepto el FNLS, pudieron configurarse para trabajar con ventanas de procesamiento y pasos (*hops*) de 0.1 segundos. En el caso la función *heppie()*, aunque no se requiere especificar el intervalo de frecuencias, si es necesario especificar la resolución de la estimación. A fin de comparar el desempeño del algoritmo HEPPIE en el escenario que le demanda mayor costo computacional, se le configuró la resolución requerida en 0.1Hz.

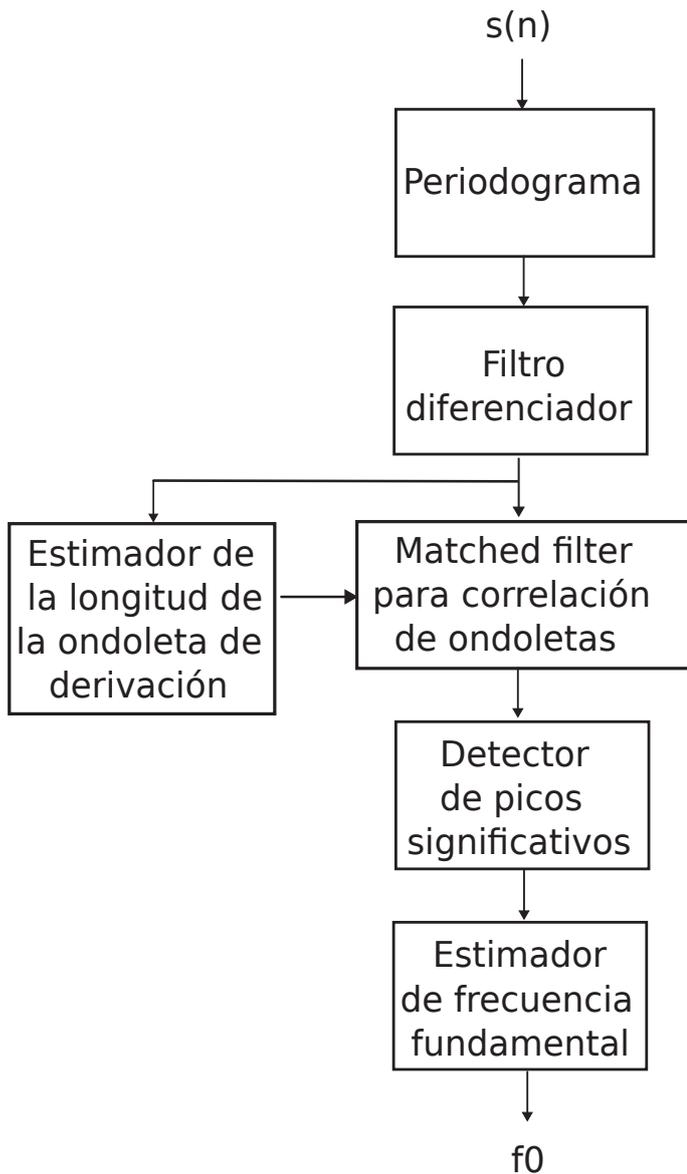


Fig. 1. Diagrama a bloques del algoritmo HEPPIE para estimación de tono.

Para comparar cuantitativamente el desempeño de todos los algoritmos, nos hemos concentrado en dos métricas: 1) el tiempo consumido, como métrica del costo computacional, y 2) la media del error absoluto, como métrica de la exactitud de las estimaciones. Específicamente, como métrica del tiempo consumido se empleó la siguiente expresión:

$$C = \frac{T}{N_{f0} \times wsize} \quad (5)$$

con T el tiempo de ejecución de la corrida, en segundos; N_{f0} el número de estimaciones realizadas y $wsize$ la duración de la ventana de procesamiento, en segundos.

El valor de C es adimensional y nos indica el tiempo consumido, por cada estimación, como una fracción de la duración de la ventana de procesamiento. El valor de C es una medida relativa del consumo de tiempo y, por lo tanto, puede expresarse como un porcentaje. El registro de la métrica C obtenidas por cada algoritmo probado, ejecutándose en una computadora soportada por un procesador Intel Core i5 @1.6GHz, se presenta en la tabla II.

TABLA II
Tiempos de ejecución relativos

Algoritmo	C [%]
FNLS	1868.7
LHS	0.68
NCF	0.17
PEF	0.70
RAPT	1.81
SRH	2.16
SWIPE	15.45
YIN	0.56
HEPPIE	9.56

Por otra parte, la media del error absoluto (MAE, por sus siglas en inglés), en Hz, no es más que el promedio del valor absoluto de las diferencias entre los valores estimados, f_{0i} , y el valor real de la frecuencia fundamental, f_t , ambas en Hz:

$$MAE = \frac{1}{N_{f0}} \sum_{i=1}^{N_{f0}} |f_{0i} - f_t| \quad (6)$$

con N_{f0} el número de las estimaciones procesadas.

Respecto de las señales de entrada para los algoritmos, se empleó una base de datos con señales sintéticas que, específicamente, incluye frecuencias fundamentales puras y notas con hasta 11 armónicos. A su vez, cada una de estas señales, puras o harmónicas, se proporcionan sin adición de ruido o con ruido AWGN, ya sea con SNR=10 o SNR=1, resultando en 6 colecciones de señales, cada una con 45 señales, tantas como frecuencias, todas con duración de 1 segundo y muestreadas a 16ksps. En la tabla III, se resume la información relevante de los 6 tipos de señales contenidas en la base de datos.

TABLA III
Colecciones con señales sintéticas

Arreglo	Descripción
sol_pure_16ksps	señales de frecuencias puras
sol_p10snr_16ksps	señales de frecuencias puras y AWGN con SNR=10
sol_p1snr_16ksps	señales de frecuencias puras y AWGN con SNR=1
sol_harm_16ksps	señales con hasta 11 armónicos
sol_h10snr_16ksps	señales con hasta 11 armónicos y AWGN con SNR=10
sol_h1snr_16ksps	señales con hasta 11 armónicos y AWGN con SNR=1
list_freq	lista de frecuencias

En particular, a fin de calcular la MAE, se emplearon las siguientes colecciones de señales:

- *sol_pure_16ksps*, denotada con la etiqueta *pure*.
- *sol_harm_16ksps*, denotada con la etiqueta *harm*.
- *sol_h10snr_16ksps*, denotada con la etiqueta *h10snr*.
- *sol_h1snr_16ksps*, denotada con la etiqueta *h1snr*.

Dada la configuración de los algoritmos, el cálculo de la MAE consideró un mínimo de 9×45 estimaciones de tono para cada uno de los tipos de señales empleadas. El registro de las MAE obtenidas, con su correspondiente desviación estándar (D.E.), y el valor MAE acumulado, se presenta en la tabla IV.

A partir los resultados de la tabla II, respecto de los tiempos de ejecución y, por ende, en cuanto al costo computacional, podemos ubicar el desempeño del algoritmo HEPPIE en una posición media, aunque cercana a los mejores tiempos de ejecución, sobre todo, tomando en cuenta que el algoritmo se operó con una resolución de 0.1 Hz, lo que le implica mayor costo computacional. Si las pruebas se repiten con una resolución de 10 Hz, se puede confirmar que el algoritmo HEPPIE logra alcanzar un tiempo de ejecución en el orden de magnitud del registrado por algoritmo YIN.

Respecto de la exactitud de los algoritmos, a partir de los resultados registrados en la tabla IV, es importante mencionar que los algoritmos LHS, PEF y SRH no pudieron configurarse para abarcar la totalidad del intervalo, ya que sus límites superiores quedaron acotados, debido a la frecuencia de muestreo empleada (16ksps), en 3198Hz, 3999Hz y 3198Hz, respectivamente, por lo que no es de sorprender el relativamente pobre desempeño obtenido. De estos algoritmos, PEF fue el menos afectado por la restricción en su límite superior (3999Hz), ya que este valor es muy cercano a la frecuencia fundamental sintetizada más alta, correspondiente a 4186Hz. Por lo demás, cabe resaltar el excelente desempeño del algoritmo FNLS, ya que teóricamente es el algoritmo con el menor error, incluso en presencia de ruido AWGN. Se puede observar que el registro MAE para el algoritmo FNLS ronda el valor de 0.4Hz, para las señales sin ruido o con poco ruido (SNR=10), y el valor de 5Hz para una señal con ruido significativo (SNR=1). Otro caso digno de mención es el desempeño del algoritmo YIN que, para las señales sin ruido, o con poco ruido (SNR=10), su registro MAE ronda el valor de 2Hz. Desafortunadamente, en presencia de ruido significativo (SNR=1) su desempeño cae notoriamente hasta alcanzar niveles de error del orden de 100Hz.

Respecto del algoritmo HEPPIE, queda clara su posición competitiva ya que muestra un registro de MAE con un orden de magnitud que ronda los 10Hz, para la mayoría de las señales, sean sin ruido o ruidosas. Esto muestra la robustez del algoritmo propuesto ante el ruido. Adicionalmente, es de resaltar que, en el caso de las señales con frecuencias fundamentales puras y sin ruido, el valor MAE para el algoritmo HEPPIE es del mismo orden de magnitud que el algoritmo mejor posicionado,

es decir el FNLS, que es un algoritmo del último estado del arte.

Para resumir en forma gráfica los resultados obtenidos, en la figura 2, se presenta una conveniente figura de mérito que permite apreciar el desempeño global para cada uno de los algoritmos probados, al considerar las dos métricas empleadas. Son notorias las posiciones de los algoritmos FNLS y NCF. El algoritmo FNLS aparece con el menor error acumulado, aunque con el mayor tiempo consumido. Por el contrario, el algoritmo NCF se posiciona con el menor tiempo consumido, aunque con el mayor error acumulado. Los demás algoritmos también quedan claramente posicionados, según su desempeño en cada una de las dos métricas, pero es evidente que la bondad de un algoritmo mejora cuando su posición relativa se aproxima al cruce de los ejes en la figura 2. En este sentido, es clara la ventaja relativa de dos algoritmos: el algoritmo YIN y el algoritmo HEPPIE, con la única diferencia de que el segundo está más cercana a la diagonal imaginaria que pasa por el origen y divide a la figura de mérito por la mitad. Esta mayor proximidad relativa del algoritmo HEPPIE a la referida diagonal, en comparación con el algoritmo YIN, significa que, aunque ambos algoritmos tienen posiciones globales similares, el algoritmo HEPPIE guarda un mejor equilibrio entre exactitud y costo computacional.

Para facilitar la reproducción los resultados reportados en este trabajo, en el repositorio [26] se deja, a disposición de los interesados, el material necesario, tanto el código Matlab de la función *heppie()* como la base de datos con la colección de señales sintéticas empleadas en las pruebas.

V. Conclusiones

Se ha presentado el algoritmo HEPPIE, como una alternativa competitiva para la estimación de la frecuencia fundamental de una señal audible, que opera con una atractiva relación exactitud/consumo de tiempo, en un amplio intervalo de frecuencias, de 50Hz a 5kHz, y que requiere únicamente cinco parámetros: la secuencia de entrada, la frecuencia de muestreo, el tamaño de la ventana de procesamiento, paso para cada estimación y la mínima resolución requerida. El algoritmo HEPPIE opera fundamentalmente en el dominio de la frecuencia, específicamente a partir del cálculo del periodograma, mediante una FFT, y su procesamiento mediante un filtro diferenciador seguido de un *matched filter*, a fin de detectar los picos espectrales significativos y la distancia entre estos.

El desempeño del algoritmo HEPPIE fue comparado contra reconocidos algoritmos de referencia y del último estado del arte, empleando, para ello, una colección de señales sintéticas, construidas a partir de los perfiles espectrales de instrumentos musicales reales. A partir de los registros de las métricas de desempeño, la media del error absoluto (MAE) y el consumo relativo de tiempo,



TABLA IV
Valores de MAE [Hz]

Algoritmo	h10snr MAE	h10snr D.E.	h1snr MAE	h1snr D.E.	harm MAE	harm D.E.	pure MAE	pure D.E.	MAE acumulada
FNLS	0.4	0.1	5.3	1.6	0.4	0.1	0.3	0.0	6.34
LHS	247.2	129.1	212.0	117.2	274.2	105.5	647.4	130.3	1380.89
NCF	877.1	133.4	883.1	654.3	961.7	0.0	1858.2	0.0	4580.08
PEF	135.5	105.3	275.8	201.6	20.1	0.2	9.3	0.2	440.67
RAPT	384.7	105.0	868.8	321.1	322.1	0.1	256.5	5.6	1832.13
SRH	335.6	170.8	294.5	186.8	345.1	22.1	1050.1	130.7	2025.35
SWIPE	50.4	82.0	409.8	329.8	19.2	5.7	9.2	4.9	488.58
YIN	2.7	0.3	152.5	58.7	2.7	0.0	2.1	0.0	159.93
HEPPIE	11.2	8.4	17.1	24.5	9.3	8.8	0.1	0.0	37.70

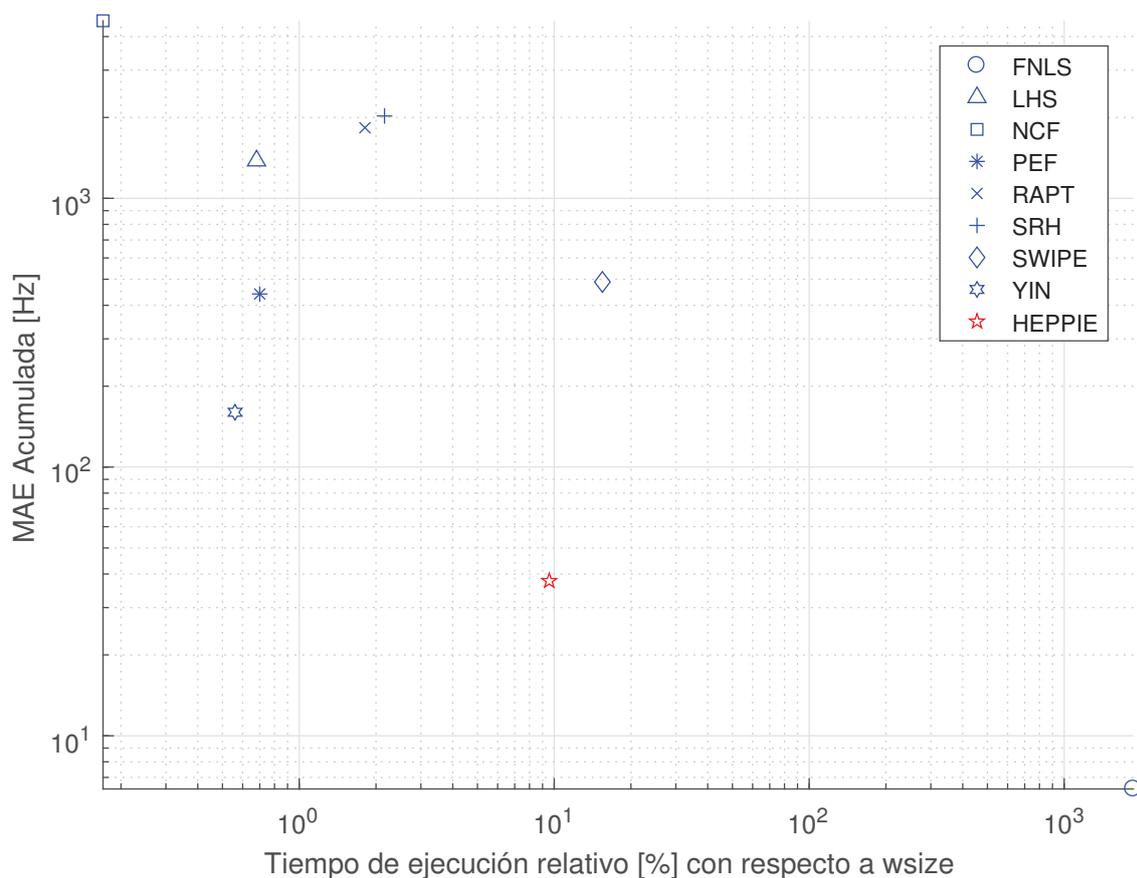


Fig. 2. Figura de mérito para el desempeño de los algoritmos probados.

se pudo comprobar que el desempeño del algoritmo HEPPIE es competitivo y que su posición global, dentro de la figura de mérito expuesta, se encuentra muy cerca del algoritmo YIN, aunque con mucho mejor tolerancia al ruido AWGN significativo.

En las pruebas realizadas, el algoritmo HEPPIE fue configurado intencionalmente para operar demandando el máximo de recursos computacionales y, a pesar de ello, su posición global en la figura de mérito resultó ser muy cercana a los algoritmos con la mejor combinación de exactitud/costo computacional. En el caso de que el algoritmo HEPPIE se empleara en aplicaciones que no requieren de una gran resolución, digamos de 10Hz, como podría ser el caso de la detección de tono en la voz humana, su consumo de tiempo se reduce significativamente hasta alcanzar un orden de magnitud similar al tiempo consumido por el algoritmo YIN, aunque con mucho mayor robustez ante el ruido.

Por todo lo anterior, podemos concluir que el algoritmo HEPPIE es una atractiva alternativa, para la estimación de la frecuencia fundamental, que permite al usuario determinar la relación exactitud/costo computacional que más le convenga.

Referencias

- [1] G. Laguna-Sanchez *et al.*, "Simple and practical algorithm for wide spectrum pitch estimation with an acceptable compromise between accuracy and computational cost," in *IV Conferencia Internacional sobre Comunicación y Tecnologías Aplicadas 2024 (ICOMTA'24)*, 2024.
- [2] MathWorks, "pitch: Estimate fundamental frequency of audio signal." The MathWorks Repository, 2024. <https://la.mathworks.com/help/audio/ref/pitch.html>.
- [3] L. Rabiner *et al.*, "A comparative performance study of several pitch detection algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 5, pp. 399–418, 1976. doi: 10.1109/TASSP.1976.1162846.
- [4] L. Shi *et al.*, "Robust bayesian pitch tracking based on the harmonic model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1737–1751, 2019. doi: 10.1109/TASLP.2019.2930917.
- [5] J. Nielsen *et al.*, "Fast fundamental frequency estimation: Making a statistically efficient estimator computationally efficient," *Signal Processing*, vol. 135, pp. 188–197, 2017. <https://doi.org/10.1016/j.sigpro.2017.01.011>.
- [6] B. Atal, "Automatic speaker recognition based on pitch contours," *Journal of the Acoustical Society of America*, vol. 52, no. 6B, pp. 1687–1697, 1972. <https://doi.org/10.1121/1.1913303>.
- [7] M. Noll, "Cepstrum pitch determination," *Journal of the Acoustical Society of America*, vol. 31, no. 2, pp. 293–309, 1967. <https://doi.org/10.1121/1.1910339>.
- [8] A. Cheveigne and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002. DOI:10.1121/1.1458024.
- [9] D. Talkin, *Speech Coding & Synthesis*, ch. A Robust Algorithm for Pitch Tracking (RAPT). Elsevier, 1995. W.B. Kleijn and K.K. Paliwal eds.
- [10] S. Zahorian and H. Hu, "A spectral/temporal method for robust fundamental frequency tracking," *Journal of the Acoustical Society of America*, vol. 123, no. 6, pp. 4559–4571, 2008. <https://doi.org/10.1121/1.2916590>.
- [11] D. Hermes, "Measurement of pitch by subharmonic summation," *Journal of the Acoustical Society of America*, vol. 83, no. 1, pp. 257–264, 1988. <https://doi.org/10.1121/1.396427>.
- [12] T. Drugman and A. Alwan, "Joint robust voicing detection and pitch estimation based on residual harmonics," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 1973–1976, 2011. .
- [13] X. Sun, "Pitch determination and voice quality analysis using subharmonic-to-harmonic ratio," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 333–336, 2002. doi: 10.1109/ICASSP.2002.5743722.
- [14] S. Gonzalez and M. Brookes, "A pitch estimation filter robust to high levels of noise (PEFAC)," in *19th European Signal Processing Conference. Barcelona*, pp. 451–455, 2011. .
- [15] A. Camacho and J. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1638–1652, 2008. <https://doi.org/10.1121/1.2951592>.
- [16] D. Wang *et al.*, "A robust and low computational cost pitch estimation method," *Sensors*, vol. 22, pp. 1–21, 2022. <https://doi.org/10.3390/s22166026>.
- [17] L. Shi, "Bayesian pitch tracking using harmonic model." GitHub Repository, 2019. <https://github.com/LimingShi/Bayesian-Pitch-Tracking-Using-Harmonic-model>.
- [18] M. Brookes, "VOICEBOX: Speech processing toolbox for MATLAB." Imperial College Repository, 2013. <http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html>.
- [19] X. Sun, "Pitch determination algorithm." MATLAB Central File Exchange, 2016. <https://la.mathworks.com/matlabcentral/fileexchange/1230-pitch-determination-algorithm>.
- [20] D. Wang, "Srh variant." GitHub Repository, 2022. https://github.com/deshengwang001/SRH_Variant.
- [21] K. Gorman, "Swipe' pitch estimator." GitHub Repository, 2012. <https://github.com/kylebgorman/swipe>.
- [22] A. Zahorian, "YAAPT pitch tracking MATLAB function." Repository of Binghamton University, 2016. <http://www.ws.binghamton.edu/zahorian/yaapt.htm>.
- [23] A. Cheveigne, "YIN, a fundamental frequency estimator for audio signals." Repository of Ecole normale superieure. <http://audition.ens.fr/adcl/>.
- [24] C. Cella *et al.*, "TinySQL: an audio dataset of isolated musical notes." Zenodo Repository, 2020. <https://zenodo.org/records/3685367>.
- [25] L. Jackson, *Digital Filters and Signal Processing: with MATLAB exercises*. Kluwer Academic Publishers, 1995.
- [26] G. Laguna-Sanchez, "HEPPIE, a simple and practical pith estimation algorithm." GitHub Repository, 2024. https://github.com/galaguna/HEPPIE_pitch_estimation_algorithm.



Herramientas de Machine Learning para IoT

1st Marcos Darío Aranda

Decanato de Ciencias Aplicadas
Universidad Siglo 21
Universidad Nacional de Catamarca
Catamarca, Argentina
marcos.aranda@ues21.edu.ar

2nd Eduardo Enrique Piray

Decanato de Ciencias Aplicadas
Universidad Siglo 21
Córdoba, Argentina
edupiray@gmail.com

3rd Paola Inés Beltramini

Grupo de Internet de las Cosas - GloT
Universidad Nacional de Catamarca
Catamarca, Argentina
pbeltramini@tecno.unca.edu.ar

4th Ricardo Elian González

Decanato de Ciencias Aplicadas
Universidad Siglo 21
Chaco, Argentina
eliangonzalez045@gmail.com

5th Gustavo Andrés Futo

Decanato de Ciencias Aplicadas
Universidad Siglo 21
Buenos Aires, Argentina
gustavoandresfu@gmail.com

5th Martin Gianotti

Decanato de Ciencias Aplicadas
Universidad Siglo 21
Buenos Aires, Argentina
lonasmartin@gmail.com

Resumen—El Machine Learning se encarga de generar algoritmos que tienen la capacidad de aprender y no tener que programarlos de manera explícita. En la actualidad el IoT permite la interconexión digital de objetos cotidianos con cualquier otro de su alrededor, éste se ha convertido en uno de los términos más populares en la industria tecnológica, utilizando para ello dispositivos de bajo consumo. Poder lograr que estos dispositivos logren incorporar machine learning, podría ser de gran utilidad para el futuro. Es por ello que el presente trabajo se centra en identificar herramientas que permitan generar modelos de machine learning para la clasificación y detección de objetos, y luego ser implementado en dispositivos de bajo consumo.

Index Terms—Machine learning, IoT, Microcontroladores, Clasificación, Detección.

I. Introducción

El Machine Learning, es una rama de la Inteligencia Artificial que se encarga de generar algoritmos que tienen la capacidad de aprender y no tener que programarlos de manera explícita. Existe un tipo de aprendizaje llamado supervisado, el cual consiste en entrenar un algoritmo de Machine Learning con datos de entrenamientos etiquetados. Además con el uso de Internet de las Cosas (IoT - Internet of Things), que se entiende como la interconexión digital de objetos cotidianos con cualquier otro de su alrededor, el mismo se ha convertido en los últimos años en uno de los términos más populares en la industria tecnológica gracias al enorme número de dispositivos que cuentan con la posibilidad de interconectarse [10]. Los dispositivos que cuentan con inteligencia artificial pueden ejecutar distintos procesos análogos al comportamiento humano, como la devolución de una respuesta por cada entrada (similar a los reflejos de los seres vivos), la búsqueda de un estado entre todos los

posibles según una acción o la resolución de problema mediante una lógica formal [1].

El objetivo de este trabajo es identificar cuales son las herramientas utilizadas para aprendizajes automáticos, que puedan ser utilizadas con un enfoque de código abierto para Machine Learning a gran escala para dispositivos de bajo consumo. [2]. Con el presente trabajo se busca contribuir a la clasificación y detección inteligente para dispositivos de bajo consumo en animales, para posteriormente el registro de información.

II. Marco Teórico

A continuación se realiza una breve descripción de las herramientas identificadas:

A. Aprendizaje Automático

En la detección de objetos se suelen emplear técnicas de Aprendizaje Automático o Machine Learning, esta área consiste en crear sistemas que pueden aprender por sí mismos, capaces de identificar una serie de patrones complejos a partir de datos de entrada. Este proceso se realiza mediante algún algoritmo de aprendizaje [3]. Existen cuatro tipos de aprendizaje automático:

- Aprendizaje supervisado: Se trata de un tipo de aprendizaje donde al sistema se le alimenta con datos de entrenamiento etiquetados. Una vez se ha completado el entrenamiento, se pueden introducir nuevos datos al sistema que no están etiquetados anteriormente, puesto que al estar entrenado el sistema es capaz de reconocer patrones para identificar los datos correspondientes. En este tipo de aprendizaje están los denominados problemas de clasificación y los de regresión [3].
- Aprendizaje no supervisado: En este tipo de aprendizaje se alimenta el sistema sin datos etiquetados.

Uno de los problemas más comunes de este tipo de aprendizaje automático es el de agrupamiento o clustering, en el que el sistema aprende a organizar los datos en distintos grupos, de modo que cada grupo contenga datos similares entre sí [3].

- Aprendizaje semisupervisado: Es una práctica que se encuentra en un punto medio entre el aprendizaje supervisado y el no supervisado. De esta manera, solo se utilizan un grupo mínimo de etiquetas. No obstante, la mayoría son grupos de datos no etiquetados, pues aumentan los costes, pero son útiles para cumplir con los objetivos. Aunque hay una supervisión sobre cómo actúa la máquina, no es un trabajo que se realiza a lo largo del trabajo con la máquina. Mientras que tendrás que etiquetar algunos resultados manualmente, otros serán propuestas automáticamente por el machine learning [11].
- Aprendizaje por refuerzo: Es una práctica que está basada en recompensar los comportamientos deseados, mientras que los no deseados serán penalizados. Es un proceso que está basado en la retroalimentación, pues la máquina aprenderá de las experiencias, a partir del desarrollo y rendimiento [11].

B. TinyML: Machine Learning

TinyML es una marca registrada de TinyML Foundation, esta asociación promueve el desarrollo de tecnología para aplicaciones de machine learning en dispositivos con capacidades de poco procesamiento [6]. Por tanto, hacer que los microcontroladores puedan ejecutar modelos de machine learning tiene el potencial para abrir nuevos escenarios disruptivos en todas las industrias. Otro aspecto relevante es el impacto ecológico que podría tener la adopción generalizada de las prácticas TinyML, que permitirían reducir el consumo de energía de cara al futuro [5].

C. Clasificación de imágenes

La tarea de identificar lo que representa una imagen se denomina clasificación de imágenes. Un modelo de clasificación de imágenes está entrenado para reconocer varias clases de imágenes. Por ejemplo, puede entrenar a un modelo para que reconozca fotografías que representen diferentes de animales. TensorFlow Lite proporciona modelos optimizados previamente entrenados que puede implementarse en aplicaciones móviles [7].

D. Detección de objetos

Un problema más complejo consiste en no solo decir si el objeto está o no presente en la imagen en cuestión, sino también en qué región de la imagen se encuentra dicho objeto [8].

III. Herramientas

A. TensorFlow Lite

Es un framework utilizado para implementación de modelos de aprendizaje automático que cuenten con poco poder de procesamiento entre ellos los dispositivos móviles, microcontroladores y dispositivos de IoT [4]. Las características más relevante de TensorFlow Lite son:

- Optimizado para el aprendizaje automático integrado en el dispositivo: No existe transferencia de datos con un servidor, logrando así disminuir la latencia. Los datos no salen del dispositivo. Con respecto a la conectividad no se necesita conexión a Internet, el tamaño del modelos y el binario son reducidos y además se obtienen mejoras en el rendimiento del consumo de energía.
- Compatible con plataformas iOS, Android, Linux embebido y microcontroladores. Soporta lenguajes Java, Swift, Objective-C, C++ y Python.
- Rendimiento para la aceleración de hardware y optimización de modelos.



Figura 1. Flujo de trabajo de TensorFlow Lite.

Con la incorporación del aprendizaje automático en microcontroladores, podemos potenciar la inteligencia de millones de dispositivos que usamos en nuestra vida, incluidos los electrodomésticos y dispositivos de la Internet de las Cosas, sin depender de hardware costoso ni de una conexión a Internet estable, que suele estar condicionada por restricciones de ancho de banda y energía, y que además, genera una latencia alta. Esto puede contribuir también a preservar la privacidad, ya que ningún dato sale del dispositivo [4]. TensorFlow Lite para microcontroladores se codificó en C++ 11 y requiere una plataforma de 32 bits. Se probó de manera exhaustiva con muchos procesadores basados en la arquitectura de la serie ARM Cortex-M y se trasladó a otras arquitecturas, como ESP32. El marco de trabajo está disponible como una biblioteca de Arduino. También puede generar proyectos para entornos de desarrollo, como Mbed. Es de código abierto y se puede incluir en cualquier proyecto de C++ 11 [9].

B. MediaPipe

Proporciona un conjunto de bibliotecas y herramientas para aplicar rápidamente técnicas de inteligencia artificial y machine learning. Permite conectar estas soluciones a las aplicaciones de manera inmediata, personalizarlas según necesidades y utilizarlas en varias plataformas de



Figura 3. Etiquetado de imágenes con labellmg.

desarrollo. MediaPipe es parte de un proyecto de código abierto [13], en la Fig. 2 se observa lo que incluye el paquete de soluciones [12].

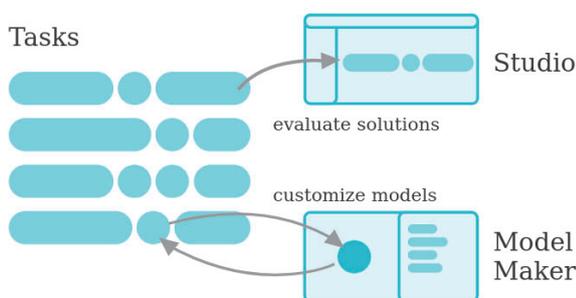


Figura 2. Paquete de soluciones de MediaPipe.

- 1) Bibliotecas y recursos que proporcionan la funcionalidad principal para cada solución:
 - MediaPipe Tasks: APIs y bibliotecas multiplataforma para implementar soluciones.
 - Modelos de MediaPipe: son modelos previamente entrenados y listos para ejecutarse para usar con cada solución.
- 2) Herramientas que permiten personalizar y evaluar las soluciones:
 - MediaPipe Model Maker: personaliza modelos para soluciones con datos.
 - MediaPipe Studio: visualiza, evalúa y compara soluciones en el navegador Web.

IV. Desarrollo

A. Búsqueda de datos

Para este trabajo se han utilizado para realizar pruebas de entrenamiento un dataset de Open Images Dataset V7, el cual es un servidor con aproximadamente 9 millones de imágenes con etiquetas, cuadros delimitadores de objetos, máscaras de segmentación de objetos, relaciones visuales y narraciones localizadas, se eligió la categoría Sheep. Posteriormente se procedió a la

descargar de categoría de las imágenes. Utilizando la herramienta OIDv4 ToolKit se procedió a la descarga de la misma.

B. Etiqueta en imágenes

Las etiquetas se generan en un formato XML, por cada una de ellas se encuentra asociada cada imagen, se utilizó la herramienta labellmg para etiquetar cuadros delimitadores de objetos en imágenes.

C. Entrenamiento

Para realizar el entrenamiento se utilizó Google Colab, el cual es un servicio colaborativo, que permite escribir y ejecutar código de Python en el navegador, el cual cuenta con las siguientes características:

- No requiere configuración previa.
- Se puede acceder a una GPU sin costo alguno.
- Facilidad para compartir.

Como continuidad de este trabajo se espera validar el modelo y obtener resultados en dispositivos de bajo consumo utilizando MediaPipe.

V. Conclusiones

En este trabajo se describió dos herramientas de aprendizaje automático que pueden ser utilizadas para la implementación de modelos TinyML en dispositivos de bajo consumo. Se identificó que ambas herramientas son de código abierto y brindan un gran soporte con respecto a la documentación para la implementación de modelo.

Como trabajo a futuro se espera validar el modelo de clasificación y detección de objetos, en dispositivos con recursos limitados, como móviles, microcontroladores o dispositivos IoT.

Agradecimientos

Agradecemos a las autoridades de la Universidad Siglo 21 por el apoyo incondicional para el desarrollo de la investigación.

Referencias

- [1] Julián Pérez Porto and Ana Gardey: Definición de inteligencia artificial. Qué es, Significado y Concepto : <https://definicion.de/inteligencia-artificial/>: 28 11 2022
- [2] Michael Bowles: Machine Learning in Python® : Essential Techniques for Predictive Analysis. Published by John Wiley and Sons, Inc: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119150202>: 15 02 2023
- [3] Abu-Mostafa, Yaser S: Learning from data: a short course (2012)
- [4] TensorFlow Lite : <https://www.tensorflow.org/lite/guide?hl=es-419> : 02 06 2023
- [5] TinyML : <https://www.sngular.com/es/tinyml-machine-learning/>: 20 03 2023
- [6] Pete Warden and Daniel Situnayake: TinyML : Published by O'Reilly Media, Inc. : 2020
- [7] Clasificación de imágenes : <https://www.tensorflow.org/lite/examples:> 20 03 2023
- [8] Rozada Raneros, Saúl and others: Estudio de la arquitectura YOLO para la detección de objetos mediante deep learning (2021)
- [9] TensorFlow microcontroller: <https://tensorflow.org/lite/microcontrollers> : 02 06 2023



- [10] Marcos D. Aranda, Paola I. Beltramini, Jesús E. Cano, Luis D. Villagrán, Juan P. Moreno, Sergio H. Gallina, Oscar Ariel Herrera Conegliano : IoT aplicado a la ganadería extensiva : RADI (PDTS) ISSN 2314-0925 : Volumen 17: 02 05 2021
- [11] Aprendizaje Automático : <https://www.keepcoding.io/blog/ tipos-de-aprendizaje-automatico/> : 27 04 2023
- [12] Google. MediaPipe Solutions Guide: <https://ai.google.dev/edge/mediapipe/solutions/guide?hl=es-419> : 20 08 2024
- [13] Google. MediaPipe GitHub Repository. <https://github.com/google-ai-edge/mediapipe> : 20 08 2024



Inclusión de tecnologías IoT y Ciencia de datos en la producción ganadera

Paola I. Beltramini
Facultad de Tecnología y Cs.Apl.
UNCA
Catamarca, Argentina
pbeltramini@tecno.unca.edu.ar

Marcos D. Aranda
Facultad de Tecnología y Cs.Apl.
UNCA
Catamarca, Argentina
maranda@tecno.unca.edu.ar

Jesús E. Cano
Facultad de Tecnología y Cs.Apl.
UNCA
Catamarca, Argentina
jecano@tecno.unca.edu.ar

Luis D. Villagrán
Facultad de Tecnología y Cs.Apl.
UNCA
Catamarca, Argentina
dvillagran@tecno.unca.edu.ar

Sergio H. Gallina
Facultad de Tecnología y Cs.Apl.
UNCA
Catamarca, Argentina
shgallina3157@gmail.com

Ivanna M. Lazarte
Facultad de Tecnología y Cs.Apl.
UNCA
Catamarca, Argentina
ilazarte@tecno.unca.edu.ar

Resumen— La ganadería extensiva, asistida por las nuevas Tecnologías de la Información y la Comunicación (TIC), se convierte en una ganadería más eficiente, rentable y sostenible, capaz de responder mejor a los desafíos actuales y futuros del sector agropecuario. A través de este trabajo se comparte la experiencia en el diseño de una solución tecnológica integral para el monitoreo y análisis de información de la actividad ganadera, utilizando tecnologías de IoT y Ciencia de Datos. Partiendo del diseño de un nodo IoT compuesto por un microcontrolador, sensores y un módulo de comunicación LoRa®, el cual se comunica a un gateway y éste al exterior via Ethernet, a través de protocolo LoRaWAN al servidor The Things Stack, quien envía la información por protocolo MQTT a Node-RED. Se utiliza Python para el procesamiento de datos y análisis estadísticos, lo que permitirá identificar patrones, tendencias y posibles problemas en la producción ganadera.

Palabras clave: IoT, Ciencia de Datos, Ganadería, LPWAN.

I. INTRODUCCIÓN

En la actualidad se evidencia un fuerte crecimiento de la automatización de las labores agrícolas y ganaderas. Esta modernización del campo, integrando tecnologías avanzadas como Internet de las Cosas (IoT) y la Ciencia de Datos, fomenta un cambio de paradigma con un impacto significativo en el trabajo rural para las nuevas generaciones [1][2]. La aplicación de estas tecnologías ofrece múltiples beneficios, entre los que destacan la eficiencia

en el uso de recursos, la reducción del impacto ambiental, el aumento de la producción y la mejora de la calidad de los productos.

Algunas de las herramientas tecnológicas que pueden ser utilizadas para mejorar ambos sectores son:

- Monitoreo y gestión remota.
- Sistemas de gestión de recursos naturales y energías renovables.
- Internet de las Cosas.
- Redes de comunicaciones inalámbricas de bajo consumo y largo alcance (LPWAN).
- Ciencia de Datos.
- Inteligencia artificial

La IoT en el sector agropecuario integra máquinas, animales y bienes de todo tipo, mediante la instalación de sensores que recopilan datos valiosos para el sector. Sin embargo, la verdadera potencia de la IoT radica en el análisis y aprovechamiento de esta gran cantidad de información, haciendo indispensable a la Ciencia de Datos.

La inclusión de estas herramientas constituye un reto complejo ya que requiere el dominio de diversas tecnologías, como el sensado de variables ambientales, físicas y biológicas, la gestión eficiente de las baterías para asegurar la autonomía de los dispositivos sensores, el uso de redes de LPWAN, el modelado dinámico, el registro y almacenamiento de datos, la seguridad de la información, así como la implementación de algoritmos y estadísticas bayesianas, entre otros.

La inclusión de estas nuevas tecnologías al sector agropecuario ya es una realidad, principalmente en aplicaciones de agricultura



de precisión, pero no es de uso intensivo en las empresas agropecuarias del noroeste argentino, más aún en la provincia de Catamarca, debido, en parte, a la escasez de dispositivos y componentes a precios accesibles, así como a la falta de operadores y personal capacitado para instalar y configurar los dispositivos y realizar el respectivo mantenimiento.

En respuesta a este desafío, se presenta este trabajo multidisciplinario que combina áreas como la electrónica, la informática, la ciencia de datos y las ciencias agropecuarias para una producción ganadera más eficiente y sostenible. El objetivo es desarrollar una solución tecnológica integral para el monitoreo y análisis de información en el sector ganadero, utilizando tecnologías IoT y Ciencia de Datos. Para lograr este objetivo se requiere recolectar y transmitir datos provenientes de sensores, capturarlos de manera remota y almacenarlos para su posterior procesamiento y análisis. Para ello se utilizan dispositivos “nodos”, cuya función primaria es la recolección y transmisión de datos, una puerta de enlace (Gateway) que realice la conexión LoRa a una red IP a través de WiFi, Ethernet, celular 3G o 4G, y plataformas de IoT que proporcionen herramientas y servicios para desarrollar y administrar aplicaciones y dispositivos conectados a Internet y visualizar la información recolectada.

II. CONTEXTO DE LA INVESTIGACIÓN

La investigación se inserta dentro de las líneas priorizadas “Plataformas Tecnológicas” y “Manejo Sustentable Agropecuario” en el marco de proyectos presentado en las convocatorias a formulación de Proyectos de Fortalecimiento de la Ciencia y de la Técnica-Vinculación Tecnológica 2022 (PFORCYT-VT 2022) y a Proyectos de Innovación y Transferencia (Pi+T)-2024, ambas de la UNCA.

El equipo de trabajo es multidisciplinario y está conformado por docentes investigadores de las carreras de Ingeniería Electrónica, Ingeniería Informática de la UNCA, que vienen trabajando y desarrollando actividades de docencia, investigación y vinculación tecnológica, en temáticas afines al proyecto. Integrantes del Laboratorio de Tecnologías de la Información y las Comunicaciones (LaTICs) y del grupo de

investigación en Internet de las Cosas (IIoT) de la Facultad de Tecnología y Ciencias Aplicadas (FTyCA) de la UNCA, con experiencia en proyectos de investigación relacionados a los sistemas embebidos, la Domótica y al desarrollo e implementación de nodos y redes inteligentes para propósitos generales y aplicaciones ganaderas. Desde el año 2019, se trabaja de manera conjunta con el INTA, a partir de un Convenio de Vinculación Tecnológica, con el objetivo de promover el desarrollo agropecuario y ganadero, atendiendo de esta manera a los crecientes requerimientos del sector, principalmente en zonas áridas donde el animal debe recorrer largas distancias diarias para buscar alimento.

El objetivo de estas convocatorias es brindar soluciones a problemas de interés directo para algún sector social y/o productivo, por lo que, durante el progreso de la investigación propuesta se busca promover y gestionar vínculos y alianzas estratégicas con Organismos públicos o privados y otras Asociaciones de productores, vinculados al desarrollo agro-tecnológico, para el fortalecimiento de la I+D y la transferencia efectiva de tecnologías surgidas del proyecto.

La implementación de IoT y Ciencia de Datos en la ganadería de precisión en la región NOA beneficiará directamente a dos grupos clave, con un impacto significativo en sus respectivas áreas:

a) Fortalecimiento de la ciencia, la investigación y la transferencia: Los grupos de investigación de la UNCA y de los sectores asociados al proyecto se fortalecerán al adquirir el conocimiento (know-how) necesario para diseñar, implementar y mantener nuevas tecnologías basadas en áreas científicas que revolucionan el siglo XXI. Este proyecto les permitirá avanzar en la frontera del conocimiento, posicionándose como referentes en la aplicación de tecnologías avanzadas en la industria ganadera.

b) Mejora de los procesos productivos: La implementación de IoT en la ganadería traerá consigo una serie de mejoras significativas en el manejo del ganado. La monitorización continua de variables permitirá detectar posibles enfermedades de forma temprana, control de comederos y bebederos,) ajuste del tamaño de parcela de pastoreo, eficiencia en el uso del agua, entre otras.

Asimismo, permitirá promover espacios de capacitación en el uso de IoT y la Ciencia de Datos con contenidos diferenciados acorde a los grupos objetivo, como así también difundir los resultados obtenidos mediante publicaciones y participaciones en reuniones científicas y académicas, permitiendo complementar la enseñanza en las carreras de Ingeniería Electrónica, Ing. en Informática de la UNCA.

III. DESCRIPCIÓN DE LA PROPUESTA

Su descripción se divide en dos partes para mejor entendimiento.

A. Sensado y almacenamiento de datos

La base del sistema es un nodo IoT, entendido como un dispositivo electrónico que permite que un sensor/actuador tradicional se comunice con un servidor o aplicación remota. Mediante las herramientas y tecnologías disponibles es sencillo incorporar un pequeño dispositivo que envíe y/o reciba datos, mediante cualquier tecnología de comunicación, a un servidor o una aplicación para la toma de decisiones a partir de la información recibida.

Un nodo IoT se distingue principalmente por:

- Pasar la mayor parte del tiempo en “modo espera” por lo que su consumo de energía depende de la relativamente baja energía que demande ese estado de reposo.
- Como parte de una red de datos, requieren mejor conectividad para subir datos que para bajarlos.
- Por su ubicación en una topología de red descentralizada y distribuida (y a veces con características de movilidad), la alimentación se realiza exclusivamente por baterías o pilas.

En la Fig. 1 se esquematiza el sistema realizado, consistente en el nodo IoT que posee un microcontrolador, varios sensores y un módulo de comunicación LoRa® integrado.

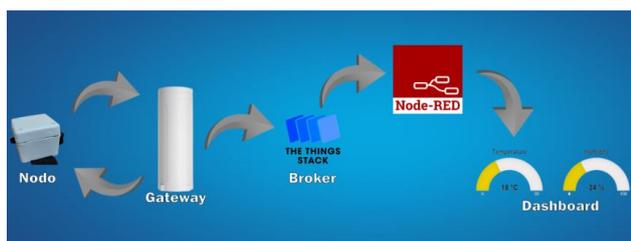


Fig. 1. Descripción de la propuesta

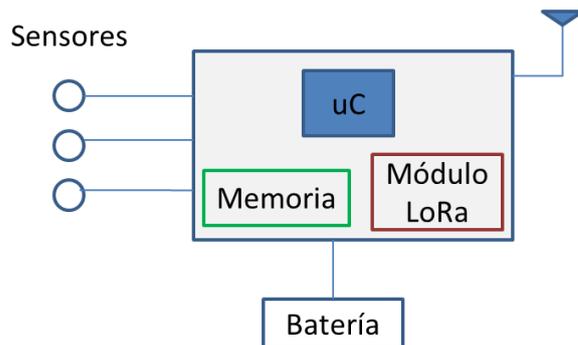


Fig. 2: Diagrama en bloques del nodo IoT.

El nodo se comunica a un gateway y éste al exterior via Ethernet, a través de protocolo LoRaWAN al servidor The Things Stack (TTS) que actúa como broker, quien envía la información por protocolo MQTT a Node-RED para su posterior visualización [3]. Se realiza a continuación una pequeña descripción de las diferentes etapas:

1. Nodo IoT [4]: se utilizó una placa Adafruit Feather M0 [5] que posee un procesador ARM Cortex M0 e incorpora un módulo de radio RFM9x LoRa 868/915 MHz. Posee un sensor de posicionamiento global GPS, para tomar datos de fecha y hora y la posición satelital del animal, sensores de temperatura y humedad ambiente, sensor de temperatura infrarrojo y una memoria EEPROM de 1024 Kbits. Se utilizan baterías de litio-polímero de 3,7V para alimentar todos los componentes del nodo. En la fig.2. se muestra su diagrama en bloque

Se configuró para realizar la transmisión a través de LoRaWAN en la banda de frecuencia ISM de 915 MHz. Un aspecto a considerar es el armado del paquete de datos a enviar, ya que las redes LPWAN están diseñadas para manejar cantidades reducidas de datos.

En un diseño posterior, y por razones de disponibilidad en el mercado, se reemplazó la placa por otra Lora32u4 de BSF France, la cual posee similares prestaciones.

2. Gateway: Se utiliza un gateway de código abierto para exteriores marca Dragino DLOS8. Permite la conexión LoRa a una red IP a través de WiFi, Ethernet, celular 3G o 4G. El DLOS8 es totalmente compatible con el protocolo LoRaWAN y posee bandas de frecuencia estándar preconfiguradas para utilizar en diferentes países [6]. Se debe registrar en la plataforma TTS y generar las credenciales correspondientes para los nodos a utilizar, en

idéntica zona y banda de frecuencias a la del nodo. Los mensajes son almacenados en un payload, en formato Hexadecimal, por lo que para entenderlos debemos convertirlos a texto.

3. Bocker: se utiliza la plataforma de IoT TTS que proporciona herramientas y servicios para desarrollar y administrar aplicaciones y dispositivos conectados a Internet. Proporciona una pila completa de tecnologías para habilitar la conectividad, la gestión de datos y la implementación de soluciones de IoT [7].

4. Node-Red: Es una herramienta de desarrollo basada en flujo para programación visual desarrollada originalmente por IBM para conectar dispositivos de hardware, API y servicios en línea como parte de la Internet de las cosas [8]. Recibe el tal cual lo hace en TTS, y dado que el payload está en base64, se debe decodificar para su correcta visualización.

B. Herramientas de Ciencia de Datos

Como herramienta de Ciencia de Datos se utilizó Python [9] por ser un lenguaje de programación de propósito general, desarrollado bajo una licencia de código abierto (administrada por Python Software Foundation), ampliamente utilizado en Ciencia de Datos debido a su versatilidad y una gran cantidad de bibliotecas especializadas. En la ganadería de precisión, Python se utiliza para el procesamiento de datos, análisis estadísticos y la creación de modelos de Machine Learning. Para la realización de tareas específicas asociadas al análisis y procesamiento de los datos, se utilizaron las librerías Pandas, NumPy, Matplotlib, Seaborn, y Folium. Esta última permite la representación y análisis efectivo de datos geospaciales en mapas interactivos, lo que facilita la comprensión de la información y ayuda a los productores a tomar decisiones más acertadas para optimizar la producción ganadera en la región.

IV. RESULTADOS OBTENIDOS

Respecto al nodo IoT, en la fig. 3 se muestra una foto de un prototipo del dispositivo (collar), que responde con éxito a los requerimientos iniciales [4]. Las principales limitaciones encontradas se circunscribieron a la falta de disponibilidad en el mercado nacional.



Fig. 3. Prototipo de Collar

La Figura 4 muestra el sistema de información (en desarrollo) que permite al usuario consultar un conjunto de datos (dataset), mostrarlos mediante una tabla, generar gráficos estadísticos y visualizar la ubicación georreferenciada del ganado.

Se generaron gráficos y visualizaciones personalizadas que muestran tendencias, valores atípicos, patrones y cambios en los datos ganaderos.

V. CONCLUSIONES

En el desarrollo de este proyecto se logró adaptar las tecnologías existentes a la realidad local, conjugar el IoT y la Ciencia de Datos, representando un enfoque innovador y una propuesta de contribución original y novedosa para la ganadería de la región noroeste del país. Durante el mismo, el equipo de investigadores adquirió una comprensión teórica y práctica de los principios fundamentales necesarios para implementar tecnologías LPWAN en la industria ganadera. A lo largo de este proceso hemos explorado detenidamente las aplicaciones potenciales de estas tecnologías en el monitoreo y control de diversas variables críticas en la gestión de la ganadería, como la salud del ganado, la ubicación y el comportamiento. Es importante tener en cuenta las limitaciones de la comunicación LoRaWAN, como ser, su capacidad limitada para manejar grandes volúmenes de datos y su alcance limitado en entornos urbanos densamente poblados.

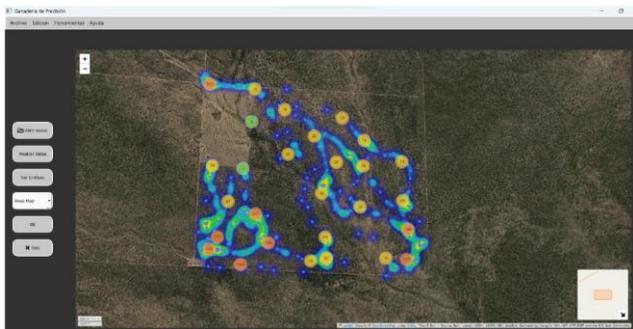


Fig. 4: Sistema de información.

Se espera continuar avanzando en el campo del IoT aplicado a la ganadería, y profundizar en el estudio de nuevas técnicas que permitan detectar patrones específicos adaptados a las necesidades de los productores ganaderos catamarqueños. de esta herramienta tecnológica en el sector.

REFERENCIAS

- [1] J.P.T. Soto, J.D.L.S.S. Suárez, A. B. Rodríguez & G. O. R. Cainaba, "Internet de las cosas aplicado a la agricultura: estado actual". Lámpsakos, (22), 86-105, 2019.
- [2] Baum, G., Artopoulos, A., Aguerre, C., Albornoz, I., & Robert, V. (2009). Libro blanco de la prospectiva tic: proyecto 2020. - 1a ed. - Buenos Aires: Ministerio de Ciencia, Tecnología e Innovación Productiva, Argentina. ISBN 978-987-1632-00-8.
- [3] M. Aranda, P. Beltramini, E. Cano, I. Lazarte & D. Villagrán, "Monitoreo de ganado a través de una red LoRaWAN en la provincia de Catamarca, JAIO, 2024
- [4] Aranda, M., Beltramini, P., Cano, J., Virragrán, L., Moreno, J., Gallina, S., & Herrera Conegliano, O. A. (2021). IoT aplicado a la ganadería extensiva. Revista Argentina de Ingeniería. Año 9, vol. 17. ISSN 2314-0925.
- [5] Adafruit Industries, Adafruit Feather M0 Radio with LoRa Radio Module. <https://www.adafruit.com/product/3178> (2024).
- [6] Dragino: DLOS8 Outdoor LoRaWAN Gateway User Manual: www.dragino.com. (2024).
- [7] The Things Industries: The carrier-grade LoRaWAN® Network Server that scales: <https://www.thethingsindustries.com/stack/> (2024)
- [8] Node RED: <https://nodered.org/>, (2024)
- [9] [10]Página oficial de Python. Accedido el 13 de septiembre de 2023. [En línea]. Disponible en: <https://www.python.org/>



Uso de redes descentralizadas en sistemas ciber-físicos basados en la placa ESP32

Ciro Edgardo Romero^{†1}, and Alejandro Elustondo^{*2}

[†]Dpto. de I + D + i, C&S Informática S.A.

Buenos Aires, Argentina

¹cromero@cys.com.ar

*XDK2MAM

Buenos Aires, Argentina

²alejandro.elustondo@nakama.io

Abstract—Existe una tendencia en desarrollar sistemas ciber-físicos basados en las placas de desarrollo ESP32. De forma paralela, se están incorporando redes sin autoridad central que favorezcan la seguridad y la confianza en los sistemas informáticos. Eventualmente estos conceptos convergen en único sistema que presentan complejidades de desarrollo de alta incertidumbre; propios del uso de nuevas tecnologías. Este trabajo describe experiencias en el desarrollo de sistemas capaces de recolectar variables ambientales integrados con redes descentralizadas, mal llamadas redes Blockchains. Se presenta una prueba de concepto, que busca comprender los problemas a resolver cuando este tipo de proyectos se integran en una red descentralizada. Al mismo tiempo, utiliza diferentes tecnologías abiertas, utilizadas en un entorno descentralizado, y analizar la viabilidad para el desarrollo productivo.

Palabras clave: ESP32; IOTA; Internet de las cosas; Blockchain.

I. Introducción

Hace unos años apareció un concepto llamado Internet de las Cosas (IoT), que utiliza componentes electrónicos para interactuar con servicios informáticos. Estos sistemas han sido impulsados por los avances de la potencia informática, la miniaturización electrónica y las interconexiones a través de internet [1]. Estos sistemas forman soluciones complejas, incorporando componentes informáticos digitales, capaces de interactuar directamente con el mundo que los rodea. En este tipo de desarrollo, la arquitectura del sistema define la ruta de un dispositivo a otro. Este camino determina la tolerancia a fallos del sistema y la capacidad de respuesta. Una de las topologías típicas de estos sistemas es la de cliente-servidor. Cuando se implementa este formato, los dispositivos son clientes de un sistema informático central, que auspicia como orquestador central. Este tipo de arquitectura se conoce como centralizada, y posee varias problemáticas conocidas y estudiadas [2].

II. Problemática de la centralización

Una de las problemáticas de la arquitectura cliente-servidor, es que todas las operaciones informáticas se llevan a cabo en un único dispositivo; el que funciona como servidor. Esto crea un punto crítico, donde una falla provoca que todo el sistema colapse. En sistemas IoT con este diseño, todos los datos recopilados desde diferentes dispositivos están bajo la autoridad del servidor. Por tal motivo, estos elementos suelen ser objetivo de varios tipos de ataques de seguridad y privacidad. [3]. Una opción para mitigar las debilidades y vulnerabilidades pueden utilizar redes descentralizadas, también conocida como DLT (por sus siglas en inglés, *Distributed Ledger Technology*).

A. Redes descentralizadas

La tecnología descentralizada resuelve (parcialmente) los problemas de seguridad que existen en un entorno público no confiable, donde dispositivos están conectados a través de internet [4]. Estos entornos son redes existentes, a las que se conectan dispositivos para acceder a Internet. La capacidad de mantener la integridad de las transacciones, elimina la necesidad de una autoridad central. Esta arquitectura ayuda a mitigar las deficiencias del modelo cliente-servidor, estableciendo una comunicación "entre iguales" para cada uno de los nodos [5]. La mejora en la tecnología DLT complementa la seguridad de sistemas IoT; tradicionalmente deficiente. Sin embargo, todavía existen algunos problemas con su adopción, como la escalabilidad, el algoritmo de consenso, la protección de datos, la eficiencia, la disponibilidad, el almacenamiento, la interoperabilidad, la estandarización, etc. Además, no existe consenso sobre modelos de referencia o mejores prácticas que definan como integrar tecnologías relevantes para cada dominio [6].

III. Adquisición de datos de forma distribuida

La implementación de un sistema ciber-físico utiliza una arquitectura que incluye interacciones, de comunicación independientes, entre diferentes elementos [7]. Con base en este concepto, se puede

entender que habrá dispositivos para tomar mediciones, y que estas serán enviadas a través de un canal de comunicación. Por otro lado, habrá dispositivos que reciban dichas mediciones, y realicen acciones en consecuencia. En la figura 1 se muestra un esquema de posibles elementos que formarían parte de un sistema IoT.

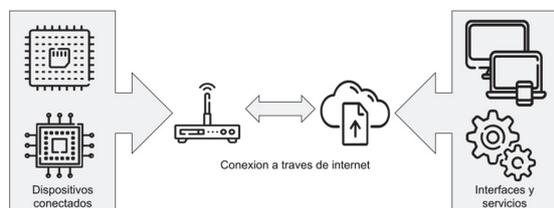


Fig. 1. Ejemplo de sistema IoT

IV. Sistemas IoT descentralizados

Desde una perspectiva teórica, los sistemas ciberfísicos pueden integrar objetos inteligentes con humanos de forma estandarizada. El objetivo del paradigma IoT es conectar operadores humanos y/o consumidores de sistemas entre sí, a través de canales seguros y confiables. Implementar una red descentralizada, es una decisión asertiva para aumentar la confianza en el diseño. Al mismo tiempo, pueda integrarse con otras redes para permitir una interoperabilidad [8]. Actualmente, el salto tecnológico necesario para lograr este objetivo se logra a través de la tecnología DLT, que proporciona una red de registros única, consensuada y descentralizada [5]. Los respectivos administradores actúan como agentes confiables, verificando la identidad y las credenciales de la red. La red estructura los datos con formato Blockchain y enlaces seguros para que la información sea rastreable [9].

La implementación de la tecnología DLT conlleva su propio conjunto de desafíos. El costo asociado a cada transacción es un punto de análisis a considerar, ya que puede ser muy elevado dependiendo del tráfico que requiera la red.

A. Estructuras de datos descentralizadas

En la arquitectura tipo Blockchain, existe el conocido como algoritmo de consenso. Bajo esta política se aceptan, o no, los bloques que sean insertados dentro de la red. La necesidad de que se llegue a un consenso, antes de insertar un nuevo bloque, provoca que no se mantenga esta sincronización. Esto provoca "bloques huérfanos", los cuales comprometen el rendimiento general de la red [10]. Esta problemática se ve resuelta con la propuesta de *IOTA Foundation* de utilizar una arquitectura propia conocida como *The Tangle*. La misma está basada en un concepto matemático llamado Grafo Acíclico Dirigido (DAG). Dicha arquitectura diseñada para admitir la transferencia de datos y valor, mientras utiliza una estrategia para aquellos bloques sin trazabilidad. En esta estructura,

los bloques "huérfanos" se fusionan nuevamente en la red [11].

Otra problemática en redes tipo Blockchain, es el tiempo que deben esperar las transacciones hasta que se incluyan en un bloque. Debido a las limitaciones en el tamaño del bloque, y al tiempo de producción de este, se crea congestión y tiempos de espera extensos. En el caso de *The Tangle* cada transacción se adjunta a transacciones cercanas. En consecuencia, el protocolo puede procesar varias cantidades de transacciones en paralelo [12]. En la figura 2 se muestra la comparativa entre una arquitectura Blockchain y por la arquitectura DAG.

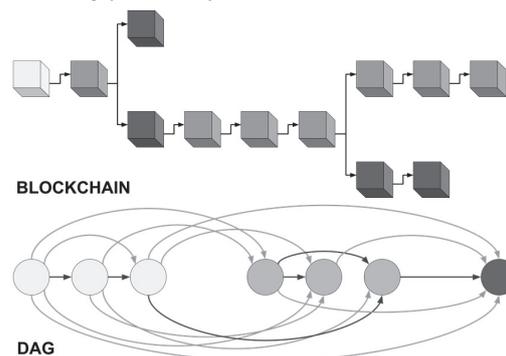


Fig. 2. Arquitectura Blockchain vs. DAG

V. Prueba de concepto

A continuación, se presenta una prueba de concepto, que intenta integrar dispositivos conectados a través de internet (un sistema IoT típico) con una red descentralizada tipo Blockchain. A partir de esta experiencia se abordaron algunos de los problemas mencionados anteriormente, así como la propuesta de soluciones. El objetivo principal es dejar un registro del proceso de desarrollo IoT, con las características antes mencionadas, que sirva como base para implementaciones similares en diferentes aplicaciones.

A. Consideraciones en sistemas ciberfísicos

Al considerar un entorno descentralizado, se pueden mostrar diferentes elementos que conforman todo el conjunto de forma compleja. Desde este punto de vista, resulta interesante realizar una prueba de concepto utilizando entidades con comportamiento básico para simplificar las interacciones que se encuentran en el sistema. En el sentido más simple, un entorno descentralizado puede describirse como un sistema de IoT. Sobre esta idea, se tienen que abordar las problemáticas de:

- Capacidades limitadas en los dispositivos
- Diversidad de tecnologías coexistiendo
- Seguridad informática
- Costos económicos y de computo
- Rigurosidad en el tratamiento de datos
- Eficiencia frente al diseño

- Complejidad del sistema general

B. Sensor de recolección de datos

La placa ESP32 es una buena opción para el desarrollo de sistemas IoT. Desarrollado por *Espressif Systems*, representa una familia de microcontroladores económicos. Gracias a su bajo consumo de energía y herramientas de código abierto, es adecuado para varios tipos de implementación. Al mismo tiempo, cuenta con documentación extensa y varias comunidades activas con ejemplos de programación en C/C++, Python, entre otros lenguajes de programación [13].

El sistema explicado en el presente trabajo se basa en la experiencia obtenida por el Departamento de Investigación, Desarrollo e Innovación de la empresa C & S Informática S.A, donde se abordaron algunas de las problemáticas anteriores [14]. A partir del trabajo mencionado, se logró la comunicación entre nodos, basados en la placa ESP32, y la red descentralizada. En la figura 3 se ilustra el dispositivo desarrollado, explicado anteriormente.



Fig. 3. Prototipo de sensor con conexión a internet

Los datos recopilados por los nodos se envían a la red IOTA, diseñada para simular la comunicación típica de sensores inteligentes en un sistema descentralizado. El objetivo es desarrollar sistema que sean fáciles de mantener desde una perspectiva unitaria, utilizando lenguajes de programación de alto nivel.

C. Código *bare-metal*

Un lenguaje de programación de código interpretado, fácil de aprender y legible, es Python [15]. Este lenguaje multiparadigma es capaz de soportar programación imperativa, programación funcional y, parcialmente, orientada a objetos. Además, es un lenguaje dinámico y compatible con diversas plataformas. En la comunidad de Python existe una reimplementación, sencilla y eficiente, optimizada para ejecutarse en microcontroladores. Este subconjunto pequeño de la biblioteca estándar, es conocido como Micropython [16].

D. Microcontrolador y sensores

Como parte del sistema IoT, se implementó un dispositivo que actúa como un nodo que recopila variables de entorno. El nodo contiene el módulo BMP180, que se basa en el sensor del mismo nombre para medir la presión, temperatura y humedad [17]. La placa realiza la configuración de sus componentes, así como los servicios diseñados para cada finalidad. Cuenta con un LED RGB, que relaciona colores con el estado de la placa. De esta forma, podrás comprobar el estado del dispositivo cuando este desconectado del

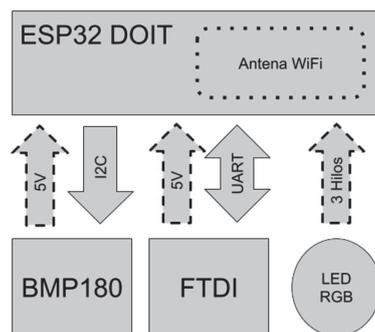


Fig. 4. Elementos que componen el nodo sensor

ordenador. En la figura 4 se muestran los componentes del nodo sensor basado en la ESP32.

El dispositivo se conecta a internet y realiza una configuración del reloj interno, a través del protocolo de tiempo de red (NTP, por sus siglas en inglés) [18]. El funcionamiento general se inicia, cuando el microcontrolador se comunica con los sensores para tomar medidas. Estos datos se almacenan en memoria y se suma un identificador único de dispositivo. Adicionalmente se recuperan la hora y fecha de medición. Antes de enviar un mensaje, se debe comprimir mediante una función criptográfica, conocida como "hash" [19]. Este es un algoritmo que convierte cualquier bloque de datos en una nueva cadena de longitud fija. Independientemente de la longitud de los datos de entrada, el valor de salida siempre tiene la misma longitud.

Finalmente, los datos se envían desde una función *request* en formato JSON. A través de este dispositivo, se restablece el funcionamiento básico del sistema de recolección y distribución de datos. Al utilizar sensores como nodos conectados a Internet, las variables físicas se pueden convertir en variables que pueden ser utilizadas por los sistemas informáticos. Estos datos luego se transmiten a la red descentralizada para que otros sistemas los utilicen.

E. Red distribuida

La propuesta de valor en un sistema distribuido, es poder crear una red topológica colectiva, de tal manera que ninguno de ellos tenga la exclusividad de filtrar la información. Siguiendo las recomendaciones de la propia comunidad IOTA, se utilizó el nodo transaccional *Hornet*. Este tipo de nodo, contiene dos funcionalidades dentro de un sistema descentralizado. En primer lugar, permite a los clientes interactuar con *The Tangle* y comunicarse con los nodos. Esto lo hace como una interfaz de envío y recepción de mensajes. Por otro lado, permite a los clientes sondear nodos en busca de nuevos mensajes y otros eventos. Esta es útil para monitorear *The Tangle* y actualizar parámetros [20].

F. Comunicación en la red

Cada nodo se puede identificar de forma individual, mediante un identificador único (del inglés *peer identity*). Las conexiones con un nodo IOTA, se establecen a través de un protocolo conocido como *peer discovery*. Este se utiliza para exponer una interfaz que proporcione una lista verificada de nodos. La implementación del protocolo proporciona una lista codificada y confiable, administrada por la comunidad y sus colaboradores [21]. Estos protocolos se identifican y autentican, utilizando criptografía de clave asimétrica. Se utilizó un nodo público provisto por la misma *IOTA Foundation*. Los dispositivos descriptos en la sección anterior, se comunican con el nodo a través de *requests* configuradas en una ruta especificada en una API intermedia, que implementa la función *hash*. En la figura 5 se muestra un diagrama para ilustrar el esquema de comunicación.

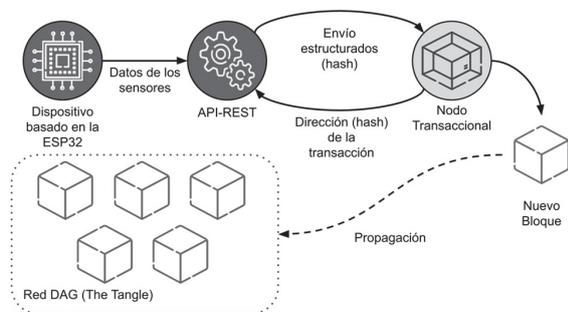


Fig. 5. Comunicación entre elementos del sistema

Al enviar el mensaje, en el caso de que falten, el nodo intentará auto completar los campos de identificación de red (*networkId*), el identificador de mensaje (*parentMessageIds*) y un número arbitrario conocido como "nonce". Si faltara el cuerpo principal del mensaje con la información (llamada *payload*, en inglés), el mensaje se generará igualmente pero vacío. A continuación, se ilustra estructura del mensaje:

```
{ "networkId": "00457867963673787875",
  "parentMessageIds": [
    "g31pxjk4q9wzmhrgptxx (...)",
    "49ch624cnwtc4a7qzh88 ( . . . ) " ,
    "q83igenu16j80013k7mj (...)",
    "c4m08e672rvcr35233xf (...)"
  ],
  "payload": { "type": 2
    ,
    "index": "4 kjr7ier33rhkiwtc9y0 ",
    "data": "bbwquq2kn (...)"
  },
  "nonce": "4402323"
}
```

Si la comunicación es exitosa, el mensaje se almacenará en *The Tangle*. Cuando la transacción es exitosa, se devuelve un *hash* como identificador de la transacción. El mismo puede ser utilizado por otros elementos dentro del sistema, si es necesario localizar y decodificar la información insertada.

VI. Conclusiones

La disponibilidad de placas ESP32 en el mercado hace que sea una buena opción para el desarrollo de diversos sistemas integrados. Además, la baja curva de aprendizaje de Micropython hace que este conjunto de opciones sean factores óptimos en el desarrollo de una prueba de concepto.

Al implementar proyectos productivos, los servicios integrados de transmisión de datos y cifrado proporcionan un nivel de seguridad interesante. Los sistemas ciber-físicos pueden aumentar la confianza en los usuarios del sistema.

La red IOTA reduce significativamente los costos asociados con la implementación. Esto beneficia a los desarrolladores de sistemas IoT, que utilizan placas como la ESP32 y otras con capacidades similares. El código que se comunica con redes descentralizadas se puede integrar en cualquier tipo de sistema de medición inteligente. Sus métodos son perfectamente transportables a otros lenguajes para otros microprocesadores. En otras palabras, se puede distribuir un sistema de producción ya instalado y agregar hardware y software en pequeñas cantidades.

El resultado final del trabajo realizado es un sistema mínimo perfectamente funcional capaz de escalar en implementaciones que requieran mediciones ambientales sin depender de un servidor central.

VII. Trabajo futuro

Actualmente existen microprocesadores con una capacidad de computo superior la versión de ESP32 utilizada en este trabajo. Espressif lanzó la versión ESP32-C6 mejorada. Este nuevo kit de desarrollo muestra más y mejores prestaciones que sus predecesores. Según el fabricante, es un microcontrolador ideal para aplicaciones IoT.

Por otra parte, la *IOTA Foundation* se encuentra en proceso de actualización de la red para mejorar la



integración. Promoviendo una versión a IOTA 2.0, dentro de la cual se promociona un proyecto llamado *Shimmer Network*¹. Es último permite una Interoperatividad con otras redes Blockchain, así como una extensión de sus prestaciones.

References

- [1] K. Rose, S. Eldridge, and L. Chapin, "The internet of things: An overview," *The internet society (ISOC)*, vol. 80, pp. 1–50, 2015.
- [2] C. Rowland, E. Goodman, M. Charlier, A. Light, and A. Lui, *Designing connected products: UX for the consumer Internet of Things*. " O'Reilly Media, Inc.", 2015.
- [3] M. ATT&CK, "Mitre att&ck," URL: <https://attack.mitre.org>, 2021.
- [4] B. Farahani, F. Firouzi, and M. Luecking, "The convergence of iot and distributed ledger technologies (dlt): Opportunities, challenges, and solutions," *Journal of Network and Computer Applications*, vol. 177, p. 102936, 2021.
- [5] J. Kehrl, "Blockchain explained," *Netguardians [en línea]*. [Data de consulta: 25 de juny de 2017]i <https://www.netguardians.ch/news/2016/11/17/blockchain-explained-part-1>, 2016.
- [6] H. F. Atlam and G. B. Wills, "Intersections between iot and distributed ledger," in *Advances in Computers*. Elsevier, 2019, vol. 115, pp. 73–113.
- [7] R. Hadidi, J. Cao, M. S. Ryoo, and H. Kim, "Robustly executing dns in iot systems using coded distributed computing," in *Proceedings of the 56th Annual Design Automation Conference 2019*, 2019, pp. 1–2.
- [8] R. Buyya and A. V. Dastjerdi, *Internet of Things: Principles and paradigms*. Elsevier, 2016.
- [9] V. Gisbert Soler and A. I. Perez Molina, "Blockchain vs iso 9001: 2015," *3C Tecnologia*, vol. 8, no. 2, pp. 37–48, 2019.
- [10] I. Foundation, "Coordinator. part 2: Iota is a dag, not a blockchain," <https://blog.iota.org/coordinator-part-2-iota-is-a-dag-not-a-blockchain-2df8ec85200f/>, 11 2018.
- [11] —, "The transparency compendium," <https://blog.iota.org/the-transparency-compendium-26aa5bb8e260/>, 06 2017.
- [12] W. F. Silvano and R. Marcelino, "Iota tangle: A cryptocurrency to communicate internet-of-things data," *Future Generation Computer Systems*, vol. 112, pp. 307–319, 2020.
- [13] A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the esp32 microcontroller module for the internet of things," in *2017 Internet Technologies and Applications (ITA)*. IEEE, 2017, pp. 143–148.
- [14] C. E. Romero, A. M. Elustondo, R. K. Der Boghosian, and M. C. Fontela, "Nodo experimental de registro e inmutabilidad de variables ambientales," in *III Simposio Argentino de Informatica Industrial e Investigacion Operativa (SIIIO 2020)-JAIIO 49 (Modalidad virtual)*, 2020.
- [15] V. Frittelli, D. Serrano, R. Teicher, F. Steffolani, M. Tartabini, J. Fernandez, and G. Bett, "Uso de python como lenguaje inicial en asignaturas de programacion," *Editor Responsable*, vol. 132, 2013.
- [16] N. H. Tollervey, *Programming with MicroPython: embedded programming with microcontrollers and Python*. " O'Reilly Media, Inc.", 2017.
- [17] Bosch, "Bmp180," https://ae-bst.resource.bosch.com/media/_tech/media/product_flyer/BST-BMP180-FL000.pdf, 04 2013.
- [18] IETF, "Simple network time protocol (sntp) version 4 for ipv4, ipv6 and osi," <https://datatracker.ietf.org/doc/html/rfc4330>, 06 2006.
- [19] B. Preneel, "Cryptographic hash functions," *European Transactions on Telecommunications*, vol. 5, no. 4, pp. 431–448, 1994.
- [20] I. Foundation, "Hornet. community driven iota node," <https://wiki.iota.org/hornet/welcome>, 11 2021.
- [21] T. Neudecker and H. Hartenstein, "Network layer aspects of permissionless blockchains," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 838–857, 2018.

¹ Detalles del proyecto: <https://shimmer.network/>



Desarrollo Abierto para Internet de las Cosas y Automatización Industrial: Evaluación de Rendimiento en Entornos Críticos

Ricardo Antonio López

Universidad Nacional de la Patagonia

SyA: Sistemas y Automatismos

Trelew, Argentina

lopez.ricardo@gmail.com

Resumen— Este trabajo se enfoca en el diseño y desarrollo de un software, con implementación de un prototipo, para aplicaciones en Internet de las Cosas y automatización industrial, que permitió realizar pruebas exhaustivas para evaluar su rendimiento y determinar factores de mérito.

Se simularon diversos entornos, incluyendo múltiples microcontroladores en redes de diferentes tipos y capas. Se recrearon escenarios con ocurrencia de eventos múltiples en tiempos breves —característicos de entornos eléctricos bajo fallo—, donde el registro preciso es crucial para diagnóstico, detección de puntos de falla y la planificación de inversiones.

Además, la implementación permitió evaluar el rendimiento de un **RTOS** (Sistema Operativo en Tiempo Real) en modos multitarea y multinúcleo, sobre un microcontrolador de 32 bits, de tamaño moderado, ampliamente compatible y popular en el mercado.

Durante el desarrollo, se prestó especial atención a una arquitectura modular del software, asegurando escalabilidad e interoperabilidad entre los diversos módulos del sistema.

Palabras clave: InternetDeLasCosas, Automatización, RTOS, Scada, Sincronización.

I. INTRODUCCIÓN

Para el desarrollo e implementación del concepto propuesto, se escribió un software al que denominaremos **ds7.1**, con tareas definidas y desplegadas en diferentes módulos, para ser portado sobre microcontroladores de amplio espectro, en cuanto a tamaño de procesador(es), memoria y entrada/salida de propósito general (**GPIO**). Se ha utilizado RTOS y en particular, se lo ha experimentado sobre algunas de las unidades que integraron la red prototipo. De estas unidades, se eligió el microcontrolador ESP32 WROOM de Espressif Systems, por su alta capacidad al poseer dos núcleos procesadores y poder así llevar a cabo un estudio más profundo del problema de sincronización horaria [2][3] y registro de eventos [8], que es en sí uno de los subproductos que surgen del presente trabajo.

Sobre el proyecto global, se plantearon desde inicio los siguientes objetivos:

1. Garantizar portabilidad del software, sobre la base de utilización de un soporte Hardware que cumplimente cierto estándar, con el objetivo de reducción de costos y facilidad de su adquisición en el mercado.

2. Utilizar protocolos de comunicaciones sobre diferentes capas físicas y de comunicaciones, todas del estándar de Sistemas Distribuidos [1], con una capa de aplicación [5] adecuada a las necesidades de cada caso a tratar.



Otro de los aspectos que se ha tenido muy en cuenta, es la de hacer posible la integración de IOT en sistemas preexistentes, para reemplazo progresivo de aquellos ya obsoletos. Esta característica puede resultar interesante en ambientes como los de producción agropecuaria, donde la evolución tecnológica suele ser lenta, o instalaciones de Pyme, que puedan poseer una electrónica que aunque obsoleta, aún sigue en funcionamiento por confiabilidad o factores económicos y de oportunidad. Concretamente para este propósito, en el ambiente de simulación se han utilizado diferentes placas de distintos fabricantes, con diferente periferia de I/O y capas físicas y de comunicaciones como Serie FDX, RS485 o Ethernet, además de la ya clásicas por su ubicuidad como WiFi, USB o Bluetooth.

II. RED DE IEDS

Se presenta aquí una descripción de la filosofía conceptual de la Red. Se define la distribución jerárquica de los IEDs, en relación con lo previsto en el software particularmente en sus tablas de configuración, para lograr un funcionamiento coherente y eficiente.

Como se consigna en la Figura 1, cada IED de la red posee un **Identificador Único (Id)**, para su uso en la capa de aplicación, independiente de su dirección de Red. Los diferentes protocolos de comunicaciones empleados en el sistema han determinado la generación de **Canales**, de modo tal que cada canal es tratado con sus particularidades en cada capa. La tabla de la Figura 2 presenta una síntesis de funciones y características de los IEDs de la red bajo estudio.

El núcleo del sistema lo compone un Maestro Concentrador denominado **CoMaster**, que se encarga de las comunicaciones externas con internet y que periódicamente, a través de la red interna consulta a diferentes IEDs esclavos (**Slaves**). Estos, a su vez están en vínculo directo con el campo a través de su GPIO.

El CoMaster, mediante consulta periódica, se nutre de los estados y eventos que se generan en el campo, los incorpora a sus propias tablas de datos, para luego permitir su requerimiento desde Servidores y/o diferentes Interfases Hombre Máquina (**MMI**), situadas tanto en la red interna o la externa (Internet), característica de un Sistema Distribuido [1][2]. Esto es, el CoMaster contiene las tablas centrales de tiempo real, de trabajo de todo el sistema subyacente y por ende es quien alimenta al Servidor WEB y a las distintas MMIs.

Podrá haber, según fuera el tamaño de la instalación, un Servidor que contenga una BDD central, relacional y más elaborada del sistema, que permita la consulta y hasta la existencia de pantallas con mímicos visuales que representen diferentes áreas para la operación del sistema bajo supervisión y control.

Por su lado, los Slaves se encargan de la adquisición directa desde el campo donde, por estar aliviados de tareas complejas de comunicación debido a la existencia de capas más simples, se desenvuelven con mayor velocidad, libres de procesos con esperas que puedan monopolizar la CPU. Estos IED poseen amplia y variada periferia, son concebidos para dedicarse a los procesos locales y de recepción de eventos, con sincronización horaria interna y precisión por debajo de 1 ms.

#	IED	Función	Hardware	BDD	Automatismo Local	Estado
1	8	Concentrador-Master	ESP32	Global	Ninguno	Desarrollado
2	10	Master - Slave	ESP8266-01	Parcial	Adquisición	Desarrollado
3	15	Master - Slave	ESP8266-01	Parcial	Adquisición	Desarrollado
4	9	Slave	Arduino DUE	Local	Semaforización	Desarrollado
5	7	Slave	UNO 32	Local	Relés Tempo	Desarrollado
6	5	Slave - Bridge	Intel Galileo Gen1	Local	Adquisición	En prueba
7	6	Slave	Intel Galileo Gen1	Local	Relés Tempo	Desarrollado
8	3	Slave	Edu-CIAA	Local	Adquisición	En prueba

Figura 2: Tabla de Funciones de IEDs



Las placas constitutivas de los slaves, típicamente están excluidas del manejo de WIFI y en general también de un Stack TCP/IP – procesos inherentemente “pesados” -, lo que les permite sobrellevar fácilmente rutinas, usualmente con interrupciones para la ejecución de tareas críticas. Normalmente, los procesos del Stack de comunicación externa, se apoderan del procesador en ocasión y con duración impredecibles, haciendo que el tratamiento interruptivo determinístico de los procesadores sea prácticamente inviable. Esto exige la adopción de placas que tengan un manejo interruptivo más predictivo. Los procesadores de la gama estándar, son de 32 bits (54 puertos digitales I/O, 8 PWM, 14 analógicos, 3 puertos serie, SPI, etc.), atributos que los hacen totalmente adecuados para escalabilidad. Ello permite alta velocidad en adquisición de eventos [12], retroalimentación y control automático generado en el propio IED y transferencia de comandos desde una MMI[15], por mencionar algunas de las actividades más críticas en el campo.

Para posibilitar la transferencia de datos en un modo ubicuo, se utiliza comunicación directa en modo WEB o indirecta mediante protocolo MQTT [15][16]. En ambos casos con un protocolo de aplicación propietario de código libre[5][7], denominado Mara-1, en su versión actual v6.1.

