

# Congreso Argentino de Sistemas Embebidos



CA  
SE

2023

10 y 11 de agosto - Bahía Blanca



Libro de trabajos

UTN  bhi

AGENCIA  
NACIONAL DE PROMOCION  
CIENTIFICA Y TECNOLÓGICA

CONICET



UNIVERSIDAD TECNOLÓGICA NACIONAL  
Facultad Regional Bahía Blanca

UNS  UNIVERSIDAD  
NACIONAL  
DEL SUR

# CASE 2023

## Libro de Trabajos

Modalidades Artículo, Foro Tecnológico y Reporte

Congreso Argentino  
de  
Sistemas Embebidos

10 y 11 de agosto de 2023

CONICET



AGENCIA

NACIONAL DE PROMOCION  
CIENTIFICA Y TECNOLOGICA



UTN  bhi

UNIVERSIDAD TECNOLOGICA NACIONAL  
Facultad Regional Bahía Blanca

UNS  UNIVERSIDAD  
NACIONAL  
DEL SUR

ISBN 978-631-90145-0-1



9 786319 014501

Congreso Argentino de Sistemas Embebidos-CASE 2023 : libro de trabajos /  
Antonelli, Maximiliano / Zacchigna, Federico G. / De Micco, Luciana / Cebedio, Celeste / Lutenberg, Ariel /  
Brenji, Diego Javier / Lipovetzky, José - 1a ed - Ciudad Autónoma de Buenos Aires : ACSE -  
Asociación Civil para la investigación, Promoción y Desarrollo de Sistemas Eléctricos  
Embebidos, 2023.  
Libro digital, PDF

Archivo Digital: descarga y online  
ISBN 978-631-90145-0-1

1. Circuitos Electrónicos. 2. Hardware. 3. Software. I. Antonelli, Maximiliano  
CDD 004

Fecha de catalogación: 08/08/2023

Libro de Trabajos  
Modalidades Artículo, Foro Tecnológico y Reporte  
**Congreso Argentino de Sistemas Embebidos - CASE 2023**

**Editores:**

Antonelli, Maximiliano	UNMDP/ICYTE/CONICET
Zacchigna, Federico G.	FIUBA
De Micco, Luciana	UNMDP/ICYTE/CONICET
Cebedio, Celeste	UNMDP/ICYTE
Lutenberg, Ariel	FIUBA/CONICET
Brenji, Diego	INTI/UNLaM/FIUBA
Lipovetzky, José	FIUBA/CBA/CONICET
Gak Szollosy, Joel	UCU

Copyright © 2023

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 Argentina, 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 forma parte del 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 2023 se realizó en modalidad presencial el 10 y 11 de agosto, en la ciudad de Bahía Blanca, organizado en conjunto por la Universidad Nacional del Sur y la Facultad Regional Bahía Blanca de la Universidad Tecnológica Nacional.

Los objetivos que persigue el congreso son:

- Ofrecer un lugar de encuentro para investigadores y becarios de todo el país, 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 argentinas.

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

- Arquitectura de procesadores
- Bioingeniería
- Procesamiento Digital de Señales (Digital Signal Processing, DSP)
- FPGAs, HDLs y ASICs
- Implementación de Sistemas Embebidos
- Internet de las Cosas (Internet of Things, IoT)
- Linux Embebido
- Hardware Abierto (Open Source Hardware)
- Protocolos y Comunicaciones
- Robótica

- Sistemas Operativos en Tiempo Real (Real-Time Operating Systems, RTOS)
- Software Embebido
- Inteligencia Artificial

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 4 trabajos en la modalidad Artículo, 12 en modalidad Foro Tecnológico y 14 en la modalidad Reporte, de un total de 36 trabajos presentados.

Esta publicación se encuentra también disponible en forma *online* en la página web: [www.sase.com.ar/case/](http://www.sase.com.ar/case/)

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

Atentamente,

**Comité Organizador CASE**

## **Auspiciantes**

- ANPCyT (Agencia Nacional de Promoción Científica y Tecnológica)
- CONICET (Consejo Nacional de Investigaciones Científicas y Técnicas)
- UNS (Universidad Nacional del Sur)
- UTN (Universidad Tecnológica Nacional)
- CLEI (Centro Latinoamericano de Estudios en Informática)
- EMTECH S.A.
- INDIE Semiconductor
- Paltatech S.A.
- CADIEEL (Cámara Argentina de Industrias Electrónicas, Electromecánicas Lumino-técnicas)
- Allegro MicroSystems

## **Entidades organizadoras**

- UTN-BHI (Facultad Regional Bahía Blanca de la Universidad Tecnológica Nacional)
- UNS (Universidad Nacional del Sur)
- 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. José Lipovetzky (FIUBA/CBA/CONICET)
- Dr. Ariel Lutenberg (FIUBA/CONICET)
- Ing. Federico G. Zacchigna (FIUBA)
- Dr. Gak Szollosy, Joel (UCU)

## **Comité organizador local**

- Ing. Adrián Laiuppa (UTN-BHI)
- Dr. Rodrigo Santos (DIEC-UNS)
- Lic. Jose Moyano (DCIC-UNS)
- Dr. Gabriel Eggly (DIEC-UNS)

## **Chairs temáticos**

- |  |                                   |
|--|-----------------------------------|
| ● Bioingeniería:                           | Ing. Juan Manuel Reta (UNER)      |
| ● Comunicaciones y protocolos:             | Ing. Ignacio Zaradnik (UNLaM)     |
| ● FPGA, HDL y ASIC:                        | Ing. Salvador Tropea (INTI)       |
| ● Implementación de<br>Sistemas Embebidos: | Dr. Julio Dondo (UNSL)            |
| ● Inteligencia Artificial:                 | Ing. Claudio Verrastro (CNEA)     |
| ● IoT:                                     | Ing. Alejandro Furfaro (UTN-FRBA) |
| ● Open Source Hardware:                    | Mg. Ing. Sergio Burgos (UTN-FRP)  |
| ● Software Embebido:                       | Dr. Ricardo Cayssials (UNS)       |

## Revisores

Alessandrini, Gustavo  
Alvarez, Nicolas  
Antonelli, Maximiliano  
Avalos Ribas, Ramiro  
Barranco, Alejandro  
Burgos, Enrique Sergio  
Carbonetto, Sebastian  
Carnaghi, Marco  
Cayssials, Ricardo  
Cazarez Castro, Nohe Ramón  
Cebedio, Maria celeste  
De Micco, Luciana  
Dondo Gazzano, Julio  
Etcheverry, Juan Alberto  
Ferreyra, Pablo Alejandro  
Filomena, Eduardo  
Furfaro, Alejandro  
Gak, Joel  
Gallina, Sergio Hilario

Gutiérrez, Marcelo  
Irrazabal, Emanuel  
Leiva, Lucas  
Lopresti, Raúl  
Lutenberg, Ariel  
Maldonado Robles, Yazmin  
Marchi, Edgardo  
Millicovsky, Martín  
Pastafiglia, Daniel  
Rabioglio, Lucas  
Ramírez de Avila, Hedy Cesar  
Ramos Fabián, Clemente Andrés  
Regueiro-Gómez, Angel  
Reta, Juan Manuel  
Ridolfi, Pablo  
Tropea, Salvador  
Uriz, Alejandro José  
Verrastro, Claudio Verrastro  
Zaradnik, Ignacio



# Índice de trabajos

<b>Introducción</b>	I
<b>Índice de trabajos</b>	VIII
<b>Artículo</b>	1
<b>Implementación de SE</b> . . . . .	1
<b>Control activo de ruido impulsivo usando un nuevo algoritmo de conmutación MVC,</b> <i>A. Soto; X. Maya; Á. Vázquez; J. Avalos; G. Sánchez; J. Sánchez</i> . . . . .	2
<b>Estimación de cobertura de cápsulas en bebedero mediante procesamiento de imágenes embebido,</b> <i>E. Carbajo; L. Leiva; S. Fernandez; F. Sagües; M. Junco; J. Toloza; I. Guerrero; M. Vázquez; C. Saumell</i> . . . . .	3
<b>Protocolos y Comunicaciones</b> . . . . .	4
<b>Estimador de ruido de Middleton clase A: implementación sobre FPGA,</b> <i>L. Rabioglio; M. Cebedio; L. Arnone; L. De Micco; J. Moreira</i> . . . . .	5
<b>Cripto-codificación en Turbo código no sistemático mediante Interleaver dinámico para estándar 5G,</b> <i>R. Lopresti; M. Antonelli; J. Moreira; L. De Micco</i> . . . . .	6
<b>Foro Tecnológico</b>	7
<b>Bioingeniería</b> . . . . .	7
<b>HEROxS,</b> <i>R. D'Eboli; R. García; J. Schmidt; J. Oreggioni</i> . . . . .	8
<b>Prototipo de sistema para el análisis viscoelástico de lágrimas,</b> <i>G. Muñoz; M. Millicovsky; J. Reta; J. Cerrudo; A. Peñalva; M. Zalazar</i> . . . . .	11
<b>FPGAs, HDLs y ASICs</b> . . . . .	14
<b>Modulador QPSK con parámetros ajustables: Diseño e implementación,</b> <i>M. Morel; L. De Micco</i> . . . . .	15
<b>Hardware Abierto</b> . . . . .	18
<b>Pico RGB2HDMI - Conversor de Video RGB a HDMI y USB basado en Microcontrolador,</b> <i>M. Lorenzati Sanz; J. Ducloux</i> . . . . .	19
<b>Implementación de SE</b> . . . . .	22

Nuevo circuito neuronal basado en sistemas neuronales tipo P spiking para el cálculo del algoritmo de Euclides, <i>L. García; M. Arroyo; G. Sánchez; J. Avalos; E. Vázquez</i> . . . . .	23
<b>Inteligencia Artificial</b> . . . . .	26
Arreglos de antenas inteligentes para validación de mensajes ADS-B con SDR y machine learning, <i>F. Escobar; D. Díaz; E. Tardivo; N. Veglia; D. Primo; D. Badino</i> . . . . .	27
<b>Protocolos y Comunicaciones</b> . . . . .	30
Impedancímetro en la banda de UHF mediante técnicas de reflectometría, <i>F. Alekseenko; B. Gluzman; R. Ribas; A. Uriz; J. Moreira</i> . . . . .	31
Micro Red IOT multipropósito, <i>W. Gemin; J. López; M. Kuzman; R. Russo</i> . . . . .	34
<b>Robótica</b> . . . . .	37
Differential-drive mobile robot controller with ROS 2 support, <i>D. Albarrán; D. Ruiz; J. Nicolodi; D. Gonzalez-Dondo; G. Perez-Paina</i> . . . . .	38
<b>Software Embebido</b> . . . . .	41
Sistema de comprobación en tiempo real de señales del autopiloto de una aeronave., <i>A. Montiel Ramos; B. Rodríguez Cárdenas; A. Matos Rodríguez</i> . . . . .	42
Implementación de la técnica del Modelo Basado en Diseño para el desarrollo de Estación Receptora ADS-B, <i>D. Primo; D. Díaz; F. Escobar; J. Oviedo; J. Vittar; D. Badino</i> . . . . .	45
Ecosistema de Interoperabilidad basado en C++/MicroPython sobre plataformas Raspberry y ESP32, <i>J. Niño; M. Politi; M. Gulfo; H. Laiz; L. Lucangioli; C. Quiroga</i> . . . . .	48
<b>Reporte</b> . . . . .	51
<b>Bioingeniería</b> . . . . .	51
Sistema para la caracterización de concentración de ozono en agua potable, <i>A. Rivas Gamallo; A. Regueiro Gómez; Y. Almeida Pichardo; L. Noa Llorens</i> . . . . .	52
<b>FPGAs, HDLs y ASICs</b> . . . . .	55
A New Fault Injection Scheme Optimized for RISCv Soft Processors, <i>P. Ferreyra; A. Edelstein; J. de Villareal; A. Gómez; J. Naguil; C. Barrientos</i> . . . . .	56
Herramienta para el diseño e implementación de filtros digitales basada en software libre, <i>L. Ledesma; A. Dachary; E. Burgos</i> . . . . .	59
<b>Implementación de SE</b> . . . . .	62
Desarrollo de una Unidad Automática para la Detección de Radiación UV de Uso 24/7, con Acceso a una Red Local Vía Wi-Fi., <i>M. Lavorato; N. Urbano Pintos; M. Pagura</i> . . . . .	63
Prototipo de Sistema de Parqueo Inteligente para la zona Edificio Central de la UPTC sede Tunja, <i>I. Ruge Ruge; O. Carolina; J. Rolando</i> . . . . .	66
Evaluación de Filtros Digitales para la Medición del Flujo de Hidrógeno, <i>I. Zaradnik; R. Spano; L. Soria</i> . . . . .	69
Sistema IoT de monitoreo vehicular para el control de flotas, <i>J. Quiroga; M. Kuzman</i> . . . . .	72
<b>Inteligencia Artificial</b> . . . . .	75
Herramienta de Machine Learning para IoT, <i>M. Aranda; E. Piray; P. Beltramini; E. Gonzalez; G. Futo</i> . . . . .	76

Segmentación de objetos en interiores a partir de sensores ToF de bajo costo, <i>N. Urbano Pintos; H. Lacomí; M. Lavorato</i> . . . . .	79
<b>Internet de las Cosas</b> . . . . .	82
Miniboyas ambientales para monitoreo de ecosistemas acuáticos, <i>I. Zaradnik; J. Slawiski; D. Lupi; D. Caccaviello; D. Turconi</i> . . . . .	83
Sistema de gestión de instrumental, <i>A. Lorenzut</i> . . . . .	86
Control de cultivos hortícolas con sensores y actuadores mediante una plataforma en la nube basada en contenedores, <i>C. Biale; A. Lutenberg</i> . . . . .	89
<b>Protocolos y Comunicaciones</b> . . . . .	92
Sistema visualizador de señales de radar empleando hardware reconfigurable., <i>L. Guevara Trujillo; M. Sánchez Posada; A. Hernández Reyes</i> . . . . .	93
<b>Software Embebido</b> . . . . .	96
Cuellos de botella en sistemas embebidos en la adquisición y transmisión de señales analógicas, <i>R. González; L. Gómez; M. Millán; J. Oreggioni</i> . . . . .	97

Categoría

**ARTÍCULO**

Implementación de SE

Área Temática

# Control activo de ruido impulsivo usando un nuevo algoritmo de conmutación MVC

Adrián E. Soto, Xochitl Maya, Ángel A. Vázquez, Juan G. Avalos, Giovanni Sánchez, Juan C. Sánchez  
SEPI ESIME Culhuacan Instituto Politécnico Nacional  
Ciudad de México, México  
{asotog1501,xmayar1300,avazquezp1301}@alumno.ipn.mx  
{javaloso,gsanchezriv,jcsanche}@ipn.mx

## Resumen

La mayoría de los sistemas de CAR prácticos están expuestos a entornos impulsivos lo que provoca inestabilidades al sistema en caso de que este no cuente con un algoritmo robusto. Recientemente, se han desarrollado diversas propuestas para mitigar este problema, por ejemplo, el algoritmo GMCC, MVC, entre otros. Sin embargo, dichos algoritmos solo se pueden configurar para obtener una alta velocidad de convergencia o un bajo nivel de desajuste. En este trabajo se propone un algoritmo de conmutación empleando dos algoritmos MVC con diferentes configuraciones. Los resultados obtenidos demuestran que la propuesta es robusta ante señales impulsivas mientras mantiene una alta velocidad de convergencia, un bajo nivel de desajuste y un bajo costo computacional.

## Palabras Clave

Control activo de ruido, ruido impulsivo, algoritmo MVC, algoritmo de conmutación, filtrado adaptativo.

# Estimación de cobertura de cápsulas en bebedero mediante procesamiento de imágenes embebido

Ezequiel Carbajo  
Depto. de Computación y Sistemas UNICEN  
Tandil, Argentina  
ecarbajo@alumnos.exa.unicen.edu.ar

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

Silvina Fernández, Federica Sagües, Milagros Junco, Inés Guerrero, Sara Zegbi y Carlos Saumell  
Centro de Investigación Veterinaria de Tandil CONICET-UNCPBA-CICPBA  
Tandil, Argentina  
{sfernand,federica,mjunco,inesg,szegbi,saumell}@vet.unicen.edu.ar

## Resumen

La digitalización y automatización en el sector agropecuario permite mejorar el proceso de producción con aumento de rindes. En particular, el suministro de medicamentos o tratamientos complementarios en animales suele ser una tarea demandante para los operarios de campos. En este trabajo se presenta un algoritmo de procesamiento de imágenes embebido para el monitoreo del nivel de cápsulas en bebederos, de un novedoso tratamiento antiparasitario. El algoritmo desarrollado fue ejecutado sobre una Raspberry Pi 3B y evaluado en un entorno real. Los resultados demostraron que la salida del algoritmo es viable y puede ser utilizada tanto para la notificación de operarios, como también entrada a un dosificador.

## Palabras Clave

Procesamiento de imágenes, sistema embebido, control parasitario, agroindustria.

## I. AGRADECIMIENTOS

Este trabajo fue financiado por el PICT 2021 I N°00142 “Encapsulación y administración de flagrans como agente de control biológico de nematodos gastrointestinales”

Categoría

**ARTÍCULO**

Protocolos y Comunicaciones

Área Temática

# Estimador de ruido de Middleton clase A: implementación sobre FPGA

L. Rabioglio, C. Cebedio, L. Arnone, L. De Micco y J. Castiñeira Moreira  
ICYTE, Departamento de Electrónica y Computación, Facultad de Ingeniería - UNMDP  
Mar del Plata, 7600, Argentina  
{lucas.rabioglio, celestecebedio, learn, ldemicco, casti}@fi.mdp.edu.ar

## Resumen

Este trabajo aborda la implementación en FPGA (Field-Programmable Gate Array), de un estimador simple de ruido de Middleton de Clase A existente. Al algoritmo original del estimador, se le realizan optimizaciones, mediante la aproximación por medianas, para obtener una arquitectura más simple y rápida. Se presenta su arquitectura y se comparan prestaciones. Los resultados obtenidos indican que la optimización del algoritmo es conveniente y se logran tiempos de ejecución en hardware muy superiores a la implementación en software.

## Palabras Clave

Arquitecturas VLSI, estimador, FPGA, Radio Cognitiva, ruido de Middleton de Clase A.



# Cripto-codificación en Turbo código no sistemático mediante *Interleaver* dinámico para estandar 5G

Raúl Eduardo Lopresti, Maximiliano Antonelli, Jorge Castiñeira Moreira y 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)*

## Resumen

El enfoque de las nuevas normas de comunicación se centra tanto en la seguridad de la transmisión como en una baja probabilidad de error. Estas características suelen ser excluyentes y requieren múltiples etapas de procesamiento de la información para lograr ambos objetivos. Sin embargo, los códigos Turbo permiten aumentar la seguridad de la transmisión sin afectar la probabilidad de error al variar temporalmente el bloque *Interleaver*. Este estudio presenta un sistema de cripto-codificación que se evalúa en términos de probabilidad de error, encriptación y complejidad del sistema final.

## Palabras Clave

Códigos Turbo, Encriptamiento, *Interleaver*, No sistemático, Seguridad.

## I. AGRADECIMIENTOS

Este trabajo fue parcialmente financiado por el CONICET (Consejo Nacional de Ciencia y Tecnología), la Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT, PICT2019-03024).

Categoría

# FORO TECNOLÓGICO

Bioingeniería

Área Temática

# HEROxS: Heart Rate and Oxygen Saturation Meter

Rosina D' Eboli, Josefina Schimtd, Rodrigo Garcia Ordeig, and Julián Oreggioni,  
 Universidad de la República, Montevideo, Uruguay  
 Email: {rosina.d,maria.josefina.schimidt,rodrigo.garcia.ordeig,juliano}@fing.edu.uy

**Abstract**—Heart rate (HR) and oxygen saturation (SpO2) are two key indicators for evaluating the cardiovascular condition of a person and detecting potential health issues. Hence, there is a need for portable solutions that can provide daily and accurate monitoring of these signals, which later can be evaluated by a healthcare professional. This work presents the design of a device capable of HR and SpO2 monitoring, consisting of a 0.91-inch OLED display that shows real-time measured values, a MSP430G2553 microcontroller and a SEN-15219 sensor board. Experimental results show that the accuracy is very good (less than 2 % of absolute error) and the autonomy can be up to 15 days.

**Index Terms**—photoplethysmography, PPG, vital signs monitoring, MAX30101, MAX32664

## I. INTRODUCTION

In Uruguay, the three leading causes of mortality between 2015 and 2020 were cardiovascular diseases, cancer, and respiratory diseases [1]. Generating information about an individual's health status during their daily activities, which can eventually be consulted by a healthcare professional, is crucial for the prevention, early diagnosis, and control of these conditions. Monitoring oxygen saturation (SpO2) and heart rate (HR) is particularly relevant as they provide a first approach to assessing a person's health status [2]. A commonly used technique to acquire these signals is Photoplethysmography (PPG). PPG use photo-diodes that capture light reflected from blood vessels (typically on the wrist or finger) after being illuminated by a LED [3].

Currently, there are several devices available in the market that utilize PPG, such as smartwatches like Apple Watch [4] or Samsung Galaxy Watch [5], as well pulse oximeter devices [6]. In our lab, we are developing our own device, and this work reports our second generation device [2], [7]. Then, this work presents the proof of concept of a portable device for HR and SpO2 acquisition and recording. The system offers two operating modes, an alarm to notify the detection of abnormal values, and the capability to transmit the recorded data to a personal computer (PC). Furthermore, the system features multi-language support for the device display and the PC.

## II. PROPOSED SOLUTION

Fig. 1 shows the block diagram of the proposed solution. The user interacts with the device by pressing a button to start taking measurements, and through a PC terminal, to add/modify user data, to set the operating mode, and to download the recorded data.

The device is composed of the MSP430G2553 microcontroller, the SEN-15219 board, the OLED (Organic light-emitting diode) display with SSD1306 controller, and a button.

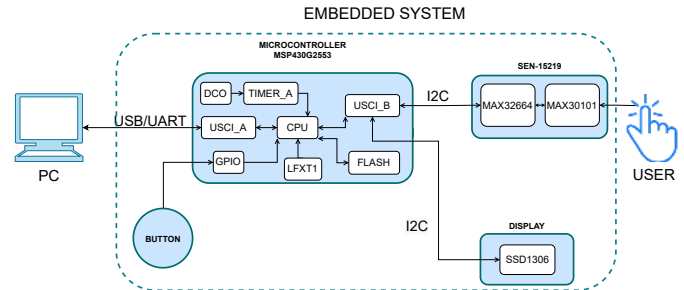


Fig. 1. Block diagram of the proposed system.

The MSP430G2553 is a low-power 16-bit microcontroller from Texas Instruments. It operates at speeds up to 16 MHz and has 16 kB of Flash memory and 512 bytes of RAM. It has built-in peripherals such as I2C (Inter-integrated circuit) and UART (Universal asynchronous receiver/transmitter).

The SEN-15219 board utilizes the MAX30101 SpO2 and HR sensor, along with the MAX32664 sensor-hub. The MAX30101 integrates an infra-red LED, a red LED, a photodetector, and analog signal conditioning circuitry. The MAX32664 controls the measurement process, performs the calculation of SpO2 and HR from the PPG signal, and provides the I2C communication interface.

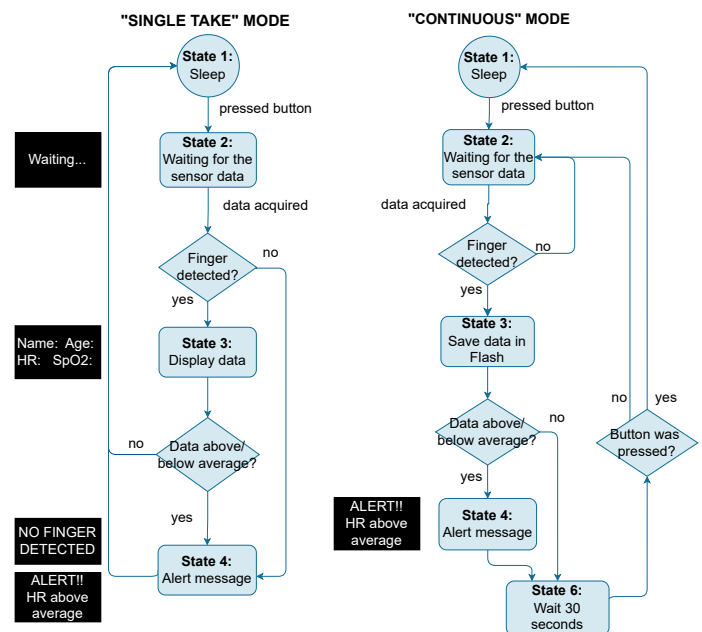


Fig. 2. Flow diagram of "single take" and "continuous" mode, some states are not shown for legibility. Display messages are depicted in black rectangles.

The device has two modes of operation (see Fig. 2). The “single take” mode is designed for occasional self-monitoring, while the “continuous” mode aims to monitor the user’s sleep cycle uninterrupted (approx. 8 hours). In both modes the device is in a low-power mode (sleep state) by default. In “single take” mode, when the user presses the button, the device takes a single measurement and shows the obtained data on the display, along with the user information. Then, the device returns to sleep. In “continuous” mode, the device takes periodic measurements and stores them in the flash memory until the button is pressed. Additionally, for both modes, an alarm message is presented on the display if the HR or SpO2 value is outside a pre-set range, or if the acquired data indicates that there was an error in the measurement (i.e. finger was not detected). For the “continuous” mode, an alarm was implemented to detect when the section of the flash memory reserved for data storage becomes full.

The embedded software architecture is Round-Robin with interruptions [8], where the task of the main program is to check the flags associated with the button’s and UART reception’s ISRs (Interrupt Service Routines) in a loop. When the associated flag is turned “on”, the corresponding handler is executed. In other case, the microcontroller is on low-power mode LPM3 until is woken up by an interruption.

The microcontroller’s internal peripheral data is handled by its ISRs. The USCI\_A (Universal serial communication interface) and USCI\_B peripherals were configured for UART and I2C protocols, respectively. UART was used for communication with the PC, while I2C was used for communication between the master (microcontroller) and the slaves (display and SEN-15219 board). Additionally, the Timer\_A peripheral was used to implement the periodic sampling described in the “continuous mode” behavior. During this time, the device enters in low-power mode LPM0 to reduce power consumption until the next timer interruption.

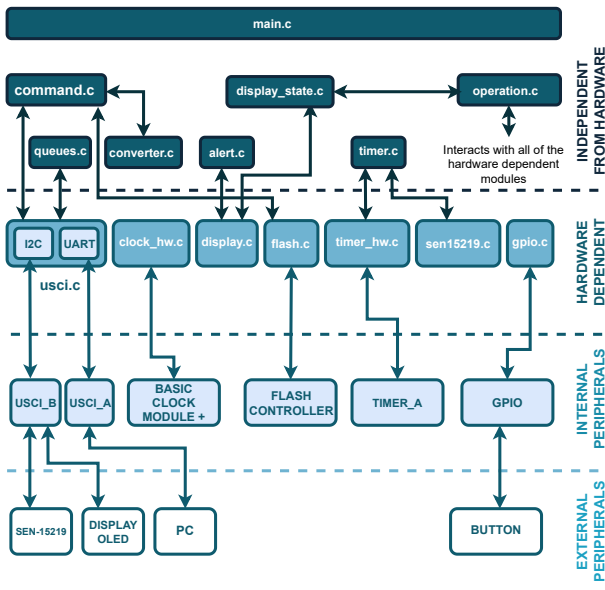


Fig. 3. Software modules.

Fig. 3 illustrates the interdependence of the software modules and their relationship with the hardware components. The modules include data acquisition and processing, communication, and user interface, among others, and work collaboratively to ensure the device’s proper operation.

### III. EXPERIMENTAL RESULTS

#### A. Current consumption and autonomy

Fig. 4 shows the experimental set-up, where the Power Profiler Kit 2 by Nordic was used to measure the current.



Fig. 4. Hardware experimental set-up (left) and the manufactured prototype (right)

Table 1 shows the current consumption of each component of the system, categorized by different modes of operation. It was observed that the SEN-15219 board exhibited high current consumption in sleep mode, accounting for approximately 98.9 % of the total sleep current consumption. Additionally, significant current consumption was also observed in single take and continuous operating modes. Optimization strategies should be implemented in future work to reduce the current consumption of the SEN-15219 board in sleep mode and improve the overall energy efficiency of the system. This work focused on reducing current consumption at the microcontroller level, achieving an operating consumption of 360  $\mu\text{A}$  and a sleep one of 88  $\mu\text{A}$ .

TABLE I  
CURRENT CONSUMPTION OF THE DIFFERENT OPERATING MODES

	Sleep	Single take	Continuous
Total	8.44 mA	18.12 mA	16.00 mA
Display	4 $\mu\text{A}$	2.71 mA	4 $\mu\text{A}$
Microcontroller	88 $\mu\text{A}$	360 $\mu\text{A}$	360 $\mu\text{A}$
SEN-15219	8.35 mA	15.50 mA	15.61 mA

Next, we estimate the device autonomy for different battery types, considering both modes of operation. In the single take mode, we considered that the user performs two measurements per day, while in the continuous mode, current was measured with the system at the steady state condition (taking measurements every 30 seconds). Table II shows the run time of different battery types for both modes of operation. It can be observed that in the continuous mode, the device is capable of lasting more than one complete sleep cycle.

TABLE II  
EMBEDDED SYSTEM AUTONOMY FOR THE DIFFERENT OPERATING MODES

Mode	CR2032	2xAAA	2xAA
Single take	1d 4h 8m	11d 12h 38m	29d 22h 33m
Continuous	14h 41m	6d 22m	15d 15h

### B. Memory

Memory usage was measured using *Code Composer Studio* from Texas Instruments. Table III shows these results.

TABLE III  
MEMORY USAGE

	Flash (kB)	Flash (%)	RAM (B)	RAM (%)
Available	12.51	100	512	100
Used	6.40	51	486	94

### C. Vital signs measurement

One hundred samples were taken to assess device accuracy. Measurements from a reference device were simultaneously obtained, using an ArgomTech smartwatch, model SkeiWatch S50 [9]. The test was conducted on a 21 year old healthy female, in an indoor space, and both our device and the reference device were placed in the right hand. Table IV outlines the statistical behavior of the acquired data, comparing results between HR and SpO2 measurements. The obtained values are consistent.

TABLE IV  
STATISTICAL ERROR ANALYSIS

	Heart Rate (bpm)		SpO2 (%)	
	Abs. error	Rel. error	Abs. error	Rel. error
Mean	1.61	2.07%	1.48	1.50%
Std. deviation	1.20	-	0.94	-

Figs. 5 and 6 show the histograms which describe the absolute error behavior for HR an SpO2 measurements respectively.

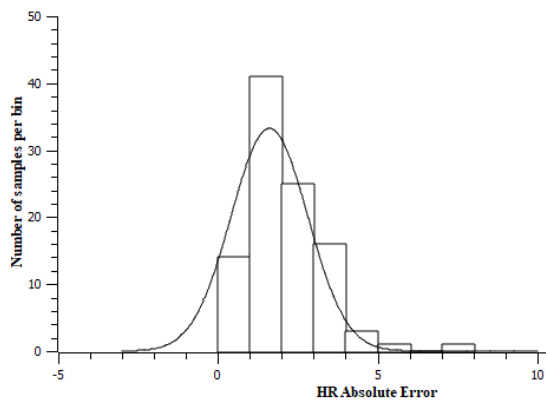


Fig. 5. HR absolute error.

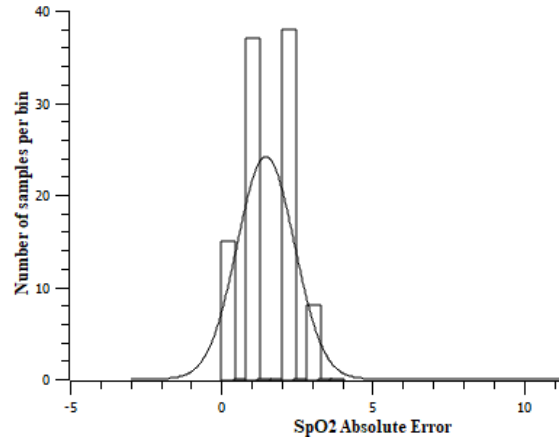


Fig. 6. SpO2 absolute error.

## IV. CONCLUSIONS

We successfully developed a device capable of monitoring vital signs. The low-power techniques utilized in the microcontroller contributed to achieve the needed autonomy. However, optimization strategies need to be implemented to minimize the current consumption of the SEN-15219 board in sleep mode to further improve the device's energy efficiency. The system offers two operating modes, an alarm to notify the detection of abnormal values, and the capability to transmit the recorded data to a PC.

## ACKNOWLEDGMENT

This work was partially funded by CSIC (Comisión Sectorial de Investigación Científica, Uruguay), and Erasmus+ Project NEON, 618942-EPP-1-2020-1-AT-EPPKA2-CBHE-JP. The experimental procedures involving human subjects described in this paper were approved by the Ethics Committee, School of Psychology, Universidad de la República, Uruguay.

## REFERENCES

- [1] I. León, A. Misa, and O. Gianneo. (2020, Setiembre) Vigilancia de la Mortalidad por todas las causas - Enero a julio 2015-2020 Informe Preliminar. Ministerio de Salud Pública, Uruguay. [Online]. Available: <https://www.gub.uy/ministerio-salud-publica/comunicacion/noticias/vigilancia-mortalidad-todas-causas>
- [2] L. Martínez Hornak, I. Morales, A. Solari, and J. Oreggioni, "Wearable device prototype for vital signs monitoring," in *XII Congreso Argentino de Sistemas Embebidos, UNLP, La Plata, Buenos Aires, Argentina*. CASE, 2022, pp. 70–72.
- [3] J. G. Webster, *Design of pulse oximeters*. CRC Press, 1997.
- [4] Apple. (2022, 06) Apple Watch Series 7. [Online]. Available: <https://www.apple.com/apple-watch-series-7/>
- [5] Samsung. (2022, 06) Galaxy Watch3. [Online]. Available: <https://www.samsung.com/latin/watches/galaxy-watch/galaxy-watch3-45mm-mystic-silver-sm-r840nzsalta/>
- [6] Pulse oximeter. SantaMedical. [Online]. Available: <https://www.amazon.com/Fingertip-Oximeter-Saturation-Measurements-Batteries/dp/B086KZ8JVH>
- [7] I. Morales, L. Martínez Hornak, A. Solari, and J. Oreggioni, "Respiratory rate estimation on embedded system," in *XII Congreso Argentino de Sistemas Embebidos, UNLP, La Plata, Buenos Aires, Argentina*. CASE, 2022, pp. 29–29.
- [8] D. E. Simon, "Survey of software architectures," in *An Embedded Software Primer*. Addison-Wesley Professional, 1999, pp. 115–136.
- [9] SkeiWatch S50. ArgomTech. [Online]. Available: <https://www.argomtech.com/products/skeiwatch-s50>

# Prototipo de sistema para el análisis viscoelástico de lágrimas

Gabriel Muñoz<sup>1</sup>, Martín Millicovsky<sup>1,2</sup>, Juan Cerrudo<sup>1</sup>, Albano Peñalva<sup>1</sup>, Juan Reta<sup>1</sup> y Martín Zalazar<sup>1,2</sup>

<sup>1</sup>Facultad de Ingeniería, Universidad Nacional de Entre Ríos (FI-UNER)

<sup>2</sup>Instituto de Investigación y Desarrollo en Bioingeniería y Bioinformática (IBB), FI-UNER-CONICET  
Entre Ríos, Argentina

gabrielguillermo2015@gmail.com, martin.millicovsky@uner.edu.ar, juan.cerrudo@uner.edu.ar, albano.penalva@uner.edu.ar,  
juan.reta@uner.edu.ar, martin.zalazar@uner.edu.ar

**Resumen**— Las lágrimas han despertado interés como fluido biológico mínimamente invasivo a partir del cual evaluar ciertas enfermedades. En este trabajo se presenta los resultados del prototipo de un sistema fabricado para aplicaciones biomédicas enfocado en la identificación de la densidad y viscosidad de lágrimas utilizando la tecnología de microbalanza de cristal de cuarzo (QCM). Un controlador proporcional-integral-derivativo (PID) fue implementado en una Raspberry Pi 4 para el control de temperatura del sistema. Además, un segundo prototipo es presentado mostrando la evolución del primer sistema. La caracterización se realizó con diferentes tipos de lágrimas artificiales.

**Palabras claves**—Lágrimas, Raspberry, Bajo costo, Microbalanza de cristal de cuarzo, Control de temperatura.

## I. INTRODUCCIÓN

La microbalanza de cristal de cuarzo (QCM) es un sensor piezoeléctrico, altamente sensible a pequeñas variaciones del líquido depositado sobre la misma. Los cambios producidos en la frecuencia de resonancia del cristal brindan información física de la sustancia depositada, características físicas como la masa y la viscosidad [1,2]. La ventaja respecto a otros instrumentos de medición es que requiere de un volumen de muestra bajo para su funcionamiento [3], consecuentemente, funcionaría para caracterizar sustancias biológicas con un bajo volumen. Por lo tanto, sería aplicable utilizar lágrimas como sustancia depositada sobre la QCM, las mismas podrían estar alteradas en sus propiedades físicas en diversas enfermedades generales lo que servirían de fuente de información clínica [4]. No obstante, aunque los sistemas QCM personalizados siguen siendo populares para adaptarse a experimentos específicos, y los sistemas comerciales apuntan a maximizar rendimientos eléctricos y térmicos con componentes de alta calidad, caros y voluminosos, no existen específicamente para el estudio de lágrimas [5].

Investigadores de Novaetech S.R.L. crearon el proyecto Open QCM, que consiste en desarrollar las herramientas necesarias para realizar trabajos de investigación con QCM. Basado en esto, un prototipo diseñado para la aplicación experimental del estudio de lágrimas fue realizado previamente por los autores de este trabajo, considerando además que los cambios de temperatura afectan la estabilidad de la frecuencia de oscilación de la QCM, que se suma a la dependencia de la viscosidad y la densidad propia del líquido a la temperatura [6]. La tecnología fue de código abierto y los materiales fueron seleccionados para que trabaje en condiciones de temperatura estable y controlada, buscando reducir costos, el mismo fue probado solo con sustancias simples [7]. El diseño se implementa con un controlador

proporcional-integral-derivativo (PID) en una Raspberry Pi 4 utilizando una celda Peltier, además el sistema integra un módulo de sensado que facilita la conexión de un instrumento de medición y la QCM, ver Fig. 1.

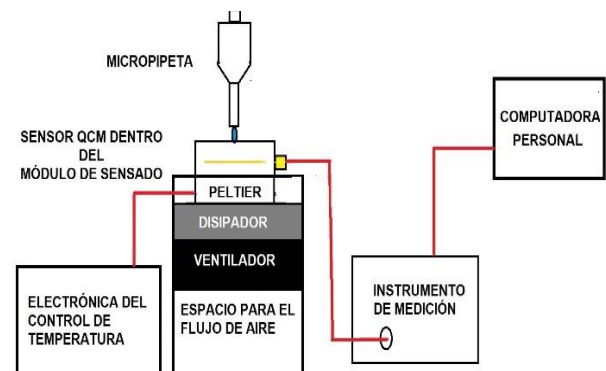


Figura 1: Diagrama en bloques del sistema QCM con control de temperatura.

En este trabajo se va a utilizar el prototipo previamente desarrollado de bajo costo y específicamente pensado para el estudio de lágrimas con el fin de caracterizar las propiedades viscoelásticas de lágrimas artificiales midiendo los cambios de frecuencia del sensor inducidos por dos volúmenes diferentes del líquido de muestra [8]. Si bien no es una sustancia natural nos permite sentar las bases para en un futuro trabajar con lágrimas humanas. Adicionalmente, se presenta un prototipo que se está desarrollando en la actualidad que reduce aún más tamaños y costos.

## II. MATERIALES Y MÉTODOS

### A. Hardware

El módulo de detección se diseñó en Autodesk Fusion 360 y se imprimió en tecnología MSLA con un equipo CreaLity LD-006 utilizando resina fotosensible. El módulo tiene un ancho de 44 mm, una longitud de 40 mm, una altura de 15 mm y un peso de 21,5 gr. Se utilizó el software de código abierto KiCad para diseñar una PCB que conecte el instrumento para medir frecuencias VNA (DG8SAQ 3, SDR-Kits) con la QCM. La conexión eléctrica se realizó a través de un conector SMA (Amphenol, bajo ruido) y pines pogo chapados en oro. El cristal piezoeléctrico adquirido de Novaetech SRL es de corte AT, tiene una frecuencia de resonancia cercana a 10 MHz. La Fig. 2 muestra la QCM utilizada en este sistema. El electrodo de oro tiene una capa de adhesión de titanio de un solo lado y los diámetros de los electrodos delanteros y trasero son de 11,5 mm y 6 mm, respectivamente. En la Fig. 3 puede verse el módulo sin el cristal y una representación con el agregado de lágrima, en la parte superior del módulo se proporciona un espacio donde se introduce el líquido. La colocación del cristal es rápida y sencilla, estando en contacto con una junta tórica de silicona que estabiliza la presión y evita el derrame.



Figura 2: A la izquierda, parte inferior del cristal. Derecha, parte superior de la QCM donde se deposita la muestra.

El control de temperatura se basó en una celda Peltier (TEC1-12706), la cual es alimentada con 12V y consume un máximo de 6A. Un disipador de aluminio, junto con grasa térmica y un ventilador, favorecen la transferencia de calor entre ambos lados de la Peltier. El control térmico tiene un rango de trabajo nominal de 18°C a 30°C, calefacción y refrigeración bidireccional. Se seleccionó una Raspberry Pi 4 como dispositivo de control, por su flexibilidad, reducido tamaño y bajo costo. La adquisición de temperatura se realiza con un termistor (NTC 3950) que proporciona un tiempo de respuesta rápido y precisión en las mediciones. El programa PID se implementó con Python, un lenguaje de código abierto de uso común. La corriente de la Peltier se controló a través de una señal de modulación de ancho de pulso (PWM) utilizando un controlador de motor paso a paso de 2 canales H-Bridge (VNH2SP30 30A). Se eligió una fuente de alimentación conmutada (12V-10A) para alimentar el controlador, el ventilador y la celda Peltier, mientras que la Raspberry lo hace desde un puerto USB de la PC. El sistema completo puede verse en la Fig. 4.

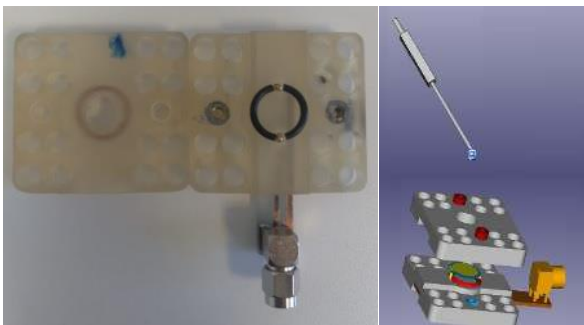


Figura 3: A la izquierda, parte superior e inferior del módulo de sensado. En la derecha, representación de la colocación de líquido, en una vista en despiece del módulo.

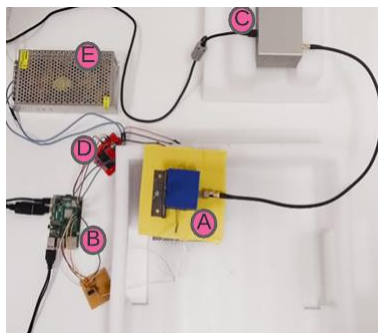


Figura 4: Sistema completo de medición, el sistema de temperatura sensa próximo al cristal mediante una NTC. A-Modulo de detección y protector del módulo. B- Raspberry Pi 4. C- VNA ((DG8SAQ 3). D-Controlador de Motor. E- Fuente.

Actualmente se desarrolla una nueva versión del prototipo que no ha llegado a probarse, entre los cambios importantes se

destaca el paso del VNA (DG8SAQ 3, SDR-Kits) a un NanoVna económico y de la Raspberry Pi 4 por una Raspberry Pi Pico, además se agregó un segundo ventilador para mejorar las transferencias térmicas. Un Hub USB interno permite que el dispositivo se conecte mediante un único cable USB a la PC para el control simultáneo del VNA y del control de temperatura que corre sobre la Raspberry Pi Pico. Todo el sistema se incorpora dentro de un gabinete plástico comercial fabricado por termoformado, ver Fig. 5 y 6.

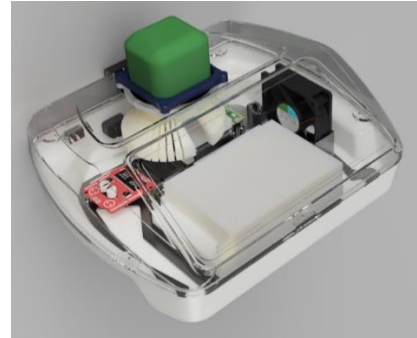


Figura 5: Renderizado del nuevo prototipo, realizado en Fusión 360.

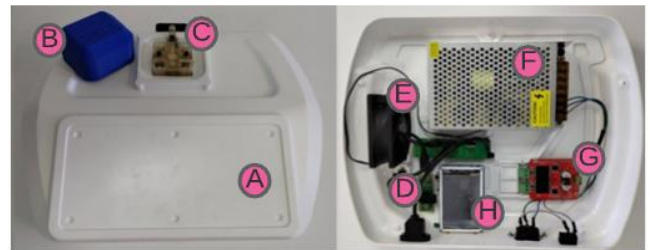


Figura 6: Nuevo sistema, cambio de Raspberry y VNA con respecto al prototipo original. A. Gabinete plástico. B.Protector módulo. C-Módulo de detección. D-Raspberry Pi Pico. E-Ventilador. F-Fuente. G-Controlador de motor. H-NanoVna.

### B. Método

Para obtener por separado la densidad y la viscosidad del líquido, simplificando el proceso de medición y mejorando la precisión. Se mide la frecuencia de resonancia de referencia ( $f_0$ ) de una QCM en aire. A continuación, se deposita un volumen de líquido de muestra  $VL_1$  sobre la superficie del sensor, lo que provocará una frecuencia de resonancia  $f_1$ . El primer cambio de frecuencia es  $\Delta f_1 = f_1 - f_0$ . Si se agrega otra muestra de líquido encima del sensor ya cargado, el volumen total será  $VL_2$  y el segundo cambio de frecuencia  $\Delta f_2 = f_2 - f_0$ . A medida que el volumen varía de  $VL_1$  a  $VL_2$ , también lo hace la frecuencia de resonancia  $\Delta f_2 - \Delta f_1$ . La densidad del líquido de la muestra  $\rho$ , se calcula de la siguiente forma [8, ec. (5)]:

$$\rho = \frac{(\Delta f_2 - \Delta f_1)}{K_{pf} \cdot C_{pf} \cdot VL_2 \cdot f_0} \quad (1)$$

$C_{pf}$ , es una constante de presión-frecuencia,  $K_{pf}$  es el coeficiente de sensibilidad a la presión del sensor QCM, y se obtiene de la siguiente manera [8]:

$$K_{pf} = \frac{f_0 \cdot K_f}{n \cdot D} \quad (2)$$

$n$  es el número armónico del sensor QCM,  $D$  es el diámetro del electrodo del sensor QCM, y  $K_f$  es la constante de fuerza-frecuencia del sensor QCM.

Por otra parte, la viscosidad del líquido  $\eta$  se calcula con la siguiente ecuación [8, ec. (6)]:

$$\eta = \frac{K_{pf} \cdot C_{pf}}{(\Delta f_2 - \Delta f_1) \cdot VL_2 \cdot f_0} \cdot \left( \frac{\Delta f_1 \cdot VL_2 - VL_1 \cdot (\Delta f_2 - \Delta f_1)}{K_{tf} \cdot C_{tf}} \right)^2 \quad (3)$$

Donde  $K_{if}$  es el coeficiente de sensibilidad a la tensión del sensor QCM y  $C_{if}$  es el coeficiente de frecuencia de tensión del sensor QCM.

### III. RESULTADOS

La caracterización se realizó calculando la viscosidad y densidad de tres diferentes marcas de lágrimas artificiales Glaucotensil® (GL), Systane® (SY) y Tratavant® (TV). Midiendo la diferencia de frecuencia entre dos volúmenes distintos a 26 °C y utilizando la (1) y (2). Se utilizaron 5 muestras por marca, cada vez que se cambiaba la muestra, el cristal se limpiaba con detergente SDS al 2% durante 3 min, se enjuagaba con agua destilada y se secaba con gas nitrógeno puro.

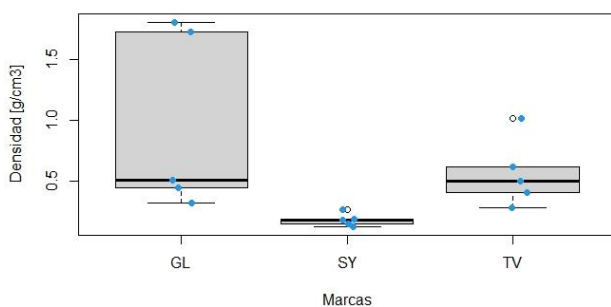


Figura 7: Variación de densidad para distintas marcas de lágrimas artificiales.

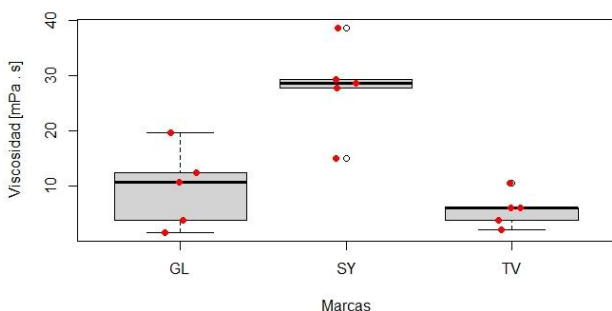


Figura 8: Variación de viscosidad para distintas marcas de lágrimas artificiales.

Las mediciones en viscosidad se encuentran dentro del rango esperable para lágrimas artificiales según la literatura [9], donde los valores de viscosidad van de 0.55 mPa.s a 35 mPa.s, en cambio se esperaría que los valores en densidad no sean menores que para el agua destilada que es de 1 g/cm<sup>3</sup>.

### IV. CONCLUSIÓN

Se presentó un prototipo QCM de código abierto y bajo costo para una aplicación oftalmológica. Está compuesto por piezas simples y de alta calidad, permiten al usuario configurar el dispositivo y su manejo de manera sencilla. Por

otra parte, también se muestra una versión actualizada con el objetivo de mejorar el equipo existente.

Se analizaron diferentes marcas comerciales de lágrimas artificiales como procedimiento de caracterización, mostrando una discriminación de densidad y viscosidad entre las distintas lágrimas. Los resultados fueron más variables de lo esperado, esto puede ser por las impurezas que quedan aun limpiando el cristal. Si bien los valores de viscosidad son acordes a la bibliografía, no los de densidad para dos de las marcas. Los próximos pasos implicarán ajustes en el método de caracterización y utilización de lágrimas naturales.

### AGRADECIMIENTOS

Este trabajo fue apoyado por la Agencia Nacional de Promoción Científica y Tecnológica, PICT StartUp 4655: Biosensor para la evaluación cualitativa y cuantitativa de la lágrima.

Queremos agradecer al Laboratorio de Prototipos Electrónicos & 3D de la Universidad Nacional de Entre Ríos por brindarnos las facilidades para desarrollar y evaluar nuestro prototipo.

Agradecemos al Doctor Rodrigo Torres, por proveernos de las lágrimas artificiales y por su constante asesoría.

### REFERENCIAS

- [1] A. Arnau, "A review of interface electronic systems for AT-cut quartz crystal microbalance applications in liquids", *Sensores*, vol. 8, pp. 370-411, Enero 2008. doi: 10.3390/s8010370.
- [2] K. K. Kanazawa, y J. G. Gordon, "Frequency of a quartz microbalance in contact with liquid", *Analytical Chemistry*, vol.57, pp. 1770-1771, Julio 1984. doi: 10.1021/ac00285a062.
- [3] L. Cervera-Chiner, M. Juan-Borrás, A. Arnau, I. Escriche, A. Montoya, y Y. Jiménez, "High Fundamental Frequency Quartz Crystal Microbalance (HF-QCM) immunosensor for pesticide detection in honey", *Food Control*, vol.92, pp. 1-6, Octubre 2018. doi:10.1016/j.foodcont.2018.04.026.
- [4] M. Dor, S. Eperon, P. Lalive, Y. Guex-Crosier, M. Hamedani, C. Salvisberg y N. Turk, "Investigation of global protein content from healthy human tears", *Experimental Eye Research*, vol. 17, pp. 64-74, Febrero 2019. doi: 10.1016/j.exer.2018.10.006.
- [5] S. Turkdogan, "Design and implementation of a cost effective quartz crystal microbalance system for monitoring small changes on any surface", *Balkan J. Electric. Comput. Eng.*, vol.7, pp. 213-217, 2019. doi:10.17694/bajece.530796.
- [6] A. Cao-Paz, L. Rodriguez-Pardo, y J. Fariña, "Temperature compensation of QCM sensors in liquid media", *Sensors and Actuators B: Chemical*, vol. 193, pp. 78-81, Marzo 2014. doi:10.1016/j.snb.2013.11.044.
- [7] G. G. Muñoz, M. J. Millicovsky, J.M. Reta, J. I. Cerrudo, A. Peñalva, M. Matchey, R. Torres y M. Zalazar, "Quartz crystal Microbalance with dissipation monitoring for biomedical applications: Open source and low cost prototype with active temperature control", *Hardware X*, vol.14, Marzo 2023. doi: 10.1016/j.ohx.2023.e00416.
- [8] F. Tan, D. Y. Qiu, L. P. Guo, P. Ye, H. Zeng, J. Jiang, Y. Tang y Y.C. Zhang, "Separate density and viscosity measurements of unknown liquid using quartz crystal microbalance", *AIP Advances*, vol. 6, Septiembre 2016. doi: 10.1063/1.4963298.
- [9] F. Che Arif, & M. R. Hilmi, K. Kamal y M. Ithnin, "Evaluation of 18 artificial tears based on viscosity and pH", *Malaysian Journal of Ophthalmology*, vol. 2, pp. 96-111, Julio 2020. doi: 10.35119/mjjo.v2i2.109.



Categoría

# FORO TECNOLÓGICO

FPGAs, HDLs y ASICs

Área Temática

# Modulador QPSK con parámetros ajustables: Diseño e implementación

Mariano Morel y 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)*

Mar del Plata, Buenos Aires, Argentina

marianomorel@fi.mdp.edu.ar

**Resumen**—En este proyecto se creó un generador de señal modulada en QPSK, utilizando una FPGA (Field Programmable Gate Array) y un DAC (convertor digital a analógico). Este sistema tiene la capacidad de recibir datos de una fuente externa o generarlos internamente mediante un generador de números pseudo aleatorios (PRNG). Una de las ventajas de este diseño es que permite ajustar la cantidad de ciclos de portadora por dato y la frecuencia de la portadora en un amplio rango, así como controlar la potencia de salida y la fase inicial de la modulación. Además, se logró una utilización eficiente de los recursos de la FPGA gracias a la implementación de la portadora mediante el almacenamiento de un cuarto de ciclo y su correspondiente lógica de lectura.

**Palabras clave**—Modulación de fase, FPGA, DAC, QPSK, PRNG.

## I. INTRODUCCIÓN

Este proyecto forma parte de una iniciativa más amplia destinada a desarrollar sensores espectrales para la tecnología de Radio Cognitiva [1]. Estos sensores tienen la tarea de detectar señales moduladas en entornos con muy baja relación señal-ruido. Para cumplir con este objetivo, es crucial contar con un generador que pueda producir señales moduladas y que permita ajustar diversos parámetros, como la potencia, la frecuencia de la portadora y la velocidad de los datos.

En la literatura existen algunos trabajos que proponen diseños basados en FPGA para moduladores QPSK (Quadrature Phase-Shift Keying). Sin embargo, en estos trabajos no se han encontrado implementaciones que ofrezcan flexibilidad de variación simultánea en los parámetros de la transmisión, tales como la capacidad de variar la frecuencia de la portadora, la cantidad de ciclos de portadora por dato, la frecuencia de los datos, la fase inicial y la potencia de salida en tiempo real.

