

CASEE 2015

2015 Sixth Argentine Conference on Embedded Systems (CASE)

■ Regular papers.



www.sase.com.ar

University of Buenos Aires, Argentina

IEEE Catalog Number CFP15C02-PRT | ISBN 978-987-45523-4-1

2015 Sixth Argentine Symposium and Conference on Embedded Systems (CASE)

Regular Papers

August 12th-14th, 2015
School of Engineering - University of Buenos Aires
Buenos Aires, Argentina



2015 Sixth Argentine Conference on Embedded Systems (CASE) : Regular Papers /
José Lipovetzky , Maximiliano Antonelli, Diego Brengi, Luciana De Micco,
Ariel Lutenberg,... [et.al.]. - 1a ed. - Ciudad Autónoma de Buenos Aires : ACSE -
Asociación Civil para la investigación, Promoción y Desarrollo
de Sistemas Eléctricos Embebidos, 2015.
50 p. : il. ; 29x20 cm.

ISBN 978-987-45523-4-1

1. Ingeniería. 2. Actas de Congresos. I. Lipovetzky, José
CDD 620.711

ISBN Date: 29/05/2015

2015 Sixth Argentine Symposium and Conference on Embedded Systems (CASE)

Regular Papers

Printed Book ISBN 978-987-45523-4-1
E-Book ISBN 978-987-45523-5-8

Printed Book IEEE CATALOG CFP15C02-PRT
IEEE XPLORE COMPLIANT CFP15C02-ART

Editors:

Antonelli, Maximiliano. UNMDP
Brengi, Diego. INTI/UNLaM
De Micco, Luciana. UNMDP/CONICET
García Inza, Mariano. FIUBA
Lipovetzky, José. IB/CNEA/CONICET
Lutenberg, Ariel. FIUBA/CONICET/ACSE

Reprint Permission is permitted with credit to the source.
All rights reserved.
Copyright 2015 by IEEE and ACSE.

Preface

The development of Embedded Systems is crucial for the development of industry, technology and science; and it is an area which has grown in the past years in Argentina and South America.

The SASE and CASE are intended to promote the development of embedded systems in the country and the region with the following activities:

- Presentation of scientific and technological works at the conference.
- Hands-on Workshops.
- Technical Lectures (Tutorials).
- Plenary sessions.
- Contest for students projects
- A program to assign electronic equipment to Universities.
- Travel and lodging grants for graduate and postgraduate students, professors and researchers of Argentina.

The objectives of the SASE are:

- To allow a more fluid exchange between academia and industry.
- To promote the exchange between researchers and students from different universities from Argentina and other countries.
- To allow the diffusion of scientific and technological development done in the region and worldwide.
- To encourage students from the country to get interested in the development of Embedded Systems.
- To coordinate and promote the actualization of curriculum contents related to Embedded Systems in undergraduate and graduate programs at the universities of Argentina.

This year, topic areas included at CASE are: Architecture of Microprocessors, ASICs, DSPs, FPGA and HDLs, Implementation of Embedded Systems, Embedded Systems Linux, Communications and Protocols, Embedded Software, Robotics, RTOS and Bioengineering. Scientific works were accepted in three categories, Regular Papers, Technical Forum, and Poster. This book contains the works accepted as Regular Papers. After an exhaustive peer review process only seven papers were accepted in this distinguished category and published here. Some of the other ones were moved to the categories of technological forum or poster.

We hope you enjoy the following papers, and the conference.

CASE 2015 Organizing Committee

Sponsors

Diamond Sponsors

- Elemon
- Emtech
- Intel
- Coradir
- VICDA Argentina
- CIKA
- Asembli
- Electrocomponentes

Platinum Sponsors

- Probattery
- Synopsys
- Telit

Gold Sponsors

- INVAP
- SUR Emprendimientos Tecnológicos
- CADIPEL
- Dai Ichi
- Freescale

Silver Sponsors

- AR-SAT
- L&R Ingeniería
- Digi International

Institutions which support SASE

Scientific Associations

- IEEE Argentina (Institute of Electrical and Electronics Engineers)
- IEEE CASS (Circuits and Systems Society)

Cameras and Industrial Associations

- ADIMRA (Asociación de Industriales Metalúrgicos de la República Argentina)
- CADIEEL (Cámara Argentina de Industrias Electrónicas, Electromecánicas y Luminotécnicas)
- CAME (Confederación Argentina de la Mediana Empresa)
- CIIECA (Cámara de Industrias Informáticas, Electrónicas y de Comunicaciones del Centro de Argentina)

Scientific and Technological Institutions

- CoNAE (Comisión Nacional de Actividades Espaciales)
- CONICET (Consejo Nacional de Investigaciones Científicas y Técnicas)
- INTI (Instituto Nacional de Tecnología Industrial)
- Fundación Sadosky (Investigación y Desarrollo en TIC)
- Fundación Fulgor

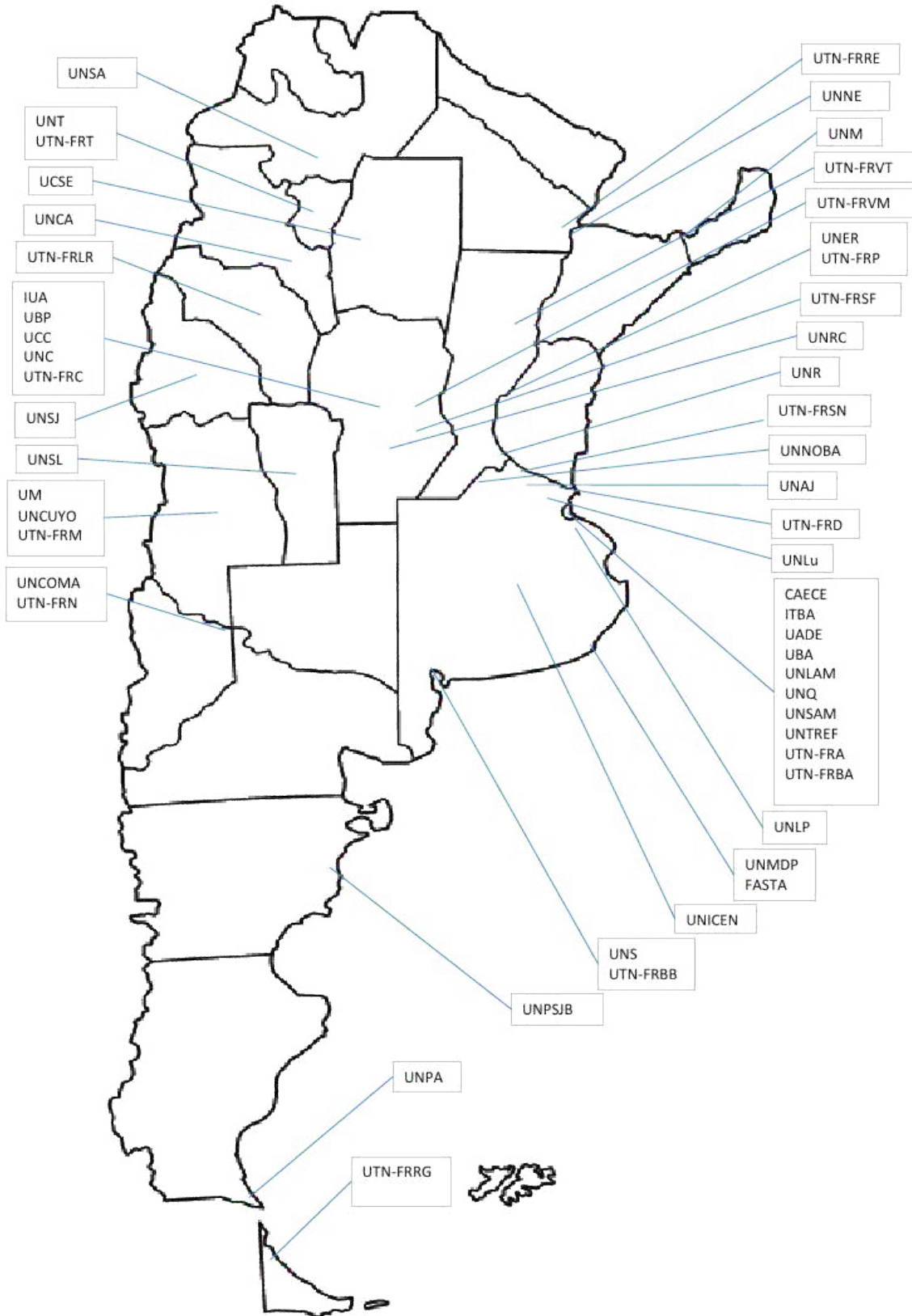
National Ministries

- MinCyT (Ministerio de Ciencia, Tecnología e Innovación Productiva)
- ANPCyT (Agencia Nacional de Promoción Científica y Tecnológica)
- Ministerio de Industria

University networks

- CONFEDI (Consejo Federal de Decanos de Ingeniería)
- RUSE (Red Universitaria de Sistemas Embebidos)

Universities which support SASE



Technical Program Committee 2015

SASE General Chair

- Dr. Ariel Lutenberg (FIUBA/CONICET/ACSE)

CASE Technical Program Chair:

- Msc. Diego Brengi (INTI/UNLaM, IEEE Senior Member)

CASE Publication Chairs:

- Dr. Luciana De Micco (UNMDP/CONICET, IEEE-CASS Member)
- Dr. José Lipovetzky (IB/CNEA/CONICET, IEEE-CASS Member)
- Dr. Mariano García Inza (FIUBA)
- Eng. Maximiliano Antonelli (UNMDP)

CASE Topic Chairs:

- Bioengineering: Eng. Juan Manuel Reta (UNER)
- DSPs: Dra. María Liz Crespo (ICTP)
- Embedded Linux: Eng. Lucas Chiesa (FIUBA)
- Embedded Software: Dr. Ricardo Medel (Ascentio Technologies)
- FPGAs, HDLs and ASIC: Eng. Salvador Tropea (INTI/UTN-FRBA)
- Implementation of Embedded Systems: Msc. Cristian Sisterna (UNSJ)
and Dr. Gustavo Sutter (UAM)
- Processor's architecture: Eng. Alejandro Furfaro (UTN-FRBA)
- Protocols and communications: Eng. Gustavo Mercado (UTN-FRM)
- Robotics: Dr. Luis Canali (UTN-FRC)
- RTOS: Dr. Ricardo Cayssials (UNS)
- Wireless communications: Dr. Leonardo Rey Vega (FIUBA)

Reviewers

Alcalde Bessia, Fabricio	Lutenberg, Ariel
Alessandrini, Gustavo	Marchi, Edgardo
Antonelli, Maximiliano	Martos, Pedro
Arias, Ricardo	Mas, Ignacio
Brenji, Diego	Mata, Walter A.
Briff, Pablo	Medel, Ricardo
Burgos, Enrique Sergio	Melo, Rodrigo
Calarco, Nicolás	Monte, Gustavo
Canali, Luis Rafael	Oliva, Rafael
Carbonetto, Sebastián	Orallo, Carlos Martin
Carrá, Martín	Perez Paina, Gonzalo F.
Cayssials, Ricardo	Perez, Martín
Cogo, Jorge	Perez, Santiago
Crespo, Maria Liz	Petrashin, Pablo Antonio
De Micco, Luciana	Pose, Claudio
Escudero, Gustavo	Pucheta, Julian
Ferrao, Hilda	Puga, Gerardo Ludovico
Ferreyra, Pablo Alejandro	Rey Vega, Leonardo
Filomena, Eduardo	Reyes, Benjamin
Fraire, Juan Andres	Ridolfi, Pablo
Furfaro, Alejandro	Rocha, Fábio
Garcia Inza, Mariano	Romeo, Marcelo
Garcia, Javier	Roncagliolo, Agustín
Gavinowich, Gabriel	Sambuco Salomone, Lucas
Gayoso, Carlos Arturo	Sisterna, Cristian
Giribet, Juan Ignacio	Sofo Haro, Miguel
Grimblatt, Victor	Sutter, Gustavo
Gwirc, Sergio N.	Taffernaberry, Carlos
Hernandez Tabares, Lorenzo	Todorovich, Elías
Juarez, Gustavo	Tropea, Salvador
Leiva, Lucas	Weisz, Rosa María
Lipovetzky, Jose	Zacchigna, Federico G.
Lozano, Clevis	Zaradnik, Ignacio

Table of Contents

Preface.	iii
Sponsors.	v
Technical Program Committee.	ix
INS/Ultrasound navigation system. <i>Patricio Moreno, Claudio Pose and Juan Giribet.</i>	1
FPGA-based floating-point UD filter coprocessor for integrated navigation systems. <i>Rodrigo Gonzalez, Gustavo Sutter, Cristian Sisterna and Héctor Daniel Patiño.</i>	7
MESA: A Formal Approach to Compute Consensus in WSNs. <i>Zacchigna, Federico G.; Lutenberg, Ariel and Vargas, Fabian.</i>	13
Accuracy of Propagation Models to Power Prediction in WSN ZigBee Applied in Outdoor Environment. <i>Teles de Sales Bezerra, José Anderson Rodrigues de Sousa, Saulo Aislan da Silva Eleutério and Jerônimo Silva Rocha.</i>	19
Low cost attitude estimation system. Performance evaluation on an airbearing based testbed. <i>F. von Bergen, J. Giribet and P. Martos.</i>	25
A queue-based scheduling algorithm for PCE-enabled Industrial Internet of Things networks. <i>Ángelo Antonio Farías and Diego Dujovne.</i>	31
FreeRTOS User Mode Scheduler for Mixed Critical Systems <i>Francisco E. Páez, José M. Urriza, Ricardo Cayssials and Javier D. Orozco.</i>	37
Author index	43

INS/Ultrasound navigation system

Patricio Moreno*, Claudio Pose*

*Grupo de Procesamiento de Señales, Identificación y Control
Facultad de Ingeniería, Universidad de Buenos Aires
Paseo Colón 850, CABA, Argentina
Email: {pamoreno|cldpose}@fi.uba.ar

Juan Giribet*†

†Instituto Argentino de Matemática
CONICET
Saavedra 15, CABA, Argentina
Email: jgiribet@fi.uba.ar

Abstract—The aim of this work is to present a navigation system for GPS denied environments. The system fuses measurements provided by an arrangement of ultrasonic range sensors with inertial measurements and a magnetometer. The core of the navigation computer are two microprocessors, an ARM Cortex M3 processor for low-level tasks, such as sensor acquisition and control, and an ARM Cortex A8 for computationally demanding tasks, as the fusion filter to compute navigation solutions. A brief description of the navigation computer is given, also the navigation algorithm is presented and validated through experimental results.

I. INTRODUCTION

There is a wide variety of sensors for implementing mobile robot navigation system; inertial sensors (accelerometers and gyroscopes) are quite common sensors because they provide data at high rate and are robust to external interferences. Moreover, MEMS (Micro Electro Mechanical Systems) technology brought the possibility of miniaturize these sensors and decrease its weight and costs. However, navigation systems based on inertial sensors have their limitations, since they compute the navigation solutions integrating accelerometers and gyroscope measurements, the navigation error increases with time, thus they cannot be used to provide relatively long term accurate navigation solutions. This is an inherent disadvantage of inertial systems and does not depend on inertial sensor technology [1]. However, it is more noticeably with MEMS inertial sensors because they have significative errors such as, bias, scale factor, non-linearities or thermal drifts. These errors continuously degrade the navigation solution in short term.

On the other hand, sensors such as GPS, cameras, RFID (Radio Frequency IDentification) signals, magnetometers, LIDAR (Laser Imaging Detection And Ranging), among others, provide navigation solution with bounded errors, making them useful for long term navigation applications. However, these (external) sensors usually provide data at low rate (which is a limitation when we are working with vehicles with high dynamics), also its measurements depend on external parameters (satellites signals, image features, beacons, magnetic fields) hence they suffer from a lack of robustness. The synergy between inertial and external sensors, brings the possibility of complementing them. While inertial sensors continuously provide navigation solution, external sensors are useful to control the navigation error increase. This is achieved by

means of a fusion filter (usually an Extended Kalman Filter [1]).

In this work, an array of ultrasonic sensors is chosen as an external navigation system. These sensors are lightweight, low-cost, and the provided information requires little processing. On the other hand, ultrasonic sensors have disadvantages, its measurements are affected by the surface in which are reflecting, the lack of directionality of these sensors also affects its performance, event though these sensors are suitable for a large range of applications, for instance they can be used in GPS denied environments. Camera sensors, for example, depend on correct detection of features to compute navigation solutions, thus they can not be used in smoke environments. Besides, vision based techniques are time-consuming and require high computing power [2]. Others sensors like LIDAR are heavy and expensive, RFID sensors depends on a dedicated infrastructure to provide navigation information [3]. RSSI (Received Signal Strength Indicator) has also been widely used for indoor positioning, using Wireless LAN networks ([4], [5]) or Wi-Fi and Bluetooth [6].

However, working with ultrasonic sensors for precise positioning is not an easy task and depends on navigation algorithm performance. There are different techniques for using ultrasonic sensors for navigation ([7],[8]). For instance, in [9] the ultrasonic sensors are used as beacons to assist in a pedestrian navigation system. Bouabdallah and Siegwart [10] used ultrasonic sensor not only to measure the altitude of a quadrotor but also to avoid obstacles, also in [11] ultrasonic sensors has been used to avoid obstacles with a quadcopter.

In this work, we first introduce the navigation computer developed in the GPSIC Lab and the board carrying ultrasonic sensors. The computer, based on 2 processors, acquire and synchronise navigation measurements (this task is performed by a 32-bit LPC1769 microcontroller), then sends (time-tagged) navigation measurements to a Beagleboard xM with a Cortex-A8 that runs an Extended Kalman Filter (EKF) to compute navigation solution. A computer based on two processors is an adequate design for navigation, guidance and control (NG&C) systems, the main motivation is because it allows to work with critical time tasks (such as sensor acquisition and control algorithms) and with high computing demanding tasks (such as navigation and guidance algorithms).

While this computer was designed for multirotor-type micro aerial vehicles (MAV), with the purpose of achieving a com-

plete autonomous flying vehicle (see [12], [13]), the navigation data may be used in different vehicles. In fact, we used a modified RC (Radio Controlled) car to validate the navigation system performance, the complete experimental setup can be seen in Fig. 5.

After describing the NG&C computer, we present the sensor board that carries the ultrasonic sensors. Also, we introduce the method to compute vehicle position from ultrasonic measurements. Then, the position data is fed into a EKF (update-step), which uses inertial sensors measurements for solving the prediction-step.

II. SYSTEM OVERVIEW

A. NG&C computer prototype

The NG&C computer was developed in the GPSIC Lab, with the objective of being used as the onboard computer in a multirotor-type MAV, particularly an hexacopter. However, as it was mentioned, for this work we installed this board on a RC car for experimental validation of the navigation system.

This computer is based on two processors. The 32-bit ARM Cortex-M3 LPC1769 [14] is the low-level processor, whose tasks include reading the sensors, adding a timestamp, executing PID control loops and sending navigation and status information to a ground station.

Although the NG&C board includes navigation sensors, several additional sensors can be added according to the application. The board includes an Inertial Measurement Unit (IMU) (MPU6000 from Invensense [15]), this IMU uses SPI protocol and provides measurements from three accelerometers and three gyroscopes. The board also allows to connect an IMU from Analog Devices; this is useful since Analog Devices offers a broad spectrum of (MEMS) IMU, and different sensors can be chosen depending on the necessary navigation performance. Another sensor included is a triple axes I²C digital compass code HMC5883 from Honeywell [16], which is useful to provide relative orientation respect to magnetic north.

Additional sensors frequently used are RS-232 GPS (ET-332 from GlobalSat [17]), I²C barometer (BMP180 from Bosch [18]) and a single I²C SRF02 [19] ultrasonic range sensor from Devantech for height control (useful when we are working with MAV). Also a GPS-RTK (MB-100 from Trimble [20]) has been tested with this board, for precise positioning of a MAV.

For the timestamp task an internal LPC1769 clock is used, its drift is corrected using the PPS signal from the GPS, if it's available. After adding a timestamp to every measurement, the navigation data is sent to the high-level processor, a Beagleboard xM with an ARM Cortex-A8 core running a debian-based embedded GNU/Linux. This high-level processor runs an Extended Kalman Filter to compute the navigation solution. The EKF estimates vehicle's attitude, position and velocity, also it estimates and compensates sensors bias, scale factors and misalignments.

Both the computer prototype and the block diagram are depicted in Fig. 1 and 2, respectively.



Fig. 1. NG&C computer prototype

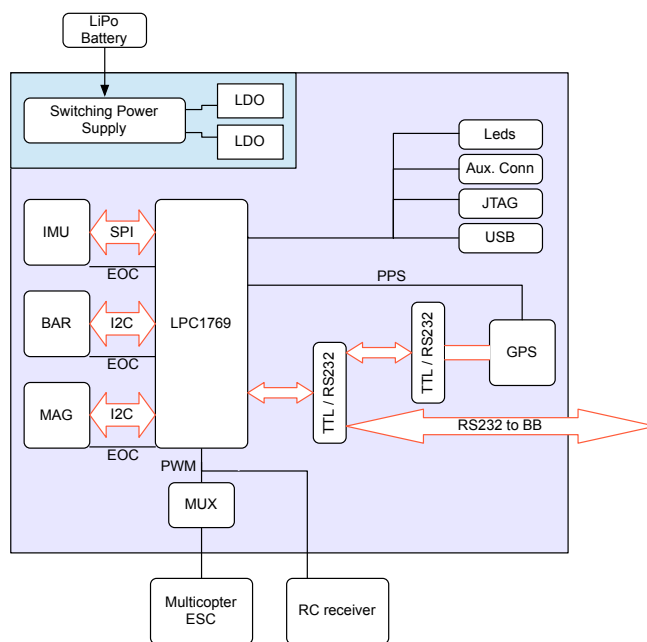


Fig. 2. Block diagram of the NG&C computer

The Beagleboard xM [21] runs an embedded Debian GNU/Linux 7 distribution with a 3.2.24-x14 Linux kernel. The position estimation algorithm and the EKF were implemented using Python 2.7.3 in sake of portability.

The two processors share information through a serial channel, originally implemented following the RS-232 standard, but later implemented through a UART. An ad-hoc protocol was implemented, which specifies a message as in Fig. 3.

Although it is not directly related with this work, it is worthy to mention that control and guidance algorithm has been implemented on this board to work with MAVs. The purpose of implementing a high-level control algorithm could be to improve the performance of the low-level attitude control, to solve the position control or implement a fault tolerant control strategy. Moreover, camera sensors has been used to provide navigation data.

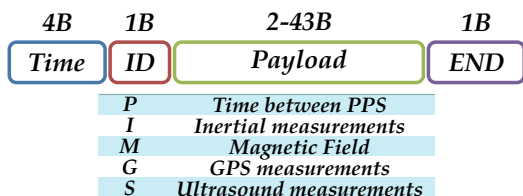


Fig. 3. UART communication protocol used



Fig. 4. Ultrasonic sensor board w/ eight sensors, over NG&C computer

B. Ultrasonic sensors board

The ultrasonic sensor that was chosen for navigation is the SRF02 transceiver from Devantech. This sensor has I²C interface with sixteen programmable addresses, which allows to use a single I²C bus with that maximum number of sensors. This transceiver allows a simple usage, starting a new measurement with a *range* command which saves the distance to the nearest object in 2 registers. The result may be in centimetres, inches or microseconds, depending on the command used. If any object is closer than approximately 15 cm the distance is saved as 0 cm and if it is 5 m or further, that value is stored. It has a good lobe-to-price relation (as this type of sensors become more expensive when they are very directional, i.e. have a narrow lobe), enough for the tests that were conducted. As most of economic ultrasonic sensors, measurements are some times corrupted by outliers. The board was designed to allow sixteen sensors, although experiments where conducted using eight and/or four.