Los procesos de comunicación y los de adquisición y control soportados en distintos procesadores, permiten tener registro de eventos con precisión horaria interna por debajo de milisegundo [9][10][11]. En general se utiliza NTP en internet, para obtención de UTC (Tiempo universal coordinado), con ajuste derivativo [2][3], para asegurar una precisión externa del orden de 10 ms.

Para ciertos casos donde la precisión horaria por requerimiento de las aplicaciones tenga una exigencia mayor, se ha previsto el agregado de GPS con Temporización por Flanco, que permite llegar a resoluciones del orden de microsegundos. Ello es particularmente útil por ejemplo, en localización de puntos de falla en líneas de alta tensión, como la línea larga Futaleufú - Puerto Madryn, de 570Kmts. Ello explica - como se verá más adelante en el apartado específico -, la búsqueda de precisiones elevadas en el registro de eventos y sincronización entre IEDs.

Los distintos IEDs simbolizados en funcionamiento jerárquico en la Figura 1, están provistos del mismo software ds7.1. Se varía su configuración para funcionalidad, mediante archivos (*.h) y de esta manera se ha logrado implementar la red descrita, para pruebas de funcionamiento con diferentes placas de distintos fabricantes, de distintos tiempos de aparición en el mercado y consecuentemente diferente tecnología, cantidad de procesadores y memoria, a saber: Uno32 (Microchip - 32 bits), Galileo Gen1 (Intel Pentium - 16bits), Arduino DUE (ARM Cortex-M3 - 32 bits), ESP8266 (Tensilica L106 - 32 bits), EDU CIAA (NXP LPC4337 dual core - 32 bits), ESP32 (Xtensa LX6 dual core - 32 bits). Esta lista no es extensiva dado que pueden agregarse otras placas de fabricantes que cumplan el estándar.

Como ya se ha indicado, las pruebas efectuadas se basaron en el IED más potente de los enunciados, a quien se ha asignado la función más crítica que es la de funcionar como CoMaster de la red. Este procesador tiene a su vez su propia GPIO que le permite interactuar con el campo como otro IED de la RED. De hecho, la consulta como IED local está inscrita en la agenda de consulta del proceso CoMaster. Esto es, el CoMaster cumple las funciones independientes de: Encuesta de la red subyacente, encuesta de su propia periferia y consulta desde un nivel superior. Debido a ello, se ha distribuido la tareas en diferentes procesos y en ciertos casos, de ser posible, en diferentes núcleos de un mismo microcontrolador.

Por otro lado, ciertos IED que se encuentran descendiendo jerárquicamente en la red, también tienen las funciones concurrentes de ser: a) Master de su sub-red, b) Adquisidor de datos y controlador de su campo, con su propia GPIO y c) Slave para el CoMaster situado jerárquicamente arriba. Por ejemplo, en la Figura 1, esta triple función la tienen los IEDs de Id: 10 y 15. Las distintas configuraciones y consecuentes roles de los microcontroladores, se logran como se ha dicho, con cambios dentro de los archivos de configuración del software.

Con todo lo descrito hasta aquí, se efectúa un resumen de las siguientes características principales contenidas en el sistema ds7.1.



II.1. CARACTERISTICAS

1. Software sobre Free RTOS Multitask y Multicore. Algunas tareas son del tipo colaborativo con Timers. Permite la realización de las tareas de sistema, comunicaciones y procesos de usuario. El usuario puede programar su propio automatismo y asignarle un slot de ejecución.
2. CPUs: ESP32, ESP8266, NodeMCU, Arduino DUE, MEGA y otros compatibles.
3. Comunicación externa por WIFI. Sincroniza la Hora Global con NTP, con sincronización externa del orden de 10 ms con ajuste derivativo.
4. Sincronización interna con precisión por debajo de 1 ms. Registro de eventos Digitales y Analógicos. Los eventos digitales se clasifican en: **Rápidos y Lentos**. Los eventos Rápidos se registran por **interrupción** del hardware, con una discriminación típica de ± 1 dms (diezmilisegundo = $s/10^{-4}$). Los eventos Lentos se registran por **polling**, con una discriminación típica de 10 ms ($s/10^{-2}$). Registro de Eventos por Tipo, Código y Motivo, con TimeStamp.
5. Software/Hardware configurable y escalable, con tablas de parámetros con Registro en EEPROM.
6. Comunicación directa e indirecta para adquisición de datos y control (Web Server y PubSub MQTT) sobre el CoMaster, que mantiene Tablas de Datos del sistema Global. MMI por Dashboard, SmartPhone (App) o PC/MAC (Aplicación WEB).
7. Comunicación interna en red mediante diferentes capas físicas estándar: WIFI, Ethernet, Bluetooth, SPI, red cableada serie (Cable físico FDX, o RS485, o Fibra óptica).
8. Entradas y salidas analógicas y digitales, escalables con distintas placas de adquisición.
9. Envío de eventos y recepción de Comandos por Mensajería Instantánea, MQTT o APP customizada.
10. Comandos de Adquisición de datos, Control, Diagnóstico y Configuración en línea, sobre subcapa de aplicación Mara 1-v6.1 (propietario / abierto).

III. SOFTWARE

Todo el concepto que se ha descrito hasta aquí, reúne la elección de arquitectura del Hardware, Comunicaciones y un Software propietario, abierto que se ha publicado en un repositorio de uso libre y dado que uno de sus atributos es su utilización en Domótica, rama de IOT, se lo ha denominado "Domo" Sapiens (ds7.1) y actualmente se encuentra en su versión 7.1.6.

III.1. ESTRUCTURA

Se describe aquí la filosofía conceptual del Software, que se ha escrito en C++ y separado en 18 fuentes (actualmente), que contienen los diferentes módulos para facilitar modificación, escalabilidad y otras necesidades que pudieren surgir. En el desarrollo se ha tenido en cuenta especialmente, la posibilidad de creación de procesos locales ad hoc, a quienes se puede asignado su respectivo "slot" de ejecución dentro de las tareas de RTOS, permitiendo la confección de los algoritmos necesarios para el proceso local del cual se encarga el microcontrolador, ello independientemente de sus tareas globales predeterminadas, como son la adquisición del estado y comunicación con el CoMaster.

Las tareas se han distribuido en diferentes Tareas del RTOS, en diferentes núcleos cuando existe esa posibilidad, asignando al Núcleo 0 el manejo de WIFI y el stack TCP/IP donde requieren sus propios tiempos y oportunidad de ejecución, algunos de ellos críticos. Se reserva el Núcleo 1 para las interrupciones del usuario y el manejo del Clock Interno con ajuste derivativo, que también requiere tiempos precisos de ejecución para mantener la granularidad y precisión que se han impuesto como principios preliminares de diseño.

Se ha dotado al software de una salida de debug que se otorga en la **FIGURA 3** donde se puede apreciar el manejo de mensajes con los IEDs, ya sea en el Rol de Master, Slave o Master-Slave. También puede observarse la ocurrencia de eventos de distinto tipo 0 al 3, especialmente.

El protocolo de Aplicación mara-1 se ha elegido en principio, por el elevado conocimiento que se tiene del mismo dado que fue objeto de proyectos específicos. Al ser abierto, posee un forma muy sencilla de crear nuevos



comandos adecuados a una aplicación específica, por encima de los generales de su inicio. Su formato es de un byte con la secuencia comando y argumentos: comX, arg1, arg2, arg3...Donde X= Nro. de Comando (de 0 a 254) y argX son los argumentos que se adicionen para el mismo. Algunos de los comandos generales que se han definido son: Requerimiento de Estado del IED, Requerimiento de Eventos, Puesta en Hora, Lectura de Variables para Debug, por listar los más importantes.

Cabe destacar que el protocolo posee, además de su capa de aplicación, dos capas de comunicaciones (Física y Transporte) donde se verifican CheckSum, secuencia de paquete, direcciones de origen y destino y bytes de encuadre. Este formato de tres capas, se utiliza especialmente en comunicación serie FDX y multipunto. En situaciones en que se requiere el estándar [1], los paquetes del protocolo son encapsulados en TCP/IP.

III.2. TIEMPO Y SINCRONIZACIÓN

En la Unidad CoMaster, el UTC se recibe por NTP. Internamente, se define una variable **PeHStatus**, que indica el modo en que el UTC es aplicado a la CPU local. Al inicio, con PeHStatus que va en aumento desde 0 hasta 2, el UTC se aplica directamente al reloj local en tanto se pueda determinar el tiempo más probable.

Consecuentemente, esta imposición del valor de UTC al reloj local hace que el mismo pueda ir hacia atrás, rompiéndose el principio de monotonía necesario para el correcto registro de eventos secuenciales. Pero ello solo sucede al arranque del microcontrolador.

En cada ajuste de tiempo, denominado **PeH** (puesta en hora), se calcula el **Sesgo** existente entre la hora local y la de llegada del paquete NTP, corregido por Algoritmo de Cristian [3]. Si el Sesgo es positivo, por convención propia, se indica que el microcontrolador adelanta respecto a la hora UTC. En el arranque, que es la situación que se ha presentado en el volcado central de la Figura 3, se observa que se efectúan lecturas consecutivas, con diferencias del orden de 15 segundos entre ellas, hasta que se toma un UTC que ocasiona un Sesgo bajo (de 23 dms, en este caso). En ese instante es cuando concluye la PeH por imposición. Se pone la variable PeHStatus = 3 y a partir de allí, el UTC que se obtenga se aplica sólo corrigiendo la deriva, de modo que el reloj local nunca vuelve hacia atrás conservando así su monotonía.

Lo antedicho implica que se garantiza que la corrección, que siempre será igual a la granularidad temporal, en este caso de ± 1 dms y se hace con un período de aplicación variable e inversamente proporcional al Sesgo resultante en la lectura, con el objetivo de acelerar la convergencia a la hora UTC.

```

Hola! Comencemos... uCNet4: Domo Sapiens1 - versión: 7.1.4. 22/jul/2024, 13:11 IED: 05. Esp32
Causa de Anterior Reset: 0. Modo Reset: 0
Previsión de apagado de Outputs... Esperando si se pulsa Boton2 para cargar la EEPROM por Default..
Lectura de Parámetros desde EEPROM ...EEPROM successfully committed
Access Point: LaVidaEsBella
Servidor MQTT: test.mosquitto.org
INICIALIZANDO I/O...I/O Inicializada.
Connecting to LaVidaEsBella .....WiFi connected . IP address: 192.168.1.135
. El Tiempo local es: 2024-6-12 17:30:12.0000 DoW: 0 DoY: 180 PeHStatus: 0
Fin del SETUP. Ahora...Loop!..... Tareas Creadas.....

Trama Publicada: uCNet4Mqtt/Ans/05/91 01 20 20 04
Trama Publicada: uCNet4Mqtt/Ans/05/-.8 Ev tipo: 3 IED: 05. IED Up.EEPROM SAVE.! . Reset:0 TimeStp: 2024/8/21 8:27:19.9
Trama Publicada: uCNet4Mqtt/Ans/05/-.7 Ev tipo: 3 IED: 05. PeH. Sesgo: 0. PeHStatus: 1 TimeStp: 2024/8/21 8:27:19.0
Trama Publicada: uCNet4Mqtt/Ans/05/-.6 Ev tipo: 3 IED: 05. PeH. Sesgo: 0. PeHStatus: 2 TimeStp: 2024/8/21 8:27:32.0
Trama Publicada: uCNet4Mqtt/Ans/05/-.5 Ev tipo: 3 IED: 05. PeH. Sesgo: -5847. PeHStatus: 3 TimeStp: 2024/8/21 8:27:47.59689
Trama Publicada: uCNet4Mqtt/Ans/05/-.4 Ev tipo: 3 IED: 05. PeH. Sesgo: -1290. PeHStatus: 3 TimeStp: 2024/8/21 8:28:2.64246
Trama Publicada: uCNet4Mqtt/Ans/05/-.3 Ev tipo: 3 IED: 05. PeH. Sesgo: -204. PeHStatus: 3 TimeStp: 2024/8/21 8:28:17.65332
Trama Publicada: uCNet4Mqtt/Ans/05/-.2 Ev tipo: 3 IED: 05. PeH. Sesgo: 23. PeHStatus: 3 TimeStp: 2024/8/21 8:28:32.23
Trama Publicada: uCNet4Mqtt/Ans/05/-.1 Ev tipo: 3 IED: 05. PERDIDA respuesta IED: 3357 TimeStp: 2024/8/21 8:28:36.3357
CountLoop00: 0. CountLoop10: 3994. CountLoop11: 766. CountLoop12: 799. Millis: 1670327
Temperatura *C *F: 22.0 71.5 Temperatura PROM *C *F: 21.7 71.1
....
Emission NTP packet...Leyendo Pack NTP. Delay[ms]: 20.60.100.140.180.
Hora NTP. Packet recibido= 48. Unix time= 1724015077. delayLectura: 2200 dms [ms/10]
El UTC corregido alg. Cristian es: 2024-8-18 18:04:38.0243 .Fracción en décimas de ms [dms] PeH: (Com 12).
ComError: 0x10. Dif.dias: -66. Dif.Horaria[s]: -2035. fraccionDMS[dms]: 5. Sesgo[dms]: 0. PeriNTP: 15.
(local...) TimeStp[i]: 0x18 8 12 12 4 26 F3
(entrante) BufAplic[i]: 0x18 8 12 12 4 26 F3
. El Tiempo local es: 2024-8-18 18:04:38.0243 DoW: 7 DoY: 231 PeHStatus: 1
...
** PeH-FL: FE 11 FF 40 02 12 18 08 12 12 09 38 88 13 07 3E 37 - Time: 2024-8-18 18:09:56.5000
    
```

Figura 3: Dump del Debug de arranque, puesta en hora inicial y emisión de PeH del IED.



Para lograr el objetivo de ajuste derivativo, se corrige el valor del incremento que se aplica al reloj a efectos que su velocidad varíe, en cada ciclo del Timer Hardware del microcontrolador. Por ejemplo, si el Sesgo obtenido es positivo, señal que el reloj local adelanta, se decrementa en 1 dms (un *Leap* dms), el valor que sistemáticamente se aplica al reloj y consecuentemente su indicación atrasará, intentando converger a un Sesgo teórico nulo en la próxima obtención del UTC. Normalmente, de un modo aleatorio de múltiples causas, la llegada de la nueva lectura traerá aparejado un nuevo Sesgo y su signo y módulo, determinará un nuevo cálculo del ajuste derivativo a aplicar al nuevo ciclo y así sucesivamente.

Si el ajuste derivativo no lograra su objetivo de convergencia, ante la existencia de un reloj deficiente que acumule un corrimiento creciente entre las sucesivas lecturas de NTP, se podrán hacer lecturas más frecuentes para mejorar el ajuste derivativo. Si este procedimiento no diera resultado, se podrá recurrir a otro método de obtención del UTC, como puede ser el agregado al hardware de un GPS con Emisión de Flanco **pps** (pulse per second), como se verá luego en el ensayo que se ha planteado.

En conclusión, con lo expuesto en este apartado, se logran buenas aproximaciones respecto al UTC, para la sincronización externa del orden de 10ms, con lectura desde internet con NTP y de pocos microsegundos con GPS con Emisión de Flanco. El valor de tiempo obtenido en el CoMaster, se puede emitir por broadcast con correcciones, a los otros IEDs de la red interna [3][10].

IV. ESCENARIO DE PRUEBA

IV.1. CONSIGNAS PRELIMINARES

Se describen aquí las condiciones del ensayo al que se sometió a un IED, para evaluación de desempeño en eventos digitales. Como se ha consignado anteriormente en Características (ítem 4), el software puede registrar Eventos Rápidos y Lentos. La situación más crítica que es la de registro de eventos rápidos, se presenta a menudo dentro de los problemas de ingeniería a resolver con microcontroladores. Esta problemática particular, se encuentra en campos diversos como Control, Robótica, Mediciones, etc. Se describe a continuación un escenario posible de medición en el campo de

la Ingeniería eléctrica, que sirve como marco de referencia para la experimentación efectuada.

Como es conocido, el disturbio que se genera ante una falla eléctrica, viaja a la velocidad de propagación electromagnética en el medio -en este caso el cable empleado para la conducción de la energía eléctrica en la línea -, llegando dicho frente en diferentes tiempos a ambos extremos de la misma. La detección y medición del tiempo de llegada de ambos eventos, en forma independiente por IEDs situados en cada extremo y sincronizados por UTC entre sí, con precisión externa de pocos microsegundos, permite definir, con errores menores a un kilómetro, la posición del punto de falla en la línea. Ello redundará en buena velocidad de reparación y reposición del servicio, con disminución del costo de falla, al disminuir el monto de las multas y demoras generalmente causantes de dificultades diversas. Esta es una síntesis muy simplificada de un desafío donde se requiere elevada precisión de registro temporal. Sobre este punto se volverá en las conclusiones, como un interesante desafío de resolución con IEDs en red.

Se eligió este escenario dado que los ambientes eléctricos - especialmente los de generación, transporte y distribución-, son de alta exigencia por el tipo, cantidad y criticidad de los eventos que ocurren en modo aleatorio. A la vez son relativamente simulables y predecibles - obviamente, no en cuanto al momento de ocurrencia -, como medio de prueba para desarrollos. La normativa emitida por el ente regulador argentino (ENRE), tiene a su vez varias exigencias, que se traducen en la necesidad de precisión en los desarrollos para cumplir con las exigencias del mercado y de la compañía Administradora del Sistema Interconectado Nacional (SIN). Esa exigencias se traducen por ejemplo, en la necesidad de registrar eventos con precisión de milisegundo, determinando que la sincronización de los distintos procesadores que conforman un gran sistema distribuido, también tengan una sincronización externa del mismo orden y que los eventos, lleguen a la compañía administradora con una antigüedad no mayor a 10 segundos, por medios de comunicaciones redundantes, de distintas tecnologías y. El objetivo central de tales requerimientos es el de contar con

información global, fidedigna del sistema para determinar causas y efectos en los caso de falla y determinar económicamente al causante de la falla para la aplicación de severas multas. Por un lado, se efectúan diagnósticos que permiten encontrar vulnerabilidades de la red y de esa forma destinar la inversión. Por otro lado, se produce un sistema de premios y castigos para los actores, componiendo así un incentivo económico para fortalecer las redes y mejorar los sistemas de protección de las mismas. Este ambiente de profusas variables es un excelente medio de prueba para desarrollos y la envergadura de los mismos puede ir de uno a múltiples procesadores interconectados en un sistema distribuido [1]. En esta situación son necesarias distintas capas físicas para interconexión, distintos protocolos [1][2] para implementar TCP/IP que es casi en su totalidad el Stack utilizado con sus diversos servicios (SNTP, HTTPS, MQTT, etc) y múltiples aplicaciones para distintos objetivos (medición comercial y de operación, estados, eventos, control de interruptores y seccionadores, soporte de protecciones, etc.). Quien esto escribe se ha desempeñado profesionalmente durante muchos años como responsable del servicio.

IV.2. ESCENARIO - ALCANCE

Para la prueba del sistema descrito hasta aquí, se ajustó el alcance de la experimentación estableciendo las siguientes consignas:

- Se cablearon las salidas de un IED ESP32 con Id:01, en forma directa (back to back), hacia sus homónimas de entrada en otro ESP32 con Id: 05 como se aprecia en la **Figura 4a** y en el circuito de la **Figura 4b**.

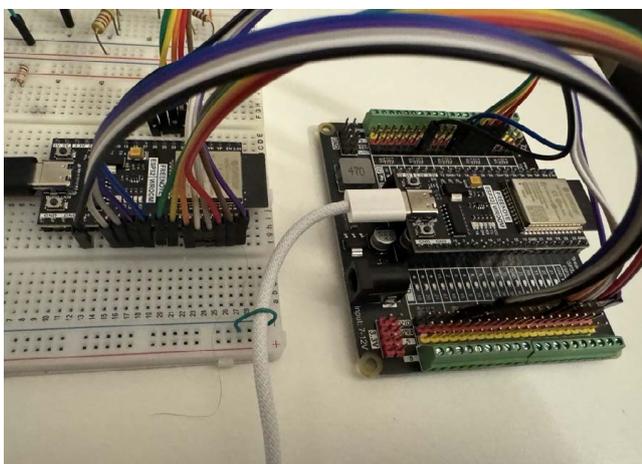


Figura 4a: Escenario de estudio.

- El IED 01 sincroniza su reloj por NTP desde los servidores públicos en internet. Se espera que en esta condición pueda llegar a existir un Sesgo Temporal con el UTC, que ronda en 10 ms. De cualquier manera, ello no es importante porque queda fuera del alcance de este estudio y no afecta la precisión que puede lograrse en la red interna.

- En el mismo procesador, se implementó una rutina de Sincronización de hora, emulando un GPS con Emisión de Flanco pps. El IED se toma como estrato de referencia de la hora actual sincronizada por NTP y el dato se emite por puerto serie 500 ms antes del inicio de cada segundo. La emisión del paquete de PeH puede observarse en la última fila del Dump presentado en la Figura 3, fila que se repite:

**** PeH-FL: FE 11 FF 40 02 12 18 08 12 12 09 38
88 13 07 3E 37 – Time: 2024-8-18 18:09:56.5000**

Obsérvese que la hora se emite por timer interno en 5000 dms desde inicio del segundo anterior. En el pasaje por cero del segundo asociado siguiente, se emite un flanco digital. Estas señales son ingresadas al IED Id:05 por las puertas adecuadas (Rx1 y DI), para ser aplicadas a la rutina de PeH. De esta forma el IED 05 toma como estrato superior al IED 01 y sincroniza su hora con él, con una precisión del orden de microsegundos, necesaria para el alcance dado a esta experimentación.

- En el IED 01, también se ha programado una rutina que permite generar ráfagas de eventos. En este caso, mediante un comando (com17) por la app en MQTT, de generación de eventos. Implementado ad hoc, permite generar ráfagas sobre 16 puertos de salida distintos, elegidos del grupo GPIO del IED 01.

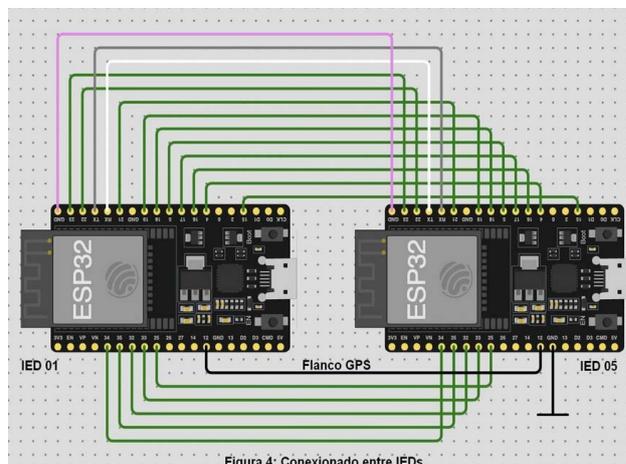


Figura 4b: Conexionado entre IEDs.



- La correspondencia entre Nro. de evento y Nro. de GPIO y además la **dirección indirecta**, de Puerto y Número de Bit, se indica en detalle en la tabla de la **Figura 5b**, columnas 1 a 4 para el IED 01 y columnas 1, 2, 7 y 8 para el IED 05.

- Los eventos de la ráfaga se emiten en un modo cuasi simultáneo, en un ciclo FOR del software, de modo que la diferencia temporal de emisión entre si para cualquier evento, está por debajo de 1 microsegundo, por lo que a los efectos de este ensayo se los considera simultáneos. Luego de emitida la primera ráfaga de 16 eventos, se hace una pausa de 2 ms y se la repite nuevamente para luego detener la emisión. O sea, se pueden emitir en cualquier momento, dos ráfagas consecutivas de 16 eventos distintos cada una, con una diferencia temporal entre ellas de 2 ms.

- En el mismo IED 01, estas direcciones se han declarado en la tabla de configuración del software para Eventos Rápidos, con lo cual los eventos que se emitan, se registrarán por loop interno también como eventos de entrada en el propio Bufer de Eventos del IED 01. Ello se hace para tener precisión del TimeStamp de emisión, de cada uno de un total de 32 eventos (2 repeticiones de 16), numerados en la columna 1 de la Tabla de Figura 5b. Estas ráfagas emitidas por el IED 01, ingresan a los puertos de entrada homónimos de la GPIO del IED 05.

- En el IED 05 se registrarán normalmente los eventos digitales de entrada y se almacenarán en su Bufer. Hay un detalle a observar en las columna 7 y 8 del IED 05 donde se ha declarado: Las primeras 13 posiciones correspondientes al el Puerto 0 de **eventos rápidos** y las 3 restantes se corresponden con las 3 primeras posiciones del Puerto 1 de **eventos lentos**. Esto se hizo así para tener un elemento de juicio en la consistencia en la información recabada, como se verá luego.

IV.3. ANÁLISIS DE RESULTADOS

El ensayo se efectuó varias veces, se ajustaron errores de implementación y cuando se obtuvieron ensayos consistentes, se registraron listados de los cuales se tomó uno de ellos, que es el volcado en la tabla de la **Figura 5a**. Ambos microcontroladores cuentan con el software ds7.1 y los resultados se clarifican en correspondencia con los Nros de evento en la planilla de la Figura 5b, que pasamos a analizar.

1. La columna 1 otorga en número del evento secuencial.
2. La columna 2 entrega la dirección física de la GPIO, por comodidad fue la misma utilizada en ambos procesadores.
3. Las Columnas 3, 4, 7 y 8 presentan la dirección indirecta de los puertos de 16 bits (**Puerto y Nro. de Bit** respectivamente) de ambos procesadores. El puerto 0 es de eventos rápidos, el puerto 1 es de eventos lentos.
4. La columna 5 (en Hexadecimal) otorga datos de identificación del evento de **salida del IED 01** (2 primeros bytes) y su TimeStamp (8 bytes restantes).
5. La columna 6 (en Hexadecimal) otorga datos de identificación del evento de **entrada del IED 05** (2 primeros bytes) y su TimeStamp (8 bytes restantes).
6. Cada registro de eventos de las columnas 5 y 6, posee 10 bytes, donde:
 1. Byte1: otorga TIPO (0 para eventos de campo) y dirección del IED (01 del emisor de eventos y 05 del receptor).
 2. Byte 2: Otorga el estado (MSB, 1 bit), el puerto (3 bits inferiores del nibble High) y el NroBit (nibble Low).
 3. Bytes 3, 4 y 5: YY MM DD (año, mes y día).

Message arrived: uCNet4Mqtt/Com/05/17.20.

COMANDO default BufApli: 17 20

Resp Comando MQTT. Canal: 3. CBufApli[Canal]: 3

```
*** Eventos. PEvIngreso: 8. PEvOut: 0. BufEventos: 05 80 18 08 12 11 2E 0A 38 0C
*** Eventos. PEvIngreso: 9. PEvOut: 0. BufEventos: 05 81 18 08 12 11 2E 0A 38 0C
*** Eventos. PEvIngreso: 10. PEvOut: 0. BufEventos: 05 82 18 08 12 11 2E 0A 39 0C
*** Eventos. PEvIngreso: 11. PEvOut: 0. BufEventos: 05 83 18 08 12 11 2E 0A 38 0C
*** Eventos. PEvIngreso: 12. PEvOut: 0. BufEventos: 05 84 18 08 12 11 2E 0A 39 0C
*** Eventos. PEvIngreso: 13. PEvOut: 0. BufEventos: 05 85 18 08 12 11 2E 0A 39 0C
*** Eventos. PEvIngreso: 14. PEvOut: 0. BufEventos: 05 86 18 08 12 11 2E 0A 38 0C
*** Eventos. PEvIngreso: 15. PEvOut: 0. BufEventos: 05 87 18 08 12 11 2E 0A 39 0C
*** Eventos. PEvIngreso: 16. PEvOut: 0. BufEventos: 05 88 18 08 12 11 2E 0A 38 0C
*** Eventos. PEvIngreso: 17. PEvOut: 0. BufEventos: 05 89 18 08 12 11 2E 0A 38 0C
*** Eventos. PEvIngreso: 18. PEvOut: 0. BufEventos: 05 8A 18 08 12 11 2E 0A 39 0C
*** Eventos. PEvIngreso: 19. PEvOut: 0. BufEventos: 05 8B 18 08 12 11 2E 0A 39 0C
*** Eventos. PEvIngreso: 20. PEvOut: 0. BufEventos: 05 8C 18 08 12 11 2E 0A 38 0C
*** Eventos. PEvIngreso: 21. PEvOut: 0. BufEventos: 05 90 18 08 12 11 2E 0A 80 0C
*** Eventos. PEvIngreso: 22. PEvOut: 0. BufEventos: 05 91 18 08 12 11 2E 0A 80 0C
*** Eventos. PEvIngreso: 23. PEvOut: 0. BufEventos: 05 92 18 08 12 11 2E 0A 80 0C
*** Eventos. PEvIngreso: 24. PEvOut: 0. BufEventos: 05 00 18 08 12 11 2E 0A 4C 0C
*** Eventos. PEvIngreso: 25. PEvOut: 0. BufEventos: 05 01 18 08 12 11 2E 0A 4C 0C
*** Eventos. PEvIngreso: 26. PEvOut: 0. BufEventos: 05 02 18 08 12 11 2E 0A 4D 0C
*** Eventos. PEvIngreso: 27. PEvOut: 0. BufEventos: 05 03 18 08 12 11 2E 0A 4C 0C
*** Eventos. PEvIngreso: 28. PEvOut: 0. BufEventos: 05 04 18 08 12 11 2E 0A 4D 0C
*** Eventos. PEvIngreso: 29. PEvOut: 0. BufEventos: 05 05 18 08 12 11 2E 0A 4D 0C
*** Eventos. PEvIngreso: 30. PEvOut: 0. BufEventos: 05 06 18 08 12 11 2E 0A 4C 0C
*** Eventos. PEvIngreso: 31. PEvOut: 0. BufEventos: 05 07 18 08 12 11 2E 0A 4D 0C
*** Eventos. PEvIngreso: 32. PEvOut: 0. BufEventos: 05 08 18 08 12 11 2E 0A 4C 0C
*** Eventos. PEvIngreso: 33. PEvOut: 0. BufEventos: 05 09 18 08 12 11 2E 0A 4C 0C
*** Eventos. PEvIngreso: 34. PEvOut: 0. BufEventos: 05 0A 18 08 12 11 2E 0A 4D 0C
*** Eventos. PEvIngreso: 35. PEvOut: 0. BufEventos: 05 0B 18 08 12 11 2E 0A 4D 0C
*** Eventos. PEvIngreso: 36. PEvOut: 0. BufEventos: 05 0C 18 08 12 11 2E 0A 4C 0C
*** Eventos. PEvIngreso: 37. PEvOut: 0. BufEventos: 05 10 18 08 12 11 2E 0A 80 0C
*** Eventos. PEvIngreso: 38. PEvOut: 0. BufEventos: 05 11 18 08 12 11 2E 0A 80 0C
*** Eventos. PEvIngreso: 39. PEvOut: 0. BufEventos: 05 12 18 08 12 11 2E 0A 80 0C
```

Figura 5a: IED 05. Segmento del Bufer de eventos.

4. Bytes 6, 7 y 8: HH mm SS (hora, minuto, segundo).
5. Bytes 9 y 10: Conforman un WORD en formato Little Endian. Contiene fracción de segundo, en dms ($s/10^{-4}$).
7. Para interpretar un evento tomamos como ejemplo el **evento 13**, de la **columna 5** (en negrita): **05 8C 18 08 12 11 2E 0A 38 0C**. La capa de Aplicación del protocolo 1-v6.1, tiene la siguiente semántica por cada byte:
 1. 05 - **TIPO**: 0 (0 para eventos de campo) y **Id** del IED: 5 (del receptor de eventos).
 2. 8C - **ESTADO**: 1 (MSB, 1 bit), **PUERTO**: 0 (3 bits inferiores del nibble High) y el **NroBit**: 12 (nibble Low).
 3. **18 08 12** (Bytes 3, 4 y 5) - **YY MM DD** (año, mes y día): 24/08/18.
 4. **11 2E 0A** (Bytes 6, 7 y 8) - **HH mm SS** (hora, minuto, segundo): 16:46:10.
 5. **38 0C** (Bytes 9 y 10) - Conforman un WORD en formato Little Endian que contiene la fracción de segundo, en dms ($s/10^{-4}$): $0x0C38 = 3128$. Luego, el **TimeStamp** del evento de la línea es: 2024/08/18 17:46:10.3128
8. Los eventos emitidos por el IED 01, se sitúan en estado 1 para los 16 pertenecientes a la primera ráfaga y en estado 0 para los 16 siguientes, completando un total de 32.

IV.4. CONSIDERACIONES

De la comparativa de los eventos registrados surge lo siguiente:

- A) El TimeStamp de la segunda ráfaga es mayor en 20 dms que lo de la primera. Ello era esperable dado que en los ensayos de Puesta en Hora y ajustes de la temporización, donde se había utilizado un Timer Hardware, la hora era muy precisa y con la monotonía esperable. Ello, salvando las diferencias aleatorias que pueden existir respecto a un patrón de hora superior, debido al *drift* del cristal del procesador local, por sus condiciones de temperatura y tensión de alimentación en funcionamiento.
- B) Los eventos recibidos por el IED 05 tienen particularidades que se destacan:
 1. El TimeStamp de cada evento recibido es igual al emitido, en casi todos los casos. Esto también es esperable

		UC 01				UC 05				
1	2	3	4	5	6	7	8			
Ev	GPIO	Puerto	NroBit	Uc 01 (Output)	Uc 05 (Input)	Puerto	NroBit	Color		
1	26	0	0	01 80 18 08 12 11 2E 0A 38 0C	05 80 18 08 12 11 2E 0A 38 0C	0	0	AZ		
2	23	0	1	01 81 18 08 12 11 2E 0A 38 0C	05 81 18 08 12 11 2E 0A 38 0C	0	1	MA		
3	22	0	2	01 82 18 08 12 11 2E 0A 38 0C	05 82 18 08 12 11 2E 0A 39 0C	0	2	RO		
4	21	0	3	01 83 18 08 12 11 2E 0A 38 0C	05 83 18 08 12 11 2E 0A 38 0C	0	3	NA		
5	19	0	4	01 84 18 08 12 11 2E 0A 38 0C	05 84 18 08 12 11 2E 0A 39 0C	0	4	AM		
6	18	0	5	01 85 18 08 12 11 2E 0A 38 0C	05 85 18 08 12 11 2E 0A 39 0C	0	5	VE		
7	5	0	6	01 86 18 08 12 11 2E 0A 38 0C	05 86 18 08 12 11 2E 0A 38 0C	0	6	AZ		
8	17	0	7	01 87 18 08 12 11 2E 0A 38 0C	05 87 18 08 12 11 2E 0A 39 0C	0	7	VI		
9	16	0	8	01 88 18 08 12 11 2E 0A 38 0C	05 88 18 08 12 11 2E 0A 38 0C	0	8	GR		
10	4	0	9	01 89 18 08 12 11 2E 0A 38 0C	05 89 18 08 12 11 2E 0A 38 0C	0	9	BL		
11	15	0	A	01 8A 18 08 12 11 2E 0A 38 0C	05 8A 18 08 12 11 2E 0A 39 0C	0	A	NE		
12	34	0	B	01 8B 18 08 12 11 2E 0A 38 0C	05 8B 18 08 12 11 2E 0A 39 0C	0	B	MA		
13	35	0	C	01 8C 18 08 12 11 2E 0A 38 0C	05 8C 18 08 12 11 2E 0A 38 0C	0	C	RO		
14	32	0	D	01 90 18 08 12 11 2E 0A 38 0C	05 90 18 08 12 11 2E 0A 80 0C	1	0	NA		
15	33	0	E	01 91 18 08 12 11 2E 0A 38 0C	05 91 18 08 12 11 2E 0A 80 0C	1	1	AM		
16	25	0	F	01 92 18 08 12 11 2E 0A 38 0C	05 92 18 08 12 11 2E 0A 80 0C	1	2	VE		
17	-	0	0	01 00 18 08 12 11 2E 0A 4C 0C	05 00 18 08 12 11 2E 0A 4C 0C	0	0	-		
18	-	0	1	01 01 18 08 12 11 2E 0A 4C 0C	05 01 18 08 12 11 2E 0A 4C 0C	0	1	-		
19	-	0	2	01 02 18 08 12 11 2E 0A 4C 0C	05 02 18 08 12 11 2E 0A 4D 0C	0	2	-		
20	-	0	3	01 03 18 08 12 11 2E 0A 4C 0C	05 03 18 08 12 11 2E 0A 4C 0C	0	3	-		
21	-	0	4	01 04 18 08 12 11 2E 0A 4C 0C	05 04 18 08 12 11 2E 0A 4D 0C	0	4	-		
22	-	0	5	01 05 18 08 12 11 2E 0A 4C 0C	05 05 18 08 12 11 2E 0A 4D 0C	0	5	-		
23	-	0	6	01 06 18 08 12 11 2E 0A 4C 0C	05 06 18 08 12 11 2E 0A 4C 0C	0	6	-		
24	-	0	7	01 07 18 08 12 11 2E 0A 4C 0C	05 07 18 08 12 11 2E 0A 4D 0C	0	7	-		
25	-	0	8	01 08 18 08 12 11 2E 0A 4C 0C	05 08 18 08 12 11 2E 0A 4C 0C	0	8	-		
26	-	0	9	01 09 18 08 12 11 2E 0A 4C 0C	05 09 18 08 12 11 2E 0A 4C 0C	0	9	-		
27	-	0	A	01 0A 18 08 12 11 2E 0A 4C 0C	05 0A 18 08 12 11 2E 0A 4D 0C	0	A	-		
28	-	0	B	01 0B 18 08 12 11 2E 0A 4C 0C	05 0B 18 08 12 11 2E 0A 4D 0C	0	B	-		
29	-	0	C	01 0C 18 08 12 11 2E 0A 4C 0C	05 0C 18 08 12 11 2E 0A 4C 0C	0	C	-		
30	-	0	D	01 10 18 08 12 11 2E 0A 4C 0C	05 10 18 08 12 11 2E 0A 80 0C	1	0	-		
31	-	0	E	01 11 18 08 12 11 2E 0A 4C 0C	05 11 18 08 12 11 2E 0A 80 0C	1	1	-		
32	-	0	F	01 12 18 08 12 11 2E 0A 4C 0C	05 12 18 08 12 11 2E 0A 80 0C	1	2	-		
-	1	TX	-			RX	-	BL		
-	3	RX	-			TX	-	GR		
-	12	FLout	-			FLin	-	NE		
-	-	GND	-			GND	-	VI		

del

Registro de eventos

Figura 5b: Tabla de el análisis



por la elevada precisión que posee el GPS de Emisión de flanco que permite sincronizar ambos procesadores con precisiones de pocos microsegundos.

2. Algunos Eventos Rápidos difieren en 1 dms en exceso, de otros que han sido coincidentes en el instante de emisión. La dispersión no es importante pero es posible imputar el corrimiento de estos eventos a una demora en la gestión de las interrupciones que efectuó el RTOS. Dado que la rutina de servicio de interrupción (ISR) es la misma para todos los eventos registrados, variando sólo el parámetro de su dirección hardware en el llamado y aunque en tiempo de ejecución, la rutina puede hacerse residir en la Internal Ram mediante configuración, para mayor velocidad de ejecución, al SO no le queda más alternativa que las distintas instancias sean ejecutadas en secuencia por cada evento, por aquellos procesadores que estén disponibles aleatoriamente para tal ejecución. Este encolamiento en la ejecución puede ser causa de diferimiento en los tiempos de registro.
3. Las elevadas diferencias que se encuentran en los registros de eventos Lentos 14, 15 y 16 de la primera ráfaga y 30, 31 y 32 de la segunda ráfaga, respecto a su tiempo de emisión y que además tienen el mismo TimeStamp en el registro, siendo que pertenecen a ráfagas distintas, se explica porque aquellos se han configurado deliberadamente para su detección por Polling. Ello se efectúa dentro de la rutina de ejecución sistemática de Toma de Estado, cada 10 ms. Esto se ha determinado así para verificar consistencia en la distinta modalidad de registro entre eventos Lentos y Rápidos. Debido a que la planificación los timers por software dispuestos en la aplicación como método de temporización de tareas, determinan que la rutina de Toma de Estado se ejecute aproximadamente en el instante que los milisegundos son múltiplos de 10 y en cero, se ha dispuesto que el proceso trunque los dígitos decimales por debajo de 10 ms. Esto es así puesto que la granularidad está determinada por el

período de la Toma de Estado y la presencia de dígitos decimales por debajo de 10 ms darían una idea equivocada de la precisión. Se observa además que el tiempo de registro de esos eventos, resulta con un TimeStamp en exceso y es igual para todos los eventos Lentos captados en el mismo ciclo. Ello es esperable porque son registrados simultáneamente. En efecto, aunque hay eventos que pertenecen a distintas ráfagas, llegando al IED 05 en diferente tiempo, ellos son recogidos con demora, en el mismo ciclo de toma de estado y se les impone el idéntico TimeStamp.

V. CONCLUSIONES

El desempeño de RTOS fue altamente satisfactorio. El desglose de las tareas con tiempos de ejecución definidos, como las interrupciones asociadas a la rutina del Reloj local y la de ingreso de eventos, se ejecutaron con sorprendente precisión, aún teniendo en cuenta que hay otras tareas temporizadas, de menor prioridad como la encuesta de la red en diferentes protocolos, el proceso local, la Toma de Estado, etc.

El funcionamiento de la red y reporte de todos los IEDs incluidos en ella se efectuó tal cual lo previsto, sin novedades que destacar, merituando aún más el funcionamiento del sistema operativo.

En el transcurso de los ensayos, se observaron algunos reinicios de la aplicación por activación del Watchdog asociado a alguna de las tareas del esquema Multitask/Multicore. Si bien se habían previsto en éstas los controles correspondientes a Watchdogs y se incluyeron retardos, que ponen en espera a los procesos de mayor prioridad, para ceder *Quantum* de ejecución a otros de menor prioridad, ha habido situaciones donde se evidencia la necesidad de un estudio más profundo de las características y configuración de RTOS para que lo indicado no suceda. Se considera que es más bien un problema de configuración o de incorrecta programación en algún punto, que una falla inherente al SO.

El desempeño del registro de eventos por su parte, ha sido altamente satisfactorio como lo evidencian las pruebas. Queda pendiente el desafío para casos de una cantidad mayor de eventos, donde podrían producirse similares



diferimientos a los ya indicados, que podrían ser inaceptables. Por su parte, el estudio más profundo del caso que se presenta con el geoposicionamiento de fallas eléctricas descrito más arriba, no se intentó abordarlo en esta etapa debido a que ello implicaría modificaciones importantes al software y la necesidad de contar con GPS con Emisión de Flanco, para determinar la incidencia de la sincronización Externa en el resultado.

A la luz de los resultados globales del proyecto, evaluados durante su desarrollo y las pruebas y en base a las conclusiones alcanzadas, se proponen algunas acciones para el futuro inmediato, tales como:

- Realizar un análisis más profundo y ensayos adicionales sobre la funcionalidad del RTOS, especialmente en lo que respecta a la separación de tareas para múltiples procesadores, dado su desempeño destacado en el trabajo realizado.

- Iniciar una línea de investigación sobre otras necesidades presentes en entornos de Misión Crítica, como los sectores Eléctrico, Químico e Industrial, que podrían beneficiarse de soluciones basadas en microcontroladores de la línea investigada.

- Evaluar la integración de algoritmos para el análisis predictivo y la detección temprana de fallas, mejorando la capacidad del sistema para anticipar y responder a eventos críticos.

Estas acciones podrían fortalecer y expandir los alcances del proyecto, permitiendo abordar nuevos desafíos y aprovechar oportunidades en la aplicación de microcontroladores en entornos críticos y de automatización industrial.

REFERENCIAS

- [1] Coulouris G., Dollimore J., Kindberg T., Bair G. "Distributed Systems, Concepts and Design". Fifth Edition. Addison-Wesley. 2012.
- [2] Tanenbaum A., VanSteen V. "Distributed Systems". Second Edition. Prentice Hall. 2002.
- [3] Cristian F., Fetzer C. "The Timed Asynchronous Distributed System Model" IEEE Transactions on Parallel and Distributed Systems, Vol 10, Nro 6 June 1999.
- [4] Fernando G. Tinetti, Ricardo A. López. "Redes de Microcontroladores: Definición, Evaluación y Perspectivas de un Sistema Distribuido". X Workshop de Investigadores en Ciencias de la Computación 2008, Fac. de Ingeniería, Universidad Nacional de La Pampa, Gral. Pico, La Pampa, Argentina, Mayo 5-6 de 2008, pp. 204 – 208.
- [5] Ricardo A. López. "Protocolos en Redes de Microcontroladores." Tesis de Magister: Redes de datos. Director: Fernando F. Tinetti. 1ra Edición - Septiembre 2010. Publicó: Fac. Informática UN La Plata. Volumen 144 hojas.
- [6] Ricardo A. López. Nahuel Defossé. "Control de Tránsito con Redes de Microcontroladores". II Congreso Virtual de Microcontroladores y sus Aplicaciones. Título: ISBN: 978-987-25855-5-6. 18 al 29 de Octubre 2010. Universidad Tecnológica Nacional. Entre Ríos. Argentina.
- [7] Ricardo A. López. Thesis Review: "Protocolos en Redes de Microcontroladores." (Protocols for Microcontrollers Networks) Special Issue on Research in Computer Science. Vol 11 - No. 1. pags 47-49, April 2011. ISSN 1666-6038. Fac. de Informática. Universidad Nal. de La Plata.
- [8] Ricardo A. López, Emilio Pincirolí, Fernando G. Tinetti. " Microcontroladores Asociados a Medición y Comunicaciones en Sistemas SCADA de Energía ". XVI Workshop de Investigadores en Ciencias de la Computación. Ushuaia, 7 y 8 de Mayo de 2014, Universidad de Tierra del Fuego.pp. 646-650.
- [9] Fernando G. Tinetti, Ricardo A. López, Sebastián Wahler. "Sincronización de Microcontroladores Interconectados: Evaluación de Factibilidad y Detalles de Implementación". X Workshop de Investigadores en Ciencias de la Computación 2008, Fac. de Ingeniería, Universidad Nacional de La Pampa, Gral. Pico, La Pampa, Argentina, Mayo 5-6 de 2008, pp. 209 – 213.
- [10] Fernando G. Tinetti, Ricardo A. López, Marcelo E. Gómez, Sebastián Wahler. "Sincronización de Microcontroladores en Red, Implementación y Evaluación". XV Congreso Argentino de Ciencias de la Computación 2009, Universidad Nacional de Jujuy, Jujuy, Argentina, Octubre 5 al 9 de 2009, ISBN 978 897 24068-4-1.
- [11] Ricardo A. López, Fernando G. Tinetti. " Sincronización Broadcast en Redes Multipunto". XVII Congreso Argentino de Ciencias de la Computación 2011, Universidad Nacional de La Plata, La Plata, Argentina, Octubre 10 al 14 de 2011, ISBN: 978-950-34-0756-1. JCS&T, Journal of Computer Science & Technology.
- [12] Marcelo E. Gómez, Sebastián P. Wahler, Fernando G. Tinetti, Ricardo A. López. " Implementación de Mensajes Rápidos y Valores de Muestreo IEC61850 sobre Ethernet con Microcontroladores". XVI Workshop de Investigadores en Ciencias de la Computación Título: Ushuaia, 7 y 8 de Mayo de 2014, Universidad de Tierra del Fuego.
- [13] Ricardo A. López. "Sistema de Semaforización Inteligente". Revista UNICA de la Asociación de Profesionales Universitarios del Agua y la Energía Eléctrica. (APUAYE) Vol. 107 pags 40-43, Agosto 2009.
- [14] Fernando G. Tinetti, Ricardo A. López, Nahuel Defossé. "Microcontroladores DSP y Aplicaciones WEB". XII Workshop de Investigadores en Ciencias de la Computación. 5 y 6 de Mayo de 2010, Calafate, Sta. Cruz. Argentina. pp. 646-650.
- [15] Nahuel Defossé, Diego Van Haaster, Lautaro Pecile, Fernando G. Tinetti, Ricardo A. López. " Implementación de Sistemas SCADA Utilizando Lenguajes de Alto Nivel ". XVI Workshop de Investigadores en Ciencias de la Computación. Ushuaia, 7 y 8 de Mayo de 2014, Universidad de Tierra del Fuego.
- [16] Defossé N., López R. A., Marcelo E. Gómez, Konstantinoff P., Wahler S., Castro L., Harris G. "Implementación de Middleware Publicador/Subscriber para Aplicaciones Web de Monitoreo". WICC 2017 XIX Workshop de Investigadores en Ciencias de la Computación 27 y 28 de Abril'17. Buenos Aires. Libro_WICC_2017 ISBN_978-987-42-5143-5 Pags 181 a 185.



Aplicaciones de Internet de las Cosas (IoT) para la conservación y gestión eficiente de la energía

Guillermo Friedrich

Universidad Tecnológica Nacional

Facultad Regional Bahía Blanca

Bahía Blanca, Argentina

gfried@frbb.utn.edu.ar

Guillermo Reggiani

Universidad Tecnológica Nacional

Facultad Regional Bahía Blanca

Bahía Blanca, Argentina

ghreggiani@frbb.utn.edu.ar

Resumen— El presente trabajo reúne los conceptos y propuestas iniciales, sobre los que se ha ido trabajando en el marco del Proyecto de I+D “Aplicaciones de Internet de las Cosas (IoT) para la conservación y gestión eficiente de la energía”. En base a los mismos se han desarrollado y ensayado algunas implementaciones a nivel de prototipo y se han elaborado algunos trabajos presentados en una jornada internacional [1], un trabajo en un congreso nacional [2] y una publicación en revista indexada internacional [3]. Asimismo, se encuentra en desarrollo una implementación experimental de MQTT sobre RS485, usando la red eléctrica como medio de transmisión mediante interfaces adecuadas, que es la base de una próxima publicación.

Palabras clave: IoT, RS485, MQTT, Power Line Communication.

I. INTRODUCCIÓN

Internet de las Cosas (IoT: Internet of Things) es un paradigma que ha venido teniendo un gran despliegue en diversas áreas, desde el ámbito doméstico hasta la industria y la salud, entre otras. El paradigma IoT resulta de interés porque facilita el diseño e implantación de aplicaciones que conectan dispositivos y objetos de uso cotidiano, haciendo posible su identificación y comunicación, con fines de monitoreo y/o control.

En los últimos años ha habido un creciente desarrollo de aplicaciones orientadas a la conservación de la energía, basadas en IoT y

computación en la nube (CC: cloud computing). Algunas de esas propuestas se orientan a la automatización de la gestión de energía, de acuerdo con modelos que intentan reducir consumos de energía y al mismo tiempo brindar bienestar a los habitantes de los denominados edificios inteligentes [4] y casas inteligentes [5]. Mientras tanto, otras propuestas se orientan a brindar realimentación a los usuarios acerca de sus consumos, a fin de que puedan administrarlos racionalmente. Hacia este último enfoque se orienta el presente trabajo.

Las facturas periódicas por los consumos de electricidad y gas también representan una forma de realimentación, pero no son tan efectivas para causar cambios en los patrones de consumo. Una adecuada realimentación puede ayudar a reducir los consumos entre un 5% y un 20% [6] si se siguen las siguientes premisas: entregada regularmente, con información presentada de manera clara y atractiva, preferentemente con detalle de consumo por artefacto, asociada con algunas metas de conservación de energía.

Los dispositivos IoT por lo general tienen una pequeña capacidad de procesamiento. CC tiene una capacidad ilimitada virtual de almacenamiento y procesamiento. La conjunción de IoT y CC permite el desarrollo de soluciones complejas, con facilidad de acceso para los usuarios mediante interfaces web y/o aplicaciones móviles. Las principales empresas que ofrecen servicios en la nube, tales como Amazon, Google y Microsoft, están orientadas



al mundo de IoT [7]; también Mathworks, que permite combinar sus herramientas analíticas con IoT.

Uno de los protocolos de comunicación ampliamente utilizado en el campo de IoT es Message Queue Telemetry Transport (MQTT) [8]. MQTT es un protocolo de la capa de aplicación, basado en un paradigma conocido como publicador-subscriptor, que permite que los clientes puedan enviar publicaciones de determinados tópicos hacia un broker, como también suscribirse a determinados tópicos, para recibir actualizaciones de los mismos cada vez que se produzcan. El protocolo de transporte habitualmente utilizado por debajo de MQTT es TCP. Por otra parte, los protocolos y tecnologías habitualmente utilizados en las capas inferiores de IoT son inalámbricos: WiFi, IEEE 802.15.4, Bluetooth; también cabe mencionar a LTE y LTE-A en el caso de las redes de telefonía celular [9].

Teniendo en cuenta que por lo general los dispositivos IoT tienen limitada capacidad de memoria y procesamiento, y es conveniente que tengan un reducido consumo de energía, este trabajo está orientado a tratar de aprovechar un estándar clásico de las comunicaciones por cable, como es RS485, para permitir la conexión de dispositivos sencillos a una red basada en MQTT. De esta manera el dispositivo IoT podría estar basado en un microcontrolador de baja gama, requiriendo sólo disponer de una interfaz serial asincrónica (UART) para la comunicación.

RS485 brinda un medio de comunicación semi dúplex, de baja velocidad y multipunto. Conecta múltiples nodos en cascada, típicamente sobre un par trenzado, usando señalización diferencial y alcanzando distancias de hasta 1 km. Las tasas de transmisión típicas están entre 9.6 kbps y 115.2 kbps.

Asimismo, una posibilidad adicional que se plantea, para evitar la necesidad de realizar un tendido de cable adicional, es utilizar interfaces adecuadas para transmitir RS485 sobre la red eléctrica. En este sentido se analiza una posible interfaz basada en el circuito integrado THVD8010 [10].

Dos trabajos relacionados en esta temática son el [11] y el [12]. El primero presenta una adaptación de 6LoWPAN para usar RS485 como capa física, a fin de posibilitar la conexión

de dispositivos de baja capacidad a una red IPv6, usando un esquema de tipo CSMA para acceder al medio compartido, de manera análoga a como lo haría IEEE 802.15.4 para acceder al medio inalámbrico. El segundo presenta una implementación de IP sobre RS485, que utiliza un esquema maestro/esclavo para regular el acceso al medio, bajo el control de un nodo (denominado IP Router). Cada nodo conectado a RS485 tiene una dirección IP y un identificador de nodo (que hace las veces de dirección MAC), por lo que el IP Router debe tener ambas asociadas para poder enviar paquetes hacia un determinado nodo del segmento RS485.

La propuesta actual tiene semejanzas con los trabajos anteriores, en particular con el segundo, en cuanto al esquema maestro/esclavo para controlar el acceso a la red RS485. Pero una diferencia significativa, que lo hace más sencillo y adecuado para dispositivos simples, es que no requiere de usar direcciones IP sobre RS485, sino que los mensajes se identifican por los tópicos MQTT. Esto implica menores tamaños de trama y menor requerimiento de procesamiento, tanto en los nodos RS485 como en el gateway que los vincula con el resto de la red.

A. *El protocolo MQTT*

A diferencia del modelo de la mayoría de los protocolos usados en Internet, basados en el paradigma cliente-servidor, el protocolo MQTT está basado en un paradigma Publicador-Suscriptor. El mismo posibilita que clientes interesados en determinados tópicos de información se suscriban a los mismos en un servidor denominado bróker. Asimismo, los clientes pueden publicar determinados tópicos y luego el bróker los reenvía a los suscriptores correspondientes. En la Fig. 1 se presenta un resumen del modelo operativo de MQTT, en la que se observan los distintos tipos de mensajes, como también que es posible que un nodo sea al mismo tiempo publicador y suscriptor

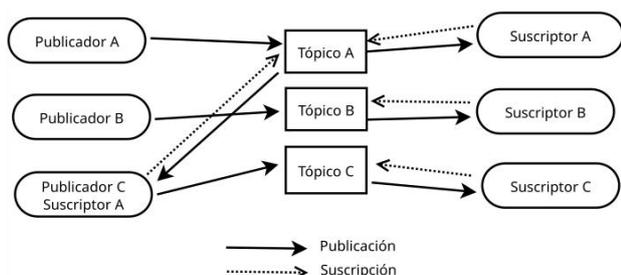


Fig. 1. Esquema básico del modelo publicador-suscriptor de MQTT

Cabe señalar que el protocolo MQTT tiene previstos tres tipos de servicio: QoS 0, QoS 1 y QoS 2. En QoS 0 cada publicación se recibe a lo sumo una vez; en QoS 1 cada publicación se recibe al menos una vez y en QoS 2 cada publicación se recibe exactamente una vez.

La presente propuesta contempla la posibilidad de operar con QoS 0 o QoS 1.

II. DESCRIPCIÓN GENERAL DEL SISTEMA PROPUESTO

El objetivo del sistema propuesto es la implementación de una red de dispositivos basada en el uso del protocolo MQTT, usando RS485 en la capa física, para posibilitar la integración de dispositivos de bajos recursos. En la Fig. 2 se presenta un esquema general de la red, donde se distinguen los dos elementos principales: los nodos RS485 y el gateway MQTT-RS485, cuyas funciones se describen seguidamente.

El gateway actúa hacia el broker MQTT como representante de los nodos RS485 que tiene bajo su control. Para ello debe tener una interfaz adecuada para conectarse a una red TCP/IP, que típicamente podrá ser Ethernet o WiFi. Por otra parte, debe tener una interfaz UART para conectar hacia la red RS485. Un criterio que se estableció para el gateway es que debe ser lo más simple y económico posible, en términos de memoria y procesamiento.

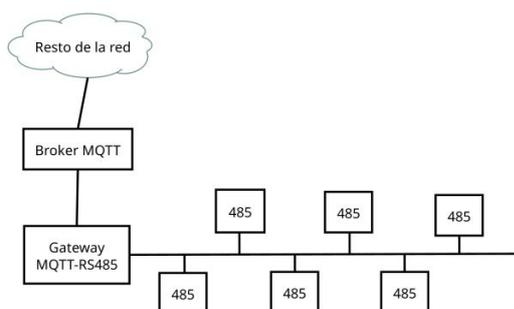


Fig. 2. Diagrama general de la red MQTT-RS485

Esto incluye no llevar un registro de las suscripciones efectuadas por cada nodo RS485.

Los nodos RS485 pueden enviar mensajes de suscripción a determinados tópicos y/o enviar publicaciones de distintos tópicos. Cuando el gateway recibe un pedido de suscripción de un nodo RS485 lo reenvía al broker, siendo el gateway el host que queda referenciado en el broker como suscriptor. Luego, cuando el gateway recibe una publicación desde el broker, la reenvía en broadcast a todos los nodos RS485, para que los interesados la acepten y procesen, y el resto las descarte. En cuanto a las publicaciones enviadas por los nodos RS485, el gateway simplemente las reenvía al broker.

Las comunicaciones en el segmento RS485 se realizan mediante un esquema maestro/esclavo bajo el control del gateway. Los nodos no tienen una dirección IP individual, sino que la dirección IP de todo el conjunto es la del gateway. Los nodos se identifican mediante un identificador de 6 bits, que el gateway utiliza durante el proceso de polling y para el envío de mensajes hacia los mismos.

A. Pila de Protocolos

La pila de protocolos que implementan los nodos RS485 está compuesta de tres capas, como se observa en la Fig. 3.

Dentro de la capa MAC cabe analizar dos elementos: el formato de trama MQTT485 y el mecanismo de polling, los que se tratan en la siguiente sección.

También resulta de interés observar las pilas de protocolos en el conjunto formado por los tres tipos de nodo, según se observa en la Fig. 4. En la misma se observa la función de reenvío que realiza el gateway. Más adelante se analiza el subconjunto de mensajes del protocolo MQTT procesados por los nodos RS485 y el gateway.

Aplicación	MQTT
MAC	Polling (trama MQTT485)
Física	RS485

Fig. 3. Pila de protocolos en los nodos RS485

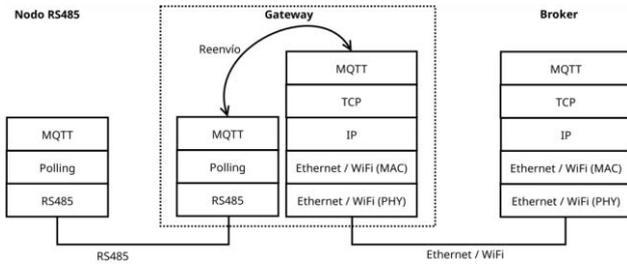


Fig. 4. Modelo de capas en la ruta entre nodo y bróker

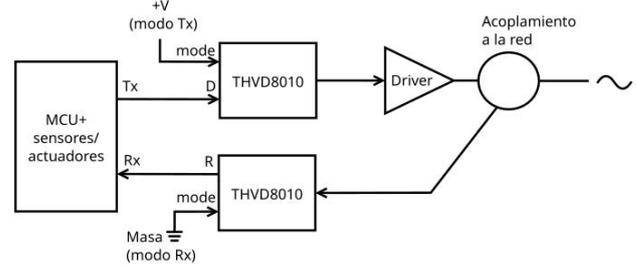


Fig. 6. Diagrama en bloques de la interfaz RS485 a la red eléctrica

B. Capa Física

La capa física de RS485 consiste en un bus de par trenzado al que se conectan los distintos nodos. La transmisión se efectúa en modo diferencial. Sólo un nodo a la vez puede transmitir; los transmisores deben estar por defecto en el estado de alta impedancia, del que salen solamente durante el intervalo de transmisión. El mecanismo de control de acceso al medio típico es del tipo maestro/esclavo. En el caso del sistema propuesto, el gateway será el maestro de la red RS485.

Como fue mencionado previamente, una variante de interés, que puede representar una ventaja significativa, es la conexión para transmisión de RS485 a través de la red eléctrica, mediante interfaces adecuadas. En tal sentido se evalúa la posibilidad de utilizar interfaces basadas en el transceiver THVD8000 [10] o THVD8010 [13], cuya implementación podría ser como la que se observa en la Fig. 5.

Tanto los nodos como el gateway se conectan a través de una UART y mediante los transceivers se conectan a la red eléctrica, evitando tener que disponer un cableado adicional para las comunicaciones mediante RS485.

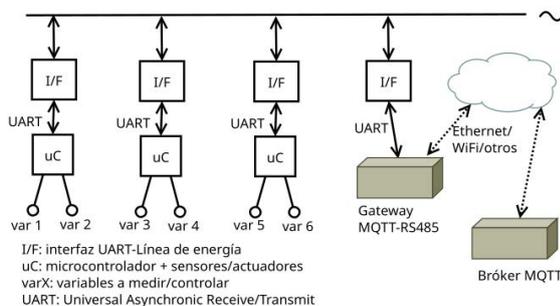


Fig. 5. Implementación de MQTT-RS485 sobre la red eléctrica

En la Fig. 6 se presenta el diagrama en bloques de un posible circuito de interfaz entre la UART del microcontrolador y la red eléctrica, en el que se observan los capacitores de desacople y la resistencia de terminación, cuyos valores se calcularon a partir de las frecuencias de red y portadora. Del lado del microcontrolador, además de las señales TxD y RxD hay una señal de control para establecer el sentido de transmisión del transceiver, teniendo en cuenta que la modalidad de trabajo es semidúplex.

La velocidad de transmisión máxima que se puede utilizar depende de la frecuencia de la portadora, que se selecciona mediante el valor de una resistencia. Para una frecuencia de 125 kHz, la tasa de transmisión puede ser de 9600 bps, mientras que para 300 kHz se puede transmitir hasta 19200 bps.

C. Formato de Trama en RS485

El formato de trama utilizado en el segmento RS485 es tal como se presenta en la Fig. 7. En la misma se observa también el detalle de los distintos campos:

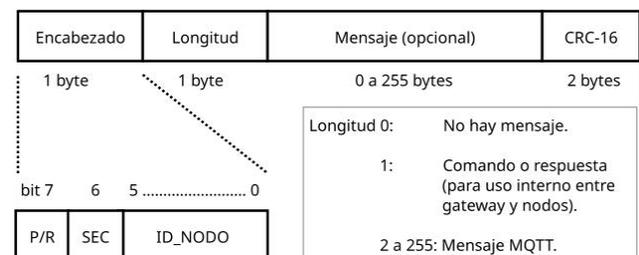


Fig. 7. Formato de trama RS485

Encabezado:

P/R (poll / respuesta): cuando el mensaje es enviado por el gateway, este bit va en uno; cuando el mensaje es una respuesta enviada por un nodo va en cero.

SEC: este bit se utiliza a modo de un número de secuencia de las tramas, alternando su valor entre uno y cero. En las tramas de respuesta, este bit va con el mismo valor que el enviado por el gateway.

ID_NODO: identificador del nodo. Cuando transmite el gateway indica el nodo interrogado o destinatario. Cuando transmite un nodo, contiene el identificador del mismo. El valor cero se utiliza para envíos en broadcast desde el gateway hacia todos los nodos. Los valores entre 1 y 63 identifican a nodos individuales.

D. Tipos de Trama

Los distintos tipos de trama utilizados en el segmento RS485 son:

Con longitud cero byte:

POLL: enviada por el gateway para habilitar a un nodo específico a transmitir.

ACK: enviada por el nodo encuestado cuando no tiene ninguna suscripción o publicación para transmitir.

Con longitud 1 byte:

INI: mensaje enviado por el gateway a cada nodo, luego de su inicio o reinicio, para solicitarle a los nodos que envíen (o reenvíen) sus suscripciones.

COM: enviada por el gateway hacia un determinado nodo, con algún comando para que el destinatario realice alguna acción de configuración o para consultar por algún estado interno. Por el momento sólo está definido el mensaje 0x01 (INICIO), el resto de los valores están reservados para futuras funciones de configuración y/o testeo.

RESP: enviada por un nodo, ante un eventual comando recibido. Por el momento no hay

mensajes definidos, sino que están reservados para su uso futuro.

Con longitud mayor o igual a dos bytes:

PUB: puede ser enviada por un nodo, cuando tiene que publicar el valor de un determinado tópico hacia el gateway, para que éste lo reenvíe al broker, como también puede ser emitida por el gateway para reenviar una publicación recibida del broker hacia los nodos RS485.

SUB: enviada por un nodo hacia el gateway, para que este se suscriba a un cierto tópico en el broker.

En la Tabla 1 se presenta el repertorio de mensajes y los valores de los distintos campos de la trama. Cabe señalar que en dicha Tabla no se menciona al campo SEC, debido a que su valor alterna entre 0 y 1, al ser utilizado para implementar un esquema de acuse de recibo implícito para ciertos tipos de tramas.

E. Mecanismo de Polling y Acuse de Recibo

La transmisión de datos en el segmento RS485 se maneja mediante un esquema de polling controlado por el gateway. Este realiza una rueda cíclica enviando una trama POLL a cada nodo.

Trama	P/R	ID NODO	Long	Mensaje
POLL	1	unicast	0	
ACK	0	unicast	0	
INI	1	unicast	1	0x01
COM	1	unicast	1	A definir
RESP	0	unicast	1	A definir
PUB (gw)	1	broadc	2 a 255	Pub
PUB (nodo)	0	unicast	2 a 255	Pub
SUB (nodo)	0	unicast	2 a 255	Subs

TABLA I
REPERTORIO DE MENSAJES EN EL SEGMENTO RS485

El bit SEC sirve a los fines de realizar un acuse de recibo implícito. Por un lado, para que el gateway confirme la recepción de una trama INI por parte del nodo destinatario. Por otra parte, para que un nodo de por recibida una trama SUB en el gateway. En la Fig. 8 se observa un ejemplo de esta secuencia de mensajes, con la pérdida de una respuesta y el uso del bit SEC (indicado entre corchetes) como acuse de recibo implícito.

El envío de mensajes SUB por parte de los nodos siempre se realiza teniendo en cuenta el acuse de recibo implícito, independientemente de que se trabaje con QoS 0 o QoS 1, a fin de asegurar que el bróker reciba la suscripción. En cambio, las publicaciones enviadas por los nodos bajo QoS 0 podrían efectuar retransmisiones si no reciben el acuse de recibo, mientras no se les haya generado una nueva publicación para enviar. En ese caso descartarán la publicación anterior y enviarán la nueva. En cambio, con QoS 1 deberán seguir retransmitiendo hasta recibir el acuse de recibo y mantener en una cola las publicaciones que se vayan produciendo.

Las publicaciones recibidas por el gateway desde el bróker, para ser retransmitidas a los nodos RS485, también admiten dos modalidades de acuerdo con QoS. Si se opera con QoS 0, el gateway envía la publicación en broadcast hacia los nodos, sin esperar confirmación. Con QoS 1 el envío debe ser individual a cada nodo, utilizando el esquema de acuse de recibo y eventual retransmisión.

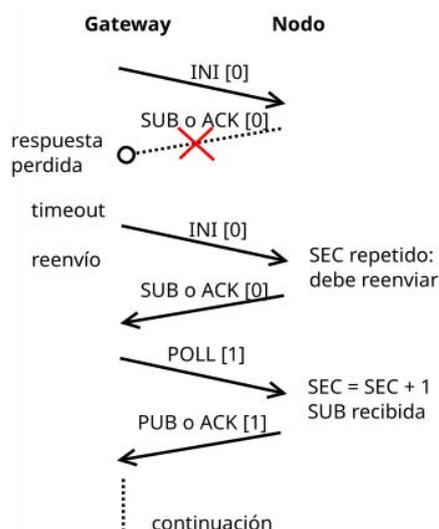


Fig. 8. Ejemplo del mecanismo de polling y acuse de recibo

Cabe observar que en el caso de operar con QoS 1, el gateway debe llevar un registro de las suscripciones de cada nodo, a fin de reenviar publicaciones sólo a los nodos suscriptos. Con QoS 0 esto no es necesario, debido a que el envío de publicaciones a los nodos es en broadcast sin acuse de recibo.

En caso de operar con QoS 0, la diferencia se presenta con las publicaciones, que no necesitan acuse de recibo. Por lo tanto, si un nodo envió PUB [n] y luego sigue recibiendo POLL [n], podría reenviarlo mientras no tenga una nueva publicación o suscripción para enviar. En tal caso descartará el que estaba reenviando y pasará a enviar el nuevo mensaje.

III. PRÓXIMOS PASOS

Continuando con los ensayos realizados y presentados en [2] y [3], las próximas actividades previstas son:

Desarrollo y ensayo de un transceiver de línea para mejorar la performance (distancia, tasa de bits y tasa de error) del THVD8000 sobre instalaciones eléctricas.

Desarrollo y ensayo de una implementación de MQTT sobre RS485, utilizando un gateway RS485-TCP/IP como nexo con el bróker MQTT. Primero utilizando RS485 convencional sobre par trenzado y luego sobre la instalación eléctrica.

REFERENCIAS

- [1] G. Friedrich and G. Reggiani, "An approach to residential energy savings using IoT and Cloud Computing to provide real-time feedback," Short Papers of the 9th Conference on Cloud Computing Conference, Big Data & Emerging Topics. Universidad Nacional de La Plata. Facultad de Informática, 2021, 2-5. <http://sedici.unlp.edu.ar/handle/10915/121564>. DOI:10.35537/10915/121564.
- [2] G. Friedrich and G. Reggiani, "Comunicación de datos para dispositivos IoT de bajos recursos: RS485 sobre la red eléctrica," Congreso Argentino de Sistemas Embebidos (CASE 2022), Universidad Nacional de la Plata, La Plata, Buenos Aires, Argentina, 18 al 19 de agosto de 2022. <http://www.sase.com.ar/case/ediciones/case-2022/>.
- [3] G. Friedrich and G. Reggiani, "Data Communication for Low Resources IoT Devices: RS485 over Electrical Wires," in IEEE Embedded Systems Letters, doi: 10.1109/LES.2023.3250162.
- [4] E. Taktak and I. Bouassida Rodriguez, "Energy consumption adaptation approach for Smart Buildings". 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications, Hammamet, Tunisia, October 30 - Nov. 3, 2017, pp. 1670-1377.
- [5] A.R. Al-Ali, I.A. Zualkernan, M. Rashid, R. Gupta and M. Alikarar, "A smart home energy management system using IoT and big data analytics approach", 2017 IEEE Transactions on Consumer Electronics, vol. 63, no. 4, pp. 426-434.



- [6] D. Vine, L. Buys and P. Morris. "The Effectiveness of Energy Feedback for Conservation and Peak Demand: A Literature Review", *Open Journal of Energy Efficiency (OJEE)*, 2013, 2, pp. 7-15.
- [7] P. Pierleoni, R. Concetti, A. Belli and L. Palma, "Amazon, Google and Microsoft Solutions for IoT: Architectures and a Performance Comparison". *IEEE Access*, vol. 8, 2020, pp. 5455-5470.
- [8] S. Ereno- Quincozes, E. R. Reginaldo-Tubino and J. Kazienk, "MQTT protocol: fundamentals tools and future directions", *IEEE Latin America Transactions.*, vol. 17, no. 9, pp. 1439-1448, Sep. 2019.
- [9] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications", *IEEE Communication Surveys & Tutorials*, Vol. 17, No. 4, Fourth Quarter 2015.
- [10] Texas Instruments, 2020. THVD8010 RS-485 Transceiver with OOK Modulation for Power Line Communication. Web: <https://www.ti.com/lit/gpn/THVD8010>.
- [11] S. Späthe and C. Prater, "Realization of IPv6 connectivity via RS485 fieldbus by adaption of 6LoWPAN", *The 2018 IEEE International Conference on Internet of Things and Intelligence System (IoT&IS)*, pp. 35-39.
- [12] H. Ochiai, H. Nakagami, Y. Teranishi and H. Esaki, "Facility networking with IP over RS485: Packet control for master-slave cascaded networks," *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Venice, Italy, 2014, pp. 410-415, doi: 10.1109/SmartGridComm.2014.7007681.
- [13] Texas Instruments, 2021. "THVD8000 RS-485 Transceiver with OOK Modulation for Power Line Communication datasheet (Rev. A)", marzo 2021, <https://www.ti.com/product/THVD8000>.

Desarrollo de un Sistema de Adquisición de Datos Basado en CIAA-Safety

Pablo N. SOLIVELLAS
Centro I+D Tec. Aeronáuticas
Fuerza Aérea Argentina
Las Higueras, Argentina
psolivellas@faa.mil.ar

Juan P. RUMIE VITTAR
Centro I+D Tec. Aeronáuticas
Fuerza Aérea Argentina
Las Higueras, Argentina
jvittar@faa.mil.ar

Darío W. DIAZ
Centro I+D Tec. Aeronáuticas
Fuerza Aérea Argentina
Las Higueras, Argentina
dw_diaz@faa.mil.ar

Resumen—En el presente trabajo se expone el análisis, diseño, desarrollo e implementación de un sistema de adquisición de datos provenientes de sensores varios, para instalarse y utilizarse en diferentes vehículos Aéreos de la FAA, a través de un microcontrolador ARM® Cortex®-R4F CPU que satisface normas de seguridad crítica, tales como IEC 61508, ISO 26262 y DO-178 (a nivel Software). Los sensores son: un GNSS (Sistema Global de Navegación por Satélite), una IMU (Unidad de Medición Inercial) y sensores de presión atmosférica y temperatura. El microcontrolador ARM fue montado sobre una placa de desarrollo denominada CIAA-Safety y su programación fue realizada bajo el formato BAREMETAL, es decir, sin Sistema Operativo en Tiempo Real.

Palabras clave—Seguridad Crítica; GNSS; IMU, UML, MISRA-C, RTOS, ISO Standards.



Fig. 1. Versión final CIAA-Safety

I. INTRODUCCIÓN

En el marco de un proyecto de investigación propuesto por Fuerza Aérea Argentina en conjunto con la Dirección General de Investigación y Desarrollo mancomunados en proyectos I+D y el Grupo de Sistema de Tiempo Real (GSTR), pertenecientes a la Facultad de Ingeniería de la UNRC (Universidad Nacional de Río Cuarto) surgió la necesidad de desarrollar un sistema (Hardware y Software) para realizar la captura y transmisión de información provenientes de sensores de posicionamiento y ubicación global (GNSS), presión atmosférica e información de actitud y aceleraciones provista por una unidad denominada Inertial Measurement Unit (IMU), con el propósito de que pueda ser utilizado en los diferentes vehículo aéreo de la Fuerza Aérea Argentina (FAA), sin importar su procedencia de fabricación. Para lograr este objetivo, se utilizó un GNSS fabricado por la Empresa SparkFun Electronics® modelo Venus638FLPx11, una IMU de Adafruit® modelo BNO055 que posee un microcontrolador marca BOSCH y un sensor de temperatura y presión atmosférica modelo BMP180. El sistema de software desarrollado, está alojado en una placa electrónica prototipo denominada CIAA-Safety [1] (Figura 1), este desarrollo sigue las buenas prácticas¹ recomendadas por los estándares de seguridad funcional para aplicaciones críticas [2], [3].

Este desarrollo se plantea como una nueva capacidad de implementación del Proyecto CIAA [4], cuyo fin está fuertemente orientado a la industria y a las PyME industriales (entre otros) de nuestro país que puedan acceder libremente a casos de estudio que les permita comprender los pasos para elaborar un sistema electrónico capaz de certificar estándares internacionales de seguridad funcional. Para alcanzar este objetivo, el proyecto CIAA-Safety está impulsado por casos de uso industrial a partir de casos de uso de sectores aeronáuticos, y que puede ser implementado en el uso de sistemas ferroviarios, aeroespaciales, automotrices, industria de procesos, etc. La extensión del proyecto abierto CIAA-Safety, tiene como desafío establecer e impulsar una metodología de desarrollo y una plataforma tecnológica de referencia como un estándar argentino para el desarrollo de sistemas seguridad crítica, equivalente a IEC 61508, ISO 26262, SIL 3, bajo un régimen de operación, a demanda. En otra arista, el desarrollo, evolución y funcionamiento de sistemas embebidos de seguridad crítica se enfrentan a cuatro retos principales:

- Cumplir con los requisitos y expectativas del usuario, tanto funcionales como no funcionales.
- Gestionar la complejidad (cada vez mayor).

- Proporcionar fiabilidad demostrable (seguridad crítica, disponibilidad e integridad).
- Encontrar métodos eficientes para el desarrollo y la evolución de los sistemas.

Este proyecto pudo concretarse gracias al aporte tecnológico y auspiciante de las siguientes instituciones (Figura 2):

- Fuerza Aérea Argentina.
- Universidad Nacional de Río Cuarto.

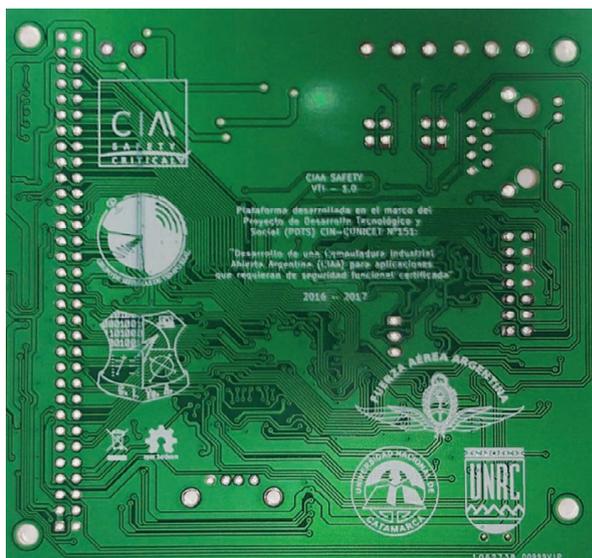


Fig. 2. PBC CIAA-Safety y entidades participantes

Técnicamente, la CIAA-Safety [1] posee un microcontrolador ARM- CortexR4F® (modelo TMS5701227LS®) en el cual cumple con las normas de seguridad crítica IEC 61508, ISO 26262, SIL 3, siendo el único dispositivo de toda la plataforma que está certificado por el fabricante. El resto del hardware utilizado, debe someterse a auditorías de certificación. El análisis y diseño del software embebido en el microcontrolador se realizó utilizando lenguaje unificado de modelado UML [5] y bajo pautas en materia de desarrollo de software relacionadas a “buenas prácticas de programación” acentuadas sobre el lenguaje de programación C, denominada MISRAC2 [6].

II. OBJETIVOS

La placa de desarrollo y múltiple propósito CIAA-Safety, se diseñó [3] con el objetivo de brindar una plataforma para desarrollos que sean canalizados en materia de seguridad funcional y también, seguridad crítica. De esta manera, se utilizó como plataforma de hardware base para ensamblar electrónica (Poncho) con componentes (GNSS, IMU y Sensor de presión atmosférica y temperatura) para lograr un prototipo denominado Sistema Registrador de Adquisición de Parámetros Portátil; SIR-APP (Figura 3).

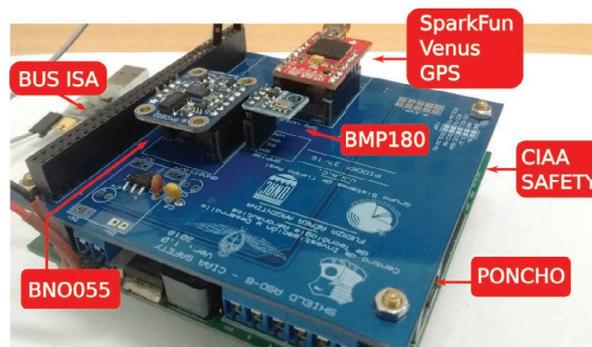


Fig. 3. CIAA-Safety + Placa Poncho

El desarrollo de este prototipo, surge de la necesidad de contar con un dispositivo que permita registrar vuelos de aeronaves. Todo lo registrado se puede representar en un sistema visual sobre diferentes tipos de topologías de mapas como así también en componentes en 3D con el propósito de otorgar valor agregado en áreas de simulación, entrenamiento y aprendizaje, tanto en la capacitación de personal de la FAA, como así también a los integrantes del GSTR, como los usuarios finales, no explicado en el presente escrito. El desarrollo, tanto de Hardware, a través de diferentes placas Poncho que pueden ser desarrolladas a futuro, como el Software, puede ser adaptado de acuerdo a las necesidades de futuros usuarios finales. El proceso de especificación y desarrollo de software siguiendo las metodologías del proceso unificado en materia de ingeniería de software [7], se realizó siguiendo los puntos ordenados a continuación:

- 1) Estudio de Artefactos de Análisis definiendo así los componentes de alto nivel.
- 2) Estudio de Artefactos de Componentes de Diseño, a partir de los Artefactos de Análisis.
- 3) Artefacto de Desarrollo del Software sobre cada componente de diseño.
- 4) Aplicación de lineamientos MISRA-C.
- 5) Integración al hardware CIAA-Safety.