En [2], se combina dos arquitecturas para el diseño del modulador digital QPSK, una de ellas utilizando la multiplicación BOOTH y para la segunda arquitectura, además, se recopilan cuatro datos de entrada combinatoriales diferentes para diferentes fases de QPSK, que se los almacena en cuatro bloques ROM diferentes. La elección de cada fase se determina mediante una secuencia de entrada de datos. Para la representación de un ciclo de portadora se almacenan 50 muestras. La cantidad de ciclos por dato es fija en uno, con una portadora de 1 MHz.

En [3], los autores proponen una modulación QPSK por desplazamiento de fase de cuatro estados, generados mediante una secuencia de dos bits. La señal portadora está compuesta

por 16 valores diferentes guardados en una memoria ROM. El modulador y el DAC se encuentran embebidos en una placa de desarrollo Spartan 3E Starter Kit board.

En [4], el diseño en VHDL fue realizado mediante la herramienta Xilinx ISE 14.1. El autor propone un modulador QPSK sin la necesidad de multiplicar señales para formar la señal modulada, sino que el sistema permita la elección de cuatro señales con fases diferentes guardadas en una memoria ROM. La señal portadora trabaja a frecuencia de 322 MHz.

En este trabajo se desarrolló un diseño que permite alcanzar la flexibilidad requerida. Consiste en almacenar una gran cantidad de muestras de portadora en memoria y seleccionar, a través de entradas, las muestras que se desean leer y entregarlas al convertor digital-analógico (DAC), que las recibe en paralelo. De esta manera, el DAC se encarga de generar la señal analógica de salida. La arquitectura mejorada de esta implementación aprovecha el almacenamiento interno de un cuarto de señal sinusoidal para generar la señal de portadora, lo que reduce significativamente la cantidad de recursos de memoria necesarios.

## II. CIRCUITO IMPLEMENTADO

El circuito implementado, cuyo código se encuentra en [5], consta de dos bloques principales. El primero se ejecuta en una placa externa que incluye un DAC (convertor digital-analógico) y un circuito de adecuación de señal, mientras que el segundo se diseña en la FPGA utilizando el lenguaje de descripción de hardware VHDL (Very High-Speed Integrated Circuit Hardware Description Language).

En la figura 1 se muestra el esquema del sistema implementado, que incluye los dos bloques principales mencionados anteriormente y las entradas de selección que permiten ajustar los parámetros de la modulación en tiempo real. Cada dato que se transmite tiene una duración de  $T_b$ , mientras que el período de la portadora es de  $T_c$ , por lo que se transmiten  $N_c$  períodos de portadora por dato ( $T_b = N_c T_c$ ).

La FPGA proporciona al DAC las muestras de la señal modulada a una frecuencia  $f_s$ , y dado que la cantidad de muestras de un ciclo de portadora es de  $4.N$ , el período de la portadora resulta  $T_c = 4.N T_s$ . De esta manera, el tiempo de dato a transmitir es  $T_b = N_c 4.N T_s$ , siendo  $T_s = 1/f_s$ .

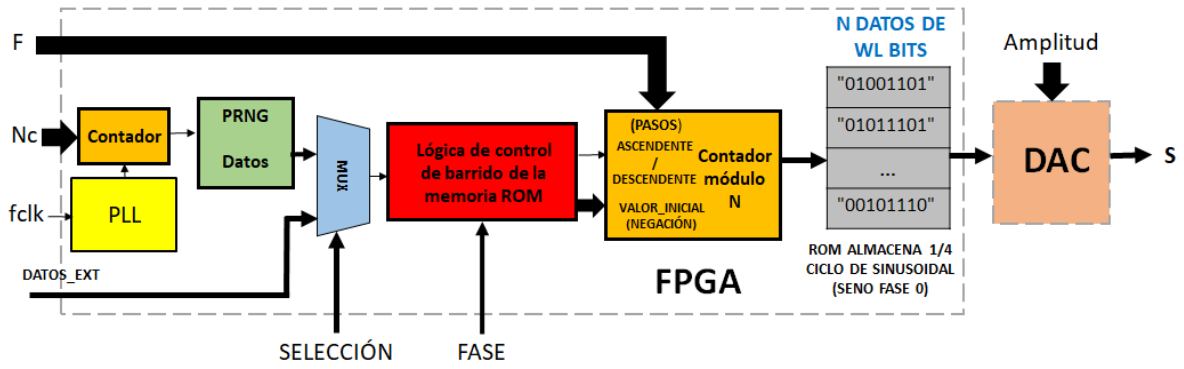


Figura 1. Esquema del circuito implementado. La parte digital implementada en la FPGA y el circuito DAC.

II-A. Circuito DAC

En [6], se describe con detalle, la placa desarrollada que contiene el DAC. Esta es la que recibe la señal digital de la FPGA y la convierte en una señal analógica QPSK.

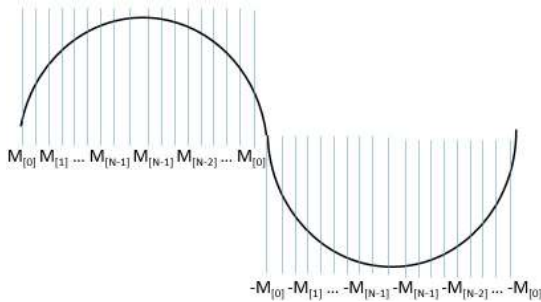


Figura 2. Lectura de memoria de tamaño N para la generación de un ciclo de portadora.

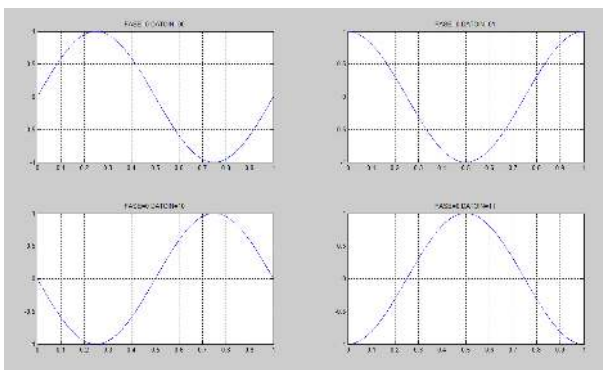


Figura 3. Para FASE=0, se simuló un período de señal sinusoidal para diferentes datos de entrada.

II-B. Diseño en la FPGA

En la figura 1 se presenta un esquema del circuito desarrollado en la FPGA. El sistema se basa en la lectura de una memoria que almacena un cuarto de ciclo de una señal sinusoidal. La lógica de control de barrido, establece el modo de operación del contador que recorre la memoria, el valor inicial y los pasos de incremento/decremento, así como también si se debe negar el valor de la memoria. La memoria se leerá según

los parámetros de entrada (Nc y F) y el dato a transmitir para que la señal sea entregada al DAC. Por ejemplo, cuando se requiere entregar un período senoidal a la mínima frecuencia de salida, se leerá la memoria en pasos de a 1 de la siguiente manera, siendo el valor inicial quien determina la fase inicial:

1. Cuenta ascendente de (valor\_inicial) a N-1.
2. Cuenta descendente de N-1 a (valor\_inicial).
3. Cuenta ascendente de (valor\_inicial) a N-1, con salida negada.
4. Cuenta descendente de N-1 a (valor\_inicial), con salida negada.

Al obtener 4N muestras de un ciclo de portadora a la frecuencia más baja posible ( $f_c = f_s / (4 \cdot N)$ ), se consigue una señal de mayor precisión. A diferencia de otros enfoques que emplean un número limitado de muestras por ciclo de portadora, este método utiliza  $N=256$  datos por cuarto de ciclo, lo que equivale a 1024 muestras por ciclo. Esto da lugar a una mayor adaptabilidad en cuanto a la variación de la frecuencia de portadora que se obtiene en la salida.

En la figura 2 se muestra cómo sería la lectura de la memoria de tamaño N para la generación de un ciclo de portadora completo. La entrada de SELECCIÓN determina si los datos modulantes son generados internamente mediante un PRNG, que en este caso se trata de un LFSR (Linear Feedback Shift Register) o se reciben de forma externa.

Este sistema permite seleccionar la posición de la constelación de la modulación QPSK en el plano de fase y cuadratura, es decir permite elegir la fase inicial del ciclo de portadora dependiendo de los datos de la señal modulante, tal como se indica en la Tabla I. La comprensión de esto, se basa en el caso de la figura 3 donde se selecciona FASE=0, dando cuatro diferentes señales de portadora, para diferentes datos de entrada.

DATOS	FASE=0	FASE=1
00	0°	45°
01	90°	135°
10	180°	225°
11	270°	315°

Tabla I  
FASE INICIAL DEPENDIENTE DE LA SEÑAL DE DATOS  
MODULANTE INTERNA O EXTERNA

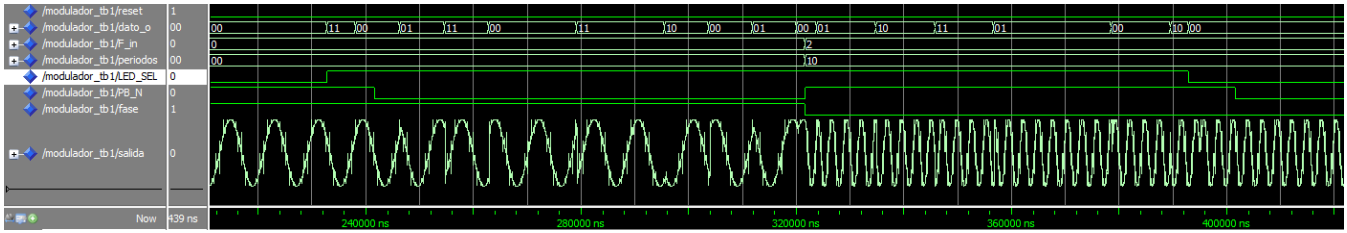


Figura 4. Simulación mediante ModelSim de la salida ante las variaciones de la frecuencia de portadora, cantidad de ciclos por dato, fase inicial y datos internos/externos.

En la Tabla II se detallan los recursos que se emplearon en la implementación del circuito en una FPGA Cyclone III de Altera. Se observa que, a pesar de almacenar un gran número de muestras de portadora para permitir una mayor flexibilidad en los parámetros, el diseño implementado logró minimizar el consumo de recursos respecto a [2] y [3].

Elementos lógicos (EL) totales (119.088)	668
Combinacional de EL	511
Registros de EL	58
Registros dedicados(121.673)	157
PLL	1
$f_{Max}$ [MHz]	181.01

Tabla II  
RECURSOS EMPLEADOS EN LA IMPLEMENTACIÓN UTILIZANDO UNA  
FPGA CYCLONE III DE ALTERA.

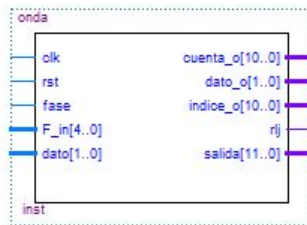


Figura 5. Bloque en Quartus II que realiza la lógica de control y contador N

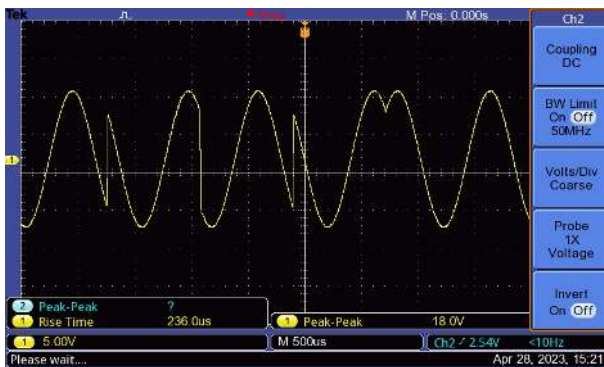


Figura 6. Se tomó una imagen del osciloscopio para mostrar la medición de la señal modulada, con FASE=1 (ver tabla I).

El diseño propuesto admite una frecuencia máxima de operación (fs) de 181,01 MHz lo que permitiría una máxima

frecuencia de portadora de 90,505 MHz, y una mínima de 176,7 kHz. En la simulación de la figura 4 se puede ver cómo varía la frecuencia de portadora ante el incremento de la entrada F, la modulación QPSK y la elección de la fase inicial. La lógica de control de barrido de la memoria ROM, como así también los pasos en que se toman las muestras, su recorrido ascendente o descendente, la generación del ciclo negativo de la señal portadora y la elección de la fase inicial, se realiza con Quartus II mediante el bloque de la figura 5. En la figura 6 se puede ver la señal de salida del DAC, tomada con el osciloscopio.

### III. CONCLUSIONES

En este proyecto se llevó a cabo sin inconvenientes el diseño e implementación de un generador de señal modulada en fase (QPSK). Gracias al diseño implementado, es posible variar diversos parámetros de la señal modulada de salida, como la frecuencia de portadora, la potencia de salida y la fase inicial, entre otros. Se logró minimizar significativamente el uso de recursos en la FPGA gracias al uso de una memoria que almacena un cuarto de ciclo sinusoidal, la cual se lee mediante una lógica de control para generar la señal de portadora. Además, se diseñó e implementó una placa que permite recibir la señal digital y entregar una señal analógica, con la posibilidad de ajustar su potencia de salida. Como trabajo a futuro, se puede reutilizar tanto el diseño en la FPGA como la placa DAC, que admiten el uso de otros tipos de modulaciones digitales.

### AGRADECIMIENTOS

El presente estudio contó con la colaboración económica del CONICET y de la Universidad Nacional de Mar del Plata (FI-UNMdP), en Argentina.

### REFERENCIAS

- [1] Y. Arjoun and N. Kaabouch, "A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions," *Sensors (Switzerland)*, vol. 19, no. 1, 2019.
- [2] B. Sujatha *et al.*, "Optimization of qpsk modem with awgn implemented in fpga," in *2017 International Conference on Inventive Systems and Control (ICISC)*. IEEE, 2017, pp. 1–6.
- [3] S. Popescu, A. Gontean, and D. Ianchis, "Implementation of a qpsk system on fpga," in *2011 IEEE 9th International Symposium on Intelligent Systems and Informatics*. IEEE, 2011, pp. 365–370.
- [4] R. Das, "Fpga implementation of digital modulation schemes: Bpsk and qpsk using vhdl," *International Journal of Electronics and Communication Technology (IJECT)*, vol. 5, no. Spl-2, 2014.
- [5] M. Morel, "GitHub," GitHub repository, 2023. [Online]. Available: <https://acortar.link/ffbTBL> (accedido el 28 de abril de 2023).
- [6] M. D. Morel, "Diseño e implementación de generador de señal bpsk con parámetros regulables," Ph.D. dissertation, Universidad Nacional de Mar del Plata. Facultad de Ingeniería, Argentina, 2022.

Categoría

# FORO TECNOLÓGICO

Hardware Abierto

Área Temática

# Pico RGB2HDMI - Conversor de Video RGB a HDMI y USB basado en Microcontrolador

Marcelo Adrián Lorenzati Sanz<sup>1</sup>, José María Ducloux<sup>2</sup>  
Especialización en Sistemas Embebidos  
Centro Regional Universitario Córdoba IUA  
Córdoba, Córdoba, Argentina  
<sup>1</sup>mlorenzati@gmail.com, <sup>2</sup>ducloux@iua.edu.ar

**Resumen-** La retrocomputación busca preservar, extender, aplicar y educar acerca de sistemas de cómputo antiguos con el fin de divulgar y consolidar las bases de la computación actual. El acto de preservar y extender involucra el desarrollo de nuevos periféricos que reemplacen a los que van desapareciendo por obsoletos y/o por desgaste de los componentes debido al paso del tiempo. Uno de los periféricos más relevantes es el controlador de video y el más usado en los años ochenta fue el RGBHV. Este artículo presenta el análisis, diseño, implementación y pruebas de un sistema de conversión en tiempo real de RGB a HDMI y captura USB, para conectar dichos sistemas retro a pantallas modernas desde un enfoque de factibilidad de desarrollo sobre un microcontrolador de propósitos generales.

**Palabras clave:** conversores, HDMI, retrocomputación, RGB

## I. INTRODUCCIÓN

Por un interés conservacionista, basado en que los sistemas modernos tienen pilares sobre la computación de las décadas del setenta, ochenta y noventa (retro computación) [1], se busca reparar, mantener y mejorar el hardware, los periféricos y programas asociados a dichas plataformas. Uno de los primeros desafíos son las interfaces que estos equipos usan, dado que por lo general entran en desuso por obsolescencia, siendo la interfaz analógica de visualización un caso de ello. De estas, la más utilizada en la década del ochenta fue el RGBHV (rojo-verde-azul-horizontal-vertical), precursor del VGA [2], el cual utilizaba tres señales analógicas para representar la información de rojo, verde y azul en el tiempo, así como dos señales digitales para sincronismo vertical y horizontal.

En la actualidad, por la escasez de monitores analógicos, estos sistemas requieren de conversores de la señal analógica a un estándar nuevo como la interfaz multimedia de alta definición (HDMI) [3]. Estos conversores son costosos y complejos dado que para cumplir con altas tasas de transferencia y latencia baja hacen uso de FPGAs, aumentando su valor y complejidad de desarrollo. Un ejemplo de esto es el proyecto Open-Source Scan Converter (OSSC) [4] que utiliza de base una FPGA Cyclone V y una interfaz analógica (AFE) ISL 51002 a menos que se presente una opción basada en un SoC [5].

En el ámbito de los sistemas embebidos hay factores que definen la elección de un SoC y su código, entre ellos están la velocidad de cómputo, la tasa de transferencia y la latencia [6]. En un SoC la mayor tasa de transferencia se logra por DMA, pudiendo transferir una palabra por ciclo de reloj sin uso del CPU [6], luego las interrupciones con mayor latencia y, por último el uso de servicios de un sistema operativo de tiempo real (RTOS) [7].

Con la aparición de los Cortex-M aparecen soluciones como el RP-2040 [8] de la Raspberry Pi, que integra dos M0+ y una serie de periféricos versátiles como es el caso de

las entradas-salidas programables (PIO) [8] que permiten interactuar, separado del CPU, con máquinas de estado que ejecutan sus propios programas y comunicarse con los diferentes buses del mismo. Esto amplía las posibilidades de aplicación, como el procesado en tiempo real de señales de video analógico, así como la posterior representación en tiempo real de la imagen en un estándar como el HDMI sin requerir un periférico dedicado de video.

En este trabajo se realiza el diseño y la implementación de un dispositivo conversor por captura de RGB analógico a HDMI y USB, utilizando un AFE [9] y los periféricos del RP-2040. Todas las funciones e incluso las más críticas (captura analógica en color y transmisión HDMI) son realizadas por el firmware del microcontrolador.

El resto del documento está diagramado de la siguiente manera. En la Sección II se expone el Desarrollo Teórico y Diseño, donde se definen las partes fundamentales del sistema y los requerimientos funcionales. En la Sección III se muestra la Implementación del hardware del sistema y del software embebido que se ejecuta en el microcontrolador. En la Sección IV se muestran las Pruebas realizadas y en la sección V se discuten los resultados obtenidos. Por último, en la sección VI se encuentran las Conclusiones.

## II. DESARROLLO TEÓRICO Y DISEÑO

En la Fig. 1 se muestra el diagrama de alto nivel del conversor RGB a HDMI propuesto. En base al este se detallan los subsistemas desde el punto de vista teórico para finalmente llevarlo al plano de implementación.

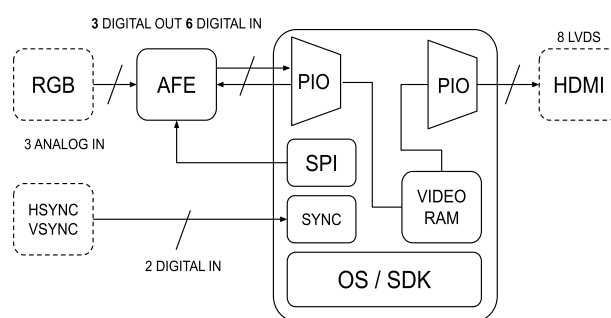


Fig. 1. Diagrama de bloques del sistema. Temporización y Captura

Esta etapa requiere la captura simultánea de las señales de rojo, verde y azul, usándose un AFE WM8213 [9] de tres canales S&H configurable por SPI que captura con tres señales: *MCLK* (reloj), *VSMP* (captura) y *RSMP* (reset), como muestra la Fig. 2a. Para el envío de datos se usa un canal de 4/8 bits permitiendo multiplexar cada color. El secuenciamiento de señales y captura se realizan aparte del CPU mediante una máquina de estados de la PIO [8] que se controla desde un canal de DMA.

El disparo de captura se realiza en base a la señal HSYNC que reporta el inicio de línea, el cálculo del reloj de píxel se formula en base a la frecuencia de HSYNC (líneas horizontales por segundo) y VSYNC (cuadros por segundo), y la cantidad de píxeles a capturar por línea se obtienen restando tiempos de porches frontales y traseros en relación a la preparación de la línea. Para el caso de modo de video RGB progresivo de 640x240 píxeles y 15.6 KHz de barrido, el número de líneas horizontales por cuadro HLines y la frecuencia de píxel PixelClk se calculan según la Fig. 2b.

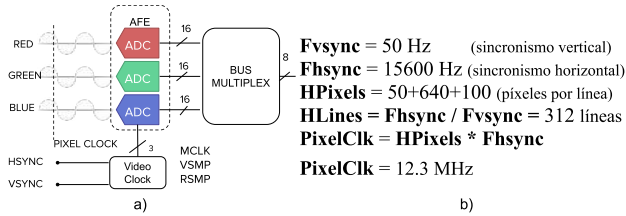


Fig. 2. (a) Módulo de temporización y captura y (b) cálculos.

A. Transporte y Conversión

El AFE puede capturar muestras de hasta 16 bits por color que se transmiten en multiplexado de 4 u 8 bits. Con 8 bits por color se obtiene la máxima densidad, por esto se configura al AFE para transmitir en ese modo. Dado que la memoria es limitada en el SOC (264 Kbytes) se debe comprimir el espacio de color en modos RGB de menor densidad, de manera de poder reservar un buffer de RAM de pantalla completa, teniendo como objetivo la resolución de 640x240 nativa del estándar RGB progresivo.

En la Tabla I se ve que el modo RGB332 [14] permite reservar la pantalla completa, usando media resolución (320x240) para el modo RGB565.

TABLA I: MODOS RGB

Modo	Bits				Bytes en 640x240
	Total	Rojo	Verde	Azul	
RGB24	24	8	8	8	460800
RGB565	16	5	6	5	307200
RGB332	8	3	3	2	153600
Monocromo	1	0	0	0	19200

Como los píxeles son transmitidos fuera del AFE en RGB24, deben ser convertidos en tiempo real al modo requerido, para esto se utiliza una máquina de estado disponible de una de las dos PIOs. La Fig. 3 muestra la secuencia que se sincroniza por DMA.

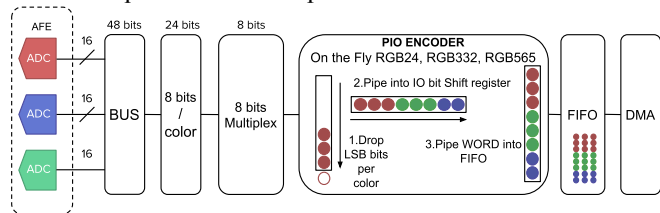


Fig. 3. Módulo de transporte y conversión.

La conversión en tiempo real aprovecha el limitado set de instrucciones de la PIO para:

1. Correr a la derecha cada byte de color el número necesario de bits.
2. Introducir en el registro de desplazamiento de IO de la PIO hasta completar los tres colores del píxel.
3. Desplazar la palabra con el píxel a la FIFO de salida.

B. Almacenamiento

En esta etapa el buffer de RAM queda disponible tanto para el proceso de captura y guardado como para el de lectura y transmisión por HDMI. Para la captura, se

configuran interrupciones por flanco en el pin de HSYNC, de manera de estar listo para tomar el contenido de video de la línea. Dicha transferencia se configura por DMA, siendo la máquina de estado de captura la que maneja el ritmo de transferencia. Dado que la línea posee un porche frontal, se configura un ciclo de captura ficticio encadenado al que transfiere al buffer de RAM como muestra la Fig. 4.

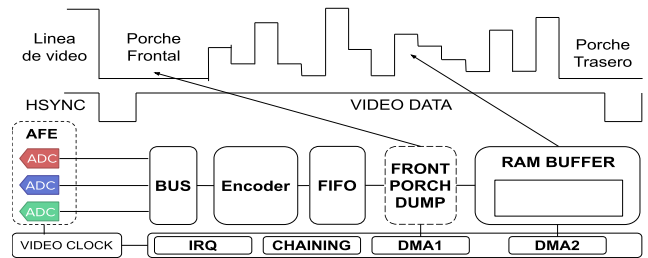


Fig. 4. Almacenamiento sincronizado.

El destino de la transferencia por DMA se calcula del mapa de memoria según el número de línea que se haya contado con HSYNC.

C. Renderizado

Este subsistema se encarga de generar la señalización diferencial de transición minimizada (TMDS) [10] sin utilizar un controlador de video dedicado. Se realiza línea por línea, con las siguientes etapas:

1. Tomar una línea del buffer según conteo de HDMI.
2. Codificar la línea RGB a un buffer de TDMS que utiliza HDMI, siendo tres tramas (rojo, verde, azul) cada una de ocho más dos bits de error y balanceado. Este proceso utiliza un núcleo del CPU más el módulo de interpolado.
3. Transmisión de TMDS balanceada usando tres máquinas de estado de la PIO y seis DMAs. Este subsistema (Fig. 5) se implementa mediante la biblioteca pico DVI [11] que se integró a la solución.

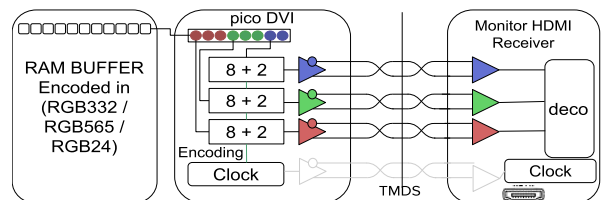


Fig. 5. Renderizado TMDS

D. Interfaz Gráfica de Usuario y USB

Al disponer de un buffer, se desarrolló una interfaz gráfica de usuario con superposición de video en tiempo real y adicionalmente al contar con módulo USB 1.0, se integró soporte de USB-CDC y una consola de comandos remotos. La Fig.6 muestra una imagen con superposición de menú capturada a USB.



Fig. 6. Captura USB de interfaz de usuario con superposición de video.

### III. IMPLEMENTACIÓN

#### A. Hardware

Éste usa placa Raspberry Pi Pico con el SoC RP-2040, un amplificador de video THS 7316 [12], el AFE WM8213 como ADC y para la salida HDMI separación por resistencias de carga. Un aspecto crítico fue el desarrollo del PCB contemplando separaciones de alimentación a base de ferritas y capacitores y el manejo de arreglo de vías entre planos de masa, dado que el SoC funciona a 250 MHz para generar señales diferenciales de TMDS y señales de sincronización del AFE a 120 MHz, lo cual produce interferencias que deben ser suprimidas o filtradas.



Fig. 7. Placas terminadas de etapas alfa y beta

#### B. Firmware y software

El firmware se desarrolló en capas desde bibliotecas de base, subsistemas críticos y módulos responsables de funciones hasta menús, manejo gráfico, almacenamiento, seguridad, escaneo, teclado y comandos. Se utiliza el SDK [13] del RP-2040, con rutinas modificadas para tiempo crítico, como el manejador de interrupciones de flanco del GPIO, que agrega latencia y jitter. Se incluyó la biblioteca pico DVI [11] para generar la señal TMDS, la cual usa un reloj a 250 MHz y requiere overclocking, así como la necesidad de correr en RAM las rutinas críticas para evitar la limitación de 130 MHz dada por la memoria QSPI externa de programa.

El software de computadora es sólo de soporte y permite, desde una aplicación de ventanas, acceder al modo de comandos por USB.

### IV. PRUEBAS Y MEDICIONES

Para probar el desarrollo se utilizó un microcomputador Commodore AMIGA 500, un monitor con soporte RGB, la placa bajo prueba y un DSO de 2 canales. Las señales a medir son HSYNC y el TP5 del SoC que es usado para medir el inicio y fin de operación de captura. Se comparan las imágenes en exactitud de color, brillo, relación de aspecto, ruido y jitter.

### V. RESULTADOS EXPERIMENTALES Y DISCUSIÓN

En la Fig. 8 se puede ver la diferencia entre el disparo de sincronismo y el procesamiento del SoC.

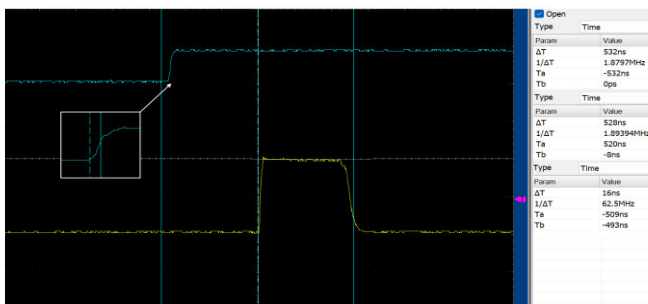


Fig. 8. Medición de señal HSYNC vs respuesta de captura.

Este tiempo es de 532 ns, como el CPU opera a 250 MHz (4 ns), la interrupción de HSYNC tarda 133 ciclos. El tiempo de procesamiento es de 520 ns (130 ciclos) y el jitter 16 ns (4 ciclos), como el reloj de píxel es de 12.3 MHz (81.3 ns) la probabilidad de capturar un píxel vecino es baja.

En la Fig. 9 se puede ver la comparativa por RGB (Fig. 9a) y por HDMI y el convertidor (Fig. 9b). A simple vista se pueden verificar diferencias ajustables como la relación de aspecto o el brillo, el número de líneas verticales en el convertidor es menor por la limitación de 320 líneas de captura. La imagen es estable, pero se observan artefactos por efectos de borde.



Fig. 9. Comparativa (a) RGB y (b) HDMI y convertidor

### CONCLUSIONES

Los resultados obtenidos de las etapas alfa y beta fueron óptimos para el uso contemplado, donde no se requiere alta resolución ni densidad de color sino velocidad de respuesta y fidelidad. Se contaron con 40 placas repartidas entre usuarios de Commodore Amiga & 128, Spectrum+, Amstrad CPC, Atari ST, MSX 2 y se busca soportar CGA y EGA.

Actualmente se está desarrollando el soporte HDMI para agregar audio. Este se envía en los tiempos ociosos de línea, donde se definen islas para metadatos como el audio.

Este proyecto será liberado a la comunidad Open-Source en busca de lograr criterios de calidad y performance, acompañado de mayores prestaciones a medida que aparezcan nuevos procesadores de la familia RP.

### AGRADECIMIENTOS

Esta solución no hubiera sido posible sin la investigación y desarrollo de Luke Wren y su biblioteca pico DVI [11].

### REFERENCIAS

- [1] Museo de Computadoras [Centro de Historia de la Computación]. Accedido: 2023-05-04. [Online] <http://www.computinghistory.org.uk>
- [2] S. D. Wolf, "Analog Video Interfaces," RGB & VGA Section, 1997.
- [3] High Definition Media Interface. Accedido: 2023-05-04. [Online] <http://www.microprocessor.org/HDMI/Specification14a.pdf>
- [4] marqs85, "Open Source Scan Converter" [GitHub]. Accedido: 2023-03-27. [Online] <https://github.com/marqs85/osscc>
- [5] P. Marwedel, "Embedded system design: Embedded systems foundations of cyber-physical systems," Springer Science & Business Media, 2011, pp. 35-38
- [6] More Throughput vs. Less Latency: Understand the Difference [National Instruments]. Accedido: 2023-03-27. [Online] <https://www.ni.com/es-cr/innovations/white-papers/13/make-it-faster--more-throughput-or-less-latency-.html>
- [7] S. L. Tan and B. A. Tran Nguyen, "Survey and performance evaluation of real-time operating systems (RTOS) for small microcontrollers," IEEE Micro, pp. 1-1, 2019, <https://doi.org/10.1109/mm.2009.56>
- [8] RP2040 Datasheet [Raspberry Pi Foundation]. Accedido: 2023-05-04. [Online] <https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf>, Chapter 3: RP2040 Pio
- [9] AFE WM8213 Datasheet [Wolfson Microelectronics]. Accedido: 2023-03-27. [Online] <https://www.cirrus.com/products/wm8213>
- [10] Digital Visual Interface (DVI) Revisión 1.0 [DDWG]. Accedido: 2023-03-24. [https://web.archive.org/web/20120813201146/http://www.ddwg.org/lib/dvi\\_10.pdf](https://web.archive.org/web/20120813201146/http://www.ddwg.org/lib/dvi_10.pdf).
- [11] Wren6991, "PicoDVI - Bitbanged DVI on the RP2040" [GitHub]. Accedido: 2023-03-24 [Online] <https://github.com/Wren6991/PicoDVI>
- [12] THS7316 Datasheet, Low-Noise High-Output-Drive Audio Amplifier [Texas Instruments]. Accedido: 2023-03-24 [Online] <http://www.ti.com/lit/ds/symlink/th7316.pdf>
- [13] Raspberry Pi Pico SDK [Raspberry Pi Foundation]. Accedido: 2023-05-04. [Online] <https://www.raspberrypi.com/documentation/pico-sdk/>
- [14] RGB Formats V4L2 [Kernel.org]. Accedido: 2023-03-24 [Online] <https://www.kernel.org/doc/html/latest/userspace-api/media/v4l/pixfmt-rgb.html>



Categoría

**FORO TECNOLÓGICO**

Implementación de SE

Área Temática

# Nuevo circuito neuronal basado en sistemas neuronales tipo P spiking para el cálculo del algoritmo de Euclides

Luis García, Moisés Rodríguez, Giovanni Sánchez, Juan G. Avalos, Eduardo Vázquez  
SEPI ESIME Culhuacan  
Instituto Politécnico Nacional  
Ciudad de México, México

jgarciap1111@egresado.ipn.mx, moises\_rodriguez1@outlook.es, gsanchezriv@ipn.mx, javaloso@ipn.mx, edvazquezf@ipn.mx

**Abstract**—This paper introduces for the first time a Euclidean circuit based on spiking neural P (SN P) systems to perform the Greatest Common Divisor (GCD). To create this compact circuit in terms of neurons and synapses, we use a cutting-edge variant of the SN P systems called spiking neural P systems with communication by request (SNQ P systems). In addition, the use of this variant has allowed us to create a circuit with low computational complexity. Therefore, its use opens new horizons in the development and improvement of cryptographic applications on embedded systems.

**Keywords**—Euclidean algorithm, GCD, SN P systems, cryptosystems

## I. INTRODUCCIÓN

Hoy en día, el algoritmo de Euclides es de gran relevancia en aplicaciones como: la teoría de control, procesamiento de señales y principalmente en el desarrollo de sistemas criptográficos [1-3]. Por ejemplo, este algoritmo es usado para desarrollar sistemas de comunicación seguros [4]. Esto se debe a que este algoritmo es simple y eficiente para calcular el máximo común divisor (MCD), el cual es usado para realizar la operación de inversión modular, siendo esta una de las operaciones más intensivas computacionalmente [5]. Una de las opciones potenciales para desarrollar esquemas de cómputo se ha propuesto recientemente en [6]. Este esquema de cómputo intenta superar las capacidades computacionales de los actuales sistemas binarios. Esto se debe al cómputo paralelo y distribuido que exhiben las redes neuronales biológicas. Inspirados en este fenómeno, se proponen los sistemas neuronales tipo P spiking. Los sistemas neuronales tipo P spiking (del inglés: SN P systems) son modelos de cómputo bio-inspirados, los cuales modelan el comportamiento de neuronas biológicas y su comunicación en el cerebro. En los sistemas neuronales tipo P spiking, las neuronas se comunican a través de impulsos eléctricos denominados spikes. Actualmente, estos sistemas neuronales son utilizados en el desarrollo de circuitos aritméticos, tales como: sumadores, restadores, multiplicadores, divisores, transformada discreta de Fourier, entre otros [7-9]. Diversos autores han demostrado la capacidad computacional de estos sistemas en aplicaciones prácticas. En este contexto, el desarrollo de nuevos circuitos neuronales abre nuevos horizontes en el mejoramiento de las aplicaciones anteriormente mencionadas. En este trabajo se presenta el diseño de un sistema neuronal tipo P spiking para el cómputo del algoritmo de Euclides de manera eficiente en términos de número de neuronas/sinapsis y ciclos de simulación.

## II. ALGORITMO DE EUCLIDES

El algoritmo de Euclides [3] es un método para calcular el MCD de dos números,  $n_1, n_2 \in \mathbb{N}$ , considerando  $n_1 > n_2$ . En este algoritmo, el MCD de  $n_1$  y  $n_2$  es expresado por  $mcd(n_1, n_2)$ , el cual se obtiene mediante una serie de divisiones de dos números enteros ( $r(i-1)$  y  $q(i)$ ) de la siguiente forma:

$$r(i-1) = q(i)r(i) + r(i+1) \quad (1)$$

donde los elementos de la división se representan como:

- $r(i-1)$ , dividendo.
- $q(i)$ , divisor.
- $r(i)$ , cociente.
- $r(i+1)$ , residuo.

En (1) se cumple que  $0 \leq r(i+1) < r(i)$  y se considera que cualquier divisor de  $r(i-1)$  y  $r(i)$  es también divisor de  $r(i)$  y  $r(i+1)$ . Por lo tanto, el MCD de  $r(i-1)$  y  $r(i)$  se expresa:

$$mcd(r(i-1), r(i)) = mcd(r(i), r(i+1)) \quad (2)$$

Si se asumen las condiciones iniciales  $r(0) = n_1$  y  $r(1) = n_2$ , entonces es posible el cálculo de las siguientes divisiones sucesivas:

$$\begin{aligned} r(0) &= q(1)r(1) + r(2) \\ r(1) &= q(2)r(2) + r(3) \\ &\dots \\ r(k-3) &= q(k-2)r(k-2) + r(k-1) \\ r(k-2) &= q(k-1)r(k-1) + r(k) \end{aligned} \quad (3)$$

donde  $k$  denota un índice entero,  $r(k)$  expresa el último residuo de las divisiones enteras sucesivas y se cumple que  $r(1) > r(2) > \dots > r(k)$ , lo cual nos permite calcular el MCD de los números,  $n_1$  y  $n_2$ , como:

$$\begin{aligned}
mcd(n_1, n_2) &= mcd(r(0), r(1)) = \dots \\
&= mcd(r(k-1), r(k)) \\
&= mcd(r(k-1), 0) \\
&= r(k-1)
\end{aligned} \tag{4}$$

En general, para calcular el  $mcd(n_1, n_2)$  mediante el algoritmo de Euclides se realiza la división de  $n_1$  dividido por  $n_2$ . Posteriormente, el valor de  $n_2$  es asignado a  $n_1$  y el residuo de la división se asigna a  $n_2$ . Este proceso continúa hasta que  $n_2$ , o también descrito con  $r(k)$  en (3), es igual a 0. Al final de esta operación, el valor de  $n_1$  es el resultado del MCD de los números enteros,  $n_1$  y  $n_2$ .

### III. CIRCUITO NEURONAL MCD TIPO P SPIKING PROPUESTO

En esta sección, se presenta el diseño de un circuito aritmético para el cálculo del algoritmo de Euclides. La estructura y el funcionamiento del circuito neuronal se basa principalmente en una variante de los sistemas neuronales tipo P spiking denominada sistemas neuronales tipo P spiking con comunicación por solicitud (del inglés: SNQ P systems) [10]. En los sistemas SNQ P, el comportamiento del sistema se determina por la activación de reglas, las cuales especifican cómo las neuronas interactúan entre ellas cuando se cumple una solicitud de comunicación. La definición formal de un sistema SNQ P de grado  $m \geq 1$ , se expresa como:

$$\Pi = (O, \sigma_1, \sigma_2, \dots, \sigma_m, a_{i_0}, out) \tag{5}$$

donde:

- $O = \{a_1, a_2, \dots, a_k\}$ .  $O$  es el alfabeto del sistema con  $k \geq 1$ , lo que indica el número de tipos de spikes.
- $\sigma_1, \sigma_2, \dots, \sigma_m$  son las neuronas, definidas por  $\sigma_i = (v_i, R_i)$  y  $m$  indica el número de neuronas, donde:
  - $v_i$  es un multiconjunto sobre  $O$ .
  - $R_i$  es el conjunto de reglas expresadas con la forma  $E/QW$ . Particularmente,  $E$  es una expresión regular sobre  $O$  y  $w$  es un conjunto finito de solicitudes expresadas como  $(a_s^p, j)$  y  $(a_s^\infty, j)$ .
- $a_{i_0}$  indica el tipo de spikes de salida y  $out$  es la neurona de salida del sistema.

En general, las operaciones de un sistema SNQ P se basan en solicitudes de comunicación. Por lo tanto, una solicitud de la forma  $(a_s^p, j)$  indica que una neurona  $\sigma_i$  solicita  $p$  copias de  $a_s$  spikes del soma de la neurona  $\sigma_j$ . Por otra parte, una solicitud de la forma  $(a_s^\infty, j)$  significa que todos los spikes ( $\infty$ ) contenidos en el soma de la neurona  $\sigma_j$  son solicitados por la neurona  $\sigma_i$ .

Una vez descrita la definición formal de los sistemas SNQ P se muestra el circuito neuronal propuesto,  $\Pi_{MCD}$  en la Fig. 1. Este circuito calcula el MCD de dos números enteros ( $mcd(n_1, n_2)$ ) empleando el algoritmo de Euclides.

Una vez validado el funcionamiento del circuito propuesto se define formalmente de la siguiente manera:

$$\Pi_{MCD} = (O, \sigma_1, \sigma_2, \sigma_3, \sigma_4, a, out) \tag{6}$$

donde:

- $O = \{a\}$ ,
- $\sigma_1 = (n_1, \{\lambda/Q(a^\infty, 3)\})$ ,
- $\sigma_2 = (n_2, \{a^+/Q(a^{n_2}, 1), \lambda/Q(a^\infty, 1)\})$ ,
- $\sigma_3 = (n_3, \{\lambda/Q(a^\infty, 2)\})$ ,
- $\sigma_4 = (0, \{a^*/Q(a^{n_3}, 2)\})$ ,
- $out = \sigma_4$

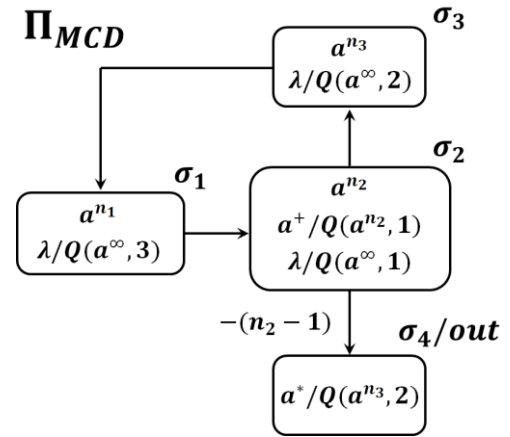


Fig. 1. Estructura neuronal del circuito  $\Pi_{MCD}$  propuesto para el cómputo del algoritmo de Euclides.

El circuito neuronal propuesto  $\Pi_{MCD}$ , calcula el algoritmo de Euclides (4) de la siguiente manera:

En el estado inicial del sistema ( $t = 0$ ), las neuronas  $\sigma_1$  y  $\sigma_2$  contienen spikes, los cuales representan a los números  $n_1$  y  $n_2$ , respectivamente. Por otra parte, los somas de las neuronas  $\sigma_3$  y  $\sigma_4$  no contienen spikes.

A partir del tiempo de simulación ( $t = 1$ ), el sistema realiza el cálculo de las divisiones sucesivas descritas en (3), donde cada operación de división se realiza mediante restas sucesivas. La neurona  $\sigma_3$  permite la sucesión de restas al almacenar una copia de  $n_2$  al aplicar su regla de solicitud  $\lambda/Q(a^\infty, 2)$ . Al mismo tiempo, las neuronas  $\sigma_1$  y  $\sigma_2$  aplican sus reglas de solicitud  $\lambda/Q(a^\infty, 3)$  y  $a^+/Q(a^{n_2}, 1)$ , respectivamente. El uso de esta estructura permite realizar restas sucesivas entre el contenido almacenado en la neurona  $\sigma_1$  menos el valor de  $n_2$ . Por lo tanto, cada ciclo de restas sucesivas termina cuando el contenido de spikes en la neurona  $\sigma_1$  es menor al contenido de spikes en la neurona  $\sigma_2$ .

En general, la finalización de un ciclo de restas sucesivas indica la realización de una división expresada en (3). La neurona  $\sigma_2$  queda sin spikes y el resultado de cada división se almacena en la neurona  $\sigma_4$  al cumplirse la regla de solicitud  $a^*/Q(a^{n_3}, 2)$  después del peso sináptico  $-(n_2 - 1)$ .

De acuerdo con (4), el proceso descrito anteriormente se repite hasta que el residuo de una división sea cero ( $r(n) = 0$ ). Finalmente, el resultado del cálculo de  $mcd(n_1, n_2)$  quedará almacenado en forma de spikes en la neurona  $\sigma_1$ .

Particularmente, si la neurona  $\sigma_1$  almacena un spike al finalizar el cálculo de  $mcd(n_1, n_2)$ , entonces se puede asegurar que  $n_1$  y  $n_2$  son números coprimos.

#### IV. EJEMPLO ARBITRARIO USANDO EL CIRCUITO NEURONAL

$$\Pi_{MCD}$$

En esta sección se describe un ejemplo arbitrario de cálculo de MCD de dos números  $mcd(n_1 = 12, n_2 = 7)$ , para demostrar el funcionamiento del circuito neuronal propuesto,  $\Pi_{MCD}$ . La Tabla 1 muestra cómo el circuito  $\Pi_{MCD}$  realiza el cálculo del  $mcd(n_1, n_2)$  en cada ciclo de simulación.

TABLA I. SIMULACIÓN DE EJEMPLO ARBITRARIO PARA CALCULAR  $mcd(12,7)$

Ciclo de simulación (t)	Neuronas			
	$\sigma_1$	$\sigma_2$	$\sigma_3$	$\sigma_4$
0	12	7	0	0
1	5	7	7	0
2	5	0	7	1
3	0	5	7	1
4	7	5	0	1
5	2	5	5	1
6	2	0	5	2
7	0	2	5	2
8	5	2	0	2
9	3	2	2	2
10	1	2	2	3
11	1	0	2	4

En  $t = 0$ , las neuronas  $\sigma_1$  y  $\sigma_2$  se inicializan con los números  $n_1 = 12$  y  $n_2 = 7$ , respectivamente. Desde  $t = 1$  a  $t = 2$ , se realiza la primer división, la cual se expresa como  $12 = (1)(7) + 5$  de acuerdo con (1). En la Tabla 1, se observa que durante el ciclo de simulación  $t = 2$  el cociente de la primer división es asignado a la neurona  $\sigma_4$  y el residuo es asignado a la neurona  $\sigma_1$ . Debido a que  $7 > 5$ , el circuito  $\Pi_{MCD}$  aún realizará dos divisiones más durante los intervalos  $t = 3$  a  $t = 6$  y  $t = 7$  a  $t = 10$ , respectivamente. Finalmente, en  $t = 11$  la última división se expresa como  $2 = (2)(1) + 0$ . Por lo tanto, como el residuo es igual a cero, el cálculo del MCD finaliza y el resultado queda almacenado en la neurona  $\sigma_1$ , es decir,  $mcd(12,7) = 1$ .

#### V. CONCLUSIONES

En este trabajo, se propuso un circuito neuronal basado en sistemas SNQ P para calcular el MCD mediante el algoritmo de Euclides. Esta propuesta es una innovadora solución que demuestra el potencial de los sistemas neuronales tipo P spiking en la resolución de problemas matemáticos complejos. El circuito propuesto ofrece una alternativa para el cálculo del MCD, lo que podría tener importantes aplicaciones en áreas como la criptografía, la seguridad informática y la computación cuántica, entre otras. Parte de nuestro trabajo a futuro es implementar el circuito neuronal propuesto en dispositivos embebidos reconfigurables (FPGA) y demostrar su eficiencia, lo cual permitiría mejorar el rendimiento del cálculo del MCD de los sistemas binarios convencionales.

#### REFERENCIAS

- [1] A. Fazzi, N. Guglielmi, and I. Markovsky, "Generalized algorithms for the approximate matrix polynomial GCD of reducing data uncertainties with application to MIMO system and control," *Journal of Computational and Applied Mathematics*, vol. 393, p. 113499, 2021, doi: [10.1016/j.cam.2021.113499](https://doi.org/10.1016/j.cam.2021.113499).
- [2] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 1996.
- [3] W. S. Brown and J. F. Traub, "On Euclid's algorithm and the theory of subresultants," *Journal of the ACM (JACM)*, vol. 18, no. 4, pp. 505-514, Oct. 1971, doi: [10.1145/321662.321665](https://doi.org/10.1145/321662.321665).
- [4] K. Sreedhar, M. Horowitz, and C. Torng, "A Fast Large-Integer Extended GCD Algorithm and Hardware Design for Verifiable Delay Functions and Modular Inversion," *Cryptology ePrint Archive*, 2021, doi: [10.46586/tches.v2022.i4.163-187](https://doi.org/10.46586/tches.v2022.i4.163-187).
- [5] M. Joye and P. Paillier, "GCD-free algorithms for computing modular inverses," in *Proceedings of Cryptographic Hardware and Embedded Systems-CHES 2003: 5th International Workshop, Cologne, Germany, Sept. 8-10, 2003*, vol. 5, Springer, Berlin, Germany, 2003, pp. 243-253, doi: [10.1007/978-3-540-45238-6\\_20](https://doi.org/10.1007/978-3-540-45238-6_20).
- [6] A. Păun and G. Păun, "Small universal spiking neural P systems," *BioSystems*, vol. 90, no. 1, pp. 48-60, 2007, doi: [10.1016/j.biosystems.2006.06.006](https://doi.org/10.1016/j.biosystems.2006.06.006).
- [7] S. Fan, P. Paul, T. Wu, H. Rong, and G. Zhang, "On applications of spiking neural P systems," *Applied Sciences*, vol. 10, no. 20, pp. 7011, Oct. 2020, doi: [10.3390/app10207011](https://doi.org/10.3390/app10207011).
- [8] L. Garcia, E. Vazquez, G. Sanchez, J. G. Avalos, and G. Sanchez, "An ultra-compact and high-speed FFT-based large-integer multiplier for fully homomorphic encryption using a dual spike-based arithmetic circuit over GF (p)," *Neurocomputing*, vol. 507, pp. 54-66, Jan. 2022, doi: [10.1016/j.neucom.2022.08.020](https://doi.org/10.1016/j.neucom.2022.08.020).
- [9] A. Vazquez, L. Garcia, J. G. Avalos, G. Sanchez, M. Nakano, K. Toscano, and J. C. Sanchez, "A high-precision multi-arithmetic neural circuit for the efficient computation of the new filtered-X Kronecker product APL-NLMS algorithm applied to active noise control," *Expert Systems with Applications*, vol. 191, pp. 116255, Dec. 2022, doi: [10.1016/j.eswa.2021.116255](https://doi.org/10.1016/j.eswa.2021.116255).
- [10] L. Pan, G. Păun, G. Zhang, and F. Neri, "Spiking neural P systems with communication on request," *International Journal of Neural Systems*, vol. 27, no. 08, pp. 1750042, Dec. 2017, doi: [10.1142/S0129065717500423](https://doi.org/10.1142/S0129065717500423).

Categoría

# FORO TECNOLÓGICO

Inteligencia Artificial

Área Temática

# Arreglos de antenas inteligentes para validación de mensajes ADS-B con SDR y machine learning

Francisco Martin ESCOBAR  
Grupo de Sistemas de Tiempo Real  
(GSTR)  
Universidad Nacional de Río Cuarto  
Río Cuarto, Cba., Argentina  
mescoabar@ing.unrc.edu.ar

Dario Walter DIAZ  
Centro I+D Tecnologías  
Aeronáuticas (CITeA)  
Fuerza Aérea Argentina  
Las Higueras, Cba.; Argentina  
dw\_diaz@faa.mil.ar

Ezequiel Alberto TARDIVO  
Grupo de Investigación, Vinculación e  
Innovación Tecnológica en Campos  
Electromagnéticos (GCEM)  
Universidad Nacional de Río Cuarto  
Río Cuarto, Cba.; Argentina  
etardivo@ing.unrc.edu.ar

Noelia María Virginia VEGLIA  
Grupo de Investigación, Vinculación e  
Innovación Tecnológica en Campos  
Electromagnéticos (GCEM)  
Universidad Nacional de Río Cuarto  
Río Cuarto, Cba., Argentina  
nveglia@ing.unrc.edu.ar

Damian Hector PRIMO  
Grupo de Sistemas de Tiempo Real  
(GSTR)  
Universidad Nacional de Río Cuarto  
Río Cuarto, Cba.; Argentina  
dprimo@ing.unrc.edu.ar

Diego Daniel BADINO  
Centro I+D Tecnologías  
Aeronáuticas (CITeA)  
Fuerza Aérea Argentina  
Las Higueras, Cba.; Argentina  
dd\_badino@faa.mil.ar

**Resumen**—Este trabajo describe el diseño de un sistema inteligente de arreglos de antenas para la recepción de mensajes del Sistema de Vigilancia Dependiente Automática (ADS-B), para abordar las vulnerabilidades del sistema ADS-B a las interferencias y falsificación de mensajes, empleando la Radio Definida por Software (SDR) y técnicas de aprendizaje automático para validar la fuente de emisión de los mensajes ADS-B. El trabajo principalmente discute el papel del arreglo de antenas inteligentes en la mejora de la recepción de mensajes ADS-B, describe la arquitectura y el diseño del sistema, así como el uso de algoritmos de aprendizaje automático para el procesamiento y análisis de señales. Los resultados presentados demuestran el uso del sistema en escenarios del mundo real y se discuten posibles casos de uso en la industria de la aviación.

**Keywords**—SDR, ADS-B, Arreglo de Antena, Red LSTM, Telemetría.

## I. INTRODUCCIÓN

El sistema ADS-B[1] permite que las aeronaves transmitan información en tiempo real sobre su posición, altitud, velocidad y otros parámetros, lo que permite a los sistemas de control en tierra seguir y supervisar sus movimientos con alta precisión. Sin embargo, el sistema presenta vulnerabilidades ante posibles interferencias e inyección de falsos mensajes con el protocolo ADS-B generando vuelos que no existen. Para abordar este problema, se presenta en este trabajo el diseño de un sistema de arreglos de antenas inteligentes que aprovecha la tecnología SDR[2] y algoritmos de aprendizaje automático para validar la fuente de emisión de los mensajes ADS-B. Se discute el papel de los arreglos de antenas inteligentes en la mejora de la recepción de mensajes ADS-B, se describe la arquitectura y diseño del sistema y se presentan resultados experimentales que demuestran su efectividad en escenarios del mundo real.

## II. METODOLOGIA

Se utilizó una configuración de cinco SDR de los cuales cuatro están sincronizados entre ellos y el quinto de los canales se conectó a una antena omnidireccional en 1090 MHz para realizar la recepción y decodificación de los mensajes ADS-B, esta misma se utiliza de referencia. Los otros 4 canales de los SDR, que se encuentran sincronizados,

fueron conectados a cuatro antenas omni- direccionales en 1090MHz con el objetivo de realizar un arreglo inteligente y validar la verdadera dirección de la transmisión del transpondedor de la aeronave.

Se aplicaron técnicas de machine learning para entrenar una red neuronal LSTM[3] con señales de RF de una trama ADS-B captada por el arreglo de antenas. La red LSTM se entrenó para determinar el ángulo de arribo de la señal del transpondedor en el espacio aéreo. Cuando la señal llega, la red devuelve una estimación de la dirección de la señal, que se envía al módulo validador. Éste calcula el ángulo con respecto a la estación receptora y compara los dos ángulos para verificar la autenticidad de la trama ADS-B.

El módulo validador compara los dos ángulos para validar si es un reporte real de una trama ADS-B o es una trama generada por una fuente ilegal. En la Figura 1 se muestra un esquema general del sistema.

