# Congreso Argentino de Sistemas Embebidos



CASE

## LIBRO DE TRABAJOS

## 2026

6 y 7 de agosto – San Rafael, Mendoza

CONICET

UTN SAN RAFAEL

ISBN 978-631-90145-4-9

ACSE

RUSE Red Universitaria de Sistemas Embebidos

# CASE 2026

## Libro de Trabajos

**Modalidades Artículo, Foro Tecnológico y Reporte**

Congreso Argentino
de
Sistemas Embebidos

6 y 7 de agosto de 2026

CONICET

UTN SAN RAFAEL

ACSE

RUSE Red Universitaria de Sistemas Embebidos

ISBN 978-631-90145-4-9

Congreso Argentino de Sistemas Embebidos : CASE 2026 : libro de trabajos / Cebedio, Celeste / Antonelli, Maximiliano / Brengi, Diego / De Micco, Luciana / Zacchigna, Federico G./
1a ed. - Ciudad Autónoma de Buenos Aires : ACSE – Asociación Civil para la Investigación, Promoción y Desarrollo de los Sistemas Electrónicos Embebidos, 2026.
Libro digital, PDF

Archivo Digital: descarga y online
ISBN 978-631-90145-4-9

1. Circuitos Electrónicos. 2. Hardware. 3. Software. I. Cebedio, Celeste
CDD 600

Fecha de catalogación: 25/08/2026

Libro de Trabajos
Modalidades Artículo, Foro Tecnológico y Reporte
**Congreso Argentino de Sistemas Embebidos – CASE 2026**

**Editores:**

| | |
|---|---|
| Cebedio, Celeste | UNMDP/ICYTE |
| Antonelli, Maximiliano | UNMDP/ICYTE/CONICET |
| Brengi, Diego | INTI/UNLaM/FIUBA |
| De Micco, Luciana | UNMDP/ICYTE/CONICET |
| Zacchigna, Federico G. | FIUBA |

Copyright © 2026
Asociación civil para la investigación, promoción y desarrollo de los sistemas electrónicos embebidos.

Se otorga permiso para copiar y redistribuir este libro de trabajos, siempre que se mantengan los mensajes de copyright y la autoría de la obra y sus partes.

# Prefacio

El diseño de sistemas embebidos es un motor clave de la industria y del desarrollo científico y tecnológico, y es un campo que en los últimos años ha crecido notablemente en la región, tanto en la academia como en la industria.

El CASE (Congreso Argentino de Sistemas Embebidos) fomenta la presentación formal de trabajos usando el modelo de revisión de pares para garantizar la calidad en la presentación de los mismos. Este congreso se realiza en forma simultánea al SASE (Simposio Argentino de Sistemas Embebidos), un evento anual que reúne a la comunidad académica y a la industria en torno a los sistemas embebidos, buscando fomentar esta temática.

El CASE 2026 se realizó en modalidad presencial el 6 y 7 de agosto, en la Facultad Regional San Rafael, Universidad Tecnológica Nacional, San Rafael, Mendoza, Argentina, organizado por la UTN-FRSR.

Los objetivos que persigue el congreso son:

- Ofrecer un lugar de encuentro para investigadores y becarios de Latinoamérica, fomentando la colaboración.
- Difundir en el medio académico los adelantos científicos y tecnológicos producidos a nivel mundial.
- Propiciar la presentación y discusión de trabajos de investigación desarrollados en Argentina y en la región.
- Estimular en los estudiantes universitarios avanzados el interés por la investigación en el área de los sistemas embebidos.
- Coordinar y actualizar los contenidos de sistemas embebidos de los programas de grado y posgrado de las universidades latinoamericanas.

Las áreas temáticas del CASE se organizan de la siguiente manera:

- Aplicaciones agro y alimentos
- Arquitectura de procesadores
- Bioingeniería
- DSPs
- FPGAs, HDLs y ASICs
- Hardware Abierto
- Implementación de SE
- Inteligencia Artificial en SE
- Internet de las Cosas
- Linux Embebido
- Protocolos y Comunicaciones
- Robótica

- RTOS
- Software Embebido

Dentro de cada una de estas áreas se permiten las modalidades Artículo, Foro Tecnológico y Reporte, según el tipo de trabajo.

Los trabajos presentados al CASE fueron sometidos a un proceso de revisión por pares doble ciego, corrección y *rebuttal*. De este modo fueron seleccionados 15 trabajos en la modalidad Artículo, 20 en modalidad Foro Tecnológico y 4 en la modalidad Reporte, dando un total de 39 trabajos aceptados de un total de 46 trabajos presentados.

Esta publicación se encuentra también disponible en forma *online* en la página web: https://case.ar/

Esperamos que los trabajos recopilados en esta memoria sean de su interés y contamos con su participación en futuras ediciones del evento.

Atentamente,

**Comité Permanente CASE**

## Auspiciantes

- Marvell
- CONICET (Consejo Nacional de Investigaciones Científicas y Técnicas)
- CTDR Los Reyunos (Centro Tecnológico de Desarrollo Regional)
- Outsource
- IEEE CAS (Circuits and Systems Society)
- Rama IEEE de la Universidad Nacional de Mar del Plata
- INDIE Semiconductor
- Municipalidad de San Rafael

## Entidades organizadoras

- Universidad Tecnológica Nacional, Facultad Regional San Rafael
- CONICET (Consejo Nacional de Investigaciones Científicas y Técnicas)
- RUSE (Red Universitaria de Sistemas Embebidos)
- ACSE (Asociación Civil para la Investigación, Promoción y Desarrollo de los Sistemas Electrónicos Embebidos)

## Instituciones a las que pertenecen los miembros del comité permanente

- CONICET (Consejo Nacional de Investigaciones Científicas y Técnicas)
- FIUBA (Facultad de Ingeniería Universidad de Buenos Aires)
- INTI (Instituto Nacional de Tecnología Industrial)
- UNLaM (Universidad Nacional de La Matanza)
- UNMDP (Universidad Nacional de Mar del Plata)
- ICYTE (Instituto de Investigaciones Científicas y Tecnológicas en Electrónica)

## Comité organizador estable

- Dr. Maximiliano Antonelli (UNMDP/ICYTE/CONICET)
- Mg. Diego Brengi (INTI/UNLaM/FIUBA)
- Ing. Celeste Cebedio (UNMDP/ICYTE)
- Dra. Luciana De Micco (UNMDP/ICYTE/CONICET)
- Dr. Federico G. Zacchigna (FIUBA)

## Comité organizador local

- Jimena Lloret (UTN FRSR)
- Gustavo Mercado (UTN FRM)
- Franco Martínez (UTN FRSR)
- Pablo Moliterno (UTN FRSR)

## Chairs temáticos

| | |
|---|---|
| • Aplicaciones agro y alimentos: | Dr. Nicolás Calarco (UCU) |
| • Bioingeniería: | Dra. Mariela Azul Gonzalez (UNMDP) |
| • FPGA, HDL y ASIC: | Ing. Salvador Tropea (INTI) |
| • Implementación de Sistemas Embebidos: | Ing. Lucas Rabioglio (UNMDP) |
| • Software Embebido: | Dr. Maximiliano Antonelli (UNMDP) |
| • Inteligencia Artificial: | Dra. Luciana De Micco (UNMDP) |
| • IoT (Internet of Things): | Dr. Ariel Lutenberg (FIUBA)<br>Esp. Claudio Omar Biale (UNAM) |
| • Protocolos y comunicaciones: | Ing. Ignacio Zaradnik (UNLAM) |

## Revisores

Alcalde, Fabricio
Alessandrini, Gustavo
Alvarez, Gonzalo
Alvarez, Nicolas
Antonelli, Maximiliano
Arizaga-Silva, Juan Antonio
Arnaldi, Luis
Arnone, Leonardo
Arzamendia, Mario
Avalos Ribas, Ramiro
Barranco Gutiérrez, Alejandro Israel
Biale, Claudio
Burgos, Enrique Sergio
Carnaghi, Marco
Cazarez Castro, Nohe Ramón
Corzi, Damian Leonel
David de Lima, Dario Manuel
De Micco, Luciana
Echarri, Martín
Ferreyra, Pablo Alejandro
Filomena, Eduardo
Funes, Marcos
Gaia, Jeremías
Galasso, Christian
Galetto, Agustín
Galleguillo, Juan
Gallina, Sergio Hilario
Garcia Inza, Mariano
Ghignone, Ramiro
Gluzman, Brian
Gómez Berisso, Mariano
Gómez López, María De Los Ángeles
Gonzalez, Mariela Azul
Gonzalez Dondo, Diego
Guerrero Moral, Gregorio Ariel
Heredia, Martín
Irrazabal, Emanuel
Leiva, Lucas
Lipovetzky, José
Lopez, Cristian
Lopresti, Raúl
Lozada Castillo, Norma Beatriz
Lutenberg, Ariel
Luviano Juarez, Alberto
Maldonado Robles, Yazmin
Marchi, Edgardo
Márquez, Daniel
Martos, Pedro
Mateos, Horacio
Medel, Ricardo
Medina, Matias
Monte, Gustavo
Oliva, Rafael
Ordoñez, Cristian
Paez, Francisco E.
Pastafiglia, Daniel
Perez Paina, Gonzalo F.
Perrone Orsi, Juan Ignacio
Pucheta, Julian
Rabioglio, Lucas
Regueiro-Gómez, Angel
Rey Vega, Leonardo
Ridolfi, Pablo
Rodriguez, Ezequiel Dario
Rossi, Silvano
Ruge Ruge, Ilber Adonayt
Russo, Rodrigo Ezequiel
Skrauba, Axel Alfredo
Urbano Pintos, Nicolás
Volentini, Esteban
Zacchigna, Federico
Zapirain, Esteban
Zerbini, Carlos Alberto

# Índice de autores


ACOSTA ACOSTA, Juan David : 66
ADAME NAJERA, Erick Zain : 45
AGUAYO, Jesus : 27
AGUAYO ALQUICIRA, Jesús : 45
AGUILA WHARTON, Alexander : 31
ALY, Mokhtar : 28
AMEZQUITA ALFONSO, Santiago : 66
ANTONELLI, Maximiliano : 8, 9, 40
ARANDA, Marcos Darío : 123
AVALOS, Juan : 28
AVALOS RIBAS, Ramiro : 40
BALIBREA LASTRE, Yosel de Jesús : 49
BARRETO, Gastón : 31
BATALLA, Nicolás : 40
BELTRAMINI, Paola Inés : 123
BERNARDI, Emanuel : 62
BURGOS, Patricio : 28
CABRAL, Rocío : 2, 20
CAÑADAS, Gabriel : 12, 105
CANO, Jesús Eduardo : 123
CASTAGNOLA, Juan : 128
CATALANO, Juan F. : 40
CEBEDIO, Maria Celeste : 7, 8, 9, 40, 87
COMELLI, Román : 29
CONTRERAS, Matías : 16
CORREA, Fabricio : 70
CORTI, Rosa Marina : 83
DE LEON, Susana : 27
DE LEÓN ALDACO, Susana Estefany : 45
DE MICCO, Luciana : 7, 8, 9, 40, 87
DÍAZ, Eliseo : 16
DUARTE, Luis Miguel : 101
FERNÁNDEZ, Agustina : 31
FERRANDO, Sofía : 74
FIGUEROA, Sergio David : 133
FLIGER, Elias Sebastián : 16
FLORES HERNÁNDEZ, Diego A. : 11
FRANCHI, Maria Luisa : 31
GAIA, Jeremías : 12, 105
GALLARDO, Fernando : 35
GALLEGUILLO, Juan : 101
GALLINA, Sergio Hilario : 123
GANEM, Elías : 58
GARECA, Martín : 58
GIRON, Juan Manuel : 105
GOMEZ, Mauro Gastón : 78, 133
GRELLET MARTINOIA, Alejandro : 119
GUTIÉRREZ FRÍAS, Octavio : 11
HERNANDEZ GOMEZ, Oscar Mauricio : 66
JAIME, Ibrahim : 62
JIMÉNEZ LÓPEZ, Fabian Rolando : 96
JUAREZ, Ivan : 27
JUÁREZ TRUJILLO, Ivan Antonio : 45
JUNCO, Sergio : 29
LANCIONI, Walter : 128
LEIVA, Lucas : 6
LLUIS, Valentina : 74
LOPRESTI, Raúl E. : 9, 40
LORUSSO, Martín : 40
MÁRQUEZ, Daniel : 54, 92, 110
MARTINEZ, Roberto Manuel : 83
MEDINA, Tomás : 6
MENTASTI, Luciana : 31
MIRETTI, Marco L. : 62
MONETTI, Gennaro : 70
MONTILLA, Alejo : 12
MORÁN FABRA, Ruben David : 16
MORENO, Juan Pablo : 123
MOZO, Demian N. : 87
MUSA, Ezequiel Alberto : 119
NAPPA, Agustina : 74
OREGGIONI, Julián : 2, 20, 70, 74
OROSCO, Eugenio : 12, 105
ORTIZ ALVAREZ, Ingrid Carolina : 96
PATARROYO GUTIERREZ, Luis David : 66
PETRASHIN, Pablo : 35, 128
PETRASHIN, Pedro Nicolas : 128
PROTTI COSENZA, Luciano : 31
RABIOGLIO, Lucas : 7, 8, 9, 40, 87
REALE, Cesar : 35
RIVA, Guillermo Gaston : 22
ROA, Andrea : 114
RODRIGUEZ, Ezequiel Dario : 7, 9
RODRIGUEZ, José : 28
RODRIGUEZ, Luciano Ariel : 78
ROMERO, Raúl : 31
ROSSI, Silvano : 31
ROSSO, Daniel : 22
RUGE RUGE, Ilber Adonayt : 66, 96, 114
RUIZ, Felipe : 28
RUIZ TATUR, Manuel : 78
SAADE, Sergio Daniel : 119
SAL MOYANO, M. Paz : 40
SALIBE, Martin : 16
SÁNCHEZ, Giovanny : 28
SÁNCHEZ GIL, Juan Camilo : 96
SANCHEZ URIZA, Edwin Javier : 96
SAÑUDO, Juan Manuel : 4
SCANDURA, Ignacio : 31
SERÉ, Andrés : 2
SIERRA, Caren : 114
SPENNATO NARVAEZ, Agustín : 8

STEINFELD, Leonardo : 2, 20
TARCETTI, Ana Evelina Yael : 16
TORRES CERVANTES, Angel A. : 11
VASQUEZ, Angel : 28
VÁZQUEZ, Martín : 6
VAZQUEZ, Rodrigo : 14
VECCHIO, Juan Pablo : 83
VELASQUEZ, Guadalupe : 27
VERSTRAETE, Enzo Gabriel : 133
VILLAGRÁN, Luis Daniel : 123
VITULICH, Carlos Alberto : 78
VOLENTINI, Esteban Daniel : 58, 119
WAGNER, Walter : 54
ZACCHIGNA, Federico G. : 14
ZERBINI, Carlos Alberto : 22
ZULIANI, Agustín : 54, 92, 110

# Índice de trabajos


Introducción I

Índice de trabajos X

Artículo 1

Aplicaciones agro y alimentos . . . 1

**Implementation and Characterization of FOTA for Extensive Livestock IoT Devices**, *Rocío Cabral; Andrés Seré; Leonardo Steinfeld; Julián Oreggioni* . . . 2

Bioingeniería . . . 3

**Desarrollo del Firmware para el HiCCE-128 en All Programmable System on Chip**, *Juan Manuel Sañudo* . . . 4

FPGAs, HDLs y ASICs . . . 5

**Implementación Eficiente en FPGA de Corrección de Distorsión Radial en Imágenes**, *Lucas Leiva; Martín Vázquez; Tomás Medina* . . . 6

**Análisis Comparativo de Generadores Pseudoaleatorios y Caóticos en Sistemas ADC Time-Interleaved Aleatorios**, *Ezequiel Rodriguez; Lucas A. Rabioglio; M. Celeste Cebedio; Luciana De Micco* . . . 7

**Plataforma FPGA para Evaluación Experimental de AES con Bloques Criptográficos Caóticos**, *Agustín Spennato Narvaez; Lucas Rabioglio; Maria Celeste Cebedio; Maximiliano Antonelli; Luciana De Micco* . . . 8

**Estimación de Parámetros de Ruido de Middleton Clase A mediante Algoritmos Genéticos en FPGA**, *Ezequiel Dario Rodriguez; Lucas Rabioglio; Maximiliano Antonelli; Luciana De Micco; Raúl Lopresti; Maria Celeste Cebedio* . . . 9

Implementación de SE . . . 10

**Embedded Open-Architecture Device for Solar Tracking Applications: Dual-Axis Control and Environmental Characterization**, *Ángel A. Torres Cervantes; Diego A. Flores Hernández; Octavio Gutiérrez Frías* . . . 11

**Strivex: Sensado Biomecánico en Tiempo Real mediante VBT con Encoder Lineal y ESP32-S3**, *Alejo Montilla; Jeremias Gaia; Eugenio Orosco; Gabriel Cañadas* . . . 12

Inteligencia Artificial en SE . . . 13

**MoodLink: An Embedded Wearable for On-Device Speech Emotion Recognition**, *Rodrigo Vázquez; Federico G. Zacchigna* . . . 14

## Internet de las Cosas . . . 15

**Nodos IIoT reconfigurables para la gestión segura de pellets de girasol en la industria aceitera**, *Elias Sebastián Fliger; Ana Evelina Yael Tarcetti; Matias Contreras; Eliseo Díaz; Rubén David Morán Fabra; Martin Salibe* . . . 16

**State-Based Energy Estimation for Zephyr RTOS in NB-IoT**, *Rocío Cabral; Julián Oreggioni; Leonardo Steinfeld* . . . 20

## Protocolos y Comunicaciones . . . 21

**Customizing NEORV32 for Matched Filtering in SDR Applications**, *Daniel Rosso; Carlos Alberto Zerbini; Guillermo Gaston Riva* . . . 22

## Software Embebido . . . 26

**SoC-FPGA Based Acquisition and Clarke Preprocessing of Three-Phase Currents**, *Ivan Juarez; Susana De Leon; Guadalupe Velasquez; Jesus Aguayo* . . . 27

**Data-Reuse RLS Based Model-Free Predictive Control for Voltage Source Inverters**, *Felipe Ruiz; Angel Vázquez; Patricio Burgos; Mokhtar Aly; Juan Avalos; Giovanny Sánchez; José Rodriguez* . . . 28

**Framework orientado a eventos con período dinámico aplicado en electrónica de potencia**, *Román Comelli; Sergio Junco* . . . 29

# Foro Tecnológico **30**

## Aplicaciones agro y alimentos . . . 30

**Sistema para la medición, registro y control de variables asociadas al cultivo interior de cannabis**, *Ignacio Scandura; Alexander Aguila Wharton; Luciano Protti Cosenza; Luciana Mentasti; Raúl Romero; Agustina Fernández; Maria Luisa Franchi; Silvano Rossi; Gastón Barreto* . . . 31

**Arquitectura Embebida Modular para Gestión de Energía Renovable: Diseño, Implementación y Evolución de un Sistema Autónomo Multipropósito**, *Pablo Petrashin; César Reale; Fernando Gallardo* . . . 35

## Bioingeniería . . . 39

**Compensación acústica para la validación de experimentos comportamentales en acuarios**, *Nicolás Batalla; Juan Catalano; Celeste Cebedio; Martín Lorusso; Lucas Rabioglio; Maximiliano Antonelli; Raúl Lopresti; Luciana De Micco; Paz Sal Moyano; Ramiro Avalos* . . . 40

## FPGAs, HDLs y ASICs . . . 44

**Digital Architecture for the Generation of Trigger Signals Applied to Dual Inverters**, *Erick Zain Adame Najera; Jesùs Aguayo Alquicira; Susana Estefany De Leòn Aldaco; Ivan Antonio Juárez Trujillo* . . . 45

**Receptor SDR sobre FPGA con Linux Embebido**, *Yosel de Jesús Balibrea Lastre* . . . 49

## Implementación de SE . . . 53

**Placa Shield MCXC444 DSI para el kit de desarrollo FRDM-MCXC444**, *Daniel Márquez; Agustín Zuliani; Walter Wagner* . . . 54

**Diseño e implementación de una placa didáctica basada en RISC-V para una asignatura de grado**, *Esteban Daniel Volentini; Elías Ganem; Martín Gareca* . . . 58
**Estimación de Posición, Orientación y Velocidad de Bajo Costo, para un Vehículo Autónomo**, *Ibrahim Jaime; Marco L. Miretti; Emanuel Bernardi* . . . 62
**Sistema Embebido para Monitoreo de Variables Eléctricas, Térmicas y Ambientales Aplicadas al Diagnóstico de Baterías de Segunda Vida**, *Ilber Adonayt Ruge Ruge; Juan David Acosta Acosta; Santiago Amezquita Alfonso; Oscar Mauricio Hernandez Gomez; Luis David Patarroyo Gutierrez* . . . 66
**Low Resources, Low Power, Real Time Multiplayer: A DDR Inspired Wearable Game**, *Fabricio Correa; Gennaro Monetti; Julián Oreggioni* . . . 70
**Ultra Low Power Solar Panel Testbench for IoT Devices Under Dynamic Partial Shading Conditions**, *Sofía Ferrando; Valentina Lluis; Agustina Nappa; Julián Oreggioni* . . . 74
**Diseño y desarrollo de sistema embebido para estaciones de monitoreo de lagos**, *Luciano Ariel Rodriguez; Mauro Gastón Gómez; Manuel Ruiz Tatur; Carlos Alberto Vitulich et al.* . . . 78

## Inteligencia Artificial en SE . . . 82

**Diseño e Implementación de un Núcleo VHDL Genérico para Redes Neuronales Artificiales en SoC**, *Juan Pablo Vecchio; Rosa Corti; Roberto Martinez* . . . 83
**Detección de Merluza en Sonar con YOLO para Sistemas Embebidos**, *Demián N. Mozo; Lucas Rabioglio; Maria Celeste Cebedio; Luciana De Micco* . . . 87

## Internet de las Cosas . . . 91

**Placa de desarrollo para IoT Industrial basada en el microcontrolador ESP32**, *Agustín Zuliani; Daniel Márquez* . . . 92
**Nodo IoT para Monitoreo Inteligente de ruido y Localización de Fuente de contaminación sonora en Zonas Urbanas Residenciales**, *Ilber Adonayt Ruge Ruge; Juan Camilo Sánchez Gil; Ingrid Carolina Ortiz Alvarez; Fabian Rolando Jiménez Lopez; Edwin Javier Sanchez Uriza* . . . 96

## Protocolos y Comunicaciones . . . 100

**Alta Disponibilidad y Resiliencia en Redes SD-WAN**, *Luis Miguel Duarte; Juan Galleguillo* . . . 101
**Arquitectura de Laboratorio Remoto para Sistemas Embebidos MSP430FR4133**, *Juan Manuel Giron; Jeremias Gaia; Eugenio Orosco; Gabriel Cañadas* . . . 105

## Software Embebido . . . 109

**Diseño de un Generador de Formas de Onda Arbitrarias Multi-canal basado en DDS y Sistemas Distribuidos**, *Agustín Zuliani; Daniel Márquez* . . . 110
**Sistema Embebido Multisensorial para Monitoreo en Tiempo Real de Variables Acústicas, Vibracionales y Térmicas en Motores**, *Andrea Roa; Caren Sierra; Ilber Ruge Ruge* . . . 114

# Reporte **118**

## Inteligencia Artificial en SE . . . 118

**Diseño e implementación de red neuronal neuromórfica**, *Alejandro Grellet Martinoia; Esteban Daniel Volentini; Ezequiel Alberto Musa; Sergio Daniel Saade* . . . 119

**Optimización Energética de Nodos IoT con GPS a través de Modelos de Aprendizaje Automático**, *Juan Pablo Moreno; Marcos Darío Aranda; Sergio Hilario Gallina; Paola Inés Beltramini; Jesús Eduardo Cano; Luis Daniel Villagrán* . . . . . 123

**Internet de las Cosas** . . . . . 127

**Arquitectura de Nodo IoT para Monitoreo de Suelos: Front-End Analógico con Conversión Tensión-Frecuencia en 130 nm CMOS versus ADC Convencional**, *Pedro Nicolas Petrashin; Juan Castagnola; Walter Lancioni; Pablo Petrashin* . . . . . 128

**Software Embebido** . . . . . 132

**Pre-Diseño del Controlador de Vuelo para un UAV**, *Enzo Gabriel Verstraete; Mauro Gastón Gómez; Sergio David Figueroa* . . . . . 133

Categoría

# Artículo

Aplicaciones agro y alimentos

Área Temática

# Implementation and Characterization of FOTA for Extensive Livestock IoT Devices

Rocío Cabral, Andrés Seré, Leonardo Steinfeld, and Julián Oreggioni,
Universidad de la República, Montevideo, Uruguay
Email: {rcabral, asere, leo, juliano}@fing.edu.uy

***Abstract*—Firmware Over-The-Air (FOTA) updates are a key requirement for the deployment of Internet of Things (IoT) devices in extensive livestock systems, where physical access to devices is limited and maintenance operations must be minimized. This work presents the integration, implementation, and experimental characterization of an Amazon Web Services (AWS) IoT Core based FOTA system within the OVIS platform, a cloud-connected wearable solution for sheep behavior monitoring. The proposed approach combines AWS cloud services, Zephyr real-time operating system, and a dual slot bootloader architecture to enable reliable remote firmware updates under constrained connectivity conditions.**

**Laboratory tests were conducted to evaluate firmware update duration and energy consumption under varying signal strengths. Results show that FOTA updates are completed in less than two minutes, with stable execution times across different conditions, while degraded signal quality leads to increased energy consumption. All FOTA updates were successfully completed without requiring a restart of the download process, demonstrating robustness to intermittent connectivity. Field deployment in a real research scenario further validated the system, where multiple FOTA updates were performed without physical access to the devices.**

***Index Terms*—remote firmware update, Internet of Things, IoT, CAT-M1, embedded systems, low-power devices**

## Acknowledgment

The authors would like to thank R. Ungerfeld, J. Pérez, and H. Cardoso. This work was partially funded by CSIC (Comisión Sectorial de Investigación Científica, Udelar), EI (Espacio Interdisciplinario, Udelar, Uruguay), ANII (Agencia Nacional de Investigación e Innovación, Uruguay) and IEEE Circuits and Systems Society (Uruguayan Chapter).

## References

[1] V. Cabrera, A. Delbuggio, H. Cardoso, D. Fraga, A. Gómez, M. Pedemonte, R. Ungerfeld, and J. Oreggioni, "Harnessing technology for livestock research: An online sheep behavior monitoring system," *IEEE Transactions on AgriFood Electronics*, vol. 2, no. 2, pp. 306–313, 2024.

[2] ——, "Research platform to study sheep behavior," in *IEEE Conference on AgriFood Electronics (CAFE)*, 2023, pp. 60–64.

[3] K. Arakadakis, P. Charalampidis, A. Makrogiannakis, and A. Fragkiadakis, "Firmware over-the-air programming techniques for IoT networks: a survey," *ACM Computing Surveys*, vol. 54, no. 9, pp. 1–36, 2021.

[4] Open Mobile Alliance, "Lightweight machine to machine technical specification: Core. Approved version: 1.2.2," San Diego, USA, 2024.

[5] B. Moran, H. Tschofenig, D. Brown, and M. Meriac, "RFC9019: A firmware update architecture for internet of things," Internet Engineering Task Force, 2021.

[6] K. Zandberg, K. Schleiser, F. Acosta, H. Tschofenig, and E. Baccelli, "Secure firmware updates for constrained IoT devices using open standards: A reality check," *IEEE access*, vol. 7, pp. 71 907–71 920, 2019.

[7] F. Mahfoudhi, A. K. Sultania, and J. Famaey, "Over-the-air firmware updates for constrained nb-iot device," *Sensors*, vol. 22, no. 9, 2022.

[8] C. Nicolas, B. Naila, and R.-C. Amar, "Energy efficient firmware over the air update for TinyML models in LoRaWAN agricultural networks," in *International Telecommunication Networks and Applications Conference*, 2022, pp. 21–27.

[9] L. Formanek, M. Kubascik, O. Karpis, and P. Kolok, "Advanced system for remote updates on ESP32-based devices using over-the-air update technology," *Computers*, vol. 14, no. 12, 2025.

[10] Nordic Semiconductor. (2026, May) AWS FOTA. https://docs.nordicsemi.com/bundle/ncs-3.2.3/page/nrf/libraries/networking/aws_fota.html. [On line].

[11] Trusted Firmware Projects. (2026, May) MCUboot. https://www.trustedfirmware.org/projects/mcuboot. [On line].

[12] R. Cabral, H. Cardoso, R. Ungerfeld, and J. Oreggioni, "OVIS farm platform: from controlled experiments to autonomous field deployment," in *IEEE Conference on AgriFood Electronics (CAFE)*, 2026.

Categoría

Artículo

Bioingeniería

Área Temática

# Desarrollo del Firmware para el HiCCE-128 en All Programmable System on Chip

Juan Manuel Sañudo
Universidad Nacional de San Juan, Facultad de Ingeniería
San Juan, Argentina
sanudojuanmanuel.77@gmail.com

***Resumen*—Este trabajo presenta el diseño e implementación del firmware completo para el HiCCE-128, una plataforma de adquisición de biopotenciales de 128 canales desarrollada por el ICTP y la Universidad Rockefeller. El firmware se implementa en un APSoC Zynq-7000, sobre la placa de evaluación y desarrollo ZedBoard, donde se aprovecha la partición del sistema entre lógica programable (PL) y sistema de procesamiento (PS). En la PL, se diseñaron en VHDL máquinas de estado finito que controlan la interfaz digital de cuatro amplificadores Intan RHA2132, que incorpora 32 canales, y cuatro ADC AD7982 de 18 bits, con una frecuencia de muestreo de 480 KSPS por conversor, dando una frecuencia de muestreo de 15 kHz por canal. El PS ejecuta FreeRTOS con el stack de red lwIP para la transmisión de datos por TCP/IP hacia un puente del protocolo Lab Streaming Layer (LSL), habilitando la compatibilidad con software de neurociencia como Medusa BCI o NeuroPype. Se diseñó y construyó una jaula de Faraday para disminuir interferencia electromagnética que impedía realizar las pruebas de validación con señales reales. Los resultados demuestran adquisición continua y sin pérdida de muestras, con reconstrucción verificada de señales sinusoidales y ECG del orden del milivolt. La integración al protocolo LSL es exitosa y permite el uso del HiCCE con software de terceros para visualización y procesamiento de señales.**

***Palabras clave*: APSoC, Zynq-7000, biopotenciales, FreeRTOS, lwIP, LSL, VHDL, HiCCE, adquisición multicanal.**

## AGRADECIMIENTOS

Se agradece a todos los docentes de la UNSJ y su dedicación diaria, al Departamento de Electrónica y Automática y el Instituto de Bioingeniería por el espacio brindado. En particular a los directores de este TFG, Cristian Sisterna y Juan Pablo Graffigna. Por último, al ICTP por proporcionar el HiCCE, lo que significó una oportunidad única de trabajar y aprender sobre un desarrollo de avanzada.

## REFERENCIAS

[1] "High Performance 128-Channel Acquisition System for Electrophysiological Signals," *IEEE Access*, 2020.
[2] L. Crockett, R. Elliot, M. Enderwitz, y R. Stewart, *The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC*. University of Strathclyde, 2014.
[3] R. Barry, *Using The FreeRTOS Real Time Kernel. A Practical Guide*. 2010.
[4] A. Dunkels, "Design and Implementation of the lwIP TCP/IP Stack," Swedish Institute of Computer Science, 2001.
[5] Intan Technologies, "RHA2000 Series Amplifier Arrays," 2010.
[6] Analog Devices, "AD7982: 18-Bit, 1 MSPS PulSAR ADC," 2020.
[7] C. Kothe, "LabStreamingLayer's Documentation. User's Guide," 2019. [https://github.com/sccn/labstreaminglayer]
[8] G. Minguez *et al.*, "MEDUSA©: A Novel Python-based Software Ecosystem to Accelerate Brain-Computer Interface and Cognitive Neuroscience Research," *Computer Methods and Programs in Biomedicine*, vol. 230, 2023.
[9] V. A. Pedroni, *Finite State Machines in Hardware*. The MIT Press, 2013.
[10] Xilinx, "PG021: AXI DMA v7.1. LogiCORE IP Product Guide," 2024.
[11] Xilinx, "PG080: AXI4-Stream FIFO v4.3. LogiCORE IP Product Guide," 2023.
[12] H. W. Ott, *Noise Reduction Techniques in Electronic Systems*, 2nd ed. John Wiley & Sons, 1988.

Categoría

Artículo

FPGAs, HDLs y ASICs

Área Temática

ISBN 978-631-90145-4-9 - Congreso Argentino de Sistemas Embebidos CASE 2026 - 6 y 7 de agosto

# Implementación Eficiente en FPGA de Corrección de Distorsión Radial en Imágenes

Lucas Leiva, Martín Vázquez
*LabSET - INTIA*
*UNICEN*
Tandil, Argentina
{lleiva,mvazquez}@labset.exa.unicen.edu.ar

Tomás Medina
*Departamento de Computación y Sistemas*
*UNICEN*
Tandil, Argentina
tmedina@exa.unicen.edu.ar

***Resumen*— La corrección de la distorsión radial constituye una etapa fundamental en sistemas de visión por computadora, particularmente en aplicaciones embebidas donde el uso de lentes de gran campo de visión es frecuente. Los métodos tradicionales, basados en mapas de corrección precalculados, presentan elevados requerimientos de memoria que dificultan su implementación en plataformas FPGA con recursos limitados, y los métodos basados en modelos matemáticos presentan alta complejidad computacional. En este trabajo se propone un método alternativo basado en la aproximación de las funciones de corrección mediante vectores precalculados. La estrategia consiste en modelar las curvas de remapeo a partir de una curva base, junto con factores de escala y términos de desplazamiento, permitiendo reconstruir el mapeo geométrico con un costo computacional reducido. Los resultados experimentales muestran que el método alcanza un error promedio de 1.11 píxeles, logrando una reducción significativa del uso de memoria, de 16 MB a 15 kB para imágenes de resolución 1080p. Además, la baja permite su ejecución en tiempo real.**

***Palabras clave*—Procesamiento de imágenes, distorsión radial, fpga.**

## REFERENCIAS

[1] D. Brown, "Decentering distortion of lenses," *Photogrammetric Engineering*, vol. 32, no. 3, pp. 444–462, 1996.

[2] K. V. Asari, "Design of an efficient VLSI architecture for non-linear spatial warping of wide-angle camera images," *Journal of Systems Architecture*, vol. 50, no. 12, pp. 743–755, 2004.

[3] B. Erdnüß, "A review of the one-parameter division undistortion model," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 1, pp. 89–96, 2021.

[4] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[5] H. Ngo, A Pipelined Architecture for Real Time Correction of Barrel Distortion, M.S. thesis, Old Dominion University, Norfolk, VA, USA, 2003.

[6] P. Y. Chen et al., "A VLSI implementation of barrel distortion correction for wide-angle camera images," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 1, pp. 51–55, Jan. 2009.

[7] H. Blasinski, W. Hai, and F. Lohier, "FPGA architecture for real-time barrel distortion correction of colour images," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, Barcelona, Spain, 2011, pp. 1–6.

[8] J. Cłapa, et al., "A fisheye distortion correction algorithm optimized for hardware implementations," in *Proc. 21st International Conference Mixed Design of Integrated Circuits and Systems (MIXDES)*, Lublin, Poland, 2014, pp. 247–252.

[9] C.-C. Huang, et al., "A low-complexity scaling scheme for barrel distortion correction," in *Proc. Sixth International Conference on Genetic and Evolutionary Computing (ICGEC)*, Kitakushu, Japan, 2012, pp. 355–358.

[10] R. Ou, et al., "Real-time image distortion correction based on FPGA," in *Proc. Fourteenth International Conference on Digital Image Processing (ICDIP 2022)*, vol. 12342, SPIE, 2022.

[11] S. Yang, et al., "Progressively complementary network for fisheye image rectification using appearance flow," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021, pp. 6348–6357.

[12] D. S. Kaputa and K. A. Derhak, "Model based design of a real time FPGA-based lens undistortion and image rectification algorithm for stereo imaging," *IEEE Access*, vol. 11, pp. 18972–18982, 2023.

[13] T. Medina, M. Vázquez, and L. Leiva. "Sistema de inspección de defectos en baldosas cerámicas implementado en FPGA." *Revista Elektron*, vol 6, no. 1, pp. 1-7, 2022.

# Análisis Comparativo de Generadores Pseudoaleatorios y Caóticos en Sistemas ADC Time-Interleaved Aleatorios

Ezequiel Rodriguez, Lucas A. Rabioglio, M. Celeste Cebedio, Luciana De Micco
ICYTE - Facultad de Ingeniería, UNMdP - CONICET
Argentina

***Resumen*—En este trabajo se presenta un análisis experimental del impacto de distintos generadores pseudoaleatorios sobre el desempeño en el rango dinámico libre de espurias (SFDR, Spurious-Free Dynamic Range) en arquitecturas de conversión analógica-digital Time-Interleaved con selección aleatoria de canales. A partir de una plataforma previamente desarrollada, se evalúan mapas caóticos, generadores clásicos y generadores basados en librerías de lenguaje de alto nivel.**

**Los resultados muestran que no solo el tipo de generador influye en el desempeño espectral, sino que, en el caso de los mapas caóticos, el método de extracción de bits resulta determinante. Asimismo, se analiza la entropía de bloque de las secuencias generadas, observándose que valores cercanos a la unidad se asocian con un mejor desempeño en SFDR.**

**Finalmente, se evidencia una fuerte correlación entre el cumplimiento de ensayos estadísticos, la entropía de bloque y la mejora en el desempeño espectral del sistema.**

***Palabras clave*—PRNG (Pseudo-Random Number Generator), caos, ADC, time-interleaving, SFDR, FPGA (Field Programmable Gate Array).**

## I. Agradecimientos

Este trabajo fue financiado por el CONICET (PIP 2025–2027, Subsidio N.º 11220250100447CO) y por el Fondo para Reparación y Actualización de Equipamiento Científico 2025 de la Fundación Williams.

# Plataforma FPGA para Evaluación Experimental de AES con Bloques Criptográficos Caóticos

Agustín Spennato Narvaez[‡], L. A. Rabioglio[†‡], M. C. Cebedio[†‡], M. Antonelli [†‡§] L. De Micco[†‡§]
*† ICYTE, ‡ FI–UNMdP, § CONICET*
agusspennato@gmail.com, {lucas.rabioglio, celestecebedio, maxanto, ldemicco}@fi.mdp.edu.ar

***Resumen*—Se presenta la implementación en FPGA de una plataforma experimental basada en Advanced Encryption Standard (AES) con bloques internos reconfigurables. El diseño permite reemplazar dinámicamente el bloque de sustitución no lineal (S-box) y el expansor de claves mediante módulos pseudoaleatorios y sistemas caóticos discretos y continuos implementados en hardware. La arquitectura propuesta permite evaluar el impacto de la cuantización, la precisión finita y la degradación digital sobre generadores caóticos integrados en sistemas criptográficos, sin necesidad de resintetizar el núcleo AES. Asimismo, la plataforma facilita el análisis comparativo entre distintas configuraciones dinámicas manteniendo una misma arquitectura hardware base. Los resultados muestran que las configuraciones dinámicas preservan métricas globales de cifrado cercanas a las del AES convencional. Asimismo, la evaluación aislada de las S-box permite distinguir sus propiedades intrínsecas de las métricas inducidas por la difusión global de AES. De este modo, la plataforma constituye un entorno de validación y experimentación para arquitecturas criptográficas reconfigurables en FPGA.**

***Palabras Clave*—AES, FPGA, S-box, sistemas caóticos, PRNG.**

## I. Agradecimientos

Este trabajo fue financiado por el CONICET (PIP 2025–2027, Subsidio N.º 11220250100447CO) y por el Fondo para Reparación y Actualización de Equipamiento Científico 2025 de la Fundación Williams.

ISBN 978-631-90145-4-9 - Congreso Argentino de Sistemas Embebidos CASE 2026 - 6 y 7 de agosto

# Estimación de Parámetros de Ruido de Middleton Clase A mediante Algoritmos Genéticos en FPGA

E. Rodriguez[†‡], L. A. Rabioglio[†‡], M. Antonelli[†‡] L. De Micco[†‡§]
R. Lopresti[†‡] y C. Cebedio[†‡]
*† ICYTE*, *‡ FI–UNMdP*, *§ CONICET*
{edrodriguez,mmedina,lucas.rabioglio,leoarn,celestecebedio,ldemicco}@fi.mdp.edu.ar

***Resumen*—Este trabajo presenta el diseño e implementación en FPGA de un estimador de parámetros para ruido impulsivo Middleton Clase A basado en Algoritmos Genéticos (GA), orientado a sistemas embebidos y aplicaciones en tiempo real. La estimación se formula mediante la minimización de la diferencia entre la Función de Distribución Acumulada Excedida empírica y la teórica.**

**Se evalúan distintas variantes de GA en punto fijo con operadores optimizados para hardware basados en manipulación de bits. Los resultados muestran estimaciones precisas y robustas, comparables a implementaciones en punto flotante, pero con menor complejidad y mayor adaptabilidad a FPGA. Finalmente, se presenta la arquitectura modular del sistema y sus principales bloques funcionales.**

***Palabras Clave*—ruido, Middleton, clase A, algoritmos genéticos, FPGA, sistemas embebidos.**

Categoría

Artículo

Implementación de SE

Área Temática

# Embedded Open-Architecture Device for Solar Tracking Applications: Dual-Axis Control and Environmental Characterization

1st Ángel A. Torres-Cervantes
*Instituto Politécnico Nacional*
*UPIITA*
Mexico City, Mexico
atorresc1502@alumno.ipn.mx

2nd Diego A. Flores-Hernández
*Instituto Politécnico Nacional*
*UPIITA*
Mexico City, Mexico
dfloreshe@ipn.mx

3rd Octavio Gutiérrez-Frías
*Instituto Politécnico Nacional*
*UPIITA*
Mexico City, Mexico
ogutierrezf@ipn.mx

***Abstract*—In the field of sustainable energy generation, photovoltaic technology is the most widely used, and its performance increases when it includes a solar tracking system. These systems require complex algorithms for trajectory control and parameter measurement to optimize performance. Therefore, this paper presents the development of an integrated embedded system with an open architecture for dual-axis solar tracking applications. The system design uses a mechatronic approach employing the VDI-2206 methodology and systems engineering principles as a framework. For tracking validation, a simulation-based assessment of PD, PID, and GPI controllers is presented through a multibody dynamic model. The PD controller achieved the lowest tracking error with a mixed RMSE of $0.1484^{\circ}$, while the GPI strategy yielded the most favorable energy profile with an estimated consumption of $2.43$ Wh. In addition, to validate the acquisition and processing capabilities of the embedded device, an experimental test under varying atmospheric conditions was carried out. The results demonstrate that the proposed platform is a scalable and reconfigurable solution for solar tracking and photovoltaic characterization, offering key advantages, including autonomous operation, low power consumption, real-time response, and direct integration with field hardware, consolidating them as an efficient alternative for performance optimization in solar energy installations.**

***Index Terms*—Embedded Device, Environmental Monitoring, Open-Architecture, Solar Energy, Solar Tracking.**

## Acknowledgment

This research was partially supported by the Instituto Politécnico Nacional – Secretaría de Investigación y Posgrado (IND-2026-0059, IND-2026-0230 y 2025-4346), and by the Sistema Nacional de Investigadoras e Investigadores – SECIHTI.

## References

[1] O. Enow, A. Ofoedu, E. Gbabo, and P. Chima, "The role of embedded systems in industrial automation: A review of global developments," *International Journal of Multidisciplinary Research and Growth Evaluation*, vol. 4, pp. 836–843, 01 2023.

[2] Y. Abou Jieb and E. Hossain, *Photovoltaic Systems: Fundamentals and Applications*, 1st ed. Switzerland: Springer Cham, 2022, published online: 07 December 2021.

[3] A. J. Khalifa and S. Al-Mutawalli, "Effect of two-axis sun tracking on the performance of compound parabolic concentrators," *Energy Conversion and Management*, vol. 39, pp. 1073–1079, 07 1998.

[4] M. Castañeda, E. Rosales-Asensio, V. González-Martínez, and A. Colmenar-Santos, "Design and implementation of a high-precision dual-axis solar tracking system based on an open-source architecture," *Energies*, vol. 18, no. 10, 2025.

[5] K. Kumba, P. Upender, P. Buduma, M. Sarkar, S. P. Simon, and V. Gundu, "Solar tracking systems: Advancements, challenges, and future directions: A review," *Energy Reports*, vol. 12, pp. 3566–3583, 2024.

[6] J. Gausemeier and S. Moehringer, "Vdi 2206- a new guideline for the design of mechatronic systems," *IFAC Proceedings Volumes*, vol. 35, pp. 785–790, 12 2002.

[7] R. Hamelin, D. Walden, and M. Krueger, "Incose systems engineering handbook v3.2, improving the process for se practioners," vol. 20, 07 2010.

[8] P. Cooper, "The absorption of radiation in solar stills," *Solar Energy*, vol. 12, no. 3, pp. 333–346, 1969.

[9] M. Blanco-Muriel, D. C. Alarcón-Padilla, T. López-Moratalla, and M. Lara-Coira, "Computing the solar vector," *Solar Energy*, vol. 70, no. 5, pp. 431–441, 2001.

[10] D. A. Flores-Hernández, S. I. Palomino-Resendiz, A. Luviano-Juárez, N. Lozada-Castillo, and O. Gutiérrez-Frías, "A heuristic approach for tracking error and energy consumption minimization in solar tracking systems," *IEEE Access*, vol. 7, pp. 52 755–52 768, 2019.

[11] R. F. Fuentes-Morales, A. Diaz-Ponce, M. I. Peña-Cruz, P. M. Rodrigo, L. M. Valentín-Coronado, F. Martell-Chavez, and C. A. Pineda-Arellano, "Control algorithms applied to active solar tracking systems: A review," *Solar Energy*, vol. 212, pp. 203–219, 2020.

[12] H. Sira-Ramírez and R. Silva-Ortigoza, *Control Design Techniques in Power Electronics Devices*. Berlin, Germany: Springer Science & Business Media, 2006.

[13] S. Palomino-Resendiz, D. Flores-Hernández, L. Cantera-Cantera, N. Lozada-Castillo, and A. Luviano-Juárez, "Design and implementation of model-based predictive control for two-axis solar tracker," *Solar Energy*, vol. 265, p. 112080, 2023.

# Strivex: Sensado Biomecánico en Tiempo Real mediante VBT con Encoder Lineal y ESP32-S3

Alejo Montilla,[*], Jeremias Gaia,[†], Eugenio Orosco,[*] Gabriel Cañadas[*]
[*] Departamento de Electrónica y Automática, Facultad de Ingeniería, Universidad Nacional de San Juan, Argentina Email: jgaia@inaut.unsj.edu.ar

***Resumen*—La cuantificación de la carga interna y externa en el entrenamiento de fuerza es fundamental para evitar el sobreentrenamiento y maximizar las adaptaciones neuromusculares. Este trabajo presenta el diseño e implementación de un encoder lineal deportivo de bajo costo basado en el microcontrolador ESP32-S3 y un encoder magnético AS5600. El sistema permite la medición en tiempo real de variables cinemáticas y dinámicas, integrando una interfaz táctil de alta resolución y conectividad BLE para la gestión de datos en dispositivos móviles. Los resultados demuestran que la precisión del muestreo y el procesamiento concurrente permiten una evaluación fidedigna del perfil carga-velocidad.**

***Palabras Clave*—Sistemas Embebidos, Tecnolog´ıa vestible, Deportes, VBT, Evaluacio´n de desempeño.**

## REFERENCIAS

[1] B. J. Schoenfeld, "Repetitions and muscle hypertrophy," *Strength & Conditioning Journal*, vol. 22, no. 6, p. 67, 2000.
[2] L. Sánchez-Medina and J. J. Gonza´lez-Badillo, "Velocity loss as an indicator of neuromuscular fatigue during resistance training," *Medicine & Science in Sports & Exercise*, vol. 43, no. 9, pp. 1725– 1734, 2011.
[3] S. Jaric, "Force-velocity relationship of muscles performing multi- joint maximum performance tasks," *International Journal of Sports Medicine*, vol. 36, no. 9, pp. 699–704, 2015.
[4] J. J. Gonzalez-Badillo and L. Sánchez-Medina, "Movement velocity as a measure of loading intensity in resistance training," *International Journal of Sports Medicine*, vol. 31, no. 5, pp. 347–352, 2010.
[5] F. Pareja-Blanco, D. Rodr´ıguez-Rosell, L. Sa´nchez-Medina, J. Sanchis-Moysi, C. Dorado, R. Mora-Custodio, J. M. Yánez Garc´ıa, D. Morales-Alamo, I. Pérez-Suárez, J. A. L. Calbet, and J. J. Gonza´lez-Badillo, "Effects of velocity loss during resistance training on athletic performance, strength gains and muscle adaptations," *Scandinavian Journal of Medicine & Science in Sports*, vol. 27, no. 7, pp. 724–735, 2017.
[6] M. Izquierdo, K. Hakkinen, J. Ibanez, M. Garrues, A. Anton, A. Zuniga, J. Larrion, and E. Gorostiaga, "Effects of strength training on muscle power and serum hormones in middle-aged and older men," *Journal of applied physiology*, vol. 90, no. 4, pp. 1497– 1507, 2001.
[7] A. Pérez-Castilla, A. Piepoli, G. Delgado-García, G. Garrido-Blanca, and A. Garc´ıa-Ramos, "Reliability and concurrent validity of seven commercially available devices for the assessment of movement velocity at different intensities during the bench press," *Journal of Strength and Conditioning Research*, vol. 33, no. 5, pp. 1258–1265, 2019.
[8] J. Weakley, B. Mann, H. Banyard, S. McLaren, T. Scott, and A. Garcia-Ramos, "Velocity-based training: From theory to application," *Strength & Conditioning Journal*, vol. 43, no. 2, pp. 31–49, 2021.
[9] Waveshare Electronics, "ESP32-S3-Touch-LCD-1.47 – User Manual and Wiki," https://www.waveshare.com/wiki/ESP32-S3-Touch-LCD-1.47, 2024, accedido: 2026-06-25.
[10] ams-OSRAM AG, "AS5600 12-bit Programmable Contactless Potentiometer Datasheet," Document v1-09, 2018, available: https://ams-osram.com/products/sensor-solutions/position-sensors/ ams-as5600-position-sensor.
[11] V.-A. Dobra, I.-M. Dobra, and S. Folea, "Real-time paddle stroke classification and wireless monitoring in open water using wearable inertial nodes," *Sensors*, vol. 25, no. 17, p. 5307, 2025.
[12] J. F. Navarro-Iribarne, D. Moreno-Salinas, and J. Sánchez-Moreno, "Low-cost portable system for measurement and representation of 3D kinematic parameters in sport monitoring: Discus throwing as a case study," *Sensors*, vol. 22, no. 23, p. 9408, 2022.
[13] A. Martínez-Cava, A. Hernández-Belmonte, J. Courel-Ibáñez, E. Conesa-Ros, R. Morán-Navarro, and J. G. Pallare´s, "Validity and concordance of a linear position transducer (vitruve) for measuring movement velocity during resistance training," *Sensors*, vol. 24, no. 19, p. 6444, 2024.

Categoría

# Artículo

Inteligencia Artificial en SE

Área Temática

# MoodLink: An Embedded Wearable for On-Device Speech Emotion Recognition

Rodrigo Vázquez and Federico G. Zacchigna
Facultad de Ingeniería, Universidad de Buenos Aires (FIUBA)
Buenos Aires, Argentina
rvazquez@fi.uba.ar, federico.zacchigna@gmail.com

***Abstract*—This paper presents MoodLink, a wearable embedded device that performs on-device Speech Emotion Recognition (SER) on an ESP32-S3 microcontroller. The device captures 4-second audio samples through an I2S Micro-Electro-Mechanical System (MEMS) microphone, extracts Mel-Frequency Cepstral Coefficients, and classifies the speaker's emotional state into seven categories using a quantized Convolutional Neural Network running on-device via TensorFlow Lite Micro. The detected emotion is communicated through a circular display and an RGB LED, providing non-verbal emotional communication support for individuals with autism spectrum disorder or attention deficit hyperactivity disorder. Deploying a convolutional SER model on a resource-constrained microcontroller presented substantial memory and latency challenges. Inference time was reduced from approximately six minutes to under six seconds through architecture simplification and INT8 quantization. A critical finding was the spectral domain shift between professional studio datasets and the MEMS microphone used on-device. This was addressed by re-recording the entire training dataset, playing back audio through a standalone speaker and capturing it with the device's own MEMS microphone, forcing the model to learn microphone-invariant emotional features. The resulting model achieves 54.6 % validation accuracy, with an inference time of approximately 4.2 seconds and a memory footprint under 2 MB of PSRAM.**

## REFERENCES

[1] S. Baron-Cohen, S. Wheelwright, J. Hill, Y. Raste, and I. Plumb, "The 'Reading the Mind in the Eyes' Test revised version: A study with normal adults, and adults with Asperger syndrome or high-functioning autism," *Journal of Child Psychology and Psychiatry*, vol. 42, no. 2, pp. 241–251, 2001.

[2] M. B. Akçay and K. Oğuz, "Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers," *Speech Communication*, vol. 116, pp. 56–76, 2020.

[3] B. W. Schuller, "Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends," *Communications of the ACM*, vol. 61, no. 5, pp. 90–99, 2018.

[4] S. R. Livingstone and F. A. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS)," *PLOS ONE*, vol. 13, no. 5, 2018.

[5] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, "CREMA-D: Crowd-sourced emotional multimodal actors dataset," *IEEE Trans. Affective Computing*, vol. 5, no. 4, pp. 377–390, 2014.

[6] NeuroByte Consulting, "Speech Emotion Recognition in TensorFlow Using CNNs," GitHub repository, 2023. https://github.com/NeuroByte-Consulting/Speech-Emotion-Recognition-in-Tensorflow-Using-CNNs

[7] Espressif Systems, "ESP32-S3 Series Datasheet," 2024. https://www.espressif.com/documentation/esp32-s3_datasheet_en.pdf

[8] TensorFlow, "TensorFlow Lite Micro," 2024. [Online]. Available: https://www.tensorflow.org/lite/microcontrollers

Categoría

# Artículo

Internet de las Cosas

Área Temática

ISBN 978-631-90145-4-9 - Congreso Argentino de Sistemas Embebidos CASE 2026 - 6 y 7 de agosto

# Nodos IIoT reconfigurables para la gestión segura de pellets de girasol en la industria aceitera

Elias S. Fliger
Instituto Argentino de Radioastronomía
Berazategui, Bs. As., Argentina
esfliger@iar.unlp.edu.ar

Ana Evelina Y. Tarcetti
Instituto Argentino de Radioastronomía
Berazategui, Bs. As., Argentina
evelina@iar.unlp.edu.ar

Matias L. Contreras
Instituto Argentino de Radioastronomía
Berazategui, Bs. As., Argentina
mcontreras@iar.unlp.edu.ar

Eliseo D. Díaz
Instituto Argentino de Radioastronomía
Berazategui, Bs. As., Argentina
ediaz@iar.unlp.edu.ar

Rubén D. Morán Fabra
Instituto Argentino de Radioastronomía
Berazategui, Bs. As., Argentina
ruben@iar.unlp.edu.ar

Martin Salibe
Instituto Argentino de Radioastronomía
Berazategui, Bs. As., Argentina
msalibe@iar.unlp.edu.ar

***Resumen*—Este trabajo presenta el diseño e implementación de nodos IIoT (*Industrial Internet of Things*) modulares y reconfigurables aplicados al monitoreo del almacenamiento de pellets de girasol en una planta elaboradora de aceite. El pellet de girasol es susceptible de autoignición espontánea por oxidación, lo que representa un riesgo crítico de seguridad para instalaciones y operarios. El sistema se basa en una placa controladora con SoC ESP32-S3 y un conjunto de placas de expansión intercambiables para interfaces industriales: termocuplas (ANITC), lazos de 4-20 mA (ANI420) y señales discretas (DIOQT). El firmware, desarrollado sobre ESP-IDF con FreeRTOS, implementa redes WiFi mesh auto-organizadas, una máquina de estados finita (FSM) y un esquema de mensajería adaptado del estándar Sparkplug B sobre MQTT con serialización JSON. La configuración dinámica de cada nodo elimina la necesidad de recompilar el firmware ante cambios operativos. El pipeline de datos se conforma con herramientas de código abierto, ofreciendo observabilidad en tiempo real del proceso y del estado interno de los nodos.**

***Palabras clave—IIoT, WiFi Mesh, redes de sensores, Industria 4.0.***

## I. INTRODUCCIÓN

La digitalización de los procesos productivos es un factor central en la transición hacia la Industria 4.0. Los establecimientos agroindustriales, como las plantas elaboradoras de aceite, poseen procesos que involucran múltiples etapas con variables críticas que requieren monitoreo continuo. Sin embargo, gran parte de las cooperativas y PyMEs industriales enfrentan restricciones económicas y técnicas que dificultan la adopción de productos comerciales IIoT o SCADA cerrados: altos costos de licenciamiento, poca adaptabilidad a la infraestructura preexistente y dependencia de integradores externos. En nuestra región, las soluciones IIoT de bajo costo y código abierto aparecen como una vía accesible para la digitalización y adaptación de nuevas tecnologías [1, 2, 3].

Sin embargo, el uso de microcontroladores de bajo costo con conectividad inalámbrica (WiFi, Bluetooth, LoRa, etc.) integrada, ha experimentado un crecimiento sostenido en aplicaciones IIoT, particularmente en contextos industriales con restricciones de inversión [6]. En este escenario, la familia de dispositivos ESP32 se ha consolidado como una solución de referencia, debido a su integración de capacidades de procesamiento, conectividad y soporte de ecosistemas de desarrollo abiertos, habilitando implementaciones de monitoreo industrial, redes de sensores y sistemas SCADA de bajo costo [7, 8].

En este caso, la Cooperativa Aceitera La Matanza (CALM) cuenta en su planta con celdas de almacenamiento de pellets de girasol. Este subproducto del proceso de extracción de aceite es altamente susceptible a la oxidación exotérmica y puede alcanzar la autoignición espontánea si la temperatura interna de las pilas no se mantiene bajo control, representando un riesgo para las instalaciones y operarios [10]. Por ende, el monitoreo manual resulta insuficiente en estos volúmenes de biomasa, donde la temperatura interior puede elevarse sin manifestaciones exteriores visibles. La instrumentación remota y continua de la temperatura y otros parámetros como el CO2 es, por tanto, una necesidad de seguridad prioritaria además de una mejora en la eficiencia general de la planta.

Anteriormente, en el marco de los Proyectos Federales de Innovación (PFI2021) [4], se desarrolló una primera versión de un sistema IIoT para la CALM con nodos de arquitectura rígida. Si bien la prueba piloto demostró su viabilidad, presentó limitaciones de escala y mantenibilidad. Para superar estos desafíos, se presentó una mejora cualitativa desarrollada con el apoyo del Fondo de Innovación Tecnológica de Buenos Aires (FITBA), instrumento de política pública orientado al fortalecimiento productivo y tecnológico de PyMEs y cooperativas industriales bonaerenses [5].

El presente artículo describe CALM2 IIoT, una versión mejorada del ecosistema, en línea con esquemas ampliamente utilizados en entornos agrícolas e industriales semejantes [9].

Las principales contribuciones son: (1) arquitectura de *hardware* modular desacoplada, con placas de expansión intercambiables, extensible a otros procesos de la cooperativa, ver Fig. 2; (2) firmware estructurado en capas sobre FreeRTOS con FSM y distribución de tareas entre los núcleos del microcontrolador; (3) esquema de mensajería MQTT adaptado de Sparkplug B con formato JSON; (4) mecanismo de reconfiguración dinámica sin necesidad de recompilación, para una rápida instalación de los dispositivos en campo; y (5) un pipeline de datos de código abierto, desplegable en *hardware* de bajo costo con base de datos InfluxDB y visualización en Grafana.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

## II. Arquitectura de Hardware

### A. *Placa controladora (UCTRL)*

El núcleo del nodo es la placa UCTRL, definido por un módulo ESP32-S3 dual-core Xtensa LX7, WiFi/Bluetooth, antena externa, 16 MB de Flash y 8 MB de PSRAM [11]. La UCTRL opera como un concentrador con tres buses de expansión: I2C, SPI y RS485 (Modbus RTU, pendiente de implementación). La interfaz UART disponible, permite la reconfiguración del nodo sin herramientas de software adicionales.

### B. *Módulos periféricos*

Los módulos periféricos, destacados en la Fig. 1, consisten en placas apilables, permitiendo configurar cada nodo según los requisitos y demandas del proceso. Las placas actualmente desarrolladas son:

**ANITC**: Adquisición de temperatura para termocuplas 'T', hasta 8 canales por placa (escalable a 32 canales por nodo). Es la placa central para el monitoreo térmico de los pellets.

**ANI420**: Adquisición analógica para lazos de corriente de 4–20 mA de uso industrial. Permite integrar variables de presión, nivel o caudal presentes en otras etapas del proceso productivo.

**DIOQT**: Entradas/Salidas discretas hasta 16 canales, optoaisladas y accionamiento mediante transistores y/o relés, proyectado para el control de actuadores en fases posteriores del proyecto.

## III. Diseño e Implementación del Firmware

El firmware CALM2 fue desarrollado sobre ESP-IDF v5.5 con FreeRTOS. La arquitectura se organiza en cuatro capas: aplicación (FSM + gestor de tareas), componentes (lógica de negocio: configuración, adquisición, red), drivers (abstracción de periféricos: I2C, SPI, GPIO, RS485) y HAL/ESP-IDF.

### A. *Distribución de tareas*

La asignación explícita de tareas a cada núcleo del ESP32-S3 minimiza la interferencia entre el *stack* de red y la lógica de aplicación:

*Core* 0: tarea WiFi/Mesh. Aísla el *stack* de red para garantizar predictibilidad en las tareas de tiempo real.

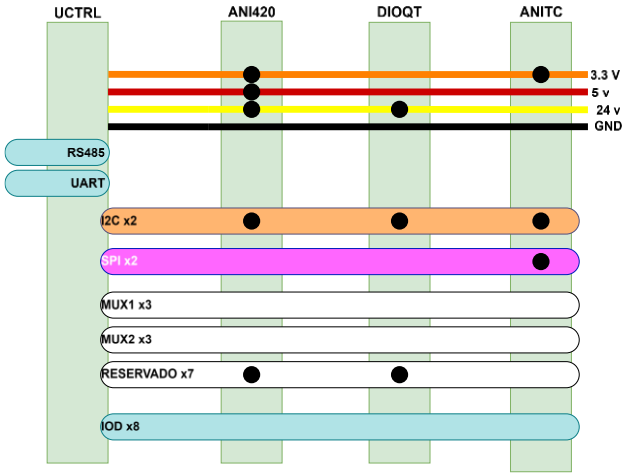



Fig. 2: Arquitectura de hardware del nodo CALM2.

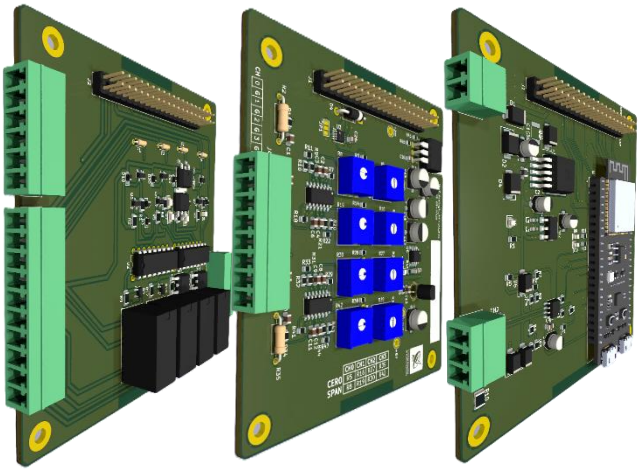


Fig. 1: Renderizado de las placas desarrolladas: DIOQT, ANI420 y UCTRL.

*Core* 1: Tareas FSM, MQTT, adquisición de datos, recepción de mensajes de la red mesh y control de LEDs indicadores del rol del dispositivo.

### B. *Máquina de Estados Finita (FSM)*

El ciclo de vida del nodo lo gestiona una FSM de cinco estados mediante grupos de eventos de FreeRTOS (`EventGroupHandle_t`), ver Fig. 3.

Los estados definidos son:

**INIT**: Inicializa GPIO, UART, drivers de placas y *watchdog* de 10 s. Lanza un temporizador de 15 s para la ventana de configuración. Transición a CONFIG si llega pedido por UART o existe configuración válida en NVS.

**CONFIG**: Decodifica el archivo JSON de configuración (desde UART o NVS) y lo aplica al sistema. Transición a CONEXION al finalizar.

**CONEXION**: Ejecuta el stack WiFi Mesh y aguarda la asignación de rol (Root/Node/Leaf). Regresa a este estado ante pérdida de enlace desde OPERACION.

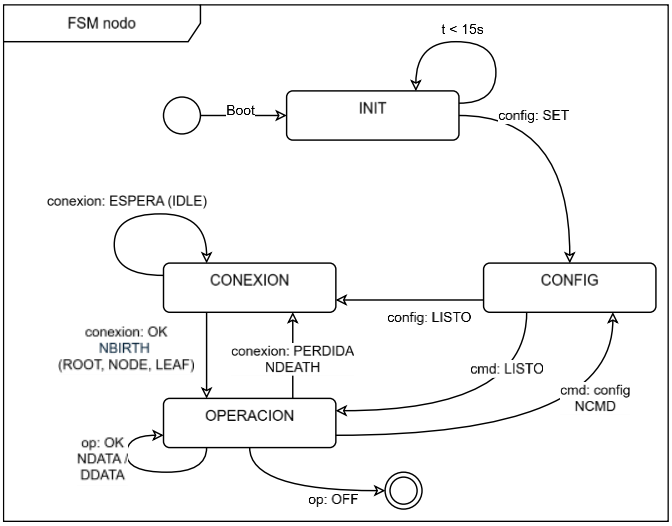



Fig. 3: Diagrama de la FSM del firmware.

**OPERACION**: Estado nominal. Activa las tareas de adquisición, MQTT y recepción Mesh. Un LED indicador muestra el rol del nodo en la red para la supervisión local.

**FIN**: Estado terminal de recuperación. Aguarda 1 s para el vaciado de logs y ejecuta `esp_restart()`, reiniciando el ciclo. Se prevé implementar registro *post mortem* en NVS para diagnóstico en campo.

### C. *Mensajería*

El sistema adopta la jerarquía semántica del estándar Sparkplug B [12] sobre MQTT [13] con serialización JSON en lugar de *Protocol Buffers* binarios, distinguiendo los mensajes de sesión (BIRTH/DEATH) de los mensajes de datos de dispositivos y nodos (DDATA/NDATA). La migración a la codificación binaria está prevista en versiones futuras.

La estructura de los diferentes tópicos se desglosa en la Tabla I.

TABLA I. TÓPICOS PUBLICADOS

| Tipo | Tópico | Frecuencia |
|---|---|---|
| NBIRTH | `calm/{proceso}/NBIRTH/{tipo}-{id}` | Al inicio |
| DBIRTH | `calm/{proceso}/DBIRTH/{tipo}-{id}/ANITC-{n}` | Al inicio |
| NDATA | `calm/{proceso}/NDATA/{tipo}-{id}` | 5s |
| DDATA | `calm/{proceso}/DDATA/{tipo}-{id}/ANITC-{n}` | 5s |

Al conectar, el Root publica los certificados NBIRTH (metadatos: MAC, firmware, sitio, placas activas) y DBIRTH por cada placa (etiquetas de canales, unidades, rangos). Periódicamente emite los tópicos NDATA (estado de red: rol, RSSI, uptime, heap libre) y DDATA (datos de proceso según la configuración del nodo).

NBIRTH (ejemplo):

```
{
  "mac": "AA:AA:AA:AA:AA:AA",
  "fw": "2.0",
  "sitio": "Celda-01",
  "node": "SNIoT-01",
  "boards": {"ANITC":[1], "DIOQT":[], "AIN420":[]}}
```

NDATA (ejemplo):

```
{
  "seq": 42,
  "mac": "AA:AA:AA:AA:AA:AA",
  "sitio": "Celda-01",
  "ip": "XX.XX.XX.XX",
  "role": "ROOT",
  "rssi": -45,
  "uptime": 3600,
  "heap": 280000}
```

DDATA/ANITC (ejemplo):

```
{
  "ch1": 64.82, "ch2": 64.74, "ch3": 65.10,
  "ch4": 64.50, "ch5": 64.92, "ch6": 65.01,
  "ch7": 64.88, "ch8": 34.53}
```

### D. *Configuración dinámica*

Cada nodo se parametriza mediante una página web de la Fig. 4, que genera el archivo JSON que especifica el proceso, sitio, las placas activas, canales (etiquetas, rangos, unidades) y red WiFi/MQTT. El archivo se transfiere por UART con un protocolo de *handshake* y se almacena en la partición NVS del ESP32-S3. Para ello, se emplean dos herramientas de asistencia al usuario: un *script* PowerShell para la transferencia UART, y una interfaz web que permite la configuración gráfica desde el navegador sin instalar herramientas de desarrollo adicionales.

## IV. PIPELINE DE DATOS

Los *payloads* MQTT, representativos de la instalación en la CALM Fig. 5, son procesados por un flujo Node-RED separando la jerarquía de tópicos y distribuyendo los datos para tres buckets de InfluxDB: `niot_proceso` (lecturas de sensores), `niot_mesh` (estado de la red) y `niot_status` (inventario y metadatos de canales). El etiquetado dinámico de canales en Grafana cruza los datos de medición con los metadatos del mensaje DBIRTH almacenados en `niot_status`, evitando nombres genéricos en los paneles operativos, Fig. 6. El *stack* completo se despliega con contenedores Docker gestionados por Portainer, tanto en equipos de desarrollo como en el *hardware* de producción en campo.

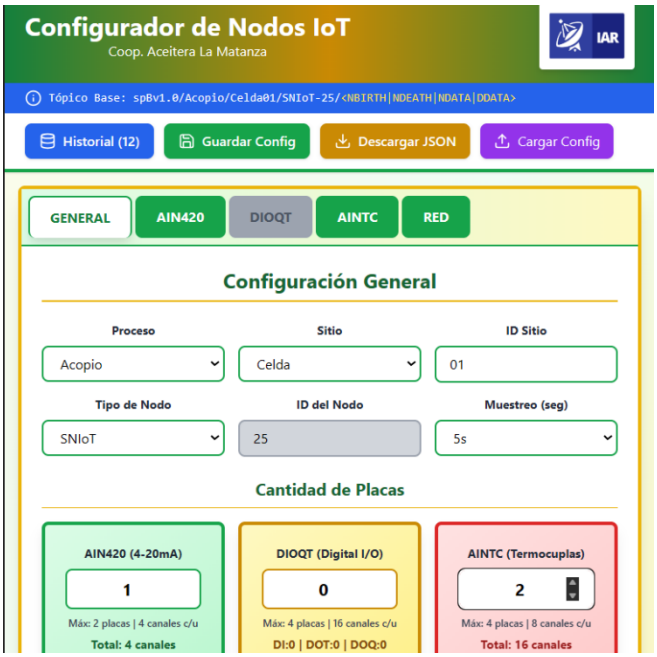



Fig. 4: Configurador web de los nodos IoT.

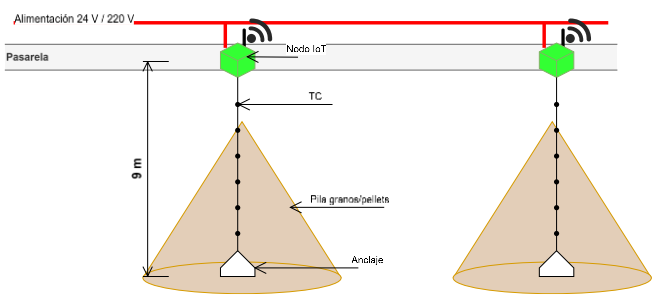



Fig. 5: Instalación de los nodos en la planta para la medición de pellets.

## V. Resultados de integración y ensayo

Se implementó un entorno de ensayo compuesto por tres nodos y un emulador de 5 nodos adicionales. En todos los casos se configuró una placa ANITC de 8 canales cada una. Este enfoque permitió evaluar la escalabilidad del sistema sin necesidad de replicar *hardware*.

Los ensayos consideraron:

**Adquisición**: verificación de lectura de los 8 canales en cada nodo, sin pérdida de datos ni conflictos en los buses SPI e I2C.

**Carga de red**: incremento progresivo del número de nodos emulados, evaluando latencia y estabilidad del *broker*, con una cadencia de mensajes de 5 s para cada nodo.

**Tolerancia a fallos**: desconexión controlada de nodos y reinicio de dispositivos para validar la recuperación automática y la persistencia de configuración.

Se logró constatar sobre un total de 26.280 s y 42.041 mensajes entre los 8 nodos (físicos y emulados) una latencia media de 15 ms con $\sigma$ = 233 ms, sin pérdida de paquetes.

El tiempo de recuperación de la red *mesh* se mantuvo por debajo de los 50 s.

## VI. Conclusiones y trabajos futuros

Se presentó CALM2 IIoT, un sistema modular reconfigurable, orientado al monitoreo distribuido de la temperatura en el almacenamiento de pellets de girasol, además de diferentes variables de otros procesos de la planta aceitera.

Los resultados de laboratorio demuestran que la arquitectura propuesta:

- soporta la adquisición de volúmenes significativos de datos en configuraciones reales de operación,
- mantiene buena estabilidad de comunicación bajo carga variable, incluso con nodos emulados,
- permite el despliegue y configuración en campo de los nodos sin dependencia de herramientas cerradas.

Posteriormente, se deberá validar la implementación en planta, para asegurar a la solución propuesta como una herramienta efectiva para la detección temprana de anomalías térmicas, contribuyendo directamente a la seguridad del personal y de los bienes de la cooperativa.

Como líneas de trabajo futuro se identifican:

- implementación de Modbus RTU sobre RS485 para integración con equipamiento industrial existente;
- adopción de *encoding* binario compatible con Sparkplug B para optimización de comunicaciones;
- incorporación de rutinas de calibración en campo para sensores analógicos;
- implementación de registro *post mortem* en memoria no volátil para diagnóstico remoto;
- programación *over-the-air* (OTA) para la actualización de los dispositivos en campo.




Fig. 6. Pantalla de Grafana del pipeline de datos durante los ensayos de integración.

## Referencias


[1] J. Motta, H. Moreno y R. Ascúa, Industria 4.0 en mipymes manufactureras de la Argentina, Documentos de Proyectos (LC/TS.2019/93), Santiago, Comisión Económica para América Latina y el Caribe (CEPAL), 2019.

[2] De Lucas Ancillo A, Gavrila Gavrila S, Fernández del Castillo Díez JR, Corro Beseler J (2022), LATAM and Spanish SME barriers to Industry 4.0. Academia Revista Latinoamericana de Administración, Vol. 35 No. 2 pp. 204–222, doi: https://doi.org/10.1108/ARLA-07-2021-0137.

[3] Ascúa, R. A. (2021). Industry 4.0 in manufacturing SMEs of Argentina and Brazil. Journal of the International Council for Small Business, 2(3), 203–222. https://doi.org/10.1080/26437015.2021.1899773.

[4] Valdéz, G., Donantueno, A., Contreras, M., Salibe, M., Fliger, E. (2024). Diseño e implementación de nodos IoT de bajo costo para monitoreo de tanques de aceite en una Cooperativa Aceitera. Congreso Argentino de Sistema Embebidos, S. M. Tucumán, Argentina.

[5] Agüero, F., Mancini, M., Ares Rossi, I. N., Bustos, F. D., & Kodric, A. I. (2025). Políticas científico-tecnológicas para el desarrollo en la provincia de Buenos Aires. Ciencia, Tecnología Y Política, 8(14), 137. https://doi.org/10.24215/26183188e137.

[6] Martikkala, Antti & David, Joe & Lobov, Andrei & Lanz, Minna & Flores Ituarte, Iñigo. (2021). Trends for Low-Cost and Open-Source IoT Solutions Development for Industry 4.0. Procedia Manufacturing. 55. 298-305. 10.1016/j.promfg.2021.10.042.

[7] Lawrence O. Aghenta, M. Tariq Iqbal. Design and implementation of a low-cost, open source IoT-based SCADA system using ESP32 with OLED, ThingsBoard and MQTT protocol[J]. AIMS Electronics and Electrical Engineering, 2020, 4(1): 57-86. doi: 10.3934/ElectrEng.2020.1.57.

[8] Maier, Alexander & Sharp, Andrew & Vagapov, Yuriy. (2017). Comparative Analysis and Practical Implementation of the ESP32 Microcontroller Module for the Internet of Things. 10.1109/ITECHA.2017.8101926.

[9] Jawad, H. M., Nordin, R., Gharghan, S. K., Jawad, A. M., & Ismail, M. (2017). Energy-Efficient Wireless Sensor Networks for Precision Agriculture: A Review. Sensors, 17(8), 1781. https://doi.org/10.3390/s17081781.

[10] Gilvari H, Cutz L, Tiringer U, Mol A, de Jong W, Schott DL. The Effect of Environmental Conditions on the Degradation Behavior of Biomass Pellets. Polymers (Basel). 2020 Apr 21;12(4):970. doi: 10.3390/polym12040970. PMID: 32326362; PMCID: PMC7240501.

[11] Espressif Systems, ESP32-S3 Series Datasheet v2.2, [Online], Disponible: https://documentation.espressif.com/esp32-s3_datasheet_en.pdf.

[12] Sparkplug Specification, Version 3.0.0, Eclipse Sparkplug Contributors, Nov. 16, 2022. [Online]. Disponible: https://sparkplug.eclipse.org/specification/version/3.0/documents/sparkplug-specification-3.0.0.pdf.

[13] MQTT Version 3.1.1, OASIS Standard, Oct. 29, 2014. [Online]. Disponible: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf.

# State-Based Energy Estimation for Zephyr RTOS in NB-IoT

Rocío Cabral, Leonardo Steinfeld, and Julián Oreggioni,
Universidad de la República, Montevideo, Uruguay
Email: {rcabral, leo, juliano}@fing.edu.uy

***Abstract*—Battery lifetime is a critical design constraint in IoT embedded systems, particularly in applications using cellular communication technologies such as NB-IoT. While external power profiling tools provide accurate measurements, they are often intrusive and difficult to use during field deployments. This work presents a lightweight state-based energy estimation framework for Zephyr-based embedded systems. The proposed approach models the system as a set of independent subsystems with associated operating states and runtime activity tracking. A configurable implementation integrated with Zephyr RTOS and nRF Connect SDK is presented without requiring modifications to existing application components. The framework was experimentally validated using an NB-IoT livestock monitoring platform based on the nRF9160 SoC. Experimental results obtained using Nordic Power Profiler Kit II measurements show low estimation error across different operating conditions and hardware instances, demonstrating the practicality and robustness of the proposed approach.**

***Index Terms*—Energy consumption, NB-IoT, Zephyr, Internet of Things.**

## ACKNOWLEDGMENT

This work was partially funded by CSIC (Comisión Sectorial de Investigación Científica, Udelar), ANII (Agencia Nacional de Investigación e Innovación, Uruguay) and IEEE Circuits and Systems Society (Uruguayan Chapter).

## REFERENCES

[1] K. Martinez, J. K. Hart, and R. Ong, "Environmental sensor networks," *Computer*, vol. 37, no. 8, pp. 50–56, 2004.

[2] J. Villaverde, D. Bouvier, C. Fernández, L. Steinfeld, and J. Oreggioni, "Low-power self-energy meter for wireless sensor networks," in *2013 IEEE 9th International Conference on Distributed Computing in Sensor Systems Workshops*. IEEE, 2013, pp. 315–317.

[3] J. Schandy, L. Steinfeld, and F. Silveira, "Average power consumption breakdown of wireless sensor network nodes using ipv6 over llns," in *2015 International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2015, pp. 242–247.

[4] J. P. García-Martín and A. Torralba, "Energy consumption analytical modeling of nb-iot devices for diverse iot applications," *Computer Networks*, vol. 232, p. 109855, 2023.

[5] D. Bortnik, V. Nikic, S. Sobot, D. Vukobratovic, I. Mezei, and M. Lukic, "Evaluation of machine learning algorithms for nb-iot module energy consumption estimation based on radio channel quality," *IEEE Access*, vol. 13, pp. 2389–2408, 2025.

[6] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He, "Software-based on-line energy estimation for sensor nodes," in *Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets)*, 2007, pp. 28–32.

[7] V. Cabrera, A. Delbuggio, H. Cardoso, D. Fraga, A. Gómez, M. Pedemonte, R. Ungerfeld, and J. Oreggioni, "Harnessing technology for livestock research: An online sheep behavior monitoring system," *IEEE Transactions on AgriFood Electronics*, vol. 2, pp. 306–313, 2024.