III. DESARROLLO

A. Artefactos de Análisis de Componentes

El estudio del análisis de los componentes parte con el agregado de un sensor de temperatura y presión atmosférica, un sensor GNSS y una unidad Inercial (IMU). En el caso del sensor GNSS, brinda información de posicionamiento global, a través de una comunicación del tipo serial cuyo Protocolo de Datos se realiza por medio del estándar NMEA. De dicho protocolo se obtiene información necesaria a través de las palabras de encabezado GGA y RMC únicamente. Por otro lado, la información que brinda el sensor IMU son datos de actitud (Roll, Pitch, Yaw, Acelerómetro y Magnetómetro en los tres ejes XYZ), y se obtiene a través del Protocolo

de Comunicación I2C. Por último, del sensor BMP180, se obtienen datos de temperatura y presión atmosférica. De este último, se puede calcular la Altitud del dispositivo sobre el cual se encuentre instalada la CIAA-Safety, en función de este valor y de un valor de referencia de presión atmosférica sobre el nivel del mar. El valor de referencia utilizado para los casos en el que se utilizó el dispositivo fue de 1013.25 hPa. Al igual que con el dispositivo IMU, sus datos se obtienen a través del protocolo I2C. En los diagramas que se presentan en la Figura 4 y 5, se muestra como se procede con la transformación del flujo de información que son enviados por los sensores mencionados, a Unidades de Ingeniería (de aquí en adelante EU que proviene del inglés Engineering Units). Contar con dicha información codificada permite brindar la misma, a programas y/o sistemas que se encuentren conectados o se comuniquen con el producto desarrollado. A continuación se presenta un diagrama de caso de uso general, del cual parten los requerimientos iniciales. El diagrama de la Figura 4 refleja los distintos componentes desarrollados:

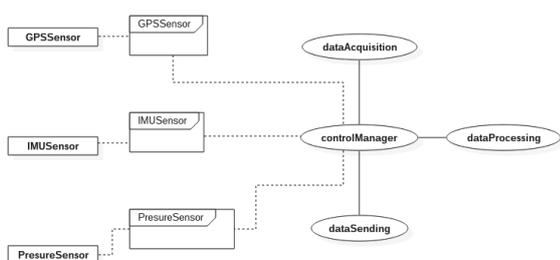


Fig. 4. Diagrama de caso de uso general

- El componente dataAcquisition es el encargado de la adquisición de datos provenientes de los diferentes sensores.
- El componente dataProcessing es el encargado de decodificar los datos recibidos y convertirlos en EU.
- El componente dataSending es el encargado de enviar los datos almacenados en EU al mundo exterior.
- El componente controlManager es el encargado de administrar al resto de los componentes, de forma coordinada y sincronizando los mensajes que interactúan entre los mismos.

La interacción de la información circundante entre los componentes, se representa por el diagrama de actividad:

El diagrama de la Figura 5 refleja la interacción de los componentes del análisis desarrollado. Básicamente la información la proporcionan los sensores mencionados para luego ser transformada (*dataProcessing*) y ser enviada a quien corresponda, en este caso a *dataSending*. Para poder analizar el 'estado' de funcionamiento de cada componente de software, se utiliza el componente de análisis controlManager, que tiene la responsabilidad

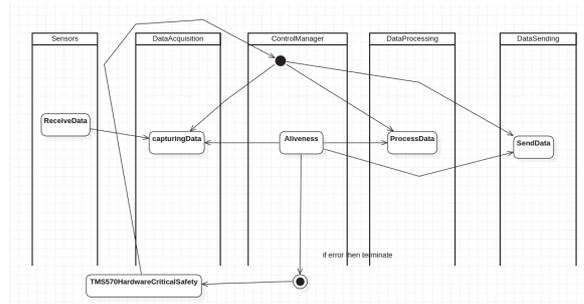


Fig. 5. Diagrama de componentes

de verificar el funcionamiento de arranque y de ejecución de todos los componentes involucrados y posterior a ello, tomar medidas en base al funcionamiento de cada sensor. Por ejemplo, si todos los sensores están en correcto funcionamiento, el Sistema estará funcionando en modo 'óptimo'. Ahora bien, si al menos un sensor está correctamente en funcionamiento, todo el sistema puede seguir con su función, solo que en modo 'degradado'. El componente controlManager, realiza una consulta periódica a cada componente para comprobar su funcionamiento. En base a la respuesta a dicha consulta, se confecciona un estado de salud (SOH, por sus siglas en inglés, State Of Health) de cada uno de los componentes. Caso contrario, ante la falla de todos los componentes (sensores) sobre las condiciones mencionadas (ej. un sensor deja de funcionar o deja de transmitir), el componente controlManager finaliza correctamente a cada uno de los componentes que forman parte del sistema, dejando la potestad de preservar la seguridad crítica al microcontrolador, cuyas cualidades lo resuelve satisfactoriamente utilizando lockstep, que es un desarrollo electrónico embebido dentro del microcontrolador que garantiza la seguridad crítica y funcional, para mas información de como funciona este mecanismo electrónico, dirigirse al sitio del fabricante [8].

B. Artefactos de Componentes de Diseño

A continuación del lineamiento de diagramas de análisis, se desarrollaron los componentes de diseño, lógicamente por la expansión de los componentes de casos de uso y análisis, respectivamente. A modo de síntesis informativa y por cuestión de espacio del presente documento, no es posible citar todos los componentes de clase de diseño realizados. Para ello, se presentará uno de ellos, el resto de los componentes en los casos de uso y clases de análisis, siguen la misma metodología y lineamientos de desarrollo que el que se presentará. El diseño de componentes es modular y adaptable siguiendo las pautas de ingeniería de software mencionadas anteriormente. Esto significa que las funciones principales denominadas como BNO055 y GNSS, hacen uso o se reutilizan las funciones de la librería

desarrollada para cada sensor respectivamente. Permitiendo que, por un lado, cualquier cambio que se realice en el software, repercuta directamente a estas funciones, heredando las mismas; y por otro lado, cambios y/o agregados de nuevas funciones sobre estas callbacks ya definidas, no repercuten sobre las funcionalidades ya desarrolladas previamente a las librerías que manejan a los sensores.

C. Desarrollo del Software

Habiendo realizado un análisis de todos los componentes involucrados, tomar medidas en base al funcionamiento de cada sensor, como así también, los componentes de diseño desarrollados por la expansión de los componentes de casos de uso y análisis de manera modular y adaptable, el desarrollo completo se llevó a cabo con la librería correspondiente de cada sensor involucrado en el sistema sobre el lenguaje C utilizando a la IDE CCS (Code Composer Studio) sobre el microcontrolador TMS5701227LS®. Para ello, se explicará como se desarrolló cada sección que interviene cada sensor, de forma modular en librerías:

- 1) Registros propios (de Hardware) del sensor.
- 2) Inicialización, bus de comunicación y calibración del Sensor.
- 3) Obtención de datos en unidades Euler (para el caso de la IMU) y Conversión de datos Euler (para el caso de la IMU) en Unidades de Ingeniería (EU).

1) *Registros de Hardware:* En la presente sección, se muestran algunos registros definidos para la unidad inercial, utilizando para este caso, los registros Euler. Dichos registros propios de la electrónica de IMU BNO055, son los necesarios para escribir en ellos, y obtener como respuesta los valores de, por ejemplo, actitud, aceleración, magnetómetro, etc.

2) *inicialización, bus de comunicación y calibración del Sensor:* El bus de comunicación seleccionado para cada uno de los sensores involucrados son:

- 1) BNO055 y BMP180 - Bus I2C.
- 2) GNSS - Bus Serial.

Para ejemplificar un caso muy particular, y siguiendo con el ejemplo anterior de la BNO055, el bus de comunicación I2C envía escrituras de palabras de datos de 8 bits (llamada función write8), de tipo unsigned int. La inicialización y calibración se realizan en conjunto. Primero se establece la comunicación inicial con el sensor, y se realiza el envío del modo de configuración normal al sensor, esto es, envío de datos de actitud, aceleraciones y magnetómetro en los tres ejes (x,y,z). Luego, se reinicializa la placa, para que tome la configuración previamente enviada y se realiza una espera de segundos, hasta que el sensor responda con un protocolo predeterminado para dar respuesta a que está en funcionamiento seleccionado. Luego se procede a deshabilitar el funcionamiento de bajo consumo enviando

o configurando al BNO055 bajo power mode: normal, y se obtiene el ID de dirección de paginado y por último el modo de funcionamiento a través de respuestas de datos a consultas por direcciones de registros. Además, se configura la utilización del sistema de disparo Trigger de datos sobre direcciones de memoria, a partir de los registros de datos propios del sensor.

3) *Obtención de datos en unidades Euler y conversión de datos Euler en unidades de Ingeniería:* Continuando con el ejemplo de la BNO055 (Para el resto de los sensores el proceso es análogo por lo que no se expondrá en este documento para evitar extender demasiado el mismo) los datos en unidad Euler, se obtienen a partir de la solicitud de los registros de orientación (recordar que hay tres tipos: Orientación, Aceleración y Magnetómetro). La información de estos registros, debe dividirse en los correspondientes 3 ejes (x,y,z) y luego se los almacena en unidades de Euler en un único vector. Posterior a ello, se transforma dicho vector en EU convirtiendo los datos de Euler a grados en los tres ejes (x,y,z). De forma similar, se realizan las conversiones correspondientes de los datos de registro de aceleración del sensor a aceleración en EU y datos de registro de magnetómetros a intensidad de campo magnético en EU. Para el sensor GNSS se implementó un componente genérico que no depende del modelo GNSS escogido, es decir, puede ser reemplazado por otro modelo si así se lo requiere. Lo que el sistema básicamente hace, es buscar en las palabras protocolo NMEA que arriban del sensor GNSS vía comunicación serial, las dos palabras utilizadas en esta implementación: *GN* y *GPRMC* siendo cabecera del protocolo genérico para cualquier constelación de posicionamiento global:

- GP es del sistema de constelación GPS.
- GA es del sistema de constelación GALILEO.
- GL es del sistema de constelación GLONASS.
- GN es una combinación de algunas de ellas.

D. MISRA-C

Es un conjunto de recomendaciones o directrices para el desarrollo de software en lenguaje C, desarrollado por The Motor Industry Software Reliability Association (MISRA). Su finalidad es proveer portabilidad, seguridad y fiabilidad al código fuente en el contexto de software embebido. Si bien, MISRA-C no es un estándar abierto y los documentos con las directrices se adquieren mediante la compra del mismo, cuando se actualiza la documentación, queda a libre disposición la versión anterior a la última vigente, siendo esta versión libre la utilizada para su análisis y aplicación en el presente trabajo. MISRA-C, básicamente, es una guía de reglas para clasificar un código fuente según los siguientes escalafones:

- Reglas/directivas preceptivas (mandatory).
- Reglas/directivas requeridas (required).
- Reglas/directivas recomendadas (advisory).

En base a estos lineamientos, se procedió a ejecutar las reglas de software desarrollado. Se mostrará a continuación, algunos de los resultados encontrados, a modo de ejemplo:

- MISRA-C:2004 6.3/A: Los typedefs que indican tamaño y signo se deben utilizar en lugar de los tipos numéricos básicos.
- MISRA-C:2004 5.2/R: Los identificadores en un ámbito interno no deben usar el mismo nombre que un identificador en un ámbito global, y por lo tanto ocultar ese identificador `scale`.
- MISRA-C:2004 10.1/R: El valor de una expresión de tipo entero no se convertirá implícitamente en un tipo subyacente diferente si no es una conversión a un tipo entero más amplio con el mismo signo.
- MISRA-C:2004 9.1/R: Todas las variables deben ser inicializadas con un valor antes de ser utilizadas (variable `quat`).
- MISRA-C:2004 17.4/R: La indexación de matrices será la única forma permitida de aritmética de punteros.

IV. INTEGRACIÓN

La CIAA-Safety, se ensambla a modo de encastre vertical por medio del BUS de comunicación ISA que sigue la arquitectura PC-104, esto es, sobre la placa base (CIAA-Safety) se integran componentes de Hardware elegidos a través un Shield o Poncho que luego se superpone (encastra) en la placa CIAA de forma vertical y a través del BUS ISA (Figura 3). Con el objetivo de realizar una validación, se optó por diseñar una test suite con ensayos aplicado al sistema presentado:

- Test case: Corroborar el funcionamiento del sistema desconectando alguno de los sensores y comprobando que el funcionamiento continúa con la captura del resto de los sensores que estén conectados. Dado que la conectividad con todos los sensores no es mandatoria, es deseable al menos un sensor conectado, asegura la continuidad de funcionamiento del sistema, aunque sea en modo degradado.
- Test case: Corroborar que el envío de la información decodificada en Unidades de Ingeniería al exterior solo se realiza si está presente la señal PPS (Pulse Per Second) del sensor GNSS. Esto se desarrolló de esta manera para utilizar el sincronismo propio que posee el sensor GNSS a través de su señal PPS, y así garantizar la frecuencia de envío de tramas de datos al exterior.
- Test case: Corroborar función de Aliveness. Para verificar que cada componente de software está funcionando correctamente, con el objetivo de verificar el funcionamiento del sistema en su fase inicial de arranque y en condiciones normales, esto es, todos sus sensores funcionando correctamente y enviando la información pertinente a cada uno.

Para cubrir de forma integral estos caso de estudio iniciales, se muestra en la Figura 6, una captura de datos, para contrastar el funcionamiento de los sensores: datos visuales del analizador lógico en el canal 0, datos codificados y enviados a una tasa de 10 muestras por segundo (10 Hz) y sincronizados por la señal propia PPS del GNSS. Posteriormente en el canal 1, la señal propia PPS del GNSS y el canal 4 las palabras GGA y RMC respectivamente, datos geográficos de posición del GNSS y por último, el canal 6 y 7 las señales de SDA y SCL de las unidades inercial (IMU) y sensor de temperatura y presión respectivamente. En la ventana interpuesta de la Figura 6, se muestran los datos ya codificados. Cabe destacar que la acción de Aliveness, se procesa continuamente. El funcionamiento normal y continuo del sistema, evidencia las acciones de este proceso.



Fig. 6. Ejemplo de Funcionamiento - Ensayo

V. ENSAYOS

Si bien, desde el momento de su desarrollo, se pretende que el dispositivo se utilice en aeronaves de la FAA, por cuestiones de Aeronavegabilidad Militar no ha sido factible realizar ensayos en los mismos. Para poder realizar dichos ensayos, se debe realizar el proceso de certificación generando toda la documentación necesaria y conseguir en una primera instancia un Certificado de Aeronavegabilidad Experimental. Sin embargo, se ha logrado ensayar en vehículos terrestres (automóviles o mo-



tocicletas) y vehículos aéreos de aviación civil (Cessna 182 Skylane), habiendo obtenido resultados esperados.

VI. CONCLUSIÓN Y TRABAJO FUTURO

El objetivo de este trabajo de desarrollo de Software y de Hardware, por cierto, llevado a cabo por ambos Centros I+D, da el puntapié inicial para comenzar a realizar la documentación y gestión para una futura certificación si así lo demandan los usuarios finales. Este Hardware, cumple con las prestaciones estipuladas en los lineamientos iniciales y el Software implantado es el mismo que el que se viene desarrollando en otros productos, por ejemplo, utilizando hardware de la empresa TI (Texas Instruments), que inicialmente se utilizaron en paralelo mientras se modelaba y fabricaba la placa multipropósito CIAA-Safety. Esta buena adaptación de Software desde los productos de TI a la placa CIAA-Safety se logró gracias a las técnicas de desarrollo de Software y buenas prácticas de programación (captura, análisis, diseño, desarrollo, implantación y test) utilizadas a lo largo de todo el proyecto. Para esto, se realizó bajo una versión Baremetal (sin Sistema Operativo) cumpliendo en ambos casos la Guía de MISRA-C. A Futuro, se prevé una versión sobre un sistema operativo RTOS certificable, para lo cual, se debe gestionar su compra. Cabe aclarar que no se utilizó manejo de interrupciones (síncronas o asíncronas), y tampoco manejo de excepciones, dado que estas técnicas y/o metodologías de software y hardware van en contra de la seguridad crítica. Los objetivos a futuro, se pretende para lo desarrollado, incorporar un sistema de alimentación autónomo, con el propósito de no interferir con el circuito eléctrico de la aeronave que vaya a ser instalado, ya que, de realizar esta modificación, se debería certificar el dispositivo en conjunto con las aeronaves que se vayan a instalar, siendo esto, complejo debido a que, se debe realizar una certificación por cada aeronave según lo requerido por Dirección General de Aeronavegabilidad

Militar Conjunta [9]. Por otro lado, se prevé ampliar la capacidad, en primera instancia, de la salida (trama) de datos, para enviar un frame que corresponda al formato SBS (Surveillance Broadcast Services) que se utiliza por el ATC (Aircraft Traffic Control) para intercambiar información proveniente de un transponder ADS-B hacia otros ATC sobre la red de satélites de posicionamiento global. Y en segunda instancia, transformar el prototipo en formato ADS-B, es decir, frames extendidos de 112 bits como lo exige el estándar ADS-B con el objetivo de convertirlo en un transponder ADS-B, agregando un poncho nuevo con un dispositivo transponder para que resuelva la comunicación inalámbrica. En ambos casos, se eliminan completamente los sensores IMU, temperatura y presión atmosférica, porque el estándar ADS-B se basa en la utilización del sensor que brinda GNSS.

REFERENCIAS

- [1] Computadora Industrial Abierta Argentina para Aplicaciones de Seguridad Crítica - CIAA Safety, Proyecto CIAA, ACSE, CADIEEL. Marzo 2022. Proyecto CIAA.
- [2] D.Díaz, G. Rodríguez, R. Manno, J. Amor, D. Fusari, 2018. Análisis de falla del Modelado de la Computadora Industrial Abierta Argentina de Seguridad Crítica bajo AADL, 5to Congreso Argentino de Ingeniería Aeronáutica. page 210-216.
- [3] D.Díaz, G. Rodríguez, R. Manno, J. Amor, D. Fusari, 2018. Modelado de la Computadora Industrial Abierta Argentina de Seguridad Crítica bajo AADL, Simposio Argentino de Sistemas Embebidos. 2018. ISBN: 978-987-46297-4-6.
- [4] <http://www.proyecto-ciaa.com.ar/http://www.proyecto-ciaa.com.ar/>, Dic 2022.
- [5] Rumbaugh, Jacobson, Booch, 2005. El Lenguaje Unificado de Modelado Manual de Referencia Segunda Edición, Addison-Wesley.
- [6] MISRA-C, 2004. Guidelines for the use of the C language in critical systems, Edition 2 reprinted July 2008 incorporating Technical Corrigendum 1, www.misra-c.com, ISBN 978-0-9524156-2-6 paperback, ISBN 978-0-9524156-4-0 PDF.
- [7] Pressman, "Ingeniería de software - Un enfoque práctico", Séptima Edición - Editorial: Mc Graw Hill, 2010, ISBN: 978-607-15-0314-5.
- [8] <https://www.ti.com/lit/ug/spnu613/spnu613.pdf?ts=1670258087654>, Dic 2022.
- [9] <http://www.fuerzas-armadas.mil.ar/Dependencias-DIGAMC-Normas-Vigentes.aspx>, Dic 2022.



Creación de un sistema IoT embebido en el microcontrolador Esp32 para la gestión de sensores y actuadores en modo AP (Punto de acceso) y STA (Estación)

Luis Carlos Triviño Torres

Universidad Santo Tomás

Ingeniería informática

Bogotá, Colombia

luistrivino@usta.edu.co

Resumen— Este proyecto, desarrollado por el semillero de investigación Ciberondas de la programa de Ingeniería Informática de la Universidad Santo Tomás, se centró en la creciente relevancia del Internet de las Cosas (IoT), transformando la interacción con dispositivos y el entorno. Su objetivo era diseñar un sistema IoT basado en el microcontrolador Esp32, que gestionara eficientemente sensores y actuadores, adaptándose a diferentes entornos de conectividad con modos de operación punto de acceso AP y STA. La metodología incluyó la integración de una API REST para comunicación entre el Esp32 y servicios externos, y el uso del protocolo MQTT, compatible con el broker EMQX en entornos locales y en la nube. La interfaz de usuario se desarrolló en HTML, CSS y JavaScript, implementada en un dashboard web para monitoreo y control en tiempo real. Los resultados destacaron la versatilidad del sistema, empleando el Esp32 como unidad central y sensores de alta precisión, contribuyendo significativamente al campo del IoT.

Palabras clave: IoT, Esp32, Monitoreo, MQTT, Conectividad

Introducción

El Internet de las Cosas (IoT, por sus siglas en inglés) ha emergido como un paradigma innovador que conecta dispositivos físicos a través de internet, permitiendo la automatización y el control remoto en diversos

entornos (Atzori, Iera & Morabito, 2010). Este proyecto, desarrollado por el semillero Ciberondas de la Universidad Santo Tomás, tiene como objetivo diseñar e implementar un sistema IoT embebido en el microcontrolador ESP32 para la gestión de sensores y actuadores en tiempo real.

El sistema está diseñado para operar tanto en modo punto de acceso (AP) como en modo STA, utilizando el protocolo MQTT y una API REST para facilitar la comunicación entre dispositivos y servicios externos. La metodología incluyó el desarrollo de un dashboard web para el monitoreo y control del sistema, así como pruebas de rendimiento en entornos controlados.

Los resultados demostraron la versatilidad y eficiencia del sistema en la gestión de sensores y actuadores, destacándose por su eficiencia energética y capacidad de adaptación a diferentes escenarios de conectividad.

Este proyecto contribuye significativamente al campo del IoT al ofrecer una solución robusta y escalable para aplicaciones que requieren monitoreo y control en tiempo real (Al-Fuqaha et al., 2015; Gubbi, Buyya, Marusic & Palaniswami, 2013).

Justificación

El proyecto se fundamenta en la creciente demanda de soluciones IoT que sean robustas tanto técnica como operativamente, además de flexibles y escalables para diversos entornos de uso. Los sistemas IoT convencionales enfrentan desafíos significativos en términos de

conectividad, gestión de datos e integración con sistemas existentes (Patel & Patel, 2016).

Al desarrollar un sistema basado en ESP32 con una API REST y el protocolo MQTT, este proyecto aborda estas limitaciones, ofreciendo una plataforma adaptable para aplicaciones tanto domésticas como industriales. Los beneficios de este sistema se extienden a múltiples sectores, incluyendo el hogar, donde facilita la implementación de soluciones domóticas; la industria, mejorando la automatización de procesos; la salud, permitiendo un monitoreo más eficiente; el agro y la ganadería, optimizando la gestión de recursos y el monitoreo en tiempo real; y la automatización en general, donde se pueden integrar fácilmente en sistemas existentes para mejorar la eficiencia operativa.

Además, fomenta la adopción de tecnologías IoT en nuevos contextos, facilitando su acceso a una amplia gama de usuarios y potenciando su aplicabilidad en diversos sectores. El sistema tiene como objetivo mostrar toda la información del dispositivo al usuario a través del dashboard, permitiendo un mejor control sobre el mismo. El sistema también permite dos modos de operación: localmente y con acceso a internet, asegurando así su flexibilidad y utilidad en diferentes condiciones de uso.

Utiliza memoria y almacenamiento interno para el firmware embebido, lo que facilita el acceso del usuario final al dispositivo. Otra ventaja significativa es la capacidad de actualizar el firmware a través del dashboard, permitiendo al usuario adaptar el dispositivo a condiciones variables de uso de manera sencilla.

Materiales y Métodos

El diseño del sistema IoT se basa en una arquitectura modular centrada en el microcontrolador ESP32. Este dispositivo puede operar en dos modos: punto de acceso (AP) y STA. En el modo AP, el ESP32 actúa como un punto de acceso Wifi independiente, mientras que en el modo STA se conecta a redes Wifi existentes, integrándose en la infraestructura de red del usuario.

En cuanto al hardware, la selección de componentes incluyó el ESP32-WROOM-32D

como unidad central, así como los sensores DHT22 y DS18B20 para la medición de temperatura y humedad. También se incorporaron actuadores como LEDs RGB y relés de estado sólido para diversas funciones de control.

Para el desarrollo del software, se utilizó el entorno Arduino IDE para programar el ESP32. El sistema incluyó la implementación de una API REST para la gestión de datos y el uso del protocolo MQTT para la comunicación eficiente entre dispositivos.

El dashboard web fue desarrollado utilizando HTML, CSS y JavaScript. Este sigue una arquitectura de Single Page Application (SPA) y utiliza websockets para permitir actualizaciones en tiempo real de los datos recogidos por el sistema IoT.

Se llevaron a cabo pruebas exhaustivas de rendimiento en entornos controlados para evaluar la conectividad del sistema, su eficiencia energética y capacidad de respuesta en diferentes escenarios de operación. Estas pruebas aseguraron la fiabilidad y funcionalidad del sistema en diversos entornos de uso.

A continuación, se muestra las características principales de los dos modos usados.

TABLA I Características del modo AP y STA

Característica	Modo AP (Punto de acceso)	Modo STA (Estación)
Función Principal	El ESP32 actúa como un punto de acceso Wi-Fi, creando su propia red.	El ESP32 actúa como un cliente, conectándose a una red Wi-Fi existente.
Conectividad	Permite que otros dispositivos se conecten a su red Wi-Fi.	Se conecta a un punto de acceso externo, como un enrutador o hotspot.
Dirección IP	El ESP32 asigna direcciones IP a los dispositivos conectados.	El ESP32 recibe una dirección IP del punto de acceso al que se conecta.
Aplicaciones Comunes	Redes locales ad hoc, configuración de dispositivos, aplicaciones sin acceso a Internet.	Conexión a Internet, comunicación con servidores remotos, control de dispositivos a distancia.
Flexibilidad	Menos dependiente de infraestructura	Depende de la existencia de un punto de acceso

	externa, ideal para aplicaciones locales.	Wi-Fi para conectarse.
Acceso a Internet	No tiene acceso directo a Internet; depende de una conexión compartida si es necesario.	Acceso total a Internet y recursos de red una vez conectado.
Configuración de Red	Crea una nueva red Wi-Fi con SSID y contraseña definidos por el usuario.	Se conecta a una red existente proporcionando SSID y contraseña del AP.
Rango de Dispositivos Conectados	Puede soportar varios dispositivos conectados simultáneamente.	El ESP32 es uno de los muchos dispositivos conectados al AP.

El modo AP del ESP32 es ideal cuando se requiere que el microcontrolador cree su propia red Wi-Fi. En este modo, el ESP32 actúa como un punto de acceso, permitiendo que otros dispositivos se conecten directamente a su red. Esta configuración es particularmente útil en situaciones donde no se dispone de un enrutador o punto de acceso externo, ya que el ESP32 gestiona completamente la red, asignando direcciones IP y permitiendo la comunicación local entre los dispositivos conectados.

Por otro lado, el modo STA es utilizado cuando el ESP32 necesita conectarse a una red Wi-Fi existente. En este caso, el microcontrolador actúa como un cliente, buscando y uniéndose a un punto de acceso externo, como un enrutador o hotspot. Este modo es esencial cuando se requiere acceso a Internet o comunicación con otros dispositivos en la misma red, permitiendo que el ESP32 interactúe con servidores remotos, envíe y reciba datos a través de la red, o sea controlado a distancia.

Resultados

El sistema IoT desarrollado demostró un rendimiento en diversos entornos de conectividad, tanto en modo AP como en STA. Los resultados clave se presentan en la siguiente tabla:

TABLA 2 Rendimiento en los modos AP y STA

Parámetro	Modo AP	Modo STA
Cobertura efectiva	30 metros en espacios abiertos	Conectividad estable en redes 2.4GHz y 5GHz
Tiempo de	150 ms	80 ms

respuesta (API)		
Consumo energético (activo)	120 mA	120 Ma
Consumo energético (bajo consumo)	10 μ A	10 μ A
Latencia en MQTT	20 ms (broker local)	100 ms (broker en la nube)

La tabla compara el rendimiento del ESP32 en los modos AP y STA según varios parámetros clave.

En cobertura efectiva, el modo AP ofrece hasta 30 metros en espacios abiertos, creando su propia red Wi-Fi. En contraste, el modo STA se conecta a redes Wi-Fi existentes, aprovechando la estabilidad de las bandas 2.4GHz y 5GHz.

El tiempo de respuesta en el modo AP es de 150 ms, debido a la gestión de la red por parte del ESP32, mientras que en el modo STA es más rápido, con 80 ms, aprovechando la eficiencia de la red Wi-Fi.

Ambos modos consumen 120 mA en activo y solo 10 μ A en bajo consumo, mostrando un rendimiento energético similar.

La latencia en MQTT es de 20 ms en el modo AP (con un broker local) y 100 ms en el modo STA (con un broker en la nube), reflejando la mayor distancia de transmisión en el segundo caso.

Discusión

Los resultados obtenidos subrayan la capacidad del ESP32 para adaptarse eficazmente en la gestión de sistemas IoT, destacando su versatilidad en los modos de operación AP y STA. Mientras que el modo STA ofrece una latencia más baja, ambos modos mantienen un consumo energético similar, lo que es ventajoso para aplicaciones en entornos remotos. Esta eficiencia energética permite al ESP32 operar durante períodos prolongados con un bajo consumo, lo que es crucial para aplicaciones en el hogar, la industria, la salud, el agro, la ganadería, y la automatización de procesos.

A pesar de estos logros, se identificaron áreas donde es posible mejorar, como el

soporte para una gama más amplia de sensores y actuadores, así como la incorporación de medidas de seguridad más avanzadas para proteger la integridad de los datos. Estos aspectos podrían potenciar aún más la adaptabilidad y la seguridad del sistema, permitiendo su uso en una variedad aún mayor de aplicaciones. Además, el proyecto destaca por su capacidad para actualizar el firmware a través de un dashboard, lo que ofrece una ventaja significativa al usuario al permitirle ajustar el dispositivo a condiciones variables de uso con facilidad.

Conclusiones

1. El sistema IoT basado en ESP32 desarrollado en este proyecto demuestra ser una solución eficiente, flexible y adaptable para la gestión de sensores y actuadores, lo que lo convierte en una opción viable para una amplia variedad de entornos, desde aplicaciones domésticas hasta industriales.
2. La integración del protocolo MQTT y una API REST no solo facilita una comunicación fluida y escalable entre dispositivos, sino que también potencia la interoperabilidad del sistema con otros servicios y plataformas, lo que incrementa su aplicabilidad en sectores como la domótica, la salud, el agro y la automatización de procesos.
3. Aunque el sistema muestra un rendimiento robusto, se identifican áreas de mejora cruciales, como la implementación de medidas de seguridad avanzadas para proteger la integridad de los datos y la ampliación de la compatibilidad con un mayor número de dispositivos y sensores. Estos aspectos serán abordados en futuros desarrollos para fortalecer la eficacia y seguridad del sistema.

Referencias

- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347-2376.
- Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787-2805.
- Espressif Systems. (2022). ESP32 Series Datasheet. Retrieved from

https://www.espressif.com/sites/default/files/documentation/Esp32_datasheet_en.pdf

Fielding, R. T., & Taylor, R. N. (2000). Architectural styles and the design of network-based software architectures. University of California, Irvine.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1-17.

Patel, K. K., & Patel, S. M. (2016). Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges. *International Journal of Engineering Science and Computing*, 6(5), 6122-6131.

Propuesta de Diseño e Implementación de Arquitecturas IoT para el Monitoreo Remoto de Pacientes

Carlos Torres

Salvador Nuñez

Raul Montiel

UTN FRRe CInATIC

UTN FRRe CInATIC

UTN FRRe CInATIC

Centro de Investigación Aplicada en
TICs

Centro de Investigación Aplicada en
TICs

Centro de Investigación Aplicada en
TICs

Resistencia, Argentina

Resistencia, Argentina

Resistencia, Argentina

carlos.ing.2013@gmail.com

salvador.nunez@gfe.frre.utn.edu.ar

raulmontiel@gfe.frre.utn.edu.ar

Jorge Roa

Sergio Gramajo

UTN FRRe CInATIC

UTN FRRe CInATIC

Centro de Investigación Aplicada en
TICs

Centro de Investigación Aplicada en
TICs

Resistencia, Argentina

Resistencia, Argentina

@gfe.frre.utn.edu.ar

sergiogramajo@gfe.frre.utn.edu.ar

Resumen—Este trabajo se enfoca en el desarrollo de sistemas inteligentes de Internet de las Cosas (IoT) aplicados al cuidado de la salud a través de monitoreo remoto. El objetivo principal es crear soluciones innovadoras que permitan una atención médica más personalizada y eficiente. Para lograrlo, se llevará a cabo un análisis exhaustivo de las arquitecturas y protocolos de telecomunicaciones utilizados en el ámbito de la salud, así como la implementación de modelos de inteligencia artificial y ciencia de datos para el procesamiento y análisis de los datos obtenidos de los dispositivos IoT. Además, se emplearán metodologías de desarrollo ágil para garantizar la rápida adaptación a las necesidades cambiantes del sector. A través del desarrollo de prototipos y la realización de pruebas rigurosas, se buscará validar la eficacia y la fiabilidad de las soluciones propuestas, con el fin de contribuir a mejorar la calidad de vida de pacientes y optimizar la gestión de los servicios de salud.

Palabras clave: e-health, IoT, IA.

I. INTRODUCCIÓN

El proceso de monitorear la salud de pacientes con internación domiciliaria o, incluso, de pacientes en los hospitales, requiere la observación constante de parámetros útiles por parte del equipo de médicos o enfermeros y profesionales [1], [2], [3], [4]. En este escenario la tecnología IoT puede consistir en dispositivos físicos de monitoreo de parámetros de interés para pacientes (glucosa, presión arterial, frecuencia cardíaca, monitoreo de actividad, etc.).

En diversas propuestas, se emplean dispositivos IoT de bajo consumo energético como sensores para recopilar datos biométricos de los pacientes, tales como temperatura, frecuencia cardíaca, saturación de oxígeno, entre otros. Estos datos son transmitidos a través de sistemas de telecomunicaciones y almacenados en plataformas remotas. Por ejemplo, en [5] se describe una solución que aprovecha la



tecnología de Radiofrecuencia para el seguimiento inteligente de parámetros de salud. A través de IoT, se facilita la transmisión remota de información biométrica, permitiendo a los profesionales sanitarios realizar un monitoreo continuo del estado de salud de los pacientes. En [6] se tiene como objetivo desarrollar un sistema web de telemonitoreo de la salud que aproveche las capacidades de la tecnología IoT. A través de una red 4G se enviarán los datos recopilados de diversos sensores médicos y ambientales para evaluar el estado de salud de los pacientes. Estos datos serán procesados y analizados mediante algoritmos de inteligencia artificial, con el fin de extraer información relevante para la toma de decisiones clínicas.

La aplicación de técnicas de aprendizaje automático y profundo en el ámbito de la salud ha demostrado ser una herramienta valiosa para la detección temprana de enfermedades, el diagnóstico preciso y la optimización de los tratamientos. En este sentido, este estudio busca contribuir al desarrollo de soluciones innovadoras que permitan mejorar la calidad de vida de los pacientes y la eficiencia de los servicios de salud [7], [8], [9], [10].

II. METODOLOGÍAS Y MÉTODOS

El trabajo se desarrollará como investigación aplicada dentro de las áreas de Ciencia de Datos y e-health, combinará herramientas de extracción de datos provenientes de dispositivos IoT empleados en internación domiciliar para su procesamiento con técnicas de Aprendizaje Automático (Machine Learning) (ML) para diseñar un modelo inteligente de gestión.

A pesar de que la literatura propone varios modelos teóricos, existe una brecha de su aplicación en entornos reales y aún más en el ámbito geográfico local, que se propone abordar el trabajo.

El enfoque metodológico consiste en la revisión y análisis de estándares para IoT, anomalías preponderantes y clasificación de algoritmos de analíticas de datos funcionales.

Consecución y generación de datasets, así como los referidos a protocolos de las arquitecturas para el tratamiento y control de outliers de sistemas de IoT aplicados en salud.

Determinación de los algoritmos de analítica de

datos útiles para el modelo.

Diseño y desarrollo del modelo, que incluyen aspectos de hardware y software.

Desarrollo de escenarios de prueba para las aplicaciones necesarias, para lo cual se seleccionarán las tecnologías de programación que se consideren más adecuadas.

Siguiendo las pautas del método científico se trabaja de la siguiente manera:

Formulación de hipótesis. Que será el eje para el diseño y creación de escenarios de prueba simulados y reales.

Recopilación de datos observados. Que abarcará aspectos como pruebas de simulaciones, aplicaciones y escenarios, lo que permitirá identificar los atributos idóneos.

Contrastación de hipótesis con los resultados obtenidos. Que permitirá medir la calidad de las aplicaciones, escenarios propuestos y resultados obtenidos y validación de la hipótesis propuesta

III. ARQUITECTURA PROPUESTA

Una vez descrito que es un sistema IoT y los protocolos de comunicación más utilizados vamos a describir el sistema propuesto. Específicamente el trabajo está enmarcado en un sub-campo denominado IoMT (Internet of Medical Things)

En la siguiente figura podemos observar las partes fundamentales del sistema, el monitoreo consta de dos componentes, ambos ubicados en la residencia del paciente (también podrían ubicarse en una habitación de internados de un sanatorio), una de estos componentes tiene la función es medir variables de interés médicas y el otro mide las condiciones ambientales en las que se encuentra el paciente. Ambos dispositivos se conectarán a la red a través de dos medios de comunicación, el primero más conocido y utilizado, la una red Wifi provista por un modem 4G ubicado en la misma residencia y el segundo medio es por LoRa (ver Fig 1).

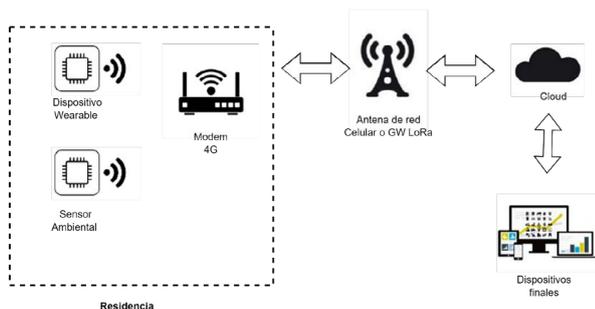


Fig. 1. Arquitectura general del sistema

Para poder asegurar la conectividad y el envío de datos a la nube, el modem 4G se va conectar a la red de dos empresas de telefonía celular. En la nube se tendrá corriendo los servicios del broker MQTT y una plataforma de monitoreo.

IV. TRABAJOS FUTUROS Y CONCLUSIONES

Los resultados de este trabajo abren nuevas perspectivas para el tratamiento de la salud digital y el monitoreo remoto de datos biométricos. La capacidad de tratar múltiples eventos y analizar datos podría permitir personalizar los tratamientos en función de las características individuales de cada paciente. Con la puesta en marcha de los prototipos creados se avanzará en la generación de mayores volúmenes de datos. A su vez se trabajará en abordar los desafíos éticos y regulatorios asociados con el uso de datos sensibles y garantizar la seguridad y la privacidad de los pacientes.

Las actividades principales abordadas son:

- Detección de patrones y predicción de eventos futuros en tiempo real.
- Desarrollar modelos que puedan integrar datos provenientes de diferentes fuentes (sensores, registros médicos, etc.) para obtener una visión más completa del estado de salud del paciente.
- Diseñar interfaces intuitivas que permitan a los profesionales de la salud visualizar los datos de manera clara, facilitando la toma de decisiones

International Conference on Computing Communication Control and Automation (ICCUBEA), ago. 2018, pp. 1-5. doi: 10.1109/ICCUBEA.2018.8697681.

[2] Z. U. Ahmed, M. G. Mortuza, M. J. Uddin, Md. H. Kabir, Md. Mahiuddin, y MD. J. Hoque, «Internet of Things Based Patient Health Monitoring System Using Wearable Biomedical Device», en *2018 International Conference on Innovation in Engineering and Technology (ICIET)*, dic. 2018, pp. 1-5. doi: 10.1109/CIET.2018.8660846.

[3] M. Alshamrani, «IoT and artificial intelligence implementations for remote healthcare monitoring systems: A survey», *Journal of King Saud University - Computer and Information Sciences*, vol. 34, n.º 8, Part A, pp. 4687-4701, sep. 2022, doi: 10.1016/j.jksuci.2021.06.005.

[4] K. C. Chowdary, K. Lokesh Krishna, K. L. Prasad, y K. Thejesh, «An Efficient Wireless Health Monitoring System», en *2018 2nd International Conference on 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, ago. 2018, pp. 373-377. doi: 10.1109/I-SMAC.2018.8653716.

[5] K. V. Sahukara, M. B. Ammisetty, G. S. K. G. Devi, S. Prathyusha, y T. S. Nikhita, «COVID-SAFE: IoT Based Health Monitoring System using RFID in Pandemic Life», en *2021 IEEE International Conference on RFID Technology and Applications (RFID-TA)*, oct. 2021, pp. 203-206. doi: 10.1109/RFID-TA53372.2021.9617312.

[6] S. Ravali y R. Lakshmi Priya, «Design and Implementation of Smart Hospital using IoT», en *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, abr. 2021, pp. 460-465. doi: 10.1109/ICCMC51019.2021.9418296.

[7] S. Durga, R. Nag, y E. Daniel, «Survey on Machine Learning and Deep Learning Algorithms used in Internet of Things (IoT) Healthcare», en *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, mar. 2019, pp. 1018-1022. doi: 10.1109/ICCMC.2019.8819806.

[8] N. Agarwal, P. Singh, N. Singh, K. K. Singh, y R. Jain, «Machine Learning Applications for IoT Healthcare», en *Machine Learning Approaches for Convergence of IoT and Blockchain*, John Wiley & Sons, Ltd, 2021, pp. 129-144. doi: 10.1002/9781119761884.ch6.

[9] C. M. Mohammed y S. Askar, «Machine Learning for IoT HealthCare Applications: A Review», *International Journal of Science and Business*, vol. 5, n.º 3, pp. 42-51, 2021.

[10] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, y R. X. Gao, «Deep learning and its applications to machine health monitoring», *Mechanical Systems and Signal Processing*, vol. 115, pp. 213-237, ene. 2019, doi: 10.1016/j.ymsp.2018.05.050.

REFERENCIAS

[1] A. Gutte y R. Vadali, «IoT Based Health Monitoring System Using Raspberry Pi», en *2018 Fourth*



Implementación de un Sistema de Gestión de Instrumental y Control de Acceso de Personal. (SiGICAP)

Alejandro Aldo Lorenzut

Grupo de Investigación y Desarrollo en Sistemas Embebidos [1]

EEST N° 6 Albert Thomas

La Plata Buenos Aires Argentina

alelorenzut@hotmail.com

Resumen—El presente proyecto muestra la implementación de un Sistema de Gestión de Instrumental y Control de Acceso de Personal de laboratorio (SiGICAP) basado en tecnología IoT. El mismo consta de registradores ubicados en cada laboratorio enlazados mediante WiFi, protocolo MQTT a un servidor que bajo la herramienta de programación visual Node-Red administra una base de datos SQL como así también la comunicación hacia los registradores y la interfaz gráfica de usuario. El sistema almacena en una base de datos la ubicación de los instrumentos, su estado, la fecha y hora de ingreso, egreso y el permiso de ingreso del operario que lo transporta.

Palabras Clave—IoT, SQL, MQTT, gestión de instrumental, base de datos.

posible resolver de manera sencilla con la forma de trabajo y recursos presentes. Esto implica la necesidad de implementar un sistema de gestión que facilite la tarea y permita establecer un mecanismo confiable y sistemático para el registro, seguimiento y control del equipamiento. En la actualidad el instituto cuenta con una base de datos donde están registrados los instrumentos, utilizando ese insumo como punto de partida la propuesta del IAR es la realización de un dispositivo automático que sea capaz de registrar el movimiento de los usuarios y de los equipos dentro del instituto.

Las funcionalidades del SiGICAP establecidas por el IAR deben ser:

I. INTRODUCCIÓN

En el marco de las prácticas profesionalizantes que deben realizar los alumnos de la especialidad Técnico en Electrónica perteneciente a la Escuela de Educación Secundaria Técnica N° 6 Albert Thomas, se llevó a cabo un vínculo con el Instituto Argentino de Radioastronomía participando como institución oferente. En este contexto el IAR dispuso como proyecto la realización de un Sistema de Gestión de Instrumental quedando el desarrollo del mismo a cargo de la institución educativa, grupo GRIDSE dando lugar de esta manera a que los alumnos de una escuela técnica intervengan, desarrollen y realicen la puesta en funcionamiento del sistema en los entornos formativos propios de la institución para luego llevar a cabo el montaje y carga de datos en las instalaciones del IAR. Surge de ello una excelente oportunidad de poner en práctica la resolución de una situación problemática basada en tecnología actual y con requerimientos específicos que son necesarios como parte de la formación de calidad de nuestros técnicos electrónicos.

II. PROPUESTA A RESOLVER

El departamento de electrónica del IAR, cuenta con un variado listado de instrumentos y equipos para uso en sus laboratorios y de campo, como se indica en la Fig. 1 se encuentra distribuido en diferentes ubicaciones dentro del instituto. La gestión del instrumental requiere atención dedicada, que actualmente no es

- Proceso de registro: Cuando una persona pasa su tarjeta identificatoria junto con la tarjeta identificatoria del instrumento que transporta por el registrador, este leerá la información y buscará los datos correspondientes en un listado interno que cada registrador tendrá precargado. Este listado interno se conformará con una lista del personal y otra lista del instrumental y deberá ser actualizado cada vez que se realicen modificaciones en el listado del personal o del instrumental en la base de datos, incluido el nivel de autorización de acceso.
- Identificación de la persona: Se mostrará en la pantalla del registrador los datos de la persona identificada, como su nombre y apellido.
- Auto completar campos: El registrador completará automáticamente los campos de hora y fecha utilizando la hora y la fecha actual obtenidas de una fuente confiable de tiempo.
- Identificación del edificio: El registrador también completará el nombre del edificio al que se está accediendo, basándose en la identificación del registrador utilizado.
- Confirmación visual: El sistema mostrará una confirmación visual en la pantalla del registrador, como un mensaje pertinente o encendiendo un led si el dispositivo registrador se encuentra en la intemperie.
- Aviso de persona no autorizada: Dentro de la base de datos general se establece niveles de

autorización para entrar en ciertas dependencias del instituto. En caso de que una persona no esté autorizada a entrar a cierta dependencia, el acceso será permitido (se abrirá la puerta), pero el sistema debe enviar una alerta al responsable de ese sector y dejar registro de esa entrada no permitida.



Fig. 1: Instituto Argentino de Radioastronomía.

Los requerimientos establecidos por el IAR son:

- Base de datos para la gestión de instrumental basada en Microsoft Access o alternativa SQL [2].
- Entorno de programación Node-Red [3].
- Sistema basado en un dispositivo SoC ESP 8266 ESP 32 o similar [4].
- Lectora de tarjetas RFID [5].
- Protocolo de comunicación MQTT [6].
- Display indicador de lectura del instrumento o personal.
- Interfaz gráfico destinada a dar alta, baja, modificación y consulta de los equipos, personal y demás datos pertinentes.

III. MATERIALES Y MÉTODO DE TRABAJO

El desarrollo se llevó a cabo durante dos ciclos lectivos (2022 - 2023).

III-A. Primera etapa

Durante el primer año se trabajó con la premisa de realizar un Sistema de Gestión de Instrumental suponiendo condiciones de conectividad de WiFi estables y permanentes entre el servidor y los dispositivos registradores dispuestos en los distintos laboratorios del predio del IAR. No se indicó particularidades sobre la frecuencia de la tarjeta RFID que llevaría asociada el instrumento.

A continuación, se detalla el procedimiento seguido como así también los elementos utilizados en cada etapa del proyecto.

1. Ensayos con el módulo lector de tarjetas TAG Rc522 de 13,56 MHz. Protocolo de comunicación SPI. Lectura de ID. Ensayo de la pantalla OLED de 0.96 pulgadas con protocolo I2C. El dato a mostrar será el nombre del instrumento. Funciona a modo de confirmación del correcto procedimiento de lectura del instrumento.

2. Ensayo de módulo esp8266 integrado en placa NodeMCU, se estableció comunicación mediante WiFi con una aplicación realizada en Node-Red.
3. Se realizó una red ad hoc entre el servidor y el dispositivo terminal. La misma permitió ensayos de comunicación y pruebas de protocolo sin la necesidad de la red propia de la institución educativa.
4. Finalmente, se eligió reemplazar la base de datos original Microsoft Access por una base de datos SQL SQLITE3, ya que ésta es de código abierto y es soportada por una gran cantidad de lenguajes de programación. Con esto se creó una base de datos CRUD (Create, Read, Update, Delete) accesible desde Node-Red mediante su interfaz gráfica.

El sistema representado en la Fig. 2 está formado por un dispositivo registrador en cada uno de los laboratorios, a saber: Cámara anecoica, Sala de control, Laboratorio de electrónica, Sala limpia, Laboratorio criogénico. El registrador lee la tarjeta RFID perteneciente al operador y la tarjeta propia del equipo, envía la información de ambos ID con sus tópicos correspondientes al servidor, éste lo recibe y ubica en la base de datos haciendo la modificación de su nueva ubicación. A continuación, devuelve el dato del nombre del instrumento que es leído en la terminal por el operario a modo de acuse de recibo correcto de la información.

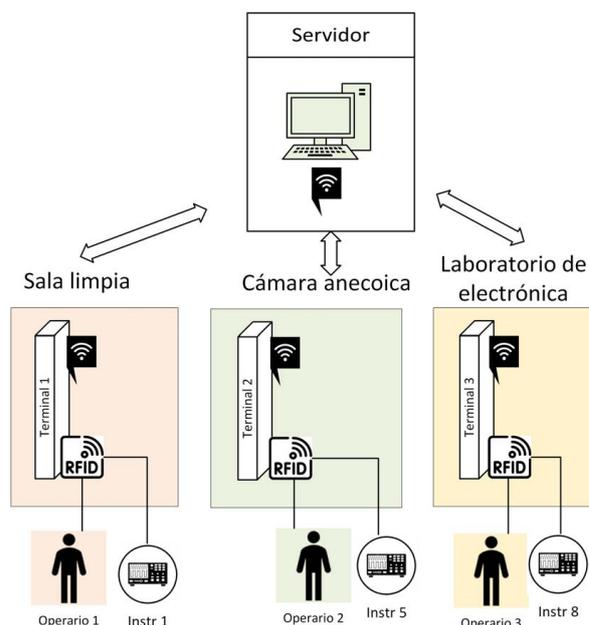


Fig. 2: Diagrama en bloques del SGI.

El servidor, esquematizado en la Fig. 3, se desarrolla en torno al lenguaje de programación Node-Red [3]. El mismo se encarga de gestionar la interfaz gráfica donde el operario en una primera instancia puede registrar los nuevos dispositivos, dar de baja a los mismos, consultar por la ubicación, consultar e ingresar fechas de reparación y estado del mismo. Así también es posible ingresar un nuevo operario y modificar su estado. Lo anterior se realiza en un

motor de base de datos SQLITE3, donde se definen tres tablas:

Tabla Laboratorio, llave primaria ID del laboratorio, campo: Nombre del laboratorio.

Tabla Instrumentos, llave primaria ID del instrumento, campos: Nombre del instrumento, Fecha y hora de ingreso, Estado, Fecha de última reparación.

Tabla Personal, clave primaria ID del personal, campos: Nombre del operario, Cargo.

Desde la plataforma de Node-Red también se gestiona la comunicación por protocolo MQTT introduciendo la IP del Broker mosquitto en la configuración pertinente. Por lo tanto, este sistema es capaz de recibir los datos provenientes de cada terminal ubicadas en los distintos laboratorios y devolver un acuse de recibo o "callback" a la terminal correspondiente donde fue ingresado el instrumento.

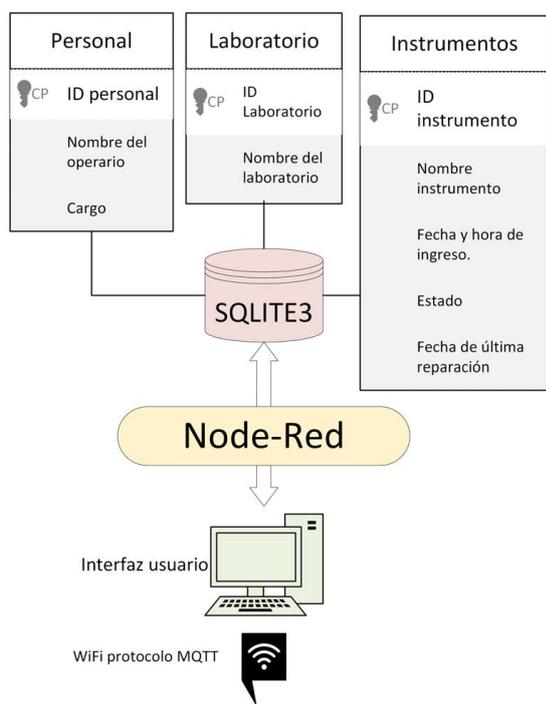


Fig. 3: Diagrama en bloques del servidor.

La terminal, representada en la Fig. 4, está realizada en torno a un dispositivo NodeMCU utilizando el IDE de Arduino para su programación.



Fig. 4: Ensayo del prototipo de la terminal.

Los periféricos asociados, lectora de tarjetas RFID y display OLED, establecen comunicación mediante los protocolos SPI e I2C respectivamente con sus correspondientes librerías.

El desarrollo del software se realizó mediante las funciones: Wifi, Callback, Reconexión, Setup y Loop.

El tamaño de la placa montada es de 16cm * 5.5cm resultando de este modo un dispositivo portátil llegando a la conclusión de alimentarlo mediante una batería recargable mediante entrada USB.

En un primer ensayo del sistema se probó con dos terminales, registrando en una base de datos el nombre del instrumento asociado a la tarjeta RFID, el estado del mismo y el laboratorio correspondiente.



Fig. 5: Terminal montada y funcionando junto con el servidor. (Etapa de prototipo)

Los datos se muestran en la interfaz gráfica de usuario, separado por bloques dividido por nombre de laboratorio (en una primera instancia sólo dos laboratorios). Luego el nombre del instrumento es devuelto a la terminal y presentado en el display como se observa en la Fig. 5.

III-B. Segunda etapa

Los cambios, actualizaciones y nuevas funcionalidades respecto a la primera etapa son: Control de Acceso de Personal en el ingreso de los distintos laboratorios.

Considerar posibles inestabilidades o cortes de la señal de WiFi en el predio del IAR.

Con estas nuevas características resultaron los siguientes cambios:

- El sistema incluirá un dispositivo actuador como un relé o similar destinado a accionar el pestillo de la puerta de acceso.
- Al quedar integrado en Sistema de Gestión de Instrumental con el nuevo Control de Acceso de Personal resulta que el mismo módulo lector de tarjetas RFID debe realizar ambas tareas, por lo tanto, y para continuar con el uso de las tarjetas RFID que ya usaba el personal en el predio del instituto se cambió el módulo lector de tarjetas RC522 de 13.56 MHz por el módulo RDM 6300 frecuencia de 125 KHz. Por el mismo motivo se cambió el protocolo de comunicación entre el microcontrolador y el módulo lector de tarjetas, de I2C a protocolo serie.
- Cada registrador debe contar con una base de datos que es enviada desde el servidor en formato json mediante protocolo MQTT . Los datos son: Nombre de personal permiso de ingreso del personal a los laboratorios, nombre de los instrumentos, permisos de ingreso de los instrumentos a los laboratorios, identificación del registrador con el laboratorio donde se encuentra ubicado y un timestamp para sincronizar a cada registrador.
- Por lo anterior cada registrador debe contar con un RTC, punto fundamental para decidir la migración hacia un microcontronador que tenga este recurso integrado.
- Para los registradores que se encontrarán en la puerta de los laboratorios, se debe utilizar una caja estanca IP65. Para mantener la estanqueidad de la caja se estableció reemplazar la forma de visualizar el estado de ingreso y otros datos pertinentes, por diodos led en lugar de pantalla oled.

Como consecuencia de los nuevos requerimientos, en esta segunda etapa se debieron realizar los siguientes cambios:

	Primera etapa	Segunda etapa
Dispositivo	ESP8266	Raspberry pi pico w
Plataforma	IDE Arduino	VSC micropython
RTC	-	Interno Raspberry
Base de datos	-	MicropyDataBase
Lectora RFID	TAG RC522	RDM 6300
Indicador visual	OLED 0.96"	OLED + LED
Control de pestillo	-	Rele de 5 V

Tabla I: Cambios realizados en el registrador

	Primera etapa	Segunda etapa
Motor de db	SQLITE 3	MY SQL
Número de tablas db	3	7
Sincronización	-	envía timestamp

Tabla II: Cambios realizados en el servidor

IV. ARQUITECTURA FINAL DEL SISTEMA

IV-A. Servidor

Del lado del servidor el cambio relevante está dado en las tablas que componen la base de datos que incluyen permisos de ingreso para usuario e instrumental como se observa en la Fig. 6.

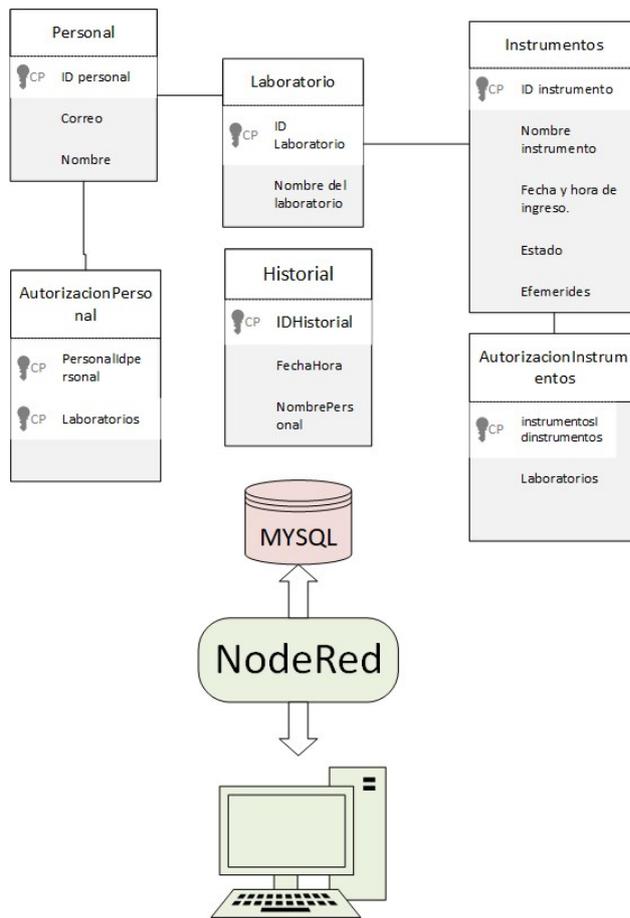


Fig. 6: Diagrama entidad relación de la base de datos MySQL

Desde el interfaz gráfico es posible dar de alta, baja, modificación y consulta de los datos ingresados. En la Fig. 7 se observa la pantalla de modificación de datos de instrumental, una vez aceptado pasa a la siguiente pantalla destinada a ingresar los permisos correspondientes. Mediante el interfaz gráfico es posible además enviar un timestamp para la sincronización de los registradores.

idInstrumento	nombreInstrumento	nombreFabricante	codigoIAR	responsable	procedimiento	deteccionDeFallas	siguienteInstrumento	laboratorio	
1	Null	Null	2023-08-28T03:...		servicio		1	0	
2	Osciloscopio 6...	Tektronix	2023-09-06T03:...	Nuevo	2023-08-28T03:...	nullbig	1	1	
3	Fuente de ali...	Hip	2023-08-28T03:...	Nuevo	2023-08-08T03:...	null	1	0	
4	Oscilador de et...	aveon	2022-04-13T03:...	Usado correcto	2024-04-17T03:...	null	1	0	
5	Fuente de alimentación	Siemens	2023-08-09T03:...	En uso voltim...	2023-08-28T03:...	Reparacion de...	1	1	
6	Osciloscopio	JEP	2023-08-28T03:...	Nuevo	2023-08-02T03:...	null	1	0	
7	Multímetros de...	BrK	2023-08-11T03:...	Nuevo	null	null	1	1	
8	Fuente de ali...	ByK	2023-08-16T03:...	Nuevo	2023-08-15T03:...	null	2	0	
9	Osciloscopio 6...	HP	2023-08-17T03:...	Nuevo	2023-08-17T03:...	no tiene fallas	2	0	
10							5	1	
11	SDC						2	1	
12	FUENTE Tasa	AGH	114-03	JUAN	FANF	ANDRADIF	ANDRADIF...	1	1
13	IKG1						8	1	
14	JIC						4	1	
15	GFJG3H	utru					5	1	
16		KLN1					1	1	

Fig. 7: Pantalla de selección de instrumento para su posterior modificación de datos

El personal encargado de realizar la carga de datos deberá completar los siguientes campos:

Datos a dar de alta Instrumental:

Tarjeta RFID (id), Nombre del instrumento, Nombre del fabricante, Código IAR, Responsable, Procedimiento, Efemerides, Detección de fallas, Autorización ingreso laboratorio.

Datos a dar de alta Usuario:

Tarjeta RFID (id), Nombre y apellido, Correo electrónico, Teléfono, Dirección, Permiso ingreso laboratorio
En total el interfaz gráfico cuenta con nueve pantallas: Alta usuario, Permisos usuario, Alta instrumental, Permiso instrumental, Modificación usuario, Modificación instrumental, Baja usuario, Baja instrumental, Sincronismo dispositivos registradores.

IV-B. Dispositivo registrador

Diagrama en bloques

El diagrama en bloques observado en la Fig.8 muestra los componentes principales del dispositivo registrador.

Según el documento de requerimientos se establece que la tensión de alimentación de los dispositivos registradores es de 5 V. El IAR provee esta tensión en cada sitio donde se ubicaran los dispositivos registradores, por lo tanto solamente se incluye en el circuito eléctrico un filtro de entrada.

El microcontrolador RP2040 dispone de dos núcleos Cortex M0+, con lo cual se implementa la utilización de los mismos de la siguiente manera:

El núcleo cero gestiona la base de datos temporal (alta, baja, modificación y consulta), consulta de base de datos interna, recibe los datos de la lectora de tarjeta RFID, envía los datos pertinentes a la pantalla OLED y los led correspondientes y controla el dispositivo actuador.

El núcleo uno se encarga de la comunicación por WiFi, protocolo MQTT, recibe la base de datos (base de datos interna) del servidor y envía los datos de la base de datos temporal hacia el servidor.

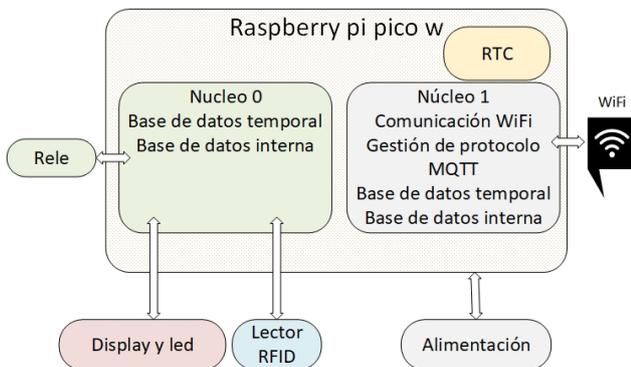


Fig. 8: Diagrama en bloques del registrador.

Los datos enviados desde el servidor hacia los dispositivos registradores son:

Del instrumental:

- id Instrumental
- Nombre del instrumento
- Permiso de ingreso laboratorio

Del usuario:

- id Usuario
- Apellido usuario
- Permiso de ingreso laboratorio

Estos datos, Instrumental y Usuario conforman la base de datos interna del registrador.

La base de datos temporal mantiene los datos de

ingreso de Usuario, Instrumental alarma de ingreso no permitido al laboratorio y timestamp hasta que son enviados al servidor; en ese momento los datos de la tabla temporal son eliminados.

Diagrama de flujo

El diagrama de la Fig.9 indica las tareas que realizan ambos núcleos. El núcleo cero recibe los datos serie del lector de tarjeta RFID, consulta con la base de datos interna si está registrado el id del usuario, el id del instrumental y la condición de permiso de ingreso al laboratorio, en el caso de no estar permitido el ingreso se activa una bandera de alarma. Se almacenan estos datos en la base de datos temporal hasta que son enviados al servidor.

El núcleo uno es destinado a la tarea de gestión de comunicación WiFi protocolo MQTT. Mientras no se establezca comunicación con el servidor los nuevos datos originados por altas en la base de datos temporal serán mantenidos en cola.

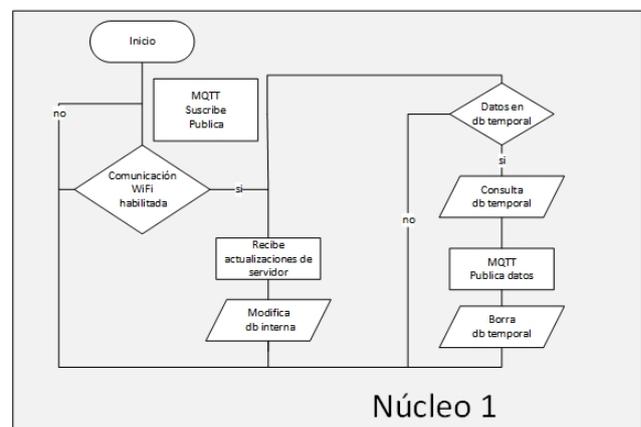
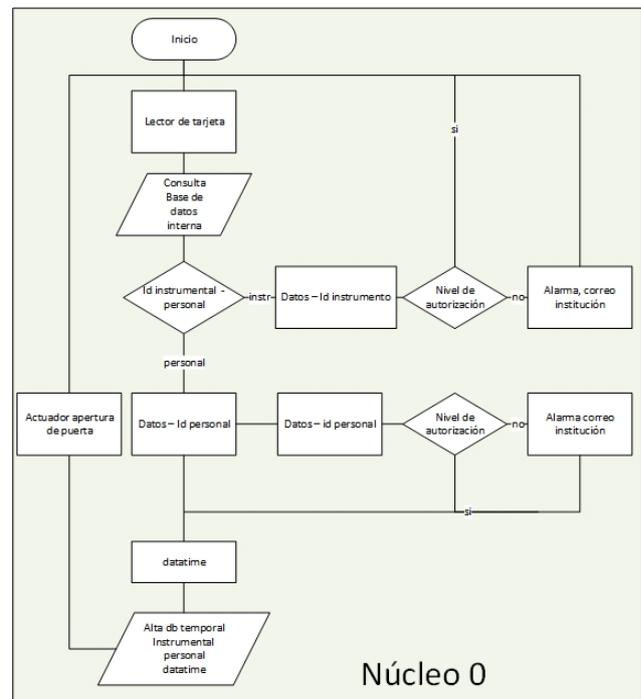


Fig. 9: Diagrama de flujo del dispositivo registrador

Circuito eléctrico

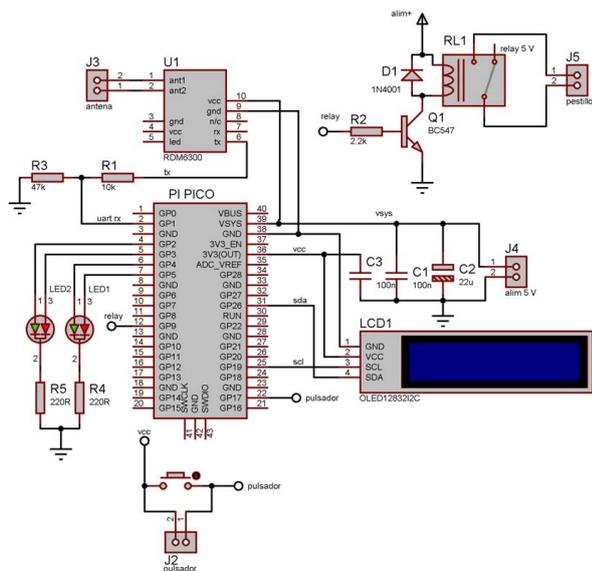


Fig. 10: Circuito eléctrico del registrador

En el circuito eléctrico que muestra la Fig.10 se observa la conexión de los pines GPIO 19 y GPIO 26 comunicación I2C hacia la pantalla OLED. El pin GPIO 1 entrada de datos serie proveniente del módulo RDM6300, los pines GPIO 2, GPIO 3, GPIO 4 y GPIO 5 destinado a controlar dos leds bicolor, el pin GPIO 12 controla el actuador y el pin GPIO 22 configurado como entrada de un pulsador.

Circuito impreso

El diseño del circuito impreso ubica la totalidad de los componentes en una placa de 6cm * 7cm. La selección de los pines GPIO se realizó para mantener la longitud de las pistas lo más cortas posible.

El consumo de corriente fluctúa desde los 50mA (funcionamiento de Raspberry pi pico W y oled sin comunicación WiFi) hasta los 150mA con comunicación WiFi, relé y led encendidos. El módulo Raspberry pi pico w se suelda del lado de las pistas.

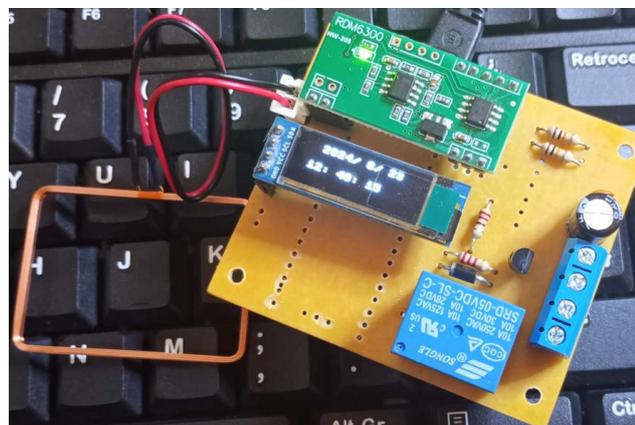


Fig. 11: Circuito registrador (El Raspberry pi pico se encuentra en el lado de abajo)

Se observa que el circuito de la Fig.11 es un prototipo donde la pantalla oled y el módulo lector RDM6300 se ubican mediante zócalos al igual que

el módulo Raspberry. Las borneras ubicadas a la derecha de la fotografía conectan la tensión de la fuente de alimentación de 5V y los contactos del relé hacia el circuito del pestillo.

V. RESULTADOS Y CONCLUSIONES

Se concluye que el sistema realizado según demanda del IAR posee las siguientes características:

- El sistema es escalable tanto en el número de terminales como en las tablas y los campos que debe almacenar su base de datos.
- Utilización de plataformas de código abierto de hardware y software.
- Los dispositivos registradores son autónomos, no necesitan estar conectadas a una computadora o vinculados a través de WiFi para su funcionamiento normal.
- La gestión de usuarios en el servidor se realiza a través de una interfaz gráfica donde facilita el ingreso y actualización de datos de personal e instrumentos.

VI. TRABAJO A FUTURO

Se realizará el diseño de producción de la placa de circuito impreso, esto es: se eliminarán los zócalos y la pantalla OLED y los leds se ubicarán en otra placa conectada mediante cable plano.

Inclusión de un nivel de seguridad TLS.

Se implementará un sistema de bajo consumo tipo Light-sleep.

Se agregará una fuente de alimentación conformada por una batería 18650 de 3000 mAh con un cargador basado en un MCP73871.

Se incluirán nuevos requerimientos y necesidades según pedido de los referentes.

Se realizará el ensayo e implementación del Sistema de gestión de instrumental en las instalaciones del Instituto de radioastronomía.

Luego de esto, en la etapa de producción, se montará la cantidad de terminales necesarias para cubrir cada laboratorio.

VII. AGRADECIMIENTO

Mi más sincero agradecimiento a los alumnos que desean transitar sus prácticas profesionalizantes en el Grupo de Investigación y Desarrollo de Sistemas Embebidos. Así también deseo agradecer al personal del Instituto Argentino de Radioastronomía y muy especialmente a los Ingenieros Elías Fliger y Martín Salibe por la confianza y el compromiso con nuestra institución educativa.

REFERENCIAS

- [1] Aban Sebastian, Acuña Ariel, Acuña Luis, Cajade Juan, Calisaya Aldo, Cavallaro Bruno, De Donato Santiago, Fontana Nicolas, Inga Sebastián, Lopez Eliseo, Ragaini Juan, Repucci Rocco, Rollie Francisco, Salto Tobias, Santillan, Lucas, Tinco Martin, Grupo GRIDSE, 2024. [Online]. Available: <https://eest6albertthomas.com/gridse>
- [2] P. C. Garcia, *SQL fácil*, 1st ed. Gran Via de les Corts Catalanes, 594: Marcombo, 2014, isbn : 978-84-267-2100-6.



- [3] B. R. Japón, *Learn IoT Programming Using Node... by Japón, Bernardo Ron Paperback / softback*, 1st ed. BPB Publications, 2008.
- [4] M. Schawartz, *Internet of Things with ESP8266*, 1st ed. 35 Livery street: Packt, 2016, vol. 1.
- [5] H. A. K. Mohd Helmy, Abd Wahab, *Design and Development of Portable Attendance System using RFID*, 1st ed. BPB Publications, March 2010.
- [6] t. Pulver, *Hands-On Internet of Things with MQTT*, 1st ed. Packt Publishing, 2019, isbn : 9781789345001.

Sistema de Monitoreo y Control con tecnología BT e IoT, utilizando la placa de desarrollo ESP32

Fernando J. Carranza, Francisco M. Cortez Nachar, Roberto M. Murdocca, Sergio F. Hernandez Velazquez
Departamento de Electrónica
Universidad Nacional de San Luis
San Luis, Argentina
e-mail: {fernandocarranza2018, francorteznachar, mmurdocc, sergio.sfhv}@gmail.com

II. DESARROLLO

Resumen - Este trabajo contribuye al conocimiento existente en el área de la IoT al presentar una arquitectura flexible y escalable para la construcción de sistemas de monitoreo y control remoto. Se buscó implementar un sistema de visualización de datos analógicos y digitales, monitoreo de variables físicas específicas y actuadores las cuales deberían poder interactuar con cualquier dispositivo portátil con conexión a Internet utilizando una interfaz gráfica. Esta última, diseñada bajo los principios de usabilidad, presenta una estructura intuitiva y elementos visuales claros.

Palabras clave: IoT, Wi-Fi, Node-Red, MQTT, Bluetooth.

I. INTRODUCCIÓN

La creciente demanda de soluciones IoT ha generado un gran interés en el desarrollo de sistemas de monitoreo y control remoto flexibles y escalables. Este trabajo contribuye a este campo al proponer una arquitectura modular y abierta que puede ser adoptada y extendida por otros investigadores. La interfaz gráfica desarrollada, basada en principios de usabilidad, establece un punto de referencia para el diseño de interfaces en sistemas IoT, facilitando la interacción entre el usuario y el sistema. Los resultados obtenidos en este proyecto abren nuevas líneas de investigación dentro del Departamento de Electrónica, como la integración de algoritmos de aprendizaje automático para la toma de decisiones autónomas y la optimización de recursos energéticos.

El sistema utiliza un microcontrolador ESP32 [1], acompañado de diversos componentes electrónicos, como un sensor de temperatura y humedad, una fotoresistencia, pulsadores, un potenciómetro, un módulo de relés, y elementos para el acondicionamiento de señales.

Para la comunicación mediante Bluetooth, se diseñó una interfaz gráfica en la aplicación móvil Bluetooth Electronics [2], que permite al usuario monitorear datos y controlar dispositivos de forma local. En cuanto a la implementación para IoT, se emplea el protocolo MQTT, utilizando una interfaz gráfica desarrollada en Node-RED [3] para la visualización y gestión de los datos proporcionados por el microcontrolador.

El sistema completo se presenta en la Fig. 1, mediante un diagrama de bloques distribuido de la siguiente manera:

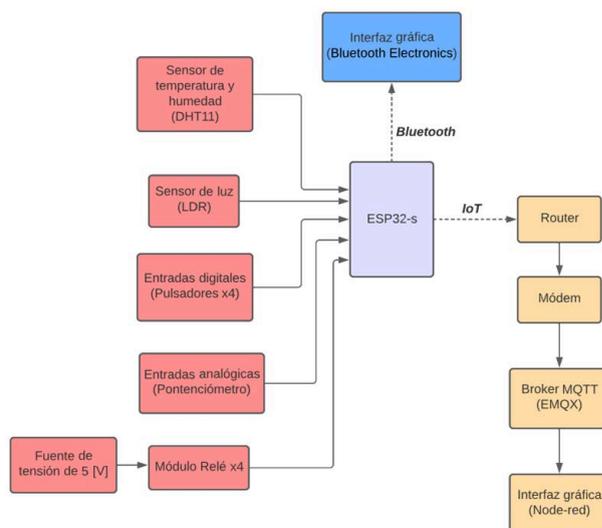


Fig. 1. Diagrama en bloques del sistema

Se tuvieron en cuenta las siguientes restricciones de diseño para asegurar el correcto funcionamiento y eficiencia del sistema:

- Conectividad dual: El sistema debe ser capaz de manejar simultáneamente conexiones Bluetooth y Wi-Fi, sin interferencias.
- Tamaño compacto: Se requiere un diseño que permita integrar todos los componentes en un espacio reducido, para facilitar su implementación en entornos pequeños.
- Compatibilidad con múltiples dispositivos: El sistema debe ser compatible con una variedad de dispositivos (smartphones, tablets, y computadoras) para asegurar una amplia accesibilidad y usabilidad.

III. DISEÑO DEL SISTEMA

A. MICROCONTROLADOR

El sistema integra un microcontrolador ESP32 (Fig.2) fabricado por Expressif Systems [1][4], el cual proporciona capacidades de comunicación Bluetooth y Wi-Fi, lo que permite la implementación simultánea de tecnologías de comunicación local y remota.

A continuación se detallan las características principales del microcontrolador:

- Integra un procesador Tensilica Xtensa 32 bits LX6, capaz de operar a velocidades de hasta 240 MHz.
- Memoria Flash de 4MB.
- Memoria RAM de 520kB.
- Wi-Fi compatible con los estándares 802.11b/g/n/e/i con velocidades de hasta 150 Mbit/s en la banda de 2.4 GHz.
- Soporta Bluetooth V4.2 en sus versiones VR/EDR (Bluetooth clásico) y Bluetooth Low Energy (BLE).
- 34 pines programables
 - 18 canales ADC de 12 bits.
 - 2 canales DAC de 8 bits.

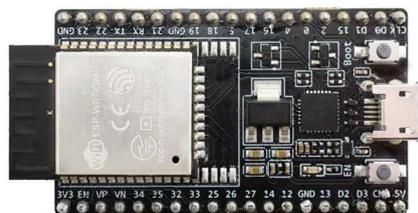


Fig. 2. ESP32.

- 16 canales PWM.
- 3 interfaces UART.
- 3 interfaces SPI.
- 2 interfaces I2C.
- Diferentes modos para gestionar el consumo de energía.

El ESP32 actúa como núcleo de procesamiento, coordinando la adquisición de datos de los sensores y el control de los actuadores.

B. SENSORES

Se detallan a continuación los diversos sensores implementados en el sistema de monitoreo:

- Sensor de temperatura y humedad: El sensor DHT11 [5] mide temperatura y humedad relativa. Este sensor utiliza un termistor y un sensor capacitivo de humedad para proporcionar datos digitales mediante una señal de un solo cable. Sus principales características son:
 - Tensión de operación: 3.3V-5.5V.
 - Corriente de operación: 0.3 mA (medición) y 60 μ A (en reposo).
 - Consumo de potencia máximo de 2.5 mW.
 - Rango de medición de temperatura: 0°C a 50°C \pm 2°C.
 - Rango de medición de humedad: 20% a 90% de humedad relativa (HR) con una precisión de \pm 5% HR.
 - Frecuencia de muestreo: 1 Hz (una vez por segundo).
 - Interfaz de salida: Serial, comunicación unidireccional.

La Fig.3 ilustra la configuración eléctrica empleada para interconectar y operar el sensor.

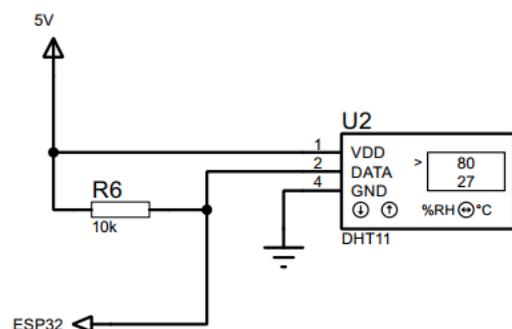


Fig.3. Circuito eléctrico DHT11.

- Fotorresistencia (LDR - Light Dependent Resistor)[6]: Es una resistencia eléctrica la cual varía su valor en función de la cantidad de luz que incide sobre su superficie. Su valor resistivo varía entre $1\text{ M}\Omega$ en la oscuridad y 100Ω con luz brillante. El consumo máximo es de 50 mW a 1W . En la Fig.4 se detalla el conexionado eléctrico, donde a mayor intensidad de luz, menor será la lectura de tensión en el pin del ESP32. Es decir, el ESP32 "verá" una tensión más alta cuando la fotorresistencia esté en la oscuridad y una tensión más baja cuando esté expuesta a la luz.

C. ENTRADAS ANALÓGICAS Y DIGITALES

En esta sección se detallan las funcionalidades de entrada y salida digital utilizadas para el sistema.

- Potenciómetro: varía de 0Ω a $10\text{ k}\Omega$ y se conecta a una entrada analógica del microcontrolador En la Fig.5, se detalla el conexionado eléctrico.
- Pulsadores: son del tipo de conexión

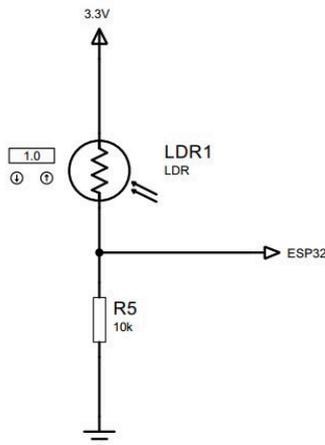


Fig.4. Circuito eléctrico del LDR.

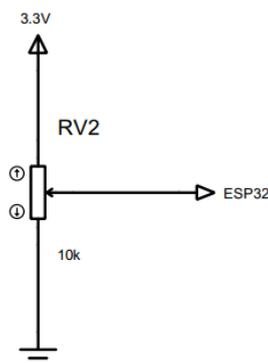


Fig. 5. Circuito eléctrico del potenciómetro.

normalmente abiertos (NA) y se encuentran configurados en la configuración pull-down, lo que garantiza que los pines digitales se mantengan en un estado bajo hasta que se active el pulsador. En la Fig.6, se detalla el conexionado eléctrico.

D. ACTUADORES

El control remoto del sistema se llevará a cabo mediante un 4 Relay Module [7] (Fig.7). Este módulo permite el control de hasta cuatro cargas eléctricas de manera independiente a través de señales TTL (Transistor-Transistor Logic) de 3.3 V a 5 V . Sus principales características son:

- 4 relés de 1 polo y 2 posiciones.
- Tensión de la bobina de 5 V .
- El circuito de control debe proveer una corriente de 15 mA a 20 mA .