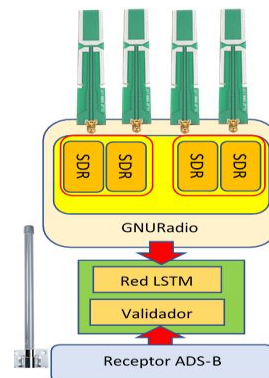


Figura 1. Esquema general del sistema de validación de mensajes

## III. DESARROLLO

Para el diseño, desarrollo y validación de un sistema de arreglo de antenas capaz de recibir los mensajes ADS-B se procedió con el desarrollo de la siguiente manera:

### A. Receptor ADS-B

La tecnología de vigilancia llamada ADS-B basada en el sistema de prevención de colisiones de tráfico (TCAS)[4]. El sistema ADS-B es más preciso, tiene menos errores y una mayor capacidad de vigilancia que el radar tradicional. Es

una tecnología que utiliza transpondedores en aeronaves para informar automáticamente su identificación, posición, velocidad y altitud a otras aeronaves y estaciones terrestres sin necesidad de una interrogación [5].

### B. Arreglo de Antenas

Para el sistema planteado se ha diseñado una estructura irradiante conformada por cuatro antenas dispuestas de manera tal que se generan cuatro sectores que abarquen un ancho angular de  $90^\circ$  cada uno. La configuración propuesta ubica a cada una de las antenas en forma de cruz, con una separación de  $\lambda/2$  (Figura 2). Con esta disposición y separación entre antenas que se encuentran opuestas entre sí, permite que, si la señal llega a una de las antenas, en la opuesta llegará con una diferencia de fase.

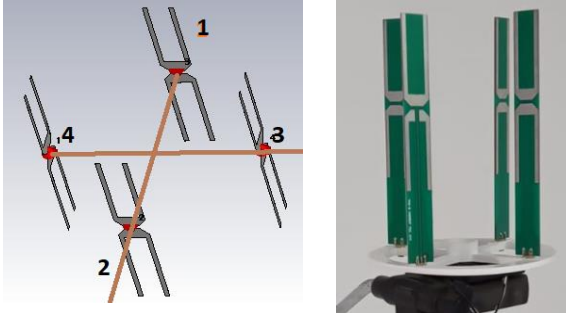


Figura 2. Arreglo de las antenas; antena 1 orientada hacia el Norte.

### C. Radio definida por Software - BladeRF de Nuand

SDR es un sistema de comunicación de radio que utiliza software para realizar funciones de radio en lugar de componentes físicos. El modelo BladeRF xA4[6] es altamente personalizable y flexible, lo que lo hace adecuado para diferentes aplicaciones de RF, y tiene dos receptores sincronizados que permiten mediciones precisas de fase y tiempo para mejorar la calidad y la confiabilidad de las comunicaciones inalámbricas.

### D. Redes neuronales

Para la resolución de este trabajo se realizó con dos SDR BladeRF, cada uno tiene dos receptores que están sincronizados entre ellos, por lo que se trabajó con dos series temporales por cada SDR, de estas señales se determina un posible ángulo de arribo que es procesada por la red neuronal y enviada al sistema validador. Dado que se desea analizar dos series temporales de 1500 puntos cada una y obtener un único valor que represente la diferencia de fase entre ellas, una arquitectura de red neuronal adecuada para este caso podría ser una Red Neuronal Recurrente RNN, tal como las redes LSTM. Estas redes están diseñadas específicamente para manejar secuencias de datos y han demostrado ser efectivas en tareas relacionadas con series temporales.

#### 1) Red neuronal recurrente LSTM

Las LSTM se componen de unidades de memoria llamadas "celdas LSTM", que tienen tres puertas (gate) que controlan el flujo de información en la celda. Las tres puertas son: puerta de entrada, puerta de olvido y puerta de salida. El proceso de entrenamiento de una red LSTM implica ajustar los pesos y los sesgos de las puertas de *entrada*, *olvido* y *salida*, a través de un algoritmo de retropropagación del error. Durante la fase de entrenamiento, la red LSTM aprende a ajustar estos pesos y sesgos para recordar u olvidar información relevante en función de los datos de entrada que recibe.

En este proyecto se realizó el preprocesamiento de datos para normalizar dos series temporales. Luego, se utilizó una capa de entrada que recibió ambas series temporales para combinarlas y alimentarlas a dos capas LSTM, que se utilizan para extraer características y capturar dependencias temporales. Posteriormente, se empleó una capa de salida con función de activación lineal para generar la diferencia de fase entre ambas series y obtener el ángulo de arribo. El modelo se entrenó con el algoritmo de optimización Adam y la función de pérdida MSE, y se controló su rendimiento en el conjunto de validación para evitar el sobreajuste.

Una vez que se entrenó y validó la red neuronal, se utilizó para predecir la diferencia de fase entre nuevas series temporales y el ángulo de arribo asociado en la función de Validación.

#### 2) Preparación de los datos para entrenar la red

El sistema completo está conformado por 4 antenas, pero se utiliza uno de los SDR que tiene dos entradas sincronizadas para comparar el desfase entre dos señales para determinar el ángulo de arribo entre dos antenas (1 y 2), el mismo procedimiento se realiza con otras dos antenas y otro SDR (3 y 4). En la Figura 3 se representa la orientación de las antenas y la dirección de la señal de RF proveniente de una fuente conocida que transmite la trama ADS-B.

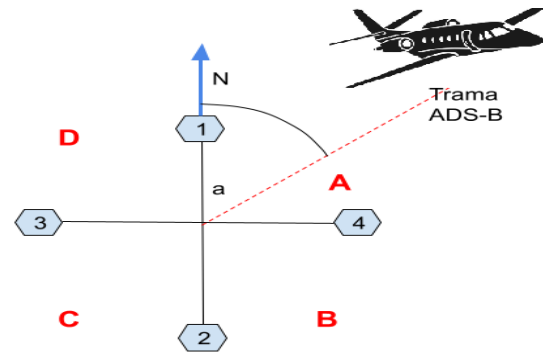


Figura 3. Diagrama de ubicación del arreglo de antenas

Las antenas 1 y 2 tienen una separación entre ellas de  $\lambda/2$ , y están ubicadas en la torre con una orientación Norte (antena 1) - Sur (antena 2); la misma separación tienen las antenas 3 y 4 con orientación Este (antena 4) - Oeste (antena 3). Si la señal de ADS-B llega con un ángulo de arribo desde el Norte o desde el Sur las señales que captan las antenas (1-2) estarán desfasadas entre ellas por  $180^\circ$  y si la señal llega por el Este u Oeste, las señales de ambas antenas están en fase; de esta manera se puede tener una relación entre el desfase de las señales y el ángulo de arribo de la fuente.

Para preparar el set de datos de entrenamiento para red neuronal se ubicó las antenas en una torre portátil a 2 mts. del suelo, y a 40 mts. de distancia se ubicó un transmisor ADS-B de baja potencia, que generaba y transmitía una trama ADS-B cada un segundo.

El software utilizado para su desarrollo es el GNURadio[7], en donde se colocaron dos bloques BladeRF (Soapy BladeRF Source) con dos salidas que fueron colocadas a las antenas 1 y 2, y las salidas del otro bloque fueron colocadas a la antenas 3 y 4; en la Figura 4 se visualiza la interconexión de los bloques. Además, se utilizó un bloque propio programado en lenguaje Python, en donde ingresan las 4 señales y las envía por un puerto TCP a otro módulo de Python que almacena los datos.

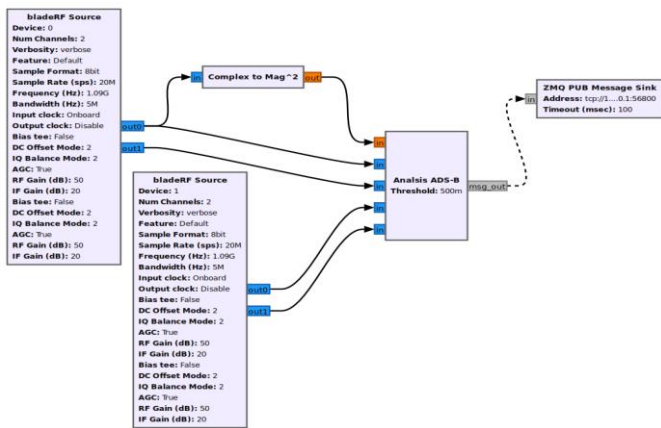


Figura 4. Bloques de GNURadio

Ambos SDR, fueron configurados para recibir a una frecuencia de 1090 MHz a una tasa de muestreo de 10MHz, con una ganancia de recepción de 73dB.

El bloque de Python, denominado Análisis ADS-B, cumple la función de recibir las cuatro señales de cada una de las antenas receptoras que forman parte de los SDR, que, cuando recibe un valor superior a un umbral de recepción preconfigurado en el bloque, significa que está ante la presencia de una señal que puede provenir desde un transmisor ADS-B. A partir de allí se almacena internamente un total de 1500 puntos por cada señal recibida. Ya que la tasa de muestreo del SDR es de 10MHz, y como una trama del protocolo ADS-B tarda 120µs, da que, con 1200 puntos se podría capturar toda la información de dicha trama, por tal motivo es que se utilizaron 1500 puntos por cada vez que se detecta señal superior al umbral.

Cuando termina de capturar y llenar 4 array de 1500 puntos, uno por cada canal receptor del SDR, toda esa información es enviada por un puerto TCP (mensajes tipo ZMQ Pub) a otro programa externo a GNURadio programado en lenguaje Python denominado "Algoritmo Validador".

3) Algoritmo Validador

Es un programa en Python que se ejecuta de manera independiente a GNURadio y del receptor ADS-B. El mismo opera en tiempo real; genera los posibles ángulos de la fuente de información transmitida por las aeronaves y los compara con lo que reporta la aeronave desde la información recibida de la trama ADS-B.

El ingreso de los datos a este algoritmo se realiza a través de dos sockets TCP (ZMQ Sub), un socket recibe desde el GNURadio la información de los 4 canales de los SDR y el otro socket recibe la trama ADS-B informando la posición geográfica reportada por las aeronaves desde el transponder ADS-B.

Ahora bien, como la red LSTM fue entrenada con dos conjuntos de datos, se utilizan dos funciones que llaman a los coeficientes del modelo LSTM (Modelo LSTM 1 y Modelo LSTM 2). Cada modelo recibe el par de señales, desde el socket de GNURadio, que corresponden a las antenas 1 - 2 y 3 - 4 dando como resultado un ángulo de arribo entre 0° y 180° cada uno de los modelos.

Por otro lado, desde el socket receptor ADS-B se recibe la posición geográfica que reporta la aeronave (Latitud,

Longitud, Altitud, etc.) y realiza un cálculo de azimut entre la posición de la estación receptora ADS-B y la posición reportada, dando como resultado un ángulo con respecto al Norte Geográfico, que será utilizado para compararlo con los estimados por las redes LSTM. En la Figura 5 se muestra el diagrama de flujo de "Algoritmo Validador".

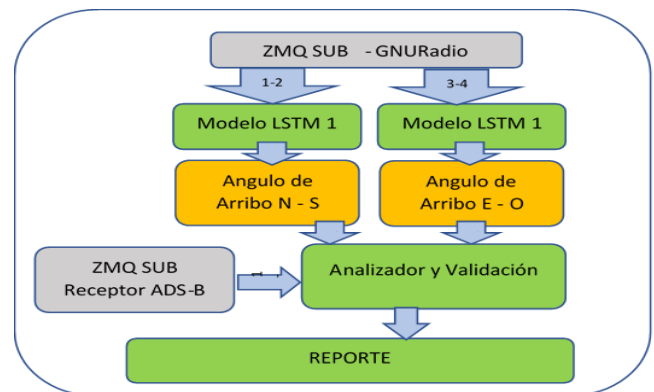


Figura 5. Algoritmo Validador

IV. RESULTADOS OBTENIDOS Y CONCLUSIONES

El sistema se puso a prueba con señales ADS-B de tráfico aéreo que circulaban en inmediaciones de la estación receptora. Los resultados iniciales de determinación angular no fueron los esperados cuando las señales de RF eran muy débiles con una S/N menor a 10dB ya que por debajo de ese valor el error era superior a ±25°, en la tabla I se muestran parcialmente un vuelo. Para mejorar los resultados, se decidió validar por cuadrante la señal de arribo.

TABLA I: Validación de reportado vs ángulo estimado

Ang ADS-B	Angulo Estimado	Error °	S/N(dB)
15	2	13	5
17	1	16	4
25	15	10	5
36	28	8	7
68	62	6	15
52	66	-14	2
...	...	...	...

REFERENCIAS

[1] RTCA, Minimum Operational Performance Standards (MOPS) for 1090 MHz Extended Squitter Automatic Dependent Surveillance - Broadcast(ADS-B) and Traffic Information Services - Broadcast (TISB). RadioTechnical Commission for Aeronautics, DO-260B 17, December 2011,pp. I3I22.

[2] Arthur (Art) Pini, "Aprenda los fundamentos de la SDR"; <https://www.digikey.com/es/articles/learn-the-fundamentals-of-software-defined-radio/>; 30/06/2020

[3] Carlos Arana, "REDES NEURONALES RECURRENTE: ANÁLISIS DE LOS MODELOS ESPECIALIZADOS EN DATOS SECUENCIALES", UCEMA: Av. Córdoba 374, C1054AAP Buenos Aires, Argentina,ISSN 1668-4575, Junio 2021

[4] RTCA/DO-300, Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System (TCAS II) For TCAS II Hybrid Surveillance. 2013.

[5] D. Primo, F. Escobar, D. Badino and J. R. Vittar, "System acquisition processing and visualization of Air Traffic Using ADS-B and IoT Connectivity", 2018 IEEE Biennial Congress of Argentina (ARGENCON), pp. 1-5, 2018.

[6] Nuand; <https://www.nuand.com/product/bladerf-xa4/>, accessed March 2023.

[7] E Gonzalez, C Zerbini, G Riva, "SDR con GNU Radio: de la teoría a la aplicación", Congreso de Microelectrónica Aplicada uEA 2016.



Categoría

# FORO TECNOLÓGICO

Protocolos y Comunicaciones

Área Temática

# Impedancímetro en la banda de UHF mediante técnicas de reflectometría

Facundo Alekseenko, Brian Gluzman, Ramiro Avalos Ribas, Alejandro José Uriz y Jorge Castiñeira Moreira

Laboratorio de Comunicaciones

ICYTE. Instituto de Investigaciones Científicas y Tecnológicas

Mar del Plata, Argentina

facundoalek@gmail.com, briangluzman@fi.mdp.edu.ar, avalosribas@fi.mdp.edu.ar, ajuriz@fi.mdp.edu.ar y casti@fi.mdp.edu.ar

**Resumen—** En este trabajo se propone el diseño e implementación de un medidor de impedancia en la banda UHF mediante técnicas de reflectometría. El dispositivo posee la capacidad de medir la relación de amplitud y la diferencia de fase de la señal reflejada de un sistema bajo prueba con respecto a una señal incidente. La principal novedad propuesta es utilizar un detector comercial y mediante un arreglo circuital implementado mediante técnicas de diseño en radiofrecuencias y procesamiento de señales, adecuarlo para que pueda medir en el rango completo de fase, distinguiendo el signo. Los valores adquiridos son procesados en una computadora para obtener la impedancia del sistema bajo prueba.

**Palabras clave--** Impedancímetro, Detector de Amplitud y Fase, AD8302, UHF.

## I. INTRODUCCIÓN

Los impedancímetros son instrumentos muy útiles a la hora de realizar diseños en electrónica. En redes o sistemas de múltiples puertos, la medición de impedancia permite caracterizar la carga que cada uno de los puertos representa para el circuito al que se interconecta, lo cual es imprescindible a la hora de diseñar las interconexiones y adaptaciones entre bloques en sistemas electrónicos complejos, para asegurar la compatibilidad electromagnética y la integridad de las señales. En bandas de frecuencias UHF, el conocimiento de la impedancia de un componente o sistema es de interés. Esto se debe a que las longitudes de onda se vuelven comparables con las dimensiones de los componentes electrónicos, y aparecen comportamientos parásitos. En este Trabajo se propone el desarrollo de un impedancímetro basado en el circuito integrado AD8302 [1] de Analog Devices. Para mejorar su rango de medición se propone implementar un arreglo de dos AD8302, con un desfase conocido entre ambos.

El trabajo se organiza de la siguiente manera: La Sección II presenta las características más importantes a medir, mientras que la Sección III describe el sistema implementado. La Sección IV presenta las pruebas realizadas. Finalmente, la Sección V presenta las conclusiones y las líneas de trabajo futuras.

## II. MEDICIÓN DE IMPEDANCIA

El sistema propuesto se basa en la relación entre el coeficiente de reflexión  $\rho$  con una carga  $Z_L$ , conectada a una línea de impedancia característica conocida  $Z_0$ . Es

posible calcular  $Z_L$ , a partir del coeficiente de reflexión  $\rho$  [2]. Para la implementación se propone utilizar el circuito integrado detector AD8302. Este dispositivo entrega dos salidas: la primera con un valor de tensión proporcional a la relación de magnitud entre sus entradas, y la segunda con un valor de tensión proporcional a la diferencia de fase entre  $0^\circ$  y  $180^\circ$ , sin distinción de signo. Esto es una desventaja, ya que el circuito integrado no permite discernir el signo de la fase. Por ello, se propone construir un arreglo con dos AD8302 con el fin de poder determinar el signo de la fase. Con este propósito, se propone incluir una diferencia de fase conocida entre detectores, con el fin de poder realizar una comparación entre ambas salidas y en base a la misma, determinar el signo de la fase y eliminar inconsistencias de la medición. De esta manera se obtiene una relación de tensión y una diferencia de fase, equivalente a  $\rho$  [2]. Si bien el cálculo es directo, hay que considerar que el uso de conectores y cables, más los errores constructivos, agregan pérdidas y aportes de fase, que varían con la frecuencia. O sea, el coeficiente de reflexión se ve afectado por el trayecto entre la carga a medir y el medidor. Por lo tanto, la medición requiere la implementación de una ecuación de calibración [3].

## III. SISTEMA IMPLEMENTADO

El sistema está compuesto por cinco módulos interconectados, los cuales se presenta en la Figura 1. El bloque controlador está compuesto por un PIC18F4550 [4], es el encargado de adquirir los datos provenientes del circuito detector, a la vez controlar al bloque generador de RF, conformado por el integrado AD4351 [5] y un amplificador de RF ERA-5+ [6]. El acoplador direccional ZABDC20-322H-S [7] separa la onda incidente de la reflejada. La interfaz con el usuario se genera en una PC.

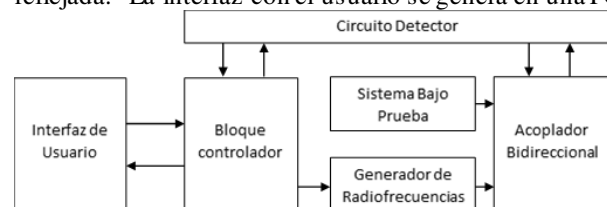


Figura 1. Diagrama en bloques completo con el sistema bajo prueba conectado.

El circuito detector, conformado por dos integrados AD8302, dos *splitter* y una diferencia de fase en los caminos que conectan los *splitter* con los integrados, se

encarga de realizar la medición de la relación de amplitud y fase entre la señal incidente y la reflejada.

A. Circuito detector

Diseñados con tecnología de guía de onda coplanar conectada a tierra (GCPW) [8], se introducen dos *splitter*, que separan las señales ingresantes en caminos de distintos largos, logrando que, al conectar los detectores, los mismos midan una diferencia de fase distinta afectada por la diferencia de caminos. La Figura 2 muestra un esquema del circuito detector.

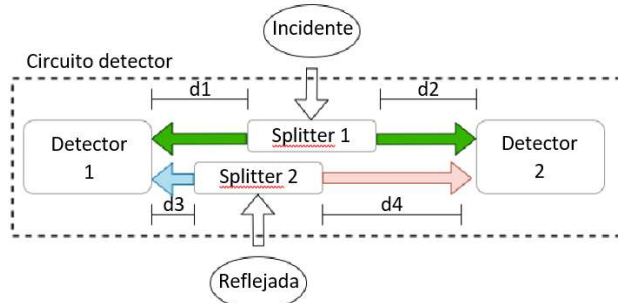


Figura 2. Diagrama en bloques del circuito detector.

A.1 Detectores

En su configuración típica propuesta por el fabricante, las mediciones entregan valores de tensión entre 0V y 1,8V utilizando las expresiones presentadas en [1]. El rango de medición de magnitud es de  $\pm 30\text{dB}$  y el de la relación de fase es de  $\pm 180^\circ$  sin distinción de signo (entrega la misma tensión con fase positiva o negativa). Mediciones por debajo de  $20^\circ$  y encima de  $160^\circ$  presentan errores significativos, por lo que es conveniente corregir valores medidos en esas zonas.

A.2 Divisor de potencia (*splitter*)

Divide la potencia de entrada en dos partes iguales. Se implementaron divisores del tipo Wilkinson [9] por presentar pérdidas de inserción cercanas a las ideales y aislamiento entre sus puertos de salida superior a los 20dB en la banda de (1,5-2GHz).

A.3 Diferencia de caminos

Para lograr la distinción de signo se propone un desfase de  $90^\circ$  a la frecuencia de 1,75GHz. Como se puede ver en la Figura 2, se coloca un *splitter* equidistante de los detectores (trayectorias d1, d2) y al otro descentrado una distancia equivalente a  $45^\circ$ , de esta manera uno mide  $45^\circ$  de menos (trayectoria d3) y el otro  $45^\circ$  de más (trayectoria d4), cumpliendo con los  $90^\circ$  entre ellos. Además, se evita introducir empalmes de líneas que puedan provocar atenuaciones indeseadas. La distancia equivalente a  $45^\circ$  sale de considerar un octavo de la longitud de onda ( $\lambda/8$ ) a la frecuencia de 1,75GHz. Dado que  $\lambda$  es inversamente proporcional a la frecuencia, en la banda de operación, se obtienen los desfases que se presentan en la Tabla 1 [2].

TABLA 1

Frecuencia (GHz)	1,5	1,6	1,75	1,9	2
Desfase (grados)	77,2	82,3	90,04	97,8	102,9

Tabla 1. Desfase – frecuencia.

Las zonas de medición indeseadas no se pueden evitar en ambos detectores a la vez, pero, con el desfase introducido, se logra que al menos uno mida fuera de esa zona, es decir, libre de la indeterminación allí presente.

La Figura 3 presenta el sistema implementado. Con el número 1 se señala la entrada de la onda directa. Con el 2 se señala la entrada de la reflejada. Los números 3 y 4 señalan los divisores de potencia 1 y 2, respectivamente. Se señalan las trayectorias d1, d2, d3 y d4. Mientras que con los números 4 y 5 se presentan los detectores 1 y 2.

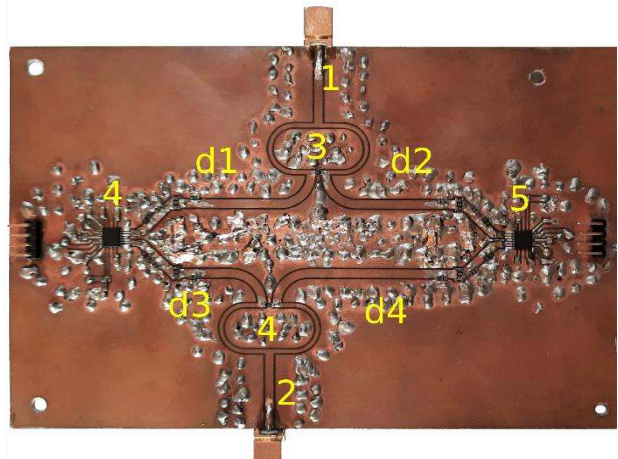


Figura 3. Fotografía del circuito detector con los divisores de Wilkinson, la red desfasadora y los detectores implementados en una topología GCPW.

B. Microcontrolador PIC18F4550

Este microcontrolador configura los registros del ADF4351 y adquiere los valores provenientes de los detectores mediante convertidores ADC de 10 bits y una tensión de referencia del ADC de 1,92V, lo cual nos provee una resolución de 1,875mV. La Figura 4 presenta un diagrama de flujo del firmware implementado.

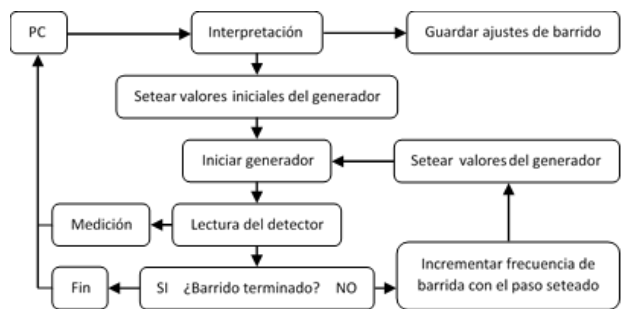


Figura 4. Diagrama de flujo del firmware.

C. Generador ADF4351

Este circuito es capaz de generar una salida senoidal en el rango 35-4400MHz. Su salida es programable mediante registros de 32bit. Se programa para generar un barrido entre 1,5GHz y 2GHz y una resolución de 1MHz.

IV. EXPERIMENTACIÓN Y ANÁLISIS DE RESULTADOS

A. Diferencia entre la fase teórica y la real

Se implementó una rutina para determinar la diferencia de fase en cada frecuencia con la condición de que los

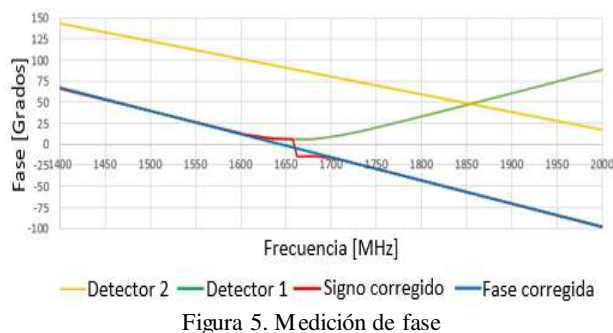
valores de ambos detectores se encuentren fuera de la zona de medición con error.

Los resultados obtenidos logran los objetivos de discernir el signo de la fase y de asegurar que al menos uno de los detectores mida fuera de la zona con error. Sin embargo, difieren de los valores teóricos ( $<10^\circ$ ), debido a variaciones entre los cálculos y las propiedades del sustrato y/o imperfecciones en el proceso de fabricación.

### B. Identificación del signo de fase y mediciones en zona de error

Se realizaron mediciones reales de un sistema bajo prueba antes de aplicar la calibración. En la Figura 5, se presenta en color verde la salida de fase del detector 1, mientras que el color amarillo se presenta la salida de fase del detector 2. Ambas curvas muestran medición de fase de los coeficientes de reflexión provenientes de cada detector. Se puede observar que la curva del detector 1, al entrar en la zona de  $20^\circ$ , comienza a perder la linealidad, también se puede apreciar el cambio de pendiente debido a la incapacidad del detector individual de distinguir el signo. A partir de  $1,7\text{GHz}$ , la separación entre las mediciones comienza a reducirse, lo que la rutina interpreta como un cambio de signo.

En la curva de color rojo (signo corregido) se observa la interpretación del signo llevada a cabo a partir de los datos de ambos detectores, el algoritmo coloca el signo que le asegure mantener la diferencia de fase correspondiente a cada frecuencia. Para compensar la incidencia en la zona de medición con error entre los  $-20^\circ$  y  $+20^\circ$ , el algoritmo emplea la medición de fase más confiable, tomando en este caso la medición del detector 2 y agregándole (o quitándole) la diferencia de fase real entre los detectores. Este proceso se puede observar en curva de color azul (fase corregida) donde se corrigen el signo y los errores de medición existentes en la roja.



### C. Mediciones de impedancia

Tomando como sistema bajo prueba cargas conocidas y previamente medidas en un analizador vectorial de redes, se realizaron las mediciones con el instrumental diseñado y calibrado [3]. Los resultados obtenidos para cargas de prueba se pueden ver en la Figura 6.

En primer lugar, para verificar la capacidad de medir impedancias complejas, se realizó una prueba emulando distintos coeficientes de reflexión. Se conectó una carga

de  $10\Omega$  con un cable de RF de por medio. Y se realizó un barrido en frecuencia entre  $1,5\text{GHz}$  y  $2\text{GHz}$ . Puede verse que, al modificar la frecuencia, el largo eléctrico del cable varía y por ello se produce un desplazamiento circular a lo largo del Ábaco [9]. El módulo del coeficiente de reflexión se mantiene igual, pero debido al camino introducido por el cable existe una variación en su ángulo. Esto último se condice con el desplazamiento mostrado en la izquierda de la Figura 6. A la derecha de la Figura 6 se aprecia una medición de una carga de  $100\Omega$  conectada directamente al impedancímetro, con el mismo rango de barrido de frecuencias. Se realiza una representación de los resultados en formato cartesiano, pudiendo apreciarse que la parte real medida es  $100\Omega$  y la imaginaria  $0$ .

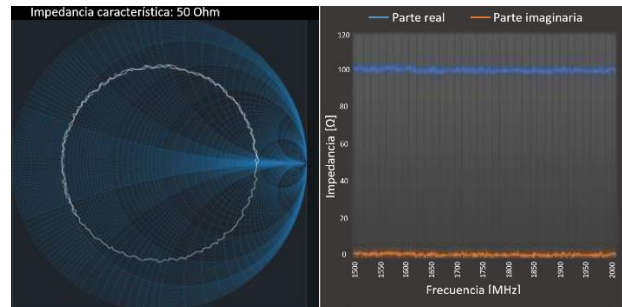


Figura 6. Mediciones de impedancia: Izquierda: Medición de impedancia,  $10\Omega$  con cable de RF, representada en un Ábaco, derecha:  $100\Omega$  representada en un gráfico.

## V. CONCLUSIONES Y DESARROLLOS FUTUROS

La implementación del impedancímetro se realizó de forma exitosa. La hipótesis y la implementación de la técnica de desambiguación de signo mediante una “diferencia de fase conocida”, resultó efectiva. Los resultados obtenidos respaldan el correcto funcionamiento de la placa diseñada y los bloques encargados del procesamiento de datos. Es importante destacar que este modelo puede ser aplicado en otros rangos de frecuencias, con los correspondientes ajustes.

En el futuro se propone incorporar circuitería para lograr implementar un analizador vectorial de redes.

## REFERENCIAS

- [1] “Analog Devices. AD8302” (Datasheet). Disponible en: <https://www.analog.com/media/en/technical-documentation/datasheets/ad8302.pdf>
- [2] David K. Cheng. “Fundamentals of Engineering Electromagnetics”, Pearson Education 2014. Capítulo 8.
- [3] Doug Rytting. “Network analyzer error models and calibration methods”. Disponible en: [https://www.rfmentor.com/sites/default/files/NA\\_Error\\_Models\\_and\\_Cal\\_Methods.pdf](https://www.rfmentor.com/sites/default/files/NA_Error_Models_and_Cal_Methods.pdf)
- [4] “Microchip. PIC18F4550” (Datasheet) Disponible en: <https://ww1.microchip.com/downloads/en/devicedoc/39632e.pdf>
- [5] “Analog Devices. ADF4351” (Datasheet) Disponible en: <https://www.analog.com/media/en/technical-documentation/datasheets/adf4351.pdf>
- [6] “Mini Circuits. ERA - 5” (Datasheet) Disponible en: <https://www.minicircuits.com/pdfs/ERA-5+.pdf>
- [7] “Mini Circuits. ZABDC20-322H-S” (Datasheet) Disponible en: <https://www.minicircuits.com/pdfs/ZABDC20-322H-S+.pdf>
- [8] B. Wadell. “Transmission line design handbook”, 1991. Página 79.
- [9] D. M Pozar. “Microwave Engineering”, Wiley 2011.

# Micro Red IOT multipropósito

Walter Gemin, Juan López, Rodrigo Russo y Melisa Kuzman  
*Instituto de Investigaciones Científicas y Tecnológicas en Electrónica (ICYTE).*  
*Facultad de Ingeniería, Universidad Nacional de Mar del Plata*  
 Mar del Plata, Argentina

Email: geminw@hotmail.com, juanml198528@gmail.com, rodrigo.e.russo@gmail.com, melisakuzman@gmail.com

**Resumen**—En este trabajo se presenta la implementación de una red IOT multipropósito, que permite la conexión en red de dispositivos controlados mediante comandos serie RS232. Esta red no necesita ser programada para su puesta en funcionamiento; únicamente se debe seleccionar la dirección de cada módulo mediante un dip swith. Luego mediante esta misma dirección es posible controlar cada dispositivo conectado. El hardware utilizado es de bajo costo y está ampliamente difundido ya que se basa en Arduino Nano y el transceptor nRF24L01. El principal logro, es que de manera muy rápida se puede implementar y comprobar el funcionamiento de una red de IoT.

**Palabras Clave**—IoT, nRF24L01, Arduino, Red.

## I. INTRODUCCION

Hoy en día es cada vez más habitual la conexión de dispositivos en red para diseños de domótica, monitoreo, o control de diversos equipos. Existen actualmente módulos o interfaces para IoT (Internet Of Things) [1] [2] que ofrecen diversas alternativas de comunicación inalámbrica como, Bluetooth [3] , ZigBee [4], Lora [5], dentro de las más conocidas. Todas estas tecnologías permiten conformar redes inalámbricas con diferentes características de alcance, consumo y topologías de red, pero tienen la dificultad de que cada una se configura de manera diferente.

Un problema habitual se presenta cuando es necesario seleccionar una tecnología adecuada para la interconexión en red de dispositivos de diversos fabricantes que funcionan con comandos propios y por defecto su interfaz de comunicación es RS232. Este problema dificulta y retrasa la concreción de proyectos.

En este trabajo se presenta una solución rápida y eficiente para crear una red local de corto alcance, de fácil configuración y con un costo muy reducido con respecto a otras alternativas.

## II. DESCRIPCIÓN GENERAL

El objetivo general es diseñar una interface de comunicación para domótica, robótica colaborativa y otras aplicaciones que requieran de una red inalámbrica de corto alcance. Esta interfaz está basada en el transceptor nRF24L01 [6] [7] y un módulo Arduino Nano [8]. Estos dos componentes están ampliamente difundidos y su costo es bajo. El diagrama en bloques de funcionamiento general se presenta en la Fig. 1.

## III. DESCRIPCIÓN DEL HARDWARE

### A. Módulo transceptor nRF24L01

El transceptor inalámbrico nRF24L01 es un módulo de RF de 2Mbps para la banda ISM (Industrial, Scientific and Medical) de 2,4 GHz. Con corrientes inferiores a 14 mA,

administración avanzada de energía (ideal para aplicaciones con baterías), y una gama de alimentación de 1,9 a 3,6 V. Son además de bajo costo y están diseñados para ser utilizados con microcontroladores por medio de una interfaz serie SPI.

Existen 2 modelos: El más básico y económico es el que se muestra en la Fig. 2.a, que básicamente es el chip NRF24L01 y sus componentes necesarios para su funcionamiento con un alcance máximo entre 20 y 30 metros. El modelo de la Fig. 2.b es más completo. Aparte del NRF24L01 posee un circuito amplificador de potencia y un circuito amplificador de bajo ruido, además de una antena SMA que en conjunto le permiten lograr un rango entre 700 y 1000 metros.

### B. Arduino Nano V3.0

Este módulo es una versión más pequeña de la placa Arduino UNO, basada en el ATmega328 SMD y se utiliza junto al transceptor nRF24L01 para conformar al maestro y los esclavos de la red.

### C. Nodo maestro o esclavo de la micro red IoT

En la Fig. 3. se observa el esquema de conexiones del nodo que puede ser utilizado como maestro o esclavo de la red seleccionando desde el dip switch la dirección correspondiente. La dirección 0 corresponde al maestro y las direcciones 1 a 15 corresponden a los esclavos. Además de la interfaz serie RS232, se aprovechan pines de entrada y salida digitales, interrupciones, PWM, entradas analógicas y sensor de temperatura digital DS18B20, que pueden ser accedidas en forma local o remota mediante comandos serie RS232 desde el módulo maestro.

### D. Implementación del nodo

En la Fig. 4. se observa la implementación del circuito impreso de dos nodos.

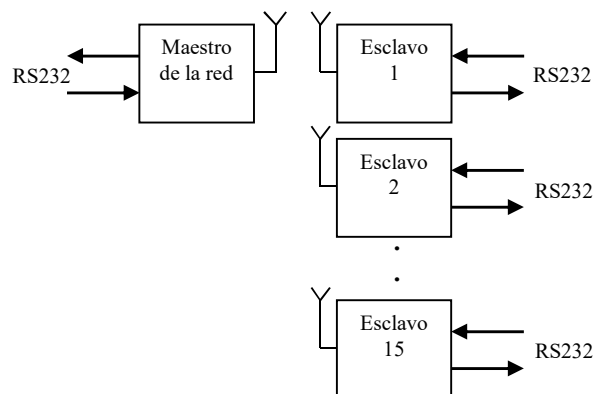


Fig. 1. Diagrama en bloques de funcionamiento.

IV. FUNCIONAMIENTO DE LA RED

Esta red está basada en las bibliotecas para Arduino RF24.h, RF24Network y RF24Mesh.h [9] que permiten utilizar los transceptores nRF24L01 para armar redes tipo mesh con dirección asignada mediante DHCP por el Maestro y un ID configurable en cada esclavo que está asociado a la dirección asignada por el maestro. Por lo tanto es posible direccionar a los esclavos por su ID, sin necesidad de conocer su verdadera dirección. En esta versión de red implementada, no se utiliza la máxima capacidad de direccionamiento que es de 255, sino que está limitada a 16 por los dip switch con el objetivo de simplificar la configuración de la misma y que el usuario pueda fácilmente armar una red sin el conocimiento de las tecnologías involucradas. Para que la red funcione en forma adecuada, el maestro debe estar encendido para que se habilite el servicio DHCP y el de ruteo. A medida que se encienden los esclavos, se les asigna una dirección dentro de la red y estos informan al maestro cuál es su ID, para que se genere la tabla de enrutamiento.

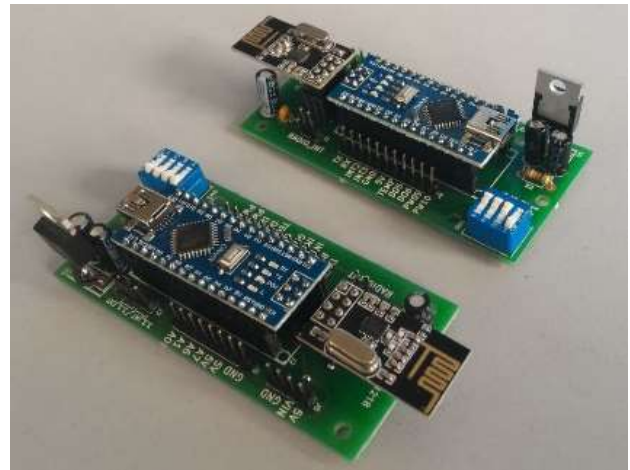


Fig. 4. Implementación de dos nodos.

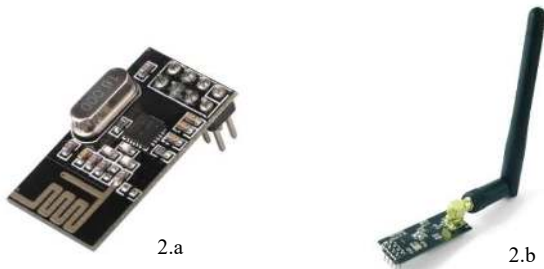


Fig. 2. Transceptor nRF24L01 de baja y alta potencia.

V. REPERTORIO DE COMANDOS

A los esclavos se los puede conectar a cualquier módulo o interfaz comercial con comunicación RS232 y adaptarlos para un trabajo en red. Para enviar mensajes o comandos a un esclavo, hay que enviarlos a través del maestro conectándose por RS232. La trama debe ser encabezada con la secuencia de caracteres ASCII de direccionamiento: IDNN, donde NN son caracteres numéricos entre 01 y 15 que identifican al esclavo correspondiente. Luego del encabezado, pueden enviarse los comandos del dispositivo conectado, el cual no recibirá el encabezado de direccionamiento, solo el comando que puede interpretar el dispositivo conectado. Otra opción es enviar luego del direccionamiento comandos que interpreta el nodo como fue mencionado en la sección C de la descripción del hardware. La trama debe finalizar con los caracteres de control especiales lf y cr. La Tabla I, muestra los comandos para acceder a las funcionalidades del nodo.

TABLA I. COMANDOS DISPONIBLES

Comando	Descripción	Respuesta
DIO5H	Salida digital 5 en alto	DIO5H\r\n
DIO5L	Salida digital 5 en bajo	DIO5L
DIN2	Lectura de entrada digital 2	DIN20\r\n ó DIN21\r\n
DIN3	Lectura de entrada digital 3	DIN30\r\n ó DIN31\r\n
INT2	Habilita la Interrupción del pin 2	INT2\r\n (para confirmar la configuración) ó Int2\r\n (cuando se produce la int.)
NOINT2	Deshabilita la Interrupción 2	NOINT2\r\n
INT3	Habilita la Interrupción del pin3	INT3\r\n (para confirmar la configuración) ó Int3\r\n (cuando se produce la int.)
NOINT3	Deshabilita la Interrupción 3	NOINT3\r\n
PWM09127	Salida PWM del pin 9 en 127 (varía entre 0 y 255)	PWM09127\r\n
PWM10060	Salida PWM del pin 10 en 60 (varía entre 0 y 255)	PWM10060\r\n
AN0	Lee entrada analógica del pin 0	AN01023\r\n
AN1	Lee entrada analógica del pin 1	AN10100\r\n
TEMP	Lee el valor suministrado por el sensor DS18B20	TEMP020\r\n

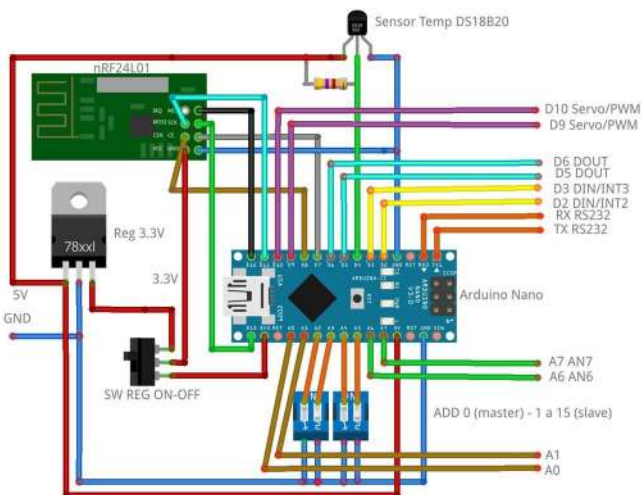


Fig. 3. Esquema de conexiones de los nodos.

VI. FIRMWARE

El firmware está diseñado en el entorno de Arduino y se divide en cuatro partes. La inicialización, donde se cargan las bibliotecas correspondientes a la red mesh y se declaran las variables utilizadas. La parte de configuración o setup donde se lee la dirección propia del nodo, se inicializa la pila de protocolos de red mesh y de acuerdo a si el nodo es maestro crea un servidor de DHCP (asigna direcciones a los esclavos) o si es esclavo se conecta al maestro y recibe una dirección. El lazo loop en ambos casos (maestro o esclavo) queda a la espera de recepción de datos desde el módulo nRF24L01. Si los datos recibidos son comandos, se ejecutan en el nodo y si no lo son, se reenvían a la interface RS232. El diagrama de flujo se visualiza en la Fig. 5.

En caso de recibir caracteres desde la interface RS232, se despierta el evento serialEvent. Dentro de esta rutina se identifica si la trama recibida corresponde a un comando, en cuyo caso es ejecutado y si no corresponde a un comando, la trama se envía a través del módulo nRF24L01 al maestro, si se trata de un esclavo o al destino si se trata del nodo maestro. El diagrama de flujo correspondiente a este evento se observa en la Fig. 6.

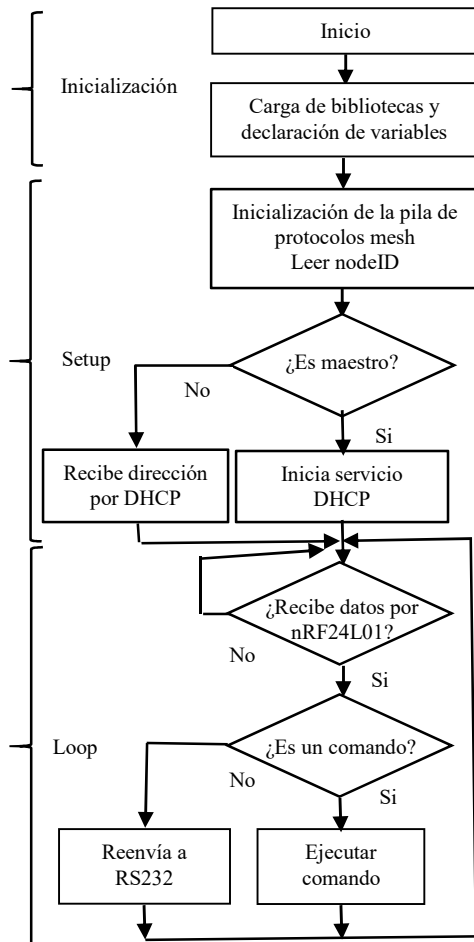


Fig. 5. Diagrama de flujo del nodo.

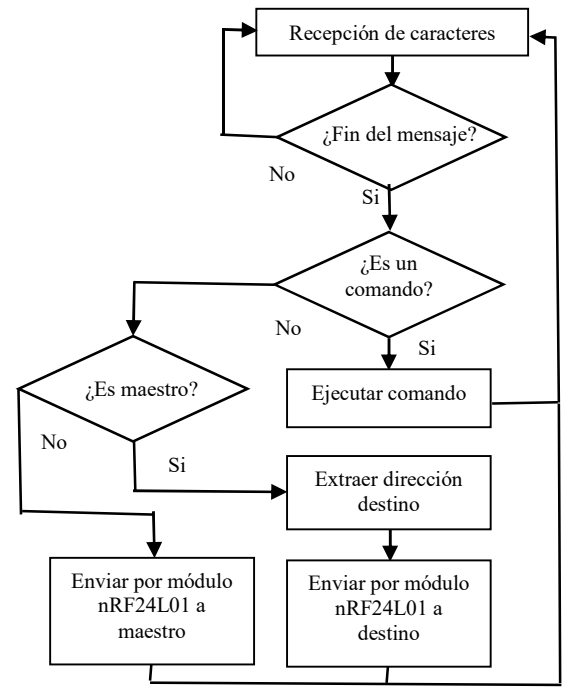


Fig. 6. Diagrama de flujo para la recepción de caracteres por RS232.

VII. CONCLUSIONES

En este trabajo se logró implementar una red local IoT multipropósito, versátil, de bajo costo y fácilmente configurable. La interfaz serie RS232 es ampliamente utilizada por muchos productos comerciales, por lo tanto es fácilmente adaptable a gran cantidad de situaciones donde productos que no fueron pensados originalmente para trabajar en red, con esta interfaz puedan ser utilizados de esta forma. También es posible armar redes rápidamente y utilizarla a través de comandos sencillos. Cabe destacar que la única configuración necesaria para poner en marcha la red es seleccionando la dirección de cada nodo por medio de un dip switch de 4 bits, que permite armar redes de hasta 16 nodos.

REFERENCIAS

- [1] ITU-T. (2023). Internet of Things Global Standards Initiative. Disponible en: <http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=y.2060>.
- [2] S. Pattar ; R. Buyya ; K. R. Venugopal ; S. S. Iyengar ; L. M. Patnaik. (2018). Searching for the IoT Resources: Fundamentals, Requirements, Comprehensive Review, and Future Directions. IEEE Communications Surveys & Tutorials. IEEE Communications Society. v. 20, n. 3.
- [3] Bluetooth. 2023. Sitio oficial: <https://www.bluetooth.com/>
- [4] ZigBee. 2023. Sitio oficial: <https://csa-iot.org/es/todas-las-soluciones/ZigBee/>
- [5] LoRa Alliance. 2023. Sitio oficial: <https://lora-alliance.org/>
- [6] Nordic Semiconductors. 2023. Sitio oficial <https://www.nordicsemi.com/Products/nRF24-series>
- [7] H. Xingna; M. Jun; Ch. Shouhong; T. Daiyu1. (2018) Design of data collection box based on NRF24L01. 2018 International Conference on Smart Materials, Intelligent Manufacturing and Automation (SMIMA 2018). MATEC Web Conference. Modeling, Analysis, and Simulation of Intelligent Manufacturing Processes. V. 173. p. 4.
- [8] Arduino Nano. 2023. Sitio oficial: <https://store-usa.arduino.cc/products/arduino-nano?selectedStore=us>
- [9] Github. 2023. Sitio oficial: <https://github.com/nRF24/RF24Mesh>

Categoría

# FORO TECNOLÓGICO

Robótica

Área Temática



# Differential-drive mobile robot controller with ROS 2 support

Gustavo Albarrán, Juan Nicolodi, Dante Ruiz, Diego González-Dondo and Gonzalo Perez-Paina

*Research Center in Informatics for Engineering (CIII)*

*Córdoba Regional Faculty, National University of Technology (UTN-FRC)*

Email: {dalbarran,druiz,dgonzalezdondo,gperez}@frc.utn.edu.ar

**Abstract**—Autonomous Mobile Robots (AMRs) are used in the internal logistics of many types of industries and production sectors. They help to reduce labor costs, enhance worker safety, increase productivity, and improve product quality. Many of the AMRs available in the market use the Robot Operating System (ROS), which is used for the development of their autonomous navigation algorithms. ROS is a set of free and open-source software libraries and tools for building robot applications. Its new version, ROS 2, was developed to be applied to production environments. This paper describes the development of a new controller for a differential-drive AMR with support for ROS 2 through its implementation for embedded system, micro-ROS. This controller is the evolution of a previous version that was used in different mobile robots for over 10 years. The design requirements, the main characteristics, and the improvements with respect to the previous version are described. Finally, the development of initial software focused on the verification of the correct operation of the controller is presented.

**Index Terms**—autonomous mobile robot, differential drive, ROS 2, micro-ROS

## I. INTRODUCTION

Automated Guided Vehicles (AGVs) are a key component in internal logistics in many types of industries that implement flexible manufacturing methods due to their ability to move products and parts efficiently in terms of handling manipulation time [1]. Currently, the navigation system of most AGVs is based on magnetic sensors for path following and laser range-finder for safety [2]. They have the limitation that the paths to follow are previously defined lacking the ability to choose a different one [3].

Recently, there have been significant advances in autonomous vehicles and their application as a service robotics platform [4], also known as Autonomous Mobile Robots (AMRs) [5], mainly focused on promoting flexibility within factories and boosting the Industry 4.0. AMRs are more flexible than AGVs, because they are reprogrammable, safe and collaborative, which is achieved by incorporating technologies reserved until a few years ago for the research field, such as [6]: map generation, computer vision, Simultaneous Localization and Mapping (SLAM) technology, 3D sensors, among others. Many of these technologies are implemented using the Robotics Operating System (ROS).

ROS is a set of free and open-source software libraries and tools for building robot applications [7], [8]. Although the first version of ROS solved many of the problems inherent to autonomous robots, it was not developed as software for production environments. For example, it has problems when working with non-ideal networks (poor quality WiFi connections), it has a main point of failure i.e. rosmaster (ROS

component that makes possible other nodes to communicate with each other), it does not include mechanisms for network security, and it does not have native support for embedded systems, among others.

Taking these limitations into account, ROS 2 emerges [9], [10], which consider other use cases such as: (a) teams of multiple robots, (b) application in embedded systems, (c) real-time systems, (d) non-ideal communication networks, and (e) production environments. ROS 2 is based on a Data Distribution Service (DDS) which is an open standard for communications in critical infrastructures [11]. DDS provides ROS 2 with support for network security, embedded and real-time systems, and operations in non-ideal networks.

Taking into account that the typical architecture of an AMR includes a network of one or more medium/high performance processors together with a set of microcontrollers ( $\mu C$ ), the use cases (b) and (c) considered by ROS 2 are of great relevance. The  $\mu C$ s are generally used to access sensors and actuators, for low latency control and security functions, etc. This is how the micro-ROS [12] project arises, which makes the ROS 2 code portable to a  $\mu C$ . The micro-ROS includes a layer with the open-source implementation of eProsima's XRCE-DDS (Extremely Resource Constrained Environments) standard, called Micro XRCE-DDS. For the interconnection between micro-ROS and ROS 2, an application known as Agent serves as a bridge between DDS and XRCE-DDS.

The present work describes the design of a new control board for a differential-drive mobile robot compatible with ROS 2 through micro-ROS, see Fig.1. The project has a

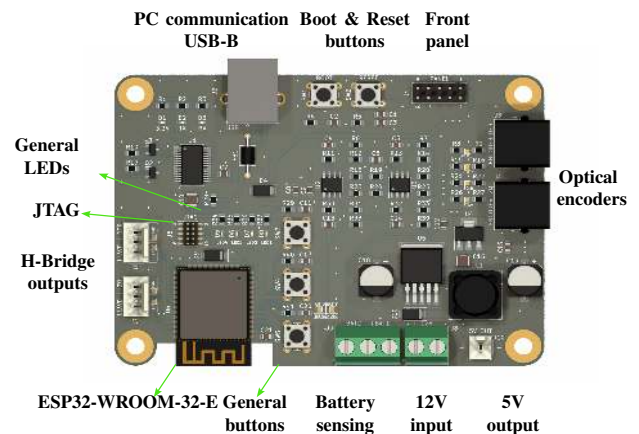


Fig. 1: New design of the differential-drive robot controller (DDRC-ESP32) board with ROS 2 support.

free license and is available in a public repository<sup>1</sup>. This development was carried out as the final project of the Electronic Engineering course of studies at the Córdoba Regional Faculty of the National Technological University [13] and the main novelty compared to the previous version [14] is the native support for ROS 2 through micro-ROS as well as other new features that will be explained later. To the best of our knowledge this is the only open hardware differential-drive robot controller board based on the ESP32 SoC with ROS 2 support.

The purpose of having an embedded controller with support for micro-ROS is to serve as the basis for the development of AMR type robots where their autonomous navigation and control algorithms are fully developed using ROS 2. Although the development was mainly focused on the hardware design, some programs developed to verify the correct operation of the board will be also presented.

## II. GENERAL DESCRIPTION

### A. Differential-drive robot

Fig. 2 shows the block diagram of a differential-drive robot. This type of robot has two drive wheels independently controlled to generate the desired movement, together with one or more non-drive wheels that serve as support. The geared motors (denoted as GM in the figure) and the incremental optical encoders used to measure the speed of rotation of each wheel can also be observed. The differential-drive robot controller operates in conjunction with two H-bridge power boards for the excitation of each of the motors.

The diagram also shows the energy system made up of batteries and the onboard computer to process the information from the sensors and execute the robot's autonomous navigation algorithms.

### B. Differential-drive robot controller

The main functions of the robot controller are:

- Reading the information from the optical encoders coupled to the drive wheels.
- Adjusting the speeds of the drive motors through PID controllers.
- Calculating the odometry with the information from the optical encoders coupled to the wheels.
- Communicating with the robot onboard PC.

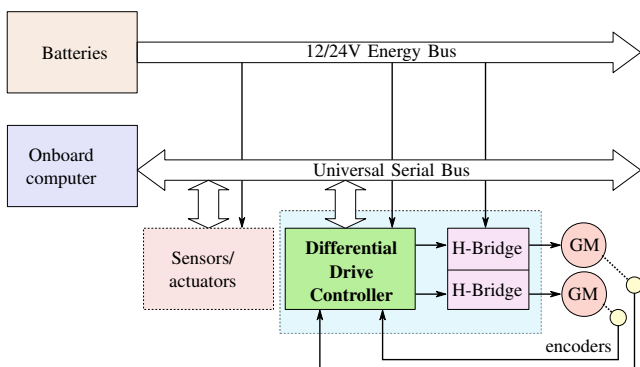


Fig. 2: Block diagram of a differential-drive robot.