The sensors were disposed for the tests in an equally spaced circular pattern, with a total of eight sensors separated 45° (as in Fig. 4), and four sensors at 90° for different tests (alternately pulling out a sensor in the figure).

The board design includes an I²C level shifter from 3.3 V to 1.8 V selectable (LPC1769 and Beagleboard levels) to 5 V (SRF02 levels), as well as some indicator leds that show the status of the power supplies.

As the frequency of the ultrasonic burst is the same for all sensors and cannot be modified, a sequential reading is performed to avoid interference between sensors. As the measuring may take up to 70 ms, a full reading may be done at a maximum frequency of ≈ 1.8 Hz for the eight sensor array, and ≈ 3.6 Hz in the four sensor case. Consequently, that is the maximum frequency for position estimation and for the EKF update-step.

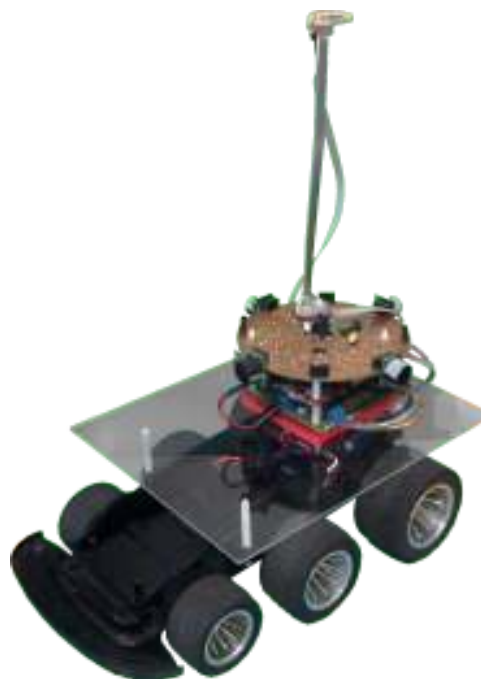


Fig. 5. Test platform

To maximize the frequency at which position estimation is carried on, the number of sensors used for experimental validation was 4. Using 8 sensors halves this frequency but doubles the information for estimate the position, hence there is a trade off between the number of sensors used and the update interval. Also, less sensors make the board lighter, allowing it's use in smaller MAVs, and cheaper, being more accessible.

III. POSITION ESTIMATION

The data transmitted from the microcontroller to the microprocessor is collected by an application running under GNU/Linux. This application, programmed in Python, arranges and enqueues the data for the data fusion filter to use it. Besides the SRF02 measurements, the attitude of the vehicle is needed. As the sensors have a minimum and maximum range, the quality of the data is based on the number of measurements in the range (0.15 m, 5 m) over the total number of measurements.

To estimate the position, the algorithm in Fig. 6 is used. When ray casting algorithm is applied all measurements are considered to be taken at the same time. It would be possible to work with measurements taken at different times and apply the same algorithm by extrapolating this measurements to a common time. Then, a measurement is defined as a pair (d_i, θ_i) where d_i is the measured value and θ_i is the orientation of the sensor. Each sensor orientation, in this test board, is calculated from the navigation system data taking in account their known fixed positions with respect to the x axis of the vehicle. Nonetheless, the algorithm considers non-fixed relative angles from one sensor to another, so each orientation

Require: $\mathcal{M} = \{(d_1, \theta_1), \dots, (d_N, \theta_N)\}$
Require: $map :=$ environment model

```

1: procedure POSITION ESTIMATION( $\mathcal{M}, map$ )
2:    $\mathcal{R} = \{m \in \mathcal{M} : d(m) \leq 5\}$ 
3:   if  $|\mathcal{R}| < 2$  then
4:     return  $\emptyset$ 
5:   end if
6:    $\mathcal{S}_x = \{d(r) \cos(\theta(r)), r \in \mathcal{R}\}$ 
7:    $\mathcal{S}_y = \{d(r) \sin(\theta(r)), r \in \mathcal{R}\}$ 
8:    $\mathcal{P}_x = \{SxToMap(\xi, map), \xi \in \mathcal{S}_x\}$ 
9:    $\mathcal{P}_y = \{SyToMap(\eta, map), \eta \in \mathcal{S}_y\}$ 
10:   $\mathcal{C} = \{\emptyset\}$ 
11:  for  $p_x \in \mathcal{P}_x$  do
12:    for  $p_y \in \mathcal{P}_y$  do
13:       $\mathbf{p} = (p_x, p_y)$ 
14:       $\hat{\mathcal{R}} = RayCast(\mathbf{p}, \{\theta(r), r \in \mathcal{R}\}, map)$ 
15:       $\Delta = \{|d(\hat{r}) - d(r)|, (\hat{r}, r) \in (\hat{\mathcal{R}}, \mathcal{R})\}$ 
16:       $\epsilon = \frac{1}{|\Delta|} \sum \delta, \forall \delta \in \Delta$ 
17:      if  $\epsilon < THRESHOLD$  then
18:         $\mathcal{C} = \mathcal{C} \cup \{(\epsilon, \mathbf{p})\}$ 
19:      end if
20:    end for
21:  end for
22:   $\mathbf{P} = \frac{1}{|\mathcal{C}|} \sum_{\forall \epsilon, \mathbf{p} \in \mathcal{C}} w(\epsilon, \mathbf{C}) \mathbf{p}$ 
23:  return  $\mathbf{P}$ 
24: end procedure

```

Fig. 6. Position estimation algorithm

is added to the measurement pair. For simplicity a function $d(\cdot)$ is defined, in the algorithm, that receives a measurement pair and returns the measured value. Analogously, the $\theta(\cdot)$ function was defined for the second element of the pair.

Firstly, measurements outside the valid range are removed. If remaining measurements aren't enough to estimate the position, an empty set is returned. In lines 6 and 7 two sets, \mathcal{S}_x and \mathcal{S}_y , are constructed with the projections of the measurements to the orthogonal axes centered in the sensor. Then, the functions $SxToMap()$ and $SyToMap()$ are used to calculate a set of possible positions for each axis from the previous sets and the environment map. Once the possible positions are computed, for each possible combination of positions, a ray casting approach is used to estimate a new set $\hat{\mathcal{R}}$ using the $RayCast()$ function. The latter function receives a point in space, the orientation of each ultrasonic sensor and the map, and returns the expected measurement for each ultrasonic sensor, consistent with the vehicle position. The returned values are compared with the ones in the \mathcal{R} set and a mean deviation, ϵ , from them is computed. If ϵ is less than a *threshold*, the point \mathbf{p} is added to a set \mathcal{C} of candidate positions. Finally, the position is estimated as a weighted unbiased mean of all elements in \mathcal{C} . The weights are

$$w(\epsilon_k, \mathcal{C}) = 1 - \frac{\epsilon_k}{\sum_{\epsilon \in \mathcal{C}} \epsilon} \quad (1)$$

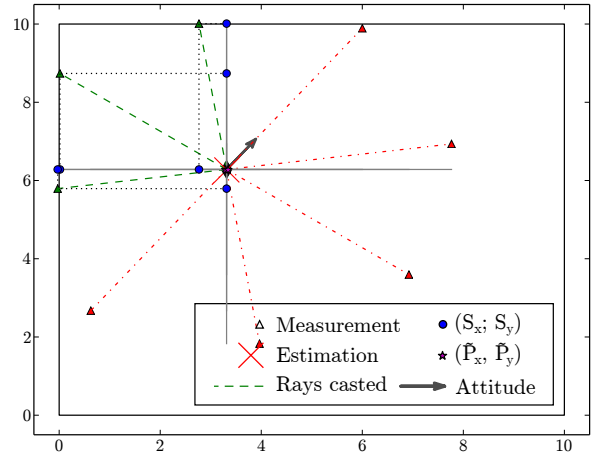


Fig. 7. Simulation of a position estimation step with 8 ultrasound sensors

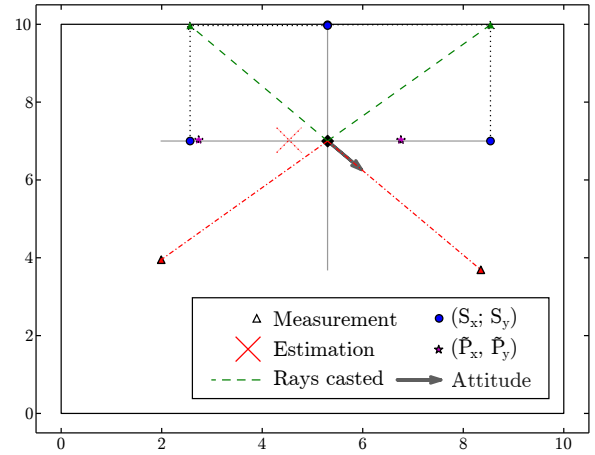


Fig. 8. Simulation of a position estimation step with 4 ultrasound sensors

In Fig. 7 and 8 a positioning step is shown, for the cases with 8 and 4 sensors respectively. In both figures, all measurement rays are shown, and the measured point is marked with a Δ . The rays that hit a wall were drawn with a dashed line, while the ones that give no location information were drawn with a dash-dotted line. A black diamond marks the real (unknown) position and a pair of axis come out of it. The projections of the useful measurements, marked with circles, are the ones saved in sets \mathcal{S}_x and \mathcal{S}_y . The sets \mathcal{P}_x and \mathcal{P}_y are formed using the functions $SxToMap()$ and $SyToMap()$, and are marked with stars. The final estimated position is marked with a cross. The arrow shows the heading of the vehicle. The difference between both figures, other than the numbers of sensors, is that in Fig. 8 the number of sensors that reach the wall is the minimum and p_y remains unknown.

IV. DATA FUSION

The Beagleboard receives three types of data: inertial, compass and ultrasound.

The inertial data is composed by the measurements of three accelerometers plus three gyroscopes. The IMU is read at 800 Hz, but a simple filtering method is used in the LPC1769

that averages ten samples, to reduce the noise (mainly from the accelerometers) resulting in a 80 Hz inertial data output (which is also used for PID control loops in the UAV). Despite the ten averaged samples being taken in ten different time instants (each one 1.25 ms apart from another), due to the slow dynamics of the system, it may be approximated that these samples are measured at the same time. So a unique timestamp is used to tag the inertial measurements.

As the raw data from magnetometers is processed in the low-level microprocessor to obtain the yaw angle, and, consequently, control the vehicle in that angle, it would be redundant to send the unprocessed data to the high-level processor and compute the yaw angle again. So, in this case, the relative yaw angle regarding the magnetic north is sent.

The ultrasonic sensors are read in sequence, as stated before but, in this case, time between measurements is greater and the simultaneous reading approximation may or may not be done, depending on the speed of the system dynamics. In consequence, each ultrasonic measurement has its own timestamp that refers to the time the data was read from the LPC1769. In this tests, the dynamics are slow and the number of ultrasonic sensors is the minimum, so the simplification can be done. If more sensors are used or dynamics becomes faster, the time tags are available to be taken into account.

As the IMU and the magnetometer are anchored to the vehicle's body, the measurements are relative to the reference frame fixed to it. The orientation of this frame of reference to the inertial frame of reference changes over time, thus the rotation matrix that relates both is needed. The mentioned rotation matrix is called C_b^i .

The dynamic equations of the system are described in Equation (2).

$$\dot{\mathbf{p}}^i = \mathbf{v}^i \quad (2a)$$

$$\dot{\mathbf{v}}^i = C_b^i \mathbf{f}^b + \mathbf{G}^i \quad (2b)$$

$$\dot{C}_b^i = C_b^i \mathbf{S}(\boldsymbol{\omega}_{ib}^b) \quad (2c)$$

being \mathbf{p} the position and \mathbf{v} the velocity expressed in the inertial frame. \mathbf{G} the gravity. \mathbf{f}^b and $\boldsymbol{\omega}_{ib}^b$ the IMU measurements, expressed in the body frame of reference.

Since it is numerically more stable and has no singularity points, the quaternion form of Equation (2) was used. The Runge-Kutta Method, RK4, was the implemented numerical method to find a numerical approximation to the solution in each step.

The state vector used in the Kalman filter is a perturbation model of the navigation variables, as in [22], with the addition of the accelerometer bias. This vector is shown in Equation (7). Position and velocity perturbations are defined as the difference between the estimated variable and the real value, as shown in Equations (3–4).

$$\delta \mathbf{p} = \hat{\mathbf{p}} - \mathbf{p} \quad (3)$$

$$\delta \mathbf{v} = \hat{\mathbf{v}} - \mathbf{v} \quad (4)$$

Consider the relation

$$C_b^i = M(\boldsymbol{\phi}) \hat{C}_b^i \quad (5)$$

$$M(\boldsymbol{\phi}) = C_b^i \hat{C}_b^i \quad (6)$$

$M(\boldsymbol{\phi})$ represents the rotation matrix needed to correct the error between the real C_b^i and the one estimated by the INS. From Equation (6) can be seen that if there's no error in the estimation, $M(\boldsymbol{\phi})$ must be $\mathbf{I} \in \mathbb{R}^{3 \times 3}$. Then, according to Euler's Theorem, $\boldsymbol{\phi} \in \mathbb{R}^3$ is the (unique) invariant vector for $M(\boldsymbol{\phi})$. Finally, \mathbf{b}_f is the accelerometer bias, and the state vector is

$$\mathbf{x} = [\delta \mathbf{p} \quad \delta \mathbf{v} \quad \boldsymbol{\phi} \quad \mathbf{b}_f]^T. \quad (7)$$

The dynamic of the state vector was modeled as shown in Equation (8)

$$\frac{d}{dt} \begin{bmatrix} \delta \mathbf{p}^i \\ \delta \mathbf{v}^i \\ \boldsymbol{\phi} \\ \mathbf{b}_f^i \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I} & 0 & 0 \\ 0 & 0 & -\mathbf{S}(C_b^i \mathbf{f}^b) & C_b^i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta \mathbf{p}^i \\ \delta \mathbf{v}^i \\ \boldsymbol{\phi} \\ \mathbf{b}_f^i \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & C_b^i & 0 \\ C_b^i & 0 & 0 \\ 0 & 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta \boldsymbol{\omega}_{ib}^i \\ \boldsymbol{\xi}_f \\ \boldsymbol{\xi}_{b_f} \end{bmatrix} \quad (8)$$

The measurements vector consists of the position estimation made by the algorithm and the measurements from the magnetometer. As the problem is solved for the 2D scenario, the constrain is added through the measurements vector, fixing the vehicle's height to a given value.

V. RESULTS

Fig. 5 shows the test platform with which the tests were done. In Fig. 9 the result of an experiment is shown. It can be seen that the estimated positions, in each step, have an error of less than 5 cm and follows the trajectory correctly. It's interesting to see that, at $y \approx 72$ cm, there's a discontinuity in the trajectory. As the test environment has a wall with a square protuberance, the trajectory discontinuity comes from an error in the environment model, where the wall is modeled as plain object.

To test the kalman filter implementation, the autocorrelation of the state innovations was computed and it's shown in Fig. 10 and 11. Although the graphics show a white-noise-like correlation, samples are not completely uncorrelated. These correlations arises from the processing that is done, sorting, weighting and taking the average of measurements related with the position instead of using the measurements directly. The higher correlation in Fig. 11 compared to Fig. 10 may, or may not, be related with the geometry of the environment, and further analysis is needed.

VI. CONCLUSIONS

A navigation capable system was designed and a working prototype was built. The low level computer runs the control loop at the stated 80 Hz and the kalman filter works correctly.

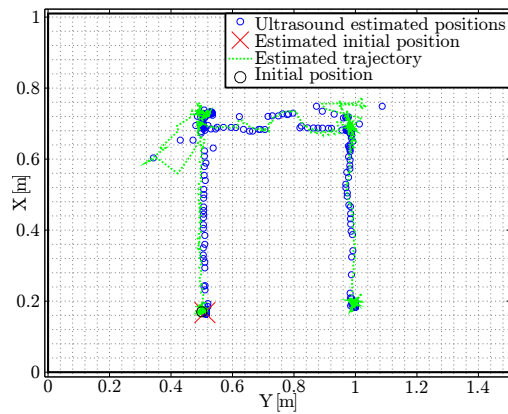


Fig. 9. Estimated trajectory in an XY plane for a real motion

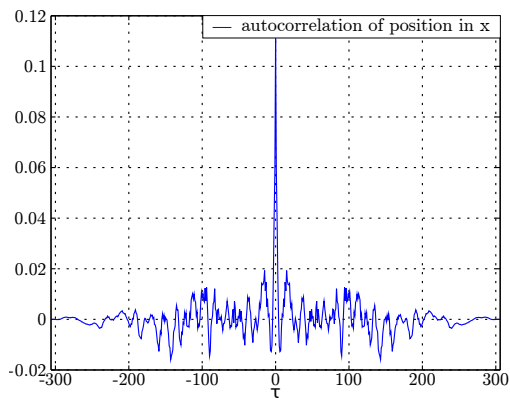


Fig. 10. Autocorrelation of position innovation for x axis

Using the algorithm discussed in previous sections, localization was done with an error of less than 5 cm. The ultrasound sensors can be used as an external aid to the INS in closed environments. This presents an improvement over the use of external beacons and broadens the use of navigation systems, as no previous configuration is needed.

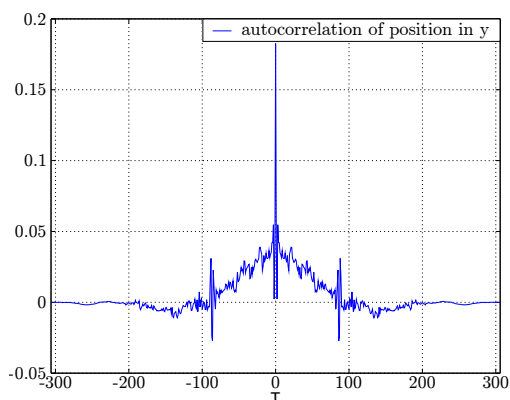


Fig. 11. Autocorrelation of position innovation for y axis

ACKNOWLEDGMENT

This work has been partially supported by ANPCyT-PICT1365 and PDTs-PI02

REFERENCES

- [1] M. España, *Fundamentos de la Navegación Integrada*. Asociación Argentina de Control Automático, 2010.
- [2] P. Alves, H. Costelha, and C. Neves, "Localization and navigation of a mobile robot in an office-like environment," in *Autonomous Robot Systems (Robotica)*, 2013 13th International Conference on, April 2013, pp. 1–6.
- [3] B. Olszewski, S. Fenton, B. Tworek, J. Liang, and K. Yelamarthi, "RFID positioning robot: An indoor navigation system," in *Electro/Information Technology (EIT)*, 2013 IEEE International Conference on, May 2013, pp. 1–6.
- [4] Y. Teramoto and A. Asahara, "Wireless LAN based indoor positioning using radio-signal strength distribution modeling," in *Indoor Positioning and Indoor Navigation (IPIN)*, 2012 International Conference on, Nov 2012, pp. 1–7.
- [5] D. Vilaseca and J. Giribet, "Indoor navigation using WiFi signals," in *Embedded Systems (SASE/CASE)*, 2013 Fourth Argentine Symposium and Conference on, Aug 2013, pp. 1–6.
- [6] A. Baniukevic, C. Jensen, and H. Lu, "Hybrid indoor positioning with Wi-Fi and bluetooth: Architecture and performance," in *Mobile Data Management (MDM)*, 2013 IEEE 14th International Conference on, vol. 1, June 2013, pp. 207–216.
- [7] F. Ijaz, H. K. Yang, A. Ahmad, and C. Lee, "Indoor positioning: A review of indoor ultrasonic positioning systems," in *Advanced Communication Technology (ICACT)*, 2013 15th International Conference on, Jan 2013, pp. 1146–1150.
- [8] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 6, pp. 1067–1080, Nov 2007.
- [9] Widyawan, G. Pirkel, D. Munaretto, C. Fischer, C. An, P. Lukowicz, M. Klepal, A. Timm-Giel, J. Widmer, D. Pesch, and H. Gellersen, "Virtual lifeline: Multimodal sensor data fusion for robust navigation in unknown environments," *Pervasive and Mobile Computing*, vol. 8, no. 3, pp. 388 – 401, 2012.
- [10] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *Intelligent robots and systems. IEEE/RSJ*, 2007, pp. 153–158.
- [11] J. Müller, A. Ruiz, and I. Wieser, "Safe and sound: A robust collision avoidance layer for aerial robots based on acoustic sensors," in *Position, Location and Navigation Symposium - PLANS 2014*, 2014 IEEE/ION, May 2014, pp. 1197–1202.
- [12] C. Pose, J. Giribet, J. Choclin, and A. Ghersin, "Sistema de navegación, guiado y control para un VANT multirroto," in *Congreso Argentino de Ingeniería Aeronáutica*, 2014.
- [13] C. Pose, F. Roasio, and J. Giribet, "Diseño de una computadora de NGC. aplicación a un vehículo aéreo no tripulado," in *Jornadas Argentinas de Robótica*, 2014.
- [14] NXP Semiconductors, "32-bit ARM Cortex-M3 microcontroller; up to 512 kB flash and 64 kB SRAM with Ethernet, USB 2.0 Host/Device/OTG, CAN," rev. 9.5, June 24, 2014.
- [15] Invensense, "MPU-60X0 6-Axis Gyro+Accel," rev. 3.1, October 24, 2011.
- [16] Honeywell, "3-Axis Digital Compass IC HMC5883L," rev. E, February 2013.
- [17] GlobalSat, "GPS Engine Board ET-332," version 1.3.1.
- [18] Bosch Sensortec, "Digital pressure sensor BMP180," rev. 2.8, May 7, 2015.
- [19] Devantech, "SRF02 Ultrasonic range finder Technical Specification."
- [20] AshTech, "MB 100 Reference Manual," 2010.
- [21] Beagleboard.org, "BeagleBoard-xM Rev C2 System Reference Manual," rev. C2, October 11, 2012.
- [22] R. Rogers, *Applied Mathematics in Integrated Navigation Systems*, ser. AIAA education series. American Institute of Aeronautics and Astronautics, 2003.

FPGA-based floating-point UD filter coprocessor for integrated navigation systems

Rodrigo Gonzalez*, Gustavo Sutter†, Cristian Sisterna‡, and Héctor Daniel Patiño§

*GridTICs and Lab. de Computación Reconfigurable, Universidad Tecnológica Nacional, Mendoza, Argentina. Email: rodralez@frm.utn.edu.ar

†Escuela Politécnica Superior, Universidad Autónoma de Madrid, España.
 Email: gustavo.sutter@uam.es

‡Instituto de Investigaciones Antisísmicas, Universidad Nacional de San Juan, Argentina.
 Email: cristian@unsj.edu.ar

§Instituto de Automática, Universidad Nacional de San Juan, Argentina.
 Email: dpatino@inaut.unsj.edu.ar

Abstract—The Kalman filter plays an essential role in an integrated navigation system. From an embedded-system design point of view, the UD filter is a convenient, numerically-stable version of the Kalman filter. In this paper, a UD filter coprocessor with single-precision floating-point format that runs in FPGA is presented. A comprehensive hardware/software exploration is carried out in order to find the combination of subsystems that achieves the best throughput. Such examination determines that the best solution is fully in hardware. In addition, it is demonstrated the convenience of using high-level development tools to synthesize in hardware algorithms that have been originally designed to run on a general purpose microprocessor. Then, a UD filter for a 21-states system is developed as a coprocessor for system-on-chip integration. The coprocessor is validated using real-world data sets. Measurements of area, power, and energy are provided. It is found that the coprocessor is suitable for mid-range FPGA. In conclusion, it is demonstrated that new hybrid FPGA are adequate devices to implement battery-operated navigation systems for robotics.