Categoría

# Artículo

Protocolos y Comunicaciones

Área Temática

ISBN 978-631-90145-4-9 - Congreso Argentino de Sistemas Embebidos CASE 2026 - 6 y 7 de agosto

# Customizing NEORV32 for Matched Filtering in SDR Applications

Daniel A. Rosso, Carlos A. Zerbini, Guillermo G. Riva
Grupo de Investigación y Transferencia en Electronica Avanzada (GInTEA)
Universidad Tecnologica Nacional, Facultad Regional Cordoba, Argentina
Email: {drosso, czerbini, griva}@frc.utn.edu.ar

***Abstract*—Root Raised Cosine (RRC) filtering is a computational bottleneck in embedded 5G NR and LTE baseband receivers: the per-symbol convolution requires $\mathcal{O}(N)$ multiply-accumulate operations per received sample, where $N$ is the filter order. On a bare-metal RISC-V soft-core without hardware multiply-accumulate support, this represents a significant fraction of the available processing budget for IoT-class devices.**

**This paper proposes two custom instructions, `fmac` and `fmac2`, implemented in the Custom Function Unit (CFU) of the open-source NEORV32 RISC-V processor. `fmac` performs a single pipelined Q15 multiply-accumulate in 2 clock cycles with zero register file round-trip overhead. `fmac2` exploits the linear-phase symmetry of RRC filters ($h[k] = h[N-1-k]$) to compute two tap products simultaneously in the same 2-cycle window, halving the instruction count for symmetric filters.**

**Hardware measurements on a Xilinx Artix-7 xc7a35t running at 100 MHz demonstrate speedups of 1.62× (`fmac`) and 1.75× (`fmac2`) over the RV32IMC software baseline for a 33-tap RRC filter, with zero measurement variance across 20 consecutive runs. Area overhead is +100 LUT / +146 FF / +2 DSP48E1 over the baseline. These values correspond to a single filtering channel; a complete in-phase/quadrature receiver applying `fmac2` to both channels requires 45.0 % more cycles than the 60 ksps real-time budget allows, so dual-channel operation needs interleaved scheduling or a reduced symbol rate.**

**The complete CFU RTL (<160 lines of VHDL) and firmware are open-source, available at https://github.com/ginteacomms-utn/2026_case_neorv32_rrc.**

***Index Terms*—RISC-V, NEORV32, Custom Function Unit (CFU), RRC filter, software-defined radio (SDR), FMAC, fixed-point DSP, FPGA, open-source.**

## I. Introduction

Software-Defined Radio (SDR) platforms implement physical-layer waveforms in software, enabling protocol-agnostic reconfigurability without hardware redesign. On FPGA-based SDR receivers (Fig. 1), a common architecture partitions the signal chain between the programmable fabric (RF front-end, ADC interface, digital down-conversion or DDC, and CIC filter) and a soft-core processor that handles the reconfigurable baseband layer (RRC filter, timing sync, carrier frequency offset or CFO correction, channel estimation, and symbol demodulation).

Digital filtering, specifically the Root Raised Cosine (RRC) matched filter, has been identified as dominant computational load in SDR receivers [1], [2]. This matched filter scheme is mandatory for any pulse-shaped waveform (BPSK, QPSK, QAM), regardless of the target standard.

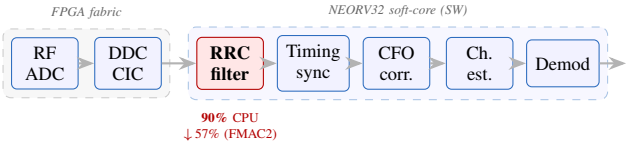



Fig. 1: SDR receive chain.

Table I shows a representative cycle budget for a single-channel QPSK receiver modelled on standard LTE parameters (15 kHz symbol rate, matching the LTE subcarrier spacing per 3GPP TS 36.211 [3], with 4× oversampling, i.e., $f_s = 60$ ksps. Based on our results and previous ones, the RRC filter consumes more than 90% of the cycle budget, leaving marginal headroom for the rest of the baseband stack.

With our optimized block FMAC2, meanwhile, the isolated single-channel filter falls to 56.8 % of the CPU budget (Table I), leaving enough headroom for concurrent execution of timing recovery, frequency correction, and demodulation within the same processor.

The NEORV32 core [4] is a modern, open-source, 32-bit system on chip (SoC) based on RISC-V architecture and optimized for FPGAs, with highly modular and customizable architecture. It exposes a *Custom Function Unit* (CFU) interface that routes any instruction with a `custom-0` or `custom-1` opcode directly to user-defined combinational or pipelined logic, without modifying the processor pipeline or the compiler backend. Instructions are invoked via the standard GCC `.insn` assembler directive, requiring no toolchain patches. We exploit this interface to implement two domain-specific instructions:

1) **`FMAC`**: accumulate one Q15 multiply-accumulate in 2 clock cycles. Eliminates the *load-use stall* (pipeline bubble caused by a `load`→use dependency) and loop overhead of the software tap loop.
2) **`FMAC2`**: accumulate two Q15 products simultaneously, using the packed `rs1` register to hold two samples and the single coefficient applied to both (valid because RRC filters have even symmetry $h[k] = h[N-1-k]$).

## II. Background

### A. Root Raised Cosine (RRC) Filter

The transmitter applies a RRC filter to limit spectral occupancy while preserving the Nyquist criterion; the receiver

TABLE I: Cycle Budget per Block at 60 ksps (1667 cy/sample Budget, NEORV32 @ 100 MHz)

| Block | cy/s | % budget |
|---|---|---|
| **RRC filter** ($N = 33$)† | 1509→946 | 90.5→**56.8** |
| RRC (I+Q, SDR chain)‡ | 2418 | 145.0 |
| CFO correction (NCO)♭ | 116 | 7.0 |
| Timing sync (Gardner TED)♭ | 169 | 10.1 |
| QPSK demodulation‖ | <5 | <1 |
| Channel estimation¶ | TBD | — |
| **Available headroom**§ | −132→431 | −7.9→**25.9** |

†*Simulated.* RTL simulation (Vivado xsim), power-of-2 circular buffer, isolated single-channel filter — distinct from the hardware-measured results of Table III (linear shift-register buffer, $N = 33$: 4 889→2 792 cy, 1.75×). The circular buffer removes shift-register overhead present in that hardware benchmark, giving a smaller 1.6× reduction (1 509→946 cy, −37.3 %) on a leaner baseline. ‡*Simulated.* Complex baseband receiver filtering both in-phase (I) and quadrature (Q) components through the integrated `demo_sdr` firmware; measured directly via RTL simulation (Vivado xsim) of the complete chain, not derived by doubling the single-channel figure above (circular-buffer reset and dual-channel bookkeeping add overhead beyond a simple 2×), exceeding the 60 ksps single-core budget by 45.0 %. ♭*Simulated.* Measured via RTL simulation (Vivado xsim) of the complete integrated `demo_sdr` firmware (this work), decoding the simulated UART output; consistent across all seven swept SNR points. ‖*Not yet measured.* Not instrumented for cycle counting in this firmware (assumed negligible, $< 1\,\%$ of budget); see Future Work. §*Derived.* Headroom computed for the isolated single-channel FMAC2 scenario: $1667 - 946 - 116 - 169 - 5 = 431$ cy (25.9 %); under the baseline RRC filter alone the budget is already exceeded ($1667 - 1509 - 116 - 169 - 5 = -132$ cy). Full I+Q chain (2418 cy) exceeds the 60 ksps budget regardless; real-time I+Q requires interleaved scheduling or a lower symbol rate. ¶*Not implemented.* AWGN channel assumed; listed as future work. Cycle cost depends on pilot density and equaliser order.

applies a second, matched RRC filter to maximise the SNR and to cancel inter-symbol interference (ISI) at the decision point; the cascade of both filters yields a raised-cosine frequency response. The RRC filter impulse response is:

$$h(t) = \frac{\sin(\pi t/T_s(1-\alpha)) + 4\alpha\frac{t}{T_s}\cos(\pi t/T_s(1+\alpha))}{\pi t/T_s\left(1-(4\alpha t/T_s)^2\right)}, \quad (1)$$

where $T_s$ is the symbol period and $\alpha \in [0, 1]$ is the roll-off factor (3GPP specifies $\alpha = 0.22$ for 5G NR and $\alpha = 0.35$ for LTE [3], [5]). The discrete-time implementation with $L$-times oversampling is an $N$-tap FIR filter whose output is:

$$y[n] = \sum_{k=0}^{N-1} h[k]\, x[n-k]. \quad (2)$$

Because (1) is an even function, the sampled coefficients satisfy $h[k] = h[N-1-k]$ (linear phase, Type I FIR), enabling the FMAC2 instruction to pair taps $k$ and $N-1-k$ and compute both products in a single 2-cycle instruction.

### B. NEORV32 Custom Function Unit (CFU)

NEORV32 implements a five-stage in-order pipeline (IF→ID→EX→MEM→WB). The CFU is a co-processor attached to the Execute (EX) stage that receives the following signals:

- `start_i`: single-cycle trigger when a CFU instruction is in the EX stage;
- `inst_i[31:0]`: full instruction word (opcode, funct3, funct7);
- `rs1_i`, `rs2_i`: source register values;
- `result_o`: returned to the destination register;
- `valid_o`: asserted when the result is ready; the CPU stalls until this signal is seen.

A 1-cycle CFU instruction has no pipeline penalty. Multi-cycle instructions stall the CPU for its duration, which is transparent to the programmer. The interface is accessed via four RISC-V opcode spaces: `custom-0` (0x0B), `custom-1` (0x2B), `OP-32` (0x3B), and `OP-IMM-32` (0x1B).

### C. Related Work

Custom ISA extensions for RISC-V targeting wireless DSP have been studied by Belhadj Amor et al. [6] (19–68 % cycle reduction for LoRa/FSK, commercial toolchain) and Gautschi et al. [7] (3.5× speedup for IoT sensor processing via near-threshold DSP extensions). The RISC-V "P" extension [8] can also be useful for this goal, but remains unratified and impractical for MCU-class targets. On the other hand, FPGA-based FIR engines for SDR [9] achieve higher throughput at the cost of programmability. From these obervations, we adopt the NEORV32 CFU without toolchain modifications, specifically addressing the RRC matched filter of 3GPP waveforms.

## III. Architecture

### A. Custom Instruction Encoding

The implemented instructions are `fmac`, `fmac2`, and `fmac_res` which reads the final Q30 product and casts it to Q15 format. All three CFU instructions use the `custom-0` major opcode (`0x0B`). The `funct3` field selects the operation; `funct7` is reserved (set to 0). Table II lists the encoding.

TABLE II: CFU Instruction Encodings (custom-0 opcode = `0x0B`)

| Mnemonic | funct3 | Cy | Operation |
|---|---|---|---|
| `fmac` | `000` | 2 | acc += rs1[15:0] × rs2[15:0] |
| `fmac2` | `001` | 2 | acc += rs1[31:16]×rs2[15:0] +rs1[15:0]×rs2[15:0] |
| `fmac_res` | `010` | 1 | rd ← acc[46:15]; acc ← 0 |

All operands are Q15 signed fixed-point (16-bit, range $[-1, +1)$). The internal accumulator is 48-bit signed; since a Q15×Q15 product occupies 32 bits (Q30), the remaining 16 bits provide headroom for accumulation, sufficient for up to $2^{16} = 65536$ taps without overflow under worst-case coefficient magnitudes.

### B. FMAC2 and Filter Symmetry

For a linear-phase RRC filter with $N$ (odd) taps, the firmware loop for FMAC2 is:

$$\texttt{fmac2}\big(\{x[n-k],\, x[n-(N-1-k)]\},\, h[k]\big), \quad k = 0, \ldots, \lfloor N/2 \rfloor - 1. \quad (3)$$

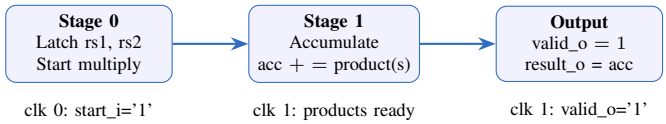



Fig. 2: FMAC/FMAC2 two-stage pipeline
. `fmac_res` completes in a single clock cycle (combinatorial path through the accumulator).

The centre tap $k = \lfloor N/2 \rfloor$ is handled with a plain `fmac`. This reduces the instruction count from $N$ to $\lceil N/2 \rceil + 1$ (including the final `fmac_res`).

### C. VHDL Microarchitecture

The CFU implements a 2-stage pipeline, as shown in Fig. 2.

Listing 1 shows the multiply-accumulate core of the VHDL pipeline (the `fmac_res` readout path is omitted here; it is listed in Table II). The multiply stages are inferred as DSP48E1 primitives by Vivado (FMAC: one DSP48; FMAC2: two DSP48s in parallel).

```
-- Stage 0 -> 1: latch operands + multiply
s1.prod_a <= resize(rs1_lo * rs2_lo, 32);
if funct3 = F3_FMAC2 then
  s1.prod_b <= resize(rs1_hi * rs2_lo, 32);
end if;

-- Stage 1 -> 2: accumulate
acc <= acc + resize(s1.prod_a, 48)
           + resize(s1.prod_b, 48);
```

Listing 1: FMAC/FMAC2 multiply-accumulate core (simplified)

## IV. Firmware Interface

Instructions are invoked without toolchain modifications via the NEORV32 intrinsic template `RISCV_INSTR_R_TYPE`, which emits a standard `.insn` directive [10] targeting the `custom-0` major opcode [11]:

```
// acc += (int16)rs1[15:0] * (int16)rs2[15:0]   (2 cy)
#define cfu_fmac(x, h) \
  RISCV_INSTR_R_TYPE(RISCV_OPCODE_CUSTOM0, 0b000, 0, x, h
      )

// acc += rs1[31:16]*rs2[15:0] + rs1[15:0]*rs2[15:0]   (2
    cy)
#define cfu_fmac2(packed_xy, h) \
  RISCV_INSTR_R_TYPE(RISCV_OPCODE_CUSTOM0, 0b001, 0,
      packed_xy, h)

// rd = acc >> 15; acc = 0   (1 cy)
#define cfu_fmac_res() \
  RISCV_INSTR_R_TYPE(RISCV_OPCODE_CUSTOM0, 0b010, 0, 0,
      0)
```

Listing 2: C intrinsic wrappers for FMAC, FMAC2, and FMAC_RES

## V. Measurements and Results

### A. Measurement Setup

Cycle counts were measured on an Arty A7-35T board (Artix-7 xc7a35t, 100 MHz) with the NEORV32 booting from on-chip IMEM. The `mcycle` CSR (`0xB00`, RISC-V Zicntr) is read via `csrr` before and after the benchmark; the difference divided by $N_s = 256$ gives cycles per sample. Tap counts $N \in \{9, 17, 33, 65\}$ are tested with a linear shift-register buffer (includes memmove overhead). Results are stable across 20 consecutive runs with zero variance. Area figures use Vivado 2024.2 post-route Out-Of-Context (OOC) synthesis at 100 MHz.

TABLE III: Cycles per Output Sample ($N_s = 256$, NEORV32 @ 100 MHz)

| N | Baseline | FMAC | FMAC2 | ×FMAC | ×FMAC2 |
|---|---|---|---|---|---|
| 9 | 1433 | 867 | 824 | 1.65× | 1.74× |
| 17 | 2585 | 1587 | 1480 | 1.63× | 1.75× |
| 33 | 4889 | 3027 | 2792 | 1.62× | 1.75× |
| 65 | 9497 | 5907 | 5416 | 1.61× | 1.75× |

TABLE IV: Resource Utilization — Artix-7 xc7a35t, post-route Vivado 2024.2, 100 MHz constraint (OOC synthesis)

| Configuration | LUT | FF | DSP48 | BRAM | $F_{\max}$ |
|---|---|---|---|---|---|
| Baseline (RV32IMC) | 1855 | 1292 | 0 | 5 | 106.5 MHz |
| +FMAC | 1916 | 1405 | 1 | 5 | 109.9 MHz |
| +FMAC2 | 1955 | 1438 | 2 | 5 | 106.2 MHz |

### B. Cycle Count Comparison

Table III shows that both FMAC and FMAC2 deliver consistent speedups across all tap counts. The baseline cycle count grows linearly with $N$, as expected for a pure load-multiply-accumulate loop. FMAC reduces cycles by 38–40 %: by eliminating the register-file round-trip of a software MAC but loop control overhead persists. FMAC2 achieves a larger reduction (42–43 %) by halving the number of CFU calls through symmetry packing; the nearly constant speedup ratio across all $N$ ($\approx 1.75\times$) reflects the shift-register overhead scaling proportionally with tap count.

**Analysis.** Hardware measurements confirm FMAC2 delivers a $1.75\times$ speedup for $N = 33$. A separate RTL simulation with a leaner circular buffer (Table I) projects a reduction from 1509 to 946 cycles/sample, cutting CPU utilisation from 90.5 % to 56.8 % at 60 ksps and leaving 721 cycles for timing recovery, CFO correction, and demodulation within the same processor.

### C. Resource Utilization

Table IV summarizes Vivado 2024.2 post-route results for the Artix-7 xc7a35t with a 100 MHz constraint.

FMAC adds one DSP48E1 slice (the Q15×Q15 multiplier). FMAC2 adds two DSP48E1 slices in parallel, processing two taps per clock. The 48-bit accumulator register adds $\sim 50$ flip-flops but minimal LUT overhead. The $F_{\max}$ impact is within post-route placement noise: FMAC shows a $+3.4$ MHz increase (106.5→109.9 MHz) and FMAC2 a $-0.3$ MHz decrease (106.5→106.2 MHz) relative to baseline, both consistent with the critical path remaining within the NEORV32 CPU control logic rather than the CFU multiplier chain. Resource figures in Table IV are post-route measurements (Vivado 2024.2, OOC

mode, xc7a35t-1). The delta over baseline is +61 LUT / +113 FF / +1 DSP48 for FMAC and +100 LUT / +146 FF / +2 DSP48 for FMAC2, confirming minimal area overhead.

### D. Throughput and Standard Compliance

At 100 MHz, the maximum sustainable sample rate with the FMAC2 kernel ($N = 33$) is:

$$f_s^{\max} = \frac{100 \times 10^6}{946} \approx 105.7 \text{ ksps}. \quad (4)$$

At $L = 4\times$ oversampling this corresponds to a symbol rate of $\approx 26.4$ ksym/s. For standard LTE at 15 kHz subcarrier spacing [3] ($f_s = 60$ ksps), the per-sample budget is $\lfloor 10^8/6\times10^4 \rfloor = 1667$ cycles.

FMAC2 consumes 946 of these (56.8 %), leaving 721 cycles for synchronisation and demodulation — sufficient to cover the Gardner TED (169 cy), NCO-based CFO correction (116 cy), and QPSK demodulation (<5 cy) reported in Table I, with $\approx$431 cycles of spare headroom. The baseline implementation requires 1509 cycles for the filter alone ($>90$ % of the budget), leaving essentially no margin for the rest of the receiver stack.

A complete I+Q receiver applies the FMAC2 kernel to both channels together, measured at 2418 cy/sample in RTL simulation of the integrated chain (Table I), which exceeds the 60 ksps single-core budget by 45.0 %; real-time I+Q operation requires interleaved scheduling or a reduced symbol rate, while the single-channel metrics demonstrate the acceleration headroom that FMAC2 provides.

### E. End-to-End QPSK Chain Validation

To confirm that the Q15 arithmetic in the CFU introduces no systematic numerical error, the complete SDR chain was simulated in software: QPSK modulation, RRC pulse shaping ($\alpha = 0.35$, $N = 33$, $L = 4$), AWGN channel, RRC matched filtering via FMAC2, Gardner timing error detector, and hard-decision demodulation. Measured BER tracks the theoretical curve $P_e = Q\left(\sqrt{2E_b/N_0}\right)$ within the statistical noise of a 496-bit block: at $E_b/N_0 = 0, 3, 6, 9$ dB the measured/theoretical pairs are $0.147/0.159$, $0.075/0.079$, $0.028/0.023$, and $0.006/0.002$, respectively, and both fall to zero detected errors for $E_b/N_0 \geq 12$ dB. This confirms that (i) the 48-bit accumulator provides sufficient headroom to prevent overflow across the 33-tap sum, and (ii) Q15 coefficient quantisation does not introduce a BER floor at the evaluated operating points. The agreement at low SNR (0 and 3 dB) additionally validates the Gardner timing loop operating on the FMAC2 filter output.

## VI. Conclusion

This paper presented `fmac` and `fmac2`, two custom instructions for the NEORV32 processor that accelerate RRC matched filtering without modifying the pipeline, register file, or compiler. Both are realized in the CFU as a 2-stage pipelined Q15 MAC; `fmac2` exploits linear-phase symmetry to pair mirror taps and compute two products per cycle.

On a 100 MHz Artix-7 FPGA, `fmac2` achieves a 1.75× speedup over baseline RV32IMC for a 33-tap filter, reducing CPU utilization from 90.5 % to 56.8 % at 60 ksps and leaving 721 cycles for tracking and demodulation. Post-route synthesis confirms minimal area impact (+100 LUT / +146 FF / +2 DSP48E1), consuming $< 10\%$ of available fabric, and timing closure is maintained at 100 MHz. End-to-end BER simulation validates the arithmetic against theoretical QPSK performance.

These single-channel figures bound one filtering path; a complete in-phase/quadrature receiver applying `fmac2` to both channels requires 2418 cy/sample, exceeding the 60 ksps single-core budget by 45.0 %, so real-time dual-channel operation requires interleaved scheduling, a reduced symbol rate, or a second core — a direction left for future work.

These results demonstrate that domain-specific ISA extensions via the NEORV32 CFU interface deliver meaningful acceleration on open-source cores with a negligible hardware footprint—offering a practical path for reconfigurable, IoT-class SDR architectures.

## Acknowledgment

Authors thank Universidad Tecnologica Nacional for the support of the present work as part of its Research and Development Projects (Proyectos de Investigacion y Desarrollo, PIDs).

## References

[1] A. P. Vinod and E. M.-K. Lai, "Low power and high-speed implementation of FIR filters for software defined radio receivers," *IEEE Trans. Wireless Commun.*, vol. 5, no. 7, pp. 1669–1675, 2006.

[2] Y. Lin, H. Lee, M. Woh, Y. Harel, S. Mahlke, T. Mudge, C. Chakrabarti, and K. Flautner, "SODA: A low-power architecture for software radio," in *Proc. 33rd Int. Symp. Computer Architecture (ISCA)*, 2006, pp. 89–101.

[3] 3GPP, *TS 36.211 v17.3.0: LTE – Physical Channels and Modulation*, 3rd Generation Partnership Project Std., 2022. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/36_series/36.211/

[4] S. Nolting, "NEORV32 – a size-optimized, customizable RISC-V soft-core processor," GitHub, 2020. [Online]. Available: https://github.com/stnolting/neorv32

[5] 3GPP, *TS 38.211 v17.3.0: NR – Physical Channels and Modulation*, 3rd Generation Partnership Project Std., 2022. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.211/

[6] H. Belhadj Amor, C. Bernier, and Z. Přikryl, "A RISC-V ISA extension for ultra-low power IoT wireless signal processing," *IEEE Trans. Comput.*, 2021.

[7] M. Gautschi, P. D. Schiavone, A. Traber, I. Loi, A. Pullini, D. Rossi, E. Flamand, F. K. Gürkaynak, and L. Benini, "Near-threshold RISC-V core with DSP extensions for scalable IoT endpoint devices," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 10, pp. 2700–2713, 2017.

[8] RISC-V International, "RISC-V P (packed-simd / DSP) extension proposal," GitHub, 2021. [Online]. Available: https://github.com/riscv/riscv-p-spec

[9] V. P. Nambiar, M. Khalil-Hani, and M. N. Marsono, "Reconfigurable FPGA-based FIR filter for SDR applications," *IEEE Access*, vol. 9, pp. 120 968–120 984, 2021.

[10] GNU Binutils, "RISC-V assembler `.insn` directive," GNU Binutils manual, 2023. [Online]. Available: https://sourceware.org/binutils/docs/as/RISC_002dV_002dFormats.html

[11] RISC-V International, *The RISC-V Instruction Set Manual, Volume I: User-Level ISA*, 2nd ed., 2017. [Online]. Available: https://github.com/riscv/riscv-isa-manual

Categoría

Artículo

Software Embebido

Área Temática

# SoC-FPGA Based Acquisition and Clarke Preprocessing of Three-Phase Currents

I. A. Juarez-Trujillo
(IEEE Student Member)
*Electronics Engineering Department, TecNM/CENIDET, Cuernavaca, Morelos, México*
*ivan.jtp@cenidet.tecnm.mx*

J. G. Velásquez-Aguilar
*Faculty of Chemical Sciences and Engineering, UAEM, Cuernavaca, Morelos, México*
*jgpeva@uaem.mx*

S. E. D. L. Aldaco
(IEEE Senior Member)
*Electronics Engineering Department, TecNM/CENIDET, Cuernavaca, Morelos, México*
*susana.da@cenidet.tecnm.mx*

*J. A. Alquicira*
*(IEEE Senior Member)*
*Electronics Engineering Department, TecNM/CENIDET, Cuernavaca, Morelos, México*
*jesus.aa@cenidet.tecnm.mx*

***Abstract*—This paper presents a graphical interface for the comparative analysis of three-phase induction-motor currents previously acquired through an architecture based on programmable logic and an embedded processor, and subsequently exported to spreadsheet files. The interface integrates dataset loading and validation, optional direct-current component removal, the Clarke transformation, and the comparison of four preprocessing techniques: unfiltered signal, Butterworth filtering, moving average, and Haar-wavelet thresholding. It also enables visualization of the original currents, the transformed components in the $\alpha\beta 0$ domain, and the trajectory in the $\alpha\beta$ plane, while computing quantitative metrics such as root-mean-square (RMS) values, the mean zero-sequence component, and processing time per block. Experimental validation was carried out with two induction motors under two acquisition scenarios of different duration, making it possible to identify differences among operating conditions and signal-conditioning strategies. The results show that the interface constitutes a useful and flexible tool for comparative analysis oriented toward induction-motor diagnosis.**

## REFERENCES

[1] G. H. Bazan, A. Goedtel, O. Duque-Perez, and D. Morinigo-Sotelo, "Multi-Fault Diagnosis in Three-Phase Induction Motors Using Data Optimization and Machine Learning Techniques," *Electronics*, vol. 10, no. 12, p. 1462, 2021, doi: 10.3390/electronics10121462.

[2] W. Liu *et al.*, "Machine Learning for Fault Diagnosis of Electric Motors in Actuator Systems: A Review," *Actuators*, vol. 14, no. 12, p. 596, 2025, doi: 10.3390/act14120596.

[3] I. A. Juárez-Trujillo, S. E. De León-Aldaco, J. G. Velásquez-Aguilar, and J. A. Aguayo-Alquicira, "Fault Analysis in Induction Motors Through Signal Acquisition With Hilbert Transform and FPGA," *IEEE Embedded Systems Letters*, 2025, doi: 10.1109/LES.2025.3546847.

[4] E. de J. Carrera Avendaño, I. A. Juarez Trujillo, M. Borunda, C. D. García Beltrán, J. G. Velásquez Aguilar, A. Acevedo Quiroz, and S. E. D. L. Aldaco, "Clarke-Domain Dyadic Wavelet Denoising for Three-Phase Induction Motor Current Signals," *Processes*, vol. 14, no. 6, p. 950, 2026, doi: 10.3390/pr14060950.

[5] C. J. O'Rourke, M. M. Qasim, M. R. Overlin, and J. L. Kirtley, "A Geometric Interpretation of Reference Frames and Transformations: dq0, Clarke, and Park," *IEEE Transactions on Energy Conversion*, vol. 34, no. 4, pp. 2070–2083, Dec. 2019, doi: 10.1109/TEC.2019.2941175.

[6] F. G. Montoya and A. H. Eid, "Formulating the geometric foundation of Clarke, Park, and FBD transformations by means of Clifford's geometric algebra," *Mathematical Methods in the Applied Sciences*, vol. 45, no. 8, pp. 4252–4277, 2022, doi: 10.1002/mma.8038.

[7] Guangdong ETCR Electronic Technology Co., Ltd., "ETCR070A split-core clamp current sensor," Guangzhou, China. [Online]. Available: https://www.etcrmeter.com/, 2026.

[8] Texas Instruments, "INA114 Precision Instrumentation Amplifier," Datasheet, Rev. B, Dallas, TX, USA, Jan. 2026.

[9] Intel, "Cyclone® V FPGA and SoC FPGA," Santa Clara, CA, USA. [Online]. Available: https://www.intel.com/content/www/us/en/products/programmable/soc/cyclone-v.html, 2026.

[10] Terasic Technologies, "DE10-Nano Development and Education Board," Hsinchu, Taiwan. [Online]. Available: https://www.terasic.com.tw/cgi-bin/page/archive.pl?CategoryNo=167&Language=English&No=1046, 2026.

# Data-Reuse RLS Based Model-Free Predictive Control for Voltage Source Inverters

F. Ruiz*, Á. A. Vázquez †, P. Burgos *, M. Aly *, J. G. Avalos ‡, G. Sánchez ‡, J. Rodríguez *
*Centro de Transición Energética (CTE), Facultad de Ingeniería, Universidad San Sebastián, Chile
†Tecnológico de Monterrey, Institute of Advanced Materials for Sustainable Manufacturing, México
‡Instituto Politécnico Nacional, SEPI ESIME Culhuacan, México
{felipe.ruiz, mokhtar.aly, jose.rodriguezp}@uss.cl, avazquezp@tec.mx, pburgosn@correo.uss.cl, {javaloso, gsanchezriv}@ipn.mx

***Abstract*—This paper presents an improved model-free finite control set model predictive control (MF-FCSMPC) strategy for three-phase voltage source inverters. The approach combines a data-reuse recursive least squares (DR-RLS) algorithm with an autoregressive exogenous (ARX) plant model. Unlike conventional RLS, the DR-RLS method reuses the same input vector to update system coefficients multiple times. This accelerates convergence and improves tracking accuracy. The DR-RLS MF-FCSMPC system delivers robust predictive control across varying operating conditions. Hardware-in-the-loop tests with two-level inverters and RL loads confirm reduced RMS error under parameter variations. Overall, the DR-RLS-based MF-FCSMPC provides a reliable and computationally efficient solution for predictive control in power electronic systems facing uncertainty and time-varying dynamics.**

***Index Terms*—ARX plant model, data-reuse RLS, model-free control, model predictive control, recursive least squares (RLS).**

## REFERENCES

[1] C. Garcia, A. Mora, M. Norambuena, J. Rodriguez, M. Aly, F. Carnielutti, J. Pereda, P. Acuna, R. Aguilera, and L. Tarisciotti, "Model predictive control in multilevel inverters part i: Basic strategy and performance improvement," *IEEE Open Journal of Industry Applications*, vol. 5, p. 428–441, 2024.

[2] M. Norambuena, A. Mora, C. Garcia, J. Rodriguez, M. Aly, F. Carnielutti, J. Pereda, C. Castillo, Z. Zhang, V. Yaramasu, L. Tarisciotti, and Y. Yin, "Model predictive control in multilevel inverters part ii: Renewable energies and grid applications," *IEEE Open Journal of Industry Applications*, vol. 5, p. 414–427, 2024.

[3] I. Harbi, J. Rodriguez, E. Liegmann, H. Makhamreh, M. L. Heldwein, M. Novak, M. Rossi, M. Abdelrahem, M. Trabelsi, M. Ahmed, P. Karamanakos, S. Xu, T. Dragičević, and R. Kennel, "Model-predictive control of multilevel inverters: Challenges, recent advances, and trends," *IEEE Transactions on Power Electronics*, vol. 38, no. 9, p. 10845–10868, Sep. 2023.

[4] F. Wang, Y. Wei, J. Rodriguez, and C. Garcia, "State-of-art, development, and challenges of model-free predictive control on motor drives," *IEEE Transactions on Power Electronics*, vol. 40, no. 8, p. 10846–10864, Aug. 2025.

[5] M. Huang, H. Zhang, K. Wang, Z. Yao, and W. Wu, "Model-free predictive control based on rls algorithm for grid-forming inverters with virtual synchronous generator," *IEEE Access*, vol. 13, pp. 57 921–57 931, 2025.

[6] N. Yang, S. Zhang, X. Li, and X. Li, "A new model-free deadbeat predictive current control for pmsm using parameter-free luenberger disturbance observer," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 11, no. 1, pp. 407–417, 2023.

[7] M. Khalilzadeh, S. Vaez-Zadeh, J. Rodriguez, and R. Heydari, "Model-free predictive control of motor drives and power converters: A review," *IEEE Access*, vol. 9, pp. 105 733–105 747, 2021.

[8] Y. Wei, H. Young, D. Ke, F. Wang, H. Qi, and J. Rodríguez, "Model-free predictive control using sinusoidal generalized universal model for pmsm drives," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 11, p. 13720–13731, Nov. 2024.

[9] Y. Wei, W. Ma, H. Xie, D. Ke, and F. Wang, "Affine ultra-local based model-free predictive control on pmsm drives with $\varepsilon$-approximation state compensation," in *2024 IEEE 13th Data Driven Control and Learning Systems Conference (DDCLS)*. IEEE, May 2024, p. 595–600.

[10] M. Kermadi, A. Rebai, L. Baghli, N. Mesbahi, S. Mekhilef, M. Mubin, and H. Mokhlis, "Model-free predictive current controller for voltage source inverters using arx model and recursive least squares," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 71, no. 5, p. 2619–2623, May 2024.

[11] R. Heydari, H. Young, F. Flores-Bahamonde, S. Vaez-Zadeh, C. Gonzalez-Castano, S. Sabzevari, and J. Rodriguez, "Model-free predictive control of grid-forming inverters with lcl filters," *IEEE Transactions on Power Electronics*, vol. 37, no. 8, pp. 9200–9211, 2022.

[12] J. Rodríguez, R. Heydari, Z. Rafiee, H. A. Young, F. Flores-Bahamonde, and M. Shahparasti, "Model-free predictive current control of a voltage source inverter," *IEEE Access*, vol. 8, pp. 211 104–211 114, 2020.

[13] C. Gonzalez-Castano, M. Norambuena, A. Navas-Fonseca, F. Flores-Bahamonde, S. Alireza Davari, H. Young, R. Heydari, and J. Rodriguez, "Model-free predictive control of five-level t-type nested neutral point clamped converter," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 13, no. 2, pp. 1969–1979, 2025.

[14] S. Shaffer and C. Williams, "Comparison of lms, alpha lms, and data reusing lms algorithms," in *Conference Record of the Seventeenth Asilomar Conference on Circuits, Systems and Computers*, 1983, pp. 260–264.

[15] C. Paleologu, J. Benesty, and S. Ciochină, "Data-reuse recursive least-squares algorithms," *IEEE Signal Processing Letters*, vol. 29, pp. 752–756, 2022.

# Framework orientado a eventos con período dinámico aplicado en electrónica de potencia

Román Comelli, Sergio Junco
*Laboratorio de Automatización y Control, FCEIA, UNR*
Rosario, Argentina
{rcomelli,sjunco}@fceia.unr.edu.ar

***Resumen*—Este trabajo presenta un *framework* basado en eventos y máquinas de estado finito jerárquicas (HSM) que cuenta con un servicio dinámico de generación de eventos temporales. En lugar de utilizar un temporizador con una configuración estática que interrumpa periódicamente al procesador para acumular pequeñas unidades de tiempo por software, en el enfoque propuesto es el propio hardware el encargado de llevar dicha cuenta. Una vez alcanzado el instante correspondiente, una rutina de servicio de interrupción se encarga de entregar los eventos temporales a los módulos de firmware que los hayan solicitado.**

**Este esquema surgió durante el desarrollo de un cargador de baterías Li-Ion/Li-Po basado en una topología *buck-boost* de cuatro llaves. La implementación se probó exitosamente sobre el hardware tras un exhaustivo testeo del tipo *processor-in-the-loop* (PIL).**

***Index Terms*—Período Dinámico, Eventos, HSM, PIL**

## Agradecimientos

Los autores desean expresar su agradecimiento al fallecido Prof. Ing. Daniel Alba por sus invaluables aportes.

## Referencias

[1] E. White, *Making embedded systems*. O'Reilly Media, Inc., 2024.
[2] B. P. Douglass, *Design patterns for embedded systems in C: an embedded software engineering toolkit*. Elsevier, 2010.
[3] P. I. D. Martos, "Identificación y clasificación de patrones de aplicación en sistemas embebidos," Ph.D. dissertation, Universidad Nacional de La Plata, 2015.
[4] B. Amos, *Hands-On RTOS with Microcontrollers: Building real-time embedded systems using FreeRTOS, STM32 MCUs, and SEGGER debug tools*. Packt Publishing Ltd, 2020.
[5] J. Ganssle, *The art of designing embedded systems*. Newnes, 2008.
[6] L. Schuermann, B. Campbell, B. Ghena, P. Levis, A. Levy, and P. Pannuto, "Tock: From research to securing 10 million computers," in *Proceedings of the ACM SIGOPS 31st Symposium on Operating Systems Principles*, 2025, pp. 36–49.
[7] A. Levy, B. Campbell, B. Ghena, D. B. Giffin, P. Pannuto, P. Dutta, and P. Levis, "Multiprogramming a 64kb computer safely and efficiently," in *Proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 234–251.
[8] E. Gamma, *Design patterns: elements of reusable object-oriented software*. Pearson Education India, 1995.
[9] J. L. Aracil Gómez del Campo, "pubsub-c: Pub/sub library for c," https://github.com/jaracil/pubsub-c, 2019, accessed: 2026-05-10.
[10] D. C. Schmidt, M. Stal, H. Rohnert, and F. Buschmann, *Pattern-oriented software architecture, patterns for concurrent and networked objects*. John Wiley & Sons, 2013.
[11] M. Samek, *Practical UML statecharts in C/C++: event-driven programming for embedded systems*. CRC Press, 2008.
[12] M. Samek and P. Montgomery, "State oriented programming," *Embedded Systems Programming*, vol. 13, no. 8, pp. 22–43, 2000.
[13] D. Harel, "Statecharts: A visual formalism for complex systems," *Science of computer programming*, vol. 8, no. 3, pp. 231–274, 1987.
[14] D. M. Cummings, "Managing concurrency in complex embedded systems," https://www.state-machine.com/doc/Cummings1006.pdf, 2014, accessed: 2026-05-11.
[15] J. Stultz, N. Aravamudan, and D. Hart, "We are not getting any younger: A new approach to time and timers," in *Proceedings of the Linux Symposium*, vol. 1, 2005, pp. 219–232.
[16] T. Gleixner and D. Niehaus, "Hrtimers and beyond: Transforming the linux time subsystems," in *Proceedings of the Linux symposium*, vol. 1, 2006, pp. 333–346.
[17] N. Biradar, "Uml-state-machine-in-c : A minimalist uml state machine framework for finite state machine and hierarchical state machine in c," https://github.com/kiishor/UML-State-Machine-in-C, 2018, accessed: 2026-05-10.

Categoría

# Foro Tecnológico

Aplicaciones agro y alimentos

Área Temática

ISBN 978-631-90145-4-9 - Congreso Argentino de Sistemas Embebidos CASE 2026 - 6 y 7 de agosto

# Sistema para la medición, registro y control de variables asociadas al cultivo interior de cannabis

Ignacio Scandura[1], Alexander Aguila Wharton[1], Luciano Protti Cosenza[1], Luciana Mentasti[1-2], Rúal E. Romero[3], Agustina Fernández[1-2], María L. Franchi[1-2], Silvano R. Rossi[1-3], Gastón Barreto[1-2]

[1]*CIFICEN (UNCPBA-CICPBA-CONICET)*

Olavarría, Argentina

iscandura@fio.unicen.edu.ar; {alexawpeter, lprotticosenza, lucianamentasti}@gmail.com; afernandez@fio.unicen.edu.ar

[2]*Facultad de Ingeniería, UNCPBA, Núcleo INMAT*

Olavarría, Argentina

{luisa.franchi, gbarreto}@fio.unicen.edu.ar

[3]*Facultad de Ingeniería, Núcleo INTELYMEC*

Olavarría, Argentina

{rromero, srossi}@fio.unicen.edu.ar

***Resumen*—En este trabajo se abordan los aspectos más importantes del diseño y desarrollo de un sistema de medición, registro y control de variables físicas y factores críticos asociados al cultivo indoor de cannabis con fines medicinales. El trabajo se centra en la integración de tecnologías que permitan configurar un entorno autónomo y de bajo costo, para facilitar la integración, flexibilidad y tratamiento eficiente de los datos obtenidos. Este enfoque busca garantizar condiciones óptimas de crecimiento, favoreciendo la calidad y estabilidad del cultivo, al tiempo que posibilita un seguimiento sistemático con propósitos científicos y de investigación aplicada. Se presentan aspectos funcionales y resultados de ensayos realizados.**

***Palabras clave—cultivo indoor, cannabis, medición y control, variables físicas***

## I. Introducción

El estudio de los sistemas de cultivo en espacios cerrados y/o invernaderos bajo condiciones controladas se ha consolidado como un área de creciente interés en la investigación agrícola contemporánea [1]. Dichos recintos se conciben y diseñan para proteger los cultivos frente a condiciones adversas, viabilizar la producción fuera de estación y brindar entornos adecuados para las especies a cultivar [2], [3]. A lo largo del tiempo, los invernaderos evolucionaron integrando equipamientos y diversos sistemas con el objetivo de mejorar las condiciones del ambiente donde se desarrolla el cultivo [4]. Por ello, resulta necesario implementar sistemas de medición y control, los que pueden poseer diferentes grados de tecnificación y automatización, para generar las condiciones ambientales adecuadas de temperatura, humedad y luminosidad, entre otras variables, para el cultivo en interiores (*indoor*) [5], [6].

En particular, el interés creciente en el uso medicinal del *Cannabis sativa L.* impulsó avances en su cultivo en ambientes controlados. Desde principios del siglo XXI, el cultivo de cannabis con fines medicinales fue ganando popularidad en todo el mundo, consolidándose especialmente en América del Norte [7]. A la luz de los lineamientos definidos en Argentina [8], se abren oportunidades importantes para el desarrollo de tecnología asociada al cultivo de cannabis en nuestro país. La modalidad del cannabis medicinal es un campo desafiante para el desarrollo de sistemas de medición, registro y control de las variables que afectan su crecimiento. La posibilidad de lograr un sistema indoor completamente autónomo, con diferentes tipos de conectividad y capaz de producir datos que brinden información relevante para el registro, permite no solo alcanzar un crecimiento de cultivo en condiciones adecuadas, fortaleciendo y otorgando las propiedades necesarias para su desarrollo, sino también lograr un seguimiento del cultivo con fines de investigación. Motivados por estas premisas, se presentan los aspectos más importantes del diseño e implementación de un sistema que integra tecnologías para la medición, registro y control de las variables físicas críticas para el desarrollo del cultivo de cannabis, en un espacio interior controlado (contenedor comercial adaptado) que se ubica en el campus de una universidad pública.

## II. Marco Teórico

Las variables físicas ambientales como la intensidad lumínica, la temperatura, la humedad, factores como la concentración de $CO_2$ en el recinto, la cantidad de agua proporcionada y la nutrición, tienen una influencia determinante en el crecimiento de las plantas en espacios interiores [9], [10]. Su medición debe hacerse de la forma más integrada posible, hecho que se ve favorecido en el caso del cultivo indoor, al poder medir y controlar el ambiente con las facilidades de las tecnologías actuales.

En los entornos de cultivo de *Cannabis sativa L.*, cuando la temperatura desciende por debajo de ciertos umbrales críticos (generalmente en torno a 15 °C para esta especie), el crecimiento vegetativo puede ralentizarse significativamente o detenerse. Por otro lado, los valores excesivos de temperatura (mayores a 30–32 °C para esta especie) pueden favorecen el marchitamiento y el estrés fisiológico [7], [11]. Los requerimientos ambientales varían significativamente según la fase de desarrollo del cultivo. En particular, durante la etapa de propagación vegetativa por esquejes o plántulas, la literatura científica y guías técnicas de producción en ambientes controlados coincide en la necesidad de mantener condiciones que minimicen la pérdida de agua y el estrés fisiológico. Se reportan valores orientativos de temperatura ambiente en el rango de 20 a 24 °C, combinado con una humedad relativa elevada mayor al 90 % y una intensidad lumínica relativamente baja, con densidades de flujo de fotones fotosintéticos (PPFD) entre 50 y 150 $\mu mol \cdot m^{-2} \cdot s^{-1}$. Estas condiciones tienden a reducir la transpiración y favorecer la formación de raíces adventicias [12], [13], [14]. Superada esta etapa, durante el crecimiento vegetativo las temperaturas óptimas se sitúan entre 20 y 25 °C, con mayores

niveles de radiación fotosintéticamente activa, mientras que en la fase de floración el cultivo responde favorablemente a temperaturas en el rango de 23 a 28 °C a intensidades lumínicas altas, mayores a 600 $\mu mol \cdot m^{-2} \cdot s^{-1}$ [15]. La temperatura también es una variable de influencia en la transición del crecimiento vegetativo al reproductivo, ya que, dependiendo de las condiciones ambientales y del genotipo, puede acelerar o retrasar el inicio de la floración [9].

Asimismo, la humedad relativa constituye un factor crítico en el cultivo de *Cannabis sativa L.* Niveles elevados, superiores al 70-80 %, a excepción de la fase de plántulas, incrementan el riesgo de condensación sobre las inflorescencias y hojas, creando un microambiente propicio para el desarrollo de patógenos fúngicos, que afectan la calidad y la seguridad del producto final [16], [17]. Mantener rangos óptimos de temperatura y humedad relativa entre 40 y 60 % durante la floración resulta esencial para minimizar la incidencia de contaminantes microbiológicos y asegurar un desarrollo fisiológico adecuado de las plantas, así como una producción homogénea y de alta calidad [18], [19].

Como se ha mencionado, la intensidad lumínica es uno de los factores más determinantes para el rendimiento del cannabis, con rangos definidos, siempre que la temperatura, la humedad y la concentración de $CO_2$ se mantengan en rangos óptimos. De acuerdo con Chandra et al. [12], [20], el enriquecimiento de $CO_2$ junto con una adecuada gestión de luz y temperatura puede potenciar la fotosíntesis y productividad de *Cannabis sativa L*. en sistemas controlados de cultivos indoor. Las Concentraciones de $CO_2$ consideradas elevadas (≈700 ppm) mejoran significativamente la fotosíntesis y la eficiencia en el uso del agua, aunque el efecto depende de la variedad. El enriquecimiento atmosférico hasta concentraciones de $CO_2$ consideradas elevadas de 800 a 1000 ppm aumenta significativamente la tasa de fotosíntesis, pero solo cuando el nivel de PPFD es superior a 500 $\mu mol \cdot m^{-2} \cdot s^{-1}$. Durante el período de oscuridad no se debe añadir $CO_2$, ya que la planta no realiza fotosíntesis [20], [21].

En relación al riego bajo ambientes controlados, el cultivo en sustratos inertes es uno de los métodos más utilizados a escala comercial, debido a su estabilidad físico-química, uniformidad y adecuada capacidad de drenaje. Este tipo de manejo de riego se implementa mediante aplicaciones frecuentes y de bajo volumen, ajustadas a la demanda del cultivo, con el objetivo de mantener condiciones óptimas de humedad y aireación en la zona radicular, evitando tanto el déficit hídrico como la saturación del sustrato [22].

La integración de los factores analizados, permiten maximizar el rendimiento y la calidad de las inflorescencias en cultivo indoor. El análisis anterior implica que, junto al sistema de medición de las variables, debe pensarse y diseñarse el uso de sistemas de iluminación, riego, ventilación, calefacción y humidificación para mantener los rangos adecuados. Recientemente, Vernon et al. [23] presentaron una revisión de tecnologías y prácticas actuales empleadas en el cultivo comercial de *Cannabis sativa* L., bajo condiciones de invernadero, destacando las variables ambientales y de proceso a monitorear para optimizar la producción. Zheng et al. [24] proponen un enfoque basado en IoT para el diseño de sistemas de monitoreo en invernaderos, orientado a la adquisición y gestión en tiempo real de variables críticas del cultivo. Ordoñez-Quintero et al. [25] presentaron un prototipo de bajo costo basado en IoT para la automatización del cultivo de cannabis medicinal en invernadero, demostrando la viabilidad de soluciones accesibles para el control de variables ambientales. De manera complementaria, Menoni et al. [26] desarrollaron un sistema de monitoreo y control para cultivo indoor de cannabis, en el cual se supervisan variables ambientales clave y se implementan lazos de control que permiten automatizar diversas tareas del sistema productivo. Muchas de estas soluciones se encuentran aún en fase de prototipo o de validación experimental. En ese contexto, se vislumbran como contribuciones del presente trabajo a aquellas que tienen que ver con la integración de tecnologías con gran potencial de aplicación a la medición, registro y control de variables físicas de interés, fundamentalmente, en este campo de aplicación, además de focalizar en el registro de datos para realizar un seguimiento del cultivo y trabajar futuramente en el concepto de trazabilidad de las mediciones.

## III. Descripción del Sistema

### A. Aspectos Generales

El cultivo se desarrolla en un contenedor adaptado, de 32 $m^2$ con sistema de intracción y extracción de aire, acondicionamiento de temperatura con sistema Split de 3000 frigorías y sistemas de iluminación para cultivo, de distintas tecnologías (lámparas de sodio y paneles LED). El sistema desarrollado constituye una estación de adquisición de datos y control de actuadores, diseñada específicamente para salas de cultivo indoor de *Cannabis sativa L.*, donde la regulación precisa del microclima resulta indispensable para acompañar las demandas fisiológicas de la planta.

### B. Aspectos Funcionales

De acuerdo a lo analizado en la Sección II, se monitorean aquellas variables cuyo impacto sobre el crecimiento y la calidad de las inflorescencias ha sido ampliamente documentado: la temperatura ambiente, la humedad relativa, la concentración de $CO_2$, así como el estado de los actuadores encargados de modificar activamente el ambiente, sistemas de extracción, iluminación, ventilación, calefacción, riego y suministro de $CO_2$. Puede hacerse un monitoreo, seguimiento y acondicionamiento, teniendo en cuenta que durante el enraizamiento de esquejes se prioriza un ambiente templado y muy húmedo para favorecer la emisión de raíces adventicias. En la fase vegetativa se busca un equilibrio que estimule el desarrollo estructural sin inducir estrés y en la floración se requiere un ambiente más seco y ligeramente más cálido, que movilice los fotoasimilados hacia los cogollos y desaliente la proliferación de patógenos fúngicos. El $CO_2$, por su parte, se adiciona únicamente durante las horas de luz y en niveles que potencian la fotosíntesis, siempre que la intensidad lumínica sea la adecuada. En la Fig. 1 se muestra el diagrama en bloques del sistema, en su fase inicial. Actualmente se emplean dos sensores digitales de temperatura y humedad ambiente, modelo DHT-22, aptos para su uso en domótica y un sensor de gas MQ-135 (ver Tabla I), de uso general para medir la concentración de $CO_2$. La resistencia interna de dicho sensor varía en presencia de diferentes gases, por lo que presenta una baja selectividad y requiere un proceso de calibración. Se debe determinar la resistencia de referencia del sensor en un ambiente conocido (~400 ppm de $CO_2$) y relacionarla posteriormente con la resistencia instantánea medida. A partir de esta relación y de la curva característica del sensor, es posible estimar una concentración equivalente en ppm, la cual constituye una aproximación de la concentración real. Los sensores mencionados se ubican en puntos estratégicos del recinto. La humedad de suelo se mide en una sola maceta, con un higrómetro anticorrosión (escalable a 8 macetas).

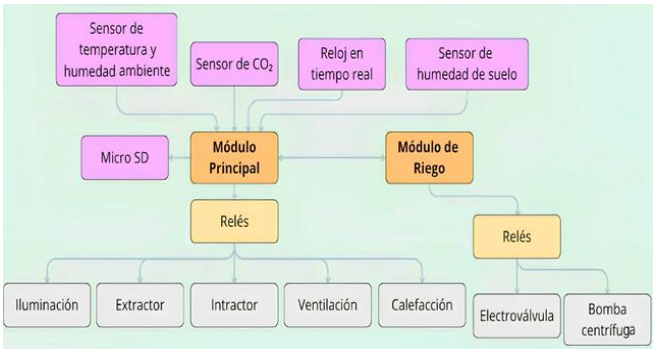



Fig. 1. Diagrama en bloques del sistema implementado.

TABLA I. DETALLES DEL SISTEMA IMPLEMENTADO

| Parte | Modelo y/o característica |
|---|---|
| Sensor de temperatura y humedad ambiente | DHT-22. Exactitud: ±0.5°C y ±2 a 5 % RH |
| Sensor de $CO_2$ | MQ-135. Exactitud dependiente de la calibración. Error relativo del orden del ±5 al 10 % |
| Sensor de humedad de suelo | Módulo HD-38 con salida digital y sonda anticorrosión |
| Sensado de intensidad lumínica | Luxómetro. Conversión a PPFD según longitud de onda del equipo de iluminación utilizado |
| Placa electrónica principal | Arduino MEGA 2560 |
| Reloj en tiempo real | Módulo RTC DS3231 |
| Pantalla | Display TFT ILI9486 de 3.5” |

En la Fig. 2 se muestra el módulo principal y el módulo de riego desarrollados. Una interfaz de potencia basada en relés controla la conexión y desconexión de los sistemas de climatización, iluminación y riego. Dado que se conocen las potencias individuales de estos equipos, es posible realizar un seguimiento del consumo energético. La interfaz hombre-máquina está compuesta por una pantalla gráfica y un teclado matricial, que permiten al operario visualizar en tiempo real las variables medidas, acceder a gráficos históricos y, si lo desea, intervenir manualmente sobre los actuadores. Por su parte, un módulo de reloj de tiempo real con respaldo de batería garantiza la correcta trazabilidad temporal de las mediciones, mientras que un sistema de almacenamiento en tarjeta SD registra los datos en un archivo estructurado. En la Tabla I se presentan los detalles principales del sistema.

Durante su operación normal, el sistema ejecuta un ciclo continuo de adquisición y respuesta. Dado que las variables monitoreadas presentan una dinámica relativamente lenta, se adoptó una frecuencia de lectura de sensores cada dos minutos, considerada suficiente para capturar tanto las tendencias graduales como eventuales fluctuaciones. La lectura es seguida de la validación de los datos y la actualización de un historial circular que conserva las últimas decenas de muestras. Se entiende por fluctuaciones aquellas variaciones que superan los umbrales críticos definidos para la estabilidad del cultivo, tales como incrementos o descensos térmicos abruptos o cambios significativos en la humedad relativa, de acuerdo al análisis realizado en la Sección II. Cuando se detecta una desviación fuera de los rangos considerados adecuados para el desarrollo del cultivo, se genera un registro inmediato en la tarjeta SD, complementando el almacenamiento periódico. Cada registro incluye la marca temporal precisa, los valores de temperatura y humedad correspondientes a ambos puntos de medición, la concentración de $CO_2$, el estado de los relés y el tiempo acumulado de funcionamiento de cada actuador.




Fig. 2. Sistema desarrollado. Módulo principal y módulo de riego.

La interacción con el usuario se realiza mediante un teclado matricial que permite la navegación entre distintas pantallas. Una pantalla principal presenta en forma destacada los valores actuales de las variables; una pantalla de control habilita la conmutación manual de los relés; y diversas pantallas gráficas muestran la evolución reciente de las variables, utilizando escalas fijas definidas dentro de rangos fisiológicamente relevantes para la especie cultivada.

## IV. ENSAYOS Y RESULTADOS

El sistema desarrollado se encuentra en etapa de validación de datos en laboratorio, junto con los primeros ensayos realizados en el entorno de cultivo. Aún no está completamente integrado, dado que se han verificado sus aspectos funcionales, pero el análisis de su impacto agronómico queda pendiente para una fase posterior. Se prevé que la plataforma sea capaz de adquirir y registrar de manera continua las variables ambientales definidas, durante períodos prolongados de tiempo, manteniendo la integridad de los datos. Actualmente el sistema detecta y almacena adecuadamente los cambios en las lecturas y la tarjeta SD conserva los datos sin corrupción. Los actuadores responden a las órdenes del usuario sin errores. En la Fig. 3 se muestra un ejemplo de visualización de datos en pantalla, obtenidos durante un ensayo. En la parte superior se presenta la pantalla principal, con los valores de las variables medidas. En la parte inferior se muestra la opción de visualización de gráfica. La Fig. 4 ilustra la variación de temperatura ambiente, registrada por dos sensores de temperatura, durante 9 h, con adquisición de datos cada 2 minutos. Este gráfico fue elaborado a partir de los datos almacenados en la tarjeta SD. De manera análoga, en la Fig. 5 se muestra la variación de humedad ambiente.




Fig. 3. Visualización de datos en pantalla. Arriba: pantalla principal. Abajo: pantalla de gráfico de variables.

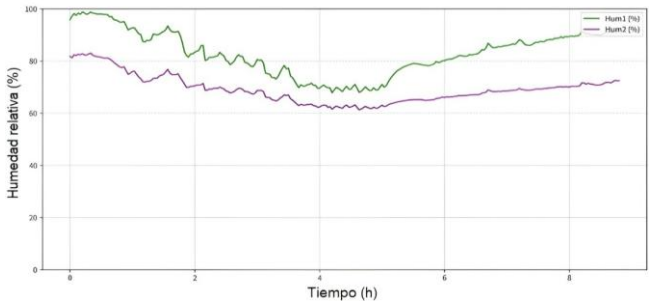



Fig. 4. Registro de temperaturas ambiente en el recinto durante 9 h.

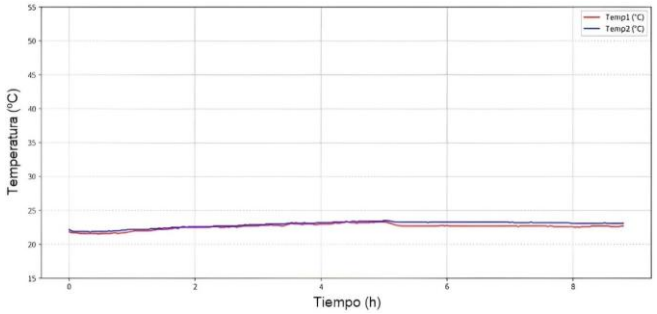



Fig. 5. Registro de humedad ambiente en el recinto durante 9 h.

## V. Conclusiones

Se desarrolló un sistema para la medición, registro y control de variables asociadas al cultivo interior de cannabis. Dicho sistema se basa en una plataforma Arduino MEGA 2560 y utiliza sensores compatibles de bajo costo relativo. El sistema integra dos módulos: uno principal y otro de riego automatizado, concebidos para satisfacer las necesidades del cultivo en condiciones controladas. Los ensayos preliminares evidencian resultados promisorios, al permitir una observación sistemática del microclima y ofrecer una herramienta de apoyo a la toma de decisiones. Se integran funciones de monitoreo, control, en concordancia con los requerimientos técnicos de los cultivos en ambientes cerrados. Las lecturas son coherentes con las condiciones ambientales de la sala de laboratorio y el recinto de cultivo, permitiendo posteriormente ajustar las calibraciones necesarias para la medición de concentración de $CO_2$.

## Agradecimientos

El equipo de trabajo agradece a la Comisión de Investigaciones científicas de la Pcia. de Buenos Aires (CIC).

## Referencias

[1] P. A. Y. Ampim, E. Obeng & E. Olvera-González, “Indoor vegetable production: an alternative approach to increasing cultivation,” *Plants*, Vol. 11, no. 21, pp. 1-28, Oct. 2022.

[2] D. Dik, E. Polyakova, A. Chelovechkova, V. Moskvin and T. Nikiforova, "The system of environment control of botanic garden greenhouses," *2018 International Multi-Conference on Industrial Engineering and Modern Technologies*, Vladivostok, Russia, Oct. 2018, pp. 1-7. https://doi.org/10.1109/FarEastCon.2018.8602559

[3] M.A . García, S. Gutiérrez, H. C. López, S. Rivera, A. C. Ruiz, “Estado del arte de la tecnología de robots aplicada a invernaderos,” *Avances en Investigación Agropecuaria*, vol 11, no. 3, 2007, pp. 53-61.

[4] P. Ponce, A. Molina, P. Cepeda, E. Lugo, and B. MacCleery, *Greenhouse design and control*, London, U.K.: CRC Press, 2015.

[5] Á. H. Alarcón-López, G. Arias-Vargas, C. J. Díaz-Ortíz, and J. D. Sotto-Vergara, “Automatic control system for climate variables to optimize greenhouse crop yields”, *Ing. Solidaria*, vol. 14, no. 24, Jan. 2018, pp. 1–11. https://doi.org/10.16925/in.v14i24.2158

[6] K. Yasuba, H. Kurosaki, M.Takaichi and K. Suzuki, “Development of an automatic data collection and communication system for greenhouse in Japan,” *SICE Annual Conference*, Tokyo, Japan, 2011, pp. 1-2.

[7] D. Jin, S. X. Jin, and J. Chen, “Cannabis indoor growing conditions, management practices, and post-harvest treatment: a review,” *American Journal of Plant Sciences*, Vol. 10, no. 6, pp. 925-946, Jun. 2019. https://doi.org/10.4236/ajps.2019.106067

[8] A. López y S. Gómez Roca, “La cadena de valor del cannabis: situación y tendencias internacionales y oportunidades para la Argentina,” Consejo para el Cambio Estructural – Ministerio de Desarrollo Productivo de la Nación, Buenos Aires, Argentina, Serie de Documentos para el Cambio Estructural, Mar. 2021. ISSN 2718-756X.

[9] A. M. VanDerZanden, “Environmental factors affecting plant growth,” Oregon State University, OSU Extension Service. Accessed: Apr. 9, 2025. [Online]. Available: https://extension.oregonstate.edu/gardening/techniques/environmental-factors-affecting-plant-growth

[10] A. Saragoça et al.,“Current context of cannabis sativa cultivation and parameters influencing its development,” *Agriculture*, Vol. 15, no. 15, Jul. 2025, pp. 1-26. https://doi.org/10.3390/agriculture15151635

[11] D. Potter, “The propagation, characterisation and optimisation of cannabis sativa L as a phytopharmaceutical,” Ph.D. Thesis, Dept. of Pharmaceutical Science Research, King’s College London, UK, 2009.

[12] S. Chandra, H. Lata, I. A. Khan and M. A. Elsohly, “Photosynthetic response of Cannabis sativa L. to variations in photosynthetic photon flux densities, temperature and $CO_2$ conditions,” *Physiol Mol Biol Plants*, Vol. 14, pp. 299–306, Oct. 2008. https://doi.org/10.1007/s12298-008-0027-x

[13] A. Mazzoni *et al.*, “Cannabis medicinal: consideraciones para el cultivo,” *Presencia*, vol. 34, no. 79, pp. 29–32, Jul. 2023.

[14] Y. Zheng, Handbook of cannabis production in controlled environments, Boca Raton, FL, USA: CRC Press, 2022.

[15] D. Llewellyn, S. Golem, E. Foley, S. Dinka, A.M.P. Jones and Y. Zheng, “Indoor grown cannabis yield increased proportionally with light intensity, but ultraviolet radiation did not affect yield or cannabinoid content,” Front. Plant Sci., Vol. 13, pp. 1-12, Sept. 2022. https://doi.org/10.3389/fpls.2022.974018

[16] Z. K. Punj, D. Collyer, C. Scott, S. Lung, J. Holmes and D. Sutton, “Pathogens and molds affecting production and quality of cannabis sativa L.,” *Front. Plant Sci*., Vol. 10, article 1120, pp. 1-23, Oct. 2019. https://doi.org/10.3389/fpls.2019.01120

[17] J. M. McPartland, “A review of cannabis diseases,” *Journal of the International Hemp Association*, Vol. 3, no. 1, pp. 19–23, 1996.

[18] Apex Publishers. (2025). Greenhouse ventilation and cooling: humidity control [online]. Available: https://greenhouse-management.com/greenhouse_management/greenhouse_ventilation_cooling/greenhouse_humidity_control.htm

[19] V. Rodriguez-Morrison, D. Llewellyn and Y. Zheng, “Cannabis yield, potency, and leaf photosynthesis respond differently to increasing light levels in an indoor environment,” *Front. Plant Sci*. Vol. 12, article 646020, pp. 1-16, May 2021. https://doi.org/10.3389/fpls.2021.646020

[20] S. Chandra, H. Lata, I. A. Khan and M. A. Elsohly, “Photosynthetic response of Cannabis sativa L., an important medicinal plant, to elevated levels of $CO_2$,” *Physiol Mol Biol Plants*, Vol. 17, no. 3, pp. 291–295, 2011.https://doi.org/10.1007/s12298-011-0066-6

[21] M. F. Westmoreland, “Environmental physiology of medical cannabis" Ph.D. Dissertation, Utah State University, Logan, Utah, USA, 2024. https://digitalcommons.usu.edu/etd2023/243

[22] D. Savvas, G. Gianquinto, Y. Tuzel and N. Gruda, “Soilless culture,” in Good Agricultural Practices for Greenhouse Vegetable Crops: Principles for Mediterranean Climate Areas, Food and Agriculture Organization of the United Nations (FAO), Rome, Italy, 2013.

[23] M. Vernon, A. Z. Kouzani, L. D. Webb and S. D. Adams, “A Survey of modern greenhouse technologies and pactices for commercial cannabis cultivation,” *IEEE Access*, Vol. 11, pp. 62077–62090, Jun. 2023. https://doi.org/10.1109/ACCESS.2023.3285242

[24] X. Zheng, S. Shao, L. Li and Y. Wang, "Research and design of monitoring system for greenhouse based on Internet of Things,"*4th Int. Symposium on Smart and Healthy Cities*, Shanghai, China, 2022, pp. 59-63. https://doi.org/10.1109/ISHC56805.2022.00021

[25] C. C. Ordoñez-Quintero, H. A. Ordoñez-Erazo y A. Ordoñez, “Low-cost prototype for automating of greenhouse medicinal cannabis production supported by IoT,” *Revista Científica*, Vol. 45, no. 3, pp. 414-421, 2022. https://doi.org/10.14483/23448350.19678

[26] C. Menoni, D. Alonso, F. Saravia, N. Pérez, y L. Steinfeld, “Sistema de monitoreo y control de cultivo indoor de cannabis,” *Ingeniería*, n.º 24, pp. 156-171, 2023. https://doi.org/10.36561/ING.24.12

# Arquitectura Embebida Modular para Gestión de Energía Renovable: Diseño, Implementación y Evolución de un Sistema Autónomo Multipropósito

**Pablo Petrashin**
*Depto. de Electrónica, FCEFyN*
*Universidad Nacional de Córdoba*
Córdoba, Argentina
ppetrashin@gmail.com

**César Reale**
*Depto. de Electrónica, FCEFyN*
*Universidad Nacional de Córdoba*
Córdoba, Argentina
creale@unc.edu.ar

**Fernando Gallardo**
*Depto. de Electrónica, FCEFyN*
*Universidad Nacional de Córdoba*
Córdoba, Argentina
fergallardo@gmail.com

***Abstract*—Se presenta el diseño, implementación y evolución de una arquitectura de sistemas embebidos modular orientada a la gestión de energía renovable, desarrollada en la FCEFyN de la Universidad Nacional de Córdoba en el marco del proyecto CONSOLIDAR 2023-2027. El sistema integra cuatro módulos desarrollados como trabajos finales de grado: (1) un controlador domótico con comunicación PLC sobre red eléctrica domiciliaria, basado en microcontrolador STM32 con medición de energía en tiempo real; (2) un cargador solar con seguimiento del Punto de Máxima Potencia (MPPT) mediante el algoritmo Perturb & Observe implementado sobre convertidor DC/DC tipo Buck, con eficiencia medida del 91.3%; (3) un módulo de acondicionamiento analógico basado en síntesis de redes activas Gm-C (filtro Butterworth de cuarto orden, sintonizable entre 1 y 18kHz) que redujo el ruido de medición de corriente de 4.2% a 0.8%; y (4) un sistema IoT de monitoreo ambiental basado en ESP32 con comunicación ESP-NOW, procesamiento de imágenes embebido en espacio YCbCr y tasa de detección del 87%. Se describen la arquitectura hardware/software de cada módulo, sus interfaces, resultados experimentales obtenidos y la proyección hacia comunicaciones LoRa/LoRaWAN e inteligencia artificial en el borde (edge AI) para el período 2025–2027.**

***Keywords—Embedded Systems, MPPT, PLC, Active Networks, GMC Filters, IoT, ESP32, LoRaWAN, Edge AI, Renewable Energy.***

## I. Introducción

La creciente demanda de sistemas de gestión de energía renovable en entornos urbanos y rurales plantea desafíos de ingeniería que abarcan múltiples dominios: electrónica de potencia para la conversión eficiente de energía solar, electrónica analógica de precisión para el acondicionamiento de señales de sensores, sistemas embebidos de tiempo real para el control y la comunicación, e inteligencia artificial para la optimización predictiva del despacho energético. La integración de estas disciplinas en un sistema cohesivo y escalable constituye un área de investigación activa con alta relevancia tanto académica como industrial [1, 5].

En el contexto de los sistemas fotovoltaicos autónomos, el seguimiento del punto de máxima potencia (MPPT) es una técnica fundamental para maximizar la energía extraída de los paneles solares bajo condiciones variables de irradiancia y temperatura. Los algoritmos de perturbación y observación (P&O) son ampliamente utilizados por su simplicidad de implementación y su buena performance en condiciones de irradiancia lentamente variable [5, 6]. Sin embargo, su integración con etapas de acondicionamiento analógico de alta calidad en el lazo de medición no ha sido suficientemente documentada en la literatura de sistemas embebidos de bajo costo.

En el dominio de la domótica, la comunicación por línea eléctrica (PLC) ofrece una alternativa de bajo costo a las tecnologías inalámbricas para el control de cargas en edificios existentes, eliminando la necesidad de cableado adicional. La integración de PLC con medición de energía en tiempo real por nodo permite implementar sistemas de gestión energética distribuida con granularidad de carga individual [1].

El monitoreo ambiental mediante sistemas IoT de bajo costo basados en ESP32 representa una tendencia consolidada en la literatura reciente. La incorporación de algoritmos de visión por computadora embebidos directamente en el microcontrolador —sin depender de infraestructura de cómputo en la nube— abre posibilidades para aplicaciones en entornos remotos con conectividad limitada [9].

Este artículo presenta la arquitectura de un sistema embebido modular para gestión de energía renovable desarrollado en el Departamento de Electrónica de la FCEFyN-UNC, cuya evolución abarca el período 2015–2025 en el marco de sucesivos proyectos CONSOLIDAR de la SeCyT UNC [3]. El enfoque modular adoptado permitió que cada subsistema fuera desarrollado, validado y publicado de forma independiente, constituyendo a la vez una pieza funcional del sistema global. La Sección II describe la arquitectura del sistema. Las Secciones III a VI detallan cada módulo. La Sección VII presenta una tabla comparativa. Las Secciones VIII y IX describen la proyección futura y las conclusiones.

## II. Arquitectura del Sistema

La arquitectura del sistema se organiza en cuatro capas funcionales interconectadas, tal como se ilustra en el diagrama de bloques de la Fig. 1. La capa de generación y almacenamiento comprende el panel fotovoltaico, el módulo MPPT y el banco de baterías AGM. La capa de distribución y control incluye el módulo domótico PLC que distribuye energía a las cargas del hogar. La capa de acondicionamiento de señal provee señales limpias de tensión, corriente e irradiancia al sistema de control mediante filtros activos Gm-C. La capa de monitoreo ambiental incorpora el sistema IoT con visión embebida.

Las interfaces eléctricas entre capas están normalizadas: el bus DC opera en el rango 12–15V, las señales analógicas acondicionadas se presentan en el rango 0–3.3V compatible

con los ADC de los microcontroladores utilizados, y la comunicación digital entre módulos utiliza los buses I2C y SPI para sensores locales y PLC para la red domótica. Esta normalización de interfaces permite reemplazar o actualizar cualquier módulo sin rediseñar el resto del sistema, característica fundamental para la escalabilidad y el mantenimiento a largo plazo.

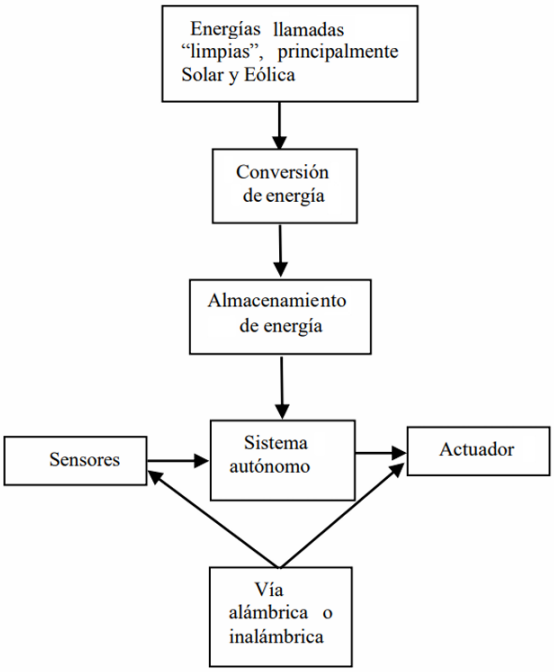



Fig 1: Diagrama de bloques del sistema propuesto

El sistema fue concebido desde el inicio con esta arquitectura modular, lo que permitió que cada módulo fuera desarrollado como un Proyecto Integrador de grado independiente, con objetivos técnicos propios y publicación de resultados parciales. Esta estrategia generó una continuidad de investigación sostenida durante más de una década, con resultados validados por pares en congresos nacionales e internacionales [1, 3, 4, 9].

## III. Módulo 1: Control Domótico con PLC

### *A. Arquitectura Hardware*

El módulo domótico fue desarrollado por Gallardo y Pistoia en el marco del proyecto de investigación previo de la FCEFyN [1]. La unidad central utiliza una Raspberry Pi 3 Model B con interfaces PLC, USB 2.0 y Ethernet 10/100. Cada nodo de la red implementa cuatro circuitos integrados principales: el CY8CPLC10 de Cypress Semiconductor para la comunicación por línea eléctrica, el microcontrolador STM32F107 de STMicroelectronics como unidad de procesamiento local, el CI CS5484 de Cirrus Logic para medición de energía eléctrica mediante conversión sigma-delta de 24 bits con interfaz SPI, y el chip MAY4409 para el control de iluminación mediante señal PWM de 10 bits con resolución de 1024 niveles de intensidad.

La topología de red PLC utiliza la línea eléctrica domiciliaria de 220V/50Hz como medio físico compartido, operando en la banda de frecuencias de 9 a 95kHz según el estándar CENELEC EN 50065. Cada nodo tiene una dirección única de 8 bits asignada durante la configuración inicial vía puerto USB 2.0, lo que permite conformar una red de hasta 256 nodos sin infraestructura adicional de cableado. La velocidad de transmisión efectiva alcanzada fue de 2.4kbps, suficiente para los mensajes de control y telemetría del sistema.

### *B. Firmware y Modos de Operación*

El firmware del STM32F107 fue desarrollado en C embebido con un scheduler cooperativo de ciclo fijo de 10ms. Cada nodo implementa cuatro modos de operación de salida configurables: interruptor simple (on/off), interruptor programado por horario con RTC interno, control de dimmer por PWM para luminarias incandescentes y halógenas, y modo automático por sensor de luminosidad con histéresis configurable. Las entradas disponibles por nodo son: luminosidad ambiente (sensor LDR con ADC de 12 bits), medición de tensión y corriente en la carga conectada (vía CS5484, resolución 0.1W) y detección de presencia por sensor PIR (entrada digital con debounce por software de 200ms).

La unidad central Raspberry Pi ejecuta un servidor Python con API REST que gestiona la red PLC, registra series temporales de consumo por nodo en base de datos SQLite y expone una interfaz web responsive para configuración remota. Las reglas de automatización son configurables por el usuario mediante una interfaz de programación visual basada en condiciones temporales y de estado de sensores.

### *C. Resultados Experimentales*

El sistema fue validado en una maqueta a escala de vivienda con 8 nodos PLC activos. La latencia de respuesta extremo a extremo para comandos de conmutación fue de 180ms promedio (σ=23ms), dentro del límite perceptible por usuarios humanos. El error de medición de potencia activa del CS5484 fue inferior al 0.5% referido al fondo de escala (calibrado contra medidor de referencia clase 0.2 certificado por INTI). El consumo propio de cada nodo en modo de espera fue de 1.8W, y el sistema completo de 8 nodos consumió 14.4W adicionales a la carga controlada.

## IV. Módulo 2: Cargador Solar con MPPT

### *A. Topología del Convertidor Buck*

El cargador solar fue desarrollado por Olguín en el marco del proyecto y presentado en IEEE CASE 2020 [4]. El convertidor DC/DC tipo Buck fue dimensionado para un panel fotovoltaico de 50W con los siguientes parámetros de placa: tensión de circuito abierto VOC=21.5V, corriente de cortocircuito ISC=3.1A, tensión en el punto de máxima potencia VMPP=17.34V y corriente en el punto de máxima potencia IMPP=2.88A. El rango de operación del convertidor es: tensión de entrada 14.5–21.5V, tensión de salida regulada 13–15V, corriente máxima de salida 2.1A.

El inductor del Buck (L=680µH) fue dimensionado para garantizar operación en modo de conducción continua (CCM) con una frecuencia de conmutación de 20kHz y un ripple de corriente de inductor inferior al 10% de la corriente nominal. El elemento de conmutación es un MOSFET de canal N IRFZ44N con driver de gate optoacoplado para aislación galvánica entre el dominio de potencia y el microcontrolador de control. La capacidad de salida (C=1000µF/25V electrolítico de baja ESR) fue seleccionada para limitar el ripple de tensión de salida a menos de 50mV pico a pico.

### *B. Algoritmo MPPT Perturb & Observe*

El algoritmo P&O implementado evalúa la potencia instantánea P(k)=V(k)·I(k) en el punto de operación actual y la compara con la muestra anterior P(k-1). La lógica de decisión es: si ΔP=P(k)-P(k-1)>0 y ΔV>0, o ΔP<0 y ΔV<0, se incrementa el duty cycle en Δd=0.5%; en caso contrario se decrementa. El ciclo de control opera con un período de muestreo de 100ms, determinado como compromiso entre la velocidad de convergencia al MPP y la inmunidad al ruido de la señal de corriente del panel.

La tensión y corriente del panel se miden mediante un divisor resistivo (tensión) y una resistencia de shunt de 0.1Ω con amplificador de instrumentación INA219 de Texas Instruments (resolución 1mA, exactitud ±0.5%). La tensión y corriente de batería se miden con el mismo esquema. El microcontrolador de 8 bits (ATmega328P a 16MHz) ejecuta el algoritmo P&O y el control del perfil de carga de la batería, con una pantalla LCD 16×2 para visualización local de los parámetros del sistema en tiempo real.

### *C. Perfil de Carga y Resultados*

La batería objetivo es una VRLA-AGM de 12V/20Ah (C10). El perfil de carga de tres etapas implementado es: (1) Bulk: corriente constante de 2.1A (C/10) hasta alcanzar 14.4V en bornes de batería, con corrección por temperatura a razón de -4mV/°C/celda; (2) Absorción: tensión constante 14.4V con reducción natural de corriente hasta caer por debajo de 200mA (C/100); (3) Flotación: tensión 13.7V para mantenimiento sin gasificación. Un relé NC (normalmente cerrado) desconecta la batería si la tensión supera 15V o la temperatura excede 45°C. Un segundo relé NA (normalmente abierto) controla la conexión de cargas de consumo con corriente máxima de 1A.

La eficiencia de transferencia de potencia medida en condiciones de laboratorio (irradiancia simulada constante a 1000W/m²) fue del 91.3% promedio en el rango de tensión de entrada 15–21V. El algoritmo P&O convergió al MPP en menos de 5 ciclos de control (menos de 500ms) ante escalones de irradiancia de ±200W/m², con una oscilación residual en estado estacionario de ±1.2W alrededor del MPP. Estos resultados superan significativamente los obtenidos con topologías de carga directa (eficiencia 72–78% documentada en [5]).

## V. Módulo 3: Redes Activas Gm-C para Acondicionamiento de Señal

Para mitigar el ruido armónico (20kHz) generado por el convertidor Buck que afectaba la estabilidad del algoritmo MPPT, se implementó un filtro paso-bajo Butterworth de cuarto orden basado en síntesis de redes Gm-C. Esta topología, seleccionada por su capacidad de sintonización electrónica, utiliza OTAs LM13700 donde la frecuencia de corte fc es linealmente ajustable mediante la corriente de polarización. Experimentalmente, el filtro alcanzó una fc=10.2kHz (2% de error), una atenuación de -80dB/década y un rango dinámico de 68dB. La integración redujo el ruido de medición de corriente del 4.2% al 0.8% del fondo de escala, estabilizando las oscilaciones del duty cycle de ±3.1% a ±0.6% ante condiciones de irradiancia estable.

## VI. Módulo 4: Sistema IoT de Monitoreo Ambiental

### *A. Motivación y Contexto*

El Embalse San Roque, principal fuente de agua potable para la ciudad de Córdoba (1.4 millones de habitantes), sufre periódicamente episodios de proliferación de cianobacterias (blooms) causados por la eutrofización del cuerpo de agua. Las cianobacterias producen toxinas (microcistinas) que representan un riesgo sanitario grave y obligan a la interrupción del suministro de agua potable. Los métodos de control disponibles incluyen la dosificación de peróxido de hidrógeno ($H_2O_2$), que oxida los pigmentos de las cianobacterias sin dejar residuos tóxicos en el agua. La dosificación manual requiere monitoreo frecuente por personal especializado, lo que motivó el desarrollo de un sistema autónomo de bajo costo [9].

### *B. Arquitectura del Sistema de Dos Nodos*

El sistema fue desarrollado por Menegozzo y Strappazzon bajo la dirección de Gallardo y Petrashin, y presentado en CASE 2025 [9]. La arquitectura consiste en dos nodos ESP32 comunicados por el protocolo ESP-NOW de Espressif, que opera sobre la capa MAC del estándar IEEE 802.11 sin necesidad de punto de acceso WiFi intermedio. Las características del enlace son: latencia de comunicación inferior a 10ms, alcance de hasta 200m en espacio abierto sin obstáculos, y velocidad de transmisión de 1Mbps en la capa MAC.