<sup>1</sup>[https://github.com/ciiutnfr/ddrc\\_esp32](https://github.com/ciiutnfr/ddrc_esp32)

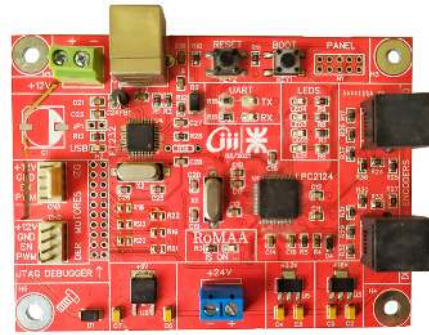


Fig. 3: Old version of the differential-drive robot controller based on a  $\mu\text{C}$  without ROS 2 support.

As mentioned, the new controller is the evolution of a previous design (see Fig.3) that was mainly used in the RoMAA-II robot [15]. It was also used in an AMR developed through a specific agreement between the UTN-FRC and the Caima-Segal S.R.L. [16], which has been operating in the Denso Manufacturing company in the city of Córdoba since the beginning of the year 2020. Both robots use ROS through a node driver developed by Perez Paina et al. [17]. Fig. 4 shows the new design of the differential-drive robot controller (DDRC-ESP32) mounted on the RoMAA-II robot.

## III. CONTROLLER DEVELOPMENT

### A. Hardware

The main requirements for the  $\mu\text{C}$  for the new design were:

- Support micro-ROS.
- Ability to drive two PWM outputs.
- Read pulses for the incremental optical encoders.
- Analog to digital converter (ADC).

In addition, another requirement was to maintain the same board size, connectors type and layout, to enable its direct replacement.

The micro-ROS is supported by some mid-range 32-bit microcontroller families and the official website provides a list of different development boards supported. From this list it was determined that the Espressif family is suitable for this application; in addition, it is available at the local market.

The selected model that meets the mentioned characteristics is the ESP32-WROOM-32E module (see Fig. 5), which has a System on Chip (SoC) model ESP32-D0WD-V3.



Fig. 4: New differential-drive robot controller board (DDRC-ESP32) mounted on the RoMAA-II robot.

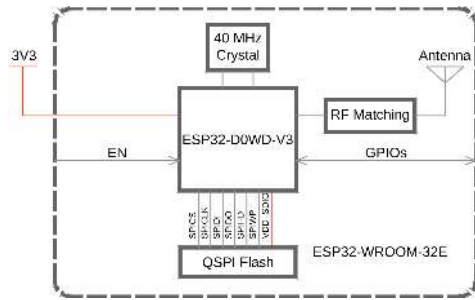


Fig. 5: Block diagram of used module, ESP32-WROOM-32E.

The main features of the module are:

- $\mu$ C Xtensa dual-core 32-bit LX6.
- Peripherals such as UART, SPI, I<sup>2</sup>C, PWM Motor, Pulse Counter, GPIO, ADC, etc.
- Communication via WiFi and Bluetooth with an integrated antenna.

The new design of the differential-drive robot controller includes the following improvements compared to the previous version [14], [17]: (i) switching power supply, (ii) conditioning circuit for measuring the robot's battery, (iii) general purpose inputs and outputs, and (iv) possibility of wireless communication via WiFi.

In summary, the main characteristics of the controller, which can be seen in Fig. 1 are:

- Inputs for incremental optical encoders (x2).
- PWM signal outputs (x2).
- Type-B USB connector for flashing the  $\mu$ C and communication with the PC.
- Reset and boot buttons.
- 12V DC power input.
- 5V DC regulated output.
- Input for 12V or 24V battery voltage measurement.
- General purpose buttons and LEDs.
- JTAG connector for program debugging.
- Front panel connector that extends the reset and boot buttons and signaling LEDs.

## B. Software

Software development is mainly focused on checking the correct functioning of the different parts that make up the differential-drive robot controller.

In the project repository there are: (i) a test program that acts as a web server which allows interaction with the board (reading optical encoders, actuation on motors and LEDs on the board, etc.) and (ii) tutorials for installing ESP32 Software Development Kit (SDK) and the steps to build and flash an example node using micro-ROS.

## IV. CONCLUSIONS AND FUTURE WORK

As a final result of the development, the hardware of a new controller for differential-drive mobile robots based on a modern System On Chip (SoC) that has support for ROS 2 through its version for embedded systems, micro-ROS, was obtained. This board is an evolution of the previous version and presents some important improvements. The development started from the design requirements. The components were selected, and the board was designed, manufactured, and

evaluated for correct operation. For the latter, some test applications of the Expressif ESP32 SoC that the controller has been developed.

Future work will include the development of applications for the evaluation and characterization of the operation of each component of the controller in an independent way, such as wheel speed measurement using the incremental optical encoders, and battery voltage measurement, among others. The goal is to finally develop a fully integrated application for the control of a robot based on micro-ROS, in order that those robots that use the controller have direct support for ROS 2.

## ACKNOWLEDGMENTS

This work is funded by the National University of Technology under the grant UTN-PID 8477, "Navigation and control of an AMR-type industrial mobile robot based on ROS 2".

## REFERENCES

- [1] G. Ullrich, *Automated Guided Vehicle Systems: A Primer with Practical Applications*. Springer Publishing Company, Incorporated, 2014.
- [2] L. Lynch, T. Newe, J. Clifford, J. Coleman, J. Walsh, and D. Toal, "Automated ground vehicle (AGV) and sensor technologies- a review," in *2018 12th International Conference on Sensing Technology (ICST)*, 2018, pp. 347–352.
- [3] F. Gul, S. S. N. Alhady, and W. Rahiman, "A review of controller approach for autonomous guided vehicle system," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, pp. 552–562, oct 2020. [Online]. Available: <https://doi.org/10.11591/ijeecs.v20.i1.pp552-562>
- [4] IFR, "Service robots," <https://ifr.org/service-robots>.
- [5] C. Cronin, A. Conway, and J. Walsh, "State-of-the-art review of autonomous intelligent vehicles (AIV) technologies for the automotive and manufacturing industry," in *2019 30th Irish Signals and Systems Conference (ISSC)*, 2019, pp. 1–6.
- [6] L. Lynch, F. McGuinness, J. Clifford, M. Rao, J. Walsh, D. Toal, and T. Newe, "Integration of autonomous intelligent vehicles into manufacturing environments: Challenges," *Proc. Manufacturing*, vol. 38, pp. 1683–1690, 2019.
- [7] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source Robot Operating System," in *Workshops at the IEEE International Conference on Robotics and Automation*, 2009.
- [8] R. D. Merrifield, A. Deguet, and G. Yang, "Powering the world's robots - 10 years of ROS," *Sci. Robotics*, vol. 2, no. 11, 2017. [Online]. Available: <https://doi.org/10.1126/scirobotics.aar1868>
- [9] B. Gerkey, "Why ROS 2?" [https://design.ros2.org/articles/why\\_ros2.html](https://design.ros2.org/articles/why_ros2.html), 2022.
- [10] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, may 2022.
- [11] D. Thomas, W. Woodall, and E. Fernandez, "Next-generation ROS: Building on DDS," in *ROSCon Chicago 2014*. Mountain View, CA: Open Robotics, sep 2014.
- [12] K. Belsare, A. C. Rodriguez, P. G. Sánchez, J. Hierro, T. Kołcon, R. Lange, I. Lütkebohle, A. Malki, J. M. Losa, F. Melendez, M. M. Rodriguez, A. Nordmann, J. Staschulat, and J. von Mendel, *Micro-ROS*. Springer International Publishing, 2023, pp. 3–55.
- [13] G. Albarrán, D. Ruiz, and J. Nicolodi, "Controlador Embebido para Robot de Tracción Diferencial," Universidad Tecnológica Nacional - Facultad Regional Córdoba, Tech. Rep., 2022.
- [14] M. Baudino and S. Pérez, "Hardware de Control de Plataforma Robótica Móvil con Arquitectura ARM y RTOS. Caracterización." Universidad Tecnológica Nacional - Facultad Regional Córdoba, Tech. Rep., 2010.
- [15] G. Perez Paina, G. Araguás, D. Gaydou, G. Steiner, and L. Rafael Canali, "RoMAA-II, an open architecture mobile robot," *Latin America Transactions, IEEE*, vol. 12, no. 5, pp. 915–921, Aug 2014.
- [16] Convenio de transferencia tecnológica: CIII-UTN-FRC y Caima Segall S.R.L., "Robot Móvil Autónomo para Transporte de Partes en Logística Interna de Planta DENSO Manufacturing Argentina," 2020.
- [17] G. Perez-Paina, D. Gaydou, and G. Araguás, "Driver de ROS para el robot móvil RoMAA," in *Proceedings of the X Jornadas Argentinas de Robótica (JAR)*, 2019.

Categoría

**FORO TECNOLÓGICO**

Software Embebido

Área Temática

# Sistema de comprobación en tiempo real de señales del autopiloto de una aeronave.

Arlet Montiel Ramos  
Departamento de Automática  
CIDNAV  
La Habana, Cuba  
montielarlet94@gmail.com

Alian Ernesto Matos Rodríguez  
Departamento de Automática  
CIDNAV  
La Habana, Cuba  
aematos@nauta.cu

Bárbaro Rodríguez Cárdenas  
Departamento de Automática  
CIDNAV  
La Habana, Cuba  
aid171alt@gmail.com

**Resumen**— Las avanzadas técnicas de visualización permiten que la información relevante se presente en el formato y en el momento adecuado por medio de interfaces visuales para que las operaciones se ejecuten de forma eficiente, al prevenir errores y malas interpretaciones que podrían conducir a situaciones de inseguridad o a fallos en los equipos. Estas constituyen una de las técnicas más relevantes dentro del campo de la informática en muchos casos empleando hardware especializados como son las Computadoras de Placas Reducidas. La principal contribución de este trabajo es un software capaz de comprobar y representar el estado técnico del autopiloto de una aeronave con el enlace sincronizado entre la interfaz hombre-máquina desarrollada y el autopiloto empleando LattePanda V1 que garantiza alta confiabilidad y bajo costo.

**Palabras claves**—interfaz hombre-máquina, LattePanda V1, sincronización de datos, UART.

## I. INTRODUCCIÓN

La interfaz de usuario es el medio por el cual las personas interactúan con las máquinas, estas han evolucionado mucho desde la inicial consola de comandos (CLI, por sus siglas en inglés), en la que los programas se utilizaban introduciendo órdenes mediante el teclado [1]. Con el devenir de los años dichas interfaces fueron progresando hasta llegar a convertirse en las Interfaces Hombre-Máquina (HMI, por sus siglas en inglés) que hoy comúnmente se emplean.

Las HMI se utilizan en varias esferas tanto en el ámbito militar como civil[2],[3],[4]. El HMI se soporta sobre diferentes hardware en dependencia de las características en donde se vayan a emplear, entre estos podemos encontrar los SBC[5]. Estos presentan una estructura de tamaño reducido por sus niveles de integración y reducción de componentes y tiene todo lo que necesita en la placa base para su empleo en entornos industriales y otros [6], [7].

Producto de la necesidad de comprobar la información de los procesos, sensores y de recopilar datos en tiempo real del autopiloto de una aeronave, que presenta déficit de piezas de repuesto e inestabilidad en su correcto funcionamiento, se propone como solución un sistema de comprobación que permite lograr la validación del autopiloto para su correcto funcionamiento empleando un cómputo de placa reducida.

## II. DESARROLLO

El software de comprobación constituye una herramienta de certificación encargada de determinar el estado técnico del autopiloto; para ello se emplea una interfaz gráfica, el autopiloto a comprobar y el hardware de placa reducida LattePanda V1.

### A. LattePanda V1

Estamos en tiempos donde los mini-ordenadores están de expansión en que podemos encontrar decenas de dispositivos en el mercado como Orange Pi PC, Raspberry pi, Odroid, Banana pi y Matrix ARM entre otros, a los cuales se suma LattePanda.

Conociendo las características de los diferentes SBC mencionados anteriormente se decide utilizar el LattePanda V1. En la Fig.1 se muestra un SBC LattePanda V1 el cual es una microcomputadora que dispone de un hardware robusto en comparación al resto de tarjetas de su misma clase, la placa es capaz de trabajar con sistemas operativos al nivel de Windows 10 y de ejecutar cualquier tipo de programas, gracias al uso de potentes procesadores que garantizan una buena calidad al momento de ejecutar varias tareas de forma simultánea[8], [9] por esto la necesidad de su empleo.



Fig. 1. Placa del LattePanda[10].

### B. Autopiloto y HMI

Se trabaja sobre un autopiloto de diseño propio del centro de trabajo. El mismo cuenta con cinco sensores analógicos (dos acelerómetros, dos giróscopos de amortiguamiento y un sensor de ángulo), un computador a bordo que funciona como sistema de control del vuelo y una fuente conmutada que brinda energía al autopiloto. Con el fin de establecer una comunicación entre el autopiloto y el SBC, se realiza una conexión inalámbrica a través de radio modem logrando la transmisión de los datos hacia el software desarrollado.

A partir de la carencia de un sistema automatizado que permitiera sustituir el engorroso trabajo de las comprobaciones que se realizaban de forma manual, provocando malas interpretaciones que conducían a situaciones de inseguridad, surge la necesidad de comprobar sus sensores con el empleo de una interfaz visual que mostrara en tiempo real los parámetros del autopiloto dando significado y contexto a su estado técnico.

Para ello se le realizan varias pruebas: determinar su cero eléctrico y mecánico, comprobación de sus giróscopos de amortiguamiento en las distintas etapas del vuelo y otras más. En la Fig. 3 se muestra la interfaz principal donde se refleja los parámetros de comprobaciones.

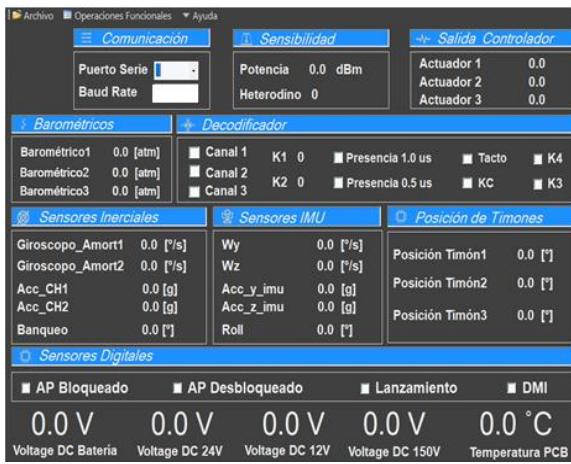


Fig. 3. Ventana principal del sistema de comprobación.

### C. Implementación del software

El software se desarrolló en el entorno de desarrollo de Visual Studio por lo que permitió editar, compilar y depurar el código en el lenguaje C# con sistema operativo Windows 10.

La comunicación del SBC con el autopiloto es mediante el protocolo de comunicación Transmisor-Receptor Asíncrono Universal (UART, por sus siglas en inglés) con la norma RS 485, el cual es muy popular debido a su capacidad para comunicarse a largas distancias, su capacidad multipunto y su inmunidad al ruido[11].

En la Fig. 4 se define la estructura de la trama de datos compuesta por cabecera, longitud, datos a procesar y CheckSum, la misma cuenta con 176 bytes que se transmiten cada 50 milisegundos a 115200 baudios lo que significa que es capaz de recibir y transmitir un máximo de 115200 bits por segundo.

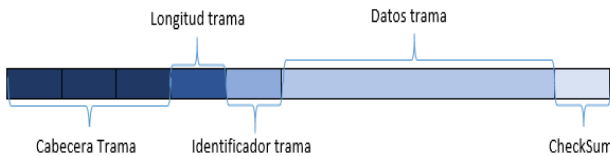


Fig. 4. Estructura de la trama de la Recepción datos y de Transmisión.

Se realiza una transmisión de datos por cada comprobación logrando una comunicación entre la aplicación y el dispositivo a comprobar. Dicha trama consta de: cabecera, la cual identifica si la trama es la indicada junto con la longitud y el identificador de la trama, así como identificador (ID) de tipo de comprobación, ID de inicio de comprobación, ID de operación y CheckSum para detectar pérdidas en la transmisión de datos con el fin de proteger la integridad de la información.

Concebida la trama de datos la implementación se organiza en cinco etapas (Recepción, Procesamiento, Visualización, Graficar y Comprobaciones de Datos), como se demuestra en la Fig. 5. Primeramente los valores son recibidos donde se analizan si corresponden con la estructura definida empleando varios métodos de la biblioteca de clases *SerialPort* como *ReadByte*, *DiscardInBuffer* entre otros, para luego procesar los datos facilitando la identificación de cada sensor del autopiloto, y luego ser representados y graficados diferentes sensores o realizar comprobaciones autónomas o combinadas.

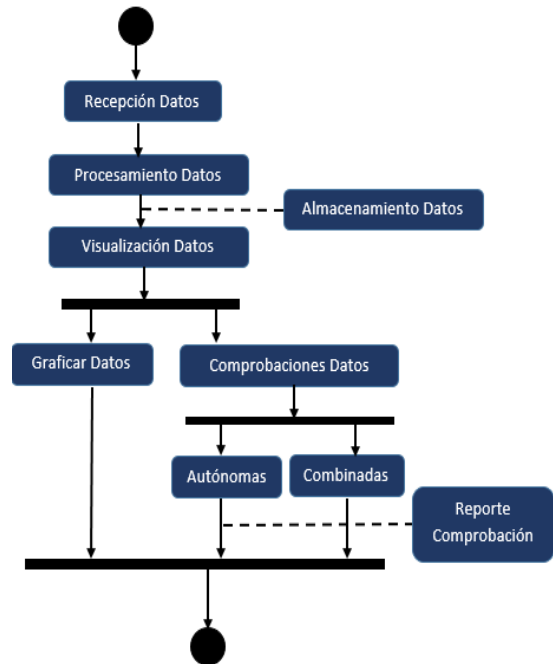


Fig. 5. Diagrama de Secuencia del Software de Comprobación.

### D. Programación con Hilo

El elevado procesamiento de datos hace necesaria la implementación de un método de programación paralela para acelerar la ejecución de la función que más tiempo consume, evitando que se dé una colisión de datos y logrando una distribución de tareas entre los núcleos del LattePanda V1.

Se puede hacer paralelismo al ejecutar aplicaciones en diferentes y concurrentes hilos de ejecución. Con la herramienta de Visual Studio se pudo determinar que el proceso de recepción de datos es el método que más tiempo consume alcanzando los 3.5 milisegundos, por tanto se incluyó dentro de la biblioteca de clase *Thread* lo que permite recibir en el mismo período que se procesan los datos llegando a reducir el tiempo de recepción a 1.8 milisegundos. Además fue necesario agregar otro hilo al software teniendo en cuenta que el proceso de graficado de parámetros y comprobaciones necesita actualizar y representar gráficamente determinados valores.

Es por ello que las aplicaciones que usan *multithreading* responden mejor a la entrada del usuario, ya que la interfaz de usuario permanece activa mientras las tareas que requieren un uso intensivo del procesador se ejecutan en subprocesos separados[12], además para evitar los problemas típicos de la programación *multithreading* de desincronización y/o comunicación se empleó una comunicación asincrónica resolviendo los problemas que pudieran surgir al esperar un tiempo mayor antes de recibir la trama siguiente porque el transmisor garantiza la velocidad de transmisión constante.

## III. EXPERIMENTOS REALIZADOS

Con el objetivo de verificar que el software desarrollado cumple con las exigencias de tiempo planteado y determinar la configuración óptima del rendimiento se realizan pruebas de experimentos bajo el principio de repetición tomando como factores controlable: afinidad del procesador y prioridad del proceso, para ello se tomaron 1200 muestras aproximadamente durante un minuto y se realizaron las

pruebas para prioridad por encima de lo normal para dos procesadores (X01) y cuatro procesadores (X02), prioridad por debajo de lo normal para dos procesadores (X03) y cuatro procesadores (X04). Como resultado se pudo determinar que la configuración óptima donde el software obtiene mejores resultados se realiza con prioridad por encima de lo normal y con afinidad de cuatro procesadores llegando a reducir hasta 2.4 milisegundos aproximadamente el tiempo de ejecución de sus procesos como se muestra en la Fig. 6 y se muestra una disminución de la distribución del tiempo de la desviación estándar para X02 demostrando que es la configuración ideal para un rendimiento eficaz del funcionamiento del software.

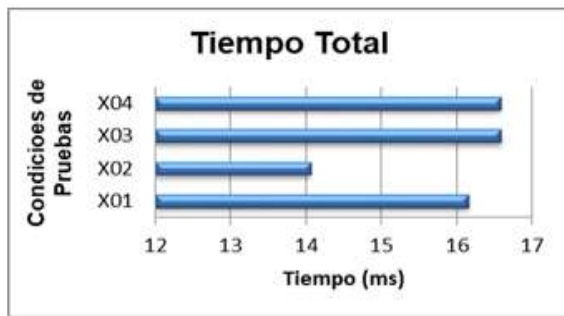


Fig. 6. Gráfica de la media del tiempo de trabajo en las diferentes condiciones de configuración.

En la Fig. 7 se representa el comportamiento de las tramas recibidas con respecto al tiempo total de la recepción obtenida en experimentos donde se tomaron 7524 muestras para 50 milisegundos de muestreo lo que indica 376.2 segundos de vuelo.

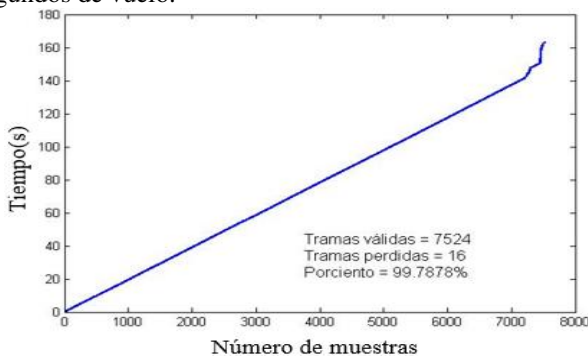


Fig. 7. Gráfica de la variable de tiempo de trabajo del autopiloto enlazado con el supervisor de comprobación.

Como resultado del muestreo realizado se obtiene una sincronización satisfactoria de 7524 tramas de las cuales se perdieron 16, lo que representa un 99% de las tramas válidas para una evaluación con profundidad de los resultados alcanzados.

#### IV. CONCLUSIONES

El software desarrollado con el empleo del cómputo de placa reducida LattePanda V1 permitió obtener una solución para la representación y comprobación de los sensores del autopiloto de una aeronave, el cual fue validado con mediciones reales mediante experimentación determinando

la influencia de los factores de programación y dispositivo en los tiempos de respuesta del sistema.

#### REFERENCIAS

- [1] Fundacionayesa. (2015). *Evolución hacia las interfaces naturales de usuario*. [En línea]. Available: [www.fundacionayesa.org/evolucion-hacia-las-interfaces-naturales-de-usuario/#:~:text=La%20interfaz%20de%20usuario%20es,sigue%20utilizando%20para%20determinadas%20tareass](http://www.fundacionayesa.org/evolucion-hacia-las-interfaces-naturales-de-usuario/#:~:text=La%20interfaz%20de%20usuario%20es,sigue%20utilizando%20para%20determinadas%20tareass)
- [2] D. L. Haro Cuadrado, "Sistema de Control con Interfaz HMI para el Proceso de Enfriamiento de las unidades de Generación de la Hidroeléctrica TOPO," Quito, Ecuador: Universidad Tecnológica Israel, 2022.
- [3] O. F. Matilla Buenapósada, "Puesta en marcha y control de un horno industrial usando un autómata programable y una pantalla HMI," 2022.
- [4] Mordorintelligence, "Mercado de Interfaz Hombre-Máquina de Europa: Crecimiento, Tendencias y pronósticos (2023 - 2028)," 2023.
- [5] S. R. Torres Gualsaqui, "Implementación de un sistema HMI mediante aplicaciones de código abierto para el control y monitoreo de un sistema dinámico real," 2021.
- [6] A. E. M. Rodríguez, L. Z. Méndez, R. S. Lima, and H. D. Pando, "Computación paralela y computadoras de placa única para aplicaciones de radar," *Serie Científica de la Universidad de las Ciencias Informáticas*, vol. 16, pp. 32-44, 2023.
- [7] P. J. Bonilla Vásquez, "Diseño de sistemas de control industrial para manipuladores basados en Industria 4.0," Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas ..., 2020.
- [8] J. J. F. P. C. A. N. Nacipucha, "Diseño de un Prototipo de Control de Acceso a través de Reconocimiento Facial Mediante la Creación de un Algoritmo Basado en Software Libre Utilizando LattePanda.," Universidad Politécnica Salesiana del Ecuador Sede Guayaquil, 2020.
- [9] LattePanda. (2023). *LattePanda V1* [Page Web]. Available: [https://docs.lattepanda.com/content/1st\\_edition/power\\_on/](https://docs.lattepanda.com/content/1st_edition/power_on/)
- [10] (2016). *LattePanda, un nuevo competidor para Raspberry Pi por tan solo 60 euros* [En línea]. Available: <https://linux.com/lattepanda-un-nuevo-competidor-para-raspberry-pi-por-tan-solo-60-euros/>
- [11] F. Marín Hernández, "Diseño e implementación de un tablero electrónico para la comunicación de posición absoluta," 2022.
- [12] C. d. Autores. (2023). *Programación en paralelo en .NET: guía de la documentación*. [En Línea]. Available: <https://learn.microsoft.com/es-es/dotnet/standard/parallel-programming/>

# Implementación de la técnica del Modelo Basado en Diseño para el desarrollo de Estación Receptora ADS-B

Damián Héctor PRIMO, Darío Walter DIAZ,  
Francisco Martín ESCOBAR  
Grupo de Sistemas de Tiempo Real  
Universidad Nacional Río Cuarto  
Río Cuarto, Córdoba, Arg.  
dprimo@ing.unrc.edu.ar

Diego Daniel BADINO, Juan Octavio  
OVIEDO, Juan Pablo RUMIE VITTAR  
Centro I+D de Tec. Aeronáuticas  
Fuerza Aérea Argentina  
Las Higuerras, Córdoba, Arg.  
dd\_badino@faa.mil.ar

**Resumen**—En el presente trabajo se expone el análisis, diseño, desarrollo e implementación de un sistema de adquisición de datos provenientes de transponder de aeronaves con tecnología ADS-B (Automatic Dependent Surveillance Broadcast) utilizando la técnica para su desarrollo denominada modelo basado en diseño (MDB, Model Based Design). Se determinan las etapas más importantes que se deben seguir para desarrollar un sistema que se basa en tecnología SDR (Software Defined Radio): planificación, simulación, producción y confirmación de código. Se usará el sistema receptor ADS-B para demostrar el uso de la metodología propuesta, con el fin de integrar las fases de desarrollo en un proceso de diseño continuo. Posteriormente, se analizarán conceptos de diseño avanzados, como la simulación, la producción de código y las pruebas en hardware.

**Palabras Claves**—SDR, ADS-B, MDB, Transponder, TCA.

## I. INTRODUCCIÓN

En el marco de un proyecto de investigación propuesto por XXXXXX XXXXX XXXXXXXXXXXX en conjunto con la XXXXXXXXXXXX XXXXXXXXXXXX xx XXXXXXXXXXXXXXXXXXXX x XXXXXXXXXXXX mancomunados en proyectos I+D, y el XXXXX xx XXXXXXXX xx XXXXXXX XXXX (XXXX), pertenecientes a la XXXXXXXXXXXX xx XXXXXXXXXXXX xx xx XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXX xx XXX XXXXXXX (XXXX) surge la necesidad de desarrollar un sistema de estaciones receptoras de señales ADS-B interconectadas que permitan recibir, procesar y almacenar información del tráfico aéreo de la zona de cobertura.

Como una de las ventajas, el desarrollo de los sistemas receptores para utilización en Tierra resulta mucho más económico que otros sistemas de vigilancia como radares primarios o secundarios debido a su menor complejidad [1][2].

En los últimos 10 años todas las aeronaves ya cuentan con esta tecnología en la que envían datos, entre otros, de identificación, de posición, de velocidad en tres dimensiones y de estado de la aeronave en sus diferentes mensajes.

En un desarrollo tradicional, generalmente un ingeniero de sistemas define la especificación de requerimientos del sistema (SRS, por sus siglas en inglés de System Requirements Specification) y la presenta como un documento de diseño (SDD, por sus siglas en inglés de System Design Document) a los desarrolladores de software, quienes tendrán la tarea de implementar esas ideas o especificaciones en una solución totalmente funcional. Sin embargo, el principal problema con este enfoque es el hecho de que, en la mayoría de los casos, las ideas presentadas por el ingeniero del sistema a través del documento de especificación pueden diferir ampliamente del software implementado. Incluso el

tipo de documentación más detallada y diligentemente preparada no siempre puede garantizar que el documento de diseño generado por los ingenieros del sistema sea completamente comprendido con precisión e interpretado correctamente por los programadores de software que lo implementan. Para evitar estos inconvenientes, una alternativa es utilizar la técnica emergente para resolver este problema denominada "Diseño Basado en Modelos" (MBD, por sus siglas en inglés de Model Based Design) [3].

## II. MODELO BASADO EN DISEÑO

En los últimos años, la metodología de MDB se ha convertido en el método preferido para diseñar, modelar y simular sistemas dinámicos complejos [4][5]. A contraparte, la metodología tradicional de diseño de sistemas embebidos, que se muestra en la Fig. 1, insume bastante tiempo y es propensa a errores por naturaleza.

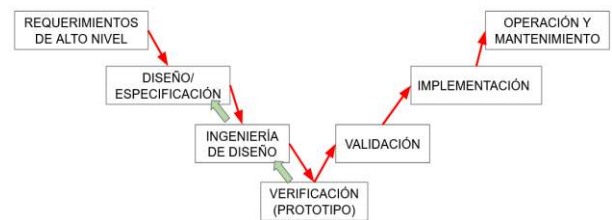


Fig. 1: Modelo tradicional para desarrollos de sistemas embebidos

En esta metodología MBD (Fig. 2), el diseño y desarrollo de un sistema y su introducción al mercado consta de 4 pasos definidos:

- **Diseño:** Se definen los requerimientos, se determinan los procedimientos, los requisitos de hardware, algoritmos y normas de referencia para cumplir con los requerimientos.
- **Implementación:** Se establece el lenguaje de programación a utilizar en función del hardware seleccionado que hacen al sistema. Se determina la herramienta visual para realizar el desarrollo de la solución.
- **Testing:** Se procede a realizar las pruebas del software y hardware para el análisis de los resultados durante el proceso de desarrollo del sistema. El modelo plantea una integración dinámica y continua durante el proceso de desarrollo.



- Producción: Completado los test de forma satisfactoria, se procede al desarrollo en serie, implantación y entrega del producto final.

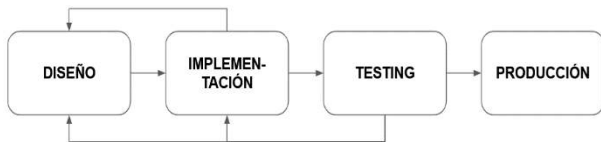


Fig. 2: Metodología de desarrollo aplicando el Modelo Basado en Diseño

La primera fase involucra la definición y diseño del sistema objetivo. Esta es la que normalmente llevan a cabo los que diseñan el sistema, quienes definen la especificación del sistema y la presentan como un documento a los diseñadores de software. Esta etapa puede incurrir en mayor riesgo, ya que es la mayor fuente de error, y diferentes desarrolladores de código pueden interpretar estas definiciones de manera diferente.

El siguiente paso es la parte de implementación. En esta etapa, los desarrolladores generarán el código de software adecuado en un lenguaje seleccionado para implementar el sistema. Para los sistemas integrados, el lenguaje de programación elegido para esta tarea es el lenguaje C para manejo de librerías y Python para procesamiento de tramas ADS-B que interactúan con el software GNU-RADIO [6]. Si las definiciones del sistema, por alguna razón, no son claras, los programadores se dirigirán al diseñador del sistema para obtener aclaraciones al respecto. Incluso después de estas aclaraciones, siempre es posible que el sistema desarrollado sea diferente del que tenían previsto los diseñadores del sistema.

Después del desarrollo del código, el siguiente paso es la prueba. Es muy común que el resultado de la prueba sea diferente de lo que los programadores o el diseñador del sistema tenían previsto según su especificación. Si las pruebas muestran que hay errores en el diseño del sistema, entonces el diseñador del sistema debe revisar su diseño y pasarlo a los desarrolladores de código para implementar las modificaciones pertinentes que resuelvan los problemas. Por supuesto, también puede darse el caso de que la implementación no se condice con la representación real de la especificación y, por lo tanto, los resultados de la prueba muestran errores.

En este caso, el programador debe asegurarse de comprender mejor la especificación y corregir la parte del código que muestra falla o simplemente depurar la aplicación. Cualquiera de estos procesos puede repetirse en forma reiterativa. Para complicar aún más la cuestión, también se puede considerar el caso en el que los procedimientos de prueba o la implementación son incorrectos y, por lo tanto, dan resultados erróneos. Posterior a las pruebas exhaustivas y solo cuando los diseñadores del sistema estén convencidos (conforme a los resultados de las pruebas) de que el sistema implementado funciona correctamente y según lo previsto y especificado, entonces puede comenzarse la producción. La complejidad de este largo proceso hace que la gestión de proyectos sea ineficaz y muy difícil de seguir.

Para mejorar el enfoque anterior y para evitar malentendidos, se definió el diseño del sistema basado en modelos. Este diseño basado en modelos es ahora un enfoque establecido para desarrollar soluciones eficientes a problemas de ingeniería complejos. En este enfoque, se pueden crear sistemas de carácter complejos mediante el uso de modelos

matemáticos que representan los componentes del sistema y sus interacciones con el entorno que los rodea.

Estos modelos tienen muchas aplicaciones en el proceso de diseño, incluida la simulación de sistemas, el análisis de estabilidad y el diseño de código de control. Mediante la introducción de tecnología avanzada de generación de código automatizada, se ha vuelto factible otra aplicación de estos modelos. Estos modelos se pueden utilizar como entrada para una herramienta de generación automática de código. Los generadores de código avanzados y de última generación pueden producir códigos fuente integrados y optimizados a partir de estos modelos.

El uso de la metodología MDB tiene como objetivo reducir el número de etapas de desarrollo al combinar las etapas de diseño, implementación y prueba en un solo proceso, como se muestra en la Fig. 2. La reducción de los pasos necesarios, en comparación con el método tradicional, y la complejidad darán como resultado una mejor gestión del proyecto y la mitigación de los riesgos de desarrollo de productos. Los resultados obtenidos con este enfoque llegan al mercado más rápido y terminan reduciendo los costos de desarrollo.

### III. DESARROLLO

#### A. Tecnología ADS-B

ADS-B es un tipo de tecnología de vigilancia basada en TCAS (Traffic Collision Avoidance System). La tecnología ADS-B tiene una mayor precisión, menos errores y una mayor capacidad de vigilancia que la tecnología de vigilancia radar tradicional.

El Sistema de Difusión de Vigilancia Dependiente Automática, es una tecnología implementada para rastrear aeronaves en forma cooperativa dado que las aeronaves poseen un transponder ADS-B que informan automáticamente su identificación y otros parámetros de la misma como ser posición, velocidad y altitud, tanto a otras aeronaves como a estaciones terrestres sin necesidad de interrogación [7].

#### B. Radio Definido por Software

La complejidad de las aplicaciones de radio definidas por software (SDR) y la dificultad de probar, depurar y validar dichos sistemas las hace particularmente adecuadas para el desarrollo con herramientas de diseño basadas en modelos como parte del proceso de diseño central. Para demostrar la viabilidad de esta metodología para el diseño, implementación y prueba de un sistema complejo, se utilizó la técnica de diseño de sistemas basada en modelos para implementar y probar un demodulador de señales ADS-B utilizando el hardware RTL2832.

#### C. Diseño: Receptor ADS-B en GNU-Radio

El demodulador fue diagramado por etapas según él y simulado en el entorno de GNU-RADIO[8], tomando como referencia lo establecido según norma aeronáutica DO-260A. El mismo se muestra en la Fig.3. Fue construido utilizando bloques estándar y drivers para el hardware SDR en cuestión utilizando una arquitectura de hardware x86 donde se ejecuta el código y con una interfaz SDR del modelo RTL.

El proceso del sistema es adquirir las señales en estándar 1090Mhz a través de la interfaz SDR y a nivel software se procesa, filtra a digital de la información, se identifica el preámbulo y cálculo del CRC para la validación de tramas que luego se utiliza en módulos posteriores para la obtención de los datos tal cual establece la norma.

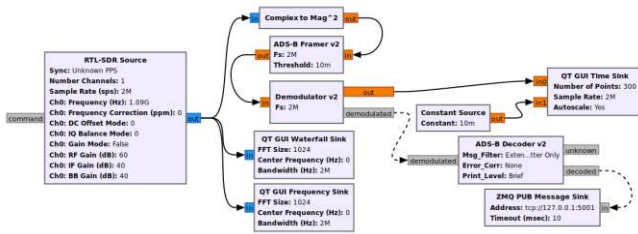


Fig. 3: Diseño del receptor con GNU-RADIO

Para probar la compatibilidad entre un modulador y el demodulador, se procedió a realizar varias capturas de diferentes vuelos en tiempo real en cercanías de la localidad de Río Cuarto, en donde son habituales los vuelos de aeronaves que transmiten su información con transponder ADS-B.

D. Implementación: Análisis de Datos Adquiridos

En la Figura 4 se muestra un ejemplo de captura utilizando el sistema desarrollado y utilizando como referencia el sitio web de flightradar24 [9]. En la misma se muestra el tráfico aéreo en tiempo real y podemos observar que los datos adquiridos y procesados coinciden como se muestra en el caso de la identificación de la aeronave.

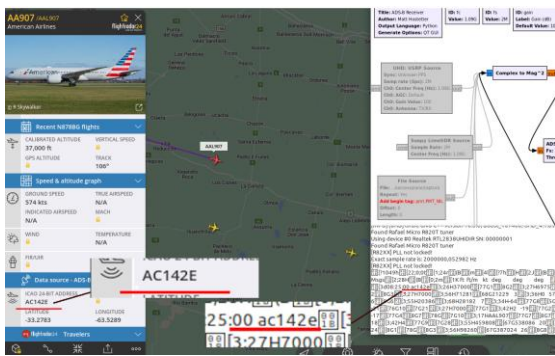


Fig. 4: Captura en tiempo real de vuelo en Flightradar24 y en GNU-RADIO

E. Integración: Sistema Prototipo Producción

A posterior de la etapa de Diseño se procede a la generación de código automática y la integración del mismo en el hardware final. En el proceso de generación, el software GNU Radio también establece las bibliotecas necesarias para los conjuntos de bloques utilizados en el modelo diseñado. Esta biblioteca creada es compatible con el hardware de destino RTL2832[10] y una OrangePI[11] con microprocesador ARM donde se ejecutará el mismo (Fig. 5).

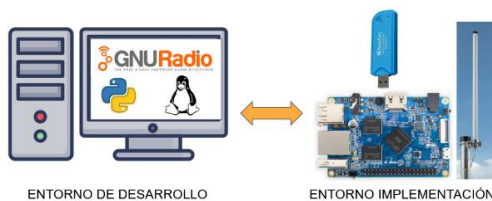


Fig. 5: Entorno de Desarrollo y entorno de Implementación

IV. CONCLUSIONES

Los resultados del desarrollo con la implementación del modelo basado en diseño demostraron que es factible reducir

el tiempo de desarrollo de sistemas embebidos, ya que utiliza un modelo de diseño por componentes para la implementación de los diferentes procesos requeridos para el procesamiento de la información ADS-B con la finalidad de describir el comportamiento del sistema. Esto permite a los diseñadores y desarrolladores trabajar de forma más eficiente y rápida. Además, realizar simulaciones y verificaciones exhaustivas del sistema antes de generar el código final (Release Candidate), lo que reduce claramente la probabilidad de errores y mejora la calidad del software.

La plataforma GNU-RADIO permitió generar código automático para embeber en el prototipo final de hardware donde se implementará el sistema receptor. Además quedó disponible un entorno, diseño y componentes de software que va a permitir realizar reutilización de código en otros desarrollos en el ambiente de investigación de la Fuerza Aérea Argentina. El impacto de esta técnica de diseño radica en el hecho de que se reduce el tiempo de desarrollo del producto, lo que tiene influencia directa en el precio del producto final y en su capacidad de venta.

Algo a destacar es que el producto desarrollado ya ha sido probado y validado en forma reiterada en la misma fase de desarrollo. Claramente es una técnica factible de adoptar para el desarrollo de sistemas embebidos en general.

REFERENCIAS

- [1] RTCA, "Minimum Operational Performance Standards (MOPS) for 1090 MHz Extended Squitter Automatic Dependent Surveillance - Broadcast (ADS-B) and Traffic Information Services - Broadcast (TISB)," Radio Technical Commission for Aeronautics, DO-260B 17, December 2011, pp. 13–122.
- [2] European Commission, "Commission implementing regulation (EU) No 1028/2014 of 26 September 2014," Official Journal of the European Union, 30, September 2014, pp. L284/7–L284/8.
- [3] MathWorks, Inc., Evolution of model-based design in aerospace. IML Group PLC. [online]. [cit. 2013-08-02], <http://www.epdtonthenet.net/article/41911/Evolution-of-modelbased-design-in-aerospace.aspx>.
- [4] X. Zhang, Y. Sun, L. Wang and D. Xu, "Product model-based design process modeling in collaborative design," 2010 IEEE International Conference on Industrial Engineering and Engineering Management, Macao, China, 2010, pp. 315-319, doi: 10.1109/IEEM.2010.5674307.
- [5] S. Perry, "Model Based Design needs high level synthesis - A collection of high level synthesis techniques to improve productivity and quality of results for model based electronic design," 2009 Design, Automation & Test in Europe Conference & Exhibition, Nice, France, 2009, pp. 1202-1207, doi: 10.1109/DATE.2009.5090845.
- [6] Cai, Xin & Zhou, Mingda & Huang, Xinming. (2017). Model-Based Design for Software Defined Radio on an FPGA. IEEE Access. PP. 1-1. 10.1109/ACCESS.2017.2692764.
- [7] D. Primo, F. Escobar, D. Badino and J. R. Vittar, "System acquisition, processing and visualization of Air Traffic Using ADS-B and IoT Connectivity," 2018 IEEE Biennial Congress of Argentina (ARGENCON), San Miguel de Tucuman, Argentina, 2018, pp. 1-5, doi: 10.1109/ARGENCON.2018.8646168.
- [8] GNU Radio Project. Accedido: 2023-04-21. [online] Available: <https://www.gnuradio.org/>.
- [9] Flightradar24, Accedido: 2023-02-11. [online] Available: <https://www.flightradar24.com/>.
- [10] RTL2832U, Accedido: 2023-02-11. [online] Available: <https://datasheetspdf.com/pdf/file/1493155/Realtek/RTL2832U/>.
- [11] OrangePi+, Accedido: 2023-02-11. [online] Available: <http://www.orangepi.org/orangepiplus2/>.

# Ecosistema de Interoperabilidad basado en C++/MicroPython sobre plataformas Raspberry y ESP32

Jorge Niño, Mg. Ing. Marcos Politi, Maximiliano Gulfo, Camilo Quiroga, Lucien Lucangioli, Dr. Hector Laiz  
*Instituto Nacional de Tecnología Industrial*  
 San Martín, Prov. de Bs. As.  
 Argentina  
 jnino@inti.gov.ar

**Resumen** — En el presente trabajo se describe una solución de bajo costo basada en el procesador ESP32 y un controlador LoRa SX1276 para control de inversores fotovoltaicos con interfaz WiFi, basados en protocolo HTTP/JSON tanto en lenguaje C++ como MicroPython. También se presenta una solución en Python para Raspberry. Para la prueba del firmware se empleará la plataforma HELTEC. Se desarrollan además el hardware del ecosistema con diversas funcionalidades y un firmware configurable para manejo de múltiples interfaces con los inversores FV.

**Palabras Clave**—LoRa, Raspberry, HTTP, JSON, ESP32, WiFi, Interoperabilidad, MicroPython, Python, IoE, IoT.

## I. INTRODUCCIÓN

Algunos modelos de inversores fotovoltaicos poseen interfaces WiFi, además de las típicas MODBUS [1], para poder acceder tanto a los datos de la operación del inversor como a los parámetros de configuración y operación del mismo. Se confeccionó una serie de equipos capaces de interactuar con el inversor en estos términos, tanto en una red local del mismo como con la red propia del inversor. Los sistemas desarrollados están basados en un microcontrolador de 32 bits de ESPRESSIF, el ESP32, que posee ya un controlador dedicado integrado para WiFi con el cual poder comunicarse hacia los inversores fotovoltaicos y en la plataforma Raspberry Pi modelo 3B+. Se implementaron dos versiones del firmware, en MicroPython y C++. Esta estrategia permite diversificar los equipos y la potencia de procesamiento, que da flexibilidad al despliegue de gran cantidad de equipos, en redes complejas. Por otro lado, como se señala en otros trabajos [2], para la interfaz entre nodos de una red distribuida y el concentrador o Gateway, es conveniente emplear un sistema de comunicación de largo alcance. El sistema de comunicación escogido es LoRa, basándose en un controlador SX1276. Para el propio Gateway se emplea tanto el ESP32 programado bajo el firmware en C++, como una Raspberry programada en Python conectada a un controlador SX1276.

## II. ESQUEMA GENERAL

### A. Topología de dispositivos

La topología de la red, descrita en la Fig. 1, muestra que se empleó la red local, a la cual se encuentra conectado el inversor, mientras que el concentrador o Gateway accederá al servidor AWS donde se encuentra la plataforma de smartgrid.ar. Por otro lado, se mantiene la conexión del equipo MODBUS-ETH para así poder administrar fuentes de diferentes datos.

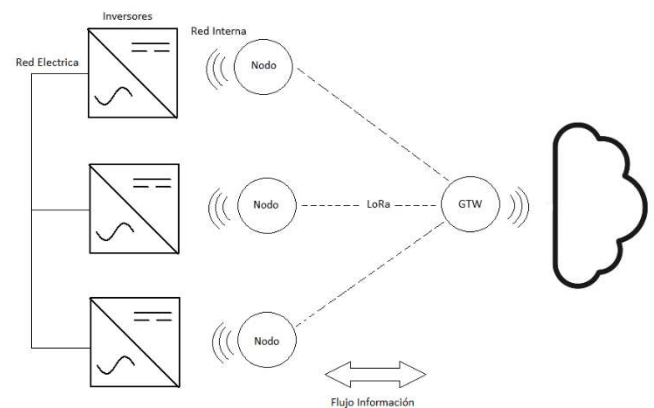


Fig. 1: Sistema de operación de inversores mediante Gateway-Nodo.

### B. Reverseibilidad y flexibilidad

La implementación de los nodos y Gateway WiFi-LoRa se realizará de tal manera que se pueda configurar sus funciones mediante una serie de macros, así se obtiene mayor flexibilidad y facilidad para su puesta a punto, además de independizar el código del hardware en cierta medida, ya que la plataforma en ambos dispositivos será la misma.

## III. HARDWARE DEL EQUIPO.

El hardware del dispositivo fue diseñado basado en un ESP32 como procesador central, un módulo LoRa RFM95 basado en el controlador SX1276. A continuación, se detallan los componentes de cada parte del circuito.

### A. Fuente de alimentación

Se empleó para la fuente de alimentación un regulador LM2576 [3], configurado para entregar 5 V desde una alimentación de rango 9-12 V externa. Además, se utilizan tres reguladores LM1117 para obtener alimentación de 3.3 V y así poder suministrar suficiente corriente para todo el sistema.

### B. Microcontrolador y Chip LoRa

Se conectó al módulo ESP32-WROOM, una serie de periféricos (conectores para sensores) y la interfaz SPI para el controlador LoRa. Por otro lado, se empleó el módulo RFM-95 para implementar el sub-circuito LoRa.

El hardware cambia en función de la modalidad del nodo (Fig. 2) para el caso del nodo implementado sobre MicroPython, se empleó el ESP32 conectado a un módulo externo, pero a través del mismo bus de datos. Mientras que la versión implementada sobre C++ se implementó directamente sobre el PCB diseñado, aunque nuevamente empleando la misma interfaz conceptual (bus SPI, interrupción de controlador, y sistema CS/RESET)

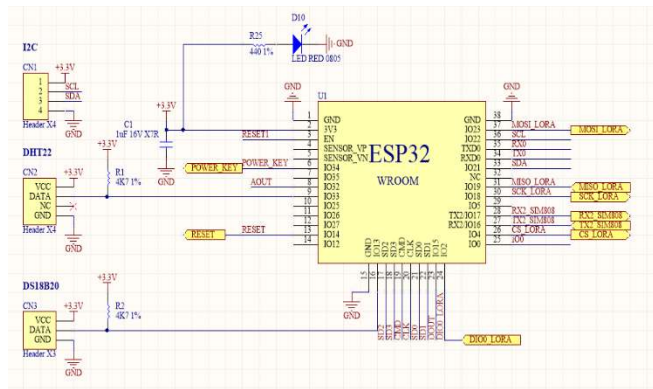


Fig. 2: Sub-circuito principal ESP32

### C. Ethernet y otros periféricos

Se desarrollaron varias versiones de PCB (Fig. 3), entre ellas, se implementó una versión ETH-LoRa y otra con GPS y LoRa, dejando un conector libre para conectar un módulo Ethernet. En el primer caso, el circuito de Ethernet consiste en el adaptador SPI-ETH W5100 [4].

Una de las versiones de PCB incluye un chip SIM808, que incorpora un controlador GSM/GPRS con un periférico GPS y Bluetooth lo cual dota al sistema de la capacidad de emplear conexiones satelitales y celulares. En esta versión se incluye asimismo, el circuito para la operación de la memoria SD necesaria para el SIM808 que emplea la memoria SIM para la operación celular, así como las conexiones Serial y SPI para comunicarse con el microcontrolador central (ESP32). Por otro lado, una vez terminado, la primera versión del sistema se implementó como GTW durante las pruebas.



Fig. 3: PCB del equipo operando como GTW por WiFi.

## IV. IMPLEMENTACIÓN DEL CÓDIGO

Para la implementación del código se emplearon las bibliotecas estándar de LoRa para ESP32, tanto en su versión para HELTEC como para módulos WROOM [5] [6] públicas. Se incluyó una biblioteca desarrollada para inversores SMA, pero solo para la lectura través de JSON/HTTP [7], la cual se modificó intencionalmente para poder no solo leer los datos del inversor sino operar los parámetros del mismo. Para el sistema basado en MicroPython, se emplearon una serie de bibliotecas estándar que están ampliamente probadas, entre ellas la “u-lora” [8], que provee todas las funciones necesarias para operar en Python el controlador SX1276. A su vez se implementó una máquina de estados general y un método de macros y condicionales, para poder configurar fácilmente características del hardware en el cual se programa el firmware para C++. El mismo modelo de máquina de estados fue empleado para realizar el núcleo operativo del sistema sobre MicroPython.

### A. Modificación de la biblioteca SMA-Reader

Para leer el inversor SMA, se empleó la biblioteca SMA-Reader. Sin embargo, dicha biblioteca no tiene implementadas las funciones de operación sobre parámetros SMA, es decir no nos permitiría controlar parámetros de inyección o de limitación. Por lo tanto, se implementaron una serie de funciones que permiten acceder a las “Keys” del SMA que contienen los parámetros del SMA que son de escritura/lectura. A continuación, se muestran los prototipos de las funciones implementadas:

1.- `getAllParams()`

2.- `setParams(const String* key, uint32 t values)`

La primera función permite extraer los parámetros sean cuales sean y sus correspondientes “Keys”. La segunda permite escribir en la “Key” correspondiente, un dato o una cadena de símbolos según sea necesario.

### B. Implementación de Máquina de estados general

Se implementó una máquina de estados como en la Fig. 4. La misma permite realizar una secuencia de operaciones para realizar la comunicación con los periféricos de inversores y mediante LoRa, en ambos casos bidireccionales.



Categoría

# REPORTE

Bioingeniería

Área Temática

# Sistema para la caracterización de concentración de ozono en agua potable

1st Alejandro Rivas Gamallo  
Departamento de  
Bioingeniería(CEBIO). CUJAE La  
Habana, CUBA  
alejandri@automatica.cujae.edu.cu

2nd Yamel Almeida Pichardo  
Departamento de  
Bioingeniería(CEBIO). CUJAE La  
Habana, CUBA  
yamel.ap@automatica.cujae.edu.cu

3st Luis Noa Llorens Departamento de  
Bioingeniería(CEBIO). CUJAE La  
Habana, CUBA  
luisnl@automatica.cujae.edu.cu

4th Angel Regueiro Gómez  
Departamento de  
Bioingeniería(CEBIO). CUJAE La  
Habana, CUBA  
regueiro@automatica.cujae.edu.cu

**Resumen:** La calidad del agua durante el consumo es de gran importancia para mantener una salud adecuada. Esta investigación aborda el diseño de un sistema para la caracterización y medición de la concentración de ozono durante el proceso de potabilización del agua mediante el método de fotometría ultravioleta. Para ello se utilizan varios programas de aplicación que facilitan el trabajo de simulación y programación de cada bloque del sistema propuesto. Experimentalmente se verifica con buenos resultados las diferentes secciones controladas por un microprocesador PIC18F4550, lográndose comprobar la calidad de las muestras posterior al tratamiento del agua con ozono, a través de la comparación de imágenes obtenidas con ayuda de un microscopio digital biológico XSP-500E (Modelo N800M).

**Palabras clave:** — Calidad del agua, esterilización, ozonificación UV.

## I. INTRODUCCIÓN

El agua es un recurso vital para la vida de los seres vivos en el planeta. En la Observación General # 15 del Comité de Derechos Económicos, Sociales y Culturales de las Naciones Unidas, se reconoce el agua como “un recurso natural limitado y un bien público fundamental para la vida y la salud” [1]. La sociedad recurre a este importante recurso natural para generar y mantener el crecimiento económico y la prosperidad, a través de actividades tales como la agricultura, la pesca comercial, la producción de energía, la industria, el transporte y el turismo, entre otras actividades. El libre flujo de un agua no contaminada resulta clave para el sostenimiento de los ecosistemas que de ella dependen, de ahí la importancia de mantener la calidad del agua para su uso y consumo, lo cual puede ser logrado por diversos métodos, entre ellos: la cloración, la ebullición, la filtración de sedimentos, la fotoemisión UV, la destilación y la ozonificación [2].

El ozono ( $O_3$ ) es una sustancia cuya molécula está compuesta por tres átomos de oxígeno, formada al disociarse los dos átomos que componen el gas de oxígeno. El ozono está presente en dos niveles de la atmósfera: en las proximidades del suelo, en la baja troposfera y en niveles altos, en la estratosfera. En este último nivel forma una capa con elevadas concentraciones que filtra los rayos ultravioletas dañinos para los seres vivos, evitando que alcancen la superficie [3].

El físico holandés Mak Van Marumom en 1785 hace mención de este compuesto por primera vez cuando descubrió que al

pasar una chispa eléctrica a través del aire aparecía una sustancia gaseosa con olor característico, que poseía fuertes propiedades oxidantes [4]. Las propiedades del ozono, hacen de este un gas con un gran poder oxidativo, lo que lo convierte en un agente con una enorme capacidad para la eliminación de microbios y microorganismos patógenos como virus, bacterias, esporas y hongos. [5].

Según la Organización Mundial de la Salud (OMS), el ozono es el desinfectante más eficiente para todo tipo de microorganismos [6] y es muy empleado en el entorno clínico-hospitalario para diversas acciones entre ellas, la ozonoterapia, capaz de eliminar agentes patógenos en diversos tipos de enfermedades

Generalmente los sistemas modernos para la generación de ozono, no disponen de la indicación de concentración generada a lo largo del tiempo, lo cual limita el empleo de estos sistemas en aplicaciones biomédicas, de ahí el interés de desarrollar nuevas alternativas para su caracterización. En este trabajo se describen los avances en el diseño de un sistema para la caracterización continua de la concentración de ozono [ $O_3$ ] durante la purificación de agua potable.

## II. MATERIALES Y MÉTODOS

La figura 1 muestra la representación del sistema propuesto para la generación y caracterización de [ $O_3$ ] a emplear en el proceso de purificación de agua potable en el Dpto. de Bioingeniería (CEBIO). Se aprecia la existencia de una sección de generación de alta tensión a partir de un circuito oscilador con acople de un transformador elevador para la obtención de alta tensión (9 kV) necesario para la generación de ozono a partir del aire que atraviesa por el tubo de emisión proveniente de una bomba de aire mediante el método de DBD (Descarga de Barrera Dieléctrica)[7].