Keywords—UD filter, Kalman, Bierman, Thornton, navigation system, FPGA, floating-point

I. INTRODUCTION

The Kalman filter (KF) is used in the context of an integrated navigation system (INS) to fusion information coming from inertial sensors, gyroscopes and accelerometers, and aiding sensor, such as a GPS receptor or a barometer. The idea behind this technique is to achieve better estimates than in the case of using such sensors in an individual fashion.

In robotics applications, an INS is an embedded system with a numerical precision usually not greater than single-precision floating point (32 bits). In practice, since an INS Kalman filter is updated through finite precision computations, it may diverge due to round-off errors [1, Ch. 6]. Some methods have been developed to prevent the Kalman filter to experiment numerical instability. One of the most convenient formulations of the Kalman filter is the UD filtering [2], [3] owing to the fact that its computation is less complex than another stable formulations [4, Sec. 6.4]. The main goal of the UD filtering is to decompose covariance matrix P as $P = UDU^T$, with U unit upper triangular matrix and D

diagonal matrix. This operation is carried out by applying the UD factorization method and the Gram-Schmidt orthonormalization process. Then, U and D are updated and propagated instead of P . This method ensures that P will be symmetric and positive-definite over the entire filter operation, which is a key condition for Kalman filter stability.

The computational requirements of the UD filter are proportional to the size of the state vector. An INS typically uses between 15 [5, Eq. 12.50, p. 384] and 36 states [6] to estimate or correct the system states, although more states may be needed.

For example, Eq. 1 shows a state vector with 21 states,

$$\delta \hat{x} = \left[\delta \hat{e}^T, \delta \hat{v}^{nT}, \delta \hat{p}^{nT}, \hat{\mathbf{b}}_g^T, \delta \hat{\mathbf{b}}_g^T, \hat{\mathbf{b}}_f^T, \delta \hat{\mathbf{b}}_f^T \right]^T, \quad (1)$$

where 3 states are for attitude correction ($\delta \hat{e}$), 3 states are for velocity correction ($\delta \hat{v}^n$), 3 states are for position correction ($\delta \hat{p}^n$), 6 states are for static and dynamic gyroscopes biases estimations ($\hat{\mathbf{b}}_g$ and $\delta \hat{\mathbf{b}}_g$), and finally 6 states are for static and dynamic accelerometers biases estimations ($\hat{\mathbf{b}}_f$ y $\delta \hat{\mathbf{b}}_f$). Thus, matrices U and D are of order 21. According to [1, Table 6.15, p. 270, and Table 6.16, p. 274], this UD filter will require 27,758 FLOPS for one iteration, plus other time-consuming tasks such as data transfers from and to the main memory, and the control of for-loops, among others. In INS for robotics, inertial sensors measurements are normally delivered with a frequency of at least 100 samples per second, which forces the UD filter to be updated in a time not longer than 10 ms. Moreover, since an INS is just one part of the guidance, navigation, and control system, other functions than the UD filter must be executed. In summary, both the required amount of floating-point operations and real-time constraints could be overwhelming for a typical embedded microprocessor, that usually works at frequencies between 200 and 600 MHz, even with floating-point hardware support.

An FPGA (Field Programmable Gate Arrays) device appears as an appropriate platform for solving the computation

of the UD filter. When the UD filter is calculated by an FPGA, the microprocessor can focus on processing less demanding algorithms. Furthermore, FPGA's particular architecture can exploit some parallelism of the UD filter. Pipelining can also be used for throughput improvement.

In the revised literature, some relevant papers tackle the synthesis in hardware of a classical KF with floating-point precision. In [7], an EKF based on traditional KF equations is used for localization and mapping of autonomous mobile robots. In [8], an alternative form of the KF is implemented in FPGA for target tracking. The work from [9] shows a sequential EKF for robot localization in Cartesian coordinates. Lastly, [10] exhibits an FPGA implementation of a regular KF for detecting discontinuities in TIG welding processes. As shown, none of these works takes into account some numerically-stable version of the KF. However, in [11] a UD filter is synthesized in FPGA but solely with fixed-point precision and up to 6-states dimension.

In this work, a 21-states UD filter implementation fully in hardware with single-precision, floating-point arithmetic is reported. The UD filter operates as a coprocessor to work in conjunction with a microprocessor within a system-on-chip. The hardware realization of the UD filter is designed to be used as part of a battery-powered INS, therefore, energy consumption is critical. Since energy is power times time, a drastic reduction in the execution time may provide energy savings. As a result, a thoroughly software/hardware exploration is done to find the filter realization with best throughput. For this purpose, two structures of the filter are taken into account: the classical version of the UD filter, and a hardware-oriented reformulation of the UD filter algorithm intended to exploit parallelism and pipelining.

The rest of this paper is organized as follows. Section II explains in detail the hardware/software co-design method. Section III provides information about the UD filter coprocessor, as area, execution time, power, and energy usages. Finally, in section IV concluding remarks are commented.

II. HARDWARE/SOFTWARE CODESIGN

The hardware/software analysis accomplished in this section is based upon to contrast the efficiencies of the classical, software-oriented UD filter [12, Appendices V.A and VI.A] and the hardware version exposed in [13]. In addition, a third version is evaluated, which arises by synthesizing in hardware the software version. The goal of this hardware/software exploration is to find the UD filter arrangement with best throughput.

A. Methodology

The hardware/software co-design studies the convenience of dividing an embedded application in a flexible part (software) and, in principle, a fixed part (hardware). This method explores the trade-off between both approaches in order to find the best realization according to some specific restrictions, as throughput, latency, or energy consumption. An algorithm purely implemented in software offers rapid deployment, but its performance could not be satisfactory for a particular application. On the other hand, if the same algorithm is executed in hardware, performance could be improved at the expense of employing more time in the development cycle.

The UD filter programs optimized for software, *bierman* and *thornton*, are exhibited in [1, Table 6.15, p. 270, and Table 6.16, p. 274], in MATLAB language. *bierman* is in charge of the measurement update, and *thornton* executes the temporal update. This sequential version of the UD filter is targeted for execution on general purpose microprocessors. It operates consecutively only on non-zero entries of matrices U and D , in an element-by-element fashion. This approach saves computation time in a microprocessor and, therefore, filter execution is accelerated. On the other hand, this strategy may not be adequate for hardware realization, since parallelism is not considered, a key advantage of reconfigurable logic implementation.

In contrast, UD filter programs restructured for hardware execution, *bierman_hw* and *thornton_hw*, are presented in [13, Fig. 1 and 2], also in MATLAB language. This strategy attempts to operate on columns and rows of matrices U and D , instead of on individual entries. Therefore, this methodology exploits the UD filter parallelism in a clever manner. For example, intermediate values of the algorithm are calculated at once and saved on temporal arrays that later will be used to update U and D . This strategy will lead to loops with homogeneous parallel structures and efficient pipelines.

In summary, two versions of the UD filter are available for hardware/software co-design. The key parameter of this exploration is throughput. Thus, the best solution has to show the shortest possible execution time.

New development tools for FPGA are an optimal framework for hardware/software algorithm partition. Vivado High-Level Synthesis (HLS) [14] is an IDE (integrated development environment) from FPGA vendor Xilinx that offers the capability of designing and testing a digital system using C/C++ languages. In this way, if a particular algorithm already written in C/C++ does not meet an expected throughput when running in a microprocessor, it can be moved to hardware without rewriting the code into a register-transfer level (RTL) description language, such as VHDL or Verilog, so as to accelerate its execution. Vivado HLS provides directives or *pragmas* to modify the resulting hardware design from the source code. Hence, the FPGA designer can adjust the RTL code outcome until the solution reaches application restrictions on area and/or throughput.

Together with new computer-aided design applications, FPGA vendors are releasing novel chipsets that ease this hardware/software partition methodology. These platforms are advanced hybrid reconfigurable devices that combine the software programmability of general purpose microprocessors with the hardware reconfigurability of FPGA.

Firstly, programs from classical and hardware-oriented versions are translated from MATLAB to C language. New *bierman* and *thornton* programs are compiled to run on a microprocessor. On the other hand, *bierman_hw* and *thornton_hw* programs are synthesized in FPGA by using Vivado HLS. Furthermore, UD filter programs for software are also synthesized in hardware, which is straightforward by using a C-to-RTL tool. In all hardware systems, an intensive use of *pragmas* is carried out to get the best possible throughput, using the maximum available hardware resources.

As a result, three solutions are proposed for the realization

of the UD filter algorithm: 1) the pure software solution (SW), 2) the hardware solution (HW), and 3) a hybrid solution (SW-HW) which results from synthesizing the software programs in reconfigurable logic.

B. Time measurements

A Zedboard development kit [15] is used to evaluate the software and hardware solutions. Zedboard's central processing unit is a Xilinx Zynq Z-7020 [16]. This is a system-on-chip that includes a dual-core ARM Cortex A-9 microprocessor and an Artix-7 FPGA [17]. This hybrid architecture enables an efficient hardware/software partition.

The ARM Cortex A-9 has an FPU (floating-point unit) to support floating-point mathematics operations in hardware. Besides, its NEON extension allows to execute 128-bits SIMD (*Single Instruction, Multiple Data*) instructions. According to ARM, both systems provide a performance boost of $\times 4$ to $\times 8$ for simple, floating-point DSP algorithms [18]. As a result, this ARM microprocessor can be considered as a high-end CPU for average embedded applications.

The precision of all involved variables for the three solutions, i.e. input, output, as well as intermediate calculations, are single-precision floating point (32 bits).

The software solution is compiled with full optimization (-O3 level), in order to make an efficient use of the CPU architecture provided by the ARM Cortex A-9. Executable code and data are load in the DDR3 external memory, which is set to work at 400 MHz.

Times from the two hardware solutions (HW and SW-HW) are taken from post-synthesis simulation. Times from software programs are measured using an independent timer in hardware (see Fig. 1). It is worth mentioning that the ARM clock runs at 400 MHz while the FPGA clock is set to 100 MHz. Both are typical frequency values for average embedded applications.

Table I resumes the execution times in microseconds (μs) for the three proposed UD filter solutions. Vivado Design Suite 2014.2 [19] is used to develop hardware and software solutions.

TABLE I: Execution times for the three proposed UD filter implementations.

	Freq. MHz	bierman μs	thornton μs	Total μs
SW	@400	160.55	529.77	690.32
HW	@100	289.59	493.33	782.92
SW-HW	@100	151.37	1108.47	1259.84
Shortest time	—	151.37	493.33	644.70

As can be seen in Table I, the UD measurement update (*bierman*) that presents the shortest execution time is interestingly the SW-HW solution, with 151.37 μs . It is a 6% faster than the SW *bierman* program and almost 2 times faster than the HW *bierman* system.

On the other hand, the UD temporal update (*thornton*) for the HW solution exhibits the shortest execution time with

493.33 μs . It is a 7% faster than the SW *thornton* program and two times faster than the SW-HW *thornton* system.

C. Result analysis

Several conclusions can be taken from looking at values from Table I. In first place, as expected, a solution entirely in hardware is the fastest one. This comes from synthesizing in the FPGA the SW-HW *bierman* and the HW *thornton* programs. The resulting UD filter realization takes 644.70 μs to execute.

Second, if total times for each of the three solutions are compared separately, the SW solution seemingly appears as the best solution. Prematurely, one may choose incorrectly this solution as the fastest for the UD filter update. Analysing each update program separately, it is concluded that the best approach is a mixing of SW-HW and HW subsystems. Therefore, it is important to make a detailed analysis of the execution times of both update algorithms in a separate way, so that finding the filter UD arrangement with best performance.

Third, it is found that *bierman_hw* does not present better throughput than the rest of the programs under analysis, as expected. A detailed inspection of *bierman_hw*'s code reveals that the bottleneck of this program is the UD factorization part [13, Fig. 1, lines 36 to 55]. This portion of *bierman_hw* keeps operating on not-null elements of matrices U and D , one entry at a time, just as the software version does. Since a strong dependency exists between previous calculated values and the present ones, parallelization of this UD factorization formulation is infeasible.

Fourth, the hardware/software exploration performed in this section gives a new perspective when attempting to run certain algorithm in hardware. In general, algorithms originally written for software execution are redesign for hardware implementation, and its code is translated to an RTL language. Tools like Vivado HLS allow to put directly in hardware the software version of an algorithm, and then compare throughputs between software and the hardware-redesign versions, both running in reconfigurable logic.

Lastly, the three systems fully comply with the 10 ms restriction as a maximum time to update the UD filter (Sec. I). In fact, the UD filter with the highest throughput (SW-HW *bierman* + HW *thornton*) can work with inertial sensors that deliver measurements up to 1,550 samples per second.

III. UD FILTER IN FPGA

This section provides details about the UD filter realization in FPGA. The UD filter procedure is updated by a coprocessor with single-precision floating-point resolution. Such coprocessor, hereafter UD coprocessor (*coP UD*), is part of a system-on-chip where the main control unit is a microprocessor.

A. Coprocessor design

New heterogeneous computing architectures, such as Xilinx Zynq-7020 [16], are appropriate platforms to run the UD coprocessor in FPGA. The Zynq-7020 provides in a single integrated circuit a microprocessor and a FPGA, both made with the same 28 nm technology and communicated through

standard high-speed buses. Hence, demanding computing algorithms can be executed on reconfigurable logic, freeing the microprocessor to run less demanding tasks.

Figure 1 shows synthesized and hard-coded components and the type of connectivity between them. A timer is included in the FPGA system in order to have an independent and precise time measurement unit.

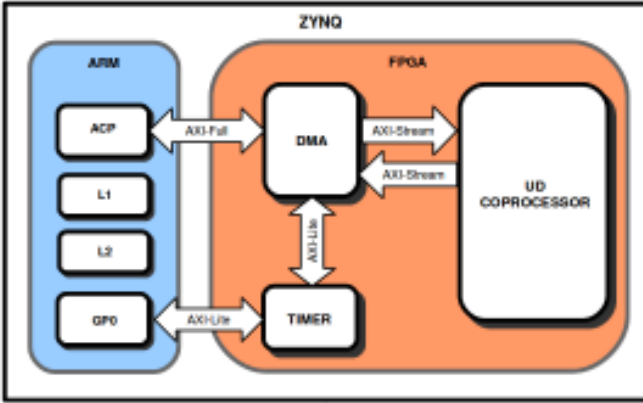


Fig. 1: Diagram of the synthesized embedded system.

As seen in Figure 1, the UD coprocessor has an AXI-Stream interface that connects to the ACP (Accelerator Coherency Port) interface, which is a 64-bit, asynchronous AXI slave port, part of the SCU (Snoop Control Unit). In the middle of both devices, a DMA (Direct Memory Access) core is instantiated with AXI interfaces. The DMA core allows the UD coprocessor directly access both the external memory DDR3 and the microprocessor L1 and L2 caches. Thus, microprocessor and coprocessor share data through all memory hierarchy, where caches have lower latency communication at the expense of having less capacity. All read transactions through the ACP interact with the SCU that verifies that the required information is stored in the L1 cache. Otherwise, SCU checks the L2 cache before forwarding the request to the main memory. Given a writing transaction, the SCU ensures that the datum is consistent within the entire memory hierarchy.

The UD coprocessor is controlled from a software application running in the microprocessor. Transferred data to the coprocessor are matrices A and G , and vector y . The UD coprocessor returns vector x_o [13, see Fig. 3]. Arrays R , Q , $U_{pr}(1)$, $d_{pr}(1)$, and $x_i = 0$ are hard-coded at C-to-RTL translation time.

Table II shows the hardware resources needed to implement the UD filter for a 21-states system (Eq. 1) in a Zynq-7020. UD coprocessor is designed with Vivado Design Suite 2014.2. Since no previous work is found in the literature about a FPGA-based UD filter, it is not possible to make a fair comparison with another implementation. Despite that, hardware resource usage from [11, Table 1] are included in Table II in order to have some reference point. It must be bear in mind that values from the first column of Table II correspond to a 12-bit fixed-point UD filter with 6 states, running in a high-range FPGA, a Xilinx Virtex-4 (XC4VFX140). Although the number of slices is only reported in [11], LUT value shown in Table II is a good approximation.

TABLE II: UD coprocessor FPGA area usage.

	Fixed-point UD filter	coP UD	Zynq-7020 resources	Utilization (%)
BRAM	–	52	280	37
DSP48E	192	171	220	77
FF	–	26,882	106,400	25
LUT	~74,018	26,193	53,200	53

Resources usage is managed through the directives provided by Vivado HLS [14]. The most used ones in the development of the UD coprocessor are UNROLL, ARRAY_PARTITION, and PIPELINE. The UNROLL directive is used to divide and parallelize certain for-loops by a fixed quantity. In Table II, it is observed a heavy use of DSP48E blocks with respect to the Zynq-7020 hardware availability. This is a deliberate design policy that is accomplished by using efficiently the UNROLL directive. The ARRAY_PARTITION directive divides a single BRAM block in several blocks to effectively increase the number of ports and, as a consequence, the memory throughput as well. Finally, the PIPELINE directive instructs the compiler to pipeline a particular for-loop. It is used by default in every for-loop in the design of the UD coprocessor.

When analysing Table II, it can be said that the UD coprocessor is a suitable IP core for midrange and onward FPGA.

Figure 2 shows the floorplan image of the synthesized system in a Zynq-7020. At first glance, it can be seen that the UD coprocessor covers approximately half of the FPGA area. It is worth pointing out that columns in green are BRAM and DSP48E blocks used by the coprocessor. In addition, it is exhibited that the DMA core takes a notorious portion of the FPGA.

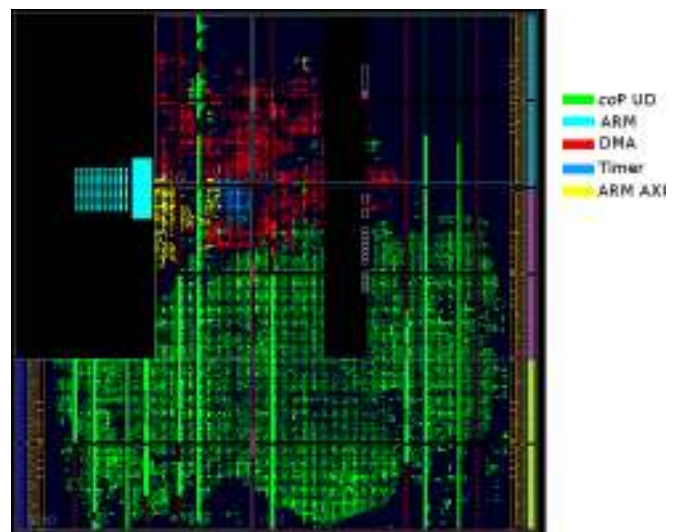


Fig. 2: Floorplan of the synthesized embedded system.

B. Power and energy utilizations

Nowadays, power and energy consumptions are key parameters in embedded systems design. Indeed, since the UD coprocessor is meant to be used in an INS for autonomous robotics, it is vital to have this values at hand to evaluate its feasibility.

Measurement of the average power consumption is made via a $10\text{ m}\Omega$ shunt resistor that is placed in the 12 V main power line of the Zedboard. Since other peripherals are powered from this line, such as audio and video chipsets, additional power consumptions are expected.

Firstly, the voltage drop across the shunt resistor V_{shunt} is measured by using a true-RMS multimeter Fluke 287. Such measurement is performed running the UD coprocessor continuously into an infinite loop. Next, the current I_{shunt} flowing through the shunt resistor is calculated by the Ohm's law. The total average power P is obtained by multiplying V_{shunt} times I_{shunt} . Finally, the average energy consumption E is calculated by multiplying the total average power times the UD coprocessor's runtime t_E . The value of t_E is obtained through the external timer (Figure 1).

Table III summarizes the measurements of power and energy for the complete embedded system, with ARM running at 400 MHz , DDR at 400 MHz , and FPGA at 100 MHz .

TABLE III: Power and energy consumptions for the embedded system

V_{shunt} (mV)	I_{shunt} (mA)	V_{12V} (V)	P (W)	t_E (μs)	E (mJ)
3.355	335.50	12.208	3.750	660.47	2.48

As shown in Table III, the energy utilization of the embedded system is 2.48 mJ , which is a suitable value for a system targeted towards being used in battery-powered autonomous robots. On the other hand, around 4 W is an affordable power budget. It is noteworthy that the UD coprocessor runtime is slight longer than the one shown in Table I. The reason is that the time from Table III includes data transfers latencies from the microprocessor to the FPGA, and vice versa.

C. Validation process

Regarding the validation of the UD coprocessor, an INS simulation environment developed in MATLAB is used [20]. A field data set provided by [21] is used to stimulate the simulation framework. It is composed by the following sensors: Crossbow IMU400CD, Novatel GPS, navigation-grade Honeywell IMU H764G-1 and differential GPS. The two latter are fused to get a reference data set.

Figure 3 displays the trajectory of the vehicle from [21]. It takes place in the vicinity of the Ohio University, USA. The globe on the left with a circle marks the beginning of the trajectory. On the other hand, the globe on the right with a square points the end. The path covers 14.67 minutes throughout which the availability of GPS signal is 100% .

The UD coprocessor is validated as follows. Ten sets of data are generated with the INS simulator. Every 100 iterations

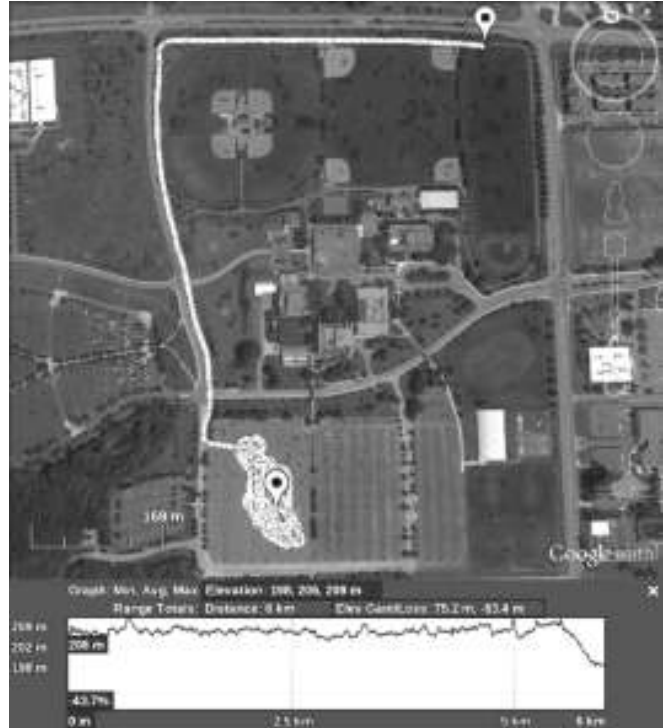


Fig. 3: Reference trajectory from [21] (image courtesy of Google Earth [22]).

from the simulated INS, a single-precision, floating-point data set is saved. Each data set consists of input arrays to the UD filter [13, see Fig. 3] and respective output vectors \mathbf{x}_o .

A testing program is developed in C to be execute in the ARM microprocessor. Input and output arrays are set at compilation time. Then, the testing routine sends the input data to the UD coprocessor through the AXI interface. Lastly, the UD coprocessor returns the output vector \mathbf{x}_o , which is compared with the corresponding output vector from the INS simulator.

As a result, negligible differences of less than $1e-5$ for the ten data sets are observed between the output vectors from the UD coprocessor and the simulation environment. In summary, it is confirmed that the UD coprocessor works properly.

IV. CONCLUSIONS

In this work, the design, implementation and validation of a FPGA-based UD filter coprocessor is exposed.

An exhaustive hardware/software exploration is performed in order to find the system that presents the shortest execution time. It is found that such a system is a combination of the classical measurement update (*bierman*) and the hardware-targeted temporal update (*thornton*), both running in reconfigurable logic.

Additionally, it is exposed the convenience of using new C-to-RTL development tools for synthesizing in hardware algorithms that originally were designed to run on general purpose microprocessors.