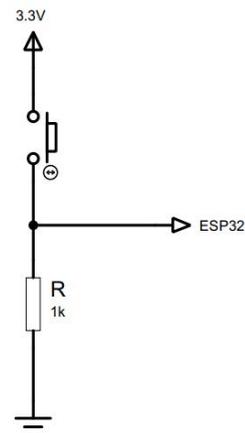


Fig. 6. Circuito eléctrico Pulsadores.

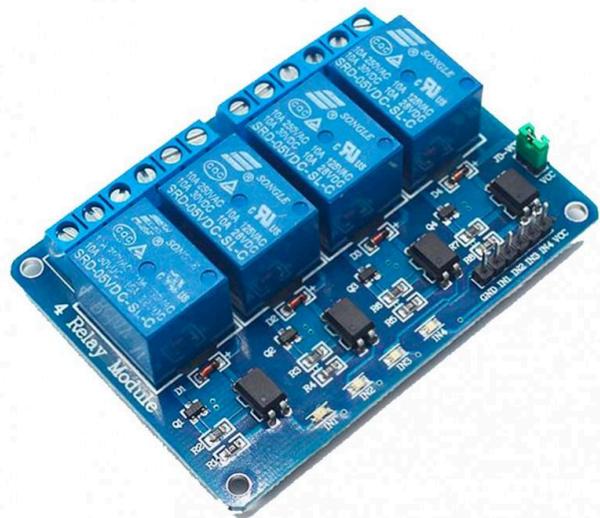


Fig. 7. 4 Relay Module.

La Fig.8 muestra el conexionado eléctrico.

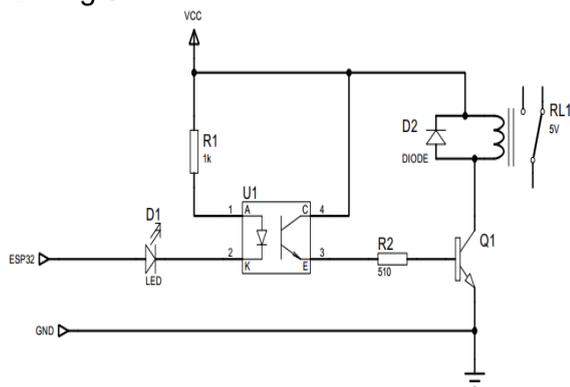


Fig.8. Circuito eléctrico de relé.

IV. INTERFAZ DE MONITOREO Y CONTROL

A. MQTT

El MQTT (Message Queuing Telemetry Transport) [8], es un protocolo de mensajería altamente eficiente y versátil, que ha demostrado ser una herramienta fundamental para el desarrollo de aplicaciones IoT y otras soluciones basadas en la comunicación máquina a máquina. Su arquitectura simple y su capacidad para manejar grandes volúmenes de datos lo convierten en una opción atractiva para investigadores y desarrolladores en diversos campos.

Emplea una arquitectura de publicación-suscripción, lo que permite que los dispositivos (clientes) envíen mensajes (publicaciones) a temas específicos. Otros dispositivos, al suscribirse a estos temas, reciben de manera eficiente los mensajes relevantes. Esta flexibilidad facilita la creación de sistemas escalables y desacoplados, donde los productores y consumidores de datos no necesitan conocerse entre sí.

B. NODE RED

Node-RED [9] es una herramienta de programación visual basada en flujo, desarrollada por IBM y ahora parte de la Fundación OpenJS. Utiliza un modelo de programación de nodos, donde cada nodo realiza una tarea específica y los datos fluyen entre ellos, facilitando la comprensión y manipulación de procesos complejos sin necesidad de conocer detalles del código.

El entorno de Node-RED se ejecuta sobre Node.js y permite a los usuarios crear

aplicaciones a través de un editor Web, conectando nodos en un espacio de trabajo.

C. CONECTIVIDAD

El sistema implementa un esquema de conectividad dual que permite la interacción local y remota mediante Bluetooth y Wi-Fi, optimizando el monitoreo y control de los dispositivos en tiempo real, como se muestra en la Fig.9.

El ESP32 puede conectarse a redes Wi-Fi 802.11b/g/n/e/i, facilitando así la comunicación remota a través de Internet. En este modo, los datos recopilados por los sensores del sistema se envían a la nube mediante el protocolo MQTT, con la ayuda del broker, que actúa como servidor para la transmisión de mensajes entre el ESP32 y el sistema de control remoto.

Para una gestión local y autónoma del sistema de monitoreo y control, se emplea Bluetooth. Esta tecnología inalámbrica permite la comunicación directa entre el ESP32 y un dispositivo móvil, sin requerir una conexión a Internet, brindando movilidad y flexibilidad al usuario.

D. PROGRAMACIÓN

La programación del sistema se dividió en dos partes principales: el desarrollo de la interfaz local mediante Bluetooth Electronics y la implementación remota utilizando Node-RED a través de MQTT. Ambas interfaces permiten al usuario monitorear los datos de los sensores y controlar los actuadores en tiempo real, proporcionando diferentes niveles de interacción, ya sea local o remota.

Para la comunicación local, se diseñó una interfaz gráfica en la aplicación móvil Bluetooth Electronics [1], que permite al usuario visualizar datos obtenidos de los sensores de temperatura, humedad, y luminosidad, además de interactuar con las salidas digitales y entradas analógicas del sistema.

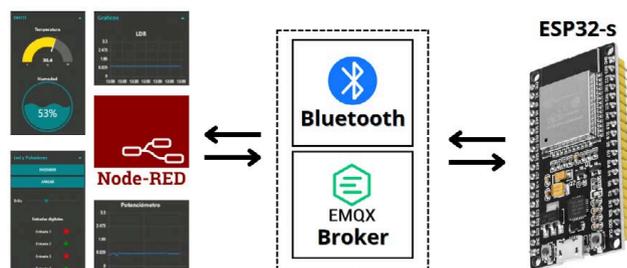


Fig.9. Esquema de conectividad.

La programación del ESP32 se realizó utilizando Arduino IDE [10], integrando librerías específicas para la comunicación Bluetooth como BluetoothSerial, esp_bt_main y esp_bt_device.

Para la implementación remota del sistema y la comunicación con el broker EMQX [11], se utilizaron las librerías WiFi.h y PubSubClient.h en el Arduino IDE, permitiendo que el ESP32 se conecte a la red Wi-Fi y publique los datos de los sensores en el servidor MQTT. Estas librerías facilitan la comunicación eficiente del ESP32 con el broker, permitiendo la transmisión en tiempo real de los datos de los sensores a la plataforma Node-RED.

En la programación en Node-RED (Fig.10) se incluyeron los nodos de suscripción y publicación en el broker MQTT, así como la configuración de nodos para la interfaz gráfica que permiten monitorear los datos y controlar las salidas del sistema.

E. INTERFAZ LOCAL

Se implementa a través de la aplicación móvil Bluetooth Electronics [1], esto permitirá la conexión directa con el ESP32.

La interfaz gráfica (Fig.11), se explica a continuación:

- Control de los relés (R1, R2, R3, R4): permiten al usuario activar (botón verde) o desactivar (botón rojo) los cuatro relés del sistema. El indicador de color azul muestra cuando un relé está activo.
- Indicadores de temperatura y humedad: se muestra la temperatura (en rojo) y la humedad (en azul). Ambas se

representan tanto numéricamente como en formato gráfico (termómetro vertical para la temperatura y gráfico de nivel para la humedad).

- Control deslizante del brillo del LED: situado justo debajo del indicador de temperatura, permite al usuario ajustar la intensidad de luz del LED.
- Indicadores LDR y POTENC.: se observa una gráfica que muestra la variación de la tensión registrada por el LDR y el potenciómetro en tiempo real. Además, se facilitan las lecturas numéricas actuales de ambos sensores.
- Indicadores P1, P2, P3 y P4: muestran el estado de las entradas digitales, representados con color naranja cuando la entrada está activa.

F. INTERFAZ REMOTA

Desarrollada en Node-RED, proporciona una visualización intuitiva del panel de monitoreo y control del sistema (Fig.12).

A continuación se detalla la estructura de la mismas:

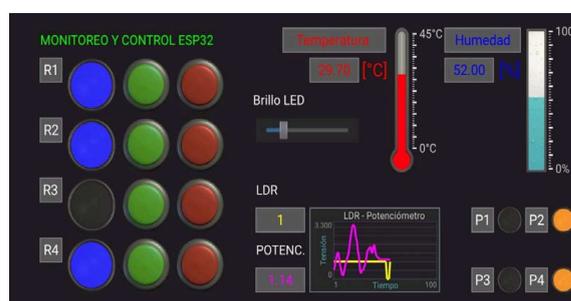


Fig.11. Interfaz local.

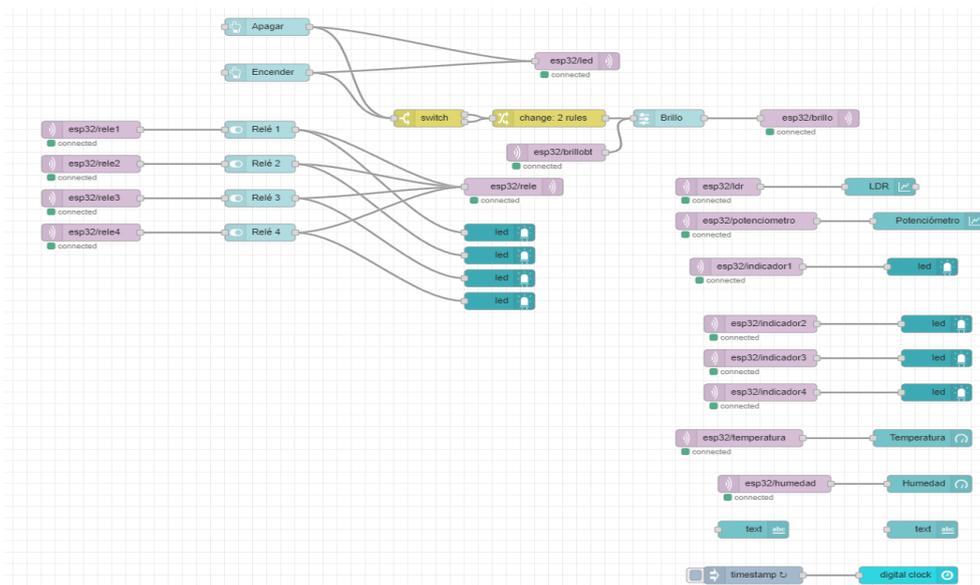


Fig.10 Programación de Node-RED.



Fig.12. Interfaz remota.

- Panel de relés: Control mediante interruptores, cada uno con su indicador on/off.
- Panel de LED y Pulsadores: Incluye un botón para encender y otro para apagar un LED, junto con un control deslizante que ajusta el nivel de brillo. También se muestran los estados de cuatro entradas digitales, representadas por indicadores luminosos.
- Panel de DHT11: Presenta un medidor analógico para la temperatura en grados Celsius y un indicador circular para la humedad relativa.
- Panel de Gráficos: Muestra los datos en tiempo real de la intensidad de luz captada por el sensor LDR y la variación de la resistencia del potenciómetro.

V. RESULTADOS

El prototipo físico de diseño puede observarse en la Fig.13. Para el mismo se lograron los siguientes apartados:

Se logró implementar y operar de manera efectiva el sistema utilizando Wi-Fi y Bluetooth de manera simultánea. Durante las pruebas, identificamos una oportunidad para optimizar el rendimiento del sistema al observar una ligera latencia en la transmisión de datos entre las diferentes plataformas. Este hallazgo nos permitirá investigar a fondo factores como la calidad de la señal, la distancia a los puntos de acceso, la congestión de red y las

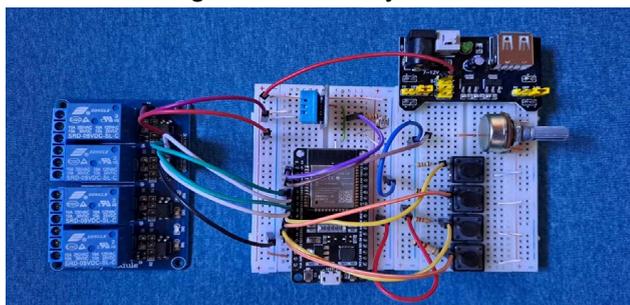


Fig. 13. Prototipo del sistema.

características del hardware utilizado para implementar soluciones que reduzcan este inconveniente y mejoren la experiencia del usuario.

Estos resultados destacan la funcionalidad básica y la interconexión efectiva lograda, mostrando el potencial del sistema en entornos donde se requiere comunicación dual, a pesar de algunas limitaciones observadas durante la fase de pruebas.

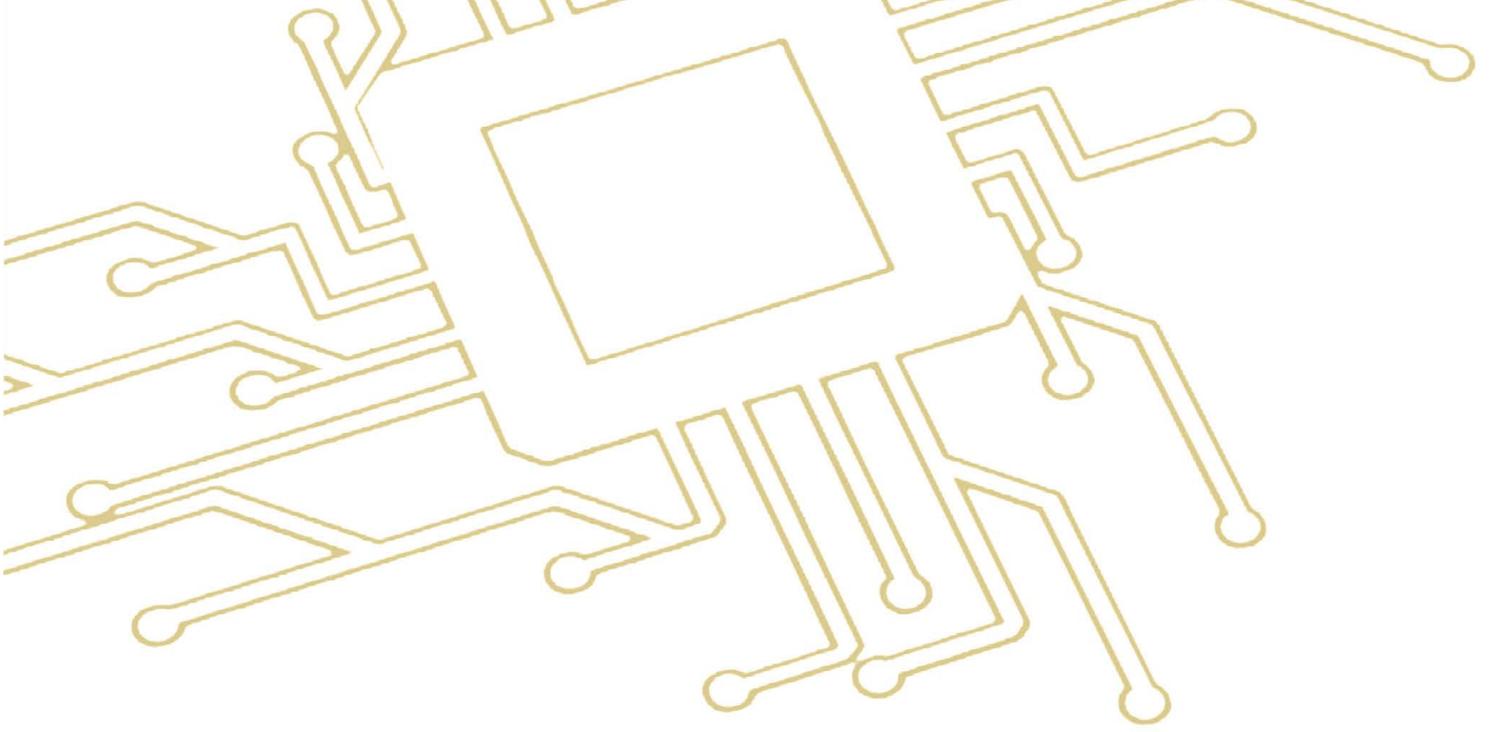
En la Fig.14, se proporciona un código QR que redirige a un video en YouTube. Este video permite observar en detalle el funcionamiento del prototipo en tiempo real, mostrando cómo el sistema interactúa a través de sus diferentes componentes y tecnologías de comunicación, como WiFi y Bluetooth.



Fig.14. QR Video funcionamiento.

REFERENCIAS

- [1] Espressif Systems (Shanghai) Co., ESP32 Series Datasheet, Version 4.5, February, 2024.
- [2] Keelsoft Electronics. Bluetooth Electronics. Obtenido de <https://www.keuwl.com/electronics.html> (Accedido 03/2023)
- [3] Node-RED. Programación de bajo código para aplicaciones basadas en eventos. Obtenido de <https://nodered.org/> (Accedido 03/2023)
- [4] Espressif Systems (Shanghai) Co., ESP32 Technical Reference Manual, Version 5.1, April, 2024.
- [5] Sensor DHT11. Product documentation. Obtenido de <https://mkelectronica.com/producto/sensor-temperatura-humedad/> (Accedido 03/2023)
- [6] MecatrónicaLATAM. Resistencia LDR. Obtenido de <https://www.mecatronicalatam.com/es/tutoriales/sensores/sensor-de-luz/ldr/> (Accedido 03/2023)
- [7] Profe Tolocka. Modulo de Relés. Obtenido de <https://www.profetolocka.com.ar/2015/05/09/modulo-de-4-rel-es-para-arduino/> (Accedido 03/2023)
- [8] Amazon Web Service, ¿Que es MQTT?, 2023, Disponible en <https://aws.amazon.com/es/what-is/mqtt/> (Accedido 03/2024)
- [9] Node-RED. Sobre Node-RED. Obtenido de <https://nodered.org/about/> (Accedido 03/2023)
- [10] Arduino IDE. Software. Obtenido de <https://www.arduino.cc/en/software> (Accedido 03/2024)
- [11] .EMQX. Disponible en <https://www.emqx.com/en> (Accedido 03/2023)



Eje Temático

Aplicaciones en Robótica



Algoritmo de control y observador hiperbólico.

1st José Guadalupe Chapa Huerta

Facultad de ciencias de la electrónica
Benemérita Universidad Autónoma de Puebla
Puebla, México
jose.chapa@alumno.buap.mx

2nd Fernando Reyes Cortés

Departamento de robótica y control de la FCE
Benemérita Universidad Autónoma de Puebla
Puebla, México
<https://orcid.org/0000-0001-5200-7632>

Abstract—En este trabajo se presenta el desarrollo de un algoritmo de control de tipo hiperbólico en el que se utiliza como estimador de velocidad un observador del mismo tipo. Se presentan resultados experimentales y de simulación para mostrar la efectividad de los esquemas propuestos.

Index Terms—Algoritmos de control, observadores, control de posición.

I. INTRODUCTION

En el control de posición (regulación) para robots manipuladores se tiene como finalidad colocar el efector final en una posición deseada constante en el tiempo, partiendo de cualquier condición inicial $[q(0), \dot{q}(0)] \in \mathbb{R}^{2 \times n}$ esto se obtiene utilizando un regulador $\tau \in \mathbb{R}^n$, dicho regulador debe diseñarse de tal forma que la velocidad de movimiento $\dot{q}(t) \in \mathbb{R}^n$ y el error de posición $\tilde{q} \in \mathbb{R}^n : \tilde{q} = q_d - q(t)$ converjan asintóticamente a cero $\forall t \geq 0$ [1], es decir:

$$\lim_{t \rightarrow \infty} \begin{bmatrix} \tilde{q}(t) \\ \dot{\tilde{q}}(t) \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{2 \times n} \forall t \geq 0 \quad (1)$$

La propuesta de la forma que debería tener dicho regulador al utilizar el método de moldeo de energía es:

$$\tau = \nabla U_a(K_p, \tilde{q}) - f_v(K_v, \dot{\tilde{q}}) + g(q) \quad (2)$$

En la que:

- $U_a(K_p, \tilde{q}) > 0; \nabla U_a^T(K_p, \tilde{q})\tilde{q} > 0$ es la energía potencial artificial; en la que $U_a(K_p, \tilde{q}) = \mathbf{0} \iff \tilde{q} = \mathbf{0} \in \mathbb{R}^n$.
- $\nabla U_a(K_p, \tilde{q}) = \frac{\partial}{\partial \tilde{q}} U_a(K_p, \tilde{q})$.
- $K_p \in \mathbb{R}^{n \times n}$ es la ganancia proporcional, matriz diagonal definida positiva.
- $K_v \in \mathbb{R}^{n \times n}$ es la ganancia derivativa, matriz definida positiva.
- $f_v(K_v, \dot{\tilde{q}})$ es una función disipativa: $\dot{\tilde{q}}^T f_v(K_v, \dot{\tilde{q}}) = \mathbf{0} \iff \dot{\tilde{q}} = \mathbf{0} \in \mathbb{R}^n$.

Ahora al tener en cuenta la forma del modelo dinámico de n grados de libertad que tiene la forma [2]:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + f_f(\dot{q}, f_e) \quad (3)$$

donde:

- $q = [q_1, q_2, \dots, q_n]^T \in \mathbb{R}^{n \times 1}$ representa el vector de posiciones articulares o coordenadas generalizadas.
- $\dot{q} = [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n]^T \in \mathbb{R}^{n \times 1}$ es el vector de velocidades articulares.
- $\tau = [\tau_1, \tau_2, \dots, \tau_n]^T \in \mathbb{R}^{n \times 1}$ es el vector de torque o pares aplicados a los servomotores.
- $M(q) \in \mathbb{R}^{n \times n}$ es la matriz de inercia, simétrica y definida positiva.
- $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ es la matriz de fuerzas centrípetas y de Coriolis.
- $g(q) \in \mathbb{R}^n$ es el vector de fuerzas o pares gravitacionales
- $f_f(\dot{q}, f_e) \in \mathbb{R}^n$ es el vector de pares de fricción (viscosa, Coulomb y estática)

Puede notarse que uno de los elementos que esta presente tanto en el algoritmo de control como en el modelo dinámico es la velocidad \dot{q} , el control de posición de robots manipuladores debe considerar que en la mayoría de los casos solo se tiene acceso a la información que representa las posiciones articulares q por lo cual la variable de estado \dot{q} se desconoce.

Al no tener acceso a dicha variable de estado puede verse entonces la necesidad de estimar dicho elemento, para ello puede utilizarse una estructura matemática cuyo propósito es como se menciona anteriormente estimar con un alto grado de fiabilidad un parámetro desconocido.

Por ejemplo, considere el siguiente sistema dinámico escalar lineal e invariante en el tiempo de primer orden:

$$\begin{aligned}\dot{x} &= -ax + bu \\ y &= cx\end{aligned}\quad (4)$$

en el que:

- $x \in \mathbb{R}$ es la variable de estado del sistema.
- $\dot{x} = \frac{dx}{dt} \in \mathbb{R}$ es la derivada temporal del estado del sistema.
- $a, b, c \in \mathbb{R}_+$ son parametros del sistema .
- $u \in \mathbb{R}$ es la entrada del sistema.
- $y \in \mathbb{R}$ es la respuesta del sistema.

Ahora al suponer que se desea estimar el estado y que para ello se utiliza un observador, solo falta elegir el tipo de observador, ya que existen varios tipos, en este caso se utiliza para ejemplificar la estructura de este tipo de elementos una estructura conocida como observador asintótico de Luenberger en su versión escalar que tiene la forma:

$$\begin{aligned}\dot{\hat{x}} &= -a\hat{x} + bu + l(y - \hat{y}) \\ y &= c\hat{x}\end{aligned}\quad (5)$$

El propósito de un observador es reproducir con un alto grado de fiabilidad un elemento desconocido con lo cual el error de convergencia \hat{x} definido como la diferencia entre el estado de la planta (elemento a estimar) $x \in \mathbb{R}^n$ y el vector estimado $\hat{x} \in \mathbb{R}^n$ tienda asintóticamente a cero, conforme el tiempo evoluciona.

Existen formas de comprobar que lo descrito anteriormente se cumple y el método utilizado en este documento para ello es el segundo método de Lyapunov, dicho método se utiliza no solo para la comprobación del correcto funcionamiento teórico de un observador, también se utiliza para la comprobación del algoritmo de control.

Dicho método consiste en proponer una función de energía artificial tal que $V(x) > 0$ es decir definida positiva, $V(0) = 0$ y cuya derivada debe cumplir que $\dot{V}(x) < 0$ o definida negativa, adicionalmente también debe comprobarse la existencia y unicidad del punto de equilibrio en $x = 0$.

II. PLANTEAMIENTO DEL PROBLEMA

Se desea diseñar un regulador hiperbólico asintótico o algoritmo de control asintótico hiperbólico para el control de posición el cual debe cumplir que, el error de posición tienda a cero conforme el tiempo evoluciona, en dicho algoritmo debe incluirse el uso de un estimador de velocidad, el cual debe ser asintótico y del

tipo hiperbólico, cumpliendo que el error entre el estado estimado y el vector a estimar tienda a cero conforme el tiempo evoluciona.

Con la información anterior se plantea entonces:

- Comprobar la existencia y unicidad del punto de equilibrio del algoritmo de control de posición hiperbólico propuesto.
- Comprobar que el error de posición y velocidad tiendan asintóticamente a cero.
- Comprobar la existencia y unicidad del punto de equilibrio del algoritmo de estimación de velocidad asintótico hiperbólico u observador asintótico hiperbólico propuesto para el caso de estudio vectorial.
- Comprobar que el error de estimación tienda asintóticamente a cero para el caso de estudio vectorial.
- Realizar la simulación correspondiente donde se compruebe que el error de estimación tienda a cero para el caso de estudio vectorial.

III. SOLUCIÓN DEL PROBLEMA

A. Comprobar la existencia y unicidad del punto de equilibrio del algoritmo de control de posición hiperbólico propuesto.

Sea entonces el algoritmo de control propuesto con compensación de gravedad para un robot manipulador de n grados de libertad:

$$K_p o - K_v p + g(q) \quad (6)$$

En el que:

- $K_p \in \mathbb{R}_+^{n \times n}$ es una matriz diagonal definida positiva que representa la ganancia proporcional
- $K_v \in \mathbb{R}_+^{n \times n}$ es una matriz diagonal definida positiva que representa la ganancia derivativa
- $o \in \mathbb{R}_+^{n \times 1}$ es un vector de parámetros definido como:
$$o = \left[\frac{\sinh(\tilde{q}_1)}{1 + \cosh(\tilde{q}_1)}, \frac{\sinh(\tilde{q}_2)}{1 + \cosh(\tilde{q}_2)}, \dots, \frac{\sinh(\tilde{q}_n)}{1 + \cosh(\tilde{q}_n)} \right]^T$$
- $p \in \mathbb{R}_+^{n \times 1}$ es un vector de parámetros definido como:
$$p = \left[\frac{\sinh(\tilde{q}_1)}{1 + \cosh(\tilde{q}_1)}, \frac{\sinh(\tilde{q}_2)}{1 + \cosh(\tilde{q}_2)}, \dots, \frac{\sinh(\tilde{q}_n)}{1 + \cosh(\tilde{q}_n)} \right]^T$$
- $g(q) \in \mathbb{R}^n$ es el vector de fuerzas o pares gravitacionales.

Al aplicar el algoritmo de control (6) en el modelo dinámico (3) se obtiene:

$$\begin{aligned}\tau &= M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) + f_f(\dot{\mathbf{q}}, \mathbf{f}_e) \\ \tau &= K_p \mathbf{o} - K_v \mathbf{p} + g(\mathbf{q}) \\ K_p \mathbf{o} - K_v \mathbf{p} + g(\mathbf{q}) &= M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) \\ &+ f_f(\dot{\mathbf{q}}, \mathbf{f}_e)\end{aligned}\quad (7)$$

Al despejar para $\ddot{\mathbf{q}}$ en (7) se obtiene:

$$\ddot{\mathbf{q}} = M^{-1}(\mathbf{q}) [K_p \mathbf{o} - K_v \mathbf{p} - C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - B\dot{\mathbf{q}}] \quad (8)$$

Ahora al definir el error de posición y su derivada se obtiene:

$$\begin{aligned}\tilde{\mathbf{q}} &= \mathbf{q}_d - \mathbf{q} \\ \dot{\tilde{\mathbf{q}}} &= -\dot{\mathbf{q}}\end{aligned}\quad (9)$$

Con estos elementos se puede expresar entonces la ecuación en lazo cerrado del modelo dinámico incluyendo la estructura del algoritmo de control propuesto la cual esta definida como [3]:

$$\frac{d}{dt} \begin{bmatrix} \tilde{\mathbf{q}} \\ \dot{\tilde{\mathbf{q}}} \end{bmatrix} = \begin{bmatrix} -\dot{\tilde{\mathbf{q}}} \\ M^{-1}(\mathbf{q}_d - \tilde{\mathbf{q}}) [K_p \mathbf{o} - K_v \mathbf{p} - C(\mathbf{q}_d - \tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}})\dot{\tilde{\mathbf{q}}} - B\dot{\tilde{\mathbf{q}}}] \end{bmatrix} \quad (10)$$

Al expresar la ecuación en lazo cerrado de dicha forma se obtiene un modelo dinámico autónomo (10), en el cual para demostrar la existencia y unicidad del punto de equilibrio se tiene que:

Con $\begin{bmatrix} \tilde{\mathbf{q}} \\ \dot{\tilde{\mathbf{q}}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$ siendo el punto de equilibrio a demostrar se tiene que en (10) $\dot{\tilde{\mathbf{q}}} = \mathbf{0} \iff -\dot{\tilde{\mathbf{q}}} = \mathbf{0}$ entonces $\dot{\tilde{\mathbf{q}}} = \mathbf{0}$.

Ahora se debe comprobar que $M^{-1}(\mathbf{q}_d - \tilde{\mathbf{q}}) [K_p \mathbf{o} - K_v \mathbf{p} - C(\mathbf{q}_d - \tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}})\dot{\tilde{\mathbf{q}}} - B\dot{\tilde{\mathbf{q}}}] = \mathbf{0}$ debido a que $M(\mathbf{q}_d - \tilde{\mathbf{q}})$ es una matriz definida positiva eso garantiza que su inversa existe y es definida positiva por lo cual $M^{-1}(\mathbf{q}_d - \tilde{\mathbf{q}})$ no es una matriz de ceros entonces el segundo miembro en la ecuación debe ser igual a cero para comprobar la existencia y unicidad del punto de equilibrio, es decir $[K_p \mathbf{o} - K_v \mathbf{p} - C(\mathbf{q}_d - \tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}})\dot{\tilde{\mathbf{q}}} - B\dot{\tilde{\mathbf{q}}}] = \mathbf{0}$.

Al desarrollar lo anterior y utilizar la propiedad de $C(\mathbf{q}_d - \tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}} = \mathbf{0}) = \mathbf{0} \in \mathbb{R}^{n \times n}$ para todo $\mathbf{q} \in \mathbb{R}^n$ y adicionalmente con $\dot{\tilde{\mathbf{q}}} = \mathbf{0}$ en $C(\mathbf{q}_d - \tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}})\dot{\tilde{\mathbf{q}}}$ se garantiza que $C(\mathbf{q}_d - \tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}})\dot{\tilde{\mathbf{q}}} = \mathbf{0}$ con todo lo anterior se obtiene:

$$K_p \mathbf{o} - K_v \mathbf{p} - B\dot{\mathbf{q}} = 0 \quad (11)$$

Al Despejar para $\dot{\mathbf{q}}$ y con B^{-1} se obtiene:

$$\dot{\mathbf{q}} = B^{-1} [K_v \mathbf{o} - K_p \mathbf{p}] \quad (12)$$

Dicha igualdad solo se cumple para $\dot{\mathbf{q}} = \mathbf{0}$ por lo cual el punto de equilibrio para $\begin{bmatrix} \tilde{\mathbf{q}} \\ \dot{\tilde{\mathbf{q}}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$ existe y es único.

B. Comprobar que el error de posición y velocidad tiendan asintóticamente a cero.

Ahora para comprobar que el error de posición y la velocidad tiendan asintóticamente a cero se utiliza el segundo método de estabilidad en el sentido de Lyapunov, para ello se procede a proponer una función candidata de Lyapunov de la forma:

$$V(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) = \frac{1}{2} \dot{\tilde{\mathbf{q}}}^T M(\mathbf{q}_d - \tilde{\mathbf{q}}) \dot{\tilde{\mathbf{q}}} + \mathbf{h}^T K_p \tilde{\mathbf{q}} \quad (13)$$

En la que

• $\mathbf{h} \in \mathbb{R}_+^{n \times 1}$ es un vector definido como:

$$\mathbf{h} = \begin{bmatrix} \sqrt{\ln(1 + \cosh(\tilde{q}_1)) - \ln(2)} \\ \sqrt{\ln(1 + \cosh(\tilde{q}_2)) - \ln(2)} \\ \vdots \\ \sqrt{\ln(1 + \cosh(\tilde{q}_n)) - \ln(2)} \end{bmatrix}^T$$

En la cual al cumplirse la siguiente propiedad : $\mathbf{x}^T Q \mathbf{x} > 0 \iff Q > 0$ la función propuesta es entonces definida positiva.

Al obtener la derivada de la función propuesta se tiene:

$$\begin{aligned}\dot{V}(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) &= \dot{\tilde{\mathbf{q}}}^T M(\mathbf{q}_d - \tilde{\mathbf{q}}) \ddot{\tilde{\mathbf{q}}} + \frac{1}{2} \dot{\tilde{\mathbf{q}}}^T \dot{M}(\mathbf{q}_d - \tilde{\mathbf{q}}) \dot{\tilde{\mathbf{q}}} + 2\mathbf{h}^T K_p \dot{\tilde{\mathbf{q}}} \\ &+ \mathbf{h}^T \dot{K}_p \tilde{\mathbf{q}}\end{aligned}\quad (14)$$

Al desarrollar (14) se obtiene:

$$\begin{aligned}\dot{V}(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) &= \dot{\tilde{\mathbf{q}}}^T M(\mathbf{q}_d - \tilde{\mathbf{q}}) \cdot \\ &[M^{-1}(\mathbf{q}_d - \tilde{\mathbf{q}}) [K_p \mathbf{o} - K_v \mathbf{p} - C(\mathbf{q}_d - \tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}})\dot{\tilde{\mathbf{q}}} - B\dot{\tilde{\mathbf{q}}}] + \\ &\frac{1}{2} \dot{\tilde{\mathbf{q}}}^T \dot{M}(\mathbf{q}_d - \tilde{\mathbf{q}}) \dot{\tilde{\mathbf{q}}} + 2\mathbf{h}^T K_p \dot{\tilde{\mathbf{q}}}\end{aligned}\quad (15)$$

Ahora al derivar con respecto al tiempo el último termino de (15) es decir \mathbf{h} se obtiene:

$$2\mathbf{h}^T K_p \dot{\mathbf{h}} = 2 \cdot \begin{bmatrix} \sqrt{\ln(1 + \cosh(\tilde{q}_1)) - \ln(2)} \\ \sqrt{\ln(1 + \cosh(\tilde{q}_2)) - \ln(2)} \\ \sqrt{\ln(1 + \cosh(\tilde{q}_n)) - \ln(2)} \end{bmatrix}^T \cdot K_p \cdot \begin{bmatrix} \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{q}_1)) - \ln(2)}} \frac{\sinh(\tilde{q}_1)}{1 + \cosh \tilde{q}_1} \\ \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{q}_2)) - \ln(2)}} \frac{\sinh(\tilde{q}_2)}{1 + \cosh \tilde{q}_2} \\ \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{q}_n)) - \ln(2)}} \frac{\sinh(\tilde{q}_n)}{1 + \cosh \tilde{q}_n} \end{bmatrix} \cdot \dot{\tilde{\mathbf{q}}} \quad (16)$$

Al expresar el vector a la derecha de la matriz K_p como una matriz simétrica de modo que se obtiene:

$$K_p = \begin{bmatrix} a & 0 & \dots & 0 \\ 0 & b & \dots & 0 \\ \vdots & \ddots & c & 0 \\ 0 & \dots & 0 & d \end{bmatrix} \quad (17)$$

En la que

$$\begin{aligned} a &= \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{q}_1)) - \ln(2)}} \frac{\sinh(\tilde{q}_1)}{1 + \cosh \tilde{q}_1} \\ b &= \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{q}_2)) - \ln(2)}} \frac{\sinh(\tilde{q}_2)}{1 + \cosh \tilde{q}_2} \\ c &= \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{q}_3)) - \ln(2)}} \frac{\sinh(\tilde{q}_3)}{1 + \cosh \tilde{q}_3} \\ d &= \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{q}_n)) - \ln(2)}} \frac{\sinh(\tilde{q}_n)}{1 + \cosh \tilde{q}_n} \end{aligned} \quad (18)$$

Como el producto entre matrices simétricas es conmutativo se procede a realizar la siguiente operación

$$2\mathbf{h}^T K_p \dot{\mathbf{h}} = \begin{bmatrix} 2\sqrt{\ln(1 + \cosh(\tilde{q}_1)) - \ln(2)} \\ 2\sqrt{\ln(1 + \cosh(\tilde{q}_2)) - \ln(2)} \\ 2\sqrt{\ln(1 + \cosh(\tilde{q}_n)) - \ln(2)} \end{bmatrix}^T \cdot \begin{bmatrix} a & 0 & \dots & 0 \\ 0 & b & \dots & 0 \\ \vdots & \ddots & c & 0 \\ 0 & \dots & 0 & d \end{bmatrix} \cdot \begin{bmatrix} k_{11} & 0 & \dots & 0 \\ 0 & k_{22} & \dots & 0 \\ \vdots & \ddots & k_{33} & 0 \\ 0 & \dots & 0 & k_{nn} \end{bmatrix} \cdot \dot{\tilde{\mathbf{q}}} \quad (19)$$

Al realizar el producto de las dos matrices se obtiene:

$$2\mathbf{h}^T K_p \dot{\mathbf{h}} = \begin{bmatrix} \frac{\sinh(\tilde{q}_1)}{1 + \cosh \tilde{q}_1} & \frac{\sinh(\tilde{q}_2)}{1 + \cosh \tilde{q}_2} & \dots & \frac{\sinh(\tilde{q}_n)}{1 + \cosh \tilde{q}_n} \\ k_{11} & 0 & \dots & 0 \\ 0 & k_{22} & \dots & 0 \\ \vdots & \ddots & k_{33} & 0 \\ 0 & \dots & 0 & k_{nn} \end{bmatrix} \cdot \dot{\tilde{\mathbf{q}}} \quad (20)$$

Como K_p es una matriz diagonal y al utilizar $\mathbf{x}^T \mathbf{y} = \mathbf{y}^T \mathbf{x}$ junto con el vector \mathbf{o} y con $\dot{\tilde{\mathbf{q}}} = -\dot{\tilde{\mathbf{q}}}$ en (15) se obtiene: $2\mathbf{h}^T K_p \dot{\mathbf{h}} = -\dot{\tilde{\mathbf{q}}}^T K_p \mathbf{o}$.

Ahora al desarrollar (10) se obtiene:

$$\begin{aligned} \dot{V}(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) &= \dot{\tilde{\mathbf{q}}}^T K_p \mathbf{o} - \dot{\tilde{\mathbf{q}}}^T K_v \mathbf{p} - \dot{\tilde{\mathbf{q}}}^T C(\mathbf{q}_d - \tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) \dot{\tilde{\mathbf{q}}} - \\ &\dot{\tilde{\mathbf{q}}}^T B \dot{\tilde{\mathbf{q}}} + \frac{1}{2} \dot{\tilde{\mathbf{q}}}^T \dot{M}(\mathbf{q}_d - \tilde{\mathbf{q}}) \dot{\tilde{\mathbf{q}}} - \dot{\tilde{\mathbf{q}}}^T K_p \mathbf{o} \end{aligned} \quad (21)$$

Al eliminar términos y con la propiedad $\frac{1}{2} \dot{\tilde{\mathbf{q}}}^T [\dot{M}(\mathbf{q}) - 2C(\mathbf{q}, \dot{\tilde{\mathbf{q}}})] = 0$ se obtiene:

$$\dot{V}(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) = -\dot{\tilde{\mathbf{q}}}^T K_v \mathbf{p} - \dot{\tilde{\mathbf{q}}}^T B \dot{\tilde{\mathbf{q}}} \leq 0 \quad (22)$$

Puesto que B es una matriz diagonal definida positiva y puesto que \mathbf{p} es una función vectorial impar, además $K_v > 0$: $\dot{\tilde{\mathbf{q}}}^T K_v \mathbf{p} > 0$ entonces $\dot{V}(\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}})$ es una función semidefinida negativa lo cual demuestra la estabilidad del punto de equilibrio.

C. Comprobar la existencia y unicidad del punto de equilibrio del algoritmo de estimación de velocidad asintótico hiperbólico u observador asintótico hiperbólico propuesto para el caso de estudio vectorial.

Para comprobar que el punto de equilibrio existe se inicia describiendo el sistema con el estado a ser estimado, entonces considere el siguiente sistema dinámico lineal invariante en el tiempo:

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{u} \\ \mathbf{y} &= \mathbf{c}^T \mathbf{x} \end{aligned} \quad (23)$$

donde:

- $\mathbf{x} \in \mathbb{R}^{n \times 1}$ es el vector de estados del sistema.
- $\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} \in \mathbb{R}^{n \times 1}$ es la derivada temporal del estado.
- $A \in \mathbb{R}_+^{n \times n}$ es la matriz de parámetros del sistema.
- $B \in \mathbb{R}_+^{n \times 1}$ es el vector de parámetros que acopla a la entrada.
- $\mathbf{c} \in \mathbb{R}^{n \times 1}$ es un vector que acopla el vector de estado con la respuesta del sistema.

- $u \in \mathbb{R}$ es la señal escalar de entrada.
- $y \in \mathbb{R}$ es la respuesta del sistema.

Ahora considere la estructura propuesta del observador en su versión vectorial:

$$\dot{\hat{x}} = K \frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} + A\hat{x} + Bu + \Gamma\tilde{x} \quad (24)$$

En la que:

- $\hat{x} \in \mathbb{R}^{n \times 1}$ es el vector de estado estimada del sistema.
- $\dot{\hat{x}} = \frac{d\hat{x}}{dt} \in \mathbb{R}^{n \times 1}$ es la derivada temporal del estado estimado.
- $\tilde{x} \in \mathbb{R}^{n \times 1}$ es el vector del error estimado.
- $A \in \mathbb{R}_+^{n \times n}$ representa la matriz del sistema a estimar.
- $B \in \mathbb{R}_+^{n \times 1}$ representa el vector de parámetros de acoplamiento con la entrada del sistema a estimar.
- $u \in \mathbb{R}$ es la entrada del sistema a estimar.
- $K \in \mathbb{R}^{n \times n}$ es una matriz diagonal de parametros del sistema.
- $\left[\frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} \right]^T = \left[\frac{\sinh(\tilde{x}_1)}{1 + \cosh(\tilde{x}_1)}, \frac{\sinh(\tilde{x}_2)}{1 + \cosh(\tilde{x}_2)}, \dots, \frac{\sinh(\tilde{x}_n)}{1 + \cosh(\tilde{x}_n)} \right] \in \mathbb{R}_+^{n \times 1}$ es un vector de parametros.
- $\Gamma \in \mathbb{R}^{n \times n}$ es una matriz diagonal definida positiva de parametros de diseño tal que :

$$\Gamma = \begin{bmatrix} \gamma_1 & 0 & \dots & 0 \\ 0 & \gamma_2 & 0 & \vdots \\ \vdots & 0 & \ddots & \vdots \\ 0 & \dots & 0 & \gamma_n \end{bmatrix}$$

Ahora se define el error entre el estado y lo estimado junto con su derivada como:

$$\begin{aligned} \tilde{x} &= x - \hat{x} \\ \dot{\tilde{x}} &= \dot{x} - \dot{\hat{x}} \end{aligned} \quad (25)$$

se substituyen los terminos de (23) y (24) en (25):

$$\dot{\tilde{x}} = Ax + Bu - \left[K \frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} + A\hat{x} + Bu + \Gamma\tilde{x} \right] \quad (26)$$

Al desarrollar (26) y substituir con (25) se obtiene:

$$\dot{\tilde{x}} = Ax + Bu - K \frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} - A\hat{x} - Bu - \Gamma\tilde{x}$$

$$\dot{\tilde{x}} = Ax - K \frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} - A\hat{x} - \Gamma\tilde{x}$$

$$\dot{\tilde{x}} = A(x - \hat{x}) - K \frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} - \Gamma\tilde{x}$$

$$\dot{\tilde{x}} = A\tilde{x} - K \frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} - \Gamma\tilde{x}$$

$$\dot{\tilde{x}} = -[\Gamma - A]\tilde{x} - K \frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} \quad (27)$$

De (27) obtenemos la representación de la evolución del sistema en el tiempo, utilizando esta representación se inicia realizando la siguiente igualdad $\dot{\tilde{x}} = -[\Gamma - A]\tilde{x} - K \frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} = 0$ entonces:

$$-[\Gamma - A]\tilde{x} - K \frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} = 0$$

$$-[\Gamma - A]\tilde{x} = K \frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})}$$

$$\tilde{x} = [-(\Gamma - A)]^{-1} K \frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} = 0 \Leftrightarrow \tilde{x} = 0 \quad (28)$$

Entonces, si se cumplen las condiciones $A \neq \Gamma$ y $[-(\Gamma - A)] \exists$ se puede ver que el único valor para el que se cumple que la igualdad (28) $[-(\Gamma - A)]\tilde{x} - K \frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} = 0$ es $\tilde{x} = 0$ con lo cual el punto de equilibrio existe y es único para el vector de ceros $\tilde{x} = 0$

D. Comprobar que el error de estimación tienda asintoticamente a cero para el caso de estudio vectorial.

Para continuar se propone una función candidata de Lyapunov de la forma:

$$\begin{aligned} V(\tilde{x}) &= \frac{1}{2} \tilde{x}^T P \tilde{x} + \left[\sqrt{\ln(1 + \cosh(\tilde{x})) - \ln(2)} \right]^T \\ &\cdot K \cdot \left[\sqrt{\ln(1 + \cosh(\tilde{x})) - \ln(2)} \right] \end{aligned} \quad (29)$$

En la que

- $P \in \mathbb{R}_+^{n \times n}$ es una matriz definida positiva .
- $K \in \mathbb{R}_+^{n \times n}$ es una matriz diagonal definida positiva
- $\left[\sqrt{\ln(1 + \cosh(\tilde{x})) - \ln(2)} \right] \in \mathbb{R}_+^{n \times 1}$ es un vector de parametros, tal que:

$$\left[\begin{array}{c} \sqrt{\ln(1 + \cosh(\tilde{x}_1)) - \ln(2)} \\ \sqrt{\ln(1 + \cosh(\tilde{x}_2)) - \ln(2)} \\ \vdots \\ \sqrt{\ln(1 + \cosh(\tilde{x}_n)) - \ln(2)} \end{array} \right]$$

Dicha función propuesta es definida positiva ya que se cumple que :

$$x^T Q x > 0 \Leftrightarrow Q > 0$$

Ahora, se obtiene la derivada de la función propuesta (29) como:

$$\begin{aligned} \dot{V}(\tilde{x}) &= \tilde{x}^T P \dot{\tilde{x}} \\ &+ 2 \cdot \left[\sqrt{\ln(1 + \cosh(\tilde{x})) - \ln(2)} \right]^T \cdot K \cdot \frac{d}{dt} \left[\sqrt{\ln(1 + \cosh(\tilde{x})) - \ln(2)} \right] \end{aligned} \quad (30)$$

Al sustituir con (30) y (27) se obtiene:

$$\begin{aligned} \dot{V}(\tilde{x}) &= \tilde{x}^T P \left[-[\Gamma - A]\tilde{x} - K \frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} \right] \\ &+ 2 \left[\sqrt{\ln(1 + \cosh(\tilde{x})) - \ln(2)} \right]^T \cdot K \\ &\cdot \begin{bmatrix} \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{x}_1)) - \ln(2)}} \frac{\sinh(\tilde{x}_1)}{1 + \cosh \tilde{x}_1} \\ \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{x}_2)) - \ln(2)}} \frac{\sinh(\tilde{x}_2)}{1 + \cosh \tilde{x}_2} \\ \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{x}_n)) - \ln(2)}} \frac{\sinh(\tilde{x}_n)}{1 + \cosh \tilde{x}_n} \end{bmatrix} \dot{\tilde{x}} \end{aligned} \quad (31)$$

Se puede dividir (31) en dos terminos como:

$$e = \tilde{x}^T P \left[-[\Gamma - A]\tilde{x} - K \frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} \right]$$

$$\begin{aligned} f &= + 2 \left[\sqrt{\ln(1 + \cosh(\tilde{x})) - \ln(2)} \right]^T \cdot K \\ &\cdot \begin{bmatrix} \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{x}_1)) - \ln(2)}} \frac{\sinh(\tilde{x}_1)}{1 + \cosh \tilde{x}_1} \\ \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{x}_2)) - \ln(2)}} \frac{\sinh(\tilde{x}_2)}{1 + \cosh \tilde{x}_2} \\ \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{x}_n)) - \ln(2)}} \frac{\sinh(\tilde{x}_n)}{1 + \cosh \tilde{x}_n} \end{bmatrix} \dot{\tilde{x}} \end{aligned} \quad (32)$$

Al desarrollar el segundo termino de la suma de (31) es decir el termino f de (32) con K siendo una matriz diagonal definida positiva se obtiene:

$$\begin{aligned} f &= 2 \left[\sqrt{\ln(1 + \cosh(\tilde{x})) - \ln(2)} \right]^T \\ &\cdot \begin{bmatrix} k_{11} & 0 & \dots & 0 \\ 0 & k_{22} & \dots & 0 \\ \vdots & \ddots & k_{33} & 0 \\ 0 & \dots & 0 & k_{nn} \end{bmatrix} \\ &\cdot \begin{bmatrix} \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{x}_1)) - \ln(2)}} \frac{\sinh(\tilde{x}_1)}{1 + \cosh \tilde{x}_1} \\ \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{x}_2)) - \ln(2)}} \frac{\sinh(\tilde{x}_2)}{1 + \cosh \tilde{x}_2} \\ \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{x}_n)) - \ln(2)}} \frac{\sinh(\tilde{x}_n)}{1 + \cosh \tilde{x}_n} \end{bmatrix} \dot{\tilde{x}} \end{aligned} \quad (33)$$

Al expresar el vector a la derecha de la matriz K como una matriz simétrica se obtiene entonces:

$$K = \begin{bmatrix} a & 0 & \dots & 0 \\ 0 & b & \dots & 0 \\ \vdots & \ddots & c & 0 \\ 0 & \dots & 0 & d \end{bmatrix} \quad (34)$$

En la que

$$\begin{aligned} a &= \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{x}_1)) - \ln(2)}} \frac{\sinh(\tilde{x}_1)}{1 + \cosh \tilde{x}_1} \\ b &= \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{x}_2)) - \ln(2)}} \frac{\sinh(\tilde{x}_2)}{1 + \cosh \tilde{x}_2} \\ c &= \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{x}_3)) - \ln(2)}} \frac{\sinh(\tilde{x}_3)}{1 + \cosh \tilde{x}_3} \\ d &= \frac{1}{2\sqrt{\ln(1 + \cosh(\tilde{x}_n)) - \ln(2)}} \frac{\sinh(\tilde{x}_n)}{1 + \cosh \tilde{x}_n} \end{aligned} \quad (35)$$

Como el producto entre matrices simétricas es conmutativo se procede a realizar la siguiente operación

$$\begin{aligned} f &= 2 \cdot \left[\sqrt{\ln(1 + \cosh(\tilde{x})) - \ln(2)} \right]^T \cdot \begin{bmatrix} a & 0 & \dots & 0 \\ 0 & b & \dots & 0 \\ \vdots & \ddots & c & 0 \\ 0 & \dots & 0 & d \end{bmatrix} \\ &\cdot \begin{bmatrix} k_{11} & 0 & \dots & 0 \\ 0 & k_{22} & \dots & 0 \\ \vdots & \ddots & k_{33} & 0 \\ 0 & \dots & 0 & k_{nn} \end{bmatrix} \dot{\tilde{x}} \end{aligned} \quad (36)$$

Al realizar el producto de las dos matrices se obtiene:

$$f = \left[\frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} \right]^T \begin{bmatrix} k_{11} & 0 & \dots & 0 \\ 0 & k_{22} & \dots & 0 \\ \vdots & \ddots & k_{33} & 0 \\ 0 & \dots & 0 & k_{nn} \end{bmatrix} \dot{\tilde{x}} \quad (37)$$

al simplificar (36) se tiene que:

$$f = \left[\frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} \right]^T K \dot{\tilde{x}} \quad (38)$$

Al tener los terminos simplificados se continua con el desarrollo de (31), se obtiene entonces:

$$\begin{aligned} \dot{V}(\tilde{x}) &= \tilde{x}^T P \left[-[\Gamma - A]\tilde{x} - K \left[\frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} \right] \right] \\ &+ \left[\frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} \right]^T K \dot{\tilde{x}} \\ \dot{V}(\tilde{x}) &= \tilde{x}^T P \left[-[\Gamma - A]\tilde{x} - K \left[\frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} \right] \right] \\ &+ \left[\frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} \right]^T K \left[-[\Gamma - A]\tilde{x} - K \left[\frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} \right] \right] \\ \dot{V}(\tilde{x}) &= -\tilde{x}^T P [\Gamma - A]\tilde{x} - \tilde{x}^T P K \left[\frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} \right] \\ &- \left[\frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} \right]^T K [\Gamma - A]\tilde{x} \\ &- \left[\frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} \right]^T K K \left[\frac{\sinh(\tilde{x})}{1 + \cosh(\tilde{x})} \right] \end{aligned} \quad (39)$$

E. Realizar la simulación correspondiente donde se compruebe que el error de estimación tiende a cero para el caso de estudio vectorial.

Al utilizar las condiciones obtenidas durante la comprobación teórica del observador se puede entonces utilizar esta información para llevar a cabo una simulación, los parámetros del sistema utilizado para dicha simulación son usados solo como ejemplo no representan algún sistema modelado.

Entonces considere al sistema (23) en el que:

$$A = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}, B = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, C = \begin{bmatrix} 4 \\ 4 \end{bmatrix}, u = \sin(t) \quad (40)$$

Ahora considere al observador propuesto (24) en el que:

$$K = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, \Gamma = \begin{bmatrix} 6 & 0 \\ 0 & 6 \end{bmatrix} \quad (41)$$

Puede comprobarse que se cumple que:

- $A \neq \Gamma \quad (42)$

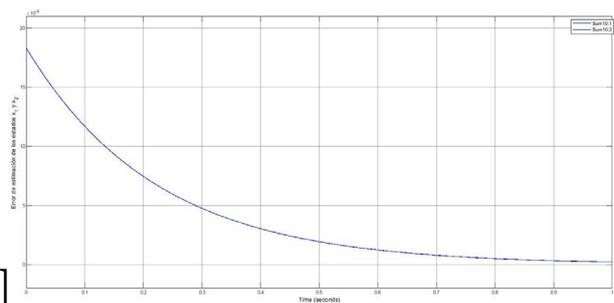
- $[-(\Gamma - A)]^{-1} \exists$ ya que el determinante de $\det[-(\Gamma - A)] = 12 \quad (43)$

con lo cual $\det \neq 0$ por lo tanto $[-(\Gamma - A)]^{-1} \exists$.

- $(\Gamma - A) > 0$ es decir definida positiva ya que cumple con el criterio de Sylvester [4] en el que

$$\begin{aligned} a_{11} &= 4 > 0 \\ |\Gamma - A| &= 12 > 0 \end{aligned} \quad (44)$$

Fig. 1: Error de estimación



Al utilizar los valores propuestos para el sistema (40) y para el observador (41) en el entorno de simulación MATLAB y adicionalmente incluir la condición inicial $u = \sin(0) = 0$ se obtiene como resultado la Fig.1 en la cual se comprueba que el error de estimación tiende asintóticamente a cero.

F. Realizar la comprobación práctica y mostrar los resultados obtenidos al utilizar el algoritmo de control y el observador propuestos

La comprobación práctica se llevo a cabo al utilizar el equipo presente en el laboratorio de robótica y control de la Facultad de Ciencias de la Electronica de la Benemérita Universidad Autónoma de Puebla constituido por:

- Robot Rotradi, constituido por tres articulaciones rotacionales las cuales son tres servomotores de transmisión directa, cada uno cuenta con su respectivo encoder óptico incremental junto con su respectivo servoamplificador.
- Tarjeta MKR Vidor 4000, una placa de desarrollo de Arduino que cuenta con una FPGA intel Cyclone 10CL016
- Interfaz de usuario programada para el correcto funcionamiento y comunicación entre ambos dispositivos.

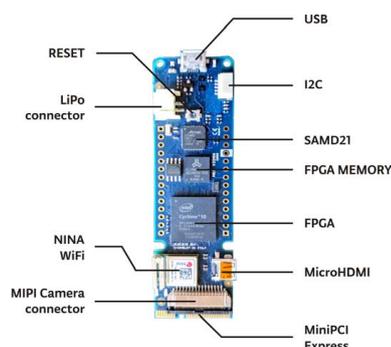


Fig. 2: Tarjeta MKR Vidor 4000



Fig. 3: Robot Rotradi

El funcionamiento de este equipo descrito de manera simplificada se detalla a continuación.

Cada servomotor del robot Rotradi cuenta con un encoder incremental el cual genera dos señales de cuadratura A y B que representan el movimiento del servomotor, dicha información es enviada y posteriormente procesada por la tarjeta MKR Vidor 4000 donde la parte correspondiente a la FPGA se utiliza como un módulo de adquisición y tratamiento de señales para así obtener lo que sería las posiciones de cada uno de los servomotores.

Al contar con la información de las posiciones de los servomotores la interfaz de usuario utiliza esta información junto con el algoritmo de control y el observador y genera una señal de salida en volts, dicha señal es enviada a los servoamplificadores donde determina el movimiento al pasar de voltaje a torque(par) entonces ya que los servomotores de transmisión directa son una fuente de par todo este proceso determina el movimiento hasta llegar a la posición deseada misma que es designada en la interfaz de usuario

Los experimentos realizados en los cuales se comprueba el funcionamiento del algoritmo de control así como del observador son dos, en el primero Fig.4 solo se prueba el algoritmo de control para conseguir una posición deseada $q_d = [45, 45, 90]$ y para estimar la velocidad se utiliza el método de derivación de Euler, en el segundo Fig.5 se implementan ambos el algoritmo de control y el observador propuestos, en el cual se propone obtener la misma posición deseada.

En el primer experimento Fig.4 se sintonizan las ganancias con el fin de obtener el mejor re-

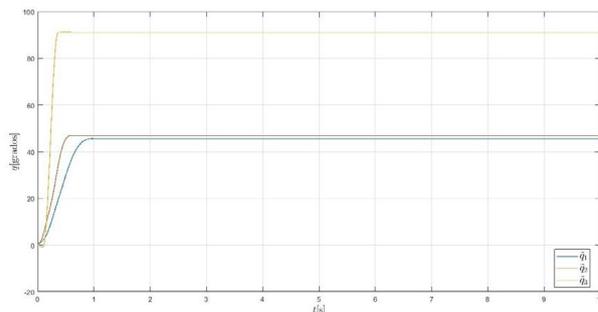


Fig. 4: Posiciones obtenidas utilizando el algoritmo de control propuesto

sultado, llegando a la posición $q = [91, 47, 46]$ en un corto periodo de tiempo y sin sobreimpulso de tal forma que se comprueba el correcto funcionamiento del algoritmo de control.

En el caso del segundo experimento al utilizar los mismo valores para las ganancias del algoritmo de control se obtienen resultados parecidos llegando a la posición $q = [76, 52, 37]$ en un poco más de tiempo que el experimento 1 y presentando un ligero sobreimpulso, el cual se puede deber a una incorrecta sintonización en los parámetros de diseño del observador, a la velocidad de estimación del observador debido a un retardo provocado por la simplificación del sistema presente en el modelo utilizado para implementar el observador de manera práctica. En el primer experimento no se observa el sobreimpulso mencionado ya que las ganancias se sintonizaron a forma de obtener el resultado más óptimo y en este caso aunque la estimación de velocidad por derivación numérica (euler) suele estar predispuesta a sufrir perturbaciones por ruido y resultados deficientes en bajas y altas velocidades las condiciones del experimento en las cuales no se presenta una cantidad considerable de ruido y las velocidades son óptimas se

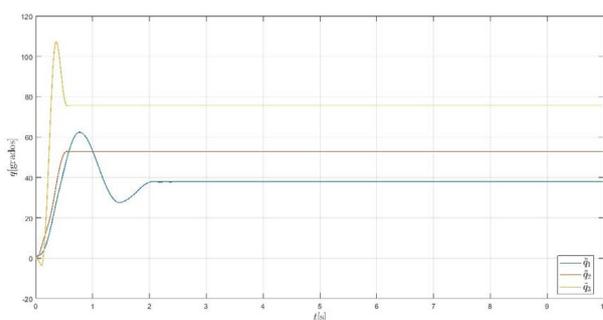


Fig. 5: Posiciones obtenidas utilizando el algoritmo de control y observador propuesto

obtienen buenos resultados .

IV. CONCLUSIONES

Este tipo de comprobaciones de estabilidad asintótica de algoritmos de control y de observadores ayudan a explicar y con ello a comprender la teoría detrás de estos elementos, si bien son ampliamente conocidos y utilizados el desarrollo y comprobación teórica de su funcionamiento brinda más ejemplos con los cuales se puede ampliar el número de recursos que se tiene para su enseñanza, por lo tanto se espera que en consecuencia dicho material sirva para ayudar a comprender de una manera más clara la teoría de su funcionamiento y a la vez utilizar la información generada para darle aplicaciones en este caso para el control de posición de robots manipuladores y en el caso del observador como un estimador de velocidad.

REFERENCES

- [1] Reyes, F. (2024). Control de robots manipuladores clase 37 capítulo 4 Robótica 1 pp.3-8.
- [2] Cortés, F. R. (2011). *Róbotica: control de robots manipuladores* pp.257-258.
- [3] Takegaki, M., & Arimoto, S. (1981). A New Feedback Method for Dynamic Control of Manipulators. *Journal Of Dynamic Systems, Measurement, And Control*, 103(2), 119-125. <https://doi.org/10.1115/1.3139651>
- [4] Cortés, F. R. (2011). *Róbotica: control de robots manipuladores* p.132.

Modelo dinámico de un robot RPP

1st Jesús Misael González Teco 2nd Moisés Gizeh Jiménez Hernández 3rd José Fernando Reyes Cortés

Facultad de Ciencias
de la Electrónica
BUAP

Puebla de Zaragoza, México
misaelgonzalezteco@ieee.org

Facultad de Ciencias
de la Electrónica
BUAP

Puebla de Zaragoza, México
moises.jimenezh@alumno.buap.mx

Facultad de Ciencias
de la Electrónica
BUAP

Puebla de Zaragoza, México
fernando.reyes@correo.buap.mx

Abstract—En el presente documento se examina minuciosamente la dinámica de un robot manipulador de tres grados de libertad con una configuración RPP, centrándose en el análisis del modelo dinámico y aspectos clave como la matriz de inercia y la presencia de los fenómenos de las fuerzas centrípetas y de Coriolis, siguiendo las ecuaciones de Euler-Lagrange para obtener el modelo cinemático, diferencial y finalmente llegar al dinámico del robot anteriormente mencionado. El objetivo principal es proporcionar una comprensión profunda de los aspectos dinámicos esenciales y se abordará la compleja interacción entre variables y su influencia en el modelo dinámico del robot para observar la dinámica acoplada dentro del mismo.

Index Terms—cinemático, diferencial, dinámico, Euler-Lagrange

I. Introducción

Cinemática directa

Es una función vectorial $f_R(l_i, \mathbf{q})$ que relaciona las coordenadas articulares $\mathbf{q} \in \mathbb{R}$ y propiedades geométricas del sistema mecánico l_i con las coordenadas cartesianas $[x, y, z]^T \in \mathbb{R}^3$ del robot y la orientación $[\theta, \phi, \psi]^T \in \mathbb{R}^3$ de la herramienta colocada en el extremo final. Es decir $f_R: \mathbb{R}^n \rightarrow \mathbb{R}^m$ tal que:

$$\begin{bmatrix} x \\ y \\ z \\ \theta \\ \phi \\ \psi \end{bmatrix} = f_R(l_i, \mathbf{q}) \quad (1)$$

donde n indica el número de grados de libertad y la dimensión del vector de coordenadas articulares \mathbf{q} , m es la dimensión conjunta de las coordenadas cartesianas y la orientación de la herramienta de trabajo. [1]

Cinemática diferencial

La cinemática diferencial directa es la derivada con respecto al tiempo de la cinemática directa

$$\frac{d}{dt} [x \ y \ z \ \theta \ \phi \ \psi]^T = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = \frac{d}{dt} f_R(\mathbf{q}) = \frac{\partial f_R(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = J(\mathbf{q}) \dot{\mathbf{q}} \quad (2)$$

Como se ve, ésta relaciona la velocidad articular $\dot{\mathbf{q}} \in \mathbb{R}^n$ con la velocidad lineal $\mathbf{v} = \frac{d}{dt} [x, y, z]^T = [\dot{x}, \dot{y}, \dot{z}]^T \in \mathbb{R}^3$ y la velocidad angular $\boldsymbol{\omega} = \frac{d}{dt} [\theta, \phi, \psi]^T = [\dot{\theta}, \dot{\phi}, \dot{\psi}]^T \in \mathbb{R}^3$, además el mapeo es descrito en términos de una matriz $J(\mathbf{q}) = \frac{\partial f_R(\mathbf{q})}{\partial \mathbf{q}} \in \mathbb{R}^{6 \times n}$ denominada jacobiano del robot o jacobiano analítico:

$$J(\mathbf{q}) = \begin{bmatrix} J_v(\mathbf{q}) \\ J_\omega(\mathbf{q}) \end{bmatrix} \quad (3)$$

$J_v(\mathbf{q}) \in \mathbb{R}^{3 \times n}$ relaciona la velocidad articular $\dot{\mathbf{q}} \in \mathbb{R}^n$ con la velocidad lineal $\mathbf{v} \in \mathbb{R}^3$, mientras que $J_\omega(\mathbf{q}) \in \mathbb{R}^{3 \times n}$ relaciona la velocidad angular $\boldsymbol{\omega} \in \mathbb{R}^3$ con la velocidad articular $\dot{\mathbf{q}} \in \mathbb{R}^n$, es decir:

$$\begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} = J(\mathbf{q}) \dot{\mathbf{q}} = \begin{bmatrix} J_v(\mathbf{q}) \dot{\mathbf{q}} \\ J_\omega(\mathbf{q}) \dot{\mathbf{q}} \end{bmatrix} \quad (4)$$

El jacobiano del robot representa una importante herramienta en robótica que sirve para caracterizar a un robot manipulador, encontrar configuraciones singulares, analizar redundancia, determinar la cinemática diferencial inversa, así como describir la relación entre la fuerza aplicada y los pares o torques resultantes del extremo final. Es indispensable para el análisis y diseño de algoritmos de control cartesiano. [1]

Convención Euler-Denavit-Hartenberg

Las variables articulares en la representación Denavit-Hartenberg se denotan con θ_i para el tipo rotacional, prismática o lineal por d_i ; este parámetro d_i también hace el papel de representar el ancho del servomotor de la articulación rotacional más el espesor de la placa metálica del eslabón, en este caso se denota por el símbolo β_i ; la longitud del eslabón se representa con l_i y el ángulo de separación entre los ejes z_i y z_{i-1} se denota con α_i .

En general, se tiene el siguiente procedimiento:

- 1) El eje z_i se asigna rígidamente a la articulación $i+1$. Es decir, z_0 es el eje de la articulación 1, z_1 es el eje de la articulación 2, y así sucesivamente.

- 2) Localizar el origen o_i del sistema de referencia $\Sigma_i(x_i, y_i, z_i)$ en la intersección del eje z_i con la normal común a los ejes z_{i-1} y z_i .
- 3) Seleccionar el eje x_{i-1} sobre la normal que une los ejes z_{i-1} y z_i en dirección de la articulación $i-1$ hacia la articulación i .
- 4) Definir el ángulo de torsión α_i , este es el ángulo entre los ejes z_i y z_{i-1} y se mide con valor positivo en el sentido de las manecillas del reloj sobre el eje x_i .
- 5) Seleccionar el eje y_i por la regla de la mano derecha.

En la representación Denavit-Hartenberg cada transformación homogénea H_{i-1}^i se representa por el producto de cuatro transformaciones básicas:

$$H_{i-1}^i = H_{R_{z_{i-1}}}(\theta_i)H_{T_{z_{i-1}}}(d_i(\beta_i))H_{T_{x_{i-1}}}(l_i)H_{R_{x_{i-1}}}(\alpha_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i(\beta_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & l_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & l_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i(\beta_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

La transformación homogénea total se obtiene como $H_n^0 = H_0^1 H_1^2 \dots H_{n-2}^{n-1} H_{n-1}^n \cdot [1]$

Modelo dinámico

El modelo dinámico del robot manipulador permite explicar todos los fenómenos físicos que se encuentran en su estructura mecánica, tales como efectos inerciales, fuerzas centrípetas y de Coriolis, par gravitacional y fricción, los cuales son fenómenos físicos propios de la naturaleza dinámica del robot. Hay varios métodos de modelado de la física como el de Newton o el de Hamilton. Sin embargo, la mejor opción como metodología de modelado la representa las ecuaciones de movimiento de Euler-Lagrange debido a las propiedades matemáticas que se deducen de manera natural, ya que facilitan el análisis y diseño de algoritmos de control. [1]

$$\tau = M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) + f_f(\dot{\mathbf{q}}, f_e) \quad (6)$$

donde:

- $\mathbf{q} \in \mathbb{R}^n$ es el vector de coordenadas generalizadas o posiciones articulares.
- $\dot{\mathbf{q}} \in \mathbb{R}^n$ es el vector de velocidades articulares.
- $\ddot{\mathbf{q}} \in \mathbb{R}^n$ es el vector de aceleraciones articulares.

- $M(\mathbf{q}) \in \mathbb{R}^{n \times n}$ es la matriz de inercia, la cual es simétrica y definida positiva.
- $C(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ es la matriz de fuerzas centrípetas y de Coriolis.
- $g(\mathbf{q}) \in \mathbb{R}^n$ es el vector de fuerzas o pares gravitacionales.
- $f_f(\dot{\mathbf{q}}, f_e) \in \mathbb{R}^n$ es el vector de pares de fricción que influye la fricción viscosa, de Coulomb y estática en cada articulación del robot.

A. Ecuaciones de movimiento de Euler-Lagrange

La suma de la energía cinética $\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}})$ más la energía potencial $\mathcal{U}(\mathbf{q})$ se representa por $\mathcal{H}(\mathbf{q}, \dot{\mathbf{q}})$ y se denomina el Hamiltoniano del robot.

$$\mathcal{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) + U(\mathbf{q})$$

donde

- $\mathbf{q}, (\dot{\mathbf{q}}) \in \mathbb{R}^n$ representan los vectores de posición y velocidad articular, respectivamente.
- La energía cinética $\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}})$ tiene una dependencia de la posición y velocidad articular.
- La energía potencial $U(\mathbf{q})$ está relacionada con el campo conservativo de la gravedad y por lo tanto únicamente depende de la posición.

El lagrangiano $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$ de un robot manipulador de n grados de libertad se define como la diferencia entre la energía cinética $\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}})$ y la energía potencial $U(\mathbf{q})$:

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) - U(\mathbf{q})$$

Las ecuaciones de movimiento de Euler-Lagrange de un robot manipulador de n grados de libertad están dadas por:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \mathbf{q}} = \tau - f_f(\dot{\mathbf{q}}, f_e)$$

donde

- $\mathbf{q} = [q_1, q_2, \dots, q_n]^T \in \mathbb{R}^n$ representa el vector de posiciones articulares o coordenadas generalizadas.
- $\dot{\mathbf{q}} = [\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n]^T \in \mathbb{R}^n$ es el vector de velocidades articulares.
- $\tau = [\tau_1, \tau_2, \dots, \tau_n]^T \in \mathbb{R}^n$ es el vector de pares aplicados.
- $f_f(\dot{\mathbf{q}}, f_e) \in \mathbb{R}^n$ es el vector de fuerzas o pares de fricción que depende de la velocidad articular ($\dot{\mathbf{q}}$) y de la fricción estática f_e presente en las articulaciones del mismo.
- $t \in \mathbb{R}_+$ representa la evolución del tiempo.
- $n \in \mathbb{N}$ es el número de grados de libertad.

[1]

1) **Matriz de Coriolis:** La matriz $C(q, \dot{q})$ está relacionada con la matriz de inercia $M(q)$ por la expresión:

$$x^T \left[\frac{1}{2} \dot{M}(q) - C(q, \dot{q}) \right] x = 0 \quad (7)$$

y, de hecho, $\frac{1}{2} \dot{M}(q) - C(q, \dot{q})$ es una matriz antisimétrica. En forma análoga la matriz $\dot{M}(q) - 2C(q, \dot{q})$ es antisimétrica y también resulta cierto que:

$$M(\dot{q}) = C(q, \dot{q}) + C(q, \dot{q})^T \quad (8)$$

[2]

2) **Vector de fuerzas o pares gravitacionales:** Se obtiene como el gradiente de la energía potencial.

$$g(q) = \frac{\partial U(q)}{\partial q} \quad (9)$$

3) **Fenómeno de fricción:** El fenómeno de fricción tiene el efecto físico de oponerse al movimiento del robot, su característica principal es un fenómeno disipativo en velocidades diferentes a cero y con entradas acotadas dentro del primer y tercer cuadrante, lo que permite considerar los modelos tradicionales de fricción viscosa y de Coulomb, para modelarlos como una combinación lineal de la fricción viscosa, Coulomb y estática. La característica disipativa de la fricción significa que convierte la energía mecánica en energía térmica.

$$f_f(\dot{q}, f_e) = B\dot{q} + F_c \text{signo}(\dot{q}) + \begin{bmatrix} [1 - |\text{signo}(\dot{q}_1)|] & \dots & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \vdots \\ 0 & \dots & 0 \\ 0 & \dots & [1 - |\text{signo}(\dot{q}_n)|] \end{bmatrix} f_e \quad (10)$$

donde $B, F_c \in \mathbb{R}^{n \times n}$ son matrices diagonales de coeficientes de fricción viscosa y de Coulomb, respectivamente.

La fricción estática f_e está representada por $f_e = [f_{e1}, f_{e2}, \dots, f_{en}]^T$ que es el vector de fricción estática que contiene los coeficientes de fricción estática de cada uno de los servomotores del robot manipulador.

B. Metodología para obtener el modelo dinámico

1) **Paso 1:** Obtener la cinemática directa del centro de masa de cada uno de los eslabones. Para el i -ésimo eslabón tomar en cuenta las longitudes anteriores l_{i-1} y ángulos q_i y q_{i-1} :

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = f_R(l_i, l_{i-1}, l_{ci}, q_{i-1}, q_i)$$

2) **Paso 2:** Calcular la cinemática diferencial del i -ésimo eslabón y deducir la rapidez lineal:

$$v_i = \frac{d}{dt} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = f_R(l_i, l_{i-1}, l_{ci}, q_{i-1}, q_i)$$

La rapidez lineal del centro de masa de cada eslabón se calcula de la siguiente forma:

$$v_i^T v_i = \dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2$$

3) **Paso 3:** Obtener el lagrangiano $\mathcal{L}(q, \dot{q})$:

$$\mathcal{L}(q, \dot{q}) = \mathcal{K}(q, \dot{q}) - \mathcal{U}(q)$$

La energía cinética $\mathcal{K}(q, \dot{q})$ incluye el movimiento de traslación y rotación, por ejemplo:

$$K_i(\dot{q}_i, q_i) = \frac{1}{2} m_i v_i^T(\dot{q}_i) v_i(\dot{q}_i) + \frac{1}{2} I_i \left[\sum_i \dot{q}_i \right]^2$$

La energía potencial no tiene forma específica como la energía cinética, depende de la geometría del robot en general $\mathcal{U}_i(q) = m_i g l_{ci} h_i(q)$, siendo $h_i(q)$ una función que indica la altura del eslabón respecto al origen del sistema de referencia del robot.

4) **Paso 4:** Aplicar las ecuaciones de movimiento de Euler-Lagrange e incluir el modelo de fricción.

$$\tau_i = \frac{d}{dt} \left(\frac{\partial L(q, \dot{q})}{\partial \dot{q}_i} \right) - \frac{\partial L(q, \dot{q})}{\partial q_i} + b_i \dot{q}_i + f_{ci} \text{signo}(\dot{q}_i) + f_{ei} (1 - |\text{signo}(\dot{q}_i)|) \quad [1]$$

II. Tabla de valores Euler-Denavit-Hartenberg

El robot a estudiar es del tipo RPP, siendo su primera articulación rotacional, la segunda prismática y la tercera del mismo modo, por lo cual es necesario disponer de cuatro sistemas de coordenadas en total, una en cada articulación y el final en la herramienta del robot.

A partir de esta convención, se obtiene la tabla de valores de Euler-Denavit-Hartenberg como sigue:

TABLE I
Tabla con los valores de la convención Euler-Denavit-Hartenberg

Eslabón	q_i	$\beta_i \oplus d_i$	l_i	α_i
1	$q_1 - \frac{\pi}{2}$	$\beta_1 + l_1 + l_{1c} + l_s$	$l_{1a} \cos \rho$	$-\frac{\pi}{2} + \rho$
2	0	d_2	0	$-\frac{\pi}{2}$
3	0	d_3	0	0

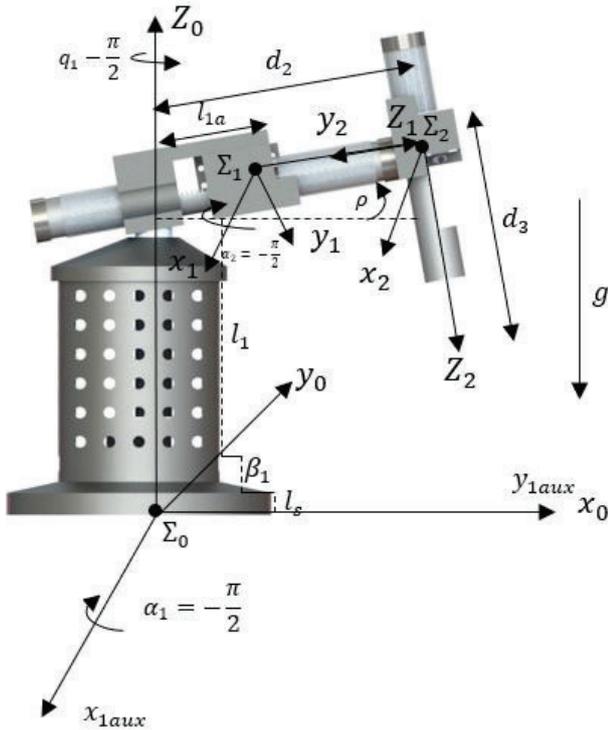


Fig. 1. Sistemas de coordenadas propuesto para el robot RPP.

III. Cinemática directa

Se obtienen las matrices de transformación homogéneas para cada articulación y se realiza la separación del primer y segundo cuadrante, correspondientes a la parte traslacional y rotacional respectivamente, esto con el objetivo de facilitar el desarrollo de la cinemática directa al momento de obtener sus respectivas derivadas.

Para H_0^1 , es decir, de Σ_1 a Σ_0

Matriz de rotación: R_0^1

$$R_0^1 = R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) \quad (11)$$

Matriz de traslación: d_0^1

$$d_0^1 = R_{z0} \left(q_1 - \frac{\pi}{2} \right) \begin{pmatrix} l_{c1} \\ 0 \\ \beta_1 + l_1 + l_{1c} + l_s \end{pmatrix} \quad (12)$$

Para $H_0^2 = H_0^1 H_1^2$, es decir, de Σ_2 a Σ_0

Matriz de transformación homogénea H_0^2 :

$$H_0^2 = R_{z0} \left(q_1 - \frac{\pi}{2} \right) \begin{pmatrix} l_{1a} \cos(\rho) \\ 0 \\ \beta_1 + l_1 + l_{1c} + l_s \end{pmatrix} + R_{z0} \left(q_1 - \frac{\pi}{2} \right) \\ \times R_{x0} \left(-\frac{\pi}{2} + \rho \right) \left[\begin{pmatrix} 0 \\ 0 \\ d_2 \end{pmatrix} + R_{x1} \left(-\frac{\pi}{2} \right) \right]$$

Matriz de rotación: R_0^2

$$R_0^2 = R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) R_{x1} \left(-\frac{\pi}{2} \right) \quad (13)$$

Matriz de traslación: d_0^2

$$d_0^2 = R_{z0} \left(q_1 - \frac{\pi}{2} \right) \begin{pmatrix} l_{1a} \cos(\rho) \\ 0 \\ \beta_1 + l_1 + l_{1c} + l_s \end{pmatrix} \\ + R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) \begin{pmatrix} 0 \\ 0 \\ d_2 \end{pmatrix} \quad (14)$$

Para $H_0^3 = H_0^1 H_1^2 H_2^3$, es decir, de Σ_3 a Σ_0 :

Matriz de transformación homogénea H_0^3

$$H_0^3 = R_{z0} \left(q_1 - \frac{\pi}{2} \right) \begin{pmatrix} l_{1a} \cos(\rho) \\ 0 \\ \beta_1 + l_1 + l_{1c} + l_s \end{pmatrix} \\ + R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) \begin{pmatrix} 0 \\ 0 \\ d_2 \end{pmatrix} \\ + R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) R_{x1} \left(-\frac{\pi}{2} \right) \begin{pmatrix} 0 \\ 0 \\ d_3 \end{pmatrix} \\ + R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) R_{x1} \left(-\frac{\pi}{2} \right) I$$

Matriz de rotación: R_0^3

$$R_0^3 = +R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) R_{x1} \left(-\frac{\pi}{2} \right) I \quad (15)$$

Matriz de traslación: d_0^3

$$d_0^3 = R_{z0} \left(q_1 - \frac{\pi}{2} \right) \begin{pmatrix} l_{1a} \cos(\rho) \\ 0 \\ \beta_1 + l_1 + l_{1c} + l_s \end{pmatrix} \\ + R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) \begin{pmatrix} 0 \\ 0 \\ d_2 \end{pmatrix} \\ + R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) R_{x1} \left(-\frac{\pi}{2} \right) \begin{pmatrix} 0 \\ 0 \\ d_3 \end{pmatrix} \quad (16)$$

IV. Cinemática diferencial

A continuación se presenta la obtención de la derivada de las matrices de rotación y traslación de cada articulación respecto al sistema Σ_0 en función de las rotaciones individuales realizadas sobre los ejes x y z .