El nodo de captura utiliza un módulo ESP32-CAM con sensor de imagen OV2640 (resolución máxima 2MP, interfaz SCCB). El nodo actuador utiliza un DOIT ESP32 DevKit v1 (ESP32-WROOM-32, 240MHz, 4MB Flash, 320KB RAM) para el control del sistema mecánico de dosificación. La comunicación bidireccional permite que el nodo actuador envíe confirmaciones de ejecución al nodo de captura, cerrando el lazo de control del sistema.

### *C. Algoritmo de Detección de Cianobacterias*

El algoritmo de detección opera en el espacio de color YCbCr, seleccionado sobre RGB por su robustez ante variaciones de iluminación natural: la componente de luminancia Y absorbe los cambios de intensidad lumínica mientras que las componentes de crominancia Cb y Cr codifican el color independientemente del brillo. Las imágenes capturadas por el OV2640 en resolución QVGA (320×240 píxeles) son convertidas internamente al espacio YCbCr mediante las ecuaciones estándar ITU-R BT.601.

Se cuenta la proporción de píxeles que satisfacen simultáneamente los umbrales empíricos de crominancia: $Cb \in [95,130]$ y $Cr \in [130,165]$, rangos calibrados con 240 muestras del embalse bajo distintas condiciones de iluminación natural. El nivel de alerta se clasifica en cuatro categorías según la proporción de píxeles positivos: normal (<5%), precaución (5–15%), alerta (15–30%) y crítico (>30%). El tiempo de ciclo completo (captura + procesamiento + transmisión ESP-NOW) fue medido en 1.8s promedio, adecuado para la dinámica del fenómeno.

### *D. Control de Dosificación y Conectividad*

El nodo actuador recibe el nivel de alerta y calcula el volumen de $H_2O_2$ requerido según la concentración objetivo (3mg/L sobre el volumen estimado del área afectada). El nivel de líquido en el depósito de $H_2O_2$ se mide con un sensor ultrasónico HC-SR04; mediante el principio de Torricelli y el diámetro calibrado de la válvula solenoide, el firmware

calcula el tiempo de apertura necesario para la dosis requerida. El control de la válvula se realiza mediante un relé de estado sólido activado por GPIO del ESP32, con resolución de tiempo de 10ms.

Para la conectividad externa, el nodo actuador opera en modo WiFi Station conectándose a la red disponible en el sitio y expone: (a) un servidor HTTP embebido para configuración de parámetros (umbral de detección, diámetro de válvula, números de contacto), y (b) notificaciones push vía la API de WhatsApp Business Cloud para alertas en tiempo real al personal técnico. La tasa de detección correcta fue del 87% sobre un conjunto de prueba de 120 muestras etiquetadas manualmente, con un 6% de falsos positivos y un 7% de falsos negativos.

## VII. Resumen Técnico de los Módulos

La Tabla I sintetiza los parámetros técnicos principales de los cuatro módulos implementados y los dos módulos proyectados para el período 2025–2027.

Tabla I: Parámetros técnicos de los módulos implementados y proyectados.

| Módulo | Hardware principal | Función / resultado clave | Interfaz |
|---|---|---|---|
| Domótico PLC | STM32F107 + RPi + CY8CPLC10 | Control cargas, dimmer, medición energía ±0.5% | PLC, USB, Ethernet |
| MPPT Solar | ATmega328P + IRFZ44N | Carga VRLA-AGM, Buck 20kHz, η=91.3% | LCD, I2C/SPI |
| Redes Activas | OTA LM13700 × 4 | Filtro Gm-C 4°orden, fc=1–18kHz, DR=68dB | Analógica 0–3.3V |
| IoT Ambiental | ESP32-CAM + ESP32 | Detección cianobacterias, $H_2O_2$, 87% exactitud | ESP-NOW, WiFi |
| LoRa (proy.) | ESP32 + SX1276 | Telemetría 10–15km, TTN/MQTT, margen >20dB | LoRaWAN 915MHz |
| Edge AI (proy.) | STM32H7 (480MHz) | LSTM irradiancia, MPC energético, lat.<100ms | WiFi / LoRa |

## VIII. Conclusiones

Se presentó una arquitectura modular de sistemas embebidos para gestión de energía renovable, íntegramente desarrollada en la FCEFyN-UNC como parte del proyecto CONSOLIDAR 2023-2027, que integra cuatro módulos validados experimentalmente: control domótico PLC con medición de energía por nodo (error <0.5%), carga solar con MPPT P&O sobre convertidor Buck (eficiencia 91.3%), acondicionamiento analógico con filtros Gm-C de cuarto orden (reducción de ruido de 4.2% a 0.8%) y monitoreo ambiental IoT con detección de cianobacterias por visión embebida (exactitud 87%).

El enfoque modular adoptado demostró ser efectivo en dos dimensiones simultáneas: técnicamente, cada módulo produjo resultados publicables de manera independiente mientras contribuía al sistema global; organizativamente, la arquitectura de bloques con interfaces normalizadas permitió que cada módulo fuera desarrollado como un Proyecto Integrador de grado en diferentes períodos, manteniendo la coherencia del sistema sin requerir rediseño de los módulos existentes.

La proyección hacia comunicaciones LoRa/LoRaWAN (margen de enlace calculado 25.3dB a 10km) y edge AI sobre STM32H7 (inferencia LSTM en menos de 100ms) extiende el sistema hacia aplicaciones de gestión energética distribuida a escala geográfica y optimización predictiva, manteniendo el mismo paradigma modular y escalable que guió el desarrollo desde sus inicios.

Como trabajo futuro inmediato se propone: (a) validación del enlace LoRa en campo con medición real de RSSI y SNR en el entorno de instalaciones fotovoltaicas rurales de la provincia de Córdoba; (b) entrenamiento y validación del modelo LSTM con los datos históricos de la estación meteorológica FCEFyN; y (c) integración end-to-end de los cuatro módulos existentes en una instalación fotovoltaica real de pequeña escala (500W) para validación sistémica en condiciones de operación reales.

## Referencias

[1] F. Gallardo, G. Pistoia, C. Reale, P. Petrashin, "Modulo Domotico Compacto con Comunicacion y Programacion por PLC", Revista de la FCEFyN, vol. 3, nro. 1, pp. 43-48, 2016.

[2] A. F. Gomez, C. Reale, "Sintesis de Redes Activas Gm-C para Acondicionamiento de Señal en Sistemas de Gestion Energetica", Pasantia de Investigacion, FCEFyN-UNC, 2025.

[3] P. Petrashin, C. Reale, F. Gallardo, "Integracion a largo plazo de la energia sostenible en la ensenanza de la Ingenieria: un caso testigo", 5to CADI / 11vo CAEI, 2021.

[4] E. Olguin, F. Gallardo, P. Petrashin, C. Reale, "Solar Battery Charger for Embedded Systems Applications", IEEE CASE 2020, agosto 2020.

[5] K.-Y. Lo, Y.-M. Chen, Y.-R. Chang, "MPPT battery charger for stand-alone wind power system", IEEE Trans. Power Electronics, vol. 26, nro. 6, pp. 1631-1638, 2011.

[6] Semtech, "SX1276/77/78/79 Low Power Long Range Transceiver", Datasheet rev. 7, 2020.

[7] STMicroelectronics, "STM32H743xI Advanced Arm-based 32-bit MCU", Datasheet DS12110, 2023.

[8] Espressif Systems, "ESP32 Series Datasheet", v4.5, 2024.

[9] B. Menegozzo, F. Strappazzon, P. Petrashin, F. Gallardo, "IoT-Based Embedded System for Autonomous H2O2 Dosing to Control Cyanobacteria in Eutrophic Lakes", CASE 2025, 2025.

## Agradecimientos

Los autores agradecen el aval académico de la SeCyT UNC (Proyecto CONSOLIDAR 2023-2027, código 33620230100068CB) y el apoyo del Laboratorio de Electrónica Analógica de la FCEFyN-UNC. Los prototipos fueron desarrollados por los ingenieros Fernando Gallardo, Griselda Pistoia, Ezequiel Olguín, Alberto Fabián Gómez, Bruno Menegozzo y Franco Strappazzon.

Categoría

# Foro Tecnológico

Bioingeniería

Área Temática

# Compensación acústica para la validación de experimentos comportamentales en acuarios

Nicolás Batalla [‡], Juan Catalano [‡], Celeste Cebedio [†‡], Martín Lorusso[*#], Lucas Rabioglio[†‡], Maximiliano Antonelli[†‡§], Raúl Lopresti[†‡§], Luciana De Micco[†‡§], Paz Sal Moyano[*#§] y Ramiro Avalos[†‡]

*† Instituto de Investigaciones Científicas y Tecnológicas en Electrónica (ICYTE)*

*‡ Facultad de Ingeniería, Universidad Nacional de Mar del Plata (FI - UNMdP)*

*§ Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)*

*∗ Instituto de Investigaciones Marinas y Costeras (IIMyC)*

*# FCEyN, Universidad Nacional de Mar del Plata-CONICET*

{celestecebedio,lucas.rabioglio,maxanto,raul.lopresti,ldemicco,avalosrivas}@fi.mdp.edu.ar

martinilorusso@gmail.com; salmoyan@mdp.edu.ar

***Resumen*—En este trabajo se presenta un avance en el desarrollo de un sistema de ecualización adaptativa destinado a compensar los efectos de multitrayecto en entornos subacuáticos artificiales, como acuarios y estanques de laboratorio. La naturaleza reverberante de estos recintos degrada significativamente la fidelidad de los estímulos sonoros, afectando la validez de experimentos comportamentales.**

**La propuesta se basa en el modelado dinámico del canal acústico y en el diseño de filtros inversos para la restauración de la señal original. Se describen la problemática, la metodología de validación en PC y los desafíos asociados a una futura implementación en hardware de alto desempeño. Los resultados obtenidos constituyen una base sólida para mejorar la precisión de los estímulos acústicos en aplicaciones de bioacústica.**

***Palabras Clave*—acústica subacuática, multitrayecto, reverberación, filtros adaptativos, sistemas embebidos.**

## I. Introducción

La fascinación humana por la vida marina, ha impulsado la necesidad de estudiar sus comportamientos en el océano. Sin embargo, el entorno marino es vasto, opaco y hostil para la instrumentación. La bioacústica marina recurre frecuentemente a experimentos en acuarios o tanques de agua para estudiar el comportamiento de especies acuáticas bajo condiciones controladas, debido a las dificultades inherentes al entorno oceánico [1], [2]. Sin embargo, la propagación acústica en estos recintos semicerrados difiere significativamente de las condiciones de campo libre. Fenómenos de absorción, dispersión y, especialmente, reflexiones múltiples generan reverberación y efectos de multitrayecto (*multipath*) que degradan severamente la señal sonora [3], [4]. Además, la proximidad de las paredes altera la relación entre presión sonora y movimiento de partículas respecto del hábitat natural [**?**], [5], produciendo un campo acústico perturbado donde la señal percibida difiere de la señal original [6].

Diversos trabajos han estudiado esta problemática mediante modelos teóricos y experimentales. En [7] se analiza cómo las dimensiones del tanque afectan la fidelidad acústica, identificando la frecuencia de resonancia y la longitud de atenuación como parámetros críticos. Otros estudios proponen metodologías de caracterización espacial del campo sonoro [1] o emplean técnicas numéricas como FEM y FDTD para modelar estas distorsiones [8]–[11]. No obstante, la mayoría de estas investigaciones se enfocan en el diagnóstico del problema o en protocolos experimentales restrictivos, existiendo escasos trabajos orientados a su mitigación activa mediante procesamiento de señales.

En comunicaciones acústicas subacuáticas, los filtros adaptativos constituyen una técnica consolidada para compensar canales dispersivos [4]. Asimismo, en acústica de recintos terrestres, los filtros inversos han demostrado ser eficaces para ecualizar entornos reverberantes [12], [13]. Sin embargo, su aplicación como herramienta activa de control acústico en tanques de laboratorio permanece poco explorada.

Por ello, este trabajo propone el diseño e implementación de un sistema de ecualización adaptativa capaz de estimar la respuesta acústica del entorno y pre-distorsionar la señal de excitación para compensar activamente los efectos del canal. Mediante filtros digitales inversos, se busca reducir la variabilidad espacial y espectral del campo sonoro, garantizando que el estímulo acústico en la posición de interés reproduzca con alta fidelidad la señal deseada y cumpla los criterios de validación bioacústica establecidos en la literatura [9].

## II. Problemática

El estudio experimental del comportamiento de especies biológicas mediante estímulos acústicos requiere la reproducción de señales de manera controlada y precisa. Sin embargo, en entornos artificiales como tanques de prueba, la señal emitida sufre distorsiones significativas que degradan la integridad del experimento. Estas distorsiones se originan principalmente por:

- **Multitrayecto:** Generado por reflexiones constantes en las paredes, el fondo y la superficie del agua.
- **Reverberación:** Un incremento artificial del tiempo de persistencia de la señal que enmascara los detalles temporales del estímulo original.
- **Interferencia constructiva y destructiva:** La superposición de múltiples copias de la señal con distintos retardos y amplitudes, lo que genera una respuesta en frecuencia altamente irregular (filtros de peine).

Como consecuencia, la señal realmente percibida por el organismo difiere considerablemente de la señal diseñada, invalidando potencialmente las conclusiones bioacústicas al no poder garantizar un campo sonoro homogéneo o controlado.

## III. Propuesta de Solución

El objetivo de este trabajo es diseñar e implementar un sistema de ecualización que, mediante la caracterización del canal acústico, permita pre-distorsionar la señal de excitación para contrarrestar los efectos del entorno. El sistema propuesto asume que el estanque se comporta como un filtro desconocido cuya función de transferencia debe ser compensada activamente. Esta transferencia, además, puede variar según la posición del transductor y del hidrófono.

### *III-A. Caracterización y Compensación del Canal*

El procedimiento fundamental se basa en obtener la respuesta del canal a partir de la relación entre la emisión y la recepción de señales de prueba. Este proceso se divide en tres etapas:

1. **Identificación del sistema:** Se emite una señal de excitación conocida $x(t)$ y se captura la señal resultante $y(t)$ mediante un hidrófono en una posición específica (emulando al individuo que recibiría la señal).
2. **Estimación de la respuesta:** A partir de la relación entre la emisión y la recepción, se estima la respuesta del canal $H(f)$ en el dominio de la frecuencia:
$$H(f) = \frac{Y(f)}{X(f)} \tag{1}$$
Esta función captura las distorsiones espectrales y temporales introducidas por el recinto.
3. **Diseño del filtro inverso:** Se calcula un ecualizador o filtro de pre-distorsión $g(t)$, cuya respuesta $G(f)$ busca ser la inversa de la del canal ($G(f) \approx 1/H(f)$), de modo que la respuesta combinada sea lo más plana posible.

### *III-B. Ecualización Adaptativa Espacial*

Dado que el canal acústico depende fuertemente de la posición relativa dentro del tanque, se propone una caracterización espacial discreta. Para ello, se define una grilla de puntos dentro del volumen de interés.

En cada nodo $i$ de la grilla, se obtiene una respuesta impulsiva $h_i(t)$ y su correspondiente filtro inverso $g_i(t)$. Bajo este enfoque, el sistema puede adaptar el pre-procesamiento en función de la posición del organismo (estimada mediante sensores o visión), seleccionando dinámicamente el filtro $g_i(t)$ adecuado para compensar localmente las distorsiones.

## IV. Metodología de Implementación

El desarrollo se estructura en dos fases para separar el estudio acústico de la implementación en hardware.

### *IV-A. Fase I: Análisis y Validación en PC*

- **Estudio de factibilidad:** Pruebas iniciales sobre banco de ensayo para verificar la capacidad de recuperación de la señal original.
- **Diseño de algoritmos:** Validación de estrategias de ecualización en un entorno de alta precisión aritmética (punto flotante).
- **Caracterización espacial:** Medición del canal en distintos puntos para evaluar la densidad necesaria de la grilla y la variabilidad de la respuesta impulsiva.

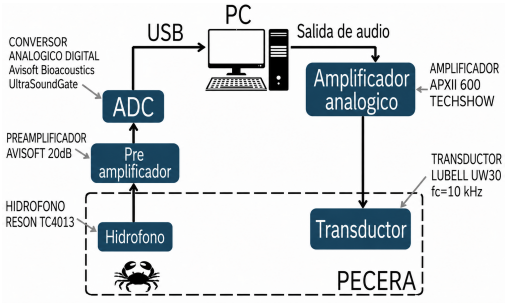



Fig. 1: Banco de Prueba

- **Optimización de algortimos para punto fijo:** Validación de estrategias de ecualización pensadas para arquitecturas de punto fijo.
- **Análisis de realimentación:** Evaluación de la necesidad de sistemas de control en tiempo real y su posible impacto acústico en el espécimen bajo estudio.

Las pruebas iniciales se realizan sobre el banco de ensayo que se muestra en 1, diseñado para el estudio de comportamiento biológico. Este entorno impone una restricción de ancho de banda con una frecuencia máxima de 10 kHz, lo cual resulta consistente con las especies de interés en este proyecto.

### *IV-B. Fase II: Implementación en Hardware Embebido*

A partir del análisis de factibilidad y los resultados obtenidos, se procede al diseño de los bloques funcionales a implementar sobre el sistema embebido. Esta etapa se centra en traducir los requerimientos algorítmicos a una arquitectura de hardware eficiente.

## V. Implementación de la Fase I: Análisis y Caracterización en PC

La primera fase del proyecto consiste en un estudio de factibilidad basado íntegramente en software. El objetivo es identificar el comportamiento del canal acústico y validar algoritmos de ecualización antes de su migración a hardware embebido.

### *V-A. Pre - Procesamiento de Señales y Alineación*

Antes de ejecutar cualquier análisis espectral o diseñar estructuras de filtrado digital, es mandatorio acondicionar las señales adquiridas. El flujo de trabajo implementado en PC comprende la remoción de componentes continuas (desplazamiento de DC), la unificación de frecuencias de muestreo (remuestreo) y la segregación de canales monoaurales. Posteriormente, se aplican dos metodologías analíticas diferenciadas para el escalado de amplitud y la sincronización temporal.

*V-A1. Normalización:* . Debido a que las señales se manejan en dominios y magnitudes físicas distintas (la señal de referencia es un archivo digital puro, mientras que la señal del hidrófono proviene de un transductor piezoeléctrico condicionado por etapas de ganancia analógica y cuantización), la comparación directa de amplitudes crudas carece de sentido analítico. Por ello, se implementaron dos estrategias complementarias de normalización:

- **Normalización por Pico (Individual):** Consiste en escalar cada señal de forma independiente dividiendo cada una por su propio valor máximo absoluto. Esta

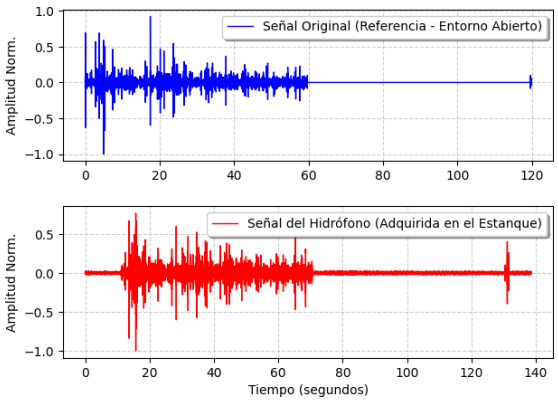



Fig. 2: Comparación temporal de la señal original y la señal adquirida por el hidrófono utilizando normalización individual por pico. Se observa la deformación de la envolvente y la persistencia de la cola de reverberación en el entorno confinado del estanque.

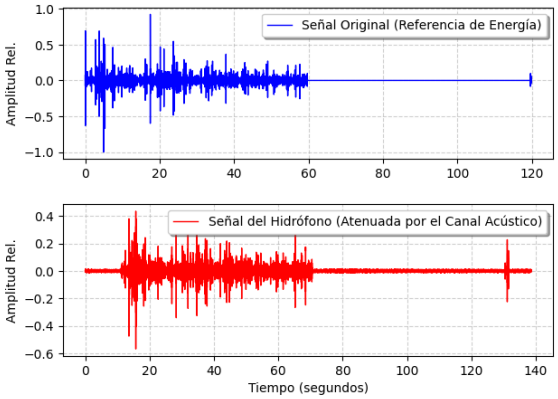



Fig. 3: Comparación temporal de la señal original y la señal adquirida por el hidrófono utilizando normalización relativa. Se observa la severa atenuación que experimenta el frente de onda al propagarse por el medio subacuático.

técnica elimina la variable de la atenuación física y es la herramienta ideal para el análisis morfológico de la forma de onda, la evaluación del ensanchamiento temporal (*smearing*) inducido por la reverberación y el cálculo matemático de la correlación cruzada. El uso de esta técnica se puede visualizar en la Fig. 2.

- **Normalización Relativa:** En este enfoque, se toma el factor de escala máximo absoluto de la señal original de referencia y se utiliza de forma unificada para dividir a ambas señales. Bajo la premisa de que las mediciones se efectúan bajo las mismas condiciones de calibración, esta estrategia conserva intactas las ganancias relativas del canal acústico y electroacústico. Visualmente, permite identificar de manera directa las pérdidas. El uso de esta técnica se puede visualizar en la Fig. 3.

*V-A2. Alineación:* como se infiere a partir de la inspección visual en la Fig. 2 y la Fig. 3, el medio acuático, la latencia inherente a la cadena de audio (filtros de reconstrucción, *buffers* del sistema operativo) y, de forma predominante, los tiempos operativos de la medición introducen un desfasaje temporal masivo. Sincronizar las señales es un requisito para que los algoritmos de ecualización operen de forma coherente muestra a muestra.

Mediante el cálculo de la función de correlación cruzada en el dominio del tiempo entre la señal original ($x_{norm}$) y la medida ($y_{norm}$), se estima el desfase temporal encontrando el retardo (*lag*) que maximiza la coincidencia geométrica de ambos vectores. Para el set de datos evaluado, se determinó un retardo bruto de $5\,732\,476$ muestras, lo que equivale a $-29{,}8566$ segundos.

Es de vital importancia discernir que este retraso macroscópico es un artefacto metodológico que contempla los tiempos humanos transcurridos entre la inicialización del software de adquisición y el disparo efectivo del estímulo sonoro.

### *V-B. Estudio de Factibilidad y Selección de Algoritmos*

En esta etapa preliminar, el objetivo es determinar si la inversión del canal es físicamente posible y qué grado de fidelidad se puede alcanzar en condiciones ideales de cómputo. Este enfoque permite que la Fase I funcione como un entorno de ”laboratorio”donde se diseña el filtro ideal para cada posición de la grilla, dejando la optimización arquitectónica para la fase de prototipado embebido.

### *V-C. El Algoritmo LS como Modelo de Referencia*

Si bien el método LS (Least Squares) es el menos eficiente para arquitecturas de hardware embebido debido a la inversión matricial requerida, actúa como un límite superior de desempeño.

Para obtener el filtro de ecualización, se implementa como primera aproximación, el método de LS por bloques [14]. A diferencia de los métodos adaptativos que actualizan coeficientes muestra a muestra, este enfoque utiliza un bloque de datos para minimizar el error cuadrático total.

La relación se plantea de forma matricial como $d \approx \mathbf{X}w$, donde $d$ es la señal deseada y $\mathbf{X}$ contiene las versiones retardadas de la señal medida. Los coeficientes del filtro inversor $w$ se obtienen mediante:

$$w = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T d \tag{2}$$

Se incorpora un término de regularización ($\lambda$) para asegurar la estabilidad numérica durante la inversión, especialmente en frecuencias donde el canal presenta atenuaciones severas.

### *V-D. Resultados Preliminares de Validación*

Para evaluar la viabilidad de la ecualización acústica, se analizó la respuesta en frecuencia del canal mediante la Transformada Rápida de Fourier (FFT), contrastando la señal original de referencia, la señal degradada por el estanque (hidrófono) y el resultado post-filtrado. Las pruebas iniciales se realizaron con un filtro FIR de $M = 8192$ coeficientes.

Como se observa en la Fig. 4, el entorno confinado del estanque introduce una distorsión espectral severa, caracterizada por profundos valles de cancelación y picos de resonancia (*comb filtering*) provocados por la interferencia destructiva y constructiva del multicamino. Al aplicar el filtro de ecualización diseñado, se logra mitigar significativamente estas alteraciones, compensando las pérdidas en las bandas críticas y suavizando la curva de respuesta del hidrófono para aproximarla a la linealidad de la señal original. Este comportamiento constituye una prueba de concepto exitosa que valida la factibilidad analítica del

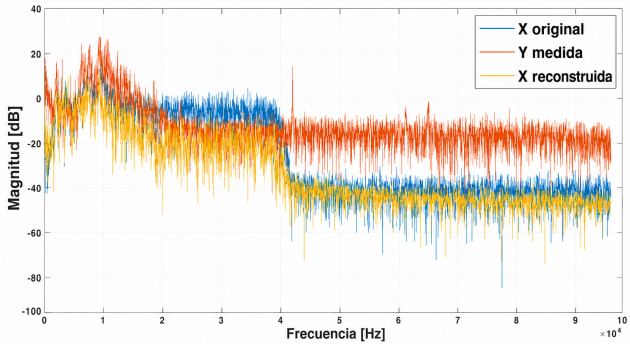



Fig. 4: Respuesta en frecuencia del canal acústico subacuático antes y después del proceso de ecualización. La curva azul representa la señal original de referencia; la curva roja detalla la señal adquirida por el hidrófono, evidenciando el severo efecto de filtro peine (*comb filtering*) provocado por el multicamino en el estanque; y la curva amarilla muestra la señal ecualizada final, demostrando la viabilidad del algoritmo.

algoritmo en software, demostrando que es técnicamente viable corregir las anomalías de transferencia del medio acústico antes de proceder a la migración del sistema al hardware embebido.

## VI. Análisis de los resultados

Los resultados presentados corresponden a una primera validación de factibilidad del enfoque propuesto. En esta etapa, el análisis se centra en verificar la capacidad del algoritmo para compensar las distorsiones introducidas por el canal acústico. Una evaluación cuantitativa representativa requerirá una campaña experimental más amplia, contemplando distintas señales de excitación y configuraciones del sistema, lo que permitirá incorporar métricas objetivas para evaluar su desempeño. Bajo este enfoque, los resultados obtenidos permiten extraer las siguientes observaciones:

- Es posible reconstruir la señal original.
- El orden del filtro ($M = 8192$) define la resolución necesaria para compensar las reflexiones de baja frecuencia en el estanque.
- La regularización $\lambda$ es indispensable para estabilizar la respuesta del ecualizador ante los ceros del canal.

### *VI-A. Desempeño del Algoritmo*

La aplicación del método LS por bloques permitió verificar la factibilidad de la inversión del canal en un entorno de punto flotante. Sin embargo, el análisis de estos resultados revela desafíos críticos de cara a la implementación embebida:

- **Precisión y estabilidad.** El uso de un término de regularización $\lambda$ resultó indispensable para estabilizar el filtro inverso ante la presencia de ceros en la respuesta del canal. Sin esta técnica, la inversión de canales altamente reverberantes tiende a la inestabilidad numérica [13].
- **Carga computacional.** El cálculo matricial directo utilizado en esta etapa posee una complejidad de $O(M^3)$. Si bien es viable en una PC para un filtro de $M = 8192$ coeficientes, resulta prohibitivo para un arquitecturas de bajo costo en tiempo real.
- **Selección de algoritmos adaptativos de bajo costo.** A partir de estos resultados, se evalúa la transición hacia implementaciones de tipo LMS (Least Mean Squares) o NLMS (Normalized LMS). Estos algoritmos permiten actualizar los coeficientes muestra a muestra con una complejidad computacional significativamente menor, de orden $O(M)$.

## VII. Conclusiones

Se presentó una propuesta preliminar para la compensación de efectos de multitrayecto en entornos subacuáticos artificiales mediante ecualización adaptativa. Los estudios preliminares permitieron validar la factibilidad de un sistema de ecualización activa para estos entornos, lo que resulta en una solución tecnológica de intervención directa y no meramente en una metodología de experimentación o un estudio pasivo de efectos acústicos.

La metodología propuesta demostró ser aplicable a experimentos biológicos en laboratorio, sistemas de acústica subacuática controlada y simulación de canales acústicos reales. Finalmente, debido a la naturaleza del diseño, el enfoque puede extenderse a otros problemas de compensación de canal en sistemas de comunicación donde la integridad de la señal sea crítica.

## VIII. Agradecimientos

Este trabajo fue financiado por el CONICET a través del proyecto PIP 2025–2027 y por la Fundación Williams.

## Referencias

[1] Carbajo and et al, "Acoustic analysis of fish tanks for marine bioacoustics research," *Journal of Marine Science and Engineering*, vol. 13, no. 7, 2025. [Online]. Available: https://www.mdpi.com/2077-1312/13/7/1253

[2] W. W. L. Au and M. C. Hastings, *Principles of Marine Bioacoustics*. New York, NY: Springer Science+Business Media, LLC, 2008.

[3] A. H. Bass and C. W. Clark, "The physical acoustics of underwater sound communication," in *Acoustic communication*. Springer, 2003, pp. 15–64.

[4] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," *IEEE communications magazine*, vol. 47, no. 1, pp. 84–89, 2009.

[5] Rogers and et al, "Parvulescu revisited: small tank acoustics for bioacousticians," in *The effects of noise on aquatic life II*. Springer, 2016, pp. 933–941.

[6] D. B. Kilfoyle and A. B. Baggeroer, "The state of the art in underwater acoustic telemetry," *IEEE Journal of oceanic engineering*, vol. 25, no. 1, pp. 4–27, 2002.

[7] T. Okumura and et al, "Analyses of small tank acoustics: Empirical and theoretical approaches," *Bioacoustics*, vol. 12, no. 2-3, pp. 330–332, 2002. [Online]. Available: https://doi.org/10.1080/09524622.2002.9753738

[8] Q. Li and et al, "Finite-element method for calculating the sound field in a tank with impedance boundaries," *Mathematical Problems in Engineering*, no. 1, p. 6794760, 2020.

[9] A. J. Duncan and et al, "Issues associated with sound exposure experiments in tanks," in *Proceedings of Meetings on Acoustics*, vol. 27, no. 1. Acoustical Society of America, 2016, p. 070008.

[10] A. Novak, M. Bruneau, and P. Lotton, "Small-sized rectangular liquid-filled acoustical tank excitation: A modal approach including leakage through the walls," *Acta Acustica united with Acustica*, vol. 104, no. 4, pp. 586–596, 2018.

[11] Y.-m. Zhang, R. Tang, Q. Li, and D.-j. Shang, "The investigation of the method for measuring the low-frequency radiated sound power in a reverberation tank," in *Proceedings of Meetings on Acoustics*, vol. 29, no. 1. Acoustical Society of America, 2016, p. 070002.

[12] M. Miyoshi and Y. Kaneda, "Inverse filtering of room acoustics," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 2, pp. 145–152, 1988.

[13] B. D. Radlović, R. C. Williamson, and R. A. Kennedy, "Equalization in an acoustic reverberant environment: robustness results," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 3, pp. 311–321, May 2000.

[14] S. Haykin, *Adaptive Filter Theory*, 4th ed. Prentice Hall, 2002.

Categoría

# Foro Tecnológico

FPGAs, HDLs y ASICs

Área Temática

ISBN 978-631-90145-4-9 - Congreso Argentino de Sistemas Embebidos CASE 2026 - 6 y 7 de agosto

# Digital Architecture for the Generation of Trigger Signals Applied to Dual Inverters

E. Z. Adame-Najera
(IEEE Student Member)
*Electronics Engineering Department*
TecNM/CENIDET
Cuernavaca, Morelos, Mexico
d20ce010@cenidet.tecnm.mx

J. Aguayo-Alquicira
(IEEE Senior Member)
*Electronics Engineering Department*
TecNM/CENIDET
Cuernavaca, Morelos, Mexico
jesus.aa@cenidet.tecnm.mx

S. E. De León-Aldaco
(IEEE Senior Member)
*Electronics Engineering Department*
TecNM/CENIDET
Cuernavaca, Morelos, Mexico
susana.da@cenidet.tecnm.mx

I. A. Juarez-Trujillo
(IEEE Student Member)
*Electronics Engineering Department*
TecNM/CENIDET
Cuernavaca, Morelos, Mexico
ivan.jtp@cenidet.tecnm.mx

***Abstract*—This paper presents the design, implementation, and functional validation of an FPGA-based digital architecture for the generation of trigger signals in a three-phase dual inverter. The proposed approach employs sinusoidal pulse-width modulation (SPWM) signals generated through lookup tables and digital comparison logic. Special emphasis is placed on the modular organization of the system, its implementation in VHDL, and its correspondence with the register-transfer level (RTL) hardware representation. The developed architecture enables the synchronized generation and distribution of trigger signals for two inverter modules operating under a common timing reference, providing a scalable and reproducible framework for dual inverter applications. Functional validation was performed through digital simulations in the ModelSim/Quartus Prime environment, confirming correct complementary switching behavior, synchronization between inverter units, and stable operation of the synchronous design. The obtained results demonstrate the feasibility of the proposed implementation as a development platform for future power electronics applications, including advanced modulation strategies, fault-tolerant operation, and experimental FPGA-based validation.**

***Index Terms*—RTL, Dual Inverter, FPGA, Trigger, Rom.**

## I. Introduction

Electrical power conversion systems have taken on a fundamental role in the development of modern industrial applications, particularly in the control of electric motors, renewable energy systems, and advanced power supplies [1], [2]. In this context, three-phase inverters represent one of the most widely used topologies for converting direct current (DC) to alternating current (AC) [3], due to their ability to generate balanced, frequency-controlled signals [4].

Traditionally, these systems have been controlled using analog platforms or microcontrollers. However, the increasing complexity of stability strategies and the need for greater temporal precision have driven the use of programmable logic devices, such as FPGAs [5]. These platforms offer significant advantages, such as parallel execution, high operating speed, and the ability to implement custom architectures [6].

Among the modulation techniques used in three-phase inverters, sinusoidal pulse width modulation (SPWM) stands out for its simplicity of implementation and its ability to generate output waveforms with low harmonic content [7]. However, implementing it in digital hardware requires a proper system architecture, particularly for reference generation, signal synchronization, and output distribution to power devices [8].

Furthermore, the use of dual three-phase inverter configurations has gained prominence in applications that demand greater flexibility, operational redundancy, or increased power capacity [9]. In such systems, proper coordination between the two three-phase inverters is critical, since any mismatch in the switching sequence or phase relationship may degrade the overall system performance [10].

In this regard, the proper definition of a digital architecture that enables the synchronized generation and distribution of switching signals is essential. Regardless of the modulation technique used, the internal organization of the control system and its alignment with the physical implementation are key design elements for FPGA-based power electronics systems.

The proposed work contributes by presenting a modular and scalable digital architecture capable of generating synchronized SPWM signals for dual three-phase inverters. This focus on dual inverter systems is particularly relevant given their accessibility and adaptability to a wide range of applications in power electronics, from renewable energy integration to advanced motor drives.

The design was functionally validated through digital simulations, ensuring reliable operation prior to hardware deployment. This approach provides a reproducible platform that can be extended or adapted for future developments, facilitating experimentation and educational use. The novelty

lies not in a new modulation strategy, but in the integration, synchronization, and organization of digital blocks within an FPGA environment, which simplifies implementation while maintaining efficiency and performance.

## II. MATERIAL AND METHODS

The proposed architecture was implemented on an Intel/Altera Cyclone IV E FPGA (EP4CE22F17C6) using VHDL and a fully synchronous design methodology. The developed architecture is based on the digital generation of modulation signals, structured into interconnected functional blocks that operate based on a common clock signal. This counter defines the resolution of the modulation system, allowing the switching cycle to be discretized. The digital carrier signal generated by the n-bit counter can be expressed as

$$c(t) = k, \quad k \in [0, 1, ..., 2^n - 1]$$

Where $c(t)$ is the carrier signal and $k$ represents its instantaneous discrete value.

The sinusoidal reference signal is generated using a lookup table stored in ROM. The table contains discretized samples of a sinusoidal waveform, avoiding the real-time computation of trigonometric functions. The reference voltage is defined as.

$$V_{ref}(t) = V_m \sin(\omega t) \tag{1}$$

Where $V_m$ is the amplitude of the reference signal and $\omega$ is its angular frequency.

The use of a lookup table reduces the computational complexity of the system by avoiding the real-time calculation of trigonometric functions.

To generate the three-phase system, a single-phase counter is used, from which three phase-shifted signals are obtained by adding constants.

$$\vartheta_A = \vartheta, \ \vartheta_B = \vartheta + 120°, \ \vartheta_C = \vartheta + 240°$$

These phase shifts are implemented digitally by offsetting the memory addresses used to access the ROM lookup table, ensuring angular coherence among the three phases.

The modulation process is performed using digital comparators that evaluate the ratio between the sinusoidal signal and the carrier:

$$PWM_x = \begin{cases} 1 & \text{Si } v_x(t) > c(t) \\ 0 & \text{Other case} \end{cases} \tag{2}$$

This method corresponds to a direct implementation of the SPWM technique in discrete time.

For each phase of the inverter, two control signals are generated corresponding to the upper and lower switches of the bridge. These signals are complementary, ensuring that there is no simultaneous conduction in both devices.

## III. SYSTEM DEVELOPMENT

### A. System Features

Figure 1 shows the block diagram of the generation of sinusoidal signals, their modulation using a digital carrier, and the subsequent distribution of signals to two six-switch three-phase inverters, enabling synchronized and efficient operation.

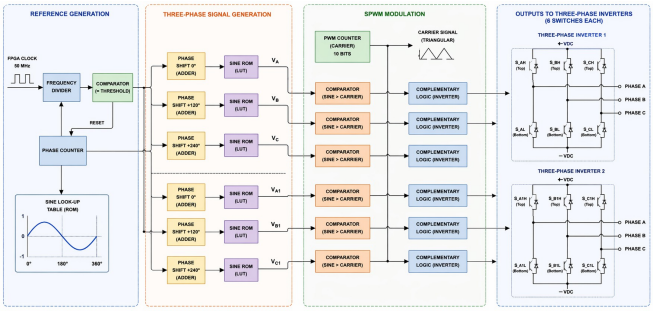



Figure 1: Diagram of the generation of sinusoidal signals.

*1) A. Digital System Architecture:* The digital architecture of the system was developed using a modular approach, in which multiple functional blocks operate concurrently, taking advantage of the parallel nature of the FPGA. The core of the system consists of a carrier signal generator based on a digital counter, which defines the timing reference for the modulation process. Simultaneously, a sinusoidal reference generator is implemented using a phase counter and ROM-type memories, enabling the generation of discrete signals with sinusouidal behavior.

*2) B. Carrier Signal Generation:* The carrier signal is generated by a 10-bit digital counter, whose value increments with each clock cycle:

$$pwm_{cnt}(k+1) = pwm_{cnt}(k) + 1 \tag{3}$$

The operating range of the counter is defined by:

$0 \leq pwm_{cnt} < 2^{10}$

The carrier signal frequency is determined by:

$$f_{PWM} = \frac{f_{clk}}{2PWM_{BITS}} \tag{4}$$

Where $PWM_{BITS}$ is the bit width of the carrier counter and was set to 10 in this implementation. Given a 50 MHz clock, this defines the base switching frequency of the system.

*3) C. Sine Wave Signal Generation:* The sine wave is generated using a 256-sample ROM, indexed by a discrete phase counter:

$$sin_{cnt}(k+1) = sin_{cnt}(k) + 1 \tag{5}$$

The progression of the counter is controlled by a frequency divider:

$$f_{out} = \frac{f_{clk}}{Nx256} \tag{6}$$

Where N=3255, which yields a fundamental frequency of approximately 60 Hz

*4) D. Three-Phase System Implementation:* The three-phase system is generated using discrete angular shifts in the ROM indices:

$$idx_A = \sin_{cnt}, \quad idx_B = \sin_{cnt} + \frac{256}{3}, \quad idx_C = \sin_{cnt} + \frac{2 \cdot 256}{3}$$

These shifts correspond to phase shifts of 120° and 240°; for the next inverter, a phase redistribution is used:

$$idx_{A1} = sin_{cnt},\ idx_{B1} = sin_{cnt} + 120°,\ idx_{C1} = sin_{cnt} + 240°$$

This allows a second three-phase system to be generated with a different configuration, while maintaining temporal synchronization.

*5) E. Implementation Using ROM:* The discrete sinusoidal signals are obtained from ROMs initialized using external files. Each phase is generated by an independent instance of the memory, allowing for simultaneous access:

$$Vref_x = ROM(idx_x)$$

Where $Vref_x$ is the discretized sinusoidal reference associated with phase $x$.

### B. RTL System Development

Based on the RTL representation, the operation of the system can be understood as a sequence of synchronous processes coordinated by the clock signal. This sequence describes the flow of information from reference generation to switching signal generation, allowing a direct relationship to be established between the hardware implementation and the functional behavior of the system.

As shown in Figure 2, the system is organized into multiple interconnected blocks that operate concurrently. On the left side is the sinusoidal signal generation block, consisting of the frequency divider $sin_{div}$ and the phase counter $sin_{cnt}$; in the central region are the summing blocks responsible for generating the angular shifts, while on the right side are the ROMs and comparators that implement SPWM modulation. Finally, the resulting signals are processed using inversion logic and distributed to the outputs corresponding to the two three-phase inverters.

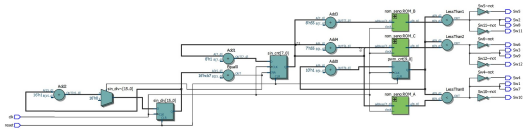


Figure 2: RTL schematic diagram.

Figure 3 shows the first stage of the system, which corresponds to the generation of the sinusoidal reference signal. In this RTL signal, a frequency divider based on the $sin_{div}$ register is implemented; this register is incremented synchronously with the clock signal. A comparator block continuously evaluates whether the divider value reaches the set threshold, allowing it to reset and enabling the increment of the $sin_{cnt}$ phase counter. This mechanism defines the rate of change of the sinusoidal signal, thereby establishing the fundamental frequency of the system. In this way, the phase counter generates a discrete representation of the electrical angle, which serves as a basis for constructing the three-phase system.

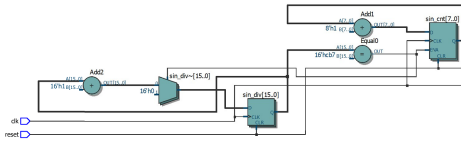


Figure 3: Generation of the sinusoidal reference signal.

The second stage of the system is shown in Figure 4, where the three-phase signals are generated based on the previously obtained phase index. In this section of the RTL, several summing blocks are identified that allow angular shifts to be applied to the base index, thereby generating signals that are 120° out of phase with one another. These indices are used to address multiple ROM-type memories, which contain discretized values of the sine wave function. As a result, the modulating signals corresponding to each phase are obtained in parallel. Additionally, a second set of reorganized indices is implemented, allowing a second three-phase system to be generated while maintaining the same time base—a fundamental characteristic for the synchronized operation of the dual inverter.

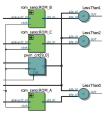


Figure 4: Schematic diagram of ROMs.

Figure 5 shows the modulation stage and the generation of switching signals. In this part of the RTL, the $pwm_{cnt}$ counter is identified, which generates the carrier signal through a continuous increment synchronized with the clock. This signal is compared with the sine wave signals using comparator blocks, resulting in the PWM signals for each phase. Subsequently, these signals are processed using inversion logic to obtain their corresponding complements, which are necessary for controlling the power switches.

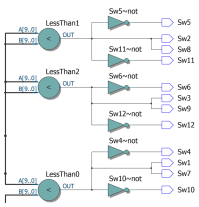



Figure 5: Generation of switching signals of dual Inverter.

**Table I:** *Parameters used in the implementation of the SPWM modulator*

| Parameter | Value | Description |
|---|---|---|
| Clock frequency | 50 MHz | Base system signal |
| Fundamental frequency | 60 Hz | Three-phase output signal |
| PWM resolution | 10 bits | Carrier resolution |
| ROM size | 256 samples | Sine wave discretization |
| Number of inverters | 2 | Dual configuration |
| Modulation type | SPWM | Sinusoidal modulation |
| Phase shift | 120° | Balanced three-phase system |
| Logic type | Synchronous | Digital control implemented in FPGA |

## IV. EXPERIMENTAL RESULT

The results presented in this section correspond to simulation-based validation of the proposed architecture, Table 1 presents the main system parameters used in the implementation of the SPWM modulator. These values define the operating conditions of the system, including the operating frequency, the modulation resolution, and the characteristics of the reference signal, all of which directly influence the performance of the three-phase dual inverter.

The proposed system was validated through digital simulations performed using the ModelSim simulator integrated within the Intel Quartus Prime development environment. Figure 6 illustrates the initial operating sequence of the proposed architecture, highlighting the effect of the reset signal and the correct initialization of the switching outputs. The results confirm that all signals are properly initialized before entering normal operation.

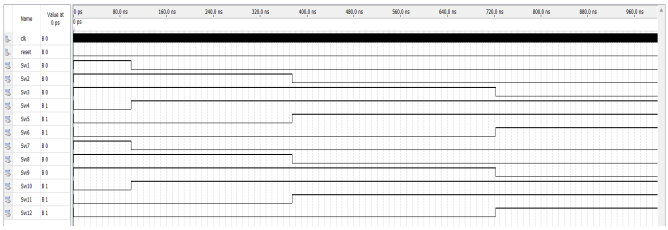


Figure 6: Trigger signal output on the dual inverter.

Figure 7 presents an extended timing diagram corresponding to steady-state operation. Several switching cycles are displayed to evaluate the periodic behavior of the generated gating signals.

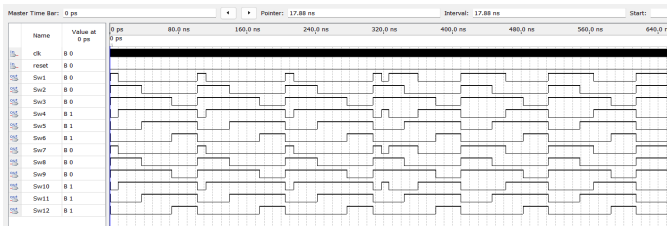


Figure 7: Detailed timing diagram of the gating signals generated for the dual three-phase inverter during steady-state operation.

The complementary operation between the upper and lower switches of each inverter leg is clearly observed. In particular, the pairs (Sw1–Sw4), (Sw2–Sw5), and (Sw3–Sw6), as well as (Sw7–Sw10), (Sw8–Sw11), and (Sw9–Sw12), maintain mutually exclusive activation states throughout the simulation interval.

## V. CONCLUSIONS

This paper presented the development of an SPWM-based system for generating gating signals in a dual three-phase inverter implemented on an FPGA, demonstrating the feasibility of integrating reference generation, digital modulation, and signal distribution within a parallel and scalable architecture.

The proposed architecture was functionally validated through digital simulations performed in the ModelSim environment, showing stable PWM signal generation, proper complementary switching behavior, and synchronized operation between both inverter units. These results confirm the correct implementation of the proposed synchronous design methodology and establish a functional basis for dual inverter applications in power electronics.

The modular structure adopted in this work facilitates future extensions toward more complex topologies, the incorporation of advanced modulation techniques, and fault-tolerant control strategies. Future work will focus on experimental validation using FPGA development platforms, spectral analysis of the generated signals, and the evaluation of hardware performance metrics such as resource utilization and operating frequency

## REFERENCES


[1] S. Hannan, S. Aslam, and M. Ghayur, "Design and real-time implementation of spwm based inverter," in 2018 International Conference on Engineering and Emerging Technologies (ICEET). IEEE, 2018, pp. 1–6.

[2] T. Bhattacharjee, M. Jamil, and A. Jana, "Design of spwm based three phase inverter model," in 2018 Technologies for Smart-City Energy Security and Power (ICSESP). IEEE, 2018, pp. 1–6.

[3] E. Z. Adame Najera, S. E. De León Aldaco, J. Aguayo Alquicira, R. E. Lozoya-Ponce, J. Á. Pecina-Sánchez, and S. Portillo Contreras, "A survey on topologies and modulation strategies of dual inverters in industrial applications," Eng, vol. 6, no. 11, p. 316, 2025.

[4] A. Ray, S. Datta, A. Biswas, and J. N. Bera, "Design of a multilevel inverter using spwm technique," in Advances in Control, Signal Processing and Energy Systems: Select Proceedings of CSPES 2018. Springer, 2019, pp. 215–229.

[5] M. A. Hosseinzadeh, M. Sarebanzadeh, M. Rivera, E. Babaei, and P. Wheeler, "A reduced single-phase switched-diode cascaded multilevel inverter," IEEE Journal of Emerging and Selected Topics in Power Electronics, vol. 9, no. 3, pp. 3556–3569, 2020.

[6] S. Yan, Y. Yang, S. Hui, and F. Blaabjerg, "A review on direct power control of pulsewidth modulation converters," IEEE Transactions on Power Electronics, vol. 36, no. 10, pp. 11 984–12 007, 2021.

[7] K. R. Kumar, M. Venkatesan, and R. Saravanan, "A hybrid control topology for cascaded multilevel inverter with hybrid renewable energy generation subsystem," Solar Energy, vol. 242, pp. 323–334, 2022.

[8] M. K. Jarjes and T. A. Hussein, "Comparative study of spwm and svpwm techniques for the control of three-phase grid connected inverter." Przegląd Elektrotechniczny, vol. 99, no. 5, 2023.

[9] D. Chattejee, C. Chakraborty, and S. Dalapati, "Pulse-width modulation techniques in two-level voltage source inverters-state of the art and future perspectives," Power Electronics and Drives, vol. 8, pp. 335–367, 2023.

[10] Q. Yin, D. Li, Z. Wu, S. H. Pun, and Y. Liu, "Design and implementation of risc-v system-on-chip for spwm generation based on fpga," IEICE Electronics Express, vol. 21, no. 24, pp. 20 240 603–20 240 603, 2024.

ISBN 978-631-90145-4-9 - Congreso Argentino de Sistemas Embebidos CASE 2026 - 6 y 7 de agosto

# Receptor SDR sobre FPGA con Linux Embebido

Yosel de Jesús Balibrea Lastre
*Observatorio Pierre Auger*
Malargüe, Mendoza, Argentina
yosel@auger.org.ar

*Abstract*— ***La instrumentación científica se ha visto ampliamente beneficiada en los últimos años por la evolución de dispositivos reconfigurables como las matrices de arreglo programable (FPGA) y, en particular, por los sistemas en chip (SoC) que pueden integrar varios procesadores, convertidores analógico-digitales (ADC) y otros periféricos especializados. Asimismo, la incorporación de sistemas operativos como Linux embebido brinda la posibilidad de utilizar herramientas de más alto nivel para el análisis, transferencia y control de los datos. El presente trabajo describe la implementación de un receptor de radio definida por software (SDR) sobre una plataforma SoC FPGA Zynq, así como su control mediante Linux embebido, utilizando principalmente herramientas de código abierto.***

**Keywords— FPGA, Linux, procesamiento digital de señales**

## I. INTRODUCCIÓN

En la actualidad muchos sistemas de comunicaciones, mediciones, control, etc. ya sea con fines industriales o científicos se benefician de sistemas reconfigurables como las FPGA y SoC FPGA. Estas plataformas permiten integrar procesamiento digital de señales, adquisición de datos y sistemas de control dentro de una misma arquitectura, ofreciendo alta flexibilidad y capacidad de adaptación a diferentes aplicaciones. El observatorio de rayos cósmicos de altas energías Pierre Auger [1] hace uso también de estas tecnologías donde destacan los sistemas de adquisición de los detectores de fluorescencia [2], los LIDAR de monitoreo atmosférico y en mayor medida los detectores de superficie. Los nuevos experimentos que se llevan a cabo también cumplen con esta tendencia. Si bien las herramientas modernas de desarrollo para FPGA permiten acelerar el diseño mediante el uso de núcleos de propiedad intelectual (IP), muchas de estas soluciones generan dependencia a fabricantes o entornos específicos. Esto puede limitar la portabilidad, la reutilización y la comprensión profunda de los sistemas implementados, especialmente en entornos académicos y científicos donde la flexibilidad y la capacidad de adaptación resultan fundamentales.

El presente trabajo desarrolla un receptor SDR en una SoC FPGA de AMD/Xilinx (Zynq) [3] mediante el lenguaje de descripción de hardware (HDL) verilog, describiendo sus principales componentes, arquitectura y resultados de simulación. Asimismo, se integra un sistema Linux embebido para el control y transferencia de datos entre la FPGA y el procesador [4]. La sección II muestra la arquitectura de este diseño, así como su funcionamiento y componentes que forman al receptor. La sección III muestra el diseño del oscilador numéricamente controlado (NCO) y los resultados de las simulaciones. La Sección IV aborda el diseño del filtro CIC y la obtención de las componentes en fase (I) y cuadratura (Q). En la sección V se realiza el proceso de demodulación, donde se toma como ejemplo la modulación de amplitud (AM). La Sección VI describe la transferencia de datos entre la FPGA y el procesador mediante Linux embebido basado en Debian 13. Finalmente, la Sección VII expone las pruebas y resultados obtenidos en simulación y hardware.

## II. ARQUITECTURA DEL SISTEMA

Las simulaciones funcionales de este SDR se realizaron mediante Icarus Verilog y algunos bancos de prueba fueron implementados en Python utilizando Cocotb. La síntesis y validación sobre hardware se realizaron utilizando el entorno Vivado. La arquitectura implementada corresponde a un receptor de conversión directa, tal como se muestra en la figura 1. La señal de entrada puede provenir de un ADC de 12 bits a 64MSPS, en este caso el AD9226 o de un generador de patrones interno utilizado para pruebas y validación. El oscilador numéricamente controlado (NCO) es utilizado para generar las señales seno y coseno correspondientes a la frecuencia de recepción. Estas señales alimentan dos mezcladores implementados mediante multiplicadores signados, obteniendo las ramas I/Q. Las salidas de ambos mezcladores son procesadas mediante filtros CIC, los cuales realizan el filtrado y decimación de la señal [5] [6]. Posteriormente, las componentes I/Q son procesadas mediante un Computador digital por rotación de coordenadas (CORDIC) que permite obtener la magnitud y fase del vector complejo asociado a la señal recibida. Finalmente, la señal resultante es filtrada y transferida desde la lógica programable (PL) hacia el sistema de procesamiento (PS) mediante una interfaz AXI-Stream conectada a un controlador DMA, permitiendo su procesamiento y visualización desde Linux embebido.

## III. OSCILADOR NUMÉRICAMENTE CONTROLADO

El oscilador numéricamente controlado (NCO) permite seleccionar la frecuencia de recepción del sistema. Su implementación se basa en una memoria LUT de 128 palabras de 12 bits en formato de punto fijo (1 bit entero y 11 bits fraccionales) que almacena un cuarto de período de la señal senoidal, junto a un acumulador [7]. Los dos bits más significativos del acumulador se utilizan para determinar el cuadrante de la señal, permitiendo recorrer la LUT en sentido directo o inverso y realizar la inversión de signo correspondiente. De esta forma se obtiene una implementación eficiente en recursos y capaz de generar señales de hasta 15 MHz con bajos niveles de ruido, considerando una frecuencia de reloj del sistema de 64 MHz. El valor de la salida coseno se logra desfasando en un cuadrante a la señal seno original. El acumulador es incrementado mediante una palabra de control de 32 bits calculada en el procesador y enviada a la FPGA mediante una interfaz genérica de entrada - salida (GPIO). La ecuación (1) muestra el cálculo de dicho parámetro, donde $f_{nco}$ corresponde a la frecuencia deseada y $f_{clk}$ es la frecuencia del reloj. En la figura 2 se muestran los resultados de la simulación para este módulo para una frecuencia de salida de 5MHz, allí se observan algunas señales espaciadas a 1MHz aproximadamente debido a ruido armónico y sus aliases producto de la cuantización al momento de crear la LUT.

$$Incremento = \frac{f_{nco}}{f_{clk}} * 2^{32} \quad (1)$$

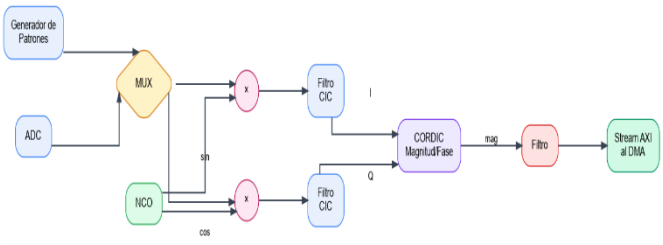



Fig. 1. Estructura del receptor SDR.

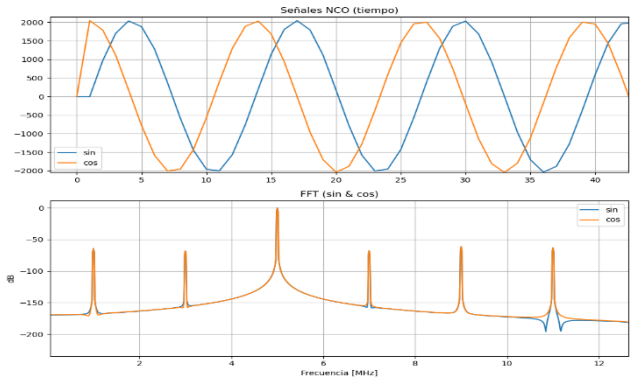



Fig. 2. Resultados de simulación del NCO.

## IV. FILTRO CIC

El filtro CIC implementado consta de cinco etapas integradoras y diferenciadoras respectivamente, con un factor de decimación D=2048 y retardo diferencial unitario [8]. Además, el diseño permite controlar la ganancia de salida mediante una entrada configurable desde el PS por medio de un GPIO. Con estos parámetros se obtiene una frecuencia de muestreo de salida de 31.25kHz. La banda pasante medida en la práctica es de aproximadamente 8 kHz, presentando una atenuación cercana a -3 dB a esta frecuencia.

La respuesta en frecuencia del filtro CIC se encuentra definida en (2) [9], donde D representa factor de decimación y M el número de etapas del filtro. El comportamiento temporal del filtro puede expresarse mediante (3). Las figuras 3 y 4 muestran la respuesta temporal y en frecuencia obtenidas mediante simulación para una señal de entrada de 4KHz con componentes de 18KHz, 35KHz y 48KHz.

Con el fin de evitar desbordamientos y garantizar la correcta operación de esta etapa, el largo de los registros usados en este módulo, deben ser seleccionados cuidadosamente. La expresión (4) muestra cómo se realizó el cálculo del ancho de estos registros que en este caso fue de 79 bits.

$$|Hcic\ (e^{j2\pi f})| = \left|\frac{\sin\left(\frac{2\pi f D}{2}\right)}{\sin\left(\frac{2\pi f}{2}\right)}\right|^{M} \quad (2)$$

$$y(n) = x(n) - x(n-D) + y(n-1) \quad (3)$$

$$Ancho = ancho_{xn} + (M * \log_2 D) \quad (4)$$

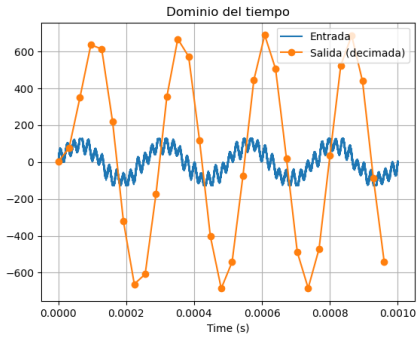



Fig. 3. Resultados de la simulación del filtro CIC.

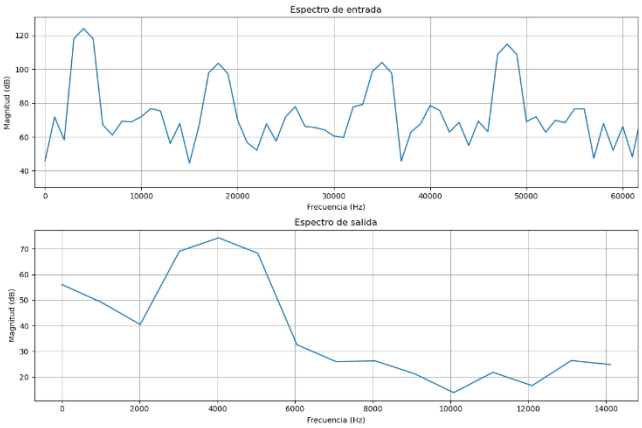



Fig. 4. Respuesta en frecuencia del filtro CIC.

## V. DEMODULADOR

Para la demodulación de las señales I/Q obtenidas se implementó un algoritmo CORDIC [10], este algoritmo permite el cálculo de cualquier función trigonométrica mediante iteraciones. Los resultados obtenidos dependen del modo de operación que se seleccione, ya sea rotacional o vectorial. El primer modo permite el cálculo de funciones como el seno y coseno, mientras que el último permite calcular el valor de magnitud y fase del vector formado por las componentes I/Q.

Con el objetivo de aumentar la eficiencia del sistema, el módulo fue implementado utilizando una arquitectura pipeline, permitiendo obtener muestras consecutivas luego de 10 ciclos de reloj, dos de estos ciclos son usados para ajustar o rotar los valores de entrada pues el algoritmo está definido para el primer y cuarto cuadrantes. El dominio angular de salida se encuentra comprendido entre -π y π, representados por los valores −32768 y 32767 respectivamente para datos signados de 16 bits. Las ecuaciones (5), (6) y (7) describen el funcionamiento iterativo del algoritmo CORDIC.

$$x_{i+1} = K_i(x_i - d_i y_i 2^{-i}) \quad (5)$$

$$y_{i+1} = K_i(d_i x_i 2^{-i} + y_i) \quad (6)$$

$$z_{i+1} = z_i - d_i \tan^{-1}(2^{-i}) \quad (7)$$

El parámetro di determina la dirección de rotación, tomando los valores -1 o 1 dependiendo del signo de yi, de esta forma se itera haciendo que el yi se aproxime a cero. Para la última iteración el valor de la fase será z(n) y la magnitud x(n), siendo n el número de iteraciones. El factor de escala Ki se muestra en (8), los ángulos de rotación se seleccionan de forma que se cumpla la expresión (9) de esta forma se

sustituyen multiplicaciones por operaciones de desplazamiento de bits, estos valores son calculados previamente en un script en Python e insertados en una pequeña LUT de la misma profundidad que el número de iteraciones, 8 elementos en este caso.

$$K_i = \cos(\theta i) \qquad (8)$$

$$\theta i = \tan^{-1}(2^{-i}) \qquad (9)$$

La figura 5 muestra los resultados obtenidos en simulación, para el caso de prueba (I = 100; Q = 0) se observa un error aproximado de 0.4 grados en la estimación de fase, considerando que cada grado corresponde a un valor de 182 unidades. Finalmente, a la salida del módulo CORDIC se implementa un pequeño filtro que permite acondicionar la señal antes de su transferencia al PS.

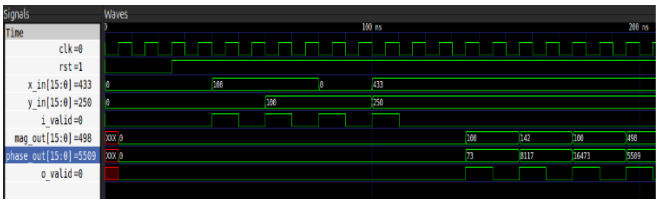



Fig. 5. Simulación del módulo CORDIC.

## VI. Transferencia y Procesamiento de Datos

La transferencia de datos hacia el PS se realizó mediante el IP *AXI Direct Memory Access* (AXI DMA) de Xilinx, el cual permite transferir bloques de datos directamente hacia la memoria RAM del sistema sin intervención continua del procesador [11]. Esta arquitectura reduce significativamente la carga computacional de la CPU. Para el acceso desde el sistema operativo Linux [12], se desarrolló un módulo de kernel encargado de mapear y copiar los datos desde el espacio de memoria física asociado al DMA hacia el espacio de usuario. La comunicación con las aplicaciones de usuario se implementó mediante un dispositivo de tipo caracter ubicado en **/dev/sdr0** [13]. Una vez en el PS, se desarrollaron dos aplicaciones, una escrita en lenguaje C y otra en Python, para el procesamiento y visualización de la señal las cuales incluyen reproducir la señal demodulada a través de dispositivos de audio, así como herramientas de análisis y visualización mediante una aplicación web. La estructura interna de estos programas está dividida en diferentes hilos de procesamiento; el primero es el encargado de tomar los datos desde el driver y guardarlos en buffers circulares, el segundo se encarga de escribir las muestras en el dispositivo de audio y el último se utiliza para el control del receptor por medio de los GPIOs y en el caso de la implementación en Python se calcula la transformada rápida de Fourier (FFT) y se implementa un pequeño servidor web. En la figura 6 se puede apreciar la estructura de este programa.

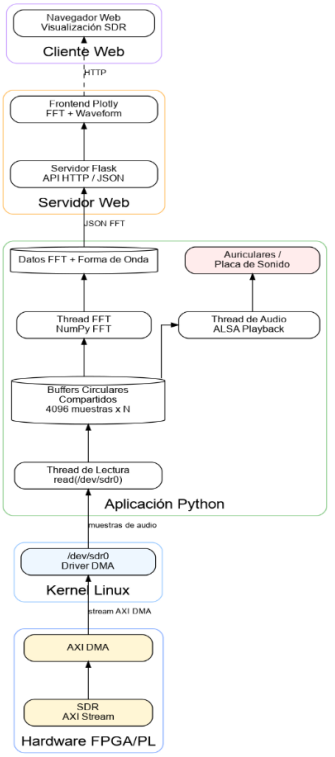



Fig. 6. Estructura del software del SDR.

## VII. Resultados y Pruebas en Hardware

La simulación del receptor SDR fue realizada utilizando un banco de pruebas desarrollado en Cocotb. Este entorno permitió generar una señal AM de estímulo, compuesta por una portadora de 1 MHz modulada por una señal de 1 kHz, además de las señales de control necesarias para validar el funcionamiento de cada una de las etapas del diseño.

La Fig. 7 muestra los resultados obtenidos para la simulación del sistema completo, verificando el correcto funcionamiento de las etapas de conversión, filtrado y demodulación implementadas. Asimismo, puede observarse el efecto del filtro removedor de corriente continua (DC removal) sobre la señal demodulada. Este módulo es el encargado de generar las señales correspondientes a la interfaz AXI-Stream (AXIS) [14] utilizada para la transferencia de datos.

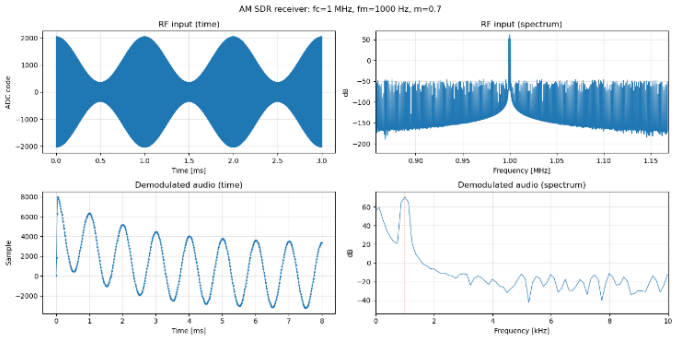



Fig. 7. Resultados de la simulacion: señal de entrada (arriba), señal de salida (abajo).

La figura 8 muestra el hardware utilizado para la implementación y validación del diseño, en este caso se utilizó la placa de desarrollo Minized de Avnet. Finalmente, en la

figura 9 se muestra la recepción de una estación (la LV19) a la frecuencia de 790KHz.

Se puede apreciar que este sistema tiene algunas limitaciones en cuanto a ruido debido a la falta de filtrado después de los CIC, además se pudo comprobar que este módulo ADC de bajo costo induce una atenuación de -8dB aproximadamente, lo cual puede ser corregido modificando el circuito. Finalmente, la figura 10 muestra el uso de recursos en el dispositivo luego de la implementación.

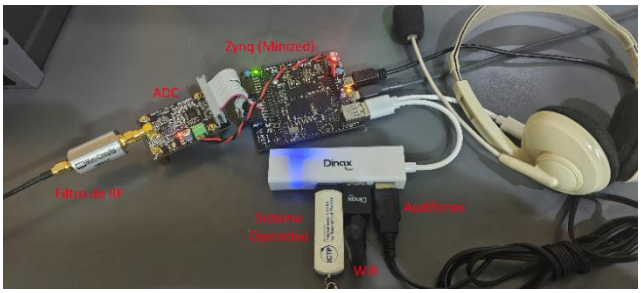



Fig. 8. Hardware utilizado.

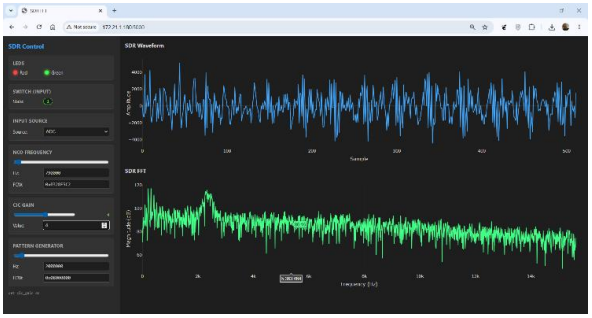


Fig. 9. Recepción de una estación AM a 790KHz.

| Resource | Utilization | Available | Utilization % |
|---|---|---|---|
| LUT | 8980 | 14400 | 62.36 |
| LUTRAM | 533 | 6000 | 8.88 |
| FF | 10494 | 28800 | 36.44 |
| BRAM | 3.50 | 50 | 7.00 |
| DSP | 4 | 66 | 6.06 |
| IO | 32 | 54 | 59.26 |

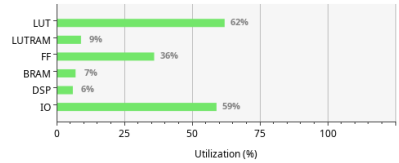



Fig. 10. Utlilización de recursos en el dispositivo.

## REFERENCIAS

[1] The Pierre Auger Collaboration, “The Pierre Auger Cosmic Ray Observatory,” *Nuclear Instruments and Methods in Physics Research Section A,* vol. 798, pp. 172-213, 2015.

[2] H. Gemmeke, “The fluorescence detector electronics of the Pierre Auger Observatory,” *Nuclear Instruments and Methods in Physics Research Section A,* vol. 518, nº 1-2, pp. 180-182, 2004.

[3] AMD Xilinx, “Zynq-7000 SoC Technical Reference Manual UG585,” 2026. [En línea]. Available: https://docs.amd.com/r/en-US/ug585-zynq-7000-SoC-TRM/Introduction?tocId=Hf6C7Oo5ABvv2hkWRoiihQ.

[4] Y. d. J. Balibrea Lastre, “Congreso Argentino de Sistemas Embebidos,” [En línea]. Available: https://case.ar/wp-content/uploads/2023/12/libro2021.pdf. [Último acceso: 18 Abril 2026].

[5] M.-B. Uwe, Digital Signal Processing with Field Programmable Gate Arrays, 2014.

[6] R. Lyons, Understanding Digital Signal Processing, Prentice Hall, 2011.

[7] Analog Devices, “A Technical Tutorial on Digital Signal Synthesis,” Application Note AN-823, 2004.

[8] R. Lyons, “A Beginner's Guide To Cascaded Integrator-Comb (CIC) Filters,” [En línea]. Available: https://www.dsprelated.com/blogimages/RickLyons/CIC_digital_filters-Ver_3-Lyons.pdf.

[9] R. Lyons, “Two Easy Ways To Test Multistage CIC Decimation Filters,” 2018. [En línea]. Available: https://www.dsprelated.com/showarticle/1171.php.

[10] R. Andraka, “A survey of CORDIC algorithms for FPGA based computers,” *Proceedings of the 1998 ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays,* p. 191–200, 1998.

[11] AMD Xilinx, “AXI DMA v7.1 LogiCORE IP Product Guide,” 2021. [En línea]. Available: https://www.amd.com/content/dam/xilinx/support/documents/ip_documentation/axi_dma/v7_1/pg021_axi_dma.pdf.

[12] B. D. Stuart, “xilinx_axidma: Zero-copy Linux driver and userspace interface for Xilinx AXI DMA and VDMA IP cores,” 2024. [En línea]. Available: https://github.com/bperez77/xilinx_axidma. [Último acceso: 5 Mayo 2026].

[13] A. R. a. G. K.-H. J. Corbet, Linux Device Drivers, Sebastopol: O’Reilly Media, 2005.

[14] ARM Ltd, “AMBA AXI and ACE Protocol Specification AXI4-Stream Protocol Specification, IHI 0051,” 2026. [En línea]. Available: https://developer.arm.com/documentation/ihi0051/latest.

Categoría

# Foro Tecnológico

Implementación de SE

Área Temática

# Placa Shield MCXC444 DSI para el kit de desarrollo FRDM-MCXC444

Daniel Márquez, Agustín Zuliani y Walter Wagner
Departamento de Sistemas e Informática (D.S.I.)
Facultad de Ciencias Exactas, Ingeniería y Agrimensura (F.C.E.I.A.)
Universidad Nacional de Rosario, Rosario, Argentina
Email: dmarquez@fceia.unr.edu.ar, azuliani02@fceia.unr.edu.ar, wagner@fceia.unr.edu.ar

***Resumen*—Este trabajo presenta el diseño, implementación y validación de una placa de expansión tipo shield destinada a acoplarse con la placa de desarrollo comercial FRDM-MCXC444, la cual integra un microcontrolador ARM Cortex-M0+. El objetivo es ampliar las capacidades de la placa base mediante periféricos orientados a la formación académica y al desarrollo de aplicaciones en sistemas embebidos. La placa incorpora salida MOSFET de potencia, relé simple inversor, LEDs de propósito general, entradas analógicas y digitales, reloj de tiempo real, memoria EEPROM, buzzer, display OLED y módulos de comunicación industrial RS-485 y CAN Bus, entre otros recursos. Se incluye además alimentación externa de 12 V con regulación a 5 V y 3,3 V, y puertos de expansión UART, SPI e I²C. La plataforma permite a los estudiantes realizar un conjunto amplio de prácticas de laboratorio para diversas asignaturas, evitando el uso de placas de prototipado que suelen presentar problemas de fiabilidad en entornos educativos, y reduciendo la dependencia de tarjetas de desarrollo de mayor costo.**

***Palabras clave*—sistemas embebidos, diseño electrónico, RTOS, Zephyr, PCB, CAN Bus, RS-485**

## I. Introducción

En el desarrollo de Sistemas Embebidos intervienen cuestiones relacionadas al hardware y al software, existiendo una sinergia entre ambas partes para un resultado exitoso. En el ambiente académico, es necesario optimizar recursos para mejorar la experiencia en el proceso enseñanza-aprendizaje, para que los estudiantes logren un aprendizaje significativo. Más aún, vincular vertientes prácticas en estos procesos, pone en marcha mecanismos experienciales en los estudiantes que logran consolidar conocimientos trabajados desde los aspectos teóricos.

Por todo esto se llevó adelante el diseño e implementación de una placa que permite a los estudiantes en diferentes materias del área de Sistemas Digitales y de Sistemas Embebidos Avanzados, desarrollar firmware embebido tanto en Bare Metal o basado en Sistema Operativo en Tiempo Real (RTOS), y al mismo tiempo, trabajar con una sólida plataforma de hardware para proyectos de aplicación, basados en estos dos paradigmas de programación.

## II. Objetivos

El objetivo de este proyecto es diseñar e implementar una placa de expansión tipo shield para acoplar a la placa de desarrollo comercial FRDM-MCXC444 [1], basada en un microcontrolador ARM Cortex-M0+. Esta placa se utiliza como herramienta transversal en la carrera de Ingeniería Electrónica, en asignaturas que abordan programación en lenguaje C, software embebido, microcontroladores, sistemas digitales, sistemas embebidos avanzados y sistemas operativos en tiempo real (RTOS).

Las distintas asignaturas presentan requerimientos de hardware específicos para cubrir las temáticas de su estructura curricular. A partir de un relevamiento de los requerimientos particulares y comunes de cada espacio, se define un conjunto mínimo de funcionalidades considerando las restricciones dimensionales impuestas por el factor de forma de la placa base FRDM y sus conectores de expansión. La placa shield incluye como mínimo:

- Etapas de potencia para implementación de sistemas de control a lazo cerrado.
- Alimentación externa de 12 V para excitar las etapas de salida.
- Sensor de temperatura para aplicaciones de control o monitoreo distribuido.
- Puertos de comunicación multipunto RS-485 y CAN Bus para prácticas específicas en diferentes asignaturas.
- Interfaz de usuario mediante display OLED.
- Puertos de expansión para acceso a pines del microcontrolador.

A partir de estos requerimientos mínimos, y considerando posibles aplicaciones académicas futuras, se definen las siguientes características adicionales:

- Alimentación de 12 V CC con regulación a 5 V y 3,3 V, con LED indicador.
- Protección con fusible térmico reseteable en etapas de potencia.
- Salida con relé simple inversor con LED indicador.
- Salida con MOSFET canal N con LED indicador.
- Dos pulsadores tipo Tactile Switch.
- Sensor de temperatura LM35 integrado en la placa o conectable externamente.
- Tres LEDs de indicación para salidas GPIO.
- Potenciómetro de ajuste (trimpot) para lectura de señal analógica.
- Memoria EEPROM $I^2C$.
- RTC DS1307 con batería de respaldo.
- Buzzer.

La disponibilidad de esta plataforma en conjunto con la placa FRDM-MCXC444 permite alcanzar los siguientes objetivos académicos:

- Desarrollar software embebido basado en RTOS para estudiar sistemas en tiempo real y analizar los recursos de implementación necesarios para alcanzar los requerimientos de respuesta.
- Ensayar periféricos, protocolos de comunicación industrial, sistemas de control y aplicaciones diversas con recursos propios de los sistemas embebidos.
- Realizar implementaciones en bare metal con modelado de software, contando con soporte directo de hardware para pruebas.
- Obtener prácticas de laboratorio con distintos enfoques según la asignatura, permitiendo variantes en las actividades.
- Implementar proyectos de distintas complejidades, incorporando funcionalidades típicas de los sistemas embebidos que acerquen al estudiante a la actividad profesional.

## III. Placa Shield MCXC444 DSI

Como se comentó en la sección anterior, la placa de desarrollo posee varios periféricos que permiten ser usados por el desarrollador para su estudio. En las siguientes secciones vamos a mostrar como fue el proceso de desarrollo para los hardware más importantes de la placa, así como también el firmware/ejemplos que se fueron realizando para verificar que todo funcionase. Presentamos ademas un diagrama en bloques general.

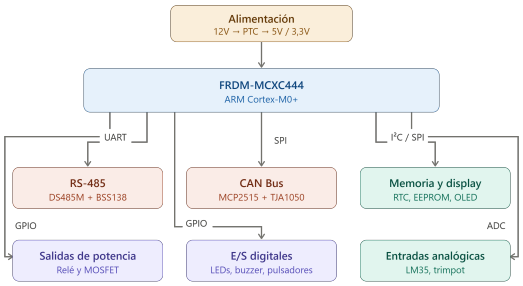



Figura 1: Diagrama de bloques general de la placa SHIELD.

### A. *Diseño del Hardware*

En el diseño del hardware se tuvieron varios aspectos presentes a la hora de su implementación. Uno de los criterios fundamentales fue que la placa shield complementara y sumara funcionalidades a los periféricos que ya posee la placa FRDM-MCXC444, entre los cuales se encuentran un sensor de luz ambiente, LED RGB multicolor y display LCD de 7 segmentos. Teniendo presente estos elementos preexistentes, en el diseño físico de la placa fue necesario realizar cortes estratégicos en el PCB de forma que dichos periféricos de la placa base no quedaran obstruidos o inutilizables al montar la shield. En la Figura 2 y la Figura 3 se muestran el diseño en 3D del lado superior (top) e inferior (bottom) de la placa que fundamentan lo antes mencionado. Además en la Figura 4 se muestra una imagen de la placa completa, con todos sus componentes soldados.

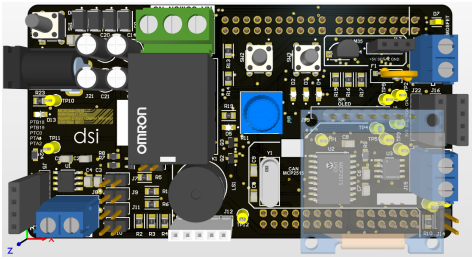



Figura 2: Imagen 3D del lado top de la placa estilo SHIELD

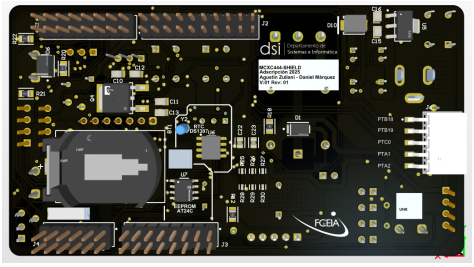



Figura 3: Imagen 3D del lado bottom de la placa estilo SHIELD

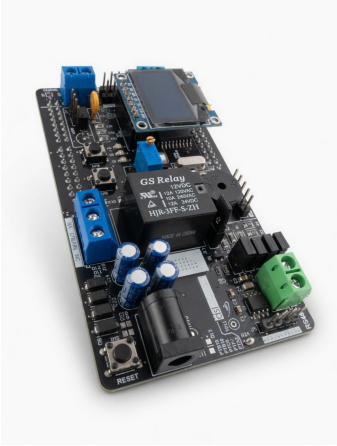



Figura 4: Imagen de la placa real en perspectiva

1. ***Consideraciones dimensionales y de compatibilidad:*** Una restricción importante es la dimensión física de la placa. Al tratarse de un diseño tipo shield que se monta sobre la placa base FRDM-MCXC444, las dimensiones no deben exceder significativamente las de esta última, manteniendo una relación de aspecto compatible con el factor de forma establecido por los conectores de expansión Arduino.
2. ***Asignación de pines y compatibilidad con firmware existente:*** Un aspecto crítico en el diseño es la selección de pines del microcontrolador para cada periférico. Esta decisión no es arbitraria, sino que se basa en la disponibilidad de

drivers y bibliotecas de software previamente desarrollados para periféricos como ADCs, interfaces RS-485, pulsadores, comunicaciones SPI e $I^2C$, entre otros.