The UD filter is synthesized in FPGA as a coprocessor for a 21-states system, with single-precision floating-point resolution. This coprocessor is suitable for being used in mid-range and onward FPGA. It can operate in autonomous robots whose dynamics require up to 1,550 samples per second from the inertial sensors. Moreover, the embedded system presents adequate power and energy consumptions for a battery-powered INS.

To conclude, it is demonstrated that new hybrid FPGA are suitable devices to put in practice battery-operated navigation systems for autonomous robots.

ACKNOWLEDGEMENT

The authors would like to thank to the Xilinx University Program for granting software licenses to use Vivado Design Suite 2014.2.

REFERENCES

- [1] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*, 3rd Ed. John Wiley & Sons, Inc., 2008.
- [2] G. J. Bierman, "Measurement updating using the U-D factorization," *Automatica*, vol. 12, no. 4, pp. 375–382, 1976.
- [3] C. L. Thornton and G. J. Bierman, "Gram-Schmidt algorithms for covariance propagation," *International Journal of Control*, vol. 25, no. 2, pp. 243–260, 1977.
- [4] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. John Wiley & Sons, Inc., 2006.
- [5] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2008.
- [6] J. A. Rios and E. White, "Fusion filter algorithm enhancements for a MEMS GPS/IMU," in *Proceedings of the 2002 National Technical Meeting of The Institute of Navigation*, San Diego, CA, January 2002.
- [7] V. Bonato, E. Marques, and G. Constantinides, "A floating-point extended Kalman filter implementation for autonomous mobile robots," in *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on*, August 2007, pp. 576–579.
- [8] P.-L. Wu, B.-B. Wang, and C.-H. Ji, "Design and realization of short range defence radar target tracking system based on DSP/FPGA," *WSEAS Transactions on Systems*, vol. 10, no. 11, pp. 376–386, 2011.
- [9] S. Cruz, D. Muñoz, M. Conde, C. Llanos, and G. Borges, "FPGA implementation of a sequential extended Kalman filter algorithm applied to mobile robotics localization problem," in *Circuits and Systems (LASCAS), 2013 IEEE Fourth Latin American Symposium on*, February 2013.
- [10] R. Hurtado, S. C. A. Alfaro, and C. Llanos, "A methodology for 'on-line' monitoring system in a welding process using FPGAs," in *Industrial Technology (ICIT), 2010 IEEE International Conference on*, March 2010, pp. 162–167.
- [11] Y. Liu, C. Bouganis, and P. Y. K. Cheung, "Efficient mapping of a Kalman filter into an FPGA using Taylor expansion," in *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on*, August 2007, pp. 345–350.
- [12] G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation*. Academic Press, 1977.
- [13] R. Gonzalez, G. Sutter, and H. D. Patiño, "Optimized UD filtering algorithm for floating-point hardware execution," in *Information Fusion (FUSION), 2014 17th International Conference on*, July 2014.
- [14] Xilinx, Inc., *Introduction to FPGA Design with Vivado High-Level Synthesis, UG998 v1.0*, July 2013.
- [15] Digilent, Inc., *ZedBoard (Zynq Evaluation and Development) Hardware user's guide*, version 2.2, January 2014.
- [16] Xilinx, Inc., *Zynq-7000 All Programmable SoC Overview*, DS190 (v1.7), October 2014.
- [17] Xilinx Inc., *7 Series FPGAs Overview*, DS180 (v1.16), October 2014.
- [18] ARM website. NEON and VFP9-S. 2014. Available online: <http://www.arm.com/products/processors/technologies/neon.php>, <http://www.arm.com/products/processors/technologies/vector-floating-point.php>.
- [19] Xilinx, Inc., *Vivado Design Suite User Guide*, ug910, April 2014.
- [20] R. Gonzalez, J. I. Giribet, and H. D. Patiño, "NaveGo: a simulation framework for low-cost integrated navigation systems," *Journal of Control Engineering and Applied Informatics*, vol. 17, issue 2, pp. 110-120, 2015.
- [21] C. Toth, D. Brzezinska, N. Politi, and A. Kealy, "Reference data set for performance evaluation of MEMS-based integrated navigation solutions," in *FIG Working Week 2011, Marrakech, Morocco*, May 2011.
- [22] Google Earth, "The Ohio State University, USA. 40.001319N, 83.039045W. Date of original imagery: 5/29/2010. Date accessed: 4/17/2014." 2014.

MESA: A Formal Approach to Compute Consensus in WSNs

Zacchigna, Federico G.
and Lutenberg, Ariel

Laboratorio de Sistemas Embebidos
Facultad de Ingeniería, Universidad de Buenos Aires
Paseo Colón 850, Ciudad Autónoma de Buenos Aires
federico.zacchigna@gmail.com, lse@fi.uba.ar

Vargas, Fabian
SiSC Group

Electrical Engineering Department
Catholic University (PUCRS)
Porto Alegre, Brasil
vargas@computer.org

Abstract—Consensus algorithms are implemented over Wireless Sensor Networks (WSN) for computing a distributed variable measurement or performing Data Fusion (DF). In a previous work the main idea of the MESA algorithm presented, but with no mathematical formalization. The MESA approach is a novel consensus algorithm that implements the transmission censoring technique aiming to reduce the energy used for reaching consensus. Moreover, it makes use of neighborhood data, which allows it to perform estimations over unknown distributed variables. This condition renders the proposed approach more efficient than the existing censoring techniques found in the literature.

In this article a formalization of the MESA algorithm over cluster based WSN is presented. It is addressed from the statistical point of view of the algorithm, which gives us a better understanding of the consequences of transmission censoring and leads us to predict the results of its utilization.

I. INTRODUCTION

This work is focused on Wireless Sensor Networks (WSN) with a Cluster Head (CH) and N nodes around it. Each node performs a measurement and a Consensus Value (CV) is achieved after that:

$$x_c = f(\mathbf{X}) \quad (1)$$

where $\mathbf{X} = [X_1 X_2 \dots X_N]^T$, X_i is the i 's node measurement and x_c is the CV. This is called Data Fusion (DF).

In this work it will be presented the statistical model that describes this process when the censoring technique is used. Section I-A will explain in more detail the measurement process and the differences between performing an estimation or a detection, and its difficulties. In section II the theoretical derivation of the algorithm is performed. The simulations, their results and the comparison with the theoretic derivation is shown in section III. Finally, conclusions are stated in section IV.

A. The measurement

The consensus process starts with the measurement of a set of initial values named initial vector or initial state. Though it is common to talk about a measurements, this is not necessarily precise. The initial vector can be obtained by measuring an environment variable with an actual sensor, for example in cognitive radios, where cooperative spectrum sensing is performed to detect the presence of a carrier by

measuring its power [3] [2], but these initial values can also be obtained, for example as a node internal variable that can represent the remaining energy stored in its battery [1]. This allows the node to discover if its battery level is above or below the average value of the other nodes in the network. Such information is of prime importance, if one desires to extend the network lifetime as a whole, as long as possible.

We will consider each of the measurements a Gaussian distributed Random Variable (RV) with mean μ and variance σ^2 . The Probability Density Function (PDF) of them may have the following characteristic:

$$X_i \stackrel{\mathcal{H}_h}{\sim} \mathcal{N}(\mu_h, \sigma^2), \text{ with } h \in \{0, \dots, h_{max}\} \quad (2)$$

or

$$X_i \sim \mathcal{N}(\mu, \sigma^2), \text{ with } \mu \in \mathbb{R} \quad (3)$$

for a detection or a estimation scheme respectively.

B. Different cases

Different schemes are shown in Table I. Cases I and IV are the standard consensus algorithms for clustered and distributed WSN respectively. Case II is the one studied in Rago's work [4]. In this article the analysis will be focused on cases I to III.

Case I provides no innovation at all, but it is derived to be used as a benchmark in this work. Case II is also not innovative, but in this work it is addressed in a different way, not from the likelihood ratio test as done by Rago [4], but from its statistics. Case III is the one that provides a new way of facing the consensus problem by using neighborhood data.

II. THEORETICAL ANALYSIS

A. Case I

Consider a WSN with a CH and N nodes, that performs an initial measurement and $\mathbf{X} = [X_1, X_2, \dots, X_N]^T$ is obtained, with $X_i \sim \mathcal{N}(\mu, \sigma^2) \forall i \in \{1, \dots, N\}$ and X_i, X_j are Independent Identically Distributed (IID) RVs $\forall i \neq j$.

Every node transmits its value to the CH, where the DF is performed as

$$X_{CH_I} = \sum_{i=1}^N w_i X_i = \mathbf{w}^T \mathbf{X} \quad (4)$$

TABLE I
 CASES CLASSIFICATION ACCORDING TO THE CENSORING TECHNIQUE. *knw*, *unkwn*, *ngbhd* AND *N/A*, ARE USED FOR *known*, *unknown*, *neighborhood* AND *Not Applicable* RESPECTIVELY.

Case	WNS type	Consensus strategy	Distribution type	Data usage	Estimation Detection
I	Cluster	standard	kwn/unkwn	N/A	Both
II		censoring	kwn	local	Detection
III		censoring	kwn/unkwn	ngbhd	Both
IV	Distributed	standard	kwn/unkwn	N/A	Both
V		censoring	kwn	local	Detection
VI		censoring	kwn/unkwn	ngbhd	Both

X_{CH_I} is the new random variable in the CH and is obtained as a Linear Combination (LC) of the \mathbf{X} 's elements. In this case the the average consensus is considered, which means that $w_i = 1/N \forall i$ and it is easy to demonstrate that

$$X_{CH_I} \sim \mathcal{N}\left(\mu, \frac{\sigma^2}{N}\right) \quad (5)$$

This consensus scheme is useful for performing estimations or detections.

B. Case II

As well as in case I, in case II the initial values are measured and \mathbf{X} is obtained, again with $X_i, X_j \in \text{IID RV } \forall i \neq j$ and X_i distributed as in (2) with $h \in \{h_0, h_1\}$. \hat{X}_{II_i} is defined as

$$\hat{X}_{II_i} = \begin{cases} X_i & \text{if } x_i \in R_i \\ X_{II_i}^{est_{\mathcal{H}}} & \text{if } x_i \in \bar{R}_i \end{cases} \quad (6)$$

where R_i and \bar{R}_i are two disjoint regions, in which the data is sent and censored respectively. Both are define by th_L and th_H , the two limits between them

$$R_i = \{x | th_L \leq x < th_H\} \quad (7)$$

$$\bar{R}_i = \{x | x < th_L \vee th_H \leq x\} \quad (8)$$

In (6) $X_{II_i}^{est_{\mathcal{H}}}$ is the estimated value, when no transmission is received and it differs according to the hypothesis \mathcal{H} taken. This estimation can be performed in several ways. As follows two examples are shown:

- 1) $X_{II_i}^{est_{\mathcal{H}_0}} = \mathbb{E}[X_i | x_i \in \bar{R}_i, \mathcal{H}_0]$: This is an unbiased estimator for \mathcal{H}_0 , but biased for \mathcal{H}_1 . By exchanging \mathcal{H}_0 y \mathcal{H}_1 and adjusting th_L and th_H accordingly, the same scheme is reached.
- 2) $X_{II_i}^{est_{\mathcal{H}_0}} = \mathbb{E}[X_i | x_i \in \bar{R}_i, \mathcal{H}_0] \mathbf{P}(\mathcal{H}_0) + \mathbb{E}[X_i | x_i \in \bar{R}_i, \mathcal{H}_1] \mathbf{P}(\mathcal{H}_1)$: This estimator is biased in both cases (but less biased than in 1). Again \mathcal{H}_0 and \mathcal{H}_1 may be exchanged for getting the symmetric case.

Fig. 1 shows a simple example of the distribution X_i y \hat{X}_{II_i} .

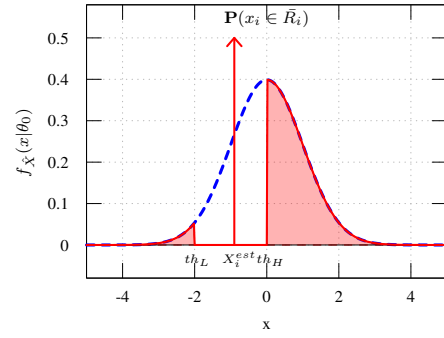


Fig. 1. Example of the \hat{X}_{II_i} PDF. In this case $th_L = -2$, $th_H = 0$, $\mu = 0$ y $\sigma = 1$.

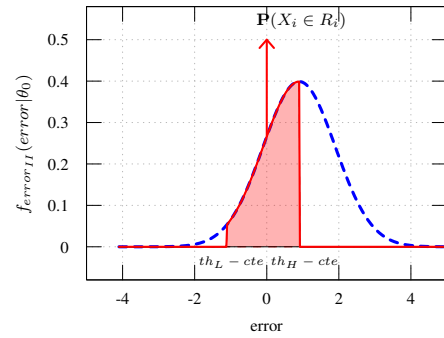


Fig. 2. Example of the $error_{II_i}$ PDF. In this case $th_L = -2 - cte$, $th_H = 0 - cte$, $\mu = 0$, $\sigma = 1$ and $cte = \mathbb{E}(x | th_L < x < th_H)$.

1) Considering the \mathcal{H}_0 hypothesis: By taking the first estimator previously explained ($X_{II_i}^{est_{\mathcal{H}_0}} = \mathbb{E}[X_i | x_i \in \bar{R}_i, \mathcal{H}_0]$) we may say

$$\mu_{\hat{X}_{II_i}} = \mu \quad (9)$$

The CH receives the i 's \hat{X}_{II_i} RV and performs de DF as

$$\hat{X}_{CH_{II}} = \sum_{i=1}^N \frac{\hat{X}_{II_i}}{N} = \frac{1}{N} \mathbf{1}^T \hat{\mathbf{X}}_{II} \quad (10)$$

The \hat{X}_{II_i} RV may be expressed as $\hat{X}_{II_i} = X_i + error_{II_i}$, which lead to

$$\hat{X}_{CH_{II}} = \sum_{i=1}^N \frac{X_i}{N} + \sum_{i=1}^N \frac{error_{II_i}}{N} = X_{CH_I} + error_{II} \quad (11)$$

where X_{CH_I} is the RV obtained in case I and $error_{II} = \sum_{i=1}^N \frac{error_{II_i}}{N}$. The $error_{II_i}$ RV PDF is shown in Fig. 2.

By applying the Central Limit Theorem (CLT), we may say that

$$\hat{X}_{CH_{II}} \overset{approx.}{\sim} \mathcal{N}\left(\mu_{\hat{X}_{CH_{II}}}, \sigma_{\hat{X}_{CH_{II}}}^2\right) \quad (12)$$

or that

$$error_{II} \overset{approx.}{\sim} \mathcal{N}\left(\mu_{error_{II}}, \sigma_{error_{II}}^2\right) \quad (13)$$

We may now say that

$$error_{II_i} = \hat{X}_{II_i} - X_i \quad (14)$$

and

$$error_{II} = \hat{X}_{CH_{II}} - X_{CH_I} \quad (15)$$

both RV have zero mean because $\mu_{\hat{X}_{CH_{II}}} = \mu_{X_{CH_I}} = \mu$, and by using CLT and considering that N is big enough:

$$\sigma_{\hat{X}_{CH_{II}}}^2 \approx \frac{\sigma_{\hat{X}_{II_i}}^2}{N} \quad (16)$$

and

$$\sigma_{error}^2 \approx \frac{\sigma_{error_{II_i}}^2}{N} \quad (17)$$

The variance of the RVs are related by their correlation $\rho_{\hat{X}_{II_i} X_i}$ and $\rho_{X_i error_{II_i}}$ respectively.

$$\sigma_{error_{II}}^2 = \frac{\sigma_{\hat{X}_{II_i}}^2 + \sigma_{X_i}^2 - 2\rho_{\hat{X}_{II_i} X_i} \sigma_{\hat{X}_{II_i}} \sigma_{X_i}}{N} \quad (18)$$

$$\sigma_{\hat{X}_{CH_{II}}}^2 = \frac{\sigma_{\hat{X}_{II_i}}^2 + \sigma_{error_{II_i}}^2 + 2\rho_{X_i error_{II_i}} \sigma_{X_i} \sigma_{error_{II_i}}}{N} \quad (19)$$

the correlation can be calculated if it is desired and it is easy to see that $\rho_{\hat{X}_{II_i} X_i} \geq 0$ and $\rho_{\hat{X}_{II_i} error_{II_i}} \leq 0$.

2) *Considering the \mathcal{H}_1 hypothesis:* Again taking the first estimator named before ($X_{II_i}^{est\mathcal{H}_1} = \mathbb{E}[X_i | x_i \in \bar{R}_i, \mathcal{H}_1]$) the error is

$$error_{II_i} = \hat{X}_{II_i} - X_i \quad (20)$$

and

$$error_{II} = \hat{X}_{CH_{II}} - X_{CH_I} \quad (21)$$

which now does not have a zero mean, because $\mu_{\hat{X}_{II_i}} = \mu + bias_{\mathcal{H}_1}$ and $bias_{\mathcal{H}_1} \neq 0$, so the RV means are

$$\mu_{\hat{X}_{CH_{II}}} = \mu + bias_{\mathcal{H}_1} \quad (22)$$

$$\mu_{error_{II}} = bias_{\mathcal{H}_1} \quad (23)$$

with variances

$$\sigma_{\hat{X}_{CH_{II}}}^2 \approx \frac{\sigma_{\hat{X}_{II_i}}^2}{N} \quad (24)$$

$$\sigma_{error_{II}}^2 \approx \frac{\sigma_{error_{II_i}}^2}{N} \quad (25)$$

respectively.

It worth to remark that the PDFs for $\hat{X}_{CH_{II}}$ and $error_{II}$ differ according to the \mathcal{H} considered and of course this leads to different $\sigma_{\hat{X}_{CH_{II}}}^2$ and $\sigma_{error_{II}}^2$.

3) *Conclusions for the case II:* The efficiency of the algorithm can be measured by means of the metrics presented in II-E and hangs on the estimator used when the transmissions are censored:

- Unbiased for \mathcal{H}_0 and biased for \mathcal{H}_1 .
- Unbiased for \mathcal{H}_1 and biased for \mathcal{H}_0 .
- Both biased but less than in the two previous cases.

The bias is a function of $\mathbf{P}(\mathcal{H}_0)$ and $\mathbf{P}(\mathcal{H}_1)$ and \bar{R}_i region should be on the side of the most probable hypothesis, so that the bias obtained will be smaller with the same amount of censored transmissions.

In [4] it is proved that for minimizing the *miss* or the *false detection* probabilities in the detection, th_L should be equal to $-\infty$, but should be different if we want to minimize another metric.

4) *Problems to be solved in case II:* In this section some useful ways of using the censoring technique are enumerated now:

- Given:
 - a transmission constraint, that is equivalent to an area limitation:

- * $th_c \geq \mathbf{P}(x_i \in \bar{R}_i | \mathcal{H}_0)$, or
- * $th_c \geq \mathbf{P}(x_i \in \bar{R}_i | \mathcal{H}_1)$, or
- * $th_c \geq \mathbf{P}(x_i \in \bar{R}_i | \mathcal{H}_1) + \mathbf{P}(x_i \in \bar{R}_i | \mathcal{H}_0)$.

where th_c is the threshold value of the constraint.

- μ and σ for \mathcal{H}_0 and \mathcal{H}_1 .

find the th_L y th_H , so that one of the metrics named in section II-E is minimized. Equivalent to find the optimum DR.

- Given:
 - a constraint in one of the metrics in section II-E.
 - μ and σ for \mathcal{H}_0 and \mathcal{H}_1 .
 find the DR (th_L and th_H) so that the number of censored transmissions is maximized:
 - $\mathbf{P}(x_i \in \bar{R}_i | \mathcal{H}_0)$, or
 - $\mathbf{P}(x_i \in \bar{R}_i | \mathcal{H}_1)$, or
 - $\mathbf{P}(x_i \in \bar{R}_i | \mathcal{H}_0) + \mathbf{P}(x_i \in \bar{R}_i | \mathcal{H}_1)$

C. Case III

As well as in case I and II the initial values are measured. The X_i s RV may be distributed as in (2) or (3). The initial vector \mathbf{X} is obtained, where X_i, X_j are IID RV $\forall i \neq j$.

In this work the following assumptions and restrictions are considered for limiting the scope and the length of this article.

- The nodes are synchronized and they transmit in order, one node per Time Slot (TS) as in [5]. In each realization:
 - The first node is randomly selected.
 - In the next TS the corresponding node decides whether to transmit or to censor the data according to the DR.
 - The realization ends when the N nodes had their TS, including the first one.
- The nodes are able to hear the data transmitted only from the previous node.

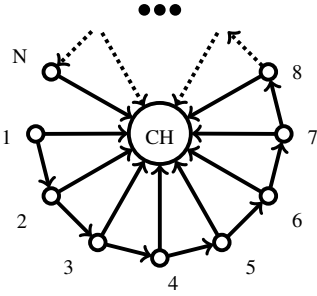


Fig. 3. This figures shows the CH and the N nodes around it and how they communicate. The i is able to send data to the CH and to hear data from the $i - 1$ node.

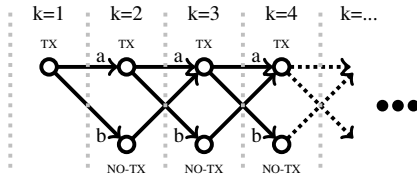


Fig. 4. Graph representing the algorithm and the transmission probability in each TS.

TABLE II
PROBABILITY IN EACH TS

Turn (k)	1	2	3	4	5	...
$\mathbf{P}(\text{TX}_k)$	1	a	$a^2 + b$	$a^3 + 2ab$	$a^4 + 3a^2 + b^2$...
$\mathbf{P}(\text{NO-TX}_k)$	0	b	ab	$a^2b + b^2$	$a^3b + 2ab^2$...

Fig. 3 shows the corresponding communication diagram used in this section. In section II-D other interesting variations to be considered are named.

The algorithm may be resumed in the following way:

- Node in the first TS sends its data.
- The k node sends it if the $k - 1$ has not send its data.
- The k node sends it if the $k - 1$ has send its data and the DR in k is fulfilled.
- The k node does not sends its data if $k - 1$ sent it and the DR in k is not fulfilled.

by saying ‘ k node’, it means the ‘the corresponding node in the k TS’. A good way of representing the algorithm is shown in Fig. 4, where the horizontal axis represents the TSs. In each TS the corresponding node has a probability $\mathbf{P}(\text{TX}_k)$ of performing the transmission and a probability $\mathbf{P}(\text{NO-TX}_k)$ of censoring it, except in the first where the transmission is mandatory.

In the first TS, it is mandatory for the node to transmit, after that, in the second TS, the probability of transmitting is a and the probability of censoring is b . Each time the previous node has transmitted, there exist a probability a for a transmission and a probability b for a censorship. If the previous node has censored the data, then is mandatory to perform the transmission, this is that the transmission probability is 1. The probabilities for each of the firsts 5 TSs is shown in Table II.

The DR used is defined as:

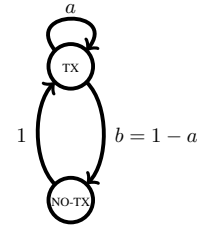


Fig. 5. The Markov chain representing the communication scheme in Fig. 4. The initial state of the chain shall always be the ‘TX’ state.