Para H_0^1

Derivada de la matriz de rotación: \dot{R}_0^1

$$\dot{R}_0^1 = S(\dot{q}_1 k_0) R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) \quad (17)$$

Derivada de la matriz de traslación: \dot{d}_0^1

$$\dot{d}_0^1 = S(\dot{q}_1 k_0) R_{z0} \left(q_1 - \frac{\pi}{2} \right) \begin{pmatrix} l_{c1} \\ 0 \\ \beta_1 + l_1 + l_{1C} + l_s \end{pmatrix} \quad (18)$$

Para H_0^2

Derivada de la matriz de rotación: \dot{R}_0^2

$$\dot{R}_0^2 = S(\dot{q}_1 k_0) R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) R_{x1} \left(-\frac{\pi}{2} \right) \quad (19)$$

Derivada de la matriz de traslación: \dot{d}_0^2

$$\begin{aligned} \dot{d}_0^2 = & S(\dot{q}_1 k_0) R_{z0} \left(q_1 - \frac{\pi}{2} \right) \begin{pmatrix} l_{1c} \cos(\rho) \\ 0 \\ \beta_1 + l_1 + l_{1c} + l_s \end{pmatrix} + \\ & S(\dot{q}_1 k_0) R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) \begin{pmatrix} 0 \\ 0 \\ d_2 \end{pmatrix} + \\ & R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) \begin{pmatrix} 0 \\ 0 \\ \dot{q}_2 \end{pmatrix} \end{aligned} \quad (20)$$

Para H_0^3

Derivada de la matriz de rotación: \dot{R}_0^3

$$\dot{R}_0^3 = S(\dot{q}_1 k_0) R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) R_{x1} \left(-\frac{\pi}{2} \right) \quad (21)$$

Derivada de la matriz de traslación: \dot{d}_0^3

$$\begin{aligned} \dot{d}_0^3 = & \dot{d}_0^2 + S(\dot{q}_1 K_0) R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) \\ & \times R_{x1} \left(-\frac{\pi}{2} \right) \begin{pmatrix} 0 \\ 0 \\ d_3 \end{pmatrix} + R_{z0} \left(q_1 - \frac{\pi}{2} \right) R_{x0} \left(-\frac{\pi}{2} + \rho \right) \\ & \times R_{x1} \left(-\frac{\pi}{2} \right) \begin{pmatrix} 0 \\ 0 \\ d_3 \end{pmatrix} \end{aligned} \quad (22)$$

Y con esto se obtiene la rapidez de cada articulación de la forma:

$$\|\dot{d}_0^n\|^2 = \dot{d}_0^{nT} \dot{d}_0^n$$

Para $\|\dot{d}_0^1\|^2$

$$\|\dot{d}_0^1\|^2 = l_{1a}^2 \cos^2(\rho) \dot{q}_1^2 \quad (23)$$

Para $\|\dot{d}_0^2\|^2$

$$\|\dot{d}_0^2\|^2 = [l_{1a}^2 + d_2^2] \cos^2(\rho) \dot{q}_1^2 + 2\dot{d}_2 l_{1a} \cos^2(\rho) \dot{q}_1 + \dot{d}_2^2 \quad (24)$$

Para $\|\dot{d}_0^3\|^2$

$$\begin{aligned} \|\dot{d}_0^3\|^2 = & [(l_{1a}^2 + d_2^2) \cos^2(\rho) + 2d_3 d_2 \cos(\rho) \sin(\rho) + \\ & d_3^2 \sin^2(\rho)] \dot{q}_1^2 - [2 \sin(\rho) \cos(\rho) \dot{d}_2 \\ & - (2 \cos^2(\rho) - 1) \dot{d}_3] 2l_{1a} \cos(\rho) \dot{q}_1 \\ & + \dot{d}_2^2 + \dot{d}_3^2 \end{aligned} \quad (25)$$

V. Modelo de energía

Se prosigue con la obtención de la energía cinética:

$$\begin{aligned} K(\mathbf{q}, \dot{\mathbf{q}}) = & \frac{1}{2} [m_1 l_{1a}^2 \cos^2(\rho) + I_{zz1} \\ & + I_{zz2} + I_{zz3} + m_2 (l_{1a}^2 + d_2^2) \cos^2(\rho) \\ & + m_3 [(l_{1a}^2 + d_2^2) \cos^2(\rho) + d_3^2 \sin^2(\rho) \\ & + 2d_2 d_3 \cos(\rho)] \dot{q}_1^2 + \frac{1}{2} [m_2 + m_3] \dot{d}_2^2 + \frac{1}{2} m_3 \dot{d}_3^2 \\ & + l_{1a} \cos^2(\rho) [m_2 + 2m_3 \sin(\rho)] \dot{d}_2 \dot{q}_1 \\ & - m_3 (2 \cos^2(\rho) - 1) l_{1a} \cos(\rho) \dot{q}_1 \dot{d}_3 \end{aligned} \quad (26)$$

Y este resultado se reescribe en su forma:

$$K(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}}$$

$$\begin{aligned} K(\mathbf{q}, \dot{\mathbf{q}}) = & \frac{1}{2} \begin{bmatrix} \dot{q}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix}^T \\ & \times \begin{bmatrix} I_{zz1} + I_{zz2} + I_{zz3} + m_1 l_{1a}^2 \cos^2(\rho) + m_2 [l_{1a} + d_2^2] \\ \times \cos^2(\rho) + m_3 [(l_{1a}^2 + d_2^2) \cos^2(\rho) + d_3^2 \sin^2(\rho) \\ + 2d_2 d_3 \cos(\rho) \sin(\rho)] \\ [m_2 + 2m_3 \sin(\rho)] (l_{1a} \cos^2(\rho)) \\ -m_3 (2 \cos^2(\rho) - 1) l_{1a} \cos(\rho) \\ (m_2 + 2m_3 \sin(\rho)) & -m_3 (2 \cos^2(\rho) - 1) \\ \times (l_{1a} \cos^2(\rho)) & \times l_{1a} \cos(\rho) \\ m_2 + m_3 & 0 \\ 0 & m_3 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix} \end{aligned}$$

Y la energía potencial queda como:

$$U(\mathbf{q}) = m_1g(l_s + \beta_1 + l_1 + l_{1c}) + m_2g(l_{1c} + l_s + \beta_1 + l_1 + d_2 \sin(\rho)) + m_3g(l_s + \beta_1 + l_1 + l_{1c} + d_2 \sin(\rho) - d_3 \cos(\rho)) \quad (27)$$

Finalizando el tercer paso con la obtención del lagrangiano, el cual es la diferencia de la energía cinética y potencial, siendo:

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} [m_1 + l_{c1}^2 + I_{zz1} + I_{zz2} + I_{zz3} + (m_2 + m_3) \times (l_{1a}^2 \cos^2(\rho))] \dot{q}_1^2 + (m_2 + m_3) \dot{d}_2^2 + m_3 \dot{d}_3^2 - [2(m_2 + m_3)(l_{1a} \cos^2(\rho))] \dot{d}_2 \dot{q}_1 - m_3 \sin(2\rho) l_{1a} \dot{d}_3 \dot{q}_1 - m_1g(l_s + \beta_1 + l_1 + l_{1c}) - m_2g(l_{1c} + l_s + \beta_1 + l_1 + d_2 \sin(\rho)) - m_3g(l_s + \beta_1 + l_1 + l_{1c} + d_2 \sin(\rho) - d_3 \cos(\rho)) \quad (28)$$

Con esto se pueden obtener las respectivas derivadas parciales y temporales con respecto a cada una de las variables.

Para τ_1

$$\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}_1} = [m_1 l_{1a}^2 \cos^2(\rho) + I_{zz1} + I_{zz2} + I_{zz3} + m_2(l_{1a}^2 + d_2^2) \cos^2(\rho) + m_3(l_{1a}^2 + d_2^2) \cos^2(\rho) + d_3^2 \sin^2(\rho) + 2d_2 d_3 \cos(\rho)] \dot{q}_1 + l_{1a} \cos^2(\rho) \times [m_2 + 2m_3 \sin(\rho)] \dot{d}_2 - m_3(2 \cos^2(\rho) - 1) \times l_{1a} \cos(\rho) \dot{d}_3 - \frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial q_1} = 0$$

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}_1} \right] = [m_1 l_{1a}^2 \cos^2(\rho) + I_{zz1} + I_{zz2} + I_{zz3} + m_2(l_{1a}^2 + d_2^2) \cos^2(\rho) + m_3(l_{1a}^2 + d_2^2) \cos^2(\rho) + d_3^2 \sin^2(\rho) + 2d_2 d_3 \cos(\rho)] \ddot{q}_1 + [2m_2 \cos^2(\rho) d_2 \dot{d}_2 + m_3 \times [2 \cos^2(\rho) d_2 \dot{d}_2 + 2d_3 \dot{d}_3 \sin^2(\rho) + 2 \cos(\rho) \sin(\rho) \times [d_2 \dot{d}_2 + d_2 \dot{d}_3]]] \dot{q}_1 + l_{1a} \cos^2(\rho) [m_2 + 2m_3 \sin(\rho)] \ddot{d}_2 - m_3(2 \cos^2(\rho) - 1) l_{1a} \cos(\rho) \ddot{d}_3$$

Para f_{c2}

$$\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{d}_2} = [m_2 + m_3] \dot{d}_2 + l_{1a} \cos^2(\rho) [m_2 + 2m_3 \sin(\rho)] \dot{q}_1$$

$$- \frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial d_2} = [[m_2 + m_3] \cos^2(\rho) d_2 + \cos(\rho) \sin(\rho) d_3] \dot{q}_1^2 + g [m_2 + m_3] \sin(\rho)$$

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{d}_2} \right] = [m_2 + m_3] \ddot{d}_2 + l_{1a} \cos^2(\rho) \times [m_2 + 2m_3 \sin(\rho)] \ddot{q}_1$$

Para f_{c3}

$$\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{d}_3} = m_3 \dot{d}_3 - m_3(2 \cos^2(\rho) - 1) l_{1a} \cos(\rho) \dot{q}_1$$

$$- \frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial d_3} = m_3 [d_3 \sin^2(\rho) + d_2 \cos(\rho) \sin(\rho)] \dot{q}_1^2 - m_3 g \cos(\rho)$$

$$\frac{d}{dt} \left[\frac{\partial \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{d}_3} \right] = m_3 \ddot{d}_3 - m_3(2 \cos^2(\rho) - 1) l_{1a} \cos(\rho) \ddot{q}_1$$

VI. Matriz de Coriolis

Dada la matriz de inercias:

$$M(\mathbf{q}) = \begin{bmatrix} I_{zz1} + I_{zz2} + I_{zz3} + m_1 l_{1a}^2 \cos^2(\rho) + m_2 [l_{1a}^2 + d_2^2] \cos^2(\rho) + d_3^2 \sin^2(\rho) + 2d_2 d_3 \cos(\rho) \sin(\rho) \\ [m_2 + 2m_3 \sin(\rho)] (l_{1a} \cos^2(\rho)) \\ -m_3(2 \cos^2(\rho) - 1) l_{1a} \cos(\rho) \end{bmatrix}$$

$$\begin{bmatrix} (m_2 + 2m_3 \sin(\rho)) & -m_3(2 \cos^2(\rho) - 1) \\ \times (l_{1a} \cos^2(\rho)) & \times l_{1a} \cos(\rho) \\ m_2 + m_3 & 0 \\ 0 & m_3 \end{bmatrix}$$

Siendo su derivada:

$$M'(\mathbf{q}) = \begin{bmatrix} 2[m_2 + m_3] d_2 \cos^2(\rho) \dot{d}_2 + 2m_3 d_3 \sin^2(\rho) \dot{d}_3 & 0 & 0 \\ + 2m_3 \cos(\rho) \sin(\rho) [d_2 \dot{d}_3 + d_2 \dot{d}_3] & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (29)$$

Y utilizando la ecuación (7), se obtiene:

$$\begin{aligned}
 & \begin{bmatrix} 2[m_2 + m_3]d_2 \cos^2(\rho)\dot{d}_2 + 2m_3d_3 \sin^2(\rho)\dot{d}_3 & 0 & 0 \\ + 2m_3 \cos(\rho) \sin(\rho)[\dot{d}_2d_3 + d_2\dot{d}_3] & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} (m_2 + 2m_3 \sin(\rho)) & -m_3(2 \cos^2(\rho) - 1) \\ \times (l_{1a} \cos^2(\rho)) & \times l_{1a} \cos(\rho) \\ m_2 + m_3 & 0 \\ 0 & m_3 \end{bmatrix} \\
 = & \begin{bmatrix} [m_2 + m_3]d_2 \cos^2(\rho)\dot{d}_2 + m_3d_3 \sin^2(\rho)\dot{d}_3 & 0 & 0 \\ + m_3 \cos(\rho) \sin(\rho)[\dot{d}_2d_3 + d_2\dot{d}_3] & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} \ddot{q}_1 \\ \ddot{d}_2 \\ \ddot{d}_3 \end{bmatrix} + \begin{bmatrix} [m_2 + m_3] \cos^2(\rho)d_2\dot{d}_2 & & \\ + m_3 \sin^2(\rho)d_3\dot{d}_3 & & 0 & 0 \\ + m_3 \cos(\rho) \sin(\rho) [\dot{d}_2d_3 + d_2\dot{d}_3] & & 0 & 0 \\ 0 & & 0 & 0 \\ 0 & & 0 & 0 \end{bmatrix} \\
 + & \begin{bmatrix} [m_2 + m_3]d_2 \cos^2(\rho)\dot{d}_2 + m_3d_3 \sin^2(\rho)\dot{d}_3 & 0 & 0 \\ + m_3 \cos(\rho) \sin(\rho)[\dot{d}_2d_3 + d_2\dot{d}_3] & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}^T \times \begin{bmatrix} \dot{q}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix} + g \begin{bmatrix} 0 \\ (m_2 + m_3) \sin \rho \\ -m_3 \cos \rho \end{bmatrix} + \begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix} \\
 \text{Por lo tanto, la matriz de coriolis queda como sigue:} \\
 C(\mathbf{q}, \dot{\mathbf{q}}) & + \begin{bmatrix} f_{c1} & 0 & 0 \\ 0 & f_{c2} & 0 \\ 0 & 0 & f_{c3} \end{bmatrix} \begin{bmatrix} \text{signo}(\dot{q}_1) \\ \text{signo}(\dot{d}_2) \\ \text{signo}(\dot{d}_3) \end{bmatrix} \\
 = & \begin{bmatrix} [m_2 + m_3]d_2 \cos^2(\rho)\dot{d}_2 + m_3d_3 \sin^2(\rho)\dot{d}_3 & 0 & 0 \\ + m_3 \cos(\rho) \sin(\rho)[\dot{d}_2d_3 + d_2\dot{d}_3] & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} f_{e1} & 0 & 0 \\ 0 & f_{e2} & 0 \\ 0 & 0 & f_{e3} \end{bmatrix} \begin{bmatrix} 1 - |\text{signo}(\dot{q}_1)| \\ 1 - |\text{signo}(\dot{d}_2)| \\ 1 - |\text{signo}(\dot{d}_3)| \end{bmatrix} \\
 & (30) \hspace{15em} (31)
 \end{aligned}$$

VII. Par gravitacional

Tomando en cuenta la ecuación (9), se tiene:

$$\begin{aligned}
 \frac{\partial U(\mathbf{q})}{\partial q_1} &= 0 \\
 \frac{\partial U(\mathbf{q})}{\partial d_2} &= (m_2 + m_3) \sin(\rho) \\
 \frac{\partial U(\mathbf{q})}{\partial d_3} &= -m_3 \cos(\rho)
 \end{aligned}$$

Por lo tanto, el vector de pares gravitacionales queda como:

$$g(\mathbf{q}) = \begin{bmatrix} 0 \\ (m_2 + m_3) \sin(\rho) \\ -m_3 \cos(\rho) \end{bmatrix}$$

VIII. Ecuaciones de movimiento Euler-Lagrange

En este último paso se obtiene finalmente el modelo dinámico del robot, siendo éste:

$$\begin{bmatrix} \tau_1 \\ f_{c2} \\ f_{c3} \end{bmatrix} = \begin{bmatrix} I_{zz1} + I_{zz2} + I_{zz3} + m_1 l_{1a}^2 \cos^2(\rho) \\ + m_2 [l_{1a} + d_2^2] \cos^2(\rho) + m_3 [(l_{1a}^2 + d_2^2) \cos^2(\rho) \\ + d_3^2 \sin^2(\rho) + 2d_2d_3 \cos(\rho) \sin(\rho)] \\ [m_2 + 2m_3 \sin(\rho)] (l_{1a} \cos^2(\rho)) \\ -m_3 (2 \cos^2(\rho) - 1) l_{1a} \cos(\rho) \end{bmatrix}$$

IX. Investigaciones futuras

El modelo dinámico del robot propuesto puede ser utilizado para desarrollar algoritmos de control avanzados, como el control predictivo basado en modelos (MPC) o el control adaptativo, para mejorar la precisión y eficiencia del robot.

X. Conclusiones

En este artículo se ha desarrollado el estudio de un robot RPP de tres grados de libertad partiendo de los valores de Euler-Denavit-Hartenberg para obtener el modelo cinemático y posteriormente concluir con el modelo dinámico.

La metodología de Euler-Denavit-Hartenberg facilita el proceso de obtención de las posiciones y sus derivadas, así como la velocidad de cada articulación en función de matrices antisimétricas. Las propiedades de estas matrices permiten simplificar los términos algebraicos, haciendo el álgebra más directa al momento de obtener la matriz de inercias y, posteriormente, el lagrangiano, cuya expresión es crucial para la derivación del modelo dinámico del robot.

La obtención del modelo dinámico de un robot manipulador constituye un elemento esencial para comprender su comportamiento y es la base fundamental para su control, cuya fundamentación se encuentra en el modelo cinemático. En el análisis de dicho modelo, se identifican diversos componentes que revelan las características específicas del robot, tales como la presencia matriz de fuerzas centrípetas y de Coriolis producto de la

articulación rotacional de la base, la fricción viscosa y la matriz de inercias, entre otros. Estas propiedades están interrelacionadas dentro del modelo completo del robot debido a la dinámica acoplada; es decir, lo que ocurre en el eslabón (i)-ésimo es consecuencia de la dinámica de sus eslabones subsecuentes.

Este prototipo puede ser implementado en la industria mediante la aplicación de un algoritmo de control adaptativo, permitiendo que el robot opere de manera óptima para cumplir funciones específicas según los requerimientos establecidos.

References

- [1] Reyes, Fernando, "MATLAB Aplicado a Robótica y Mecatrónica". Alfaomega, 2012.
- [2] Kelly, R. and Santibáñez, V. "Control de movimiento de robots manipuladores". Pearson Educación, 2003.
- [3] Takegaki M. and Arimoto S. "A new feedback method for dynamic control of manipulators". ASME J. Dyn. Syst. Meas. Control. 1981.
- [4] Vidyasagar M. "Nonlinear Systems Analysis". Prentice-Hall, Englewood Cliffs, NJ. 1993.
- [5] Murray R M, Li Z and Sastry S S. "A Mathematical Introduction to Robotic Manipulation"- CRC Press. 1994.
- [6] Horn Roger A and Charles R. Johnson. "Matrix analysis". Cambridge University Press. 1996.
- [7] Reyes F. and Kelly R. "A direct drive robot for control research". Proc. IASTED International Conference, Applications of Control and Robotics. Orlando. FL. 1996
- [8] Kelly R., V. Santibáñez y F. Reyes. "On saturated-proportional derivative feedback with adaptive gravity compensation of robot manipulators". Int. J. of Adaptive Control and Signal Processing. John Wiley & Sons, Ltd. 10(4): 1996
- [9] Reyes F. and Kelly R. "Experimental Evaluation of Identification Schemes on a Direct Drive Robot". Robotica. Cambridge University Press. 1997
- [10] Reyes F and Kelly R. "On parameter identification of robot manipulators". Proceedings of the 1997 IEEE International Conference on Robotics and Automation. Albuquerque, New Mexico. April 1997
- [11] J. Denavit and R.S. Hartenberg. "A kinematic notation for lower-pair mechanisms based on matrices". Transactions ASME J. Appl. Mech., 1955.

Control de un robot RPP

1st Moisés Gizeh Jiménez Hernández 2nd Jesús Misael González Tecó 3rd José Fernando Reyes Cortés

Facultad de Ciencias
de la Electrónica
BUAP

Puebla de Zaragoza, México
moises.jimenezh@alumno.buap.mx

Facultad de Ciencias
de la Electrónica
BUAP

Puebla de Zaragoza, México
misaelgonzalezteco@ieee.org

Facultad de Ciencias
de la Electrónica
BUAP

Puebla de Zaragoza, México
fernando.reyes@correo.buap.mx

Abstract—En el presente documento se implementa un control con compensación de gravedad a un robot manipulador de tres grados de libertad (gdl) con configuración rotacional-prismático-prismático (RPP), el comportamiento cinemático y dinámico han sido previamente modelado. Se implementa un algoritmo de control proporcional-derivativo (PD) que permita dirigir el robot hacia posiciones deseadas. La validez del control se demuestra mediante la existencia y unicidad del punto de equilibrio, complementado con un análisis de estabilidad basado en la regla de diseño de una función candidata de Lyapunov. Además, se lleva a cabo una sintonización orientada a mantener la energía de los servomotores por debajo de un umbral preestablecido.

Index Terms—control, equilibrio, Lyapunov, sintonización

metodología sistemática. El primer paso consiste en demostrar la existencia y unicidad del punto de equilibrio basado en el modelo dinámico del robot con la ecuación en lazo cerrado (modelo dinámico del robot manipulador más la estructura matemática del control de posición). Este análisis inicial asegura que el sistema pueda alcanzar y mantener un estado de equilibrio estable.

A continuación, se comprueba que la función de Lyapunov utilizada es definida positiva, lo cual garantiza que el punto de equilibrio identificado es estable. Asegurando que las perturbaciones del sistema se reducirán y que el robot podrá mantener un comportamiento predecible y controlado.

I. Introducción

El estudio y desarrollo de sistemas de control son fundamentales para garantizar precisión y estabilidad en una amplia gama de procesos, así como para regular las respuestas de señales y mejorar el rendimiento del sistema. En el campo de la robótica, estos sistemas son esenciales para la coordinación efectiva de los movimientos y acciones de los robots manipuladores. Además, permiten una retroalimentación continua de los sensores, lo que posibilita ajustes en tiempo real y, por ende, mejora la eficiencia operativa y la precisión.

El control o regulador de posición de robots manipuladores, busca colocar el extremo final del robot en una posición deseada \mathbf{q}_d (constante en el tiempo) a partir de cualquier condición inicial $[\mathbf{q}(0), \dot{\mathbf{q}}(0)] \in \mathbb{R}^{2n}$. El objetivo es el diseño de un regulador τ tal que la velocidad de movimiento $\dot{\mathbf{q}}(t)$ y el error de posición $\tilde{\mathbf{q}}(t)$ converjan asintóticamente a cero, es decir,

$$\lim_{t \rightarrow \infty} \begin{bmatrix} \tilde{\mathbf{q}}(t) \\ \dot{\tilde{\mathbf{q}}}(t) \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{2n}, \forall t \geq 0.$$

[3]

En el diseño de un sistema de control adecuado para un robot manipulador, es necesario seguir una

Una vez demostrada la existencia del punto de equilibrio y su estabilidad, se retoma el modelo dinámico del robot y se adapta el esquema de control a los tres grados de libertad del prototipo, introduciendo parámetros como distancias, masas y tiempos de simulación. Esta fase implica una simulación detallada utilizando el método ode45 Runge Kutta en **MATLAB** [1], que permite observar cómo responde el sistema bajo diferentes condiciones y ajustes.

Para asegurar que las ganancias de control obtenidas no provoquen saturación ni mal funcionamiento de los servomotores, es necesario diseñar una regla de sintonía adecuada. Este proceso de sintonía implica encontrar los valores numéricos óptimos para cada ganancia de control, de modo que el sistema funcione de manera eficiente y sin comprometer la integridad de los componentes físicos [1].

El algoritmo de control proporcional-derivativo (PD) es uno de los esquemas de control más simples y populares que se pueden utilizar en robots manipuladores. En 1981, Takegaki y Arimoto demostraron que la ecuación en lazo cerrado del modelo dinámico del robot manipulador con control proporcional-derivativo tiene un punto de equilibrio global y asintóticamente estable. La ley de

control PD más la compensación de gravedad está dada por la siguiente ecuación:

$$\tau = K_p \tilde{\mathbf{q}} - K_v \dot{\mathbf{q}} + g(\mathbf{q}),$$

[4]

donde $\tilde{\mathbf{q}} \in \mathbb{R}^n$ es el vector error de posición, que se define como la diferencia entre la posición deseada y la posición actual del robot, $K_p \in \mathbb{R}^{n \times n}$ y $K_v \in \mathbb{R}^{n \times n}$ corresponden a la ganancia proporcional y derivativa respectivamente, ambas definidas positivas [4].

La ecuación en lazo cerrado en variables de estado que define el problema de control de posición es:

$$\frac{d}{dt} \begin{bmatrix} \tilde{\mathbf{q}} \\ \dot{\tilde{\mathbf{q}}} \end{bmatrix} = \begin{bmatrix} -\dot{\tilde{\mathbf{q}}} \\ M(\mathbf{q})^{-1} [K_p \tilde{\mathbf{q}} - K_v \dot{\tilde{\mathbf{q}}} - C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\tilde{\mathbf{q}}}] \end{bmatrix},$$

la cual es una ecuación diferencial autónoma [3].

II. Modelo dinámico y control proporcional derivativo

Para el análisis, se considera un robot de 3 grados de libertad (gdl) con configuración rotacional-prismático-prismático (RPP) como se muestra en la Fig. 1 utilizando los parámetros de Table 1 con fines de simulación.

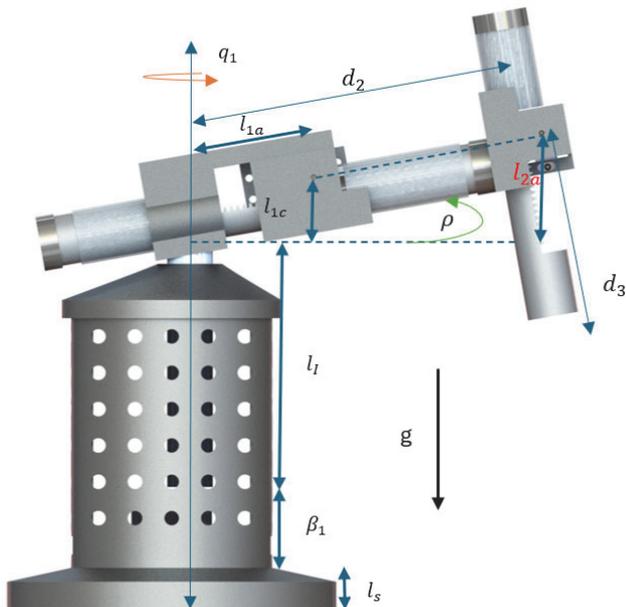


Fig. 1. Prototipo de robot 3gdl RPP.

TABLE I
Parámetros Robot RPP.

Eslabón	Parámetros del robot de 3 gdl.		
	Significado	Notación	Valor
Hombro	Masa 1.	m_1	3.25 Kg
	Longitud 1.	l_s	0.02291 m
	Longitud 2.	l_I	0.13 m
	Alt. Servomotor.	β_1	0.0401 m
	Inercia 1.	I_{z1z1}	$1.26 Nm \frac{seg^2}{rad}$
	Coef. fricc. visc. 1.	b_1	$2.288 Nm \frac{seg}{rad}$
	Lím. de Sat. 1.	τ_1^{max}	120 Nm
Codo	Masa 2.	m_2	0.85 Kg
	Cent. de masa 1.	l_{1a}	0.07382 m
	Cent. de masa 2.	l_{1c}	0.03814 m
	Eslabon 2.	d_2	0.17137 m
	Inercia 2.	I_{z2z2}	$0.15 Nm \frac{seg^2}{rad}$
	Inclinación.	ρ	10°
	Coef. fricc. visc. 2.	b_2	$0.175 Nm \frac{seg}{rad}$
Lím. de Sat. 2.	τ_2^{max}	150 Nm	
Muñeca	Masa 3.	m_3	0.35 Kg
	Longitud 3.	l_{2a}	.07224 m
	Eslabon 3.	d_3	0.1015 m
	Inercia 3.	I_{z3z3}	$0.093 Nm \frac{seg^2}{rad}$
	Coef. fricc. visc. 3.	b_3	$0.165 Nm \frac{seg}{rad}$
Lím. de Sat. 3.	τ_3^{max}	50 Nm	

^aDatos de un modelo digital.

A continuación se presenta el modelo dinámico de un robot prototipo de 3gdl considerando la aceleración debido a la gravedad (g) como $9.81 \frac{m}{seg^2}$ [3]:

$$\tau = M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + f_f(\dot{\mathbf{q}}, f_e)$$

donde:

- $\mathbf{q} \in \mathbb{R}^n$ es el vector de coordenadas generalizadas o posiciones articulares.
- $\dot{\mathbf{q}} \in \mathbb{R}^n$ es el vector de velocidades articulares.
- $\ddot{\mathbf{q}} \in \mathbb{R}^n$ es el vector de aceleraciones articulares.
- $M(\mathbf{q}) \in \mathbb{R}^{n \times n}$ es la matriz de inercia, la cual es simétrica y definida positiva.
- $C(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ es la matriz de fuerzas centrípetas y de Coriolis.
- $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ es el vector de fuerzas o pares gravitacionales.
- $f_f(\dot{\mathbf{q}}, f_e) \in \mathbb{R}^n$ es el vector de pares de fricción que influye la fricción viscosa, de Coulomb y estática en cada articulación del robot.

$$\begin{aligned}
 & \begin{bmatrix} \tau_1 \\ f_{c2} \\ f_{c3} \end{bmatrix} \\
 &= \begin{bmatrix} I_{zz1} + I_{zz2} + I_{zz3} + m_1 l_{1a}^2 \cos^2(\rho) \\ + m_2 [l_{1a} + d_2^2] \cos^2(\rho) + m_3 [(l_{1a}^2 + d_2^2) \cos^2(\rho) \\ + d_3^2 \sin^2(\rho) + 2d_2 d_3 \cos(\rho) \sin(\rho)] \\ \\ [m_2 + 2m_3 \sin(\rho)] (l_{1a} \cos^2(\rho)) \\ -m_3 (2 \cos^2(\rho) - 1) l_{1a} \cos(\rho) \end{bmatrix} \\
 & \begin{bmatrix} (m_2 + 2m_3 \sin(\rho)) & -m_3 (2 \cos^2(\rho) - 1) \\ \times (l_{1a} \cos^2(\rho)) & \times l_{1a} \cos(\rho) \\ m_2 + m_3 & 0 \\ 0 & m_3 \end{bmatrix} \\
 & \times \begin{bmatrix} \ddot{q}_1 \\ \ddot{d}_2 \\ \ddot{d}_3 \end{bmatrix} + \begin{bmatrix} [m_2 + m_3] \cos^2(\rho) d_2 \dot{d}_2 \\ + m_3 \sin^2(\rho) d_3 \dot{d}_3 & 0 & 0 \\ + m_3 \cos(\rho) \sin(\rho) [\dot{d}_2 d_3 + d_2 \dot{d}_3] & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
 & \times \begin{bmatrix} \dot{q}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix} + g \begin{bmatrix} 0 \\ (m_2 + m_3) \sin \rho \\ -m_3 \cos \rho \end{bmatrix} + \begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_2 & 0 \\ 0 & 0 & b_3 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{bmatrix} \\
 & + \begin{bmatrix} f_{c1} & 0 & 0 \\ 0 & f_{c2} & 0 \\ 0 & 0 & f_{c3} \end{bmatrix} \begin{bmatrix} \text{signo}(\dot{q}_1) \\ \text{signo}(\dot{d}_2) \\ \text{signo}(\dot{d}_3) \end{bmatrix} \\
 & + \begin{bmatrix} f_{e1} & 0 & 0 \\ 0 & f_{e2} & 0 \\ 0 & 0 & f_{e3} \end{bmatrix} \begin{bmatrix} 1 - |\text{signo}(\dot{q}_1)| \\ 1 - |\text{signo}(\dot{d}_2)| \\ 1 - |\text{signo}(\dot{d}_3)| \end{bmatrix} \quad (1)
 \end{aligned}$$

Se implementa un algoritmo de control proporcional - derivativo (PD) en el robot manipulador de 3 grados de libertad (gdl) para desarrollar su controlador:

$$\begin{aligned}
 \tau &= K_p \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{q}^2}}{\tilde{q}^2 + e^{-\alpha \tilde{q}^2}} \right) \tilde{\mathbf{q}} \\
 &- K_v \text{diag} \left(\frac{1 - \alpha e^{-\rho \dot{\tilde{q}}^2}}{\dot{\tilde{q}}^2 + e^{-\rho \dot{\tilde{q}}^2}} \right) \dot{\tilde{\mathbf{q}}} + \mathbf{g}(\mathbf{q}) \quad (2)
 \end{aligned}$$

- 1) $K_p \in \mathbb{R}^{n \times n}$ es la ganancia proporcional, matriz diagonal definida positiva.
- 2) $K_v \in \mathbb{R}^{n \times n}$ es la ganancia derivativa, matriz diagonal definida positiva.
- 3) La siguiente notación se utiliza:

$$K_p \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{q}^2}}{\tilde{q}^2 + e^{-\alpha \tilde{q}^2}} \right) =$$

$$\begin{bmatrix} \left(\frac{1 - \alpha e^{-\alpha \tilde{q}^2}}{\tilde{q}^2 + e^{-\alpha \tilde{q}^2}} \right) & 0 & \dots & 0 \\ 0 & \left(\frac{1 - \alpha e^{-\alpha \tilde{q}^2}}{\tilde{q}^2 + e^{-\alpha \tilde{q}^2}} \right) & \dots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \dots & \left(\frac{1 - \alpha e^{-\alpha \tilde{q}^2}}{\tilde{q}^2 + e^{-\alpha \tilde{q}^2}} \right) \end{bmatrix}$$

$$K_v \text{diag} \left(\frac{1 - \alpha e^{-\rho \dot{\tilde{q}}^2}}{\dot{\tilde{q}}^2 + e^{-\rho \dot{\tilde{q}}^2}} \right) =$$

$$\begin{bmatrix} \left(\frac{1 - \alpha e^{-\rho \dot{\tilde{q}}^2}}{\dot{\tilde{q}}^2 + e^{-\rho \dot{\tilde{q}}^2}} \right) & 0 & \dots & 0 \\ 0 & \left(\frac{1 - \alpha e^{-\rho \dot{\tilde{q}}^2}}{\dot{\tilde{q}}^2 + e^{-\rho \dot{\tilde{q}}^2}} \right) & \dots & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \dots & \left(\frac{1 - \alpha e^{-\rho \dot{\tilde{q}}^2}}{\dot{\tilde{q}}^2 + e^{-\rho \dot{\tilde{q}}^2}} \right) \end{bmatrix}$$

1) $0 < \alpha < 1; 0 < \rho < 1.$

2) $\tilde{\mathbf{q}} = \begin{bmatrix} \tilde{q}_1 \\ \tilde{q}_2 \\ \tilde{q}_3 \end{bmatrix}$ Donde $\tilde{q}_i = q_{di} - q_i; i = 1, 2, 3.$

3) $\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}.$

III. Existencia y unicidad del punto de equilibrio

El esquema de control (regulador) proporcional derivativo (PD) está dado por:

$$\begin{aligned}
 \tau &= K_p \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{q}^2}}{\tilde{q}^2 + e^{-\alpha \tilde{q}^2}} \right) \tilde{\mathbf{q}} \\
 &- K_v \text{diag} \left(\frac{1 - \alpha e^{-\rho \dot{\tilde{q}}^2}}{\dot{\tilde{q}}^2 + e^{-\rho \dot{\tilde{q}}^2}} \right) \dot{\tilde{\mathbf{q}}} + \mathbf{g}(\mathbf{q}) \\
 &= M(\mathbf{q}) \ddot{\tilde{\mathbf{q}}} + C(\mathbf{q}, \dot{\tilde{\mathbf{q}}}) + B \dot{\tilde{\mathbf{q}}} + \mathbf{g}(\mathbf{q}) \quad (3)
 \end{aligned}$$

Matemáticamente obtenemos una compensación de la gravedad en la ecuación (3).

Tomamos el vector de errores, despejamos \mathbf{q} :

$$\mathbf{q} = \mathbf{q}_d - \tilde{\mathbf{q}}$$

Obtenemos: $\dot{\tilde{\mathbf{q}}} = -\dot{\tilde{\mathbf{q}}}$.

La ecuación de lazo cerrado (modelo dinámico del robot manipulador más la estructura matemática del control de posición) se encuentra expresada como:

$$\frac{d}{dt} \begin{bmatrix} \tilde{\mathbf{q}} \\ \dot{\tilde{\mathbf{q}}} \end{bmatrix} = \begin{bmatrix} -\dot{\tilde{\mathbf{q}}} \\ M^{-1}(q_d - \tilde{\mathbf{q}}) \left[K_p \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{q}^2}}{\tilde{q}^2 + e^{-\alpha \tilde{q}^2}} \right) \tilde{\mathbf{q}} - K_v \text{diag} \left(\frac{1 - \alpha e^{-\rho \dot{\tilde{q}}^2}}{\dot{\tilde{q}}^2 + e^{-\rho \dot{\tilde{q}}^2}} \right) \dot{\tilde{\mathbf{q}}} - C(q_d - \tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}) \dot{\tilde{\mathbf{q}}} - B \dot{\tilde{\mathbf{q}}} \right] \end{bmatrix} \quad (4)$$

La demostración de existencia y unicidad del punto de equilibrio $[0, 0]^T \in \mathbb{R}^{2 \times n}$ es la siguiente:

1) Para $\tilde{\mathbf{q}}$:

$$-\dot{\mathbf{q}} = -I\dot{\mathbf{q}} = \mathbf{0} \Leftrightarrow \dot{\mathbf{q}} = \mathbf{0} \quad (5)$$

2) Para $\dot{\mathbf{q}}$

$$M^{-1}(q_d - \tilde{\mathbf{q}}) \begin{bmatrix} K_p \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) \tilde{\mathbf{q}} \\ -K_v \text{diag} \left(\frac{1 - \alpha e^{-\alpha \dot{\mathbf{q}}^2}}{\dot{\mathbf{q}}^2 + e^{-\alpha \dot{\mathbf{q}}^2}} \right) \dot{\mathbf{q}} \\ -C(q_d - \tilde{\mathbf{q}}, \dot{\mathbf{q}}) \dot{\mathbf{q}} \\ -B\dot{\mathbf{q}} \end{bmatrix} = \mathbf{0} \quad (6)$$

Aplicamos ($\dot{\mathbf{q}} = 0$), simplificando la expresión queda por demostrar:

$$M^{-1}(q_d - \tilde{\mathbf{q}}) \left[K_p \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) \tilde{\mathbf{q}} \right] = 0$$

$M(q_d - \tilde{\mathbf{q}}) > 0 \Rightarrow M^{-1}(q_d - \tilde{\mathbf{q}}) > 0$, dado que la matriz es definida positiva y diagonal.

Por definición, K_v y K_p son matrices diagonales definidas positivas, por lo tanto, no pueden ser cero, ya que cumplen los criterios de Sylvester:

Definimos a la matriz $A \in \mathbb{R}^{n \times n}$, donde A denota a las matrices K_v y K_p :

- 1) La matriz es simétrica: $A = A^T$.
- 2) El primer elemento de la matriz A , a_{11} , debe ser positivo, es decir, $a_{11} > 0$.
- 3) Todos los determinantes menores deben ser positivos, consecuentemente el determinante de la matriz $|A| > 0$.

El único caso que queda por demostrar es que:

$$\text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) \tilde{\mathbf{q}} = 0$$

Para que esto se cumpla, se necesita cumplir alguna de las siguientes condiciones:

$$\text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) = 0 \quad \text{o} \quad \tilde{\mathbf{q}} = 0$$

Analizamos:

$$\text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) > 0$$

La expresión $e^{-\alpha \tilde{\mathbf{q}}^2} > 0$.

Por su parte, $\tilde{\mathbf{q}}^2 > 0$ para cualquier $\tilde{\mathbf{q}} \neq 0$.

$$\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2} > 0$$

Analizamos $1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}$:

$$0 < 1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2} < 1$$

Podemos concluir que:

$$\text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) > 0$$

$$\therefore K_p \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) \tilde{\mathbf{q}} = 0 \Leftrightarrow \tilde{\mathbf{q}} = 0$$

Por lo tanto, el punto de equilibrio es:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \mathbb{R}^{2 \times n}$$

La ecuación de lazo cerrado tiene un punto de equilibrio que existe y es único.

IV. Función de Lyapunov

La función candidata de Lyapunov tiene como regla de diseño la ecuación:

$$V(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T M(\mathbf{q}) \dot{\mathbf{q}} + U_a(K_p, \tilde{\mathbf{q}}) \quad (7)$$

La técnica de moldeo de energía nos permite diseñar una familia extensa de esquemas de control para un sistema dinámico, en este caso; un robot de 3 grados de libertad (gdl) con configuración RPP, tenemos:

$$\tau = \nabla U_a(K_p, \tilde{\mathbf{q}}) - f_v(K_v, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q})$$

Integramos $\int \nabla U_a(K_p, \tilde{\mathbf{q}}) d\tilde{\mathbf{q}}$

$$\int \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) \tilde{\mathbf{q}} d\tilde{\mathbf{q}}$$

Obtenemos como resultado:

$$\int \nabla U_a(K_p, \tilde{\mathbf{q}}) d\tilde{\mathbf{q}} = \frac{1}{2} \text{In} |\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}| + C$$

Con $c = 0$, tenemos:

$$V(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T M(q_d - \tilde{\mathbf{q}}) \dot{\mathbf{q}} + \begin{bmatrix} \frac{1}{2} \text{In} |\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}| \\ \frac{1}{2} \text{In} |\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}| \\ \frac{1}{2} \text{In} |\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}| \end{bmatrix}^T K_p \begin{bmatrix} \frac{1}{2} \text{In} |\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}| \\ \frac{1}{2} \text{In} |\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}| \\ \frac{1}{2} \text{In} |\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}| \end{bmatrix} \quad (8)$$

Es posible reescribir la ecuación (8) matricialmente como una función candidata de Lyapunov ($V(\mathbf{x}) = \mathbf{x}^T P \mathbf{x}$, $x \in \mathbb{R}^n$, $P \in \mathbb{R}^{n \times n}$) [1]:

$$\left[\begin{array}{c} \dot{\mathbf{q}} \\ \left[\begin{array}{c} \frac{1}{2} I_n \tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2} \\ \frac{1}{2} I_n \tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2} \\ \frac{1}{2} I_n \tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2} \end{array} \right] \end{array} \right]^T \left[\begin{array}{cc} M(q_d - \tilde{\mathbf{q}}) & 0 \\ 0 & K_p \end{array} \right] \left[\begin{array}{c} \dot{\mathbf{q}} \\ \left[\begin{array}{c} \frac{1}{2} I_n \tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2} \\ \frac{1}{2} I_n \tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2} \\ \frac{1}{2} I_n \tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2} \end{array} \right] \end{array} \right] \quad (9)$$

Analizando nuestra función candidata de Lyapunov (9) es posible observar que la matriz $\begin{bmatrix} M(q_d - \tilde{\mathbf{q}}) & 0 \\ 0 & K_p \end{bmatrix} > 0$ puesto que cumple los criterios de Sylvester.

Derivamos (8) con respecto al tiempo, aplicamos $\dot{\mathbf{q}} = -\dot{\tilde{\mathbf{q}}}$:

$$\dot{V}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{q}}^T M(q_d - \tilde{\mathbf{q}}) \dot{\mathbf{q}} + \frac{1}{2} \dot{\mathbf{q}}^T \dot{M}(q_d - \tilde{\mathbf{q}}) \dot{\mathbf{q}} - \dot{\mathbf{q}}^T K_p \dot{\mathbf{q}}$$

Sustituimos $\tilde{\mathbf{q}}$, $\tilde{\mathbf{q}} = \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) \tilde{\mathbf{q}}$ y $\mathbf{q} = q_d - \tilde{\mathbf{q}}$, y simplificamos al aplicar la propiedad:

$$\frac{1}{2} \dot{\mathbf{q}}^T [\dot{M}(q_d - \tilde{\mathbf{q}}) - 2C(q_d - \tilde{\mathbf{q}}, \dot{\mathbf{q}})] \dot{\mathbf{q}} \equiv 0$$

Obtenemos la ecuación:

$$\dot{V}(\mathbf{q}, \dot{\mathbf{q}}) = -\dot{\mathbf{q}}^T K_v \text{diag} \left(\frac{1 - \alpha e^{-\rho \dot{\mathbf{q}}^2}}{\dot{\mathbf{q}}^2 + e^{-\rho \dot{\mathbf{q}}^2}} \right) \dot{\mathbf{q}} - \dot{\mathbf{q}}^T B \dot{\mathbf{q}}$$

Factorizamos:

$$\dot{V}(\mathbf{q}, \dot{\mathbf{q}}) = -\dot{\mathbf{q}}^T \left[K_v \text{diag} \left(\frac{1 - \alpha e^{-\rho \dot{\mathbf{q}}^2}}{\dot{\mathbf{q}}^2 + e^{-\rho \dot{\mathbf{q}}^2}} \right) - B \right] \dot{\mathbf{q}} < 0 \quad (10)$$

A la ecuación (10) se le conoce como la ecuación algebraica de Lyapunov $\dot{V}(x) = -\mathbf{x}^T Q \mathbf{x} < 0$, donde la matriz $Q \in \mathbb{R}^{n \times n}$ es una matriz definida positiva [1].

Y por lo tanto nuevamente podemos concluir: $\dot{\mathbf{q}}^T \left[K_v \text{diag} \left(\frac{1 - \alpha e^{-\rho \dot{\mathbf{q}}^2}}{\dot{\mathbf{q}}^2 + e^{-\rho \dot{\mathbf{q}}^2}} \right) - B \right] \dot{\mathbf{q}} > 0$

V. Regla de sintonía

La regla de sintonía tiene la finalidad de operar a los servoamplificadores en la parte lineal, es decir, no saturar sus límites físicos $\tau_i^{\text{máx}}$, $i = 1, 2, 3$. La sintonía de las ganancias proporcional queda en función del troque máximo y de la condición inicial: $[\tilde{q}_i(0), \dot{q}_i(0)]^T$, por su parte la ganancia derivativa es un cierto porcentaje de la ganancia proporcional [1].

Acotamos el algoritmo de control (2):

$$\|\tau\| = \|K_p \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) \tilde{\mathbf{q}} - K_v \text{diag} \left(\frac{1 - \alpha e^{-\rho \dot{\mathbf{q}}^2}}{\dot{\mathbf{q}}^2 + e^{-\rho \dot{\mathbf{q}}^2}} \right) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})\| \quad (11)$$

Tenemos:

$$\begin{aligned} & \left\| K_p \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) \tilde{\mathbf{q}} - K_v \text{diag} \left(\frac{1 - \alpha e^{-\rho \dot{\mathbf{q}}^2}}{\dot{\mathbf{q}}^2 + e^{-\rho \dot{\mathbf{q}}^2}} \right) \dot{\mathbf{q}} \right. \\ & \left. + \mathbf{g}(\mathbf{q}) \right\| \leq \left\| K_p \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) \tilde{\mathbf{q}} \right\| \\ & \quad + \left\| K_v \text{diag} \left(\frac{1 - \alpha e^{-\rho \dot{\mathbf{q}}^2}}{\dot{\mathbf{q}}^2 + e^{-\rho \dot{\mathbf{q}}^2}} \right) \dot{\mathbf{q}} \right\| + \|\mathbf{g}(\mathbf{q})\| \end{aligned}$$

Continuando:

$$\begin{aligned} & \left\| K_p \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) \tilde{\mathbf{q}} \right\| + \left\| K_v \text{diag} \left(\frac{1 - \alpha e^{-\rho \dot{\mathbf{q}}^2}}{\dot{\mathbf{q}}^2 + e^{-\rho \dot{\mathbf{q}}^2}} \right) \dot{\mathbf{q}} \right\| \\ & + \|\mathbf{g}(\mathbf{q})\| \leq \lambda_{K_p}^{\text{max}} \left\| \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) \tilde{\mathbf{q}} \right\| \\ & + \lambda_{K_v}^{\text{max}} \left\| \text{diag} \left(\frac{1 - \alpha e^{-\rho \dot{\mathbf{q}}^2}}{\dot{\mathbf{q}}^2 + e^{-\rho \dot{\mathbf{q}}^2}} \right) \dot{\mathbf{q}} \right\| + K_g \end{aligned}$$

Cada norma contiene vectores, por lo tanto:

$$\begin{aligned} \left\| \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) \tilde{\mathbf{q}} \right\| &= \sqrt{\sum_{i=1}^n \left(\frac{1 - \alpha e^{-\alpha \tilde{q}_i^2}}{\tilde{q}_i^2 + e^{-\alpha \tilde{q}_i^2}} \tilde{q}_i \right)^2} \\ \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) &< 1 \end{aligned}$$

$$\Rightarrow \sqrt{\sum_{i=1}^n \left(\frac{1 - \alpha e^{-\alpha \tilde{q}_i^2}}{\tilde{q}_i^2 + e^{-\alpha \tilde{q}_i^2}} \tilde{q}_i \right)^2} \leq \sqrt{\sum_{i=1}^n (\tilde{q}_i)^2} \quad (12)$$

$$\begin{aligned} \left\| -K_v \text{diag} \left(\frac{1 - \alpha e^{-\rho \dot{\mathbf{q}}^2}}{\dot{\mathbf{q}}^2 + e^{-\rho \dot{\mathbf{q}}^2}} \right) \dot{\mathbf{q}} \right\| &= \sqrt{\sum_{i=1}^n \left(\frac{1 - \rho e^{-\rho \dot{q}_i^2}}{\dot{q}_i^2 + e^{-\rho \dot{q}_i^2}} \dot{q}_i \right)^2} \\ \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) &< 1 \\ \Rightarrow \sqrt{\sum_{i=1}^n \left(\frac{1 - \rho e^{-\rho \dot{q}_i^2}}{\dot{q}_i^2 + e^{-\rho \dot{q}_i^2}} \dot{q}_i \right)^2} &\leq \sqrt{\sum_{i=1}^n (\dot{q}_i)^2} \quad (13) \end{aligned}$$

Entonces:

$$\left\| K_p \text{diag} \left(\frac{1 - \alpha e^{-\alpha \tilde{\mathbf{q}}^2}}{\tilde{\mathbf{q}}^2 + e^{-\alpha \tilde{\mathbf{q}}^2}} \right) \tilde{\mathbf{q}} \right\| + \left\| K_v \text{diag} \left(\frac{1 - \alpha e^{-\rho \dot{\mathbf{q}}^2}}{\dot{\mathbf{q}}^2 + e^{-\rho \dot{\mathbf{q}}^2}} \right) \dot{\mathbf{q}} \right\| + \|g(\mathbf{q})\| \leq \lambda_{K_p}^{\max} \|\tilde{\mathbf{q}}\| + \lambda_{K_v}^{\max} \|\dot{\mathbf{q}}\| + K_g$$

Finalmente:

$$\lambda_{K_p}^{\max} \|\tilde{\mathbf{q}}\| + \lambda_{K_v}^{\max} \|\dot{\mathbf{q}}\| + K_g \leq \|\tau^{max}\| \quad (14)$$

Proponemos a $\lambda_{K_v}^{\max} = \epsilon \lambda_{K_p}^{\max}$, $0 < \epsilon < 1$, factorizamos y obtenemos:

$$\lambda_{K_p}^{\max} \leq \frac{\|\tau^{max}\| - \|g(\mathbf{q})\|}{\|\tilde{\mathbf{q}}\| + \epsilon \|\dot{\mathbf{q}}\|} \quad (15)$$

Aplicamos un criterio de 0.8 para permanecer lejos de la región de saturación donde encontramos vibración mecánica, dinámica no modelada y juego mecánico. Este criterio nos va a asegurar trabajar el driver en la zona lineal.

$$\lambda_{K_p}^{\max} \leq (0.8) \frac{\|\tau^{max}\| - \|g(\mathbf{q})\|}{\|\tilde{\mathbf{q}}\| + \epsilon \|\dot{\mathbf{q}}\|} \quad (16)$$

VI. Simulación

Para propósitos de simulación se requiere el modelo dinámico numérico del robot manipulador, el cual se obtuvo al reemplazar cada uno de los parámetros correspondientes del modelo dinámico (1) en **MATLAB**, sin tomar en cuenta a la fricción estática y de Coulomb. Proponemos $\epsilon = 0.5$, $\alpha = 0.7$, $\rho = 0.5$, $\tau^{max} = [120, 150, 50]^T$, $K(g) = [0, 0, 0]^T$, $\tilde{\mathbf{q}}(0) = [\frac{\pi}{2}, 0.15, 0.05]^T$.

Obtenemos los valores de $K_{p1} \approx 60.8082$ y $K_{v1} \approx 30.4041$ para el primer eslabón, $K_{p2} \approx 76.0102$ y $K_{v2} \approx 38.0051$ para el segundo eslabón y $K_{p3} \approx 25.336$ y $K_{v3} \approx 12.668$ para el último eslabón.

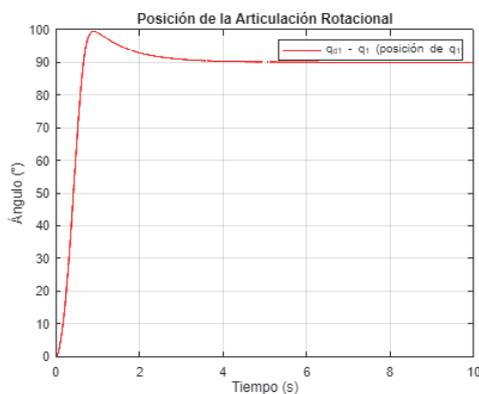


Fig. 2. Posición articulación rotacional

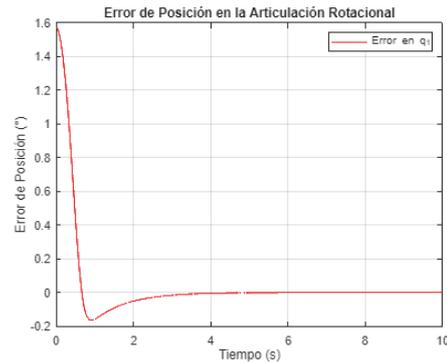


Fig. 3. Comportamiento del error articulación rotacional

En la Fig. 2 se muestra cómo el algoritmo de control proporcional-derivativo (PD) ajusta la posición de la primera articulación rotacional desde su posición inicial hacia un punto deseado en su espacio de trabajo, que en este caso es 90° . El control sigue con precisión la referencia deseada. En la Fig. 3, se observa que, a medida que la articulación se aproxima al punto objetivo, el error se reduce progresivamente hasta alcanzar un valor de cero.

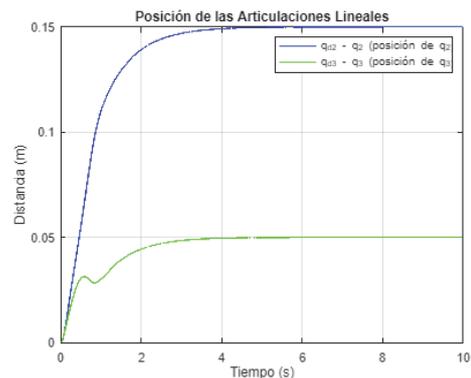


Fig. 4. Posición de las articulaciones lineales

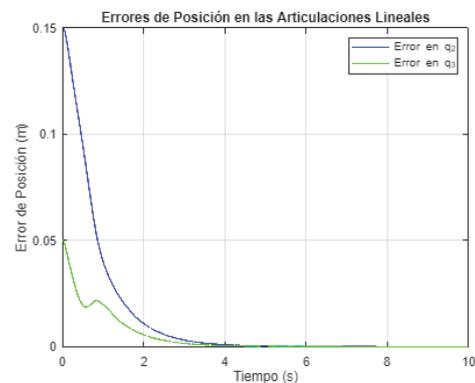


Fig. 5. Error en las articulaciones primáticas

Las articulaciones prismáticas del sistema, partiendo desde la posición inicial hacia el punto deseado, presentan el comportamiento mostrado en la Fig. 4. El pico observado en la posición de la articulación 3 se atribuye al efecto Coriolis, un fenómeno relevante en la dinámica de sistemas robóticos con movimiento acoplado. La Coriolis surge de la interacción entre las velocidades angulares y lineales, especialmente cuando se produce un cambio rápido en la velocidad de las articulaciones, a medida que las articulaciones se aproximan al punto deseado, el error se reduce hasta alcanzar un valor de cero.

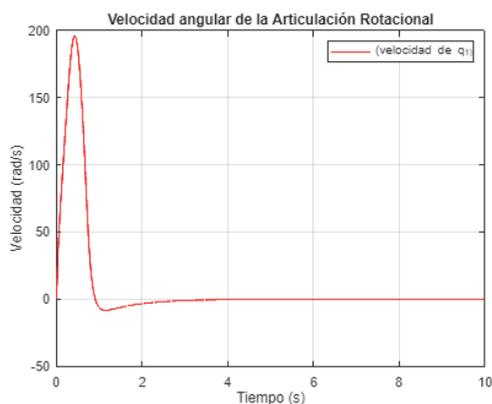


Fig. 6. Velocidad angular articulación rotacional

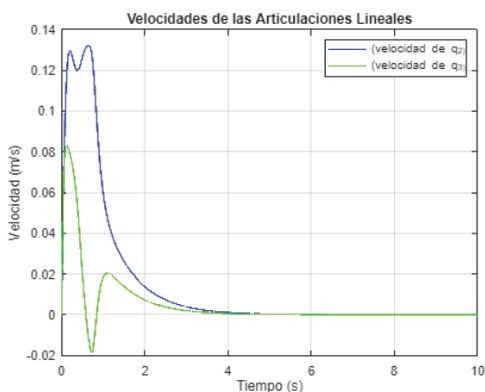


Fig. 7. Velocidad de las articulaciones lineales

A medida que las articulaciones alcanzan el punto deseado, la velocidad de la articulación rotacional Fig. 6 y de las articulaciones lineales Fig. 7 se aproximan a cero, ya que el robot mantiene una posición constante sin experimentar cambios adicionales.

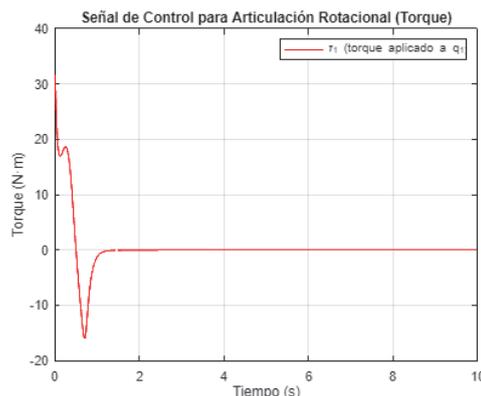


Fig. 8. Control articulación rotacional

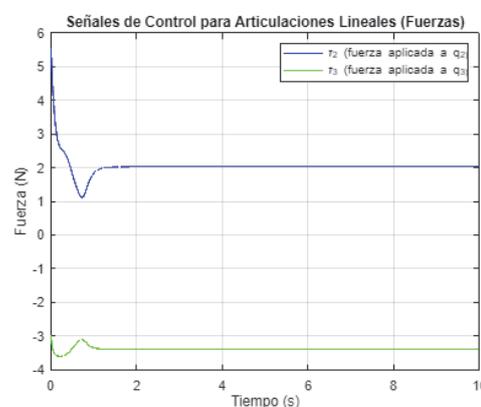


Fig. 9. Control articulaciones lineales

Una vez que la articulación rotacional alcanza su posición deseada, la señal de torque Fig. 8 se anula, ya que no hay fuerzas externas que influyan en el comportamiento de esta articulación. Sin embargo, las dos articulaciones lineales requieren de una pequeña fuerza constante Fig. 9 debido a la influencia de la gravedad sobre ambas.

VII. Investigaciones futuras

El algoritmo de control proporcional derivativo (PD) empleado en este estudio tiene un amplio rango de aplicaciones y no se limita al sistema aquí analizado. Puede ser implementado en robots con múltiples grados de libertad y diversas configuraciones de articulaciones. Además, la sintonización del control, permite la selección arbitraria de algunos valores, lo que ofrece flexibilidad al usuario para ajustar el rendimiento del sistema según observaciones específicas, lo que podría conducir a mejoras significativas en su desempeño.

VIII. Conclusiones

El algoritmo de control propuesto cumple su función en el manejo de la posición del robot, permitiendo que el punto final alcance cualquier posición deseada dentro de su rango operativo. Sin embargo, para evitar la saturación del motor, es esencial establecer una regla de sintonización que garantice el funcionamiento dentro de la zona lineal del sistema. En ausencia de esta precaución, los torques pueden exceder los límites físicos de los motores, especialmente si los valores de K_p y K_v no son óptimos. Por lo tanto, es crucial que la sintonización de los controladores se realice considerando las capacidades específicas de los motores. Adicionalmente, los valores de α , ρ y ϵ fueron seleccionados de manera arbitraria en este estudio. Esta elección intencionada permite al usuario explorar diferentes configuraciones para identificar aquellas que puedan mejorar el rendimiento. Este enfoque promueve una sintonización empírica, alentando al usuario a ajustar los parámetros basándose en observaciones prácticas y resultados experimentales, lo que puede llevar a un desempeño óptimo del sistema de control.

References

- [1] Reyes Cortés F. "Robótica: Control de robots manipuladores". Alfa Omega. 2011.
- [2] D. A. Wells, "Dinámica de Lagrange con un Estudio de Ecuaciones del Movimiento de Euler: Principio y Ecuaciones de Hamilton", Serie de Compendios Schaum. New York, NY, USA: McGraw-Hill, 1972.
- [3] Kelly, R. and Santibáñez, V. "Control de movimiento de robots manipuladores". Pearson Educación, 2003.
- [4] Takegaki M. and Arimoto S. "A new feedback method for dynamic control of manipulators". ASME J. Dyn. Syst. Meas. Control. 1981.
- [5] J. F. Reyes Cortés, M. A. Limón Díaz, y E. J. González Galván, "Regulación Saturada con Ganancia Variable Derivativa de Robots Manipuladores", Revista Iberoamericana de Automática e Informática Industrial, [En línea]. Disponible en: <https://polipapers.upv.es/index.php/RIAI/article/view/9195/9237>.
- [6] Kelly R., V. Santibáñez y F. Reyes. "On saturated-proportional derivative feedback with adaptive gravity compensation of robot manipulators". Int. J. of Adaptive Control and Signal Processing. John Wiley & Sons, Ltd. 10(4): 1996



Diseño de un controlador digital de la velocidad de un motor de corriente directa basado en microcontrolador DSPic30f4012.

Ing. Bárbaro Rodríguez Cárdenas

Ing. Arlet Montiel Ramos

Ing. Frank E. Aties Rojas

Centro de Investigación y Desarrollo
Naval

Centro de Investigación y Desarrollo
Naval

Centro de Investigación y Desarrollo
Naval

La Habana, Cuba

La Habana, Cuba

La Habana, Cuba

aid171alt@gmail.com

montielarlet94@gmail.com

frankaties0312@gmail.com

Resumen—Este artículo presenta el diseño y la implementación de un controlador digital para la velocidad de un motor de corriente continua, utilizando el microcontrolador DSPic30F4012. Se aborda la necesidad de un control preciso, donde la variabilidad en la carga puede afectar el rendimiento del motor. Se describe el procedimiento del diseño del algoritmo de control PID, que permite ajustar dinámicamente la señal de control en función de la medición de los sensores de velocidad. Se muestran las técnicas de simulación computacional y las pruebas experimentales para evaluar la respuesta temporal del sistema de control respecto a los requerimientos establecidos. En este trabajo se refleja el procedimiento del grupo de investigación para el desarrollo de aplicaciones microcontroladas.

Palabras clave: Controlador digital, motor de corriente directa, DSPic30F4012, algoritmo de control PID, sFunction de Matlab.

I. INTRODUCCIÓN

El control de la velocidad de motores de corriente directa (CD) es un área crítica en la ingeniería automática, con aplicaciones que abarcan desde la robótica hasta la automatización industrial. La necesidad de sistemas de control precisos y eficientes ha llevado a la investigación en el diseño de controladores digitales, especialmente aquellos basados en microcontroladores como el DSPic30f4012 [1],[2],[3]. Este artículo aborda el diseño e implementación de un controlador

digital de velocidad para motores de CD, utilizando un algoritmo PID y simulaciones computacionales en Matlab.

El modelado matemático de los motores de CD es una herramienta esencial en el diseño de control. Un modelo capaz de describir el comportamiento dinámico garantiza una gran eficacia del sistema de control. Por ejemplo, [1] y [2] han propuesto modelos que integran las ecuaciones eléctricas y mecánicas, aportando un marco robusto para el análisis del sistema a pesar de las limitantes que presentan. Estos modelos consideran factores como la inductancia, resistencia y constante de tiempo del motor, lo cual es esencial para el desarrollo de un controlador efectivo [1], [4], [5].

En cuanto al diseño del controlador digital, el algoritmo PID ha sido ampliamente utilizado debido a su simplicidad y eficacia. Investigaciones recientes, como las de [3] y [4], han explorado diversas técnicas para la sintonización de los parámetros del PID, destacando la importancia de una sintonización adecuada para lograr un rendimiento óptimo en sistemas no lineales. Además, el uso de métodos como el Ziegler-Nichols y algoritmos genéticos ha sido propuesto como una forma de mejorar la respuesta del sistema ante perturbaciones.

La simulación computacional se ha convertido en una herramienta esencial para validar el desempeño del controlador antes de

su implementación física. Matlab, junto con su capacidad para trabajar con sFunctions, permite modelar sistemas complejos y realizar simulaciones detalladas. Estudios como los de [5] y [6] han demostrado cómo estas técnicas pueden ser empleadas para evaluar el rendimiento del controlador bajo diversas condiciones operativas, facilitando así el proceso de diseño.

Finalmente, este artículo se estructurará en cinco secciones: primero, se presentará la introducción; segundo, se abordará el modelado matemático de la dinámica del movimiento del motor CD; tercero, se discutirá el diseño del controlador digital; cuarto, se describirá la implementación del control basado en el microcontrolador DSPic30f4012; y quinto, se presentarán los resultados experimentales obtenidos.

II. MODELADO DINÁMICO DEL MOVIMIENTO DEL MOTOR CD

A. Definiciones, consideraciones y suposiciones.

La técnica de la modelación matemática que describe el comportamiento dinámico del sistema es una herramienta muy empleada en el desarrollo de los sistemas de control. Para el análisis del modelo se deben tener en cuenta aspectos fundamentales como: definiciones, consideraciones y suposiciones que garanticen la descripción de la dinámica del sistema con la menor complejidad posible [6],[7],[8].

La representación simplificada del motor de corriente directa se ilustra en el siguiente diagrama:

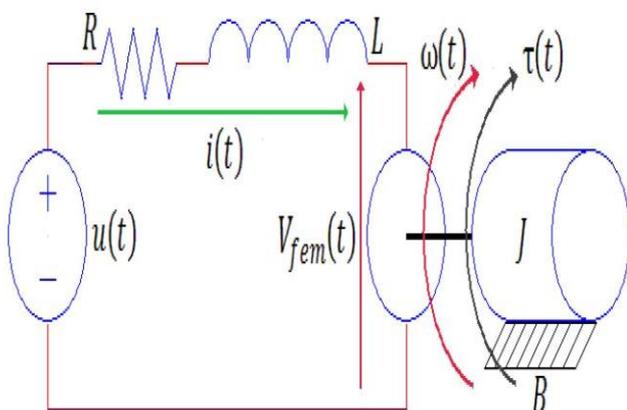


Figura 1. Esquema electro-mecánico simplificado de un motor de corriente directa.

Son estas cuatro expresiones las más significativas para el modelado de un motor de corriente directa:

$$u(t) = R \cdot i(t) + L \cdot di(t)/dt + V_{fem}(t). \quad (1)$$

$$\tau(t) = B \cdot \omega(t) + J \cdot d\omega(t)/dt + T_i(t). \quad (2)$$

$$\tau(t) = K_T \cdot i(t). \quad (3)$$

$$V_{fem}(t) = K_E \cdot \omega(t). \quad (4)$$

Dónde:

R: Resistencia óhmica de los embobinados del rotor.

L: Inductancia de los embobinados del rotor.

J: Momento de inercia del rotor.

B: Coeficiente de fricción viscosa entre el rotor y el estator.

u(t): Fuente de excitación del sistema.

$\omega(t)$: Velocidad angular del rotor.

$\tau(t)$: Par del rotor.

i(t): Corriente del sistema.

$V_{fem}(t)$: Tensión eléctrica inducida.

K_T : Constante mecánica, considerando un flujo magnético constante (También puede ser nombrada como "Constante de campo").

K_E : Constante eléctrica, considerando un flujo magnético constante (También puede ser nombrada como "Constante de armadura").

Son estas cuatro expresiones las más significativas para el modelado de un motor de corriente directa. Los parámetros del motor se obtienen utilizando técnicas experimentales propuestas, en su gran mayoría, las mencionadas en (Modelado de un motor de corriente directa).

B. Simulación computacional.

Se procede a crear el diagrama de bloques correspondiente en la herramienta Simulink dentro del entorno de Matlab.

Se representa el diagrama de bloques correspondiente a las ecuaciones eléctrica y mecánica representada en las expresiones 1 y 2 considerando también las expresiones 3 y 4 que vinculan la parte eléctrica con la mecánica en su función transferencial en el dominio de Laplace.

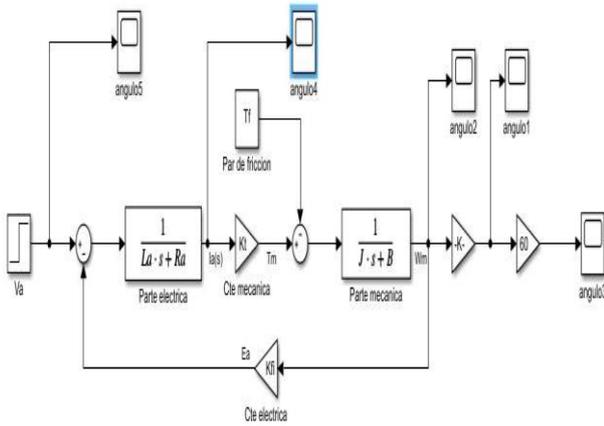


Figura 2. Diagrama en bloques de la dinámica del motor de corriente directa.

Se realiza la simulación y se comparan los resultados con algunos valores puntuales referentes a la tabla 2 considerando como señal de entrada la tensión de armadura. Se compara la velocidad angular para 5, 9 y 12V de tensión.

Tabla 1. Comparación de los resultados.

u(t) (V)	$\omega(t)$ (rad/s) real	$\omega(t)$ (rad/s) modelo	Error (%)
5	167.7610	165.2	1.53
9	328.6106	330.5	0.57
12	450.0017	454.5	1

III. DISEÑO DEL CONTROLADOR DIGITAL

Se describe el procedimiento para obtener el controlador digital de algoritmo PID [9] a partir del sistema modelado representado en su función de transferencia en el dominio de Laplace en el capítulo anterior [10].

Se establecen las especificaciones deseadas en el rendimiento del sistema de control [11], [12], [13]:

- Tiempo de establecimiento (t_e) < 0.5 segundos
- Sobreimpulso (M_p) < 10%
- Error en estado estable ($|e|$) < 0.5 rad/s

Se calculan los parámetros: proporcional (K_p) e integral (K_i). Para la sintonización de los parámetros se emplea el método de ajuste basado en función transferencial del bloque PID del Simulink acorde para modelo de sistemas lineales. Se obtienen los parámetros que garantizan los requerimientos establecidos expresados en el dominio de Laplace. Ver Figura 3.

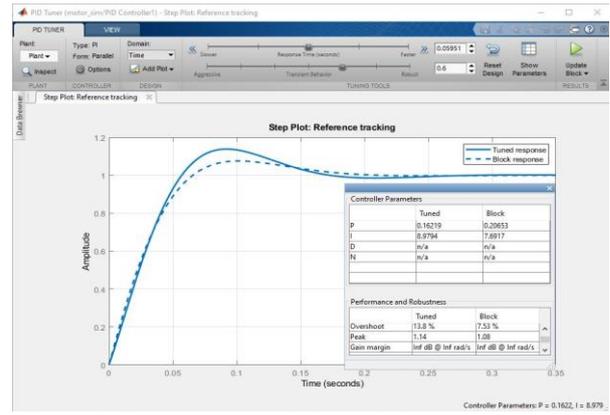


Figura 3. Ventana para el ajuste de los parámetros del controlador PID.

Se aplica el método de Tustin para discretizar con un período de muestreo de un milisegundo para obtener su representación en el dominio en Z. Como parte del procedimiento de nuestro grupo de investigación se emplean las sFunction Builder de Simulink. Ver Figura 4 donde se simulan los códigos a implementarse en el microcontrolador y así reducir los errores que se ejecutan por transcripción de las ecuaciones en diferencia finita.

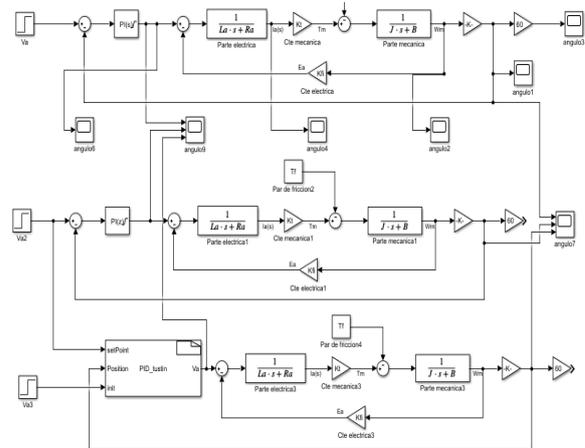


Figura 4. Representación del sistema de control en el dominio de Laplace, en el dominio en Z y en la sFunction Builder.

Se muestran los resultados de la simulación (Figura 5) y se evalúa el comportamiento del sistema de control (Tabla 2), además, de realizar comparación para los tres esquemas mostrados.

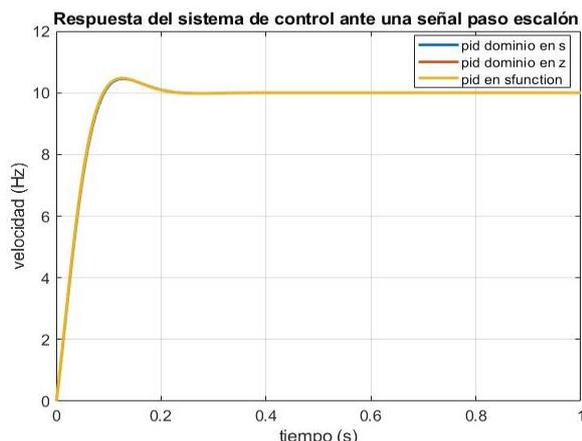


Figura 5. Comportamiento del sistema en simulación y comparación entre los tres esquemas.

Tabla 2. Resultados del comportamiento temporal de la respuesta.

Parámetros	Deseados	Reales
Tiempo de establecimiento	<0.5 seg	0.168 seg
Sobreimpulso	<10%	7.53%
Error en estado estable	<0.5 Hz	±0.02Hz

Una vez diseñado el sistema de control de velocidad, es necesario implementarlo sobre un microcontrolador para su uso práctico. Esto implica desarrollar el firmware que ejecute el algoritmo de control, así como la interfaz con los sensores y actuadores del motor.

IV. IMPLEMENTACIÓN DEL CONTROL BASADO EN EL MICROCONTROLADOR DSPIC30F412.

A. Criterios de selección.

La selección del microcontrolador dsPIC30F4012 se basa en diferentes criterios fundamentales [14]. También se elige el entorno de desarrollo adecuado que para esta familia de controladores y se trabaja con el IDE MPLAB versión v8.50.

El dsPic30F4012 se destaca por su alta velocidad de procesamiento de hasta 30MIPS (Millones de Instrucciones por Segundo). Con una arquitectura Harvard modificada de señal digital de 16 bits, lo que es adecuado para aplicaciones que requieren un proceso rápido de datos y cálculos intensivos [15].

Con una memoria flash integrado de hasta 24 KB, 2 KB de RAM y 1 KB de EEPROM no volátil garantiza espacio suficiente para implementar algoritmos complejos y manejar datos [15].

Dispone de 20 puertos de entrada/salida lo que permite la conexión de múltiples periféricos, facilitando la integración en sistemas más complejos. Incluye módulos de Modulación por Ancho de Pulso (PWM) de hasta 6 canales, conversores A/D de 10 bits y temporizadores lo que es esencial para el control de los motores [15].

Es compatible con voltajes de alimentación de 2.5V a 5.5V y puede operar en un amplio rango de temperaturas, desde -40°C hasta 125°C, garantizando su funcionamiento confiable en entornos adversos. Presenta disponibilidad de documentación técnica, facilidades de adquisición y ejemplos de implementación fundamental para hacer del dsPIC30F4012 el microcontrolador adecuado como unidad de procesamiento y cálculo para el sistema [15].

B. Algoritmo General.

Se incluyen todas las librerías que se emplean para el adecuado funcionamiento del controlador como son: "math.h" para las operaciones matemáticas, "INCAP.h" para la configuración del módulo de captura, "TIMER.h" para la configuración de los temporizadores, "pwm.h" para la configuración de PWM y "uart.h" para la comunicación con otros periféricos.

Se definen las macros a través de la ventana de herramienta "Configuration Bits" del MPLAB como se muestra en la Figura 6 ya que esta muestra cómo se desea el estado de los pines cuando están en desuso.

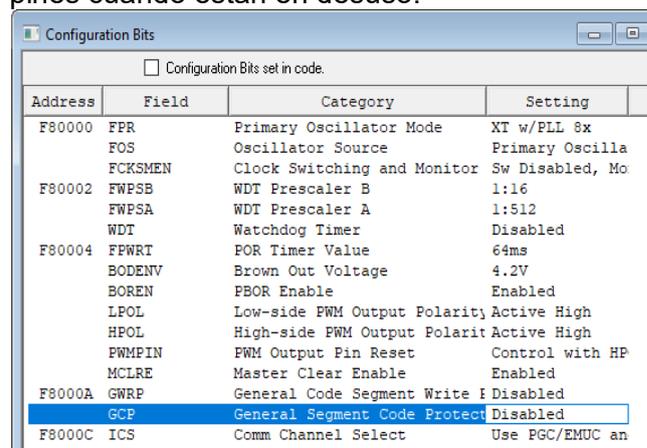


Figura 6. Ventana de Configuration Bits para la configuración de las macros.

En el programa principal se configuran los puertos de entrada y salida; el PWM a través de una función propia que tiene como

parámetros de entrada la FCY, frecuencia del PWM deseada, ciclo útil y canal a emplear (estos pueden ser 1 y/o 2 y/o 3); se configuran los temporizadores, se inicializa el controlador PID, el módulo de captura, la comunicación UART y las interrupciones con sus diferentes órdenes de prioridad todo través de las funciones definidas en las librerías del compilador C30. Luego de realizar todas las configuraciones anteriormente mencionadas el programa realiza la instrucción "while" y espera hasta que alguna interrupción configurada se active.

C. Atención a la subrutina de interrupción.

La interrupción del módulo de captura con una prioridad de seis, cuenta la frecuencia de la señal de entrada por el pin asignado a esta función. En la atención a la subrutina se procesa este dato multiplicado por el TCY y el preescalador para llevarlo a segundos y se almacena. Esta función se emplea para obtener la frecuencia de la señal de realimentación de los motores entregada por los sensores de efecto Hall. Se debe tener en cuenta que para un buen funcionamiento se debe deshabilitar la función de ADC de estos pines ya que están configurador así por defecto [15].

La función Gate del temporizador (el uno en este caso) proporciona una interrupción en cada frente de subida de la señal y cuenta el tiempo hasta el próximo flanco de caída es decir se obtiene el ancho del pulso de la señal [15].

La atención a la subrutina de interrupción del temporizador 4 en este caso se realiza estrictamente cada 1 ms que es el tiempo de muestreo exigido por el control PID en las simulaciones realizadas con una prioridad de seis.

En la misma se carga el valor del ancho del pulso en segundos, se establecen los límites y mapea a frecuencia desde los 29 Hz (mínima velocidad establecida) hasta los 82 Hz (máxima velocidad) para determinar la velocidad de los motores deseada en Hz. Se comprueba si existe realimentación, de no ser así en cuales quiera de los motores, no se realiza el control de velocidad, se carga cero en el buffer a transmitir por UART y solo se actualiza en valor del ciclo útil del PWM. Para cuando exista realimentación se carga el buffer a transmitir con el valor de la realimentación y se crea una

función que como parámetros de entrada tiene la velocidad deseada y la velocidad de ambos motores y actualiza el ciclo útil necesario para cada motor que garantice una velocidad de giro similar. De esta manera en la Figura 7 se realiza el control de la velocidad de los motores implementado en el código de microcontrolador.

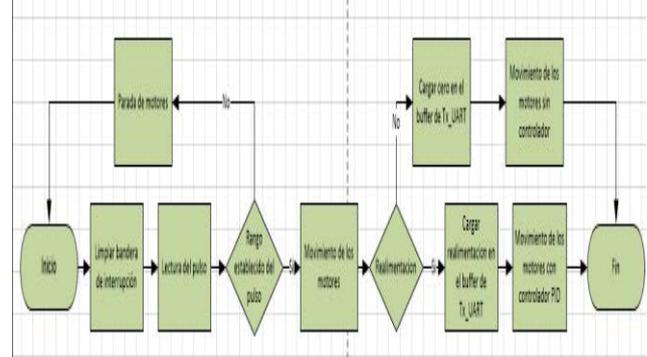


Figura 7. Diagrama en bloque del proceso general.

V. RESULTADOS EXPERIMENTALES

Se procede con las pruebas a nivel de laboratorio donde se integran los elementos que conforman el sistema de dirección y control del vehículo naval. En las Figuras 8 y 9 se muestran las tarjetas de control y de potencia que se encargan de procesar la señal proveniente del pixhawk, proporcional a la velocidad deseada en los motores.