La estrategia adoptada consiste en mapear los periféricos de la shield a aquellos pines del MCXC444 que ya cuentan con código validado, lo que permite:

- Acelerar el proceso de validación funcional.
- Reutilizar código existente y probado.
- Facilitar la portabilidad entre proyectos.
- Reducir el tiempo de desarrollo de firmware.

A continuación se describen dos de los periféricos más relevantes implementados en la placa:

1. **Módulo CAN BUS**: Para la implementación del bus CAN se utiliza el controlador **MCP2515** de Microchip [2], el cual se comunica con el microcontrolador mediante interfaz **SPI**. Este integrado actúa como controlador CAN completo, gestionando el protocolo de comunicación de forma autónoma. El MCP2515 requiere un oscilador externo, para lo cual se incorpora un cristal de 8 MHz con sus capacitores de carga asociados.
   Complementando al MCP2515, se integra un transceiver **TJA1050**, cuya función es convertir las señales diferenciales del bus CAN (CANH/CANL) a niveles lógicos TTL compatibles con el MCP2515. Esta configuración en dos etapas (controlador + transceiver) permite:
   - Aislamiento entre la lógica del microcontrolador y el bus físico.
   - Protección contra transitorios y sobretensiones del bus.
   - Compatibilidad con redes CAN industriales estándar.

   El módulo incluye resistencia de terminación de 120 Ω conmutable mediante jumper, permitiendo configurar el nodo como extremo de red cuando sea necesario.
   **Conexiones principales:**
   - SPI: MOSI, MISO, SCK (interfaz con MCXC444).
   - CS (Chip Select): control de selección.
   - INT: señal de interrupción.
   - CANH/CANL: conexión al bus físico mediante bornera.
2. **Módulo RS-485**: La interfaz RS-485 está implementada mediante un transceiver que realiza la conversión entre niveles lógicos TTL y los niveles diferenciales del estándar RS-485. Este módulo permite comunicaciones multipunto half-duplex en entornos industriales con alta inmunidad al ruido.
   El circuito integra un driver diferencial basado en el integrado DS485M (compatible MAX-485), que convierte las señales UART del microcontrolador (TX/RX) en señales diferenciales balanceadas (A/B). Dado que el transceiver opera a 5 V y el microcontrolador a 3,3 V, se incorpora un level shifter basado en el MOSFET BSS138 en la línea de recepción (RO) para garantizar la compatibilidad de niveles lógicos.
   **Características principales:**
   - Control de dirección (DE/RE) para operaciones half-duplex.
   - Resistencia de terminación de 120 Ω conmutable mediante jumper.
   - Adaptación de niveles 3,3 V/5 V en la línea de recepción.

### *B. Ensamblado y validación inicial del prototipo*

Una vez recibidas las primeras unidades del circuito impreso (PCB), se procedió al ensamblado manual de los prototipos iniciales con fines de validación funcional. Esta etapa resultó importante dentro del ciclo de desarrollo, ya que permitió identificar oportunidades de mejora tanto en el ruteado de pistas como en la disposición de componentes, la serigrafía y aspectos generales del diseño orientados a facilitar el montaje y la mantenibilidad del sistema.

Cabe destacar que, debido a la disponibilidad en el mercado local y a criterios de costo, se adoptó como estrategia de implementación el reaprovechamiento de componentes provenientes de módulos comerciales. Diversos dispositivos fueron desoldados de placas existentes para su posterior integración en el prototipo desarrollado. Esta decisión permitió reducir costos y tiempos de aprovisionamiento, manteniendo la funcionalidad requerida para la validación del sistema.

La estrategia de verificación se estructuró de manera incremental, validando subsistemas de forma independiente antes de su integración completa. Inicialmente se ensayaron las entradas y salidas digitales básicas, incluyendo indicadores LED y pulsadores. Posteriormente se incorporaron y verificaron periféricos de mayor complejidad, tales como el display OLED, la interfaz de comunicación RS-485 junto con su correspondiente adaptación de niveles, el módulo de comunicación CAN Bus, el reloj de tiempo real (RTC) y el resto de los dispositivos integrados. Este enfoque permitió aislar errores, simplificar el proceso de depuración y garantizar el correcto funcionamiento de cada bloque funcional previo a la validación del sistema completo.

### *C. Desarrollo de firmware y validación funcional*

En paralelo al proceso de ensamblado y validación de hardware, se desarrolló un conjunto de aplicaciones de prueba específicas para cada uno de los subsistemas integrados en la plataforma. Estas implementaciones permitieron verificar el correcto funcionamiento de los periféricos y, al mismo tiempo, sirvieron como base para la construcción de abstracciones de mayor nivel.

Para cada etapa de testeo se generaron archivos de firmware independientes, orientados a validar de manera aislada las funcionalidades de los distintos módulos, tales como entradas/salidas digitales, adquisición analógica, interfaces de comunicación y dispositivos periféricos. Este enfoque modular facilitó la detección de fallas, la validación progresiva del sistema y la reutilización de código en etapas posteriores del desarrollo.

A partir de estos desarrollos iniciales, se implementaron controladores (drivers) específicos para cada uno de los recursos de hardware de la plataforma, siguiendo criterios de

abstracción y reutilización de código. Dichos drivers fueron concebidos no sólo como herramientas de validación, sino también como componentes didácticos dentro de la asignatura Sistemas Digitales II, en la cual se emplea esta plataforma como soporte experimental.

En este contexto, los estudiantes disponen de los drivers desarrollados como capa de acceso al hardware, lo que les permite concentrarse en la resolución de aplicaciones de mayor nivel. De esta manera, se promueve un enfoque de desarrollo estructurado, alineado con prácticas profesionales en ingeniería de sistemas embebidos, facilitando la comprensión de la interacción entre hardware y software en sistemas reales.

### D. *La plataforma como recurso didáctico*

La plataforma actúa como un valioso recurso didáctico en Ingeniería Electrónica, promoviendo la participación activa de los estudiantes. Un aspecto clave fue la incorporación de un estudiante avanzado mediante una adscripción académica, quien —como coautor de este trabajo— llevó adelante una parte importante del diseño. Esto demuestra lo necesario de integrar la formación con desarrollos de ingeniería reales, consolidando competencias técnicas como diseño de hardware, selección de componentes y validación. Asimismo, su uso en las asignaturas ha motivado a otros alumnos a proponer mejoras, evidenciando que un sistema tangible desarrollado en el propio entorno actúa como un potente disparador de la iniciativa estudiantil.

## IV. Aplicaciones y ejemplo de uso

A continuación se presentan dos aplicaciones representativas desarrolladas sobre la plataforma, correspondientes a las asignaturas Sistemas Digitales II y Sistemas Embebidos Avanzados II.

### A. *Comunicación CAN Bus*

Uno de los ejemplos de aplicación (ver [3]) es la comunicación CAN Bus. Este protocolo se aborda en la asignatura Sistemas Digitales II de la carrera de Ingeniería Electrónica, donde se estudian protocolos de comunicación industrial con firmware desarrollado en bare metal en lenguaje C, utilizando máquinas de estado finitas.

La prueba consiste en la comunicación entre dos placas shield, cada una conectada a su respectiva placa base FRDM-MCXC444, configuradas como nodo productor y nodo consumidor. El bus CAN se configura a una velocidad 125 kbps en modo normal. El nodo productor transmite cada 100 mS una trama con identificador 0x01 y un byte de datos con valor 0x0A, mientras que el nodo consumidor recibe y verifica la trama. En las Figuras 5 y 6 se muestra la salida por terminal de ambos nodos, donde se observa la transmisión y recepción exitosa de los mensajes.

```
Mensaje transmitido - ID: 0x1, DLC: 1, Data[0]: 0xA
Mensaje transmitido - ID: 0x1, DLC: 1, Data[0]: 0xA
Mensaje transmitido - ID: 0x1, DLC: 1, Data[0]: 0xA
Mensaje transmitido - ID: 0x1, DLC: 1, Data[0]: 0xA
```

Figura 5: Salida por terminal del nodo productor

```
Mensaje recibido - ID: 0x1, DLC: 1, Data[0]: 0xA
Mensaje recibido - ID: 0x1, DLC: 1, Data[0]: 0xA
Mensaje recibido - ID: 0x1, DLC: 1, Data[0]: 0xA
Mensaje recibido - ID: 0x1, DLC: 1, Data[0]: 0xA
```

Figura 6: Salida por terminal del nodo consumidor

### B. *Lectura de LM35*

Se desarrolla un ejemplo (ver [4]) para la asignatura Sistemas Embebidos Avanzados II. La aplicación realiza la lectura de temperatura mediante el sensor LM35 integrado en la shield y transmite el valor medido por puerto serie a la PC. A diferencia del ejemplo anterior, esta implementación se realiza mediante el RTOS Zephyr [5], lo que permite abordar conceptos de planificación de tareas y temporización en tiempo real.

En la Figura 7 se muestra la salida por puerto serie, donde se observa el valor crudo del ADC, la tensión correspondiente y la temperatura calculada.

```
[0:00:82.050,000] <inf> MAIN: Sample A3: ADC=888,
Voltage=716.6 mV, Temp=27.35°C
[0:00:87.052,000] <inf> MAIN: [TIMER PERIODIC]
Lectura del ADC
```

Figura 7: Salida por puerto serie de la lectura del LM35

## V. Conclusión

En este trabajo se presentó el diseño, implementación y validación de una placa Shield para el kit de desarrollo FRDM-MCXC444. La plataforma demostró ser una herramienta versátil tanto para actividades académicas como para proyectos de aplicación en sistemas embebidos, permitiendo trabajar con periféricos industriales, protocolos de comunicación y distintos paradigmas de programación. Habiendo cumplido los objetivos propuestos, se plantea como trabajo futuro evaluar la conveniencia de desarrollar un kit completo con microcontrolador propio.

## VI. Agradecimientos

Ing. José Luis Simón: Ingeniero Electrónico (UNR). Actual Director del Departamento de Sistemas e Informática, FCEIA-UNR.

Ing. Ignacio Sosa: Ingeniero eléctrico con orientación electrónica (UNR). Ex Director del Departamento de Sistemas e Informática, FCEIA-UNR.

## Referencias

[1] NXP, “Frdm-mcxc444: Freedom development board for kinetis mcxc mcus,” En línea, 2024, disponible en: https://www.nxp.com/design/design-center/development-boards-and-designs/FRDM-MCXC444.

[2] *MCP2515 Stand-Alone CAN Controller with SPI Interface*, Microchip Technology Inc., 2023.

[3] DSI-FCEIA, “SHIELD-MCXC444-DSI: Hardware Repository,” En línea, 2024, disponible en: https://github.com/sd2-dsi-fceia/SHIELD-MCXC444-DSI.

[4] ——, “SEA2 Zephyr - Shield MCXC444 Firmware,” En línea, 2024, disponible en: https://github.com/sea2-dsi-fceia/SEA2_ZEPHYR/tree/main/Shield-MCXC444.

[5] Linux Foundation, “Zephyr project: A scalable real-time operating system (rtos),” En línea, 2024, disponible en:https://www.zephyrproject.org/.

# Diseño e implementación de una placa didáctica basada en RISC-V para una asignatura de grado

Esteban Daniel Volentini, Elías Ganem, Martín Gareca
*Departamento de Electricidad, Electrónica y Computación*
*Facultad de Ciencias Exactas y Tecnología*
San Miguel de Tucumán, Argentina
evolentini@herrera.unt.edu.ar, eliasgfac@gmail.com, mfgareca36@gmail.com

***Abstract*—Este trabajo presenta el diseño, desarrollo y evaluación de una placa didáctica basada en RISC-V para las actividades de laboratorio de la asignatura *Sistemas con microprocesadores y microcontroladores*. La placa se concibió para disponer de una plataforma propia de bajo costo que permita avanzar hacia la unificación del caso de estudio con la asignatura *Arquitectura de Computadoras*, la cual adoptó RISC-V en 2019. Se describe el proceso de selección del microcontrolador, diseño de hardware orientado a objetivos didácticos, prototipado iterativo y producción de una serie final para uso en laboratorio. La solución adoptada integra un microcontrolador GD32VF103RBT6 con una placa RP2040 Zero emulando un CMSIS-DAP, y permitió resolver limitaciones detectadas en una primera aproximación basada en módulos comerciales. Como resultado, se fabricaron 50 unidades funcionales empleadas en el primer dictado de la asignatura con esta plataforma. Al final se analizan los resultados de esta primera experiencia docente.**

***Index Terms*—RISC-V, sistemas embebidos, educación en ingeniería, microcontroladores, laboratorio didáctico, hardware abierto.**

## I. Introducción

Este artículo presenta el diseño, desarrollo y evaluación de una placa didáctica basada en RISC-V destinada a las actividades de laboratorio de una asignatura que enseña el funcionamiento y uso de microprocesadores de baja capacidad de cálculo, utilizados habitualmente en sistemas embebidos, dentro de un enfoque de aprendizaje activo centrado en laboratorio [1], [2].

El desarrollo de las placas se realizó en forma conjunta entre el Laboratorio de Microprocesadores y el Laboratorio Docente de Electrónica. Si bien las placas se utilizan en una asignatura de la carrera de Ingeniería en Computación, el proyecto fue ejecutado por dos estudiantes de Ingeniería Electrónica, bajo la supervisión del docente responsable de la asignatura usuaria de las placas y en colaboración con docentes de la asignatura Electrónica I.

### A. Antecedentes

La carrera de Ingeniería en Computación de la Universidad Nacional de Tucumán estructura estos contenidos en dos asignaturas de una misma cátedra:

- *Sistemas con microprocesadores y microcontroladores*: orientada al funcionamiento de una computadora básica y su aplicación en sistemas embebidos. El caso de estudio evolucionó de CPU08 a Cortex-M4, incorporando además placas didácticas propias para laboratorio [3], [4]. Esta asignatura trabaja programación en ensamblador, camino de datos simple e interrupciones/excepciones.
- *Arquitectura de Computadoras*: enfocada en técnicas avanzadas de rendimiento (segmentación, superescalaridad, caché y memoria virtual). En 2019 migró su caso de estudio de MIPS a RISC-V, en línea con la bibliografía principal, consolidando a RISC-V como referencia académica de la cátedra.

Tras la adopción de RISC-V en *Arquitectura de Computadoras* (2019), la cátedra se propuso unificar el caso de estudio en ambas asignaturas. En ese momento, la escasa oferta de microcontroladores RISC-V con soporte de laboratorio impidió concretar el cambio en la asignatura de sistemas embebidos. En particular, se buscaba mantener prácticas sin sistema operativo y con acceso de bajo nivel al hardware, requisito clave para la secuencia didáctica de la materia.

### B. Alcances

El objetivo de este artículo es presentar el diseño de una nueva placa didáctica, basada en un microcontrolador con arquitectura RISC-V, que permitió unificar los casos de estudio de las dos asignaturas impartidas por la cátedra.

Se analizan las diversas opciones evaluadas y los criterios técnicos y económicos para la elección del microcontrolador, los objetivos pedagógicos que fundamentaron la selección de periféricos, el proceso de desarrollo y fabricación del hardware, y los primeros resultados del dictado con esta plataforma.

## II. Desarrollo

El desarrollo se estructuró en varias etapas: selección del microcontrolador, diseño electrónico de la placa, prototipado iterativo, producción de la serie final y aplicación en el aula.

### A. Elección del microcontrolador RISC-V

Se evaluaron distintas alternativas buscando bajo costo (menos de U$S 5 en mercado minorista), programación en ensamblador y operación sin sistema operativo. La opción SiFive E300 se descartó por costo y volumen mínimo de compra (la oferta más accesible implicaba una inversión del orden de U$S 20.000 para 1.000 unidades). También se evaluó ESP-32C con pruebas de laboratorio; aunque se logró compilar sin sistema operativo en escenarios acotados, la

dependencia práctica de FreeRTOS para el ecosistema oficial y la disponibilidad limitada de entradas/salidas lo hicieron poco adecuado para la placa didáctica.

Finalmente se evaluaron dos clones del STM32F103 que reemplazan el procesador ARM Cortex-M3 por un procesador RISC-V de 32 bits. La primera opción fue el CH32V103 de la empresa Nanjing Qinheng Microelectronics Co [5], que implementa un procesador propio denominado Qing V3A. La otra opción fue el GD32VF103 de la compañía GigaDevice [6], que implementa un procesador Bumblebee, una variante del procesador N200 de la empresa Nuclei Systems [7]. De estos dispositivos se eligió la familia GD32VF103 principalmente por la interfaz de depuración. El CH32V103 utiliza una interfaz de depuración propietaria que funciona únicamente con un POD propietario y con una versión modificada de OpenOCD, mientras que el GD32VF103 utiliza una interfaz JTAG estándar.

### *B. Diseño de la placa didáctica*

En esta etapa del desarrollo, la primera actividad consistió en la definición de los objetivos didácticos de las actividades de laboratorio. En el diseño de la placa se distinguieron dos conjuntos. En primer lugar, los objetivos que la asignatura ya tenía fijados con anterioridad y que se deseaba mantener:

- Utilizar entradas y salidas digitales.
- Gestionar una interfaz de entrada mediante pulsadores directamente conectados a un GPIO (no multiplexado).
- Generar señales PWM de salida.
- Implementar multiplexado de salida en pantallas de siete segmentos.

En segundo lugar, y con menor prioridad, un conjunto de objetivos adicionales pensado para el uso de la placa en proyectos educativos o en otras asignaturas, que motivó la inclusión de periféricos complementarios como sensores analógicos, teclado matricial, buses de expansión y un encoder rotativo.

En base a ambos conjuntos de objetivos se decidió que la placa didáctica debía implementar los siguientes dispositivos de entrada/salida:

- Una pantalla de cuatro dígitos de siete segmentos multiplexados.
- Cuatro teclas de función ubicadas debajo de cada uno de los dígitos de la pantalla.
- Dos teclas denominadas “Aceptar” y “Cancelar”.
- Un teclado matricial de 16 teclas multiplexadas.
- Un LED RGB controlado por tres terminales con capacidades PWM.
- Un parlante conectado a un terminal con capacidades PWM.
- Un sensor de temperatura LM35 conectado a una entrada analógica.
- Un sensor de luz LDR conectado a una entrada analógica.
- Un puerto de expansión I2C con terminales de alimentación.
- Un puerto de expansión SPI con terminales de alimentación.
- Un POD JTAG para programación del microcontrolador.
- Un puerto serial conectado al POD que permita disponer de una consola serial utilizando una conexión USB única.

En una primera aproximación se eligió desarrollar la placa de periféricos como una ampliación de una placa de evaluación Longan Nano, de la empresa Sipeed [8], que utiliza un microcontrolador GD32VF103CBT6 [6], compatible *pin a pin* con el STM32F103CBT6 y de factor de forma similar a las placas Blue Pill. Por esta razón se comenzó el diseño pensando en una placa de periféricos compatible con ambas placas de evaluación.

Por razones de ruteo del PCB se descartó rápidamente la compatibilidad con ambas placas; sin embargo, se planteó como alternativa un diseño que permitiera cambiar fácilmente la placa del microcontrolador sin necesidad de alterar todo el diseño. En la Fig. 1 se puede observar que esta característica se mantuvo en el diseño final como una línea de corte que permite dividir la placa en dos secciones: una del microcontrolador y otra de periféricos. Al dividirlas se puede soldar un conector de 2x21 vías, de paso 100 *mils*, en cada una de las placas. Esto permite usarlas en otros proyectos y, eventualmente, volver a conectarlas entre sí.

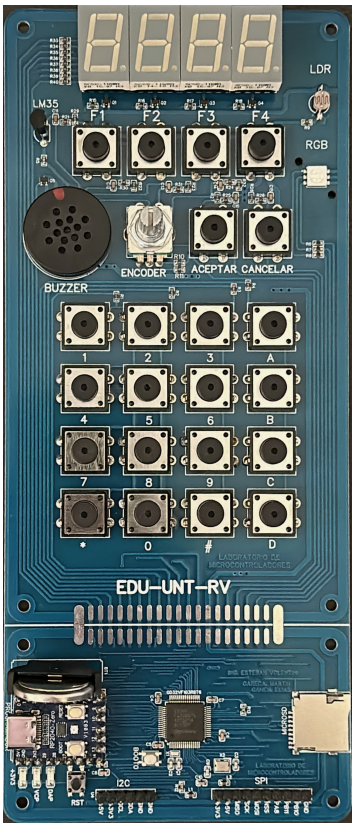



Fig. 1. Placa desarrollada con separación entre módulo de microcontrolador y módulo de periféricos didácticos.

Esta versión no implementaba POD de programación ya que los terminales para esta función en la placa Longan Nano son laterales. Para esta función se seleccionó una RV-Debugger de la empresa Sipeed [9], que utiliza un BL702. Por el tamaño de los componentes, este prototipo se pudo resolver con trazos de 20 *mils*, por lo que se fabricó localmente.

### C. Prototipado

Para la validación del primer prototipo se implementó un software de demostración que utiliza todos los recursos integrados en la placa, es decir, la pantalla de siete segmentos, las seis teclas con conexión exclusiva, el teclado matricial, el sensor de luz, el sensor de temperatura, el LED RGB y el parlante. Estas pruebas revelaron una serie de inconvenientes con el diseño elegido:

- Limitaciones en la cantidad de terminales disponibles: en la placa Longan Nano quedan disponibles 28 terminales para conexión de dispositivos, que resultaban insuficientes para conectar todos los dispositivos que se querían integrar. Para resolver este problema se decidió utilizar los mismos terminales para el teclado y la pantalla de siete segmentos. Agregando diodos en las teclas es posible intercalar multiplexados de entrada y de salida en las mismas líneas digitales. Sin embargo, algunas resistencias de pull-up integradas en la placa Longan Nano interferían con la configuración, por lo que una de las columnas del teclado quedaba inutilizada.
- Problemas con el POD de programación: cuando se decidió utilizar un POD externo se analizó la robustez del conjunto, sobre todo teniendo en cuenta que las placas se prestan a los alumnos durante todo el cursado. Se eligió el RV-Debugger porque la salida es lateral, igual que la Longan Nano, y entonces se podían vincular ambas placas con un arnés de cables a medida razonablemente robusto. Sin embargo, el programador no funcionaba correctamente. Una actualización de firmware podía resolver el problema; sin embargo, no se pudo completar la actualización por problemas con la herramienta que transfiere la imagen usando el *bootloader* del BL702.

Ante esta situación se optó por eliminar la placa Longan Nano e integrar el microcontrolador directamente en la placa base. Esta decisión permitió la libre elección del microcontrolador a utilizar, el cual se cambió por un GD32VF103RBT6 [6] que implementa las mismas cantidades de memoria pero utiliza un encapsulado LQFP64, lo que permitía disponer de más terminales para conectar dispositivos. También se decidió implementar un POD en la placa base, para lo cual se eligió una placa RP2040 Zero de la empresa Waveshare, a la que se le instaló el firmware FreeDAP [10] para emular un CMSIS-DAP con soporte para Serial Wire Debug y JTAG.

Estas decisiones aumentaron el nivel de tecnología necesaria para fabricar el PCB, ya que ahora se requerían trazos tan delgados como 8 *mils* de ancho. Por esta razón se encargó la fabricación del siguiente prototipo a la empresa china JLCPCB. Para optimizar costos y tiempo en esta versión se concentraron la mayoría de los componentes de montaje superficial en una misma cara, y se encargó la provisión y soldadura al mismo fabricante. De esta versión se fabricaron cinco unidades: dos se destinaron a validaciones, mientras que las otras tres se utilizaron para comenzar a generar el material de apoyo necesario para el dictado de la asignatura.

En un repositorio propio [11] se publicó la estructura de compilación y enlazado, junto con un instructivo de configuración del entorno (WSL, Visual Studio Code y usbipd), un script de instalación del GNU RISC-V Embedded Toolchain y OpenOCD, el esquemático de la placa y ejemplos de código. Dado que uno de los objetivos de las prácticas es explicar el proceso de *reset* del procesador junto con el vector de interrupciones y excepciones, el repositorio permite compilar proyectos en ensamblador y en C, con *startup* propio o bibliotecas del fabricante. Los ejemplos sirven como punto de partida para los ejercicios de laboratorio. La depuración se realiza con OpenOCD sobre la interfaz CMSIS-DAP integrada en la placa.

### D. Producción

Para validar esta segunda versión se modificó y extendió el software utilizado en la validación del primer prototipo. Al cambiar el microcontrolador por una versión con más terminales se reasignaron varias de las conexiones de los periféricos. Por lo tanto, la primera tarea fue corregir los recursos asociados a cada dispositivo en el software de prueba y verificar el funcionamiento de todos los dispositivos didácticos. A continuación se incluyeron validaciones para los dispositivos que originalmente estaban integrados en la placa Longan Nano y ahora forman parte de la placa base: el reloj de tiempo real con la batería CR2032, el cristal externo para el sistema de reloj, el zócalo para tarjetas TransFlash y los puertos de expansión I2C y SPI.

Todos los dispositivos del segundo prototipo funcionaron correctamente. El único error detectado fue la conexión de la UART del microcontrolador con el puerto serie auxiliar del POD de programación. Esta conexión se corrigió en uno de los prototipos para tener una versión totalmente funcional antes de la fabricación del lote final. A partir de las pruebas también se hicieron algunos cambios, como alejar el botón de *reset* de un conector que dificultaba el acceso.

Para la fabricación de la versión final se decidió mover todos los componentes de montaje superficial a la cara superior de la placa y encargar su montaje junto con la fabricación. De esta forma solo se montaron en el laboratorio los componentes más grandes y pesados: los *displays* de siete segmentos, los botones y el POD de programación. El encoder rotativo no se montó porque no se utiliza en la asignatura. Tampoco se montaron los zócalos de las pilas CR2032 para mantener la altura de las placas en torno a 1 cm y disminuir el volumen del despacho.

Con estos cambios, y después de una revisión exhaustiva del equipo de trabajo formado por dos alumnos y de una segunda revisión con el docente a cargo del proyecto, se encargó la fabricación de 50 unidades de esta versión final, con un costo total de U$S 640 incluyendo el flete (aproximadamente U$S 12,80 por unidad). Este valor corresponde únicamente al PCB con sus componentes de montaje superficial; no incluye los elementos montados en el laboratorio, a saber: los *displays* de siete segmentos (U$S 6), los botones (U$S 12) y el POD de programación (U$S 7). Con estos componentes adicionales, el costo aproximado por placa completamente funcional asciende a unos U$S 40. El proyecto comenzó en octubre de 2024, con

recepción de prototipos en marzo y julio de 2025, y de la versión final el 16 de octubre de 2025.

### *E. Aplicación en el aula*

Las placas se utilizaron por primera vez en clase el 20 de octubre de 2025, durante el segundo cuatrimestre de ese año. Los cambios en el contenido de la asignatura se implementaron desde el inicio de ese cuatrimestre y se planificaron desde el momento en que se comenzó la construcción de la placa didáctica. Estos cambios incluyeron la reformulación de algunos objetivos de la asignatura, la rearticulación completa de los contenidos y la adaptación de todo el contenido teórico y práctico a la nueva arquitectura.

En el segundo cuatrimestre de 2023 (2C-2023) cursaron 21 estudiantes: 3 abandonaron la asignatura, 4 desaprobaron y 14 aprobaron, con un promedio de 7 entre los aprobados. En 2C-2024 cursaron 33 estudiantes: 5 abandonaron y los 28 restantes aprobaron, con un promedio de 7,03. En 2C-2025 cursaron 23 estudiantes: 5 abandonaron, 2 desaprobaron y 16 aprobaron, con un promedio de 8 entre los aprobados. Cada estudiante utilizó una placa en préstamo durante el cursado. No se usaron repuestos ni se registraron fallas de hardware; adicionalmente, se utilizaron placas específicas para docencia y para pruebas exhaustivas de los periféricos integrados. En cuanto a la carga horaria, que se mantuvo sin modificaciones respecto a cursadas anteriores, el dictado incluyó 60 horas de teoría, 12 horas de prácticos y 18 horas de laboratorio presencial, complementadas con aproximadamente 30 a 40 horas de trabajo autónomo de laboratorio por parte de los estudiantes.

Es importante señalar que la comparación entre cohortes incluye cambios simultáneos a la migración de plataforma. Entre ellos, en 2C-2025 se eliminaron los exámenes parciales y se adoptó una evaluación totalmente distribuida en prácticos y laboratorios; este cambio metodológico explica en parte las diferencias de rendimiento observadas y condiciona la interpretación de las tasas de aprobación. Además del cambio de arquitectura, la asignatura pasó de un enfoque centrado en lenguaje ensamblador a un enfoque mixto, donde se mantiene una parte en ensamblador y se incorpora la transición a C—lenguaje que los estudiantes ya estudian en el primer año—para el desarrollo de dispositivos más avanzados, como los temporizadores. Por este motivo, las diferencias observadas no pueden atribuirse de manera directa a un único factor.

## III. Conclusiones

El diseño, desarrollo y evaluación de una placa didáctica basada en RISC-V permitió avanzar hacia un objetivo estratégico de la cátedra: unificar la base conceptual y tecnológica de *Sistemas con microprocesadores y microcontroladores* con *Arquitectura de Computadoras*. Técnicamente, la adopción de RISC-V resultó viable combinando criterios pedagógicos y económicos: tras iteraciones desde módulos comerciales, la placa con GD32VF103RBT6 y POD CMSIS-DAP integrado permitió fabricar 50 unidades usadas sin fallas en laboratorio.

En relación con los resultados académicos, la tasa de aprobación sobre inscriptos en 2C-2025 (69,6%) fue inferior a la de 2C-2024 (84,8%), aunque similar a la de 2C-2023 (66,7%). No obstante, el promedio entre aprobados ascendió a 8 en 2C-2025, frente a 7,03 en 2C-2024 y 7 en 2C-2023. Dado el tamaño reducido de las cohortes, la eliminación de exámenes parciales en favor de una evaluación distribuida en prácticos y laboratorios, y la coexistencia de cambios de contenidos y de plataforma, estos resultados no permiten extraer conclusiones definitivas sobre el impacto de los cambios en el cursado. Aun así, es razonable pensar que, al menos, no afectaron de forma negativa el dictado de la asignatura.

La ausencia de fallas de hardware durante el cursado confirma el éxito técnico-operativo de la plataforma y su viabilidad en la dinámica de laboratorio de la asignatura. La evaluación pedagógica de la experiencia, en cambio, permanece preliminar.

Como línea de trabajo futuro se plantean placas adaptadoras para reutilizar el módulo de periféricos con otras plataformas (p. ej. Blue Pill o ESP32). El repositorio publica el esquemático y la guía de instalación para facilitar la replicación por otras cátedras. Como limitación, la implementación coincidió con cambios curriculares y metodológicos, por lo que aún no se dispone de una evaluación causal del impacto pedagógico atribuible exclusivamente a la migración de plataforma.

## Agradecimientos

Este trabajo se realizó en la Facultad de Ciencias Exactas y Tecnología, en el marco del Proyecto de Investigación “Sistemas Embebidos, Comunicaciones y Aplicaciones en Internet de las Cosas” (PIUNT E743).

## References

[1] M. Prince, “Does active learning work? a review of the research,” *Journal of Engineering Education*, vol. 93, no. 3, pp. 223–231, 2004.

[2] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, “Active learning increases student performance in science, engineering, and mathematics,” *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8410–8415, 2014.

[3] E. D. C. Esteban D. Volentini, Guillermo Rosemberg, “Nuevos paradigmas en trabajos de laboratorio,” in *2014 Fifth Argentine Symposium and Conference on Embedded Systems SASE/CASE 2014 (CASE)*, 2014, pp. 54–57.

[4] G. R. Esteban D. Volentini, Eduardo D. Cohen, “Redesign of laboratory works : A proposal for integration,” in *2018 IEEE Biennial Congress of Argentina (ARGENCON)*, 2018, pp. 1–6.

[5] N. Q. M. Co., “Ch32v103 product documentation,” https://www.wch-ic.com/products/CH32V103.html, 2025, [Online; accessed May 7, 2026].

[6] G. S. Inc., “Gd32vf103 series risc-v 32-bit mcu datasheet and user manual,” https://www.gigadevice.com/product/mcu/main-stream-mcus/gd32vf103-series/, 2025, [Online; accessed May 7, 2026].

[7] N. S. Technology, “Nuclei bumblebee core documentation,” https://doc.nucleisys.com/nuclei_spec/isa/core.html, 2025, [Online; accessed May 7, 2026].

[8] Sipeed, “Longan nano documentation,” https://wiki.sipeed.com/hardware/en/longan/Longan_Nano/Longan_nano.html, 2025, [Online; accessed May 7, 2026].

[9] ——, “Rv-debugger documentation,” https://wiki.sipeed.com/hardware/en/rv-debugger/rv-debugger.html, 2025, [Online; accessed May 7, 2026].

[10] A. Taradov, “Free dap,” https://github.com/ataradov/free-dap, 2025, gitHub repository, [Online; accessed May 7, 2026].

[11] L. de Microprocesadores, “micro-muju: Entorno de trabajo para la asignatura de sistemas con microprocesadores unt (versión risc-v),” https://github.com/labmicro/micro-muju, 2025, gitHub repository, [Online; accessed June 22, 2026].

# Estimación de Posición, Orientación y Velocidad de Bajo Costo, para un Vehículo Autónomo

1st Ibrahim Jaime
*AC&ES-RG*
*UTN Facultad Regional San Francisco*
*San Francisco, Argentina*
ibrahimjaime@gmail.com

2nd Marco L. Miretti
*AC&ES-RG*
*UTN Facultad Regional San Francisco*
*San Francisco, Argentina*
marco.miretti@gmail.com

3rd Emanuel Bernardi
*AC&ES-RG*
*UTN Facultad Regional San Francisco*
*San Francisco, Argentina*
bernardiemanuel@gmail.com

***Abstract*—Este trabajo aborda el diseño y desarrollo del sistema de estimación de posición, orientación y velocidad, de un vehículo autónomo. El sistema integra un Sistema Global de Navegación Satelital (GNSS), una Unidad de Medición Inercial (IMU) y un magnetómetro como sensores principales, combinando sus mediciones para proporcionar información precisa y confiable. Se aplicaron técnicas de calibración y filtrado para reducir los efectos del ruido y las perturbaciones externas, particularmente en los sensores inerciales y magnéticos. Para fusionar los datos provenientes de los sensores inerciales y del magnetómetro se emplearon filtros complementarios. Estos permiten combinar las fortalezas de cada sensor, compensando sus limitaciones individuales. Este enfoque mejora la estimación de la orientación, garantizando mayor robustez en escenarios dinámicos. El sistema logró un tiempo de muestreo de 10 ms, garantizando una tasa de actualización en tiempo real compatible con la dinámica de un prototipo terrestre con velocidades de hasta 25 km/h. Los resultados obtenidos destacan la viabilidad de integrar sensores de bajo costo y técnicas avanzadas de procesamiento de señales para alcanzar mediciones fiables.**

***Index Terms*—GNSS, IMU, magnetómetro, filtro complementario, calibración de sensores, fusion de sensores.**

## I. Introduction

En la actualidad, la movilidad autónoma se ha consolidado como un área de gran interés en investigación y desarrollo, siendo aplicada exitosamente en vehículos terrestres, marítimos y aéreos. Su implementación presenta ventajas económicas, ambientales y de seguridad en la movilidad. Partiendo de dicho contexto y teniendo en cuenta que la agricultura es una de las principales actividades económicas de nuestro país, el grupo de investigación AC&ES-RG (Applied Control and Embedded Systems - Research Group) de la Universidad Tecnológica Nacional, Facultad Regional San Francisco, inició el desarrollo de un vehículo autónomo capaz de realizar diversas actividades en el ámbito de la agricultura. Una de las partes fundamentales del sistema de navegación de dicho vehículo es la adquisición de datos cinemáticos, posición, orientación y velocidad. En consecuencia, mediante el presente estudio se abordó el desarrollo e implementación de un sistema de determinación de posición y actitud del vehículo. El mismo contó con una Unidad de Medición Inercial (IMU), un magnetómetro y un Sistema Global de Navegación Satelital (GNSS).

El sistema desarrollado, proporcionó estimaciones en un período de muestreo uniforme, y se ha priorizado reducir al máximo dicho tiempo para posibilitar el correcto funcionamiento del sistema de control. Consecuentemente, se ha realizado la calibración y adaptación de señales de cada uno de los sensores, y luego se combinaron con el fin de lograr un incremento en la fiabilidad. Así mismo, la combinación permitió reducir el tiempo de muestreo por debajo del tiempo de operación del GNSS. Esto solucionó una limitación del sistema, ya que el periodo de este último es considerablemente mayor en comparación con los demás sensores. Para determinar la técnica adecuada de fusión de sensores se tuvieron en cuenta las características de los mismos, la capacidad de procesamiento del microcontrolador seleccionado y los resultados de pruebas experimentales.

El presente trabajo corresponde al desarrollo de un Proyecto Final de la Carrera de Ingeniería Electrónica de la Universidad Tecnológica Nacional, Facultad Regional San Francisco.

### A. Estado del Arte

Existen numerosos estudios relacionados que involucran a sensores inerciales, los cuales abordan sus características de ruido, las desventajas de cada tipo de sensor y las técnicas para realizar una calibración adecuada [1]. Además, existen diversas publicaciones acerca de la combinación de sensores para mejorar las estimaciones. Generalmente, la fusión de sensores se realiza empleando un filtro de Kalman [2] [3] [4] [5], o un filtro complementario. En [6] se presenta una descripción y comparación de ambos, mientras que en [7] se realiza una fusión similar comparando los métodos: filtro extendido de Kalman (EKF) y filtro extendido de información (EIF), con el fin de mejorar las estimaciones de la dinámica de un vehículo. Otra investigación que emplea filtros de Kalman extendidos se presenta en [8] donde se implementa un sistema considerando aceleraciones de media cero con respecto a cualquier marco de referencia, lo cuál mejora las estimaciones en terrenos irregulares. Por otro lado, [9] describe la implementación de filtros complementarios para fusionar los datos de un giróscopo y un acelerómetro, con el fin de mantener la estabilidad de una plataforma. Finalmente, en [10] se valida la fiabilidad de la fusión de datos mediante

filtros complementarios en un contexto dinámico, similar al del presente trabajo.

## II. Metodología

### A. Componentes del sistema

Para interactuar e integrar los diversos dispositivos, se optó por utilizar el microcontrolador de 32 bits 'RP2040'.

La IMU seleccionada fue la 'MPU-6050', la cual cuenta con un giróscopo de 3 ejes y un acelerómetro de 3 ejes. La escala seleccionada para el giróscopo fue de $\pm 250$ °/s, mientras que para el acelerómetro se utilizó un rango de $\pm 4$ g. Su filtro pasa bajos fue configurado con un ancho de banda de $5,05$ Hz y su frecuencia de muestreo se configuró en 1 kHz.

El magnetómetro utilizado fue el 'QMC5883l', capaz de brindar mediciones independientes en tres ejes perpendiculares. El rango de medición empleado fue de $\pm 2$ G mientras que la frecuencia de muestreo seleccionada fue de 200 Hz. Además, se empleó una tasa de sobre-muestreo con lo que se redujo al máximo el ancho de banda del filtro pasa-bajos del sensor.

El módulo GNSS utilizado fue el NEO-M8N, que alcanza una exactitud de hasta $\pm 2,5$ m en condiciones óptimas. Luego de la realización de pruebas experimentales se decidió emplear solo la recepción del sistema de navegación GPS. Debido a que no existen problemas de conectividad en el área de utilización y además es el único que permite una frecuencia de muestreo de 10 Hz.

Mientras que unidades de navegación inercial de grado militar o industrial poseen costos que oscilan entre cientos y miles de dólares, los sensores comerciales empleados en este trabajo representan una fracción mínima de dicha inversión.

### B. Calibración de componentes

Para calibrar los acelerómetros, giróscopos y magnetómetros se realizó un proceso similar al utilizado por [1].

*1) Acelerómetro:* La calibración del acelerómetro consistió en adquirir mediciones mientras el dispositivo se giraba lentamente 360 ° sobre cada eje. Idealmente, las mediciones deben ubicarse dentro del rango de $\pm 1$ g y formar un círculo unitario centrado en el origen al graficar dos ejes ('XY', 'XZ' o 'YZ'). Los desplazamientos y errores de escala se corrigieron mediante la Ecuación 1.

$$A_{cal} = (A_{medida} - A_{desp}) * A_{escala} \tag{1}$$

*2) Giróscopo:* El proceso de calibración de este dispositivo consiste en adquirir una gran cantidad de datos en estado de reposo. Luego, las mediciones se promedian para obtener el valor de desplazamiento presente en las mediciones. La Ecuación 2 describe la calibración.

$$\omega_{cal} = \omega_{medida} - \omega_{desp} \tag{2}$$

*3) Magnetómetro:* Las distorsiones que afectan a este dispositivo se dividen en 'hard-iron' producidas por elementos ferro-magnéticos que se encuentran magnetizados de forma permanente y 'soft-iron' generadas por objetos de alta permeabilidad magnética.

Para identificar distorsiones magnéticas, el dispositivo se giró 360 ° sobre cada eje y se graficaron pares de mediciones ('XY', 'XZ' o 'YZ'). Idealmente, los datos deben formar un círculo unitario centrado en el origen. El efecto 'hard-iron' se manifiesta como un desplazamiento, corregido mediante la Ecuación 3:

$$M_{cal} = M_{medida} - M_{desp} \tag{3}$$

Por otra parte, el efecto 'soft-iron' produciría una deformación elíptica en las gráficas obtenidas. Sin embargo, dicho efecto no resultó significativo en las mediciones realizadas, por lo que no fue necesario aplicar correcciones adicionales.

### C. Determinación de magnitudes

Las ecuaciones empleadas para calcular los ángulos de cabeceo y balanceo a través de las mediciones del acelerómetro ($A_X$, $A_Y$ y $A_Z$) son:

$$\text{cabeceo} = \arctan\left(-\frac{A_X}{\sqrt{A_Y^2 + A_Z^2}}\right) \cdot \frac{180}{\pi} \tag{4}$$

$$\text{balanceo} = \arctan\left(-\frac{A_Y}{\sqrt{A_X^2 + A_Z^2}}\right) \cdot \frac{180}{\pi} \tag{5}$$

Por su parte, el giróscopo mide velocidad angular ($G_X$, $G_Y$ y $G_Z$) por lo que se debe integrar dicho valor para obtener la orientación angular. Para realizar una integración en tiempo discreto se empleó la Ecuación 6.

$$\text{ángulo}_n = \text{ángulo}_{n-1}^{G} + G_{eje} \cdot \Delta t \tag{6}$$

Siendo $\Delta t$ el intervalo de muestreo.

Para el cálculo del ángulo azimut se emplearon las mediciones del magnetómetro en la Ecuación 7.

$$\text{azimut} = \arctan\left(\frac{MAG_X}{MAG_Y}\right) \cdot \frac{180}{\pi} \tag{7}$$

Con el objetivo de calcular la velocidad del vehículo, se empleó la Ecuación 8 a partir de las mediciones obtenidas mediante el sistema GNSS. El procesamiento de las coordenadas y la determinación de variables cinemáticas se realizó siguiendo fundamentos de procesamiento GNSS descritos en [12]. Para ello, las coordenadas de latitud y longitud se transformaron de grados a kilómetros empleando una aproximación esférica de la Tierra.

$$v = \frac{d_{pos}}{\Delta t} \tag{8}$$

Donde $d_{pos}$ es la distancia entre dos posiciones consecutivas y $\Delta t$ el intervalo de tiempo entre las mediciones.

Si bien el tiempo de muestreo del GNSS es variable, se mantuvo en cada período de muestreo el último valor medido, alcanzando un tiempo de muestreo menor y uniforme. Pese a tener una componente de error asociado, esto es de gran utilidad a la hora de realizar un controlador apropiado para el vehículo.

### *D. Fusión de sensores*

Para obtener estimaciones precisas en sistemas de navegación, se asegura la medición de cada variable con dos dispositivos, los cuales son posteriormente fusionados para lograr una mayor precisión.

El análisis experimental de los sensores permitió determinar que el filtro complementario proporciona la precisión requerida con un bajo costo computacional.

El filtro complementario fusiona señales con distintas características de ruido mediante filtros pasa bajos y pasa altos, combinando las componentes de baja frecuencia de una señal con las de alta frecuencia de otra. De este modo, se obtiene una estimación más estable y precisa que utilizando cada sensor por separado.

Un ejemplo de implementación del método se presenta en [10] donde se analiza cómo la fusión de datos de diferentes sensores mediante filtros complementarios mejora la exactitud para la estimación de la orientación de un cuadricóptero en tiempo real. De manera similar, [9] aborda el uso de filtros complementarios para equilibrar el ruido proveniente de distintos sensores en sistemas de navegación.

La Ecuación 9 describe la implementación de un filtro complementario con el fin de fusionar las mediciones de un giróscopo y un acelerómetro.

$$\theta_k = \alpha(\theta_{k-1} + \theta_{\text{gir}}\Delta t) + (1-\alpha)\theta_{\text{acel}} \tag{9}$$

Donde el término $\alpha(\theta_{k-1} + \theta_{\text{gir}}\Delta t)$ se corresponde con el filtro pasa altos, mientras que el término $(1-\alpha)\theta_{\text{acel}}$ corresponde al pasa bajos. La constante $\alpha$ es el coeficiente de ponderación (de dominio entre 0 y 1) cuyo objetivo es ponderar cuánto contribuye cada fuente de señal al resultado final.

En el presente trabajo se emplearon tres filtros complementarios. Dos de ellos fueron utilizados para fusionar las mediciones del acelerómetro y el giróscopo en la estimación de los ángulos de cabeceo y balanceo, mientras que el tercero se utilizó para fusionar los datos del giróscopo y el magnetómetro en la estimación del ángulo azimut.

## III. Resultados Experimentales

En la tabla I se muestran los valores empleados para la calibración de los componentes del sistema.

TABLE I: Valores de calibración

| Sensor | Eje X | Eje Y | Eje Z |
|---|---|---|---|
| Acelerómetro | 0 g | 0 g | 0 g |
| Giróscopo | -1,6022 °/s | 0,2899 °/s | 1,8464 °/s |
| Magnetómetro | 0,037 G | -0,126 G | N/A |

Mediante la secuencia de calibración del acelerómetro se determinó innecesaria la compensación o la escala de las mediciones. Por su parte, la adquisición de mediciones del giróscopo se efectuó durante 90 minutos. Mientras que mediante la calibración del magnetómetro se descartó la presencia del efecto 'soft-iron'. Los avances logrados con el prototipo hasta dicha instancia fueron presentados y publicados en [11].

Con el objetivo de validar el funcionamiento del sistema y optimizar su desempeño, se llevaron a cabo una serie de pruebas exhaustivas tanto en condiciones de reposo como de movimiento. En las Figuras 1, 2 y 3 se presentan los resultados experimentales.

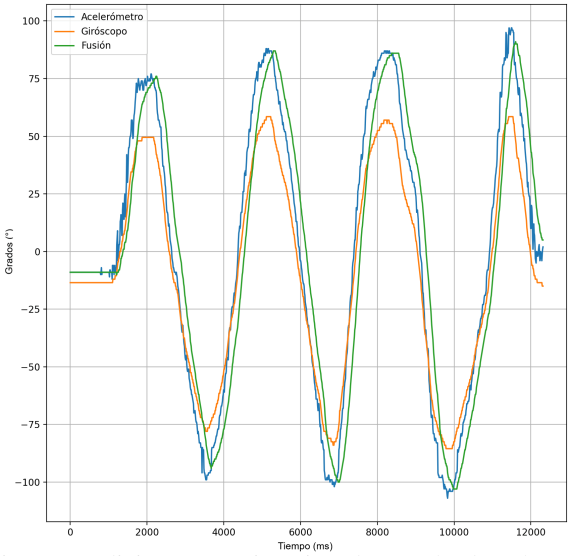



Fig. 1: Mediciones y estimación de ángulo de cabeceo

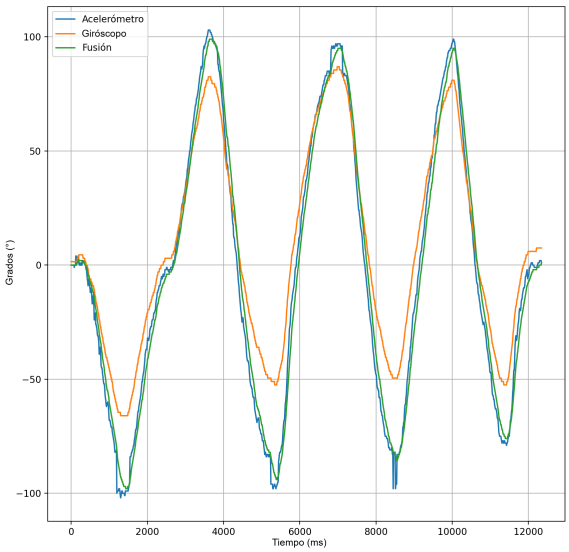



Fig. 2: Mediciones y estimación de ángulo de balanceo

Mediante la realización de pruebas se determinaron las constantes de los filtros complementarios ($\alpha$), para cada ángulo: cabeceo $= 0,92$, balanceo: $= 0,92$ y azimut: $= 0,5$.

En lo que respecta a la estimación de posición y velocidad, se realizaron pruebas dinámicas utilizando un automóvil como plataforma de validación. Para evaluar la precisión de las

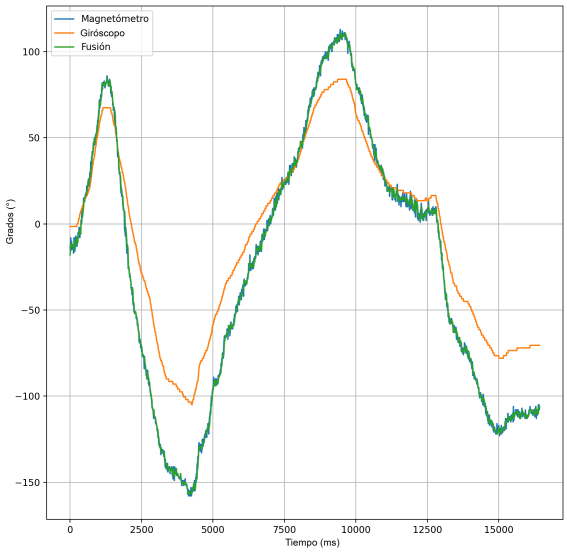



Fig. 3: Mediciones y estimación de ángulo azimut

mediciones de posición, los datos recolectados se proyectaron sobre un mapa (Figura 4), permitiendo una comparación visual con las trayectorias reales. El perfil de velocidad estimado fue evaluado cualitativamente para verificar su coherencia respecto a las aceleraciones y detenciones efectuadas durante el trayecto.

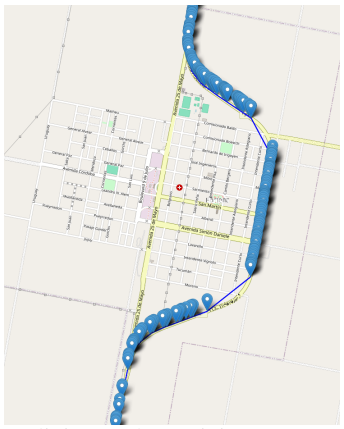


Fig. 4: Mediciones de posición en movimiento.

La fusión de velocidad y posición entre el GNSS y el acelerómetro fue descartada debido a la imposibilidad de medir con precisión la aceleración en el plano horizontal. Esto se debe a que incluso configurando el dispositivo en su máxima sensibilidad, las aceleraciones generadas por la conducción normal de un automóvil resultan prácticamente imperceptibles en comparación con las aceleraciones causadas por las irregularidades del terreno. Como resultado, el cálculo de la velocidad mediante el acelerómetro se tornó inviable.

## IV. Conclusiones

En el presente trabajo se logró calibrar los sensores de una IMU, alcanzando un nivel de precisión apropiado para su integración en la estimación de orientación. Se corrigió el efecto 'hard-iron' presente en el magnetómetro y se compensó el desplazamiento del giróscopo.

Las constantes de los filtros complementarios se determinaron a partir de pruebas experimentales. Durante las cuales se observó que las mediciones del acelerómetro se veían afectadas por la irregularidad del terreno. Por esta razón, se asignó una mayor ponderación a los valores proporcionados por el giróscopo. Por otro lado, en la fusión de datos entre el giróscopo y el magnetómetro, se asignó igual ponderación a ambas mediciones.

Las pruebas realizadas con el sistema GNSS demostraron limitaciones significativas en entornos urbanos, afectando tanto la frecuencia de actualización como la confiabilidad de las mediciones. Como trabajo futuro, se propone la implementación de tecnología RTK, basada en técnicas de posicionamiento diferencial GNSS [13], con el objetivo de mejorar la precisión y estabilidad de las mediciones.

Por otro lado, la estimación de velocidad mediante la combinación de mediciones del acelerómetro y el GNSS no resultó viable bajo las condiciones de prueba. Como alternativa, se propone integrar las mediciones GNSS con un encoder acoplado a una de las ruedas del vehículo.

Los resultados obtenidos demuestran la viabilidad de emplear sensores de bajo costo y técnicas de fusión de sensores para aplicaciones de navegación autónoma en tiempo real.

## References

[1] Á. López Revuelta, "Orientation estimation and movement recognition using low cost sensors," 2017.

[2] B. Suwandi, T. Kitasuka, and M. Aritsugi, "Low-cost IMU and GPS fusion strategy for apron vehicle positioning," in Proc. IEEE Region 10 Conf. (TENCON), 2017, pp. 449–454.

[3] A. Laurell, E. Karlsson, and Y. Naqqar, "GPS and IMU sensor fusion to improve velocity accuracy," Bachelor's thesis, Uppsala Univ., Uppsala, Sweden, 2022.

[4] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, "GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects," *Information Fusion*, vol. 7, no. 2, pp. 221–230, 2006.

[5] P. Gui, L. Tang, and S. Mukhopadhyay, "MEMS based IMU for tilting measurement: Comparison of complementary and Kalman filter based data fusion," in *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, 2015, pp. 2004–2009.

[6] M. Miretti, F. Busano, E. Bernardi, H. Pipino, and G. C. Peretti, "Estimación híbrida de posición angular en dispositivos de captura de imágenes aéreas," in *VIII Jornadas de Ciencia y Tecnología para Alumnos (CyTAL)*, 2018.

[7] K. Saadeddin, M. F. Abdel-Hafez, and M. A. Jarrah, "Estimating vehicle state by GPS/IMU fusion with vehicle dynamics," *Journal of Intelligent & Robotic Systems*, vol. 74, pp. 147–172, 2014.

[8] S. Sasani, J. Asgari, and A. R. Amiri-Simkooei, "Improving MEMS-IMU/GPS integrated systems for land vehicle navigation applications," *GPS Solutions*, vol. 20, pp. 89–100, 2016.

[9] S. Colton, "The balance filter: a simple solution for integrating accelerometer and gyroscope measurements for a balancing platform," *Chief Delphi White Paper*, 2007.

[10] E. Malinen, "Fusion of data from quadcopter's inertial measurement unit using complementary filter," Bachelor's thesis, School of Energy Systems, Lappeenranta Univ. Technol., Lappeenranta, Finland, 2015.

[11] I. Jaime, F. Beck Ferrero, M. L. Miretti, and E. Bernardi, "Caracterización y calibración de unidades de medición inercial de bajo costo," 2024. [Online]. Available: https://ria.utn.edu.ar/server/api/core/bitstreams/7a31b183-3d8b-4fbe-b916-9ed34d02cd11/content

[12] K. Fletcher, *GNSS Data Processing, Vol. I: Fundamentals and Algorithms*. ESA Communications, ESA TM-23/1, 2013.

[13] E. D. Kaplan and C. J. Hegarty, "Chapter 8: Differential GPS," in *Understanding GPS: Principles and Applications*, 2nd ed. Norwood, MA, USA: Artech House, 2006, pp. 379–458.

ISBN 978-631-90145-4-9 - Congreso Argentino de Sistemas Embebidos CASE 2026 - 6 y 7 de agosto

# Sistema Embebido para Monitoreo de Variables Eléctricas, Térmicas y Ambientales Aplicadas al Diagnóstico de Baterías de Segunda Vida

Ilber Adonayt Ruge Ruge
Programa de Ingenieria Electronica
Universidad Pedagogica y Tecnologica de Colombia
Tunja - Boyacá
ilber.ruge@uptc.edu.co

Juan David Acosta Acosta
Programa de Ingenieria Electronica
Universidad Pedagogica y Tecnologica de Colombia
Tunja - Boyacá
juan.acosta06@uptc.edu.co

Santiago Amezquita Alfonso
Programa de Ingenieria Electronica
Universidad Pedagogica y Tecnologica de Colombia
Tunja - Boyacá
santiago.amezquita@uptc.edu.co

Oscar Mauricio Hernandez Gomez
Programa de Ingenieria Electronica
Universidad Pedagogica y Tecnologica de Colombia
Tunja - Boyacá
oscarmauricio.hernandez@uptc.edu.co

Luis David Patarroyo Gutierrez
Programa de Ingenieria Electronica
Universidad Pedagogica y Tecnologica de Colombia
Tunja - Boyacá
luis.patarroyo@uptc.edu.co

***Abstract*— This paper presents the design and implementation of an embedded system for monitoring electrical, thermal, and environmental variables related to vehicular battery operation. The system acquires voltage, current, battery temperature, ambient temperature, and relative humidity data. It also provides Bluetooth wireless communication, Comma-Separated Values (CSV) data storage, and real-time supervision through a MATLAB-based graphical interface. The development was conducted within the international project led by Universidade Federal do Amazonas, with participation from Universidad Pedagógica y Tecnológica de Colombia in embedded systems for battery monitoring. Experimental validation was performed under dynamic operating conditions, including idling, acceleration, deceleration, and braking. Results demonstrated stable operation of data acquisition, transmission, storage, and visualization processes, with good agreement between measured values and reference measurements. The proposed system represents a scalable platform for future energy analysis and battery state-of-health (SoH) assessment in electromobility applications.**

***Keywords*— *Embedded systems, battery monitoring, state of health (SoH), second-life batteries, data acquisition***

## I. Introducción

Las baterías reutilizadas constituyen una alternativa relevante para el almacenamiento energético en movilidad eléctrica y sistemas aislados, particularmente en regiones con acceso limitado a infraestructura energética convencional. Su aprovechamiento permite reducir costos y extender la vida útil de sistemas electroquímicos con capacidad operativa remanente. Este enfoque adquiere especial importancia en escenarios de transición energética y contextos sociales vulnerables, donde la reutilización tecnológica contribuye al desarrollo de soluciones sostenibles para movilidad y suministro energético [1], [2], [3].

El uso de baterías reutilizadas requiere sistemas de monitoreo capaces de supervisar variables eléctricas, térmicas y ambientales relacionadas con su comportamiento operativo y degradación. Parámetros como voltaje, corriente, temperatura y condiciones ambientales permiten evaluar el estado eléctrico, capacidad energética y desempeño de los sistemas de almacenamiento [4], [5]. En los últimos años, los sistemas embebidos y tecnologías de adquisición de datos han impulsado el desarrollo de plataformas de bajo costo para monitoreo energético en tiempo real [6]–[8]. Sin embargo, muchas soluciones comerciales presentan limitaciones asociadas a altos costos, baja escalabilidad y reducida integración con herramientas de supervisión y almacenamiento histórico de datos.

Este trabajo presenta un sistema embebido para el monitoreo de variables eléctricas, térmicas y ambientales orientado al análisis operativo de baterías. El sistema integra sensores de voltaje, corriente, temperatura y humedad, comunicación Bluetooth, visualización mediante pantalla de Diodo Orgánico Emisor de Luz (OLED), almacenamiento en formato CSV y supervisión remota desde MATLAB. El desarrollo se realizó en el marco de un proyecto internacional liderado por la Universidade Federal do Amazonas con participación de la Universidad Pedagógica y Tecnológica de Colombia. El prototipo constituye una plataforma escalable para futuras estrategias de estimación del estado de salud (SoH) de baterías [9], [10], validada preliminarmente en motocicletas.

## II. Metodología

La metodología propuesta se basa en el diseño e implementación de un sistema embebido para la adquisición, transmisión, almacenamiento y visualización de variables eléctricas, térmicas y ambientales asociadas al funcionamiento de baterías de motocicleta. La arquitectura, presentada en la Figura 1, integra módulos de sensado, procesamiento y comunicación inalámbrica para el monitoreo en tiempo real de voltaje, corriente, temperatura de batería, temperatura ambiente y humedad relativa. Además, incorpora almacenamiento local en formato CSV mediante microSD y supervisión remota desde una interfaz gráfica en MATLAB, lo que facilita el análisis de la operación de la batería y el registro de las variables que son de utilidad para la evaluación de futuras estrategias de diagnóstico del estado de salud (SoH) de la batería.

Las variables de interés consideradas en la propuesta incluyen mediciones directas de voltaje, corriente, temperatura y humedad relativa, así como variables estimadas orientadas al análisis del estado operativo de la batería, entre

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

ellas la resistencia interna y el tiempo de descarga [11], [12], [13]. Con este conjunto de variables medidas se propone una base experimental para futuras estrategias de estimación del SoH y análisis de degradación de baterías en aplicaciones vehiculares.

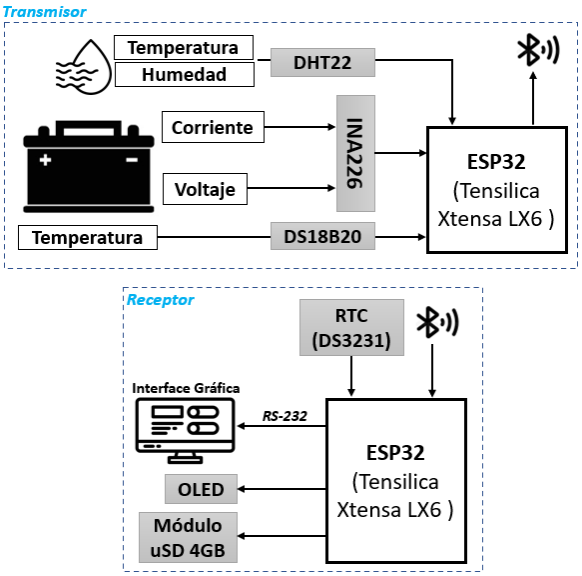



Fig. 1. Arquitectura general del sistema de monitoreo y variables analizadas de la batería.

La arquitectura propuesta consta de dos bloques un transmisor y un receptor. El bloque transmisor realiza la adquisición y transmisión inalámbrica de variables asociadas al comportamiento operativo de la batería. Para la medición de voltaje y corriente se emplea el sensor INA226, que integra un ADC de 16 bits y comunicación I2C, efectuando mediciones bidireccionales de alta precisión y bajo consumo energético. Estas variables permiten analizar procesos de carga y descarga, así como identificar condiciones de sobrecarga [14], [15], descarga profunda y degradación potencial de la batería. El monitoreo térmico se implementa mediante el sensor digital DS18B20, mientras que las variables ambientales de temperatura y humedad relativa son adquiridas mediante el sensor DHT22, estas mediciones se utilizan para evaluar a futuro el análisis correlacional de las condiciones externas sobre el desempeño del sistema [16]. El procesamiento se realiza mediante un microcontrolador ESP32-WROOM-32, basado en arquitectura Tensilica Xtensa LX6 de doble núcleo y 32 bits, encargado de la adquisición de datos y de la transmisión inalámbrica vía Bluetooth hacia el módulo receptor para su almacenamiento y visualización.

El bloque receptor está orientado a la recepción, almacenamiento y visualización de la información transmitida desde el sistema embebido instalado en la motocicleta. La recepción inalámbrica se realiza mediante un microcontrolador ESP32-WROOM-32, que gestiona la comunicación con los módulos de almacenamiento y supervisión externa. Esta arquitectura desacopla la adquisición de señales de los procesos de registro y análisis, mejorando la estabilidad operativa del sistema. Para el almacenamiento histórico de las variables se emplea un módulo microSD de 4 GB con comunicación SPI, utilizado para registrar la información en formato CSV. Adicionalmente, se incorpora un reloj de tiempo real DS3231 para generar marcas temporales asociadas a cada registro. La visualización local se realiza mediante una pantalla OLED de 0.96 pulgadas con resolución de 128 × 64 píxeles. Finalmente, la supervisión y análisis de datos se implementa mediante una interfaz gráfica desarrollada en MATLAB App Designer, permitiendo visualización en tiempo real e importación de registros históricos para análisis posteriores.

El desarrollo del sistema embebido se realizó utilizando el entorno de programación Arduino IDE versión 2.3.2. La programación del microcontrolador ESP32 se implementó mediante un lenguaje basado en Processing y C/C++, desarrollando rutinas de adquisición, procesamiento y transmisión de datos. Adicionalmente, se emplearon librerías especializadas para la comunicación con los sensores INA226 (INA226_WE.h), DS18B20 (OneWire.h y DallasTemperature.h) y DHT22 (DHT.h), así como para el control de la pantalla OLED (Adafruit_GFX.h y Adafruit_SSD1306.h), almacenamiento en memoria microSD (SPI.h y SD.h) y comunicación Bluetooth (BluetoothSerial.h). La utilización de estas librerías facilitó la integración de los módulos de medición, visualización y almacenamiento requeridos por el sistema de monitoreo propuesto.

La Figura 2 presenta los diagramas de flujo de los bloques transmisor y receptor del sistema de monitoreo. En el bloque transmisor, el algoritmo configura las variables $T_A$, $H_A$, $I_{bat}$, $V_{bat}$ y $T_{bat}$, además de la comunicación Bluetooth a 9600 baudios. La adquisición de datos se ejecuta cada 10 s mediante una interrupción Timer0, transmitiendo posteriormente la información hacia el bloque receptor. En el bloque receptor, el algoritmo inicializa la comunicación serial y los periféricos OLED, Reloj en Tiempo Real (RTC) y memoria microSD. La recepción de datos se realiza mediante una interrupción serial RS232. Posteriormente, el sistema genera la marca temporal, almacena la información en formato CSV, actualiza la visualización local en la pantalla OLED y transmite los datos hacia la interfaz gráfica desarrollada en MATLAB.

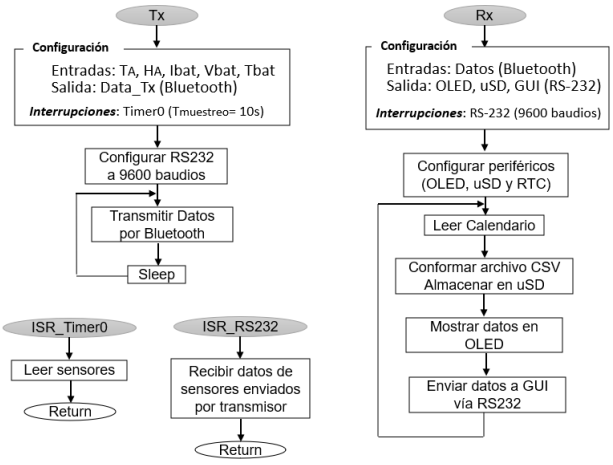



Fig. 2. Algoritmos de los bloques transmisor y receptor del sistema de monitoreo.

Las pruebas experimentales se realizaron utilizando una batería de motocicleta GX MF-12N9-BS de 12 V y 9 Ah bajo condiciones normales de operación vehicular, incluyendo estados de encendido, ralentí y variaciones de carga eléctrica. Durante cada ciclo de muestreo de 10 s, las variables adquiridas fueron transmitidas inalámbricamente hacia el bloque receptor para su almacenamiento en formato CSV con marcas temporales generadas por el RTC DS3231. La adquisición continua de estas variables permite establecer una

base experimental orientada al análisis del estado eléctrico y capacidad energética de la batería, considerando parámetros como voltaje en reposo, voltaje bajo carga, corriente de descarga, comportamiento térmico y condiciones ambientales de operación. La Figura 3 muestra la interfaz gráfica y un ejemplo de datos almacenados en la memoria microSD. La visualización en tiempo real facilita el monitoreo de variables medidas, mientras el almacenamiento asegura trazabilidad para futuros análisis energéticos y estimaciones del SoH de la baterías. [17], [18].




Fig. 3. GUI de monitoreo y almacenamiento de datos en formato CSV.

## III. Análisis de resultados

La validación experimental del sistema se realizó sobre una TVS Apache RTR 160 2V bajo condiciones reales de operación vehicular. Las pruebas incluyeron diferentes ciclos de conducción, como aceleración, desaceleración, frenado y ralentí, permitiendo analizar el comportamiento de la batería frente a distintas demandas energéticas del sistema eléctrico. Durante la operación se adquirieron continuamente variables eléctricas, térmicas y ambientales, las cuales fueron transmitidas inalámbricamente, almacenadas en formato CSV y visualizadas en tiempo real mediante la pantalla OLED y la interfaz gráfica desarrollada en MATLAB. La configuración experimental permitió verificar la estabilidad operativa de la arquitectura propuesta y validar el correcto funcionamiento de los procesos de adquisición, transmisión, almacenamiento y supervisión de datos en tiempo real.

### A. *Validacion de medicion de voltaje y corriente*

Con el propósito de validar el desempeño del sistema embebido propuesto, se realizaron mediciones experimentales en tiempo real de voltaje y corriente sobre la batería de una motocicleta bajo diferentes condiciones dinámicas de operación vehicular, incluyendo estados de ralentí y variaciones de carga eléctrica del sistema. Los valores registrados por el prototipo fueron comparados con las mediciones obtenidas mediante un multímetro digital de referencia Fluke 17B. En la Tabla I se registraron las mediciones de voltaje y corriente, valores de referencia y la precisión en las mediciones en distintos instantes del ciclo de prueba, esta información permitió evaluar el comportamiento del sistema de adquisición y estimar la precisión de medición implementada. La precisión se determinó mediante la ecuación (1), donde *Vref* y *Vmed* corresponden a los valores medidos por el instrumento de referencia y el sistema embebido, respectivamente.

$$Precision(\%) = \left(1 - \frac{|Vref - Vmed|}{Vref}\right) * 100 \qquad (1)$$

TABLA I. COMPARACIÓN DE MEDICIONES ENTRE EL SISTEMA EMBEBIDO Y EL FLUKE 17B.

| Tiempo (s) | Voltaje medido (V) | Referencia (V) | Precisión (%) |
|---|---|---|---|
| 20 | 12.14 | 12.68 | 95.74 |
| 60 | 12.02 | 12.61 | 95.32 |
| 120 | 11.93 | 12.47 | 95.67 |
| 180 | 11.71 | 12.29 | 95.28 |
| 240 | 11.52 | 12.10 | 95.21 |
| 300 | 11.08 | 11.89 | 93.19 |
| **Tiempo (s)** | **Corriente medida (A)** | **Referencia (A)** | **Precisión (%)** |
| 20 | 1.74 | 1.92 | 90.63 |
| 60 | 2.05 | 2.22 | 92.34 |
| 120 | 2.48 | 2.71 | 91.51 |
| 180 | 2.96 | 3.14 | 94.27 |
| 240 | 2.61 | 2.84 | 91.90 |
| 300 | 1.87 | 2.01 | 93.03 |

Los resultados evidenciaron precisiones promedio de 95.07 % para voltaje y 92.28 % para corriente, confirmando una adecuada concordancia respecto al instrumento patrón. Adicionalmente, la medición de corriente resulta fundamental para el análisis energético y futuras estrategias de estimación del SoH de baterías, debido a su relación directa con procesos de carga, descarga y profundidad de descarga [19], [20].

La Figura 4 presenta evidencia experimental de la implementación del sistema de monitoreo sobre la motocicleta durante las pruebas de adquisición de datos, donde se presenta la integración física del prototipo y la visualización en tiempo real de las variables eléctricas monitoreadas, verificando el correcto funcionamiento de la arquitectura propuesta bajo condiciones reales de operación.




Fig. 4. Validación experimental del sistema de monitoreo en la motocicleta.

### B. *Validacion de medicion de temperatura y vatiables ambientales*

El sistema embebido permitió el monitoreo continuo de la temperatura de la batería, temperatura ambiente y humedad relativa durante diferentes condiciones de operación de la motocicleta. La Figura 5 muestra los registros obtenidos desde la interfaz gráfica desarrollada en MATLAB, evidenciando la evolución temporal de las variables monitoreadas en tiempo real. Los resultados confirmaron el correcto funcionamiento y

estabilidad del sistema de adquisición implementado, permitiendo supervisar el comportamiento térmico bajo distintas condiciones eléctricas y ambientales. Además, los datos registrados constituyen una base experimental relevante para futuros trabajos orientados a la estimación del SoH y degradación de baterías.

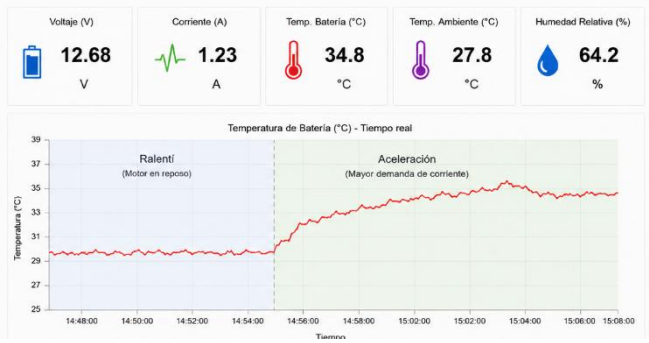



Fig. 5. Registro en tiempo real de la temperatura de la bateria y variables ambientales.

## IV. Conclusiones

Los resultados experimentales validaron el funcionamiento de la arquitectura embebida propuesta para el monitoreo de variables eléctricas, térmicas y ambientales en baterías vehiculares. La integración entre adquisición de datos, transmisión inalámbrica, almacenamiento CSV y supervisión en tiempo real mediante MATLAB evidenció estabilidad operativa durante diferentes condiciones dinámicas de la motocicleta, incluyendo ralentí, aceleración, desaceleración y frenado.

Las pruebas de validación evidenciaron una adecuada concordancia entre el sistema embebido y el instrumento de referencia Fluke 17B, obteniéndose precisiones promedio de 95.07 % en voltaje y 92.28 % en corriente. Además, el monitoreo térmico y ambiental permitió registrar condiciones operativas durante los ciclos de conducción. Los resultados demuestran la viabilidad del sistema para monitoreo energético y adquisición de datos en tiempo real, proporcionando una base experimental para futuras estrategias de estimación del estado de salud (SoH) de baterías.

## Referencias

[1] Y. Xiao, J. Wen, L. Yao, J. Zheng, Z. Fang and Y. Shen, “A comprehensive review of the lithium-ion battery state of health prognosis methods combining aging mechanism analysis,” Journal of Energy Storage, vol. 65, pp. 1–15, 2023, doi: https://doi.org/10.1016/j.est.2023.107347.

[2] N. Belmajdoub, R. Lajouad, A. El Magri and S. Boudoudouh, “State of charge and state of health estimation of a lithium-ion battery for electric vehicles: A review,” IFAC-PapersOnLine, vol. 58, no. 13, pp. 460–465, 2024, doi: https://doi.org/10.1016/j.ifacol.2024.07.525.

[3] R. Swarnkar, R. Harikrishnan and M. Singh, “Analysis of Electric Vehicle Battery State Estimation Using Scopus and Web of Science Databases from 2000 to 2021: A Bibliometric Study,” World Electric Vehicle Journal, vol. XX, no. XX, pp. XX–XX, 2022, doi: https://doi.org/10.3390/wevj13080157.

[4] J. A. Braun, R. Behmann, D. Schmider and W. Bessler, “State of charge and state of health diagnosis of batteries using voltage-controlled equivalent circuit models,” Journal of Power Sources, vol. 544, pp. 1–11, 2022, doi: https://doi.org/10.1016/j.jpowsour.2022.231828.

[5] J. Tian, R. Xiong, and W. Shen, “State-of-health estimation based on differential temperature for lithium-ion batteries,” IEEE Transactions on Power Electronics, vol. 35, no. 10, pp. 10363 – 10373, 2020, doi: https://doi.org/10.1109/TPEL.2020.2978493.

[6] S. Haldar, S. Gol, A. Mondal and R. Banerjee, “IoT-enabled advanced monitoring system for tubular batteries: Enhancing efficiency and reliability,” e-Prime - Advances in Electrical Engineering, Electronics and Energy, vol. 9, pp. 1–14, 2024, doi: https://doi.org/10.1016/j.prime.2024.100709.

[7] H. Li, “IoB: Internet-of-batteries for electric Vehicles–Architectures, opportunities, and challenges,” Green Energy and Intelligent Transportation, vol. 2, no. 6, pp. 2–22, 2023, doi: https://doi.org/10.1016/j.geits.2023.100128.

[8] D. Kerley, J. Hyun and D. Ha, “Automotive lead-acid battery state-of-health monitoring system,” in IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society, 2016, doi: https://doi.org/10.1109/IECON.2015.7392714.

[9] Z. Wei, H. Ruan and Y. Li, “Multi-stage state of health estimation of lithium-ion battery with high tolerance to heavily-partial charging,” IEEE Transactions on Power Electronics, vol. 37, no.6, pp. 7432–7442, 2022, doi: http://dx.doi.org/10.1109/TPEL.2022.3144504.

[10] J. Gong et al., “Predictive Modeling for Electric Vehicle Battery State of Health: A Comprehensive Literature Review,” Energies, vol. 18, no. 2, pp. 1–37, 2025, doi: https://doi.org/10.3390/en18020337.

[11] X. Bian, Z. Wei, W. Li and J.Pou, “State-of-Health Estimation of Lithium-Ion Batteries by Fusing an Open Circuit Voltage Model and Incremental Capacity Analysis,” IEEE Transactions on Power Electronics, vol. 37, no. 2, pp. 2226-2236, 2022, doi: https://dx.doi.org/10.1109/TPEL.2021.3104723.

[12] Y. Li, M. Abdel-Monem, R. Gopalakrishnan, M. Berecibar, E. Nanini-Maury, N. Omar, P. van den Bossche and J. van Mierlo, “A quick on-line state-of-health estimation method for Li-ion battery with incremental capacity curves processed by Gaussian filter,” Journal of Power Sources, vol. 373, pp. 40–53, 2018, doi: https://doi.org/10.1016/j.jpowsour.2017.10.092.

[13] B. Acuña, D. Chérrez, J. Rodriguez, F. Grijalva and L. Pereira, “Robust Data-Driven State of Health Estimation of Lithium-Ion Batteries Based on Reconstructed Signals,” Energies, vol. 18, no. 10, 2025, doi: https://doi.org/10.3390/en18102459.

[14] A. Gismero, K. Norregard, B. Johnsen, L. Stenhoj and E. Schaltz, “Electric vehicle battery state of health estimation using incremental capacity analysis,” Journal of Energy Storage, vol. 64, pp. 1–8, 2023, doi: https://doi.org/10.1016/j.est.2023.107110.

[15] H. Liu, C. Li, X. Hu, J. Li, K. Zhang, Y. Xie, R. Wu and Z. Song, “Multi-modal framework for battery state of health evaluation using open-source electric vehicle data,” Nature Communications, vol. 16, no. 1137, pp. 1–12, 2025, doi: https://doi.org/10.1038/s41467-025-56485-7.

[16] R. Sun, J. Chen and C. Piao, “Battery health features extraction and state of health estimation based on real-time online vehicle driving data,” Journal of Power Sources, vol. 645, pp. 1–13, 2025, doi: https://doi.org/10.1016/j.jpowsour.2025.236784.

[17] Y. Wu et al., “State-of health estimation for lithium-ion batteries based on healthy features and long short-term memory,” IEEE Access, vol. 8, pp. 28533–28547, 2020, doi: https://doi.org/10.1109/ACCESS.2020.2972344.

[18] B. Gou, Y. Xu, and X. Feng, “State-of-health estimation and remaining useful life prediction for lithium-ion battery using a hybrid data-driven method,” IEEE Transactions on Vehicular Technology, vol. 69, no. 10, pp. 10854 - 10867, 2020, doi: https://doi.org/10.1109/TVT.2020.3014932.

[19] M. Karthikeyan, N. Anirudh, N. Krishnan, C. Columbus and C. Aravind, “Optimizing battery health monitoring in electric vehicles using interpretable CART–GX model,” Results in Engineering, vol. 27, 2025, doi: https://doi.org/10.1016/j.rineng.2025.106043.

[20] J. Zhao, D. Li, Y. Li, D. Shi, J. Nan and A. Burke, “Battery state of health estimation under fast charging via deep transfer learning,” iScience A Cell Press Journal, vol. 28, no. 5, pp. 1–24, 2025, doi: https://doi.org/10.1016/j.isci.2025.112235.