- If $(X_i - X_{i-1}) > th$ then the transmission is performed.
- If $(X_i - X_{i-1}) \leq th$ then the transmission is censored.

it uses th as parameter and $a = f(\sigma, th) = \mathbf{P}(X_i - X_{i-1} > th)$, $b = 1 - a$ and $\mathbf{P}(\text{NO-TX}_k) = 1 - \mathbf{P}(\text{TX}_k)$. The X_{i-1} PDF is considered Gaussian in both situations (TX and NO-TX). This is a valid approximation used in this work for small values of th , thus a and b may be considered constant for every k . There exists different ways of finding the probabilities in each TSs. As follows two of them are shown:

- 1) Through recurrence equations. We know:

$$\mathbf{P}(\text{TX}_k) = a \mathbf{P}(\text{TX}_{k-1}) + \mathbf{P}(\text{NO-TX}_{k-1}) \quad (26)$$

$$\mathbf{P}(\text{NO-TX}_k) = b \mathbf{P}(\text{TX}_{k-1}) \quad (27)$$

combining (26) and (27) leads us to the recurrence equation (28) with the initial conditions in (29).

$$\mathbf{P}(\text{TX}_k) = a \mathbf{P}(\text{TX}_{k-1}) + b \mathbf{P}(\text{TX}_{k-2}) \quad (28)$$

$$\mathbf{P}(\text{TX}_1) = 1, \mathbf{P}(\text{NO-TX}_1) = 0 \quad (29)$$

This particularly equation is homogeneous and with constant coefficients and thus has a very simple solution:

$$\mathbf{P}(\text{TX}_k) = A x_0^k + B x_1^k$$

x_0 and x_1 are the characteristic polynomial roots of (28) ($t^2 - at - (1-a) = 0$) A and B are found through (29).

- 2) Through Markov chains. The process in Fig. 4 may be considered as the Markov chain showed in figure 5. The Markov operator may be defined as:

$$\mathbf{K} = \begin{bmatrix} a & 1 \\ b & 0 \end{bmatrix} = \begin{bmatrix} a & 1 \\ 1-a & 0 \end{bmatrix} \quad (30)$$

and the probability in the k TS can be calculated as:

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{P}(\text{TX}_k) \\ \mathbf{P}(\text{NO-TX}_k) \end{bmatrix} = \mathbf{K}^{k-1} \mathbf{P}_0, \mathbf{P}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Both solutions leads as to the same result showed in Fig. 6, that shows $\mathbf{P}(\text{TX}_k)$ and $\mathbf{P}(\text{NO-TX}_k)$, as a function of k . It can be seen that process converges and the convergence rate is $\propto a$.

These probabilities are needed for computing the RV obtained in the CH after the DF. In each TS, the CH incorporates a new \hat{X}_{III_i} to the set used to perform the DF:

$$\hat{X}_{III_i} = \begin{cases} X_i & \text{if } x_i \in R_i \\ X_{i-1} & \text{if } x_i \in \bar{R}_i \end{cases} \quad (31)$$

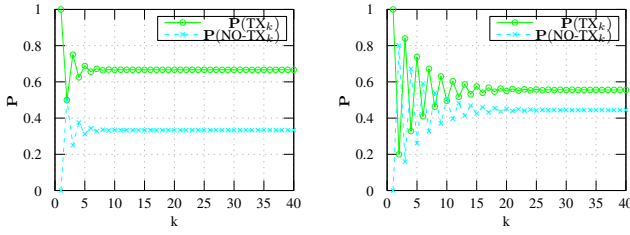


Fig. 6. The evolution of the process of Fig. 4 for $a = 0.5$ and $a = 0.2$ and $N = 40$.

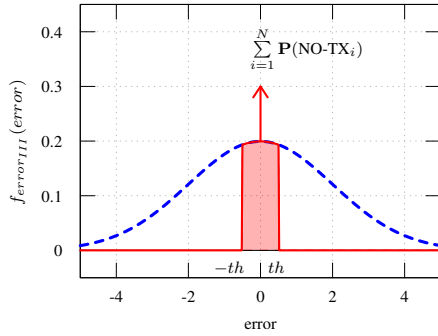


Fig. 7. The error PDF for the case III.

Calculating the variance of $\hat{X}_{CH_{III}} = \sum_{i=1}^N \hat{X}_{III_i}$ is not easy because of the complex correlation that exists between \hat{X}_{III_i} and \hat{X}_{III_j} . For dealing with this problem we rewrite

$$\hat{X}_{III_i} = X_i + error_{III_i} \quad (32)$$

with

$$error_{III_i} = \begin{cases} 0 & \text{if } x_i \in R_i \\ X_i - X_{i-1} & \text{if } x_i \in \bar{R}_i \end{cases} \quad (33)$$

for small values of th its PDF can be considered as a delta in 0 with area $\mathbf{P}(\text{TX}_i)$ and the rest of the area is a trimmed Gaussian RV in $\pm th$ with area $\mathbf{P}(\text{NO-TX}_i)$. The original Gaussian PDF has a zero mean and variance equal to $2\sigma^2$. The RVs $error_{III_i}$ and $error_{III_j}$ are not independent, but because of nature of the algorithm it is possible to demonstrate that $\rho_{error_{III_i}, error_{III_j}} = 0$. The $error_{III}$ PDF is shown in Fig. 7.

By doing this, we may calculate the $\hat{X}_{CH_{III}}$ value in the CH as:

$$\hat{X}_{CH_{III}} = \left(\frac{1}{N} \sum_{i=1}^N X_i \right) + \left(\frac{1}{N} \sum_{i=1}^N error_{III_i} \right) = X_{CH_I} + error_{III} \quad (34)$$

where X_{CH_I} is the one from case I, the ‘ideal case’. With these approach we may say, that in the CH we compute the mean of every X_i plus an error. This error ($error_{CH_{III}}$) is known and we are able to present its statistics:

$$error_{III} = \hat{X}_{CH_{III}} - X_{CH_I} \quad (35)$$

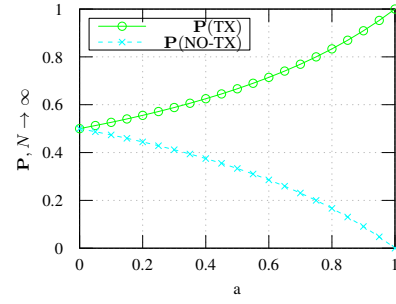


Fig. 8. The convergent values of $\mathbf{P}(\text{TX})$ and $\mathbf{P}(\text{NO-TX})$ as a function of a probability, when $N \rightarrow \infty$.

which has zero mean because $\mu_{\hat{X}_i} = \mu$, and by using CLT and considering that N is big enough:

$$\sigma_{error_{III}}^2 \approx \frac{\sigma_{error_{III_i}}^2}{N} \quad (36)$$

The variance of the RVs are related by their correlation $\rho_{\hat{X}_{III_i}, X_i}$

$$\sigma_{error_{III}}^2 = \frac{\sigma_{\hat{X}_{III_i}}^2 + \sigma_{X_i}^2 - 2\rho_{\hat{X}_{III_i}, X_i} \sigma_{\hat{X}_{III_i}} \sigma_{X_i}}{N} \quad (37)$$

It is easy to see that $\rho_{\hat{X}_{III_i}, X_i} \geq 0$.

To conclude this section, Fig. 8 shows the convergence values for $\mathbf{P}(\text{TX}) = \sum_{i=1}^N \mathbf{P}(\text{TX}_i)$ and $\mathbf{P}(\text{NO-TX}) = \sum_{i=1}^N \mathbf{P}(\text{NO-TX}_i)$ as a function of a probability, when $N \rightarrow \infty$. It can be interpreted as the superior limit of the Censored Transmission Rate (CTR) as a function of a . For $N < \infty$, the CTR decreases, the first mandatory and the subsequent ones do not influence too much in the convergence.

D. Case III variations

Possible variations of case III are: *a)* The asynchronous case III. *b)* The case III, when two or more previous nodes can be heard. *c)* The case III, when the amount of nodes than can be heard depends on i . *d)* The case III, when there exists a hearing probability of others node’s transmissions (depending on the distance, energy, etc.), becoming random graphs. *e)* Any combination of the previous cases.

They will not be analyzed in this work.

E. Metrics

The following metrics are suggested for evaluating the algorithm performance:

- Statistic distance between distributions $d(f_{X_{CH}}, f_{\hat{X}_{CH}})$.
- The ‘miss’ and ‘false detection’ probabilities.
- Mean Square Error (MSE) as in [6].
- CTR.
- Time or iterations needed for reaching consensus.

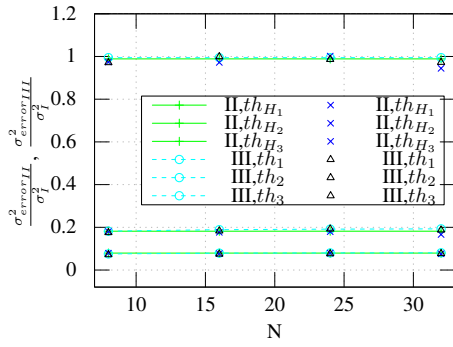


Fig. 9. The relative variance of the error with respect to the variance of the case I as function of N . Theoretic results plotted with lines and markers, simulated ones only with markers.

III. SIMULATIONS AND RESULTS

This section deals with the validation of the proposed mathematical analysis. The considered network topology is the one presented in section II.

The first results to be shown are the ones that validates the theory, this is the error variance in Fig. 9, where it can be seen that the predicted results are in accordance with the simulated ones. This figures shows the relative variance of the error wrt. the variance in case I ($\frac{\sigma_{error}^2}{\sigma_{CH_I}^2}$).

The most interesting results from the point of view of the algorithm performance are show in Fig. 10. The aim of performing DF is to obtain a better measurement, this is the measurement with the smallest possible variance. In the standard case $\sigma_{CH}^2 \propto 1/N$. In this figure, the following convention is adopted: In case I, no transmission is censored, so the variance should not depend on CTR, it should be interpreted as having less quantity of sensors. For example, given N nodes and $CTR = r$, is equivalent to have (rN) nodes, which leads to a $\sigma_{CH}^2 = \frac{\sigma^2}{rN}$. The only objective of doing this, is to have a benchmark value to compare the rest of the results.

From Fig. 10 the following results may be extracted:

- $\sigma_{CH_{III}}^2 = \sigma_{CH_I}^2$ for $r \in \{0, 0.5\}$: The algorithm becomes dummy for those values of r , no DR is evaluated. For $r = 0$ every node performs the transmission and for $r = 0.5$ every odd node performs the transmission, being the same as having $\lfloor \frac{N}{2} \rfloor$ nodes.
- $\sigma_{CH_{III}}^2 < \sigma_{CH_I}^2$ for $0 < r < 0.5$. In this region is where the algorithm achieves better results, between 0.10 to 0.35 the $\sigma_{CH_{III}}^2$ remains almost the same, while the censored transmissions increase. Moreover, the performance is similar to the one in case II.
- For CTR above 0.35, the algorithm derates and case II outperforms it. This is clearly because case II has the advantage of knowing the distribution a priori, but at the same time case II is not able of performing estimations.

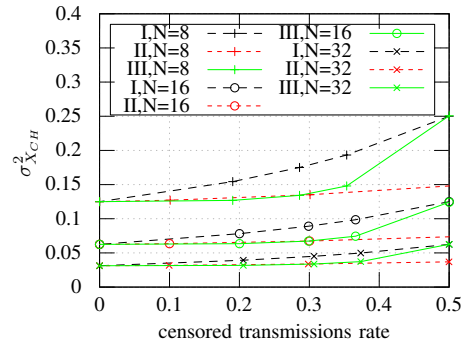


Fig. 10. The X_{CH} variance as a function of the CTR. Case I works as benchmark, in it no transmission is censored and this rate shall be interpreted as having less sensor data to perform the DF. For example, when CTR $r = 0.25$ and $N = 8$, the result for case I is the same as having 6 sensors.

IV. CONCLUSIONS

MESA is a novelty consensus algorithm over WSN for performing distributed detection or estimation, while using a censoring technique. The preliminary theoretic model of the MESA algorithm, has been successfully derived.

In particular the scheme presented in this work, where only one previous transmission can be heard by the nodes, the maximum CTR can rise up to $\lfloor \frac{N}{2} \rfloor \frac{1}{N}$. The algorithm demonstrated to have the better performance when the CRT is not close to zero or close to its upper limit, but around rates between 0.10 and 0.35. This rates may increase for other topologies.

The results of the work encourage us to develop the theory for the variations presented in II-D, which are left for future works.

REFERENCES

- [1] Dong Wentao; Li Wenfeng; Zhang Fan, "Consensus algorithm for energy consumption of wireless sensor networks," Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on , vol., no., pp.2620,2625, 9-12 Oct. 2011 doi: 10.1109/ICSMC.2011.6083992
- [2] Chunhua Sun; Wei Zhang; Ben, K., "Cluster-Based Cooperative Spectrum Sensing in Cognitive Radio Systems," Communications, 2007. ICC '07. IEEE International Conference on , vol., no., pp.2511,2515, 24-28 June 2007 doi: 10.1109/ICC.2007.415
- [3] Ian F. Akyildiz, Brandon F. Lo, and Ravikumar Balakrishnan. 2011. Cooperative spectrum sensing in cognitive radio networks: A survey. Phys. Commun. 4, 1 (March 2011), 40-62. DOI=10.1016/j.phycom.2010.12.003 http://dx.doi.org/10.1016/j.phycom.2010.12.003
- [4] Rago, Constantino; Willett, P.; Bar-Shalom, Y., "Censoring sensors: a low-communication-rate scheme for distributed detection," Aerospace and Electronic Systems, IEEE Transactions on, vol.32, no.2, pp.554,568, April 1996 doi: 10.1109/7.489500
- [5] Zacchigna, Federico G.; Lutenberg, Ariel, "A novel consensus algorithm proposal: Measurement estimation by silent agreement (MESA)," Embedded Systems (SASE/CASE), 2014 Fifth Argentine Symposium and Conference on , vol., no., pp.7,12, 13-15 Aug. 2014 doi: 10.1109/SASE-CASE.2014.6914462
- [6] Pereira, S.S.; Pages-Zamora, A., "Consensus in Correlated Random Wireless Sensor Networks," Signal Processing, IEEE Transactions on , vol.59, no.12, pp.6279,6284, Dec. 2011 doi: 10.1109/TSP.2011.2166552

Accuracy of Propagation Models to Power Prediction in WSN ZigBee Applied in Outdoor Environment

Teles de Sales Bezerra,
José Anderson Rodrigues de Sousa
and Saulo Aislan da Silva Eleutério
Federal Institute of Education, Science
and Technology of Paraíba
Campina Grande - PB, Brazil
Email: (teles, andersonrodrigues,
saulo_eleuterio)@ieee.org

Jerônimo Silva Rocha
Federal Institute of Education, Science
and Technology of Paraíba
Campina Grande - PB, Brazil
Institute of Advanced Studies on Communications - IECOM
Email: jeronimo@iecom.org.br

Abstract—Propagation in Radiofrequency is a constant concern in the application of Wireless Sensor Networks (WSN), the behavior of an environment determines how good the quality of signal reception. The sensor network environment suffers undesirable effects which affect the communication between the modules, the correct modeling of environmental effects is realized by received power prediction that is important for the deployment in the desired environment. The objective of this paper is to analyze the behavior of a WSN applied in brazilian forests and places of beans agriculture, where environmental variables are present, and correlate the captured values of received signal strength (RSSI) with models to power prediction in WSN, and show the best model for this application.

Keywords—WSN, RSSI, prediction, model.

I. INTRODUCTION

WSNs have attracted a growing interest in recent years due to their versatility of being employed in different application fields [1]. The most common performance questions for wireless communications include the range performance due to the large role that the environment has on radio frequency signals. Buildings, trees, obstructions and lack of antenna height can all contribute to a decrease in signal strength at the receiving end. The RF propagation in indoor environment is hard to predict, due to the extensive interference and thus a consistent propagation model is difficult to be derived [2]. The study of propagation is important for understanding the wireless communications because it provides the required physical modelling leading to a good estimate of power required to establish the communications link to provide reliable communication. The propagation effects and other signal losses are generally grouped together and classified as channel properties [3].

To ensure an acceptable level of quality of service for users in wireless data network, network designers rely on signal propagation path loss models. Radio wave propagation models are a series of mathematical calculation developed to predict path characteristics and losses in a given environment [4]. For example, propagation models have traditionally focused on predicting the average received signal strength at a given distance from the transmitter, plus the variability in the signal intensity near a particular location area. Thus, propagation

models are mathematical tools used by engineers and scientists to plan and optimize wireless network systems [5].

Coverage problems are one of the most active research topics related to WSN. It is generally necessary to deploy multiple sensors to cover an entire WSN area so as to provide services within the area. Each sensor used in WSN has a limited sensing radius range [6].

The objectives of this paper is show the Radio Frequency Propagation into areas of brazilian forests and places of agriculture beans, and we made a comparison with real measurements of real RSSI values collected from seven propagations models. This paper is organized in this way: Section 2 shows the wireless sensor networks characteristics. In Section 3 and 4 is showed the RSSI metric and Arduino Platform, respectively, in Section 5 is showed the related works, the Section 6 shows the propagation models used in this paper, in Section 7 is explained the methodology of the measurements. Finally, in Section 8 and 9, are presented the results of measurements and conclusions, respectively.

II. WIRELESS SENSOR NETWORKS CHARACTERISTICS

most promising technologies of the new generation, because its applicable on several problems, i.e. industrial control systems, and becomes more usable on the day-to-day; the low cost of its implementation; and the multi-functional sensors that perform surveillance functions and action control, making this technology to disseminate [7].

A reason for the accelerated grown of wireless communications technologies is their capacity of provide both flexibility and mobility on the network. Another benefit is the dynamic web creation, with low cost and easy implementation. Usually the wireless networks are made with one of three technologies: Wi-Fi, Bluetooth and ZigBee [8].

Unlike technologies as Wi-Fi and Bluetooth, ZigBee offers an interesting option for communication. One of its advantages is its standard that makes possible to uniform the development of devices and applications, easing the leave of proprietary systems [9]. Wireless devices have some limitations. Some of them are not cheap, and have energy use issues. Others, like Bluetooth ones, limit how many nodes may communicate

at any specific time. Wireless sensors do not need high transmission rates, but low latency and energy use instead, in order to grant higher battery autonomy. ZigBee technology has been showing itself as a reasonable alternative for use in WSNs, compared to the other ones available. It works with low power and it is capable of provide the needed underlying structure to allow a great amount of devices to communicate. It also provides low latency communication, with no need to synchronize the network delays [10]. ZigBee technology has been growing faster, but not as fast as its acceptance. ZigBee devices work under IEEE 802.15.4 Standard, written to promote the grown and interoperability between the existent technologies and those that may exist [11].

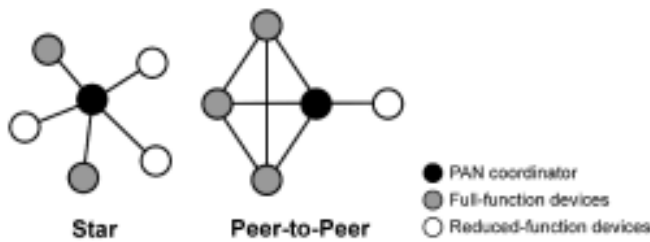


Fig. 1. Topologies of WSNs.

Specifically the standard ZigBee suggests that the ZigBee-based WSN can be composed by three different types of devices, such as: ZigBee Coordinator (ZC) or sink node, a mandatory device in the network; ZigBee Router (ZR) or the Full Function Device (FFD) of the network responsible of routing function; ZigBee End Device (ZED) or the Reduced Function Device (RFD) of the network responsible of detecting events. For our purposes we consider a simple WSN composed by a ZC device (the transmitter) and a ZR device (the receiver) [2].

ZigBee technology stands as an option to supply WSN related application needs. Its differentials are its vantages facing other communication standards, like Wi-Fi and Bluetooth. ZigBee technology's protocol supports mesh, star and tree network topologies, in such a way that this topology diversity increases network packet deliverance trustworthiness. This technology has been gaining notoriety for its simplified code and protocol, together with low development costs and modularization, the latter making easier the assembly of ZigBee devices.

The IEEE Standard that defines ZigBee Data Link layer is IEEE 805.15.4 [11]. It is capable of operate in three band frequencies, 868MHz in Europe, 915MHz in the USA and 2.4GHz on the rest of the world. Using the quoted IEEE Standard, it implements Physical (PHY) and Medium Access Control (MAC) layers. The ZigBee Alliance defines the others layers [12]. XBee is one of the modules technologies used on the assembling of required Hardware on ZigBee networks. Packet transmission is made by an antenna, which maximum power is 60mW, and frequency band ranging from 2.40GHz to 2.48GHz. There are two models, XBee and XBee-Pro [11].

III. RSSI

RSSI (Received Signal Strength Indicator) is used as a metric to estimate the transmission quality between two nodes

based on their relative distance. RSSI is implemented under IEEE 802.11 Standard [13]. This method uses relative distance to estimate the transmitted signal quality by comparing the received signal with probability distributions and location measures basing on the statistic analysis method [14]. RSSI importance is due to the severity of fading effects on wireless communications, making their existence directly affect the performance of wireless communications systems [15], [16], [17].

There are several fading processes affecting the signal propagation, i.e. reflection on obstacles standing on the propagation path. The electromagnetic waves travel through different paths, each one with different lengths, and their interaction of the receiver creates interference. The principal fading processes during the electromagnetic wave propagation are due to reflection, diffraction and scattering [17].

IV. ARDUINO

Arduino is a small micro-controller board, connected to a PC through an USB connection, allowing so a connection between the board and the PC. Moreover, an Arduino board contains several other terminals that allow connection with external devices such as motors, relays, light sensors, LEDs, speakers and so on. This board's project is open-source, that means that any user could construct Arduino-compatible boards. The code opening resulted in a board's cost reduction.

The Arduino's development platform was the base-device during the assembly of RSSI measurer prototypes. The model used was an Arduino Uno R3, chosen due to its great application field and its easy integration with the XBee module's platform. Figure 2 shows the used Arduino device.



Fig. 2. Arduino Uno R3.

V. RELATED WORKS

Various papers have been published with the purpose of investigate the effects on the propagation of radio signals in the devices *ZigBee*. The authors in [18] have investigated the effects caused by external factors in the RF signals. Specifically analyse the RF activity outdoors for 24 hours in order to investigate the influence of time on the RSSI measurements and therefore to estimate the difference between day and night measurements, due the effects of the communication are aleatory and moving human that possibly were present in the area. On the other hand, some works analysed the effects of internal factors about on RSSI measurements, as the effect of polarization antenna between the transmitter and the receiver [19], or the effect of the conception of hardware devices [20]. And other authors in [2] performed a RF propagation analysis using collected RSSI values indoors.

During the studies were also consulted references that not only verified the performance of wireless systems, but it also performed the systems modelling procedures. The authors in [21] have carried out studies with the COST 235 models and ITU-R to model forest environments in Turkey, the authors in [22] performed measurements of performance in wireless systems in an open courtyard at a university in Lisbon - Portugal, the authors in [23] propounded a new propagation model for the characterization of forests in India, which have some features present in the Brazilian forests finally the authors in [24] performed measurements of RSSI values compared to propagation models in a pine forest in Portugal.

VI. PROPAGATION MODELS

Propagation models are fundamental tools for designing and deploying any wireless communication system including WSN in outdoor environments. The models are closely related to the system working environment and characteristics. In general, propagation models are methods and algorithms used to predict the signal strength level along with description of signal level variability. Their main purpose is to predict the distortion and attenuation of the RF signal that will reach the receiver [25].

Currently there are various mathematical models with the objective of predict the average strength of the wireless signal transmission between two network devices. These models are useful in estimating the radio area of coverage of a transmitter and are called propagation models, featuring the signal strength when there is separation between transmitter and receiver. The PL_{dB} , path loss, represents the losses of the model, d represents the distance, f represents the frequency, λ is a wavelength and all losses from these models is given in dBm.

A. Free Space Model

$$PL_{dB} = -10 * \log\left(\frac{G_t * G_r * \lambda^2}{(4 * \Pi)^2 * d^2}\right)$$

Where, G_t is Antenna transmitter gain and G_r is Antenna receiver gain.

B. Log-Distance Model

Defined by [4] [22]:

$$PL_{dB} = PL(d_0) + 10 * n * \log\left(\frac{d}{d_0}\right)$$

The Table I shows the values for the coefficients (n).