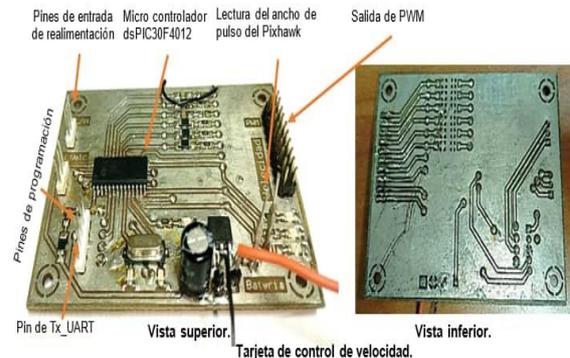


Figura 8. Tarjeta de control.



Figura 9. Tarjeta de potencia que manipula el motor.

Se realiza el diseño, fabricación y montaje de las tarjetas, como parte de la investigación, dentro de la institución. En la Figura 10 se muestra como queda conformado el sistema integrado de dirección y control.



Figura 10. Autopiloto.

Se observa (Figura 11) el medio de prueba para evaluar el sistema de control de los motores. Con el mando Futaba se varía la velocidad deseada de los motores y se transmite la señal en ppm al autopiloto proporcional a la velocidad. Se procesa la información en la tarjeta de control y manipula la tarjeta de potencia. La velocidad de ambos motores se mide a través de los sensores de efecto hall cerrando el lazo de control. Se registran las señales de ambos sensores en el osciloscopio y se evalúa el comportamiento del sistema de control.



Figura 11. Integración del sistema en pruebas de laboratorio.

Se muestra uno de los resultados obtenidos en la ejecución de las pruebas para un estímulo simulando una señal paso escalón de amplitud proporcional a una velocidad de 31Hz y se observa en el osciloscopio la señal de salida de los sensores.

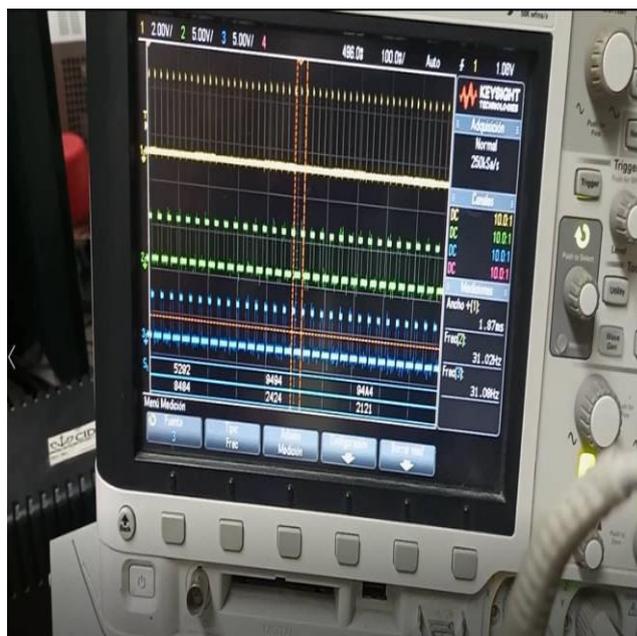


Figura 12. Resultados experimentales del sistema de control.

CONCLUSIONES

Mediante el modelado matemático se obtuvo una representación del comportamiento dinámico del motor de corriente directa. La parametrización se realiza mediante métodos experimentales.

El algoritmo de control PID demostró ser una solución efectiva para el control de la velocidad del motor, logrando un balance adecuado entre la rapidez de respuesta y la estabilidad del sistema.

Se validaron los procedimientos para el diseño e implementación del controlador digital empleando las simulaciones computacionales. Se evaluó el comportamiento del controlador mediante el empleo de las sFunction de Matlab garantizando así un desarrollo más eficiente y seguro antes de la implementación física.

La implementación del controlador digital en el microcontrolador DSPic30f4012 evidenció su capacidad para manejar tareas de control cumpliendo con el periodo de muestreo, proporcionando un rendimiento óptimo y una respuesta rápida ante variaciones en la carga del motor.

Las pruebas experimentales en laboratorio confirmaron que el controlador PID cumple con los requisitos de diseño establecidos, al cumplir un seguimiento de la señal de referencia y una reducción significativa del error en estado estacionario.

REFERENCIAS

- [1] S. Katke and S. Rangdal, "Speed control of DC motor using microcontroller," *International Journal of Scientific Research in Science and Technology*, vol. 1, no. 2, 2015.
- [2] M. Khine, M. Htwe, and M. Y. Y. Mon, "Simulation DC motor speed control system by using PID controller," *Int. J. Trend Sci. Res. Develop.*, vol. 3, no. 4, pp. 1418-1423, 2019.
- [3] C. Alvarez, A. Soto, and F. Watkins, "Simulación de controladores digitales," *Ingeniare. Revista chilena de ingeniería*, vol. 17, no. 3, pp. 309-316, 2009.
- [4] J. L. Inca Rodríguez, "Diseño de un Sistema de Control de Posición de un motor DC usando Procesador Digital de Señales," 2002.
- [5] J. Fonseca-Beltrán, G. Cazarez-Ayala, and F. K. L. Dablantes, "NUEVAS TECNOLOGÍAS EN LOS CONTROLADORES DIGITALES DE SEÑALES, ESTUDIO Y APLICACIONES NEW TECHNOLOGY IN DIGITAL SIGNAL CONTROLLERS, STUDY & APPLICATIONS," *Ra Ximhai*, vol. 8, no. 2, pp. 249-256, 2012.
- [6] S. Neira Rosales, M. A. Gutiérrez Zamarripa, and N. González Morales, "Cálculo de características de motores de CD mediante instrumentación virtual," *Ingenierías*, vol. 10, no. 35, pp. 20-25, 2007.
- [7] C. G. Hernández Capacho and M. G. Quijano Ortega, "Obtención experimental de los parámetros del motor

- [8] A. C. Aráoz, "Cómo obtener los parámetros de un motor de corriente continua e imán permanente," ed: May, 2009.
- [9] Z. Zhou, "Modern PID Controller Parameter Design for DC Motor," 2015.
- [10] V. Chesaru, C. Dan, and M. Bodea, "Pid Algorithm for controller processors," in *2006 International Semiconductor Conference*, 2006, vol. 2, pp. 429-432: IEEE.
- [11] A. V. Rao, "Mechanical Vibrations: Lecture Notes for Course Eml 4220," URL: <http://vdol.mae.ufl.edu/CourseNotes/EML4220/vibrations.pdf>, 2009.
- [12] R. C. D. R. H. Bishop, *Modern control systems*. 2011.
- [13] R. Isermann, *Digital control systems*. Springer Science & Business Media, 2013.
- [14] J. A. C. Osorio, J. A. M. Vargas, and J. A. M. Escobar, "Criterios de selección de las herramientas de desarrollo para microcontroladores pic," *Scientia et Technica*, vol. 3, no. 43, pp. 187-192, 2009.
- [15] Microchip, *Data Sheet dsPic30F4012* 2008, p. 238.

Interfaz háptica para el control remoto de robots móviles

Mateo N. Medina Morales, Pressello Lucas, Roberto M. Murdocca, Sergio F. Hernandez Velazquez

Departamento de Electrónica

Universidad Nacional de San Luis

Ciudad de San Luis-San Luis-Argentina

email: {teo.nmedina,lucaspressellounsl,mmurdocc,sergio.sfhv}@gmail.com

Resumen— La presente investigación desarrolla un prototipo de vehículo eléctrico teleoperado mediante un guante háptico. Este dispositivo permite el control remoto del vehículo a través de gestos manuales, proporcionando una interfaz intuitiva. Adicionalmente, el vehículo se encuentra equipado con una cámara que transmite video en tiempo real a un dispositivo móvil, permitiendo al operador una visualización directa del entorno (ver Fig.1).

Palabras clave: vehículo, inalámbrico, Wi-Fi, teleoperación, ESP32.

I. INTRODUCCIÓN

Con el fin de contribuir al avance de la robótica móvil, se ha diseñado y construido un vehículo teleoperado. Este sistema robótico, basado en una arquitectura modular y flexible, ofrece un marco de trabajo ideal para la investigación y desarrollo de algoritmos de navegación,

percepción y manipulación.

II. DESCRIPCIÓN

El vehículo se controla a distancia con un guante el cual es capaz de detectar movimientos de la mano y a partir de estos movilizarlo. La comunicación entre guante-vehículo se lleva a cabo vía Wi-Fi aprovechando las ventajas de conectividad que provee el microcontrolador ESP32.

Este sistema de control a su vez puede ser reemplazado por un smartphone utilizando el mismo tipo de comunicación, pero aplicando el protocolo Telnet.

También se cuenta con una cámara integrada que permite la transmisión instantánea de imágenes desde el vehículo hacia un teléfono inteligente, aprovechando el mismo medio de comunicación que el control háptico.

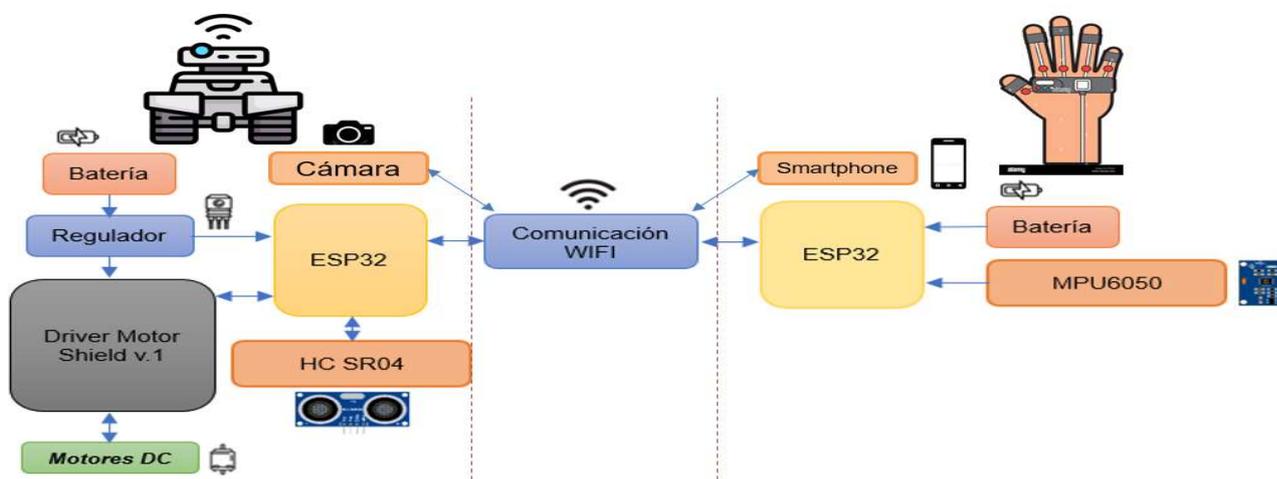


Fig.1 Diagrama en bloques del sistema.

III. CONTROL HÁPTICO

A. HARDWARE

El hardware principal del guante se detalla a continuación:

- ESP32
- Módulo *acelerómetro y giroscopio MPU-6050*
- *Batería*

El control se implementa con un guante detector de movimientos el cual capta los movimientos de la mano en tres ejes (x,y,z), esto lo realiza mediante el sensor acelerómetro y giroscopio. El microcontrolador será el encargado de leer los valores de aceleración y velocidad angular en los 3 ejes medidos por el sensor y a partir de ellos calcular los ángulos de inclinación en dichos ejes.

La ESP32 se conecta mediante Wi-Fi como cliente UDP al punto de acceso del vehículo para de esta manera según el movimiento que se realice con la mano movilizarlo. Este control es alimentado por una batería de 12 V (ver Fig.2).

Además, se cuenta con tres pulsadores donde dos de ellos se utilizan para controlar el movimiento del motor paso a paso que mueve la cámara en el vehículo y el restante le indica al auto si debe moverse o permanecer en reposo. Este último se agrega debido a que la mano solo tiene cuatro grados de libertad de movimiento y para el control del vehículo se requieren cinco (adelante, atrás, izquierda, derecha y reposo).

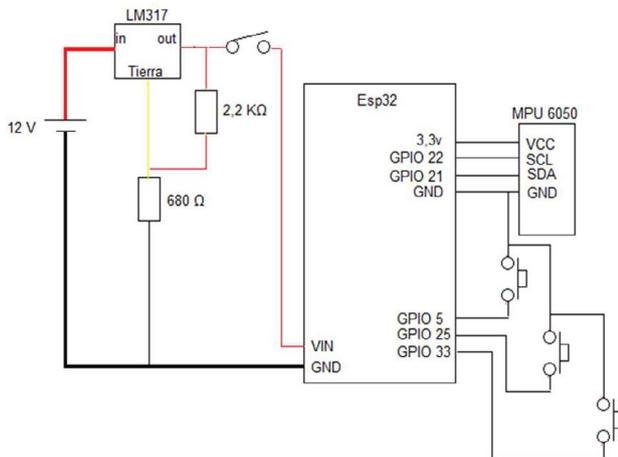


Fig. 2. Esquema eléctrico de Control Háptico.

B. FIRMWARE

El firmware para el funcionamiento del control háptico se desarrolla a partir de una MEF (máquina de estado finito), teniendo como principales características (Fig.3):

Tarea 1: realiza la lectura del módulo MPU-6050. Luego se procede a expresar los valores sensado en ángulos para que el control háptico detecte los ángulos de inclinación de la mano y a partir de estos movimientos indicar para dónde se debe desplazar el vehículo. A su vez, en esta tarea se realiza la lectura de tres pulsadores.

Tarea 2: se realiza el procesamiento de los datos leídos en la tarea anterior, por lo que se le asigna a una variable la cual va a ser enviada vía Wi-Fi al vehículo un valor correspondiente según la lectura de los movimientos hápticos realizados por el usuario.

Tarea 3: aquí se envía mediante Wi-Fi utilizando un protocolo UDP la variable con el valor asignado anteriormente al vehículo.

IV. VEHÍCULO TELEOPERADO

A. Hardware

El hardware principal del vehículo se detalla a continuación:

- ESP32
- Módulo Driver Motor Shield Arduino
- Sensor Ultrasónico HC-SR04
- Batería
- Forzador y disipador de calor
- Motores DC con caja reductora
- Motor paso a paso generico
- Cámara con conectividad WI-FI

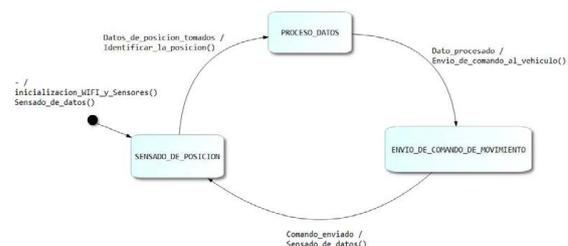


Fig. 3. MEF control háptico.

El vehículo funciona controlando 4 motores de corriente continua (DC) mediante el Motor Driver Shield de Arduino, a su vez dicho shield se adaptó creando una interfaz para poder ser controlado por una ESP32.

El Motor Driver Shield de Arduino funciona con 2 circuitos integrados L293D (cada uno de estos son los encargados de hacer que dos motores funcionen, se detengan y cambien de sentido) y un registro de desplazamiento 74HC595 (éste configura la dirección de todos los motores usando solo 3 cables de datos en lugar de 8). La ESP32 es la encargada de poner en funcionamiento el shield, enviando 4 señales PWM (Modulación de Ancho de Pulso) las cuales establecen las velocidades de los motores, y 3 señales de control al registro de desplazamiento (señal de reloj, sentido de giro, habilitación de salida). Además, la ESP32 funciona como un punto de acceso UDP (Protocolo de Datagramas de Usuario), método por el cual recibirán los datos para el movimiento del vehículo.

El vehículo cuenta también con una cámara la cual es capaz de grabar y transmitir en tiempo real. La señal de vídeo es transmitida a un celular haciendo uso de la comunicación Wi-Fi. También, se le integra un sensor de ultrasonido con el objetivo de crear un sistema anti-impactos.

Por último, para garantizar el control inalámbrico y la autonomía tanto del vehículo como del guante, estos poseen baterías independientes recargables. El auto es alimentado por una batería 10.8 V / 4400 mAh, mediante dos reguladores de tensión se derivan las tensiones de 5 V utilizada por sensores, cámara y microcontrolador, por otra parte, se utiliza otro regulador de 9 V el cual es el encargado de alimentar el shield y el forzador perteneciente al disipador de calor (Fig.4).

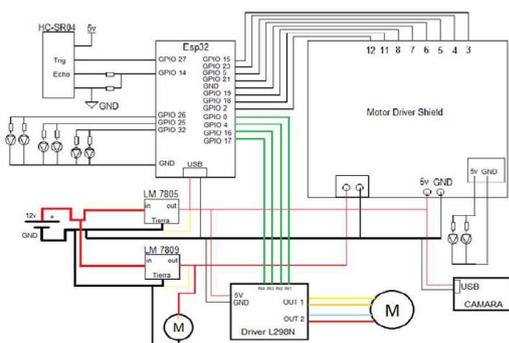


Fig. 4. Esquema eléctrico Vehículo Teleoperado.

B. FIRMWARE

La realización del firmware del vehículo teleoperado también se realizó mediante una MEF de dos tareas (Fig.5):

MEF tarea 1: esta tarea es la encargada de leer el dato recibido mediante Wi-Fi y ejecutar o hacer llamado a la función que ejecuta el movimiento tanto del vehículo como del motor que rota la cámara.

MEF tarea 2: en esta tarea se desarrolla el sistema anti-impacto del vehículo, por lo que se realiza lecturas del sensor ultrasónico el cual detecta la distancia de posibles colisiones, en caso de que la distancia leída sea menor a 30cm aproximadamente, se asigna a la variable de movimiento la orden de detenerse/girar.

También se cuenta con 11 funciones independientes, que se detallan a continuación:

Función dataProcess: en esta función se lee el dato recibido mediante Wi-Fi (enviado por el control háptico) y se procesa este mismo asignando a una variable el código correspondiente al movimiento ordenado.

Función moveDataProcess: encargada de ejecutar la función de movimiento correspondiente.

Función movement: controla el registro de desplazamiento del driver motor shield.

Función setSpeed: gestiona las velocidades de cada uno de los motores DC.

Funciones de movimiento: aquí se hace llamado a la función movement a la cual se le pasa como parámetro un arreglo el cual contiene los valores correspondientes que ponen en funcionamiento los motores DC (Fig.6). Luego aprovechando las funcionalidades provistas por la librería AccelStepper.h se hace llamado a funciones pertenecientes a esta para movilizar el motor paso a paso encargado de darle movimiento a la cámara.

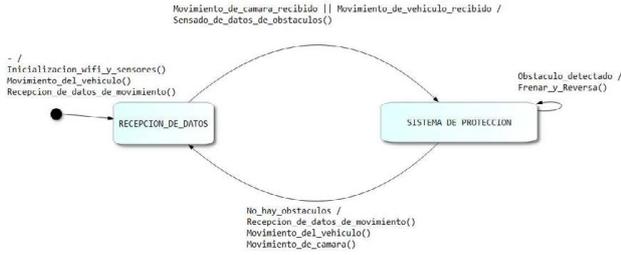


Fig. 5. MEF vehículo teleoperado.

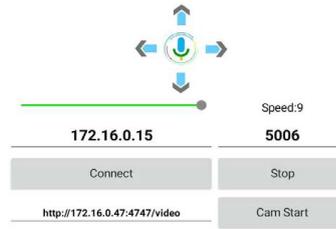


Fig. 7. App Wifi Robot.

74HC595 Out	Bits	number	motor	direction
QC	2	4	M1	clockwise
QD	3	8	M1	counterclockwise
QB	1	2	M2	clockwise
QE	4	16	M2	counterclockwise
QF	5	32	M3	clockwise
QH	7	128	M3	counterclockwise
QA	0	1	M4	clockwise
QG	6	64	M4	counterclockwise

Fig. 6. Matriz de movimiento de motores.

V. COMUNICACIÓN

La comunicación vehículo-control se realiza haciendo uso de una comunicación UDP, de esta forma se garantiza la velocidad de respuesta sin delay dado que en este protocolo no se realiza un control de los datagramas que llegan al receptor.

Esto se implementa utilizando dos ESP32, una de estos funciona como punto de acceso (AP) y el restante funciona como cliente, esto permite realizar la comunicación y envío de datos entre ambas. A nivel firmware esto se implementa haciendo uso de la librería WiFiUdp.h, WiFi.h.

El vehículo también puede ser controlado mediante el smartphone utilizando el protocolo Telnet (Telecommunication Network), el cual permite controlar de forma remota otro dispositivo mediante una conexión TCP/IP (Transmission Control Protocol / Internet Protocol). Esta conexión se realiza mediante un punto de acceso creado en el celular y utilizando al vehículo como cliente. El movimiento del vehículo se realiza mediante cualquier app de Telnet en el celular (Fig.7).

Para realizar el envío de video en tiempo real, la cámara crea un punto de acceso el cual se comunica con el cliente (smartphone) mediante Wi-Fi utilizando el protocolo TCP/IP. La aplicación utilizada para la visualización del vídeo es Sports XDV (Fig.8).

VI. RESULTADOS

Se obtuvo como resultado final un prototipo el cual puede ser tomado como base para distintas áreas del desarrollo de la robótica tanto de exploración como de búsqueda, siendo así de gran utilidad en la resolución de problemas de estas características (Fig.9 y Fig.10).

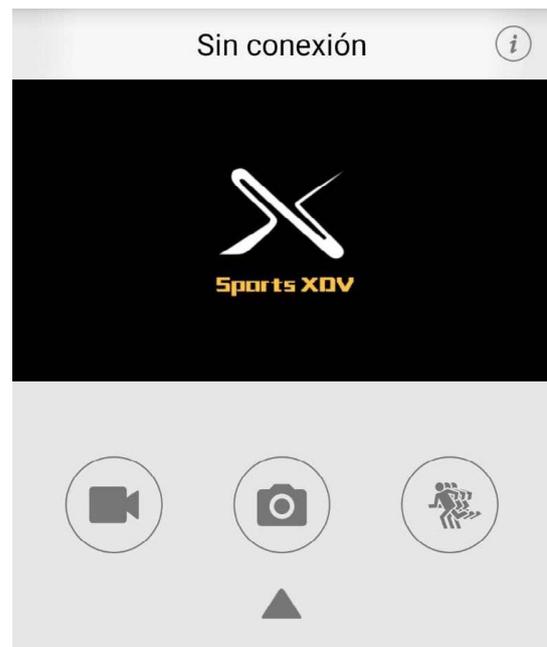


Fig. 8. App Sports XDV.



Fig. 9. Prototipo final guante de control háptico.

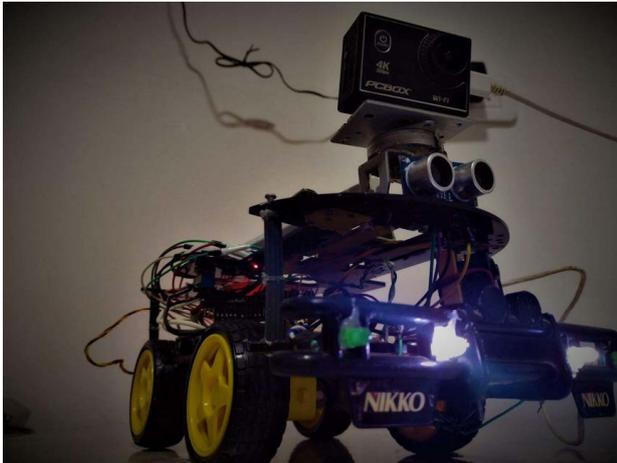
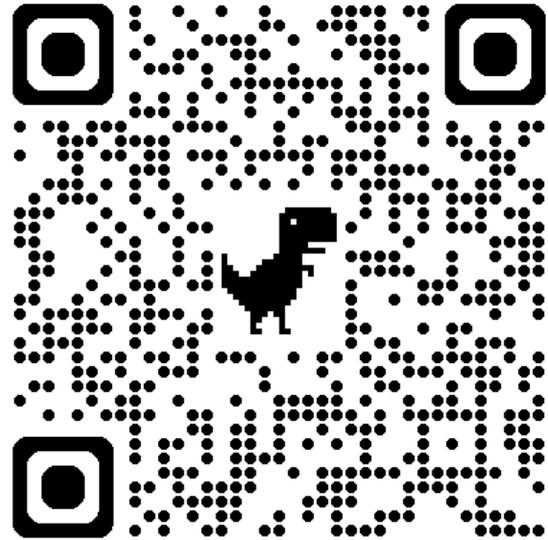


Fig. 10. Prototipo final de vehículo teleoperado.

En el siguiente código QR se puede visualizar el progreso del desarrollo de este prototipo

desde una versión inicial hasta la actual siendo evidente el progreso alcanzado al día de hoy.



VIII.REFERENCIAS

- [1] Espressif Systems, *ESP32 Series Datasheet*, Espressif Systems, 2023.
- [2] N. Medina Morales and L. A. Pressello, *Vehículo eléctrico con control inalámbrico*, 2023.
- [3] M. Gul, *wifi-nodemcu-robot-esp8266* [Repositorio GitHub], 16 de marzo de 2023 - <https://github.com/mahirgul/wifi-nodemcu-robot-esp8266>
- [4] SunFounder, *Motor Driver Shield*, 16 de marzo de 2023 - http://wiki.sunfounder.cc/index.php?title=L293D_Motor_Driver_Shield
- [5] Arduino Spain, *Arduino Motor Shield*, 16 de marzo de 2023 - <https://sp.arduino-france.site/shield-l293d/>
- [6] J. Lipovský, [Blog post], 16 de marzo de 2023 - <http://blog.janlipovsky.cz/2016/03/?m=1>
- [7] Leantec, *Datasheet L293N* - <https://leantec.es/wp-content/uploads/2019/05/LEANTEC-Documentacion-L298N-Rojio.pdf>



Diseño e Implementación de una banda transportadora automática con el microcontrolador PIC18f45k50 y una interfaz de control local y remota

línea 1: 1st Jeisson Avila
línea 2: Universidad ECCI
línea 3: Universidad ECCI
línea 4 Bogotá D.C, Colombia
línea 5: <https://orcid.org/0009-0004-4688-3803>

línea 1: 2nd Juan Pablo Blanco
línea 2: Universidad ECCI
línea 3: Universidad ECCI
línea 4 Bogotá D.C, Colombia
línea 5: <https://orcid.org/0009-0009-0245-4325>

línea 1: 3rd Robin Blanco
línea 2: Universidad ECCI
línea 3: Universidad ECCI
línea 4 Bogotá D.C, Colombia
línea 5: <https://orcid.org/0000-0002-6105-3689>

Resumen—Se ha diseñado una banda transportadora automatizada controlada por un microcontrolador PIC18F45K50, la cual incorpora un sistema de distribución automatizada de piezas mediante un servomotor. El sistema también integra elementos de visualización como un LCD de caracteres y un display de 7 segmentos, que proporcionan información en tiempo real sobre el conteo de piezas. Este conteo se lleva a cabo a través de un sensor de ultrasonido estratégicamente ubicado en la parte superior de la banda transportadora.

La velocidad de la banda, la parada de emergencia y la distribución de piezas se gestionan a través de una interfaz local y remota, utilizando tecnología Bluetooth para la comunicación inalámbrica. Adicionalmente, un teclado matricial permite al usuario ingresar la cantidad de piezas que se desea contar. Una vez alcanzada la cantidad predeterminada, se activa una señal sonora como indicativo.

Este artículo se centra en la facilidad de replicación del modelo de banda transportadora, proporcionando todos los planos mecánicos y eléctricos necesarios para su construcción.

Palabras clave:

1. Banda transportadora.
2. Automatización.
3. PIC18f45K50.
4. Tecnología.

5. Diseñar

I. INTRODUCCIÓN

El proyecto propuesto titulado "Diseño e Implementación de una banda transportadora automática con el microcontrolador PIC18f45k50 y una interfaz de control local y remota", tiene como objetivo desarrollar una herramienta didáctica avanzada. Este proyecto busca integrar tecnologías de control con microcontroladores, programación, sensórica, automatización e instrumentación para crear una banda transportadora automatizada. La automatización de estas bandas con el uso de microcontroladores, sensores y actuadores puede mejorar significativamente su rendimiento y fiabilidad. El enfoque principal es el diseño y construcción de una banda transportadora que funcione como una plataforma educativa práctica, la cual será controlada por un microcontrolador PIC18f45K50 y equipada con un sensor de ultrasonido para el conteo preciso de las piezas. Se incluirá una tarjeta electrónica que integrará todos los elementos de control e instrumentación necesarios. Además, se incorporarán componentes como un módulo Bluetooth, sensores ultrasónicos, una interfaz de usuario visual con pantalla LCD caracteres y teclado matricial para una interacción intuitiva y eficiente. El proyecto también se centra en la optimización del rendimiento de la banda, asegurando un funcionamiento eficiente y continuo.



Se busca desarrollar una implementación práctica y fácilmente replicable para estudiantes que están aprendiendo sobre microcontroladores, en la que se utilicen tanto sensores como actuadores de manera sencilla, pero con aplicaciones que tengan un enfoque real en entornos industriales. El objetivo de este proyecto es proporcionar una plataforma educativa accesible y didáctica que permita a los estudiantes comprender los fundamentos de la electrónica y la programación, mientras trabajan con tecnologías y componentes que son ampliamente utilizados en la industria moderna.

II. MARCO TEÓRICO

1. Microcontrolador PIC18F45K50

Este Microcontrolador desarrollado por Microchip cuenta con algunas características que lo hacen ideal para este proyecto. Cuenta con un oscilador interno que puede variar desde los 31KHz hasta los 16 MHz el cual le permite realizar operaciones con mayor rapidez y precisión, también cuenta con 36 pines I/O (Entradas o salidas) es decir que entrega o recibe hasta 36 señales una más que el tradicional PIC18F4550.

1.1 Herramientas de programación

MPLAB X IDE:

Es el entorno de desarrollo integrado oficial de Microchip para sus microcontroladores PIC XC8

Compiler: Es el compilador de Microchip para la familia de microcontroladores PIC de 8 bits

Mplab code configurator: Herramienta gráfica que permite configurar periféricos y generar código automáticamente

1.2 Lenguajes de programación c:

Lenguaje de programación más utilizado para programar microcontroladores PIC

Ensamblador: Lenguaje de bajo nivel específico para el hardware del microcontrolador.

2. Fundamentos del Modo Captura El microcontrolador

PIC18F45K50 cuenta con módulos CCP (Capture/Compare/PWM), que permiten trabajar en tres modos: captura, comparación y PWM (modulación por ancho de pulso). En el modo captura, el valor actual del contador de un temporizador (Timer) se guarda en un registro específico cuando ocurre un evento en uno de los pines de captura designados, como CCP1 o CCP2.

2.1 Funcionamiento del Modo Captura El modo

captura se basa en la combinación de un Timer y un módulo CCP. El Timer es un contador que puede ser configurado para contar en diferentes velocidades, mientras que el módulo CCP captura el valor del Timer cuando detecta un evento específico en el pin de captura.

2.2 Registros Involucrados en el Modo Captura

- **CCPxCON** (Capture/Compare/PWM Control Register): Este registro controla el funcionamiento del módulo CCP. Los bits CCPxM configuran el módulo en el modo captura, y los bits asociados definen si la captura se realizará en el borde ascendente, descendente o en ambos.
- **CCPRxH** y **CCPRxL**: Estos registros almacenan el valor capturado del Timer cuando se detecta el evento. Se divide en dos partes, **CCPRxH** (parte alta) y **CCPRxL** (parte baja) para poder almacenar valores de 16 bits.

3. Sensor Ultrasonido

Un sensor de ultrasonido funciona mediante la emisión y recepción de ondas sonoras de alta frecuencia,

inaudibles para el oído humano. Estos sensores son utilizados en diversas aplicaciones, como la medición de distancias

3.1 Principios de funcionamiento

3.1.1 Emisión de ondas ultrasónicas

El sensor ultrasónico consta de un transductor que emite ondas sonoras de alta frecuencia (usualmente entre 20 kHz y 40 kHz) en un breve pulso.

3.1.2 *Reflexión del pulso* Estas ondas viajan a través del aire hasta que encuentran un objeto en su camino, momento en el cual se reflejan y regresan al sensor.

3.1.3 *Recepción del ECO* El mismo transductor, o en algunos casos un transductor separado, recibe las ondas reflejadas (eco) que retornan al sensor.

3.1.4 *Cálculo del tiempo de vuelo* El sensor mide el tiempo transcurrido desde que se emite el pulso hasta que se recibe el eco. Este tiempo se conoce como tiempo de vuelo (ToF, por sus siglas en inglés).

3.1.5 *Cálculo de la distancia* Conociendo la velocidad del sonido en el aire (aproximadamente 343 metros por segundo a 20°C), el sensor utiliza la fórmula:

3.2 Uso Industrial Del sensor

Los sensores de ultrasonido tienen una amplia variedad de aplicaciones en la industria debido a su precisión y fiabilidad, en este caso se utilizarán para la detección de objetos.

4. Bandas Transportadoras

4.1. Principios de funcionamiento

4.1.1. Componentes Básicos

- Cinta transportadora.
- Rodillos o poleas.
- Motor.
- Estructura de soporte.

4.1.2. Movimiento de la cinta

El motor está conectado a un engranaje el cual girará al momento que se accione el motor.

4.1.3. Descarga

Los materiales se descargan en el punto final de la cinta, donde pueden ser recogidos, empaquetados o enviados a otra etapa del proceso.

4.2. Ventajas

- Eficiencia: Permiten el transporte continuo y eficiente de grandes volúmenes de materiales.
- Versatilidad: Pueden ser diseñadas para manejar una amplia variedad de materiales y tamaños.
- Seguridad: Reducen el riesgo de lesiones al minimizar la manipulación manual de materiales pesados.

III. METODOLOGÍA

La metodología empleada en el desarrollo de este proyecto se fundamenta en diversas estructuras previamente establecidas. En particular, la banda transportadora se controla a través de un microcontrolador Microchip PIC18f45K50, programado mediante el entorno de desarrollo integrado (IDE) MPLAB X, especializado en la programación de este tipo de microcontroladores. El funcionamiento de la banda se visualiza en (Figura 1).

El sistema de la banda transportadora incluye un único motor y un potenciómetro para regular su velocidad. Además, cuenta con un módulo Bluetooth que posibilita el envío de comandos desde dispositivos móviles y la recepción de los valores medidos por el potenciómetro. Un sensor ultrasónico se encarga de detectar y

contar las piezas que transitan por la banda, operando en un rango de 7 a 9 cm.

Para la interfaz de usuario, se incorpora un visualizador LCD caracteres que proporciona mensajes de bienvenida, indica la cantidad de piezas contadas y muestra la cantidad total esperada. Un LED RGB cambia de color al alcanzar decenas específicas en el conteo, mientras que las pantallas LCD caracteres y 7 segmentos muestran las unidades actuales contadas.

Además, se integra un teclado matricial que permite al usuario ingresar el número de piezas a contar y realizar acciones como reiniciar el conteo, activar el paro de emergencia, iniciar el conteo y borrar entradas erróneas. Los cuales se pueden evidenciar en (Figura 2).

En cuanto a la estructura mecánica (Figura 3), la banda transportadora se impulsa mediante un motor de corriente continua (motor reductor) conectado directamente a ella. La banda está construida con una estructura de acrílico y la correa es de tela, cuenta con un mecanismo controlado por un servomotor MG90S (Figura 4) el cual permite una distribución controlada de piezas sobre la banda.

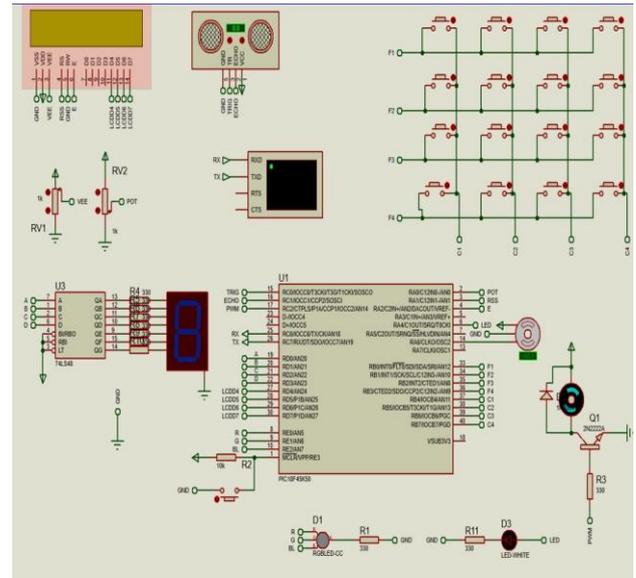


Figura 2. Mecanismo de funcionamiento.

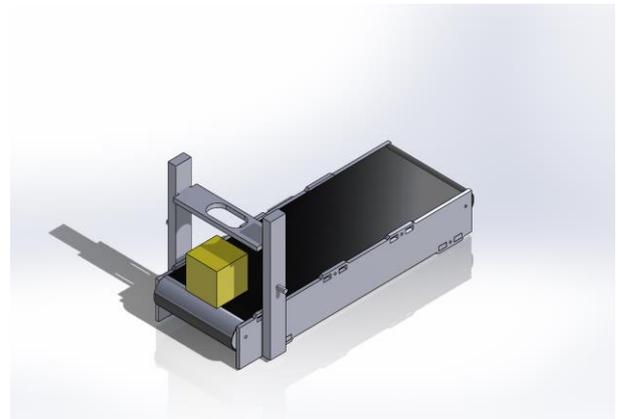


Figura 3. Estructura mecánica.

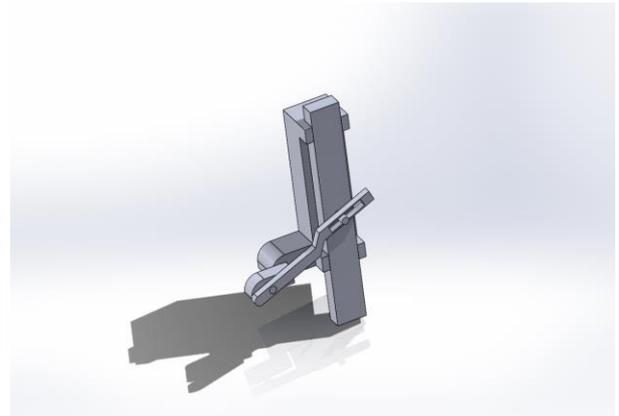


Figura 4. Mecanismo de empuje.

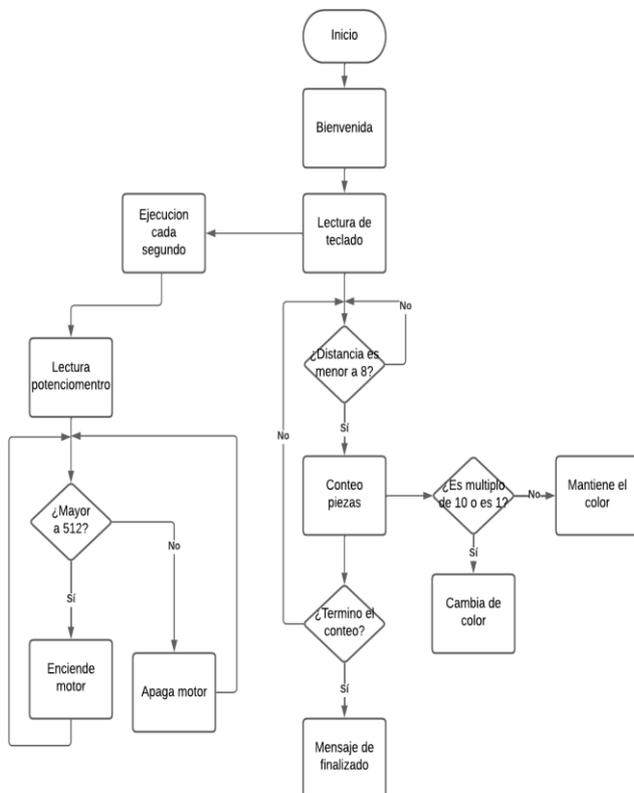


Figura 1. Funcionamiento de la banda.

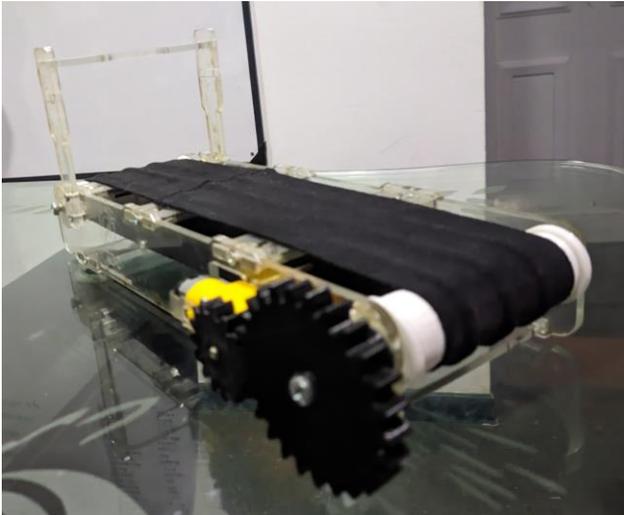
IV. RESULTADOS

La interfaz local, mediante la pantalla LCD, mostró un excelente desempeño en la visualización del conteo de piezas en tiempo

real. Durante las pruebas, el sistema fue capaz de actualizar el número de piezas sin retraso perceptible, proporcionando información instantánea al operador. El ingreso de la cantidad objetivo de piezas se realizó sin problemas a través del teclado matricial, y la pantalla reflejó de forma precisa tanto el número de piezas como el estado del sistema, mostrando un mensaje de finalización al alcanzar el número deseado.

Por otro lado, la interfaz remota, implementada mediante un módulo Bluetooth, permitió la conexión con dispositivos móviles (Android) de manera rápida y estable. Las pruebas demostraron que el emparejamiento se realizó en un promedio de 10 segundos, y la comunicación se mantuvo constante en un rango de hasta 7 metros sin interferencias físicas.

La interfaz remota demostró ser altamente funcional y estable, permitiendo un control y monitoreo flexible, incluso en situaciones donde la distancia entre el operador y la banda transportadora era significativa.



RECONOCIMIENTOS

Los autores desean expresar su sincero agradecimiento al Profesor Robin Alfonso Blanco Cañon su valiosa orientación y apoyo durante el desarrollo de este proyecto. Su experiencia y consejos fueron fundamentales para el éxito de este trabajo.

REFERENCIAS

- [1] R. J. Moreno, J. E. Martínez Baquero, y C. A. Romero Molano, Introducción a la programación de microcontroladores PIC, 1st ed. Villavicencio, Colombia: Editorial Universidad de los Llanos, 2022. [En línea]. Disponible en: <https://editorial.unillanos.edu.co/index.php/editorial-unillanos/catalog/book/3>. [Accedido: ago. 15, 2024].
- [2] R. Jiménez M., Programación de microcontroladores PIC: Lenguaje de alto nivel orientado a gama alta. Bogotá, Colombia: ASEUC. [En línea]. Disponible en: <https://unilibros.co/gpd-programacion-de-microcontroladores-pic-lenguaje-de-alto-nivel-orientado-a-gama-alta.html>. [Accedido: ago. 17, 2024].
- [3] Microchip Technology Inc., PIC16F83/84A Data Sheet: 18-Pin Enhanced Flash/EEPROM 8-Bit Microcontrollers, 2nd ed., Microchip Technology Inc., Chandler, AZ, USA, 2001. [En línea]. Disponible en: <https://ww1.microchip.com/downloads/en/devicedoc/30000684b.pdf>. [Accedido: ago 20, 2024].
- [4] R. A. Blanco C., "Micros 4AM 2023-II," Google Classroom, 2023. [En línea]. Disponible en: <https://classroom.google.com/u/1/c/NTg4OTkwNjI4NTEy>. [Accedido: ago 27, 2024].

Diseño y fabricación de un Robot Delta

Elio Ruben Ogas
Laboratorio de Mecatrónica
FICA - UNSL

Villa Mercedes, San Luis, Argentina
ogaselio@gmail.com

Gabriel Iglesias
Laboratorio de Mecatrónica
FICA - UNSL

Villa Mercedes, San Luis, Argentina
gabrieliglesias22@gmail.com

Oscar Daniel Moran
Laboratorio de Mecatrónica
FICA - UNSL

Villa Mercedes, San Luis, Argentina
odanielmoran@gmail.com

Resumen— Este artículo presenta el diseño y la fabricación de un robot Delta educativo en el Laboratorio de Mecatrónica (LABME) de la Facultad de Ingeniería y Ciencias Agropecuarias (FICA-UNSL). El proyecto se enfoca en el desarrollo de un robot económico utilizando dispositivos y materiales disponibles en Argentina; destinado a aplicaciones educativas en ingeniería. El diseño se desarrolló en el laboratorio y para la fabricación se empleó una impresora 3D y una cortadora láser, para las partes estructurales. El robot Delta tiene tres articulaciones motorizadas con motores paso a paso NEMA 17 y sistemas de transmisión con poleas y correas. Se desarrollaron herramientas intercambiables para el efector final: un electroimán y una pinza. El sistema de control incluye una placa Arduino Mega con una shield Ramp 1.4 y drivers A4988, controlado mediante comandos G. La interfaz gráfica en Visual .NET y C# facilita la interacción y control del robot, validando su utilidad en entornos educativos.

Palabras claves— robot, delta, cinemática paralela.

I. INTRODUCCIÓN

El desarrollo de robots Delta, caracterizados por su estructura paralela y tres brazos conectados a una plataforma móvil, ha revolucionado múltiples aplicaciones industriales debido a su alta velocidad y precisión. Estos robots, inventados por Raymond Clavel en los años 80, son especialmente valorados por su capacidad de ejecutar movimientos rápidos y precisos en un espacio tridimensional, manteniendo una rigidez estructural destacable [1].

En un contexto educativo, en el Laboratorio de Mecatrónica (LABME) de la Facultad de

Ingeniería y Ciencias Agropecuarias de la UNSL se emprendió el desafío de diseñar y fabricar un robot Delta accesible y económico. Utilizando materiales y componentes disponibles en Argentina. Este proyecto busca proporcionar una herramienta versátil para la enseñanza y experimentación en carreras de ingeniería.

El presente artículo describe detalladamente el proceso de diseño y fabricación del robot Delta, incluyendo la selección de materiales, la metodología empleada y los resultados obtenidos en su implementación y pruebas, destacando su potencial como recurso educativo.

II. DESARROLLO

A. Descripción y geometría básica

El robot desarrollado está conformado por una estructura fija unida a una plataforma móvil mediante tres cadenas cinemáticas paralelas, Fig. 1. Cada cadena tiene como actuadores motores paso a paso y cuatro articulaciones tipo rótulas, dispuestas para formar un paralelogramo. Esta disposición le permite a la plataforma móvil tener tres grados de libertad de traslación en la dirección de los ejes del sistema de coordenadas de referencia.

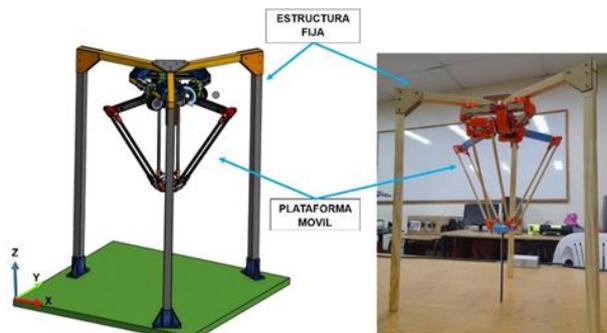


Fig. 1. Ensamblaje en software y robot fabricado.

B. Diseño del robot

El diseño del robot Delta se realizó utilizando el software de diseño CAD, Onshape. Este

software permitió crear un modelo detallado y preciso del robot, facilitando su posterior fabricación. El dispositivo cuenta con tres articulaciones motorizadas, cada una comandada por un brazo dispuesto a 120 grados entre ellos, Fig.2.



Fig. 2. Modelo CAD en software de diseño.

C. Fabricación

La fabricación del robot se llevó a cabo en el LABME utilizando impresoras 3D para crear componentes en PLA y ABS. Las partes estructurales y algunos soportes, fueron cortados en madera mediante una cortadora láser, Fig. 3.



Fig. 3. Robot Delta, componentes impresos en 3D y fabricados con cortadora láser.

D. Componentes Mecánicos

Motores y Sistema de transmisión: Cada articulación del robot es impulsada por un motor paso a paso NEMA 17 conectado a un sistema de transmisión por polea y correa dentada. Utiliza dos poleas dentadas (relación 2:1) y una correa cerrada de 150 mm tipo GT2 6 mm. Los tensores de las correas consisten en pivotes con

ruedas de 16 mm de diámetro exterior, ajustables mediante tornillos para mejorar la tensión de las mismas, Fig. 4.

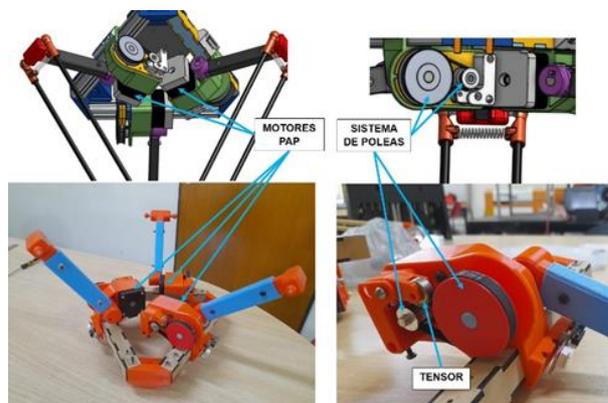


Fig. 4. Motores y sistema de poleas y correa dentada.

Soporte: El soporte que contiene los motores de cada articulación fue diseñado y fabricado con partes impresas en 3D y con una cortadora láser en madera MDF. Dicho soporte tiene una forma triangular, debido a la disposición de 120 grados de los brazos requerido por el diseño de estos robots, Fig. 5.

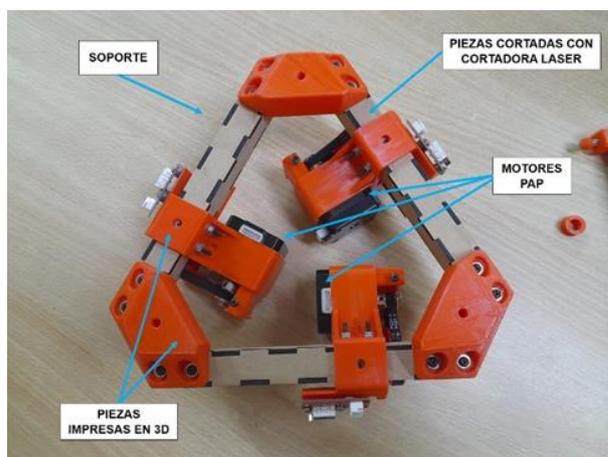


Fig. 5. Soporte que contiene los motores y sistema de poleas

Brazos y Uniones: El eje de la polea transmitida, permite la tracción para realizar el movimiento circular de cada articulación del robot, Fig. 6. Este eje está conectado a un brazo de 150 mm de largo, de sección rectangular 20x40mm. En el extremo de cada brazo se encuentran articulaciones del tipo rótulas, unidas con ejes de 3 mm y esferas de 5mm de diámetro impresas en 3D en material ABS. Las uniones permiten el movimiento de dos varillas de sección circular de 8 mm de diámetro y 300 mm de largo, conectadas a la plataforma del efector final.

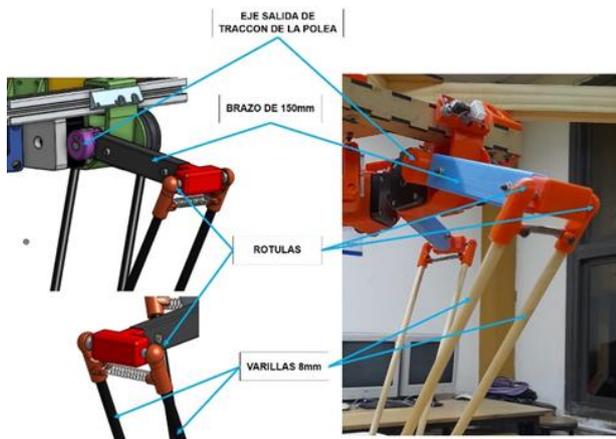


Fig. 6. Brazos, uniones y rótulas de la plataforma móvil.

Efactor Final y Herramientas: La plataforma del efector final fue diseñada e impresa en 3D, incluyendo un acople rápido central para herramientas, Fig. 7. Se desarrollaron dos tipos de herramientas: una con un electroimán y otra con una pinza de dos dedos. Ambas herramientas incluyen una articulación rotacional, accionada por un servomotor, que permite la orientación del extremo. Para el accionamiento de los dedos de la pinza, también se utiliza un servomotor.

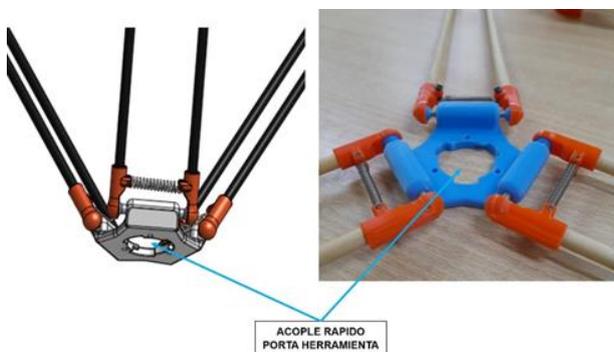


Fig. 7. Efactor final y acople central porta herramienta

E. Sistema de control.

El sistema de control del robot Delta se implementa mediante una placa Arduino Mega, Fig. 8, la cual contiene el firmware de control y las ecuaciones de la cinemática correspondiente, que calcula los ángulos de movimiento y genera las señales necesarias para mover los motores de las articulaciones. Además, dicho firmware controla el accionamiento de los servos de las herramientas y de la articulación de orientación del extremo [2]. Sobre la placa Arduino se encuentra montada una placa shield Ramp 1.4 la cual permite el conexionado de los drivers de los

motores paso a paso A4988 con el Arduino. Para detectar la posición inicial (Home) de las articulaciones, se utilizan finales de carrera mecánicos que entran en contacto con los brazos rectangulares durante la rutina de búsqueda de posición inicial.

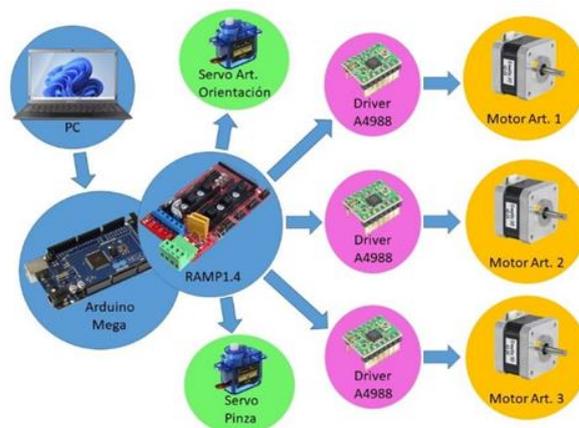


Fig. 8. Diagrama de bloques de dispositivos de control

F. Firmware del Arduino

El firmware contenido en el Arduino comanda los motores de acuerdo a las peticiones recibidas desde una computadora (PC). La comunicación con la PC, donde se ejecuta la interfaz gráfica de usuario, se logra mediante un puerto serie por hardware del Arduino. El firmware contiene un módulo que recibe e interpreta las peticiones de movimiento y configuración del robot, implementadas mediante código G. Este código es un lenguaje de programación utilizado en máquinas CNC para controlar el movimiento y las operaciones de estas máquinas. Se compone de comandos que indican movimientos específicos y configuraciones.

El módulo de entrada del firmware recibe las peticiones mediante una cadena de texto que contiene el código G y sus parámetros. Realiza la separación de las partes de la cadena, identifica el comando y determina si los parámetros son válidos. Los comandos se dividen en dos grupos: los de movimiento y los de control/configuración.

Comandos de Movimiento: Incluyen los comandos de movimiento lineal (G0-G1), movimiento por arco (G2-G3), movimiento angular de cada articulación (G6) y movimiento a posición inicial "Home" (G28).



Fig. 9. Diagrama de bloques de comandos de movimientos

Tales comandos son los encargados de realizar movimientos en el robot de determinadas maneras. Para ello se utilizan una serie de interpoladores para generar trayectorias suaves y controladas para que el robot se mueva de manera precisa y eficiente en un espacio tridimensional. En tales comandos se procede de la siguiente manera para realizar los movimientos, Fig.9.

El firmware contenido en el Arduino comanda los motores de acuerdo a las peticiones recibidas desde una computadora (PC). La comunicación con la PC, donde se ejecuta la interfaz

- Definición del objetivo:** Se define la posición o punto final al que se desea que el robot llegue en coordenadas cartesianas en el espacio tridimensional (x, y, z).
- Cálculo de la cinemática inversa:** A partir de la posición objetivo definida, se utilizan ecuaciones y algoritmos de cinemática inversa, propios de los robots Delta, para determinar los ángulos de articulación necesarios en las articulaciones del robot para alcanzar esa posición. Esto implica resolver las ecuaciones que relacionan las coordenadas espaciales con los ángulos de las articulaciones. Esto permite corroborar que los ángulos obtenidos son posibles de realizar para continuar con la interpolación.
- Interpolación de trayectoria:** Una vez que se han calculado los ángulos de articulación para el punto objetivo, se divide la trayectoria entre la posición actual del robot y la posición objetivo en

segmentos más pequeños. Para cada uno de estos segmentos, se calculan los ángulos de articulación correspondientes mediante la interpolación de los valores de ángulo entre la posición actual y la posición objetivo.

- Ejecución del movimiento:** Con los ángulos de articulación calculados para cada segmento de la trayectoria, el robot o sistema mecánico ejecuta los movimientos necesarios para desplazarse de manera suave y controlada desde su posición actual hasta la posición objetivo.

De acuerdo al comando se utilizan diferentes interpoladores, tanto lineales como circulares de la siguiente manera:

Comandos G0 - G1 (Interpolación lineal): Estos utilizan la interpolación lineal para mover el robot desde su posición actual hasta una posición objetivo en línea recta. Se calculan segmentos lineales de longitud menor al segmento entre la posición actual y la posición objetivo, y se planifican los movimientos del robot a lo largo de estos segmentos. Por ejemplo, los comandos pueden ser:

G1 X50 Y25 Z-250 F200

Mueve el extremo a la posición X=50mm, Y=25mm, Z=-250 con una velocidad de 200 mm/min. Si no se especifica la velocidad se mantiene la anteriormente especificada.

G1 X80 F150

Mueve el extremo a la posición X=80mm, manteniendo la coordenada de Y y Z constantes, con una velocidad de 150 mm/min.

G1 W90

Mueve en el extremo la articulación de orientación al ángulo especificado en parámetro W manteniendo constantes los valores de posición X, Y, Z.

Comandos G2 - G3 (Interpolación circular): Estos utilizan la interpolación circular para mover el robot en trayectorias curvas. Se definen arcos de círculo de longitud menor al arco entre la posición actual y la posición objetivo, y el robot se mueve a lo largo de estos arcos de manera suave y continua. Por ejemplo:

G2 X10 Y10 I5 J5 F150

Mueve el extremo en forma de arco a la posición final (X=10, Y= 10) con centro en la posición (X=5, Y=5) y con una velocidad de 150 mm/min. G2 realiza el movimiento en el sentido de las agujas del reloj y G3 en el sentido contrario.

En los interpoladores mencionados se deben resolver las ecuaciones de la cinemática inversa para cada punto final de los segmentos intermedios interpolados, así de esta manera obtener la trayectoria del segmento total que une el punto inicial con el objetivo. A continuación, Fig. 10, se pueden observar las ecuaciones implementadas en el código del firmware del Arduino, para obtener los ángulos de las articulaciones (Ang.Tita1, Ang.Tita2, Ang.Tita3) correspondientes a cada punto mencionado con coordenadas X,Y,Z. Siendo RD_RF la longitud del brazo y RD_RE la longitud de las varillas, Fig. 5.

```
bool CinematicaInversa(float X, float Y, float Z, Angle sAng){
  if (!CalculoAngulo(X, Y, Z, Ang.Tita1)) return false;
  if (!CalculoAngulo(X*cos120 + Y*sin120, Y*cos120 - X*sin120, Z, Ang.Tita2)) return false;
  if (!CalculoAngulo(X*cos120 - Y*sin120, Y*cos120 + X*sin120, Z, Ang.Tita3)) return false;
  return true;
}

bool CalculoAngulo(float x0, float y0, float z0, float sTita){
  y1 = -0.5 * tan30 * RD_RF;
  y0 = 0.5 * tan30 * RD_E;

  a = (x0*x0 + y0 * y0 + z0 * z0 + RD_RF * RD_RF - RD_RE * RD_RE - y1 * y1) / (2.0*z0);
  b = (y1 - y0) / z0;
  d = -(a + b * y1)*(a + b * y1) + b*b*RD_RF * RD_RF + RD_RF * RD_RF;
  if (d < 0) return false;

  yj = (y1 - a * b - sqrt(d)) / (b*b + 1.0);
  zj = a + b * yj;
  Tita = atan(-zj / (y1 - yj)) * RAD_TO_DEG + ((yj > y1) ? 180.0 : 0.0);
  return true;
}
```

Fig. 10. Código de implementación de la cinemática inversa.

Comandos de Control y Configuración:

Incluyen los comandos tales como pausa en ejecución (G4), control de las herramientas (M3-M4-M5), configuración de velocidad y aceleración máxima (M203-M204), selección de la herramienta actual (M360), y almacenamiento de parámetros (M500-M501).

Comando de pausa en ejecución (G4): Este comando permite detener el programa en código G un determinado tiempo especificado en sus parámetros en milisegundos. Por ejemplo:

G4 P2000

Realiza una pausa en la ejecución del programa especificada mediante el parámetro P en milisegundos.

Comandos de control de herramientas (M3-M4-M5): Mediante estos comandos se puede controlar el accionamiento de las herramientas. Si se utiliza el electroimán como herramienta instalada en plataforma móvil, mediante el comando M3 se activa dicho dispositivo atrayendo el objeto metálico a sujetar y mediante M4 se desactiva el dispositivo liberando el objeto. Si se utiliza la pinza, mediante el comando M3 se controla la apertura de la misma. Utilizando su parámetro asociado se especifica desde 0 a 100 el porcentaje de apertura de la pinza. El comando M5 permite la apertura completa de la pinza. Por ejemplo:

M3

Colocando solo el comando M3, sin ningún parámetro, se realiza la activación del electroimán.

M4

Colocando solo el comando M4, sin ningún parámetro, se realiza la desactivación del electroimán.

M3 S60

Utilizando el comando M3 acompañado del parámetro S podemos especificar el porcentaje de apertura de la pinza (en este caso 60 % de apertura, siendo 0% totalmente cerrada y 100% la posición de apertura máxima)

M5

Utilizando solo el comando M5, sin ningún parámetro especificado, se realiza la apertura completa de la pinza.

Comando de configuración de velocidad y aceleración máxima (M203-M204): Mediante los mismos se modifican los valores de velocidad y aceleración máxima admitidos por robot para llevar a cabo sus movimientos. Por ejemplo:

M203 S200

Se especifica una velocidad máxima mediante el parámetro S, en este caso de 200 mm/min

M204 S1200

Se especifica una aceleración del movimiento mediante el parámetro S, en este caso de 1200 mm/s².

Comando de selección de la herramienta actual (M360): Mediante este comando podemos especificar dentro de un listado

numérico la herramienta a utilizar en el efector final. De esta manera se deciden las restricciones y la lógica de funcionamiento acorde a la herramienta elegida.

G. Interfaz gráfica de usuario (GUI)

La interfaz gráfica de usuario (GUI) del robot Delta fue desarrollada utilizando Visual .NET y lenguaje C#. Se trata de una aplicación de escritorio para Windows que incluye todos los controles necesarios para interactuar y controlar el robot Delta de manera eficiente. La GUI se compone de varios sectores, cada uno diseñado para facilitar diferentes aspectos del control del robot, Fig. 11. A continuación, se describen detalladamente las funciones y características de cada sector de la interfaz.

Sector de Conexión: Este sector permite establecer la conexión serial con el Arduino, que actúa como controlador del robot. Los usuarios pueden seleccionar el puerto serial y la velocidad de comunicación (baud rate) adecuados para iniciar la conexión. La correcta conexión se indica mediante un mensaje de estado.

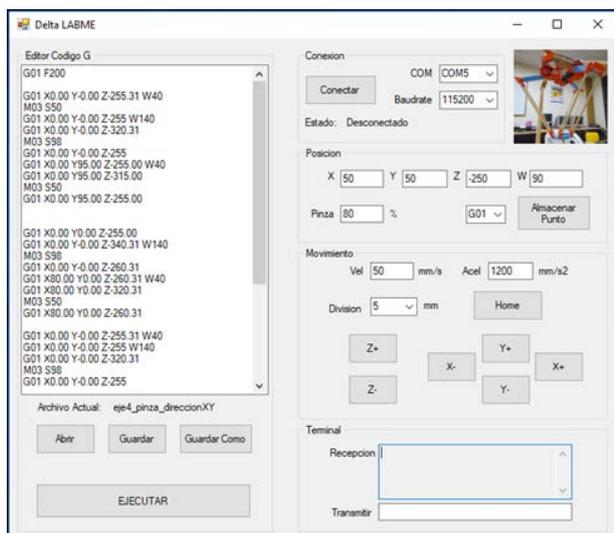


Fig. 11. Interfaz gráfica de usuario.

Sector de Posición Actual: En este sector se muestran las coordenadas espaciales actuales (X, Y, Z) del extremo del robot. Los usuarios tienen la posibilidad de modificar estos valores para mover el extremo del robot a una nueva posición deseada. Además, se puede ajustar la orientación angular del extremo (campo W) y el porcentaje de apertura de la pinza (campo Pinza), si la misma se encuentra instalada como herramienta en el robot. Una función destacada de este sector es la capacidad de almacenar la

posición actual y generar automáticamente el código G de movimiento lineal (G1) utilizando como parámetros los valores de dicha posición (botón "Almacenar Punto").

Sector de Movimiento: Este sector permite a los usuarios modificar la posición del extremo del robot en las direcciones de los ejes coordenados (X, Y, Z) de manera manual. Mediante botones de incremento y decremento, se pueden ajustar los valores de posición mencionados. Además, aquí se configuran la velocidad y la aceleración de los movimientos, así como el valor de incremento/decremento aplicado por los botones de control.

Sector de Terminal: En el sector de terminal, los usuarios pueden enviar comandos G o comandos de configuración reconocidos por el Arduino. Este sector también permite visualizar las respuestas recibidas desde el Arduino, facilitando la interacción y el monitoreo del estado del robot.

Sector de Programación en Código G: Este sector se dedica a la gestión y edición de los programas en código G. Los usuarios pueden cargar, editar, guardar y ejecutar programas en código G que controlan las trayectorias y operaciones del robot Delta. La interfaz facilita la creación de nuevos programas, así como la modificación de los existentes, permitiendo una mayor flexibilidad y control sobre el robot.

La interfaz gráfica de usuario desarrollada para el robot Delta en Visual .NET y C# proporciona una herramienta poderosa y versátil para el control y la programación del robot. Su diseño intuitivo y la integración de múltiples funciones permiten a los usuarios interactuar con el robot de manera eficiente, mejorando la usabilidad y el rendimiento en aplicaciones industriales y de investigación. Implementar una interfaz gráfica de usuario (GUI) para el control del robot ofrece numerosas ventajas, tanto para operadores como para desarrolladores, tales como:

1. Facilidad de Uso

- **Intuitiva y Amigable:** Una GUI facilita la interacción con el robot mediante una interfaz visual clara y organizada, lo cual reduce la curva de aprendizaje para los nuevos usuarios.
- **Accesibilidad:** Los controles visuales permiten que personas con menos



experiencia técnica puedan operar el robot de manera efectiva.

2. Eficiencia Operativa

- **Control Manual y Automático:** La GUI permite tanto el control manual preciso del robot como la ejecución de programas automatizados en código G, optimizando los tiempos de operación.
- **Gestión Simplificada:** Funciones como la generación automática de código G desde las posiciones actuales agilizan la programación y ajuste de las trayectorias del robot.

3. Monitoreo

- **Visualización de la posición:** La visualización en todo momento de las coordenadas actuales del robot y de otros parámetros importantes permite un monitoreo continuo y preciso.
- **Feedback Inmediato:** La terminal integrada que muestra la respuesta del Arduino proporciona un feedback inmediato sobre los comandos ejecutados, facilitando el diagnóstico y resolución de problemas.

4. Flexibilidad y Personalización

- **Configuración Personalizada:** La GUI permite configurar parámetros clave como velocidad, aceleración y posiciones de destino de manera sencilla y rápida.
- **Adaptabilidad:** Es fácil de adaptar la interfaz a diferentes necesidades y aplicaciones, ya sea mediante la adición de nuevas funcionalidades o la modificación de las existentes.

5. Mejora en la Seguridad

- **Control Preciso:** La capacidad de controlar manualmente cada movimiento del robot permite operar en entornos donde la precisión es crucial, reduciendo el riesgo de errores y accidentes.
- **Monitoreo Visual:** La GUI facilita el monitoreo visual del estado del robot, ayudando a prevenir situaciones peligrosas mediante una supervisión constante.

6. Productividad Aumentada

- **Automatización de Tareas:** La integración de comandos automatizados y la posibilidad de almacenar y reutilizar programas en código G aumenta la productividad al reducir el tiempo necesario para configurar y ejecutar tareas repetitivas.
- **Documentación y Registro:** La capacidad de almacenar puntos y generar código G automáticamente ayuda en la documentación y registro de operaciones, lo que es útil para análisis posteriores y mejoras continuas.

7. Compatibilidad y Escalabilidad

- **Compatibilidad con Hardware Existente:** Al utilizar Visual .NET y C#, la GUI es compatible con una amplia gama de dispositivos y puede integrarse fácilmente con otros sistemas de control.
- **Escalabilidad:** La arquitectura de la GUI permite futuras expansiones y actualizaciones sin necesidad de rediseñar la interfaz desde cero.

III. RESULTADOS

La construcción del robot Delta resultó en un dispositivo funcional y versátil, adecuado para aplicaciones educativas. El robot demostró una alta precisión en la ejecución de movimientos gracias a la combinación de motores paso a paso y sistemas de transmisión con poleas y correas. Las herramientas intercambiables permitieron realizar diversas tareas, demostrando la flexibilidad del diseño.

Las pruebas realizadas incluyeron la manipulación de objetos con el electroimán y la pinza, así como la ejecución de trayectorias programadas para cada una de las herramientas. Los resultados mostraron un desempeño consistente y fiable, validando el diseño y la fabricación del robot.

IV. CONCLUSIONES

El desarrollo y fabricación del robot Delta en el LABME - FICA - UNSL ha sido un proyecto exitoso que ofrece valiosas oportunidades educativas y de investigación. El robot, diseñado con software Onshape y fabricado utilizando tecnologías de impresión 3D y corte láser, demuestra la viabilidad de crear herramientas educativas avanzadas con recursos accesibles



y de bajo costo en Argentina. Incorporar un robot Delta didáctico en el entorno educativo tiene un impacto significativo en la formación de ingenieros. Los estudiantes pueden comprender de manera práctica los conceptos teóricos de la cinemática, dinámica y control en un sistema robótico real, facilitando una comprensión más profunda y retentiva de los conceptos aprendidos en clase.

El uso del robot en entornos educativos permitirá a los estudiantes y profesores explorar conceptos complejos de ingeniería de manera práctica y dinámica. La interacción con un robot Delta permite a los estudiantes desarrollar habilidades prácticas en diseño, fabricación, programación y control de robots, esenciales para su futura carrera profesional. Futuras mejoras podrían incluir la integración de sensores adicionales y el desarrollo de nuevas herramientas, ampliando aún más las capacidades del robot Delta. También se experimentará con la integración del robot Delta en una Celda de Fabricación Flexible que se está desarrollando en el citado laboratorio.

Trabajar con robots didácticos estimula la creatividad y la innovación, desafiando a los estudiantes a diseñar y construir soluciones para problemas reales utilizando tecnología robótica. De esta manera, el proyecto no solo contribuye al aprendizaje académico, sino que también prepara a los futuros ingenieros para enfrentar desafíos profesionales con habilidades prácticas y un enfoque innovador.

REFERENCIAS

- [1] Sarvesh Patil, Samuel C. Alvares, Pragna Mannam, Oliver Kroemer, F. Zeynep – “DeltaZ: An Accessible Compliant Delta Robot Manipulator for Research and Education” Arxiv - 2022
- [2] Ahmad Taher Azar; Quanmin Zhu; Alaa Khamis; Dongya Zhao - “Control design approaches for parallel robot manipulators: a review” International Journal of Modelling, Identification and Control (IJMIC), Vol. 28, No. 3, 2017.



Aplicación de Visión Artificial para Robótica Educativa: Reconocimiento y Manipulación de Objetos por Robots Antropomorfo y Delta.

Elio Ruben Ogas

Laboratorio de Mecatrónica

FICA - UNSL

Villa Mercedes, San Luis, Argentina

ogaselio@gmail.com

Gabriel Iglesias

Laboratorio de Mecatrónica

FICA - UNSL

Villa Mercedes, San Luis, Argentina

gabrieliglesias22@gmail.com

Funes Matias

Laboratorio de Mecatrónica

FICA - UNSL

Villa Mercedes, San Luis, Argentina

matiasfunes1991@gmail.com

Oscar Daniel Moran

Laboratorio de Mecatrónica

FICA - UNSL

Villa Mercedes, San Luis, Argentina

odanielmoran@gmail.com

Resumen—En este trabajo se presenta el desarrollo e implementación de una aplicación de visión artificial para robots manipuladores educativos, específicamente un robot antropomorfo y un robot Delta, desarrollados en el laboratorio de Mecatrónica de la Facultad de Ingeniería y Ciencias Agropecuarias de la Universidad Nacional de San Luis. La aplicación, que utiliza Machine Learning y la plataforma online "Teachable Machine", permite el reconocimiento y clasificación de objetos para tomar decisiones sobre los movimientos de los robots manipuladores. Este sistema se desarrolló en Python utilizando la biblioteca OpenCV, y su eficacia fue validada en un entorno educativo y de investigación.

Palabras claves—visión, Python, Machine Learning, robótica.

I. INTRODUCCIÓN

El avance acelerado de la inteligencia artificial (IA) y la visión artificial ha transformado radicalmente múltiples sectores industriales, destacándose especialmente en el ámbito de la robótica. Estas tecnologías no solo han optimizado procesos productivos, sino que también han abierto nuevas fronteras en la educación, el entrenamiento y la investigación. Este artículo se centra en una aplicación pionera que aprovecha estos avances para mejorar significativamente la interacción con robots manipuladores educativos.

En particular, se examina el desarrollo de dos robots clave en el laboratorio de Mecatrónica de la Universidad Nacional de San Luis, un robot antropomorfo y un robot Delta. Estos robots no solo representan la culminación de la ingeniería mecatrónica moderna, sino que también sirven como plataformas experimentales para explorar cómo la visión artificial puede potenciar sus capacidades educativas y funcionales.

El robot antropomorfo, con su capacidad para imitar acciones humanas precisas, y el robot Delta, reconocido por su rapidez y precisión en tareas repetitivas, son ejemplos emblemáticos de cómo la tecnología puede adaptarse al entorno educativo. En este contexto, la implementación de sistemas avanzados de visión artificial no solo permite a estos robots identificar y manipular objetos con alta precisión, sino que también facilita un aprendizaje interactivo y experimental para estudiantes y profesionales del campo.

Este trabajo explora cómo la combinación de Machine Learning [1], redes neuronales y herramientas como la plataforma "Teachable Machine" y la biblioteca OpenCV [2], están siendo utilizadas para entrenar modelos que permiten a los robots reconocer una variedad de objetos en tiempo real. Estos avances no solo mejoran la capacidad de los robots para tomar decisiones autónomas basadas en la percepción visual, sino que también establecen un puente crucial entre la teoría y la práctica en la formación de futuros ingenieros y científicos.

Así, este trabajo no solo resalta los logros técnicos alcanzados en el laboratorio de

Mecatrónica, sino que también subraya el potencial transformador de la inteligencia artificial y la visión artificial en la educación y la investigación en robótica.

II. DESARROLLO

A. Descripción del sistema de visión

Los dispositivos involucrados para el desarrollo del sistema son: una cámara web Logitech C922, una computadora, un robot antropomorfo de seis grados de libertad (CXN2) y un robot Delta, ambos desarrollados en el laboratorio con sus controladores asociados, Fig.1. La finalidad de este sistema es permitir que los robots manipuladores educativos, realicen tareas de manipulación de objetos mediante el reconocimiento de los mismos en el entorno de trabajo, utilizando visión artificial.

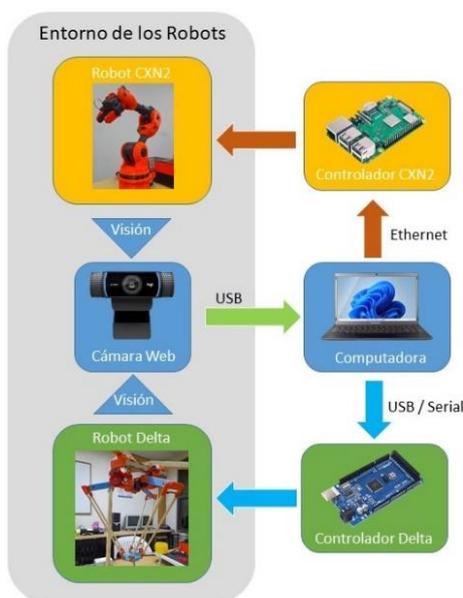


Fig. 1. Sistema de visión artificial.

La cámara web utilizada es la Logitech C922, que ofrece una alta calidad de imagen con resolución Full HD 1080p y una velocidad de fotogramas de hasta 60 fps. Esta cámara permite capturar imágenes o video del entorno de trabajo de los robots y transmitirlos al software de visión artificial implementado en la computadora. La computadora contiene el software desarrollado en este trabajo utilizando el lenguaje Python y se comunica con los controladores de los robots para realizar acciones sobre los mismos. El controlador del robot antropomorfo se conecta a través de una conexión Ethernet utilizando el protocolo TCP/IP, lo que asegura una comunicación rápida y confiable, permitiendo que el software

de visión artificial envíe comandos precisos al robot. Simultáneamente, la computadora se conecta al controlador del robot Delta (Arduino) mediante una conexión USB. Esta conexión serial permite que la computadora envíe comandos directamente al Arduino, que controla el robot Delta.

B. Plataforma de aprendizaje automático

La implementación de la aplicación se basa en Machine Learning, una subdisciplina de la inteligencia artificial (IA) que permite a los sistemas aprender y mejorar automáticamente a partir de la experiencia sin ser explícitamente programados. En este trabajo, se utiliza el concepto para entrenar modelos que pueden reconocer y clasificar objetos en imágenes.

En este contexto, el aprendizaje automático se aplica para el reconocimiento de objetos mediante la plataforma "Teachable Machine". Dicha plataforma online permite a los usuarios entrenar modelos de IA utilizando datos ejemplos proporcionados a través de una cámara web o desde imágenes almacenadas. La plataforma utiliza redes neuronales profundas, que son modelos de aprendizaje automático inspirados en el funcionamiento del cerebro humano. Las redes neuronales son modelos computacionales que consisten en capas de nodos interconectados, donde cada nodo representa una neurona artificial que procesa una parte de la información y transmite los resultados a la siguiente capa. Estas redes procesan la información y permiten la clasificación de los objetos mediante imágenes. El proceso inicia con la recopilación de los datos de entrenamiento los cuales se proporcionan a la red neuronal para llevar a cabo el aprendizaje supervisado, Fig. 2. Una vez entrenada la red neuronal obtenemos el modelo, el cual luego es utilizado para reconocer el tipo de objeto de imágenes nuevas tomadas a los mismos en los entornos de los robots.

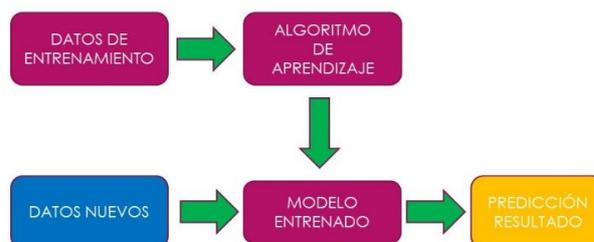


Fig. 2. Proceso de aprendizaje automático supervisado.

Proceso de recopilación de datos: El procedimiento comienza con la recopilación de imágenes de los diferentes objetos a reconocer utilizando la cámara web. Estas imágenes se cargan en la plataforma "Teachable Machine", donde se entrena la red neuronal. Para ello se crean las etiquetas que identificarán a cada objeto, en este caso "Objeto 1" y "Objeto 2", y se capturan con la cámara las imágenes en diferentes ángulos, distancias y condiciones de iluminación, para cada objeto a reconocer, Fig. 3. Este paso es crucial, ya que un conjunto de datos diverso mejora la precisión del modelo de aprendizaje automático.



Fig. 3. Datos de entrenamiento.

Preparación del modelo: En la preparación del modelo la plataforma construye la red neuronal adecuada a los datos de entrenamientos cargados, Fig. 4. Para ello se establecen los parámetros de configuración tales como:

- **Épocas:** Se han configurado 50 épocas, lo que significa que el modelo pasará por todo el conjunto de datos 50 veces durante el entrenamiento.
- **Tamaño del Lote:** Se ha establecido un tamaño de lote de 16. Esto indica que el modelo procesará 16 muestras de imágenes a la vez antes de actualizar los parámetros del modelo.
- **Tasa de Aprendizaje:** La tasa de aprendizaje se ha fijado en 0.001, controlando la magnitud de las actualizaciones de los parámetros del modelo en cada paso del entrenamiento.

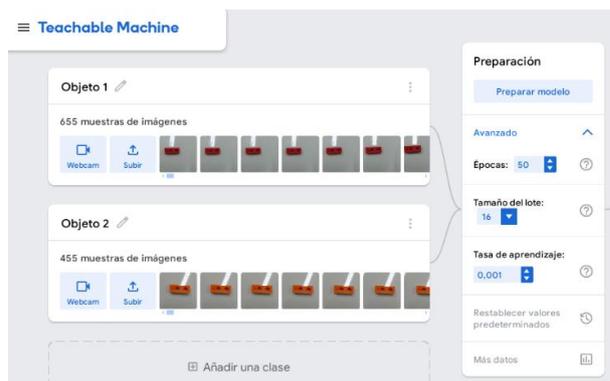


Fig. 4. Entrenamiento de la red neuronal

Entrenamiento del modelo: Al hacer clic en "Preparar modelo", la plataforma inicia el proceso de entrenamiento, Fig. 4. Durante este proceso, las imágenes de cada objeto se utilizan para ajustar los pesos y sesgos de la red neuronal, de modo que el modelo entrenado luego pueda distinguir entre los diferentes objetos.