A la salida del flujo de aire ozonizado se acopla un arreglo óptico para la detección de [ $O_3$ ] (Bloque de transducción), cuya salida es conectada al canal de acondicionamiento encargado de la conversión I-V (Corriente-Tensión), amplificación y filtrado analógico de las señales obtenidas, y que se acopla a la etapa de conversión análogo-digital (A/D-12 b) en la unidad de control, procesamiento y visualización de la información, basada en el empleo de un microcontrolador (PIC18F4550).

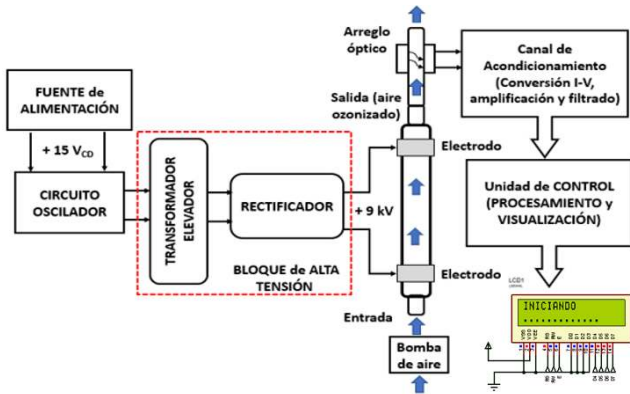


Figura 1. Bloques del sistema para la caracterización de la [O<sub>3</sub>].

Durante el desarrollo de la investigación se utilizaron varias herramientas, entre ellas: Proteus Design Suite (V8.13), FilterPro (V3.1.0) de Texas Instruments para el diseño y simulación de las diferentes secciones del sistema propuesto y el Microscopio Digital Biológico XSP-500E (Modelo N800M, ANJUE, China) y la aplicación SCOPEIMAGE-9 para la verificación de la calidad del proceso de ozonización del agua potable a través de imágenes digitales.

A. Etapa de generación

La figura 2 muestra la sección simplificada del bloque de generación, donde a partir de un oscilador, un transformador elevador y un multiplicador de tensión se obtiene la tensión de 9 kV necesaria para producir el efecto corona en el gas (disociación de los átomos de oxígeno).

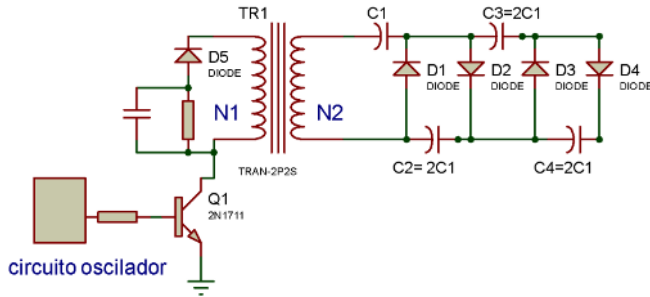


Figura 2. Representación simplificada del bloque generador de alta tensión.

B. Etapa de transducción

Con ayuda de un arreglo óptico para la detección continua de la [O<sub>3</sub>] en el flujo de aire ozonizado proveniente de una bomba, se detecta la concentración generada a partir de fotometría ultravioleta debido a la gran banda de absorción que presenta el ozono en dicha región del espectro[8]. En el arreglo óptico se emplea un LED UV, capaz de transmitir una señal lumínica que será modulada por la concentración de ozono en el flujo de aire generado, y la señal resultante llegará al receptor para detectar los pequeños cambios de corriente en el mismo.

C. Etapa de acondicionamiento

La figura 3 muestra la representación eléctrica del canal de acondicionamiento analógico de la señal. Para la simulación se ha empleado un acoplador óptico que simula el cambio de concentración obtenido en el proceso de ozonización al paso del flujo de aire. La pequeña corriente de salida circula por

un convertidor corriente-tensión que permite incorporar cierta ganancia en el proceso.

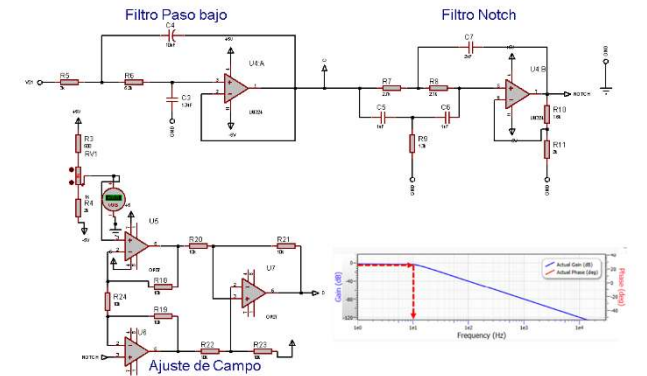
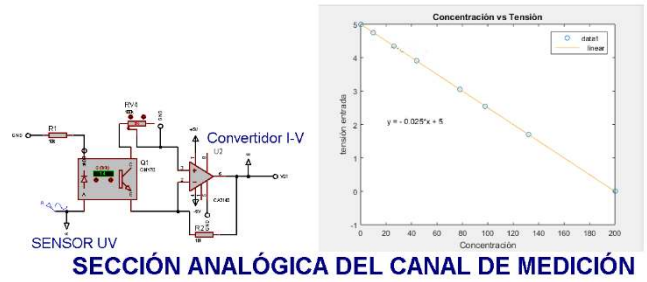


Figura 3. Representación del canal de acondicionamiento de señal.

En la sección inferior se aprecia el filtrado de la señal, iniciando con una etapa paso bajo ( $f_c = 10$  Hz) en cascada con una etapa de filtro Notch para minimizar las interferencias acopladas de la red industrial [9]. La señal resultante de este bloque se acopla a un amplificador para el adecuado ajuste de nivel y del margen dinámico de entrada del convertidor A/D en el microcontrolador.

D. Etapa de control y visualización

La figura 4 muestra el diseño físico del microcontrolador seleccionado, el diagrama de flujo simplificado y ejemplos de visualización. Inicialmente el sistema procede a realizar una comprobación del diseño físico (Memoria, puertos de E/S, alarmas y visualizador, etc.) y si todo está adecuadamente, comienza la secuencia lógica para la adquisición de señal.

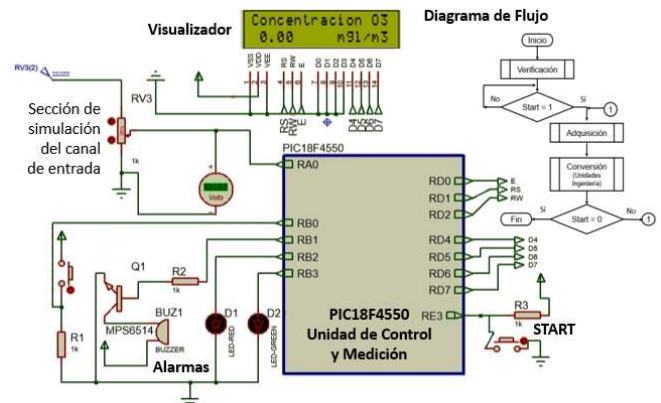


Figura 4. Etapa de control y visualización.

La rutina de adquisición permite la captura de la información con una frecuencia de muestreo de 100 Hz, obtenida a partir de la interrupción generada de uno de los contadores internos (Timer 0) del procesador. Al detectar la solicitud de interrupción asociada al fin de conversión, se lee el dato



correspondiente y se realiza la conversión a unidades de ingeniería ( $\text{gO}_3/\text{m}^3$ ) para su posterior visualización, interrumpiéndose el proceso si se detecta que la medición está fuera del intervalo preestablecido: 0-200  $\text{gO}_3/\text{m}^3$ , activando una alarma a través del empleo de un *buzzer* y un indicador lumínico mientras dure esta situación.

### III. RESULTADOS Y DISCUSIÓN

La figura 5 muestra el sistema propuesto durante las pruebas desarrolladas en laboratorio. La bomba (sección izquierda) impulsa el flujo de aire por el tubo de ozonización excitado por una alta tensión (9 kV) permitiendo la conformación de ozono en el aire conducido hacia el reservorio de agua potable (botellón).



Figura 5. Montaje experimental de prototipo en laboratorio.

Durante el tiempo de excitación, se puede detectar un olor característico del ozono generado y se percibe un cambio notable en el sabor del agua almacenada en el reservorio.

La figura 6 muestra el proceso desarrollado para la verificación preliminar de las muestras ozonizadas de agua, las cuales fueron estudiadas a partir de comparaciones de las imágenes capturadas desde un microscopio digital con ayuda de una aplicación informática (SCOPEIMAGE-9), que permitió establecer comparaciones entre las muestras iniciales y finales obtenidas posterior al proceso de ozonificación para diferentes intervalos de tiempo.

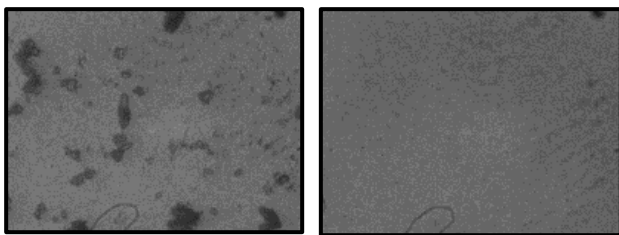


Figura 6. Ejemplo de comparación de imágenes (muestras) de  $\text{H}_2\text{O}$  antes y después (izquierda y derecha respectivamente) de la aplicación de ozono.

El proceso de verificación de muestras ozonizadas se ha desarrollado con periodicidad semanal desde diciembre 2022 hasta la actualidad en el laboratorio, mostrando similares índices de calidad en los resultados obtenidos de agua potable tratada. Actualmente la propuesta del canal de adquisición y visualización de la  $[\text{O}_3]$  se evalúa para ser adicionada a un sistema comercial (ACOZO AF201) producido en el país y que está en uso a partir de su certificación industrial por la entidad regulatoria después de un proceso de calibración y verificación industrial del fabricante.

### IV. CONCLUSIONES

Durante el desarrollo del diseño de la maqueta de laboratorio se logró verificar con buenos resultados los diferentes bloques funcionales: unidad de generación, canal de medición y unidad de control y visualización. Con los experimentos realizados basados en la comparación de imágenes en el laboratorio se pudo verificar la adecuada purificación del reservorio de agua después del tratamiento desarrollado, lo cual permite disponer de agua potable para el consumo.

### REFERENCIAS

- [1] O. d. N. Unidas, "Objetivos para el desarrollo sostenible (#6: Agua limpia y saneamiento), Agenda 2030-ONU," 2015. [Online]. Available: <https://www.un.org/sustainabledevelopment/es/water-and-sanitation/>.
- [2] FiltraShop. "10 Métodos de Purificación de agua altamente efectivos- Filtrashop." <https://filtrashop.com/10-metodos-de-purificacion-de-agua-mas-usados-y-efectivos/> (accessed 21/10/2022, 2022).
- [3] A. Archibald *et al.*, "Tropospheric Ozone Assessment Report A critical review of changes in the tropospheric ozone burden and budget from 1850 to 2100," *Elementa: Science of the Anthropocene*, vol. 8, no. 1, 2020.
- [4] A. S. Tapia and G. Martínez-Sánchez, "La ozonoterapia y su fundamentación científica," *Ozone Therapy Global Journal*, vol. 2, no. 1, pp. 163-198, 2012.
- [5] M. B. Rubin, "The history of ozone. VII. The mythical spawn of ozone: antozone, oxozone, and ozohydrogen," *Bulletin for the History of Chemistry*, vol. 34, no. 1, p. 39, 2009.
- [6] M. Fernández Cuadros, M. Albaladejo Florín, S. Álava Rabasa, D. Peña Lora, and O. Pérez Moro, "Ozono y COVID-19: bases fisiológicas y sus posibilidades terapéuticas según el estadio evolutivo de la infección por SARS-CoV-2," *Revista de la Sociedad Española del Dolor*, vol. 28, no. 1, pp. 27-36, 2021.
- [7] M. P. Silva, "Un nuevo método para desinfectar el agua: Ozono," *Hypatia - Revista de Divulgación Científico - Tecnológica del Estado de Morelos*, vol. 28, p. 2, 2010.
- [8] M. OJEDA, J. C. GOMEZ, and M. S. DE AJA, "INCREMENTO DE LA RADIACION UV-B EN LA ANTARTIDA COMO CONSECUENCIA DE LA REDUCCION DEL OZONO ESTRATOSFERICO," in *Actas del segundo Symposium Español de Estudios Antárticos: Madrid, 13 al 15 de julio de 1987, 1987*: Editorial CSIC-CSIC Press, p. 137.
- [9] D. López Marín, "Historia, definición, descripción, tipos y aplicaciones de filtros electrónicos.," p. 31, 2003.

Categoría

**REPORTE**

FPGAs, HDLs y ASICs

Área Temática

# A New Fault Injection Scheme Optimized for RISC-V Soft Processors

<p>1<sup>st</sup> Adrián M. Edelstein  <i>GDEI-FAMAF y LCSR-FCEFYN</i>  <i>Universidad Nacional de Córdoba</i>            Córdoba, Argentina            adrian.edelstein@mi.unc.edu.ar</p>	<p>2<sup>nd</sup> Pablo A. Ferreyra  <i>GDEI-FAMAF y LCSR-FCEFYN</i>  <i>Universidad Nacional de Córdoba</i>            Córdoba, Argentina            pablo.ferreyra@unc.edu.ar</p>	<p>3<sup>rd</sup> José G. Luján Pérez  <i>GDEI-FAMAF y LCSR-FCEFYN</i>  <i>Universidad Nacional de Córdoba</i>            Córdoba, Argentina            jose.lujan@mi.unc.edu.ar</p>
<p>4<sup>th</sup> Alberto F. Gómez  <i>Departamento de Ciencias Básicas</i>  <i>Universidad Nacional de Chilecito</i>            Chilecito, Argentina            albertogomez@hotmail.com</p>	<p>5<sup>th</sup> Jorge L Naguil  <i>FRSC y UARG</i>  <i>UTN y UNPA</i>            Río Gallegos, Argentina            jnaguil@gmail.com</p>	<p>6<sup>th</sup> Carlos J. Barrientos  <i>Unidad de Formación Superior</i>  <i>CONAE</i>            Córdoba, Argentina            carlos.barrientos@conae.gov.ar</p>

**Abstract**—Open source RISC-V soft processors are becoming increasingly popular in various fields, including critical applications. These applications must be characterized in different scenarios. Single or Multiple Event upset induced failures is one of them. This work contributes with a new fault injection scheme optimized for applications running on RISC-V processors. This optimized tool is validated with preliminary results.

**Index Terms**—RISC-V, fault, injection, SEU, Optimization

## I. INTRODUCTION

Open Source RISC-V soft processors are becoming increasingly popular in various fields of application, [1]–[4]. One area that currently stands out is space systems. Such systems have critical aspects that require a certain level of fault tolerance in order to withstand the hostility of the space environment. The fact that their architecture and source code are open, allows us to implement various ways to characterize the reliability and availability in RISC-V based systems. One of these characterizations is based on fault injection that simulate Single Event Upsets (SEUs) or Multiple Event Upsets (MBUs). Several papers deal with SEU and MBU type fault injections techniques, as can be seen in references [6]–[9]. Those previous works address this problem

This work was supported by SECyT-UNC.

with general purpose and non-optimal tools, as can be seen in [10]–[13]. In contrast, this work contributes with a new fault injection scheme optimized for applications running on RISC-V processors. In section II the simplified implementation of a RISC-V processor developed for this work is briefly described. In section III the developed fault injection tool is also briefly described. The tool is optimized for injecting SEU or MBU in the internal registers of the RISC-V processor. In section IV it is shown that with the fault injection tool we can obtain a parameter known in the literature as the error rate. The tool is used to obtain this parameter in a simple algorithm so the results and the tool can both be validated by analysis. Finally, in section V some conclusions and future works are given.

## II. SIMPLIFIED RISC-V PROCESSOR

Fig. 1 is a simplified block diagram of the RISC-V processor implemented in an FPGA.

It shows how an R Format instruction works. An example of an instruction that uses this format is *add rd, rs1, rs2* which adds the data stored in the registers *rs1* and *rs2* and stores the result in *rd*. Highlighted in green we can see the relevant connections and control lines for these type of instructions. Table I shows how this format is structured.

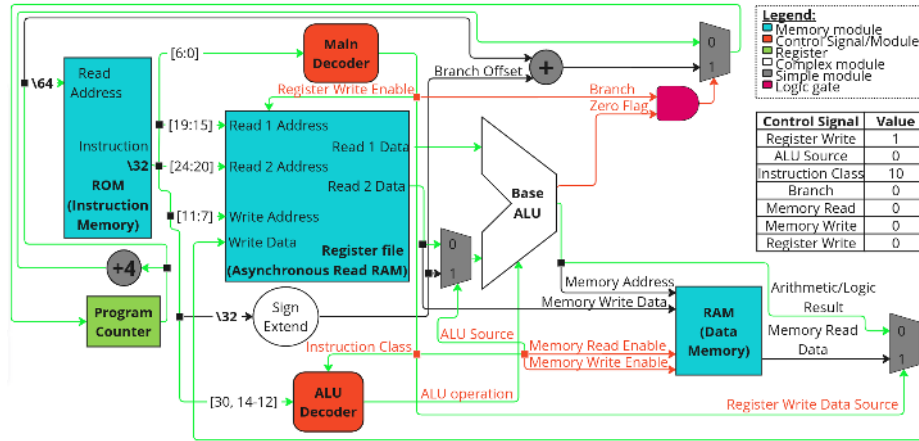


Fig. 1: Simplified RISC-V Implementation Block Diagram.

TABLE I: R Format Instruction Fields.

[31:25]	[24:20]	[19:15]	[14:12]	[11:7]	[6:0]
funct7	rs2	rs1	funct3	rd	opcode

In RISC-V, the 7 least significant bits of the instruction field always represent the *opcode* which implies the instruction format. In the case of this type of instruction, the *funct3* and *funct7* fields distinguish the ALU operation. Instructions that have the field *rd* as destination set the control signal Register Write Enable. If the Branch control signal is set, we replace the current PC with the branch address we calculate if the conditions for branching are satisfied. Although not shown, note that the branch address offset must be shifted by one bit; this is set in the Sign Extend module.

### III. FAULT INJECTION TOOL

Fig. 2 is a simplified block diagram of the Fault Injection Tool. Initially when programming

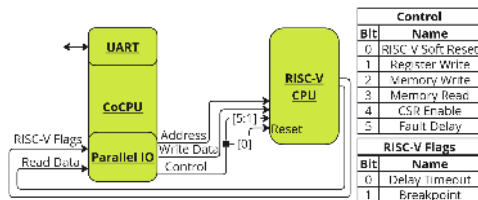


Fig. 2: Optimized Fault Injection Tool Block Diagram.

the FPGA the signal RISC-V Soft Reset is set and the co-processor does not have a program. Thus we must download onto the co-processor a program that sets the control signals properly for fault injection. First we setup a delay measured in RISC-V instruction cycles by setting the address control signal to point to the Delay Count CSR. Since the first 12 bits belong to the RISC-V CSR addresses we use the 13th bit to address more registers. We set the delay amount using the data write bus; then we set the control signals Fault Delay Enable and CSR Enable for one RISC-V clock cycle and we clear CSR Enable to disable writing the register twice; lastly we clear the data set in the Data Write bus to avoid setting any wrong bits when injecting a fault. After that we wait using the function *usleep()*, disable the RISC-V Soft Reset signal to start the processor and wait in a loop until the RISC-V CPU notifies the co-processor when the desired time has elapsed using the Delay Timeout flag. Then we can select a register and set a bit mask (the bits that are set in the mask will be flipped in the selected general purpose register). Lastly we disable the Register Write Enable signal and we wait until a timeout or for the Break flag to be set, after which we can read the results from UART.

### IV. EVALUATION AND VALIDATION

For preliminary testing purposes we decided to use a simple program that calculates  $\sum_{n=1}^N n$  where the error rate is the amount of times the program

returned the wrong value or did not finish due to a timeout. Injection was done on the registers that the program utilized. Among these we can highlight  $ra$ (return address),  $a_0$ (function return value) and  $s_0$ (frame pointer). Events in  $ra$  almost always guaranteed a timeout, given that after returning from a function we could have jumped to a random point in the program. The way the program is structured is that if we jump outside the scope of the program we get stuck in a loop. Events in  $a_0$  yielded the correct result most of the time, this is because this register is set at the end of the *sum* function and thus the only cases where it yielded a wrong result was when we injected a fault after the calculation had finished. In Fig. 3 we can see the results.

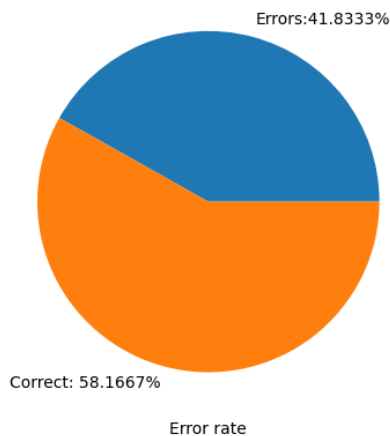


Fig. 3: SEU results using the program

## V. CONCLUSIONS AND FUTURE WORK

In section IV it is shown that with the fault injection tool we can obtain a parameter known in the literature as the error rate. With this parameter and some calculations, we can obtain in the future the reliability of an application running on a RISC-V processor.

## ACKNOWLEDGMENT

To the Secretaría de Ciencia y Tecnología of the Universidad Nacional de Córdoba, Argentina, for supporting this work.

## REFERENCES

- [1] E. Matthews and L. Shannon, "Taiga: A new risc-v soft-processor framework enabling high performance cpu architectural features," in 2017 27th International Conf. on Field Programmable Logic and Applications (FPL), pp. 1–4, Sep. 2017.
- [2] C. Duran, L. Rueda, G. Castillo, A. Agudelo, C. Rojas, L. Chaparro, H. Hurtado, J. Romero, W. Ramirez, H. Gomez, H. Hernandez, J. Amaya, and E. Roa, "A 32-bit microcontroller featuring a RISC-V core, 2016. [Online]. Available: <https://github.com/onchipuis/mriscv>
- [3] M. Samsoniuk, "Opensource RISC-V DarkRISCV," 2019. [Online]. Available: <https://github.com/darklife/darkriscv>
- [4] W. F. Heida, "Towards a fault tolerant risc-v softcore," Ph.D. dissertation, Delft Univ. of Technology, 2016, <http://resolver.tudelft.nl/uuid:cee5e97b-d023-4e27-8cb6-75522528e62d>.
- [5] K. Douglas Almeida dos Santos, Lucas Matana Luza, Cesar Albenes Zeferino, Luigi Dilillo, Douglas Rossi Melo, "A Low-Cost Fault-Tolerant RISC-V Processor for Space Systems," DTIS 2020 - 15th International Conference on Design and Technology of Integrated Systems in Nanoscale Era, Apr 2020, Marrakech, Morocco. pp.1-5, 10.1109/DTIS48698.2020.9081185. lirmm-03026132
- [6] J. Laurent, V. Berouille, C. Deleuze, F. Pebay-Peyroula, and A. Papadimitriou, "On the Importance of Analysing Microarchitecture for Accurate Software Fault Models," in 2018 21st Euromicro Conference on Digital System Design (DSD), 2018, pp. 561–564.
- [7] M. Rebaudengo, M. S. Reorda, and M. Violante, "Analysis of SEU effects in a pipelined processor," in Eighth IEEE International On-Line Testing Workshop (IOLTW), 2002, pp. 112–116.
- [8] D. Gil, J. Gracia, J. C. Baraza, and P. J. Gil, "Analysis of the influence of processor hidden registers on the accuracy of fault injection techniques," in Ninth IEEE International High-Level Design Validation and Test Workshop (IEEE Cat. No.04EX940), 2004, pp. 173–178.
- [9] L. Dureuil, G. Petiot, M.-L. Potet, T.-H. Le, A. Crohen, and P. de Choudens, "FISSC: A Fault Injection and Simulation Secure Collection," 2016, pp. 3–11.
- [10] Ebrahimi, M., Mohammadi, A., Ejlali, A., and Miremadi, S.G., "A fast, flexible, and easy-to-develop FPGA-based fault injection technique," *Microelectron. Reliab.*, 54, 2014, pp. 1000-1008.
- [11] D. de Andres, J. C. Ruiz, D. Gil and P. Gil, "Fault Emulation for Dependability Evaluation of VLSI Systems," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 4, pp. 422-431, April 2008, doi: 10.1109/TVLSI.2008.917428.
- [12] M. Solinas, A. Coelho, J. A. Fraire, N. E. Zergainoh, P. A. Ferreyra and R. Velazco, "Preliminary results of NETFI-2: An automatic method for fault injection on HDL-based designs," in 18th IEEE Latin American Test Symposium (LATS), Bogota, Colombia, 2017, pp. 1-4, doi: 10.1109/LATW.2017.7906748.
- [13] W. Mansour and R. Velazco, "An Automated SEU Fault-Injection Method and Tool for HDL-Based Designs," in *IEEE Transactions on Nuclear Science*, vol. 60, no. 4, pp. 2728-2733, Aug. 2013, doi: 10.1109/TNS.2013.2267097. 2011. 10.1016/j.microrel.2011.06.017

# Herramienta para el diseño e implementación de filtros digitales basada en software libre

Luciano Leonel Ledesma  
U.T.N. Facultad Regional Paraná  
Paraná, Entre Ríos, Argentina  
lucianoledesma@alu.frp.utn.edu.ar

Alejandro Raúl Dachary  
Laboratorio de Procesamiento y  
Transmisión de Señales  
U.T.N. Facultad Regional Paraná  
Paraná, Entre Ríos, Argentina  
adachary@frp.utn.edu.ar

Enrique Sergio Burgos  
Laboratorio de Informática  
U.T.N. Facultad Regional Paraná  
Paraná, Entre Ríos, Argentina  
sergioburgos@frp.utn.edu.ar

**Resumen**—En este trabajo se presenta una herramienta de cálculo denominada *Filter Designer* para el diseño de filtros digitales basada en software libre, utilizando GNU Octave como motor de cálculo y el paquete *signal*. Como característica distintiva se puede mencionar que utiliza representaciones numéricas de punto fijo y permite la generación automática de código fuente en lenguaje Verilog para su posterior implementación. Se incluye también el resultado de las pruebas realizadas con filtros pasa bajas y su implementación en dos kits de desarrollo de FPGA (EDU-FPGA y Digilent Zybo).

**Palabras claves:**—filtros digitales, FPGA, GNU Octave, IIR, FIR

## I. INTRODUCCIÓN

El diseño de filtros utilizando herramientas de cálculo no es nuevo y ha sido tratado en varias ocasiones. Entre otros trabajos podemos mencionar [1] donde a partir del uso de Matlab y el paquete de procesamiento de señales se diseñan diferentes tipos de filtros. En [2] se analiza la aplicación de Matlab para el diseño de filtros FIR y en [3] se utiliza el paquete *FDATool* a fin de diseñar filtros IIR comprobando su resultado en señales de audio. Si bien todos son casos de éxito, el uso de software licenciado puede generar una dificultad, no solamente por el costo de la licencia sino también por los requerimientos de hardware indicados para su uso.

A partir de esto, y considerando antecedentes de compatibilidad entre Matlab y GNU Octave [4] se optó por GNU Octave versión 7.1 como base para el desarrollo.

El trabajo se estructura del siguiente modo, en la sección II se presentan la metodología utilizada para el desarrollo de la herramienta y el cálculo de los coeficientes, en la sección III se presentan las pruebas realizadas con dos plataformas de hardware y software y, en la sección IV, se presentan comentarios respecto de los resultados encontrados y futuras funcionalidades que se consideran de interés para nuevas versiones de la aplicación.

## II. METODOLOGÍA

GNU Octave, a través del *toolbox signal* [5], incorpora funcionalidades que permiten el diseño de filtros siendo este el punto de partida considerado en este trabajo. Las funcionalidades provistas por esta herramienta posibilitan trabajar

utilizando toda la precisión disponible en los tipos de datos de punto flotante. Si bien los resultados así obtenidos son de utilidad en algunas implementaciones, en los casos de interés basados en FPGA, resulta conveniente ajustar la precisión a fin de satisfacer los requerimientos esperados con el menor consumo de recursos.

Esto es posible de lograr a través de la utilización de valores de punto fijo, ya que permiten indicar la cantidad de bits a emplear en la representación de las partes enteras y fraccionarias de un valor de tipo real, logrando así acotar el consumo de recursos de la implementación.

El uso de esta forma de representación numérica implica tener en cuenta dos factores. Por un lado, la magnitud de la parte entera de cada valor fijará la cantidad mínima de bits a utilizar. Por otro lado, al disminuir la cantidad de bits a utilizar en la representación fraccionaria, se tendrá menos precisión en la respuesta, lo que podría generar resultados que se alejen de lo esperado. En este documento denominaremos cuantización al proceso por el cuál se establece la representación de un valor con una cantidad arbitraria de bits a partir de su representación en punto flotante.

A fin de lograr una aplicación que simplifique el proceso de diseño y cuantización de los filtros, se implementó una interfaz gráfica que denominamos “Filter Designer” que permita la manipulación de los diferentes parámetros. En este sentido, vale mencionar que GNU Octave no cuenta con una herramienta que permita la construcción de interfaces gráficas de usuario, por lo que se utilizó el entorno de desarrollo *guiEditor* [6].

La aplicación desarrollada consta de solo dos ventanas de diálogo. A través de la primera ventana de diálogo (Fig.1), es posible elegir las características del filtro de interés y analizar su respuesta. Hasta este punto el análisis y procesamiento se realiza utilizando álgebra de punto flotante. Una vez obtenida la respuesta deseada es posible cuantizar los coeficientes del filtro y establecer el tamaño de los datos de entrada y salida a través de una segunda ventana de diálogo. Además, desde esta segunda ventana también es posible visualizar el valor de los coeficientes, el valor aproximado de estos según la representación de punto fijo elegida y la representación entera

de los valores cuantizados en punto fijo, lo que permite su uso en otros sistemas digitales que utilicen este formato de representación.

Una vez obtenidos los valores en punto fijo, es posible generar el código fuente asociado al filtro en lenguaje Verilog. La cantidad de archivos de código fuente generados dependerá del tipo y orden del filtro ya que en el caso de los filtros IIR los coeficientes de cada sección de segundo orden se almacenan por separado. El formato base utilizado para la implementación es el propuesto en el material didáctico de [7], siguiendo los fundamentos planteados en [8].

### III. RESULTADOS

A fin de comprobar el funcionamiento de los filtros diseñados con esta herramienta, se realizaron múltiples pruebas de diversos tipos de filtros. Se generó el código fuente asociado y se verificó su respuesta utilizando herramientas de simulación para FPGA incluidas en dos suites de desarrollo. Con el actual grado de avance de la herramienta presentada, no es posible analizar el resultado del proceso de filtrado. Esto se debe a que al operar matemáticamente con los valores de punto fijo, GNU Octave los trata como valores de punto flotante, lo que impide observar los resultados asociados a la pérdida de información debido a la cantidad de bits considerados.

Además de las simulaciones también se realizaron pruebas sobre la *EDU-CIAA-FPGA* [9] (FPGA ICE40HX4K) utilizando el entorno de desarrollo *IceCube2*. Las simulaciones en este caso se realizaron utilizando Mentor ModelSim, incorporado en la distribución de la aplicación para sistema operativo MS Windows. En la figura 2 puede observarse la salida obtenida al aplicar una señal de 9 bits (1 bit de signo y 8 de parte fraccionaria) a un filtro pasa baja, IIR, de orden 4, con una frecuencia de corte de 150 Hz. Aquí el valor de la señal de entrada fue generado desde GNU Octave como la suma de dos señales senoidales positivas, de 20.25 y 405 Hz, respectivamente, con una frecuencia de muestreo de 2025 Hz. Estas señales fueron cuantizadas así como los coeficientes del filtro, pero utilizando para estos 20 bits (14 bits para la parte fraccionaria, 5 bits para la parte entera y 1 bit de signo).

Una comprobación adicional se realizó utilizando un módulo basado en el dispositivo PCF8591 [10], que incorpora un convertor analógico a digital de 4 entradas y un convertor digital a analógico de 1 canal, ambos con una resolución de 8 bits. Se implementó un módulo que permita la adquisición de una entrada analógica, para luego realizar el filtrado y generar la salida filtrada. La arquitectura de este sistema fue la que definió los parámetros de la simulación indicada anteriormente ya que se logró una frecuencia de muestreo de 2025 Hz.

A la hora de tratar el signo de las señales utilizadas y los valores de entrada y salida de los conversores, fue necesario tener una consideración particular. Para las pruebas, los valores resultantes de la conversión realizada por el ADC fueron tratados como parte fraccionaria de la señal. Luego, restando la constante 128 al valor adquirido se logró una señal con

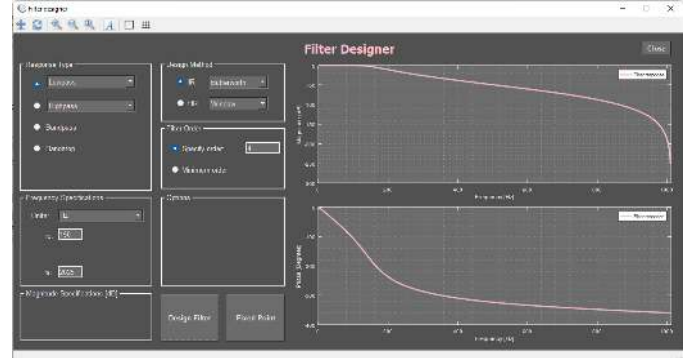


Fig. 1. Aspecto de la interfaz de diseño de la herramienta presentada. Puede observarse las opciones de diseño así como la respuesta a la configuración indicada.

valores positivos y negativos. Se siguió un procedimiento similar para calcular el valor a aplicar a la entrada del DAC. Al resultado del proceso de filtrado se le sumó el valor 128 para luego eliminar el bit más significativo, generando un resultado congruente con las simulaciones realizadas.

En las tablas I y II se presenta el consumo de recursos del sistema y la relación entre los recursos consumidos y disponibles en el dispositivo. Cabe mencionar que en este resumen se incluyen, además de los recursos utilizados por el filtro, los asociados al módulo principal y la máquina de estados responsable de controlar los conversores (módulo PCF8591).

También se realizaron pruebas similares a las anteriores utilizando la placa de desarrollo Zybo de la empresa *Digilent*, que incorpora el SoC Z-7000 de la empresa *Xilinx*. En este caso la adquisición de la señal se realizó utilizando la interfaz XADC y, a través de las herramientas de depuración incorporadas en el entorno de desarrollo de Vivado, fue posible obtener la representación de la salida obtenida en el filtro. Si bien el comportamiento obtenido fue similar, el consumo de recursos observado fue menor. En la tabla III puede observarse el consumo de recursos de la implementación del filtro digital IIR, de orden 4, utilizando 20 bits para representar los coeficientes (1 bit de signo, 5 bit para entera y 14 para la parte fraccionaria) y 9 bits (1 bit de signo y 8 bits para la parte fraccionaria) para la entrada y salida.

Tabla I: Recursos utilizados por una implementación que utiliza un filtro IIR, pasa bajas, de orden 4

Número de LUTs	2594
Número de DFFs	207
Número de Carrys	674
Cantidad de RAMs	0
Cantidad de ROMs	0

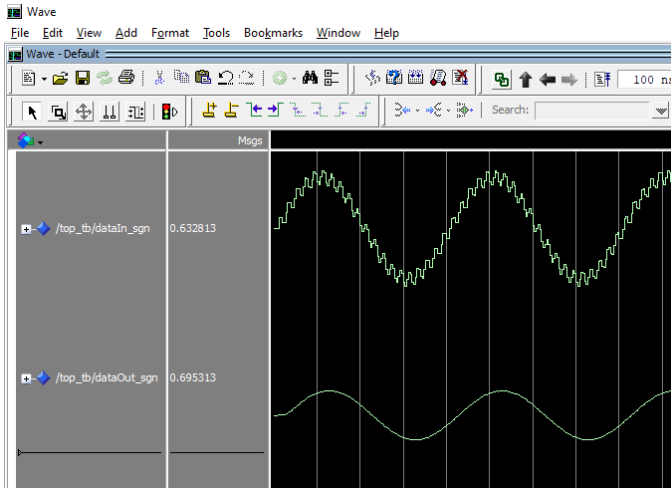


Fig. 2. Resultado de la simulación considerando un filtro IIR pasa bajas con frecuencia de corte de 150 Hz y orden 4. La señal de entrada es la suma de dos señales senoidales de 20.25 y 405.0 Hz.

Tabla II: Recursos utilizados en la FPGA ICE40HX4K de una aplicación que utiliza un filtro IIR, pasa bajas y orden 4.

LogicCells	2634/3520
PLBs	367/440
BRAMs	0/20
IOs y GBIOs	9/107
PLLs	1/2

Tabla III: Consumo observado en una implementación utilizando Vivado y la placa de desarrollo Zybo.

Slice LUTs	897/17600
Slice Registers	124/35200
DSPs	6/80
Bonded IOB	20/100
BufGCtrl	1/32

#### IV. CONCLUSIONES Y TRABAJOS FUTUROS

La herramienta presentada, que ha demostrado un correcto funcionamiento en las condiciones comprobadas, aún se encuentra en desarrollo. Aunque los tipos de filtros que permite diseñar son limitados, se espera incrementar las características y funcionalidades, mediante las posibilidades del paquete *signal*.

Dentro de las funcionalidades que consideramos significativas de incorporar se encuentra la posibilidad de generar código fuente en lenguaje C para realizar la implementación de filtros en microcontroladores partiendo de modelos típicos.

Una futura línea de trabajo que potenciará el desarrollo de esta herramienta es la implementación de un tipo de dato nativo en GNU Octave que permita la representación de valores de punto fijo. Esto permitiría la expansión del trabajo realizado a la simulación dentro del entorno de diseño utilizando valores de entrada predefinidos, y la posibilidad de diseñar controladores en aplicaciones de control de procesos. Si bien se ha documentado un antecedente de este tipo de implementación en [11], actualmente no forma parte de los paquetes mantenidos por la comunidad de software libre.

Aún cuando se trata de un proyecto que continúa en desarrollo, la versión actual es funcional y puede descargarse gratuitamente de [12].

#### REFERENCIAS

- [1] D. Báez-López, D. Báez-Villegas, R. Alcántara, J. J. Romero, and T. Escalante, "Package for filter design based on matlab," *Computer Applications in Engineering Education*, vol. 9, no. 4, pp. 259–264, 2001.
- [2] A. A. Eleti and A. R. Zerek, "FIR digital filter design by using windows method with matlab," in *14th International Conference on Sciences and Techniques of Automatic Control & Computer Engineering-STA'2013*. IEEE, 2013, pp. 282–287.
- [3] S. R. Sutradhar, N. Sayadat, A. Rahman, S. Munira, A. F. Haque, and S. N. Sakib, "FIR based digital filter design and performance analysis," in *2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*. IEEE, 2017, pp. 1–6.
- [4] E. E. Tarifa, S. L. Martínez, S. F. Domínguez, and Á. F. Núñez, "Análisis de entornos matemáticos numéricos gratuitos y de código abierto para simulación," *Difusiones*, vol. 20, no. 20, pp. 88–104, 2021.
- [5] M. M. (maintainer). (2023) *Signal Package signal processing tools, including filtering, windowing and display functions*. [Online]. Available: <https://gnu-octave.github.io/packages/signal/>
- [6] E. S. Burgos and E. J. Adam, "Graphical user interface editor for octave applications," *Engineering Reports*, vol. 2, no. 11, p. e12269, 2020.
- [7] D. A. Pola, "Diseño Digital Avanzado," <https://fcefyn.unc.edu.ar/facultad/secretarias/extension/area-de-formacion-continua/disenio-digital-a-vanzado-dda-2021/>, 2022, curso de Posgrado, Universidad Nacional de Córdoba.
- [8] S. A. Khan, *Digital design of signal processing systems: a practical approach*. John Wiley & Sons, 2011.
- [9] S. web del proyecto CIAA, "Sitio web del proyecto EDU-CIAA-FPGA," <http://www.proyecto-ciaa.com.ar/devwiki/doku.php?id=desarrollo:edu-fpga>, 2023, [Online; accedido 17/04/2023].
- [10] N. Semiconductors, "Hoja de datos del dispositivo," <https://www.nxp.com/docs/en/data-sheet/PCF8591.pdf>, 2023, [Online; accedido 17/04/2023].
- [11] D. Bateman, L. Mazet, V. Buzenac-Settineri, and M. Muck, "A fixed-point type for octave," *arXiv preprint cs/0604039*, 2006.
- [12] L. Ledesma, A. Dachary, and S. Burgos, "Repositorio oficial del proyecto," <https://gitlab.com/labinformatica/fdsn>, 2023, [Online; accedido 17/04/2023].



Categoría

**REPORTE**

Implementación de SE

Área Temática

# Desarrollo de una Unidad Automática para la Detección de Radiación UV de Uso 24/7, con Acceso a una Red Local Vía Wi-Fi.

1er Mario Blas Lavorato  
Grupo TAMA

Facultad Regional Haedo  
Universidad Tecnológica Nacional,  
Buenos Aires, Argentina  
mlavorato@frh.utn.edu.ar

2do Nicolás Urbano Pintos  
Grupo TAMA

Facultad Regional Haedo  
Universidad Tecnológica Nacional,  
Buenos Aires, Argentina  
nurbano@frh.utn.edu.ar

3er Mauricio Roberto Pagura  
Grupo TAMA

Facultad Regional Haedo  
Universidad Tecnológica Nacional,  
Buenos Aires, Argentina  
mpagura@frh.utn.edu.ar

**Resumen** — Se presentan los resultados del desarrollo de un Radiómetro que cuenta con dos sensores para medir la Radiación UV (UVA + UVB e Índice UV). Además, la unidad cuenta con otro dispositivo que mide en forma simultánea la Presión Atmosférica, la Temperatura exterior y la Humedad Relativa Ambiente. El dispositivo de control, adquisición, resguardo de datos y comunicación vía Wifi es un módulo de desarrollo de Firmware abierto basado en ESP32 (NODEMCU32 ESP-WROOM-32s). Este Microcontrolador cumple con las especificaciones requeridas y es el más apto para realizar las tareas simultáneas de adquisición de señales, cálculo y conversión de unidades; lleva asociado un reloj de tiempo real y un dispositivo SD para el registro y salvaguarda de datos. Las señales adquiridas y preprocesadas son transmitidas a un 'router' externo para la lectura de los parámetros atmosféricos adquiridos en tiempo real. Los resultados son visualizados por cualquier sistema informático externo conectado a la misma red (Smartphone, Tablet, Notebook o PC). Este equipo no posee 'display' de monitoreo; esta tarea la realiza el dispositivo informático utilizado. Se muestran los resultados procesados correspondientes a las mediciones realizadas durante 3 días.

**Keywords**— Radiación UV, Índice UV, Sensores, ESP32, Microcontroladores, Smartphone.

## I INTRODUCCIÓN

Los procesos meteorológicos y climáticos impactan directa o indirectamente sobre la vida en la Tierra. Estos procesos afectan el medio ambiente sobre todo en la producción agrícola-ganadera y en la salud de la población empeorando su calidad de vida. Uno de los factores que puede afectar el clima es la Radiación Solar que llega a la Tierra. Si realizamos un espectro de la radiación Solar, se pueden asociar con bandas de colores, como los del arcoíris; la gama de colores del arcoíris solo representa el espectro de radiación visible (es la que detecta el ojo humano). Por debajo y por encima de esta banda hay también radiación, solo que el ojo humano no la percibe, es otro el órgano que la detecta, y es en forma de calor (la piel). El espectro general de radiación es mucho mayor tal como lo muestra la Figura 1; en ella se distinguen las Bandas de Radiación visible que van desde los 400 nm hasta los 750 / 800 nm, de allí en más comienza la denominada radiación infrarroja (IR) cercana (800 nm a 4000 nm), luego sigue la IR lejana (4  $\mu\text{m}$  a 50  $\mu\text{m}$ ) y más allá las ondas de radio. Del otro lado del espectro visible, se encuentra la radiación UV (UVA + UVB + UVC + UV Vacío) y más allá los rayos X [1]. Siendo el Sol la principal fuente de Radiación que llega a la Tierra el experimento Lowtran 7 [2] realizado por la NASA permitió conocer con certeza el espectro de Radiación proveniente del Sol (dos dispositivos: uno fuera de la atmósfera y el otro a nivel del mar). En la Figura 2 se pueden observar las bandas de absorción de la

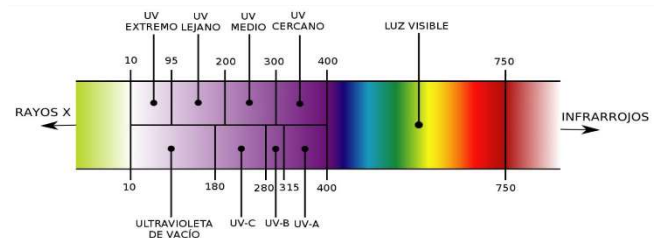


Figura 1. Espectro de Radiación general. Se pueden observar las bandas de: Ultra Violeta (UV) – Visible – Infrarroja (IR).

atmósfera y sobre todo la cantidad de radiación Ultra Violeta que llega al suelo. La Radiación UV es la que más afecta la vida en la Tierra por un lado, gracias a su presencia se desarrolló la vida vegetal y animal en nuestro planeta; un exceso de la misma puede ser letal para las plantas y los seres vivos. La radiación UV representa solo el 5 % o 6 % de la Radiación Solar total que llega al suelo. Es invisible al ojo humano, pero no a la sensibilidad de la piel y cubre la banda espectral entre los 180 nm y los 400 nm. El rango de radiación UV proveniente del sol fue dividido arbitrariamente en tres bandas: UVC de 180 nm a 280 nm es letal para la vida, pero es filtrada en su totalidad por la atmósfera superior; la UVB de 280 nm a 320 nm atraviesa la atmósfera y puede dañar seriamente el material genético de las células provocando cáncer. Por último, la UVA de 320 nm a 400 nm es la menos dañina ya que solo < 10 % de la misma puede afectar a los organismos vivos pero el resto favorece el desarrollo de la vida. La radiación UV tiene un alto impacto sobre los seres vivos y todas las sustancias orgánicas. Esta radiación es esencial para la vida ya que solo la radiación UV estimula la síntesis de algunas vitaminas necesarias para la vida. Los efectos de la exposición a la radiación UV, es la quemadura

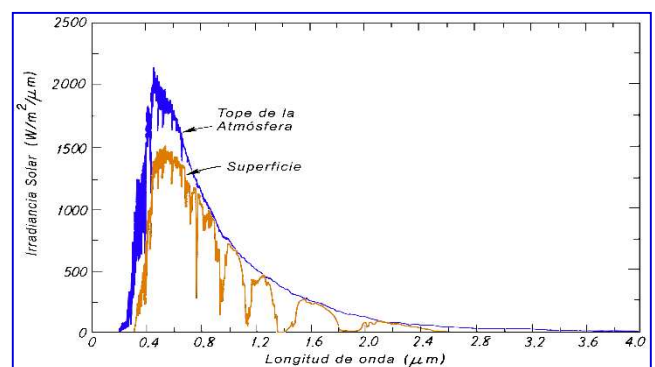


Figura 2. Experimento Lowtran 7 – Radiación Solar fuera de la atmósfera vs. Radiación Solar que llega a la Tierra.

IUV	Grado	Radiación [ W/m2 ]	Índice UV	0	1	2	3	4	5
0 - 2	Bajo	< 0,075	Índice UV	0	0,025	0,05	0,075	0,1	0,125
3 - 5	Moderado	0,075 - 0,149	Índice UV	6	7	8	9	10	11
6 - 7	Alto	0,150 - 0,199	Radiación UVE [ W/m2 ]	0,15	0,175	0,20	0,225	0,25	0,275
8 - 10	Muy Alto	0,200 - 0,269							
11+	Extremo	> 0,270							

Figura 3. Tabla y Diagrama de colores correspondiente a los distintos niveles de Radiación UVE y su correspondencia en Densidad de Potencia.

de Sol o eritema de piel. Por ello se definió la UVE (Ultra Violeta dosis Eritémica) que resulta de combinar toda la banda de UVB y una parte de la UVA [3].

El Servicio Nacional de Meteorología y la Agencia de Protección Ambiental de los Estados Unidos (EPA, por sus siglas en inglés) [4] han desarrollado el Índice UV diario para medir la intensidad de la Radiación UV. Basado en parte en la climatología del índice UV de OMI (Ozone Monitoring Instrument), este índice UV diario predice la intensidad de UV utilizando una escala de 0 a 16+, donde 0 representa ‘nada’ de riesgo y los valores superiores a 11 representan un riesgo extremo de exposición. Cuando se habla del Índice UV son muchas las representaciones disponibles tanto en números como en colores. En la Figura 3 se muestran las dos más difundidas a nivel mundial.

En el presente trabajo se muestra el desarrollo de un prototipo de Instrumento de medición autónomo para medir la radiación UV total y el Índice UV durante todo el día (24/7), tomando como unidad de procesamiento y control el ESP32 (NODEMCU32 ESP-WROOM-32s) [5].

## II EXPERIMENTACIÓN

El prototipo que se presenta es la base para la fabricación de varias unidades que puedan ser instaladas en diferentes locaciones lo que va a permitir realizar un mapa de Radiación UV regional. El principal objetivo es advertir a la población cuando los niveles radiación comienzan a ser peligrosos para la salud. Los equipos comerciales miden la Radiación UV como densidad de potencia con gran precisión y de allí se calcula el Índice UV o, dispositivos que miden directamente el Índice UV. Los mismos son extremadamente costosos (Eko, Kipp&Zonen) y si el objetivo es realizar un mapa de diferentes zonas, se van a necesitar varias unidades; el presupuesto resultaría inalcanzable. Surge entonces, la necesidad de desarrollar equipos propios que midan la Radiación UV y que puedan determinar el Índice UV correspondiente. El prototipo cuenta con el microcontrolador ESP32 con un Procesador Tensilica Xtensa 32bits LX6 hasta 240MHz, Bluetooth: v4.2 BR/EDR y Bluetooth Low Energy (BLE), Wi-Fi: 802.11b/g/n/e/i (802.11n @ 2.4 Ghz hasta 150 Mbit/s), memoria Flash embebida: 4 MiB (ESP32-PICO-D4 SIP module), periféricos compatibles: ADC, DAC, I2C, UART, Interfaz CAN 2.0, SPI, I2S, RMII y PWM entre otros, una SRAM: 520 KiB, etc. El dispositivo construido (ver Figura 4) cuenta con una fuente de ‘switching’ con salida de 5 V estabilizados para la alimentación del ESP32. La unidad puede ser alimentada por dos baterías de Ion-Li o una fuente regulada 220 - 9 V (las del Arduino) o una pantalla Solar o una combinación entre ellas. Como Sensores UV se utilizan dos dispositivos, el ML8511 que mide la columna total de Radiación UV (UVB + UVA) en W/m2 y el UVM-30A que permite medir directamente el Índice UV. Como accesorios adicionales se agregó el DS3231 (reloj de tiempo real), el BME/BMP280 que permite medir la Temperatura, Presión y

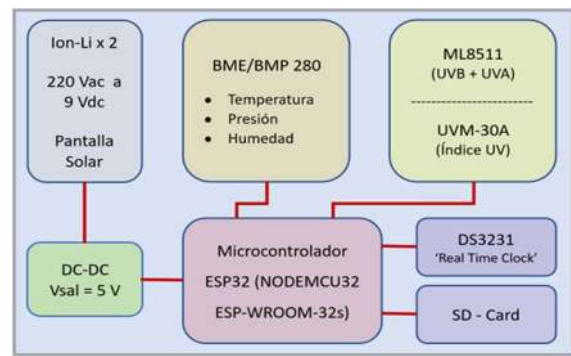


Figura 4. Diagrama esquemático del Prototipo implementado



Figura 5. Fotos de la placa del ESP32 y accesorios, de los Sensores UV y de parámetros Meteorológicos.

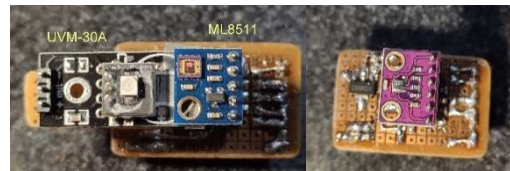


Figura 6. Montaje del Sensor UV en la zona de medición (1300 m).

Humedad relativa ambiente y un dispositivo SD-Card para el registro histórico de datos (‘backup’) [6] [7] [8].

## III RESULTADOS

En las Figuras 5 y 6 se muestra el prototipo armado y en operación con los sensores UV y el dispositivo meteorológico. Se muestran las mediciones realizadas durante los días 17, 18 y 19 de marzo del 2023 en la localidad de Carpintería a 1300 metros de altura. Los días 17 y 19 se observa la presencia de nubes mientras que el día 18 el cielo permaneció completamente despejado. Las variables radiométricas y meteorológicas son: la Radiación -UV [ W/m2 ], el Índice-UV [ s/u ], Temperatura [ °C ], Presión [ hPa ] y Humedad [ % ]. En la Figura 7 podemos observar la locación de las mediciones. En la Figura 8 (marzo 17), 9 (marzo 18) y 10 (marzo 19) observamos la evolución temporal de la Radiación UV, del Índice UV y, de los parámetros meteorológicos locales: Temperatura, Presión y Humedad.



Figura 7. Fotografía del sitio a 1.300 m de altura el 19/03/2023 con presencia de nubes intermitentes

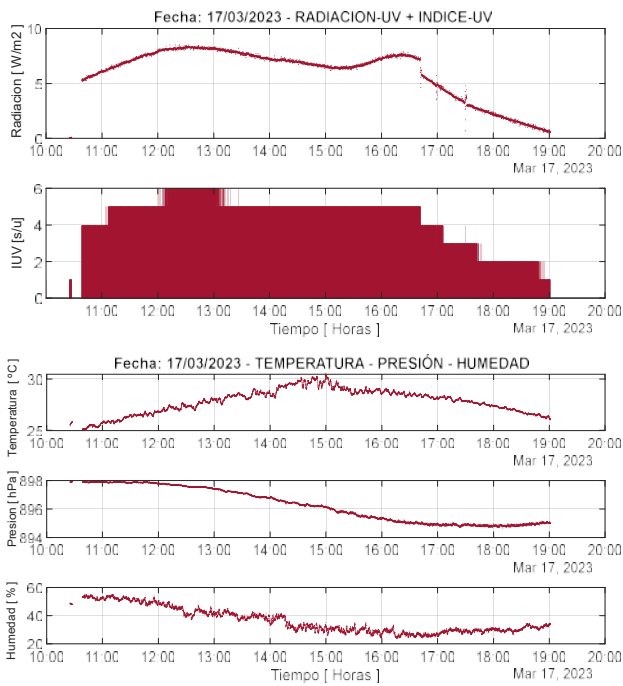


Figura 8. Resultados del 17/03/2023 con presencia de nubes.

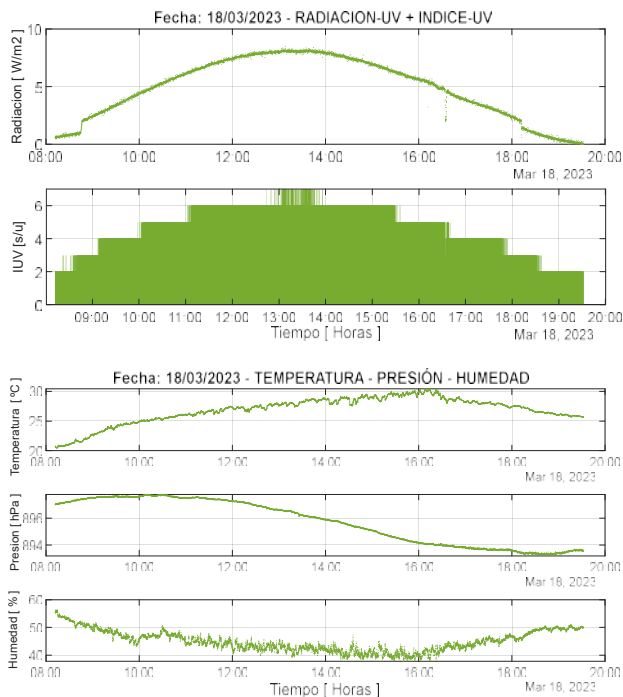


Figura 9. Resultados del 18/03/2023 cielo despejado.