TABLE I. N EXPONENTS.

Environment	Path loss exponent (n)
Free space	2
Urban area	2.7 to 3.5
Shadowed urban	3 to 5
Obstructed in building	4 to 6

C. Shadowing Adapted Model

Defined by [26]:

$$PL_{dB} = -10 * \beta * \log(d) + X [dB]$$

$$X [dB] = 9$$

D. Tewari, Swarup e Roy Model

Defined by [23]:

$$PL_{dB} = 88 + 20 * \log(f_{MHz}) + 40 * \log(R_{Km}) - 20 * \log[H_t(m) * H_r(m)] + L_f(dB)$$

Where, H_t is Antenna transmitter height, H_r is Antenna receiver height and R is distance.

E. Weissberger Model

Case $d \leq 14m$:

$$PL_{dB} = 0.45 * f^{0.284} * d$$

Case $14m \leq d \leq 400m$:

$$PL_{dB} = 0.45 * f^{0.284} * d^{0.588}$$

F. ITU-R Model

$$PL_{dB} = 0.2 * f^{0.3} * d^{0.6}$$

G. COST 235 Model

$$PL_{dB} = 15.6 * f^{-0.009} * d^{0.26} - \text{With leaf}$$

$$PL_{dB} = 15.6 * f^{-0.2} * d^{0.5} - \text{Whitout leaf}$$

VII. METHODOLOGY

The methodology for the assembly process, development and implementation of nodes, measurements and analyze data were taken 3 steps, namely:

- Step 1: At this stage was performed the assembly of the RSSI measuring device, using the platform Arduino Uno R3 and XBee modules, in order to verify communication between the devices, and this prototype serves like a RSSI collector. The devices were assembled using a Shield Arduino XBee to correct coupling of the modules to Arduino, in each device, this prototype are showed in Figure 3.



Fig. 3. Prototype developed.

- Step 2: In order to evaluate the propagation behavior in an agriculture environment, the experiment was performed in an area of forests located in Campina Grande city in Paraba, Brazil, and a area of beans agriculture, in the same city, these areas are showed in Figure 4 and Figure 5. With fixation of the transmitter module at one point and performing the displacement of the receiver module along a measurement path until no further communication between module. For the data collection module is placed at a height of 150 cm above the floor in the first data collection modules have a distance of 1 meter from each other immediately after the receiver has been moved 1 meter.
- Step 3: In collecting the data for each point 100 samples were collected RSSI values. In order to evaluate the samples collected was initiated the step of processing the samples with 14 measuring points collected 100 samples and these samples the average RSSI, in total of 14000 samples of RSSI values in the forest, and we collected 45 measuring points with 100 samples in each point, in total 45000 samples os RSSI values on agriculture measurements.



Fig. 4. Brazilian Forest.



Fig. 5. Area of beans agriculture.

VIII. RESULTS

With the conclusion of analysis of data obtained, which is the last stage of the work methodology were obtained satisfactory results, which are desired by the work proposed. In the performed analysis was used the dispersion method of analysis of the samples collected in the environmental preservation forest through the standard deviation calculation, shown at Table II.

TABLE II. STANDARD DEVIATION OF FORESTS SAMPLES.

Distance (m)	Deviation (dBm)
1	6.7172
2	2.8066
3	3.9786
4	1.1101
5	1.3119
6	1.6970
7	5.4259
8	3.7930
9	1.9410
10	1.9410
11	5.1520
12	1.2978
13	2.9802
14	1.4822

Also about the results, Figure 6 shows the curve obtained from the real RSSI values in comparison with the power prediction curves for each propagation model.

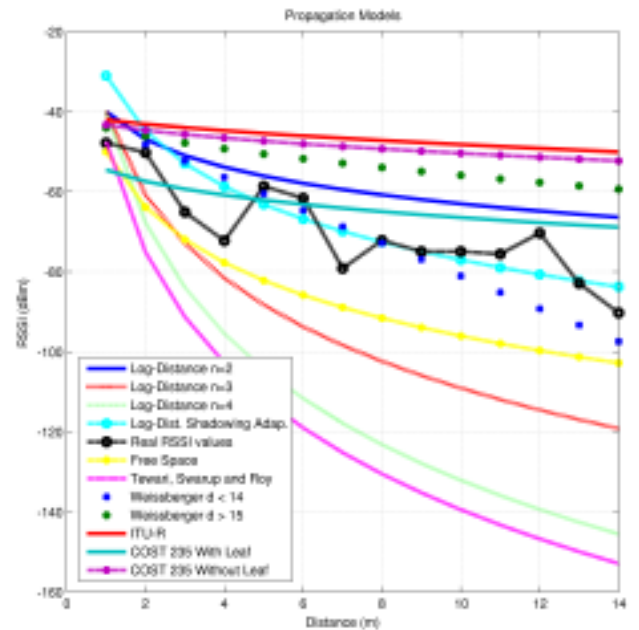


Fig. 6. Curves of Propagation Models on Forest.

The mean square error calculation MSE was used as statistical method to measure the approximation of the actual measurement with the proposed models, the results are described in Table III.

This measurements reveals that the most suitable the prediction model by approximating the value 0 was the Log-distance Shadowing Model, this model presented a MSE equals

TABLE III. TABLE OF MSE ON FOREST.

Model	Mean Square Error
Log-Distance $n=2$	177.977255
Log-Distance $n=3$	779.089054
Log-Distance $n=4$	2217.898427
Log-Distance Shadowing	67.249643
Free Space	324.826798
ITU-R	623.990297
Weissberger $d < 14$	86.079551
Weissberger $d > 15$	347.679306
Tewari Swarup e Roy	2893.533802
COST 235 Without Leaf	532.544086
COST 235 With Leaf	104.051883

67.249643 in the general average of all points, other models also obtained approximate values. The results to the analysis of the measurements made in the field of agriculture were observed with the same statistical methods, wherein the Table IV shows the standard deviation values in the samples collected being taken at a given distance meters and the deviation dBm. Already in the comparative analyzes with the models presented

TABLE IV. TABLE OF STANDARD DEVIATION OF THE SAMPLES IN AGRICULTURE.

Dist.	Deviation	Dist.	Deviation	Dist.	Deviation
1	0	16	1.3345	31	1.8870
2	0	17	1.3345	32	1.8870
3	1.2400	18	1.3345	33	1.8870
4	1.5771	19	1.3345	34	1.8870
5	0.6333	20	0.8562	35	1.6303
6	0.6333	21	0.8562	36	1.0849
7	0.6333	22	0.8562	37	1.7268
8	0.6333	23	0.8562	38	1.2952
9	0.6333	24	0.8562	39	1.6511
10	1.0187	25	1.3219	40	1.7129
11	1.0187	26	1.3219	41	2.4323
12	1.0187	27	1.3219	42	1.0234
13	1.0187	28	1.3219	43	0.8257
14	1.0187	29	1.3219	44	2.2098
15	1.3345	30	1.8870	45	1.9972

by the curves prediction models as compared to measurements are shown in Figure 7 and MSE values are shown in Table V.

TABLE V. MSE TABLE FOR AGRICULTURE.

Model	Mean Square Error
Log-Distance $n=2$	36.527096
Log-Distance $n=3$	3437.407205
Log-Distance $n=4$	7836.074104
Log-Distance Shadowing	415.634733
Free Space	1497.073504
ITU-R	340.914842
Weissberger $d < 14$	5895.590677
Weissberger $d > 15$	43.389256
Tewari Swarup e Roy	9145.218476
COST 235 Without Leaf	277.161223
COST 235 With Leaf	57.416623

It is possible see from the MSE values that the model had a greater efficiency was the Log-Distance model $n=2$, with a MSE of 36.527096, other models also had good results as the model Weissberger $d > 15$ and COST 235 with foliage, wherein the MSE values were 43.389256 and 57.416623, respectively.

IX. CONCLUSIONS

Deployment of a WSN in an forests environment requires the study of signal propagation in order to find the best way of positioning sensor nodes and the power prediction of

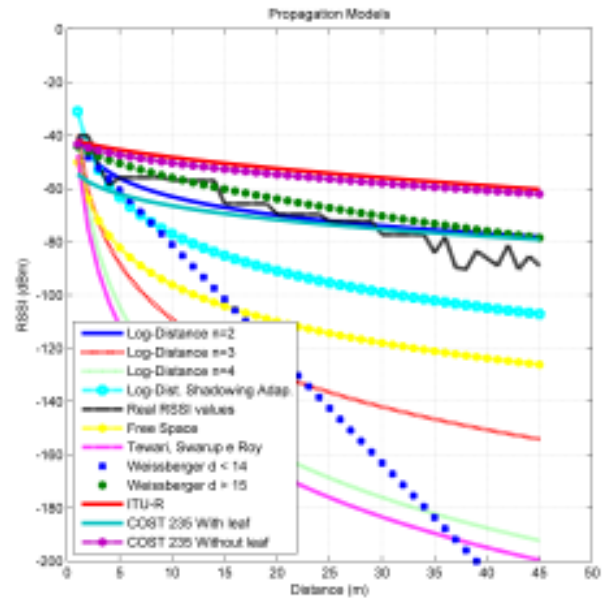


Fig. 7. Curves of propagation models on agriculture.

RSSI values is very important to building this network. This work contributes to the deployment of WSNs in the region of the Campina Grande - Paraba / Brazil in order to optimize the use of resources such as water and electricity. From the analysis of the models propagation to power prediction, we conclude which the Log-Distance Shadowing is the best way to modelling the power prediction in these brazilian forests, but other models like Weissberger Model and COST 235 has good results.

Considering that the time of planting is standardized, there are only natures parameters that can be varied and hinder the communication between the sensors within the limits of distances raised. And the results shows the Log-Distance $n=2$ is the best way to modelling the power prediction in agriculture places.

For this work, we chose to study only the transmission range of the sensors in the forests environments as a future work we will study the impact of climate on these measurements, varying schedules of data collection and on different days with different climates and temperatures.

X. ACKNOWLEDGEMENT

The authors thanks:

- IEEE Student Branch - Campina Grande;
- Federal Institute of Science and Technology of Paraiba Education (IFPB);
- Laboratory of Cognitive Systems and Personal Networks (Labe);
- National Council for Scientific and Technological Development (CNPq);
- Group of Communications (GCOM-IFPB).

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *Communications magazine, IEEE*, vol. 40, no. 8, pp. 102–114, 2002.
- [2] R. M. Pellegrini, S. Persia, D. Volponi, and G. Marcone, "Rf propagation analysis for zigbee sensor network using rssi measurements," in *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on*, pp. 1–5, IEEE, 2011.
- [3] S. Haykin and M. Moher, *Sistemas modernos de comunicações wireless*. Bookman, 2008.
- [4] T. S. Rappaport, *Comunicações sem fio: Princípios e Práticas*. Pearson Prentice Hall, 2009.
- [5] R. Timoteo, D. Cunha, and G. Cavalcanti, "A proposal for path loss prediction in urban environments using support vector regression," in *AICT 2014, The Tenth Advanced International Conference on Telecommunications*, pp. 119–124, 2014.
- [6] W. Kong, M. Li, L. Han, and A. Fukuda, "An smt-based accurate algorithm for the k-coverage problem in sensor network," in *UBICOMM 2014, The Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 240–245, 2014.
- [7] T. d. S. Bezerra, S. Silva, J. Souza, E. Silva, and M. Sousa, "Análise de desempenho em redes de sensores sem fio a partir de um medidor de rssi aplicada ao monitoramento de áreas de preservação ambiental e ambientes para prática desportiva," in *XXI Congr. Intl. de Ingeniería Electronica, Eléctrica y Computación INTERCON 2014*, pp. 209–214, IEEE Sección Peru, 2014.
- [8] A. S. L. F. Fernandes, "Comunicação ad hoc em equipes de robôs móveis utilizando a tecnologia zigbee," mestrado, Universidade de Coimbra, 2012.
- [9] C. E. V. Amador, "Ap wifi/zigbee para suporte à localização baseada em redes sem fios," mestrado, Universidade de Aveiro, 2010.
- [10] T. Tose, A. S. Garcia, A. M. Frasson, R. L. A., and D. N. Sicari, "Redes de sensores sem fio zigbee aplicada em uma estação de tratamento de esgoto," 2012.
- [11] T. Bezerra, S. Silva, E. Silva, M. Sousa, and M. Cavalcante, "Performance evaluation of zigbee transmissions on the grass environment," in *UBICOMM 2014, The Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 287–290, 2014.
- [12] M. Saleiro and E. Ey, "Zigbee uma abordagem prática," 2009.
- [13] K. Benkic, M. Malajner, P. Planinsic, and Z. Cucej, "Using rssi value for distance estimation in wireless sensor networks based on zigbee," in *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on*, pp. 303–306, IEEE, 2008.
- [14] C. Park, D. Park, J. Park, Y. Lee, and Y. An, "Localization algorithm design and implementation to utilization rssi and aoa of zigbee," in *Future Information Technology - FutureTech, 2010 5th International Conference on*, pp. 1–4, IEEE, 2010.
- [15] N. W. Lo, D. D. Falconer, and A. U. Sheikh, "Adaptive equalization for a multipath fading environment with interference and noise," in *Vehicular Technology Conference, 1994 IEEE 44th*, pp. 252–256, IEEE, 1994.
- [16] J.-L. Chu and J.-F. Kiang, "Multipath effects on beacon performances," in *Networking, Sensing and Control, 2004 IEEE International Conference on*, vol. 1, pp. 635–638, IEEE, 2004.
- [17] R.-H. Wu, Y.-H. Lee, H.-W. Tseng, Y.-G. Jan, and M.-H. Chuang, "Study of characteristics of rssi signal," in *Industrial Technology, 2008. ICIT 2008. IEEE International Conference on*, pp. 1–3, IEEE, 2008.
- [18] E. Jafer, B. O'Flynn, C. O'Mathuna, and R. Spinar, "A study of the rf characteristics for wireless sensor deployment in building environment," in *Sensor Technologies and Applications, 2009. SENSORCOMM'09. Third International Conference on*, pp. 206–211, IEEE, 2009.
- [19] M. Barralet, X. Huang, and D. Sharma, "Effects of antenna polarization on rssi based location identification," in *Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on*, vol. 1, pp. 260–265, IEEE, 2009.
- [20] J. Hightower, C. Vakili, G. Borriello, and R. Want, "Design and calibration of the spoton ad-hoc location sensing system," *unpublished, August, 2001*.
- [21] O. Kurnaz, M. Bitigan, and S. Helhel, "Procedure of near ground propagation model development for pine tree forest environment," in *Progress In Electromagnetics Research Symposium Proceedings*, pp. 1403–1406, 2012.
- [22] R. M. P. Jacinto, "Modelação da propagação numa rede de sensores sem fios," 2012.
- [23] R. Tewari, S. Swarup, and M. Roy, "Radio wave propagation through rain forests of india," *Antennas and Propagation, IEEE Transactions on*, vol. 38, no. 4, pp. 433–449, 1990.
- [24] J. A. R. Azevedo and F. E. Santos, "Propagação em ambientes florestais," *Floresta*, vol. 2, no. 35, pp. 4–37, 2008.
- [25] T. Stoyanova, F. Kerasiotis, A. Prayati, and G. Papadopoulos, "A practical rf propagation model for wireless network sensors," in *Sensor Technologies and Applications, 2009. SENSORCOMM'09. Third International Conference on*, pp. 194–199, IEEE, 2009.
- [26] A. Fanimokun and J. Frolik, "Effects of natural propagation environments on wireless sensor network coverage area," in *System Theory, 2003. Proceedings of the 35th Southeastern Symposium on*, pp. 16–20, IEEE, 2003.

Low cost attitude estimation system. Performance evaluation on an airbearing based testbed

F. von Bergen

GPSIC - Dept. of Electrical Engineering
University of Buenos Aires
Buenos Aires, Argentina
Email: federicovonbergen@gmail.com

J. Giribet

GPSIC - Dept. of Electrical Engineering
University of Buenos Aires
Argentine Institute of Mathematics
CONICET
Buenos Aires, Argentina
Email: juan.ignacio.giribet@gmail.com

P. Martos

GPSIC - Dept. of Electrical Engineering
LSE - Dept. of Electrical Engineering
University of Buenos Aires
Buenos Aires, Argentina
Email: pmartos@fi.uba.ar

Abstract—In this work, a low-cost attitude navigation system for spacecrafts is presented. The system fuses measurements from a three axis MEMS gyroscope and a CCD camera that identifies a known pattern of points (which emulate the stars). The system has been implemented on an ARM Cortex-A8 processor, running a GNU/Linux operating system. The developed attitude system was tested on an air-bearing testbed as experimental validation.

I. INTRODUCTION

ATTITUDE Determination and Control Systems (ADCS) are fundamental in the estimation of satellites orientation. A standard formulation to solve the attitude estimation problem is based on inertial measurements (provided by three axis gyroscopes) and external references, such as line-of-sight to known observed stars.

A system that depends only on inertial sensors is called INS (Inertial Navigation System). The inertial sensors provide high-rate information, making it possible to track and control a vehicle with high speed dynamics. Furthermore, assuming gyroscopes measurements are always available, the system is robust respect to external disturbances. Since attitude is estimated by integrating gyroscopes measurements, the main drawback of INS is that the error, due to sensors inaccuracies (noise, bias, scale factor, misalignment, among others) grows unbounded. Those imprecisions are more noticeable in low-cost MEMS (MicroElectroMechanical Systems) gyroscopes, since these sensors have high noises, biases and scale factors errors. In fact, considering that in this technology navigation errors grow rapidly, it is impossible to implement an INS based only on low-cost MEMS sensors.

On the other side, an attitude navigation non-inertial sensor, such as a star-tracker, provides absolute information of vehicle's orientation, by measuring a star pattern and correlating it with a known stars catalog. With this technique it is possible to achieve a navigation solution with bounded errors. However, since image processing is a time consuming task, it is not suitable for vehicles with high speed dynamics. Besides, star-trackers can be affected by external disturbances that make it impossible to identify the stars (for instance, the moon can affect the ability of the star-tracker to see the stars). These

temporally occlusions are a limitation of reference based non-inertial sensors.

As stated, low-cost inertial sensors or star-trackers, in some cases, are not enough to solve the attitude estimation problem as stand-alone navigation sensors. A system based on both types of sensors allows to preserve their advantages and compensate their disadvantages. Essentially the inertial sensors provide attitude information in between star-tracker observations. When a star-tracker measurement is available the drift of the inertial navigation system is compensated. A fusion filter is necessary in order to achieve this, and applied to navigation is called an integration navigation system. Although it requires higher computational effort than an INS, it makes possible to implement a navigation system based on MEMS gyroscopes, since the drift errors will be bounded and estimated by consecutive star-trackers measurements. The recent advances in microprocessors manufacture increased the computational power, while costs remain on hold. This progress permitted the implementation of fusion filters not to be a limitation. In consequence a low-cost, small and lightweight attitude navigation system can be achieved, useful for small satellites or spacecrafts.

This paper provides the reader a standard system model for attitude estimation, using inertial sensors and a CCD camera. The camera will identify a known pattern of points, which emulate the stars. Moreover, it describes the developed software, and its implementation on an ARM Cortex-A8 board connected to the sensors mentioned. The whole project will be tested in an environment with the Hardware In the Loop (HIL) SESCO, which in spanish refers to air-bearing testbed for attitude control systems performance evaluation.

Attitude estimation HIL systems have been used from aeronautical companies to universities [1]. The first testbeds appeared 55 years ago and were based on an air flux suspension technology. Although there exist other types of simulators (e.g. with magnetic suspension), the ones with pneumatic suspension permits the manipulation of the hardware in a practically null friction and torque environment, useful in cases where vehicles are not affected by aerodynamic forces. Regarding the technological difficulties in building an ideal rotative system with three degrees of freedom (3DOF), it is common to have a range of movement restriction in one of the

This work has been partially supported and funded by ANPCyT-PICT1365 and CTP/SEAM-“Concurso de proyectos “General Manuel Nicolás Savio””.

axis. There are a great variety of possible testbeds operated with a pressurized injection of a thin air layer suspension, which can be classified as planar or rotative systems, being the last one the case of the SESCO. The SESCO system (see Fig. 1) will be equipped with the attitude estimation system detailed in this work.

Testbeds allow rigorous, replicative, risk-less and transparent experiments for evaluating and validating, on earth, ADCS. They provide the capabilities for the verification of the development of hardware and software before launching. The direct impact is the cost savings of possible losses of expensive gear by identifying errors prematurely.



Fig. 1. SESCO - air-bearing testbed system

II. PRELIMINARIES

A. Notation

Vectors and matrices are written in bold letters, vectors are in small letters, while matrices are in capital letters. Given $\mathbf{A} \in \mathbb{R}^{n \times m}$, \mathbf{A}^T denotes its transpose. A vector $\mathbf{x} \in \mathbb{R}^3$ written in the coordinate system a is denoted \mathbf{x}^a . Given $\mathbf{v} \in \mathbb{R}^3$, the cross product operator is $[\mathbf{v} \times] = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}$

B. Quaternions

A quaternion can be seen as an extension of the field of complex numbers to a higher dimension space [2]. Let $b_i \in \mathbb{R}$, a quaternion \mathbf{b} is defined as:

$$\mathbf{b} = b_1 + b_2i + b_3j + b_4k = b_1 + \vec{\mathbf{b}} \quad (1)$$

The quaternion product (denoted \circ) is defined based on the following rules: $i \circ i = j \circ j = k \circ k = -1$, $i \circ j = k$, $j \circ k = i$ and $k \circ i = j$. As in complex numbers we define its real and imaginary parts, for a quaternion \mathbf{b} we can define its real (b_1) and vectorial ($\vec{\mathbf{b}} = [b_2 \ b_3 \ b_4]^T$) components. Representing quaternions as elements in \mathbb{R}^4 , lets the quaternion product in matrix form become:

$$\mathbf{b} \circ \mathbf{c} = b_1c_1 - \vec{\mathbf{b}}^T \vec{\mathbf{c}} + b_1\vec{\mathbf{c}} + c_1\vec{\mathbf{b}} + \vec{\mathbf{b}} \times \vec{\mathbf{c}} = \mathbf{Q}_b \mathbf{c} \quad (2)$$

where,

$$\mathbf{Q}_b = \begin{bmatrix} b_1 & -\vec{\mathbf{b}}^T \\ \vec{\mathbf{b}} & b_1 \mathbf{I}_{3 \times 3} + [\vec{\mathbf{b}} \times] \end{bmatrix} \quad (3)$$

It is possible to define the conjugate quaternion. Given a quaternion $\mathbf{b} = b_1 + \vec{\mathbf{b}}$, its conjugate is defined as $\mathbf{b}^* = b_1 - \vec{\mathbf{b}}$. Notice that $\|\mathbf{b}\|^2 = \mathbf{b} \circ \mathbf{b}^*$; and if $\mathbf{b} \neq 0$, $\mathbf{b}^{-1} = \frac{\mathbf{b}^*}{\|\mathbf{b}\|^2}$

C. Rectangular coordinate systems and rotations

Rectangular coordinate systems are composed by three orthogonal axis defined by the right-handed convention. This work will refer to two reference frames: an inertial frame (denoted i , which is supposed known), and a body frame (denoted b , which is fixed to the vehicle).

The vehicle orientation will be determined by the rotation between the vehicles frame (b) and the known inertial frame (i). There exist different representations for these rotation. Two examples are the Euler angles and the invariant vector.

$SO(3)$ denotes the set of rotation matrices $\mathbf{C} \in \mathbb{R}^{3 \times 3}$, i.e., orthogonal matrices with determinant equal 1:

- 1) $\mathbf{C}^{-1} = \mathbf{C}^T$
- 2) $\det(\mathbf{C}) = 1$

An example of a rotation matrix is $\mathbf{C}_i^b \in SO(3)$, which relates the inertial and body frames. Given a vector in the inertial frame \mathbf{v}^i , then $\mathbf{v}^b = \mathbf{C}_i^b \mathbf{v}^i$. The matrix $\mathbf{C}_i^b = (\mathbf{C}_b^i)^T$ gives vehicle's attitude information.