Prueba del modelo entrenado: Mediante la misma plataforma se corrobora el correcto aprendizaje del modelo obtenido, adquiriendo imágenes nuevas mediante la cámara web o cargadas desde archivos, observando la correcta predicción obtenida sobre el tipo de objeto y el porcentaje de certeza de la predicción, Fig. 5.

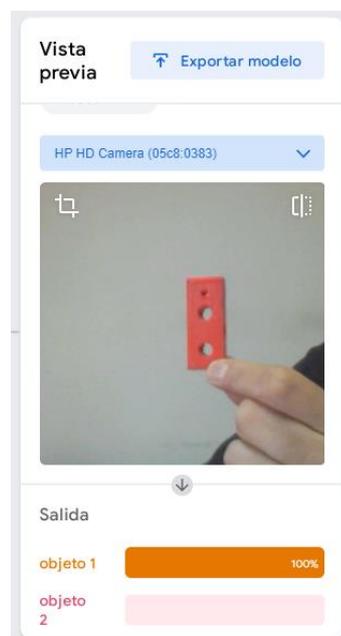


Fig. 5. Prueba del modelo entrenado.

Descarga del modelo entrenado: Una vez finalizado el entrenamiento, se descarga el modelo el cual incluye archivos como el de etiquetas y el de pesos y sesgos, que representan el conocimiento adquirido por la red neuronal. Dicho archivos son utilizados luego por el software de visión realizado en Python.

C. Software de visión artificial

El software desarrollado en el laboratorio fue implementado mediante lenguaje Python y la biblioteca OpenCV, Fig. 6. Dicha biblioteca de visión artificial de código abierto proporciona herramientas para el procesamiento y la mejora de imágenes, tales como el redimensionamiento de imágenes, la eliminación de ruido y el aumento del contraste, entre otras, funciones que preparan las imágenes para ser utilizadas como entrada en el modelo entrenado.

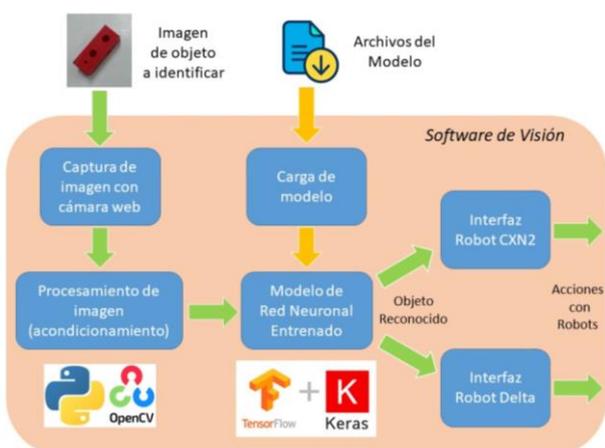


Fig. 6. Diagrama de bloques de software de visión.

Dicho modelo previamente entrenado en Teachable Machine, se carga en el software desarrollado para ser utilizado por la aplicación. Para ello se utiliza la biblioteca Keras de código abierto que proporciona una interfaz de alto nivel para el diseño y el entrenamiento de redes neuronales artificiales. Originalmente desarrollada como una biblioteca independiente, Keras se ha integrado completamente en TensorFlow, que es una plataforma de código abierto desarrollada por Google para la computación numérica y el aprendizaje automático. Una vez cargado el modelo y las etiquetas utilizadas, mediante la aplicación se pueden tomar imágenes de los objetos a reconocer, mediante la cámara web, en los entornos de los robots manipuladores, Fig 7. Las mismas son procesadas por el modelo cargado obteniendo la etiqueta del objeto reconocido y el

porcentaje de certeza de la predicción, Fig.8. A partir de ello se toman las decisiones y acciones correspondientes sobre los robots manipuladores.

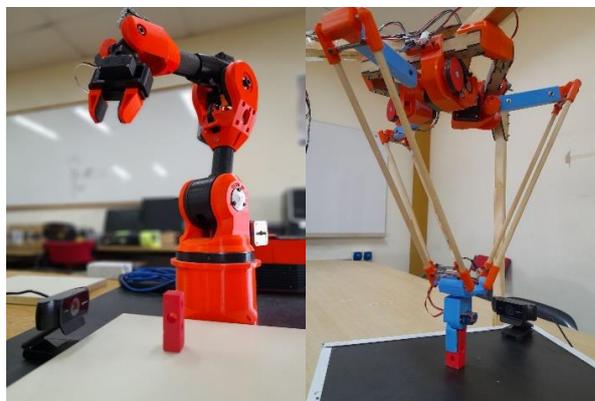


Fig. 7. Sistema de visión en el entorno de los robots manipuladores.

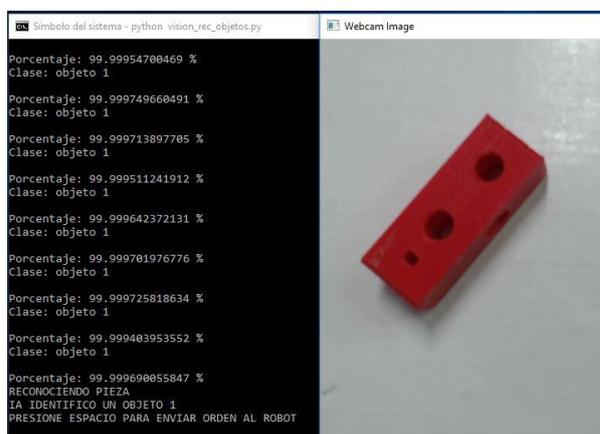


Fig. 8. Aplicación de visión identificando un objeto.

Para la interacción con los robots manipuladores, el software desarrollado envía los comandos específicos de cada robot para ejecutar tareas, según la clase de objeto identificada. En el caso del robot antropomorfo, el software se comunica con el controlador del robot mediante una red Ethernet y sockets TCP/IP. Dichos sockets son un mecanismo de comunicación que permite la transferencia de datos entre dispositivos en una red, asegurando la entrega confiable de datos. Esto permite enviar órdenes al robot para que recoja el objeto identificado de una posición conocida y lo transporte a una posición establecida para cada clase de objeto reconocido.

En el caso del robot Delta, la comunicación se realiza a través de una conexión serial USB, con la placa Arduino (controlador del robot). En este trabajo, para controlar el robot Delta, se envían desde el software desarrollado los



códigos G de las tareas correspondientes a cada objeto identificado. Una vez que el software de visión artificial detecta y clasifica un objeto, determina la acción específica que el robot Delta debe llevar a cabo. Esta acción puede implicar mover el robot hacia la ubicación del objeto identificado, recogerlo y transportarlo a una posición designada previamente, todo esto pre-programado mediante código G.

III. RESULTADOS

La implementación del sistema de visión artificial para los robots manipuladores educativos ha demostrado ser altamente eficaz en diversos aspectos. Los resultados obtenidos se pueden desglosar en varias áreas clave:

Precisión del Reconocimiento de Objetos: Utilizando la plataforma "Teachable Machine" para entrenar modelos de machine learning, se logró una alta precisión en el reconocimiento y clasificación de objetos. Las pruebas demostraron que el modelo podía identificar correctamente los objetos con una tasa de acierto superior al 95%, incluso en condiciones de iluminación y ángulos variados.

Rendimiento del Sistema en Tiempo Real: El sistema de visión artificial desarrollado en Python, utilizando OpenCV, fue capaz de procesar imágenes en tiempo real. Esto permitió a los robots tomar decisiones inmediatas sobre la manipulación de objetos, demostrando la viabilidad del sistema para aplicaciones que requieren respuestas rápidas y precisas.

Interacción Robot-Controlador: La comunicación entre la computadora y los controladores de los robots (mediante TCP/IP para el robot antropomorfo y USB serial para el robot Delta) se realizó de manera fluida y confiable. Esto permitió una integración efectiva entre el software de visión artificial y los sistemas de control de los robots, asegurando que las acciones de los robots fueran precisas y coordinadas según el tipo de objeto identificado.

Entornos Educativos y de Investigación: En el contexto educativo, el sistema proporcionó una plataforma interactiva para que los estudiantes aprendieran sobre visión artificial y robótica de manera práctica. Los estudiantes pudieron observar cómo los robots reconocían y manipulaban objetos en tiempo real, lo cual enriqueció significativamente su experiencia de aprendizaje. En el ámbito de la investigación, el sistema demostró ser una herramienta valiosa

para explorar nuevas aplicaciones de machine learning y visión artificial en robótica, facilitando experimentos y desarrollos futuros.

Flexibilidad y Adaptabilidad: La capacidad del sistema para ser adaptado a diferentes tipos de objetos y tareas sin requerir modificaciones significativas en el hardware o software subraya su flexibilidad. Esta adaptabilidad es crucial para su aplicación en diversos escenarios educativos y de investigación, permitiendo una amplia gama de experimentos y proyectos.

IV. CONCLUSIONES

La integración de la visión artificial y el Machine Learning en robots manipuladores educativos ha mostrado ser una herramienta poderosa y versátil para la enseñanza y la investigación en robótica. Algunas de las principales conclusiones derivadas de este trabajo son:

El sistema desarrollado facilita un aprendizaje más interactivo y práctico para los estudiantes de ingeniería y ciencias afines. Al permitir que los estudiantes trabajen con robots que pueden reconocer y manipular objetos en tiempo real, se crea un puente crucial entre la teoría y la práctica, mejorando la comprensión de conceptos complejos y su aplicación real.

Utilizar plataformas accesibles como "Teachable Machine" y bibliotecas robustas como OpenCV demuestra que es posible desarrollar aplicaciones avanzadas de visión artificial con recursos relativamente limitados. Esto abre la puerta para que instituciones educativas con presupuestos restringidos puedan implementar tecnologías de vanguardia en sus programas.

La capacidad de los robots para reconocer y manipular objetos con precisión en tiempo real mejora significativamente la interacción robot-humano. Esto no solo es beneficioso en entornos educativos, sino que también tiene aplicaciones potenciales en la industria, donde robots colaborativos pueden trabajar junto a humanos de manera más efectiva y segura.

Este trabajo establece una base sólida para futuras investigaciones y desarrollos en el campo de la robótica educativa. La combinación de machine learning, redes neuronales y visión artificial ofrece un amplio campo para explorar nuevas aplicaciones y mejorar las capacidades de los robots manipuladores.



La implementación exitosa de este sistema en el laboratorio de Mecatrónica subraya el potencial transformador de la inteligencia artificial y la visión artificial en la educación y la investigación en robótica. Al incorporar estas tecnologías avanzadas en el currículo educativo, se fomenta una cultura de innovación y experimentación entre los estudiantes y profesionales del campo.

REFERENCIAS

- [1] Z. Q. Zeng, S. X. Wu, and H. Z. Huang - "Vision-based Robot Control in Education: A Review" *Journal of Robotics and Automation* – 2021.
- [2] Robert Laganière - "OpenCV Computer Vision Application Programming Cookbook" - Packt Publishing. Año: 2017.

Desarrollo de Aplicaciones Interactivas para la Robótica Educativa con Godot Engine y ESP32

Ing. Andrés Tapari

Paraná, Entre Ríos, Argentina

andrestapari@alu.frp.utn.edu.ar

Resumen — Este trabajo presenta una plataforma educativa de robótica basada en la integración de Godot Engine y ESP32, utilizando el protocolo UDP para su comunicación en tiempo real. La plataforma permite a los estudiantes controlar un robot didáctico, ejecutando tareas a través de una aplicación interactiva. El enfoque fomenta el aprendizaje práctico, aprovechando tecnologías accesibles y de código abierto. Los resultados evidencian una interacción fluida entre la aplicación y el robot, proporcionando una herramienta flexible para diversos entornos educativos.

Palabras clave: Robótica educativa, ESP32, Godot Engine, Protocolo UDP, interacción en tiempo real.

I. INTRODUCCIÓN

La robótica educativa es una herramienta útil en la enseñanza de habilidades tecnológicas, permitiendo que los estudiantes aprendan sobre programación, electrónica y control mediante la interacción con robots. Con el avance de tecnologías de código abierto como ESP32 [1] y el motor de videojuegos Godot Engine [2], se han generado nuevas oportunidades para crear entornos de aprendizaje interactivos y accesibles. Sin embargo, la integración de hardware y software en tiempo real de manera económica sigue siendo un reto en muchos entornos educativos.

Este trabajo presenta una solución que combina estas dos tecnologías, utilizando el protocolo UDP [3] para permitir la comunicación en tiempo real y facilitar el control de un robot didáctico. La plataforma [4] no sólo es accesible para la enseñanza de robótica, sino que

también subraya el potencial de estas tecnologías para mejorar el proceso educativo.

II. CARACTERÍSTICAS

A. Arquitectura del sistema

El sistema se compone de dos elementos principales: una aplicación desarrollada en Godot Engine, que actúa como interfaz gráfica y servidor de comunicación, y un robot didáctico, Fig. 1, controlado por el ESP32. La comunicación se realiza a través de una red Lan Wi-Fi, que permite el envío rápido de comandos desde la interfaz al SoC.

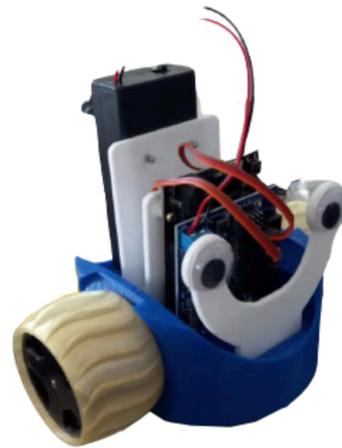


Fig. 1. Robot didáctico modular, basado en el proyecto JUNO ROVER [5].

B. Servidor UDP

El servidor UDP, nativo del motor, recibe las instrucciones del usuario desde la interfaz gráfica, transformándolas en comandos que se envían al ESP32 a través de una red Lan Wi-Fi.

C. ESP32 y Control del Robot

Este SoC, elegido por su accesibilidad y su

capacidad de conectarse a redes Wi-Fi, facilita el control remoto del robot. Este dispositivo recibe comandos enviados desde la aplicación y los convierte en acciones físicas, como el movimiento de motores o la activación de diferentes sensores.

D. Interacción y control de usuario

La aplicación permite a los usuarios diseñar rutinas de movimiento personalizadas para que el robot las ejecute según las actividades requeridas, ya sea en un contexto educativo o lúdico. Esta interfaz ha sido concebida para ser intuitiva y flexible, ofreciendo la posibilidad de adaptar los comandos y funciones a las necesidades específicas del entorno educativo. Esto facilita la creación de experiencias interactivas, ajustadas a las habilidades y objetivos pedagógicos de los estudiantes.

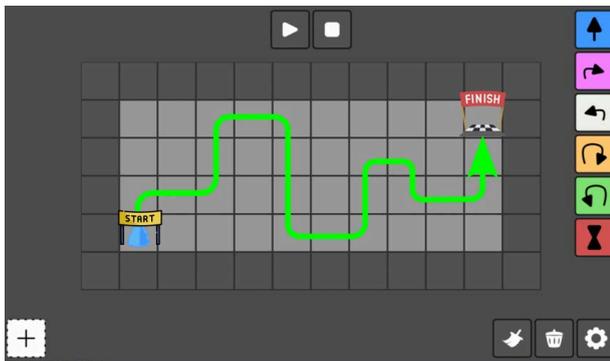


Fig. 2. Aplicación "Domino" desarrollada en Godot Engine 3.x

III. RESULTADOS

Los resultados obtenidos durante las pruebas [6] mostraron que el sistema es capaz de gestionar la comunicación en tiempo real entre Godot y el ESP32, permitiendo a los estudiantes interactuar fácilmente con el robot didáctico. Además, la plataforma demostró ser fácilmente escalable, controlando múltiples robots sin pérdida de rendimiento, lo que permite su aplicación en entornos de colaboración o competencia.

IV. CONCLUSIONES

Este trabajo demuestra que la integración de Godot Engine y ESP32 mediante el protocolo UDP ofrece una solución accesible y eficaz para la enseñanza de robótica en entornos educativos. Permite plantear distintas actividades áulicas, pudiendo adaptarse a

distintos niveles educativos, como otros instrumentos innovadores de la educación, este sistema podría influir en la motivación de los alumnos.

A medida que las tecnologías de código abierto continúan evolucionando, esta plataforma puede expandirse para abordar nuevas necesidades educativas y tecnológicas, como por ejemplo, el puerto bluetooth del egine [7] aún no está implementado de forma nativa, lo que podría replantear el uso del servidor UDP.

REFERENCIAS

- [1] Espressif Systems, "ESP32 datasheet," *Espressif Systems*, [En línea]. Disponible en: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. [Accedido: 01-sep-2023].
- [2] Godot Engine, "Godot Engine – Free and open-source 2D and 3D game engine," *Godot Engine*. [En línea]. Disponible en: <https://godotengine.org/>. [Accedido: 01-sep-2023].
- [3] Godot Engine, "Class UDP Server," *Godot Engine Documentation*. [En línea]. Disponible en: https://docs.godotengine.org/en/stable/classes/class_udpserver.html. [Accedido: 01-sep-2023].
- [4] A. Tapari, *Domino*, GitHub repository. [En línea]. Disponible en: <https://github.com/andresTapari/domino>. [Accedido: 01-sep-2023].
- [5] ExploreMaking, "Juno Rover: Intro to electronics and coding," *Thingiverse*, 2016. [En línea]. Disponible en: <https://www.thingiverse.com/thing:1720394>. [Accedido: 01-sep-2023].
- [6] A. Tapari, "Conectar ESP32 Parte 4/4," YouTube, 2023. [En línea]. Disponible en: <https://www.youtube.com/watch?v=YJ-H4M448ls>. [Accedido: 01-sep-2023].
- [7] GDWired, *GDSimpleBLE*, GitHub repository. [En línea]. Disponible en: <https://github.com/GDWired/GDSimpleBLE>. [Accedido: 01-sep-2023].



Diseño de Robot de Exploración para Entornos Peligrosos con Sistema de Monitoreo y Control Basado en el microcontrolador ESP32

Edwin Yair Osorio Cuatecatl

Ingeniería Mecatrónica

Universidad del Valle de Puebla

Puebla, México

im43864@uvp.edu.mx

Emanuel Ojeda Perez

Ingeniería Mecatrónica

Universidad del Valle de Puebla

Puebla, México

im44099@uvp.edu.mx

Alma S. Lizet Salazar Valdez

Ingeniería Mecatrónica

Universidad del Valle de Puebla

Puebla, México

coord.mecatronica@uvp.mx

Resumen— Este trabajo presenta el desarrollo de un sistema robótico de exploración para entornos inaccesibles o peligrosos, como cuevas o zonas con presencia de amenazas químicas o gases. El sistema se basa en un microcontrolador ESP32 [1], que gestiona la comunicación y el control del robot. La comunicación entre los módulos del robot (motores, sensores, actuadores) se realiza mediante el protocolo IBUS [2], una solución robusta y eficiente. El ESP32 permite la integración de diversos sensores para la detección de gases [3], temperatura, humedad y otros parámetros ambientales. Los datos recogidos se transmiten de forma inalámbrica a una estación base para su visualización y análisis en tiempo real. El sistema IBUS facilita la escalabilidad del robot, permitiendo la incorporación de nuevos módulos. Este sistema robótico ofrece una solución flexible y adaptable para la exploración remota de entornos peligrosos, minimizando el riesgo para los operadores humanos y proporcionando información crucial para la evaluación de riesgos y la toma de decisiones en situaciones de emergencia. El uso de ESP32 e IBUS permite una implementación eficiente y de bajo costo, abriendo posibilidades para la investigación y exploración en áreas previamente inaccesibles.

Palabras clave: Microcontrolador, Robot explorador, ESP32 e IBUS.

I. INTRODUCCIÓN

El diseño de robots exploradores para entornos peligrosos es una estrategia fundamental en la resolución de problemas,

especialmente en áreas inaccesibles o que presentan riesgos para la salud humana, como la exploración espacial [4], la inspección de zonas de desastre [5], o la manipulación de materiales peligrosos [6]. Estos robots permiten que las habilidades humanas se integren sin poner en riesgo la salud o la integridad física, especialmente en áreas de difícil acceso en la naturaleza [7].

Para detectar las necesidades estructurales, mecánicas, electrónicas y de control que se requieren para brindar una solución concreta al problema, se tiene que verificar en el diseño del robot para el rastreo de personas. Se han tenido en consideración agentes del entorno natural, pensando en las condiciones severas de los terrenos a explorar, actuadores mecánicos capaces de proporcionar la potencia necesaria al robot, una etapa de control y un procesamiento óptimo de información que permitan una navegación deseada y enfocada a cumplir con el objetivo principal de detectar gases peligrosos [8]. Para lo cual se requiere un sistema embebido o computadora que será idónea para procesar la información de los sensores y a su vez que pueda controlar los movimientos del robot. En este caso, la elección del microcontrolador ESP32 se justifica por su conectividad Wi-Fi y Bluetooth [9], que facilita la comunicación y control remoto; su potente procesador dual-core de 32 bits, ideal para procesar algoritmos complejos; su amplio número de pines GPIO, que permite conectar múltiples sensores y actuadores; su bajo consumo energético, crucial para misiones prolongadas; y su costo accesible.

En el trabajo se muestra el diseño del robot, incluyendo las piezas diseñadas para las articulaciones [10]. Para el diseño se utilizaron 2 softwares: Solidworks [11] y Autodesk: Fusion 360 [12]. Se observa el análisis de los componentes electrónicos, así como la simulación del circuito, verificando las capacidades del microcontrolador ESP32 para su aplicación en un robot explorador.

La elección de Solidworks y Autodesk Fusion 360 se justifica por su amplio uso en el diseño de robots [13], su capacidad para modelar piezas complejas en 3D y la posibilidad de realizar simulaciones para comprobar el funcionamiento del diseño antes de su construcción [14]. El análisis del circuito electrónico y la simulación de este permiten asegurar la correcta integración de los diferentes componentes y prever posibles problemas antes de la implementación física [15].

II. DISEÑO DEL ROBOT EXPLORADOR EN CAD

El diseño del robot fue realizado en conjunto con los softwares CAD Solidworks y Autodesk Fusion 360, destacando en primer lugar el brazo articulado, diseñado para la verificación in situ de materiales presentes en el terreno. En la Figura 1, se presenta un modelo 3D del brazo articulado, donde se pueden apreciar sus diferentes componentes, como las articulaciones, los eslabones y el efector final (garra o herramienta). Este brazo robótico se ha diseñado con el objetivo de alcanzar y manipular objetos o muestras en el entorno del robot, permitiendo la recolección de datos o la interacción física con el terreno. El diseño de brazos robóticos articulados se basa en principios cinemáticos y dinámicos que permiten un amplio rango de movimiento y precisión en la manipulación de objetos [16].

El efector final, en este caso una garra, puede ser diseñado con características adaptativas para mejorar la capacidad de agarre en diferentes tipos de objetos y superficies. Las garras adaptativas, que pueden ajustar su forma y fuerza de agarre en función de las características del objeto, se han utilizado con éxito en robots para la manipulación de objetos delicados o de formas irregulares [17].

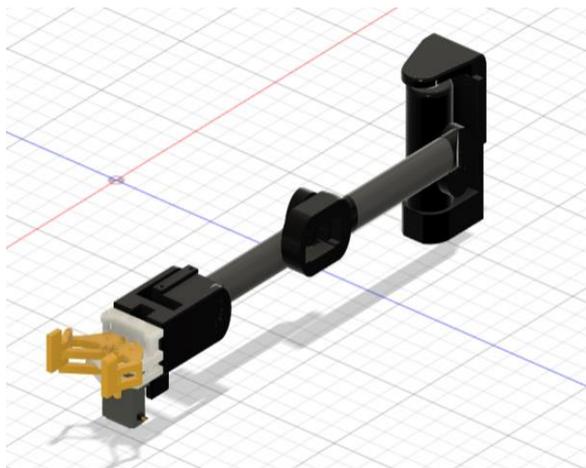


Fig. 1. Diseño de prototipo en Solidworks del brazo articulado del robot para verificación de materiales

En la Figura 2, se puede apreciar un modelado 3D realizado en Autodesk Fusion 360 del sistema rocker-bogie de seis ruedas. Su sistema de locomoción se basa en estas seis ruedas que operan de forma sincronizada, garantizando la estabilidad y movilidad del robot. Dicha figura ilustra la composición del sistema, mostrando la disposición de las ruedas, los brazos de suspensión (bogies) y el mecanismo de enlace central (rocker). Además, la Figura 2 permite visualizar cómo este sistema opera, destacando la capacidad de las ruedas para adaptarse a terrenos irregulares gracias a la flexibilidad del rocker y los bogies. El sistema rocker-bogie ha demostrado ser una solución eficiente para la locomoción de robots en terrenos irregulares, como se ha evidenciado en misiones espaciales como la del Rover Curiosity [18].

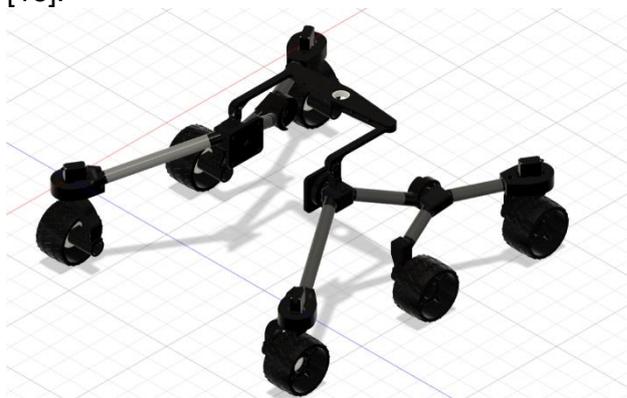


Fig. 2. Diseño de prototipo en Autodesk Fusion 360 del sistema rocker-bogie para la locomoción del robot.

Fig. 5. Circuito electrónico representativo de funcionamiento del robot

La Figura 5 ilustra la arquitectura del sistema robótico desarrollado. El núcleo del sistema es una placa de desarrollo TTGO basada en el microcontrolador ESP32 [20], que integra una pantalla OLED para la monitorización del estado y la visualización de datos. El ESP32 gestiona el procesamiento de las señales de control, la comunicación con los demás módulos y el envío de comandos a los actuadores. Todos los demás componentes del sistema se conectan al ESP32 a través de sus pines GPIO (General Purpose Input/Output) y buses de comunicación como I2C, SPI y UART. La elección del ESP32 como microcontrolador principal se justifica por su versatilidad, bajo consumo energético y capacidad de procesamiento, características que lo hacen ideal para aplicaciones robóticas. Como señalan Maiolino et al. [21], los microcontroladores ESP32 se han convertido en una opción popular para el desarrollo de sistemas robóticos debido a su combinación de características como conectividad WiFi/Bluetooth, procesamiento potente y bajo costo.

La comunicación inalámbrica de largo alcance se logra mediante un módulo TTGO LoRa [22] integrado, conectado al ESP32 mediante SPI, que facilita la teleoperación del robot en entornos extensos o con obstáculos que limitan la comunicación por radiofrecuencia convencional. La tecnología LoRa se caracteriza por su bajo consumo energético y su capacidad para transmitir datos a largas distancias, lo que la hace adecuada para aplicaciones de telemetría y control remoto en entornos hostiles. Centenaro et al. [23] destacan que la tecnología LoRa se ha utilizado con éxito en aplicaciones de monitoreo ambiental y control industrial, donde se requiere una comunicación inalámbrica de largo alcance y bajo consumo energético.



Fig. 6. Prueba de LoRa con dos placas de desarrollo TTGO LoRa32 2.1 1.6

El sistema de propulsión está compuesto por seis motores DC de 12V y 40 rpm, controlados individualmente por dos controladores de velocidad electrónicos (ESC). Los ESC se conectan al ESP32 a través de pines PWM (Pulse Width Modulation), permitiendo al microcontrolador regular la velocidad de los motores. Esta configuración sugiere una plataforma de tracción de seis ruedas o un sistema de movimiento omnidireccional, permitiendo un control preciso del movimiento del robot. El uso de motores DC con ESC proporciona una solución eficiente y flexible para el control de la velocidad y la dirección de los motores en robots móviles. Precup. et al. [24] explican que la modulación por ancho de pulso (PWM) es una técnica ampliamente utilizada para el control de la velocidad de motores DC en robots, ya que permite una regulación precisa y eficiente de la potencia entregada a los motores.

Para la teleoperación visual, se incorpora una cámara FPV que captura imágenes en tiempo real y las transmite a un receptor remoto (no mostrado) a través de un módulo transmisor de video. El módulo transmisor de video se conecta al ESP32, que puede controlar la transmisión de video o capturar imágenes estáticas según sea necesario. Esta funcionalidad proporciona al operador una vista en primera persona del entorno, mejorando la navegación y la toma de decisiones. La integración de sistemas de visión en robots exploradores permite la teleoperación remota y la toma de decisiones en tiempo real, mejorando la eficiencia y la seguridad de las misiones. Como se describe en [25], la visión artificial se ha convertido en una herramienta fundamental para la robótica, permitiendo a los robots percibir e interactuar con su entorno de manera más autónoma e inteligente.

Un receptor RC FLYSKY se encarga de recibir



las señales de control enviadas por un transmisor operado por el usuario. El receptor RC se conecta al ESP32, que decodifica las señales recibidas y las traduce en comandos para controlar el movimiento del robot. El uso de sistemas de radio control (RC) proporciona una forma intuitiva y flexible para la teleoperación de robots, especialmente en aplicaciones donde se requiere un control manual preciso. Kim et al. [26] afirman que los sistemas de radio control (RC) son ampliamente utilizados en la robótica para la teleoperación de robots móviles y manipuladores, permitiendo a los operadores controlar los movimientos del robot a distancia.

El diseño del sistema contempla la integración de un módulo de navegación, como un GPS, una brújula o un sistema de posicionamiento inercial. Este módulo se comunicaría con el ESP32 a través de interfaces como I2C o UART, proporcionando datos de posición y orientación al microcontrolador para la navegación autónoma o el seguimiento de rutas. La incorporación de sistemas de navegación permite a los robots exploradores determinar su posición y planificar rutas de forma autónoma, mejorando la eficiencia y la seguridad de las misiones. En [27] se describe cómo los sistemas de navegación inercial (INS) y GPS se utilizan en robots exploradores para proporcionar información precisa sobre la posición y orientación del robot, permitiendo la navegación autónoma en entornos desconocidos.

La alimentación del sistema se realiza mediante una batería de polímero de litio de 3 celdas (3S Lipo). Un convertidor DC-DC regula la tensión de la batería para adaptarla a los requerimientos de los diferentes componentes. El ESP32 puede monitorizar el voltaje de la batería a través de un pin analógico y gestionar el consumo de energía del sistema. La elección de baterías de polímero de litio se justifica por su alta densidad energética y su capacidad para proporcionar una corriente de descarga elevada, características importantes para robots móviles que requieren una autonomía considerable. Goodenough and Kim [28] señalan que las baterías de polímero de litio (LiPo) son ampliamente utilizadas en robots móviles debido a su alta densidad energética, su bajo peso y su capacidad para proporcionar una corriente de descarga elevada.

Finalmente, se incluye un servo motor para el

control de la dirección de la cámara FPV o la gestión de otros mecanismos auxiliares del robot. El servo motor se conecta al ESP32 a través de un pin PWM, permitiendo al microcontrolador controlar su posición con precisión. Los servos motores son actuadores electromecánicos que permiten un control preciso de la posición angular, lo que los hace ideales para aplicaciones como el control de la dirección de cámaras o la manipulación de objetos. Como se explica en [29], los servos motores son ampliamente utilizados en robótica para el control de la posición de articulaciones y mecanismos, gracias a su capacidad para proporcionar un control preciso de la posición angular.

Movilidad y Control del Robot Explorador

El robot explorador se ha diseñado teniendo en cuenta las necesidades de movilidad, robustez, y capacidad de carga útil para la integración de sensores y actuadores. Se ha optado por una configuración de seis ruedas con tracción independiente, que proporciona una gran estabilidad y maniobrabilidad en terrenos irregulares. Los materiales utilizados en la construcción del chasis y las ruedas se han seleccionado por su resistencia a los impactos y a las condiciones ambientales adversas.

La elección de una configuración de seis ruedas con tracción independiente se justifica por su capacidad para superar obstáculos y mantener la estabilidad en terrenos irregulares, características importantes para robots exploradores que operan en entornos hostiles. Thrun et al. [30] afirman que la configuración de seis ruedas con tracción independiente es una solución popular para robots móviles que operan en terrenos irregulares, ya que proporciona una mayor estabilidad y capacidad para superar obstáculos.

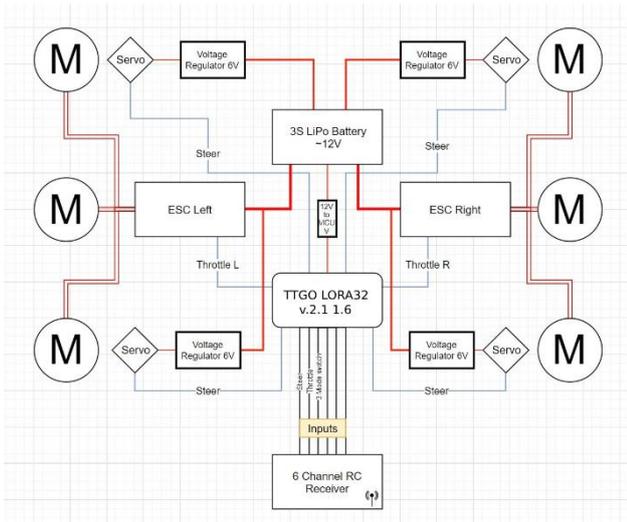


Fig. 7. Diagrama electrónico de control de movimiento del robot explorador

El sistema de control del robot se basa en el microcontrolador ESP32 [31], que ofrece un buen equilibrio entre potencia de procesamiento, capacidad de comunicación inalámbrica, y bajo consumo energético. Para la detección de gases se ha seleccionado un sensor específico para el tipo de gas que se desea detectar, mientras que para la medición de otros parámetros ambientales se han integrado sensores de temperatura, humedad, y presión. Los motores DC se han elegido por su eficiencia y facilidad de control, y se han dimensionado para proporcionar la potencia necesaria para el movimiento del robot. La selección de los sensores y actuadores del robot se basa en las necesidades específicas de la aplicación, teniendo en cuenta factores como la precisión, el rango de medición, el consumo energético y la robustez. Como se describe en [32], la elección de los sensores y actuadores para un robot explorador depende de las necesidades específicas de la aplicación, como el tipo de entorno en el que operará, las tareas que debe realizar y los requisitos de precisión y robustez.

La interacción entre las ruedas del robot y el terreno se modela utilizando el criterio de falla de Coulomb, que describe la relación entre las tensiones normales y tangenciales en la interfaz rueda-suelo. Este modelo permite predecir el comportamiento del robot en diferentes tipos de terreno a partir de los parámetros del suelo, como la cohesión, el ángulo de corte, y el módulo de deformación por corte. [33] El modelo de Coulomb es ampliamente utilizado para la simulación de la interacción rueda-terreno en robots móviles, ya que proporciona una buena

aproximación del comportamiento del suelo en un amplio rango de condiciones. Bekker [34] describe en detalle el modelo de Coulomb y su aplicación en la mecánica de suelos.

Para lograr una navegación autónoma eficiente, es necesario que el robot pueda estimar los parámetros del terreno en tiempo real. Se han propuesto diferentes técnicas para lograr esto, incluyendo métodos analíticos, filtros de Kalman, y redes neuronales. [35] La estimación de los parámetros del terreno en tiempo real es un desafío importante para la navegación autónoma de robots, ya que permite al robot adaptar su comportamiento a las condiciones del terreno y mejorar la eficiencia y la seguridad de la navegación. Olson et al. [36] presentan un ejemplo de cómo se pueden utilizar técnicas de visión estereó para la estimación del movimiento del robot y la caracterización del terreno.

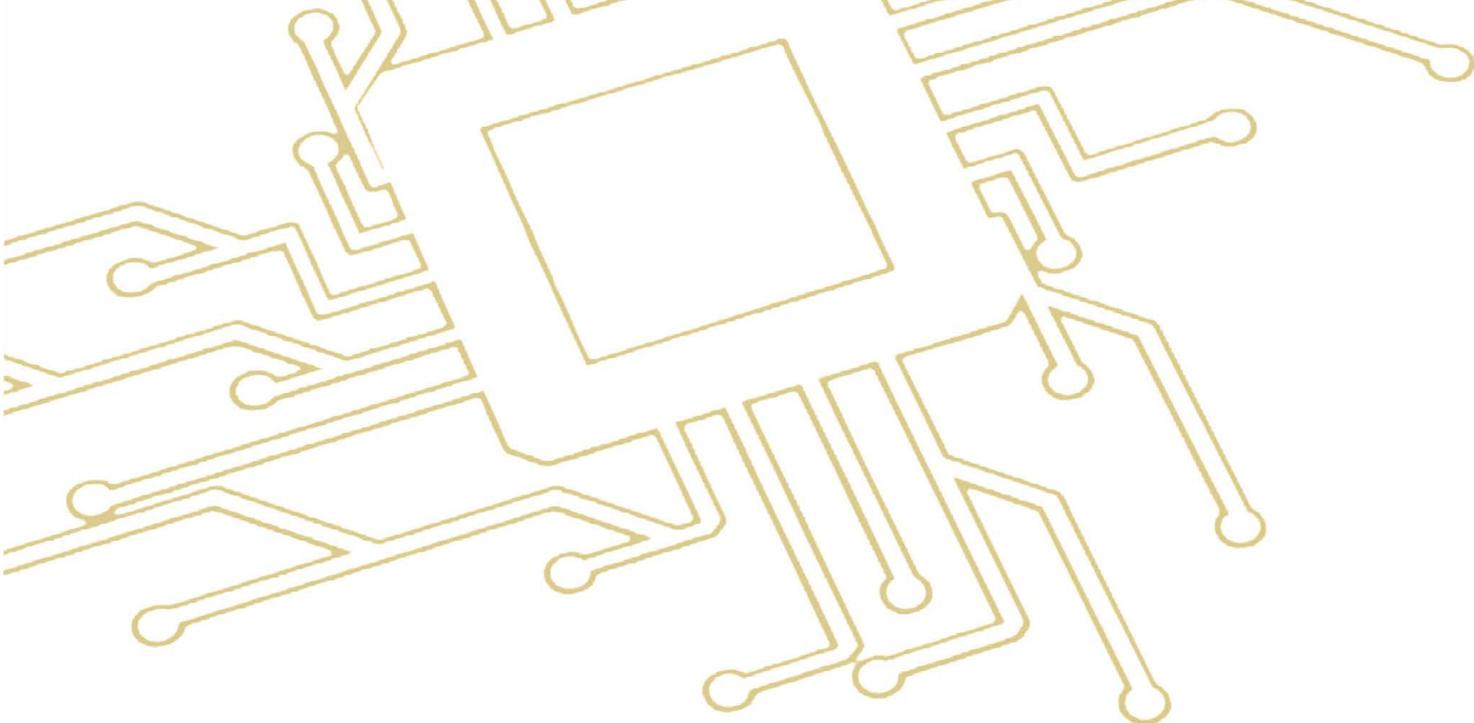
Se planea realizar pruebas de campo del robot en un entorno real, para validar su funcionamiento en condiciones más complejas. Además, se trabajará en la mejora de los algoritmos de control y en la integración de nuevas funcionalidades. La realización de pruebas de campo en entornos reales es fundamental para validar el funcionamiento de los robots exploradores y asegurar su capacidad para operar de forma segura y eficiente en las condiciones para las que fueron diseñados. Wettergreen et al. [37] describen un ejemplo de pruebas de campo realizadas con un robot explorador en un entorno desértico, destacando la importancia de validar el diseño del robot en condiciones reales de operación.

REFERENCIAS

- [1] Espressif Systems. (2023). ESP32 Series Datasheet. Retrieved from https://www.espressif.com/sites/default/files/documentation/ESP32_datasheet_en.pdf.
- [2] TTGO. (2023). T-Beam V1.1 ESP32 433/868/915/923MHz WiFi Bluetooth LoRa SX1278/SX1276 18650 Battery GPS NEO-6M SMA 0.96 Inch Blue OLED. Retrieved from <https://github.com/LilyGO/TTGO-T-Beam>
- [3] Yi, S., Wang, X., & Fringer, O. B. (2017). Wireless Sensor Networks for Gas Leak Detection: A Review. *Sensors*, 17(11), 2554.
- [4] Thangavelautham, J. (2012). Humanoid robots in spacecraft: The challenges in developing robots for space exploration. In 2012 IEEE International Conference on Robotics and Automation (pp. 5490-5495). IEEE.
- [5] Murphy, R. R. (2004). Disaster robotics. In Springer Handbook of Robotics (pp. 1351-1373). Springer, Berlin, Heidelberg.
- [6] Micire, M. J. (2007). Mobile robots for nuclear environments: A literature review. *Journal of Robotic Systems*, 24(1), 31-41.



- [7] Yoshida, K., Hamano, H., & Koyanagi, E. (2003). Development of an all-terrain rover for search and rescue operations. *Advanced Robotics*, 17(3), 249-262.
- [8] Siciliano, B., & Khatib, O. (Eds.). (2016). *Springer handbook of robotics*. Springer.
- [9] M. Ma, Y. Zhang, and W. Trappe, "ESP32: A Low-Cost, Low-Power SoC with Integrated Wi-Fi and Bluetooth for Internet of Things Applications," in 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), 2017, pp. 146-151.
- [10] Norton, R. L. (2019). *Diseño de maquinaria: un enfoque integrado* (5a ed.). Pearson Educación.
- [11] Solidworks. (s.f.). ¿Qué es SOLIDWORKS? Recuperado de <https://www.solidworks.com/es/solutions/what-is-solidworks>
- [12] Autodesk. (s.f.). Fusion 360. Recuperado de <https://www.autodesk.com/products/fusion-360/overview>
- [13] Mataric, M. J. (2007). *The robotics primer*. MIT press.
- [14] Corke, P. I. (2017). *Robotics, vision and control: fundamental algorithms in MATLAB® second*,
- [15] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed. Pearson Education, 2005.
- [16] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. MIT Press, 2011.
- [17] M. T. Mason and J. K. Salisbury, *Robot Hands and the Mechanics of Manipulation*. MIT Press, 1985.
- [18] NASA. (s.f.). Mars Science Laboratory Curiosity Rover. Recuperado de <https://mars.nasa.gov/msl/>
- [19] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, "Rover Navigation Using Stereo Ego-Motion," *Robotics and Autonomous Systems*, vol. 43, no. 4, pp. 215-229, 2003.
- [20] "DIY Mars Perseverance Rover Replica - Arduino based Project." *How To Mechatronics*. Accedido el 26 de septiembre de 2024. [En línea]. Disponible: <https://howtomechatronics.com/projects/diy-mars-perseverance-rover-replica-with-arduino/>
- [21] R. Maiolino, P. Gonçalves, and J. Sequeira, "ESP32-based Educational Mobile Robot for Industry 4.0," in 2020 IEEE Global Engineering Education Conference (EDUCON), 2020, pp. 1448-1455.
- [22] LilyGO, "TTGO LoRa32 V2.1 1.6 Datasheet," 2023. [Online]. Available: <https://github.com/LilyGO/TTGO-LoRa32-V2.1.6/blob/master/Datasheet/TTGO%20LoRa32-OLED%20V1.pdf>
- [23] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-Range Communications in Unlicensed Bands: The Rising Stars in the IoT and Smart City Scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60-67, 2016.
- [24] C. Precup, S. Preitl, and R. -E. Precup, "PI-Fuzzy Control of Mobile Robots," in *Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, 2010, pp. 1-6.
- [25] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer, 2011.
- [26] D. H. Kim, J. H. Kim, and J. S. Lee, "Development of a Teleoperated Mobile Robot System for Remote Operations in Hazardous Environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2001, pp. 3162-3167.
- [27] J. Borenstein, H. R. Everett, and L. Feng, *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd., 1996.
- [28] J. B. Goodenough and Y. Kim, "Challenges for Rechargeable Li Batteries," *Chemistry of Materials*, vol. 22, no. 3, pp. 587-603, 2010.
- [29] K. Ogata, *Modern Control Engineering*, 5th ed. Prentice Hall, 2010.
- [30] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.
- [31] J. Krajewski, "Imagen del diseño del robot explorador," Mars-Rover GitHub repository, Issue #2, 2019. [Online]. Available: <https://github.com/jakkra/Mars-Rover/issues/2>
- [32] J. L. Jones and A. M. Flynn, *Mobile Robots: Inspiration to Implementation*, 2nd ed. A. K. Peters, Ltd., 1999.
- [33] Nicol, C., Ellery, A., & Cloutis, E. (2013). On-Line Estimation of Soil Parameters Using the Kapvik Micro Rover. *Proceedings of the 7th International Conference on Sensing Technology (ICST)*, 165-174.
- [34] M. G. Bekker, *Introduction to Terrain-Vehicle Systems*. University of Michigan Press, 1969.
- [35] M. Cross, A. Ellery, and A. 'Qadi. "Estimating terrain parameters for a rigid wheeled rover using neural networks". In: *Journal of Terramechanics* 50 (2013), pp. 165-174.
- [36] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, "Rover Navigation Using Stereo Ego-Motion," *Robotics and Autonomous Systems*, vol. 43, no. 4, pp. 215-229, 2003.
- [37] D. Wettergreen, H. Thomas, and M. Buehler, "Experiments with the Ames Marsokhod Rover in Mojave Desert Terrain," in *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1999, pp. 1432-1437.



Eje Temático

Aplicaciones en Seguridad



Sensor de Inclinación y Vibración para Monitoreo de Apuntalamientos en Estructuras Colapsadas

Juan M. Souto*, Claudio J. Paz†

Centro de Investigación en Informática para la Ingeniería,
Facultad Regional Córdoba, Universidad Tecnológica Nacional
Emails: *juanmanuelsouto004@gmail.com, †cpaz@frc.utn.edu.ar

Abstract—Este trabajo presenta el desarrollo y la implementación de un primer prototipo funcional de un sensor de inclinación y vibración diseñado para su uso en escenarios de búsqueda y rescate en estructuras colapsadas. El sensor, basado en tecnologías MEMS, permite detectar cambios en la inclinación como así también vibraciones de la estructura donde se encuentra, facilitando la identificación de riesgos inminentes en estos entornos peligrosos. Se describe el diseño del sistema, los algoritmos de procesamiento de señal implementados, y las pruebas realizadas en entornos controlados que validan su eficacia. Los resultados demuestran que el sensor es capaz de operar en condiciones adversas, proporcionando información crítica que mejora la seguridad y efectividad de las operaciones de rescate.

Index Terms—MEMS, Inclinación, Vibración, Búsqueda y Rescate, Estructuras Colapsadas

I. INTRODUCCIÓN

Los sismos son uno de los desastres naturales más destructivos que enfrenta la humanidad. La liberación repentina de energía acumulada en las fallas geológicas provoca movimientos sísmicos que pueden colapsar edificaciones, dejando a su paso destrucción y caos. En estos escenarios, la labor de los equipos de búsqueda y rescate es crucial para salvar vidas atrapadas bajo los escombros. Los rescatistas se enfrentan a un entorno peligroso, con estructuras inestables que pueden colapsar en cualquier momento debido a réplicas o fallas. Para evitar estos peligros mientras los rescatistas realizan los procedimientos de búsqueda y rescate, se realizan apuntalamientos de las estructuras donde se trabaja. Los apuntalamientos son procedimientos mediante los cuales se refuerza la estructura en cuestión con *fortificaciones* realizadas con puntales de madera unidos con clavos, con la finalidad de evitar movimientos o colapso de la misma. En este contexto, es vital desarrollar tecnologías que permitan monitorear continuamente las condiciones de las estructuras colapsadas y los apuntalamientos, proporcionando alertas tempranas sobre posibles riesgos. Los parámetros que se desean relevar de forma continua son la inclinación y vibraciones que superen cierto umbral.

Para obtener la inclinación de un cuerpo se puede utilizar una combinación de acelerómetros y giróscopos. Estos sensores tienen características complementarias que se pueden aprovechar para determinar la orientación.

Este trabajo presenta un sensor de inclinación y vibración basado en tecnologías MEMS, diseñado específicamente



Fig. 1. SIVI: Sensor de Inclinación y Vibración.

para operar en escenarios de búsqueda y rescate. En la Fig. 1 se puede ver la primera versión del prototipo. El sensor fue evaluado en simulaciones y en pruebas de campo realizadas por el grupo *Rescate Internacional TOPOS*, donde demostró su utilidad en escenarios de estructuras colapsadas apuntaladas. Se espera que el sensor desempeñe un papel fundamental en la protección de los rescatistas, al proporcionar datos críticos sobre la estabilidad estructural en tiempo real.

Este trabajo continúa con tres secciones: La Sección II donde se formalizarán algunos conceptos utilizados en el desarrollo del dispositivo, así como las bases trigonométricas y algorítmicas para explicar su funcionamiento. La Sección III donde se detalla la implementación del dispositivo, tanto *software* como *hardware*, sus características y modo de uso. Y finalmente la Sección IV donde se resume el trabajo realizado, las experiencias logradas y los objetivos a mediano y largo plazo del uso que se le espera dar a este dispositivo.

II. METODOLOGÍA

Aunque *inclinación* y *orientación* son términos que a menudo se utilizan de manera intercambiable, tienen significados distintos, siendo uno una generalización del otro.

La inclinación se refiere a los ángulos que un objeto forma con respecto a un plano de referencia, generalmente el plano horizontal o la dirección de la gravedad. La inclinación describe cómo un objeto está inclinado en relación con la gravedad en uno o más ejes. Los ángulos de inclinación más comunes son *pitch* y *roll*. Suponiendo que un objeto tiene en su centro geométrico un sistema de ejes coordenados con el eje x apuntando hacia adelante y el eje y apuntando hacia la derecha del cuerpo, el *pitch* es el ángulo de inclinación alrededor del eje y , o sea, indica si el objeto está inclinado hacia adelante o hacia atrás. El *roll* es el ángulo de inclinación alrededor del eje x , esto quiere decir que indica si el objeto está inclinado hacia los lados (izquierda o derecha).

La inclinación es particularmente relevante en aplicaciones donde se necesita medir cómo se desvía un objeto de la verticalidad, como en dispositivos de nivelación o en la estabilización de plataformas [1].

En cambio la orientación describe la posición completa de un objeto en el espacio tridimensional, considerando todos los ángulos de rotación en torno a sus tres ejes principales: x , y , y z . La orientación se suele representar mediante alguna secuencia de tres ángulos conocidos como *ángulos de Euler* (*roll*, *pitch*, *yaw*), o mediante cuaterniones [2].

La orientación no solo incluye la inclinación, sino también la dirección en la que está apuntando el objeto en relación con un sistema de coordenadas fijo. Esto es importante en aplicaciones como la navegación, exploración o mapeo donde es importante conocer la dirección en la que se mueve un objeto, además de su inclinación.

Este trabajo solo tiene en cuenta la inclinación de un cuerpo, donde se considera que los ángulos son muy pequeños permitiendo que usar los ángulos de *roll* y *pitch* desacoplados, considerando solo las rotaciones independientes alrededor de los eje x y y .

A. Sensores MEMS

Los sensores *MEMS* (por *Microelectromechanical Systems*) son dispositivos miniaturizados que combinan componentes mecánicos y eléctricos en escalas micrométricas. Estos sensores son ampliamente utilizados en diversas aplicaciones debido a su tamaño reducido, bajo costo y capacidad de medir una amplia gama de parámetros físicos, como aceleración o velocidad angular, siendo los sensores más usados los *acelerómetros* y *giróscopos*.

A pesar de su nombre, los acelerómetros no miden aceleraciones, miden las fuerzas específicas actuando sobre el sensor. Por ejemplo, si se deja un acelerómetro en reposo sobre una superficie estática, se puede decir que el sensor no está sometido a ninguna aceleración, sin embargo la lectura en este caso es de $1g$ en el eje vertical debido a la fuerza ficticia de la superficie que *evita la caída libre* del sensor.

Los acelerómetros MEMS de tres grados de libertad se utilizan para medir esta fuerza proyectada en los tres ejes, proporcionando información sobre la inclinación relativa respecto a la gravedad.

La orientación angular, específicamente los ángulos de *pitch* (θ alrededor del eje x) y *roll* (ϕ alrededor del eje y), se puede calcular utilizando las ecuaciones

$$\theta_{\text{accel}} = \arctan\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \quad (1)$$

y

$$\phi_{\text{accel}} = \arctan\left(\frac{-a_x}{a_z}\right) \quad (2)$$

donde a_x , a_y y a_z son las componentes de la aceleración medidas por el sensor en los ejes x , y y z respectivamente. Estas ecuaciones suponen que el sensor está en reposo o se mueve a velocidades constantes, de modo que la aceleración medida proviene principalmente de la gravedad.

Aunque el acelerómetro puede dar una estimación de la orientación usando estas ecuaciones, sufre de limitaciones importantes. Una de ellas es que es sensible a las aceleraciones dinámicas, es decir, las aceleraciones debidas al movimiento del dispositivo, lo que puede introducir errores significativos en los cálculos de orientación. Además, el ruido de alta frecuencia inherente al sensor puede afectar la precisión de las mediciones, lo que hace necesario el uso de técnicas de filtrado y fusión de sensores para obtener estimaciones más precisas.

Los giróscopos son sensores que miden la velocidad angular de un objeto en movimiento. Funcionan mediante la detección del *efecto Coriolis*, que se manifiesta cuando una masa interna vibra y experimenta una fuerza proporcional a la rotación del sensor. Estos giróscopos son ampliamente utilizados en aplicaciones como estabilización de imágenes, navegación y control de vehículos, debido a su pequeño tamaño, bajo costo y alta precisión.

A partir de la velocidad angular se puede determinar la orientación angular del sistema si se integra estos valores respecto del tiempo. Suponiendo que la velocidad angular alrededor del eje x es $\omega_x(t)$, el ángulo θ alrededor de x se puede calcular mediante la integración de la velocidad angular con

$$\theta_{\text{gyro}}(t) = \theta_{\text{gyro}}(t-1) + \int \omega_x(t) dt \quad (3)$$

como así también para ϕ_{gyro} con

$$\phi_{\text{gyro}}(t) = \phi_{\text{gyro}}(t-1) + \int \omega_y(t) dt \quad (4)$$

A pesar de su precisión a corto plazo los giróscopos sufren de deriva en el tiempo debido a una componente constante en las mediciones del giróscopo llamada *bias* dada por su principio de funcionamiento. Esto provoca errores acumulativos en la estimación de la inclinación por integración.

Para mitigar este problema, se suelen combinar con otros sensores, como acelerómetros, en algoritmos de fusión de sensores, mejorando así la estimación de la orientación [3].

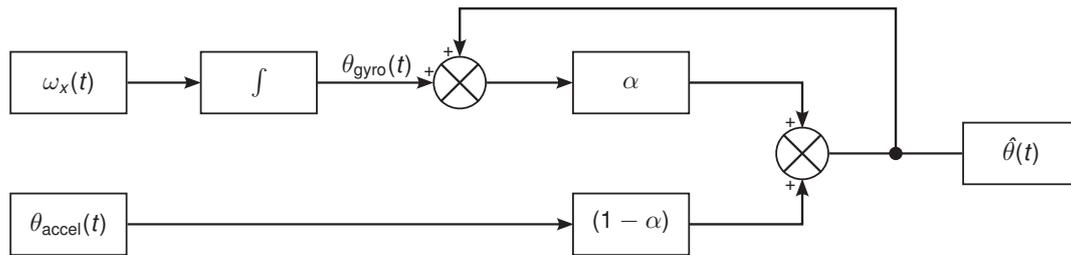


Fig. 2. Diagrama en bloques del filtro complementario para cálculo de roll (θ)

B. Filtro Complementario

Una de las opciones principales para mitigar las limitaciones de los sensores MEMS es usar fusión sensorial, generalmente usando *Filtros de Kalman* o *Filtros Complementarios* [4].

El Filtro de Kalman es una versión lineal de unos algoritmos secuenciales llamados *Filtros bayesianos*. Estos filtros pueden estimar algún estado de un sistema, suponiendo conocido el modelo del mismo, de cada sensor y el ruido que los afecta. Este tipo de filtros es muy eficiente para eliminar el ruido del sistema a la vez que fusiona la información provista por cada sensor. Pero necesita realizar recursivamente ecuaciones con matrices que pueden generar una carga excesiva en microcontroladores de bajo costo.

En cambio, el Filtro Complementario es un método de fusión de señales utilizado para combinar la información de diferentes fuentes, aprovechando las ventajas de cada una de ellas en diferentes bandas de frecuencia. Este tipo de filtro es útil cuando se tienen dos señales que contienen información sobre un mismo estado, pero con diferentes tipos de ruido o incertidumbres, y se desea estimar dicho estado de forma precisa. Es especialmente útil cuando una de las señales es más confiable en bajas frecuencias, mientras que la otra es más confiable en altas frecuencias.

Existen muchas variantes de filtros complementarios de diversa complejidad [5], [6], sin embargo, la idea principal del filtro complementario es aplicar un filtro paso bajo a una de las señales y un filtro paso alto complementario a la otra, de modo que la combinación de ambas cubra todo el espectro de frecuencia relevante para la estimación del estado. En términos generales, un filtro complementario puede expresarse como

$$\hat{x}(t) = \alpha \cdot x_1(t) + (1 - \alpha) \cdot x_2(t) \quad (5)$$

donde $x_1(t)$ es la señal más confiable en altas frecuencias (filtrada con un filtro paso altos), $x_2(t)$ es la señal más confiable en bajas frecuencias (filtrada con un filtro paso bajos), α es un factor de ponderación que determina cómo se combinan ambas señales, con $0 \leq \alpha \leq 1$. En este caso, $\hat{x}(t)$ es la estimación del estado buscado en el instante t . Este esquema es la versión más sencilla de filtro complementario y permite mejorar la estimación de un estado al usar las características complementarias de las señales disponibles.

Para determinar la orientación se puede implementar un filtro complementario que combine los datos del acelerómetro y del giróscopo, mejorando la precisión de la estimación de la inclinación. El filtro complementario permite hacer énfasis en las fortalezas de ambos sensores: la capacidad del acelerómetro para proporcionar una referencia a largo plazo (basada en la detección de la orientación de la gravedad) y la capacidad del giróscopo para detectar cambios rápidos en la orientación a corto plazo. Una manera más empírica de verlo es que al combinar estas lecturas usando factores complementarios realiza lo que se podría definir como un promedio ponderado.

Entonces, para estimar el ángulo de inclinación $\hat{\theta}$ para el instante t se hace

$$\hat{\theta}(t) = \alpha \cdot \theta_{gyro}(t) + (1 - \alpha) \cdot \theta_{acc}(t) \quad (6)$$

donde $\theta_{acc}(t)$ es el ángulo estimado a partir de la aceleración en el tiempo t y α es un factor de ponderación que determina cuánto se confía en las mediciones del giróscopo frente a las del acelerómetro, generalmente en el rango de $0 \leq \alpha \leq 1$. Un valor alto de α (más cerca de 1) da más peso a las mediciones del giróscopo, mientras que un valor bajo da más peso a las mediciones del acelerómetro. Este valor se ajusta experimentalmente.

En la Fig. 2 se muestra un esquema del filtro complementario propuesto.

C. Detección de Vibraciones

La detección de vibraciones se puede realizar utilizando las lecturas del acelerómetro. Las vibraciones se manifiestan como cambios rápidos en las lecturas de aceleración en los tres ejes (x , y , z). La magnitud de la vibración indicará la intensidad de la vibración.

La magnitud de la vibración se puede calcular utilizando la norma euclidiana de las lecturas de aceleración en los tres ejes, lo que proporciona un valor escalar que indica la intensidad de la vibración. Entonces,

$$V = \sqrt{a_x^2 + a_y^2 + a_z^2} \quad (7)$$

donde a_x , a_y , a_z son las componentes de la aceleración en los ejes x , y y z , respectivamente, y V es la magnitud de la vibración.

Para eliminar el ruido de alta frecuencia y mejorar la precisión en la detección de vibraciones, se debe implementar

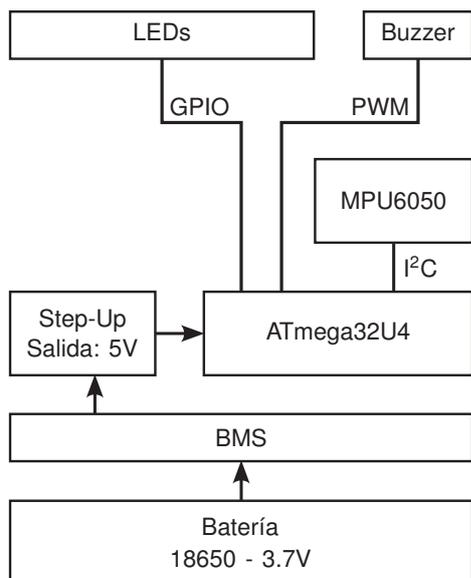


Fig. 3. Diagrama en bloques del sistema completo

un filtro paso bajo a lecturas del acelerómetro. Este filtrado suaviza las señales, permitiendo una mejor identificación de las vibraciones. El filtro pasa bajos de primer orden se puede escribir como

$$q_k = \beta \cdot q_{k-1} + (1 - \beta) \cdot p_k \quad (8)$$

donde p_k es la entrada al filtro, esto es la medición del acelerómetro en el instante de tiempo k . La salida del filtro en la medición anterior es q_{k-1} y β es el coeficiente de suavizado, con $0 \leq \beta \leq 1$. Finalmente, q_k es la salida del filtro para la medición actual. El valor de β se ajusta según la frecuencia de corte deseada y las características del ruido presente en la señal.

Aún con la aplicación del filtro pasa bajos, se debería incluir un umbral para detectar eventos de vibración significativos. Un pico indica un cambio repentino en la aceleración, lo que puede corresponder a una vibración fuerte o un impacto. Estos picos se pueden detectar comparando la magnitud de la vibración filtrada con un umbral predefinido. Si la magnitud supera este umbral, se considera que ha ocurrido un evento de vibración.

III. IMPLEMENTACIÓN

El sensor de inclinación y vibración fue implementado utilizando una placa de desarrollo Arduino Pro Micro, que ofrece una plataforma compacta y de bajo consumo. Esta placa tiene un microcontrolador ATmega32U4, que cuenta con 32kB de memoria flash y una frecuencia de reloj de 16MHz, proporcionando un equilibrio entre rendimiento y eficiencia energética.

Para las lecturas de aceleración y velocidad angular se utilizaron los acelerómetros y giróscopos que se encuentran en el circuito integrado MPU6050 de *InvenSense Inc*. Se eligió el módulo de desarrollo GY-521 que cuenta con

los periféricos necesarios para utilizar el MPU6050 con los niveles de tensión del módulo Pro Micro. La comunicación entre el módulo los sensores y el microcontrolador se realiza mediante el bus I²C disponible en ambos. Para proveer de energía a todo el sistema se incluyó una batería Li-Ion 18650 con un BMS genérico para una celda y una fuente *step-up*. Con esta configuración, se calcula que el sistema puede funcionar por más de 24 horas.

Para dar referencia de los valores estimados, el sistema tiene señalización auditiva y lumínica. Mediante cuatro LEDs y un buzzer pasivo se provee gran parte de la información disponible. Un LED verde se enciende cuando se energiza el sistema mientras que el segundo LED verde titila una vez por segundo mientras el sistema esté en funcionamiento haciendo la detección. El LED rojo comienza a titilar cuando se detecta un cambio en la inclinación a una frecuencia que será proporcional al cambio. Un LED amarillo se enciende cuando se detecta vibraciones por encima de un umbral determinado. El buzzer suena acompañando a cualquiera de los dos LEDs. En la Fig. 3 se muestra el esquema completo mientras que en la Fig. 4 se muestra la distribución interna de los módulos.

A. Forma de Uso y Experiencia en Campo

Para utilizar el sensor, hay que fijarlo a la estructura a monitorear. Para esto, se cuenta con un orificio en el centro del dispositivo el cual fue diseñado para su uso con tornillos tirafondo estándar. Se puede usar directamente sobre un puntal de madera o una pared.

Una vez fijo, se debe encender el dispositivo con el *switch* lateral. El durante el primer segundo el dispositivo solo espera sin hacer ninguna estimación. Este momento está pensado para que luego de la activación, el rescatista suelte completamente el dispositivo y permanezca inmóvil para no afectar la



Fig. 4. Distribución interna de los módulos



Fig. 5. Prototipo activo funcionando en el laboratorio

estimación inicial. Durante el siguiente segundo, el dispositivo determina la orientación actual, y la configura como *cero*. Será este el valor considerado como *de referencia*. A partir del tercer segundo, el sistema comienza a funcionar, dando una única alerta mediante un breve sonido y un encendido momentáneo de todos los LEDs para corroborar que el sistema está funcionando correctamente. La carcasa del dispositivo fue implementado usando tecnología de modelado por deposición fundida (FDM) usando filamentos de PLA. En la Fig. 5 se puede ver una imagen del prototipo funcionando en el laboratorio, fijo a una estructura de madera simulando un apuntalamiento.

El prototipo fue utilizado en un simulacro llevado adelante en el 8vo Curso Internacional de Intervención en Desastres, realizado en la Ciudad de Centenario, provincia de Neuquén

en el mes de agosto de 2024. Tuvo una recepción positiva por parte de los rescatistas quienes validaron tanto su forma, tamaño y utilidad, debido principalmente a su sensibilidad. Además, a pesar de los cálculos realizados, el sistema estuvo activo 4 días con la misma carga, debido a que en gran medida el dispositivo solo procesaba las señales y no emitía alertas ya que las estructuras no sufrían modificaciones.

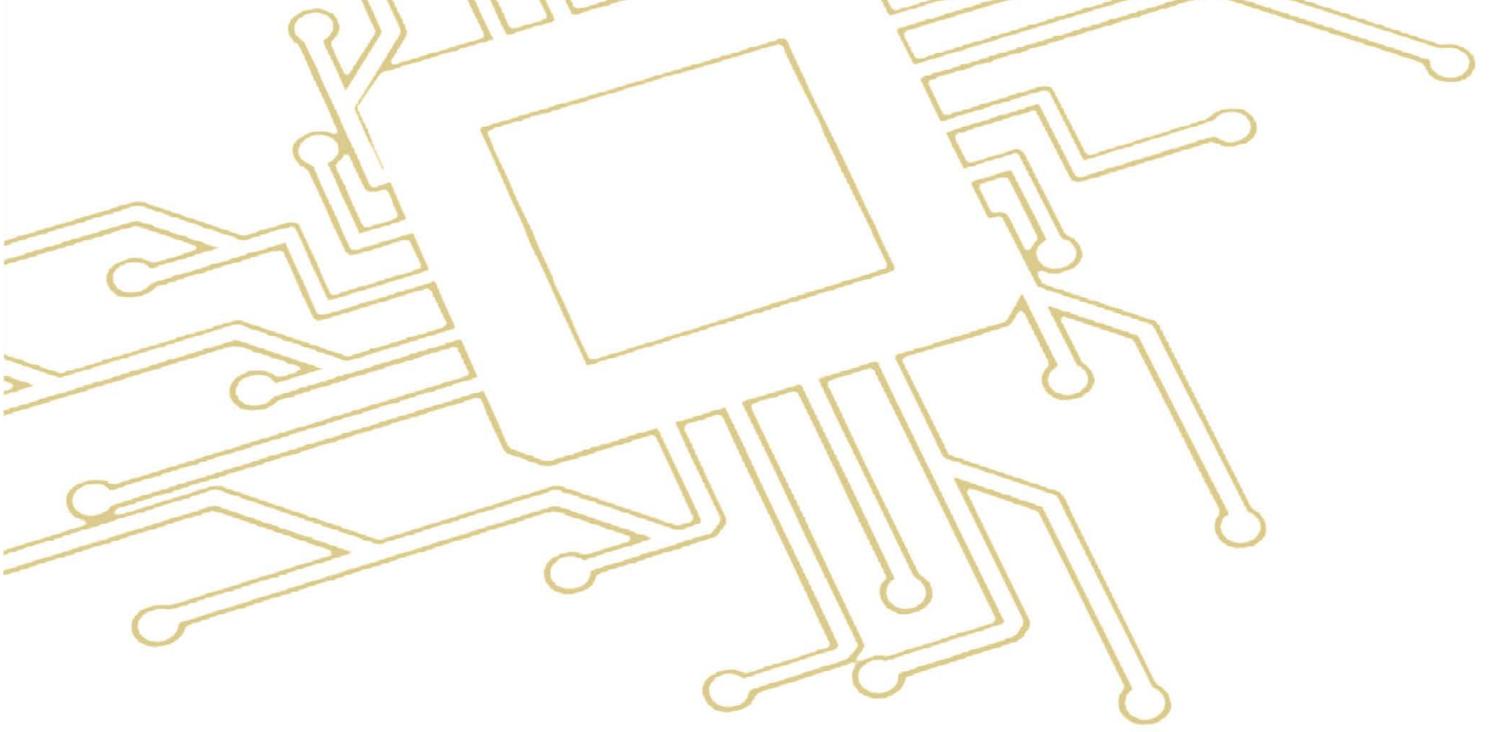
IV. DISCUSIÓN Y TRABAJOS FUTUROS

Este trabajo presenta el primer prototipo de un sensor de inclinación y vibración para monitoreo de apuntalamientos en estructuras colapsadas. Se presentó la teoría subyacente para la estimación de inclinación y detección de vibración como así también una detallada descripción de la implementación del sistema, el cual ya está en uso y continúa su desarrollo. En las pruebas realizadas en la simulación de desastres llevada adelante por rescatistas, el prototipo mostró buen desempeño en la tarea de monitorear los apuntalamientos bajo control.

Se prevé en la siguiente etapa realizar una validación experimental de la estimación de inclinación comparando diferentes versiones de Filtros Complementarios y Filtros de Kalman, incluyendo comparaciones con diferentes arquitecturas de bajo consumo, como el Cortex-M3.

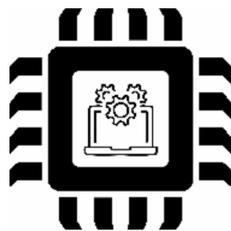
REFERENCES

- [1] A. A. S. Shaon, S. Bhowmik, B. K. Bhawmick, P. Das, and N. K. Das, "Design and implementation of a self-balancing robot," in *Proc. Int. Conf. on Mechanical Eng. And Renewable Energy*, 2017.
- [2] M. A. Pucheta, C. J. Paz, and M. E. Pereyra, "Representaciones cinemáticas de orientación y ecuaciones de estimación," in *Mecánica Computacional Vol XXXIII, Proc. del XXI Congreso sobre Métodos Numéricos y sus Aplicaciones*, Setiembre 2014, pp. 2303–2324.
- [3] C. J. Paz, *Estimación eficiente de la posición y orientación de un vehículo en tiempo real por medio de sensores de bajo costo*. Ph.D. Thesis. UTN-FRC, 2016.
- [4] G. P. Paina, D. Gaydou, J. Redolfi, C. Paz, and L. Canali, "Experimental comparison of kalman and complementary filter for attitude estimation," in *Jornadas Argentinas de Informática*. Córdoba, Argentina: AST, Argentine Symposium on Technology, 2011, pp. 205–215.
- [5] P. Narkhede, S. Poddar, R. Walambe, G. Ghinea, and K. Kotecha, "Cascaded complementary filter architecture for sensor fusion in attitude estimation," *Sensors*, vol. 21, no. 6, p. 1937, 2021.
- [6] L. Ryzhkov, "Complementary filter design for attitude determination," in *2018 IEEE 5th International Conference on Methods and Systems of Navigation and Motion Control (MSNMC)*. IEEE, 2018, pp. 214–217.



Eje Temático

Aplicaciones en Software



telBot, una biblioteca para la implementación de clientes de mensajería Telegram

E. Sergio Burgos

Facultad Regional Paraná

Universidad Tecnológica Nacional

Paraná, Entre Ríos, Argentina

sergioburgos@frp.utn.edu.ar

Resumen—En este trabajo se presentan antecedentes del uso de clientes de mensajería como interfaces de comunicación con sistemas embebidos para contextualizar las posibilidades que brindan. Se propone una biblioteca en lenguaje C para la implementación de clientes de mensajería orientada al control de dispositivos e instrumentación utilizando el *framework* de la empresa Espressif y la familia de dispositivos ESP32.

Palabras clave—ESP32, Telegram, chat bot, monitoreo, control

I. INTRODUCCIÓN

En el momento tecnológico en el que nos encontramos, diferentes factores han confluído permitiendo facilitar la interacción entre procesos, aplicaciones y dispositivos. La aparición en el mercado de módulos de bajo costo tales como los dispositivos ESP8266 y ESP32, que permiten establecer comunicación con redes WiFi, incorporando procesadores con abundantes recursos de memoria y la proliferación de aplicaciones de mensajería han facilitado el control y monitoreo de variables de diferente índole.

Entre las aplicaciones de mensajería, dos muy populares son Whatsapp y Telegram, ambas presentando la posibilidad de implementar aplicaciones (bots) con las que los usuarios pueden interactuar. En el caso de Telegram, se distingue por permitir la creación de cuentas para ser utilizadas por estas aplicaciones de modo automatizado y gratuito.

Existen diferentes antecedentes de uso de Telegram como sistema de intercambio de mensajes en sistemas embebidos. Uno de ellos [1] no solo hace uso de la red a fin de establecer comunicación entre el usuario y un dispositivo sino que además remite fotos como evidencia del estado de ocupación de una habitación, haciendo el procesamiento requerido a través de una computadora Raspberry PI. En [2] se utiliza una idea similar a fin de vigilar niños, pero en este caso se utiliza la plataforma Arduino junto a un módulo ESP8266. Casos como este, donde los recursos de procesamiento son relativamente limitados, son de interés en este trabajo.

Más específicamente, en la implementación presentada en [3] se demuestran formas en las que se po-

dría interactuar con dispositivos a través de redes de mensajería a fin de obtener el valor de variables físicas y en [4] además se agrega la posibilidad de controlar dispositivos.

En este trabajo se toman las ideas anteriores proponiendo una biblioteca que busca abstraer el problema de la implementación de la comunicación con el sistema de mensajería Telegram para dispositivos ESP32 y sus variantes, utilizando comandos, realizando la implementación en lenguaje C y Visual Studio Code con PlatformIO como entorno de desarrollo.

II. SOBRE EL SISTEMA DE MENSAJERÍA

La estructura del protocolo está ampliamente documentada [5] y de allí se extraen algunas ideas sobre su funcionamiento que se describen a continuación.

El sistema de mensajería de la plataforma Telegram utiliza clientes (aplicaciones web o móviles) y servidores a fin de permitir la propagación de mensajes entre usuarios. Cada usuario es identificado dentro del protocolo de comunicación a través de un número entero que es incluido como un elemento dentro del mensaje.

Los mensajes pueden tener diferente contenido (imágenes, sonidos, archivos, etc.) y ser de diferente tipo. Además de intercambiar texto entre clientes es posible definir comandos cuando uno actúa en la comunicación es una aplicación. Estos comandos, una vez definidos, son propuestos a quién desea comunicarse. Si bien es un tipo de mensaje específico del conjunto de alternativas disponibles, la estructura de la información asociada es similar a los demás. La información intercambiada entre clientes y servidores es enviada utilizando el protocolo HTTPS (Hyper Text Transfer Protocol/Protocolo de Transferencia de Hipertexto encriptado a través de SSL o TLS) y los mensajes junto a sus atributos son codificados utilizando JSON (Java Script Object Notation) [6]. Luego, las diferentes operaciones que se desean realizar sobre el sistema de mensajería son indicadas a través de URLs utilizando el método de petición GET o POST (propios del protocolo HTTP), la petición POST permite enviar las cadenas JSON requeridas para el envío de mensajes.



Se prevén dos maneras para que los clientes conozcan si se ha remitido un mensaje hacia ellos, a saber:

- Realizando encuestas periódicas al servidor de mensajería.
- Utilizando *webhooks*. En este modo de trabajo, la implementación del bot debe estar asociada a un servidor web que, luego de vincularse con los servidores de mensajería enviando solicitudes HTTPS, recibirá notificaciones del mismo modo.

El uso de *webhooks* permite optimizar el procesamiento de los mensajes pero requiere de un servidor web que sea accesible desde los servidores de Telegram a fin de poder recibir actualizaciones. Esto presenta una limitante para el caso interés ya que requiere medidas adicionales de seguridad para la implementación del sistema embebido por estar accesible desde una dirección pública o requerir dispositivos adicionales que resuelvan la seguridad.

Al utilizar encuestas periódicas al servidor a fin de obtener notificaciones de nuevos mensajes se enfrenta el problema que, el tiempo entre consultas a los servidores de mensajería fijará la demora en la respuesta del sistema, lo cuál puede ser una limitante para dispositivos que deban cuidar el consumo de energía y tener rápidas respuestas. No obstante, en este trabajo se optó por esta alternativa. También es importante considerar que los mensajes serán mantenidos en el servidor de mensajería por 24 horas, luego de lo cuál, sino son procesados serán eliminados de forma automática.

La obtención de notificaciones se realiza a través del método *getUpdates*, el cuál utiliza diferentes parámetros a fin de establecer su comportamiento y retorna objetos de la clase *update* que describen diferentes aspectos del mensaje. La implementación propuesta utiliza dos argumentos, a saber

- *limit*: permite establecer la cantidad de objetos *update* que se desea recibir y que por defecto es 100.
- *offset*: identificador del primer objeto *update* a ser retornado, indicado como un número entero.

Los mensajes son marcados como leídos en el servidor cuando se solicita un objeto *update* con su valor de *offset*, este es un valor entero que identifica la entidad. La implementación propuesta obtiene los mensajes uno por uno (*limit* = 1) y, a partir de obtener un primer objeto *update* utiliza su identificación (llamada *update_id*) para obtener los mensajes siguientes. De este modo, en cada petición de actualización es posible acceder a todos los mensajes recibidos en las últimas 24 horas que no han sido procesados aún.

Cada objeto *update*, representado como una entidad JSON, cuenta con múltiples campos opcionales, que describen características particulares. Estos también son objetos, siendo significativos para esta implementación la representación del mensaje (atributo *message*) y, en ese objeto, el identificador del chat al que pertenece

(campo *id*). Se utiliza este último valor a fin de identificar el destinatario del mensaje de respuesta.

III. DETALLES DE IMPLEMENTACIÓN

De lo expuesto puede concluirse que la implementación de un bot de mensajería tiene dos requerimientos:

- Realizar peticiones HTTPS: a diferencia de las peticiones HTTP, estas utilizan un certificado (archivo de texto) que permite la comunicación segura con el servidor, siendo necesario incorporarlo en la implementación. Este archivo es utilizado para cifrar la información que se intercambia en la comunicación, impidiendo que terceros puedan acceder a su contenido o modificarla.
- Procesar objetos JSON: si bien se trata de cadenas de caracteres que incorporan la descripción de entidades, al utilizar composición entre diferentes objetos a fin lograr objetos de más alto nivel, su interpretación puede requerir de algoritmos complejos y consumir recursos de procesamiento significativos.

La realización de peticiones HTTPS está provista en el *framework* proporcionado por el fabricante de los dispositivos ESP32 [7], por lo que solo debe preverse la incorporación de un archivo de certificado válido como parte del proyecto. Este es uno de los motivos por los que se ha optado por este dispositivo. Debe considerarse que los certificados tienen períodos de validez, por lo que la fecha de caducidad del mismo debe ser considerada. En caso de no contar con un certificado válido, la comunicación no podrá ser realizada.

El procesamiento de cadenas de caracteres conteniendo la descripción de objetos JSON se realizó a través de la biblioteca *microjson* [8]. Esta tiene como característica distintiva no requerir realizar alojamiento dinámico de memoria (no utiliza la función *malloc*) y no consume una cantidad de recursos significativos ya que ha sido diseñada para ser utilizada en entornos con recursos reducidos.

Para procesar una cadena de caracteres JSON, la biblioteca *microjson* requiere que previamente se realice una descripción de las propiedades del objeto utilizando estructuras inicializadas en lenguaje C. La inicialización establece las variables donde se realizará el alojamiento del valor de los campos. Los diferentes valores de cada atributo del objeto son almacenados en variables globales que deben definirse previos a la inicialización de la estructura. Luego el procesamiento de cada cadena se realiza a través de invocaciones a la función *json_read_object*.

Debido a la forma de definición y límites de los valores enteros utilizados por la biblioteca *microjson* y el tamaño de almacenamiento de los identificadores de chat y usuario utilizados por el sistema de mensajería (ambos de tipo entero), ha sido necesario definir estas propiedades

como valores reales de doble precisión a fin de soportar las características de almacenamiento requeridas.

IV. ESTRUCTURA DE LA BIBLIOTECA TELBOT

La biblioteca está conformada por cinco archivos, *config.h* en el cuál se establece la configuración de funcionamiento a través de la definición de constantes simbólicas, los archivos *telBot.h* y *telBot.c* conteniendo las implementaciones específicas relacionadas con el procesamiento del sistema de mensajería y finalmente los archivos *commands.h* y *commands.c* con la definición de los comandos que serán considerados.

Dentro de las opciones de configuración debe indicarse la cadena de identificación del bot. Este valor, equivalente a la identificación de un usuario en el sistema de mensajería, es un argumento utilizado en cada una de las peticiones realizadas al servidor. Debe ser obtenido a través de un cliente de Telegram solicitando la creación de un nuevo bot a un sistema automático (bot) llamado *BotFather*.

Debido a que se utiliza alojamiento estático para los mensajes, también es necesario indicar los límites de almacenamiento para las cadenas de caracteres que los contendrán.

Dos parámetros adicionales deben ser indicados y que tienen directa relación con el comportamiento de la librería, a saber:

- **USE_DEFAULT_RESPONSE:** la librería ha sido desarrollada a fin de ejecutar comandos en el microcontrolador al ser recibidos como mensajes. Pero es posible que reciba valores no se correspondan con los comandos esperados o definidos. Ante esta situación es posible optar por dos alternativas, o utilizar un mensaje de respuesta predeterminado definiendo esta constante o ignorarlo no generando una respuesta.
- **REGISTER_COMMANDS:** los comandos que se deseen utilizar pueden configurarse a través del bot *BotFather*, pero también es posible indicarlos desde la librería. Si se define esta constante simbólica, al inicializar la biblioteca, los comandos registrados serán configurados como válidos en la plataforma.