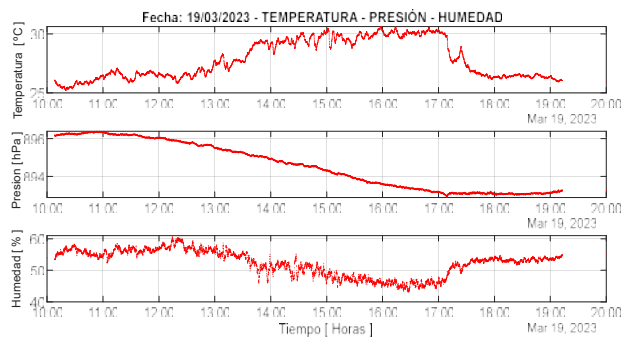
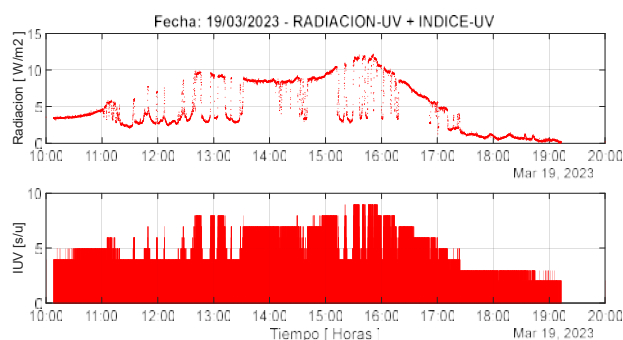


Figura 10. Resultados del 19/03/2023 nubes alternadas.

#### IV CONCLUSIONES

Hay que aclarar que el Índice UV carece de precisión, es un número (generalmente de 0 a 12) o color (5 tonalidades diferentes) fácilmente reconocible para advertir si los niveles detectados pueden o no afectar a la población. Está mundialmente reconocido y es un dato que da el SMN (Servicio Meteorológico Nacional) pero, un número o color no sirve si se da una vez al día, por ello proponemos la construcción de dispositivos económicos fácilmente reproducibles. El prototipo construido operó en óptimas condiciones pese a encontrarse sometido a altas temperaturas y fuertes vientos. Se muestran 3 de los 7 registros de datos adquiridos a partir de las 8:30 hs o 10:00 hs hasta el anochecer (aprox. 19:30 hs). Los datos meteorológicos se condicen con datos locales obtenidos por otros sistemas, con una variación del 5 % de respecto a los datos locales (ubicados a 10 km). Los resultados de la Radiación UV estuvieron acorde con la información provista por el Servicio Meteorológico Local.

#### V TRABAJO A FUTURO

Finalizar con las pruebas del prototipo sobre todo la conexión Wifi, en la actualidad se está construyendo una pre-serie de 3 unidades a ser instaladas en la UTN Facultad Regional Haedo y en la Reserva Natural Urbana de Morón. Se está trabajando en la conexión Wifi ya que la transmisión de datos depende del acceso a la red disponible en la locación.

#### VI AGRADECIMIENTOS

Se agradece a las autoridades de la UTN Facultad Regional Haedo por el apoyo económico y humano.

#### REFERENCIAS

- [1] Estefania Coluccio Leskow "Espectro visible". De: Argentina. Para: Concepto .de. Disponible en: <https://concepto.de/espectro-visible/>. Última edición: 16 de julio de 2021.
- [2] Proyecto Lowtran 7 - <https://apps.dtic.mil/sti/pdfs/ADA206773.pdf>. (Reporte del Proyecto Lowtran 7).
- [3] Enrique D. Albizzati, Germán H. Rossetti y Orlando M. Alfano. "Mediciones de la Radiación Solar UVB y Predicción de la Radiación Solar UV Eritémica Bajo Cielos Claros y sin Nubes". Avances en Energías Renovables y Medio Ambiente Vol. 6, N° 2, 2002. Impreso en la Argentina. ISSN 0329-5184.
- [4] EPA-USA - <https://espanol.epa.gov/>
- [5] ESP32 (NODEMCU32 ESP-WROOM-32s) [https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf)
- [6] ML8511 - <https://datasheetspdf.com/pdf/1188772/LAPIS/ML8511/1>.
- [7] BME/BMP-280-<https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf>.
- [8] UVM-30A - [Interfacing UVM30A UV Sensor Module with Arduino](https://www.circuits-diy.com/interfacing-uvm30a-uv-sensor-module-with-arduino/)

# Prototipo de Sistema de Parqueo Inteligente para la zona Edificio Central de la UPTC sede Tunja

Ilber Adonayt Ruge Ruge  
 Universidad Pedagógica y Tecnológica  
 de Colombia.  
 Grupo de Investigación I2E  
 Tunja, Boyacá  
 ilber.ruge@uptc.edu.co

Ingrid Carolina Ortiz Alvarez  
 Universidad Pedagógica y Tecnológica  
 de Colombia.  
 Tunja, Boyacá  
 ingrid.ortiz05@uptc.edu.co

Fabian Rolando Jiménez López  
 Universidad Pedagógica y Tecnológica  
 de Colombia.  
 Grupo de Investigación I2E  
 Tunja, Boyacá  
 fabian.jimenez02@uptc.edu.co

**Resumen**—El aumento de población en las ciudades, el incremento de flujo vehicular y una infraestructura vial invariante en el tiempo, conlleva a una crisis de movilidad. Las tecnologías de la información y las comunicaciones (TICs) son herramientas que facilitan el desarrollo de sistemas inteligentes para la gestión de espacios públicos como lugares de parqueo. El presente artículo expone el desarrollo de un prototipo de sistema de parqueo inteligente para la zona central de la Universidad Pedagógica y Tecnológica de Colombia sede Tunja. La metodología comprende: Sistema electrónico para identificación de espacios de parqueo, desarrollo de interface gráfico (GUI) y registro de índice de ocupación en memoria EEPROM.

**Palabras clave**—Sistema de parqueo inteligente, Arduino MEGA2560, GUI Visual Studio, Índice de Ocupación en EEPROM.

## I. INTRODUCCION

"En la era del mundo moderno, gran parte de la población tiene un vehículo personal y se ha convertido en una necesidad básica para los seres humanos" [1]. El uso de vehículos aumenta rápidamente cada año, y cada vez es más difícil encontrar lugares de estacionamiento en las ciudades [2-3].

Las zonas de parqueo de la Universidad Pedagógica y Tecnológica de Colombia sede Tunja no es ajena a esta situación. Por tanto, las Escuelas de Ingeniería Electrónica e Ingeniería de Transporte y Vías crean interés colaborativo de saberes para desarrollar un prototipo para la identificación de espacios de parqueo que ayuden a los conductores a encontrar lugares donde haya estacionamiento disponible, y recopilación de información de la dinámica de ocupación que a futuro pueda ayudar a soportar lineamientos y/o políticas institucionales para la gestión y optimización del uso de estos espacios.

La arquitectura del prototipo de sistema de estacionamiento inteligente consta de tres componentes principales: Identificación de espacio de estacionamiento (instrumentación electrónica), aplicación para indicar al conductor el espacio disponible (interface gráfica de usuario GUI) y registro de información de índice de ocupación.

## II. METODOLOGIA

### A. Justificación de la investigación

La Universidad Pedagógica y Tecnológica de Colombia (UPTC), es una universidad oficial del orden nacional con sede principal en la ciudad de Tunja, capital del departamento de Boyacá.

La UPTC alberga cerca de 30.000 estudiantes de pregrado y 2950 de posgrado. Sin considerar el número de profesores y personal administrativo, es evidente la situación crítica por los espacios de estacionamiento. La Fig. 1 ilustra la situación que motiva la realización de la fase prototipo del sistema de parqueo inteligente expuesto en el presente artículo.



Fig. 1. Situación actual de no disponibilidad de espacios de parqueo.

Para el desarrollo del prototipo se considera la zona de parqueo del Edificio Central, que dispone de 120 lugares de parqueo aproximadamente.

### B. Descripción del sistema

El sistema propuesto se utiliza para indicar al usuario sobre la disponibilidad de los lugares de estacionamiento. El prototipo se realiza sobre un modelo a escala del lugar de interés y considerando una escala de 1:4 en términos de lugares de parqueo. Para emular la ocupación de cada lugar de parqueo se dispone de interruptores táctiles capacitivos, que facilitan al usuario la dinámica de ocupación, la opción de visualizar la disponibilidad (color Rojo indica ocupado) y el bajo precio (\$US2.5/5unidades). El interruptor táctil entrega un voltaje de 5V (1 lógico) cuando está activo y 0V (0 lógico) cuando está inactivo: estos valores lógicos serán los que el procesador digital usará para identificar los espacios disponibles de parqueo. La Fig. 2 muestra un esquema del prototipo elaborado para el desarrollo del sistema de parqueo inteligente. El sector A y sector B (zonas demarcadas en color violeta) se han dispuesto de 7 interruptores respectivamente, que en una escala 1:4 (un interruptor representa 4 espacios de parqueo) corresponderán a 28 espacios de estacionamiento en cada sector, para un total de 56 espacios reales de parqueo correspondiente al 50% del total real de espacios de parqueo del sector central (120 en total aproximadamente). Los espacios de parqueo demarcados en color amarillo no fueron instrumentados, dado que el sistema de instrumentación planteado deberá ser escalable para cubrir la totalidad de espacios que se desean monitorear.

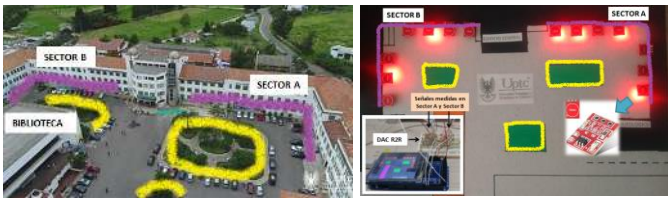


Fig. 2. Prototipo zona de parqueo zona Central UPTC.

C. Diagrama de bloques

La Fig. 3 muestra el diagrama de bloques del sistema de parqueo inteligente propuesto.

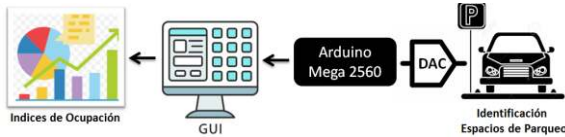


Fig. 3. Arquitectura del sistema de parqueo inteligente (prototipo).

III. MATERIALES Y METODOS

A. Identificación de espacios de parqueo (Instrumentación electrónica)

Para emular la ocupación de los espacios de parqueo se usan interruptores táctiles capacitivos basados en el circuito integrado TTP223 Ton Touch [4], diseñado para reemplazar el tradicional interruptor. Este interruptor presenta un comportamiento similar a un pulsador, pero puede ser activado con poca o ninguna presión, además de proveer el modo de operación Toggle (TOG Pin): ambas características facilitan generar los cambios requeridos en la dinámica de ocupación de espacios de parqueo. Por otro lado, el principio de activación por presión es homologado al planeado para la siguiente fase de desarrollo del sistema de parqueo inteligente (celdas de carga). Otra ventaja de este tipo de interruptores es que no requieren de contacto físico para realizar el disparo, siendo suficiente acercar el dedo a 1-5mm del sensor. Esta otra característica permite escalar y ampliar las opciones de selección de los futuros sensores de detección de ocupación de espacio de parqueo a alternativas como los sensores ultrasónicos.

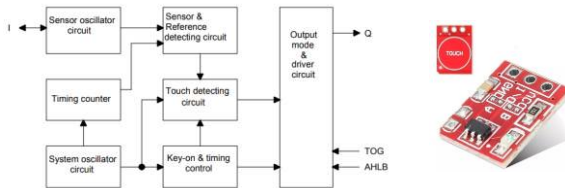


Fig. 4. Diagrama de bloques del interruptor táctil TTP223.

El diseño del sistema electrónico para la detección de espacios de parqueo en la fase de prototipo, debe considerar la escalabilidad al número real de lugares de parqueo (120 espacios de parqueo zona Central UPTC: 56 espacios emulados en escala 1:4 (50% de ocupación)). Se propone el uso de arquitectura de conversor Digital-Analógico (DAC) [5], donde las entradas digitales corresponden a los estados lógicos entregados por los sensores de ocupación (estados emulados por los interruptores táctiles), y la salida será un voltaje analógico proporcional al valor digital generado por

la ocupación de los espacios de parqueo. La Fig. 5 muestra el circuito electrónico propuesto para la medición.

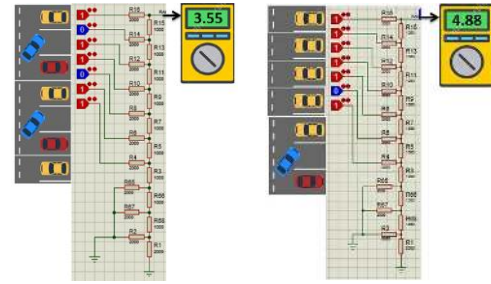


Fig. 5. DAC arquitectura R2R resolución 10 bits para identificación de espacios disponibles en zona de parqueo (prototipo).

Para una zona de medición (ZONA A o ZONA B) se requieren 7 estados lógicos de entrada (7 interruptores táctiles por zona), entonces un solo canal analógico de entrada del procesador es suficiente para determinar los espacios disponibles de cada zona a partir del voltaje analógico correspondiente. Se usa un conversor DAC R2R de 10 bits y se consideran solamente los 7 bits más significativos (D9-D3) para la generación del voltaje analógico (los bits (D2-D0) se conectan a GND): lo anterior para mejorar la resolución de la conversión. El procesador digital requiere de conversor ADC con resolución de 10 bits y dos (2) entradas analógicas para la medición de los voltajes analógicos de las dos zonas de parqueo en el prototipo a una escala 1:4 (Zona A y Zona B).

La board de desarrollo Arduino MEGA2560 [6] dispone de 16 canales analógicos de entrada, siendo una excelente alternativa para la escalabilidad del total de espacios de parqueo que se requieren monitorear (17 canales para monitorear 120 espacios de estacionamiento:  $120_{\text{espacios}}/7_{\text{bits}} = 17.14$ ). De igual manera, dispone de canales de comunicación serial necesarios para la comunicación con la interface gráfica de usuario GUI y periféricos, así como la cantidad de pines digitales entrada/salida para propósitos de validación del prototipo.

Para la validación del proceso de identificación de los espacios disponibles en cada zona de parqueo, se utiliza una pantalla TFT 5" 800x480 HDMI Adafruit 2260 [7] con una sencilla interface gráfica que indica al usuario los espacios de parqueo disponibles. La pantalla representa una imagen del prototipo con indicadores distribuidos de la misma manera que en el prototipo: el color rojo indica "OCUPADO" y el color blanco "DISPONIBLE". La Fig. 6 muestra un ejemplo de validación.

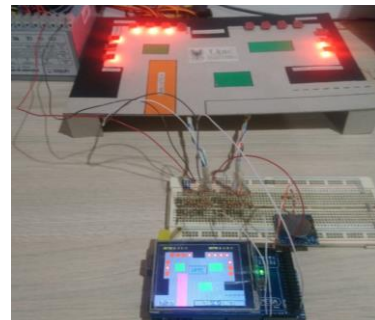


Fig. 6. Validación grafica de identificación de espacios disponibles de parqueo usando pantalla TFT (prototipo).

### B. Interface Gráfica de Usuario (GUI)

El objetivo de la interface gráfica de usuario GUI, es facilitar la información de disponibilidad de espacios de parqueo de manera fácil y efectiva. Por tanto, la plataforma de desarrollo en la primera fase de prototipado debe considerar aspectos como:

- **Accesibilidad:** Posibilidad de acceder a la aplicación desde lugares remotos.
- **Conectividad:** Manera de establecer conexión con el sistema electrónico de identificación de espacios de parqueo.

En términos de accesibilidad, la disponibilidad de la información en una página web facilita al usuario su consulta en tiempo real siempre que disponga de una conexión a internet. Para la conectividad entre hardware/software, se adopta un protocolo de comunicación serial estándar : RS232 es un protocolo simple, compatible con gran variedad de dispositivos de transmisión tanto alámbricos como inalámbricos y de amplia aplicabilidad en plataformas de desarrollo Arduino.

La Interface Gráfica de Usuario (GUI) se desarrolló en Visual Studio 2019 [8]. La aplicación se realiza mediante Windows Forms App (.NET framework) usando lenguaje de programación C#. La comunicación con la tarjeta de desarrollo Arduino MEGA2560 se realiza mediante el Toolbox SerialPort (Component) configurado a 9600 baudios. La Fig. 7 muestra un resultado del sistema inteligente de parqueo (prototipo).

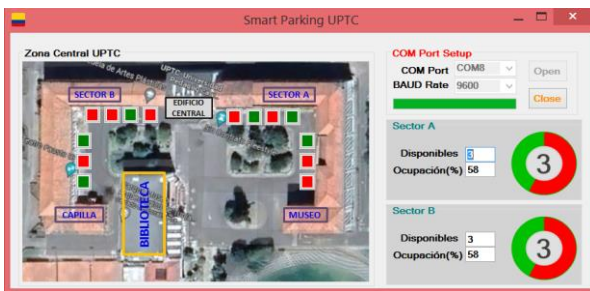


Fig. 7. Aplicación en Visual Studio 2019.

### C. Registro de información de índice de ocupación

El registro de información del índice de ocupación de los espacios de parqueo, es un dato para conocimiento de usuario en tiempo real, y de almacenamiento en memoria ROM con propósitos de análisis estadístico que pueda soportar lineamientos y/o políticas institucionales para la optimización, uso adecuado y eficiente de los espacios de estacionamiento en la UPTC. Los datos para almacenar en memoria ROM son: Fecha (DD/MM/AA), hora (HH:MM) e Índice de Ocupación (% de Ocupación de las zonas de Parqueo Zona A – Zona B). Se usa un reloj- calendario de tiempo real (RTC) de referencia DS1307 [9], usa comunicación síncrona I2C compatible con microcontrolador, alimentación compatible con TTL, modo de operación de bajo consumo de energía que extiende la durabilidad de la batería de respaldo y requiere solamente 8 pines para su conexión. El módulo RTC, contiene además una memoria EEPROM AT24C32 [10],

capacidad de almacenamiento de 4kB, interface I2C y voltaje de operación compatible con TTL.

El mapa de memoria para la gestión de almacenamiento del índice de ocupación en memoria EEPROM se gestiona para que a partir de una fecha específica del año en curso, haya una posición única de memoria. Se registran 8 muestras por día durante horario laboral. Por ejemplo, la muestra\_actual(3) del día en curso corresponde al registro que se hace a las 12am. Así, para el 30 de Enero a las 5pm (Dia(30), Mes(1), Muestra\_Actual(6)) la posición de memoria para almacenar el valor del índice de ocupación en 238 y para el 31 de Diciembre a las 11am corresponde la posición 2970. La ecuación (1) determina el valor de posición de memoria (M) en función del Día/Mes/Hora(Muestra\_Actual):

$$M = ((Dia * 8) + ((31 * 8) * (Mes - 1)) - (8)) + Muestra_Actual \quad (1)$$

### IV. CONCLUSION Y TRABAJO FUTURO

El sistema de parqueo inteligente propuesto está basado en un esquema de identificación de espacios disponibles usando conversor DAC, el cual es un circuito electrónico de bajo costo (usa solo resistencias) escalable a un gran número de espacios que se requieran monitorear. Como trabajo futuro, se plantea la alternativa de usar sensores basados en celdas de carga (identificación por presión) o ultrasonido (identificación por distancia), así como incorporar ajustes del algoritmo por variaciones de temperatura y ruido eléctrico. Por otro lado, la realización de la página web, donde estará disponible la información recopilada en la actual GUI junto con los resultados del análisis estadístico de la información de índice de ocupación almacenada en memoria EEPROM del módulo de monitoreo basado en Arduino MEGA2560.

### REFERENCIAS

- [1] M. Chandran, N. Mahrom, and Y. Wahab, "An IoT based Smart Parking System," International Conference Computer Science and Engineering, 2019. doi:10.1088/1742-6596/1339/1/012044.
- [2] A. Aditya, S. Anwarul, R. Tanwar and S. K. Vamsi, "An IoT assisted intelligent parking System (IPS) for Smart Cities", International Conference on Machine Learning and Data Engineering, Procedia Computer Science 218, 2023, pp. 1045-1054.
- [3] Y. Allbadi, J. N. Shehab and M. Jasim, "The smart parking system using ultrasonic control sensors". 2nd International Scientific Conference of Engineering Science (ISCES 2020), 2021, doi:10.1088/1757-899X/1076/1/012064.
- [4] TTP223 TonTouch, 1 key touch padderector IC, disponible en: [https://datasheet.lcsc.com/szlcsc/TTP223-BA6\\_C80757.pdf](https://datasheet.lcsc.com/szlcsc/TTP223-BA6_C80757.pdf).
- [5] T. Floyd, "Fundamentos de sistemas digitales", 9 Edición, Editorial Pearson Prentice Hall, 2006, pp. 867-872.
- [6] Arduino MEGA2560 Overview. Disponible en: <https://store.arduino.cc/products/arduino-mega-2560-rev3>.
- [7] Adafruit 5" 800x480 TFT HDMI Backpack, disponible en: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-5-800x480-tft-hdmi-monitor-touchscreen-backpack.pdf>.
- [8] Aprender a codificar en Visual Studio. Disponible en: <https://visualstudio.microsoft.com/es/vs/getting-started/>
- [9] DS1307 64x8 Serial Real/Time Clock. Disponible en: <https://www.sparkfun.com/datasheets/Components/DS1307.pdf>.
- [10] AT24C32 2-wire Serial EEPROM, disponible en: <https://ww1.microchip.com/downloads/en/devicedoc/doc0336.pdf>

# Evaluación de Filtros Digitales para la Medición del Flujo de Hidrógeno

Ignacio Zaradnik

*Departamento de Ingeniería e Investigaciones Tecnológicas*  
*Universidad Nacional de la Matanza*  
 Buenos Aires, Argentina  
 izaradnik@unlam.edu.ar

Leandro Jaimes Soria

*Departamento de Ingeniería e Investigaciones Tecnológicas*  
*Universidad Nacional de la Matanza*  
 Buenos Aires, Argentina  
 ljaimessoria@unlam.edu.ar

Rodrigo Spano

*Departamento de Ingeniería e Investigaciones Tecnológicas*  
*Universidad Nacional de la Matanza*  
 Buenos Aires, Argentina  
 rospano@alumno.unlam.edu.ar

**Resumen**— El control del flujo de hidrógeno es un factor determinante tanto en los procesos de generación como de consumo. En el ámbito de la generación, se puede destacar la importancia de la correlación entre la volumen y naturaleza del hidrógeno generado con el tipo y cantidad de energía eléctrica utilizada para dicho fin. En cuanto al consumo, se resalta la necesidad de regular el uso de hidrogeno en la celda de combustible que se encarga de producir energía eléctrica. En este trabajo, se presenta el progreso alcanzado en el desarrollo de un sistema de medición de flujo de gas, diseñado específicamente para aplicaciones en generación y consumo de hidrógeno. Luego de una breve introducción y descripción del hardware utilizado, se detallan los distintos filtros digitales implementados con el fin de reducir el ruido presente en la señal asociada la medición de flujo. Se discuten los resultados obtenidos en relación a la reducción del ruido, la atenuación de la señal y el retardo introducido entre la entrada y la salida, llegando a la conclusión de que lo resultados obtenidos son satisfactorios y extensibles a aplicaciones futuras.

**Palabras claves**— *Sensor de flujo, Hidrógeno, Filtrado digital, Microcontrolador.*

## I. INTRODUCCIÓN

Hoy en día, el 82% de la energía mundial proviene de combustibles fósiles (carbón, petróleo y gas), siendo estos los principales responsables de la emisión de gases de efecto invernadero [1]. Esta situación ha impulsado el interés en el hidrógeno como vector energético a nivel nacional [2] e internacional [3]. El hidrógeno puede utilizarse para almacenar el excedente de energía producido por fuentes renovables y posteriormente reconvertirlo en energía eléctrica. El control del flujo de hidrógeno es crucial tanto en los procesos de generación como de consumo. En el ámbito de la generación, la medición del flujo es importante no solo para el proceso en sí, sino también para certificar que el hidrógeno proviene de fuentes renovables (hidrógeno verde), lo cual posibilita su exportación. Para ello, es necesario desarrollar sensores inteligentes de hidrógeno que permitan la descentralización de la generación energética y la participación de todas las provincias en este cambio tecnológico, fomentando la creación de empleo y fortaleciendo la soberanía tecnológica del país [4].

Los dispositivos de medición de flujo de gas más utilizados incluyen medidores de flujo de masa térmica y medidores de presión diferencial. No obstante, garantizar mediciones precisas y fiables del flujo de gas en todas las condiciones sigue siendo un reto. Se necesitan soluciones innovadoras y técnicas de calibración avanzadas para obtener mediciones precisas y fiables del flujo de gas en tiempo real. En este contexto, el desarrollo de la electrónica de acondicionamiento de la lectura del sensor de flujo de gas y su procesamiento posterior adquiere gran relevancia.

Este informe se enfoca en el filtrado digital de la señal del sensor, con el objetivo de lograr mediciones más precisas y respuestas rápidas en tiempo real.

## II. DESARROLLO

### A. Selección de método de medición y sensor

Como se menciona en [5], los métodos de medición de masa térmica y presión diferencial son los más adecuados para este tipo de aplicaciones. Del mismo modo, aunque centrado en el análisis de factibilidad de un controlador de flujo másico, en [6] se llega a la misma conclusión y se seleccionan sensores específicos para hidrogeno. Debido a los costos y las dificultades para obtener estos dispositivos, se decidió comenzar a trabajar con un sensor no apto para hidrógeno y luego replicar las pruebas en futuros estudios con el sensor adecuado, realizando la medición de flujo de hidrógeno en laboratorios con instalaciones apropiadas. El sensor utilizado fue el FS1012-1100-NG [7], teniendo en cuenta la medición del flujo principal del sistema a través de un capilar, como se describe en [8]. Este sensor se basa en el principio de disipación térmica que ocurre en una pequeña pastilla ubicada dentro del tubo, como se muestra en la Figura N°1. Incluye un elemento calefactor y dos sensores de temperatura. Dependiendo del flujo en el interior del tubo, los sensores reciben mayor o menor cantidad de calor, lo que permite procesar dicha medición y estimar el flujo. Este proceso se basa en el método térmico para la medición del flujo volumétrico [9].

### B. Caracterización y tratamiento de la señal

La Figura N°2 muestra la curva característica de los elementos sensores de temperatura del FS1012-1100-NG en función del flujo que atraviesa el tubo [7]. Si bien es posible utilizar uno solo de ellos (conexión simple), se optó por utilizar ambos elementos sensores en una conexión diferencial debido a las ventajas propias de esta configuración.

Se puede observar que las curvas de cada uno de los elementos sensores son lineales por tramos, presentando la curva de TP1 (termopar 1) seis tramos y la de TP2 (termopar 2) tres tramos. A partir de estas, se determinó la curva diferencial que se implementó en el procesador para calcular el flujo en función de la tensión medida. Esto se llevó a cabo obteniendo, para cada tramo de las curvas mencionadas, las ecuaciones correspondientes y calculando la diferencia entre ellas, dando como resultado una curva con un total de seis tramos lineales, Figura N°3.

Es importante destacar que, antes del filtrado digital de la señal, se acondicionó la señal diferencial para su correcta lectura por el microcontrolador. Para lograr esto, se implementaron una etapa de filtrado analógico y otra de amplificación. El filtro anti-alias diseñado consiste en un



filtro pasa bajos de 50 Hz de orden 4 en una topología Sallen-Key, seguido de un amplificador de instrumentación AD620 y circuitos complementarios para la generación de la tensión negativa necesaria y el ajuste de la ganancia.

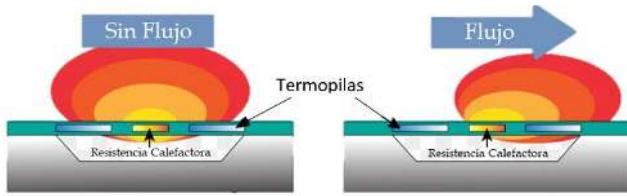


Fig. 1. Principio de funcionamiento del sensor FS1012 (Copyright 2023, Alicat Scientific).

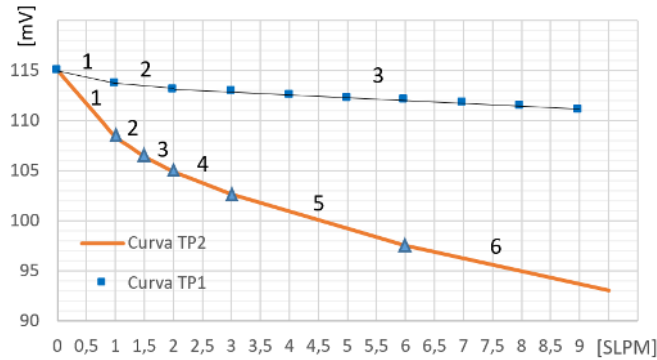


Fig. 2. Curvas características del sensor FS1012.

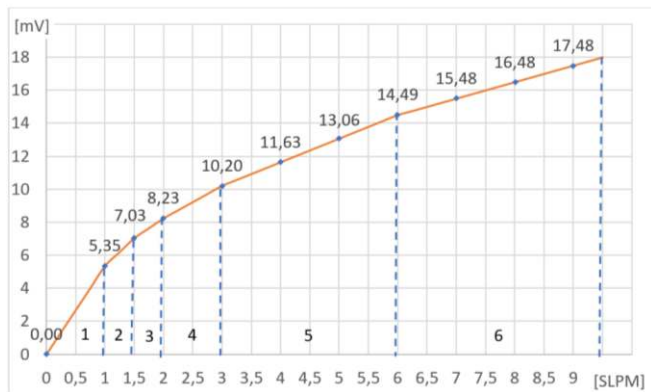


Fig. 3. Curva diferencial del sensor FS1012.

### C. Filtrado digital

Con los niveles de señal adecuados para el ADC, se digitalizó la señal y se implementó un filtro pasa-bajo digital con el objetivo de reducir aún más el efecto del ruido en la señal. Para ello, se calcularon y evaluaron tres topologías de filtros: filtro de media móvil exponencial (1), filtro IIR Chebyshev (2) y filtro IIR Butterworth (3). Donde:

- $Y(n)$  = valor filtrado
- $Y(n - i)$  = valor filtrado previo
- $X(n)$  = muestra actual
- $\alpha$  = coeficiente de ponderación

$$Y(n) = (\alpha \cdot X(n)) + ((1 - \alpha) \cdot Y(n - 1)) \quad (1)$$

El valor del coeficiente de ponderación  $\alpha$  seleccionado fue de 0,05, el cual fue ajustado para obtener una atenuación porcentual mayor al 80% de la señal de entrada. Las ecuaciones en diferencia asociadas a los Chebyshev (2) y Butterworth (3) fueron obtenidas planteando los siguientes

requerimientos: frecuencia de corte 2 Hz, frecuencia de muestreo 100 Hz, ripple en la banda de paso de 0,1 dB y orden del filtro igual a 2.

$$Y(n) = 0.0112 X(n) + 0.0223 X(n - 1) + 0.0112 X(n - 2) + 1.6980 Y(n - 1) - 0.7432 Y(n - 2) \quad (2)$$

$$Y(n) = 0.0036 X(n) + 0.0072 X(n - 1) + 0.0036 X(n - 2) + 1.8227 Y(n - 1) - 0.8372 Y(n - 2) \quad (3)$$

La implementación de estos filtros fue realizada en un Arduino Uno, el cual integra un microcontrolador ATmega328P [10-13]

## III. RESULTADOS

La Figura N°4 muestra los resultados obtenidos de la implementación del filtro de media móvil exponencial, mientras que las Figuras N°5 y N°6 presentan los resultados de los filtros IIR Chebyshev y Butterworth, respectivamente. La Figura N°7 ofrece análisis comparativo de los tiempos de retardo de cada filtro ante las variaciones de la señal. Al analizar la respuesta del filtro de media móvil exponencial (Figura N°4), se observa que mientras la señal muestreada tiene un valor pico-pico de 25 mV, la señal filtrada presenta un valor menor a 3 mV pico-pico, lo que evidencia una atenuación del 88% de la amplitud del ruido. Además, el filtro tiene un retardo de 170 ms en reaccionar a las variaciones de la señal de entrada (Figura N°7). En cuanto al filtro IIR Chebyshev, la señal de entrada presenta un valor pico-pico de 25 mV y la señal filtrada muestra un valor pico-pico de 3,5 mV (Figura N°5), lo que implica una reducción similar al filtro de media móvil (86%). Sin embargo, se nota una atenuación de 7 mV constante entre las lecturas. El tiempo de respuesta del filtro es de 220 ms (Figura N°7). Por último, en la respuesta del filtro IIR Butterworth, se observa que la señal de entrada tiene un valor pico-pico de 25 mV y la señal filtrada es prácticamente continua, con un valor pico-pico menor a 1 mV (atenuación del 96%). Asimismo, se aprecia una atenuación constante de 5 mV (Figura N°6) y un tiempo de respuesta de 380 ms (Figura N°7).

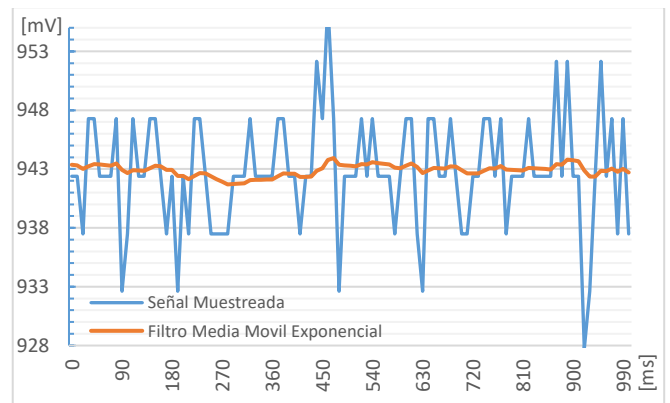


Fig. 4. Respuesta del filtro de media móvil exponencial.

## IV. DISCUSIÓN

Al analizar los resultados obtenidos, es fundamental tener en cuenta las aplicaciones específicas y los requisitos de diseño al elegir el filtro más apropiado. Si el objetivo principal es minimizar el ruido en la señal, el filtro IIR Butterworth sería la opción más conveniente. Sin embargo, si se necesita una respuesta más veloz a las variaciones en la señal de entrada, el filtro de media móvil exponencial sería la mejor alternativa, aunque su capacidad para atenuar el ruido

no es tan efectiva como en los filtros IIR. La atenuación, observada en las mediciones de los filtros IIR, no es un factor que se considere crítico al momento de la selección, ya que a misma puede ser compensada por un ajuste de los coeficientes del filtro o una compensación durante el procesamiento dentro de procesador empleado.

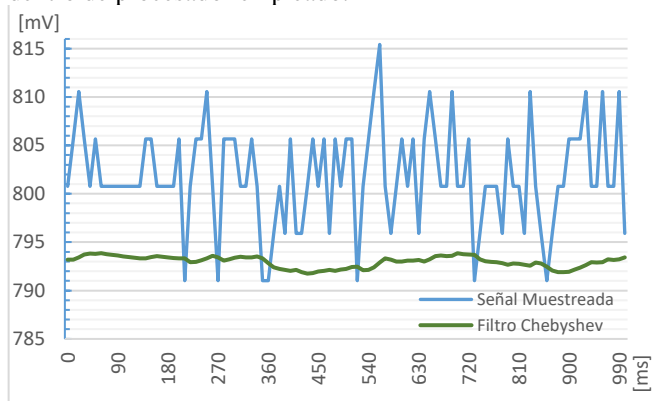


Fig. 5. Respuesta del filtro IIR Chebyshev.

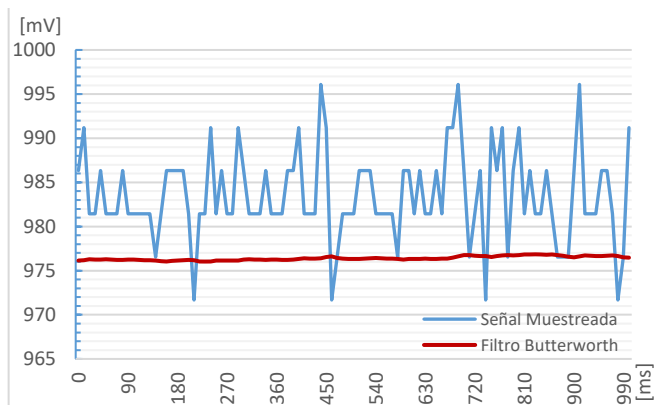


Fig. 6. Respuesta del filtro IIR Butterworth.

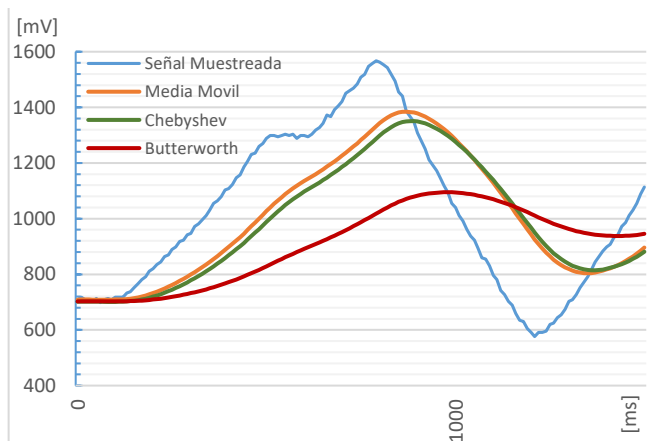


Fig. 7. Tiempo de respuesta de los filtros.

## V. CONCLUSIONES

Se ha logrado la implementación de distintos tipos de filtros digitales para la medición precisa del flujo de hidrogeno. Las opciones evaluadas en este trabajo ofrecen reducciones en los costos respecto a soluciones analógicas ya que aprovechan el procesador disponible en un medidor de flujo. Estas alternativas proporcionaron una supresión del ruido superior al 85% en todos los casos, alcanzando incluso una atenuación máxima del 96%. Los tiempos de retardo obtenidos, que fluctúan entre 170 y 400 ms, resultan apropiado

para monitorear las variaciones de flujo en aplicaciones donde típicamente los controladores trabajan con muestras cada un segundo.

Los resultados obtenidos, sobre la base de un sensor no apto para hidrogeno, inicialmente pueden extenderse al trabajo futuro previsto con el sensor ESRF-HF-600NL-01-02-01-01 de la marca "ES Systems" (ya que posee una salida analógica que requerirá un acondicionamiento similar al presentado). Sin embargo, para garantizar esto, será necesario realizar la caracterización del ruido de ambos sistemas y verificar que tengan el mismo comportamiento.

En el contexto en que se realizó la evaluación presentada, el filtro IIR Butterworth representa la mejor opción, ya que tiene una mejor reducción del ruido y la demora de introducida a la salida no afecta el funcionamiento del sistema.

Este estudio no solo resalta el potencial de las técnicas de filtrado digital en la medición del flujo de hidrógeno, sino que también sienta las bases para futuras investigaciones y desarrollos en este ámbito.

## VI. REFERENCIAS

- [1] Bp. (2022). Statistical Review of World Energy [Online]. Disponible: <https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/statistical-review/bp-stats-review-2022-full-report.pdf>
- [2] Honorable cámara de diputados de la Nación, República Argentina. (2019). Proyecto de Ley de Promoción de la Movilidad Sustentable [Online]. Disponible: <https://www.hcdn.gob.ar/proyectos/textoCompleto.jsp?exp=4602-D-2019&tipo=LEY>.
- [3] International Energy Agency. (2021). Hydrogen in Latin America [Online]. Disponible: <https://www.iea.org/reports/hydrogen-in-latin-america>.
- [4] Consejo Económico y Social, República Argentina. (2021). Hacia una estrategia nacional de Hidrógeno 2030 [Online]. Disponible: [https://www.argentina.gob.ar/sites/default/files/segundo\\_documento\\_ces\\_hidrogeno.pdf](https://www.argentina.gob.ar/sites/default/files/segundo_documento_ces_hidrogeno.pdf)
- [5] Fauroux, Degaetani y Juárez. "Análisis comparativo de sensores para flujo de hidrogeno". Revista Digital del Departamento de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de la Matanza, num 7.
- [6] Leandro Soria, Ignacio Zaradnik, Rodrigo Spano, Diego Brengi, Alejandro Martínez. "Selección de sensores y actuadores para un controlador de flujo másico de hidrogeno". Libro de actas 6º Congreso Argentino de Ingeniería y 12avo Congreso Argentino de enseñanza de Ingeniería, Resistencia (Chaco) y Corrientes (Capital), 2022.
- [7] Renesas Electronics Corporation. (2018). Datasheet Gas and Liquid Flow Sensor Module FS1012 [Online]. Disponible: <https://www.renesas.com/us/en/document/dst/fs1012-datasheet?r=344046>
- [8] Marcelo Juárez, Juan P. Vázquez, Leandro Jaimes Soria. "Consideraciones generales para el desarrollo de caudalímetros para hidrogeno". XV COINI 2022 – Congreso Internacional de Ingeniería Industrial, Mar del Plata Argentina, 2022.
- [9] Alicat Scientifics. (2023). Types of gas flow instruments [Online]. Disponible: <https://www.alicat.com/choosing-an-instrument/types-of-gas-flow-meters/>
- [10] Sophocles J. Orfanidis. "Exponential Smoothing" en Applied Optimum Signal Processing, Rutgers University, 2018, pp 221-314.
- [11] S. Bakshi et al., "Design and Comparison Between IIR Butterworth and Chebyshev Digital Filters Using Matlab," 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2019, pp. 439-446.
- [12] M. Sandhu, S. Kaur, J. Kaur. "A Study on Design and Implementation of Butterworth, Chebyshev and Elliptic Filter with MatLab". International Journal of Emerging Technologies in Engineering Research (IJETER). Volume 4, Issue 6, June (2016), pp 111-114.
- [13] Matlab Mathworks. (2023). Butterworth filter design [Online]. Disponible: <https://www.mathworks.com/help/signal/ref/butter.html>

# Sistema IoT de monitoreo vehicular para el control de flotas

1<sup>st</sup> Juan Nicolás Quiroga<sup>1</sup>

Dpto. de Sistemas<sup>1</sup>

Universidad CAECE Mar del Plata

Mar del Plata, Argentina

nicolasq994@gmail.com

2<sup>nd</sup> Melisa Kuzman<sup>1,2</sup>

Dpto. de Electrónica y Computación - ICYTE<sup>2</sup>

Universidad Nacional de Mar del Plata

Mar del Plata, Argentina

melisakuzman@fi.mdp.edu.ar

**Resumen**—En este trabajo se presentan las bases para el desarrollo de un sistema web para controlar y monitorear el estado de una flota de vehículos. El propósito es presentar información a los usuarios correspondiente a la velocidad del vehículo, nivel de combustible disponible, cantidad de carga transportada, botón de pánico, estado del airbag e inhabilitación de encendido. En el trabajo se abordan aspectos del diseño del sistema embebido como la adquisición y procesamiento de datos con una ESP32, y su presentación al usuario usando Node-RED.

**Palabras claves**—ESP32, IoT, monitoring, vehicles, connectivity, Node-RED.

## I. INTRODUCCIÓN

El advenimiento de las tecnologías IoT, hacen posible monitorear en forma remota las actividades y los parámetros de funcionamiento y operación de vehículos. Las organizaciones que cuentan con parques vehiculares, necesitan disponer y conocer en tiempo real datos tales como la posición, el desplazamiento y el estado general de los vehículos. Los beneficios que se obtienen con la aplicación de esta tecnología, son la mejora de la seguridad, la reducción de los gastos de operación, el control del consumo de combustible, incremento en la productividad y en términos generales mejorar la eficiencia de los procesos administrativos involucrados.

El propósito de este proyecto es desarrollar un sistema para monitorear y controlar el estado de una red de camiones de transporte de cargas. En el sitio web se observará la información correspondiente a la velocidad del vehículo, nivel de combustible disponible, cantidad de carga transportada, estado del vehículo (si está encendido o no), botón de pánico (para eventos no deseados), estado del airbag, inhabilitación de encendido y geolocalización. Además, permitirá el envío de notificaciones de alerta por medio del correo electrónico, la exportación de datos en una planilla de cálculos para su posterior análisis y la personalización de la interfaz por parte los administradores del sistema.

Con los objetivos mencionados, el sistema se compone por un microcontrolador, un módulo de internet a través de datos telefónicos y GPS integrado, un sensor de carga, un escáner bluetooth para la conexión con el vehículo, un servidor web basado en Node-Red con una base de datos SQL. Para la comunicación entre el microcontrolador y el servidor web se utiliza un servidor de mensajes MQTT como bróker.

## II. DIAGRAMA DEL SISTEMA

En la Fig.1 se aprecia un esquema de los componentes más relevantes del sistema. Como primera instancia el vehículo se conecta mediante el puerto OBD2 al escáner ELM327 [1]. Este último se empareja mediante bluetooth con el microcontrolador ESP32 [2], quién envía los comandos correspondientes para consultar y registrar los datos de interés, junto con la información adquirida del sensor de carga y de geolocalización. Al enviar los datos a través de un bróker MQTT que se encuentra en la nube, es necesario contar con internet. Con este fin, se conecta por interfaz serie un módulo de telefonía que cuenta con conectividad a internet.

Desde el lado del usuario se crea una interfaz visual desarrollada con Node-RED para configurar los parámetros de interés del sistema remotamente, como el nombre del vehículo, la información relevante, los umbrales de alarmas, entre otros. Los datos de configuración y el recorrido de cada vehículo son almacenados en una base de datos relacional MySQL. Este diseño posibilita el envío de señales de control remotas hacia el cliente, tales como el apagado del móvil a distancia.

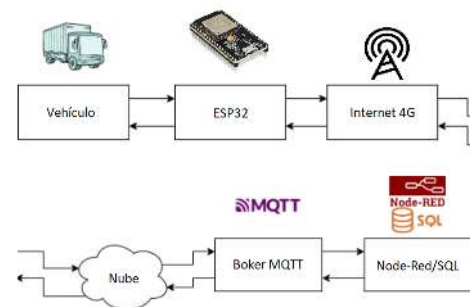


Fig. 1. Esquema general

## III. MÓDULOS CIRCUITALES

### A. Microcontrolador ESP32

La elección del microcontrolador es fundamental para el desarrollo del proyecto debido a que en el mismo se centralizan la adquisición de datos del vehículo, la interconexión

entre módulos y la comunicación con el servidor. Para realizar estas tareas se elige al Microcontrolador ESP32 debido a las siguientes características:

- Conectividad Bluetooth integrada en su chip. Esto permite conectarlo con el escáner OBD2 con bastante facilidad.
- Tiene 34 entradas/salidas de propósito general, las cuales son más que suficientes para conectar los sensores y actuadores del proyecto.
- Posee 3 UARTs por hardware. Permite conectarse al módulo de conectividad a internet y GPS, y otro para debug.
- Bajo costo en el mercado.
- Tamaño es reducido para las características que ofrece integrando varias tecnologías.

### B. Escáner ELM327 OBD2

La mayoría de los vehículos producidos hoy en día proporcionan una interfaz de conexión para equipos de diagnóstico. Los datos de transferencia de estas interfaces siguen varios estándares, pero ninguno de ellos puede usarse directamente en dispositivos como PCs o Smartphones. Para evitar esto se utiliza el escáner ELM327, que está diseñado para actuar como un puente entre estos puertos de diagnóstico a bordo (OBD2) y otro dispositivo. Para comunicar este escáner con el microcontrolador se utiliza la librería ELMDuino [3], permitiendo usar comandos sencillos para consultar las variables de interés en este trabajo, tales como velocidad del vehículo, el estado del ABS, consumo de combustible, entre otros.

### C. Módulo de conectividad a internet y GPS

Debido a que la información se debe transmitir a la nube, es necesario brindar soporte de internet a cada uno de los vehículos, tanto para transmitir como para recibir la información. Por ese motivo, y dado a su gran cobertura incluso en zonas lejanas a los centros urbanos, se elige el módulo SIM7600 [4] que soporta comunicación 2G, 3G y 4G, incluyendo aquellas que pertenecen a las bandas utilizadas en Argentina. Además, para garantizar la seguridad de cada móvil se registra su geolocalización en tiempo real. Este módulo integra tecnología GPS, y a través de comandos serie se obtienen de forma directa los datos de latitud, longitud a donde se encuentra el vehículo.

### D. Celda de carga

En la mayoría de los vehículos no existe un sensor para medir el peso de la carga transportada, por lo que es necesario agregar un sensor para ello. Con el fin de simular la carga, y para el desarrollo de este trabajo, se utiliza una celda de carga de 5 kg con un amplificador de señal. Esta se conecta a una entrada analógica del microcontrolador y permite generar diferentes situaciones de cargamento. Con esta información se pueden inferir datos indirectos, como por ejemplo verificar que el material transportado se descargue en el lugar correspondiente.

## IV. CONTROL Y PROCESAMIENTO DE LA INFORMACIÓN

### A. Comunicación MQTT

Para comunicar a los diferentes nodos del sistema se elige trabajar con el protocolo MQTT [5], un método ampliamente utilizado en proyectos IoT. En esta topología estrella existe un nodo central conocido como bróker que se encarga de filtrar y reenviar los mensajes a cada uno de los clientes. Este tipo de mensajería de publicación/suscripción es extremadamente simple y liviana, está diseñada para dispositivos restringidos y redes de bajo ancho de banda, tales como en el presente trabajo. El bróker MQTT utilizado en este proyecto es Mosquitto [6], un software libre y de código abierto, permitiendo una comunicación eficiente entre el microcontrolador ESP32 y el servidor web, lo que asegura una adecuada transmisión de los datos de monitoreo de la flota de vehículos. Para obtener una comunicación más confiable, en el bróker se configuran ficheros de usuarios y contraseñas para evitar que ningún dispositivo sin autenticación accedan a los datos de los clientes.

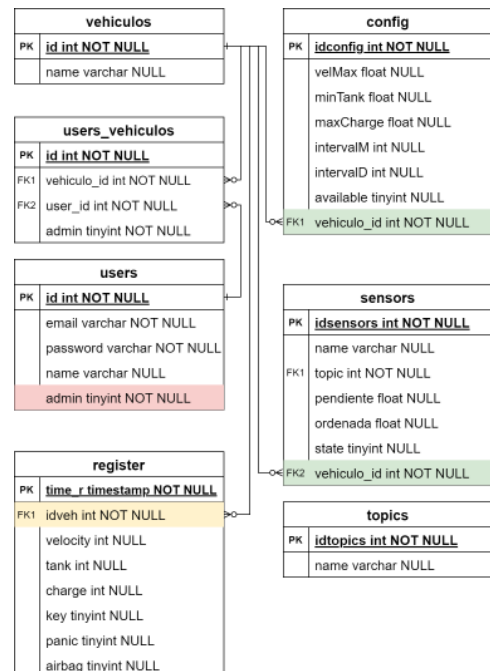


Fig. 2. Base de datos MySQL implementada

### B. Servidor Web y Base de Datos

La interfaz web creada presentada en este trabajo permite monitorear el estado del vehículo en tiempo real, además de acceder al histórico de recorridos del vehículo. Para el desarrollo del servidor web se eligió utilizar Node-Red [7] una interfaz de programación Low-Code, de desarrollo comunitario, basada en nodos utilizada especialmente en IoT. La elección se fundamenta en la sencillez para conectar dispositivos de hardware, API y servicios en línea.

Para almacenar esta información y otros datos del sistema, se usa una base de datos relacional MySQL con la estructura

presentada en la Fig. 2. La conexión entre Node-RED y MySQL se realiza mediante el uso del nodo “node-red-node-mysql”, el cual permite realizar consultas y operaciones en la base de datos de manera sencilla y directa.

El servidor web es un componente fundamental del sistema IoT, ya que se encarga de recibir los datos enviados por el microcontrolador y almacenarlos en la base de datos para su posterior análisis. Además, proporciona una interfaz web para que el usuario pueda configurar los sensores y visualizar los datos de una manera amigable, como se muestra en la Fig. 3.



Fig. 3. Intefaz web

Dentro de sus funcionalidades más relevantes se encuentran:

- Registro y LogIn: A través un correo electrónico y contraseña se acceden a los datos de cada usuario.
- Vehículo: aquí se ingresan los datos provenientes de cada móvil que conforma la flota.
- Configuración: este panel permite configurar los parámetros de adquisición del sistema. El usuario puede agregar, eliminar o modificar los sensores y establecer los umbrales de alerta para cada uno de ellos. También se pueden configurar otros parámetros como la frecuencia de muestreo y controlar la habilitación del encendido. Esto se presenta en la Fig. 4.
- Geolocalización: permite ver en tiempo real la ubicación de cada vehículo de la flota. La librería OpenStreetMaps facilita la visualización en un mapa. La información se visualiza como se presenta en la Fig 5.
- Indicadores: estos proveen una iterfaz visual que simplifica el análisis del estado del vehículo: combustibles, velocidad carga, alarma de pánico y airbag.

## V. CONCLUSIONES

En este proyecto se desarrolló un sistema de monitoreo para vehículos, con el objetivo de mejorar la seguridad y eficiencia de transporte de cargas. Para ello se utilizó un microcontrolador ESP32, un módulo con conectividad a internet y GPS integrado, un escáner OBD2 y tecnología MQTT para la comunicación de datos. Durante el proceso, se logró la implementación exitosa del sistema, permitiendo la captura y envío de datos en tiempo real a través de la red.

En cuanto a los resultados obtenidos, se logró crear un sistema funcional que cumple con los objetivos planteados,



Fig. 4. Panel de configuración

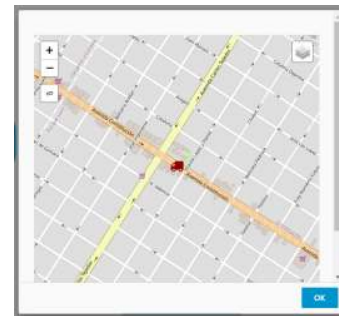


Fig. 5. Geolocalización en tiempo real

usando hardware presente en el mercado local. Además, se sentaron las bases para futuros trabajos en la mejora y evolución del sistema. En este sentido, el acceso a los datos tiene una primera etapa de seguridad con ingreso a través usuario y contraseña, pero el uso microcontroladores más modernos con circuitería especializada en criptografía como el ESP32-S2 permitiría implementar capas de seguridad como TLS en tiempo real, más robusta que al implementarlo por software. Por su parte, la celda de carga utilizada para simular el cargamento deberá ser reemplazada por un transductor que permita medir el peso real transportado en cada vehículo de transporte.