The Hamilton set \mathbb{H} is defined as those quaternions \mathbf{q} such that $\mathbf{q} \circ \mathbf{q}^* = 1$. There is a two to one map between \mathbb{H} and $SO(3)$, in fact $\mathbf{q} \in \mathbb{H}$ and $-\mathbf{q} \in \mathbb{H}$ represent the same rotation.

Given a quaternion $(\mathbf{b}^i)^T = [0 \ \vec{\mathbf{x}}^i]$, where $\vec{\mathbf{x}}^i \in \mathbb{R}^3$ represents a vector in the inertial frame, the quaternion in the body frame is:

$$\mathbf{b}^b = \mathbf{q}_i^b \circ \mathbf{b}^i \circ (\mathbf{q}_i^b)^{-1} = \begin{bmatrix} 1 & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{3 \times 1} & \mathbf{C}_i^b(\mathbf{q}_i^b) \end{bmatrix} \mathbf{b}^i \quad (4)$$

Recall that, given $\mathbf{I} \neq \mathbf{C}_i^b \in SO(3)$, there is a unique real (normalized) eigenvector vector for \mathbf{C}_i^b . Every rotation can be characterized by this invariant vector ($\Phi \in \mathbb{R}^3$) and the (positive) angle rotated around this vector. In fact, if $\Phi/\|\Phi\|$ is the normalized eigenvector and $\|\Phi\|$ is the angle then [3]:

$$\mathbf{C}_i^b = \mathbf{I} + \frac{\sin(\|\Phi\|)}{\|\Phi\|} [\Phi \times] + \frac{1 - \cos(\|\Phi\|)}{\|\Phi\|^2} [\Phi \times]^2 \quad (5)$$

The quaternion representation related with Φ is:

$$\mathbf{q}_i^b = \begin{bmatrix} \cos\left(\frac{\|\Phi\|}{2}\right) \\ \frac{\Phi_x}{\|\Phi\|} \sin\left(\frac{\|\Phi\|}{2}\right) \\ \frac{\Phi_y}{\|\Phi\|} \sin\left(\frac{\|\Phi\|}{2}\right) \\ \frac{\Phi_z}{\|\Phi\|} \sin\left(\frac{\|\Phi\|}{2}\right) \end{bmatrix} \quad (6)$$

Another rotation representation can be given in Euler angles. Although this representation is not useful for algorithm

implementations, due to its singularities, it has an intuitive interpretation. This representation consists of three successive rotations. Each rotation is specified by one of the three axis, where the planar rotations happen, in small letter in a left to right order; separated by the “-” character. Regarding that in intrinsic rotations some of the axis may change, the “~” character will be added after the axis symbol of those changed axis. The rotation matrix is composed of a sequence of matrices associated to the Euler angles rotations. The name of each of these matrices is the axis over which the rotation is happening in capital letters, with the argument being the Euler angles.

In this work the following sequence of rotations is defined: x - y' - z'' . The angles are define as follows:

- ϕ (*roll*): intrinsic rotation over the x axis.
- θ (*pitch*): intrinsic rotation over the y' axis.
- ψ (*yaw*): intrinsic rotation over the z'' axis.

The Euler angles rotation matrix becomes:

$$\mathbf{C}_i^b = Z(\psi) \cdot Y(\theta) \cdot X(\phi) \quad (7)$$

Since there are different possibilities of defining a rotation, it is expected to have formulas that relates them. The conversion of the quaternion representations to Euler angles is:

$$\theta = \arcsin(-2(q_{i2}^b \cdot q_{i4}^b + q_{i1}^b \cdot q_{i3}^b)) \quad (8)$$

$$\phi = \operatorname{atan2}\left(q_{i3}^b \cdot q_{i4}^b - q_{i1}^b \cdot q_{i2}^b, \frac{1}{2} - [(q_{i2}^b)^2 + (q_{i3}^b)^2]\right) \quad (9)$$

$$\psi = \operatorname{atan2}\left(q_{i2}^b \cdot q_{i3}^b - q_{i1}^b \cdot q_{i4}^b, \frac{1}{2} - [(q_{i3}^b)^2 + (q_{i4}^b)^2]\right) \quad (10)$$

$$(11)$$

The conversion of Euler angles to quaternion is:

$$\mathbf{q}_i^b = \begin{bmatrix} \frac{1}{2}\sqrt{1 + c_{i11}^b + c_{i22}^b + c_{i33}^b} \\ \frac{c_{i32}^b - c_{i23}^b}{4 \cdot q_{i1}^b} \\ \frac{c_{i13}^b - c_{i31}^b}{4 \cdot q_{i1}^b} \\ \frac{c_{i21}^b - c_{i12}^b}{4 \cdot q_{i1}^b} \end{bmatrix} \quad (12)$$

D. Euler angles matrix and quaternion dynamics

Attitude can change over time. If $\boldsymbol{\omega}^b$ is defined as the angular velocity of frame b respect to frame i , expressed in b frame, then the rotations representation dynamics are [3]¹:

$$\dot{\mathbf{C}}_b^i = -\mathbf{C}_b^i \cdot \boldsymbol{\Omega}^b, \quad \boldsymbol{\Omega}^b = [\boldsymbol{\omega}^b \times] \quad (13)$$

$$\dot{\mathbf{q}}_i^b = \frac{1}{2} \mathbf{Q}_{\mathbf{q}_{\omega^b}} \cdot \mathbf{q}_i^b, \quad \mathbf{q}_{\omega^b}^T = \begin{bmatrix} 0 & (\boldsymbol{\omega}^b)^T \end{bmatrix} \quad (14)$$

¹Notice that, since inertial sensors are attached to the vehicle, this sensors provide measurements respect to the body frame (this is called a strap-down inertial system).

III. ATTITUDE ESTIMATION

$k \in \mathbb{N}$ is associated to the successive measurements of the sensors, and $t_k \in \mathbb{R}$ refers to the time variable. Consequently $t_{k+1} \in \mathbb{R}$ refers to the next timestep.

There are several techniques for attitude estimation [4], although those based on the Extended Kalman Filter (EKF) are commonly applied. This section proposes an attitude estimation algorithm, based on inertial sensors and camera measurements. It is based on an EKF and consists of two steps: prediction and update.

In previous subsections II-C and II-D, two alternatives were presented for representing rotations between frames and their relation with the angular velocity ($\overline{\boldsymbol{\omega}}^b$) from an IMU (Inertial Measurement Unit). The over-lined angular velocity ($\overline{\boldsymbol{\omega}}^b$) refers to the measurements corrupted with errors. Although a more complex error model was implemented for the gyroscope measurements, this text presents a simplified version of it (a detailed gyroscope error model can be found in [5])

$$\overline{\boldsymbol{\omega}}^b = \boldsymbol{\omega}^b + \mathbf{b}_{\boldsymbol{\omega}^b} + \boldsymbol{\eta}_{\boldsymbol{\omega}^b} \quad (15)$$

$$\hat{\mathbf{b}}_{\boldsymbol{\omega}^b} = \boldsymbol{\eta}_{\mathbf{b}_{\boldsymbol{\omega}^b}} \quad (16)$$

In this model, $\boldsymbol{\omega}^b$ are the real values of the gyroscopes measurements, $\mathbf{b}_{\boldsymbol{\omega}^b}$ is the bias, $\boldsymbol{\eta}_{\boldsymbol{\omega}^b}$ and $\boldsymbol{\eta}_{\mathbf{b}_{\boldsymbol{\omega}^b}}$ are zero mean additive white gaussian noises (AWGN) with their standard deviations $\sigma_{\boldsymbol{\eta}_{\boldsymbol{\omega}^b}}$ and $\sigma_{\boldsymbol{\eta}_{\mathbf{b}_{\boldsymbol{\omega}^b}}}$ respectively. The EKF algorithm estimates the errors from the angular velocity biases ($\delta \mathbf{b}_{\boldsymbol{\omega}^b}$). The biases estimations can be corrected applying $\delta \mathbf{b}_{\boldsymbol{\omega}^b}$ as feedback. An estimation of the angular velocity $\hat{\boldsymbol{\omega}}^b$ is achieved, after the subtraction of the estimated biases from the angular velocity measurements. The latter will be the angular velocity being used for the propagation of the equations. The estimations are obtained through:

$$\hat{\mathbf{b}}_{\boldsymbol{\omega}^b}(t_{k+1}) = \hat{\mathbf{b}}_{\boldsymbol{\omega}^b}(t_k) + \delta \mathbf{b}_{\boldsymbol{\omega}^b} \quad (17)$$

$$\hat{\boldsymbol{\omega}}^b = \overline{\boldsymbol{\omega}}^b - \hat{\mathbf{b}}_{\boldsymbol{\omega}^b}(t_{k+1}) \quad (18)$$

The camera acts as a star-tracker and provides us the position of n identifiable points ($\mathbf{p}_{\text{CAM}}^b$) at a lower rate. $l \in \mathbb{N}^+$ refers to the position point number reference. The camera measurement model for each (l) of the $n \in \mathbb{N}^+$ points is defined by:

$$\mathbf{p}_{\text{CAM}l}^b = \mathbf{p}_l^b + \boldsymbol{\eta}_{\mathbf{p}_{\text{CAM}}^b} \quad (19)$$

\mathbf{p}_l^b represents the real position of the point in body frame and $\boldsymbol{\eta}_{\mathbf{p}_{\text{CAM}}^b}$ is the zero mean additive white gaussian noise (AWGN) with standard deviation $\sigma_{\boldsymbol{\eta}_{\mathbf{p}_{\text{CAM}}^b}}$.

The objective of the algorithm is to fuse the navigation sensors information. The inertial measurements will allow the propagation of the rotation matrix or quaternion, while the camera will correct the estimations. The rotation matrix representation is intuitive, but quaternions are more stable from a numerical point of view and make use of only 4 variables against 9 from the matrix. Consequently, the quaternion representation is a better choice for the prediction phase.

Our attitude algorithm uses primary quaternions, which are part of the propagation step, where numerical errors arise.

The rotation matrix is utilized when there is the need to add the correction obtained from the data of the camera. The implemented algorithm makes use of the equations defined in subsection II-D but in discretized form.

The following subsections III-A and III-B introduce the algorithm's equations.

A. Prediction

$\delta\mathbf{q}_i^b$ represents as a quaternion perturbation of \mathbf{q}_i^b . Therefore the perturbed quaternion rotation can be written as:

$$\bar{\mathbf{q}}_i^b = \delta\mathbf{q}_i^b \circ \mathbf{q}_i^b \quad (20)$$

The propagation of the perturbed quaternion is defined as follows:

$$\bar{\mathbf{q}}_i^b(t_{k+1}) = \left[\cos(\|\mathbf{w}\|) + \frac{\sin(\|\mathbf{w}\|)}{\|\mathbf{w}\|} \mathbf{W} \right] \bar{\mathbf{q}}_i^b(t_k) \quad (21)$$

$$\mathbf{W} = \begin{bmatrix} 0 & -\mathbf{w}^\top \\ \mathbf{w} & [\mathbf{w} \times] \end{bmatrix} \quad (22)$$

$$\mathbf{w} = \int_{t_k}^{t_{k+1}} \hat{\boldsymbol{\omega}}^b(\tau) d\tau \quad (23)$$

$\delta\phi$ is an invariant vector that represents a rotation matrix. This matrix corresponds to the rotation error of \mathbf{C}_i^b . The linear approximation of the perturbed Euler angles matrix is:

$$\bar{\mathbf{C}}_i^b = (\mathbf{I}_{3 \times 3} + [\delta\phi \times]) \mathbf{C}_i^b \quad (24)$$

\mathbf{P} is defined as the covariance matrix of $\delta\phi$ (an invariant vector representing the error of the rotation matrix) and $\Delta\mathbf{b}_{\omega^b}$ (the bias error). The propagation of \mathbf{P} is [3]:

$$\mathbf{P}(t_{k+1}) = \Phi(t_k) \mathbf{P}(t_k) \Phi(t_k) + \mathbf{Q}_d \quad (25)$$

$$\Phi(t_k) = \begin{bmatrix} \Theta(t_k) & \Psi(t_k) \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (26)$$

$$\Theta(t_k) = \mathbf{I}_{3 \times 3} - \frac{1}{\|\hat{\boldsymbol{\omega}}^b\|} \sin(\mathbf{w}) \left[\hat{\boldsymbol{\omega}}^b \times \right] + \quad (27)$$

$$+ \frac{1}{\|\hat{\boldsymbol{\omega}}^b\|^2} (1 - \cos(\mathbf{w})) \left[\hat{\boldsymbol{\omega}}^b \times \right]^2 \quad (28)$$

$$\Psi(t_k) = -\Delta t_k \mathbf{I}_{3 \times 3} + \frac{1}{\|\hat{\boldsymbol{\omega}}^b\|^2} (1 - \cos(\mathbf{w})) \left[\hat{\boldsymbol{\omega}}^b \times \right] - \quad (29)$$

$$- \frac{1}{\|\hat{\boldsymbol{\omega}}^b\|^3} (\mathbf{w} - \sin(\mathbf{w})) \left[\hat{\boldsymbol{\omega}}^b \times \right]^2 \quad (30)$$

$$\Delta t_k = t_{k+1} - t_k \quad (31)$$

$$\mathbf{Q}_d = \int_{t_k}^{t_{k+1}} \mathbf{Q}_c d\tau \quad (32)$$

$$\mathbf{Q}_c = \begin{bmatrix} \sigma_{\eta_{\omega^b}}^2 \mathbf{I}_{3 \times 3} + \sigma_{\eta_{b_{\omega^b}}}^2 \Psi \Psi^T & \sigma_{\eta_{b_{\omega^b}}}^2 \Psi \\ \sigma_{\eta_{b_{\omega^b}}}^2 \Psi^T & \sigma_{\eta_{b_{\omega^b}}}^2 \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (33)$$

B. Update

The reference points position in the inertial frame (\mathbf{p}_i^i) are known. Therefore, each of them in the body frame can be estimated with:

$$\mathbf{p}_{\text{INS}_l}^b = \bar{\mathbf{C}}_i^b \cdot \mathbf{p}_i^i \quad (34)$$

$\mathbf{p}_{\text{INS}_l}^b$ refers to the points positions estimated by the prediction phase, while $\mathbf{p}_{\text{CAM}_l}^b$ is the measurement of the camera. The subtraction of $\mathbf{p}_{\text{INS}_l}^b$ from $\mathbf{p}_{\text{CAM}_l}^b$ gives:

$$\mathbf{z}_l = \mathbf{p}_{\text{INS}_l}^b - \mathbf{p}_{\text{CAM}_l}^b = -[\mathbf{p}_i^i \times] \delta\phi - \boldsymbol{\eta}_{\text{PCAM}}^b \quad (35)$$

\mathbf{z}_l are called the innovations. Considering there are $n \in \mathbb{N}^+$ position points, hence:

$$\mathbf{H} = \begin{bmatrix} -[\mathbf{p}_1^i \times] \\ -[\mathbf{p}_2^i \times] \\ \vdots \\ -[\mathbf{p}_n^i \times] \end{bmatrix} \quad (36)$$

The update EKF equations are:

$$\begin{bmatrix} \delta\phi \\ \Delta\mathbf{b}_{\omega^b} \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_n \end{bmatrix} \quad (37)$$

$$\mathbf{P}(t_{k+1}) = (\mathbf{I}_{6 \times 6} - \mathbf{K}\mathbf{H}) \mathbf{P}(t_k) \quad (38)$$

$$\mathbf{K} = \mathbf{P}\mathbf{H}^\top (\mathbf{H}\mathbf{P}\mathbf{H}^\top + \mathbf{R})^{-1} \quad (39)$$

$$\mathbf{R} = \sigma_{\eta_{\text{PCAM}}^b}^2 \mathbf{I}_{3n \times 3n} \quad (40)$$

The first two equations provide us the error angles ($\delta\phi$) and the update of the covariance matrix.

The propagated quaternion ($\bar{\mathbf{q}}_i^b$) is corrected with the use of the error angles from the previous equation, using the formulas from subsection II-C and the properties from subsection II-B.

IV. HARDWARE DESCRIPTION

The described attitude estimation system is composed by the single-board computer Beagleboard-xM, a MPU9150, a LogitechC920 camera and a network wireless USB hub TP-LINK TL-WN721N.

- BeagleBoard-xM: is an open hardware design board with a 1GHz ARM Cortex-A8 processor, 512MB LPDDR RAM, 4 USB 2.0 ports and an ethernet port among other ports and circuits.
- MPU9150: is the IMU of our system; it is a low-cost nine-axis (gyroscope, accelerometer and compass) MEMS technology device. The integrated circuit is in a 11 pins breakout board from SparkFun which is connected to the computer via a bi-directional level shifter board.
- LogitechC920: is a CCD camera used as a star tracker for our algorithm and is connected to the BeagleBoard-xM through one of the USB ports.

The system makes no use of special hardware or SIP (Semiconductor Intellectual Property) blocks (e.g. DSP) of the board, but it serves as a practical implementation of an embedded system in a Debian GNU/Linux environment for testing the ARM Cortex-A8 microprocessor. It provides the insights and limitations of running an attitude estimation algorithm in a processor without the aid of other processing components. Moreover, given that there was no guarantee of a propitious environment, the magnetometer was not added in the attitude system.

V. SOFTWARE DESCRIPTION

The software regards to the attitude system and all the necessary components for the implementation on the testbed. It is composed of several modules written in C++ and executed via bash scripting. The system modules can be separated in three categories, host, client and communication.

The host modules are the attitude algorithm in charge of the attitude estimation, the IMU which allows the acquisition of the inertial measurements and the camera as interface for the CCD sensor. There are API's built for both sensors, IMU and camera. On top of those API's is the specific code to the sensors being used. This coding schema permits the exchange of sensors without modifying the algorithms. The IMU API contains functions for the estimation of the sensor errors. A camera calibration routine was necessary for the estimation of its intrinsic parameters in order to obtain undistorted images.

The client module is the real-time 3D plotter of the attitude estimations of the host. Fig. 2 is an image captured from a simulation.



Fig. 2. Attitude simulation plot.

The communication regards to the OpenSSL module, which is a custom build of a server and a client that serves as a communication channel between the host and the clients.

In our case the host refers to the BeagleBoard-xM, the clients to personal computers, while the communication is being established with two wireless network USB hubs.

VI. RESULTS

Two case studies were implemented in order to illustrate the working attitude algorithm. They will provide real tests and measurements, which will help to conclude that the estimation of the state variables and the simulator attitude are correct.

The first case corresponds to an estimation of the angular velocity biases (\mathbf{b}_{ω^b}) in a static environment, with an initial

value of zero ($\mathbf{0}_{3 \times 1}$). Before this case the IMU biases from 1000 gyroscopes measurements were estimated and the results were: $\mathbf{b}_{\omega^b} = [-0.0439237 \quad -0.0489388 \quad 0.0350172]$. Figs. 3 and 4 show the estimated biases (rad/s), bounded by their standard deviation obtained from the covariance matrix ($\mathbf{P}(t_k)$). The obtained results of the case were: $\mathbf{b}_{\omega^b} = [-0.0439234 \quad -0.0489392 \quad 0.0350167]$. Considering that the standard deviation for the biases is around $5 \cdot 10^{-7}$, we can conclude that the algorithm is working properly.

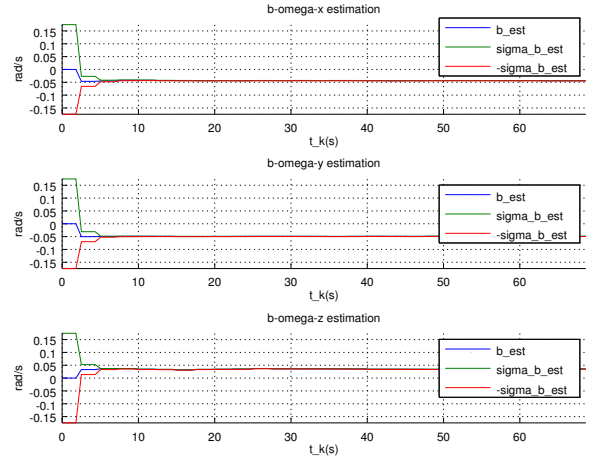


Fig. 3. First case of study - Biases estimation.

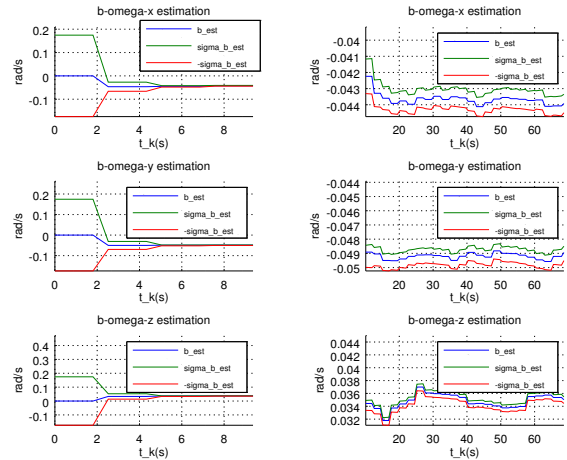


Fig. 4. First case of study - Biases estimation zooming from 0s to 10s and from 10s and forward.

Fig. 5 corresponds to the autocorrelation of the innovations from the first case. The innovations shown in the 8 graphs are the x and y axis from 4 position points acquired by the camera. The stars remained always on the field of view, because the simulator angular velocity was practically null.

The graphs are in accordance with the expectations. It has been able to estimate the biases with a low standard deviation in spite of having a great initial error and drift errors. This simulation was done to show the proper functioning of the system, but our algorithm estimates previously initial biases from gyroscope measurements to improve the initial estimations.

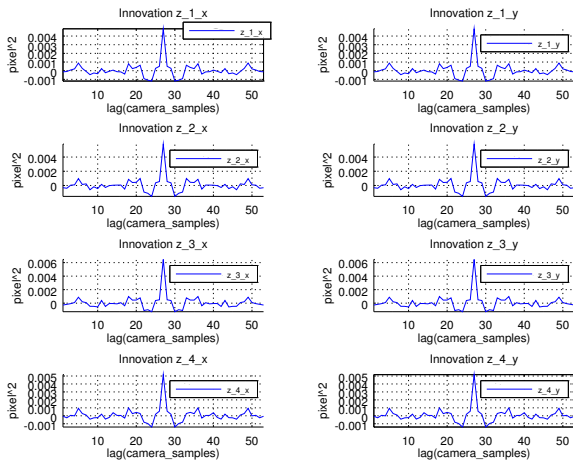


Fig. 5. First case of study - Innovations.

The second case corresponds to an estimation of the euler angles, in a non-static experiment, bounded by their standard deviation obtained from the covariance matrix $(\mathbf{P}(t_k))^2$. The results of the second case are presented in the Fig. 6. Fig. 7 shows the standard deviations without the estimations.

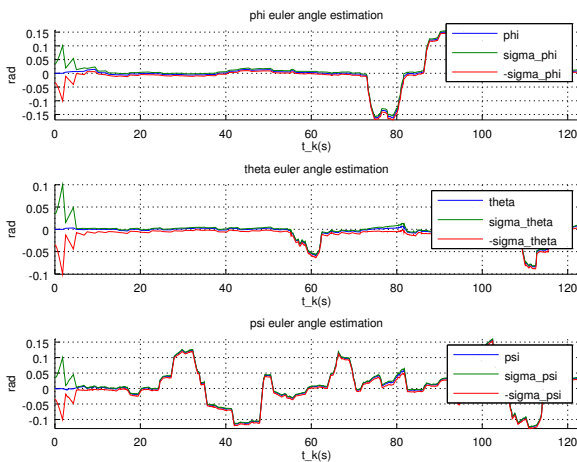


Fig. 6. Second case of study - Euler angles estimations.