ISBN 978-631-90145-4-9 - Congreso Argentino de Sistemas Embebidos CASE 2026 - 6 y 7 de agosto

# Low Resources, Low Power, Real Time Multiplayer: A DDR Inspired Wearable Game

Fabricio Correa, Gennaro Monetti and Julián Oreggioni
Universidad de la República, Montevideo, Uruguay
Email: {fabricio.correa, gennaro.monetti, juliano}@fing.edu.uy

***Abstract*—This paper presents the implementation of a video game inspired by Dance Dance Revolution on a 16-bit microcontroller (MCU). Using only 385 B of RAM, 13 kB of flash memory, and having the CPU master clock at 14 MHz, the system consists of: game logic, control of a 128x32 pixel OLED display, user input handling, and communication between MCUs for multiplayer mode, both via direct communication or over the Internet through a PC. The proposed solution demonstrates how low-cost embedded platforms are able to support non-trivial interactive applications while maintaining a low power consumption. In fact, the system is capable of supporting more than 80 hours of continuous gameplay powered by a 230 mAh CR2032 battery, which is equivalent to more than 2400 singleplayer games.**

***Index Terms*—embedded video game, MSP430, low-power**

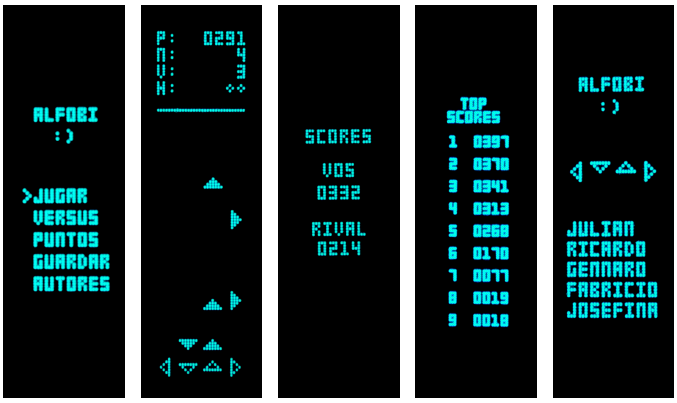
## I. Introduction

In recent years, wearable devices have experienced sustained global adoption. In particular, 611.5 million wearable devices were manufactured worldwide in 2025, with a year-over-year growth of 9.1% [1]. Consequently, wearable platforms have become attractive for exploring novel low-power interactive applications.

Embedded systems with limited computational resources and strict energy constraints are typically designed for dedicated tasks, making the implementation of highly interactive real-time applications particularly challenging. Video games constitute a demanding application in this context, since they require low-latency input processing and immediate feedback to the player [2]. Among them, rhythm games are especially demanding due to their strong dependence on audiovisual and motor synchronization.

Dance Dance Revolution (DDR) [3], [4] was one of the first video games to popularize the rhythm game genre [5], where audio, visual, and motor coordination play a central role in the gameplay experience [6]. Since its introduction, DDR has attracted interest beyond the entertainment domain and has been the subject of multiple academic studies. Previous works have reported positive effects on players' social life and physical health, including improvements in endurance, muscle strength, and sense of rhythm, while also fostering social interaction and the creation of new friendships [5]. In addition, DDR has been explored as a potential exercise therapy for patients with Huntington's disease [6], and its emotional impact on players has also been studied [7].

This work presents the design and implementation of a portable DDR-inspired game on a very low-resource embedded platform featuring a small OLED display and push buttons instead of a dance pad.




(a) Menu (b) Game (c) End (d) Scores (e)Authors

Figure 1: System screens

## II. Game Mechanics

Similarly to the original game [3], [4], the system displays sequences of arrows moving along four columns. The arrows appear at the top of the screen and descend until either they disappear at the bottom edge or the corresponding button is pressed (the system includes one button for each column and an additional "special" button). At the bottom of each column, an outline of the corresponding arrow indicates the *hit area* (see Fig. 1.b).

The objective of the game is to obtain the highest possible score while avoiding the loss of the available *lives* (i.e., the number of mistakes allowed by the system). When the player presses the button corresponding to a column, the system evaluates the position of the lowest arrow in that column relative to the hit area. If the arrow overlaps the outline, a score proportional to the overlap degree is awarded. In all cases, such arrow is removed. The system subtracts lives both when a button is pressed without any arrow being sufficiently close to the hit area and when an arrow disappears at the bottom of the screen without being hit. When the latter occurs, temporary invincibility is granted, during which the player does not lose lives if another arrow reaches the end of the screen.

The game difficulty increases over several levels. The level determines the descending speed of the arrows, as well as the probability of multiple arrows being introduced simultaneously at each sequence step.

The user can navigate through different screens by interacting with menus (some of them shown in Fig. 1).

After booting the system presents the main menu (Fig. 1.a), where several actions may be performed: start a *single*

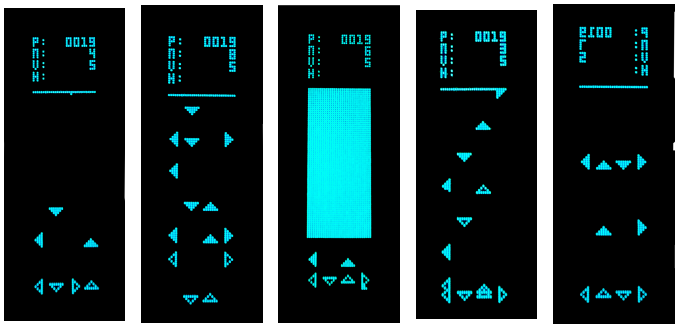


Figure 2: Handicaps

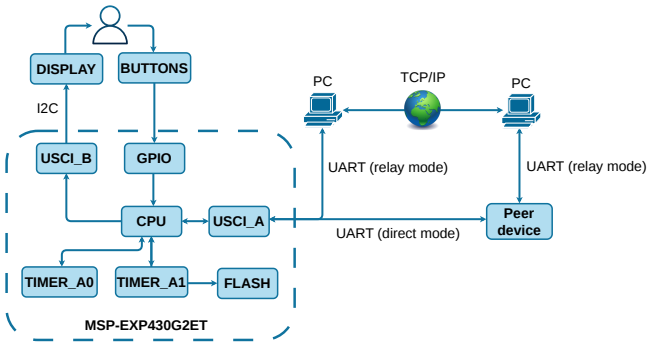



Figure 3: Block diagram

*player* game, start a *versus* game, go to the scoreboard, or persist the current scoreboard state. The scoreboard (Fig. 1.d) displays up to nine of the highest scores achieved. Results are kept in volatile memory until the user decides to save them into persistent memory. The game screen (Fig. 1.b) shows an information section at the top, including three indicators identified by letters corresponding to their initials in Spanish: score (P, *Puntaje*), level (N, *Nivel*), and lives (V, *Vidas*). Additionally, in versus games, a handicap indicator (H) is displayed, represented by zero to five asterisks, showing the difficulty of the disadvantage that would be applied to the opponent (see versus matches below).

There are two types of matches: "singleplayer" and "versus". Single player matches end when all levels are completed or all lives are lost. In this mode, the special button displays a pause menu, allowing the player to restart the match or return to the main menu.

In versus matches, players compete using their own hardware connected to the opponent's hardware. Both players receive the same arrow sequence and, as their score increases, unlock handicaps that can be sent to the rival. Fig. 2 shows the available handicaps. From left to right: *Swap columns* randomly exchanges the position of two columns; *Change outline height* moves the hit area of two columns to random positions; *Flash* hides a large portion of the display; *Armored* arrows, as outlines, require two hits to be removed; and finally, *Mirror* vertically flips the entire display. The duration of each handicap depends on its type. Versus matches end when all lives are lost or when the maximum score of 9999 is reached. Once both players finish, the opponent's score is displayed (Fig. 1.c).

## III. Hardware

Fig. 3 shows the block diagram of the proposed solution. The hardware consists of the MSP430G2553 microcontroller (MCU) from Texas Instruments, configured to operate at 14 MHz. The MCU includes 16 kB of Flash memory, 512 B of RAM, up to 16 general-purpose input/output (GPIO) pins, two hardware timer modules, and two USCI (Universal Serial Communication Interface) serial communication modules. Both USCI modules were used: one for UART (Universal Asynchronous Receiver-Transmitter) inter-MCU communication during versus matches, and the other for $I^2C$ communication with the display. The UART implementation allows MCUs to connect either directly through their pins or remotely over the Internet through a PC-based relay, using the Transport Control Protocol (TCP) for reliable communication. Five GPIO pins were used for the push buttons, along with the two timer modules: TIMER 0 for handling "logical timers" (see Section IV-C1) and TIMER 1 for button debouncing. Finally, Flash memory was used for persistent storage of user scores.

Additionally, a 0.91" OLED display with an SSD1306 controller was used. The display has a resolution of 32 x 128 pixels and includes a Graphic Display Data RAM (GDDRAM) organized into 4 pages and 128 columns. When viewed horizontally, the columns correspond to the 128 pixel columns and each page groups 8 rows of pixels. This display receives two types of messages: control messages for configuration and GDDRAM pointer positioning, and data messages for writing bytes into the selected page and column. To send a message to the display, a byte indicating the message type must first be transmitted, followed by another byte containing the payload. In the case of a data message, the bits of the second byte each represent one of the 8 pixels in the selected page and column.

## IV. Embedded Software

### A. Design Philosophy

The main priority was meeting appropriate response times and efficiently using memory resources. Furthermore, the system was designed to be easily scalable and configurable without requiring major modifications. To achieve this, a hardware abstraction layer was implemented, decoupling application logic from hardware-specific details. This modular separation presents high cohesion and low coupling [8]. Moreover, implementation parameters (e.g., screen positions, buffer sizes, etc.) were abstracted into constants and variables to facilitate modifications while avoiding hardcoded values, which also improves legibility. Regarding extensibility, data structures and interfaces were established. For example, implementing a new handicap would only require writing its logic and then using the established interfaces to register it without additional modifications.

### B. Architecture

The system is implemented using a task queue, where hardware interrupts capture asynchronous events and enqueue them [9]. The solution is implemented in *bare-metal*, providing more direct hardware control while avoiding race conditions. Once the function queue becomes empty, the MCU enters

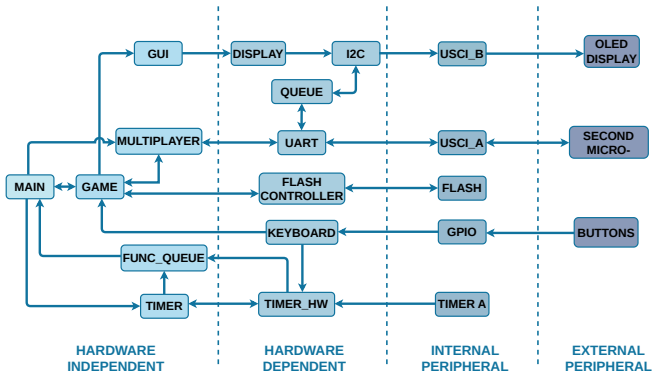



Figure 4: System diagram

Low Power Mode 3 (LPM3), where only the auxiliary clock remains active.

The following interrupts are handled: empty $I^2C$ and UART buffers, timer expiration and button presses.

Most of the game logic is executed based on *game ticks*, which are added to the function queue every 20 ms by a timer. On each tick, the game state is updated and user input is processed. Section IV-C2 explains how these ticks are processed.

Fig. 4 shows the system's module diagram, where arrow direction represents information flow. The next section details the implementation of the most important modules.

### C. Implementation Details

*1) Timer:* This module manages the different timing-control needs of the system. Based on TIMER 0, multiple "logical timers" with arbitrary intervals and callbacks are managed. These logical timers can be configured at runtime through functions. Currently, two logical timers are used: one for generating game ticks and the other one for multiplayer protocol timeouts.

*2) Game:* This module is responsible for handling user input, arrow movement, pseudo-random generation of arrow sequences, score and life management, and handicap management.

Fig. 5 shows the main entry point of the Game module. It should be noted that the diagram represents a simplification, as handicap sending/receiving and sequence generation logic are omitted. In the figure, *lower_arrow_ticks* is a counter used to control arrow movement speed.

For pseudo-random number generation, the *rand* function from *stdlib* [10] is used, whose implementation uses a linear congruential generator with a period of $2^{31}$ values, which is adequate for generating gameplay sequences without repetition. The *rand* function requires an initial value: the "seed". The seed is captured upon starting the game from a counter that is incremented in every iteration of the main loop.

Arrow sequences are generated dynamically during gameplay. Once arrows have moved enough to make room for new arrows, the system pseudo-randomly decides in how many columns (zero to four) and in which specific columns arrows will be inserted. The probability of generating arrows in a certain number of columns on the same game tick depends on the current level.

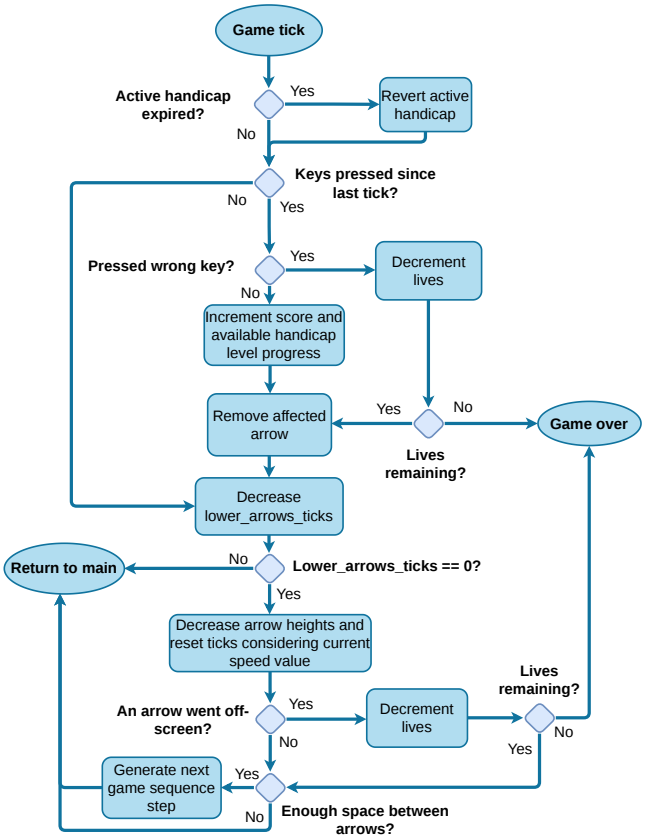



Figure 5: Execution flow of a *game tick*

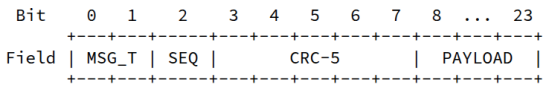



Figure 6: Multiplayer message format

*3) Multiplayer:* This module is responsible for implementing the communication protocol between MCUs at the application layer. The protocol aims to improve the reliability that would be obtained using the parity bit provided by UART and to guarantee message ordering. To achieve this, a 3-byte message protocol was implemented, whose header (see Fig. 6) includes: message type identifier, sequence number and a CRC-5-USB error detection code [11], capable of detecting burst errors of up to 5 bits.

The protocol supports the following message types: ACK, SEED, HANDICAP and END_OF_GAME. ACK is an acknowledgment message that must be sent in response to any other message. Messages are retransmitted every second until the corresponding ACK is received.

To start a versus match, MCUs asynchronously exchange their seeds (SEED messages). To ensure that both players get the same arrow sequence, the sum of the exchanged values is used as the game seed.

A HANDICAP message tells the recipient to apply the handicap type specified in the message payload.

Finally, END_OF_GAME messages indicate that the sender has finished the match, delivering the achieved score as the payload of the message.

*4) Flash:* The MCU makes it possible to pre-allocate Flash segments in multiples of 512 bytes for custom program data usage. In order to write a byte into memory, the entire segment must be erased first. Erase operations take several seconds and degrade memory cells, limiting their lifetime [12].

To minimize erase operations, a circular memory management algorithm was implemented. Storing metadata about valid memory positions would require using a fixed (known) section of the segment, which would need to be overwritten every time data is saved. To avoid this, erasure is performed by leaving all segment bits set to 1, so that the last valid position can be found by scanning the segment until finding a word different from 0xFFFF (valid score values go up to 0x270F, i.e., 9999).

## V. Experimental Results

### A. Current Consumption

Current consumption was characterized using the *Power Profiler Kit II* from Nordic Semiconductor. Fig. 7 shows the current consumption during a single player session (there is almost no difference between this mode and versus mode in terms of power consumption). The figure indicates the different operating states throughout the session: main menu (1 and 6), game in progress (2 and 4), pause (3), and game over screen (5). Table I breaks down the current consumption according to stage and hardware component.

Average current consumption during gameplay is 2.6 mA, where the display consumes 1.9 mA (70%) and the microcontroller 0.8 mA (30%). The implementation presents high energy efficiency: operating on a 230 mAh CR2032 battery, the system would be able to run for 80 hours of continuous gameplay, equivalent to more than 2400 matches (considering two minutes per match).

### B. Flash and RAM Usage

The system implementation required 13 kB of the available 16 kB of Flash memory (80%) and 385 B of the available 512 B of RAM (75%).

### C. Multiplayer Interaction Latency

The time elapsed between a player pressing a button and the handicap being applied on the opponent's MCU was measured. 40 measures were collected. The observed values range from 15.82 ms to 52.67 ms, with an average of 34.66 ms. This latency includes the time required to debounce the button's input, the time between game ticks (up to 20 ms on each player's MCU) and the UART transmission and message processing.

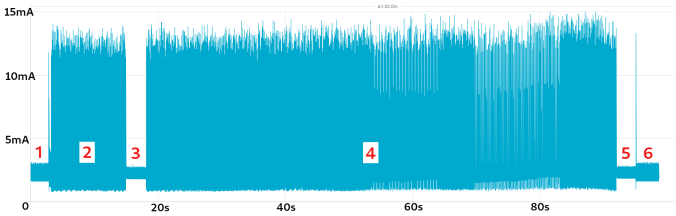



Figure 7: Power consumption during a match.

Table I: Current consumption by system stage

| Stage | Total (mA) | MCU (mA) | Display (mA) |
|---|---|---|---|
| Main menu | 2.00 | 0.08 | 1.92 |
| Gameplay | 2.60 | 0.74 | 1.86 |
| Pause | 2.00 | 0.08 | 1.92 |
| Game over | 1.90 | 0.08 | 1.82 |

## VI. Conclusions

The proposed game was successfully implemented under the imposed hardware constraints. The modular architecture decouples game logic from hardware-specific details, improving maintainability, extensibility, and portability to other platforms and components (display and buttons), while achieving low current consumption and low resource utilization. Although no formal user experience study was conducted, the authors found the game to provide a fun gameplay experience.

Future improvements include adding hibernation functionality to further reduce power consumption while the game is idle, incorporating sound support to provide a more immersive rhythm-game experience, as well as a greater variety of handicaps, which would be straightforward thanks to the modular implementation. Additionally, a Wi-Fi module could be incorporated to eliminate dependence on a PC relay and thus enable a native online experience. The source code for this project is publicly available on GitHub[1].

## Acknowledgment

This work was partially funded by CSIC (Comisión Sectorial de Investigación Científica, Udelar) and IEEE Circuits and Systems Society (Uruguayan Chapter).

## References

[1] International Data Corporation, "Worldwide wearable device market results," https://www.idc.com/promo/wearablevendor/, 2025, [On line, Accessed: May, 2026].

[2] J. Pérez Mauri et al, "U-Tetris: Tetris controlado por UART," in *XII Congreso Argentino de Sistemas Embebidos, CASE, Evento virtual, Argentina, pp. 149-151.*, 2021.

[3] Konami, "Dance Dance Revolution," http://www.konami.com/, 1998, [On line, Accessed: May, 2026].

[4] H. Ishikawa et al, "Game machine, game device control method, information storage medium, game distribution device, and game distribution method," Feb. 24 2004, US Patent 6,695,694.

[5] J. Hoysniemi, "International survey on the dance dance revolution game," *Computers in Entertainment (CIE)*, vol. 4, no. 2, pp. 8–es, 2006.

[6] A. D. Kloos et al, "Video game play (dance dance revolution) as a potential exercise therapy in huntington's disease: a controlled clinical trial," vol. 27, no. 11, pp. 972–982, 2013.

[7] E. Hutton and S. S. Sundar, "Can video games enhance creativity? effects of emotion generated by dance dance revolution," *Creativity Research Journal*, vol. 22, no. 3, pp. 294–303, 2010.

[8] B. Scabbia, "High cohesion, low coupling and strong encapsulation," https://benscabbia.co.uk/programming/high-cohesion-low-coupling-and-strong-encapsulation, April 2018, [On line, Accessed: May, 2026].

[9] D. E. Simon, "Survey of software architectures," in *An Embedded Software Primer*. Addison-Wesley Professional, 1999, pp. 115–136.

[10] "C Standard Library," https://cplusplus.com/reference/cstdlib/, [On line, Accessed: May, 2026].

[11] USB Implementers Forum, "Cyclic redundancy check in usb," https://www.usb.org/sites/default/files/crcdes.pdf, 1999, [On line, Accessed: May, 2026].

[12] K. Yasar, "Flash memory," https://www.techtarget.com/searchstorage/definition/flash-memory, TechTarget, June 2023, [On line, Accessed: May, 2026].

[1] https://github.com/Nanush7/ALFOBI

ISBN 978-631-90145-4-9 - Congreso Argentino de Sistemas Embebidos CASE 2026 - 6 y 7 de agosto

# Ultra Low Power Solar Panel Testbench for IoT Devices Under Dynamic Partial Shading Conditions

Sofía Ferrando, Valentina Lluis, Agustina Nappa, and Julián Oreggioni,
Universidad de la República, Montevideo, Uruguay
Email: {sofia.ferrando.bergeret, valentina.lluis, agustina.nappa, juliano}@fing.edu.uy

***Abstract*—This work presents a prototype of a solar panel testbench for IoT applications. The platform is designed to study the response of small photovoltaic panels under partial and time varying shading conditions, such as those caused by fixed structures or passing clouds. To this end, the platform performs real time monitoring of the panel open circuit voltage together with relevant environmental variables.**

**The proposed platform is based on the MSP430G2553 microcontroller and periodically acquires temperature, atmospheric pressure, relative humidity, and ambient light, in addition to the panel open circuit voltage. Environmental sensing is implemented using the BME280 and OPT3001, while voltage measurements are obtained through the on chip analog to digital converter.**

**The platform allows user configuration of alarm thresholds, sampling frequency and other parameters. Measurements and alarms are transmitted in real time via Bluetooth to a host device.**

**Experimental results demonstrate that the proposed platform can characterize the dynamic behavior of small photovoltaic panels under realistic irradiance fluctuations, providing a practical tool for the design and validation of energy harvesting IoT systems, while maintaining an average current consumption below 80 $\mu$A during sensing and acquisition cycles (excluding wireless transmission).**

## I. Introduction

Energy autonomy is a critical requirement for electronic systems deployed in remote or hard to access locations, where connection to the electrical grid is unavailable and battery replacement is costly or impractical. Typical examples include environmental sensing stations [1], wildlife tracking devices [2], remote property surveillance platforms, livestock monitoring systems [3]–[5], and distributed nodes in smart agriculture or infrastructure monitoring. In such applications, battery capacity directly limits operational lifetime and increases maintenance requirements [6].

Energy harvesting from renewable sources has therefore become a key enabling technology for long term IoT deployments. Among the available options, solar energy stands out as the most practical solution for applications with moderate power demand, due to its high availability and energy density in outdoor environments [3], [6]. However, ensuring reliable operation requires proper dimensioning and characterization of the photovoltaic source, taking into account both load requirements and environmental conditions.

Photovoltaic performance is strongly affected by partial shading and environmental variability [7]. Even limited shading can significantly reduce output power due to irradiance loss and electrical mismatch between cells. These effects alter the current voltage characteristics, reducing the short circuit current and the maximum power point voltage, and may lead to the appearance of multiple power maxima [8]. In addition, irradiance fluctuations caused by moving clouds, together with panel temperature and wind conditions, introduce dynamic effects that are relevant in real world deployments [9].

This work presents the design and preliminary validation of an embedded solar panel test bench that periodically measures the panel open circuit voltage, together with illuminance, temperature, relative humidity, and atmospheric pressure. Measurements are timestamped, alarm thresholds can be configured, and data is transmitted wirelessly via Bluetooth for real time monitoring. The main contributions of this paper are: i) development of a low power embedded platform for in situ characterization of small photovoltaic panels; ii) integrated acquisition of electrical and environmental variables enabling the analysis of irradiance dynamics; iii) real time wireless monitoring capabilities for field experimentation; and iv) experimental validation under partial and time varying shading conditions.

## II. Proposed platform

The proposed platform (see Fig. 1) is based on the Texas Instruments MSP430G2553 microcontroller (MCU).

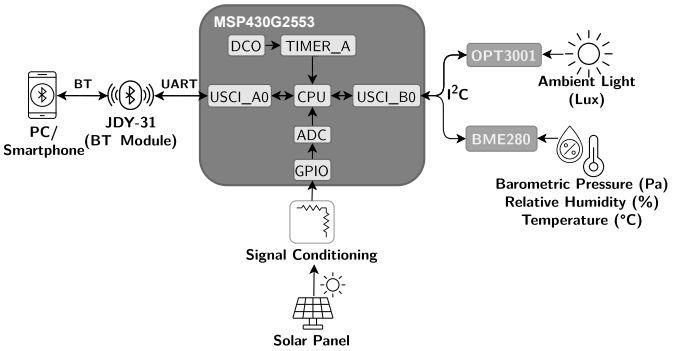



Fig. 1: Block diagram of the proposed platform.

The platform simultaneously records the solar panel open circuit voltage using its Analog to Digital Converter (ADC), while acquiring ambient light levels through the OPT3001 sensor, and ambient humidity, temperature, and pressure through the BME280 sensor. To ensure temporal consistency, a software implemented Real Time Clock (RTC), based on an external 32768 kHz crystal, assigns a timestamp to each acquisition, enabling synchronized analysis of environmental

variations. Data is transmitted to a host device (smartphone or personal computer, PC) via the JDY-31 Bluetooth (BT) module. The acquisition frequency is user configurable. The platform also detects predefined conditions of interest, such as low panel voltage, low ambient light levels, or high ambient temperature, and generates alarms that are transmitted via BT to the host device. Alarm thresholds are defined by the user at system startup through a console interface over the BT link.

### A. Hardware

The MSP430G2553 is integrated into the MSP-EXP430G2 LaunchPad development kit. This MCU features a 16 bit RISC architecture, 16 kB of Flash memory, 512 bytes of RAM, and five low power modes (LPM), along with integrated Universal Serial Communication Interfaces (USCI) supporting UART (Universal Asynchronous Receiver Transmitter) and $I^2C$ (Inter Integrated Circuit) protocols.

Environmental variables are acquired using the BOOSTXL-SENSORS BoosterPack from Texas Instruments, which integrates the OPT3001 and BME280 sensors. Both sensors interface with the MCU via the $I^2C$ bus. The required pull up resistors are provided by the BoosterPack.

The solar panel (SM811K08TF) provides a maximum open-circuit voltage of 5.5 V. Since the MSP430 ADC input range is limited to 3 V, a resistive voltage divider was implemented to scale the panel output. Resistor values of $R_1 = 7.5$ k$\Omega$ and $R_2 = 9.1$ k$\Omega$ were selected to ensure safe ADC operation while maintaining low quiescent current consumption.

Wireless communication is handled by the JDY-31 module, based on the BT 3.0 Serial Port Profile (SPP). The module was integrated following a preliminary validation stage, in which it was interfaced with a PC using an FTDI TTL-232R-PCB adapter. During this phase, the module was configured via AT commands to operate at a baud rate of 115200 bps and with hardware flow control disabled, ensuring compatibility with the MSP430 USCI_A0 UART interface.

### B. Embedded Software

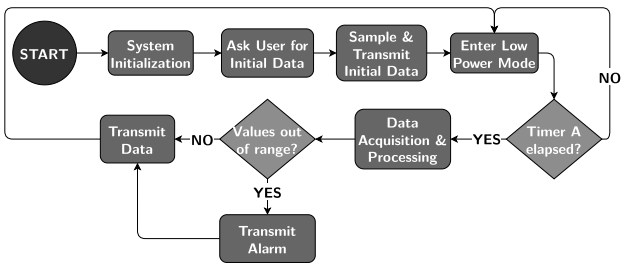



Fig. 2: Flowchart of the platform's embedded software.

The platform operation is summarized in the flowchart shown in Fig. 2. Following initialization and user configuration, the MCU enters LPM. A timer periodically generates events that trigger sensor acquisition according to the configured intervals. The measurements are processed, evaluated against predefined thresholds, and transmitted via BT in a structured format that includes a timestamp and alarm status.

The main program is implemented using a Round-Robin architecture with interrupts [10]. This approach is based on a main loop that processes tasks according to flags set by interrupt service routines. To optimize energy efficiency, the MCU enters Low Power Mode 3 (LPM3) whenever no events are pending. In this mode, the Central Processing Unit (CPU) and most internal clocks are disabled, while the auxiliary clock (ACLK) remains active, significantly reducing current consumption (typical current consumption of 0.7 $\mu$A). Periodic data acquisition is governed by Timer A. Sensor readings, alarm evaluations, and data transmissions are executed only when their corresponding flags are set.

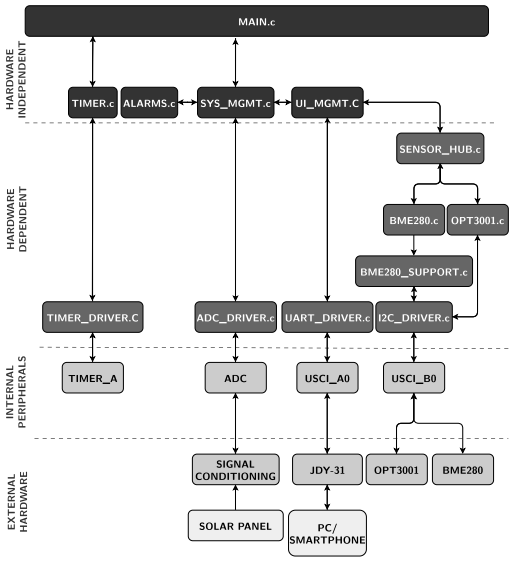



Fig. 3: Software and hardware module diagram.

The embedded software is organized in a modular architecture (Fig. 3) centered around the main application (MAIN.c), which coordinates system operation through modules for timing (TIMER.c), alarms (ALARMS.c), system management (SYS_MGMT.c), and user interface (UI_MGMT.c). Sensor acquisition is handled by SENSOR_HUB.c, interfacing with device specific modules (BME280.c, OPT3001.c). Low level access to MCU peripherals is provided by dedicated drivers (ADC, TIMER, UART, and I2C), enabling a clear separation between hardware dependent and independent layers.

The ADC configuration for solar panel voltage sensing sets the reference voltages to $V_{R+} = V_{CC} = 3.3$ V and $V_{R-} = GND$, with a sample and hold time of 64 clock cycles to ensure signal stability. The ADC is clocked using the internal ADC10OSC, with a nominal frequency of 5 MHz and a prescaler of 4. The UART module is configured using the sub main clock (SMCLK) as the clock source to achieve a stable 115200 bps rate. The $I^2C$ module is also configured using SMCLK, with a clock divider that results in an approximate bus frequency of 100 kHz. Finally, Timer A operates in "Up Mode" using ACLK as the clock source, with a capture compare register value of 8192 defining the base timing interval, and the CPU operates at 1MHz.

## III. Experimental results

### A. Independent validation of sensing subsystems

The experimental phase included independent qualitative validation of all sensing subsystems. The OPT3001 and BME280 sensors were evaluated through controlled tests, confirming correct response to variations in illuminance and ambient conditions. In the case of the BME280, measurements across different building levels showed the expected pressure variation with altitude. The solar panel voltage acquisition chain was validated in two stages. First, the MSP430 ADC was characterized using a signal generator (Analog Discovery 2), confirming linearity within the operating range. Then, the complete acquisition path, including the resistive divider, was verified by comparing software estimated voltages with direct multimeter measurements at the panel terminals.

Figure 4 shows the mobile interface used for system interaction. The initial section displays the user prompts corresponding to system configuration, including sampling periods, alarm thresholds, and date and time settings. This is followed by the real time data stream transmitted by the device. Each record includes a timestamp and the measured variables, where L denotes illuminance, T temperature, P atmospheric pressure, H relative humidity, and V the solar panel voltage. Finally, alarm indicators are appended to each entry.




Fig. 4: Screenshot of the cellphone interface.

### B. 8-hours Continuous Monitoring Analysis

To evaluate the platform performance under real world conditions, a continuous 8 hour monitoring session was conducted. The platform was installed in a fixed position, with the sampling period for all sensors set to 15 minutes. The location and orientation were selected to ensure direct solar exposure for the panel during most of the evaluation window (09:30 to 17:30), while keeping the environmental sensors unobstructed to provide accurate ambient measurements throughout the test.

Fig. 5 illustrates the platform response to environmental changes, including cloud cover and intermittent shading from nearby trees. These effects are reflected as abrupt variations in both the panel output voltage and the corresponding illuminance levels, particularly between 12:00 and 13:30.

The relationship between these variables is further analyzed in Fig. 6, which presents the aggregated data from the 8 hour period. The results confirm the expected direct relationship between illuminance and open circuit voltage, with a peak voltage recorded at 14:30. This consistent tracking of environmental and electrical parameters supports the use of the proposed platform as a tool for small photovoltaic panels characterization.

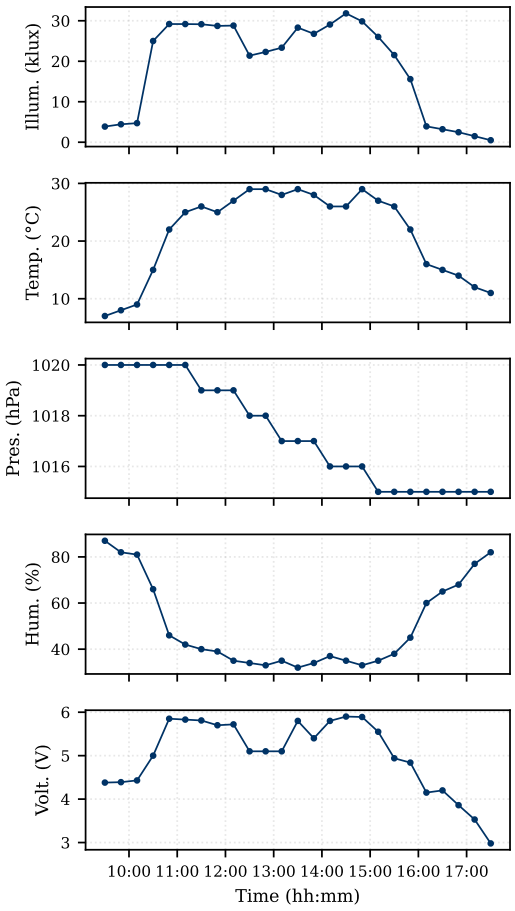



Fig. 5: Monitored variables during the 8-hours trial.

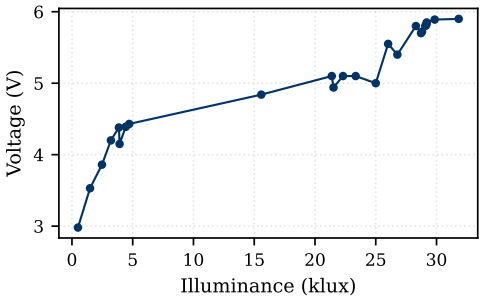



Fig. 6: Correlation plot of panel voltage vs illuminance.

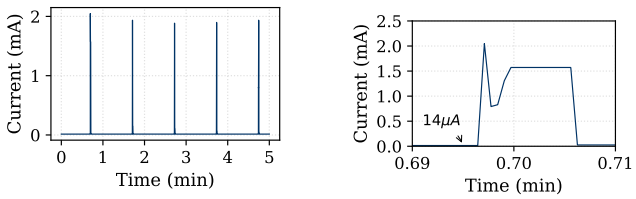
### C. Current Consumption Characterization

A detailed current consumption characterization was conducted to evaluate the platform energy efficiency. For this

purpose, the Texas Instruments EnergyTrace tool was used.

Fig. 7a shows the current consumption profile over a five minute operational period. A repetitive pattern is observed, with current peaks associated with each acquisition event. For this test, the sampling interval was set to 1 minute.

Fig. 7b shows the power profile during a single acquisition cycle. The current increase (approximately 1.5 mA) is caused by the activation of the BME280, the OPT3001, and the internal ADC of the MCU, including its voltage reference. Additionally, the CPU exits low power mode to process and format the acquired data. The system remains active for approximately 0.01 minutes per cycle and then returns to LPM3, reducing the current consumption to approximately 14 $\mu$A. Since the system spends most of its operating time in sleep mode, the estimated average current consumption is approximately 28.8 $\mu$A for a sampling rate of one acquisition per minute.




(a) Complete 5-min profile. (b) Detailed acquisition event.

Fig. 7: platform power profile.

To further analyze the measured average current consumption, two additional test were conducted to isolate the contribution of each sensor group. In the first configuration, the OPT3001 and the ADC were sampled once per minute, while the BME280 sampling period was set to 1 hour. In the second configuration, these sampling periods were inverted. In both cases, the measured current also included the contribution of the CPU, UART, and $I^2C$ modules. The measured average currents were 18.7 $\mu$A for the configuration including the OPT3001 and ADC, 18.8 $\mu$A for the configuration including the BME280, and 29.6 $\mu$A for the complete system. Table I summarizes the estimated contribution of each subsystem to the average current consumption. The obtained values are consistent with the previously measured average current consumption of approximately 28.8 $\mu$A, supporting the validity of the proposed measurement methodology.

TABLE I: Average current consumption

| Component Group | Average Current ($\mu$A) |
|---|---|
| OPT3001 + ADC | 10.8 |
| BME280 | 10.9 |
| CPU + UART + $I^2C$ | 7.9 |
| Total | 29.6 |

It should be noted that this characterization excludes the JDY-31 module, since it is powered from the 5 V rail while EnergyTrace monitors only the 3.3 V domain. According to the manufacturer datasheet, the module adds approximately 7.3 mA during operation. Unlike the MCU and sensors, the JDY-31 lacks low power modes, resulting in a nearly constant current draw. Replacing this module with a Bluetooth Low Energy (BLE) solution could reduce communication-related power consumption by more than 95%.

## IV. Conclusions

This work presented the design and preliminary validation of an ultra low power embedded test bench for the characterization of small photovoltaic panels for IoT applications under realistic operating conditions. The proposed platform integrates electrical and environmental sensing capabilities, enabling the simultaneous acquisition of open circuit voltage, illuminance, temperature, humidity, and pressure. Experimental results demonstrated the ability of the platform to capture the impact of environmental variability, including partial and time varying shading, on small photovoltaic panels behavior. In addition, current consumption analysis confirmed the effectiveness of the low power design, with a baseline consumption in the order of tens of microamperes.

The current implementation relies on a conventional BT communication module, and the adoption of BLE would further reduce power consumption, enabling long term autonomous operation. In addtion, future work will explore alternative low power wireless communication technologies and the integration of current sensing capabilities for direct harvested power estimation and enhanced photovoltaic panel characterization.

## Acknowledgment

This work was partially funded by CSIC (Comisión Sectorial de Investigación Científica, Udelar) and IEEE Circuits and Systems Society (Uruguayan Chapter).

## References

[1] K. Martinez, J. K. Hart, and R. Ong, "Environmental sensor networks," *IEEE Computer*, vol. 37, no. 8, pp. 50–56, 2004.

[2] J. Azziz et al., "Selective audio recording device for wildlife research using embedded machine learning," in *2025 IEEE Latin Conference on IoT (LCIoT)*, 2025, pp. 65–68.

[3] N. Barreto, J. Oreggioni, and L. Steinfeld, "Energy harvesting and storage solutions for low-power iot devices in livestock industry," in *IEEE 15th Latin America Symposium on Circuits and Systems (LASCAS)*, 2024.

[4] V. Cabrera et al., "Harnessing technology for livestock research: An online sheep behavior monitoring system," *IEEE Transactions on AgriFood Electronics*, vol. 2, no. 2, pp. 306–313, 2024.

[5] P. Castro-Lisboa, J. Oreggioni, and E. Machado, "System and device for monitoring the reproductive activity of animals," Mar. 2020, US Patent 10,575,501.

[6] H. Elahi et al., "Energy harvesting towards self-powered iot devices," *Energies*, vol. 13, no. 21, 2020. [Online]. Available: https://www.mdpi.com/1996-1073/13/21/5528

[7] H. Oufettoul et al., "Comparative performance analysis of pv module positions in a solar pv array under partial shading conditions," *IEEE Access*, vol. 11, pp. 12 176–12 194, 2023.

[8] A. Ibrahim, et al., "Pv maximum power-point tracking using modified particle swarm optimization under partial shading conditions," *IEEE Chinese Journal of Electrical Engineering*, vol. 6, no. 4, pp. 106–121, 2020.

[9] E. Kaplani and S. Kaplanis, "Thermal modelling and experimental assessment of the dependence of pv module temperature on wind velocity and direction, module orientation and inclination," *Solar Energy*, vol. 107, pp. 443–460, 2014.

[10] D. E. Simon, "Survey of software architectures," in *An Embedded Software Primer*. Addison-Wesley Professional, 1999, pp. 115–136.

# Diseño y desarrollo de sistema embebido para estaciones de monitoreo de lagos

Luciano A. Rodriguez*, Mauro G. Gómez*†, Carlos A. Vitulich*†,
Manuel A. Ruiz Tatur*, Andrés Rodriguez*‡
*Laboratorio de Hidráulica, FCEFyN, UNC, Córdoba, Argentina
†CIA, DGIyD, Fuerza Aérea Argentina, Córdoba, Argentina
‡IDIT, CONICET, UNC, Córdoba, Argentina

***Resumen*—Este trabajo presenta el diseño y desarrollo del firmware para una Estación de Monitoreo de Lagos (EML), orientada a la adquisición de variables meteorológicas y de calidad de agua en entornos remotos. El sistema se basa en un microcontrolador ARM Cortex-M4 y utiliza el sistema operativo de tiempo real FreeRTOS para gestionar tareas críticas de sensado, almacenamiento y comunicación. La estación integra sensores meteorológicos (radiación, viento, temperatura y humedad) y de calidad de agua (pH, conductividad eléctrica, nivel, entre otros) utilizando el protocolo de comunicación Modbus RTU. Los datos recolectados se almacenan localmente en una memoria SD como respaldo y se transmiten a un servidor remoto mediante tecnología LTE-M/NB-IoT utilizando módulos SIMCom (SIM7000/7070G). Además, se monitorean tanto el consumo como las fuentes de energía del sistema. Los resultados ganan robustez manteniendo continuidad operativa e integridad de los datos, debido a las constantes mejoras para obtener un sistema confiable a mediano y largo plazo sin intervención técnica física.**

***Palabras clave*—firmware, EML, FreeRTOS, Modbus RTU, LTE-M/NB-IoT.**

## I. Introducción

La obtención de parámetros hidrometeorológicos y de calidad de agua in situ es de suma importancia para desarrollar modelos que permitan caracterizar la evolución de los embalses de la provincia de Córdoba, algunos de los cuales presentan eutrofización [3].

En el Laboratorio de Hidráulica de la Universidad Nacional de Córdoba se encuentra vigente un proyecto de desarrollo e instalación de Estaciones de Monitoreo de Lagos (EML) de la Provincia de Córdoba [1][2]. Las EML desarrolladas se estructuran sobre una plataforma flotante tubular con su sistema de anclaje y señalización, donde se instalan la instrumentación, la electrónica asociada y la planta de energía.

Cada estación recolecta, almacena de forma local y transmite las variables de forma inalámbrica hacia el servidor remoto. Una vista de una EML instalada en campo se ilustra en la Fig. 1. Actualmente se encuentran operativas estaciones en la provincia de Córdoba (San Roque, Los Molinos, Embalse Ministro Pistarini, La Viña, Cruz del Eje) y Salta (Cabra Corral).

## II. Arquitectura funcional del sistema

La distribución de los bloques que componen el hardware y la topología general de la estación se describen en el esquema jerárquico de la Fig. 2.

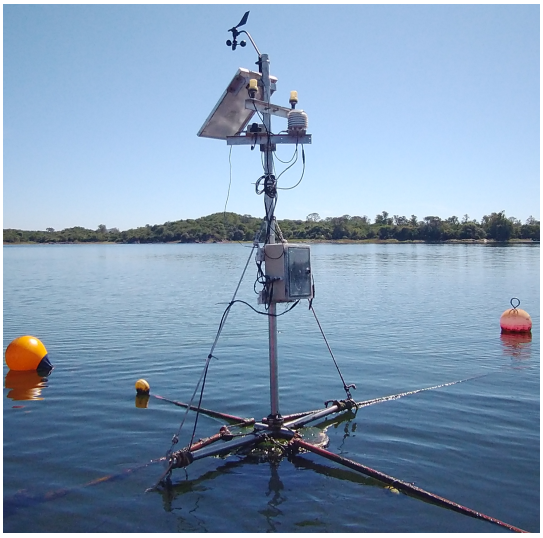


Figura 1. Estación de monitoreo de lagos instalada.




Figura 2. Arquitectura funcional del sistema.

### *II-A. Sensores meteorológicos*

La instrumentación superior, orientada al registro meteorológico en superficie, se agrupa conceptualmente como se visualiza en la Fig. 3, y abarca los siguientes dispositivos:

- **Sensor de Humedad, Temperatura y Presión (HTP):** Se utiliza un módulo integrado que combina un sensor digital SHTC3 para humedad relativa, un sensor barométrico BMP280 para presión atmosférica y un termistor PS102J2 para temperatura del aire. Se refugia de la radiación solar mediante una torre de platos impresa en 3D, y de la condensación con un capuchón de tela hidrófuga, disminuyendo errores de medición.
- **Anemómetro Davis 6410:** Mide dirección y velocidad de

viento mediante cazoletas y veleta, con un amplio rango de operación y una precisión adecuada para monitoreo ambiental.
- **Sensor de radiación solar directa Davis:** Convierte intensidad de la luz solar en una señal eléctrica proporcional.

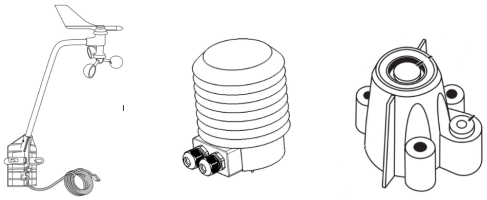


Figura 3. Sensores meteorológicos.

### II-B. *Sensores de calidad de agua*

- **Cadena de temperatura de la columna de agua:** Cadena de múltiples nodos interconectados unidos a una soga con lastre en extremo inferior. Cada nodo posee un termistor con el objetivo de detectar la estratificación térmica del lago.
- **Sensor de pH:** Una sonda S628H con un diseño robusto para la inmersión total. Cuenta con una adaptación electrónica asociada con un amplificador de ultra-alta impedancia, digitalización y comunicación para la EML.
- **Sensor de conductividad eléctrica:** RK500-13, electrodo de acero inoxidable para inmersión total.
- **Sensor de profundidad:** Nos permite la medición del nivel a partir de la presión hidrostática. Por eso cuenta con un sensor de presión integrado, y una adaptación para la adquisición y comunicación de la medición.

### II-C. *Sistemas electrónicos*

- **Suministro de energía:** Las fuentes de energía son un panel solar (20W), y de manera secundaria, una batería de gel (12V / 9Ah). Realiza una regulación de carga y descarga de la fuente secundaria, además de una protección de la misma y regulación para el resto del sistema. Mediante el mismo protocolo que los sensores, se monitorean voltajes y corrientes de las fuentes de energía y consumo.
- **Computadora a bordo:** Es el núcleo principal que coordina la adquisición, almacenamiento y transmisión de los datos. Además nos da una puerta de enlace para *debug* y *testing*.
- **Interfaces:** Mediante esta interfaz se vinculan los sensores de todos los puertos con la computadora a bordo, además de la gestión de la energía de los mismos.
- **Módulo de comunicaciones:** A partir de un módulo SIMCom (SIM7070G), permite la transmisión de datos a un servidor remoto, mediante un protocolo MQTTS y red celular.

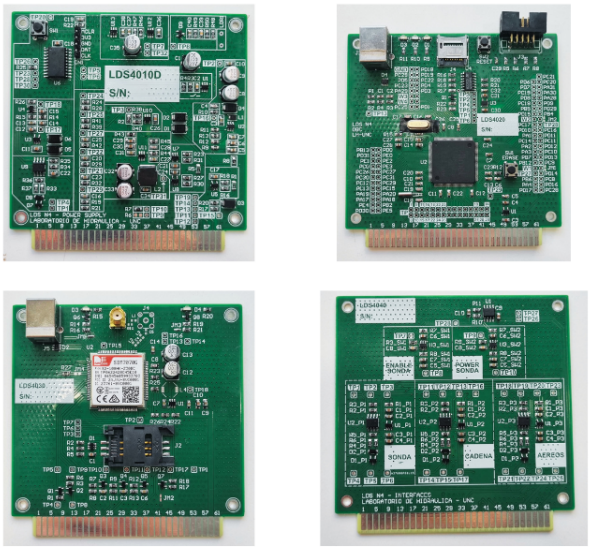



Figura 4. Placas electrónicas versión 4.

El diseño electrónico actual Fig. 4 presenta una evolución tecnológica respecto a la versión previa del proyecto (EML-V3) [1][2] donde se estructuraba sobre solo dos placas: una de control de energía y otra ”master”que concentraba la totalidad de las funciones (procesamiento, interfaces, almacenamiento y comunicaciones). Con el objetivo de mitigar fallas de acoplamiento y optimizar las tareas de mantenimiento y diagnóstico constructivo tanto *in situ* como en laboratorio, la versión 4 desacopla el hardware distribuyendo el sistema en los cuatro pilares independientes descritos anteriormente, incrementando la modularidad y robustez global de la plataforma electrónica.

### II-D. *Servidor y Gestión de Datos*

La infraestructura de *backend* se compone de un servidor principal y una unidad de respaldo alojados en el laboratorio, garantizando la disponibilidad del servicio. El proceso de gestión de datos se divide en tres etapas:
- **Recepción:** El servidor actúa como un *broker* MQTT (utilizando TLS para seguridad), al cual las estaciones se conectan para publicar los paquetes de datos crudos en tópicos específicos.
- **Procesamiento:** Una vez recibidos, los datos pasan por una capa de adaptación donde se aplican las funciones de transferencia y calibración de laboratorio. Esto permite convertir las lecturas en unidades físicas (ej. 180°C, hPa, mg/L) y mitigar los errores sistemáticos de medición de cada variable ambiental de forma centralizada.
- **Almacenamiento y Visualización:** La información procesada se ingesta en una base de datos de series temporales (*InfluxDB*), optimizada para el manejo de grandes volúmenes de datos ambientales. Finalmente, los datos son visualizados mediante tableros dinámicos en *Grafana* Fig. 5, permitiendo el monitoreo remoto en tiempo real, el análisis histórico de los parámetros de los lagos y la gestión operativa ante banderas de falla en las estaciones.

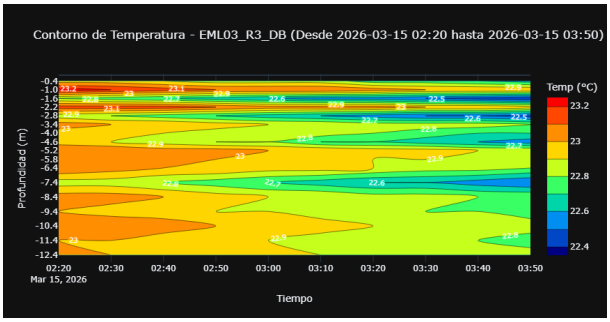



Figura 5. Visualización a partir de los datos ingestados.

Esta infraestructura de *backend* representa una mejora crítica en términos de eficiencia y ciberseguridad respecto a las primeras iteraciones de la red de estaciones. En las versiones anteriores, la transferencia de archivos hacia el servidor se realizaba mediante el protocolo FTP (File Transfer Protocol). La migración hacia una arquitectura basada en el protocolo MQTTS (MQTT sobre TLS) no solo garantiza la confidencialidad e integridad tanto de los datos como la conexión, sino que además optimiza el ancho de banda consumido, adaptándose a las restricciones energéticas y de conectividad de la tecnología LTE-M/NB-IoT.

## III. FIRMWARE

El sistema presenta una arquitectura donde el desarrollo del firmware se divide en dos componentes principales: el firmware de la Computadora de a Bordo (OBC, por sus siglas en inglés) y el firmware propio de cada nodo sensor.

### *III-A. Firmware Computadora de a Bordo*

El firmware principal fue desarrollado sobre el ecosistema de FreeRTOS, permitiendo una arquitectura modular y determinista. La elección de un RTOS responde a la necesidad de gestionar múltiples periféricos de forma concurrente sin afectar la precisión temporal del sensado. El microcontrolador utilizado en la OBC es el modelo ATSAM4E16E de Microchip (ARM Cortex-M4).

Se definieron cuatro hilos de ejecución principales:

- **Tarea de sensado:** Actúa como maestro en el bus Modbus RTU. Realiza un escaneo dinámico de los nodos esclavos para identificar los sensores presentes en los puertos de la estación. Una vez detectados, ejecuta la lectura de los registros, encapsula los parámetros en una estructura de datos y la envía hacia la tarea de almacenamiento mediante una cola de mensajes (*Queue*) de FreeRTOS.
- **Tarea de guardado:** Es la encargada de garantizar la persistencia local. Recibe la estructura de datos de la cola y realiza una doble escritura en la tarjeta microSD utilizando la biblioteca FatFS: un archivo en formato *.csv* para facilitar la auditoría manual y lectura directa en campo, y un archivo de tipo binario (*.bin*). El uso de archivos binarios permite el manejo de índices de lectura y escritura, funcionando como una cola persistente que optimiza la memoria RAM y mitiga significativamente el riesgo de pérdida de información ante eventuales desbordamientos de la pila de transmisión o reinicios imprevistos del sistema.
- **Tarea de comunicación:** Gestióna el módulo SIM7070G y la pila de protocolos de red. Esta tarea indexa el archivo binario buscando registros pendientes de envío. La transmisión se realiza mediante el protocolo MQTTS, asegurando el cifrado de la información. Una vez confirmada la recepción por el *broker*, la tarea actualiza los índices en la SD. Adicionalmente, tras cada envío exitoso, habilita una ventana de escucha en un tópico de comandos, permitiendo la ejecución de configuraciones y diagnóstico a través de una conexión remota.
- **Tarea de comandos CLI:** Implementa una interfaz de línea de comandos utilizando la biblioteca *FreeRTOS-CLI* a través de un puerto virtual serie (*USB CDC*). Permite la interacción directa para la ejecución de funciones aisladas de diagnóstico o el lanzamiento de ciclos de prueba completos. Asimismo, actúa como un monitor en tiempo real (*logger*), permitiendo al operador visualizar el flujo de información de los sensores y el estado del módem directamente sobre el bus USB durante el mantenimiento físico en la estación.

### *III-B. Gestión de Periféricos y Drivers de Bajo Nivel*

Para dar soporte a las tareas del sistema operativo, se desarrollaron capas de abstracción de hardware y rutinas de servicio de interrupción optimizadas:

- **Protocolo Modbus sobre la OBC:** Se utilizan los periféricos UART/USART del microcontrolador. Las rutinas de dichos periféricos actúan sobre una librería generalizada de Modbus, permitiendo desplegar el protocolo de manera transparente sobre cualquier puerto físico de la placa de expansión o *backplane*.
- **Tiempos de rutinas:** Para llevar la base de tiempo de la estación de manera estricta se utilizan en conjunto el módulo RTC (Reloj de Tiempo Real) y el RTT (Temporizador de Tiempo Real). El RTT asiste en la periodicidad fina de los muestreos, mientras que el RTC permite la gestión de energía general del sistema en intervalos prolongados.
- **Señalización y Balizas:** Se gestiona el encendido nocturno de las balizas de seguridad mediante lógica redundante: sincronización horaria por red (*UNIX timestamp*) o, ante la falta de señal celular, mediante umbrales críticos de lectura local del panel solar y radiación.

### *III-C. Detección Dinámica de Sensores Modbus*

Una particularidad crítica del firmware es la capacidad de descubrir de forma activa qué sensores están conectados en los puertos físicos al iniciar o tras una falla del bus. Al comenzar el ciclo de sensado, el firmware recorre las direcciones de esclavos posibles y solicita su ID único (por ejemplo, el anemómetro - ID 500). Este mecanismo resuelve el problema de la intercambiabilidad en entornos remotos. Ante una avería, permite la sustitución física directa de cualquier

sensor en campo sin interrumpir el servicio ni requerir una reprogramación del firmware de la OBC.

Una vez referenciado y validado el sensor, se asocia dinámicamente a una estructura en memoria RAM que mapea las funciones de sus respectivos archivos de cabecera (*headers*). Esto permite la escalabilidad del sistema: un nuevo sensor puede ser admitido en la estación sin necesidad de recompilar el firmware del *datalogger*, requiriendo únicamente un ID único asignado y una estructura de registros Modbus compatible.

### *III-D. Estructura de la Memoria Flash y Parametrización*

La memoria flash interna del ATSAM4E16E se estructuró estratégicamente para dividir el mapa en múltiples sectores funcionales. Esto permite alojar un binario secundario para soportar la funcionalidad de actualizaciones de firmware remotas del tipo *safe-upgrade* (FOTA), asegurando la posibilidad de revertir al binario base ante fallas en la transferencia.

Adicionalmente, se reservó un sector dedicado a banderas (*flags*) parametrizables individuales para cada estación. Mediante comandos CLI o vía MQTTS, es posible alterar dinámicamente los intervalos de muestreo, punteros de memoria y parámetros de APN celular, persistiendo los cambios de forma local.

### *III-E. Estandarización de Interfaces en Nodos Sensores*

Para simplificar el mantenimiento y desacoplar el hardware del sensor del código de la OBC, se definió que cada nodo sensor de desarrollo propio implemente nativamente el protocolo *Modbus RTU sobre la capa física RS-485*. Cada nodo embebido corre un firmware interno dedicado a la adquisición cruda del transductor local (vías ADC, I2C, etc.), aplica su calibración y expone la información en un mapa de registros estandarizado:

- ***Input Registers:*** Alojan los valores físicos resultantes del sensado que no sufren alteraciones (ej. temperatura de columna, pH).
- ***Coils:*** Actúan como banderas booleanas de comandos para disparar acciones inmediatas, como forzar un ciclo de muestreo.
- ***Holding Registers:*** Almacenan variables de configuración locales (calibración del cero, tiempos de integración) que pueden ser modificadas por los operarios mediante la estación sin necesidad de reprogramar el nodo sensor individual.

### *III-F. Evaluación de Consumo y Gestión de Errores*

Los ensayos sobre el hardware de la versión 4, alimentado a 12.5 VDC con la batería, determinaron un consumo de entre 20 y 30 mA en modo activo con la cadena de temperatura, sensores meteorológicos y un sensor de calidad de agua operando. En inactividad profunda, el firmware ejecuta rutinas de desconfiguración (*deinit*) de interfaces y pasa al modo *Deep Power Down*, reduciendo el consumo total a un rango de entre 3 y 5 mA en *sleep*, con un despertar por *wakeup* de los módulos RTC/RTT.

Respecto a fallas, el firmware prioriza la continuidad del ciclo de adquisición de manera simplificada, sin aislamiento automatizado en el bus. Ante una avería o desconexión en campo, las anomalías por pérdida de tramas, desbordes o lecturas fuera de escala se evalúan de forma pasiva en la ingesta del *backend* o mediante la auditoría del personal especializado. Esto delega el diagnóstico fino al procesamiento centralizado del servidor, manteniendo un firmware liviano en la OBC.

### *III-G. Entornos de Desarrollo y DevOps*

La gestión del código se realiza en *GitHub Enterprise* mediante repositorios independientes para la OBC, nodos sensores y *scripts* de testeo. El sistema de construcción utiliza *Makefiles* optimizados para Linux y Windows, basado en la *toolchain* de GNU Arm. Para asegurar la robustez antes del despliegue en campo, se implementó un *pipeline* de Integración y Despliegue Continuo (CI/CD) que ejecuta compilaciones automatizadas y pruebas de tipo *Hardware-in-the-Loop* (HIL) sobre un banco de ensayos en el laboratorio, validando la estabilidad del *kernel* de FreeRTOS sobre el hardware real.

## IV. Conclusiones

La madurez tecnológica alcanzada en la versión 4 de la EML consolida una plataforma robusta y eficiente para el monitoreo ambiental continuo en entornos remotos. Las principales contribuciones de este trabajo se centran en el desacoplamiento y la modularidad del firmware basado en FreeRTOS, lo que simplifica la mantenibilidad y el diagnóstico del sistema. Asimismo, la migración hacia una arquitectura basada en el protocolo MQTTS optimizó el uso del ancho de banda bajo redes LTE-M/NB-IoT garantizando la seguridad de la información. Finalmente, la implementación de estrategias avanzadas de gestión energética redujo el consumo en estado de inactividad profunda a rangos de entre 3 y 5 mA, asegurando la autonomía y disponibilidad del sistema a largo plazo sin necesidad de intervención técnica física.

## Referencias

[1] A. Rodríguez, M. G. Gómez, C. A. Vitulich, P. Ruiz Diaz, "Estaciones de Monitoreo de Lagos en Córdoba, Argentina: Avances en el Desarrollo," en *Actas del XXVIII Congreso Nacional del Agua*, CABA, Argentina, 2023, pp. 274–275. [En línea]. Disponible en: https://conagua.ina.gob.ar/archivos/Libro-resumenes-CONAGUA2023.pdf

[2] A. Rodríguez *et al.*, "Red de estaciones para medición de parámetros hidrometeorológicos en embalses de Córdoba, Argentina," en *2020 IEEE Congreso Bienal de Argentina (ARGENCON)*, Resistencia, Argentina, 2020, pp. 1–6, doi: 10.1109/ARGENCON49523.2020.9505437.

[3] Pérez, Gabriel y Restrepo, John. (2008). Fundamentos de limnología Neotropical.

[4] FreeRTOS, "FreeRTOS Real-Time Operating System for Microcontrollers," *FreeRTOS Official Website*, 2026. [En línea]. Disponible en: https://www.freertos.org. [Accedido: 24-Jun-2026].

[5] Microchip Technology, "ATSAM4E16E: SAM4E Series 32-bit ARM Cortex-M4 Microcontroller Datasheet," *Microchip Technology Inc.*, Documento DS60001308B, 2014.

[6] SIMCom Wireless Solutions, "SIM7070G Hardware Design: LPWA Module Product Technical Specification," *SIMCom Inc.*, v1.02, 2020.

Categoría

# Foro Tecnológico

Inteligencia Artificial en SE

Área Temática

ISBN 978-631-90145-4-9 - Congreso Argentino de Sistemas Embebidos CASE 2026 - 6 y 7 de agosto

# Diseño e Implementación de un Núcleo VHDL Genérico para Redes Neuronales Artificiales en SoC

Juan Pablo Vecchio
Facultad de Ciencias Exactas, Ingeniería y Agrimensura – Centro de Estudios Interdisciplinarios
Universidad Nacional de Rosario
jvecchio@fceia.unr.edu.ar

Rosa Corti
Facultad de Ciencias Exactas, Ingeniería y Agrimensura – Centro de Estudios Interdisciplinarios
Universidad Nacional de Rosario
rcorti@fceia.unr.edu.ar

Roberto Martínez
Facultad de Ciencias Exactas, Ingeniería y Agrimensura – Centro de Estudios Interdisciplinarios
Universidad Nacional de Rosario
romamar@fceia.unr.edu.ar

***Resumen*—Este artículo presenta el diseño de un núcleo VHDL genérico para la implementación eficiente de Redes Neuronales Artificiales (RNAs) de tipo perceptrón multicapa en dispositivos SoC (System-on-Chip). El flujo de desarrollo combina el entrenamiento del modelo de RNA en la nube utilizando PyTorch, con la implementación en el borde mediante un núcleo parametrizable. Se destacan las ventajas del enfoque a nivel de RTL, que optimiza el rendimiento y la reutilización del diseño, subrayando la importancia de la combinación de procesadores escalares y lógica reconfigurable para aplicaciones en inteligencia artificial. Finalmente, se aborda un caso de estudio considerando la utilización de recursos y la velocidad de procesamiento de la red implementada en lógica reconfigurable. Esto permite analizar los retardos introducidos por los procesadores escalares en el sistema y comparar la eficiencia respecto de la implementación del mismo modelo de red en una GPU (Graphics Processing Unit).**

***Palabras clave— Codiseño nube-borde, FPGA, PyTorch, Perceptrón Multicapa, Cuantización***

## I. INTRODUCCIÓN

La evolución conjunta de la computación en la nube y la computación en el borde ha impulsado nuevas aplicaciones de inteligencia artificial con requisitos estrictos de latencia y procesamiento en tiempo real, especialmente en sistemas embebidos cercanos a sensores [1].

El aumento de aplicaciones de IA en el borde impulsa el uso de plataformas neuromórficas eficientes en energía y área, capaces de aprender y adaptarse localmente sin depender de la nube [2] [3]. Simultáneamente, el campo de la IA ha experimentado progresos notables, las Redes Neuronales Artificiales (RNAs) han demostrado un gran potencial en aplicaciones de inteligencia artificial proporcionando algoritmos de alta precisión para diversas aplicaciones. Sin embargo, muchos de estos algoritmos están diseñados para ejecutarse en plataformas informáticas convencionales y no son directamente compatibles con el hardware neuromórfico [4]. Los estudios han indicado que la implementación de redes neuronales profundas puramente en software puede encontrar restricciones significativas en su implementación de hardware, específicamente en la utilización de parámetros críticos como memoria, recursos computacionales y energía. La implementación de RNAs en dispositivos SoC, que combinan procesadores escalares y lógica reconfigurable, ofrece una solución ideal para aprovechar las fortalezas de ambas tecnologías. En este sentido, los entornos de desarrollo que incluyen herramientas que permiten aplicar conceptos de codiseño, poseen sintetizadores que abordan las limitaciones mencionadas, aprovechando las ventajas de la implementación en hardware, como la segmentación y el paralelismo, para mejorar el rendimiento de los FPGA (Field-Programmable Gate Arrays) [5]. Estos dispositivos de lógica reconfigurable exhiben una gran flexibilidad, a diferencia de los ASICs (Application-Specific Integrated Circuits), que los hace muy adecuados para adaptarse a la evolución de los modelos de redes neuronales, facilitando la creación de prototipos y la implementación de nuevos diseños. Además, presentan bajos valores de latencia y sus requerimientos de potencia son menores a los de los GPU y CPU (Central Processing Unit) [6].

Es importante destacar que el procesamiento en la nube y en el borde no son excluyentes, por el contrario, son tecnologías que pueden complementarse para lograr aplicaciones más eficientes, escalables y con mejor capacidad de respuesta; como concluyen los autores de [1] luego de realizar un extenso análisis de situación y de perspectivas a futuro en la academia y en la industria. En este sentido, el trabajo que aquí se presenta consiste en una metodología de desarrollo que combina ambos procesamientos, y se enfoca en implementar en un dispositivo SoC diversos modelos de RNAs utilizando un núcleo parametrizable que se adapta a las necesidades de los usuarios en distintas aplicaciones. El enfoque adoptado disminuye el tiempo de desarrollo para cada aplicación específica y permite dotarlas de la velocidad y eficiencia inherentes a la lógica reconfigurable. Diversos artículos reportan mejoras significativas al implementar RNAs y arquitecturas neuromórficas sobre FPGA, destacando ventajas en latencia, eficiencia energética y flexibilidad [7], [8], [9], [10]. En esta línea, consideramos que las principales contribuciones de este trabajo son: (i) un núcleo VHDL parametrizable que permite instanciar mediante genéricos, MLPs de distinta cantidad de entradas, capas, neuronas por capa y funciones de activación, sin reescribir el diseño “a medida”; (ii) un flujo de desarrollo nube–borde que toma un modelo entrenado en PyTorch, lo cuantiza y lo despliega sobre un SoC combinando lógica reconfigurable y procesadores escalares. La publicación se organiza de la siguiente manera, en la sección II se detalla la metodología aplicada y en la sección III se describe el núcleo VHDL que se ha desarrollado. Finalmente, la sección IV describe los resultados obtenidos al implementar un caso de estudio y resume las conclusiones alcanzadas.

## II. METODOLOGÍA

El flujo de trabajo que se eligió para desarrollar e implementar una red neuronal se divide en dos grandes bloques: la nube, donde se selecciona la arquitectura de red, se realiza el entrenamiento y evaluación de la cuantización

de los parámetros e hiperparámetros; y la implementación en el borde basada en un SoC que combina lógica reconfigurable con procesadores escalares. Para ello se traduce la red virtualizada a un lenguaje de descripción de hardware y se desarrolla un driver en lenguaje C que facilita el intercambio de información de un procesador con la red implementada en el FPGA.

El diseño del núcleo VHDL se enfocó en dotarlo de las siguientes características: modularidad y adaptabilidad, conectividad simple con otros módulos en un SoC y velocidad para las operaciones críticas de una red neuronal.

## III. Descripción del Núcleo VHDL

### *A. Características del núcleo VHDL*

Se orientó el diseño del núcleo VHDL para obtener un módulo que presentara las siguientes características:

- *Módulo combinacional:* Se aprovecha el paralelismo inherente al hardware para lograr velocidad. La operatoria se despliega "espacialmente" en el chip.
- *Módulo genérico:* Los parámetros permiten al usuario personalizar el núcleo para implementar RNAs que se adapten al modelo definido en PyTorch.
- *Módulo que trabaja como esclavo de un procesador:* Intercambia información con el procesador usando el sistema de buses en el chip. Para ello se lo empaqueta como un esclavo AXI con el fin de conectarlo con los demás módulos de la plataforma HW.

### *B. Estructura de la red*

El núcleo VHDL se utiliza para implementar una RNA de tipo MLP, una red unidireccional que consta de una capa de entrada, una cantidad variable de capas ocultas y una capa de salida. Este tipo de RNA se adapta a una gran cantidad de aplicaciones, entre las cuales podemos mencionar regresión y clasificación.

Conceptualmente una red neuronal puede describirse mediante una jerarquía de tres niveles, neuronas artificiales que conforman un número variable de capas y una RNA como estructura superior que las engloba. No obstante, en nuestro desarrollo adoptamos una jerarquía de dos niveles. El número de neuronas que constituyen cada capa es variable, pero todas ellas trabajan con la misma función de activación y despliegan la misma operatoria sobre sus entradas. Además, las salidas de las neuronas de cada capa ingresan como entradas a todas las neuronas de la siguiente capa. Por lo tanto, se utilizó una jerarquía en dos niveles: La red como nivel superior de la estructura, donde se instancian e interconectan las capas de neuronas que constituyen el nivel inferior, como se puede apreciar en la Fig. 1. Este esquema es más simple, lo que facilita la implementación, aporta legibilidad y resulta suficiente para incorporar todos los elementos necesarios para lograr un núcleo genérico, que se adapte a las necesidades de las distintas aplicaciones.

### *C. Parametrización del diseño*

Una característica clave para el diseño del núcleo VHDL se refiere a su adaptabilidad, con el fin de evitar el trabajo "a medida" en la implementación de los modelos de RNAs. Se incorporaron al esquema jerárquico un conjunto de parámetros para personalizar el diseño.

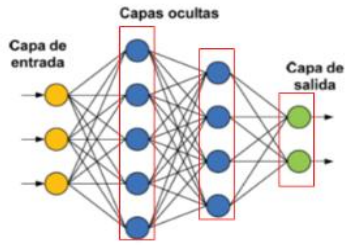



Fig.1. Estructura jerárquica en dos niveles de un MLP

La mayoría son parámetros de tamaño y se refieren a la cantidad de entradas a la red, cantidad de neuronas de cada capa, número de bits de los datos de entrada y número de bits de pesos y sesgos. Además, se agregó un parámetro de características, que permite definir la función de activación de las neuronas de cada capa de acuerdo con las necesidades del modelo a implementar. En el caso de estudio se utilizó la función ReLU para la capa oculta y la función lineal para la capa de salida. La sección de código que se presenta a continuación, muestra la definición de los parámetros mencionados.

```
generic (N: integer;  -- Nro. de entradas a la capa
         M: integer; -- Cantidad de Neuronas de la capa
         funcion:F_Activacion;  -- Funcion de activacion para la capa
         N_bits_ent:integer;  -- Nro bits de cada entrada a la capa
         N_bits_peso:integer;   -- Nro bits de cada peso (Wij)
         N_bits_u:integer);  -- Nro bits de cada sesgo
```

### *D. Representación de datos – Operatoria*

El tipo de datos que se selecciona tiene un importante impacto en la complejidad de la operatoria y por lo tanto en la cantidad de recursos del FPGA que se necesitan al implementar el circuito. Esta selección depende de las características de las variables propias de la aplicación que se aborda, de los valores de los parámetros obtenidos a partir del entrenamiento con PyTorch y de los errores admisibles para los resultados que entregará la red. Se hace necesario realizar un balance entre precisión de las conclusiones y posibilidades de implementación en el dispositivo elegido. La cuantización es una estrategia de optimización, enfocada en la velocidad de respuesta y la precisión de los resultados. Los pesos y sesgos que se obtienen a partir del entrenamiento de la red mediante PyTorch son datos en punto flotante, que la mayoría de los autores desaconseja para implementaciones en FPGA, por la complejidad de la operatoria que redunda en una elevada utilización de recursos [11]. Por estos motivos, se aplicó cuantización lineal a los parámetros de los modelos obtenidos con el procesamiento en la nube, trabajando con datos en punto fijo.

El SoC utilizado en este desarrollo ofrece la ventaja de poder delegar determinadas tareas de procesamiento en los Cortex-A9 que incluye; de esta manera se evita que se afecte la velocidad de trabajo de la red neuronal implementada en lógica reconfigurable. Por lo tanto, se optó por representar los datos con los que opera el núcleo VHDL como números enteros (en complemento a dos), mientras que el proceso de conversión a representaciones con parte entera y fraccionaria se asignó a uno de los procesadores mencionados.

Se ensayaron modelos que trabajan con datos en punto fijo utilizando el mismo núcleo para la implementación de la red. La cantidad de bits para parte entera y parte fraccionaria se definió para cada caso buscando el balance mencionado entre complejidad y precisión.

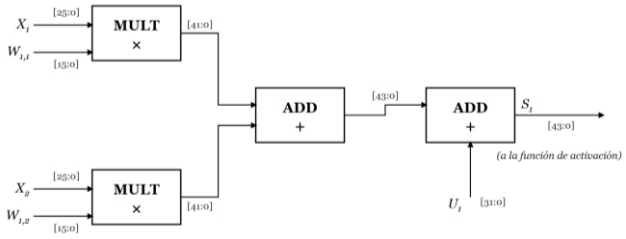



Fig. 2. Implementación de una neurona con dos entradas.

Las neuronas de cada capa realizan la misma operatoria sobre las entradas, pero el tamaño y cantidad de datos sobre los que operan es variable, dependiendo del modelo de red que se implemente. Para abordar esta situación, los datos de entrada, los parámetros de la red (pesos y sesgos), y también los resultados obtenidos, se organizan en arreglos VHDL sin restringir, en los cuales el número de ítems que los constituyen y el tamaño de cada uno, se define al momento de la síntesis a partir de los genéricos incorporados al diseño. Las operaciones que realiza cada neurona antes de aplicar la función de activación, corresponden a la expresión (1)

$$S_{z(i)} = \sum_j X_j W_z(i,j) + U_z(i) \quad (1)$$

donde $X_j$ son entradas, $W_z(i,j)$ pesos, $U_z(i)$ sesgos y $S_{z(i)}$ la salida de la neurona (i) en la capa (z).

La Fig. 2 muestra cómo se despliega en el dispositivo la operatoria que realiza cada neurona de la red. En este ejemplo se presenta el caso de una neurona en una capa a la que ingresan dos entradas, y puede apreciarse cómo se incrementa el tamaño de los datos de salida a partir de las operaciones realizadas.

## IV. Resultados

Con el objetivo de evaluar la funcionalidad y flexibilidad del núcleo VHDL propuesto, se seleccionó un caso para su estudio, el cálculo de una regresión no lineal múltiple, aplicación representativa del tipo de tareas comunes en el campo de las redes neuronales. Este caso permite mostrar la capacidad del núcleo para modelar relaciones complejas entre variables continuas. En este sentido, el problema analizado examina la idoneidad del núcleo para aproximar funciones de varias entradas, lo que permite evaluar la precisión numérica de la inferencia, así como su capacidad para modelar relaciones no lineales. La parametrización del diseño del núcleo, facilita la exploración de configuraciones con diferentes profundidades de red y tamaños de capa oculta. El diseño de la red y su entrenamiento fueron realizados en la nube utilizando PyTorch, y posteriormente sus parámetros fueron exportados para ser utilizados en la implementación en hardware. Esto permite demostrar el flujo completo propuesto: diseño y entrenamiento en la nube + inferencia en el borde con lógica reconfigurable.

### *A. Caso: Regresión no lineal múltiple*

Se pretende modelar las relaciones no lineales entre una variable dependiente y dos variables independientes, es decir, hallar la función 'f' tal que y = f(x1, x2) cumpla con la relación entre variables indicada en la Tabla I.

Tabla I. Datos del caso de regresión no lineal múltiple

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **X1** | 3 | 2 | 5 | 1 | 5 | 6 | 7 | 4 | -4 | -7 |
| **X2** | 20 | 45 | 33 | 12 | 15 | 22 | 33 | 19 | 20 | 4.1 |
| **y** | 67 | 110 | 103 | 39 | 68 | 90 | 114 | 69 | 23.2 | 4.1 |

Tabla II. Formato de entradas y parámetros de la red.

| Entradas | Pesos capa oculta | Pesos capa salida | Sesgos capa oculta | Sesgos capa salida |
|---|---|---|---|---|
| Q7.1 | Q4.12 | Q4.12 | Q11.13 | Q7.25 |

Para encarar el problema, se diseñó en PyTorch una red neuronal de dos entradas (x1, x2) y una salida (y). Se ensayaron diversas configuraciones para la red, variando la cantidad de capas ocultas, neuronas por capa y funciones de activación. En base a los resultados obtenidos se buscó un balance entre complejidad y precisión, eligiendo una red conformada por una capa oculta de dos neuronas con funciones de activación ReLU y una capa de salida de una neurona con función de activación lineal. Una vez entrenada, con un error cuadrático medio (MSE) en la salida de 1,088 %, se obtuvieron los valores de pesos y sesgos para su implementación en hardware.

### *B. Implementación del caso de prueba en SoC*

La aplicación se implementó utilizando la placa ZYNQ - 7020 de la empresa Digilent, basada en un SoC de la familia Zynq 7000 de Xilinx. La personalización del núcleo VHDL descrito en la sección III se realizó en base a la cuantización de los parámetros de red obtenidos durante el entrenamiento en PyTorch, que permitió definir los valores de los genéricos VHDL asociados según consta en la siguiente sección de código.

```
generic (N: integer := 2;  -- Cantidad de entradas a la red
M1: integer := 2;  -- Cantidad de Neuronas de la capa oculta
M2: integer := 1;  -- Cantidad de Neuronas de la capa salida
N_bits_ent:integer := 8;  -- Nro bits de cada entrada a la red
N_bits_peso:integer := 16;  -- Nro bits de cada peso
N_bits_u1:integer:=24; -- Nro bits del bias de capa oculta
N_bits_u2:integer:=32 );  -- Nro bits del bias de capa salida
```

En la Tabla II se presentan los formatos de las entradas y de los parámetros de la red neuronal.

El circuito de la red neuronal propiamente dicha se mapeó en el FPGA de la familia Artix 7 que integra el SoC, y el diagrama en bloques correspondiente se muestra en la Fig. 3.

El núcleo VHDL de red neuronal implementado en lógica reconfigurable se empaquetó en el entorno de desarrollo Vivado como un esclavo AXI, con la finalidad de conectarlo con los otros módulos que componen el sistema que soporta la aplicación, como puede apreciarse en la Fig. 4

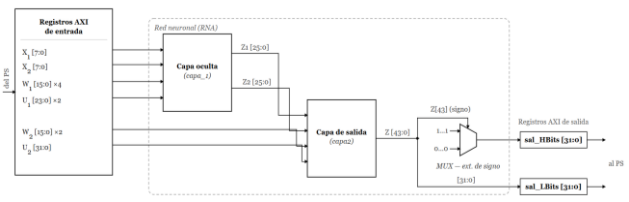



Fig. 3. Esquemático de la red neuronal para el caso de regresión no lineal múltiple.

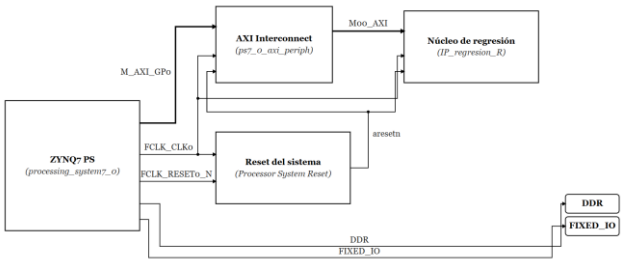



Fig. 4. Circuito del problema de regresión no lineal múltiple

Los ensayos realizados arrojaron un error cuadrático medio del 1,089 % respecto de la salida esperada, resultando prácticamente el mismo que el obtenido durante el entrenamiento en PyTorch.

Los retardos de la red propiamente dicha según el informe de implementación del entorno de desarrollo se presentan en la Tabla III. En la misma queda en evidencia que la demora total de la red, medida sobre el paso crítico, se reparte aproximadamente en partes iguales entre lógica y enrutamiento. Esto muestra un balance adecuado entre ambos componentes, lo que valida el diseño como una solución escalable.

La interconexión del núcleo en el FPGA con el Cortex-A9 potencia las fortalezas de ambas tecnologías haciendo responsable a cada una de ellas de las tareas para las cuales demuestra ser más eficiente; pero introduce un retardo significativo en el funcionamiento del sistema según se aprecia en la Tabla IV, pese a lo cual, la latencia de la aplicación resulta ser un orden de magnitud menor que la presentada por la red implementada en GPU.