## REFERENCIAS

- [1] (2016, Jul.) Página Web de Elm Electronics. [En línea], Disponible en: <https://www.elmelectronics.com/wp-content/uploads/2016/07/ELM327DS.pdf>
- [2] (2023, Ene.) Página Web Espressif Systems. ESP 32,[En línea], Disponible en: [https://espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)
- [3] (2022, Sep.) Página Web de Arduino. [En línea], Disponible en: <https://www.arduino.cc/reference/en/libraries/arduino/>
- [4] (2022, Dic.) Página Web de SIMCom Wireless Solutions.[En línea], Disponible en: <https://www.simcom.com/product/SIM7600X.html>
- [5] (2022, Ene.) Página Web de MQTT. [En línea], Disponible en: <https://mqtt.org/>
- [6] (2023, Mar.) Mosquitto. [En línea], Disponible en: <https://mosquitto.org/>
- [7] (2022, Jul.) Página Web de Node-RED. [En línea], Disponible en: <https://nodered.org/>

Categoría

# REPORTE

Inteligencia Artificial

Área Temática

# Herramienta de Machine Learning para IoT

1<sup>st</sup> Marcos Darío Aranda

*Decanato de Ciencias Aplicadas  
Universidad Siglo 21*

*Universidad Nacional de Catamarca  
Universidad Nacional de General Sarmiento  
Catamarca, Argentina  
marcos.aranda@ues21.edu.ar*

2<sup>nd</sup> Eduardo Enrique Piray

*Decanato de Ciencias Aplicadas  
Universidad Siglo 21*

*Córdoba, Argentina  
edupiray@gmail.com*

3<sup>rd</sup> Paola Inés Beltramini

*Grupo de Internet de las Cosas - GIoT  
Universidad Nacional de Catamarca*

*Catamarca, Argentina  
pbeltramini@tecno.unca.edu.ar*

4<sup>th</sup> Ricardo Elian González

*Decanato de Ciencias Aplicadas  
Universidad Siglo 21*

*Chaco, Argentina  
eliangonzalez045@gmail.com*

5<sup>th</sup> Gustavo Andrés Futo

*Decanato de Ciencias Aplicadas  
Universidad Siglo 21*

*Buenos Aires, Argentina  
gustavoandresfu@gmail.com*

**Resumen**—El Machine Learning se encarga de generar algoritmos que tienen la capacidad de aprender y no tener que programarlos de manera explícita. En la actualidad el IoT permite la interconexión digital de objetos cotidianos con cualquier otro de su alrededor, éste se ha convertido en uno de los términos más populares en la industria tecnológica, utilizando para ello dispositivos de bajo consumo. Poder lograr que estos dispositivos logren incorporar machine learning, podría ser de gran utilidad para el futuro. Es por ello que el presente trabajo se centra en identificar herramientas que permitan generar modelo de machine learning para la clasificación de imágenes, y luego puedan ser implementado en dispositivos de bajo consumo.

**Index Terms**—Machine learning, IoT, Microcontroladores.

## I. INTRODUCCIÓN

El Machine Learning, es una rama de la Inteligencia Artificial que se encarga de generar algoritmos que tienen la capacidad de aprender y no tener que programarlos de manera explícita. Existe un tipo de aprendizaje llamado supervisado, el cual consiste en entrenar un algoritmo de Machine Learning con datos de entrenamientos etiquetados. Además con el uso de Internet de las Cosas (IoT - Internet of Things), que se entiende como la interconexión digital de objetos cotidianos con cualquier otro de su alrededor, el mismo se ha convertido en los últimos años en uno de los términos más populares en la industria tecnológica gracias al enorme número de dispositivos que cuentan con la posibilidad de interconectarse [9]. Los dispositivos que cuentan con inteligencia artificial pueden ejecutar distintos procesos análogos al comportamiento humano, como la devolución de una respuesta por cada entrada (similar a los reflejos de los seres vivos), la búsqueda de un estado entre todos los posibles según una acción o la resolución de problema mediante una lógica formal [1].

El objetivo de este trabajo es identificar cuales son las herramientas utilizadas para aprendizajes automáticos, que puedan ser utilizadas con un enfoque de código abierto para Machine Learning a gran escala para dispositivos de bajo

consumo. [2]. Con el presente trabajo se busca contribuir a la clasificación inteligente para dispositivos de bajo consumo en animales, para posteriormente el registro de información.

## II. MARCO TEÓRICO

A continuación se realiza una breve descripción de las herramientas identificadas:

### A. Aprendizaje Automático

En la detección de objetos se suelen emplear técnicas de Aprendizaje Automático o Machine Learning, esta área consiste en crear sistemas que pueden aprender por sí mismos, capaces de identificar una serie de patrones complejos a partir de datos de entrada. Este proceso se realiza mediante algún algoritmo de aprendizaje [3]. Existen cuatro tipos de aprendizaje automático:

- Aprendizaje supervisado: Se trata de un tipo de aprendizaje donde al sistema se le alimenta con datos de entrenamiento etiquetados. Una vez se ha completado el entrenamiento, se pueden introducir nuevos datos al sistema que no están etiquetados anteriormente, puesto que al estar entrenado el sistema es capaz de reconocer patrones para identificar los datos correspondientes. En este tipo de aprendizaje están los denominados problemas de clasificación y los de regresión [3].
- Aprendizaje no supervisado: En este tipo de aprendizaje se alimenta el sistema sin datos etiquetados. Uno de los problemas más comunes de este tipo de aprendizaje automático es el de agrupamiento o clustering, en el que el sistema aprende a organizar los datos en distintos grupos, de modo que cada grupo contenga datos similares entre sí [3].
- Aprendizaje semisupervisado: Es una práctica que se encuentra en un punto medio entre el aprendizaje supervisado y el no supervisado. De esta manera, solo se utilizan un grupo mínimo de etiquetas. No obstante,

la mayoría son grupos de datos no etiquetados, pues aumentan los costes, pero son útiles para cumplir con los objetivos. Aunque hay una supervisión sobre cómo actúa la máquina, no es un trabajo que se realiza a lo largo del trabajo con la máquina. Mientras que tendrás que etiquetar algunos resultados manualmente, otros serán propuestas automáticamente por el machine learning [10].

- **Aprendizaje por refuerzo:** Es una práctica que está basada en recompensar los comportamientos deseados, mientras que los no deseados serán penalizados. Es un proceso que está basado en la retroalimentación, pues la máquina aprenderá de las experiencias, a partir del desarrollo y rendimiento [10].

### B. TinyML: Machine Learning

TinyML es una marca registrada de TinyML Foundation, esta asociación promueve el desarrollo de tecnología para aplicaciones de machine learning en dispositivos con capacidades de poco procesamiento [6]. Por tanto, hacer que los microcontroladores puedan ejecutar modelos de machine learning tiene el potencial para abrir nuevos escenarios disruptivos en todas las industrias. Otro aspecto relevante es el impacto ecológico que podría tener la adopción generalizada de las prácticas TinyML, que permitirían reducir el consumo de energía de cara al futuro [5].

### C. Clasificación de imágenes

La tarea de identificar lo que representa una imagen se denomina clasificación de imágenes. Un modelo de clasificación de imágenes está entrenado para reconocer varias clases de imágenes. Por ejemplo, puede entrenar a un modelo para que reconozca fotografías que representen diferentes de animales. TensorFlow Lite proporciona modelos optimizados previamente entrenados que puede implementarse en aplicaciones móviles [7].

## III. HERRAMIENTA

### A. TensorFlow Lite

Es un framework utilizado para implementación de modelos de aprendizaje automático que cuentan con poco poder de procesamiento entre ellos los dispositivos móviles, microcontroladores y dispositivos de IoT [4]. Entre las características más relevante en TensorFlow Lite podemos encontrar:

- **Optimizado para el aprendizaje automático integrado en el dispositivo:** No existe transferencia de datos con un servidor, logrando disminuir la latencia. Los datos no salen del dispositivo. Con respecto a la conectividad no se necesita conexión a Internet, el tamaño del modelos y el binario son reducidos y además se obtienen mejoras en el rendimiento del consumo de energía.
- **Compatible con plataformas iOS, Android, Linux embebido y microcontroladores.** Soporta lenguajes Java, Swift, Objective-C, C++ y Python.
- **Rendimiento para la aceleración de hardware y optimización de modelos.**

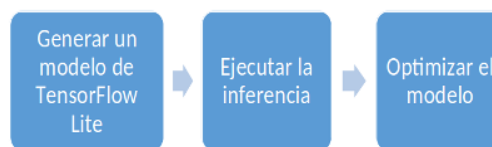


Figura 1. Flujo de trabajo de TensorFlow Lite.

Con la incorporación del aprendizaje automático en microcontroladores, podemos potenciar la inteligencia de millones de dispositivos que usamos en nuestra vida, incluidos los electrodomésticos y dispositivos de la Internet de las Cosas, sin depender de hardware costoso ni de una conexión a Internet estable, que suele estar condicionada por restricciones de ancho de banda y energía, y que además, genera una latencia alta. Esto puede contribuir también a preservar la privacidad, ya que ningún dato sale del dispositivo [4]. TensorFlow Lite para microcontroladores se codificó en C++ 11 y requiere una plataforma de 32 bits. Se probó de manera exhaustiva con muchos procesadores basados en la arquitectura de la serie ARM Cortex-M y se trasladó a otras arquitecturas, como ESP32. El marco de trabajo está disponible como una biblioteca de Arduino. También puede generar proyectos para entornos de desarrollo, como Mbed. Es de código abierto y se puede incluir en cualquier proyecto de C++ 11 [8].

## IV. DESARROLLO

### A. Búsqueda de datos

Para este trabajo se han utilizado para realizar pruebas de entrenamiento un dataset de Open Images Dataset V7<sup>1</sup>, eligiendo la categoría Sheep. Posteriormente se procedió a la descarga de categoría de las imágenes. Utilizando la herramienta OIDv4 ToolKit se procedió a la descarga de la misma.

### B. Etiquetar imágenes para TensorFlow

Las etiquetas para TensorFlow deben estar en formato XML, para generar cada etiqueta XML asociada a cada imagen usamos la herramienta labelImg<sup>2</sup>.

### C. Entrenamiento

Para realizar el entrenamiento se utilizó Google Colab, el cual es un servicio colaborativo, que permite escribir y ejecutar código de Python en el navegador, el cual cuenta con las siguientes características:

- No requiere configuración previa.
- Se puede acceder a una GPU sin costo alguno.
- Facilidad para compartir.

<sup>1</sup>Servidor de imágenes de aproximadamente 9 millones de imágenes con etiquetas, cuadros delimitadores de objetos, máscaras de segmentación de objetos, relaciones visuales y narraciones localizadas.

<sup>2</sup>Es una herramienta para etiquetar cuadros delimitadores de objetos en imágenes.





Figura 2. Etiquetado de imágenes con labelImg.

Una vez entrenado el modelo se exporto a TensorFlow Lite, para implementar el modelo en un dispositivo de bajo procesamiento. Como continuidad de este trabajo se espera validar el modelos y obtener resultados en dispositivos de bajo consumo.

#### V. CONCLUSIONES

En este trabajo se describio una herramienta de aprendizaje automático que pueden ser utilizada para la implementación de modelos TinyML en dispositivos de bajo consumo. Se identifico una herramienta de código abierto como TensorFlow y lenguajes de programación como python que brindan bastante documentación para la implementación de modelo.

Como trabajo a futuro se esperar validar el modelo de

clasificación de imagen, generado con TensorFlow Lite en dispositivos con poco poder de procesamiento con dispositivos móviles, microcontroladores o dispositivos de IoT.

#### AGRADECIMIENTOS

Agradecemos a las autoridades de la Universidad Siglo 21 por el apoyo incondicional para el desarrollo de la investigación.

#### REFERENCIAS

- [1] Julián Pérez Porto and Ana Gardey: Definición de inteligencia artificial. Qué es, Significado y Concepto : <https://definicion.de/inteligencia-artificial/>: 28 11 2022
- [2] Michael Bowles: Machine Learning in Python® : Essential Techniques for Predictive Analysis. Published by John Wiley and Sons, Inc: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119150202>. 15 02 2023
- [3] Abu-Mostafa, Yaser S: Learning from data: a short course (2012)
- [4] TensorFlow Lite : <https://www.tensorflow.org/lite/guide?hl=es-419> : 02 06 2023
- [5] TinyML : <https://www.sngular.com/es/tinyml-machine-learning/>: 20 03 2023
- [6] Pete Warden and Daniel Situnayake: TinyML : Published by O'Reilly Media, Inc. : 2020
- [7] Clasificación de imágenes : <https://www.tensorflow.org/lite/examples>: 20 03 2023
- [8] TensorFlow microcontroller: <https://tensorflow.org/lite/microcontrollers> : 02 06 2023
- [9] Marcos D. Aranda, Paola I. Beltramini, Jesús E. Cano, Luis D. Villagrán, Juan P. Moreno, Sergio H. Gallina, Oscar Ariel Herrera Conegliano : IoT aplicado a la ganadería extensiva : RADI (PDTS) ISSN 2314-0925 : Volumen 17: 02 05 2021
- [10] Aprendizaje Automático : <https://www.keepcoding.io/blog/tipos-de-aprendizaje-automatico/> : 27 04 2023

# Segmentación de objetos en interiores a partir de sensores ToF de bajo costo

1<sup>er</sup> Nicolás Urbano Pintos  
*Grupo TAMA - UTN FR Haedo*  
*DRL - CITEDEF*  
 Villa Martelli, Argentina  
 urbano.nicolas@gmail.com

2<sup>do</sup> Héctor Alberto Lacomi  
*Grupo ASE - UTN FR Haedo*  
*DRL - CITEDEF*  
 Villa Martelli, Argentina  
 hlacomi@citedef.gob.ar

3<sup>er</sup> Mario Blas Lavorato  
*Grupo TAMA*  
*UTN FRH*  
 Haedo, Argentina  
 mlavorato@frh.utm.edu.ar

**Resumen**—La segmentación de entornos interiores es fundamental para la detección y clasificación de objetos en aplicaciones de robótica. La información provista por cámaras monoculares es insuficiente, ya que no posee datos de profundidad, por este motivo se utilizan cámaras tipo ToF (Time of Flight) de alta resolución. Dichos dispositivos proporcionan un mapa de profundidad de gran definición, que luego se segmenta con técnicas de aprendizaje automático. Las cámaras ToF tienen costos restrictivos para su utilización en equipos de bajo costo. En este trabajo, se estudia la segmentación de objetos en entornos interiores a partir de sensores ToF de baja resolución y bajo costo. A partir de la información obtenida de 3 sensores ToF de 8x8 píxeles cada uno, se segmenta el entorno empleando, en primer lugar, el algoritmo RANSAC (Random sample consensus) para la búsqueda de planos y luego se aplican diversas técnicas de agrupamiento basadas en aprendizaje automático, evaluando su desempeño. El objetivo de este trabajo es allanar el camino para la implementación de este tipo de sensores ToF en tareas de reconocimiento de objetos para dispositivos electrónicos de bajo costo.

**Palabras Claves**—Segmentación de objetos, Sensores ToF, Aprendizaje automático

## I. INTRODUCCIÓN

En aplicaciones como la navegación autónoma, la robótica industrial, o los robots de rescate y búsqueda urbana, es necesario conocer el entorno circundante. Es por ello que estos dispositivos disponen de sensores con distintas tecnologías complementarias como son: cámaras digitales, sensores Radar, sensores de ultrasonido, sistemas LIDAR (Light Detection and Range) y cámaras ToF [1]. Las cámaras ToF entregan un mapa de profundidad, en la que cada píxel tiene asociado la información correspondiente a la distancia. El costo de estas cámaras puede ser una restricción a la hora de usarlas en sistemas de bajo costo. Por otra parte, en dispositivos donde el tamaño, el peso y el consumo de energía son críticos, como el caso de los teléfonos móviles, las características de estas cámaras hacen que su implementación sea dificultosa. Por esta razón, la aparición de nuevos sensores ToF de bajo costo, poco consumo energético, y tamaño diminuto, están abriendo el camino a nuevas aplicaciones [2] [3]. En la actualidad, se encuentran en el mercado diversos sensores ToF, como es el caso del sensor VL53L5CX del fabricante STMicroelectronics (ST), se encuentra disponible en el mercado desde mediados del año 2021 con un costo aproximado de 5 USD, un

tamaño del orden de los milímetros, y un consumo en modo automático de unos 11mW. El mismo posee una resolución de 8x8 píxeles y puede medir distancias de hasta 4 metros. La información se presenta como un mapa de profundidad [4]. Respecto a otros sistemas de adquisición de información de 3 dimensiones, como es caso de la visión estereo [5], estos sensores presentan la ventaja de no verse afectados por sombras o deficiencias en la iluminación.

Los mapas de profundidad se pueden representar como nubes de puntos en coordenadas cartesianas, para analizarlas es necesario agrupar los puntos que cuenten características similares en regiones homogéneas, y de este modo segmentar la información. Este desarrollo es el punto inicial para la detección y clasificación de objetos, y/o extracción de características.

En este trabajo se adquiere información espacial de escenarios interiores con diferentes objetos, a partir de 3 sensores VL53L5CX superpuestos, de modo de aumentar la resolución espacial. Y, a partir del algoritmo RANSCAC [6] se encuentran los planos, que representan a las paredes o fondos, y con las técnicas de agrupamiento K-Means [7] y DB-SCAN [8] se segmenta de los objetos. Contribuciones:

- Evaluación de técnicas de aprendizaje automático en nubes de puntos extraídas de nuevos sensores ToF de bajo costo.
- Generación de un conjunto de datos de escenarios interiores [9] capturados con cámara RGB y sensores ST VL53L5CX para futuras evaluaciones de algoritmos de segmentación y/o detección de objetos, y publicación en línea de una aplicación web [10] para la evaluación de los conjuntos de datos con hiperparámetros ajustables.

## II. EXPERIMENTACIÓN

### A. Obtención de mapas de profundidad y transformación a nubes de puntos

A partir de la interfaz gráfica de evaluación provista por ST [11], se midieron 8 escenarios, con objetos distintos, utilizando las placas de desarrollo P-NUCLEO-53L5A1 del mismo fabricante. Además, sobre la placa de expansión, se montaron 2 placas de desarrollo VL53L5CX-SATEL. De este modo, se adquirió la información de los escenarios con 3

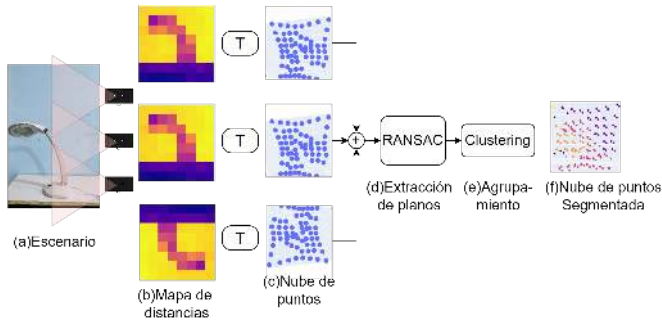


Fig. 1: Diagrama de obtención de datos y procesamiento

sensores cuasi superpuestos. En la Fig. 1.a se observa la imagen de un escenario, representado por una lámpara de escritorio. A continuación, en (b) se observan 3 mapas de profundidad obtenidos por los sensores ToF.

Para realizar la conversión de cada uno de los mapas de profundidad a nubes de puntos se utilizó la Ec. 1 recomendada en la hoja de datos del fabricante.

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} \cos \alpha_k \cos \theta_i \frac{d_k}{\sin \theta} \\ \sin \alpha_k \cos \theta_i \frac{d_k}{\sin \theta} \\ d_k \end{bmatrix} \quad (1)$$

Siendo  $x_i$ ,  $y_i$  y  $z_i$ , las coordenadas cartesianas de la nube de puntos. Donde  $d_k$  representa la distancia medida por cada elemento de sensor,  $\alpha$  el ángulo de inclinación, y  $\theta$  el ángulo de acimut. Para evaluar los algoritmos presentados en este trabajo se realizó una segmentación manual a partir de la observación de las nubes de puntos y delimitaron las clases utilizando restricciones en las coordenadas cartesianas.

### B. Corrección del sistema de coordenadas de los sensores ToF

Respecto a la posición relativa entre sensores, existe un sensor central y 2 sensores laterales (derecha e izquierda). El sensor central es la posición  $(0, 0, 0)$  del sistema, y a partir de la información provista por el fabricante de las placas de desarrollo, se obtuvo la posición relativa de los sensores derecha e izquierda. Con dichos valores se realizó un desplazamiento en las coordenadas  $x$  e  $y$  de las nubes de puntos obtenidas por los sensores laterales. Para comprobar la corrección, se realizó una medición de una placa plana (formada por una madera rectangular montada en un trípode), como se observa en la Fig. 2a. La placa se encuentra a una distancia de 1 m de los sensores, en forma individual se computaron los puntos correspondientes al plano de la placa de cada sensor. Luego se calcularon los coeficientes de los 3 planos con RANSAC y se graficaron de forma conjunta restringido por los puntos correspondientes. En la Fig. 2b se observan que los planos son coincidentes, y que representan en el espacio el mismo objeto. Siendo azul, la medición obtenida por el sensor central, y en rojo y verde las obtenidas por los sensores laterales. Vale destacar, que la placa de ensayo se encuentra centrado con el sensor central, por este motivo el plano azul representa a la placa completa, y los planos azul y rojo representan una porción de la placa de ensayo.

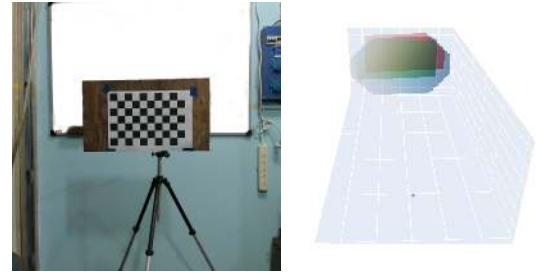


Fig. 2: Comprobación de coordenadas

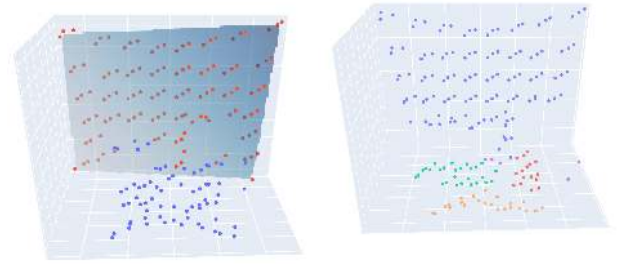


Fig. 3: Detección de pared y agrupamiento

### C. Detección de planos y agrupamiento

Para la detección de planos se utiliza RANSAC [6], el cual es un método iterativo para calcular los parámetros del modelo matemático de un conjunto de datos observados que contienen valores atípicos. En este caso se eligen 3 puntos del conjunto de datos aleatoriamente y se determina un plano. Luego se mide la distancia de cada punto a dicho plano y se cuantifica la cantidad de puntos que están por debajo de un umbral de distancia. Estos pasos se realizan de forma iterativa hasta encontrar el plano que mayor cantidad de puntos interiores contiene. Se definió la cantidad máxima de iteraciones en 50 veces. El umbral se determinó computando los puntos vecinos más cercanos a cada punto mediante un Árbol k-dimensional (KD) [12]. En este caso, se determinó  $k$  igual a 8 vecinos. Luego de obtener los 8 vecinos más cercanos a cada punto, se promediaron las distancias a cada punto, fijando este valor como el umbral del algoritmo. En la Fig. 3a se observa el plano encontrado por el algoritmo y en color rojo se grafican los puntos que pertenecen al plano, y en azul los elementos atípicos. Luego de detectar el plano correspondiente a las paredes de los escenarios, se procedió a agrupar los puntos detectados como atípicos. Para dicha tarea se evalúan dos algoritmos, K-Means [7] y DB-SCAN [8]. El parámetro  $K$  de K-means se definió en cada escenario por la cantidad de clases determinadas en la separación manual. En el caso de DBSCAN, se realizó un barrido de  $\text{min\_samples}$  entre 2 y 10, y de  $\text{eps}$  entre 0 y 200. En la totalidad de las combinaciones de parámetros, se calcularon la cantidad de clústeres generados, y se eligieron como candidatos a las combinaciones en donde los clústeres eran iguales a las clases reales de cada escenario. Luego, dentro de los candidatos se evaluó con el índice

Esc.	K Means		DBscan				
	k	Jaccard	Rand	eps	m_s	Jaccard	Rand
1	89	<b>0.9492</b>	<b>0.9245</b>	97	9	0.8199	0.8941
2	99	<b>1.0000</b>	<b>1.0000</b>	67	2	0.9793	0.9969
3	88	<b>0.9793</b>	<b>0.9853</b>	92	9	0.8823	0.8132
4	89	0.7066	0.9116	79	4	<b>0.9492</b>	<b>0.9208</b>
5	94	0.8916	0.9546	83	2	<b>0.9793</b>	<b>0.9796</b>
6	95	0.7534	0.9461	70	4	<b>0.8550</b>	<b>0.9780</b>
7	92	0.648	<b>0.8552</b>	76	2	<b>0.7454</b>	0.8494
8	20	0.5058	0.6837	220	7	<b>0.7611</b>	<b>0.7820</b>

TABLA I: Parámetros y métricas obtenidas en cada escenario

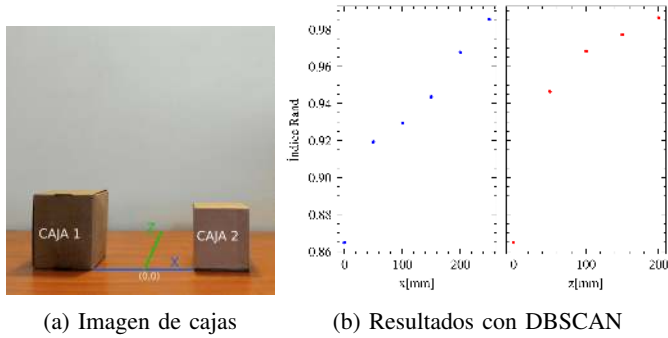


Fig. 4: Ensayo de distancias de cajas

Rand [13], y se seleccionaron aquellas combinaciones de  $\text{min\_samples}$  y  $\text{eps}$  que obtuviera mayor índice. En la Fig. 3b se observa en azul, los puntos pertenecientes a la pared, obtenidos por RANSAC, y en verde, rojo y anaranjado, los grupos detectados por DBscan y en violeta, se representan los puntos que no fueron agrupados.

#### D. Ensayos variando la distancia entre objetos

Además, de los escenarios descritos anteriormente, se realizaron muestras de escenarios formados solamente por 2 cajas, como se observa en la Fig. 4a. En este ensayo se varió la posición relativa entre ambas cajas en 2 ejes, la posición inicial fue ambas cajas contiguas y en el mismo plano. En primer lugar, se separaron ambas cajas en el eje  $x$ , adquiriendo muestras cada 50 mm, hasta los 250 mm. Luego, de forma análoga, y desde la posición inicial, se desplazó la caja 2, hacia el fondo de la escena en el eje  $z$ , con saltos de 50 mm hasta los 200 mm.

### III. RESULTADOS

Para evaluar las técnicas de agrupamiento se utilizó el índice Jaccard [14] y el índice Rand [13], calculados a partir de la segmentación manual. En la Tabla I se observan los parámetros y las métricas obtenidas con K Means y DBscan para cada escenario, en negrita se aprecia el mejor método para cada escenario.

A la izquierda de la Fig. 4b se observan los resultados de la segmentación calculados por el índice Rand, cuando se varía la distancia en el eje  $x$ . A la derecha se observa el índice Rand cuando se varía la profundidad de la caja 2. En ambos casos, se nota que cuando los objetos se encuentran contiguos, el índice Rand baja a valores menores a 0.9 y para distancias

mayores a 50 mm, se obtiene un índice mayor a 0.9. Siendo mayor el aumento del índice cuando las cajas se alejan en el eje  $x$  que cuando se alejan en el eje  $z$ .

### IV. CONCLUSIONES

En este trabajo se logró representar objetos en escenarios interiores a partir de la utilización de 3 sensores ToF, gracias a una transformación de los mapas de profundidad y un desplazamiento de coordenadas. Dicha representación fue comprobada a través de un ensayo de medición de una placa plana. Respecto a la segmentación, el algoritmo RANSAC separó correctamente a los puntos que representan a las paredes de los puntos que representan los objetos en todos los escenarios. En el agrupamiento, en escenarios donde los objetos se encuentran más cercanos, K-means obtuvo las métricas más altas, tal cual se observa en la tabla I para los escenarios 1,2 y 3. En cambio, en escenarios donde los puntos que representan a los objetos se encuentran separados, el algoritmo DBscan obtuvo los mejores resultados. A su vez, se evaluaron los límites del algoritmo DBscan, realizando un ensayo donde se varió la distancia entre objetos (cajas) en 2 ejes. A partir de este ensayo se determinó que el algoritmo DBscan es capaz de segmentar objetos con distancias mayores a 50 mm, con un índice Rand mayor a 0.91. A su vez, se comprobó que la segmentación se comporta mejor cuando varía distancia en profundidad entre los objetos.

### REFERENCIAS

- [1] L. Li *et al.*, “Time-of-flight camera—an introduction,” *Technical white paper*, no. SLOA190B, 2014.
- [2] I. Ostovar, V. Niculescu, H. Müller, T. Polonelli, M. Magno, and L. Benini, “Demo abstract: Towards reliable obstacle avoidance for nano-uavs,” in *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2022, pp. 501–502.
- [3] A. Ruget, M. Tyler, G. M. Martín, S. Scholes, F. Zhu, I. Gyongy, B. Hearn, S. McLaughlin, A. Halimi, and J. Leach, “Real-time, low-cost multi-person 3d pose estimation,” *arXiv preprint arXiv:2110.11414*, 2021.
- [4] “Time-of-flight 8x8 multizone ranging sensor with wide field of view,” March 2023. [Online]. Available: <https://www.st.com/resource/en/datasheet/vl53l5cx.pdf>
- [5] K. Y. Kok and P. Rajendran, “A review on stereo vision algorithm: Challenges and solutions,” *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 13, no. 2, p. 112–128, Aug. 2019. [Online]. Available: <https://ph01.tci-thaijo.org/index.php/ecticit/article/view/194324>
- [6] K. G. Derpanis, “Overview of the ransac algorithm,” *Image Rochester NY*, vol. 4, no. 1, pp. 2–3, 2010.
- [7] D. Sculley, “Web-scale k-means clustering,” in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 1177–1178.
- [8] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, “Dbscan revisited, revisited: why and how you should (still) use dbscan,” *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.
- [9] nurbano, “Dataset segobj-vl53l5cx,” [https://github.com/nurbano/objseg\\_vl53l5cx](https://github.com/nurbano/objseg_vl53l5cx), 2023.
- [10] “Segobj-vl53l5cx app,” 2023. [Online]. Available: <https://objseg-vl53l5cx.streamlit.app>
- [11] “Gui for vl53l5cx expansion boards,” Jun 2021. [Online]. Available: <https://www.st.com/en/embedded-software/stsw-img024.html>
- [12] *Orthogonal Range Searching*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 95–120. [Online]. Available: [https://doi.org/10.1007/978-3-540-77974-2\\_5](https://doi.org/10.1007/978-3-540-77974-2_5)
- [13] D. Steinley, “Properties of the hubert-arable adjusted rand index,” *Psychological methods*, vol. 9, no. 3, p. 386, 2004.
- [14] L. d. F. Costa, “Further generalizations of the jaccard index,” *arXiv preprint arXiv:2110.09619*, 2021.

Categoría

# REPORTE

Internet de las Cosas

Área Temática

# Miniboyas ambientales para monitoreo de ecosistemas acuáticos

Lupi, Oreste Daniel  
Departamento de Ingeniería  
Universidad Nacional de la Matanza  
Buenos Aires, Argentina  
olupi@unlam.edu.ar

Zaradnik, Ignacio José  
Departamento de Ingeniería  
Universidad Nacional de la Matanza  
Buenos Aires, Argentina  
izaradnik@unlam.edu.ar

Slawiski, Javier Martín  
Departamento de Ingeniería  
Universidad Nacional de la Matanza  
Buenos Aires, Argentina  
jlawiski@unlam.edu.ar

Turconi, Diego Horacio  
Departamento de Ingeniería  
Universidad Nacional de la Matanza  
Buenos Aires, Argentina  
dturconi@unlam.edu.ar

Caccaviello, Diego Ignacio  
Departamento de Ingeniería  
Universidad Nacional de la Matanza  
Buenos Aires, Argentina  
dcaccaviello@unlam.edu.ar

**Resumen—** En la actualidad existe una creciente preocupación por el deterioro del medioambiente y por el impacto que determinadas actividades humanas pueden causar sobre él. En especial los factores que afectan a los recursos naturales destacándose el agua, que es un elemento básico para la vida. En función de esto, se propone el desarrollo experimental de un conjunto de miniboyas que efectúen la medición de un conjunto de parámetros adecuados a la situación de control, y que se comuniquen con un sistema remoto de toma de datos. El presente artículo expone los avances logrados en el desarrollo de dicho sistema. Los mismos involucran mejoras en la estructura mecánica, el desarrollo de una interfaz gráfica propietaria y la implementación de una política de bajo consumo. En lo que respecta a las mejoras mecánicas, se describen los cambios realizados y los beneficios obtenidos. Asociado a la interfaz, se resume la estructura implementada y los softwares utilizados. Finalmente se detallan los consumos de los distintos bloques de la miniboya, los tiempos de ejecución de las tareas involucradas en la medición y transmisión de los parámetros y se presenta la secuencia de trabajo implementada con el fin de optimizar el uso de la energía.

**Palabras claves—** Ecosistemas acuáticos, Internet de las Cosas, Medioambiente, Miniboyas

## I. INTRODUCCION

Tal como se ha mencionado en trabajos previos [1][2][3], el agua representa un recurso de vital importancia para el desarrollo socioeconómico, la energía, la producción de alimentos, los ecosistemas y la supervivencia de los seres humanos. Por lo tanto, la medición de su calidad es fundamental. Es tal la importancia que se le da al agua en la Argentina que no sólo el ministerio de Salud se ocupa de este tema, sino que existe el Instituto Nacional del Agua (INA) dependiente del ministerio de obra pública [4]. El mismo intenta satisfacer los requerimientos de estudio, investigación, desarrollo y prestación de servicios especializados en el campo del aprovechamiento y preservación del agua. En función de lo mencionado, se planteó el desarrollo de un sistema de miniboyas para el monitoreo de los ecosistemas acuáticos.

## II. DESARROLLO

### A. Descripción general

El sistema implementado de miniboyas para monitoreo de ecosistemas acuáticos está compuesto por dos elementos: la miniboya y la interfaz gráfica. En el trabajo presentado, se utilizó la red celular (la red 3G más específicamente) como

tecnología de comunicación entre ambos elementos [2]. El módulo celular, al contar con un stack TCP/IP embebido, permitió la conexión de la miniboya a Internet y enviar los datos directamente al servidor donde la interfaz gráfica está instalada. En la Fig. 1 se puede observar el diagrama de bloques de la miniboya.

Dado que la calidad del agua describe la composición general de la misma con referencia a sus propiedades químicas, físicas y biológicas, se estudiaron numerosos trabajos, los cuales se resumen en [1], para analizar los parámetros más adecuados a medir. En función de esto, se decidió que la miniboya integrara sensores de temperatura, pH, conductividad y oxígeno disuelto.

La selección de los productos comerciales asociados a los parámetros a medir se detalla en [2], los cuales fueron: SEN0237-A, el DFR0300-H y el SEN0249. El módulo GNSS empleado es el modelo SL869V2 mientras que, el módulo celular es el modelo UL865-NAD. El microcontrolador utilizado es el ATSAM4S16, el cual se encuentra integrado en una placa de desarrollo Xplained. En lo que respecta a la batería, se hará mención cuando se hable del estudio de consumo del dispositivo.

La integración de estos componentes se realizó a través de dos circuitos impresos. La Fig. 2 presenta una imagen de estos. La placa inferior integra la fuente de alimentación, la cual va a tomar la tensión de las baterías y generar los niveles adecuados para cada componente, las etapas de acondicionamiento de señal de los sensores y permite la conexión del kit de desarrollo del microcontrolador seleccionado. La placa superior integra el módulo GNSS y el módulo celular.

En lo que respecta al firmware, si bien en [2] se presentó el diagrama de flujo del bucle principal y su explicación, el mismo no contemplaba una política de bajo consumo. El firmware actual, una vez inicializado el dispositivo y enviado el primer paquete a la interfaz gráfica apaga el módulo GNSS, el módulo celular y pone al microcontrolador en bajo consumo. Pasado un tiempo configurado a través de la consola, el microcontrolador se despierta, enciende los distintos elementos y transmite los valores de los sensores.

### B. Cambios mecánicos

Como consecuencia de que la estructura mecánica planeada para la miniboya presentó varios problemas se diseñó una nueva, la cual se puede ver en la Fig. 3. La primera de las mejoras consistió en realizar el rediseño completo del

cuerpo principal con el objetivo de obtener un cuerpo superior completamente sellado contra el ingreso de agua, protegiendo la electrónica alojada. A continuación, se incorporaron cuatro flotadores laterales capaces de asegurar la flotabilidad de la boya con todos los sistemas electrónicos alojados en su interior. Adicionalmente, se colocó un alojamiento para batería, el cual no se había tenido en cuenta en el prototipo inicial. El alojamiento se ubicó en la parte central de la boya de forma tal de mantener el sistema balanceado.

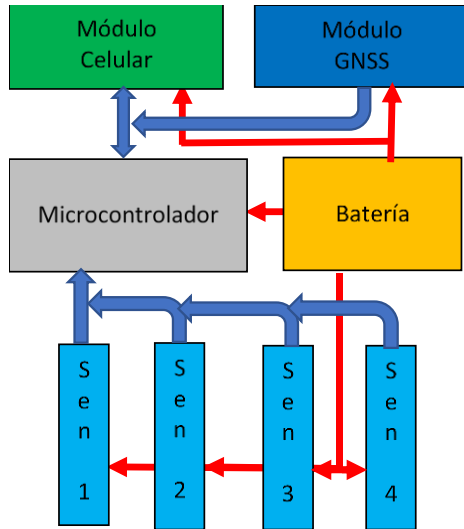


Fig. 1. Diagrama en bloque de miniboya.



Fig. 2. Integración de la electrónica.



Fig. 3. Estructura mecánica de la miniboya.

Con el fin de asegurar el funcionamiento de los sensores y evitar que estos se vean afectados por malezas u otro tipo de elementos presentes en el ambiente de operación, se rediseñó el alojamiento de los sensores colocando cuatro aberturas laterales y una abertura inferior del tamaño adecuado para permitir el ingreso de agua y evitar el ingreso de objetos extraños. Finalmente, se colocó una manija superior con el objetivo de mejorar el transporte de la boya.

### C. Nueva interfaz gráfica

Si bien inicialmente para el desarrollo de la interfaz gráfica del sistema de miniboyas se empleó el servicio de Nube de la empresa Telit [2], como evolución se planteó una interfaz basada en LabWindows, un Broker MQTT (Eclipse Mosquitto) y en bases de datos MySQL. Además, se contemplo que la interfaz grafica pueda recibir datos a través de un puerto TCP para soportar dispositivos que no puedan implementar el protocolo MQTT. Este nuevo enfoque buscó evitar los costos de la plataforma en caso de que se escale la aplicación. La Fig. 4 presenta la arquitectura de la aplicación y la Fig. 5 la pantalla principal. Detalles de la presente interfaz gráfica se encuentran en [3].

### D. Estudio de consumos

El presente estudio buscó caracterizar los consumos de los distintos elementos que conforman la miniboya. El mismo fue realizado en las inmediaciones de la laguna de Chascomús, provincia de Buenos Aires, con cielo parcialmente nublado (condición que afecta la recepción de la señal satelital del módulo GNSS). Con los datos obtenidos se pretendió detectar las fuentes de mayor consumo a fin de poder definir un tiempo de reporte y una batería que maximice la relación entre la autonomía y el costo. Por haberse realizado la integración de la miniboya con kits de desarrollo y placas de evaluación, los datos obtenidos son más bien de carácter cualitativos.

El sistema fue alimentado con una fuente regulada de 30V-3A (UNI-T UTP3313TFL-II alimentada con un invertir de onda pura de 12V y 500VA) ajustada en 7,4V y el consumo fue medido a la salida de esta por medio de un multímetro (UNI-T UT71A). A continuación, se detallan los consumos individuales: electrónica general (fuente y lógica), sensores (temperatura y pH) y microcontrolador operativo (12MHz), 72 mA; electrónica general, sensores y microcontrolador en bajo consumo, 57 mA; módulo GNSS, 73 mA; módulo celular en proceso de registro, 65 mA; módulo celular transmitiendo, 78 mA.

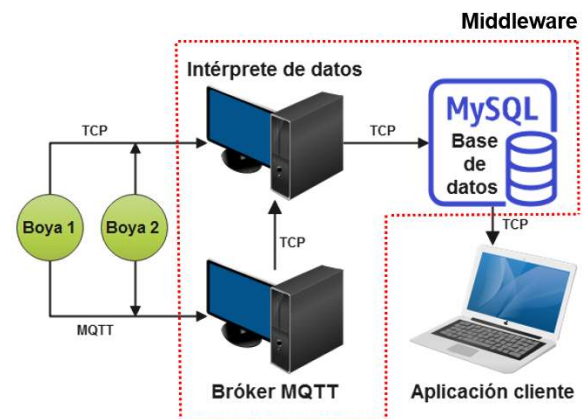


Fig. 4. Arquitectura de la intefaz grafica.

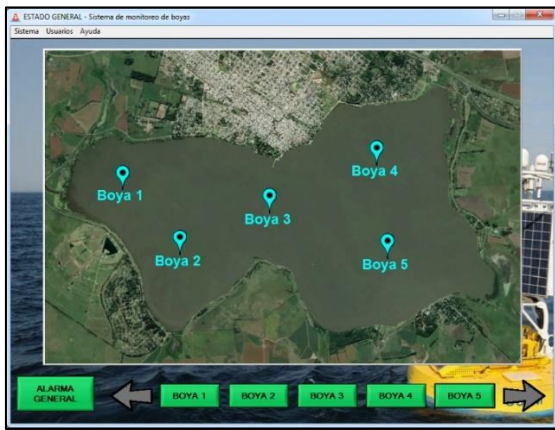


Fig. 5. Pantalla principal de la interfaz gráfica.

Teniendo en cuenta que el tiempo medido en el cual el módulo GNSS toma señal es 60 segundos (cold start), el tiempo en el que el módulo celular obtiene registro es 30 segundos y el tiempo en que este realiza la transmisión de los datos es 30 segundos, se implementó una secuencia en la cual se solapan durante los últimos 30 segundos la obtención de la señal de GNSS y el registro del módulo celular. La tabla I presenta los consumos de cada una de las etapas de la secuencia implementada, mientras que la Fig. 6 presenta un gráfico de consumo con respecto al tiempo en donde se resumen los valores detallados. El tiempo entre que el microcontrolador se activa y comienza el proceso de obtención de señal del módulo GNSS es menor al segundo, por lo que dicho tiempo no se aprecia en el gráfico. El tiempo total en que el sistema se despierta y transmite la información a la interfaz gráfica es de aproximadamente 1,5 minutos.

TABLA I. CONSUMOS EN CADA ETAPA DE LA SECUENCIA IMPLEMENTADA

Etapa	Estado	Consumo
1	Placa encendida / microcontrolador en bajo consumo	57 mA
2	Placa encendida / microcontrolador activo	72 mA
3	Placa encendida / microcontrolador activo / GNSS obteniendo señal	145 mA
4	Placa encendida / microcontrolador activo / GNSS obteniendo señal / modulo celular registrandose	210 mA
5	Placa encendida / microcontrolador activo / modulo celular transmitiendo	150 mA

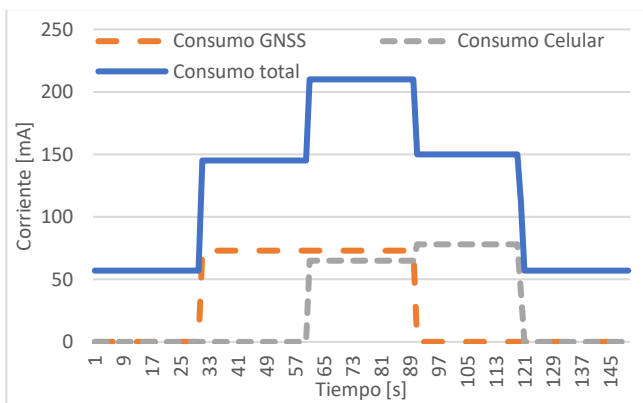


Fig. 6. Gráfica de consumos.

Con estos datos se pudo estimar que por cada ciclo de toma de datos y transmisión se deben estimar 2,775mAh

(extra a lo que consumiría el sistema en bajo consumo), lo cual es consecuencia de la activación de los módulos GNSS y celular. Suponiendo una transmisión por hora y un sistema de alimentación formado por 4 baterías de litio-ion 18650 de 3500mAh (conectadas 2 en serie y luego cada par en paralelo), se calculó una autonomía de aproximadamente 117 horas. Posteriores pruebas realizadas en la Ciudad Autónoma de Buenos Aires y en las localidades de Arrecife, San Martín y La Matanza, de la provincia de Buenos Aires, validaron la aproximación realizada.

### III. CONCLUSIONES

Se lograron avances en el desarrollo de un sistema de monitoreo de ecosistemas acuáticos de carácter experimental: mejora de aspectos mecánicos, desarrollo de interfaz gráfica en lugar de usar una plataforma comercial e implementación de política de bajo consumo.

En lo que respecta a la mecánica de miniboya, si bien se logró mejorar la flotabilidad, así como otros aspectos antes mencionados, se contempló el relleno de los flotadores con espuma de poliuretano expandido.

El desarrollo de una interfaz gráfica propia fue realizado satisfactoriamente en lo que respecta a la comunicación y la presentación de información, existiendo posibilidad de mejora en aspectos estéticos y de seguridad de datos. Se contemplan a futuro la inclusión de mapas dinámicos, ya que lo que se encuentra implementado actualmente es a través de una imagen fija, y de la implementación del protocolo SSL en la comunicación.

Finalmente, en lo que respecta al estudio de consumo, si bien se suponía que los consumos que se obtuvieran no serían adecuados para una aplicación que requiere gran autonomía (mayor a un mes), se ha identificado varios aspectos en los cuales mejorar el hardware implementado para aumentar la autonomía. A continuación, se detallan algunos aspectos que redundarían en la reducción de los 57 mA de piso que se tiene: implementar circuitos de apagado de sensores, uso de fuente de alimentación de mayor eficiencia e integración del microcontrolador en lugar de usar un kit de desarrollo. A fin de reducir el consumo en el ciclo de toma de datos y transmisión las mejoras a implementar serían: utilizar un módulo celular con GNSS integrado, emplear la red 4G LTE CAT M en lugar de la red 3G y utilizar una antena pasiva para el módulo GNSS.

### REFERENCIAS

- [1] Oreste D. Lupi, Ignacio J. Zaradnik, Monica B. Canziani. "Estado del arte de los sistemas de monitoreo de calidad de agua". Revista Digital de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de la Matanza, vol. 5, nro. 2, 2020.
- [2] Oreste D. Lupi, Diego H. Turconi, Javier M. Slawiski. "Monitoreo de ecosistemas acuáticos". Revista Digital de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de la Matanza, vol. 6, nro. 2, 2021.
- [3] Ignacio Zaradnik, Facundo Dominguez, Augusto Kumvich, Oreste D. Lupi, Diego Caccaviello. "Desarrollo de middleware y aplicación cliente para sistema de miniboyas ambientales". Libro de actas XXIV Workshop de Investigadores en Ciencias de la Computación, pp.438-442, 2022.
- [4] Ministerio de Obras Públicas, República Argentina. (n.d). Instituto Nacional del Agua [Online]. Disponible: <https://www.argentina.gob.ar/ina>



# Sistema de gestión de instrumental

Alejandro Aldo Lorenzut

Grupo de Investigación y Desarrollo en Sistemas Embebidos [1]

EEST N° 6 Albert Thomas

La Plata Buenos Aires Argentina

alelorenzut@hotmail.com

**Resumen**—El presente proyecto muestra la implementación de un Sistema de Gestión de Instrumental de laboratorio (SGI) basado en tecnología IoT. El mismo consta de terminales ubicadas en cada laboratorio enlazadas mediante WiFi, protocolo MQTT a un servidor que bajo la herramienta de programación visual Node-Red administra una base de datos SQL como así también la comunicación hacia las terminales y la interfaz gráfica de usuario. El sistema almacena en la base de datos la ubicación de los instrumentos, su estado, la fecha de ingreso, egreso y el operario que lo transporta.

**Palabras Clave**—IoT, SQL, MQTT, gestión de instrumental, base de datos.

## I. INTRODUCCIÓN

En el marco de las prácticas profesionalizantes que deben realizar los alumnos de la especialidad Técnico en Electrónica perteneciente a la Escuela de Educación Secundaria Técnica N° 6 Albert Thomas, se llevó a cabo un vínculo con el Instituto Argentino de Radioastronomía participando como institución oferente. En este contexto el IAR dispuso como proyecto la realización de un Sistema de Gestión de Instrumental quedando el desarrollo del mismo a cargo de la institución educativa, grupo GRID-SE dando lugar de esta manera a que los alumnos de una escuela técnica intervengan, desarrollen y realicen la puesta en funcionamiento del sistema en los entornos formativos propios de la institución para luego llevar a cabo el montaje y carga de datos en las instalaciones del IAR. Surge de ello una excelente oportunidad de poner en práctica la resolución de una situación problemática basada en tecnología actual y con requerimientos específicos que son necesarios como parte de la formación de calidad de nuestros técnicos electrónicos.

## II. PROPUESTA A RESOLVER

El departamento de electrónica del IAR, cuenta con un variado listado de instrumentos y equipos para uso en laboratorio y de campo, como se indica en la Fig. 1 se encuentra distribuido en diferentes ubicaciones dentro del instituto.

La gestión del instrumental requiere atención dedicada, que actualmente no es posible resolver de manera sencilla con la forma de trabajo y recursos presentes. Esto implica la necesidad de implementar un sistema de gestión que facilite la tarea y permita establecer un mecanismo confiable y sistemático para el registro, seguimiento y control del equipamiento.

En la actualidad el instituto cuenta con una base de datos donde están registrados los instrumentos, utilizando ese insumo como punto de partida la propuesta del IAR es la

realización de un dispositivo automático que sea capaz de registrar el movimiento de los equipos dentro del instituto.



Fig. 1: Instituto Argentino de Radioastronomía.

Los requerimientos establecidos por el IAR son:

- Base de datos para la gestión de instrumental basada en Microsoft Access o alternativa SQL [2].
- Entorno de programación Node-Red [3].
- Sistema basado en un dispositivo SoC ESP 8266 [4].
- Lectora de tarjetas RFID [5].
- Protocolo de comunicación MQTT [6].
- Display indicador de lectura del instrumento o personal.
- Interfaz gráfico destinada a dar alta, baja, modificación y consulta de los equipos, personal y demás datos pertinentes.

## III. MATERIALES Y MÉTODO DE TRABAJO

A continuación, se detalla el procedimiento seguido como así también los elementos utilizados en cada etapa del proyecto.

1. Ensayos con el módulo lector de tarjetas TAG Rc522 de 13,56 MHz. Protocolo de comunicación SPI. Lectura de ID. Ensayo de la pantalla OLED de 0.96 pulgadas con protocolo I2C. El dato a mostrar será el nombre del instrumento. Funciona a modo de confirmación del correcto procedimiento de lectura del instrumento.
2. Ensayo de módulo esp8266 integrado en placa NodeMCU, se estableció comunicación mediante WiFi con una aplicación realizada en Node-Red.
3. Se realizó una red ad hoc entre el servidor y el dispositivo terminal. La misma permitió ensayos de comunicación y pruebas de protocolo sin la necesidad de la red propia de la institución educativa.
4. Finalmente, se eligió reemplazar la base de datos original Microsoft Access por una base de datos SQL SQLITE3, ya que ésta es de código abierto y es soportada por una gran cantidad de lenguajes

de programación. Con esto se creó una base de datos CRUD (Create, Read, Update, Delete) accesible desde Node-Red por una interfaz gráfica.

#### IV. ARQUITECTURA DEL SISTEMA

##### IV-A. El sistema

El sistema representado en la Fig. 2 está formado por una terminal en cada uno de los laboratorios, a saber: Cámara anecoica, Sala limpia, Laboratorio de electrónica, Sala limpia, Laboratorio criogénico. La terminal lee la tarjeta RFID perteneciente al operador y la tarjeta propia del equipo, envía la información de ambos ID con sus tópicos correspondientes al servidor, éste lo recibe y ubica en la base de datos haciendo la modificación de su nueva ubicación. A continuación, devuelve el dato del nombre del instrumento que es leído en la terminal por el operario a modo de acuse de recibo correcto de la información.

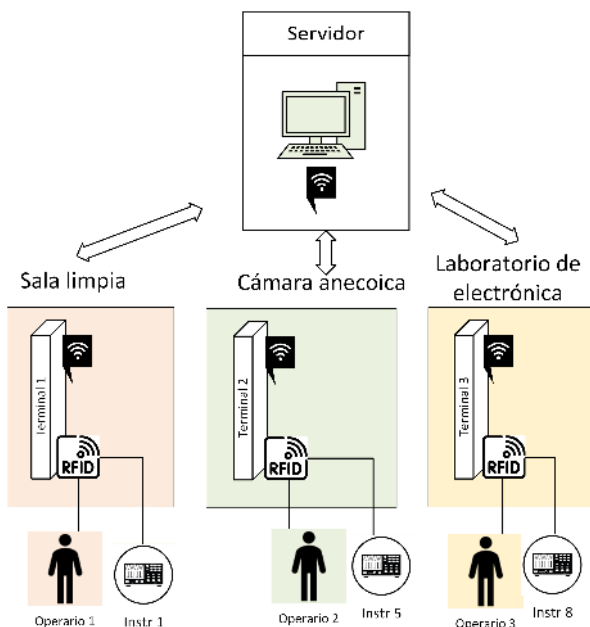


Fig. 2: Diagrama en bloques del SGI.

##### IV-B. El servidor

El servidor, esquematizado en la Fig. 3, se desarrolla en torno al lenguaje de programación Node-Red [3]. El mismo se encarga de gestionar la interfaz gráfica donde el operario en una primera instancia puede registrar los nuevos dispositivos, dar de baja a los mismos, consultar por la ubicación, consultar e ingresar fechas de reparación y estado del mismo. Así también es posible ingresar un nuevo operario y modificar su estado. Lo anterior se realiza en un motor de base de datos SQLITE3, donde se definen tres tablas:

Tabla Laboratorio, llave primaria ID del laboratorio, campo: Nombre del laboratorio.

Tabla Instrumentos, llave primaria ID del instrumento, campos: Nombre del instrumento, Fecha y hora de ingreso, Estado, Fecha de última reparación.

Tabla Personal, clave primaria ID del personal, campos: Nombre del operario, Cargo.

Desde la plataforma de Node-Red también se gestiona la comunicación por protocolo MQTT introduciendo la

IP del Broker mosquito en la configuración pertinente. Por lo tanto, este sistema es capaz de recibir los datos provenientes de cada terminal ubicadas en los distintos laboratorios y devolver un acuse de recibo o “callback” a la terminal correspondiente donde fue ingresado el instrumento.

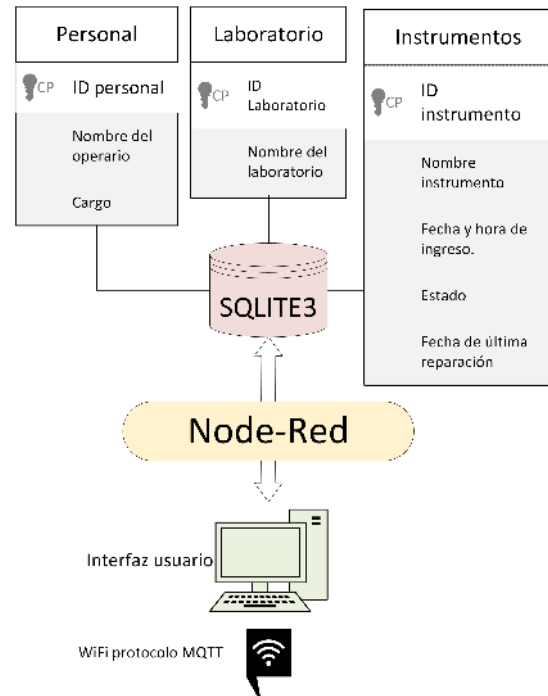


Fig. 3: Diagrama en bloques del servidor.

##### IV-C. La terminal

La terminal, representada en la Fig. 4, está realizada en torno a un dispositivo NodeMCU utilizando el IDE de Arduino para su programación.

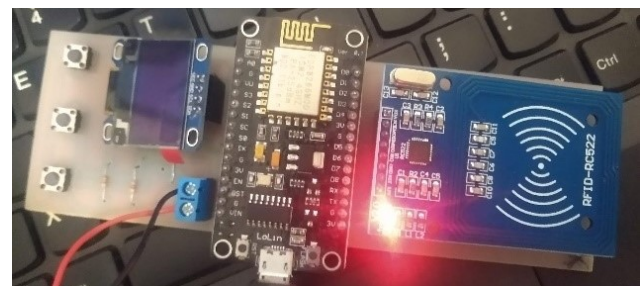


Fig. 4: Ensayo del prototipo de la terminal.