La interacción con la biblioteca se realiza a través de un par de funciones, siendo ellas las responsables de realizar el procesamiento de los mensajes en el servidor y responder de modo adecuado. La función *startBotTask* toma como argumento el intervalo de tiempo (expresado en mili segundos) al cabo del cuál se realizarán las consultas al servidor en busca de nuevos mensajes. Esta función crea una tarea estática de *FreeRTOS* que comprobará la disponibilidad de nuevos mensajes en el servidor obrando en consecuencia e invocando a los comandos (funciones) que corresponda.

A fin de poder enviar respuestas a los mensajes recibidos se incorpora la función *sendMessage*, la cuál

recibe como argumento el mensaje a enviar y la cantidad de intentos de envío que se desea realizar, retornando verdadero o falso a fin de indicar el éxito de la operación. La cadena que representa el mensaje debe contener una representación de un objeto JSON conteniendo los atributos *chat_id* (identificación del destinatario) y *text* (el mensaje propiamente dicho).

Los comandos que serán reconocidos por el bot son definidos en el archivo *commands.c*, realizándose aquí las implementaciones requeridas para cada aplicación. Cada comando es identificado por una estructura en lenguaje C conteniendo tres campos, una cadena de caracteres que contiene la identificación del comando propiamente dicho, un puntero a función que será utilizado invocar a la función asociada al comando cuando sea recibido y la descripción de la tarea realizada. Este último campo, en caso de configurar la opción de uso de respuesta predeterminada, será utilizado para enviar un mensaje descriptivo de los comandos válidos ante un comando no reconocido.

La cadena de identificación del comando debe iniciar con el carácter */*, y el puntero a función utilizado debe tener la siguiente definición:

```
typedef void (*pFcnDoAction)(int, char*[], uint64_t);
```

Los dos primeros argumentos corresponden con la cantidad palabras recibidas y su valor, con un concepto similar al de los argumentos *argc* y *argv* de la función *main* en las aplicaciones desarrolladas en lenguaje C. El último argumento corresponde con la identificación del chat de donde proviene el mensaje. Con este valor y utilizando la función *sendMessage* provista por la librería es posible contestar al remitente de un mensaje.

Todos los comandos válidos deben definirse como elementos de un arreglo llamado *commands* cuya definición se encuentra en el archivo *commands.c*. Por cada comando se tendrá una función cuyo prototipo deberá realizarse en *commands.h* y su implementación en *commands.c*. El último elemento del arreglo debe indicarse con el valor cero.

Con estas implementaciones realizadas, cada vez que se reciba un nuevo mensaje, la biblioteca procesará el texto del mismo identificando sus partes. Si se encuentra un comando cuyo nombre tenga coincidencia con la primer palabra del mensaje se invocará a la función asociada, proporcionándole los argumentos adecuados.

V. PRUEBAS REALIZADAS

Una versión preliminar de la biblioteca fue utilizada en la implementación de un sistema de monitoreo de temperatura utilizado en el Centro de Comunicaciones de la Facultad Regional Paraná. El sistema funciona actualmente y fue instalado en diciembre de 2023.

El desarrollo utiliza un sensor DS18B20 [9] a fin de registrar la temperatura ambiente a intervalos regulares de tiempo (5 segundos). Este sensor es de tipo digital

y utiliza el protocolo 1-Wire a fin de establecer comunicación con el microcontrolador. La conexión entre ambos se realizó utilizando 3 conexiones, dos para energizar el dispositivo mientras que la tercera para establecer la comunicación bidireccional. Se prefirió este dispositivo por ser de tipo digital, no requerir de calibración y contar con una presentación tipo "sonda", permitiendo mantener el sensor de temperatura protegido y seleccionar el punto donde realizar las mediciones de modo simple.

La comunicación con el sensor de temperatura se implementó utilizando la biblioteca esp32-owb [10], la cuál utiliza el módulo RMT (Remote Control Transceiver) presente en el dispositivo ESP32 para establecer la comunicación.

El valor de temperatura registrado puede ser consultado a través de un comando enviado al sistema a través del sistema de mensajería. En esta implementación también se agregó la posibilidad de registrar alarmas, para esto el bot permite agregar el valor de identificación del chat desde el que se envía el mensaje y en caso de que la temperatura supere límites superiores o inferiores envía mensajes reportando la situación. Utilizando comandos también es posible establecer los valores máximos y mínimos de referencia.

El registro de los identificadores de chat a los que deben enviarse las notificaciones se almacenaron en la memoria flash del dispositivo, accediendo a la misma como una partición SPIFFS [11].

A modo de prueba de concepto, la versión aquí presentada ha sido probada con dos aplicaciones simples que proporcionan ejemplos de su uso. En un caso se utiliza un ESP32 con 4 comandos que responden un mensaje alusivo a su nombre al ser invocados. Una segunda implementación se realizó utilizando un ESP32-C6-DevKit1 [12] a fin de controlar el color e intensidad del led RGB presente en la placa. En la figura 1 puede observarse la interacción a través del cliente de mensajería para teléfonos móviles en el sistema operativo Android.

Ambas aplicaciones, junto al código fuente de la biblioteca, se encuentran disponibles en el repositorio del proyecto [13]. En ambos ejemplos la configuración requerida para el acceso a la red WiFi ha sido incluida en el archivo de configuración, no haciéndose énfasis en otras alternativas más prácticas en favor de mantener la simplicidad de los ejemplos.

VI. CONCLUSIONES

VI-A. Plataforma de mensajería como interfaz de comunicación

Una de las conclusiones a las que se arribó a través de las pruebas realizadas con el sistema de monitoreo de temperatura es que el uso de sistemas de mensajería permite características de monitoreo y control similares a las requeridas en aplicaciones de sistemas embebidos que, en otras circunstancias, motivan la implementación

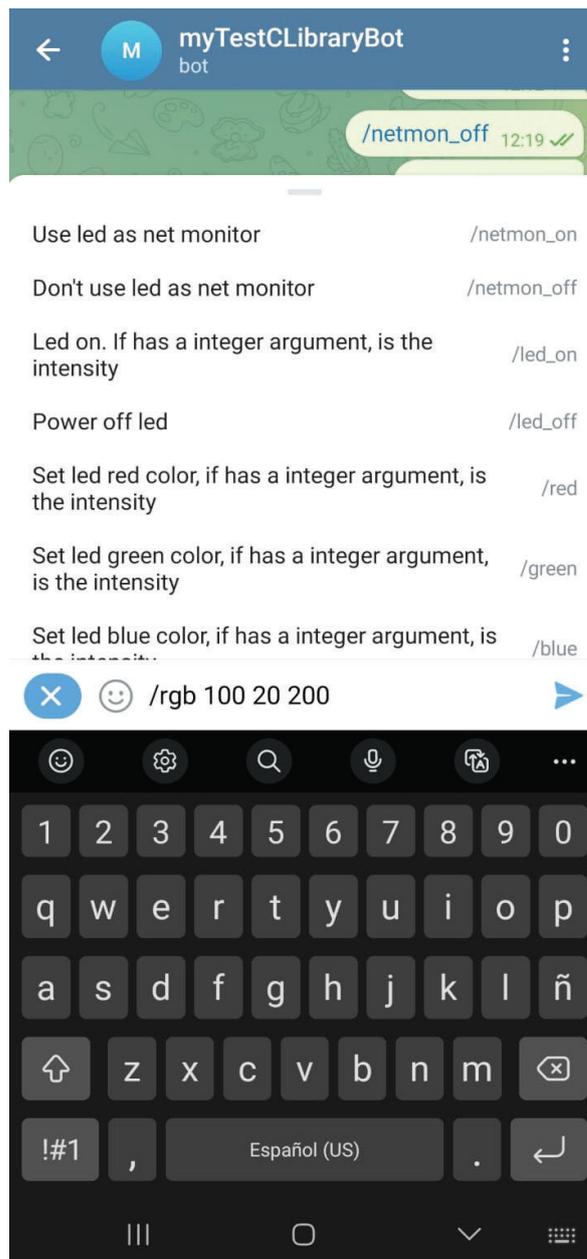


Figura 1. Interacción con el ejemplo de aplicación para ESP32-C6. Puede observarse el conjunto de comandos registrados así como la interacción con un comando que incluye parámetros.

de aplicaciones móviles. En este trabajo y a partir de la librería desarrollada solo se consideró el uso de comandos, pero esto es solo una de las alternativas existentes en la plataforma. Es posible con métodos similares agregar definiciones de botones que permiten registrar acciones de modo gráfico, sin necesidad de escribir un comando particular.

Procesos de autenticación que habiliten la interacción con el bot no han sido cubiertos en este trabajo pero puede resultar de importancia en caso realizar tareas de

control. Una versión básica de este tipo validación podría realizarse agregando la opción de iniciar una sesión a través de un comando que reciba como argumento una clave de acceso. Luego, al recibir un comando de acción, puede ser ejecutado si se cuenta con una sesión iniciada para un identificador de chat específico. Esta implementación podría realizarse con la biblioteca propuesta ya que la clave de acceso en cuestión puede agregarse como un argumento de un comando. Esta propuesta también cubre el caso de acceso múltiple, al registrarse una sesión iniciada desde un cliente puede bloquearse el acceso desde otros a fin de evitar la ejecución de acciones contradictorias.

Desde un punto de vista más general, la posibilidad de acceder a diferentes recursos a través de mensajes abre un conjunto interesante de posibilidades al poder utilizar un sistema embebido como origen o destinatario de mensajes, archivos de audio, imágenes o videos.

VI-B. Biblioteca telBot

La implementación cumple con el objetivo que motivó el desarrollo, contar con una implementación que realice el procesamiento de mensajes de forma independiente respecto de la definición de los comandos. Este hecho brinda la posibilidad de incorporar la librería a implementaciones existentes sin requerir cambios significativos. No obstante, no implementa el procesamiento de muchos campos opcionales, definidos en el protocolo de comunicación de Telegram.

En lo que respecta al consumo de recursos de memoria RAM depende fundamentalmente del tamaño máximo definido para los espacios de almacenamiento asociados a los mensajes recibidos desde el servidor de mensajería y de la extensión del mensaje. Se tiene un consumo de recursos adicional al activar la opción de registro de los comandos en el servidor de mensajería. Esto es así debido a que este proceso debe realizarse a través de un mensaje que incluye las palabras utilizadas

como comandos así como su descripción. Por otro lado, si se utiliza la opción de registro de los comandos en el servidor, debe verificarse que la definición utilizada para el espacio de almacenamiento del mensaje sea adecuada considerando las longitudes de los comandos y sus descripciones.

REFERENCIAS

- [1] I. M. Hazri, W. Z. W. Ismail, I. Ismail, S. A. Rahman, F. S. Hussin *et al.*, "Automated motion detection security system notifier using raspberry pi with telegram," en *2020 IEEE Symposium on Industrial Electronics & Applications (ISIEA)*. IEEE, 2020, pp. 1–6.
- [2] M. D. Akbari y R. Roestam, "Arduino baby monitoring with telegram bot and wifi connection," *IT for Society*, vol. 4, no. 2, 2020.
- [3] D. N. Bestari y A. Wibowo, "An iot-based real-time weather monitoring system using telegram bot and thingsboard platform." *International Journal of Interactive Mobile Technologies*, vol. 17, no. 6, 2023.
- [4] D. K. B. A. Othman, C. A. Mai, y A. Y. Ibrahim, "Preagriot: Precision agriculture iot monitoring and controlling system," *MALAYSIAN JOURNAL OF INNOVATION IN ENGINEERING AND APPLIED SOCIAL SCIENCES (MYJIEAS)*, vol. 2, no. 01, pp. 37–45, 2022.
- [5] telegram.org. (2024) Bots: An introduction for developers. [Online]. Disponible en: <https://core.telegram.org/bots>
- [6] T. Bray, "The JavaScript Object Notation (JSON) Data Interchange Format," RFC 8259, Dec. 2017. [Online]. Disponible en: <https://www.rfc-editor.org/info/rfc8259>
- [7] E. Systems. (2024) Esp http client. [Online]. Disponible en: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/protocols/esp_http_client.html
- [8] E. S. Raymond. (2024) microjson project. [Online]. Disponible en: <https://gitlab.com/esr/microjson>
- [9] A. Devices. (2024) Ds18b20, programmable resolution 1-wire digital thermometer. [Online]. Disponible en: <https://www.analog.com/en/products/ds18b20.html>
- [10] D. Antliff. (2023) esp32-owb home page. [Online]. Disponible en: <https://github.com/DavidAntliff/esp32-owb>
- [11] E. Systems. (2024) Spiffs filesystem. [Online]. Disponible en: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/storage/spiffs.html>
- [12] —. (2024) Esp32-c6-devkitc-1 v1.1. [Online]. Disponible en: https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32c6/esp32-c6-devkitc-1/user_guide_v1.1.html
- [13] E. S. Burgos. (2024) Repositorio de la biblioteca telbot. [Online]. Disponible en: <https://gitlab.com/labinformatica/telbot>



Implementación de Modelos de Software en la Formación de Sistemas Embebidos: Un enfoque Metodológico

Roberto M. Murdocca, Sergio F. Hernandez Velazquez, Claudio Caballero
Departamento de Electrónica
Universidad Nacional de San Luis
San Luis, Argentina
e-mail: {mmurdocc, sergio.sfhv, cnc.cabal}@gmail.com

Resumen - Este trabajo presenta una estrategia de diseño para sistemas embebidos que optimiza el proceso de desarrollo en las clases prácticas de laboratorio. Mediante el empleo de máquinas de estado finito, se crea un modelo del sistema que facilita la simulación y la detección temprana de errores. Esta aproximación reduce significativamente el tiempo de diseño garantizando un desarrollo más rápido, de mayor calidad y confiabilidad. Esta estrategia se incorporó hace unos años en la materia Procesadores II.

Palabras clave: Sistemas embebidos, Máquinas de estado finito, Modelado, Simulación, Verificación.

I. INTRODUCCIÓN

La presente investigación propone una metodología de diseño para sistemas embebidos que prioriza la modelización, simulación y verificación antes de la implementación. El enfoque se centra en el uso de Máquinas de Estados Finitos como herramienta para representar el comportamiento del sistema. Inicialmente, se desarrolla un modelo del sistema utilizando esta representación, lo que permite visualizar de forma clara y concisa los estados, transiciones y eventos que lo gobiernan.

Posteriormente, se lleva a cabo una simulación del modelo para validar su funcionamiento y detectar posibles errores o incoherencias en la lógica del diseño. Esta etapa es crucial para garantizar que el sistema se comporte como se espera antes de invertir recursos en su implementación física.

Finalmente, se procede a la codificación del sistema, utilizando el esqueleto de código generado por la herramienta de modelado y simulación. Esta metodología permite reducir significativamente el tiempo de desarrollo y los costos asociados a la detección y corrección de errores en etapas posteriores del proyecto.

II. DESCRIPCIÓN

En el diseño de sistemas embebidos, la complejidad inherente a la integración de hardware y software exige enfoques metodológicos que garanticen la confiabilidad y eficiencia del sistema final. En ciertos sistemas de mediana a alta complejidad, no alcanza con utilizar lenguajes de alto nivel y HAL (Hardware Abstraction Layers). Un enfoque ampliamente reconocido es el uso de modelos de software como las Máquinas de Estado Finito (FSM, por sus siglas en inglés) [1][2] durante las etapas tempranas del desarrollo. Estos modelos ofrecen una representación abstracta y formal del comportamiento del sistema, permitiendo a los diseñadores conceptualizar, simular y validar la lógica de funcionamiento antes de proceder con la implementación del código. El empleo de máquinas de estados finitos facilita la comprensión de las interacciones complejas y las transiciones entre diferentes estados del sistema, proporcionando un marco estructurado para capturar el comportamiento reactivo y secuencial característico de muchos sistemas embebidos.

Además, la simulación basada en FSM permite identificar errores lógicos y condiciones límite que pueden pasar desapercibidos en etapas posteriores del desarrollo, reduciendo

significativamente el riesgo de fallos en la fase de integración y despliegue.

A medida que los sistemas embebidos se vuelven más sofisticados y críticos para aplicaciones en áreas como la automoción, la robótica y la IoT, la necesidad de enfoques de diseño robustos y predecibles se hace más evidente. La modelización con máquinas de estado finito no solo es una herramienta esencial en el desarrollo de sistemas embebidos, sino que también representa una oportunidad única para transformar la educación en ingeniería, al promover un aprendizaje activo, fomentar la resolución de problemas complejos y preparar a los estudiantes para diseñar sistemas inteligentes que respondan a los desafíos del siglo XXI.

Este trabajo explora la importancia de la modelización con máquinas de estados finitos como una herramienta esencial para la simulación y validación lógica en el desarrollo de sistemas embebidos, argumentando que esta práctica no solo mejora la eficiencia del proceso de diseño, sino que también asegura la confiabilidad y robustez del sistema final.

III. Software

A. MODELOS DE SOFTWARE

En el desarrollo de sistemas automatizados, el uso de modelos se ha vuelto cada vez más común, ya que ofrecen una descripción clara del software de los sistemas embebidos, facilitando la comprensión del sistema y permitiendo un diseño con un nivel de abstracción superior al que se logra con los lenguajes de programación.

Un modelo es una representación simplificada de un sistema que destaca sus propiedades esenciales desde una perspectiva particular. Esto facilita una comprensión más profunda del problema y promueve el intercambio de ideas entre los diseñadores. En el ámbito del diseño de software para sistemas embebidos, las FSM han sido tradicionalmente utilizadas para modelar el comportamiento de sistemas electrónicos e informáticos. Si bien existen otros modelos que se pueden utilizar como Redes de Petri, Statecharts, diagramas de flujo, y otros. Como una extensión avanzada de las FSM, los diagramas de estado (Statecharts) se han integrado al estándar UML (Lenguaje Unificado de Modelado) para describir el comportamiento de sistemas en modelos abstractos.

Por razones de facilidad de uso se optó por el uso de Máquinas de Estado Finito (Fig.1). Estos diagramas ilustran el conjunto de estados por los cuales un objeto transita a lo largo de su ciclo de vida en una aplicación. Como plantea Baron Ruiz (2009) *“Una máquina de estados es un modelo computacional, basado en la teoría de autómatas, que se utiliza para describir sistemas cuyo comportamiento es función de los eventos actuales y de los eventos que ocurrieron en el pasado. En cada instante de tiempo la máquina se encuentra en un estado, y dependiendo de las entradas, actuales y pasadas, que provienen del ambiente, la máquina cambia, o no, de estado pudiendo realizar acciones que a su vez influyen en el ambiente”*. El análisis de cómo un objeto responde a eventos y cómo estas respuestas se vinculan con acciones y transiciones es central en estos diagramas. Los elementos clave de estos diagramas son definidos así:

- Estado: es un contexto en el que se encuentra el sistema en un momento dado. Cada estado representa un punto del código en el que el sistema está esperando eventos o condiciones que le permitan cambiar a otro estado.
- Evento: es una ocurrencia o condición que desencadena una transición entre estados. Los eventos pueden ser externos, como una entrada del usuario o una señal de hardware, o internos, como un temporizador que desborda.
- Transición: son los cambios o movimientos entre estados que ocurren en respuesta a ciertos eventos o condiciones. Una transición define la regla o el criterio bajo el cual el sistema pasa de un estado a otro.
- Acción: son actividades o procesos que se llevan a cabo cuando el sistema está en un estado específico o durante una transición entre estados. Las acciones pueden incluir tareas como el envío de una señal, la actualización de una variable o la ejecución de una función.



Fig. 1. Máquina de Estado Finito.

- Estado inicial: estado a partir del cual se empieza a ejecutar la FSM.

B. Herramienta software

Para llevar a cabo la modelización y simulación se analizaron varias herramientas existentes, entre ellas, IART Visual State [3], Matlab Stateflow [4], QM de Quantum Leaps [5], uModel Factory [6][7][8].

El análisis de estas herramientas concluyo en que el desarrollo nacional implementado por un grupo de investigación de la UTN FRBA, uModel Factory (Fig.2) era la mejor elección debido a su simplicidad de uso, licencia Open Source y que cumple con todas las actividades necesarias tanto para crear el modelo, realizar la simulación lógica y generar el código. Además permite realizar la documentación en forma automática utilizando el formato HTML.

C. Plataforma Hardware

Históricamente en la materia se trabajó con microcontroladores PIC de las familias 16F877, luego se pasó al 18F46K20 a través de una placa de desarrollo MCE Starter Student. Actualmente se utilizan dos plataformas embebidas; por un lado se utiliza los Stick LPC1769 más baseboard de expansión (Fig.3) y por otro lado la ESP32 (Fig.4).

En el caso de la LPC1769 se utiliza lenguaje de programación C con uso de HAL CMSIS y LPCOpen con el IDE MCUXpresso [9][10][11][12]. Para la ESP32 [13][14] se usa el IDE Vscode con Framework Arduino[15].

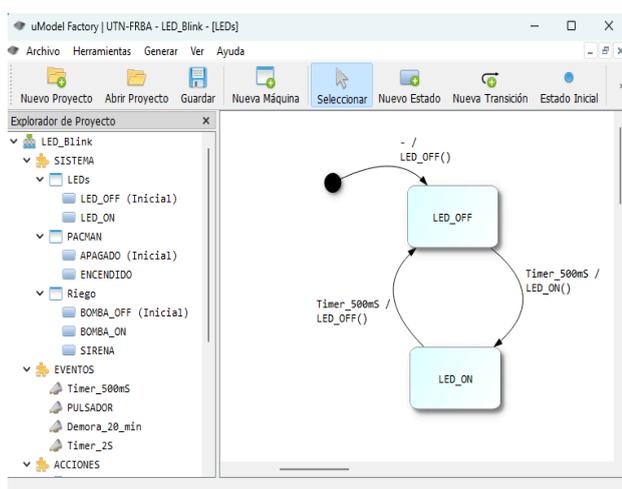


Fig. 2. Interface de uModel Factory.

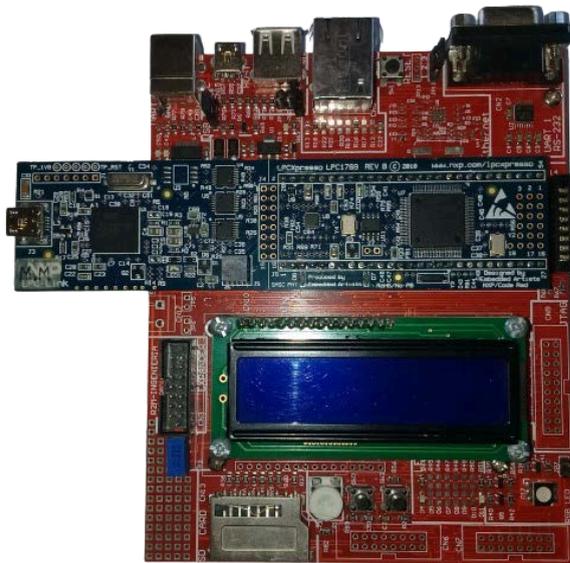


Fig. 3. Placa de desarrollo LPC1769 con baseboard.



Fig. 4. ESP32

IV. Enfoque metodológico

A. FLUJO DE TRABAJO

En el diagrama de la Fig.5 se presenta la secuencia de pasos utilizada en la materia para la realización de una actividad práctica. Como podemos ver se parte de los requisitos de diseño para luego realizar una división y establecer especificaciones de hardware y software. Por un lado se eligen los componentes de hardware, como microcontrolador o placa de desarrollo y componentes auxiliares según las especificaciones, se realiza el diseño y realizan pruebas básicas de estos componentes. Por ejemplo, si se conecta un relé, se verifica la funcionalidad del mismo. Una verificación que se debe realizar siempre al conectar distintos módulos o sensores es la vinculación de la referencia o GND entre circuitos.

De las especificaciones de software surge el diseño del modelo o grafo de estados. El estudiante inicia por comprender a profundidad el problema a resolver, identificando los estados clave que puede adoptar el sistema. A continuación, se definen los eventos que pueden desencadenar transiciones entre estos estados y las acciones correspondientes a cada

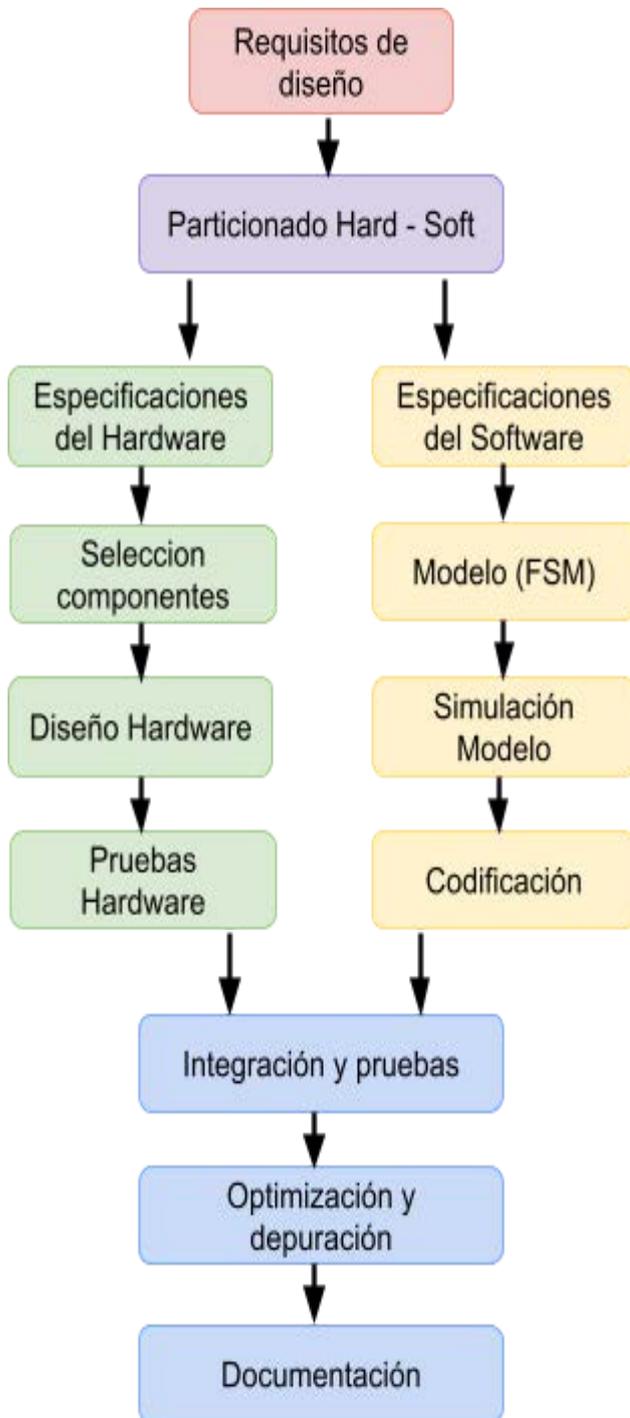


Fig. 5. Etapas del enfoque metodológico aplicado.

evento. Esta conceptualización inicial puede evolucionar a medida que se profundiza en el análisis del problema. Es importante destacar que este proceso es independiente del lenguaje de programación a utilizar. Una vez que se ha definido la máquina de estados finita, se procede a simular su comportamiento para verificar si los estados, eventos y acciones seleccionados son adecuados y completos. La simulación permitirá validar el modelo antes de su implementación en cualquier lenguaje de programación específico.

La simulación lógica se lleva a cabo usando uModel Factory, una vez que la simulación es correcta se pasa a la generación del código. La herramienta genera la FSM usando una estructura switch-case donde el estudiante debe completar el código según el lenguaje de programación elegido. Acto seguido se realiza una integración de hardware y software, para esto se compila y se graba la memoria del microcontrolador para realizar una prueba en circuito. En esta etapa se realiza la depuración y optimización en caso de ser necesario.

Finalmente se documenta el trabajo. Se elaborará un informe técnico detallado que documentará el proceso completo. Este informe incluirá la justificación de cada decisión, los resultados obtenidos en cada etapa y los códigos fuente utilizados.

B. EJEMPLO PRÁCTICO

A modo de ejemplo se plantea un ejercicio clásico utilizado en las prácticas que consiste en controlar un semáforo con LEDs usando una FSM.

Este caso de uso describe el funcionamiento de un semáforo de tres luces (rojo, amarillo, verde) en un entorno urbano. El semáforo regula el tráfico en una calle garantizando el flujo seguro y ordenado de vehículos y peatones. Para simplificar se implementa en principio solo un sentido, para luego agregar el sentido opuesto y cruce peatonal. En la Fig. 6 se muestra la FSM del semáforo propuesto en la actividad. Dentro de los objetivos del ejercicio, si bien es un ejercicio básico, pero se destacan el uso de Timer SysTick e interrupciones, uso de una base de tiempo fija y ampliación usando una variable TICKS para ampliar el tiempo de demora.

También se tiene como objetivos de la actividad práctica realizar correctamente la lógica de funcionamiento a través del modelo de la FSM y su simulación lógica en la herramienta uModel Factory. Luego con esta herramienta se genera el código esqueleto (FSM) para el programa. Cabe aclarar que este código esqueleto es genérico en lenguaje C y se debe agregar todo el resto del código para los GPIO, Timers. Configuraciones, etc. Además, se debe realizar la conexión de los LEDs a la placa utilizada teniendo en cuenta los valores de corriente de los puertos para el cálculo de la resistencia limitadora correspondiente en cada LED (Fig 7).

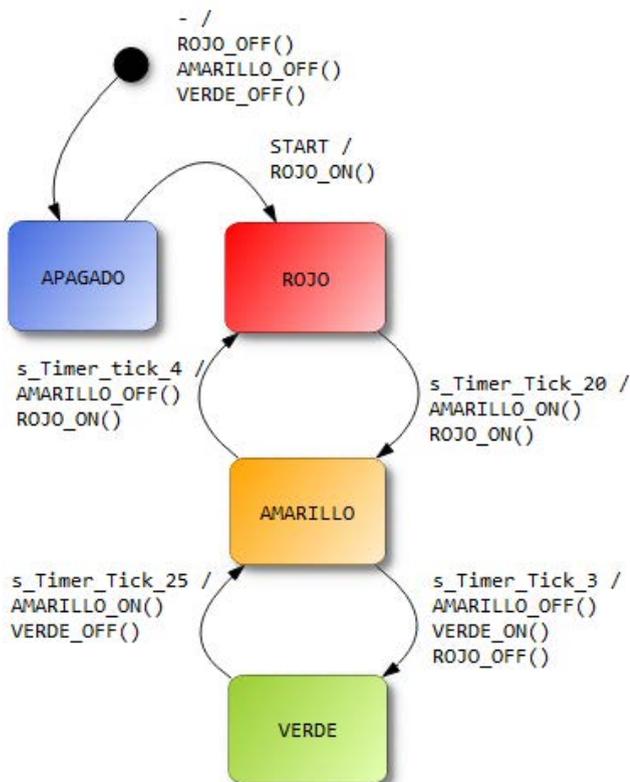


Fig. 6. FSM del semáforo ejemplo.

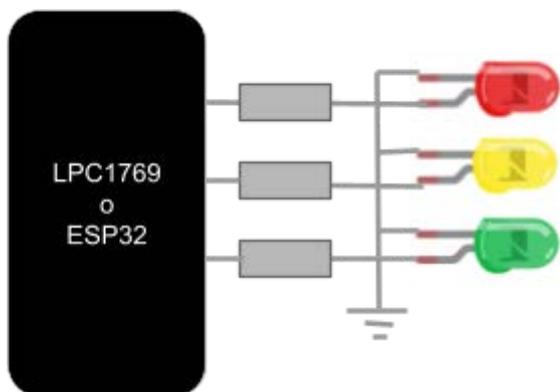


Fig. 7. Conexión eléctrica del semáforo.

VI. RESULTADOS

A través de la implementación de esta metodología se ha observado una mejora en el tiempo de desarrollo para las actividades prácticas y además la comprensión de los conceptos utilizados. Los estudiantes también observaron que a través del uso de la FSM como modelo de software es más fácil realizar mantenimiento y modificación del código, por ejemplo, si tienen que agregar una funcionalidad al sistemas diseñado o realizar mantenimiento después de un cierto tiempo.

VI. CONCLUSIONES

El uso de modelos de software para el diseño de sistemas embebidos plantea una sustancial mejora en cuanto a la comprensión de los conceptos. Además, se logra una estructura más clara y ordenada del código permitiendo fácilmente realizar cambios o mantenimiento del mismo. Esto es debido a que la FSM nos proporciona un mapa de la lógica de funcionamiento del código. La incorporación de esta herramienta facilitó el trabajo de los docentes a cargo de las actividades como así también a la hora de explicar las teorías usando estos modelos.

Por otra parte, los estudiantes lograron desarrollar sus proyectos en un tiempo acotado, demostrando una rápida curva de aprendizaje al adaptarse a diferentes niveles de conocimiento y experiencia, desde estudiantes principiantes hasta aquellos con mayor experiencia en programación.

Si bien se identificaron algunas limitaciones en términos de escalabilidad para proyectos muy grandes, los resultados obtenidos son prometedores y abren nuevas vías para la investigación en el campo de la educación en ingeniería.

Como trabajo a futuro se propone realizar un estudio de los modelos basados en Statecharts para sistemas más complejos.

A modo de cierre en la actividad práctica se les pidió a los estudiantes que envíen una palabra que se les ocurra luego de la práctica que describiera la actividad realizada o que les aparezca acorde a la misma y luego con estas palabras se armó nube de ideas como se muestra la Fig. 8.



Fig. 8 Nube de palabras aportada por los estudiantes después de la práctica.



VIII.REFERENCIAS

- [1] Wagner, F., Schmuki, R., Wagner, T., & Wolstenholme, P. (2006). *Modeling software with finite state machines: A practical approach*. CRC Press.
- [2] D. Harel. Statecharts: a visual formalism for complex systems. *Science of Computer Programming*. Vol 8, N° 3, 231-274 (1987).
- [3] IAR Systems. (2024). IAR Visual State. Disponible en <https://www.iar.com/products/iar-visual-state/> (accedido 9/9/24 - 21:26 h).
- [4] MathWorks. (2024). Stateflow. <https://la.mathworks.com/products/stateflow.html> (accedido 9/9/24 - 21:27 h).
- [5] QM de Quantum Leaps. (2024). QM. <https://www.state-machine.com/products/qm>
- [6] Facultad Regional Buenos Aires, Disponible en <https://www.frba.utn.edu.ar/electronica/umodelfactory-2/>
- [7] CoNaII SI. (2015). uModelFactory: Software para modelado de sistemas embebidos.
- [8] Prieto, M., Sugezky, L., González, N., Giura, M., Kuo, Y., Trujillo, M., & Cruz, J. M. (2017). Evaluation of the uModel Factory software used for the modeling of embedded systems with concurrent states. *IEEE*.
- [9] NXP Semiconductors, MCUXpresso IDE User Guide, Rev. 10.2.0, 14 May, 2018.
- [10] NXP Semiconductors, LPC1769/68/67/66/65/64/63 Datasheet, Rev. 6 , 25 August 2010
- [11] R2M Ingeniería, Esquemático BB1769_CPU_R01_SCH, Rev.1, 26 August 2013.
- [12] NXP Semiconductors, LPCOpen v2.xx LPCXpresso quickstart, Rev. 1, February 8, 2013.
- [13] Espressif Systems (Shanghai) Co., ESP32 Series Datasheet, Version 4.5, February, 2024.
- [14] Espressif Systems (Shanghai) Co., ESP32 Technical Reference Manual, Version 5.1, April, 2024.
- [15] Espressif Systems (Shanghai) Co., ESP-IDF Programming Guide, Version 4.3, Disponible en <https://docs.espressif.com/projects/esp-idf/en/v4.3/esp32/get-started/index.html> (accedido 8/9/24 - 11:30 h).
- [16] Mariano Baron Ruiz, "Curso de Programación de Sistemas Embebidos con Statecharts", *IEEE-RITA*, Vol4, N° 1, (2009).



Aporte a la confiabilidad de software embebido escrito en C mediante contratos y análisis estático con VeriFast

1st Jordán Francisco Insfrán

Laboratorio de Informática y Computación
Aplicada / Facultad de Ingeniería / Universidad
Nacional de Entre Ríos

Oro Verde, Argentina

jordan.insfran@uner.edu.ar

2nd Javier Eduardo Díaz Zamboni

Laboratorio de Informática y Computación
Aplicada / Facultad de Ingeniería /
Universidad Nacional de Entre Ríos

Oro Verde, Argentina

javier.diaz@uner.edu.ar

Resumen—Los sistemas embebidos son claves en la tecnología actual, y C es uno de los lenguajes más importantes para su programación. Sin embargo, los errores en lenguaje C suelen ser difíciles de detectar por pruebas convencionales, ya que suelen ser sutiles, no causan un fallo limpio y tienen efectos impredecibles difíciles de diagnosticar.

Para aportar a la confiabilidad del software embebido programado en C, se presenta una estrategia que combina la programación por contratos y la verificación estática con VeriFast, y se aplica a dos casos simples de código en C. Se hace foco en software crítico, que puede constituir una línea de base para un despliegue amplio, donde una falla en producción implica importantes pérdidas. Este puede ser el caso de drivers empleados en arquitecturas basadas en capas.

Consideramos que el enfoque propuesto aporta a mejorar la confiabilidad del software, porque contribuye a sistematizar las pruebas de correctitud.

Palabras clave: programación por contratos, análisis estático, driver, VeriFast, confiabilidad.

I. INTRODUCCIÓN

Los sistemas embebidos son una pieza clave en el desarrollo de la tecnología actual, y C es uno de los lenguajes más potentes y utilizados para su programación, proporcionando la lógica y el control necesarios para su funcionamiento. Sin embargo, los errores en programas en lenguaje C suelen ser muy difíciles de detectar por pruebas convencionales (*testing*) o la revisión del código, ya que a menudo son

sutiles y normalmente no causan un fallo limpio, sino que tienen efectos impredecibles difíciles de diagnosticar, [1].

La detección de estos errores es muy importante en contextos de aplicaciones críticas, ya que podrían permitir ciberataques, causar pérdidas económicas importantes o producir lesiones graves a las personas o el ambiente, [1]. Por esta razón, en nuestro laboratorio, hemos sistematizado y analizado nuestras prácticas de diseño y desarrollo en el ámbito de la docencia, investigación y desarrollo, considerando esencial la incorporación de conceptos, metodologías y herramientas para la mejora de la confiabilidad del software. Entre las metodologías que estamos incorporando, se destacan la programación por contratos y el análisis estático, objetos de estudio en este trabajo para mejorar la confiabilidad del software.

Esta integración es crucial, ya que las aplicaciones de nuestro interés, y aquellas en las que nuestros futuros graduados podrían trabajar, pueden ser de carácter crítico. Ejemplos de estas aplicaciones pueden incluir desarrollo de software integrado en automóviles, aviones, aplicaciones espaciales, nucleares, médicas o químicas; sistemas complejos de procesamiento de datos bioinformáticos que contribuyen al diagnóstico basado en información de pacientes, controladores para dispositivos en aplicaciones de Internet de las Cosas en entornos críticos como hospitales, entre otros.

En estos contextos la confiabilidad de los sistemas es prioritaria, y mucho del software



desarrollado en sistemas críticos se programa en lenguaje C. Esto se debe a que es un lenguaje que produce un código máquina eficiente y de alto rendimiento, es portable entre sistemas, y tiene una gran y activa comunidad de programadores. A pesar de sus virtudes, los errores de programación en C, como los accesos ilegales a memoria y las carreras de datos (*data races*), no son fáciles de detectar y pueden facilitar ciberataques o causar graves daños.

Para estos casos, una apropiada combinación de programación por contratos y análisis estático pueden aportar una forma sistemática de detectar errores en C, aportando a mejorar la confiabilidad del software. Para mostrar esto, en este trabajo se presenta un análisis de códigos simples de uso de programación por contratos en C, para ser analizados y verificados con la herramienta VeriFast. La estrategia de desarrollo con la metodología de programación por contratos se plantea a partir de algunos requerimientos básicos mostrando la conexión entre estos y el código. En forma general, el enfoque planteado permite aumentar la confiabilidad del firmware en su conjunto, la legibilidad y la testabilidad del código y, además, favorece el pensamiento sistémico, computacional y algorítmico del programador/a. En particular, para este trabajo, se hace foco en aquel software que constituye la línea de base para organizaciones o empresas que pretenden un amplio despliegue de su firmware donde una falla en producción podría implicar importantes pérdidas. Este es el caso de drivers empleados en arquitecturas basadas en capas, [2].

II. PROGRAMACIÓN POR CONTRATOS

El diseño por contratos es una metodología en ingeniería de software, propuesta por Bertrand Meyer, que está pensada para mejorar la calidad de la construcción de software, haciéndolo más confiable, propiedad que combina robustez y correctitud, [3], [4].

En esta metodología se debe definir claramente el comportamiento esperado de cada función –y en métodos y clases de lenguajes que soportan orientación a objetos–. Más concretamente, el diseño por contrato, enfatiza el uso de los conceptos de precondiciones, postcondiciones e invariantes y

los incorpora estratégicamente en el concepto de contrato. La metodología propone hacer explícitas y sistemáticas las verificaciones del contrato. Las precondiciones y postcondiciones forman parte de la interfaz de las funciones, y establecen propiedades lógicas que deben satisfacerse antes y después, respectivamente, de la ejecución de la tarea u operación que la función implementa, [4], [5], [6].

Las precondiciones son los criterios lógicos que deben cumplirse cuando se invoca una función, mientras que las postcondiciones son los criterios lógicos que deben cumplirse cuando una función completa su tarea de manera correcta. En general, las funciones reciben argumentos sobre los que se tienen determinadas expectativas para que la operación programada pueda proceder. En parte, estas refieren a los tipos de dato de los argumentos, algo que por ejemplo C++ y otros lenguajes compilados y fuertemente tipados validan durante la compilación. Es decir, el compilador asegura que los argumentos sean del tipo correcto. Sin embargo, existen otras expectativas sobre los parámetros. Por ejemplo, el conjunto de posibles valores válidos que pueden tomar los argumentos en la invocación de la función. Y algo similar sucede con los valores de retorno. También se espera sobre ellos que, si la función se ejecuta correctamente, cumpla una expectativa lógica. Contar con información de precondiciones y postcondiciones es muy útil para el desarrollador, para los usuarios de la función y para los testers, [7].

Algunos lenguajes modernos incorporan contratos nativamente y otros lo pueden hacer con librerías. C, por su parte, no provee nativamente soporte para contratos, pero existe la biblioteca de QuantumLeaps [8], que provee soporte para C embebido, y en otros casos se implementa con anotaciones *ad hoc*.

La programación por contratos es la forma en la que se programa en el diseño por contrato y se refiere a la implementación concreta del diseño por contratos, ya sea con el uso de notación *ad hoc*, librerías o con el soporte nativo del lenguaje. Esta forma de programación contrasta con el enfoque adoptado en programación defensiva, [9].

III. ANÁLISIS ESTÁTICO CON VERIFAST

La verificación estática es una técnica, generalmente automatizada con herramientas, para análisis estático de código. Entra dentro de la categoría de verificación formal de software, que consiste en la aplicación de técnicas matemáticas rigurosas para la verificación de software. La verificación formal, y en general, los métodos formales de desarrollo, no son muy utilizados. Esto puede deberse a múltiples razones, aunque algunas de ellas podrían ser más mitos que realidades. Entre las razones más conocidas se puede mencionar que las metodologías formales son complejas y requieren un alto nivel de formación y experiencia. También se suele argumentar que son más costosas en términos de tiempo y recursos, y que requieren una gran carga de documentación adicional. Otra razón que se suele mencionar como desventaja de la aplicación de técnicas formales, es la limitación de disponibilidad de herramientas, es decir, las herramientas de desarrollo de software que soportan metodologías formales son menos comunes y pueden no estar tan bien integradas en los entornos de desarrollo modernos.

De las herramientas de verificación estática para el lenguaje C se ha desarrollado y usado ampliamente LCLint, [10], para chequear código y, posteriormente se desarrolló VeriFast, [11], que es una herramienta de verificación estática para los lenguajes para C y java, y más recientemente para C++, [12]. Lo interesante de estas herramientas es que requieren de una sintaxis diferenciada para poder analizar el código fuente. En particular, VeriFast utiliza anotaciones *ad hoc* en las se especifica el contrato de cada función.

VeriFast verifica ciertas propiedades de correctitud de programas. Esto lo hace analizando programas que pueden estar compuestos por uno o más archivos de código fuente. Como resultado, puede indicar que no se han detectado errores o, en caso contrario, señalar la ubicación exacta de un posible problema. De acuerdo a los autores, si VeriFast no encuentra errores significa que el programa, [1]:

- no realiza accesos ilegales a la memoria, como leer o escribir un campo de instancia de estructura después de que la instancia de estructura haya sido liberada, o leer o escribir

más allá del final de una matriz (conocido como desbordamiento de búfer, la causa más común de vulnerabilidades de seguridad en sistemas operativos y servicios de Internet) y

- no incluye un cierto tipo de errores de concurrencia conocidos como carreras de datos, es decir, accesos conflictivos no sincronizados del mismo campo por múltiples hilos. Los accesos se consideran conflictivos si al menos uno de ellos es un acceso de escritura. Y

- cumple las precondiciones y postcondiciones de las funciones especificadas por el programador en forma de comentarios especiales (conocidos como anotaciones) en el código fuente.

Para detectar todos los errores, VeriFast realiza una ejecución simbólica modular del programa. En concreto, VeriFast ejecuta simbólicamente el cuerpo de cada función del programa, partiendo del estado simbólico descrito por la precondición de la función, comprobando que los permisos están presentes en el estado simbólico para cada posición de memoria a la que accede una sentencia, actualizando el estado simbólico para tener en cuenta el efecto de cada sentencia, y comprobando, siempre que la función retorna, que el estado simbólico final satisface la postcondición de la función. Un estado simbólico consta de un montículo simbólico, que contiene permisos (conocidos como chunks) para acceder a determinadas posiciones de memoria, un almacén simbólico, que asigna un valor simbólico a cada variable local, y una condición de ruta, que es el conjunto de suposiciones sobre los valores de los símbolos utilizados en el estado simbólico en la ruta de ejecución actual. La ejecución simbólica siempre termina, porque gracias al uso de invariantes de bucle, cada cuerpo de bucle sólo necesita ejecutarse simbólicamente una vez, y la ejecución simbólica de una llamada a función sólo utiliza la precondición y la postcondición de la función, no su cuerpo, [1].

Una ventaja del enfoque de análisis estático con VeriFast es que no requiere incorporar su notación de contratos en el código ejecutable. Por lo tanto, no interfiere con el desempeño del firmware, por lo que el tiempo de ejecución no se verá afectado, [2].

IV. METODOLOGÍA

Como se indicó previamente, este trabajo se orienta al desarrollo de firmware para sistemas embebidos en lenguaje C. Para este fin, se propuso la siguiente metodología donde se omite intencionalmente el diseño y desarrollo de pruebas (testing) para limitar la extensión de esta presentación. Obviamente, las pruebas deberían estar siempre presentes en el ciclo de desarrollo.

La estrategia empleada consiste en:

1. Selección de casos y definición de requerimientos: *drivers* y biblioteca de funciones para uso en drivers.
2. Declaración de funciones y estructuras de cada driver y biblioteca.
3. Incorporación de contratos y aserciones en cada caso.
4. Desarrollo de aplicación de prueba que utiliza el driver y/o la biblioteca.
5. Primer análisis estático con VeriFast con revisión y corrección si es necesario.
6. Incorporación de aserciones en la aplicación.
7. Segundo análisis estático con VeriFast con revisión y corrección si es necesario.
8. Verificación y validación. Se vuelve a etapas previas si es necesario.

Dentro de esta secuencia de pasos, el análisis estático con VeriFast es la etapa central. El paso 6, podría adelantarse sin mayores inconvenientes porque constituyen afirmaciones sobre lo que se asume como un comportamiento correcto para un punto en el camino de ejecución de un programa.

V. RESULTADOS Y DISCUSIÓN

A continuación se analizan brevemente dos códigos de prueba para observar con algún detalle cómo se emplea VeriFast y la notación de contratos. El primer código refiere al desarrollo de una función para sumar dos valores enteros no negativos. El código, con la notación de contratos más simple posible se muestra en la Fig. 1. Se observan dos cláusulas expresadas como comentarios iniciados con *//@*: *requires* y *ensures*. Estas cláusulas constituyen las precondiciones y postcondiciones, respectivamente para cada función. Y para estas funciones se establece el caso más general posible en que se asume que

la función es correcta, sin importar los valores actuales que adoptan los argumentos formales y el algoritmo que implementa la función para dar lugar al valor de retorno.

```

1  int sum(int a, int b)
2      //@ requires true;
3      //@ ensures true;
4  {
5      return a + b;
6  }
7
8  unsigned short sum2(unsigned short
9  a, unsigned short b)
10     //@ requires true;
11     //@ ensures true;
12 {
13     return a + b;
14 }

```

Fig. 1. Funciones que suman dos números enteros no negativos con notación de contratos.

La función principal (*main*) que emplea estas funciones se muestra en la Fig. 2. Notar que, al igual que antes, se emplean las pre y postcondiciones más débiles posibles (es decir, las más abarcativas).

```

1  int main()
2      //@ requires true;
3      //@ ensures true;
4  {
5      int x = 10;
6      int y = 2147483637;
7      int result = sum(x, y);
8
9      unsigned short x2 = 10;
10     unsigned short y2 = 65525;
11     unsigned short result2 = sum2(x2,
12     y2);
13     printf("%i\n", result);
14     printf("%u\n", (unsigned
15     int)(result2));
16     return 0;
17 }

```

Fig. 2. Función principal con ejemplo de uso de las funciones que suman números no negativos.

La principal ventaja de este sencillo enfoque consiste en permitir aplicar la herramienta de verificación estática VeriFast de forma muy

simple para realizar un primer análisis del código. Como se indicó previamente, VeriFast realiza una ejecución simbólica del código C considerando errores comunes como fallas en alocaación y liberación de memoria (con *malloc* y *free*), divisiones por 0, acceso a direcciones de memoria no reservada, entre otros errores. Por ejemplo, si se añaden previo al retorno las líneas `int a=1; int b=0; int r=a/b;` el analizador indica “*Denominator might be 0. (Cannot prove !(0 = 0).)*” en la división `a/b`. Aunque en este caso, el problema es obvio, el analizador rastrea si la variable “b” posee un valor diferente de cero en los diferentes caminos de ejecución por los que puede transitar el programa. Esencialmente, realiza de forma sistemática un proceso de análisis del flujo de ejecución del código fuente similar al que realiza un/a programador/a cuando razona sobre la secuencia de pasos del programa que escribe para implementar un algoritmo. La ventaja del enfoque es directa, el analizador simbólico puede llevar la cuenta de una cantidad muy superior de valores actuales de argumentos formales de lo que lo puede hacer el/la más experimentado/a programador/a.

Volviendo la atención al código de la Fig. 2, la salida que resulta de ejecutar este programa se muestra en la Fig. 3. En este caso, los valores mostrados corresponden a los extremos superiores de cada rango de números enteros que pueden ser representados por los tipos de datos seleccionados.

1	2147483647
2	65535

Fig. 3. Salida por terminal del programa que suma números enteros no negativos.

Esta salida es un caso extremo y, si se modifican los valores `x` y `x2` a 11, en lugar de 10, la salida es como muestra la Fig. 4. Es sencillo apreciar que los resultados son incorrectos y sucedió un desbordamiento de rango. De aquí que las precondiciones y postcondiciones generales empleadas como primera aproximación no aportan la información necesaria al analizador para detectar el problema de desbordamiento.

1	-2147483648
2	0

Fig. 4. Salida por terminal del programa que suma

números enteros no negativos cuando `x=11` y `x2=11`.

La Fig. 5 muestra el código que define las funciones `sum` y `sum2` con precondiciones para que el analizador pueda detectar estáticamente el desbordamiento de rango. En este caso, el analizador indica “*Cannot prove condition. (Cannot prove (11 + 2147483637) <= 2147483647.)*” para la línea 2 de la Fig. 5, y “*Cannot prove condition. (Cannot prove (11 + 65525) <= 65535.)*” para la línea 10 de la misma figura. Estas detecciones se realizan en forma secuencial, es decir, primero surge una y debe corregirse el programa, luego surge la otra y debe corregirse el programa nuevamente. En este caso, la corrección consiste en que los valores a sumar sean menores que los valores máximos permitidos por los tipos de dato seleccionados.

```

1  int sum(int a, int b)
2      //@ requires a + b <= INT_MAX;
3      //@ ensures true;
4  {
5      return a + b;
6  }
7
8  unsigned short sum2(unsigned short
   a, unsigned short b)
9      //@ requires a + b <= USHRT_MAX;
10     //@ ensures true;
11 {
12     return a + b;
13 }
```

Fig. 5. Funciones que suman dos números enteros no negativos con notación de contratos que previene el desbordamiento de rango.

Otra posibilidad es que uno de los sumandos sea negativo para la función `sum`, lo cual no está permitido según la idea inicial para estas funciones de suma. Así, si `x = -5` en la línea 5 de la función principal mostrada en la Fig. 2, el analizador estático no detectará el problema. Por lo tanto, siguiendo la misma estrategia debería restringirse la precondición para aceptar valores que sean no negativos únicamente. Esto se observa en el código la Fig. 6. donde el contrato establece ahora como precondición adicional para las entradas de la función `sum` que sean valores no negativos. Múltiples condiciones se unen con el conector `&&` que corresponden a la operación lógica de

conjunción (o *and*) para VeriFast, [1]. Con esta precondition, el analizador estático indicará con un mensaje “*Cannot prove condition. (Cannot prove 0 <= (0 - 5).)*” para la línea 2 de la Fig. 6 en caso de ser $x=-5$ en la línea 5 de la función principal mostrada en la Fig. 2.

```

1  int sum(int a, int b)
2      //@ requires a >= 0 && b>= 0
   && a + b <= INT_MAX;
3      //@ ensures true;
4  {
5      return a + b;
6  }
```

Fig. 6. Funciones que suman dos números enteros no negativos con notación de contratos que previene valores actuales negativos y desbordamiento de rango.

Otro tipo de condiciones pueden ser añadidas en el código fuente mediante las aserciones. A modo de ejemplo, se añade dos aserciones de VeriFast a la función principal en las líneas 8 y 13 de la Fig. 7. Las aserciones funcionan como afirmaciones y son similares a las que se provee la biblioteca *assert.h*, con la diferencia de que es un comentario y no afecta el tiempo de ejecución, esto es, no añade comprobaciones en tiempo de ejecución. Como desventaja encontramos que es necesario comprobar el código con el analizador estático para que la aserción sea comprobada estáticamente.

```

1  int main()
2      //@ requires true;
3      //@ ensures true;
4  {
5      int x = 10;
6      int y = 2147483637;
7      int result = sum(x, y);
8      //@ assert result == 2147483647;
9
10     unsigned short x2 = 10;
11     unsigned short y2 = 65525;
12     unsigned short result2 = sum2(x2,
13     y2);
14     //@ assert result2 == 65535;
15     printf("%i\n", result);
16     printf("%u\n", (unsigned
17     int)(result2));
18     return 0;
```

```

19 }
```

Fig. 7. Función principal que implementa las pruebas de las funciones que suman números no negativos.

Cuando se aplica el análisis estático sobre el código de la Fig. 7 con los contratos en las funciones de suma de actualizados hasta la Fig. 6, el analizador indica “*Assertion might not hold. (Cannot prove result = 2147483647.)*” para la línea 8 de la Fig. 7. Esto se debe a que a partir del contrato de la función *sum* no es posible vincular simbólicamente las entradas “a” y “b” con el resultado “result”; ver Fig. 5. Para resolver este inconveniente, el contrato de la función *sum* se modifica como se muestra en la Fig. 8 donde se hace referencia al valor de retorno con *result*, [1].

```

1  int sum(int a, int b)
2      //@ requires a >= 0 && b>= 0
   && a + b <= INT_MAX;
3      //@ ensures a + b == result;
4  {
5      return a + b;
6  }
7
8  unsigned short sum2(unsigned short
   a, unsigned short b)
9      //@ requires a + b <= USHRT_MAX;
10     //@ ensures a + b == result;
11 {
12     return a + b;
13 }
```

Fig. 8. Funciones que suman dos números enteros no negativos con notación de contratos que previenen desbordamiento de rango y considera el vínculo entre las entradas y la salida.

Es importante destacar que la elección de los tipos de datos de las entradas a las funciones constituye en sí mismo una restricción, como lo son las preconditiones. En efecto, al elegir el tipo de dato se define el rango de conjunto de valores que podrá adoptar la entrada. Es por este motivo que, para el caso de la función *sum2*, no fue necesario añadir la precondition “ $a \geq 0 \ \&\& \ b \geq 0$ ” debido a que es el propio tipo de dato el que impone esta restricción implícitamente.

Otro aspecto de importancia que debe notarse en las funciones *sum* y *sum2* es que el cuerpo de ambas ha permanecido

extremadamente simple (`return a + b;`). Quedan como comentarios por fuera del cuerpo de ambas funciones las restricciones tanto para las entradas como para el valor de retorno. Por otra parte, estas restricciones que constituyen el contrato se rigen, únicamente, por el requerimiento establecido, esto es, que las funciones implementan la suma de valores enteros no negativos.

VI. CASO DRIVER DE SENSOR LUMÍNICO

Retomando la metodología propuesta, para el primer punto se propone el driver de un sensor LDR. Para este fin la especificación propuesta se muestra en la Tabla I donde aparecen dos columnas. En la primera se indica el código del requerimiento funcional (RF) o requerimiento no funcional (RNF), y en la segunda se describe brevemente el requerimiento.

TABLA I
ESPECIFICACIÓN PARA DRIVER DE SENSOR LDR

RF / RNF	Referencia
RF-1	Inicialización del LDR Configurar el ADC para iniciar la lectura del sensor LDR y establecer los valores iniciales en la estructura 'LDR_read'. El rango de valores va de 0 a 4095, y un valor alto significa menor intensidad lumínica para este módulo.
RF-2	Lectura Promediada del LDR Realizar 10 lecturas en 1 ms, promediar los valores obtenidos.
RF-3	Verificación de Desconexión del LDR Determinar si hay una alta probabilidad de que el módulo LDR se encuentre desconectado. Esta verificación se hace comparando la lectura promedio con un umbral de desconexión. Este umbral es un valor que a partir del cual se sabe que podría deberse a una desconexión entre el módulo y el microcontrolador.
RNF-4	Estructura de datos Tanto la lectura, códigos de error, como el umbral de desconexión deben gestionarse en una única estructura de datos de C (struct).

El driver del sensor LDR está programado para una esp32, y consiste en dos archivos de código fuente: `ldr_module.h` y `ldr_module.c`. El contenido del módulo `.h` se muestra en la Fig. 9.

```

1  #ifndef LDR_DRIVER_H
2  #define LDR_DRIVER_H
3
4  #include "esp_adc/adc_oneshot.h"
5
6  struct LDR_read {
7      uint16_t light_intensity;
8      esp_err_t error_code; // ESP_OK
9      uint16_t
10     disconnection_threshold;
11 };
12 // Initialize the LDR by configuring
13 // the ADC
14 void ldr_init(adc_unit_t p_unit_id,
15 struct LDR_read* ldr_read);
16
17 // Take 10 readings in 1 ms, average
18 // the values, and check if the module
19 // is disconnected
20 void ldr_read_average(struct
21 LDR_read* ldr_read);
22
23 // Check if the LDR module is
24 // disconnected
25 bool ldr_is_disconnected(struct
26 LDR_read* ldr_read);
27
28 #endif // LDR_DRIVER_H

```

Fig. 9. Contenido de `ldr_module.h`.

Y, el contenido del módulo `.c` para el mismo driver se muestra en la Fig. 10.

```

1  #include "ldr_module.h"
2  #include <rom/ets_sys.h>
3
4  static adc_oneshot_unit_handle_t
5  adc_handle;
6  static adc_oneshot_unit_init_cfg_t
7  init_config;
8
9  void ldr_init(adc_unit_t p_unit_id,
10 struct LDR_read* ldr_read) {

```

```

8      // Initial configurations
9
10     ldr_read->disconnection_threshold =
11     100; // Threshold to detect
12     disconnection
13     init_config.unit_id = p_unit_id;
14
15     ESP_ERROR_CHECK(adc_oneshot_new_unit
16     (&init_config, &adc_handle));
17
18     //-----ADC
19     Config-----//
20     adc_oneshot_chan_cfg_t config =
21     {
22         .atten = ADC_ATTEN_DB_12,
23         .bitwidth = ADC_BITWIDTH_12,
24     };
25
26     ESP_ERROR_CHECK(adc_oneshot_config_c
27     hannel(adc_handle, ADC_CHANNEL_0,
28     &config));
29 }
30
31 void ldr_read_average(struct
32 LDR_read* ldr_read) {
33     int sum = 0;
34     const int num_samples = 10;
35     int read_raw; // adc value
36
37     esp_err_t err, err_aux;
38     err = ESP_OK;
39     err_aux = ESP_OK;
40
41     for (int i = 0; i < num_samples;
42     i++) {
43         err_aux =
44         adc_oneshot_read(adc_handle,
45         ADC_CHANNEL_0, &read_raw);
46
47         if (err_aux != ESP_OK) {
48             err = err_aux;
49             break;
50         }
51         sum += read_raw;
52         ets_delay_us(100); // Wait
53         100 microseconds for a total of 1 ms
54     }
55
56     ldr_read->light_intensity = sum
57     / num_samples;
58     ldr_read->error_code = err;

```

```

43 }
44
45 bool ldr_is_disconnected(struct
46 LDR_read* ldr_read) {
47     return ldr_read->light_intensity
48     < ldr_read->disconnection_threshold;
49     // Check if the value is below the
50     threshold
51 }

```

Fig. 10. Contenido de ldr_module.c.

La estructura de directorios del firmware para una esp32 se muestra en la Fig. 11. Se observa en esta figura que el driver para el módulo LDR se encuentra dentro del directorio *devices* que, a su vez, se encuentra en el directorio *drivers*. Esta estructura permite la reutilización de los drivers en diferentes proyectos lo que favorece la creación de código reutilizable.

Para este trabajo, se implementa un proyecto de prueba compuesto por tres archivos ubicados en el directorio main (dentro del directorio test_project): test_app.h, test_app.c y test_project.c. Los primeros dos archivos de la aplicación implementan funciones que incluyen e invocan funciones de los drivers para implementar sus funcionalidades, y el código fuente test_project.c incluye a test_app.h. De esta forma, se logra una arquitectura de software por capas similar a la expuesta en [2].

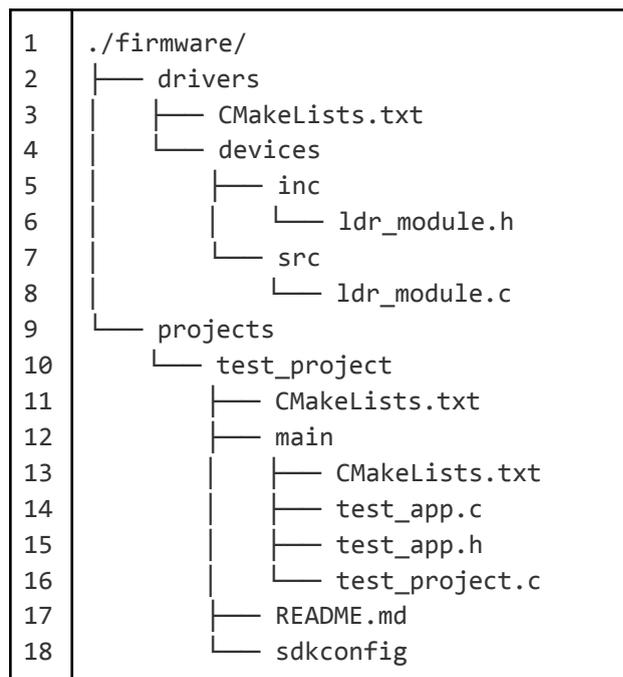


Fig. 11. Árbol de directorios para el firmware.

Para añadir la notación de contratos de VeriFast al driver se incorporan las anotaciones al archivo de cabecera como muestra la Fig. 12. Se observa la utilización de un predicado (*predicate*) para proveer concisión a los contratos y ocultar información del módulo en otros módulos lo utilicen LDR, [1]. Por otra parte, la notación empleada se ha mantenido minimalista con el fin de centrar la atención en cómo se incorporan al módulo. La forma de realizar verificaciones a partir de los argumentos se realiza de forma similar a cómo se mostró con la función de suma. Es importante destacar que si los predicados son recursivos VeriFast no verifica su terminación, y no informa de recursión o bucles infinitos por lo que todavía es responsabilidad del programador/a tener cuidado en este punto, [1].

```

1  #ifndef LDR_DRIVER_H
2  #define LDR_DRIVER_H
3
4  #include "esp_adc/adc_oneshot.h"
5
6  /*@
7  predicate LDR_read_pred(struct
8  LDR_read *ldr_read, uint16_t
9  the_light_intensity, esp_err_t
10 the_error_code, uint16_t
11 the_disconnection_threshold) =
12   ldr_read->light_intensity |->
13   the_light_intensity
14   &*& ldr_read->error_code |->
15   the_error_code
16   &*&
17   ldr_read->disconnection_threshold
18   |-> the_disconnection_threshold;
19 @*/
20
21 struct LDR_read {
22     uint16_t light_intensity;
23     esp_err_t error_code; // ESP_OK
24     = 0, means no error
25     uint16_t
26     disconnection_threshold;
27 };
28
29 // Initialize the LDR by configuring
30 the ADC
31 void ldr_init(adc_unit_t p_unit_id,
```

```

21 struct LDR_read* ldr_read);
22     /*@ requires true;
23     /*@ ensures
24     LDR_read_pred(ldr_read, _, _, _);
25
26 // Take 10 readings in 1 ms, average
27 the values, and check if the module
28 is disconnected
29 void ldr_read_average(struct
30 LDR_read* ldr_read);
31     /*@ requires true;
32     /*@ ensures
33     LDR_read_pred(ldr_read, _, _, _);
34
35 // Check if the LDR module is
36 disconnected
37 bool ldr_is_disconnected(struct
38 LDR_read* ldr_read);
39     /*@ requires true;
40     /*@ ensures true;
41
42 #endif // LDR_DRIVER_H
```

Fig. 12. Contenido de ldr_module.h con contratos.

Otro aspecto de relevancia para realizar el análisis estático es la forma de vincular las inclusiones de los módulos a analizar, con aquellos que provee el *framework* de desarrollo. Por ejemplo, en el caso del archivo de cabecera ldr_module.h es necesario incluir el módulo esp_adc/adc_oneshot.h (línea 4 de la Fig. 12) para poder emplear la definición de *adc_unit_t* que se encuentra allí. Sin embargo, incluir directamente el código fuente que provee el *framework* de desarrollo derivaría en que el analizador comience a analizar el código fuente de este último. Para evitar esto, una alternativa es crear módulos *mocks* en un directorio separado e indicar estas inclusiones al analizador. De esta forma, el código fuente del driver permanece inalterado respecto a las inclusiones. La Fig. 13 muestra el árbol de directorios del directorio mock, dentro del cual se encuentran los diferentes directorios *include* que se proporcionan al analizador previo a realizar el análisis estático. El árbol replica parcialmente el del *framework* en caso de que sea necesario añadir *mocks* adicionales.

Una alternativa analizada que cabe destacar consistía en añadir directivas de preprocesador al módulo a analizar para realizar inclusiones

condicionales. No obstante, ello implicaba alterar el código fuente original se descartó por tratarse de una nueva posible fuente de errores. Por ejemplo, por error podría quedar activa la inclusión de un archivo de cabecera *mock* en la compilación de producción.

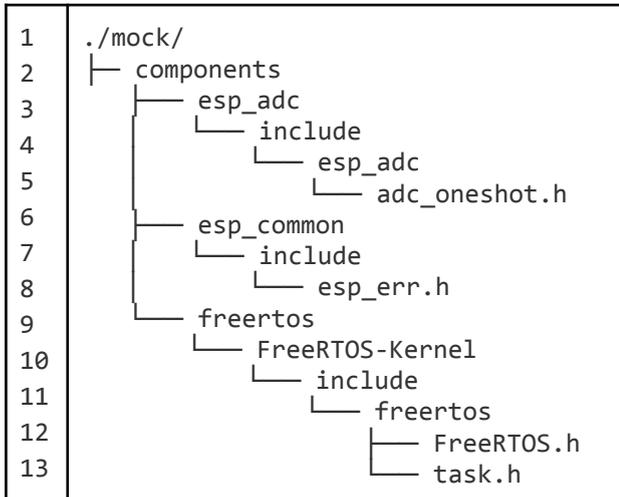


Fig. 13. Árbol de directorios para mocks.

Para comprobar el módulo LDR estáticamente, se crea una función de prueba en un módulo *test_app.c* que puede observarse en la Fig. 14. En este código se invocan las diferentes funciones del módulo *ldr_module.h*. También, se observa la incorporación de un bucle que realiza sucesivas lecturas una vez iniciado el programa en el microcontrolador. Para este fin, es necesario incorporar un invariante de bucle que provea al ámbito del bucle de las expresiones lógicas que emplea en su interior, es decir, el invariante *//@ invariant true;* no funciona. El invariante añadido permite analizar la correctitud del cuerpo del bucle, pero no asegura su terminación. De hecho, en este ejemplo, lo correcto es que el bucle sea infinito.

```

1  #include "test_app.h"
2
3  // inclusion of drivers
4  #include "ldr_module.h"
5
6  // dependencies
7  #include "freertos/FreeRTOS.h"
8
9
10 void test_ldr_module()
    
```

```

11  //@ requires true;
12  //@ ensures true;
13  {
14      struct LDR_read ldr_read;
15
16      ldr_init(ADC_UNIT_1, &ldr_read);
17      // Initialize the LDR. ADC_UNIT_1
18      // corresponds to GPIO36 on the ESP32
19      while (1)
20          //@ invariant
21          LDR_read_pred(&ldr_read,
22          ?light_intensity, ?error_code,
23          ?disconnection_threshold);
24          {
25              ldr_read_average(&ldr_read);
26              if
27              (ldr_is_disconnected(&ldr_read)) {
28                  //@ open
29                  LDR_read_pred(&ldr_read,
30                  light_intensity, error_code,
31                  disconnection_threshold);
32                  printf("LDR probably
33                  disconnected! (LDR Value: %u,
34                  threshold: %u)\n",
35                  (unsigned
36                  int)(ldr_read.light_intensity),
37                  (unsigned
38                  int)(ldr_read.disconnection_threshol
39                  d));
40                  //@ close
41                  LDR_read_pred(&ldr_read,
42                  light_intensity, error_code,
43                  disconnection_threshold);
44              } else {
45                  //@ open
46                  LDR_read_pred(&ldr_read,
47                  light_intensity, error_code,
48                  disconnection_threshold);
49                  printf("LDR Value (100
50                  to 4095, higher means low light
51                  intensity): %u\n", (unsigned
52                  int)(ldr_read.light_intensity));
53                  //@ close
54                  LDR_read_pred(&ldr_read,
55                  light_intensity, error_code,
56                  disconnection_threshold);
57              }
58          }
59          vTaskDelay(pdMS_TO_TICKS(1000)); //
60          Wait 1 second
    
```

```

33 // @ Leak
   LDR_read_pred(&ldr_read,
   light_intensity, error_code,
   disconnection_threshold);
34 }
35 }
    
```

Fig. 14. Contenido de test_app.c con contratos.

Una captura de pantalla a modo ilustrativo de la verificación de este código en el IDE de VeriFast se muestra en la Fig. 15. donde puede observarse un pequeño árbol de ejecución a la derecha, y muy pequeño el código de la Fig. 14 en el editor. El editor permite, a través del árbol de ejecución navegar por diferentes caminos de ejecución simbólica del programa.

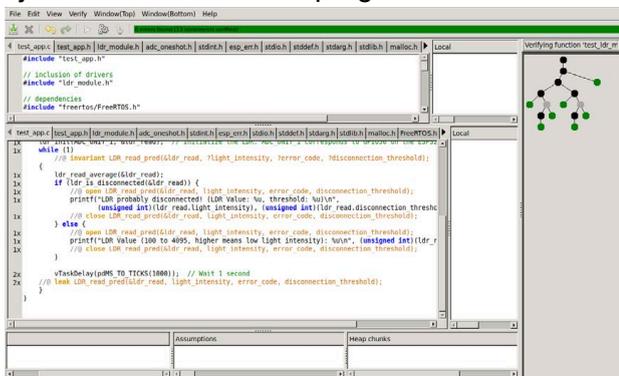


Fig. 15. Captura de pantalla del IDE de VeriFast.

El código mostrado para el módulo LDR posee contratos que le permiten verificar correctamente la ejecución simbólica. Por lo tanto, la estrategia a partir de este punto, consiste en hacer crecer el detalle de los contratos con el objetivo de que el analizador simbólico siga realizando verificaciones correctas.

VII. CONCLUSIONES

En este trabajo se ha presentado una estrategia que combina programación por contratos y análisis estático a software embebido en lenguaje C con la herramienta VeriFast. El enfoque involucra la teoría de programación por contratos donde las precondiciones y poscondiciones se verifican estáticamente de forma simbólica.

Se han analizado dos códigos, uno para sumar valores positivos y otro para un driver de un módulo LDR. En este último caso, se muestra la estructura de directorios y una forma de acoplar el analizador estático al desarrollo.

El enfoque adoptado ha sido gradual, partiendo por una integración de la herramienta con aserciones débiles que, posteriormente, una vez que se tiene una versión verificable del programa, fueron fortaleciéndose.

Una desventaja de la herramienta es que su utilización requiere aprender una nueva sintaxis y comprender el funcionamiento general del analizador. Y, por el contrario, una ventaja importante es que permite ejercitar y agudizar el pensamiento algorítmico y lógico matemático que, en última instancia, proporciona sentido a los programas que se desarrollan para ser ejecutados en un procesador digital moderno.

Consideramos que esta forma de combinar contratos y análisis estático aporta a mejorar la confiabilidad, porque contribuye a las pruebas de correctitud del software. Esto es importante en general para software crítico, y puede ser de mucho valor e importancia para el software que constituye la línea de base para organizaciones o empresas que pretenden un amplio despliegue de su firmware donde una falla en producción podría implicar importantes pérdidas.

Una cuestión importante de mencionar es que lo presentado en relación al análisis estático y, en general, los métodos formales son herramientas y metodologías muy valiosas para el análisis de software. No obstante, es crucial tener en cuenta que, para cualquier lenguaje Turing-completo, como el lenguaje C, encontrar todos los posibles errores de tiempo de ejecución en un programa arbitrario o, en forma más general, cualquier violación de la especificación en el resultado final de un programa, es un problema indecidible. Por lo tanto, no existe un método computacional que pueda garantizar siempre con certeza si un programa arbitrario presentará errores de tiempo de ejecución. A pesar de esta limitación, la investigación y los avances en métodos formales siguen siendo de gran importancia, ya que contribuyen al desarrollo de soluciones aproximadas que pueden mejorar significativamente la confiabilidad del software.

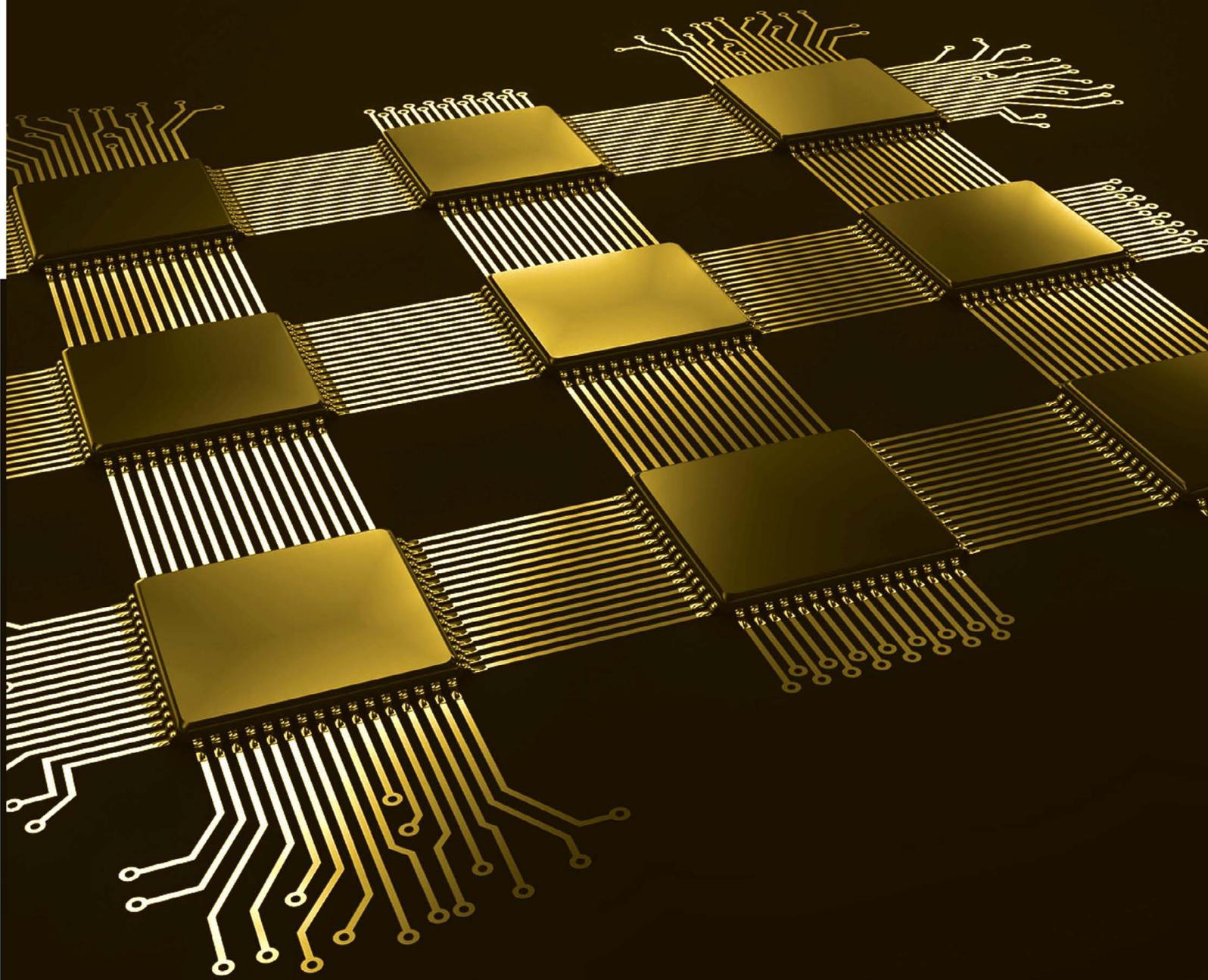
REFERENCIAS

[1] B. JACOBS, J. SMANS, AND F. PIESSENS, "THE VERIFAST PROGRAM VERIFIER: A TUTORIAL." IMEC-DISTRI NET, DEPARTMENT OF COMPUTER SCIENCE, KU LEUVEN - UNIVERSITY OF LEUVEN, BELGIUM, NOV. 28, 2017. [ONLINE]. AVAILABLE:



[HTTPS://PEOPLE.CS.KULEUVEN.BE/~BART.JACOBS/VERIFAST/TUTORIAL.PDF](https://people.cs.kuleuven.be/~bart.jacobs/verifast/tutorial.pdf)

- [2] J. BENINGO, *REUSABLE FIRMWARE DEVELOPMENT. A PRACTICAL APPROACH TO APIs, HALS AND DRIVERS*, 1ST ED. ACCESSED: AUG. 28, 2024. [ONLINE]. AVAILABLE: [HTTPS://LINK.SPRINGER.COM/BOOK/10.1007/978-1-4842-3297-2](https://link.springer.com/book/10.1007/978-1-4842-3297-2)
- [3] B. MEYER, “DESIGN BY CONTRACT,” INTERACTIVE SOFTWARE ENGINEERING INC, TECHNICAL, 1986. [ONLINE]. AVAILABLE: [HTTPS://SE.INF.ETHZ.CH/~MEYER/PUBLICATIONS/OLD/DBC_CHAPTER.PDF](https://se.inf.ethz.ch/~meyer/publications/old/dbc_chapter.pdf)
- [4] B. MEYER, “APPLYING ‘DESIGN BY CONTRACT,’” *COMPUTER*, VOL. 25, NO. 10, ART. NO. 10, OCT. 1992, DOI: 10.1109/2.161279.
- [5] B. MEYER, *OBJECT-ORIENTED SOFTWARE CONSTRUCTION*, 2ND ED. UPPER SADDLE RIVER, N.J: PRENTICE HALL PTR, 1997. [ONLINE]. AVAILABLE: [HTTPS://BERTRANDMEYER.COM/WP-CONTENT/UPLOADS/OOSC.PDF](https://bertrandmeyer.com/wp-content/uploads/OOSC.pdf)
- [6] B. MEYER, *TOUCH OF CLASS*. BERLIN, HEIDELBERG: SPRINGER, 2009. DOI: 10.1007/978-3-540-92145-5.
- [7] B. STROUSTRUP, *THE C++ PROGRAMMING LANGUAGE*, FOURTH EDITION. UPPER SADDLE RIVER, NJ: ADDISON-WESLEY, 2013.
- [8] “DESIGN BY CONTRACT (DBC) FOR EMBEDDED C AND C++.” [ONLINE]. AVAILABLE: [HTTPS://GITHUB.COM/QUANTUMLEAPS/DBC-FOR-EMBEDDED-C](https://github.com/QuantumLeaps/dbc-for-embedded-c)
- [9] S. McCONNELL, *CODE COMPLETE*, 2ND ED. REDMOND, WASH: MICROSOFT PRESS, 2004.
- [10] D. EVANS, J. GUTTAG, J. HORNING, AND Y. M. TAN, “LCLINT: A TOOL FOR USING SPECIFICATIONS TO CHECK CODE,” IN *PROCEEDINGS OF THE 2ND ACM SIGSOFT SYMPOSIUM ON FOUNDATIONS OF SOFTWARE ENGINEERING*, IN SIGSOFT '94. NEW YORK, NY, USA: ASSOCIATION FOR COMPUTING MACHINERY, DEC. 1994, PP. 87–96. DOI: 10.1145/193173.195297.
- [11] B. JACOBS (EDITOR), *VERIFAST 21.04*. (APR. 18, 2021). ZENODO. DOI: 10.5281/ZENODO.4705416.
- [12] N. MOMMEN AND B. JACOBS, “VERIFICATION OF C++ PROGRAMS WITH VERIFAST,” DEC. 28, 2022, *ARXIV*: ARXIV:2212.13754. DOI: 10.48550/ARXIV.2212.13754.



**4to Congreso Virtual
de Microcontroladores y
sus Aplicaciones**

Universidad Tecnológica Nacional
Facultad Regional Paraná
Departamento de Ingeniería Electrónica
Laboratorio de Técnicas Digitales