La utilización de recursos en el FPGA que se presenta en la Fig. 5 resulta compatible con una red poco densa como la utilizada para el caso de prueba. Cabe aclarar que en el dispositivo elegido la red se puede escalar hasta incluir 161 neuronas. El componente limitante cuando se piensa en abordar aplicaciones más complejas resulta ser el número de patillas del dispositivo. En la implementación realizada esta limitación se subsana utilizando registros AXI que funcionan como interfaces de comunicación entre el Cortex-A9 y el FPGA. Se espera en el futuro estudiar otras aproximaciones como multiplexado, serialización o incorporación de memoria física para abordar el acceso a los datos necesarios para el funcionamiento de la red propiamente dicha, considerando siempre un balance adecuado entre los recursos del FPGA que se necesite incorporar y la demora agregada en la obtención de los resultados.

TABLA III. RETARDOS EN EL REPORTE DE IMPLEMENTACIÓN DE VIVADO.

| Tipo de Análisis | |
|---|---|
| Retardo Total | 31.788 ns |
| Retardo Lógico | 16.767 ns |
| Retardo de ruteo | 15.022 ns |
| Niveles Lógicos | 20 |

TABLA IV. COMPARACIÓN DE LOS RETARDOS CORRESPONDIENTES A LA IMPLEMENTACIÓN DE LA RED EN GPU (TESLA T4) Y SOC.

| Métrica | GPU (baseline) | SoC (propuesto) | PL (lógica reconfigurable) |
|---|---|---|---|
| Tiempo de inferencia | 10.5ms | 678us | 31.788ns |

| Name | Slice LUTs (53200) | Slice (13300) | LUT as Logic (53200) | DSPs (220) | Bonded IOB (200) |
|---|---|---|---|---|---|
| Red_Neuronal | 136 | 41 | 136 | 6 | 176 |
| capa2 (Capa_Red_Gen__parameterized1) | 7 | 3 | 7 | 0 | 0 |
| capa_1 (Capa_Red_Gen) | 129 | 40 | 129 | 4 | 0 |

Fig. 5. Recursos del FPGA utilizados en implementación.

### *C. Conclusiones y trabajo futuro*

Los resultados validan la factibilidad del flujo híbrido nube–borde propuesto. El núcleo VHDL parametrizable permite implementar RNAs MLP con baja latencia, alta reutilización y desempeño competitivo frente a GPU en aplicaciones embebidas. Como trabajo futuro se plantea modificar el diseño del núcleo para extender el rango de aplicaciones abordables sin que se degrade de manera significativa el throughput de la red. Esto implicará explorar estrategias de multiplexado, serialización e incorporación de memoria dedicada para escalar a RNAs más densas, sin incrementar significativamente la latencia del sistema.

## REFERENCIAS

[1] A. S. George, A. S. H. George, and T. Baskar, "Edge Computing and the Future of Cloud Computing: A Survey of Industry Perspectives and Predictions," Partners Universal Int. Res. J. (PUIRJ), vol. 2, no. 2, pp. 1–9, Apr.–Jun. 2023, doi: 10.5281/zenodo.8020101.

[2] Y. Himeur, A. N. Sayed, A. Alsalemi, F. Bensaali, and A. Amira, "Edge AI for Internet of Energy: Challenges and Perspectives," Internet of Things, vol. 25, Art. no. 101035, 2024, doi: 10.1016/j.iot.2023.101035.

[3] L. Liu, H. Zhu, T. Wang, and M. Tang, "A Fast and Efficient Task Offloading Approach in Edge-Cloud Collaboration Environment," Electronics, vol. 13, no. 2, Art. no. 313, 2024, doi: 10.3390/electronics13020313.

[4] N. Rathi, I. Chakraborty, A. Kosta, A. Sengupta, A. Ankit, P. Panda, and K. Roy, "Exploring Neuromorphic Computing Based on Spiking Neural Networks: Algorithms to Hardware," ACM Comput. Surv., vol. 55, no. 12, Art. no. 243, pp. 1–49, Mar. 2023, doi: 10.1145/3571155.

[5] M. Harrane and R. Touhami, "MLP on FPGA: Optimal Coding of Data and Activation Function," in 2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Genova, Italy, 2019, pp. 338–341, doi: 10.1109/ICECS46596.2019.8924355.

[6] M. Reis, M. Véstias, and H. Neto, "Designing Deep Learning Models on FPGA with Multiple Heterogeneous Engines," ACM Trans. Reconfig. Technol. Syst., vol. 17, no. 1, Art. no. 6, pp. 1–30, Jan. 2024, doi: 10.1145/3615870.

[7] S. E. El Maachi, A. Chehri, and R. Saadane, "Efficient Hardware Acceleration of Spiking Neural Networks Using FPGA: Towards Real-Time Edge Neuromorphic Computing," in 2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring), Singapore, 2024, pp. 1–5, doi: 10.1109/VTC2024-Spring62846.2024.10683049.

[8] A. Z. Ghonem, S. M. Elewa, and E. Y. Tawfik, "Towards Efficient and Reconfigurable Neuromorphic Computing Using an Embedded FPGA in 22nmFDX," in 2024 22nd IEEE Interregional NEWCAS Conference (NEWCAS), Sherbrooke, QC, Canada, 2024, pp. 138–142, doi: 10.1109/NewCAS58973.2024.10666113.

[9] H. Johnson, T. Fang, A. Perez-Vicente, and J. Saniie, "Reconfigurable Distributed FPGA Cluster Design for Deep Learning Accelerators," in 2023 IEEE International Conference on Electro Information Technology (eIT), Romeoville, IL, USA, 2023, pp. 1–5, doi: 10.1109/eIT57321.2023.10187228.

[10] D. Maheshwari, A. R. Young, P. Date, S. R. Kulkarni, B. Witherspoon, and N. R. Miniskar, "An FPGA-Based Neuromorphic Processor with All-to-All Connectivity," in 2023 IEEE International Conference on Rebooting Computing (ICRC), San Diego, CA, USA, 2023, pp. 1–5, doi: 10.1109/ICRC60800.2023.10386808.

[11] R. Wu, X. Guo, J. Du, and J. Li, "Accelerating Neural Network Inference on FPGA-Based Platforms—A Survey," Electronics, vol. 10, no. 9, Art. no. 1025, Apr. 2021, doi: 10.3390/electronics10091025.

# Detección de Merluza en Sonar con YOLO para Sistemas Embebidos

Demián N. Mozo ‡, Lucas Rabioglio†‡, Celeste Cebedio †‡, Luciana De Micco†‡§
† *Instituto de Investigaciones Científicas y Tecnológicas en Electrónica (ICYTE)*
‡ *Facultad de Ingeniería, Universidad Nacional de Mar del Plata (FI - UNMdP)*
§ *Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)*
dmozo@alumnos.fi.mdp.edu.ar, {celestecebedio,lucas.rabioglio,ldemicco}@fi.mdp.edu.ar

***Resumen*—En este trabajo se presenta el entrenamiento, evaluación y análisis de factibilidad de una variante liviana preexistente de la familia YOLO para la detección automática de ecos compatibles con merluza en imágenes de sonar. El modelo fue entrenado con un conjunto reducido de datos compuesto por 85 imágenes, obteniendo en la partición de validación valores de mAP@50 de 0.9539 y mAP@50–95 de 0.7493. Se analiza la arquitectura empleada, su complejidad computacional y su potencial de utilización en plataformas embebidas con recursos limitados. Asimismo, se evalúa la cuantización post-entrenamiento del modelo como estrategia para reducir los requerimientos de memoria y facilitar un futuro despliegue en sistemas embebidos, analizando su impacto sobre las métricas de detección. Los resultados obtenidos constituyen una evaluación preliminar y ponen de manifiesto la necesidad de ampliar el conjunto de datos y de evaluar el desempeño ante una mayor diversidad de condiciones operativas y especies.**

***Index Terms*—YOLO, detección de objetos, ecosonda, merluza, sistemas embebidos, cuantización**

## I. Introducción

La detección automática de merluza a partir de registros de sonar resulta de interés para el monitoreo pesquero, la estimación de biomasa y la asistencia a la toma de decisiones durante las campañas de relevamiento. En la práctica, la inspección manual de ecogramas demanda tiempo y depende de la experiencia del operador, mientras que las condiciones de adquisición —ruido, bajo contraste, variabilidad de los ecos y disponibilidad limitada de datos etiquetados— dificultan la automatización del análisis. En este contexto, disponer de herramientas capaces de localizar automáticamente ecos asociados a merluza puede contribuir al procesamiento sistemático de grandes volúmenes de información acústica.

Las redes neuronales profundas han demostrado ser efectivas para tareas de detección de objetos. En particular, las arquitecturas de la familia YOLO (You Only Look Once) [1], [2] permiten realizar detección y localización en una única etapa de procesamiento, por lo que resultan atractivas para aplicaciones con requerimientos de baja latencia. Trabajos previos han demostrado su utilidad en la detección de peces en imágenes submarinas [3] y, recientemente, en el procesamiento de imágenes de sonar, donde deben abordarse problemas de bajo contraste, ruido y resolución limitada. [4], [5].

Sin embargo, el despliegue de estos modelos cerca del sistema de adquisición plantea restricciones de memoria, consumo y capacidad de cómputo. Por este motivo, se han propuesto variantes livianas y técnicas de reducción de precisión orientadas a plataformas con recursos limitados [6], [7]. En este trabajo se analiza la utilización de YOLOv8n, una variante compacta de la familia YOLO, y su cuantización post-entrenamiento como una etapa preliminar hacia un futuro despliegue en una plataforma embebida conectada al sistema de sonar.

En este marco, el presente trabajo propone:

- entrenar y evaluar YOLOv8n para la detección de ecos asociados a merluza en imágenes de sonar;
- analizar su desempeño sobre un conjunto de datos reducido;
- evaluar el impacto de la cuantización post-entrenamiento INT8 sobre el tamaño del modelo y las métricas de detección;
- discutir su factibilidad como etapa previa a un futuro despliegue en plataformas embebidas.

## II. Metodología

### *II-A. Dataset*

Se utilizó un conjunto de datos compuesto por 85 imágenes de sonar provistas por la empresa Seaman Electronics, correspondientes a registros acústicos reales adquiridos en condiciones operativas. Las imágenes fueron etiquetadas manualmente, identificando la presencia de peces mediante cajas delimitadoras.

Las imágenes corresponden a ecogramas generados a partir de registros acústicos discretos y no a una secuencia de video. El archivo analizado contiene 10625 pings, una frecuencia de muestreo de 773.0825 kHz y hasta aproximadamente 152412 muestras por ping. Para el entrenamiento, las imágenes fueron redimensionadas a $1024 \times 1024$ píxeles. Por ello, no se informa una tasa de cuadros por segundo equivalente a la de una cámara.

Previo al entrenamiento, se aplicó un preprocesamiento de los datos orientado a mejorar la calidad visual y la consistencia de las imágenes. Este incluyó la selección de regiones de interés, la normalización de los niveles de intensidad y la eliminación de zonas irrelevantes como el fondo, con el objetivo de resaltar las estructuras asociadas a ecos de peces.

El problema se planteó como una tarea de detección de una única clase, denominada *fish* dentro del formato de

etiquetado empleado por YOLO. Esta etiqueta corresponde exclusivamente a ecos identificados como asociados a merluza en las imágenes disponibles. En consecuencia, el modelo fue entrenado para localizar dichas estructuras acústicas, mientras que el fondo, el ruido y las estructuras no anotadas actúan como ejemplos negativos durante el entrenamiento. No obstante, la capacidad de discriminación frente a otras especies de peces deberá ser evaluada mediante la incorporación de datos multiespecio en trabajos futuros.

### *II-B. Entrenamiento*

El conjunto de 85 imágenes se dividió a nivel de imagen en 68 muestras para entrenamiento y 17 para validación. El modelo se entrenó durante 80 épocas utilizando un tamaño de lote de 4 y entradas de $1024 \times 1024$ píxeles. Se utilizó el optimizador automático de Ultralytics YOLO y aumento de datos mediante *mosaic*, inversión horizontal y variación HSV.

### *II-C. Arquitectura: YOLOv8n*

Se seleccionó como arquitectura base la red YOLOv8n (nano). Esta versión específica está optimizada para operar en entornos con recursos de hardware limitados, al poseer una cantidad significativamente menor de parámetros y una menor profundidad de capas en comparación con las variantes estándar de la familia YOLO.

El modelo entrenado presenta:

- 130 capas
- 3,011,043 parámetros
- 8.2 GFLOPs
- 1 clase de salida

La arquitectura corresponde a una variante ligera de YOLO basada en bloques *Conv*, *C2f* y *SPPF*, con cabeza de detección multiescala.

## III. Resultados

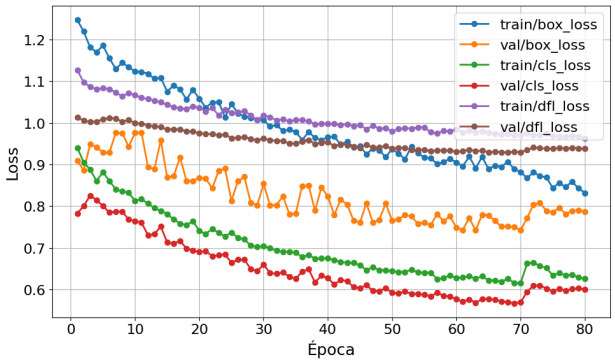



Figura 1. Evolución de las funciones de pérdida durante el entrenamiento y validación del modelo.

### *III-A. Funciones de pérdida*

La Figura 1 muestra la evolución de las funciones de pérdida asociadas al entrenamiento del modelo:

- **box_loss**: error en la localización de las cajas delimitadoras.
- **cls_loss**: error de clasificación de los objetos detectados.
- **dfl_loss**: mide el error en el ajuste fino de los bordes de las cajas delimitadoras.

Se presentan tanto las pérdidas de entrenamiento (*train*) como de validación (*val*), lo cual permite evaluar la capacidad de generalización del modelo.

### *III-B. Análisis del entrenamiento*

La Fig. 1 muestra una disminución de las pérdidas de entrenamiento y validación, compatible con una convergencia estable. Las leves fluctuaciones finales se atribuyen al tamaño reducido del conjunto de datos.

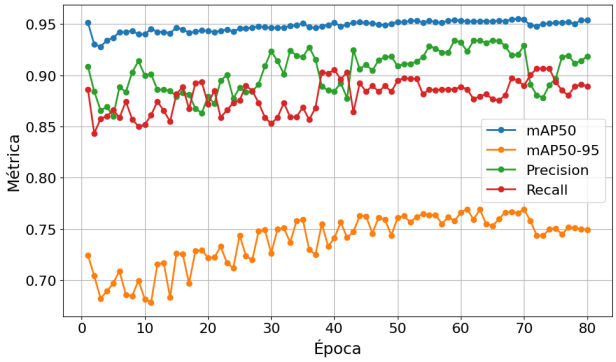



Figura 2. Evolución de métricas de validación durante el entrenamiento del modelo YOLO.

### *III-C. Descripción de métricas*

En la Figura 2 se presentan las principales métricas de evaluación utilizadas durante el entrenamiento:

- **mAP@50**: corresponde al *mean Average Precision* calculado con un umbral de intersección sobre unión (IoU) de 0.5. Esta métrica evalúa la capacidad general del modelo para detectar correctamente los objetos.
- **mAP@50-95**: representa el promedio del mAP calculado para valores de IoU entre 0.5 y 0.95. Se trata de una métrica más estricta que evalúa tanto la detección como la precisión en la localización de las cajas.
- **Precision**: indica la proporción de detecciones correctas respecto del total de detecciones realizadas, reflejando la tasa de falsos positivos.
- **Recall**: mide la proporción de objetos correctamente detectados respecto del total de objetos presentes, indicando la capacidad del modelo para no omitir detecciones.

### *III-D. Análisis de resultados*

Como se observa en la Figura 2, el modelo presenta una convergencia rápida y estable a lo largo de las épocas. El valor de mAP@50 se mantiene cercano a 0.95, lo que indica una alta capacidad de detección general. Por su parte, el mAP@50-95 alcanza valores alrededor de 0.75, reflejando un buen desempeño en la localización precisa de los objetos.

Asimismo, la precisión se mantiene elevada durante todo el entrenamiento, con valores superiores a 0.9, lo que sugiere una

baja tasa de falsos positivos. El recall, ligeramente inferior, se sitúa en torno a 0.88–0.90, indicando que el modelo detecta la mayoría de los objetos, aunque con un comportamiento levemente conservador. Este balance entre precisión y recall resulta adecuado para el tipo de aplicación considerada.

En conjunto, los resultados muestran un desempeño preliminar favorable sobre la partición de validación, aun considerando el tamaño reducido del conjunto de entrenamiento. Las métricas obtenidas fueron:

- Precisión: 0.9185
- Recall: 0.8894
- mAP@50: 0.9539
- mAP@50-95: 0.7493

### *III-E. Resultados cualitativos*

La Figura 3 muestra un ejemplo de inferencia del modelo sobre una imagen de sonar. Se observa que el modelo es capaz de identificar correctamente las estructuras características asociadas a la presencia de peces, las cuales aparecen como ecos de alta intensidad en la imagen. Las detecciones presentan valores de confianza mayormente elevados, lo que indica que el modelo ha aprendido representaciones discriminativas del objetivo. Asimismo, se observa que las detecciones siguen patrones coherentes con la estructura física del fenómeno acústico.

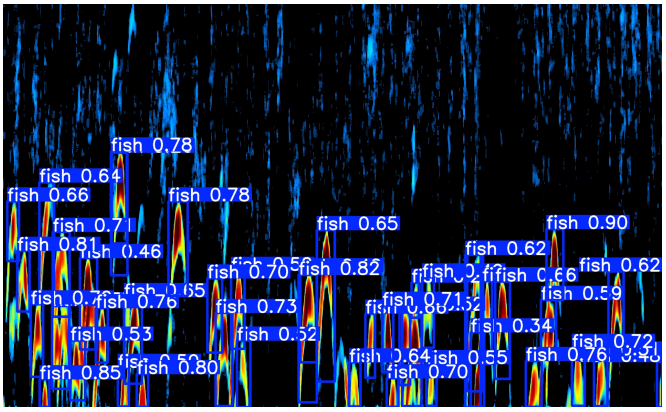



Figura 3. Ejemplo de inferencia del modelo YOLO sobre una imagen de sonar. Las cajas delimitadoras indican las detecciones de la clase *fish* junto con su probabilidad asociada.

Con el objetivo de evaluar el comportamiento del modelo en condiciones más exigentes, la Figura 4 presenta un caso correspondiente a una situación operativa más compleja. En este escenario, asociado a una mayor velocidad de desplazamiento de la embarcación, la imagen presenta estructuras acústicas más alargadas, mayor dispersión y menor definición visual de los ecos, lo que dificulta la tarea de detección.

A pesar de estas condiciones adversas, el modelo logra detectar regiones compatibles con la presencia de peces, aunque con valores de confianza más moderados que en el caso anterior. En conjunto, estos resultados sugieren que la red conserva capacidad de generalización aun en escenarios menos favorables, aunque también evidencian la necesidad de ampliar el conjunto de datos con condiciones operativas diversas.

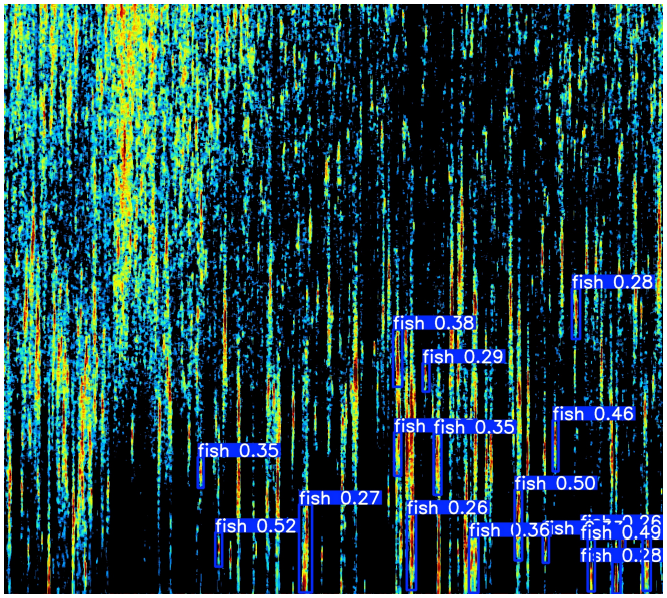



Figura 4. Ejemplo cualitativo de detección en un registro adquirido durante el desplazamiento de la embarcación a aproximadamente 10 nudos. Esta condición, superior a la velocidad habitual de búsqueda o pesca (∼4 nudos), presenta ecos más débiles y dispersos.

## IV. Discusión

Los resultados obtenidos sobre la partición de validación sugieren que el modelo puede localizar ecos asociados a merluza en las condiciones representadas en el conjunto disponible. Sin embargo, el número reducido de imágenes, la ausencia de validación cruzada y la baja diversidad de especies, fondos y condiciones operativas limitan la capacidad de generalización que puede atribuirse al sistema. En consecuencia, los resultados deben considerarse preliminares. Como trabajo futuro se propone ampliar el conjunto de datos, incorporar otras especies y clases de fondo, realizar una evaluación multiclase con validación cruzada y analizar el desempeño ante diferentes condiciones de adquisición y navegación.

## V. Cuantización para Inferencia Edge

Con el objetivo de analizar el impacto de técnicas de reducción de precisión orientadas al despliegue en sistemas edge y plataformas embebidas, el detector YOLOv8n entrenado fue exportado desde PyTorch al formato ONNX [8] utilizando la librería Ultralytics YOLO.

Posteriormente, se aplicó cuantización post-entrenamiento (*post-training quantization*) INT8 mediante las herramientas de cuantización de ONNX Runtime [9], obteniendo una versión cuantizada del modelo orientada a reducir los requerimientos de memoria y complejidad computacional durante la inferencia. Cabe destacar que este proceso modifica principalmente la representación numérica del modelo, sin implicar una optimización específica de la arquitectura hardware utilizada para ejecutar la inferencia.

La Tabla I resume la comparación entre el modelo original en precisión flotante y el modelo cuantizado.

Cuadro I
COMPARACIÓN ENTRE MODELO ORIGINAL Y CUANTIZADO

| Modelo | Precisión | Tamaño |
|---|---|---|
| YOLOv8n ONNX | FP32 | 11.70 MB |
| YOLOv8n ONNX | INT8 | 3.20 MB |

La cuantización redujo el tamaño del modelo un 72.6 %. La evaluación del modelo INT8 se realizó con ONNX Runtime sobre CPU, en una estación de trabajo con procesador Intel Core i9 y GPU NVIDIA GeForce RTX 3060, sin utilizar aceleración por GPU. Se obtuvieron precisión de 0.801, *recall* de 0.804, mAP@50 de 0.829 y mAP@50–95 de 0.404.

Si bien se observa una degradación respecto del modelo FP32, la reducción de memoria resulta de interés para plataformas con recursos limitados. No se caracterizaron la latencia, los FPS ni el consumo de memoria y energético, cuya evaluación sobre una plataforma embebida objetivo queda como trabajo futuro.

## VI. PROPUESTA DE INTEGRACIÓN CON SISTEMA SONAR

Como etapa posterior al procesamiento realizado por el sonar, se propone una arquitectura de integración orientada a plataformas embebidas, esquematizada en la Fig. 5. El sistema de adquisición entrega los registros acústicos o las imágenes de sonar procesadas a una unidad de cómputo embebida mediante la interfaz de comunicación disponible en el equipo, por ejemplo Ethernet, USB, puerto serie o almacenamiento local. La unidad embebida recibe las imágenes o datos acústicos, realiza las etapas de preprocesamiento necesarias para adecuar la entrada al formato requerido por la red —selección de región de interés, normalización y redimensionamiento— y ejecuta la inferencia del modelo YOLO cuantizado. Como salida, el sistema genera las cajas delimitadoras, la probabilidad asociada a cada detección y, eventualmente, una señal de alarma o un registro de eventos para su visualización, almacenamiento o transmisión a un sistema externo.
La cuantización INT8 analizada en este trabajo constituye una etapa preliminar para reducir los requerimientos de memoria del modelo y facilitar su posterior ejecución en una plataforma con recursos limitados. Sin embargo, la presente contribución no incluye aún una integración física con un sonar comercial ni una implementación directa sobre FPGA u otra plataforma embebida. La selección de la interfaz de comunicación, la caracterización temporal del flujo de datos, la medición de latencia y la evaluación de recursos y consumo energético serán abordadas en trabajos futuros.

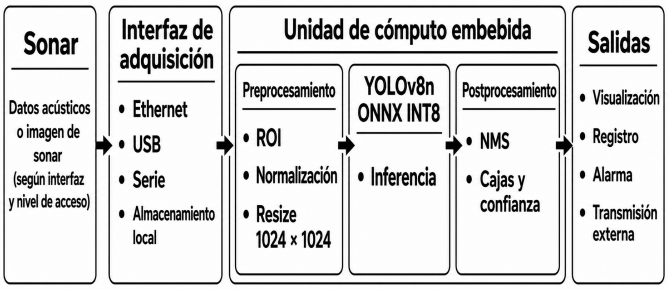



Figura 5. Esquema conceptual de integración del detector con un sistema de sonar.

## VII. CONCLUSIONES Y TRABAJO FUTURO

Se desarrolló un modelo YOLO eficiente para la detección de merluza en imágenes de sonar, alcanzando altos valores de precisión incluso con un dataset reducido.

Los resultados muestran el potencial de este enfoque como etapa preliminar hacia aplicaciones embebidas, especialmente al combinarlo con técnicas de cuantización. Sin embargo, será necesario caracterizar la latencia, los cuadros por segundo, el consumo de memoria y el consumo energético sobre una plataforma objetivo antes de establecer su viabilidad para operación en tiempo real. Como trabajo futuro se propone:

- Ampliar el conjunto de datos e incorporar múltiples especies y clases de fondo.
- Evaluar técnicas de optimización del modelo mediante pruning, distillation y cuantización consciente del entrenamiento (QAT)
- Explorar representaciones de menor precisión (4 bits)
- Considerar implementaciones sobre FPGA, incluyendo arquitecturas basadas en DPU y aceleradores de hardware
- Utilizar herramientas de diseño y entrenamiento orientado a hardware.

## AGRADECIMIENTOS

Este trabajo fue financiado por el CONICET (PIP 2025–2027, Subsidio N.º 11220250100447CO) y por el Fondo para Reparación y Actualización de Equipamiento Científico 2025 de la Fundación Williams. Asimismo, los autores agradecen a AMD por la provisión de placas de desarrollo a través del AMD University Program.

## REFERENCIAS

[1] J. Redmon and et al, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
[2] Ultralytics, "Ultralytics yolo documentation," https://docs.ultralytics.com, 2024, accedido: 2026-04-29.
[3] A. Al Muksit and et al, "Yolo-fish: A robust fish detection model to detect fish in realistic underwater environment," *Ecological Informatics*, vol. 72, p. 101847, 2022.
[4] B. Xing *et al.*, "Fish sonar image recognition algorithm based on improved yolov5," *Research Square*, 2023. [Online]. Available: https://doi.org/10.21203/rs.3.rs-3369498/v1
[5] S. Yu, "Sonar image target detection based on deep learning," *Mathematical Problems in Engineering*, no. 1, p. 5294151, 2022.
[6] N. Yang *et al.*, "Ss-yolo: A lightweight deep learning model focused on side-scan sonar target detection," *Journal of Marine Science and Engineering*, vol. 13, p. 66, 2025.
[7] X. Cui *et al.*, "Small object detection in side-scan sonar images based on soca-yolo and image restoration," *Frontiers in Marine Science*, vol. 12, p. 1542832, 2025.
[8] ONNX Community, "Onnx: Open neural network exchange," 2026, accessed: 2026-05-09. [Online]. Available: https://onnx.ai/
[9] ONNX Runtime, "On-device training," 2026, accessed: 2026-05-09. [Online]. Available: https://onnxruntime.ai/docs/get-started/training-on-device.html

Categoría

# Foro Tecnológico

Internet de las Cosas

Área Temática

# Placa de desarrollo para IoT Industrial basada en el microcontrolador ESP32

Agustín Zuliani, Daniel Márquez
Departamento de Sistemas e Informática (D.S.I.)
Facultad de Ciencias Exactas, Ingeniería y Agrimensura (F.C.E.I.A.)
Universidad Nacional de Rosario, Rosario, Argentina
Email: azuliani02@fceia.unr.edu.ar, dmarquez@fceia.unr.edu.ar

***Resumen*—En esta publicación se presenta el desarrollo de una placa de tipo PLC para implementaciones industriales embebidas, basada en el microcontrolador ESP32 de Espressif. La placa incorpora periféricos estándar en dispositivos programables industriales, tales como entradas digitales optoaisladas de 12 V a 24 V, salidas de potencia con relé simple inversor, salidas de estado sólido con TRIAC, reloj calendario de tiempo real (RTCC) y conectores de expansión mediante $I^2C$. Adicionalmente, dispone de memoria EEPROM para almacenamiento no volátil, pantalla OLED comandada por $I^2C$, alarma sonora y LEDs para monitoreo de variables, conformando una plataforma versátil para diversas aplicaciones. Las capacidades de conectividad WiFi y Bluetooth del ESP32 posibilitan implementaciones en Internet de las Cosas Industrial (IIoT) y telemetría. Más allá del uso industrial, su campo de aplicación se extiende a la domótica y la enseñanza de automatización y sistemas embebidos. Desde la perspectiva académica, esta placa cumple un rol importante en la formación de ingenieros electrónicos, al proporcionar prácticas que favorecen el aprendizaje experiencial.**

***Palabras clave*—PLC, ESP32, RTOS, FreeRTOS, diseño electrónico, PCB, domótica**

## I. Introducción

El ESP32 [1] es un microcontrolador versátil que posee características relevantes como conexión WiFi de 2,4 GHz, Bluetooth, Ethernet (en modelos recientes) y comunicación CAN, entre otros. Estas características lo convierten en una opción adecuada para desarrollos electrónicos con enfoque en Industria 4.0 o IIoT, donde se requiere que los sistemas puedan ser gestionados a través de una red central, facilitando además la recopilación de datos para análisis predictivo de procesos.

En el ámbito industrial resulta necesario contar con etapas de potencia que permitan manejar diferentes tipos de cargas —electroválvulas, motores, actuadores, entre otros—, manteniendo una lógica programada para realizar acciones bajo determinadas condiciones de trabajo. Por ende, no basta con disponer únicamente de un microcontrolador, sino que además se requiere integrar hardware de potencia como relés, MOSFET y TRIAC para la gestión de cargas que deben ser accionadas por la lógica implementada.

Se plantea entonces el diseño y desarrollo de un hardware robusto que dé soporte al ESP32 para su utilización en el ámbito industrial, alcanzado mediante el diseño de etapas electrónicas que satisfacen exigencias de voltajes de alimentación, niveles de entrada, potencias de salida, así como criterios de diseño del circuito impreso (PCB) que garanticen correctas aislaciones y una adecuada inmunidad al ruido electromagnético.

### A. *Estado del arte*

En los últimos años han surgido diversas plataformas comerciales y de código abierto que combinan el ESP32 con hardware orientado a entornos industriales. Entre las más representativas se encuentran los productos de Industrial Shields, que ofrecen una familia de PLCs basados en ESP32, montaje en riel DIN, Ethernet (W5500), RS-485 y expansión por $I^2C$, con precios desde aproximadamente 100 USD para el modelo de 14 E/S [2]. Por otro lado, NORVI comercializa controladores industriales ESP32 con entradas optoaisladas, salidas a relé, pantalla OLED, RS-485 y opciones de conectividad, orientados a aplicaciones de IIoT [3]. Asimismo, el proyecto de código abierto OpenPLC [4] permite programar microcontroladores en lenguaje Ladder conforme al estándar IEC 61131-3 [5], aunque no provee un hardware industrial dedicado.

La Tabla I presenta una comparación entre estas alternativas y la placa propuesta en el presente trabajo.

| Característica | Propuesta | Ind. Shields | NORVI |
|---|---|---|---|
| Entradas dig. optoaisladas | Sí | Sí | Sí |
| Entradas analógicas | Sí (0–10 V) | Sí | Sí |
| Salidas relé | Sí | Sí | Sí |
| Salidas TRIAC | 5 | No | No |
| OLED integrada | Sí | No | Sí |
| RTC con respaldo | Sí | No | Sí |
| EEPROM | Sí | No | No |
| WiFi / Bluetooth | Sí | Sí | Sí |
| Ethernet | No* | Sí | Sí* |
| RS-485 | No* | Sí | Sí |
| Costo estimado (USD) | $\sim$30 | $>$100 | $>$60 |
| Código abierto (HW) | Sí | No | No |

*Previsto en futuras revisiones.

Tabla I: Comparación con alternativas comerciales basadas en ESP32.

La principal diferenciación de la placa propuesta radica en tres aspectos: su carácter completamente de código abierto (circuito esquemáticos y PCB publicados en un repositorio público [6]), la incorporación de salidas de estado sólido con TRIAC, ausentes en las otras alternativas, y un costo de producción significativamente inferior al de las soluciones comerciales, lo que la posiciona como una herramienta accesible

para el ámbito académico y para profesionales que requieran un punto de partida personalizable.

### B. *Requerimientos de diseño*

Los requerimientos de la placa se plantean de la siguiente manera:

- Voltaje de alimentación versátil para niveles industriales entre 12 V y 24 V CC.
- Entradas digitales optoaisladas con soporte de niveles entre 5 V y 24 V.
- Entradas analógicas 0 V–10 V con alta impedancia de entrada, resueltas mediante el ADC interno del ESP32 con 12 bits de resolución.
- Salidas digitales con LEDs testigos para monitoreo de estados.
- LED RGB para monitoreo de estados.
- Salidas de potencia mediante relés simple inversor (modelo G5LE-1-E DC5, 10 A/250 VAC) optoaislados de la lógica.
- Salidas de estado sólido mediante TRIAC (BTA16-600B, 16 A/600 V) con disparo optoaislado (MOC3021).
- Almacenamiento de datos no volátiles en memoria EEPROM (AT24C32, 32 Kbit, $I^2C$).
- Reloj calendario DS3231 con batería de respaldo.
- Puerto de comunicación USB-UART (CP2102) para interfaces HMI.
- Puerto UART auxiliar para conexión de interfaces serie adicionales, por ejemplo, un transceptor RS-485.

La conectividad WiFi del ESP32 se contempla para implementar diferentes protocolos de comunicación e interfaces de usuario, entre ellos: Modbus TCP, MQTT para telemetría y control remoto, WebSocket, servidor web embebido con interfaz HTML y HTTP/REST API. La disponibilidad de bibliotecas probadas en el ecosistema ESP-IDF y Arduino para estos protocolos, tales como `AsyncWebServer`, `PubSubClient` (MQTT), `ModbusIP_ESP8266` y el stack TCP/IP de lwIP, facilita la integración de estas funcionalidades sin requerir desarrollo desde cero.

La versatilidad del ESP32 permite trabajar en lenguaje C y diseñar aplicaciones sobre un sistema operativo en tiempo real como FreeRTOS [7], e incluso programarlo en lenguaje Ladder mediante el proyecto OpenPLC [4], compatible con el estándar IEC 61131-3 [5].

## II. Diseño del hardware

En esta sección se detallan los aspectos relacionados con el hardware de la placa ESP32 [6]. En la Tabla II se presentan todas las señales de entrada y salida disponibles.

### A. *Entradas*

Las señales de entrada digital están optoacopladas para garantizar un aislamiento seguro del microcontrolador. Cada canal incorpora un LED indicador que permite verificar visualmente su activación. Mediante el *jumper* JP2 se selecciona la referencia de tensión utilizada. En la Figura 1 se presenta el circuito correspondiente a una entrada digital.

| Etiqueta | Descripción | Rango |
|---|---|---|
| *Alimentación* | | |
| $V_{in}$ | Entrada de alimentación principal | +24 V |
| $V_{ref}$ | Tensión de ref. para entradas | +10 V |
| *Entradas digitales* | | |
| I0.0 | Entrada digital 0 | +24 V |
| I0.1 | Entrada digital 1 | +24 V |
| I0.2 | Entrada digital 2 | +24 V |
| I0.3 | Entrada digital 3 | +24 V |
| *Entradas analógicas* | | |
| A0.0 | Entrada analógica 0 | 0–10 V |
| A0.1 | Entrada analógica 1 | 0–10 V |
| *Salidas digitales* | | |
| Q0.0 | Relé 1 (simple inversor) | 220 VAC / 24 VDC |
| Q0.1 | Relé 2 (simple inversor) | 220 VAC / 24 VDC |
| Q0.2 | Relé 3 (simple inversor) | 220 VAC / 24 VDC |
| Q0.3 | TRIAC 1 (estado sólido) | 220 VAC |
| Q0.4 | TRIAC 2 (estado sólido) | 220 VAC |
| Q0.5 | TRIAC 3 (estado sólido) | 220 VAC |
| Q0.6 | TRIAC 4 (estado sólido) | 220 VAC |
| Q0.7 | TRIAC 5 (estado sólido) | 220 VAC |

Tabla II: Entradas y salidas del hardware de la placa ESP32.

Las entradas analógicas presentan un rango de 0 V a 10 V y se conectan al ADC interno del ESP32, que posee una resolución de 12 bits (4096 niveles). Se emplea un divisor resistivo de alta impedancia para adaptar el rango de tensión industrial al rango de entrada del conversor (0–3,3 V).

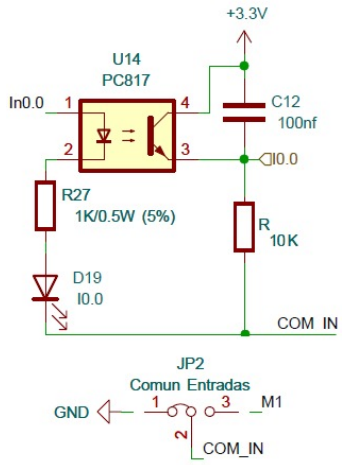



Figura 1: Circuito eléctrico de la entrada digital.

### B. *Salidas*

Las salidas digitales, al igual que las entradas, se encuentran optoaisladas. Para las salidas a relé se emplea el optoacoplador PC817, mientras que para las salidas a TRIAC se utiliza el opto-TRIAC MOC3021.

Los relés utilizados son del modelo G5LE-1-E DC5, con las siguientes características: tensión de bobina de 5 VDC, capacidad de conmutación de 10 A a 250 VAC (carga resistiva) y configuración simple inversor (SPDT). Cada salida a relé incluye un diodo 1N4007 en antiparalelo con la bobina (*freewheeling*) para proteger el transistor *driver* frente a los picos de tensión inductivos, como se observa en la Figura 2.

Las salidas de estado sólido emplean TRIAC del tipo BTA16-600B, con capacidad de 16 A y tensión de bloqueo de 600 V, disparados mediante el optoacoplador MOC3021. Este esquema, presentado en la Figura 3, resulta adecuado para el control de cargas AC como motores monofásicos, resistencias calefactoras y luminarias. La decisión de incorporar salidas TRIAC, además de las salidas a relé, responde a la necesidad de contar con conmutación rápida y sin desgaste mecánico para cargas AC.

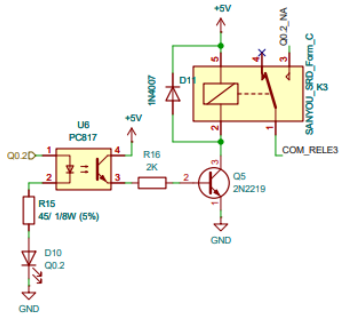



Figura 2: Salida con relé simple inversor.

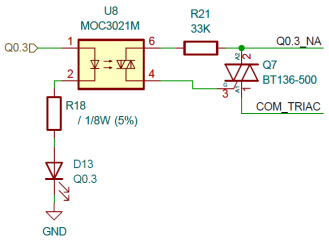



Figura 3: Salida de estado sólido con TRIAC.

### C. *Periféricos*

Se incorporan periféricos que enriquecen las capacidades de la placa: una pantalla OLED (SSD1306, 128×64) controlada mediante $I^2C$, un *buzzer* para señalización sonora, una memoria EEPROM (AT24C32, 32 Kbit, $I^2C$, 1 MHz) para almacenamiento no volátil de configuraciones y registros de datos, y un reloj de tiempo real DS3231 con cristal de 32,768 kHz y pila de respaldo, que permite registrar marcas de tiempo asociadas a eventos del sistema. Los periféricos $I^2C$ se conectan a un bus compartido con conectores de expansión accesibles, lo que posibilita agregar módulos adicionales (por ejemplo, conversores ADC externos o expansores de E/S) sin modificar el diseño base.

### D. *Diseño del PCB*

El diseño del PCB se realiza en KiCad [8] siguiendo las pautas descritas a continuación. Las dimensiones de la placa no superan los 100 $cm^2$. Al tratarse de un prototipo de primera versión, se prioriza el uso de componentes de montaje pasante (THT), con el fin de facilitar su adquisición y reemplazo. Los anchos de pista se dimensionan según la corriente máxima admisible en cada sección.

Una decisión de diseño relevante consiste en la realización de muescas (*slots*) alrededor de los conectores de relé como medida de aislación eléctrica, con el objetivo de prevenir arcos de voltaje al manejar cargas de 220 VAC. Para las etapas de potencia, las pistas se trazan en ambas capas (TOP y BOTTOM) a fin de incrementar la sección efectiva de conducción, reduciendo la resistencia y el sobrecalentamiento.

En la Figura 4 se presenta el resultado final del prototipo ensamblado.



Figura 4: Prototipo de la placa ESP32 ensamblada.

## III. Aplicaciones y Ejemplo

A modo de ejemplo integrador, se describe la implementación de un sistema de "control de temperatura de caldera con sistema de emergencia de presión y telemetría de variables", que pertenece a la categoría de IIoT.

En este sistema se utiliza una entrada analógica para medir la temperatura de la caldera mediante un sensor LM35 (ver Figura 5). La estrategia de control sobre el elemento calefactor resistivo consiste en un lazo cerrado SI-NO con histéresis. Una entrada digital optoaislada se emplea para conectar un sensor de detección de alta presión: ante un evento de sobrepresión, el sistema desactiva el calefactor y acciona una salida a relé que activa un ventilador de enfriamiento. Es relevante destacar que el calefactor opera con 12 VDC mientras que el ventilador funciona con 220 VAC, lo que demuestra la capacidad de la placa para manejar cargas con diferentes niveles de tensión simultáneamente.

Como interfaz local de tipo HMI se utiliza la pantalla OLED para visualizar la temperatura medida y el valor de consigna (*set-point*), el cual se asigna mediante comunicación USB-serie desde una PC.

En cuanto al *firmware*, se desarrolla sobre FreeRTOS [7], lo que permite modelar el software de forma ordenada y escalable, garantizando tiempos de respuesta adecuados ante eventos de lectura de temperatura y detección de alta presión. Para la telemetría, se implementa una aplicación web embebida en HTML accesible desde un navegador (ver Figura 6), que permite monitorear el estado del sistema de forma remota.

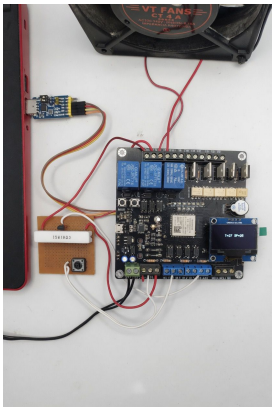


Figura 5: Montaje del proyecto de aplicación con la placa ESP32.




Figura 6: Interfaz de la aplicación web embebida.

## IV. Futuras mejoras

El presente desarrollo corresponde a una primera versión del prototipo. Se prevé que en futuras revisiones se incorporen las siguientes funcionalidades de hardware:

- Puerto de comunicación Ethernet (W5500 o LAN8720).
- Puerto de comunicación RS-485 con soporte Modbus RTU.
- Puerto de comunicación CAN Bus (MCP2515 + TJA1050).
- Fuente de alimentación conmutada reductora (*buck*) DC-DC.
- Conectores de expansión para módulos adicionales de E/S.

Desde el punto de vista del *firmware*, la incorporación de estos puertos de comunicación habilitará protocolos industriales estándar (Modbus RTU/TCP, CANopen) para la interoperabilidad con otros dispositivos programables.

## V. Conclusión

En el presente trabajo se desarrolla una placa basada en el microcontrolador ESP32, orientada a aplicaciones industriales e IIoT. El diseño integra periféricos esenciales para entornos industriales: entradas digitales optoaisladas, salidas de potencia mediante relés G5LE-1-E (10 A/250 VAC) y TRIAC BTA16-600B (16 A/600 V), memoria EEPROM AT24C32, reloj de tiempo real DS3231 con respaldo de batería y pantalla OLED, conformando una plataforma versátil y autónoma.

Se aplican criterios de diseño orientados a la robustez eléctrica, incluyendo aislamiento óptico en entradas y salidas, protección mediante diodos *freewheeling* en las etapas de relé, y dimensionamiento de pistas acorde a la corriente máxima admisible. En comparación con alternativas comerciales como Industrial Shields y NORVI, la placa propuesta se diferencia por su carácter completamente de código abierto, la incorporación de salidas TRIAC para conmutación de cargas AC sin desgaste mecánico, y un costo de producción estimado en aproximadamente 30 USD, significativamente inferior al de las soluciones existentes.

La versatilidad del ESP32 permite programar la placa en C con FreeRTOS, así como en lenguaje Ladder mediante OpenPLC, lo que lo hace compatible con el estándar IEC 61131-3 [5] y lo acerca al paradigma de los PLC industriales tradicionales.

Si bien el desarrollo presentado corresponde a un prototipo de primera versión, sienta las bases para una plataforma de mayor escala con incorporación futura de Ethernet, RS-485 y CAN Bus.

Su uso en el ámbito académico representa un aporte a la formación de ingenieros electrónicos en la automatización y sistemas embebidos. Los estudiantes no solo consolidan los conceptos teóricos, sino que también desarrollan competencias técnicas clave, preparándolos de manera óptima para enfrentar los desafíos profesionales en constante evolución.

## Agradecimientos

Ing. José Luis Simón: Ingeniero Electrónico (UNR). Actual Director del Departamento de Sistemas e Informática, FCEIA-UNR.

Ing. Ignacio Sosa: Ingeniero eléctrico con orientación electrónica (UNR). Ex Director del Departamento de Sistemas e Informática, FCEIA-UNR.

Nicolás Ramos: Estudiante avanzado de Ingeniería Electrónica, FCEIA-UNR.

## Referencias

[1] Espressif Systems, *ESP32 Series Datasheet*, Espressif Systems, [Disponible en:] https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

[2] Boot & Work Corp., S.L., “Industrial ESP32 PLC family,” https://www.industrialshields.com/industrial-hardware-solutions-based-on-esp32, 2024, consultado: junio 2025.

[3] NORVI Controllers, “NORVI IIOT – ESP32 industrial controller,” https://norvi.io/products/norvi-iiot-esp32-industrial-controller/, 2024, consultado: junio 2025.

[4] Autonomy Logic, “The OpenPLC Project,” https://autonomylogic.com, 2024.

[5] International Electrotechnical Commission, “IEC 61131-3: Programmable controllers – part 3: Programming languages,” International Electrotechnical Commission, Geneva, Switzerland, International Standard, 2013.

[6] U. N. d. R. Departamento de Sistemas e Informática (D.S.I.), F.C.E.I.A., “PLACA_IOT_INDUSTRIAL: Placa de desarrollo IoT industrial basada en ESP32,” https://github.com/sea2-dsi-fceia/PLACA_IOT_INDUSTRIAL, 2024, departamento de Sistemas e Informática, F.C.E.I.A., Universidad Nacional de Rosario.

[7] Amazon Web Services, *FreeRTOS Kernel Developer Docs*, Amazon Web Services, Inc., [Disponible en:] https://www.freertos.org/Documentation/RTOS_book.html.

[8] KiCad EDA, “KiCad – a cross platform and open source electronics design automation suite,” https://www.kicad.org/, 2024, versión 8.0.

ISBN 978-631-90145-4-9 - Congreso Argentino de Sistemas Embebidos CASE 2026 - 6 y 7 de agosto

# Nodo IoT para Monitoreo Inteligente de ruido y Localización de Fuente de contaminación sonora en Zonas Urbanas Residenciales

Ilber Adonayt Ruge Ruge
Programa de Ingenieria Electronica
Universidad Pedagogica y Tecnologica de Colombia
Tunja - Boyacá
ilber.ruge@uptc.edu.co

Juan Camilo Sánchez Gil
Programa de Ingenieria Electronica
Universidad Pedagogica y Tecnologica de Colombia
Tunja - Boyacá
juan.sanchez24@uptc.edu.co

Ingrid Carolina Ortiz Alvarez
Programa de Ingenieria Electronica
Universidad Pedagogica y Tecnologica de Colombia
Tunja - Boyacá
ingrid.ortiz05@uptc.edu.co

Fabian Rolando Jimenez Lopez
Programa de Ingenieria Electronica
Universidad Pedagogica y Tecnologica de Colombia
Tunja - Boyacá
fabian.jimenez02@uptc.edu.co

Edwin Javier Sanchez Uriza
Programa de Ingenieria Electronica
Universidad Pedagogica y Tecnologica de Colombia
Tunja - Boyacá
edwinjavier.sanchez@uptc.edu.co

***Resumen*— La contaminación sonora en áreas residenciales requiere sistemas de monitoreo continuo que permitan identificar tanto la intensidad como la orientación de las fuentes de ruido. Este trabajo presenta el diseño e implementación de un sistema embebido de Internet de las Cosas (IoT) basado en Arduino UNO para monitoreo inteligente de contaminación sonora. La propuesta integra un sensor acústico MAX4466, una brújula digital GY-271 y un mecanismo de barrido angular mediante servomotor para identificar la dirección predominante de la fuente sonora. El sistema incorpora almacenamiento local en microSD, visualización mediante una interfaz gráfica desarrollada en MATLAB App Designer y transmisión remota hacia ThingSpeak. Los resultados experimentales evidenciaron errores inferiores a 1 dB en el rango de 50–79 dB, validando la viabilidad del sistema para aplicaciones de monitoreo acústico residencial.**

***Palabras clave*—** *Contaminación sonora, Monitoreo acústico, Arduino UNO, ThingSpeak*

## I. Introducción

La contaminación sonora se ha convertido en una de las problemáticas ambientales más relevantes en entornos urbanos debido a su impacto directo sobre la salud y el bienestar de la población. La exposición prolongada a niveles elevados de ruido en zonas residenciales puede generar alteraciones del sueño, estrés, disminución del rendimiento cognitivo y enfermedades cardiovasculares, incluso cuando las intensidades acústicas no superan niveles considerados extremos. Estudios recientes han evidenciado que la dinámica temporal y espacial del ruido urbano presenta una alta variabilidad asociada al tráfico vehicular, actividades comerciales y comportamiento social, lo que limita la efectividad de los mapas acústicos estáticos utilizados tradicionalmente en planificación urbana y control ambiental [1]. En este contexto, la necesidad de implementar sistemas de monitoreo continuo y distribuido ha impulsado el desarrollo de soluciones tecnológicas orientadas a la adquisición de datos acústicos en tiempo real [2].

El avance de plataformas embebidas de bajo costo y tecnologías Internet de las Cosas (IoT) ha favorecido la aparición de Redes Inalámbricas de Sensores Acústicos (WASN), capaces de realizar monitoreo ambiental de forma escalable y accesible [3]. Investigaciones recientes demuestran que microcontroladores como Arduino y ESP8266 permiten implementar nodos acústicos con capacidades de adquisición, procesamiento local y transmisión remota de información sonora [4]. Asimismo, se han desarrollado sistemas capaces de integrar técnicas de monitoreo inteligente y análisis distribuido para la supervisión acústica en ciudades inteligentes, mejorando la cobertura espacial y la disponibilidad de datos ambientales [5],[6]. En este trabajo se presenta el diseño e implementación de un sistema IoT de bajo costo para monitoreo direccional de contaminación sonora en áreas residenciales, integrando adquisición acústica, referencia espacial, almacenamiento local y telemetría inalámbrica para supervisión remota en tiempo real.

## II. Metodología

La metodología propuesta se fundamenta en el desarrollo de un sistema embebido orientado al monitoreo inteligente de contaminación sonora en entornos residenciales, integrando adquisición, procesamiento, almacenamiento y visualización de señales acústicas. Como se observa en la Figura 1, la arquitectura del sistema se compone de un módulo de captura de audio, una unidad de procesamiento central, un sistema de referencia espacial y una etapa de almacenamiento local para el registro histórico de mediciones. A diferencia de los sistemas convencionales orientados únicamente a la cuantificación de niveles de presión sonora, la propuesta incorpora un mecanismo automatizado de barrido angular que permite identificar la orientación predominante de la fuente de ruido dentro del entorno monitoreado, proporcionando una caracterización acústica más completa mediante la integración de variables de intensidad y dirección [4],[7]. Para ello, el sistema realiza el procesamiento digital de la señal acústica mediante el cálculo del valor eficaz (RMS), permitiendo estimar el nivel de presión sonora expresado en decibelios (dB) [8], mientras que la referencia espacial posibilita asociar cada medición con una orientación angular específica. Adicionalmente, la arquitectura implementa almacenamiento local de datos en formato CSV y una interfaz gráfica desarrollada en MATLAB para la supervisión y análisis histórico de tendencias acústicas, estableciendo una base experimental para futuras aplicaciones de monitoreo ambiental inteligente mediante plataformas embebidas.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

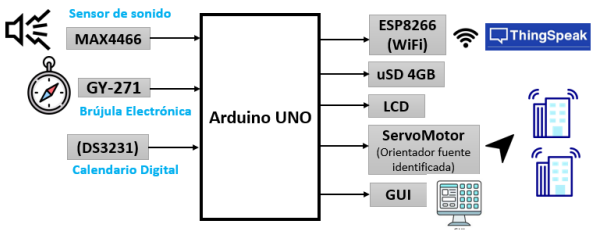



Fig. 1. Diagrama de bloques del monitor inteligente de contaminación sonora.

El sistema propuesto utiliza la plataforma Arduino UNO con microcontrolador ATmega328P como unidad central para la adquisición y gestión de variables monitoreadas. La información acústica se captura mediante un sensor MAX4466 con preamplificador de bajo ruido y salida analógica ajustable, apropiado para monitoreo ambiental [4], [9]. La orientación espacial se determina mediante una brújula digital GY-271 con magnetómetro de tres ejes e interfaz I2C, permitiendo estimar la dirección predominante de la fuente sonora [3]. Un servomotor automatiza el barrido angular del micrófono. Arduino UNO fue seleccionado por su bajo costo, disponibilidad y capacidad para adquisición, procesamiento, almacenamiento y comunicaciones.

El sistema incorpora una pantalla de cristal líquido (LCD) para la visualización local en tiempo real de las variables monitoreadas. El almacenamiento de información se realiza mediante una memoria microSD de 4 GB bajo el protocolo Serial Peripheral Interface (SPI), registrando datos históricos en formato Comma-Separated Values (CSV). Un módulo Real-Time Clock (RTC) proporciona la referencia temporal de cada medición almacenada. Para supervisión avanzada, el sistema establece comunicación serial Recommended Standard 232 (RS232) con una interfaz gráfica desarrollada en MATLAB App Designer, permitiendo el monitoreo en tiempo real y el análisis histórico. Además, un módulo WiFi ESP8266 transmite los datos hacia la plataforma ThingSpeak mediante comunicación Universal Asynchronous Receiver-Transmitter (UART) a 3.3 V [10], [11]. La integración física de los módulos implementados se presenta en la Figura 2.

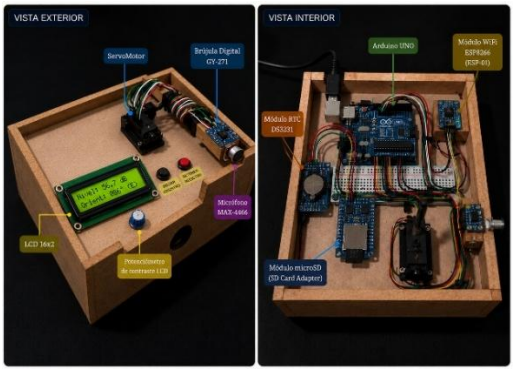



Fig. 2. Prototipo físico del monitor inteligente de contaminación sonora.

Las variables de interés consideradas en la propuesta corresponden al nivel de presión sonora expresado en decibelios (dB), el valor eficaz de la señal acústica (VRMS) y la orientación angular de la fuente predominante de ruido [8]. El cálculo de estas variables se realiza a partir del procesamiento digital de la señal acústica adquirida por el sensor de sonido y muestreada mediante el conversor analógico-digital del microcontrolador. Debido a que la señal presenta comportamiento alterno alrededor de un nivel de offset, se implementa el cálculo del valor eficaz (RMS) con el fin de estimar la energía efectiva de la señal acústica, como se expresa en (1):

$$V_{RMS} = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1}(v[n] - V_{offset})^2} \quad (1)$$

A partir del valor RMS calculado, el sistema estima el nivel de presión sonora expresado en decibelios (dB) mediante una aproximación logarítmica calibrada experimentalmente, como se muestra en (2):

$$L_P \approx 20Log_{10}\left(\frac{V_{RMS}}{V_{ref}}\right) + K \quad (2)$$

donde V*ref* representa un valor de referencia y K corresponde al factor de calibración obtenido durante el proceso experimental. Este procedimiento permite convertir la señal analógica capturada en niveles de intensidad sonora asociados al entorno monitoreado.

Para la identificación direccional de la fuente contaminante, el sistema implementa un mecanismo de barrido angular mediante servomotor, registrando el nivel de intensidad sonora en diferentes orientaciones espaciales. La referencia angular es obtenida mediante la brújula digital a partir de las componentes horizontales del campo magnético terrestre, utilizando la relación, como se muestra en (3):

$$\theta = atan2(M_y, M_x)x\frac{180}{\pi} \quad (3)$$

donde M*x* y M*y* corresponden a las componentes medidas por el magnetómetro. Durante el proceso de barrido, el sistema compara los niveles acústicos registrados en cada orientación y selecciona automáticamente la dirección asociada a la mayor intensidad sonora, permitiendo orientar el micrófono hacia la fuente predominante de contaminación acústica.

El procesamiento de las variables de interés se implementó en Arduino IDE mediante lenguaje C/C++. El firmware realiza la adquisición acústica, el cálculo RMS, la estimación del nivel de presión sonora, el control del barrido angular y la lectura de orientación espacial. La comunicación con la brújula digital GY-271 y el módulo RTC utiliza Wire.h bajo protocolo I2C. El almacenamiento histórico en memoria microSD emplea SPI.h y SD.h para generar archivos CSV. El servomotor y la pantalla LCD se controlan mediante Servo.h y LiquidCrystal.h. La comunicación UART se configuró a 9600 baudios para el intercambio de datos con MATLAB App Designer y el módulo ESP8266, utilizando ThingSpeak.h para la transmisión hacia la plataforma IoT.

La caracterización y calibración del sensor acústico se realizaron para garantizar la confiabilidad de las mediciones de intensidad sonora. El módulo fue alimentado con 5 VDC, generando una señal analógica adquirida por el convertidor analógico-digital (ADC) de 10 bits del Arduino UNO, con un rango de 0 a 1023 niveles digitales. Para estimar la energía efectiva de la señal se implementó un algoritmo de cálculo RMS, utilizado para relacionar experimentalmente el valor VRMS con el nivel de presión sonora en decibelios (dB). La calibración se efectuó comparando las mediciones del

prototipo con la aplicación SoundMeter para Android, ubicando ambos dispositivos entre 5 y 10 cm de una fuente sonora de 200 Hz. Los pares experimentales VRMS-dB se presentan en la Tabla I, obteniéndose una aproximación adecuada para la función de conversión. La validación metrológica con instrumentación certificada se plantea como trabajo futuro.

TABLA I. DATOS EXPERIMENTALES DE CALIBRACION.

| Vrms | dB | Vrms | dB | Vrms | dB |
|---|---|---|---|---|---|
| **0.0663** | 21 | 0.0929 | 29 | 1.9084 | 74 |
| **0.0636** | 20 | 0.0931 | 30 | 2.1065 | 76 |
| **0.0719** | 23 | 0.0940 | 31 | 2.2048 | 78 |
| **0.0737** | 24 | 0.1000 | 40 | 2.3132 | 79 |
| **0.0720** | 23 | 0.1254 | 46 | 2.3471 | 80 |
| **0.0707** | 22 | 0.1556 | 50 | 2.3826 | 81 |
| **0.0765** | 24 | 0.2208 | 55 | 2.4125 | 82 |
| **0.0817** | 25 | 0.3852 | 60 | 2.4577 | 83 |
| **0.0825** | 26 | 0.7183 | 65 | 2.4933 | 86 |
| **0.0855** | 28 | 1.1065 | 68 | 2.5000 | 87 |

Para la supervisión y análisis de las variables acústicas medidas se realizó una interfaz gráfica desarrollada en MATLAB App Designer, permitiendo visualización en tiempo real, análisis estadístico y representación histórica de los niveles de presión sonora y orientación detectada. La Figura 3 presenta la interface gráfica de usuario GUI implementada para la gestión local del sistema de monitoreo acústico.

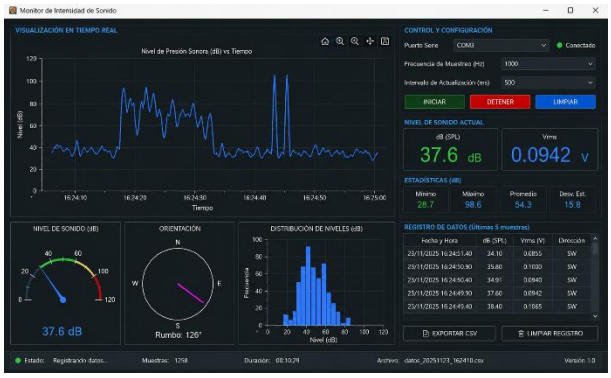



Fig. 3. GUI del monitor inteligente de contaminación sonora.

## III. ANÁLISIS DE RESULTADOS

La validación experimental del sistema se realizó comparando las mediciones del monitor acústico con los valores registrados por la aplicación patrón SoundMeter. Los resultados presentados en la Tabla II evidencian errores inferiores o iguales a 1 dB en el rango de 50 a 79 dB, confirmando la efectividad del ajuste logarítmico implementado para aplicaciones de monitoreo residencial. Sin embargo, en niveles bajos de intensidad acústica cercanos a 22 dB se observó una sobreestimación aproximada de 13 dB, atribuida al fenómeno de noise floor característico de sensores acústicos de bajo costo. A pesar de esta desviación, el sistema obtuvo un Error Cuadrático Medio (RMSE) de aproximadamente 4.45 dB. Dado que la propuesta está orientada al monitoreo de ruido ambiental en zonas residenciales, cuyos límites normativos se encuentran entre 55 y 65 dB según la reglamentación colombiana [12], el desempeño alcanzado resulta adecuado para la detección y seguimiento de eventos asociados a contaminación sonora.

TABLA II. VALIDACIÓN EXPERIMENTAL DEL SENSOR ACÚSTICO.

| dB Ref (App) | dB Prot. (Monitor) | Error E (dB) | Error $E^2$ ($dB^2$) |
|---|---|---|---|
| **79** | 79 | 0 | 0 |
| **76** | 76 | 0 | 0 |
| **71** | 71 | 0 | 0 |
| **65** | 64 | 1 | 1 |
| **60** | 59 | 1 | 1 |
| **56** | 56 | 0 | 0 |
| **54** | 53 | 1 | 1 |
| **51** | 50 | 1 | 1 |
| **46** | 41 | 5 | 25 |
| **22** | 35 | 13 | 169 |

La identificación direccional constituyó una de las principales contribuciones del sistema, integrando información de intensidad acústica y orientación espacial durante el monitoreo. El micrófono y la brújula digital GY-271 fueron montados sobre un soporte acoplado a un servomotor SG90, permitiendo realizar barridos angulares en cinco posiciones discretas separadas 45° (0°, 45°, 90°, 135° y 180°). En cada posición se efectuó una adquisición de datos durante 5 s, registrando simultáneamente el nivel de presión sonora y la orientación espacial para identificar la dirección asociada a la mayor intensidad acústica detectada [13]. El ciclo completo de exploración requirió aproximadamente 25 s. Para minimizar perturbaciones electromagnéticas, la brújula fue ubicada a 10 cm del servomotor y las lecturas se realizaron únicamente después de finalizar cada movimiento. La calibración experimental presentada en la Figura 4 confirmó la consistencia de las mediciones angulares obtenidas.




Fig. 4. Validación experimental de orientación mediante brújula digital GY-271.

El prototipo fue concebido como un nodo de monitoreo acústico portátil alimentado mediante una batería externa (power bank) de 5 VDC y 15000 mAh, diseñado para operar en ambientes interiores o exteriores cubiertos debido a la ausencia de encapsulado de protección. Las pruebas experimentales se realizaron en entornos residenciales con fuentes sonoras controladas y ruido ambiental cotidiano. El consumo energético, medido con un UNI-T UT658, registró valores promedio de 5.08 V, 0.32 A y 1.63 W, evidenciando una operación estable y de bajo consumo. Estos valores permitieron estimar una autonomía superior a 40 horas. Los resultados demostraron una identificación consistente de la dirección de máxima intensidad acústica y una adecuada visualización espacial de los focos de ruido mediante MATLAB.

La visualización y almacenamiento de datos constituyen un componente fundamental del sistema, permitiendo supervisión en tiempo real y análisis histórico del comportamiento acústico. La interfaz gráfica desarrollada en MATLAB App Designer, presentada en la Figura 5, permite representar la evolución temporal de los niveles de presión sonora e incorpora herramientas de inspección, análisis estadístico y selección de ventanas temporales, facilitando la identificación de eventos acústicos y el análisis de tendencias sonoras en diferentes intervalos de tiempo.

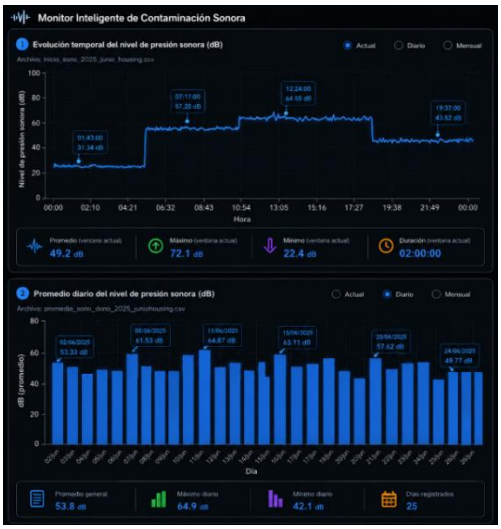



Fig. 5. GUI para visualización y almacenamiento de datos acústicos del sistema IoT.

La integración con la plataforma IoT ThingSpeak permitió implementar supervisión remota y almacenamiento en la nube de las variables acústicas registradas por el sistema. Durante las pruebas experimentales, el módulo ESP8266 transmitió periódicamente niveles de presión sonora y orientación detectada hacia canales de visualización remota accesibles desde Internet. Como se muestra en la Figura 6, ThingSpeak permitió registrar históricamente las variaciones de intensidad sonora, facilitando análisis temporal y monitoreo distribuido. Asimismo, la sincronización entre el almacenamiento local en microSD y la plataforma IoT proporcionó redundancia de información y mayor confiabilidad ante interrupciones de conectividad. La integración desarrollada demostró viabilidad para aplicaciones de monitoreo acústico residencial inteligente y supervisión ambiental distribuida.

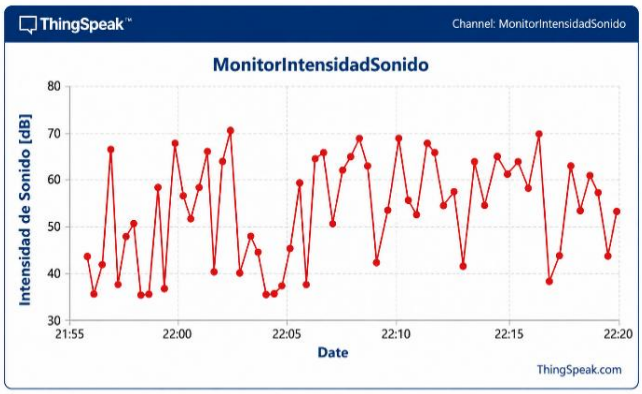



Fig. 6. Visualización remota de intensidad sonora mediante ThingSpeak.

## IV. Conclusiones

Se desarrolló e implementó un sistema IoT embebido de bajo costo orientado al monitoreo inteligente de contaminación sonora, integrando adquisición acústica, identificación direccional, almacenamiento local y supervisión remota. La validación experimental evidenció un comportamiento adecuado en el rango de operación residencial, obteniendo errores inferiores a 1 dB en la mayoría de las mediciones comprendidas entre 50 y 79 dB, lo que demuestra la viabilidad de utilizar plataformas basadas en Arduino UNO y sensores acústicos económicos para aplicaciones de monitoreo ambiental.

La incorporación de un sistema de barrido angular mediante servomotor y brújula digital permitió identificar la orientación predominante de la fuente contaminante, aportando información espacial adicional frente a soluciones convencionales centradas únicamente en la medición de intensidad sonora. Asimismo, la integración de visualización local, almacenamiento histórico en microSD y comunicación IoT mediante ThingSpeak proporciona una plataforma funcional para supervisión residencial en tiempo real y futuras aplicaciones relacionadas con análisis inteligente de ruido ambiental y ciudades inteligentes.

## Referencias

[1] X. Zhang and S. Zhou, “Building a City with Low Noise Pollution: Exploring the Mental Health Effect Thresholds of Spatiotemporal Environmental Noise Exposure and Urban Planning Solution,” International Journal of Environmental Research and Public Health, vol. 20, no. 5, p. 4222, 2023.

[2] R. Subraja, T. Meyyapan, A. Padmapriya and S. Santhosh Kumar, “Smart noise pollution monitoring system,” World Journal of Advanced Engineering Technology and Sciences, vol. 13, no. 1, pp. 899–908, 2024. DOI: 10.30574/wjaets.2024.13.1.0472.

[3] A. Mishra, “Edge Computing and IoT-Based Urban Environment Monitoring Solutions,” Computational Engineering and Technology Innovations, vol. 1, no. 3, pp. 139–149, 2024. DOI: 10.48314/ceti.v1i3.34.

[4] J. Picaut, A. Can, N. Fortin, J. Ardouin and M. Lagrange, “Low-Cost Sensors for Urban Noise Monitoring Networks—A Literature Review,” Sensors, vol. 20, no. 8, p. 2256, 2020.

[5] B. Peng, K. I.-K. Wang and W. H. Abdulla, “An Integrated Hierarchical Wireless Acoustic Sensor Network and Optimized Deep Learning Model for Scalable Urban Sound and Environmental Monitoring,” Applied Sciences, vol. 15, no. 4, p. 2196, 2025.

[6] R. M. Alsina-Pagès, F. Alías and M. Hervás, “Noise Events Monitoring for Urban and Mobility Planning in Smart Cities Using WASN,” Environments, vol. 6, no. 2, p. 24, 2019.

[7] J.-A. Fernández-Prieto, J. Cañada-Bago and M.-A. Gadeo-Martos, “Wireless Acoustic Sensor Nodes for Noise Monitoring in the City of Linares (Jaén),” Sensors, vol. 20, no. 1, p. 124, 2020.

[8] L. E. Kinsler, A. R. Frey, A. B. Coppens and J. V. Sanders, Fundamentals of Acoustics, 4th ed. New York, NY, USA: John Wiley & Sons, 2000.

[9] E. Vidaña-Vila, J. M. Cobos and F. J. Ferri, “Low-Cost Distributed Acoustic Sensor Network for Real-Time Urban Sound Monitoring,” Electronics, vol. 9, no. 12, p. 2119, 2020.

[10] R. M. Alsina-Pagès, F. Alías, M. Hervás and C. Canalís, “Design of a Mobile Low-Cost Sensor Network Using Urban Buses for Real-Time Ubiquitous Noise Monitoring,” Sensors, vol. 17, no. 1, p. 57, 2017.

[11] J. S. Botero-Valencia, D. E. Atehortúa and J. A. Serna-Montoya, “Low-Cost Air, Noise, and Light Pollution Measuring Station with IoT Connectivity,” HardwareX, vol. 13, e00373, 2023.

[12] Ministerio de Ambiente, Vivienda y Desarrollo Territorial, Resolución 0627 de 2006: Por la cual se establece la norma nacional de emisión de ruido y ruido ambiental, Bogotá, Colombia, 2006.

[13] L. Luo, D. Wu and J. Zhang, “Wireless Sensor Networks for Noise Measurement and Acoustic Event Recognitions,” Sensors, vol. 20, no. 7, p. 2093, 2020.

Categoría

# Foro Tecnológico

Protocolos y Comunicaciones

Área Temática

# Alta Disponibilidad y Resiliencia en Redes SD-WAN

Luis Miguel Duarte, Juan Galleguillo
Maestría en Ciencias de la Ingeniería Mención Telecomunicaciones
Facultad de Ciencias Exactas, Físicas y Naturales
Universidad Nacional de Córdoba
Córdoba, Argentina
luis.duarte@mi.unc.edu.ar, juan.galleguillo@unc.edu.ar

***Resumen*—Este estudio aborda la transición de las arquitecturas de red Wide Area Network (WAN) convencionales hacia modelos definidos por software, con el objetivo de mejorar la alta disponibilidad y la resiliencia operativa en infraestructuras corporativas. Ante las limitaciones de flexibilidad y costos de las redes MPLS tradicionales, se plantea un modelo de implementación SD-WAN híbrido que integra transporte MPLS e Internet mediante una orquestación centralizada. La metodología comprende el diseño de una topología Hub-and-Spoke en el entorno GNS3, empleando firewalls de próxima generación (NGFW) para la abstracción de los planos de red. Además, se evalúan mecanismos de conmutación automática (failover) y balanceo de carga dinámico basados en métricas de desempeño en tiempo real, como latencia, jitter y pérdida de paquetes. Los resultados evidencian que la integración de túneles IPsec con políticas orientadas a la aplicación permite mantener la continuidad del servicio, preservar la operación del negocio y mejorar el uso del ancho de banda sin intervención manual. Se concluye que SD-WAN constituye una solución robusta para incrementar la visibilidad de la red en entornos distribuidos.**

***Palabras clave*—SD-WAN, MPLS, alta disponibilidad, SLA, failover, balanceo de carga, redes WAN.**

## I. Introducción

Las redes WAN empresariales han evolucionado significativamente impulsadas por la adopción de servicios en la nube, aplicaciones distribuidas y entornos híbridos. Tradicionalmente, MPLS (Multiprotocol Label Switching) ha sido la tecnología predominante debido a su capacidad para ofrecer calidad de servicio (QoS), ingeniería de tráfico y segmentación mediante VRFs (Virtual Routing and Forwarding). Sin embargo, las redes MPLS pueden presentar limitaciones asociadas a su menor flexibilidad operativa, dependencia de servicios provistos por operadores de telecomunicaciones y mayores costos frente a alternativas basadas en SD-WAN [1].

En este contexto, SD-WAN (Software-Defined Wide Area Network) emerge como un paradigma que desacopla el plano de control del plano de datos, permitiendo la orquestación centralizada de políticas y el uso simultáneo de múltiples medios de transporte. Esta tecnología habilita decisiones de enrutamiento basadas en desempeño en tiempo real, optimizando la experiencia de usuario y la resiliencia de la red.

El presente trabajo propone y valida una arquitectura híbrida SD-WAN/MPLS, orientada a garantizar alta disponibilidad mediante mecanismos de conmutación automática y balanceo dinámico, evaluados en un entorno controlado de laboratorio [2], [3], [4].

## II. Fundamentación Técnica

### *II-A. Redes MPLS (Multiprotocol Label Switching)*

MPLS basa su funcionamiento en la conmutación de etiquetas en lugar del reenvío por direcciones IP, lo que permite trayectorias predecibles mediante LSPs (Label Switched Paths). Su ventaja principal es la integración de servicios como L3VPN (Layer 3 Virtual Private Network) y la clasificación de tráfico en clases de servicio (CoS) para priorizar aplicaciones críticas [5], [6].

### *II-B. Arquitectura SD-WAN*

Este modelo separa los planos de control, datos y gestión. Se compone de un Controlador (lógica centralizada), un Orquestador (gestión global y Zero-Touch Provisioning) y equipos Edge o CPE que ejecutan las políticas y establecen túneles IPsec cifrados. A diferencia de MPLS, SD-WAN permite el uso de transportes heterogéneos bajo un esquema multi-path.

### *II-C. Alta disponibilidad en WAN*

La alta disponibilidad se logra mediante redundancia de enlaces y mecanismos de failover. SD-WAN mejora este enfoque mediante:

- Monitoreo continuo de SLA
- Selección dinámica de rutas
- Balanceo activo-activo

La arquitectura SD-WAN permite la incorporación de enlaces LTE o 5G como respaldo dinámico (backup link) para servicios críticos. Estos actúan como una ruta inalámbrica alternativa capaz de mantener la disponibilidad de aplicaciones esenciales (como VoIP o sistemas financieros) durante fallas en los medios cableados principales [6].

## III. Metodología y Modelo Propuesto

Como se muestra en la figura 1, la arquitectura propuesta consiste en una sede principal (Hub) y sucursales interconectadas mediante enlaces MPLS e Internet a los cuales se les asignó un nombre para su reconocimiento, la simulación de los firewalls se realizó con equipos Fortinet [7]. La arquitectura diseñada integra una red híbrida simulada en GNS3 [8], [9].

En la Fig. 2 se muestra la topología implementada, y en la Tabla I se describen las asignaciones de direccionamiento correspondientes.

Por limitaciones técnicas del software de emulación (GNS3), la conectividad LTE no fue implementada en el escenario de laboratorio. No obstante, el diseño propuesto es plenamente compatible con el uso de equipos CPE modernos que integran módulos 4G/5G [7], [10] para una gestión unificada del respaldo desde el orquestador.

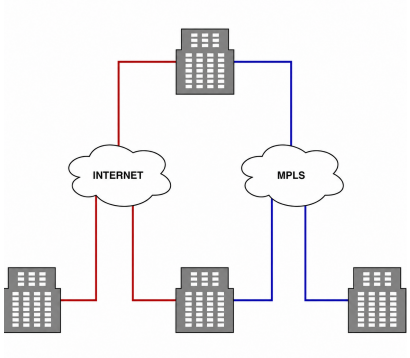



Fig. 1: Modelo de red propuesto

### III-A. *Diseño de la Topología y Nodos*

- **Sede Miramontes (Hub):** Punto central de concentración conectado a MPLS e Internet
- **Sitios Remotos:** Lomas (MPLS/Internet), Tepeyac (MPLS) y Roatán (Internet vía túnel S2S).
- **Seguridad:** Implementación de firewalls NGFW para filtrado de paquetes con estado, control de aplicaciones e inspección SSL.
- **Segmentación de Red:** Se definieron direccionamientos específicos para las interfaces WAN (redes 172.16.x.x y 190.16.x.x) y redes LAN (192.168.x.0/24) para simular el tráfico corporativo.

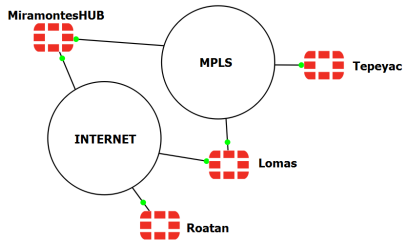



Fig. 2: Red Simulada en GNS3

| Sitio | WAN | LAN | Conectividad |
|---|---|---|---|
| **Miramontes HUB** | 172.16.1.2/30<br>190.16.1.2/30 | 192.168.10.0/24 | MPLS / Internet |
| Tepeyac | 172.16.2.2/30 | 192.168.20.0/24 | MPLS |
| Lomas | 172.16.3.2/30<br>190.16.3.2/30 | 192.168.30.0/24 | MPLS / Internet |
| Roatán | 190.16.4.2/30 | 192.168.40.0/24 | Internet |

Tabla I: Direcciones WAN/LAN y conectividad por sitio

### III-B. *Componentes de Software y Orquestación*

Para la abstracción de los planos de control y datos, se emplearon instancias virtuales de firewalls de próxima generación (NGFW FortiGate) [7] como equipos Edge para la ejecución de políticas de seguridad y enrutamiento. La gestión se centralizó mediante un orquestador que habilitó el Zero-Touch Provisioning, la visibilidad de telemetría en tiempo real y la generación de tableros de control para métricas históricas. Asimismo, las redes LAN fueron emuladas con nodos virtuales para la validación de conectividad mediante el protocolo ICMP.

### III-C. *Configuración de Parámetros de Performance SLA*

La metodología incluyó la definición de umbrales críticos de rendimiento dentro del motor de SD-WAN para la toma de decisiones automática indicados en la figura 3.

- **Latencia máxima:** 100 ms.
- **Jitter máximo:** 100 ms.
- **Pérdida de paquetes permitida:** 10 %.
- **Frecuencia de monitoreo:** Mediciones constantes de calidad hacia los underlays para determinar el cumplimiento de los acuerdos de nivel de servicio (SLA).

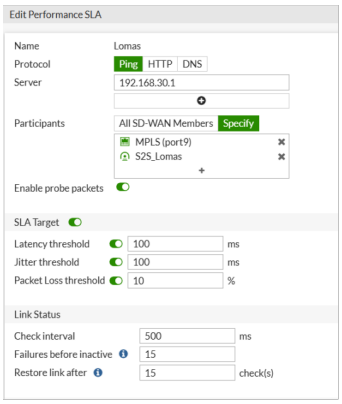



Fig. 3: Parámetros configurados

### III-D. *Escenarios de Validación (Pruebas de Estrés)*

La efectividad de la arquitectura se validó mediante la simulación de fallas de enlace. El procedimiento consistió en:

1. Establecer una sesión de tráfico constante entre la LAN de un sitio remoto (ej. Lomas) y el Hub.
2. Desactivación manual de la interfaz MPLS para forzar una condición de falla.
3. Monitoreo del tiempo de convergencia y verificación del reencaminamiento inmediato del tráfico a través del túnel IPsec sobre Internet, asegurando la continuidad del servicio sin intervención manual.