Los periféricos asociados, lectora de tarjetas RFID y display OLED, establecen comunicación mediante los protocolos SPI e I2C respectivamente con sus correspondientes librerías.

El desarrollo del software se realizó mediante las funciones: Wifi, Callback, Reconexión, Setup y Loop.

El tamaño de la placa montada es de 16cm \* 5.5cm resultando de este modo un dispositivo portátil llegando a la conclusión de alimentarlo mediante una batería recargable mediante entrada USB.

#### V. RESULTADOS Y CONCLUSIONES

En un primer ensayo del sistema se probó con dos terminales, registrando en una base de datos el nombre

del instrumento asociado a la tarjeta RFID, el estado del mismo y el laboratorio correspondiente.



Fig. 5: Terminal montada y funcionando junto con el servidor. (Etapa de prototipo)

Los datos se muestran en la interfaz gráfica de usuario, separado por bloques dividido por nombre de laboratorio (en una primera instancia sólo dos laboratorios). Luego el nombre del instrumento es devuelto a la terminal y presentado en el display como se observa en la Fig. 5. Se concluye que el sistema realizado según demanda del IAR posee las siguientes características:

- El sistema es escalable tanto en el número de terminales como en las tablas y los campos que debe almacenar su base de datos.
- Utilización de plataformas de código abierto de hardware y software.
- Las terminales son independientes, no necesitan estar conectadas a una computadora.
- La gestión de usuarios en el servidor se realiza a

través de una interfaz gráfica donde facilita el ingreso y actualización de datos de personal e instrumentos.

- La autonomía de las terminales queda garantizada mediante un sistema de alimentación por batería recargable mediante un conector micro USB.

## VI. TRABAJO A FUTURO

Durante este año se realizará:

El ensayo e implementación del Sistema de gestión de instrumental en las instalaciones del Instituto de radioastronomía.

Inclusión de un nivel de seguridad TLS.

Se implementará un sistema de bajo consumo tipo Light-sleep.

Se reemplazarán los dispositivos NodeMCU por ESP-12.

Se agregará una fuente de alimentación conformada por una batería 18650 de 3000 mAh con un cargador basado en un MCP73871.

Se incluirá un sistema de respaldo de información en cada terminal tal que se ponga en funcionamiento ante cortes de energía.

Se incluirán nuevos requerimientos y necesidades según pedido de los referentes.

Luego de esto, en la etapa de producción, se montará la cantidad de terminales necesarias para cubrir cada laboratorio.

## VII. AGRADECIMIENTO

Mi más sincero agradecimiento a los alumnos que desean transitar sus prácticas profesionalizantes en el Grupo de Investigación y Desarrollo de Sistemas Embebidos. Así también deseo agradecer al personal del Instituto Argentino de Radioastronomía y muy especialmente a los Ingenieros Elías Fliger y Martín Salibe por la confianza y el compromiso con nuestra institución educativa.

## REFERENCIAS

- [1] Aban Sebastian, Acuña Ariel, Acuña Luis, Cajade Juan, Calisaya Aldo, Cavallaro Bruno, De Donato Santiago, Fontana Nicolas, Inga Sebastián, Lopez Eliseo, Ragaini Juan, Repucci Rocco, Rollie Francisco, Salto Tobias, Santillan, Lucas, Tinco Martin, Grupo GRIDSE, 2022. [Online]. Available: <https://eest6albertthomas.com/gridse>
- [2] P. C. Garcia, *SQL fácil*, 1st ed. Gran Via de les Corts Catalanes, 594: Marcombo, 2014, isbn : 978-84-267-2100-6.
- [3] B. R. Japón, *Learn IoT Programming Using Node...* by Japón, Bernardo Ron Paperback / softback, 1st ed. BPB Publications, 2008.
- [4] M. Schawartz, *Internet of Things with ESP8266*, 1st ed. 35 Livery street: Packt, 2016, vol. 1.
- [5] H. A. K. Mohd Helmy, Abd Wahab, *Design and Development of Portable Attendance System using RFID*, 1st ed. BPB Publications, March 2010.
- [6] t. Pulver, *Hands-On Internet of Things with MQTT*, 1st ed. Packt Publishing, 2019, isbn : 9781789345001.

# Control de cultivos hortícolas con sensores y actuadores mediante una plataforma en la nube basada en contenedores

Claudio Omar Biale

Departamento de Informática,  
Facultad de Ciencias Exactas,  
Químicas y Naturales  
Universidad Nacional de Misiones  
Apóstoles, Argentina  
claudiobiale@fceqyn.unam.edu.ar

Ariel Lutenberg

Laboratorio de Sistemas Embebidos,  
Facultad de Ingeniería  
Universidad de Buenos Aires  
Buenos Aires, Argentina  
alutenb@fi.uba.ar

**Resumen**—En este trabajo se describe el diseño, desarrollo e implementación de un sistema de control de cultivos hortícolas bajo cubierta orientado al uso en ambientes urbanos y semiurbanos. El sistema está compuesto por hardware de bajo costo y una plataforma de software de dos niveles, en los planos local y en la nube. En el plano local los nodos sensores y actuadores recopilan datos y realizan acciones de control. La plataforma en la nube basada en contenedores recopila la información enviada por los nodos y se encarga de impartir órdenes de control de acuerdo a reglas definidas y asociadas a cada nodo. Los datos recopilados y el estado de los actuadores pueden ser visualizados en tiempo real por el usuario mediante una página web.

**Palabras Clave**—Internet de las cosas, MQTT, CouchDB, EMQX

## I. INTRODUCCIÓN

En los últimos años la agricultura ha enfrentado muchos desafíos, desde una creciente población mundial a ser alimentada, hasta requisitos de sostenibilidad y restricciones ambientales debido al cambio climático y el calentamiento global [1].

La existencia de diversos problemas sociales y económicos también amenaza la productividad de la horticultura. El envejecimiento de los productores, el poco interés en las generaciones más jóvenes para convertirse en agricultores, la falta de mano de obra y la reducción de la tierra utilizada para la agricultura pueden provocar una disminución de la productividad [1][2].

Esto obliga a implementar soluciones que permitan modernizar las prácticas agrícolas. En este contexto, la Agricultura 4.0 representa la última evolución de la agricultura de precisión. La misma se encuentra basada en el concepto de agricultura inteligente, donde convergen el uso de Internet de las cosas, computación en la nube, aprendizaje automático para el análisis de grandes volúmenes de datos, vehículos no tripulados y robótica [3][4].

A diferencia de la agricultura de precisión donde las tareas de gestión sólo se basan en datos de acuerdo a su ubicación, en la agricultura inteligente también se consideran los datos de acuerdo al contexto, la situación y los disparadores de eventos [5].

El desarrollo de las tecnologías asociadas a la Agricultura 4.0 han permitido que un invernadero se transforme

en un sistema de producción de plantas de ambiente controlado, propicio para ser usado en agricultura urbana, donde el control y monitoreo remoto usando nodos conectados a Internet brindan información precisa para la toma de decisiones. Esto permite reducir las labores humanas y optimizar la productividad y la calidad de los cultivos; protegiéndolos de plagas, enfermedades y de los efectos adversos de los fenómenos meteorológicos extremos [6][7].

El mercado de invernaderos inteligentes se encuentra en constante crecimiento. En el año 2020 se lo valoró en 1380 millones de dólares y se prevé que alcance los 2820 millones de dólares para el año 2028 [8].

En este contexto, el objetivo de este trabajo radica en el diseño, desarrollo e implementación de un prototipo de sistema que automatice el control y monitoreo de cultivos hortícolas bajo cubierta orientado al uso en ambientes urbanos y semiurbanos.

## II. DISEÑO E IMPLEMENTACIÓN

En la Fig. 1 se muestra un diagrama en bloques en el que se pueden observar, de forma simplificada, las interacciones que se producen entre las distintas partes del sistema.

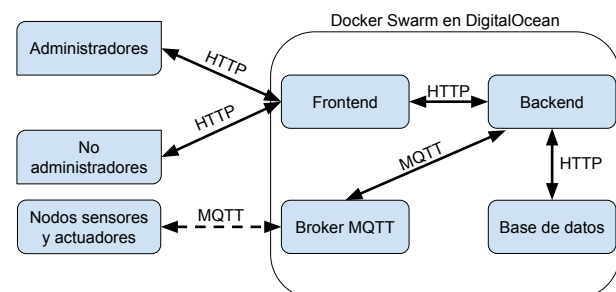


Fig. 1: Diagrama en bloques del sistema.

El sistema se desplegó en el proveedor de servicios en la nube DigitalOcean donde se conformó un clúster de Docker Swarm entre las distintas máquinas virtuales o droplets y está constituido por cuatro aplicaciones:

- Una aplicación web (frontend) desarrollada utilizando Svelte,

- Un servidor REST (backend) desarrollado utilizando Fastify,
- Una base de datos documental CouchDB y
- Un broker MQTT EMQX.

A nivel de protocolos, en la capa de aplicación se usan HTTP y MQTT. El primero se utiliza para la interacción de usuarios de tipo administrador y no administrador con el frontend, la comunicación entre el frontend y el backend, y entre el backend y la base de datos CouchDB. Por otro lado, MQTT es el protocolo utilizado para la comunicación entre los nodos sensores y actuadores con el broker MQTT, y entre este y el backend.

Con el objetivo de que las imágenes del frontend y backend se encuentren disponibles para los nodos del clúster se configuró un repositorio privado en Docker Hub.

Para la gestión de los contenedores dentro del clúster se utilizó un conjunto de scripts de Bash y Portainer Community Edition, una herramienta de gestión de contenedores de código abierto.

Las funcionalidades existentes en la aplicación del frontend permiten gestionar nodos, tipos de sensores, tipos de actuadores, reglas y usuarios. La gestión de nodos incluye el cambio en el tiempo entre envíos de datos, el acceso al dashboard, a la asociación de reglas, al historial de mediciones y al historial de logs.

En el backend se implementaron rutas y plugins que aceptan solicitudes del frontend y del broker MQTT. Se definió la forma en que el backend y los nodos sensores y actuadores se comunican utilizando el protocolo MQTT.

En la etapa inicial se desarrolló una aplicación en Node.js que simula el funcionamiento de un nodo sensor y actuador. La aplicación publica cada determinado tiempo un mensaje MQTT con valores aleatorios correspondientes a sensores de humedad ambiente, humedad de suelo y temperatura ambiente, y tiene como actuador una bomba de agua.

Se ha desarrollado un prototipo de nodo sensor y actuador sobre un módulo ESP32-WROOM-32, constituido por los siguientes componentes:

- Sensor de temperatura y humedad ambiente DHT22.
- Sensor de humedad del suelo HL-69.
- Válvula solenoide.
- Relés.
- Transformador de 12 V.

Para el desarrollo del firmware se utilizó Mongoose OS un framework y sistema operativo de tiempo real de código abierto que facilita el desarrollo de firmware aplicado a Internet de las cosas.

En [9] se encuentra disponible la memoria del trabajo final de posgrado donde se especifican cuestiones como el modelo de datos y el detalle de las acciones asociadas a nodos y backend en la comunicación usando el protocolo MQTT.

### III. EVALUACIÓN

Para el despliegue del sistema en DigitalOcean se utilizaron dos maquinas virtuales o droplets provistas de 1 CPU compartida, 1 GB de memoria RAM, 25 GB de disco SSD y 1000 GB de transferencia.

Con el sistema desplegado se realizaron pruebas manuales de escalamiento horizontal del broker EMQX y

la base de datos CouchDB. Ambas pruebas consistieron en agregar un nuevo contenedor, realizar las acciones correspondientes para el armado del clúster y verificar que el clúster se haya conformado.

Adicionalmente se realizó una prueba de integración y funcionamiento de todas las partes. Los pasos de configuración de la prueba consistieron en:

- Acceder a la aplicación web con un usuario de tipo “administrador”.
- Crear un nodo.
- Crear los tipos de sensores correspondientes a temperatura ambiente, humedad ambiente y humedad del suelo.
- Crear un tipo de actuador correspondiente a una bomba de agua con funcionamiento en modo “on-off”.
- Definir dos reglas que permitan:
  - Activar la bomba de agua si la humedad del suelo es menor o igual al 30 %.
  - Desactivar la bomba de agua si la humedad del suelo es mayor o igual al 50 %.
- Asociar las reglas al nodo.
- Programar el prototipo de nodo sensor y actuador.

Una vez configurado el entorno de prueba se procedió a verificar que el sistema responda a cada una de las reglas asociadas al nodo. Para que se cumpla la primera regla de prueba se dejó el sensor de humedad del suelo sin contacto con agua y se verificó el estado del relé y los cambios a nivel del dashboard del nodo. De igual modo, para que se cumpla la segunda regla de prueba se colocó el sensor de humedad del suelo en agua y se verificó el estado del relé y los cambios a nivel del dashboard del nodo. Como paso final se procedió a verificar que los datos se almacenaron correctamente en la base de datos.

### IV. COMPARACIONES CON OTROS TRABAJOS

Pliot [10] permite la obtención de datos mediante una red de sensores instalados en el invernadero que transmiten los datos de forma inalámbrica para su almacenamiento en la nube. Mediante una aplicación web o un celular se puede acceder a visualizar los datos actuales e históricos y a la administración de alertas. Soporta el cálculo de índices y algoritmos de predicción de acuerdo a los datos recolectados. La empresa ofrece la instalación, mantenimiento y calibración de equipos que son otorgados en comodato.

Seedmech [11] posibilita la obtención de datos de una huerta o invernadero mediante estaciones meteorológicas, estos datos se almacenan en la nube. Mediante una aplicación web o un celular se puede acceder a visualizar los datos actuales, históricos y distintos valores calculados. La disponibilidad de sensores y del sistema de riego viene dado por el modelo de estación meteorológica pero es posible agregar sensores adicionales. Hay modelos que disponen de un sistema de alertas mediante SMS. Las estaciones meteorológicas y los sensores son provistos por la empresa.

En [12] se describe el desarrollo de una plataforma compuesta por un servidor web NGINX, una aplicación desarrollada en Node-Red, un broker MQTT EMQX y una base de datos MySQL. Se utilizó como visor de datos a

Grafana y se desarrolló una aplicación en Python que se encarga de almacenar los datos que llegan al broker en la base de datos. Los nodos fueron implementados sobre un módulo ESP32 constituido por una electroválvula, un led RGB, un módulo GPS y sensores de temperatura y humedad ambiente y de humedad del suelo.

En [13] se desarrolló un sistema compuesto por nodos y un gateway conectados mediante el protocolo LoRa. El gateway se encarga de recibir los datos enviados por cada nodo y de enviarlos a una base de datos Firebase donde también se configuró un dashboard para una visualización amigable de la información. Los nodos se implementaron en módulos Arduino Uno y el gateway en una placa HELTEC LoRa 32.

En el trabajo desarrollado los datos se almacenan en la nube en servidores que son administrados por la empresa y no en servidores manejados por terceros como es el caso de Pliot y Seedmech. El soporte a distintos tipos de actuadores en Seedmech es limitado porque solo se proporciona de forma adicional el uso de actuadores para riego. Los tipos de sensores soportados por Pliot y Seedmech viene limitado porque se pueden usar aquellos provistos por las respectivas empresas. En [12] y [13] no se especifica la forma de adicionar nuevos sensores o actuadores. En el trabajo desarrollado se soportan sensores y actuadores a decisión del usuario final siempre que se cumpla con el protocolo de comunicación y que los tipos de sensores y actuadores se encuentren registrados en el sistema. Pliot permite administrar un sistema de alertas y Seedmech tienen soporte en algunos modelos de estaciones meteorológicas, este soporte no se encuentra en la implementación del trabajo desarrollado. Pliot, Seedmech y del libro de referencia.[13] soportan el uso de protocolos de comunicación adicionales pero estos protocolos no son necesarios para ambientes urbanos o semiurbanos. En [13] el control de cambio en los actuadores se encuentra especificado en el código de los nodos. En [12] el cambio en los actuadores se puede realizar de forma manual desde el dashboard o en caso de pérdida de conexión de acuerdo a lo especificado en el código de los nodos. En el trabajo desarrollado el cambio en los actuadores se produce de acuerdo a las reglas asociadas a cada nodo.

## V. CONCLUSIONES

En este trabajo se completó el diseño, desarrollo e implementación de un prototipo de sistema escalable que automatiza el control y monitoreo de cultivos hortícolas bajo cubierta orientado al uso en ambientes urbanos y semiurbanos.

El sistema se desplegó en un proveedor de servicios en la nube. El uso de un clúster de Docker Swarm permitió que el sistema sea escalable horizontalmente y use los servicios de descubrimiento y balanceo de carga. Se comprobó el funcionamiento del escalado horizontal y la conformación de un clúster tanto en la base de datos como también en el broker MQTT.

A futuro se pretende mejorar el prototipo de nodo sensor y actuador, extender el sistema de reglas, automatizar la configuración del clúster al escalar horizontalmente los servicios, analizar alternativas en la forma de almacenamiento de los valores de sensores y actuadores en la base

de datos, trasladar el control de reglas de activación de actuadores al borde de la red y desarrollar un módulo que almacene los datos al borde de la red en situaciones donde no exista conexión de Internet, este módulo puede extenderse para definir en que caso se transfieren datos a la nube.

## REFERENCIAS

- [1] Rafael Simionato et al. «Survey on connectivity and cloud computing technologies: State-of-the-art applied to Agriculture 4.0». En: *Revista Ciência Agronômica* 51 (2021).
- [2] Lukas Zagata y Lee-Ann Sutherland. «Deconstructing the ‘young farmer problem in Europe’: Towards a research agenda». En: *Journal of Rural Studies* 38 (2015), págs. 39-51.
- [3] Sara Oleiro Araújo et al. «Characterising the Agriculture 4.0 Landscape—Emerging Trends, Challenges and Opportunities». En: *Agronomy* 11.4 (2021), pág. 667.
- [4] Verónica Saiz-Rubio y Francisco Rovira-Más. «From smart farming towards agriculture 5.0: A review on crop data management». En: *Agronomy* 10.2 (2020), pág. 207.
- [5] Cor Verdouw et al. «Digital twins in smart farming». En: *Agricultural Systems* 189 (2021), pág. 103046.
- [6] Antonis Tzounis et al. «Internet of Things in agriculture, recent advances and future challenges». En: *Biosystems engineering* 164 (2017), págs. 31-48.
- [7] Redmond R Shamshiri et al. «Advances in greenhouse automation and controlled environment agriculture: A transition to plant factories and urban agriculture». En: *International Journal of Agricultural and Biological Engineering* 11.1 (2018), págs. 1-22.
- [8] Verified Market Research. *Smart Greenhouse Market Size, Share, Trends, Opportunities & Forecast*. <https://www.verifiedmarketresearch.com/product/global-smart-greenhouse-market-size-and-forecast-to-2025/>. Abr. de 2022. (Visitado 06-05-2022).
- [9] Laboratorio de Sistemas Embebidos Universidad de Buenos Aires. *Consulta de Trabajos Finales - Listado Completo*. <https://se.posgrados.fi.uba.ar/trabajo-final/archivo-historico/consulta-de-trabajos-finales-listado-completo>. (Visitado 06-06-2023).
- [10] Pliot Smart Agriculture. *Pliot*. <https://pliot.ar/>. (Visitado 24-11-2022).
- [11] Seedmech Latinamerica SRL. *Home — SEED-MECH*. <https://www.seedmech.com/>. (Visitado 24-11-2022).
- [12] Facundo A. Chazarreta et al. «Diseño de una plataforma para servicios IoT con aplicaciones en el sector productivo». En: *Libro de Trabajos del CASE 2022, Congreso Argentino de Sistemas Embebidos, Argentina*. 2022.
- [13] Matías N. Mardegan et al. «Sistema embebido escalable para el monitoreo de parámetros en invernaderos mediante IoT». En: *Libro de Trabajos del CASE 2022, Congreso Argentino de Sistemas Embebidos, Argentina*. 2022.

Categoría

**REPORTE**

Protocolos y Comunicaciones

Área Temática

# Sistema visualizador de señales de radar empleando hardware reconfigurable

Lisvan Guevara Trujillo  
Facultad de Informática  
Universidad Tecnológica de La Habana  
La Habana, Cuba  
<https://orcid.org/0000-0002-1830-2045>

Marcelino Sánchez Posada  
Facultad de Informática  
Universidad Tecnológica de La Habana  
La Habana, Cuba  
<https://orcid.org/0000-0001-8906-1149>

Ariel Hernández Reyes  
Facultad de Informática  
Universidad Tecnológica de La Habana  
La Habana, Cuba  
<https://orcid.org/0009-0009-1133-3779>

**Resumen**—Conceptualmente un sistema de radar consta de cinco componentes: un generador, un receptor, un amplificador, un procesador y un visualizador. En la actualidad existen radares de aproximación de precisión basados en tecnología analógica, por lo que presentan inestabilidad durante su funcionamiento. El objetivo de este trabajo es implementar un sistema de procesamiento y visualización digital que cumpla con las exigencias operativas del sensor. En este documento se presenta una implementación combinada hardware-software empleándose un kit FPGA y un ordenador de escritorio disponibles comercialmente. El kit FPGA de conjunto con una tarjeta ADA posibilita la recepción y conversión de la señal analógica a digital, así como implementar algoritmos de procesamiento de los datos del radar. La representación de la información se realizó en el computador de escritorio. La comunicación entre estos dispositivos se realizó por la interfaz de alta velocidad PCI Express. Se logra una sustitución tecnológica de los sistemas de procesamiento y visualización analógicos, cumpliendo con el tiempo de actualización de la información impuesta por el radar de aproximación de precisión. Se obtuvo una solución modular, genérica y flexible, con posibilidad de adicionarle nuevas funcionalidades.

**Palabras claves**—procesador de señales de radar, visualizador de radar, radar de aproximación de precisión.

## I. INTRODUCCIÓN

Conceptualmente, un sistema de radar consta de cinco componentes: un generador, un receptor, un amplificador, un procesador y un visualizador [1, 2]. El procesador permite obtener de las señales recibidas la información deseada. El visualizador es un dispositivo electrónico que se utiliza para representar en un formato adecuado la información obtenida del procesador [3]. Los visualizadores de radar deben presentar al observador una imagen gráfica continua, que permita comprender de forma fácil la posición relativa de los objetos detectados, lo que contribuye a la eficiencia del proceso de toma de decisión por los operadores [4, 5].

Actualmente, Cuba cuenta con radares de aproximación de precisión que datan de la década del 50 construidos con tecnología analógica [6], los cuales aún se emplean en el aterrizaje de precisión de aeronaves. Estos dispositivos están sometidos a largos períodos de mantenimiento provocado fundamentalmente por el envejecimiento de sus partes y piezas. Además son altos consumidores de energía eléctrica, poseen gran volumen y peso, sus piezas están obsoletas y existe incapacidad de adquirir los repuestos en el mercado internacional [6]. La inestabilidad en el funcionamiento y los altos costos que provocaría la modernización del sistema de

procesamiento y visualización analógico en países extranjeros, determinan la necesidad de obtener un sistema procesador y visualizador que cumpla con las exigencias operativas del radar de aproximación de precisión.

En la actualidad, gracias a los avances en la tecnología digital, es posible procesar y presentar la información de señales de radar con el empleo de medios de cómputo [7, 8], siendo esta una de las vías para la solución de los problemas planteados. Para la implementación de un sistema de procesamiento digital de señales de radar se suelen emplear dispositivos electrónicos programables como microcontroladores, DSPs (*digital signal processing*), FPGAs (*field-programmable gate array*) y CPLDs (*Complex Programmable Logic Device*) [9]. Varios autores han abordado la implementación de sistemas de procesamiento y visualizadores digitales cumpliendo con las exigencias de tiempo y procesamiento que impone el radar, empleándose diversas plataformas hardware y software [7, 8].

Los autores de [1, 2] para realizar el procesamiento de la señal del radar se auxiliaron de dispositivos FPGA. Como resultado se obtuvo una modernización de los componentes procesador y visualizador del radar, capaz de realizar el proceso de adquisición, interpretación y presentación en tiempo real.

En [10] se presenta un sistema de adquisición, procesamiento y visualización de datos de radar. Los procedimientos de procesamiento y visualización de los datos de radar se implementaron como módulos software que se ejecutan en una PC (*personal computer*) disponible comercialmente. En este trabajo se plantea que el procesamiento y la visualización de datos de radar son operaciones computacionalmente muy exigentes, por lo que para lograr un rendimiento en tiempo real fueron aprovechadas las capacidades de procesamiento del CPU (*central processing unit*).

Por su parte en [7, 8] se desarrolló un visualizador digital de la situación aérea para remplazar uno antiguo basado en tubo de rayos catódicos. Por razones de rendimiento la aplicación fue escrita en C++ y la representación de la situación aérea se realizó en un hilo independiente.

A partir del análisis de estos trabajos se pudo apreciar el uso de los kit FPGA, ya que en el desarrollo de las nuevas generaciones de radar, las plataformas de procesamiento de señales y datos son ampliamente utilizadas [9, 11].



## II. DESARROLLO

### A. Diseño e implementación

Para la solución de los problemas presentes en el radar analógico, el sistema debe cumplir con varios requisitos. Uno de los requerimientos más importantes del sistema es realizar la adquisición, procesamiento y presentación de la información en tiempo real. Además, debe ser un sistema flexible ante futuras modificaciones que permita poder ser actualizado. Una variante factible en la implementación de estos sistemas lo constituye la combinación de un FPGA con un procesador estándar en un ordenador personal, sobre todo por su accesibilidad en el mercado y su potencia de cálculo. La combinación de una FPGA con un procesador generalmente se dificulta por la necesidad de establecimiento de un enlace de datos de alta velocidad entre el kit FPGA y el ordenador. En la revisión bibliográfica se pudo evidenciar una de las vías de comunicación a alta velocidad entre un kit FPGA y un computador es a través de la interfaz PCI (*peripheral component interconnect*) Express [12-14]. La Figura 1 muestra el enlace de comunicación entre el kit de FPGA y el computador de escritorio a través de la interfaz PCI Express.

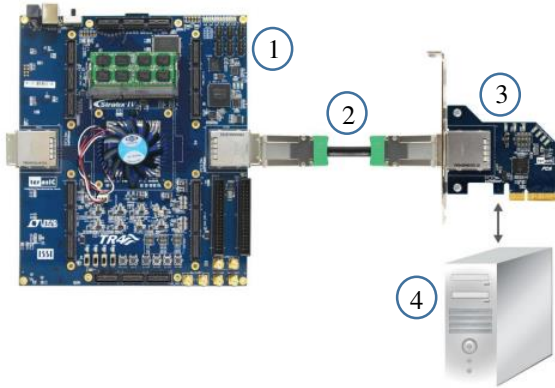


Fig. 1. Conexión del kit FPGA al ordenador de escritorio.

La figura muestra los cuatro elementos fundamentales que intervienen en la comunicación. El primero es el kit TR4 de Terasic, el cual cuenta con un FPGA Stratix IV GX EP4SGX230. El segundo elemento es el cable de conexión PCIe x4, el tercero es la tarjeta PCA3 que se conecta por interfaz PCIe al cuarto elemento que es una PC. La transmisión de datos a alta velocidad mantiene un rendimiento y latencia aceptables para operar en tiempo real.

Como parte de la modernización se empleó un kit FPGA que dispone de una gran cantidad de recursos (228000 elementos lógicos), memoria interna e interfaz de comunicación con el ordenador, lo cual se fundamenta por la incorporación en el mismo dispositivo de múltiples algoritmos de procesamiento, tanto de señales, entre los que se encuentra el filtrado del ruido, tanto frecuencial como espacial, la detección óptima, etc., como de datos para la detección y seguimiento de objetos aéreos. El procesamiento que realiza este dispositivo para garantizar la visualización de la información de las señales de radar permite llevar un cúmulo relativamente grande de datos a la cantidad de píxeles a representar en la pantalla. Véase en la Figura 2 la etapa de procesamiento de señales que se realiza para garantizar la visualización de las señales de radar.

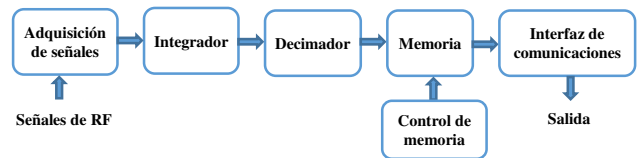


Fig. 2. Diagrama en bloque del procesamiento de señales.

La Figura 2 muestra como el dispositivo FPGA recibe las señales de radiofrecuencia que provienen del front-end original del radar el cual es analógico, conectándose a una etapa en frecuencia intermedia. Se emplea el convertor AD9254, el cual forma parte de una tarjeta ADA que se acopla al kit TR4 mediante interfaz HSMC (*High Speed Mezzanine Card*) que muestrea la señal de entrada para realizar el procesamiento digital de las señales. Teniendo en cuenta que la distancia máxima que se desea representar en el visualizador es de 37500 metros y el tiempo que demora el pulso en recorrer esa distancia es de 250 microsegundos, al ser muestreada a 25 MHz se obtuvieron 6250 muestras por barrido. Luego, estas muestras se entregaron al integrador que disminuye la influencia de picos ruidosos. Para esto se realizaron promediaciones cada 5 muestras. Posteriormente, se decima la señal, al tomarse el mayor valor del resultado de 2 promediaciones, que corresponden a 2 grupos de 5 muestras, es decir 10 muestras. El integrador disminuye la amplitud de los picos de ruido y el decimador evita una reducción mayor del valor de amplitud del blanco. Como resultado del empleo del integrador y el decimador, al tomar 1 valor cada 10 muestras de los 6250 valores, se obtienen 625 valores por barrido. Los 80000 datos obtenidos como resultado de los 128 barridos que realiza el radar en una exploración del espacio aéreo, se almacenan en una memoria del kit FPGA, que es compartida entre el kit FPGA y la PC. En el FPGA se creó un bloque encargado de escribir los valores en memoria, donde cada espacio tiene un tamaño de 256 bits. Teniendo en cuenta que se tienen que almacenar en memoria 80000 datos, cada uno de 8 bits se necesitarían 2500 espacios de memoria para almacenar esta información. La escritura de los datos en memoria se realiza a través de un control de memoria que se encarga de organizar la estructura de los datos a escribir. Para garantizar que no se realice la lectura desde la PC en el mismo instante de tiempo que el FPGA esté escribiendo, existe un espacio de memoria de la FPGA que es leído por la PC para saber si se realizó la escritura de todos los datos y si se puede leer la memoria. Para la lectura de los datos desde la PC se hizo uso de una biblioteca de enlace dinámico (DLL, por sus siglas en inglés) realizada por TERIC, la cual sólo está disponible para los sistemas operativos Windows XP/7/10. La transferencia de datos a alta velocidad es realizada por DMA (*Direct Memory Access*).

Una vez leídos los datos se realiza la representación de la información, para ello se empleó la indicación de tipo E. La exploración de la situación aérea se representará a través de imágenes. Los píxeles que componen cada imagen representan la intensidad de las muestras de la señal. El ordenador de escritorio también fue empleado para el registro de la información en una base de datos, lo cual permite su análisis posterior. Además el ordenador se encarga de distribuir la información recibida desde el kit TR4 a otros dispositivos por interfaz Ethernet posibilitando representar la información en diferentes lugares y en varios formatos de indicaciones.

### III. RESULTADOS Y DISCUSIÓN

Una vez implementada la arquitectura hardware-software se hace necesario verificar si cumple con la restricción temporal de 40 milisegundos que impone el sensor. Con el objetivo de determinar la configuración óptima del sistema visualizador y determinar los factores que influyen de manera significativa en el tiempo de ejecución, se empleó el diseño de experimentos. El proceso experimental a analizar es la ejecución del software encargado de la lectura de los datos ubicados en las direcciones de memoria del FPGA a través de PCI Express y la representación de los datos del radar. Se define como rendimiento del proceso, el tiempo en milisegundos que emplea el sistema en realizar la lectura y representación de los datos correspondientes a una exploración del radar. La unidad experimental sobre la cual se ejecutarán todos los tratamientos es un ordenador que posee un procesador Intel Core i3-2120 a 3.30 GHz, 4 GB de RAM (*Random Access Memory*) y el sistema operativo Windows 7.

Los factores controlables identificados son afinidad del procesador y prioridad del proceso. El factor no controlable identificado son los procesos ejecutados por el sistema operativo, ya que en el momento en que se ejecuta el software hay procesos ejecutándose en el sistema operativo, que consumen recursos arbitrariamente y no pueden ser detenidos. Para cada uno de los factores seleccionados en el experimento existen dos niveles, uno de ellos se considera como el valor bajo y otro como el alto. En la Tabla 1 se muestran los factores con los niveles asociados a cada uno de ellos:

TABLA I. Niveles de los factores seleccionados.

Factor	Nivel bajo	Nivel alto
Afinidad del procesador	2	4
Prioridad del proceso	Por debajo de lo normal (DN)	Por encima de lo normal (EN)

Para realizar el diseño de experimentos se decidió hacer un diseño factorial completo. Al tener el factor controlable afinidad del procesador 2 niveles y prioridad del proceso 2 niveles, se realizaron 4 combinaciones y cada combinación será replicada 5 veces para disminuir el error experimental, por lo que en total se realizaron 20 corridas. Luego de obtener los valores de rendimiento para los tratamientos generados, se procedió a generar un conjunto de gráficas. La Figura 3 muestra la gráfica de interacción, la cual representa la respuesta media de dos factores para todas las posibles combinaciones de sus niveles, lo que permite determinar la configuración de factores que optimiza los resultados.

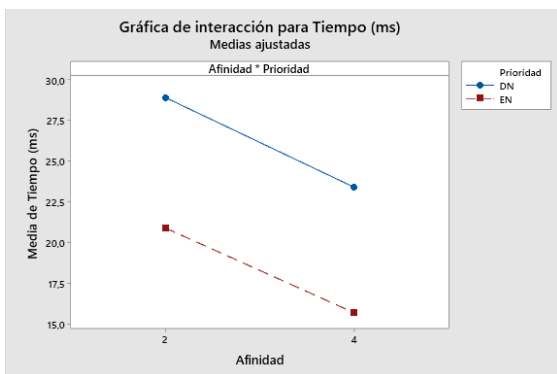


Fig.3. Gráfica de efectos principales.

La gráfica muestra que ambos factores influyen en el tiempo de ejecución del software. Además se aprecia que con la ejecución del software con prioridad por debajo de lo normal se aumenta el tiempo de ejecución con respecto a cuándo se ejecuta el software con prioridad por encima de lo normal, independientemente de la afinidad del procesador. Al ejecutarse el software con prioridad por encima de lo normal y con una afinidad de 4 procesadores se logró el menor tiempo, con una media de 15.7 ms, siendo esta última la configuración óptima de los factores.

### IV. CONCLUSIONES

La arquitectura hardware-software propuesta logra una sustitución tecnológica del sistema de procesamiento y visualización, cumpliendo con el período de actualización del radar de aproximación de precisión. Se propone una solución de hardware modular, genérico y flexible, sin la necesidad de diseño y fabricación de placas de circuito impreso personalizadas, permitiendo realizar el procesamiento digital de las señales de radar y una fácil modificación ante futuros cambios.

### REFERENCIAS

- [1] N. Acosta, M. A. Tosini, and M. F. Mezzanotte, "Desarrollo de un Visualizador de Señales de Radar," presented at the XII Congreso Argentino de Ciencias de la Computación, Argentina, 2006.
- [2] N. Acosta, M. A. Tosini, M. F. Mezzanotte, and M. Tommasi, "Improving radar visualization system," *Journal of Computer Science & Technology*, vol. 7, pp. 1-7, April 2007.
- [3] V. Kavyashree, P. N. Madhu Chaitra, H. Vinutha, and S. Rajeshwari, "A Radar Target Generator for Airborne Targets," *International Journal of Advance Research and Innovative ideas in Education*, vol. 2, pp. 249-261, 2017.
- [4] P. Kaushik, "Radar Displays," *International Journal of Innovative Research in Technology*, vol. 1, pp. 472-476, 2014.
- [5] Sulistyaningsih, Y. P. Saputera, M. Wahab, and Y. Y. Maulana, "Design of radar display of Indonesian airspace monitoring application," *TELKOMNIKA*, vol. 17, pp. 1176-1184, 2019.
- [6] L. G. Trujillo, B. N. S. Hernández, L. Z. Méndez, W. H. Mazo, and M. A. Ampuero, "Sistema visualizador para un radar de seguimiento con reducción de costo, consumo eléctrico y tamaño," *Revista Cubana de Ciencias Informáticas*, vol. 15, pp. 76-104, 2021.
- [7] M. Stamatovic, M. Jevtić, U. Kisić, and M. Tatarević, "Design and implementation of a modern radar display for air surveillance applications," presented at the 20th Telecommunications Forum (TELFOR), 2012.
- [8] M. Stamatović, M. Jevtić, U. Kisić, M. Tatarević, T. Pajić, and K. Marković, "Modern Air Situation Picture Display for Air Surveillance Radar Applications," *Telfor Journal*, vol. 5, pp. 54-59, 2013.
- [9] F. A. S. Gómez, "Metodología de codiseño hardware-software para procesamiento de señales radar en sistemas embebidos," Magíster en Ingeniería Electrónica Tesis de Maestría, Facultad de Ingeniería, Pontificia Universidad Javeriana, Colombia, 2017.
- [10] M. Jevtić and M. Stamatović, "Radar data processing and visualization on desktop platforms," presented at the 17th Telecommunications forum (TELFOR), Belgrade, Serbia, 2009.
- [11] S. M. López Estrada, "Detección Automática en Señales de Radar Basada en Conocimiento," Tesis Doctoral, Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, Mexico, 2009.
- [12] L. Boyang, "Research and Implementation of XDMA High Speed Data Transmission IP Core Based on PCI Express and FPGA," presented at the 1st International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Kunming, China, 2019.
- [13] M. A. Khelif, J. Lorandel, O. Romain, M. Regnery, D. Baheux, and G. Barbu, "Toward a hardware man-in-the-middle attack on PCIe bus," *Microprocessors and Microsystems*, vol. 77, p. 103198, 2020.
- [14] Y. I. Guerra, A. R. Zaldívar, A. H. Violat, and N. R. Ramírez, "Implementación Hardware-Software de un Sistema de Estimación de DOA con Súper Resolución," *Revista de ingeniería electrónica, automática y comunicaciones*, vol. 43, pp. 16-32, 2022.

Categoría

# REPORTE

Software Embebido

Área Temática

# Cuellos de botella en sistemas embebidos en la adquisición y transmisión de señales analógicas

Luisina Gómez, Rodrigo González, María José Millán, y Julián Oreggioni,  
Universidad de la República, Montevideo, Uruguay

Email: {luisina.gomez, rodrigo.gonzalez.suarez, maria.millan, juliano}@fing.edu.uy

**Resumen**—Los microcontroladores estándar de rango medio especifican la disponibilidad de varios canales analógicos y altas frecuencias de muestreo de sus conversores analógico-digitales. Este trabajo explora de forma preliminar qué tan factible es aprovechar esas altas tasas de muestreo, identificando cuellos de botella, y eventualmente proponiendo soluciones. Para ello, se desarrolló el prototipo de un sistema embebido, basado en el microcontrolador MSP430G2553, que adquiere señales analógicas, las procesa y las transmite a una computadora personal mediante comunicación serial, buscando operar a la máxima capacidad posible.

## I. INTRODUCCIÓN

Los microcontroladores estándar de rango medio, como el ATmega328PB de Microchip o el MSP430 de Texas Instruments, especifican la disponibilidad de varios canales analógicos (hasta ocho en los mencionados microcontroladores) y altas frecuencias de muestreo de sus conversores analógico-digitales (ADC), por ejemplo 80 kpsps (kilo muestras por segundo) en el caso de ATmega328PB, o 200 kpsps en el MSP430. Este trabajo explora qué tan factible es aprovechar esas altas tasas de muestreo, identificando cuellos de botella, y eventualmente proponiendo soluciones. Para ello, se desarrolló el prototipo de un sistema embebido, basado en el microcontrolador MSP430G2553, que adquiere señales analógicas, las procesa y las transmite a una computadora personal (PC) mediante comunicación serial, buscando operar a la máxima capacidad posible. Asimismo, se implementó en Python en el PC una interfaz gráfica de usuario (GUI) que permite recibir los datos en el PC, visualizarlos y configurar ciertas características de la adquisición, tales como el número de canales a adquirir y la frecuencia de muestreo de la señal.

## II. DESCRIPCIÓN DEL SISTEMA

En la Fig. 1 se muestra el diagrama funcional y de conexión del sistema, donde las señales adquiridas provienen del generador de señales Analog Discovery 2 de Digilent.

Se utiliza la placa de desarrollo MSP-EXP430G2ET (Launchpad) del microcontrolador MSP430G2553. Se trata de un microcontrolador de 16 bits, que cuenta con un ADC de 10 bits y tasa de muestreo de hasta 200 kpsps, multiplexado en su entrada para admitir hasta 8 canales. Posee además, memoria flash de 16 kB y RAM de 512 bytes, varios modos de operación de bajo consumo, e interfaz de comunicación universal (USCI) de 2 Mbps de tasa máxima de comunicación por UART (Universal Asynchronous Receiver-Transmitter). Si

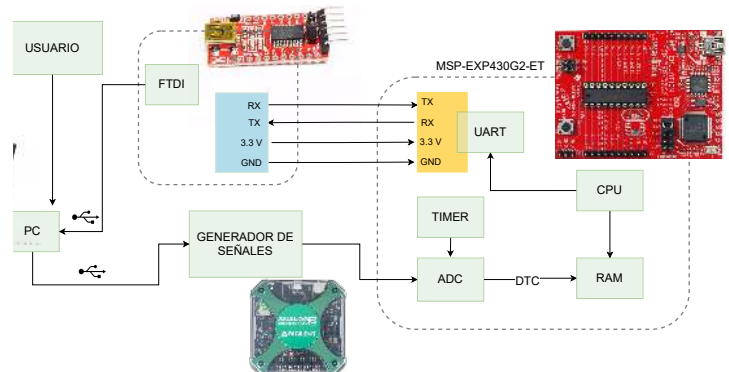


Figura 1. Diagrama funcional y de conexión.

bien el microcontrolador tiene ocho canales analógicos, en el sistema implementado la cantidad de canales disponibles se reduce a seis, puesto que la UART comparte dos pines con el ADC. El ADC incluye además un “Data Transfer Controller” (DTC), el cual permite la transferencia de los datos digitalizados desde el ADC a un lugar de RAM predefinido, sin la intervención del microprocesador. Dichas transferencias son realizadas en bloques de largo configurable. Esto permite que el microprocesador realice otras tareas en paralelo mientras se realiza la transferencia aumentando la capacidad de adquisición de datos.

Además de proveer un mecanismo para depurar y programar, el Launchpad ofrece un sistema para medir el consumo de energía (Energy Trace) y un adaptador UART/USB cuya velocidad máxima es 115 kbps (ver Tabla 4 en pág. 19 de [1]). Este adaptador constituye por tanto el primer cuello de botella que se solucionó fácilmente incorporando un adaptador UART/USB externo que permite transmitir a una frecuencia máxima de 3 Mbps.

El software se implementó en lenguaje C utilizando el Code Composer Studio (CCS) de Texas Instruments como IDE. Se utiliza *Round-Robin con interrupciones* como arquitectura de software embebido [2]. En la Fig. 2 se presenta un diagrama de tres capas con los distintos módulos de software implementados.

**main.c** implementa el bucle principal del Round-Robin con interrupciones (ver Fig. 3). *flag\_adc* se enciende en la ISR del ADC cuando se finaliza la adquisición de un nuevo bloque de datos. Se apaga en el main, luego de procesar y transmitir el último dato del bloque. *flag\_cmd* se enciende en la ISR de recepción de la UART cuando se recibió del PC un

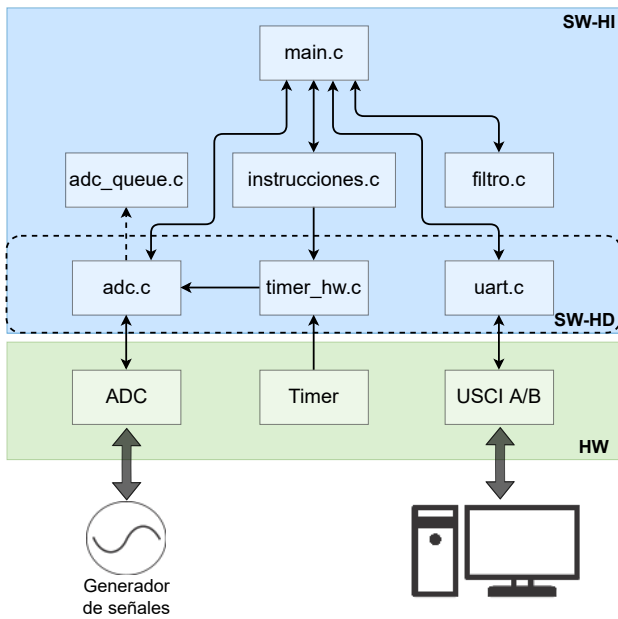


Figura 2. Diagrama de módulos. Las flechas continuas indican la comunicación entre módulos. La flecha punteada señala la transferencia de datos adquiridos por el ADC hacia una sección de memoria específica (DTC).

comando completo. Se apaga en la *main* luego de decodificar el comando recibido por UART.

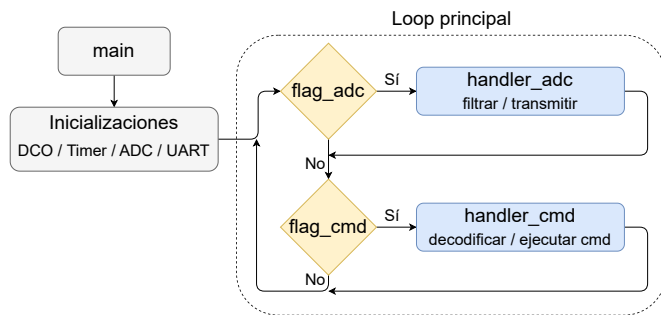


Figura 3. Diagrama de flujo del módulo principal (*main.c*).

En *adc\_queue.c* se definen las funciones de acceso a la cola de datos adquiridos mediante el ADC.

El módulo *instrucciones.c* cuenta con la función *get\_cmd* que permite decodificar el mensaje obtenido por UART, interpretando el comando y tomando la acción correspondiente según el caso. Los comandos disponibles son: inicio y fin de adquisición, configuración de frecuencia de muestreo ( $f_s$ ) y baudrate (ver Tabla I) y selección de cantidad de canales de adquisición (en la actual versión se permite adquirir uno o dos).

En el módulo *adc.c*, entre otras cosas, se configura para que el ADC use su oscilador interno (ADC10OSC), que tiene una frecuencia de 5 MHz, para que trabaje en modo *un canal - una conversión*, y para que los resultados de la conversión se guarden directamente en memoria RAM, en bloques de N datos, usando Data Transfer Controller (DTC).

Se implementaron cuatro rutinas de atención a interrupciones ISR: una para el Timer, una para el ADC y dos para la

Tabla I  
VALORES QUE PUEDE TOMAR LA FRECUENCIA DE MUESTREO  $f_s$ .

Parámetro	baudrate (bps)	$f_s$ (ksps)
0	9600	0.48
1	115200	5.7
2	256000	12.8
3	460800	22.8
4	921600	44

UART (transmisión y recepción).

Las interrupciones del Timer (*isr\_timer*) son periódicas y fijan la frecuencia de muestreo, ya que en su ISR se inicia la conversión del ADC. Cuando se trabaja con más de un canal de adquisición, previo a iniciar la conversión, se realiza la conmutación del canal. La ISR del ADC (*isr\_adc*) se dispara cuando se finaliza la transferencia a memoria de un bloque completo de N datos. Dentro de esta ISR simplemente se enciende la bandera *flag\_adc* cuyo estado es verificado en el *main*. Luego de adquirirse un bloque entero de datos, desde el *main* se inicia la transmisión. Se guardan los datos en la cola de transmisión de la UART y se habilitan las interrupciones de transmisión. En la ISR (*isr\_tx*) se copia el dato en el buffer de transmisión y se actualizan los punteros de la cola. Cuando la cola queda vacía se deshabilitan las interrupciones. Las interrupciones de recepción se disparan cuando llega un comando por UART. En esta ISR (*isr\_rx*) se obtiene el dato del buffer de recepción y se guarda en la cola de recepción de la UART. Finalmente se enciende la bandera *flag\_cmd* cuyo estado es verificado en el *main*.

### III. ANÁLISIS DE LOS CUELLOS DE BOTELLA

#### III-A. Reloj del sistema

El objetivo de muestrear a la máxima frecuencia posible implica utilizar un reloj que sea capaz de hacer funcionar el sistema en estas condiciones. El microcontrolador tiene integrado un Digitally-Controlled Oscillator (DCO) cuya frecuencia de trabajo puede variar en un rango amplio dentro de ciertos valores discretos. Para determinar con precisión las frecuencias a las que trabaja el DCO se observó en un osciloscopio la señal generada por éste en dos placas diferentes usando dos configuraciones diferentes. En la Tabla II se presentan los mayores valores obtenidos.

Tabla II  
MEDIDA DE LA FRECUENCIA REAL DEL DCO EN 2 PLACAS DIFERENTES USANDO RSELX = 15 Y MODX = 0.

Placa	Frecuencia (DCOx = 3)	Frecuencia (DCOx = 7)
1	13.7 MHz	19 MHz
2	13.4 MHz	18 MHz
Valor esperado [3]	12-18.5 MHz	16-26 MHz

Ya que el reloj principal del microprocesador no puede trabajar a más de 16 MHz, se configura el DCO para que trabaje a  $f_{DCO} = 13.7$  MHz, quedando determinada en ese valor la frecuencia del MCLK y el SMCLK.

### III-B. Velocidad del ADC

La frecuencia máxima de adquisición del ADC se midió conmutando una señal digital de salida en cada interrupción/adquisición del ADC. Se obtuvo una señal cuadrada de frecuencia 104 kHz por lo que se concluye que la frecuencia de muestreo es 208 kHz (ver Fig. 4), valor que supera los 200 ksp especificados en la hoja de datos.

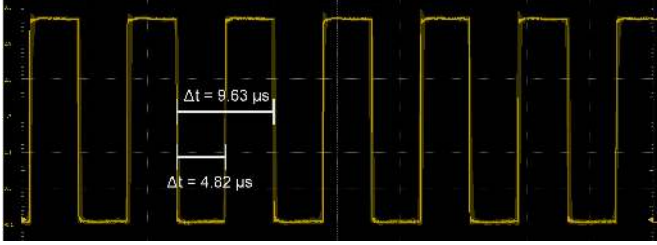


Figura 4. Máxima velocidad de adquisición del ADC

### III-C. Procesamiento de los datos

El tiempo de procesamiento  $t_{proc}$  se define como el tiempo entre que se dispara la adquisición de una muestra y ésta se guarda en el buffer de transmisión, e incluye cualquier procesamiento que se haga con los datos. Asimismo, se considera el mínimo procesamiento posible, que consiste solamente en guardar el dato adquirido en la cola de transmisión. El ADC con DTC interrumpe cuando finaliza la adquisición de un bloque. La medida del tiempo de procesamiento arrojó  $t_{proc} = 8,1 \mu s$  en la primer muestra del bloque y  $t_{proc} = 7,15 \mu s$  para cada una de las restantes, pues en estas últimas no se ingresa a la ISR del ADC. El bloque utilizado fue de tamaño 50 muestras. En la Ec. (1) se presenta el cálculo del tiempo requerido para el procesamiento de  $N = 50$  muestras con DTC.

$$t_{proc}^{conDTC} = 49 \times 7,15 + 8,1 = 358,45 \mu s \quad (1)$$

De esta manera, se obtiene que la condición que debe cumplir la frecuencia de muestreo está dada por la Ec. 2.

$$f_s < \frac{N}{t_{proc}^{conDTC}} \Rightarrow f_s < 123,4 \text{ kHz} \quad (2)$$

### III-D. Transmisión por UART

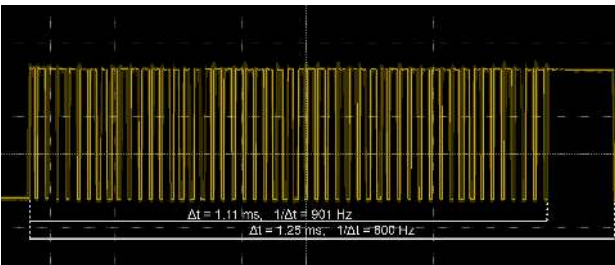


Figura 5. Tiempos de transmisión a una tasa de 921.6 kbps

Se configura la UART para enviar datos a una tasa de 921.6 kbps (se intentó usar valores más altos pero no se

logró que funcionara). En la Fig. 5 se muestran las medidas realizadas. Se registra un tiempo de transmisión de  $22.7 \mu s$  y  $1.11 \text{ ms}$  para las 50 muestras, se concluye por tanto que la máxima frecuencia de muestreo para una tasa de 921.6 kbps en transmisión es:

$$f_s^{max} = 44 \text{ kHz}$$

## IV. DISCUSIÓN Y CONCLUSIONES

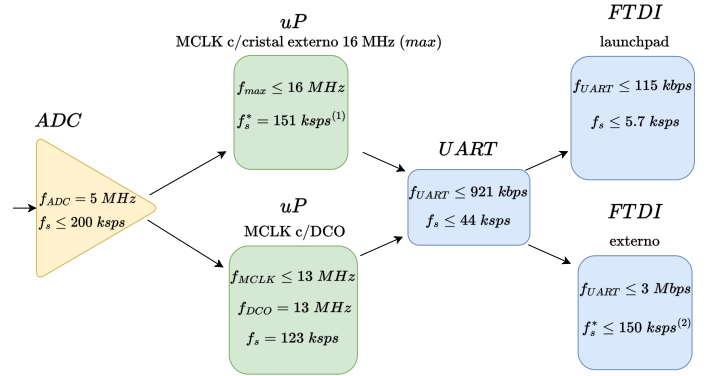


Figura 6. Máxima  $f_s$  alcanzada según cada etapa. (1) y (2) son estimaciones

El resumen de las restricciones se condensa en la Fig. 6. Se verificó que el ADC es capaz de adquirir 200 ksp. El procesamiento mínimo (encolar el dato adquirido para transmitir) reduce a 123 ksp la frecuencia de muestreo usando el DCO. Una posible solución a la variabilidad del DCO entre placas sería usar un cristal de 16 MHz, esto además permitiría llevar la velocidad de reloj interno del microprocesador a su límite, aumentando el tiempo disponible para procesamiento y por tanto la máxima frecuencia de muestreo a 151 ksp. El primer cuello de botella detectado fue el adaptador USB/UART del Launchpad que solo permite muestrear a 5,7 ksp. Eso se solucionó agregando un adaptador externo, que permite llevar la frecuencia de muestreo a 150 ksp. Finalmente, el cuello de botella es el periférico UART del microcontrolador, que limita la frecuencia de muestreo a 44 ksp. A futuro nos proponemos sustituir el AD2 por el Front-End analógico desarrollado en nuestro laboratorio basado en [4], y realizar pruebas en personas usando el sistema desarrollado.

## AGRADECIMIENTOS

Este trabajo fue financiado por CSIC (Comisión Sectorial de Investigación Científica, Uruguay) y Erasmus+ Project NEON, 618942-EPP-1-2020-1-AT-EPPKA2-CBHE-JP.

## REFERENCIAS

- [1] *MSP Debuggers User's Guide*, Texas Instrument, 2015. [Online]. Available: <https://www.ti.com/lit/ug/slau647o/slau647o.pdf>
- [2] D. E. Simon, *An embedded software primer*. Addison-Wesley Professional, 1999, vol. 1.
- [3] *Mixed signal microcontroller*, Texas Instrument, 2013. [Online]. Available: <https://www.ti.com/lit/ds/symlink/msp430g2553.pdf>
- [4] J. Oreggioni, A. Caputi, and F. Silveira, "Current-efficient preamplifier architecture for cmrr sensitive neural recording applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no. 3, pp. 689–699, jun 2018.

**Congreso Argentino de Sistemas Embebidos  
CASE 2023  
Libro de trabajos en modalidad artículo, foro tecnológico y reporte**

ISBN 978-631-90145-0-1