## IV. Resultados

En esta sección se presentan los datos obtenidos durante la fase de validación en el entorno emulado, enfocándose en la conectividad inicial, la observabilidad del tráfico y la resiliencia del sistema ante fallas de enlace.

### *IV-A. Validación de Conectividad Inter-sitios*

Las pruebas iniciales, ilustradas en las Figs. 4–6, confirmaron la correcta configuración del enrutamiento y los túneles VPN. Mediante el uso de paquetes ICMP, se verificó la comunicación desde las redes LAN de los sitios remotos hacia el Hub central. Como se detalla en las pruebas de comunicación, la latencia promedio en condiciones normales se mantuvo por debajo de los 10 ms para los sitios con conectividad MPLS directa, validando la estabilidad de los underlays.

```
PC3> show ip

NAME        : PC3[1]
IP/MASK     : 192.168.20.5/24
GATEWAY     : 192.168.20.1
DNS         :
MAC         : 00:50:79:66:68:00
LPORT       : 10001
RHOST:PORT  : 127.0.0.1:10002
MTU:        : 1500

PC3> ping 192.168.10.5
84 bytes from 192.168.10.5 icmp_seq=1 ttl=59 time=8.797 ms
84 bytes from 192.168.10.5 icmp_seq=2 ttl=59 time=4.527 ms
84 bytes from 192.168.10.5 icmp_seq=3 ttl=59 time=8.962 ms
84 bytes from 192.168.10.5 icmp_seq=4 ttl=59 time=6.172 ms
84 bytes from 192.168.10.5 icmp_seq=5 ttl=59 time=7.644 ms
```

Fig. 4: Prueba de comunicación Tepeyac - HUB

```
PC2> show ip

NAME        : PC2[1]
IP/MASK     : 192.168.30.5/24
GATEWAY     : 192.168.30.1
DNS         :
MAC         : 00:50:79:66:68:01
LPORT       : 20208
RHOST:PORT  : 127.0.0.1:20209
MTU         : 1500

PC2> ping 192.168.10.5

84 bytes from 192.168.10.5 icmp_seq=1 ttl=59 time=4.348 ms
84 bytes from 192.168.10.5 icmp_seq=2 ttl=59 time=3.787 ms
84 bytes from 192.168.10.5 icmp_seq=3 ttl=59 time=3.869 ms
84 bytes from 192.168.10.5 icmp_seq=4 ttl=59 time=4.852 ms
84 bytes from 192.168.10.5 icmp_seq=5 ttl=59 time=3.504 ms
```

Fig. 5: Prueba de comunicación Lomas - HUB

```
CLI Console (1)
Roatan # execute ping-options source 192.168.40.1

Roatan # execute ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1): 56 data bytes
64 bytes from 192.168.10.1: icmp_seq=0 ttl=255 time=11.2 ms
64 bytes from 192.168.10.1: icmp_seq=1 ttl=255 time=5.9 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=255 time=6.1 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=255 time=7.2 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=255 time=8.7 ms

--- 192.168.10.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 5.9/7.8/11.2 ms

Roatan # execute ping 192.168.20.1
PING 192.168.20.1 (192.168.20.1): 56 data bytes
64 bytes from 192.168.20.1: icmp_seq=0 ttl=251 time=8.5 ms
64 bytes from 192.168.20.1: icmp_seq=1 ttl=251 time=48.4 ms
64 bytes from 192.168.20.1: icmp_seq=2 ttl=251 time=13.5 ms
64 bytes from 192.168.20.1: icmp_seq=3 ttl=251 time=11.2 ms
64 bytes from 192.168.20.1: icmp_seq=4 ttl=251 time=8.3 ms

--- 192.168.20.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 8.3/17.9/48.4 ms

Roatan #
```

Fig. 6: Prueba desde Roatán (Internet) hacia HUB y Tepeyac

### *IV-B. Análisis de Telemetría y Visibilidad de Aplicación*

El uso de un orquestador centralizado permitió obtener una granularidad superior en la supervisión del tráfico en comparación con los modelos WAN tradicionales.

- **Distribución de Carga:** La Fig. 7 indica la identificación del consumo por interfaz de origen (LAN, MPLS, S2S_Hub), donde el sistema muestra volúmenes de tráfico de 7.30 MB en la LAN y 2.85 MB a través del túnel S2S, demostrando un balanceo dinámico efectivo.

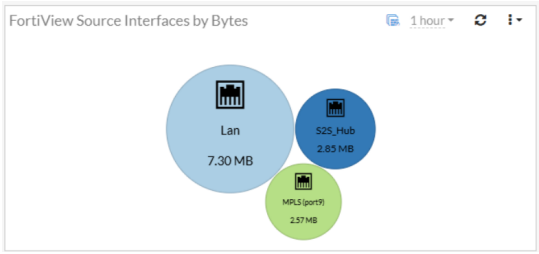



Fig. 7: Consumo por interfaz

- **Detalle del tráfico:** Se logró identificar el consumo categorizándolo por su origen (IP), desde el orquestador central, lo cual se detalla en las Figs. 8 y 9:

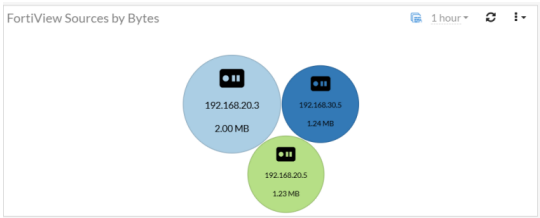



Fig. 8: Consumo por origen

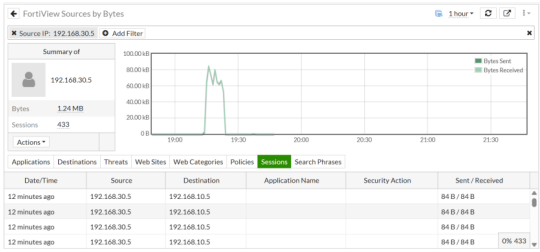



Fig. 9: Detalle del consumo por origen

### *IV-C. Desempeño de SLA y Resiliencia (Failover)*

La prueba crítica consistió en la degradación provocada del enlace MPLS en el sitio ”Lomas”, el cual cuenta con conexión MPLS e Internet, esto para evaluar la correcta conmutación y resiliencia del sistema.

- **Activación de Umbrales:** El sistema fue configurado con umbrales de SLA de 100 ms de latencia/jitter y un 10 % de pérdida de paquetes.
- **Detección de falla:** Al desactivar la interfaz física (port9, Puerto WAN), el monitoreo de Performance SLA registró un incremento crítico en la pérdida de paquetes (alcanzando un 24 % en el momento del evento) y una caída inmediata de la conectividad MPLS, representada por la línea azul en el gráfico de telemetría y manteniendo la comunicación por el túnel S2S vía Internet, esto ilustrado en la figura 10. Este valor del 24 %, al superar

ampliamente el umbral del 10 % configurado, disparó el algoritmo de selección de ruta, validando la sensibilidad del motor de SD-WAN.

- **Conmutación Automática:** Como resultado de la política de SD-WAN, el tráfico fue reencaminado de forma transparente a través del túnel S2S_Hub (Internet). Las pruebas de comunicación continua evidenciaron que, durante la indisponibilidad del enlace MPLS, la conectividad a través de Internet se mantuvo estable, sin interrupción de las sesiones activas. Este comportamiento permitió preservar la alta disponibilidad del servicio. En la figura 11, se identifica el instante de la conmutación, representado por las líneas rojas.

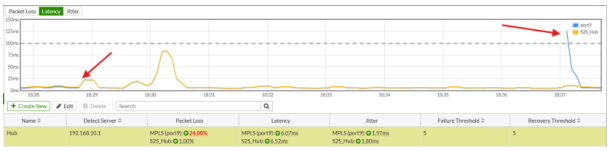


Fig. 10: Prueba de Conmutación

```
PC2> show ip

NAME        : PC2[1]
IP/MASK     : 192.168.30.5/24
GATEWAY     : 192.168.30.1
DNS         :
MAC         : 00:50:79:66:68:01
LPORT       : 20228
RHOST:PORT  : 127.0.0.1:20229
MTU         : 1500

PC2>  ping 192.168.10.5

84 bytes from 192.168.10.5 icmp_seq=1 ttl=62 time=5.113 ms
84 bytes from 192.168.10.5 icmp_seq=2 ttl=62 time=8.225 ms
84 bytes from 192.168.10.5 icmp_seq=3 ttl=62 time=12.905 ms
84 bytes from 192.168.10.5 icmp_seq=4 ttl=62 time=8.494 ms
84 bytes from 192.168.10.5 icmp_seq=5 ttl=62 time=5.996 ms
```

Fig. 11: Comunicación Lomas - Hub vía Internet

### *IV-D. Recuperación del Servicio (Switch-back)*

Una vez restaurada la interfaz MPLS, el motor de SD-WAN detectó que el enlace volvía a cumplir con los umbrales de recuperación configurados (5 checks exitosos). El sistema realizó la transición de retorno al enlace primario de forma automática, optimizando nuevamente el uso de los recursos de transporte según la prioridad definida en el orquestador.

## V. Conclusiones

El desarrollo de esta investigación permitió validar la transición de las arquitecturas WAN tradicionales hacia modelos definidos por software, demostrando que la integración de SD-WAN sobre infraestructuras MPLS constituye una evolución crítica para la resiliencia corporativa. Las conclusiones derivadas de este estudio se detallan a continuación:

### *V-A. Cambio de Paradigma y Eficiencia*

La arquitectura SD-WAN marca un cambio fundamental desde redes rígidas hacia modelos autoadaptativos, donde las decisiones de enrutamiento se basan en el desempeño real (latencia, jitter, pérdida de paquetes) y políticas de negocio, en lugar de topologías estáticas.

### *V-B. Alta Disponibilidad y Resiliencia*

Los mecanismos de failover automático y balanceo dinámico activo-activo garantizan la continuidad operativa incluso ante fallas críticas de enlace. Los resultados experimentales confirman que este modelo permite reducir drásticamente el Tiempo Medio de Recuperación (MTTR), manteniendo niveles de disponibilidad superiores al 99.9 % (SLA) sin requerir intervención manual.

### *V-C. Seguridad y Visibilidad Centralizada*

La implementación demostró que las capacidades de seguridad integradas (cifrado IPsec y segmentación lógica) superan las limitaciones de las WAN tradicionales. Asimismo, el uso de orquestadores centralizados proporciona una visibilidad proactiva de la red, permitiendo una toma de decisiones informada basada en el análisis histórico de la telemetría.

### *V-D. Trabajo Futuro*

El modelo de laboratorio propuesto constituye una base escalable. Se recomienda para futuras investigaciones extender este escenario hacia la integración con arquitecturas SASE (Secure Access Service Edge) y el uso de enlaces 5G como respaldo inalámbrico, consolidando redes aún más ágiles y predictivas mediante el uso de inteligencia artificial (AIOps).

## Referencias


[1] Zscaler, “SD-WAN vs. MPLS: What’s the Difference?.´´ [En línea]. Disponible: https://www.zscaler.com/zpedia/sd-wan-vs-mpls

[2] M. Terry, “What is SD-WAN and understanding its value,” Agility Communications, 2020. [En línea]. Disponible: https://agilitycommunications.net/what-is-sd-wan-and-understanding-its-value/.

[3] Cisco Systems, “What is SD-WAN? Software-Defined WAN (SD-WAN),” [En línea]. Disponible: https://www.cisco.com/site/us/en/learn/topics/networking/what-is-sd-wan.html.

[4] Cisco Systems, “What is the difference between SD-WAN and MPLS?,” [En línea]. Disponible: https://www.cisco.com/site/us/en/learn/topics/networking/what-is-the-difference-between-sd-wan-and-mpls.html#tabs-9da71fbd27-item-1288c79d71-tab

[5] A. Froehlich, “What is Class of Service (CoS)?,” TechTarget, 2021. [En línea]. Disponible: https://www.techtarget.com/searchnetworking/definition/Class-of-Service-CoS

[6] CloudFlare Inc., “SD-WAN vs. MPLS: ventajas e inconvenientes de la SD-WAN”[En línea]. Disponible: https://www.cloudflare.com/es-es/learning/network-layer/sd-wan-vs-mpls/

[7] Fortinet Inc., “Secure SD-WAN de Fortinet – Soluciones Secure SD-WAN de Fortinet,” [En línea]. Disponible: https://www.fortinet.com/lat/products/sd-wan.

[8] S. Sidheeq, “Build a FortiGate lab using GNS3 – Step by Step Guide.´´ GetLabsDone, Ene. 6, 2022. [En línea]. Disponible: https://getlabsdone.com/build-a-fortigate-lab-using-gns3-step-by-step-guide/

[9] Fortinet Community, “Technical Tip: How to add a FortiGate VM into the GNS3,” Fortinet Community Knowledge Base, Jan. 5, 2023. [En línea]. Disponible: https://community.fortinet.com/fortigate-3/technical-tip-how-to-add-a-fortigate-vm-into-the-gns3-127045

[10] Palo Alto Networks, “About SD-WAN,” Palo Alto Networks Documentation. [En línea]. Disponible: https://docs.paloaltonetworks.com/sd-wan/getting-started/about-sd-wan.

# Arquitectura de Laboratorio Remoto para Sistemas Embebidos MSP430FR4133

Juan Manuel Giron*, Jeremias Gaia†, Eugenio Orosco* Gabriel Cañadas*
* Lab. de Electronica Digital y Telecomunicaciones, Facultad de Ingeniería, Universidad Nacional de San Juan, Argentina
Email: jgaia@inaut.unsj.edu.ar
† INAUT (UNSJ-CONICET), San Juan, Argentina

***Abstract*—Este trabajo presenta el diseño e implementación de un laboratorio remoto para el microcontrolador MSP430FR4133 orientado a la enseñanza de sistemas embebidos. La arquitectura propuesta integra un backend basado en Python/FastAPI, una plataforma Tiva C para la emulación física de periféricos, retroalimentación visual mediante streaming de video, acceso desde Moodle mediante el protocolo LTI y conexión segura bajo filosofía Zero Trust. La contribución diferencial respecto al estado del arte reside en exponer al estudiante a un flujo de trabajo estándar de la industria: el ciclo completo de edición, compilación, carga y depuración en circuito en tiempo real se realiza desde VSC con acceso remoto SSH, prescindiendo de IDEs propietarios o simplificados. El paradigma thin-client elimina la brecha de hardware entre estudiantes, requiriendo únicamente un navegador web y VSC en el equipo del alumno.**

***Index Terms*—Laboratorio remoto, sistemas embebidos, MSP430FR4133, FastAPI, Tiva C, LTI, VSC, SSH.**

## I. Introducción

La enseñanza de sistemas embebidos requiere interacción directa con hardware real para el dominio de la programación a bajo nivel y la observación de comportamientos en tiempo real. Este acceso a laboratorios especializados suele restringirse por factores geográficos, institucionales o de equipamiento [1]. Los laboratorios remotos constituyen una respuesta concreta a esta problemática [2], permitiendo la teleoperación de hardware físico real desde ubicaciones distantes, a diferencia de los laboratorios virtuales basados en simulación, que no siempre replican con exactitud las restricciones físicas del hardware [3]. Existen antecedentes relevantes en esta línea. Trabajos como [4] y [5] proponen arquitecturas de laboratorio remoto para cursos de sistemas embebidos con retroalimentación visual en tiempo real, demostrando la viabilidad técnica del enfoque. La integración con plataformas de gestión de aprendizaje mediante el protocolo LTI (Learning Tools Interoperability) ha sido abordada en [6], donde se implementa un laboratorio para programación de microcontroladores accesible desde Moodle, y en [7], que describe sistemas de federación de laboratorios remotos basados en LTI [8]. A diferencia de simuladores (como Proteus) que omiten restricciones físicas, o laboratorios remotos comerciales (como LabsLand) y académicos que exponen IDEs web sin posibilidad de sesiones de depuración, esta arquitectura aporta una solución para la familia MSP430 integrando un entorno profesional estándar VSC (Visual Studio Code) vía SSH (Secure Shell), permitiendo depuración en circuito en tiempo real.

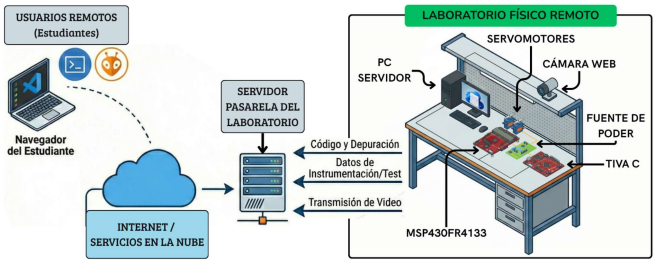



Fig. 1: Arquitectura del sistema de laboratorio remoto de sistemas embebidos. Se observan los componentes principales ordenados por flujo de datos: pc alumno → internet/servicio en la nube → servidor del laboratorio → laboratorio físico

Este trabajo aborda el diseño e implementación de un laboratorio remoto para el microcontrolador MSP430FR4133, presentándose como innovación, a la implementación del conjunto de las tecnologías no presentes en el estado del arte. La arquitectura propuesta integra un servidor backend basado en Python/FastAPI, un firmware de control en tiempo real sobre plataforma Tiva C para la emulación física de periféricos mediante servomotores y señales PWM, retroalimentación visual mediante streaming de video, y acceso seguro bajo filosofía Zero Trust. La experiencia del estudiante se unifica a través de VSC con acceso remoto SSH, eliminando la necesidad de instalar toolchains locales y democratizando el acceso independientemente del hardware del cliente.

## II. Arquitectura del Sistema

La arquitectura propuesta sigue una topología cliente-servidor, cuya interacción se ilustra en la Fig. 1. Se adoptó un paradigma thin-client [9], donde el procesamiento computacional se realiza en el servidor. La imagen refleja los componentes del sistema ordenados según el flujo de datos al momento de iniciar una conexión: cliente (alumno) → internet → servidor del laboratorio → laboratorio físico. Los *estudiantes*, a través de *internet*, *acceden* al hardware de experimentación que se encuentra en el *laboratorio físico*.

**Cliente.** La PC del alumno, centraliza la interacción a través de su interfaz visual.




Fig. 2: Ventana de acceso con: contraseña, acceso directo al workspace remoto en VSC y botón para borrar proyecto.




Fig. 3: Interfaz web del laboratorio: panel de control de periféricos y stream de video de la placa.

**Internet.** Provee los servicios de conectividad para el acceso remoto al laboratorio.

**Servidor.** Nodo de control local que gestiona el hardware, tránsito de clientes y el ciclo de vida de las sesiones.

**Laboratorio Físico.** En él se encuentra el hardware de experimentación MSP430FR4133, junto con los periféricos de E/S necesarios para la interacción y el monitoreo de sus estados operativos.

A continuación, la Sección III detalla el proceso de integración entre todas las capas mencionadas anteriormente.

## III. Implementación

### A. *Interfaz y Experiencia de Usuario - UIX*

La interfaz de usuario cuenta con tres ventanas interactivas: Acceso, Panel de control con video y VSC Remote.

**Ventana de Acceso.** Al inicio de cada sesión, luego de entrar desde Moodle, el estudiante es redirigido a una pantalla de logueo (Fig. 2). Allí, mediante un botón, el alumno copia una contraseña segura autogenerada, requerida posteriormente para programar el hardware objetivo. Como segundo paso, un botón lanza localmente VSC apuntando al workspace remoto mediante el esquema URI (Uniform Resource Identifier) *vscode://vscode-remote/ssh-remote+(IPHost)(RutaAlProyecto)*, dejando de lado configuraciones manuales de red por parte del cliente. Además, el alumno puede borrar su proyecto del servidor mediante un botón de confirmación doble (justificado en IV-B).

**Panel de Control con Video.** Una vez que el usuario ha sido identificado, puede acceder al panel de control, mostrado en la Fig. 3. Contiene a la derecha el video que capta la cámara remota para observar leds, pantalla LCD, y la presión de los botones. A la izquierda las acciones que se pueden realizar sobre el kit. Éstas incluyen: accionar S1 y S2 en modo momentáneo o de enclavamiento y ajustar una tensión analógica entre 0 y 3.3v mediante un potenciómetro virtual. Estos estímulos se traducen a acciones físicas y reales sobre la placa objetivo.

Cuando se utilizan los botones S1 y S2 en modo momentáneo, el sistema capta el accionar del mouse, permitiendo que los botones físicos del MSP430FR4133 reciban el mismo estímulo que recibe el mouse del alumno. En el modo enclavamiento, los botones de la placa se mantienen presionados hasta el segundo click que se realice.

El potenciómetro virtual genera acciones sobre el módulo ADC del microcontrolador, el cual recibe la señal analógica, que queda lista para su procesamiento. Para implementarlo, se utilizó una técnica de control de flujo que limita la frecuencia de envío de datos. Al deslizar el control, el sistema espera 75 ms de inactividad antes de transmitir el valor definitivo al servidor, un tiempo por debajo del umbral de 100 ms asociado a la percepción de respuesta instantánea. Esto evita la saturación del ancho de banda del puerto serie (UART) con ráfagas de valores intermedios. La sección III-C explica cómo el servidor procesa estos datos recibidos desde el frontend.

**Visual Studio Code.** En la ventana de VSC el alumno interactúa con un entorno estándar de la industria del software embebido. A través de la extensiones SSH-Remote y PlatformIO[1] se logra integrar el toolchain MSP430-GCC y los servicios de backend del laboratorio remoto. La interfaz expone un flujo de trabajo completo y consistente con la cultura de desarrollo de la industria electrónica contemporánea: edición, compilación, carga y depuración del firmware. Estas acciones se realizan sobre la placa MSP430FR4133 a través de túneles SSH, y la retroalimentación se presenta en tiempo real mediante la consola integrada, tabla de expresiones y de variables. Además el workspace se mantiene simplificado para el estudiante al ocultar carpetas y archivos con los que no es necesario interactuar.

### B. *Infraestructura de Conexión Remota*

El sistema gestiona una red privada virtual (VPN), como capa de transporte. Sobre esta capa, se despliega una arquitectura de comunicación híbrida que utiliza: estándar LTI, HTTP/HTTPS, WebSockets y túneles SSH.

[1] https://platformio.org/

**Red Privada Virtual.** La infraestructura de red se implementó mediante una Mesh VPN, estableciendo un perímetro de seguridad bajo la filosofía Zero Trust. La solución utiliza el protocolo WireGuard para crear túneles cifrados punto a punto (P2P) entre el equipo del estudiante y el servidor de laboratorio. El despliegue del agente en el host permite realizar un NAT traversal automático, eliminando la necesidad de exponer puertos sensibles a internet. Este esquema garantiza que el acceso al hardware de experimentación esté restringido exclusivamente a nodos autorizados y autenticados previamente.

**Estándar LTI.** El acceso al laboratorio se integró con la plataforma Moodle mediante el estándar LTI. En esta arquitectura, Moodle actúa como herramienta consumidora (TC) y el backend del laboratorio como herramienta proveedora (TP). El flujo de acceso comienza con un click en la herramienta externa (TE) configurada en el TC, se envía una solicitud HTTP POST firmada mediante el protocolo OAuth a la TP. Mediante el par consumer key y shared secret configurado en la TE, se garantiza que el acceso provenga exclusivamente del aula virtual autorizada y elimina la necesidad de gestionar una base de datos de usuarios externa.

**HTTP/HTTPS.** Se emplea HTTPS para la carga de la interfaz web y el streaming de video mediante el endpoint */video_feed*, implementado con *multipart/x-mixed-replace* para la entrega secuencial de fotogramas MJPEG con baja latencia. Se utilizan cabeceras *no-cache* para evitar el estancamiento visual por caché del navegador.

**WebSockets.** Se implementó WebSockets (RFC 6455) para el control bidireccional en tiempo real, transmitiendo el estado de los periféricos (S1, S2 y potenciómetro) en formato JSON. Un heartbeat cada 2000 ms reporta el estado de conectividad de la Tiva C y la cámara al banner de alertas.

**Protocolo SSH.** El acceso al entorno de programación VSC se gestióna mediante el protocolo SSH (RFC 4253) con la extension Remote - SSH de VSC instalada en el cliente, de esta manera, se logra cumplir con el paradigma de thin-client. El equipo local opera como terminal gráfica, delegando procesos de alto consumo (análisis sintáctico, compilación cruzada y enlazado de binarios) al servidor central. Permite encriptar y autenticar todo el canal y crea un túnel de archivos (VFS) que es un sistema de archivos virtual, permitiendo al alumno ver la carpeta de su proyecto como si estuviera en su propio disco duro. Esto último permite editar el código en tiempo real sin tener que subir archivos por cada cambio realizado.

### *C. Backend*

La gestión integral del laboratorio remoto se centraliza en un módulo de gestión desarrollado bajo el framework FastAPI, el cual coordina la lógica de hardware, la seguridad del sistema operativo y la persistencia de datos mediante una arquitectura asíncrona. Este componente actúa como el núcleo de toma de decisiones, garantizando la integridad del sistema y la exclusión mutua en el acceso a los recursos físicos. El funcionamiento del backend se estructura siguiendo el flujo del ciclo de vida de la sesión, organizado en cuatro etapas: i) autenticación y asignación de recursos, ii) aprovisionamiento del entorno y seguridad, iii) operación y telemetría, iv) persistencia y liberación de recursos. Todo esto se ejecuta sobre Windows 11 Pro, con un usuario genérico por banco de laboratorio.

**Autenticación y Asignación de Recursos.** El ciclo se inicia con la recepción de la solicitud HTTP POST nombrada en III-B, la cual es validada por el backend para autenticar la identidad académica del estudiante. Tras esto, el sistema extrae los identificadores de usuario y de grupo, almacenándolos en la sesión para gestionar el acceso exclusivo a los bancos de laboratorio. La asignación del espacio de trabajo se realiza bajo un estricto control de exclusión mutua garantizando que no existan colisiones de hardware y verificando que el grupo del estudiante no posea ya un laboratorio activo en la plataforma. Una vez validado el ingreso, el servidor identifica las placas Tiva C y MSP430 correspondientes al slot mediante sus números de serie únicos registrados en un archivo config.py, asegurando una vinculación física consistente y persistente durante toda la sesión.

**Aprovisionamiento del Entorno y Seguridad.** Una vez asignado el recurso, el servidor genera una contraseña de sesión dinámica aplicada al usuario Windows del banco, inyecta una regla de firewall que restringe el puerto 22 exclusivamente a la IP del cliente autenticado, sincroniza el workspace desde el almacenamiento persistente del alumno o desde una plantilla base, y aplica listas de control de acceso (ACL) NTFS otorgando permisos exclusivos al usuario asignado. Esto garantiza un entorno privado e inmune a interferencias entre usuarios concurrentes.

**Operación y Telemetría.** Durante el uso del laboratorio físico, el backend gestiona flujos de datos concurrentes mediante programación asíncrona, actuando como puente entre la interfaz web y el hardware. Las acciones del estudiante se reciben vía WebSockets en formato JSON y se procesan en el módulo de gestión de estado (state.py), el cual funciona como la única fuente de verdad del sistema. Estas actualizaciones son procesadas por un módulo traductor que genera tramas binarias compactas —incluyendo bytes de sincronización y un código de verificación XOR acumulativo— para su despacho por el puerto UART (115200 bps) hacia la placa Tiva C. En paralelo, el servidor gestiona hilos independientes para la captura de video mediante OpenCV y recibe el heartbeat del frontend para reportar el estado de conectividad de los periféricos. Todo el intercambio de comandos y credenciales se documenta en registros de auditoría y archivos de texto diarios, garantizando la trazabilidad completa de la sesión.

**Persistencia y Liberación de Recursos.** La sesión concluye mediante un protocolo de cierre garantizado, activado por la desconexión del usuario o por el Session Reaper, una tarea asíncrona programada que monitorea el cumplimiento del límite de tiempo de 60 minutos. En esta fase, el backend realiza una sincronización final del código fuente desde el slot hacia el almacenamiento persistente del estudiante, preservando su progreso antes de eliminar el directorio de trabajo temporal. Finalmente, se ejecuta el comando taskkill para

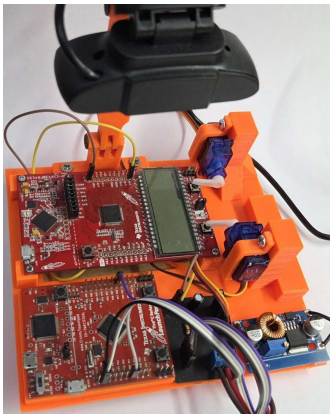


Fig. 4: Banco de laboratorio físico.

terminar forzosamente todos los procesos activos vinculados al usuario del slot (como el servidor de VSC o procesos de compilación), liberando los recursos de memoria y garantizando un entorno neutral para el próximo acceso.

### D. Laboratorio Físico

Se diseñó un modelo 3D que funciona como base del laboratorio físico remoto. Sobre este modelo se posiciona todo el hardware, permitiendo la integración y conexiones entre los diferentes componentes y el servidor. En la Fig. 4 se muestra el banco de laboratorio físico construido, donde los servomotores SG90 quedan posicionados sobre los botones S1 y S2 del MSP430FR4133 y la cámara USB apunta hacia la placa. El PCB dedicado integra el filtro RC pasa-bajos y los conectores de distribución de señales, disminuyendo el cableado libre entre componentes.

## IV. Resultados

La validación del sistema se realizó en un entorno de pruebas controlado, evaluando cuatro dimensiones: desempeño de la teleoperación, estabilidad del servicio, persistencia de datos y seguridad. Un video demostrativo del sistema completo se encuentra disponible en línea *https://bit.ly/4oPuGUz*.

### A. Latencia y Retroalimentación Visual

Las pruebas de teleoperación arrojaron una latencia de control inferior a 100 ms entre el comando emitido desde la interfaz web y la respuesta física de los actuadores, lo cual permite una interacción percibida como instantánea para los fines de las prácticas académicas. El sistema de streaming basado en MJPEG demostró estabilidad operativa entregando video a 640×480 píxeles a una tasa de entre 15 y 20 FPS, suficiente para observar en tiempo real el movimiento de los servomotores, el encendido de LEDs y el contenido de la pantalla LCD del MSP430FR4133.

### B. Estabilidad, Persistencia y Seguridad

El mecanismo de recuperación automática ante cortes de energía, basado en el inicio de sesión automático y el Programador de Tareas de Windows, restableció el backend sin intervención humana. Ante desconexiones forzadas de WebSockets, el código fuente fue sincronizado íntegramente al almacenamiento persistente antes de liberar el slot, preservando el progreso del estudiante; el comportamiento de restauración automática de archivos eliminados desde VSC justificó además el botón ”Borrar Todo” de la interfaz. Finalmente, las pruebas de acceso cruzado confirmaron que las ACL NTFS y las reglas dinámicas de firewall bloquearon exitosamente tanto el acceso entre slots como los intentos de conexión externos a la plataforma Moodle. Pruebas adicionales validaron el rechazo exitoso de conexiones concurrentes sin disponibilidad, y el bloqueo de accesos no autorizados mediante aislamiento de red, incluso ante credenciales comprometidas. Funcionalmente, se comprobó la exactitud del accionamiento de E/S y la generación/lectura de señales PWM (verificado con osciloscopio).

## V. Conclusiones

Se diseñó e implementó un laboratorio remoto para la enseñanza de sistemas embebidos sobre MSP430FR4133, validando su viabilidad técnica y pedagógica. La integración de VS Code con SSH y PlatformIO en el servidor constituye la contribución diferencial del trabajo, exponiendo al estudiante a un flujo de programación, compilación, ejecución y depuración estándar en la industria sobre hardware físico real. La arquitectura es de bajo costo, replicable y escalable, con operación desatendida garantizada por su mecanismo de recuperación automática. Como trabajo futuro se prevé escalar los bancos físicos, integrar un sistema de turnos en Moodle, y exponer una API/CLI para soportar pruebas automatizadas (Hardware-in-the-Loop) y despliegue continuo (CI/CD).

## References


[1] G. R. Alves, M. A. Marques, A. V. Fidalgo, J. García-Zubía, M. Castro, U. Hernández-Jayo, F. García-Loro, and C. Kreiter, “A roadmap for the visir remote lab,” *European Journal of Engineering Education*, vol. 48, no. 5, pp. 880–898, 2023.

[2] V. J. Bhute, P. Inguva, U. Shah, and C. Brechtelsbauer, “Transforming traditional teaching laboratories for effective remote delivery—a review,” *Education for Chemical Engineers*, vol. 35, pp. 96–104, 2021.

[3] A. Van den Beemt, S. Groothuijsen, L. Ozkan, and W. Hendrix, “Remote labs in higher engineering education: engaging students with active learning pedagogy,” *Journal of Computing in Higher Education*, vol. 35, no. 2, pp. 320–340, 2023.

[4] A. Maiti, A. Raza, and B. H. Kang, “Teaching embedded systems and internet-of-things supported by multipurpose multiobjective remote laboratories,” *IEEE Transactions on Learning Technologies*, vol. 14, no. 4, pp. 526–539, 2021.

[5] J. A. Nestor, “Experiences with remote teaching an embedded systems course,” in *Proceedings of the 2021 Great Lakes Symposium on VLSI*, 2021, pp. 437–442.

[6] A. Morales, “Laboratorios remotos en sistemas embebidos,” in *XX Congreso Internacional de Informática Educativa*, 2015, pp. 450–453.

[7] I. Ruano Ruano, E. Estévez, J. Gómez Ortega, and J. Gámez García, “Uso del estándar lti para integrar sistemas de gestión de aprendizaje y laboratorios online,” *XL Jornadas de Automática*, 2019.

[8] S. Shotarova and S. Stoyanova-Petrova, “Interactive learning in engineering education based on lms and lti,” in *Edulearn23 Proceedings*. IATED, 2023, pp. 1803–1810.

[9] J. T. Sinclair and M. Merkow, *Thin Clients Clearly Explained*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.

Categoría

# Foro Tecnológico

Software Embebido

Área Temática

# Diseño de un Generador de Formas de Onda Arbitrarias Multi-canal basado en DDS y Sistemas Distribuidos

Agustín Zuliani, Daniel Márquez
Departamento de Sistemas e Informática (D.S.I.)
Facultad de Ciencias Exactas, Ingeniería y Agrimensura (F.C.E.I.A.)
Universidad Nacional de Rosario, Rosario, Argentina
Email: azuliani02@fceia.unr.edu.ar, dmarquez@fceia.unr.edu.ar

***Resumen*—En este trabajo se presenta el diseño e implementación de un generador de formas de onda arbitrarias (AWG) de dos canales basado en Síntesis Digital Directa (DDS [1]). El sistema se estructura en una arquitectura distribuida con dos placas con sus propios microcontroladores: Una placa de control basada en el microcontrolador RP2040, encargada de la generación de señales, y una placa de panel frontal basada en el microcontrolador STM32H755ZI, a cargo de la Interfaz Hombre-Máquina (HMI). El núcleo de síntesis utiliza el circuito integrado AD9102, sincronizado por el oscilador programable SI5351 a una tasa de muestreo de hasta 180 MSPS, lo que permite generar formas de onda entre 1 Hz y 10 MHz. Cada canal permite configurar de forma independiente la amplitud (±10 Vp), el offset de continua, el ciclo de trabajo y la fase. La interacción con el usuario se realiza a través de un display TFT a color táctil de 3,5", un encoder rotativo de 20 posiciones, dieciséis pulsadores configurables, y conectividad USB Tipo C y Ethernet. Ambos firmwares se desarrollaron sobre el RTOS Zephyr, aprovechando su capa de abstracción de hardware para lograr un código portable, mantenible y escalable. La comunicación entre placas se realiza mediante un protocolo IPC propio sobre SPI, con tramas de tamaño fijo y verificación de integridad CRC-16. Una tercera placa, diseñada para generar los voltajes de alimentación, provee los rieles de ±15V, ±5V y 3,3V requeridos por las etapas analógicas y digitales.**

***Palabras clave*—Generador de Funciones, RP2040, RTOS, Zephyr, Diseño Electrónico, PCB, DDS, STM32H7, LVGL.**

## I. Introducción

Un generador de formas de onda arbitrarias (**AWG**, del inglés *Arbitrary Waveform Generator*) es un instrumento de laboratorio de electrónica capaz de sintetizar señales de forma configurable por el usuario, las cuales pueden inyectarse a un equipo bajo prueba para verificar su correcto funcionamiento o detectar fallas. A diferencia de los generadores convencionales, que producen formas de onda fijas (senoidal, cuadrada y triangular), un AWG permite definir la señal de salida de forma arbitraria, por ejemplo mediante una tabla de muestras almacenada en memoria.

La síntesis de señales se basa en la técnica de **Síntesis Digital Directa** (**DDS**, del inglés *Direct Digital Synthesis*), en la cual un acumulador de fase genera direcciones sobre una tabla de onda almacenada en memoria, cuya salida alimenta un Convertidor Digital-Analógico (**DAC**). La frecuencia máxima sintetizable queda acotada por el teorema de Nyquist a la mitad de la frecuencia de muestreo del sistema. La salida se entrega típicamente a través de un conector **BNC** con terminación de 50 Ω. Los equipos de mayor prestación incorporan además etapas de acondicionamiento analógico que permiten controlar la amplitud, el offset de continua y la fase de la señal generada.

Si bien existen soluciones comerciales en este rango de prestaciones, su arquitectura cerrada limita la posibilidad de adaptación, estudio y modificación en contextos académicos y de laboratorio. En este trabajo se presenta el diseño completo de un AWG de dos canales, de código abierto, basado en el circuito integrado **AD9102** [2] como núcleo DDS y el microcontrolador **RP2040** [3] como unidad de control. El sistema permite sintetizar señales entre 1 Hz y 10 MHz, con control independiente de amplitud (±10 Vp), offset, ciclo de trabajo y fase por canal. El sistema se estructura en dos placas principales: una placa de control basada en el RP2040, que incluye al AD9102 como núcleo DDS comunicándose por SPI, y una placa de panel frontal basada en el STM32H755ZI para la interfaz con el usuario. El firmware de ambas placas basa su desarrollo sobre el **RTOS Zephyr** [4], priorizando portabilidad y mantenibilidad del código. Se describe la arquitectura del sistema, el diseño del hardware compuesto por tres placas: **control**, **panel frontal** y **alimentación** y la organización del firmware.

## II. Arquitectura del Sistema

El sistema completo se compone de tres placas de circuito impreso con funciones distintas: una placa de **control**, una placa de **panel frontal** y una placa de **alimentación**. En la Figura 1 se muestra el diagrama en bloques general del sistema.

La idea de generar un sistema distribuido permite separar tareas y responsabilidades. En la placa de control se ejecutan las tareas de síntesis de señal, mientras que la placa de panel frontal gestiona la interfaz con el usuario sin imponer latencias sobre la generación.

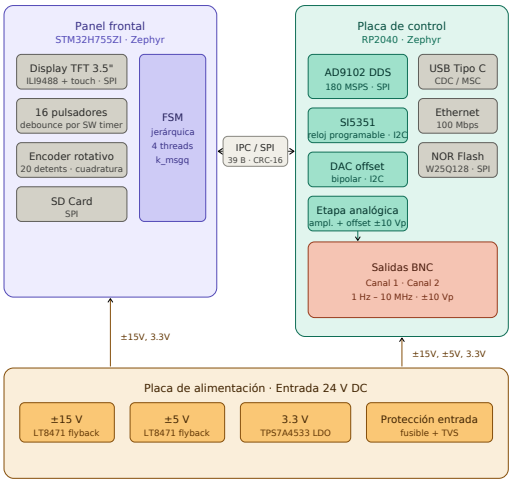



Figura 1: Diagrama en bloques del Hardware.

### A. *Comunicación entre placas*

La comunicación entre la placa de control y la de panel frontal se realiza mediante un protocolo **IPC** (*Inter-Processor Communication*) propio, implementado sobre **SPI** en configuración maestro-esclavo entre el microcontrolador RP2040 y el STM32H755.

La estructura de la trama se muestra en la Tabla I. Cada paquete tiene un tamaño fijo de 39 bytes. El campo `magic` (0xAF) permite la sincronización en el flujo de datos, mientras que el número de secuencia `seq` posibilita la correlación entre comandos y respuestas. La integridad de los datos se verifica mediante un CRC-16/CCITT-FALSE calculado sobre los primeros 37 bytes.

Tabla I: Estructura de la trama IPC de 39 bytes.

| Byte(s) | Campo | Tamaño | Descripción |
|---|---|---|---|
| 0 | Magic | 1 B | Sincronización (0xAF) |
| 1 | Version | 1 B | Versión del protocolo |
| 2 | Seq | 1 B | Número de secuencia |
| 3 | Cmd | 1 B | Comando o estado |
| 4–36 | Payload | 33 B | Datos del comando |
| 37–38 | CRC-16 | 2 B | CRC-16/CCITT-FALSE |

## III. Hardware

### A. *Placa de control*

Esta placa corresponde a la parte más importante del sistema. Se encarga de generar las señales definidas por el usuario mediante el **DDS AD9102**, el cual posee una memoria de tipo SRAM en su interior, permitiendo cargar una tabla de valores definida por el usuario, la cual se ve reflejada en la salida con su **DAC** (Digital Analog Converter) integrado de 14 bits. Este DDS es controlado a través del **RP2040**, transfiriendo la información mediante el protocolo SPI maestro-esclavo. Para lograr el ajuste de offset se utiliza un DAC externo, **MCP4725** con una resolución de 12 bits, manipulado mediante $I^2C$, que junto con un circuito de adaptación permite generar una señal de continua bipolar de $\pm 5$ V DC. El ajuste de amplitud se logra al variar la tensión de referencia del propio DDS, esta parte requiere de alta precisión, por lo que para dicha referencia se utiliza el **DAC8562** que posee una resolución de 16 bits. En la Figura 2 puede verse la placa de control en la vista superior, donde además aparecen ciertos periféricos como **Ethernet** y **USB-C** para la comunicación con el usuario e interfaz gráfica desde la PC.

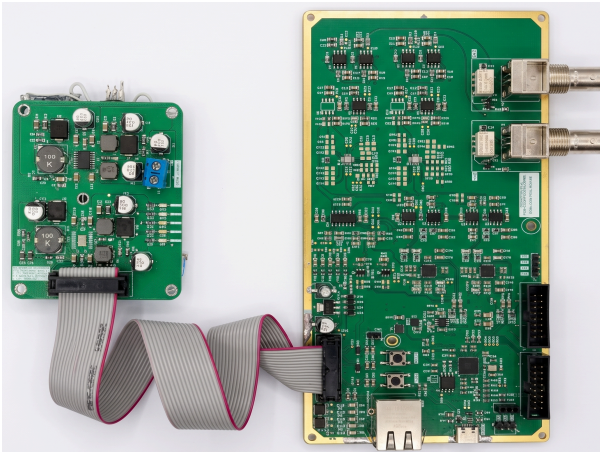


Figura 2: Vista superior de la placa de control (RP2040+AD9102) y fuente de alimentación (LT8471).

### B. *Placa de panel frontal*

La placa de panel frontal (Figura 3) se encarga de gestionar la interfaz con el usuario mediante el microcontrolador **STM32H755ZI** de doble núcleo (Cortex-M7 + Cortex-M4), operando el firmware principal sobre el M7 bajo Zephyr. Dentro de los periféricos que se pueden encontrar tenemos un **display TFT a color de 3.5”**, con controlador **ILI9488**, panel táctil resistivo y tarjeta **SD**. Todo esto se encuentra controlado mediante bus **SPI** de alta velocidad. Por último, la placa posee 16 pulsadores y un encoder rotativo de 20 vueltas. A cada señal de entrada se le incorporó un **filtro amortiguador RC** para evitar el rebote de los contactos mecánicos y además un comparador tipo Schmitt Trigger con el integrado **SN74HC14**.

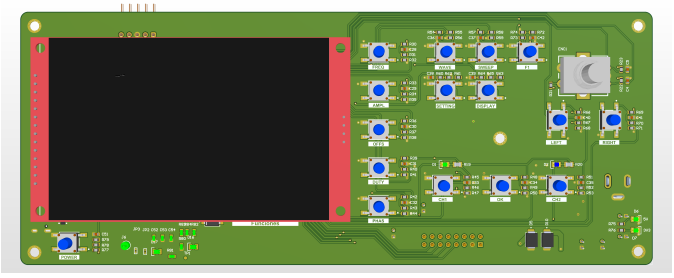


Figura 3: Vista 3D superior de la placa de panel frontal (STM32H755ZI). Posee un diseño de 4 capas.

### C. *Placa de alimentación*

La fuente de alimentación fue diseñada como una placa independiente para aislar los convertidores conmutados de las etapas analógicas. Se alimenta con una entrada de 24 V DC y genera los rieles de $\pm 15$ V y $\pm 5$ V mediante dos convertidores **LT8471** a 500 kHz. El riel de 3,3 V se obtiene mediante el LDO **TPS7A4533**, seleccionado por su alto rechazo al ruido. Cada riel cuenta con protecciones de polaridad, ESD y un LED indicador de presencia. En la Figura 2 se muestra el diseño de la placa.

### D. *Diseño del PCB*

Durante el diseño se tuvieron en cuenta muchos criterios de routeo, más aún sobre la **placa de control**, dado que las señales digitales que circulan por las pistas son de media-alta velocidad, y además la señal analógica de salida debe ser acondicionada para lograr hasta un máximo de 20 Vpp. Esto se llevó adelante con un diseño de **6 capas**, ubicando cada etapa de manera que no interfiera con el resto, por ejemplo la **etapa digital** se encuentra toda del lado izquierdo, mientras que la **etapa analógica** del lado derecho, Figura 2. Además para evitar un acople de ruido sobre los rieles de alimentación, se utilizaron ferrites que dividen ambos dominios digital y analógico, dando como resultado en la práctica un ripple menor a 100 mVpp (0.34 % de 15 V).

## IV. Firmware

El firmware de ambas placas basa su desarrollo sobre el **RTOS Zephyr** [4], aprovechando su capa de abstracción de hardware (HAL), sistema de colas nativas y configuración mediante *Device Tree*. Esto permite obtener un código portable, mantenible, escalable, e independiente del microcontrolador utilizado.

### A. *Placa de panel frontal*

La arquitectura del firmware se organiza en cuatro threads concurrentes comunicados exclusivamente mediante colas `k_msgq` de Zephyr, evitando variables globales compartidas. La lógica de aplicación se implementa mediante una **FSM jerárquica** basada en una tabla de transiciones, con dos regiones: una principal (reset, inicialización, reposo) y una de edición de señal (frecuencia, amplitud, offset, duty, fase y forma de onda). La comunicación con la placa de control se realiza desde un thread dedicado que ensambla y transmite tramas IPC por SPI.

## V. Resultados experimentales

Se realizaron mediciones preliminares sobre la placa de control con el fin de verificar el correcto funcionamiento del núcleo de síntesis. Se configuró el AD9102 para generar una señal senoidal a 100 kHz, con el reloj de muestreo provisto por el SI5351. La señal de salida fue capturada mediante un osciloscopio digital OWON VDS1022I y se analizó su espectro utilizando la función FFT del instrumento con ventana de Hanning. En la Figura 4 se muestra la forma de onda temporal junto con su espectro en frecuencia. A partir del análisis espectral se determinó que el armónico dominante corresponde al tercero (300 kHz), con un nivel de $-47$ dBV, mientras que el segundo armónico (200 kHz) se ubicó por debajo de $-52$ dBV. Se obtuvo un SFDR de aproximadamente 58 dB y una THD de $-56{,}6$ dB (0,15 %).

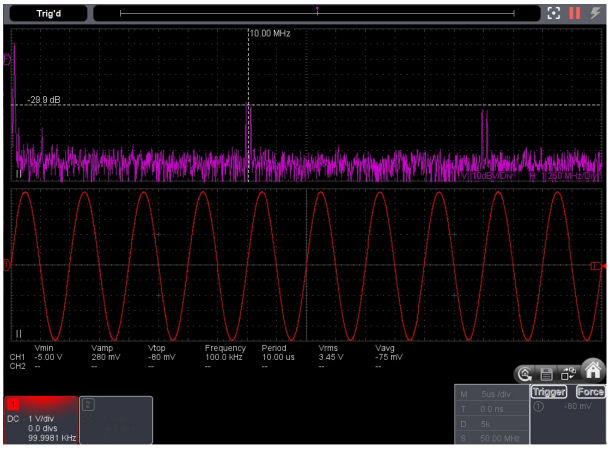



Figura 4: Señal senoidal de 100 kHz y su espectro obtenido mediante FFT del osciloscopio.

Para verificar la operación multicanal, se configuraron ambos canales del sistema para generar señales senoidales independientes a 100 kHz. En la Figura 5 se observan las dos salidas capturadas simultáneamente, confirmando el correcto funcionamiento de ambos canales de síntesis de forma independiente.

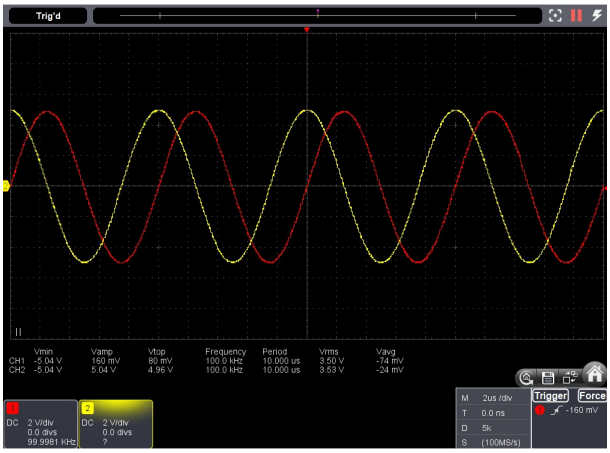



Figura 5: Salida simultánea de ambos canales generando señales senoidales de 100 kHz.

Adicionalmente, se verificó la capacidad de generación de formas de onda arbitrarias mediante la carga de una tabla de muestras en la memoria SRAM del AD9102. En la Figura 6 se muestra una forma de onda tipo sinc generada a 100 kHz, validando el correcto funcionamiento del modo de reproducción por SRAM.

El control del instrumento se realizó mediante una aplicación de escritorio desarrollada en Python con PySide6, comunicándose con la placa de control a través de Ethernet mediante

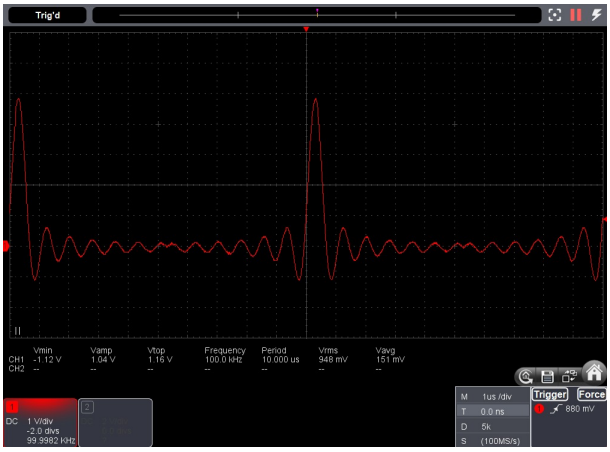



Figura 6: Forma de onda sinc generada mediante tabla de muestras cargada en la SRAM del AD9102 a 100 kHz.

el protocolo binario propietario de 39 bytes. En la Figura 7 se muestra la interfaz gráfica, donde se pueden observar los parámetros de configuración del canal, la previsualización de la forma de onda y el registro de tramas transmitidas y recibidas.

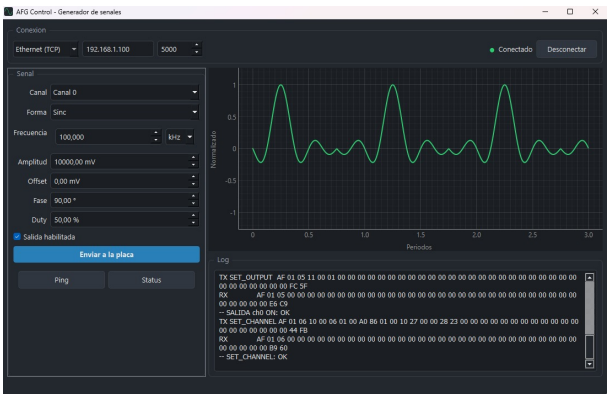



Figura 7: Interfaz gráfica de la aplicación de control mostrando la configuración de una señal sinc a 100 kHz y el log del protocolo IPC. Enlace mediante Ethernet.

## VI. Marco institucional

Este desarrollo se encuadra en el Proyecto Final de Carrera de Ingeniería Electrónica de la FCEIA (UNR). El proyecto propone una alternativa de código abierto frente a soluciones comerciales de arquitectura cerrada, permitiendo a la institución disponer de un instrumento con control total sobre su diseño. Su arquitectura distribuida, el uso de Zephyr RTOS y el protocolo IPC propio constituyen casos de estudio aplicables a múltiples asignaturas del área de sistemas embebidos. A futuro, se proyecta equipar los laboratorios con unidades del instrumento y habilitar el control remoto mediante Ethernet para laboratorios a distancia.

## VII. Futuras mejoras

Como continuación del trabajo presentado, se identifican las siguientes implementaciones:

- Incorporación de un filtro de reconstrucción a la salida del DAC para eliminar las imágenes de Nyquist y el escalonado inherente a la conversión digital-analógica. Esto permitiría extender el rango de frecuencias útiles hasta los 60 MHz, aprovechando la tasa de muestreo máxima de 180 MSPS del AD9102.
- Puesta en marcha de la placa de panel frontal (STM32H755ZI) e integración completa del firmware de interfaz con el usuario mediante LVGL.

## VIII. Conclusión

En este trabajo se presentó el diseño e implementación de un generador de formas de onda arbitrarias de dos canales basado en síntesis digital directa. El sistema integra tres placas de circuito impreso diseñadas a medida: una placa de control basada en el RP2040 con el AD9102 como núcleo DDS, una placa de panel frontal basada en el STM32H755ZI, y una placa de alimentación multi-rail. El firmware se desarrolló sobre Zephyr RTOS [4], logrando una arquitectura modular, portable y mantenible. Se verificó el correcto funcionamiento del sistema mediante mediciones experimentales, obteniendo una THD de $-56{,}6$ dB y un SFDR de 58 dB a 100 kHz. Se demostró la operación simultánea de ambos canales y la generación de formas de onda arbitrarias mediante carga de tablas en la SRAM del AD9102.

## Agradecimientos

Ing. José Luis Simón: Ingeniero Electrónico de la Universidad Nacional de Rosario. Actual Director del Departamento de Sistemas e Informática de la Facultad de Ciencias Exactas, Ingeniería y Agrimensura de la Universidad Nacional de Rosario.

Ing. Pablo Vozza: Ingeniero Electrónico de la Universidad Nacional de Rosario. Actual Secretario Académico de la Escuela de Ingeniería Electrónica de la Facultad de Ciencias Exactas, Ingeniería y Agrimensura de la Universidad Nacional de Rosario.

## Referencias

[1] E. Murphy and C. Slattery, "All about direct digital synthesis," *Analog Dialogue*, vol. 38, no. 3, 2004. [Online]. Available: https://www.analog.com/en/resources/analog-dialogue/articles/all-about-direct-digital-synthesis.html

[2] Analog Devices, Inc., *AD9102: Low Power, 14-Bit, 180 MSPS, Digital-to-Analog Converter and Waveform Generator*, rev. b ed., Analog Devices, Inc., 2024. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/AD9102.pdf

[3] Raspberry Pi Ltd, *RP2040 Datasheet: A microcontroller by Raspberry Pi*, Raspberry Pi Ltd, 2021. [Online]. Available: https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf

[4] Zephyr Project, "Zephyr project," 2024, real-Time Operating System (RTOS) for connected, resource-constrained devices. [Online]. Available: https://zephyrproject.org

# Sistema Embebido Multisensorial para Monitoreo en Tiempo Real de Variables Acústicas, Vibracionales y Térmicas en Motores

Andrea Catalina Roa Salcedo
Escuela de Ingeniería Electrónica
Universidad Pedagógica y Tecnológica de Colombia (UPTC)
Tunja, Colombia
andrea.roa03@uptc.edu.co

Caren Ximena Sierra Viasus
Escuela de Ingeniería Electrónica
Universidad Pedagógica y Tecnológica de Colombia (UPTC)
Tunja, Colombia
caren.sierra@uptc.edu.co

Ilber Ruge Ruge
Escuela de Ingeniería Electrónica
Universidad Pedagógica y Tecnológica de Colombia (UPTC)
Tunja, Colombia
ilber.ruge@uptc.edu.co

***Abstract*— This work presents the design and implementation of a multisensor embedded system for motor anomaly detection through acoustic, vibrational, and thermal variable analysis. The proposed architecture integrates a MAX4466 sensor, an MPU6050 module, a DHT11 sensor, microSD storage, temporal synchronization using an RTC DS3231 module, and wireless communication with the ThingSpeak platform.The system uses an ATmega328P microcontroller as the main processing unit, combining analog and digital acquisition through ADC, I²C, and SPI protocols. Experimental validation was performed on a 12 V DC motor under normal operating conditions and simulated mechanical fault scenarios.The obtained results demonstrated stable behavior during acquisition, storage, and wireless transmission stages, obtaining differences below 10% with respect to reference instruments in acoustic measurements.The proposed architecture constitutes a low-cost and scalable embedded solution for preventive monitoring and remote supervision of motors.***

***Keywords*— Embedded systems, anomaly detection, IoT, motor monitoring, multisensor acquisition, ATmega328P.***

## I. INTRODUCCIÓN

La detección temprana de anomalías en motores es fundamental para reducir fallas inesperadas y mejorar la confiabilidad operativa en aplicaciones industriales. En los últimos años, los sistemas embebidos y las tecnologías IoT han permitido desarrollar soluciones de bajo costo para el monitoreo en tiempo real de variables asociadas al funcionamiento de motores [1], [4].

Diversos trabajos han abordado la supervisión de motores mediante variables como vibración, sonido o temperatura [3], [6]. Sin embargo, muchas de estas propuestas se enfocan en una única variable o requieren plataformas especializadas. En este contexto, la integración de múltiples variables físicas dentro de una misma arquitectura embebida permite obtener una visión más completa del comportamiento operativo del sistema.

Este trabajo presenta el diseño e implementación de un sistema embebido multisensorial basado en el microcontrolador ATmega328P para el monitoreo de variables acústicas, vibracionales y térmicas. La arquitectura integra sensores de sonido, vibración e inclinación, almacenamiento local mediante microSD, sincronización temporal con RTC DS3231 y supervisión remota a través de ThingSpeak.

La validación experimental se realizó utilizando un ventilador DC de 12 V bajo condiciones normales y escenarios de anomalía inducidos de manera controlada. Los resultados permitieron verificar la estabilidad de la arquitectura propuesta y evidenciaron que las variables acústicas y vibracionales son las más representativas para identificar cambios en las condiciones de operación del sistema.

## II. METODOLOGÍA

La metodología propuesta se orientó al diseño e implementación de un sistema embebido multisensorial de bajo costo para la adquisición, procesamiento, almacenamiento y transmisión de variables asociadas al comportamiento operativo de motores. La arquitectura implementada integra módulos de adquisición acústica, monitoreo vibracional, sensado térmico, almacenamiento local y supervisión remota dentro de una plataforma centralizada basada en el microcontrolador ATmega328P. La Figura 1 presenta la arquitectura general del sistema y la distribución de los periféricos utilizados.

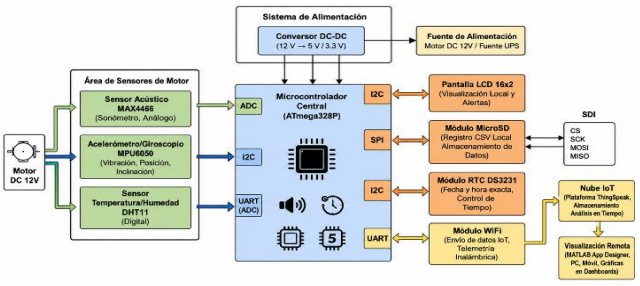



Fig. 1. Arquitectura general del sistema embebido multisensorial propuesto.

El procesamiento principal se desarrolló utilizando el microcontrolador ATmega328P de 8 bits debido a su capacidad para integrar adquisición analógica, protocolos seriales y temporización interna dentro de una arquitectura de bajo consumo y fácil integración [1], [2]. El sistema incorpora el sensor MAX4466 para adquisición acústica, el acelerómetro MPU6050 para monitoreo de vibración e inclinación, y el sensor DHT11 para supervisión térmica. Adicionalmente, se integraron un módulo RTC DS3231 para sincronización temporal y una memoria microSD para almacenamiento histórico de registros sensoriales.

En una primera etapa de desarrollo se implementó la adquisición y procesamiento de señales acústicas mediante el convertidor analógico-digital interno del ATmega328P. El firmware fue desarrollado inicialmente en Microchip Studio utilizando lenguaje C/C++, permitiendo realizar adquisición de muestras, cálculo RMS y conversión de amplitud acústica a decibeles. Posteriormente, debido a la integración de múltiples variables físicas y periféricos digitales, el sistema fue migrado hacia Arduino IDE con el fin de optimizar la modularidad del firmware y facilitar la implementación de librerías de comunicación I²C y SPI.

La ejecución del sistema fue organizada mediante temporizadores e interrupciones internas por hardware, eliminando retardos por software y garantizando actualización periódica de variables en tiempo real. Esta estrategia permitió desacoplar los procesos de adquisición, visualización y almacenamiento, manteniendo estabilidad operativa durante la supervisión simultánea de múltiples sensores. La Figura 2 muestra el flujo general de procesamiento implementado en el microcontrolador.

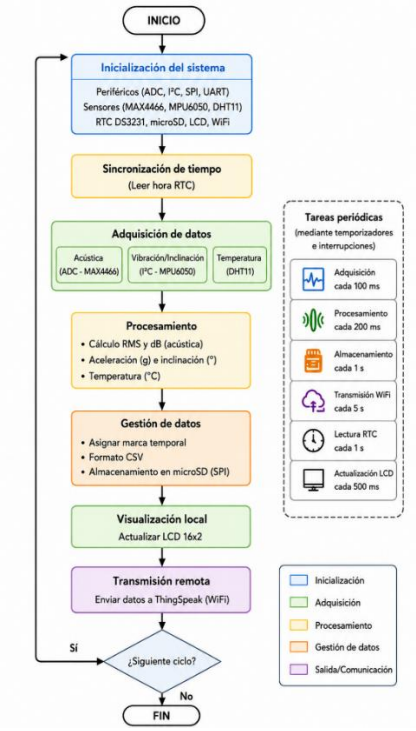



Fig. 2. Flujo de procesamiento y gestión de tareas implementado en el sistema embebido.

La integración multisensorial permitió combinar señales acústicas, vibracionales, inclinaciones y térmicas dentro de una misma arquitectura de monitoreo. La adquisición vibracional e inclinacional se realizó mediante el MPU6050 utilizando comunicación I²C, mientras que el almacenamiento histórico de datos fue implementado mediante SPI sobre memoria microSD, organizando los registros en formato CSV con referencia temporal proporcionada por el módulo RTC DS3231 [5], [7].

Para la supervisión operativa, el sistema incorporó una pantalla LCD de 16 × 2 caracteres para visualización local de variables y estados del sistema. Paralelamente, se implementó transmisión inalámbrica mediante comunicación WiFi hacia la plataforma ThingSpeak, permitiendo supervisión remota, almacenamiento en la nube y visualización histórica de las variables adquiridas [8], [9]. Adicionalmente, se desarrolló una interfaz gráfica en MATLAB para la representación simultánea de las variables monitoreadas y validación funcional del sistema. La Figura 3 presenta el prototipo implementado y la interfaz de supervisión desarrollada.




Fig. 3. Prototipo físico implementado y entorno de supervisión desarrollado en MATLAB y ThingSpeak.

La validación experimental se realizó utilizando un ventilador DC de 12 V montado sobre una base de madera. Para evaluar la respuesta del sistema ante diferentes condiciones operativas, se generaron de manera controlada anomalías acústicas y mecánicas, además de realizar pruebas de validación térmica. Las variaciones acústicas fueron inducidas modificando el voltaje de alimentación del ventilador, produciendo cambios en su velocidad de giro y nivel sonoro. Las anomalías mecánicas se generaron mediante la obstrucción parcial de las aspas del ventilador, provocando desbalanceo y vibraciones detectadas por el MPU6050. Adicionalmente, la respuesta del canal térmico fue validada aplicando aire caliente sobre el sensor DHT11 mediante un secador de cabello. Durante las pruebas se registraron continuamente variables acústicas, vibracionales y térmicas, las cuales fueron procesadas, almacenadas y transmitidas mediante la arquitectura propuesta.

En las primeras etapas se trabajó con sensores analógicos y adquisición mediante ADC del ATmega328P, permitiendo validar el procesamiento básico de señales y el cálculo de niveles acústicos. Posteriormente, la incorporación de sensores digitales mejoró la estabilidad y precisión de las mediciones, reduciendo efectos asociados al ruido eléctrico y optimizando la comunicación entre dispositivos.

La integración del acelerómetro MPU6050 mediante protocolo I²C permitió simplificar el cableado y aumentar la escalabilidad del sistema embebido. Asimismo, el módulo RTC DS3231 facilitó el registro cronológico de eventos y almacenamiento histórico de datos en memoria microSD.

La migración hacia Arduino IDE facilitó la modularidad del firmware y la integración de librerías para comunicación serial, almacenamiento y monitoreo IoT. Además, la interfaz desarrollada en MATLAB permitió visualizar variables en tiempo real, mientras que la plataforma ThingSpeak habilitó supervisión remota y almacenamiento en la nube. Las mediciones acústicas obtenidas mediante el sensor MAX4466 fueron comparadas con los valores registrados por una aplicación móvil de sonómetro utilizada como referencia comparativa durante las pruebas experimentales. Bajo las mismas condiciones de operación y ubicación respecto a la fuente sonora, se observaron diferencias inferiores al diez por ciento entre ambas mediciones, validando la confiabilidad del sistema desarrollado para aplicaciones de monitoreo preventivo de bajo costo.

## III. ANÁLISIS DE RESULTADOS

La validación experimental del prototipo permitió evaluar el desempeño de la arquitectura propuesta en la adquisición simultánea de variables acústicas, vibracionales y térmicas asociadas al funcionamiento del motor. Durante las pruebas, el sistema registró continuamente las señales provenientes de los sensores, evidenciando estabilidad operativa en condiciones normales de funcionamiento. En particular, la temperatura se mantuvo cercana a 26 °C, mientras que la inclinación en el eje X presentó valores concentrados entre 26 y 27 unidades, correspondientes a un comportamiento estable del sistema. No obstante, se identificaron variaciones puntuales en las señales acústicas y vibracionales asociadas a condiciones anómalas del motor.

Con el propósito de analizar el comportamiento del sistema bajo diferentes condiciones operativas, se estableció una condición normal de referencia a partir de registros estables de las variables monitoreadas. Posteriormente, dichas mediciones fueron comparadas con muestras correspondientes a condiciones anómalas. Los resultados evidenciaron que las mayores desviaciones se presentaron en la señal acústica y en la inclinación medida mediante el sensor MPU6050, mientras que la temperatura permaneció prácticamente constante. La Tabla I resume las muestras representativas obtenidas durante la validación experimental.

TABLA I. COMPARACIÓN ENTRE CONDICIÓN NORMAL Y CONDICIONES ANÓMALAS DEL SISTEMA.

| Condición | Fecha y hora | Sonido (dB) | Incl X (°) | Temperatura (°C) | Observación |
|---|---|---|---|---|---|
| Normal | 05/11/2025 09:14:32 | 48,7 | 26,34 | 26,5 | Operación estable |
| Anómala 1 | 05/11/2025 09:27:45 | 63,2 | 25,91 | 26,4 | Incremento acústico |
| Anómala 2 | 05/11/2025 09:41:18 | 42,1 | 30,15 | 26,6 | Desviación mecánica |

La condición normal presentó un nivel acústico de 48,7 dB, una inclinación en X de 26,34 y una temperatura de 26,5 °C, valores coherentes con una operación estable del motor. En contraste, la primera condición anómala evidenció un incremento del nivel sonoro hasta 63,2 dB, indicando la presencia de una perturbación mecánica o un aumento del ruido operativo del sistema. Este comportamiento demuestra la sensibilidad de la variable acústica frente a cambios en las condiciones de funcionamiento.

Por otra parte, la segunda condición anómala presentó una variación significativa en la inclinación, alcanzando un valor de 30,15 en el eje X, mientras que la temperatura permaneció estable. Este resultado indica que la anomalía detectada estuvo asociada principalmente a alteraciones mecánicas reflejadas en el comportamiento vibracional del sistema y no a fenómenos térmicos. En este sentido, la integración de variables acústicas y vibracionales permitió mejorar la capacidad de identificación de anomalías respecto al análisis individual de una sola señal.

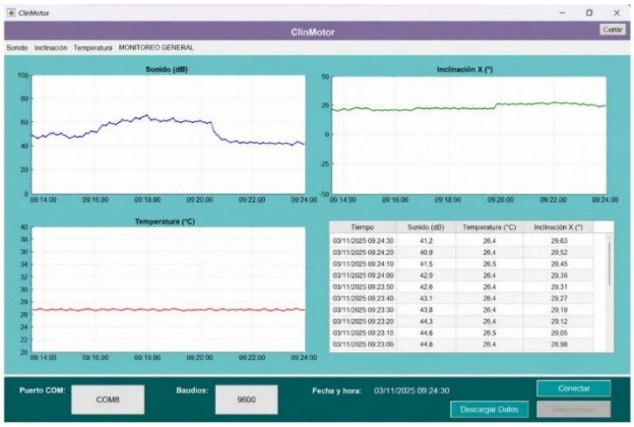



Fig. 4. Registro temporal de las variables acústicas, vibracionales y térmicas durante la validación experimental.

La Figura 4 presenta la evolución temporal de las variables monitoreadas durante las pruebas experimentales. Se observa que las condiciones normales mantienen un comportamiento estable en la mayor parte del intervalo de medición, mientras que los eventos anómalos generan perturbaciones claramente identificables en las señales de sonido e inclinación. Estos resultados evidencian la capacidad del sistema para detectar variaciones operativas en tiempo real mediante el análisis multisensorial de variables mecánicas y acústicas.

La Figura 5 compara las condiciones normales y anómalas observadas durante las pruebas, evidenciando variaciones en las señales acústicas y vibracionales.

En conjunto, los resultados obtenidos validan el funcionamiento de la arquitectura propuesta para el monitoreo preventivo de motores de bajo costo. La fusión de variables acústicas, vibracionales y térmicas permitió diferenciar condiciones normales y anómalas de operación, evidenciando que las señales de sonido e inclinación constituyen los indicadores más representativos para la detección de alteraciones mecánicas. Asimismo, la estabilidad en la adquisición y registro de datos confirma la viabilidad del

prototipo como plataforma experimental para futuras estrategias de diagnóstico y mantenimiento predictivo.




Fig. 5. Comparación entre condición normal y condición anómala en las señales acústicas y vibracionales del sistema.

## IV. Conclusiones

Los resultados experimentales validaron el funcionamiento del sistema embebido multisensorial para el monitoreo de anomalías en motores mediante variables acústicas, vibracionales y térmicas. La integración entre adquisición de datos, procesamiento embebido, almacenamiento CSV y supervisión remota mediante MATLAB y ThingSpeak evidenció estabilidad operativa durante las etapas de adquisición, procesamiento y transmisión bajo condiciones normales y escenarios de anomalía inducidos de manera controlada.

Las pruebas de validación evidenciaron variaciones significativas en las variables de sonido e inclinación durante condiciones anómalas del motor. El nivel acústico alcanzó valores de hasta 63,2 dB, mientras que la inclinación en el eje X presentó variaciones de hasta 30,15 respecto a la condición normal de referencia. Asimismo, la temperatura permaneció prácticamente estable alrededor de 26 °C durante las pruebas, indicando que las anomalías detectadas estuvieron principalmente asociadas al comportamiento mecánico y vibracional del sistema.

Adicionalmente, la comparación de las mediciones acústicas con una aplicación móvil de sonómetro utilizada como referencia evidenció diferencias inferiores al diez por ciento, confirmando la confiabilidad del prototipo para aplicaciones de monitoreo preventivo de bajo costo. Los resultados obtenidos demuestran la viabilidad de la plataforma para aplicaciones de supervisión embebida y adquisición de datos en tiempo real, proporcionando una base experimental para futuras estrategias de mantenimiento predictivo y clasificación inteligente de fallas en motores.

## Referencias


[1] R. Waswani, A. Pawar, M. Deore and R. Patel, “Fault detection, protection and speed control of induction motors using Arduino,” in Proc. International Conference on Computing, Communication, Control and Automation, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8276071

[2] J. Serrano Prieto, “Detección de anomalías de funcionamiento en motores eléctricos,” Trabajo de grado, Universidad de Oviedo, 2024. [Online]. Available: https://digibuo.uniovi.es/dspace/handle/10651/74043

[3] L. F. Ugarte Vega, J. L. Morejón Aucancela and J. A. Choez Rodríguez, “Identificación de anomalías de máquinas asíncronas mediante monitoreo no intrusivo con sensores de vibración,” Escuela Superior Politécnica del Litoral, 2023. [Online]. Available: https://www.dspace.espol.edu.ec/handle/123456789/6052 6

[4] J. A. Gomez Garcia, “Sistema de adquisición y monitoreo en tiempo real para la detección temprana de fallas en motores mediante análisis de señales de vibración y temperatura bajo un ambiente IoT,” Universidad Nacional Abierta y a Distancia, 2025. [Online]. Available: https://repository.unad.edu.co/handle/10596/69921

[5] H. Li, “A study of the effect of motor structural components on harmonic noise,” Scientific Reports, vol. 15, 2025. [Online]. Available: https://www.nature.com/articles/s41598-025-91861-9

[6] A. Suman, C. Kumar and P. Suman, “Early detection of mechanical malfunctions in vehicles using sound signal processing,” Applied Acoustics, vol. 190, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003682X21006721

[7] K. Mykoniatis, “Real-time condition monitoring and maintenance management system for industrial low-voltage motors using Internet of Things technologies,” Procedia Computer Science, vol. 170, pp. 873–880, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978920306090

[8] A. Nadar and R. Sangam, “Condition monitoring of rolling element bearing by acoustic analysis using LabVIEW,” International Journal of Recent Technology and Engineering, 2019. [Online]. Available: https://www.scopus.com/pages/publications/85065155947

[9] V.-K. Nguyen et al., “Real-time anomaly detection in electric motor operation noise,” 2024. [Online]. Available: https://www.researchgate.net/publication/386304486_Real-time_anomaly_detection_in_electric_motor_operation_noise

[10] W. Jung et al., “Multimodal dataset including vibration, acoustic, temperature and current measurements for rotating motor diagnosis,” Data in Brief, vol. 49, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340923001671

[11] S. Arciniegas, “IoT device with TinyML for bearing fault detection,” Discover Internet of Things, 2025. [Online]. Available: https://link.springer.com/article/10.1007/s43926-025-00142-4

[12] M. Antonini et al., “Adaptive and unsupervised TinyML system for edge anomaly detection,” Sensors, vol. 23, no. 4, 2023. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC9962960

Categoría

# Reporte

Inteligencia Artificial en SE

Área Temática

# Diseño e implementación de red neuronal neuromórfica

1st Grellet Martinoia, Alejandro
*Facultad de Ciencias Exactas y Tecnología*
*Universidad Nacional de Tucumán*
San Miguel de Tucumán, Argentina
grelletalejandro@gmail.com

2nd Volentini, Esteban Daniel
*Facultad de Ciencias Exactas y Tecnología*
*Universidad Nacional de Tucumán*
San Miguel de Tucumán, Argentina
evolentini@herrera.unt.edu.ar

3rd Musa, Ezequiel Alberto
*Facultad de Ciencias Exactas y Tecnología*
*Universidad Nacional de Tucumán*
San Miguel de Tucumán, Argentina
eamusa@herrera.unt.edu.ar

4th Saade, sergio Daniel
*Facultad de Ciencias Exactas y Tecnología*
*Universidad Nacional de Tucumán*
San Miguel de Tucumán, Argentina
ssaade@herrera.unt.edu.ar

***Resumen*—Este trabajo presenta el diseño, implementación y simulación de una solución en hardware para el procesamiento de una red neuronal de picos, desarrollada mediante el lenguaje de descripción de hardware VHDL. La propuesta se enfoca en la modelización de neuronas del tipo Leaky Integrate-and-Fire y en la implementación y validación de mecanismos de procesamiento temporal compatibles con arquitecturas neuromórficas. Se describe la arquitectura del sistema, la dinámica de propagación de spikes y la gestión de los parámetros sinápticos.**

***Palabras Clave*—VHDL, arquitecturas neuromórficas, redes neuronales basadas en picos, inteligencia artificial.**

## I. Introducción

La computación neuromórfica es un área emergente de la inteligencia artificial orientada al diseño de modelos cercanos a sistemas biológicos complejos, como las presentes en el cerebro humano. En este contexto surgen las redes neuronales de picos (*Spiking Neural Networks*, SNN), cuya unidad básica de memoria y procesamiento es la neurona y sus sinapsis. Las SNN operan por eventos: permanecen en reposo hasta recibir un estímulo externo, activando únicamente los módulos de hardware necesarios para su procesamiento. Adicionalmente, sus mecanismos de aprendizaje se inspiran en la plasticidad sináptica descripta por la neurociencia [1].

Estas premisas buscan obtener modelos de cómputo energéticamente eficientes y con alto nivel de adaptabilidad a nivel aprendizaje, con un gran potencial en la industria.

Este trabajo busca presentar una nueva arquitectura de hardware de una red neuronal basada en picos simple, implementada en VHDL, distando, por ejemplo, de la presentada en [4], con el fin estudiar su comportamiento en entornos de desarrollo, analizar su esquema de comunicación y validar los modelos teóricos estudiados.

## II. Modelo de neurona

Una neurona LIF es un modelo computacional simplificado de neurona que integra corrientes de entrada con fuga exponencial en su potencial de membrana, disparando un pico de salida cuando el potencial alcanza un umbral definido [2]. El modelo matemático de una neurona LIF se basa en la siguiente ecuación diferencial (1), que describe la dinámica del potencial de membrana:

$$\tau_m \frac{dV(t)}{dt} = -\left(V(t) - V_{\text{rest}}\right) + R_m I(t), \tag{1}$$

donde:

- $V(t)$: potencial de membrana en el tiempo $t$,
- $V_{\text{rest}}$: potencial de reposo de la membrana,
- $\tau_m = R_m C_m$: constante de tiempo de la membrana ($R_m$ es la resistencia de membrana y $C_m$ es la capacitancia),
- $I(t)$: corriente de entrada en el tiempo $t$,
- $R_m I(t)$: contribución de la corriente sináptica o externa al potencial de membrana.

y de acuerdo con la condición de disparo establecida en la ecuación (2):

$$\text{Si } V(t) \geq V_{\text{th}}, \text{ entonces } V(t) \rightarrow V_{\text{reset}} \tag{2}$$

Así la actividad de una neurona LIF respecto a los picos de entrada puede verse con mayor detalle en la Fig. 1:

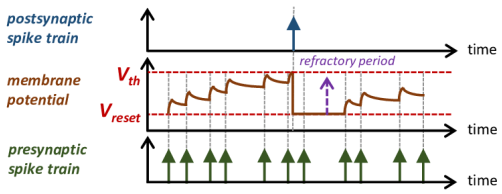



Fig. 1. Ilustración de la dinámica de una neurona LIF.

### A. Mecanismos de regulación sináptica

Debido al esquema de procesamiento descrito anteriormente, las redes neuronales de picos (se convierten en sistemas dinámicos de alta dimensionalidad y no linealidad. En

estas arquitecturas, las neuronas evolucionan colectivamente hacia el comportamiento deseado mediante retroalimentación, adaptación e interacciones con el entorno, a pesar de la incertidumbre en el comportamiento preciso de cada unidad neuronal individual [3].

La dinámica de atractores constituye una forma de dinámica colectiva que permite obtener cómputos útiles y no triviales. Una estrategia para el diseño de SNN con garantías de convergencia consiste en verificar la condición de Lyapunov. Bajo este marco, el Algoritmo de Competitividad Local (LCA, *Local Competition Algorithm*) es un ejemplo de mecanismo que proyecta una señal de entrada sobre un conjunto de neuronas que se inhiben mutuamente mediante conexiones recurrentes, induciendo competencia hasta converger a una representación dispersa de la entrada [3].

Cabe aclarar que, en la implementación presentada en este trabajo, el módulo LCA se incorpora como regulador de actividad a nivel hardware sin validar formalmente la condición de Lyapunov ni el comportamiento de convergencia teórico.

## III. Implementación en hardware

Este proceso se llevo a cabo teniendo en mente la escala probable que tendría la implementación de una red avocada a tareas de clasificación simple, intentando mantener un diseño modular y escalable en caso de que, si la misma sufriese cambios, estos no implicasen una reestructuración completa de la implementación.

El criterio de diseño en cada módulo se basó en maximizar el paralelismo y baja latencia en todas las instancias, teniendo en mente su implementación futura en una FPGA, siguiendo los principios de la computación neuromórfica y tomando como referencia las decisiones de diseño de Loihi [1], sin pretender equiparar su nivel de sofisticación dado el alcance de este trabajo. En consecuencia, se decidió no implementar el mecanismo de entrenamiento, optando por un modelo de inferencia que soporte la arquitectura resultante del entrenamiento por software y que permita modificar parámetros y configuraciones de forma flexible.

### A. *Módulos de procesamiento*

La estructura general propuesta para la neurona consiste en una serie de módulos que recogen la actividad de la red de forma pasiva y la procesa para obtener un resultado de forma autónoma.

Para ser posible programar todos los subsistemas en lenguaje VHDL y lograr un código reutilizable que no implique un ajuste fino para cada posible caso que pueda darse dentro de la red, se recurrió a la estrategia de modularizar los elementos funcionales en fragmentos de código pequeños que converjan en los módulos principales mediante la exportación a modo de librerías. A partir de esta premisa se estableció un esquema de abstracción bajo las siguientes directivas:

- Se considera candidato a modularizar a cualquier proceso que no pueda ejecutarse con primitivas simples o un grupo pequeño de directivas.
- Se considera candidato a cualquier proceso en el que concurran una gran cantidad de entradas y/o señales.
- Se considera candidato a modularizar a cualquier proceso que pueda construirse a través de módulos más pequeños.

Por otra parte, se estableció como tipo de dato un formato signado de punto fijo Q1.6 [1]. Esta elección se justifica en el rango de activaciones de modelos entrenados en software, que oscila entre $-0{,}5$ y $1$, y en la conveniencia de operar con un tamaño de palabra estándar de 8 bits. Se evaluó la incorporación de un bit fraccionario adicional, concluyendo que el incremento marginal en precisión no justifica el coste adicional en recursos de hardware. En consecuencia, todo valor inferior a la resolución mínima se trunca a cero, y todo valor fuera del rango representable se satura al extremo correspondiente.

El error de cuantización introducido por este formato es de $\pm 0{,}0078125$ en el peor caso (la mitad del paso de cuantización). Dado que los pesos sinápticos y los potenciales de membrana del modelo entrenado se encuentran en el rango $[-0{,}5,\ 1]$, este error relativo se considera aceptable para el propósito de validación funcional de la arquitectura.

Bajo estas consideraciones se obtuvieron los siguientes módulos:

- Tabla de rutas (*RouteTable*): implementa una estructura de memoria direccionable que almacena múltiples identificadores en registros individuales y permite compararlos en paralelo con una palabra de entrada proveniente del bus de datos. Su función principal es determinar si el identificador presente en el bus coincide con alguno de los previamente almacenados, generando una señal de coincidencia para cada registro. Este módulo contiene un arreglo de registros parametrizable en número de neuronas (`M`) y ancho de palabra (`N`), de modo que cada registro puede almacenar un identificador distinto. Durante cada flanco ascendente del reloj, los valores de entrada (`data_in`) se escriben en los registros seleccionados por la señal de habilitación de escritura (`we`). En cada ciclo de evaluación, la palabra de entrada (`input_word`) se compara con el contenido de todos los registros en paralelo. El resultado de estas comparaciones se refleja en el vector `match_out`, donde cada bit indica si el registro correspondiente contiene un valor coincidente.
- LCA (*Lateral Competition Array*): implementa un mecanismo de competencia lateral paralelizada para generar una máscara de activación selectiva a partir de los datos de potencial de membrana almacenados. El módulo cuenta con un conjunto de `M` registros de 4 bits que pueden ser actualizados de manera independiente mediante señales de habilitación de escritura (`we`), permitiendo modificar los datos de entrada (`data_in`) durante los flancos ascendentes del reloj. La señal de lectura global (`re`) activa el proceso de evaluación sobre los registros alma-

[1] Q1.6 es un formato de punto fijo con signo que usa 1 bit para la parte entera (incluido el signo) y 6 bits para la parte fraccionaria. Representa valores entre aproximadamente -2 y +1.984375, con una resolución de 0.015625.

cenados. Internamente, los registros se descomponen en vectores de bits individuales, y cada vector se somete a una reducción lógica OR para detectar si algún registro contiene un '1' en esa posición de bit. A partir de estas señales globales, se generan las señales de selección que determinan cuál grupo de registros permanece habilitado. Finalmente, la máscara de salida se construye combinando las señales de selección con los bits correspondientes de cada registro, produciendo una activación localizada en aquellos que poseen el bit dominante según la prioridad definida.

- Árbol acumulador (*TreeAdder*): encapsula la suma de los pesos sinápticos almacenados en un array de registros mediante una estructura de sumadores en forma de árbol balanceado, capaz de procesar en paralelo `M` entradas de 8 bits. Este enfoque reduce la complejidad del hardware de crecimiento lineal a logarítmico, a costa de introducir una latencia también logarítmica, logrando un equilibrio entre coste y rendimiento. Esta arquitectura se adopta para mejorar la escalabilidad del número de sinapsis por neurona, evitando los cuellos de botella de implementaciones secuenciales o acumuladores lineales, que aumentarían el retardo crítico y limitarían la frecuencia máxima del sistema.
- Actualización del potencial de membrana (*LIFUpdate*): implementa la ecuación de evolución del potencial de membrana en una neurona de tipo LIF, modelando el comportamiento de integración y decaimiento del voltaje neuronal en función de la corriente de entrada y una constante configurable de pérdida (*loss*). El módulo recibe como entradas la corriente sináptica discretizada en formato fijo Q1.6 y una señal de reloj, junto con una señal de control que permite cargar dinámicamente la constante de decaimiento en un registro interno. El proceso de actualización se realiza en cuatro etapas principales: en primer lugar, se calcula la diferencia entre la corriente de entrada y el valor actual del potencial de membrana; posteriormente, este resultado se multiplica por una constante de pérdida almacenada en un registro configurable mediante la señal `load_loss`, lo que permite ajustar el factor de decaimiento temporal del sistema; a continuación, el valor obtenido se normaliza y se trunca al formato Q1.6 para mantener la coherencia numérica en toda la arquitectura; finalmente, el voltaje de membrana se actualiza de manera secuencial en cada flanco ascendente del reloj, aplicando la ecuación discreta (3).

$$v(t+1) = v(t) + \Delta v = v(t) + (I_{in} - v(t)) \times \text{loss} \quad (3)$$

- Sistema de umbral (*ThresholdSys*): implementa la función de comparación del potencial de membrana con un valor umbral configurable, que permite determinar el instante de disparo de una neurona en el modelo LIF. El módulo recibe una señal de reloj y un valor de entrada discretizado en formato Q1.6, junto con un conjunto de señales de control que permiten cargar dinámicamente el umbral de comparación mediante un registro interno. Su funcionamiento se organiza en dos procesos, uno gestiona la carga del valor de umbral en el registro de memoria cuando la señal de habilitación *load_en* se activa en un flanco ascendente del reloj, mientras que el restante, realiza la comparación entre el valor de entrada y el valor almacenado en el registro, generando una salida lógica en alto cuando el potencial de entrada supera o iguala el umbral definido mediante la salida *ge_out*. Este impulso un único registro donde se almacena el identificador de la neurona, publicando su actividad en la red.

### B. Comunicación

Para lograr una comunicación efectiva, cada neurona posee un identificador único dentro de la red. Este se enviará al momento del disparo a un bus compartido y permitirá a las demás neuronas identificar si el disparo es relevante para su actividad o no.

En segundo lugar, y en consecuencia de lo arriba mencionado, se resolvió que las neuronas solo estarían influenciadas por aquellas cuya fuerza en sus conexiones sinápticas supere un umbral determinado. De esta forma se reduce en gran medida el coste de cada neurona teniendo en cuenta la existencia de conexiones recurrentes en la capa intermedia, lo que supone un crecimiento casi exponencial en complejidad a nivel neurona y a nivel red si se desea hacer crecer la arquitectura en un futuro.

### C. Módulo de control

Para la simulación de esta red neuronal se concibió un módulo de control que cumpla con las mínimas funcionalidades de control de señales, con el fin de agilizar el proceso de evaluación de la red, en base a un pulso de reloj que activa las señales de control para cada etapa de todas las neuronas de forma secuencial, realiza un polling de la lista de neuronas para permitirles compartir su actividad por la red arbitrando de forma simplificada el bus de datos, para finalmente repetir el ciclo.

## IV. Pruebas y resultados

Las pruebas de validación se realizaron con datos controlados y de trazabilidad conocida, cargados mediante señales de control auxiliares. En todas las simulaciones se utilizó la herramienta Waveform provista por Quartus, con un reloj de 100 MHz y los siguientes parámetros de red: $M = 16$ neuronas en la capa recurrente, $N = 8$ bits de ancho de palabra.

Se validó en primera instancia cada módulo de forma individual y luego el modelo completo de la neurona LIF. La Fig. 2 muestra las formas de onda obtenidas. Se observa la lectura correcta de la tabla de ruteo (*RouteTable*), el filtrado por el mecanismo LCA al eliminar a la neurona de ID 1 frente al potencial de membrana mayor de la neurona ID 2 como se observa en la señal *enable*, la acumulación de pesos a través del *TreeAdder* y el disparo con envío de identificador al detectar que el potencial supera el umbral. Las señales

visibles de arriba a abajo corresponden a: identificadores de neuronas de entrada (`input_word`), potencial de membrana entrante (`membrane_in`), habilitador de lectura de actividad presináptica *re*, señales habilitadoras internas (`tree_sig`, `loss_sig`, `thresh_sig`), señales de control de salida de cada módulo (`tree_out`, `match_out`, `enable`, identificador de salida (`ID_OUT` y señales de salida y reset (`spike_out`, `spike_reset`. Posteriormente se realizó una prueba con estímulos externos ingresados por la señal `input_spike`, cuyo resultado se aprecia en la Fig. 3.

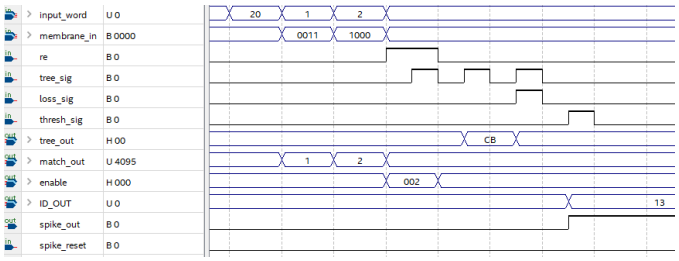



Fig. 2. Forma de onda de un *spike* generado a partir de la actividad simulada de otras neuronas en la red. La secuencia muestra la integración del potencial de membrana (`v_mem`) hasta superar el umbral y el consiguiente disparo (`spike_out`). Escala temporal: 10 ns/div.

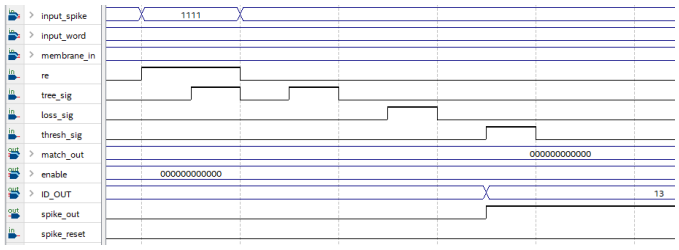



Fig. 3. Forma de onda generada por estímulos externos. La secuencia muestra la integración del potencial de membrana (`v_mem`) hasta superar el umbral y el consiguiente disparo (`spike_out`). Escala temporal: 10 ns/div.

Las pruebas sobre la red ya completamente conectada se basaron en controlar la correcta comunicación y propagación a través del bus de datos y la coordinación de señales globales. La red en sí fue alimentada con datos de ruteo sencillos generados por un contador y valores de pesos iguales a 0xFF para todas las neuronas. La prueba consistió en demostrar 2 ciclos de red espaciados temporalmente por una cantidad de ciclos suficiente para comprobar la correcta persistencia del estado de la red. En las Fig. 4 y 5 se puede apreciar cómo el primer ciclo fue alimentado con un estímulo fuerte de entrada (*input_spike*) para asegurar la actividad en la red. En el segundo ciclo sin estímulo externo, al activar otra vez las señales de control de las neuronas se vuelve a obtener un disparo, lo cuál valida exitosamente el mantenimiento y comunicación de los impulsos en la red.

## V. Conclusiones

La arquitectura resultante es modular y escalable, aunque esta flexibilidad introduce una latencia adicional por ciclo completo de procesamiento. Se implementó un modelo de

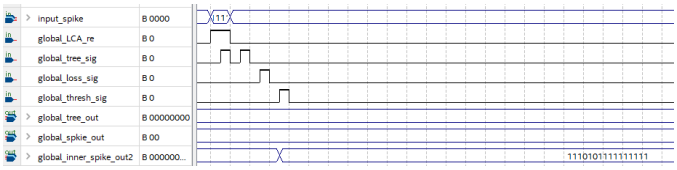



Fig. 4. Simulación del primer ciclo de procesamiento en la red neuronal de picos con estímulos externos. Escala temporal: 10 ns/div.

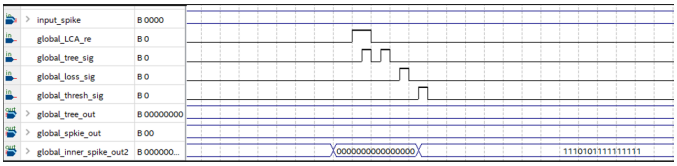



Fig. 5. Simulación del primer ciclo de procesamiento en la red neuronal de picos sin estímulo externo. Evidencia la propagación interna de actividad entre neuronas. Escala temporal: 10 ns/div.

cómputo no Von-Neumann con paralelismo a nivel de neurona y un módulo de control de complejidad reducida, prescindiendo de una máquina de estados finitos explícita. El diseño admite la incorporación de un módulo de aprendizaje sin requerir modificaciones estructurales significativas en la configuración actual. Con ajustes menores en el módulo de control, el sistema puede procesar estímulos externos a medida que ingresan, lo que constituye un paso hacia la operación en tiempo real con menor consumo energético respecto a arquitecturas Von-Neumann convencionales.

Como trabajo futuro se prevé: (i) la síntesis e implementación sobre una FPGA para obtener métricas reales de recursos y frecuencia máxima; (ii) el análisis del impacto de la cuantización en Q1.6 sobre la exactitud del modelo entrenado; y (iii) la integración de un mecanismo de aprendizaje en línea compatible con la arquitectura propuesta.

## Acknowledgment

Este trabajo se realizó en la Facultad de Ciencias Exactas y Tecnología, en el marco del Proyecto de Investigación “Sistemas Embebidos, Comunicaciones y Aplicaciones en Internet de las Cosas” (PIUNT E743).

## References

[1] M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya, G. A. Fonseca Guerra, P. Joshi, P. Plank y S. R. Risbud, “Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook,” *Proceedings of the IEEE*, vol. 109, no. 5, pp. 911–934, 2021.

[2] N. Rathi, I. Chakraborty, A. Kosta, A. Sengupta, A. Ankit, P. Panda y K. Roy, “Exploring Neuromorphic Computing Based on Spiking Neural Networks: Algorithms to Hardware,” *ACM Computing Surveys*, vol. 55, no. 12, art. 243, 2023.

[3] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose y J. S. Plank, “A Survey of Neuromorphic Computing and Neural Networks in Hardware,” arXiv:1705.06963, 2017.

[4] A. Rosado-Muñoz and M. Bataller-Mompeán and J. Guerrero-Martínez, “FPGA implementation of Spiking Neural Networks” in *IFAC Proceedings Volumes*, 2012.

ISBN 978-631-90145-4-9 - Congreso Argentino de Sistemas Embebidos CASE 2026 - 6 y 7 de agosto

# Optimización Energética de Nodos IoT con GPS a través de Modelos de Aprendizaje Automático

line 1: 1st Given Name Surname
xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxx

line 1: 1st Given Name Surname
xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxx

line 1: 1st Given Name Surname
xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxx

line 1: 1st Given Name Surname
xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxx

line 1: 1st Given Name Surname
xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxx

line 1: 1st Given Name Surname
xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxx

***Abstract*—***Este trabajo presenta una comparación de arquitecturas de optimización energética para nodos IoT aplicados al monitoreo de animales en ganadería extensiva, integrando comunicación LoRa, posicionamiento GPS y técnicas de inteligencia artificial para la predicción de movimiento. Se desarrolla un modelo detallado de consumo energético considerando múltiples estrategias de operación, incluyendo transmisión periódica, transmisión basada en eventos y agrupamiento de datos. Adicionalmente, se propone un modelo de un clasificador Random Forest embebido para reducir activaciones innecesarias del GPS. Los resultados muestran que el GPS representa más del 70% del consumo total del sistema. Mediante la incorporación de predicción de movimiento, se logra reducir las activaciones del GPS hasta en un 70%, disminuyendo el consumo promedio en más del 50% y extendiendo la vida útil de la batería de aproximadamente 8-10 días a más de 28-30 días. La propuesta permite reducir el mantenimiento en sistemas IoT autónomos en entornos rurales. El trabajo se encuentra en fase de validación experimental; no se presentan aún mediciones de campo. Se discuten limitaciones, escenario de prueba y trabajo futuro.*

***Keywords*—** *IoT, LoRa, GPS, Optimización Energética, Monitoreo de Ganado, Predicción de movimiento, Random Forest, Bajo Consumo.*

## I. Introducción

El uso de tecnologías de Internet de las Cosas (IoT) en la ganadería extensiva ha permitido avances significativos en trazabilidad, monitoreo del comportamiento animal y optimización de recursos productivos. Sin embargo, uno de los principales desafíos en este tipo de aplicaciones es la limitada disponibilidad energética y capacidad de procesamiento de los nodos, los cuales deben operar de forma autónoma durante largos períodos en zonas remotas, sin acceso a infraestructura de carga [1] [2].

En este contexto, las tecnologías LPWAN, como LoRa, han demostrado ser adecuadas para este tipo de aplicaciones debido a su bajo consumo y largo alcance [2] [3]. No obstante, los sistemas de posicionamiento global (GPS) continúan siendo uno de los principales consumidores de energía, limitando significativamente la autonomía del sistema [5].

Diversos trabajos han explorado la optimización del consumo mediante mejoras en la comunicación [6]. Estudios recientes comienzan a integrar aprendizaje automático en sistemas IoT ganaderos [7]. Sin embargo, la aplicación de modelos embebidos para la gestión adaptativa del GPS decidiendo cuándo encenderlo en función del movimiento estimado sigue siendo un área con escasa validación cuantitativa. [7]. Este trabajo combina y evalúa ambas aproximaciones, con énfasis en la gestión inteligente del GPS como principal consumidor energético.

Un clasificador Random Forest (RF) embebido, entrenado con estadísticas de un acelerómetro de bajo consumo, puede reducir las activaciones del GPS en al menos un 60% sin comprometer la precisión del seguimiento, extendiendo significativamente la vida útil de la batería.

Este artículo presenta un modelo energético teórico y una propuesta arquitectural. Los resultados experimentales y la validación del clasificador se encuentran en curso y serán publicadas en futuras versiones.

## II. Arquitectura del sistema

El nodo IoT propuesto integra cuatro subsistemas funcionales: sensor/actuador, microcontrolador de bajo consumo (MCU), módulo de posicionamiento GPS y radio LoRa. Los componentes de referencia utilizados para el modelado son: MCU familia Kinetis KL [9]; transceptor LoRa SX1276/RFM95 [10, 11]; receptor GPS u-blox NEO-6/7/8 [12]; y acelerómetro LIS2DW12 [13] (habilitado en el escenario con IA, consumo 50 nA en power-down, 1 uA en modo activo).

El nodo opera en modo super-loop alternando estados de bajo consumo (deep sleep), adquisición de datos (GPS), procesamiento (MCU) y transmisión (LoRa). La duración de cada estado determina el consumo energético total del ciclo.

## III. Modelos de consumo energético

### A. Formulación

La corriente promedio del nodo (Iavg) y el tiempo de vida útil de la batería (Tvida), se estiman como:

$$Iavg = (\sum (Ii * ti)) / T \quad [mA] \qquad (1)$$

$$Tvida = Cbateria / Iavg \quad [h] \qquad (2)$$

Donde: Ii: Corriente de cada subbloque [mA]
ti: Tiempo de activación de cada subbloque [s]
T: Tiempo de ciclo [s]
Cbatería: Capacidad de la batería [mAh]

### B. *Análisis de diferentes escenarios de trabajo*

1. Transmisión periódica (E1): lectura GPS cada 5 minutos, transmisión inmediata de cada dato. Máxima frecuencia de muestreo pero mayor uso de LoRa.

2. Transmisión por eventos (E2): lectura GPS periódica pero transmisión sólo cuando se detectan cambios de posición que sean relevantes, las mediciones menores a un cierto límite establecido se descartan. Reduce transmisiones, no reduce activaciones GPS.

3. Transmisión acumulada (E3): almacenamiento local cada 300 s y transmisión en bloque cada 1800 s. Reduce overhead de TX pero incrementa el tiempo de procesamiento en MCU. A diferencia del escenario E2, no se descartan mediciones.

### C. *Resultados por escenario*

La Tabla I muestra el consumo energético por fase para los tres escenarios. Los valores de tiempo son promedios horarios sobre el ciclo de 3600s.

TABLA I. Consumo por fase en diferentes escenarios de operación

| Fase | I [mA] | Escenario 1 | | Escenario 2 | | Escenario 3 | |
|---|---|---|---|---|---|---|---|
| | | t [s/h] | mAs/h | t [s/h] | mAs/h | t [s/h] | mAs/h |
| Sleep MCU | 0,010 | 3213 | 32,1 | 3233 | 32,3 | 3226 | 32,3 |
| MCU activo | 7 | 3 | 21 | 3 | 21 | 12 | 84 |
| GPS (fix cal.) | 40 | 360 | 14400 | 360 | 14400 | 360 | 14400 |
| TX LoRa | 130 | 24 | 3120 | 4 | 520 | 2 | 260 |
| **Total [mAs/h]** | | | **17573** | | **16013** | | **14776** |
| **Iavg [mA]** | | | **4,88** | | **4,15** | | **4,1** |
| **Tvida [días]** | | | **~8** | | **~10** | | **~10** |

El proceso que lleva obtener una posición válida (fix start) del GPS tiene un consumo de 40 mA, representa consistentemente el componente de mayor consumo, con 14400 mAs/h en todos los escenarios (82-90% del total). La optimización producida en los escenarios E2 y E3 con esta estrategia de comunicación, aumenta la vida útil de 8 a 10 días, mejora modesta respecto al potencial de optimizar las lecturas del GPS.

La siguiente Figura 1 evidencia la participación del GPS en el gasto energético total de un nodo bajo E1.

Estos resultados validan la hipótesis central: la gestión inteligente del GPS, y no la optimización de la comunicación, es la palanca de mayor impacto energético.

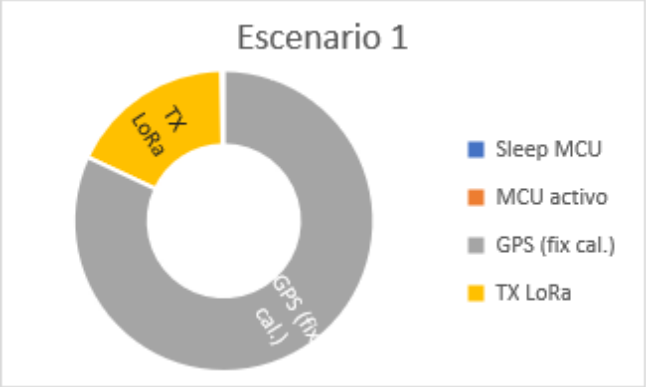



Figura 1: Participación del GPS y LoRa en el consumo de energía del nodo

## IV. Predicción de movimiento basada en RF

En este escenario (E4) se introduce una capa de inteligencia embebida mediante un modelo de Random Forest, orientado a mejorar la calidad de la detección de movimiento a partir de los datos de un acelerómetro. A diferencia de los enfoques anteriores, donde la activación del sistema GPS o la decisión de transmitir se basan en umbrales fijos o estrategias temporales, en E4 se utiliza un modelo entrenado previamente para clasificar el estado del dispositivo (por ejemplo: movimiento significativo vs. reposo, o movimiento válido vs. ruido). En términos operativos, el dispositivo realiza lecturas continuas del acelerómetro de bajo consumo, y sólo activa el GPS y/o la transmisión LoRa cuando el modelo predice un evento de movimiento relevante con alta probabilidad. Esto permite reducir tanto la cantidad de activaciones del GPS como las transmisiones innecesarias, optimizando el consumo energético global sin sacrificar la calidad de la información recolectada.

### A. *Descripción del modelo*

Los trabajos basados en doble integración del acelerómetro (dead-reckoning) [15] acumulan errores de deriva que crecen cuadráticamente con el tiempo, requieren calibración periódica de offset gravitacional y son sensibles a la orientación del dispositivo. Se propone reemplazar la integración inercial por un clasificador RF embebido [14], entrenado offline sobre los datos Inertial Measurement Unit (IMU) recolectados en campo y que opera directamente sobre las características estadísticas de la ventana FIFO del acelerómetro.

El modelo aprovecha las características etológicas bovinas: los animales se mueven lentamente (0.5-1 m/s), con alta redundancia espacial. Esto permite establecer si el animal se desplazó lo suficiente para justificar encender el GPS, usando únicamente estadísticas del acelerómetro de bajo consumo.

### B. *Extracción de características*

Se define el vector F = [a_mean, a_var, delta_t, E, P_desp] donde:

- a_mean: media de aceleración horizontal (ejes X e Y, a_xy = Raiz cuadrada ($x^2 + y^2$), eje Z, a_z = a_z - 9.8)
- a_var: varianza de aceleración
- delta_t: tiempo desde la última medición de GPS (RTC del MCU)
- E: energía de la ventana de aceleración = sum(a_i^2)/N
- P_desp: predicción del desplazamiento acumulado estimado por el RF

Todas las características se calculan en aritmética de punto fijo 32-bit en el MCU, sin biblioteca de punto flotante, minimizando tiempo de cómputo y consumo [18].

### C. *Entrenamiento del modelo*

El RF se entrena con N = 32 árboles (igual a la cantidad de muestras del FIFO del LIS2DW12 antes de interrumpir al MCU). Cada árbol produce una predicción P_desp; el promedio del bosque estima el desplazamiento del animal en esa ventana.

La salida binaria Y se define como:

Y = 1 (GPS ON) si P_desp >= umbral

Y = 0 (GPS OFF) si P_desp < umbral

Si se acumulan predicciones consecutivas Y=1 desde una posición inicial conocida, se estima el desplazamiento total y, al superar el umbral preestablecido, se activa el GPS para confirmar posición y se establece un nuevo punto de partida.

El entrenamiento requiere datos de campo con ground truth GPS: sesiones de 4-8 h en al menos 5 animales instrumentados, con etiquetado por comparación directa contra GPS de referencia. El modelo resultante ocupa ~4-6 KB de flash y es transferible a nuevos establecimientos con reentrenamiento fino [6].

### D. *Evaluación energética*

Para cuantificar el impacto del clasificador RF, se adopta un escenario representativo de 3600s donde el GPS se activa en 3 oportunidades (Y=1), lo que corresponde a una reducción de activaciones de ~70% respecto al escenario E1 (12 activaciones/hora). El cálculo de corriente media es:

MCU activo (leer, procesar, decidir) $7 \cdot 2 \cdot 12 = 168$ [mAs]

GPS + MCU: $40 \cdot 30 \cdot 3 = 3600$ [mAs]

MCU + LoRa TX: $130 \cdot 2 \cdot 3 = 780$ [mAs]

Acelerómetro activo $0.001 \cdot 3600 = 3{,}6$ [mAs]

Sleep: $0{,}01 \cdot [3600-(90 + 6+6)] = 34{,}98$ [mAs]

I_avg = ( (168 +3600+780+3,6+34,98) ) / 3600≈1.27 [mA]

Con batería de 1000 mAh entonces podemos calcular la duración teórica de la batería en:

T_vida = (1000 [mAh]) / (1,23 [mA]) = 813 h ≈ 33 días

Esto representa una mejora en más de tres (3) veces respecto a cualquiera de los tres escenarios (E1, E2, E3) sin predicción del desplazamiento.

La participación del GPS en esta modalidad de trabajo se reduce a valores cercanos al 70%, siendo todavía el valor más significativo en el gasto energético.

## V. Resultados y Discusión

El análisis cuantitativo permite extraer tres conclusiones estructurales:

1. El GPS representa más del 70% del consumo total en todos los escenarios analizados. La optimización de la estrategia de comunicación (E1 a E3) produce una mejora marginal de consumo del 16% mientras que la reducción de activaciones GPS mediante el modelo predictivo produce una mejora superior al 75%.
2. La optimización de la comunicación tiene impacto limitado: pasar de 24 transmisiones/hora (E1) a 2 transmisiones/hora (E3) reduce la vida útil sólo de 8 a 10 días. Esto confirma que el cuello de botella energético no es el radio LoRa sino el módulo GPS.
3. El modelo predictivo reduce el consumo de forma significativa: el clasificador RF embebido, con un footprint de 4-6 KB de flash, despreciables respecto a la cantidad de memoria flash en diferentes versiones del CORTEX M0 (128-256 KB) y latencia de inferencia menor a 1 ms [7], permite extender la autonomía a ~33 días sin agregar hardware de costo significativo respecto a los Escenarios 1-3.

## VI. Limitaciones y trabajo futuro

El estudio presenta limitaciones que deben considerarse en la interpretación de resultados. Los cálculos energéticos, para el caso de predicción de movimiento con RF, se basan en valores típicos de datasheets y modelos teóricos; las mediciones experimentales de campo están programadas como siguiente etapa de validación. El modelo energético no contempla factores ambientales como variación de temperatura, envejecimiento de la batería, ni pérdidas por regulación.

El modelo RF fue evaluado en escenarios simulados. La colecta de datos de campo para entrenamiento (instrumentación de al menos 5 animales durante 5 días) y la evaluación de transferibilidad del modelo entre establecimientos constituyen el trabajo experimental en curso.

El trabajo futuro incluye: (i) validación experimental en campo abierto; (ii) evaluación de clasificadores alternativos (SVM cuantizado, redes 1-D) en términos de exactitud vs. footprint de flash; (iii) implementación de hot start GPS para reducir el tiempo de fix de ~30s a ~5s, con impacto directo en el consumo energético; y (iv) diseño de PCB propio con gestión de energía solar.

## VII. Conclusiones

Se presentó una arquitectura de optimización energética para nodos IoT ganaderos basada en la integración de un clasificador Random Forest embebido para la gestión adaptativa del GPS. El modelo energético detallado para tres escenarios de recopilación/transmisión de datos demuestra

que el GPS representa más del 70% del consumo total en todos los casos analizados.

La incorporación del clasificador RF permite reducir las activaciones del GPS hasta en un 70%, disminuyendo la corriente media de 4.1-4.9 mA a aproximadamente 1.2-1,6 mA (dependiendo del umbral de decisión) y extendiendo la vida útil de la batería de 8-10 días a 28-33 días con una celda de 1000 mAh. Esto representa una mejora de más de tres veces la autonomía (Figura 2).

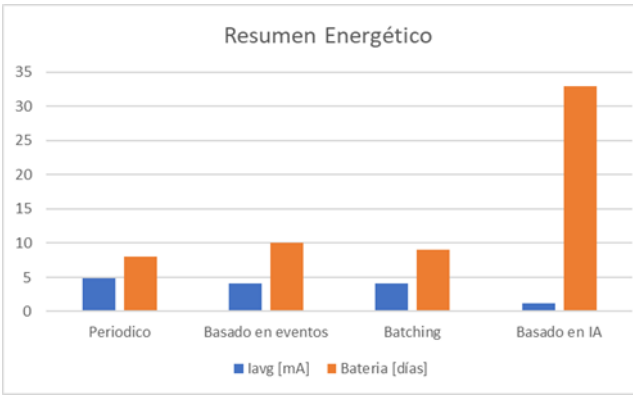



Figura 2: Gráfico comparativo de corriente promedio y duración de una batería de 1000mAh

La estrategia resulta especialmente efectiva en ganadería extensiva, donde los patrones de movimiento bovino son predecibles y el clasificador puede generalizarse a nuevos entornos con reentrenamiento fino. El modelo ocupa ~5 KB de flash, compatible con microcontroladores de 32 bits de bajo costo, y no requiere operaciones de punto flotante en inferencia.

Los resultados teóricos indican que la optimización energética en sistemas IoT de trazabilidad no debe centrarse únicamente en la comunicación sino principalmente en la gestión inteligente de los subsistemas del nodo, siendo el GPS el componente crítico en aplicaciones de seguimiento animal. Se espera que la validación experimental en curso, confirme las estimaciones presentadas en este trabajo.

## Referencias

[1] L. Germani et al., "An IoT Architecture for Continuous Livestock Monitoring Using LoRa LPWAN," Electronics, vol. 8, no. 12, p. 1435, 2019.

[2] J. P. Becona et al., "LoRa, Sigfox, and NB-IoT: An Empirical Comparison for IoT LPWAN Technologies in the Agribusiness," IEEE Embedded Syst. Lett., vol. 16, no. 3, pp. 283-286, 2024.

[3] H. Rajab et al., "Evaluation of energy consumption of LPWAN technologies," EURASIP J. Wireless Commun. Networking, 2023.

[4] H. Mohapatra, "A LoRa IoT Framework with Machine Learning for Remote Livestock Monitoring in Smart Agriculture," arXiv preprint, 2025.

[5] S. Suryavansh et al., "Ambrosia: Reduction in Data Transfer from Sensor to Server for Increased Lifetime of IoT Sensor Nodes," arXiv preprint, 2021.

[6] T. Valletta et al., "Applications of machine learning in animal behaviour studies," Animal Behaviour, vol. 124, pp. 203-220, 2017.

[7] P. Warden y D. Situnayake, TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers. O'Reilly Media, 2019.

[8] A. Al-Shammari et al., "Service Embedding in IoT Networks," IEEE Access, 2019.

[9] NXP Semiconductors, "MKL26Z64VLH4 Microcontroller Data Sheet." [Online]. https://www.nxp.com

[10] Semtech Corporation, "SX1276/77/78/79 LoRa Transceiver Data Sheet." [Online]. https://www.semtech.com

[11] HopeRF, "RFM95/96/97/98 LoRa Module Data Sheet." [Online]. https://www.hoperf.com

[12] u-blox, "NEO-6/7/8 GNSS Modules Data Sheet." [Online]. https://www.u-blox.com

[13] STMicroelectronics, "LIS2DW12 MEMS ultra-low-power accelerometer," DS11546 Rev. 5, 2020.

[14] S. Giri et al., "Application of machine learning to predict transport modes from GPS, accelerometer, and heart rate data," Int. J. Health Geogr., vol. 21, no. 1, p. 19, 2022.

[15] J. Sanz, O. Salazar, "Determinación de la aceleración, velocidad y desplazamiento utilizando acelerómetros micromaquinados," Scientia et Technica, no. 24, 2004.

[16] A. Gonzalez et al., "Evaluation of GPS-GSM livestock tracking devices," Computers and Electronics in Agriculture, vol. 198, p. 107024, 2022.

[17] Los autores, "Nodo IoT de Ultra-Bajo Consumo para Seguimiento Ganadero Basado en GPS Adaptativo y Red LoRaWAN," CASE 2025 (trabajo previo).

[18] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825-2830, 2011.

Categoría

# Reporte

Internet de las Cosas

Área Temática

# Arquitectura de Nodo IoT para Monitoreo de Suelos: Front-End Analógico con Conversión Tensión-Frecuencia en 130 nm CMOS versus ADC Convencional

Pedro N. Petrashin
Facultad de Ingeniería – Universidad Católica de Córdoba. Córdoba, Argentina
nicopetrashin@gmail.com

Juan Castagnola
Facultad de Ingeniería – Universidad Católica de Córdoba. Córdoba, Argentina
juanluiscastagnola@gmail.com

Walter Lancioni
Facultad de Ingeniería – Universidad Católica de Córdoba. Córdoba, Argentina
walterlancioni@gmail.com

Pablo A. Petrashin
Facultad de Ingeniería – Universidad Católica de Córdoba. Córdoba, Argentina
pablo.petrashin@ucc.edu.ar

***Resumen*—Se presenta una arquitectura de nodo IoT para monitoreo multiparámetro de suelos agrícolas (pH, temperatura, humedad, nitratos) basada en un ASIC CMOS de 130 nm con conversión tensión-frecuencia (V-F) en el front-end analógico. El núcleo del sistema es un conversor tensión-corriente (V-I) de tres transistores que controla un oscilador de relajación compartido por los cuatro canales mediante multiplexado en tiempo. La salida codificada en frecuencia es leída directamente por el timer capture/compare de un microcontrolador de bajo consumo, eliminando la necesidad de un ADC externo. Se presenta un análisis comparativo detallado entre el enfoque V-F propuesto y una solución basada en ADC SAR convencional, evaluando consumo energético, área de silicio, resolución efectiva e inmunidad a ruido en campo. Los resultados de simulación SPICE con modelos BSIM4 sobre el proceso IHP SG13G2 muestran una no linealidad integral (INL) de 2,21 % sobre el rango completo de 0–400 mV del sensor de pH, una sensibilidad de 4,4 kHz/mV y un consumo total del ASIC inferior a 35 µA a VDD = 1,1 V. El ciclo completo de medición de los cuatro parámetros se completa en menos de 10 ms, compatible con operación LoRa de muy bajo consumo.**

***Palabras clave—nodo IoT, agricultura de precisión, conversor tensión-frecuencia, ASIC CMOS 130 nm, ADC SAR, sensor de pH, sistemas embebidos***

## I. Introducción

La agricultura de precisión requiere el monitoreo continuo de múltiples parámetros del suelo. El pH determina la disponibilidad de nutrientes; la temperatura afecta la actividad microbiana; la humedad volumétrica es crítica para el riego; y la concentración de nitratos indica el estado nutricional del cultivo. Monitorear estos cuatro parámetros simultáneamente y de forma inalámbrica exige nodos IoT de ultra bajo consumo desplegables en grandes extensiones con autonomía de meses [1].

El diseño del front-end analógico de estos nodos plantea una disyuntiva fundamental: convertir las señales de los sensores mediante un ADC convencional o mediante una cadena de conversión tensión-frecuencia (V-F). La solución ADC es la más difundida en sistemas comerciales, pero impone restricciones de consumo, área y ruido que resultan problemáticas en nodos IoT de muy bajo consumo alimentados por baterías o energía solar [2]. El enfoque V-F, en cambio, codifica la información en la frecuencia de un oscilador, cuya salida digital puede ser leída directamente por el timer de cualquier microcontrolador sin hardware adicional.

Este trabajo presenta una arquitectura de nodo IoT completa basada en un ASIC de conversión V-F diseñado en el proceso IHP SG13G2 de 130 nm de código abierto [3], y realiza un análisis comparativo sistemático entre el enfoque V-F propuesto y una solución ADC SAR de referencia. El objetivo es proveer criterios cuantitativos para la selección de arquitectura de front-end en sistemas embebidos de monitoreo agrícola. La Sección II describe la arquitectura del nodo IoT. La Sección III detalla el front-end analógico V-F. La Sección IV presenta el comparativo con ADC SAR. La Sección V reporta los resultados de simulación. Las Secciones VI y VII discuten y concluyen.

## II. Arquitectura del Nodo IoT

### *A. Visión General del Sistema*

El nodo IoT propuesto (Fig. 1) se compone de cuatro bloques principales: el ASIC de front-end analógico, un microcontrolador de bajo consumo (MCU), un módulo de radio LoRa y la gestión de energía (celda solar + supercapacitor). Los cuatro sensores de suelo se conectan directamente al ASIC, que entrega sus señales codificadas en frecuencia al MCU mediante líneas digitales estándar.

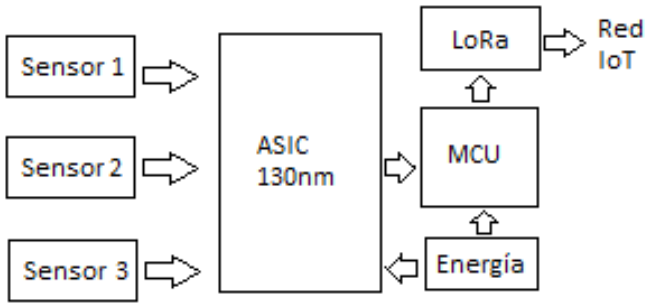



Fig. 1 Arquitectura del nodo IoT para monitoreo

Las especificaciones principales del nodo se resumen en la Tabla I. El ciclo de operación es el siguiente: (1) el MCU despierta desde deep sleep cada intervalo programable (típicamente 15 min); (2) habilita el ASIC mediante una señal de enable; (3) mide secuencialmente los cuatro canales por multiplexado en tiempo usando su timer capture/compare

interno; (4) construye el paquete de datos; (5) transmite por LoRa y vuelve a deep sleep. El tiempo total de actividad es inferior a 10 ms, lo que permite una autonomía de años con una batería de 2000 mAh o un supercapacitor recargado por panel solar pequeño (5 cm²).

**TABLA I. ESPECIFICACIONES DEL NODO IOT**

| Parámetro | Valor | Condición |
|---|---|---|
| Parámetros medidos | pH, T°, humedad, $NO_3^-$ | — |
| Tensión de alimentación | 1,1 V (ASIC) / 1,8 V (MCU) | Batería Li-ion |
| Corriente ASIC activo | <35 µA | Todos los canales |
| Corriente ASIC sleep | <100 nA | Enable=0 |
| Tiempo de medición | <10 ms | 4 canales, mux |
| Interfaz MCU | Timer capture, 1 pin/canal | Sin ADC externo |
| Radio | LoRa 868 MHz | SF7, BW125 |
| Autonomía estimada | >18 meses | Ciclo 15 min, bat. 2 Ah |

### *B. Multiplexado en Tiempo de los Canales*

Los cuatro módulos de front-end del ASIC (V-I para pH, R→I para temperatura, oscilador capacitivo para humedad y potenciostato para nitratos) comparten un único oscilador de relajación mediante multiplexado en tiempo controlado por el MCU. Una señal de selección de 2 bits conecta el oscilador al módulo activo en cada instante. El tiempo de establecimiento entre canales es inferior a 50 µs, despreciable frente a la ventana de medición de 1 ms por canal.

Este esquema de multiplexado reduce el área del ASIC al eliminar cuatro osciladores independientes, y garantiza que el MCU solo necesita cuatro pines de entrada digital para leer todos los parámetros, sin conversor analógico-digital de ningún tipo.

## III. FRONT-END ANALÓGICO: CONVERSOR V-I PARA PH

### *A. Topología*

El sensor electroquímico de pH entrega una tensión diferencial de 0–400 mV siguiendo la ecuación de Nernst (−59,2 mV/pH). El módulo V-I convierte esta tensión en una corriente proporcional que carga el capacitor C1 del oscilador de relajación, produciendo una frecuencia de oscilación $f \propto I_{out}$.

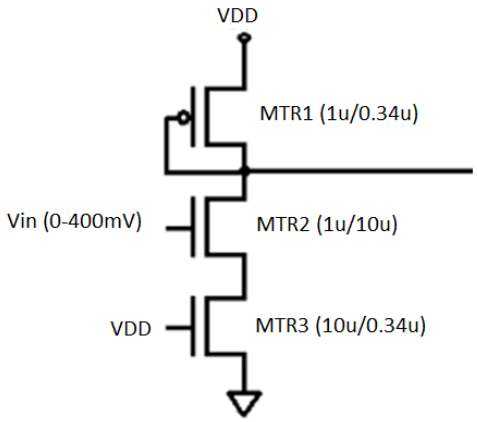



Fig. 2. Diagrama a nivel transistor del conversor V-I

La celda de conversión emplea tres transistores (Fig. 2): MTR1 (PMOS diodo, W/L = 1/0,34 µm) como carga activa; MTR2 (NMOS canal largo, W/L = 1/10 µm) como transistor de señal; y MTR3 (NMOS triodo profundo, W/L = 10/0,34 µm) como resistencia de degeneración de fuente integrada. Una red resistiva Rhi = 530 kΩ / Rlo = 700 kΩ / Rin = 1 kΩ desplaza la señal del sensor al rango de compuerta 0,625–1,025 V necesario para operación lineal de MTR2 en saturación.

### *B. Análisis de Pequeña Señal*

La transconductancia efectiva del conversor es $G_{m,eff} = g_{m2}/(1 + g_{m2} \cdot R_t)$, donde $R_t \approx 5$ kΩ es la resistencia en triodo de MTR3. Al punto medio (Vin = 200 mV): $g_{m2} \approx 38$ µA/V, factor de degeneración $1 + g_{m2} \cdot R_t \approx 1{,}19$, $G_{m,eff} \approx 32$ µA/V. La fuente dominante de no linealidad es la variación de Rt con la tensión de fuente de MTR3 ($\Delta R/R \approx 30$ %), que produce una INL de 2,21 % — aceptable para el requisito agronómico de ±0,1 pH [4].

## IV. COMPARATIVO: ENFOQUE V-F VS. ADC SAR

### *A. Arquitectura de Referencia ADC SAR*

La alternativa convencional para este tipo de nodo es un ADC de aproximaciones sucesivas (SAR) externo o integrado en el MCU. Los MCUs de ultra bajo consumo actuales (p. ej. STM32L0, MSP430FR) incluyen ADC SAR de 12 bits con frecuencias de muestreo de hasta 1 MSPS y consumo típico de 150–300 µA durante la conversión. Para leer cuatro canales analógicos, el MCU debe permanecer activo con el ADC encendido durante todo el tiempo de adquisición, y requiere además un buffer de entrada (seguidor de tensión) para no cargar los sensores electroquímicos de alta impedancia.

En el enfoque V-F propuesto, el ASIC realiza la conversión de forma autónoma con su propio oscilador y entrega una señal digital al MCU. El MCU solo necesita abrir una ventana de conteo de 1 ms por canal con el timer en modo capture, un periférico de consumo insignificante (~2 µA) que no requiere que el núcleo ARM esté ejecutando código durante la medición.

### *B. Comparación Cuantitativa*

La Tabla II presenta la comparación sistemática entre ambos enfoques para el caso de uso específico de este nodo: cuatro canales de sensores de suelo, ciclo de medición cada 15 minutos, alimentación a 1,8 V para el MCU y 1,1 V para el ASIC.

**TABLA II. COMPARATIVO V-F (ESTE TRABAJO) VS. ADC SAR CONVENCIONAL**

| Criterio | V-F (este trabajo) | ADC SAR (ref.) |
|---|---|---|
| Corriente conversión | 35 µA (ASIC) | 150–300 µA (ADC+buffer) |
| Tiempo activo/ciclo | <10 ms | 10 ms + overhead MCU |
| Carga de batería/ciclo | 0,35 nAh | 0,8–1,2 nAh |
| Autonomía (bat. 2 Ah, 15 min) | >18 meses | 8–12 meses |
| Resolución efectiva | ~10 bits (1 ms gate) | 12 bits (SAR) |
| Pines MCU requeridos | 4 (timer capture) | 4 (ADC mux) + ref. externo |
| Buffer de entrada | No requerido | Sí (op-amp externo) |
| Inmunidad a ruido | Alta (digital out) | Media (señal analógica) |
| Área ASIC estimada | ~0,05 mm² | N/A (ADC en MCU) |
| Complejidad PCB | Baja (1 chip) | Media (pasivos, buffer) |
| Costo componentes | Bajo | Medio (op-amp externo) |

### C. Resolución Efectiva del Enfoque V-F

La resolución del enfoque V-F está determinada por la ventana de conteo T_gate y el rango de frecuencia. Con T_gate = 1 ms y f_max = 2,03 MHz, el número máximo de cuentas es N_max = f_max·T_gate = 2030, equivalente a $\log_2(2030) \approx 11$ bits de resolución. Descontando la INL de 2,21 % (equivalente a ~0,9 bits de pérdida), la resolución efectiva (ENOB) es de ~10,1 bits, comparable a un ADC SAR de 10 bits y suficiente para los requisitos agronómicos de todos los parámetros medidos [4].

El error de frecuencia del oscilador por variaciones PVT (±20 % sin calibración) es compensado por el mecanismo de ajuste Vcal integrado en el ASIC, que permite calibrar el sistema en campo con una solución de pH conocida en el encendido inicial, procedimiento estándar en sensores electroquímicos.

### D. Limitaciones del Enfoque V-F

El enfoque V-F presenta dos limitaciones principales respecto al ADC SAR: (i) la resolución no es arbitrariamente ampliable sin aumentar T_gate (y por ende el tiempo activo), mientras que un ADC SAR de 16 bits existe en silicio sin incrementar el tiempo de conversión; y (ii) la no linealidad intrínseca del oscilador requiere calibración individual por unidad en aplicaciones de alta precisión (pH < ±0,02). Para el caso de uso agrícola con requisito ±0,1 pH, estas limitaciones no son determinantes.

## V. Resultados de Simulación

### A. Linealidad DC del Conversor V-I

Se realizó un barrido DC de 0 a 400 mV en LTspice con modelos BSIM4 (LEVEL 14) del proceso IHP SG13G2. La Tabla III compara la INL de tres iteraciones de diseño del conversor V-I, mostrando el impacto de las decisiones de dimensionamiento.

TABLE I. **TABLA III. Evolución del Diseño: INL y Rango de Frecuencia**

| Configuración | INL p-p (%) | f mín (kHz) | f máx (MHz) |
|---|---|---|---|
| Triodo, L2=4 µm, C1=100 pF | 10,47 | 195 | 0,81 |
| Rext=40 kΩ, L2=8 µm, C1=10 pF | 1,48 | 256 | 1,86 |
| Final: triodo W=10 µm, L2=10 µm | 2,21 | 256 | 2,03 |

### B. Característica de Transferencia f(Vin)

La Fig. 3 muestra la frecuencia de oscilación en función de la tensión de entrada para el diseño final (C1 = 10 pF). La curva varía cuasi-linealmente de 256 kHz en Vin = 0 V a 2,03 MHz en Vin = 400 mV, con una sensibilidad promedio de 4,4 kHz/mV. La desviación máxima respecto a la recta ideal es de ±22 kHz, ocurriendo en el punto medio (Vin ≈ 200 mV), consistente con la INL pico a pico de 2,21 %.

### C. Sensibilidad Térmica

La Tabla IV muestra la variación de la frecuencia de salida al punto medio (Vin = 200 mV) en el rango de temperatura industrial. La variación de ±21 % es compensable por calibración con Vcal o por corrección por software usando la medición simultánea de temperatura del mismo nodo.

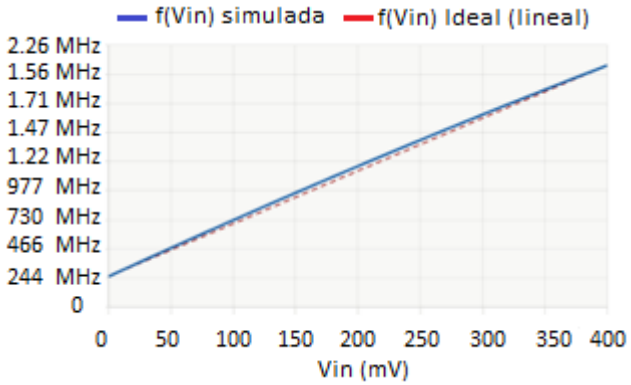



Fig. 3. Comparativa f(Vin) del conversor V-I con salida de oscilación

**TABLA IV. Sensibilidad Térmica (Vin = 200 mV)**

| Temperatura | Iout (µA) | f salida (kHz) | Δf/f @ 25 °C |
|---|---|---|---|
| −25 °C | 9,8 | 892 | −11,2 % |
| 25 °C (nom.) | 11,1 | 1005 | — |
| 85 °C | 13,4 | 1217 | +21,1 % |

### D. Presupuesto Energético del Ciclo Completo

La Tabla V compara la energía consumida por ciclo de medición completo (4 canales) en ambos enfoques, considerando el MCU STM32L0 como plataforma de referencia para el caso ADC SAR. La reducción de consumo del enfoque V-F permite más que duplicar la autonomía del nodo.

**TABLA V. Energía por Ciclo de Medición Completo (4 canales)**

| Bloque | V-F (este trabajo) | ADC SAR (ref.) |
|---|---|---|
| ASIC front-end | 0,35 nAh | N/A |
| MCU (timer activo) | 0,02 nAh | — |
| MCU (ADC + núcleo) | — | 0,83 nAh |
| Buffer op-amp | N/A | 0,35 nAh |
| Radio LoRa (tx) | 1,1 nAh | 1,1 nAh |
| Total por ciclo | 1,47 nAh | 2,28 nAh |
| Autonomía (bat. 2 Ah, 15 min) | >18 meses | ~11 meses |

## VI. Discusión

### A. Criterios de Selección de Arquitectura

Los resultados del comparativo permiten establecer criterios cuantitativos para la selección de front-end en nodos IoT agrícolas. El enfoque V-F es superior cuando: (i) la resolución requerida es ≤12 bits; (ii) la autonomía es crítica (>12 meses); (iii) el presupuesto de PCB es reducido (elimina op-amp y pasivos); y (iv) el entorno tiene ruido electromagnético significativo (campo agrícola, motores, riego por aspersión). El enfoque ADC SAR es preferible cuando se requiere resolución > 12 bits sin calibración, o cuando el sistema ya integra un MCU con ADC de alta calidad y el consumo no es el factor limitante.

### B. Escalabilidad a Otros Parámetros

La arquitectura V-F es directamente aplicable a los otros tres canales del nodo. El canal de temperatura usa un conversor R→I basado en espejo de corriente con degeneración que convierte la resistencia del termistor PTC en

corriente. El canal de humedad emplea un oscilador capacitivo a ~350 MHz cuya frecuencia varía con la permitividad del suelo [1]. El canal de nitratos usa un potenciostato de tres electrodos con conversión de corriente amperométrica a frecuencia. Los cuatro módulos comparten el mismo oscilador de relajación, reduciendo el área total del ASIC y garantizando consistencia en la calibración de frecuencia.

### *C. Trabajo Futuro*

Las principales líneas de trabajo futuro son: (i) simulaciones de esquinas PVT completas (SS/FF/SF/FS a −25 °C y 85 °C) para validar los márgenes de diseño; (ii) análisis de Montecarlo de tolerancias de resistencias y descasamiento de transistores; (iii) layout del módulo V-I con extracción RC y simulación post-layout; e (iv) integración del ASIC completo en una placa de prototipo con MCU nRF9160 (NB-IoT) para validación en campo en parcelas experimentales de la región de Córdoba.

## VII. Conclusión

Se presentó una arquitectura de nodo IoT para monitoreo multiparámetro de suelos agrícolas basada en conversión tensión-frecuencia en un ASIC CMOS de 130 nm. El front-end analógico propuesto emplea un conversor V-I de tres transistores que alcanza una INL de 2,21 % sobre 0–400 mV sin resistencias poli ni componentes externos, con consumo total inferior a 35 µA a 1,1 V.

El análisis comparativo con un front-end ADC SAR convencional demuestra que el enfoque V-F reduce la energía por ciclo de medición en un 35 %, duplicando la autonomía del nodo de ~11 a >18 meses con la misma batería. La salida codificada en frecuencia se conecta directamente al timer del MCU sin ADC externo, simplificando el hardware y mejorando la inmunidad al ruido en entornos agrícolas.

Estos resultados posicionan al enfoque V-F como alternativa cuantitativamente superior al ADC convencional para aplicaciones IoT de agricultura de precisión con restricciones severas de energía y costo.

## Referencias

[1] W. Lancioni, P. A. Petrashin y J. Castagnola, "A Novel 350 MHz Capacitive Soil Moisture Sensor for Precision Agriculture," JAREE, 2025.

[2] J. Gubbi, R. Buyya, S. Marusic y M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, 2013.

[3] IHP Microelectronics, "IHP Open Source PDK (SG13G2)," 2024.

[4] S. Musa, "Electrochemical pH sensing for precision agriculture: A review," Sensors, vol. 21, no. 14, art. no. 4855, 2021.

[5] A. Veeravalli, E. Sánchez-Sinencio y J. Silva-Martínez, "Transconductance amplifier structures with very small transconductances," IEEE Journal of Solid-State Circuits, vol. 37, no. 6, pp. 770–775, Jun. 2002.

[6] STMicroelectronics, STM32L0 Series Ultra-Low-Power MCUs: Datasheet, 2023.

[7] E. Sánchez-Sinencio y J. Silva-Martínez, "CMOS transconductance amplifiers, architectures and active filters: a tutorial," IEEE Circuits Syst. Mag., vol. 1, no. 2, pp. 4–12, 2001.

[8] A. J. Bandodkar et al., "Electrochemical Sensors for Sustainable Precision Agriculture—A Review," Frontiers in Chemistry, vol. 10, art. no. 848320, May 2022.

Categoría

# Reporte

Software Embebido

Área Temática

ISBN 978-631-90145-4-9 - Congreso Argentino de Sistemas Embebidos CASE 2026 - 6 y 7 de agosto

# Pre-Diseño del Controlador de Vuelo para un UAV

Enzo Gabriel Verstraete Mauro Gastón Gómez Sergio David Figueroa
*Dept. de Ing. Electrónica, FCEFyN - UNC*
*Dept. Sistemas Electrónicos, Informáticos y Telecomunicaciones, CIA - FAA*
Córdoba, Argentina
enzo.verstraete@mi.unc.edu.ar, mg.gomez@unc.edu.ar, sdfigueroa@faa.mil.ar

***Abstract*—Este trabajo presenta el portado y la validación funcional preliminar de ArduPilot sobre una plataforma basada en el microcontrolador STM32H753ZI, empleada como base para el desarrollo de un controlador de vuelo personalizado para UAV. La implementación se realizó mediante la configuración de una capa de abstracción de hardware (HAL) específica, definida a través del archivo `hwdef.dat`, integrando sensores inerciales, barométricos, magnéticos y GNSS mediante interfaces SPI, $I^2C$ y UART. Además, se incorporó un subsistema de telemetría externa mediante scripting en Lua, utilizando un enlace UART con un dispositivo Castle Serial Link para la adquisición de variables del ESC, como tensión, corriente, velocidad y temperatura. La validación se llevó a cabo mediante pruebas funcionales en laboratorio, verificando la inicialización del firmware, el reconocimiento de periféricos, la comunicación MAVLink con Mission Planner y la ejecución del script de telemetría. Los resultados obtenidos permiten verificar la viabilidad del portado y establecer una base técnica para una futura implementación en PCB propio.**

***Index Terms*—UAV, ArduPilot, STM32H7, hardware abstraction layer, firmware porting, embedded systems, Lua scripting**

## I. Introducción

El desarrollo de Vehículos Aéreos No Tripulados (UAV) se apoya ampliamente en controladores de vuelo comerciales que integran hardware y firmware en plataformas cerradas, limitando la capacidad de adaptación del sistema a requerimientos específicos y el acceso a configuraciones de bajo nivel.

El uso de firmware de código abierto como ArduPilot permite desacoplar el software de la plataforma física, habilitando su ejecución sobre hardware embebido de propósito general. El portado constituye un desafío técnico, requiriendo la correcta configuración de la capa de abstracción de hardware (HAL), la asignación de periféricos y la gestión de recursos en tiempo real.

El presente trabajo aborda el portado de ArduPilot sobre una plataforma basada en un microcontrolador STM32, validada inicialmente mediante la placa de evaluación del mismo. La arquitectura se organiza en torno a una unidad de procesamiento central encargada de ejecutar el firmware, gestionar los periféricos de sensado y mantener la comunicación con la estación de tierra. El enfoque se centra en la configuración del sistema mediante el archivo `hwdef.dat`, donde se define la correspondencia entre los recursos físicos de la plataforma y su representación dentro de ArduPilot, permitiendo la integración de sensores y módulos de comunicación.

Adicionalmente, se explora la capacidad de extensión funcional del sistema mediante scripting en lenguaje Lua, permitiendo la integración de dispositivos externos sin modificar el núcleo del firmware. Como caso de estudio, se implementa la adquisición de telemetría de un controlador electrónico de velocidad (ESC) mediante un enlace serial dedicado (Castle Serial Link).

La validación se realiza mediante pruebas funcionales en laboratorio, verificando la inicialización del sistema, la detección de periféricos y la comunicación con la estación de tierra. En particular, se comprueba la consistencia de las variables de estado reportadas por el sistema, incluyendo posición GPS y ángulos de actitud (roll, pitch y yaw), frente a estímulos controlados sobre la plataforma, así como la ejecución de tareas extendidas mediante scripting.

## II. Arquitectura del Sistema

La arquitectura se divide en cuatro bloques funcionales principales: procesamiento, sensado, comunicación/almacenamiento y telemetría extendida.

### A. Unidad de Procesamiento

Como núcleo del sistema se seleccionó el microcontrolador STM32H753ZI, basado en ARM Cortex-M7. Esta elección responde a la necesidad de disponer de una plataforma compatible con el ecosistema ArduPilot, con suficiente capacidad de memoria Flash, RAM y desempeño computacional para ejecutar tareas de estimación, comunicación y gestión de periféricos en tiempo real.

El microcontrolador opera como punto central de integración entre el firmware, los sensores externos y los módulos de comunicación. La configuración específica del hardware se realiza mediante el archivo `hwdef.dat`, donde se definen el modelo de MCU, el orden de los puertos serie, los buses SPI e $I^2C$, las salidas PWM, las entradas ADC, la interfaz USB-OTG y el almacenamiento externo.

### B. Subsistema de Sensado

El subsistema de sensado integra sensores inerciales, magnéticos, barométricos y GNSS, con el objetivo de verificar la compatibilidad funcional del firmware con distintos periféricos, buses y dispositivos definidos en la HAL.

Se empleó una IMU ICM-42688-P como sensor inercial principal, junto con un magnetómetro MMC5983MA y un barómetro BMP581 mediante SPI, priorizando buses dedicados, mayor tasa de transferencia y selección independiente por dispositivo. Como complemento, se incorporó una IMU ICM-20948 mediante $I^2C$, debido a las restricciones de su placa de evaluación y para validar dicho bus, mientras que el módulo GNSS u-blox se integró por UART, interfaz adecuada para la recepción periódica de tramas de posicionamiento a baja tasa de datos. La presencia de múltiples sensores permite además evaluar una capacidad de redundancia soportada por la arquitectura.

Las especificaciones técnicas detalladas de los sensores utilizados, incluyendo rangos dinámicos, resolución y niveles de ruido, pueden consultarse en sus respectivas hojas de datos [5]–[10].

### C. Comunicación y almacenamiento

La comunicación con la estación de tierra se realiza mediante MAVLink a través del puerto USB-OTG de la placa de evaluación, permitiendo la visualización de variables de estado en Mission Planner. Adicionalmente, se habilitan interfaces UART para módulos externos, incluyendo GNSS y telemetría auxiliar.

Para el almacenamiento de registros y la ejecución de scripts se incorporó una tarjeta MicroSD mediante la interfaz SDMMC. Este recurso resulta necesario para habilitar funcionalidades avanzadas de ArduPilot, como el registro de datos de operación y la ejecución de rutinas en Lua.

### D. Telemetría extendida mediante scripting

Como extensión funcional del sistema, se integró un dispositivo Castle Serial Link mediante la interfaz UART6. Esta conexión permite adquirir variables provenientes del controlador electrónico de velocidad (ESC) del motor, tales como tensión, corriente, velocidad de giro y temperatura.

La adquisición se implementó mediante un script en Lua ejecutado dentro del entorno de scripting de ArduPilot. El script consulta periódicamente registros del dispositivo Castle Serial Link, convierte los valores crudos a magnitudes físicas y los reporta a la estación de tierra mediante mensajes MAVLink de texto. De esta forma, se valida la capacidad del sistema para incorporar periféricos externos sin modificar el núcleo del firmware.

El flujo funcional implementado puede resumirse como:

$$\text{ESC} \rightarrow \text{Castle Serial Link} \rightarrow \text{UART6} \rightarrow \text{Script Lua} \rightarrow \text{ArduPilot} \rightarrow \text{Mission Planner}$$

Esta solución permite evaluar la extensibilidad de la plataforma, aunque introduce dependencias adicionales sobre el scheduler, la memoria disponible y el sub-sistema de almacenamiento requerido para scripting.

### E. Interconexiones

La interconexión de estos componentes se detalla en el diagrama de bloques de la Fig. 1, donde se especifican los buses de comunicación utilizados.

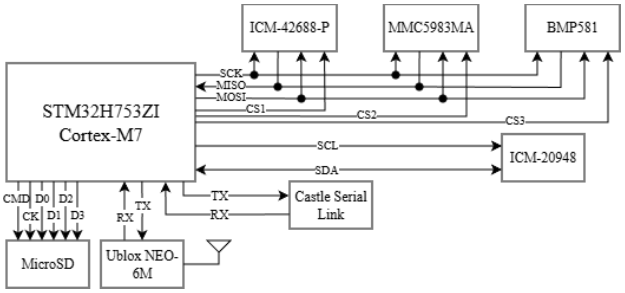



Fig. 1. Diagrama de bloques del hardware validado.

## III. Implementación de Firmware y Portado

El portado de ArduPilot hacia la plataforma STM32H753ZI se realizó mediante la configuración de una capa de abstracción de hardware (HAL) específica, permitiendo la adaptación del firmware a los recursos disponibles en la placa de evaluación. Este proceso se basa en la definición del archivo `hwdef.dat`, el cual establece la correspondencia entre los periféricos físicos y su representación lógica dentro del sistema.

El enfoque adoptado consistió en una configuración incremental del sistema, comenzando por la inicialización del microcontrolador y la comunicación básica, seguida de la incorporación progresiva de periféricos y funcionalidades.

### A. Configuración Base del Microcontrolador

Se definió la plataforma de ejecución sobre la familia del MCU, especificando el modelo, el oscilador externo de 8 MHz y el tamaño de memoria Flash disponible. Estos parámetros permiten establecer la base para la generación de frecuencias internas y la organización del firmware.

```
MCU STM32H7xx STM32H753xx
APJ_BOARD_ID AP_HW_NUCLEO_H753ZI
OSCILLATOR_HZ 8000000
FLASH_SIZE_KB 2048
```

### B. Gestión de Memoria y RTOS

Se configuraron parámetros críticos del sistema operativo en tiempo real ChibiOS, incluyendo tamaños de stack para distintos hilos y la base de tiempo del scheduler. Asimismo, se reservó espacio en memoria Flash para el bootloader y almacenamiento persistente.

ChibiOS se emplea por su integración con ArduPilot en plataformas STM32, aportando planificación de hilos, temporización y abstracción de recursos de bajo nivel para tareas concurrentes de sensado, comunicación y almacenamiento.

Durante el proceso de portado, la correcta asignación de memoria resultó un factor crítico para la estabilidad del sistema, siendo necesario ajustar tamaños de stack para evitar fallos en tiempo de ejecución asociados a desbordamientos.

```
MAIN_STACK 0x800
IDLE_STACK 0x400
#define HAL_STORAGE_THREAD_STACK_SIZE 8192
FLASH_RESERVE_START_KB 128

STM32_ST_USE_TIMER 12
#define CH_CFG_ST_RESOLUTION 16
```

### C. Configuración Global del Sistema

Se definió el orden de los puertos serie y la configuración de buses internos, permitiendo la correcta correspondencia con los índices SERIALx utilizados por ArduPilot.

### D. Interfaces de comunicación

Se configuraron múltiples interfaces UART para la conexión de módulos externos, incluyendo GNSS y telemetría. Asimismo, se habilitó el bus CAN y la interfaz USB-OTG para comunicación con la estación de tierra.

```
PB6 USART1_TX USART1
PB7 USART1_RX USART1
...
PD0 CAN1_RX CAN1
PD1 CAN1_TX CAN1
```

### E. Interfaces de alta velocidad y almacenamiento

El bus SPI1 se utilizó como interfaz principal para sensores, mientras que la memoria MicroSD se implementó mediante el periférico SDMMC en modo de 4 bits, permitiendo el almacenamiento de registros y la ejecución de scripts.

```
PA5 SPI1_SCK SPI1
PA6 SPI1_MISO SPI1
PB5 SPI1_MOSI SPI1
PC8  SDMMC1_D0 SDMMC1
...
#define HAL_OS_FATFS_IO 1
#define FATFS_HAL_DEVICE SDCD1
```

### F. Asignación de Periféricos Funcionales

Se configuraron salidas PWM para el control de actuadores y entradas ADC para la medición de variables de energía. Estas configuraciones permiten la integración con sub-sistemas típicos de control de vuelo.

```
PE9 TIM1_CH1 TIM1 PWM(1)
...
PB15 TIM1_CH3N TIM1 PWM(12)
PC4 BAT_VOLTAGE ADC1_INP14 ADC1 SCALE(1)
PC5 BAT_CURRENT ADC1_INP15 ADC1 SCALE(1)
```

### G. Integración de Sensores

Los sensores se integraron mediante buses SPI e $I^2C$, definiendo dispositivos específicos dentro de la HAL. La IMU principal se configuró sobre SPI, mientras que una IMU secundaria se conectó mediante $I^2C$ como respaldo funcional.

```
PD14 IMU_CS CS
...
SPIDEV icm42688 SPI1 DEVID1 IMU_CS MODE3 2*MHZ 8*MHZ
SPIDEV mmc5983  SPI1 DEVID2 MAG_CS MODE0 1*MHZ 1*MHZ
SPIDEV bmp581   SPI1 DEVID3 BARO_CS MODE3 7.5*MHZ 12*
    MHZ
IMU Invensensev3 SPI:icm42688 ROTATION_NONE
IMU Invensensev2 I2C:0:0x68 ROTATION_NONE
BARO BMP581 SPI:bmp581
```

Listing 1. Integración de sensores y redundancia

### H. Configuración de GPIO

Se definieron estados por defecto para los pines GPIO con el fin de evitar condiciones indeterminadas durante el arranque del sistema. También se habilitó la entrada PPS del GPS para sincronización temporal.

### I. Integración de scripting

Se habilitó el subsistema de scripting de ArduPilot, el cual permite la ejecución de scripts en lenguaje Lua utilizando almacenamiento externo. Sobre esta base, se implementó un script para la adquisición de telemetría desde un dispositivo Castle Serial Link mediante UART.

La ejecución del script introduce una carga adicional sobre el sistema, afectando el uso de memoria y el comportamiento temporal del scheduler. Durante las pruebas, fue necesario ajustar parámetros de memoria para evitar fallos asociados a la ejecución concurrente de tareas del sistema y scripting.

Este mecanismo permite extender la funcionalidad del sistema sin modificar el firmware principal.

## IV. Resultados y Validación

La validación del sistema se realizó mediante pruebas funcionales en entorno de laboratorio, con el objetivo de verificar la correcta integración del firmware ArduPilot sobre la plataforma STM32H753ZI y la interacción con los periféricos definidos en la HAL. Para estas pruebas se realizo una placa de integración observada en la Fig. 2.

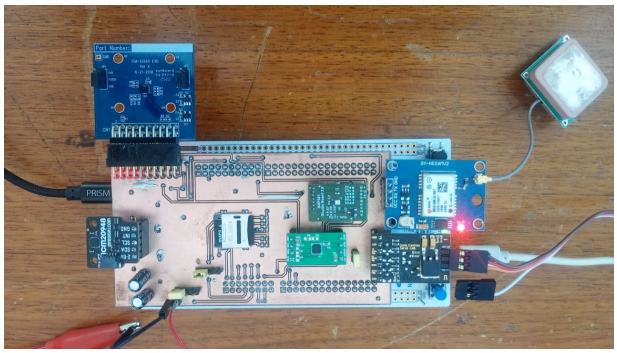


Fig. 2. Placa de interconexión.

### A. Inicialización y estabilidad del sistema

El sistema logró una inicialización completa del firmware sin fallos críticos, validando la correcta configuración de la capa de abstracción de hardware. Durante la operación continua no se observaron reinicios espontáneos ni bloqueos, evidenciando una configuración adecuada de los recursos del sistema y del entorno de ejecución basado en ChibiOS.

### B. Comunicación con la estación de tierra

Se estableció un enlace de telemetría mediante MAVLink a través del puerto USB-OTG, permitiendo la interacción con la estación de tierra Mission Planner. Se verificó la recepción continua de variables de estado del sistema, incluyendo actitud, posición y estado general, sin pérdidas evidentes de comunicación durante las pruebas. Como se ve en la Fig. 3.

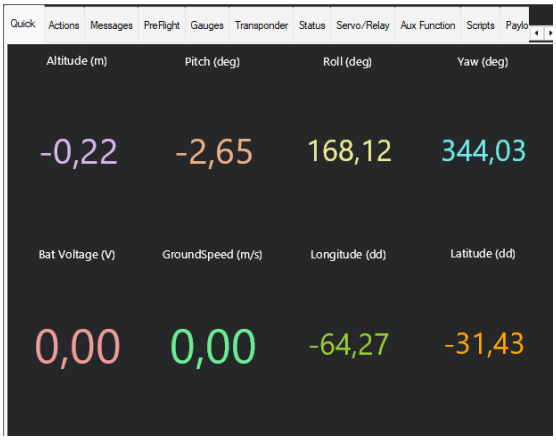



Fig. 3. Validación de conexión con Mission Planner.

### C. Integración y validación funcional del sensado

El sistema permitió la detección automática de los sensores configurados en el archivo `hwdef.dat`, observándose su correcta identificación dentro de la estación de tierra. Este comportamiento valida la correspondencia entre la configuración de hardware y la inicialización de dispositivos dentro de la HAL. Como se observa en la Fig. 4

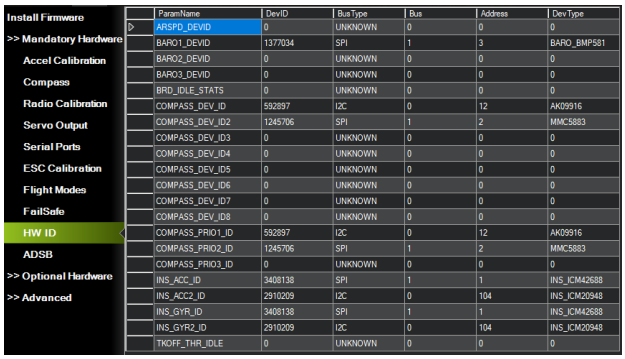



Fig. 4. Validación de módulos de hardware con Mission Planner.

Adicionalmente, se realizaron pruebas mediante la aplicación de movimientos controlados sobre la plataforma, verificando la consistencia de los ángulos de actitud (roll, pitch y yaw) reportados por el sistema. Asimismo, se comprobó la recepción de datos de posición provenientes del módulo GNSS, confirmando el correcto funcionamiento del pipeline de adquisición (flujo de procesamiento) y procesamiento de sensores.

### D. Validación de telemetría externa

Se verificó la correcta ejecución del subsistema de scripting mediante la adquisición de variables provenientes de un controlador electrónico de velocidad (ESC) a través de un dispositivo Castle Serial Link. El sistema permitió la lectura periódica de tensión, corriente, velocidad de giro y temperatura, las cuales fueron transmitidas a la estación de tierra mediante mensajes MAVLink de texto como se ve en la Fig. 5

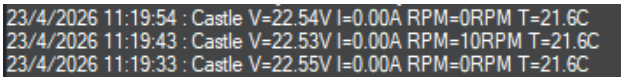



Fig. 5. Validación de Castle Serial Link.

La actualización de estas variables se realizó en intervalos del orden de segundos, consistente con la implementación del script y las restricciones temporales del entorno de ejecución.

## V. Conclusiones

Se demostró la viabilidad del portado del firmware ArduPilot sobre una plataforma basada en el microcontrolador STM32H753ZI, logrando la correcta inicialización del sistema y la integración funcional de sensores y periféricos mediante la configuración de la HAL a través del archivo `hwdef.dat`.

Asimismo, se validó la capacidad de extensión funcional mediante scripting en Lua, integrando telemetría externa a través de un dispositivo Castle Serial Link sin modificar el núcleo del firmware, evidenciando la flexibilidad de la plataforma.

La validación realizada es de carácter funcional, quedando como trabajo futuro la caracterización cuantitativa del desempeño, el análisis de consumo de recursos y la implementación en un PCB propio junto con estudios de integridad de señal y compatibilidad electromagnética.

## References

[1] ArduPilot Dev Team, ”ArduPilot Firmware Source Code,” GitHub Repository, 2026. [Online]. Available: https://github.com/ArduPilot/ardupilot. [Accessed: 20-Apr-2026].

[2] ArduPilot Team, ”ArduPilot Documentation,” 2026. [Online]. Available: https://ardupilot.org/ardupilot/. [Accessed: 20-Apr-2026].

[3] ArduPilot Community, ”ArduPilot Discourse Forum,” 2026. [Online]. Available: https://discuss.ardupilot.org/. [Accessed: 20-Apr-2026].

[4] STMicroelectronics, ”STM32H753xI Datasheet: High-performance and DSP with FPU Arm Cortex-M7 MCU,” DS12110, 2023. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32h753vi.pdf.

[5] STMicroelectronics, ”UM2407: STM32 Nucleo-144 boards (MB1364) user manual ” 2018. [Online]. Available: https://www.st.com/resource/en/user_manual/um2407-stm32-nucleo144-boards-mb1364-stmicroelectronics.pdf

[6] TDK InvenSense, ”ICM-42688-P High Performance 6-Axis SmartMotion Hub Datasheet,” v1.7, 2023. [Online]. Available: https://invensense.tdk.com/wp-content/uploads/2021/11/DS-000347-ICM-42688-P-v1.7.pdf.

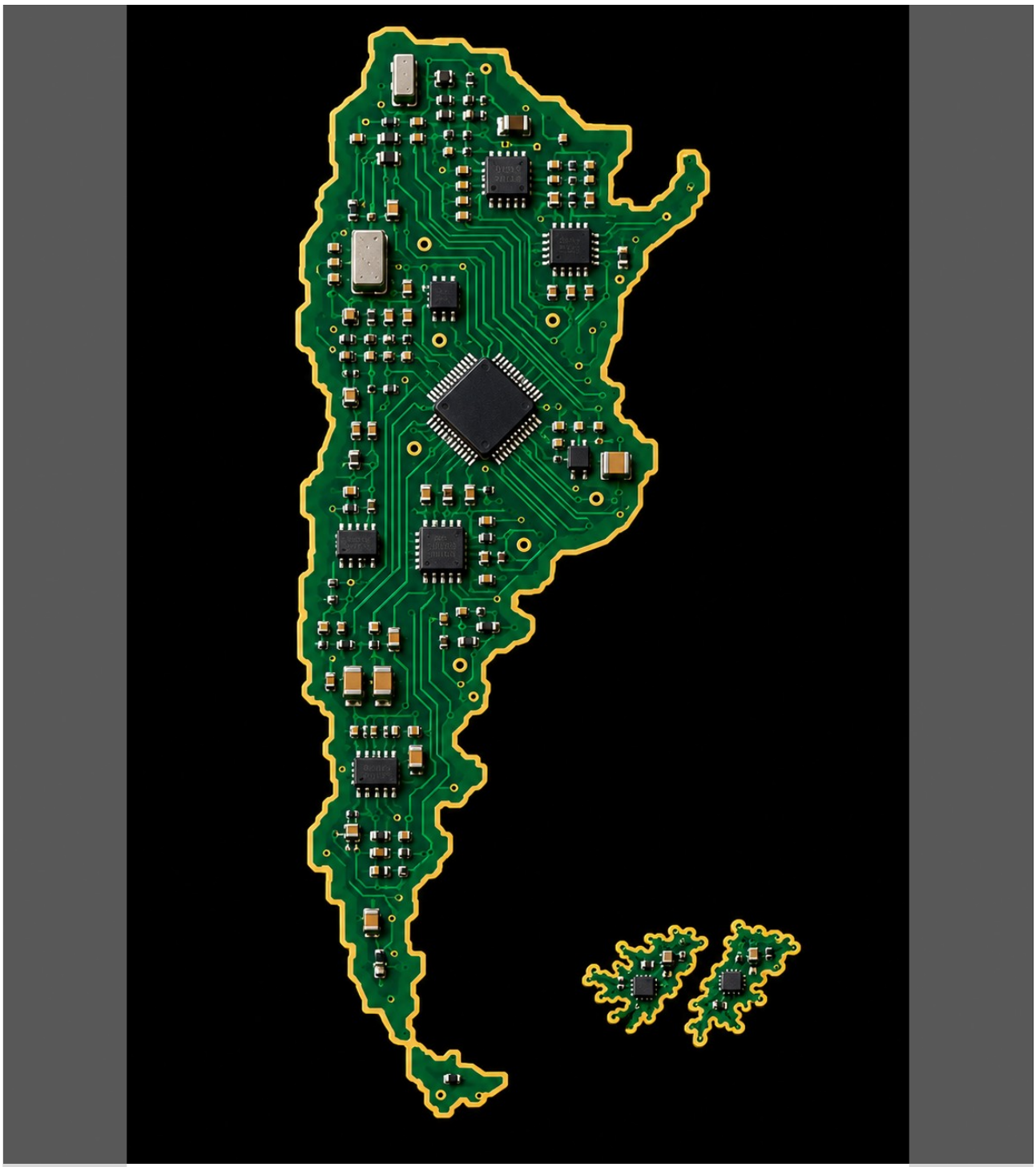
[7] TDK InvenSense, ”ICM-20948 9-Axis MotionTracking Device Datasheet,” v1.3, 2017. [Online]. Available: https://invensense.tdk.com/wp-content/uploads/2016/06/DS-000189-ICM-20948-v1.3.pdf.

[8] Asahi Kasei Microdevices, ”AK09916 3-axis Electronic Compass Datasheet,” 2016. [Online]. Available: https://www.akm.com/content/dam/documents/products/magnetic-sensors/magnetic-compass/ak09916c/ak09916c-en-datasheet.pdf.

[9] Bosch Sensortec, ”BMP581 Barometric pressure sensor Datasheet,” v1.2, 2022. [Online]. Available: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmp581-ds001.pdf.

[10] MEMSIC Inc., ”MMC5983MA High Performance Magnetic Sensor Datasheet,” 2021. [Online]. Available: https://www.memsic.com/Public/Uploads/uploadfile/files/20220119/MMC5983MA_Datasheet.pdf.

[11] u-blox, ”NEO-6 u-blox 6 GPS Modules Data Sheet,” v15, 2011. [Online]. Available: https://content.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf.



ISBN 978-631-90145-4-9

Congreso Argentino de Sistemas Embebidos

CASE 2026

**Libro de trabajos en modalidad artículo, foro tecnológico y reporte**