Fig. 8 is the autocorrelation of the innovations of the second case. These innovations are the x and y axis from 4 position points obtained by the camera.

The simulations are aligned with the expectations. The angles were estimated bounded with a low error and were in accordance with the movement done by the object. The highest peak observed in Fig. 7 at $t_k \simeq 80s$ is caused by 3 missing camera measurements.

It is worth mentioning that these simulations were done at a low angular velocity, regarding that propagation and update times were fixed at a duration of 0.03s and 0.72s respectively.

²The first three elements of the $diag(\mathbf{P}(t_k))$ is an approximation of the standard deviation, when small angles are considered for the Euler angles.

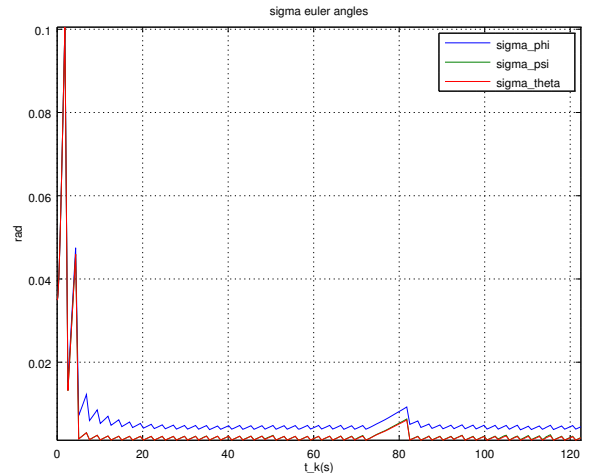


Fig. 7. Second case of study - Sigma Euler angles.

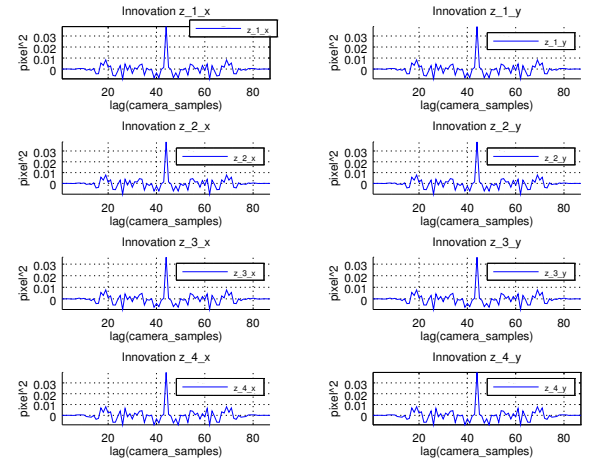


Fig. 8. Second case of study - Innovations.

VII. CONCLUSIONS

The goal of developing a precise low-cost attitude estimation system was achieved. The most important fact is that the computing power needed was fulfilled only by the use of an ARM Cortex-A8 processor and a standard GNU/Linux distribution; without the aid of specific hardware or SIP blocks. The implementation allows the study of attitude control systems in a controlled environment. As described in the results a limitation arises from the processing power of the board, making it only viable for low angular velocities.

REFERENCES

- [1] Schwartz, J. L., Peck, M. A., & Hall, C. D. (2003). Historical review of air-bearing spacecraft simulators. *Journal of Guidance, Control, and Dynamics*, 26(4), 513-522.
- [2] Kuipers, J. B. (1999). *Quaternions and rotation sequences* (Vol. 66). Princeton: Princeton university press.
- [3] Rogers, R. M. (2003). *Applied mathematics in integrated navigation systems* (Vol. 1). AIAA.
- [4] Markley, F. L., & Crassidis, J. L. (2014). *Fundamentals of Spacecraft Attitude Determination and Control*. New York, NY: Springer New York.
- [5] España, M. *Fundamentos de la Navegación Integrada*, AADECA, 2010.

A queue-based scheduling algorithm for PCE-enabled Industrial Internet of Things networks

Ángelo Antonio Farías, Diego Dujovne
Escuela de Informática y Telecomunicaciones
Facultad de Ingeniería,
Universidad Diego Portales, Santiago, Chile
{angelo.farias},{diego.dujovne}@mail.udp.cl

Abstract—The rise of the Industrial Internet of Things as a new paradigm for instrumentation and control has brought the use of internet-enabled constrained devices to complement and replace standard wired networks with wireless connectivity reducing deployment and maintenance costs. This work has been supported by the development of several industry standards, such as ISA100.11a, WirelessHART, 802.15.4e and the current 6TiSCH WG, to enable IPV6 for Time-Scheduled Channel Hopping (TSCH) networks. The most relevant features of TSCH networks are the intrinsic robustness and predictable delay, given a predefined schedule. In this paper, we propose a PCE-enabled scheduling approach to distribute cells within the slotframe by queuing the cell requirements derived from a number of different RPL DODAGs.

Keywords—Internet of Things, WSN, WPAN, 802.15.4e, scheduling, Industrial.

I. INTRODUCTION

The Industrial Internet of Things (IIoT) has evolved as a wireless alternative to wired networks of sensors and actuators. Industrial production plants have strict communication constraints, involving reliability (extremely low packet loss), delay predictability and jitter limitations. Time-Scheduled Channel Hopping [26] was proposed as a solution to this problem, and further evolved through applications and standardization by several industry-wide boards, such as ISA100.11a, WirelessHART, IEEE802.15.4e and currently IETF at the 6TiSCH Working Group. The core of this technology is based on the simultaneous Time-Division Multiplexing and Channel Hopping through all the available wireless channels from the 802.15.4 spectrum. This results in a two-dimensional structure called slotframe, where the resource allocation unit, called cell, represents the availability of one wireless channel for a limited amount of time to transmit or receive data. By limiting not only the transmission time but also the receivers' duty cycle, power consumption is reduced to the minimum, while the end-to-end delay can be bounded.

All the standards defined above specify the mechanisms and methods to allocate resources from the slotframe; nevertheless, the algorithms dedicated to this task are not defined and left open to different implementations from the industry. Our goal is to propose a centralized scheduling algorithm based on an allocation request queue from the RPL DODAG requirements. By using simulations with different topologies and tree depths,

we show that our proposal allocates efficiently the cells according to the system limitations while maximizing the distribution of cells along the slotframe in order to reduce intra-slotframe delay.

The rest of this document is structured as follows: Section II presents related work in the area, section III describes the network stack components, section IV presents the theoretical network and conflict model, section V provides a detailed description of the algorithm, section VI describes the simulation scenarios, section VII analyses the results and finally section VIII presents the conclusions and future work.

II. RELATED WORK

Synchronous and Time-Division Multiplexing MAC layers in wireless sensor networks have been developed and presented in several publications during the last decade, the most comprehensive surveys to date, from Huang et al. [17] and Bashir et al. [16] show a number of different proposals where the resource allocation is done in a distributed manner, such as in MMSN [18], TMCP [19], MC-LMAC [21], Y-MAC [22] and MuChMAC [23] or the centralized GBCA [20] without taking into account pre-established delay and bandwidth requirements. Moreover, the convergence time of the solutions due to negotiation stages or the repair probability is not taken into account. Furthermore, there are no routing constraints considered except for the assumption of static routing.

Regarding specifically to TSCH, a distributed approach from Accettura et al. [25] called DETAS and a centralized version from Palattella et al. [24] called TASA were recently published, thus showing the increasing interest in defining scheduling algorithms for TSCH-enabled networks. Although TASA scheduling is based on topology and traffic constraints, it obtains traffic patterns from local and global queues, while our proposal is based on traffic requests from the nodes. Furthermore, the authors try to keep allocated timeslots sequential while our approach tends to separate them evenly when possible, thus reducing the intra-slotframe delay. Finally, TASA includes an iterative phase, while our algorithm allocates only the bandwidth requests from the request queue, in one iteration.

III. NETWORK STACK ELEMENTS

The 6TiSCH IPv6 stack is built on top of the 802.15.4e TSCH MAC specification. The function of this stack is to generate seamless IPv6 services and connectivity between the resource-constrained nodes and the Internet. The 802.15.4e MAC layer provides all the primitives and services to enable the usage of TSCH on top of the standard 802.15.4 PHY layer; while the proposed 6TiSCH layer manages resource allocation with the 6top layer. It is this layer which reserves, maintains and deallocates cells according to upper layer requirements and individual cell statistics. The routing layer is based on RPL (Routing Protocol for Low Power and Lossy Networks [29]), while application protocols such as CoAP provide uniform access to the network resources.

A. Time-Scheduled Channel Hopping

The slotframe structure is defined as a time-frequency offset channel distribution matrix (CDU) where each cell represents a limited period of transmission time corresponding to a fixed timeslot in a specific frequency offset. The use of Frequency offsets increases link robustness by reducing the influence of narrowband interference and/or multipath fading effects. In the case of the 802.15.4 2.4GHz band, there are 16 different allocated channels. The 6top sublayer is in charge of allocating, deallocating and maintaining link quality given the bandwidth requirements between neighbours. Each allocated cell obtains one of the following flags: TX, RX, shared and timekeeping [30], while the ability to reallocate the cell is defined by the hard cell (fixed) or soft cell (can be reallocated). As a result, the TSCH schedule can be managed either in a centralized manner by a Path Computation Element (PCE) by using a fully distributed scheme or by adapting an initial fixed schedule with dynamic resource allocation.

B. Scheduling

Current TSCH minimal configuration [27] defines 16 wireless channels and 101 timeslots to configure the slotframe. Only the first and the last timeslots are reserved for signalling purposes, while the remaining 99 timeslots are initially unused. However, the slotframe length can vary from one implementation to another, according to the network size, delay and bandwidth requirements.

The TSCH architecture draft [28] defines four slotframe management paradigms:

1) **Static scheduling:**

This mechanism is the simplest one where all the nodes share the same slotframe. This slotframe can be configured either as the initial phase or as recovery mechanism after a network failure. The slotframe allocation is fixed and either preloaded on the nodes or transferred from a Path Computation Element (PCE).

2) **Neighbour to neighbour scheduling:**

This mechanism allows neighbours to negotiate bandwidth in a peer-to-peer basis. Cell reservation is not broadcasted to the rest of the network; as a consequence, it is prone to collisions. In order to reduce this effect

without centralizing cell allocation, the current 6top draft [31] describes a monitoring service to relocate cells according to packet delivery rate statistics at the 6top sublayer.

3) **Remote monitoring and management:**

The architecture draft defines a generic data model which can be used to monitor and manage the nodes' resources remotely. This is made available by a number of Constrained Application Protocol (CoAP) commands, such as *put*, *get*, *post* or *delete*.

4) **Hop by hop scheduling:**

A node can allocate cells throughout the route towards destination. It is the responsibility of the TSCH mechanism at each node along the route to monitor these cells and trigger relocation when needed.

C. RPL

Low-Power and Lossy Networks are characterized by their constrained, energy, processing and storage resources together with a higher packet loss than their wired counterparts, given the wireless channel impairments and low-power transceivers. Among the numerous routing proposals available on the literature, RPL has become the industry standard (by the IETF) for IPv6 routing in this field.

The design goal of this protocol is to build a Destination-Oriented Directed Acyclic Graph (DODAG) starting from the root node (called DAGRoot) and creating a loop-free network tree among the participating nodes. There can be multiple DAGRoots within the same network, although in this work we focus on the most common case of a single DAGRoot.

RPL uses four values to identify and manage the topology:

- RPLInstanceID: Identifies one or more DODAGs. A network can support multiple RPLInstanceIDs, each of them having one or multiple DODAGs. Each of the DODAGs is called RPL Instance.
- DODAGID: Identifies one DODAG within an Instance.
- DODAGVersionNumber: When the DODAG is restructured the version number is increased.
- Rank: Defines the partial order in the DODAG version, assigning the individual position of a node with respect to the DAGRoot.

RPL provides ascending routes towards the DAGRoot. All the RPL nodes build and maintain these DODAGs through DODAG Information Object (DIO) messages. The Objective Function (OF) defines how the RPL nodes select and optimize routes within an RPL instance. The OF is identified by an Objective Code Point (OCP), included in the RPL configuration options. An OF defines how the nodes translate one or more metrics and limitations to the Rank and also establishes the parent node selection criteria for the nodes. The DODAG is built using a distributed algorithm: First, a node becomes the DAGRoot and the other nodes announce themselves the DODAG where it is affiliated, the routing cost and the metrics as multicast packets. Second, the nodes listen to the DIOs and use this information to join to a new DODAG, according to the Objective Function and their respective neighbour Rank.

Finally, the nodes provide an entry to the routing tables for each of the destinations specified on the DIOs, through the DODAG parents. The nodes that decide to join to a specific DODAG can provide one or more parents as a second hop for the default route and a number of external routes for the association stage.

IV. THEORETICAL MODEL

The network and conflict model have been detailed in the work from Palattella et al. [24]. We have based our algorithm on the definitions described on the cited work. In the following, we briefly recall the main characteristics from that publication. Our proposed algorithm is only an approximate solution, given that this kind of scheduling problems is remained as NP-Complete, as mentioned in [24].

A. Network

The network is an oriented graph G with N nodes, namely $\{n_0, n_1, \dots, n_{N-1}\}$ where n_0 is the root node, while the rest n_i are generic nodes. Each node has a parent node, p_i , each parent has a number of child nodes, $ch(n_i)$. Each node is connected to its parent p_i with a dedicated link.

B. Conflict

Given a Physical Connectivity graph, an edge between two different nodes n_i and n_j , is defined when both nodes can hear or interfere each other at the same time. A node n_i cannot transmit nor receive at the same time, and cannot receive from multiple nodes simultaneously. These links, as mentioned in [24], are called Duplex-Conflict links, DC_i for a node n_i ; This includes the links between node pairs in G which cannot transmit simultaneously on the same slot reserved to the pair (n_i, p_i) . We find two main limitations here: *First*, a collision occurs when two nodes n_i transmit at the same time; and *Second*, the parent node p_i cannot receive when it is transmitting.

V. QUEUED SCHEDULING

A. Model

The current model design is built to fulfill the bandwidth requirements of highly demanded nodes, when the DODAG tree below them has a high number of descendants. Figure 1 depicts an example of a DODAG.

In order to compensate the bandwidth requirements given the DODAG structure, the number of allocated slots to enable the communication between neighbours will vary according to the overload value of the link. The overload value is proportional to the number of descendants that share the available bandwidth of the link. For example, on Figure 1(b) link $1 \rightarrow 3$ would be the highest demanded link, given that the DAGRoot node has two descendants plus the traffic from node 3 to node 1.

The overload value is a metric to establish the replication factor k , which is proportional to the number of descendants of the destination node, and defines the number of allocated cells per link, distributed as uniform as possible on the slotframe.

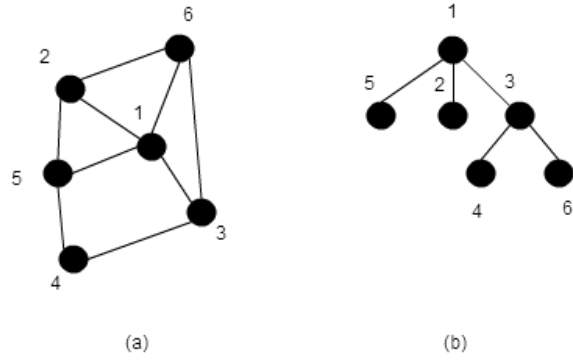


Fig. 1: Figure 1(a) shows the typical grid topology using 6 nodes. Figure 1(b) shows the routing DODAG derived from that topology.

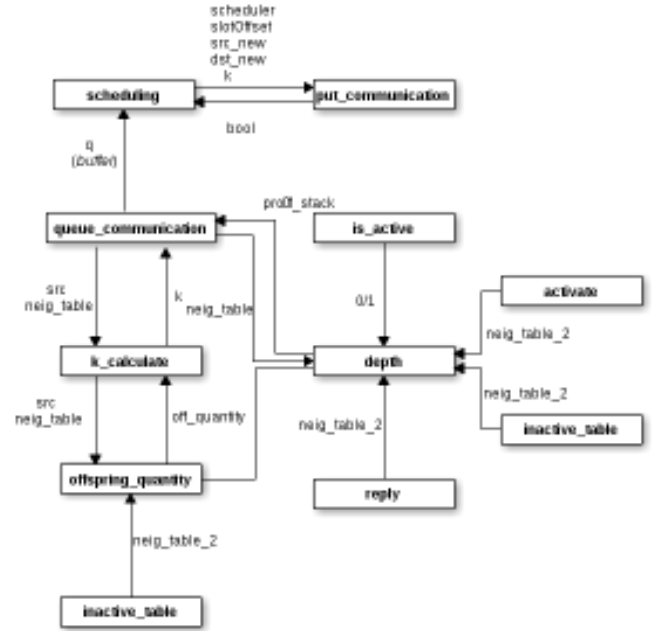


Fig. 2: Proposed Cell Model allocation process

According to the requirements above, we propose the cell allocation process model as described on Figure 2.

On the first stage, the *depth* function calculates the DODAG node depth from the neighbour table. The second step is the *queue_communication* which is a communication buffer which stores the origin and destination nodes plus the replication factor k on each register. The third step is the *put_communication* function, in charge of slot and channel validation from the physical and hardware restrictions to avoid interference: A node cannot transmit on two frequencies simultaneously and a node cannot transmit and receive on the same cell unless it is a shared cell. We also include the limitations described in section IV in this module. Finally, the last step is the *scheduling* function, which finds a valid location in the slotframe for each of the required links in the

queue and the corresponding replications along the slotframe.

The function *put_communication* calls two functions to define the k replication factor. First, *offspring_quantity* calculates the number of descendants for each node and second, *k_calculate* assigns the highest descendant number node with the highest replication factor, where the range (for the slotframe size considered) is between 1 and 16.

VI. SIMULATION

A. Software tools

In this work we have selected two different but complementary tools to evaluate the channel allocation algorithm performance. The first tool is the OpenWSN project to generate the DODAG tree and extract neighbour information, whilst the second tool is Matlab, to implement the scheduler and provide performance results.

OpenWSN is an open source project based on the development of firmware and software for the implementation of TSCH on IIoT networks. OpenWSN firmware covers a wide range of architectures and includes a simulated python platform called OpenSim. Unlike other type of network simulators, OpenSim runs the same Operating System, Stack and Applications on an emulated node as it would run on a real node.

The simulator structure includes five main elements:

- **Propagation**, defines the physical layer, including propagation behaviour, the node entities and the wait times for queued packets.
- **Topology**, is used to configure the network at the link level.
- **SimEngine**, is in charge of managing the timeslots.
- **Mote**, represents the node state with all the physical variables.
- **SimSettings**, contains the simulation configuration parameters.

The neighbour lists and the DODAG configuration are fed to a number of process written in Matlab, as described above, to run the cell allocation algorithm for different topology sizes and distributions.

B. Hardware

The hardware used to simulate the schedule allocation algorithm is composed by a portable computer to run the Matlab routines and a server to run the simulator engine.

Portable computer

- **Processor:** 4x Intel(R) Core(TM) i5-3210M CPU 2.50GHz
- **Memory:** 8Gb
- **Operating System:** Ubuntu 12.04.4

Server

- **Processor:** 8x Intel(R) Xeon(R) CPU E31230 3.20 GHz
- **Memory:** 8272 MB
- **Operating System:** Ubuntu 13.10

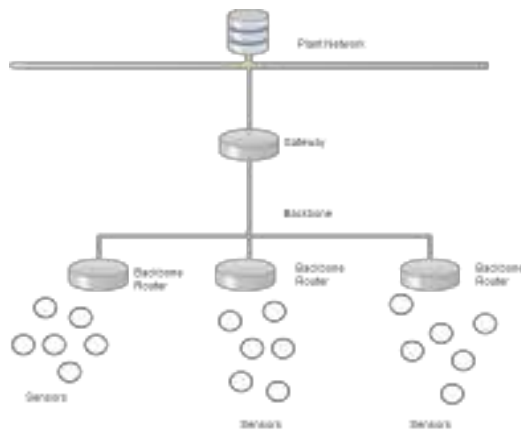


Fig. 3: Backbone topology

C. Scenarios

In order to define the industrial scenarios for the simulation stage, we extracted from current literature the available parameters that characterize these environments. Most of the applications can be satisfied with a network size of 10 to 200 devices, with a maximum path length of 20 hops [10]. Moreover, an important number of industrial applications require realtime data acquisition, thus generating a higher bandwidth demand on the network. We assume that this requires an individual end-to-end flow between each of the collecting node and the DAGRoot.

The most typical industrial topology includes a main backbone as depicted on Figure 3 which interconnects multiple LLNs. This setup provides higher transmission rates and enables the interconnection of multiple and smaller low power networks.

According to [15], the latency and packet loss in a 802.15.4-based industrial network has a mean value of $95 \mu\text{s}/\text{byte}$ with a background noise value of -100 dBm . In this study, we consider 4.1ms latency for a payload of 23 bytes.

In order to provide statistical variability for the results, we generated 20 different topologies of 10, 20, 30 and 40 nodes with the OpenSim simulator. We then exported the routing tree generated by RPL such as the one depicted on Figure 4. This structure is used to build a neighbour table. Finally, the information is sent as input to the Matlab-based scheduler to evaluate the performance of the algorithm.

VII. RESULTS

In this section, we analyse the simulations results. Due to space constraints, we have selected the most representative results regarding the percentage occupation of the slotframe for different topology sizes. From these results, we can derive the delay and energy consumption: Since all requested slots are allocated, delay is only linear to the k replication factor, while energy consumption will be directly proportional to the number of scheduled cells within a slotframe (constant for size 10, 20, 30 and 40 topologies).

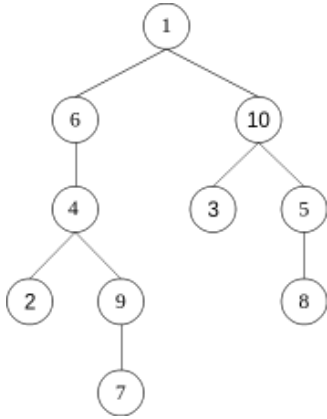


Fig. 4: Example of RPL DODAG used to generate the neighbour table. Numbers are Node IDs.

It can be observed on Fig. 5 that for an increasing k replication factor number, the behaviour is linear and increases the slotframe occupancy when the topology size grows. Nevertheless, the increase on the topology size does not guarantee a fixed increase on efficiency, as it can be observed on the distance between the 30- and 40-node topologies. The slotframe occupancy is small given the available resources even for typical DODAG tree cases as the ones we selected in this paper; this is a direct consequence of the use of a single transceiver, disallowing simultaneous frequency and time slot utilization on the same node. The full time-frequency capacity availability can be then reused to add, for example, leaf nodes on unbalanced branches, or for triggering a route optimization process. Another possibility would be the use of multiple DODAG trees, to reduce the bandwidth constraints on the DAGRoot.

Fig. 6 shows relative slotframe occupation against the tree depth for all the topologies. The relative slotframe occupation is calculated as the ratio of the current occupation value to the product of the number of used channels and the number of timeslots. As a consequence, this ratio reflects the resource allocation density for each of the routing topologies. The best function fit depicts a linear behavior, displaying the slotframe use inefficiency of using very deep DODAG trees. Nevertheless, the use of a replication factor is useful for deep tree branches, in order to distribute the cells in the most uniform manner along the slotframe and reduce latency. Although the apparent schedule utilization rate is low, this corresponds to a single RPL tree, and thus our proposal permits the localization of the cells in a restricted area, leaving the remaining channels to other (possible) RPL instances or to increase the number of nodes on the network. Further improvements such as frequency reuse and interference limitations are part of our current work to improve this algorithm.

Table I represents at a small scale an example of a TSCH slotframe fill according to the proposed algorithm. Gray-background shows allocated cells. It can be observed that after all the requested resources are allocated, there are two

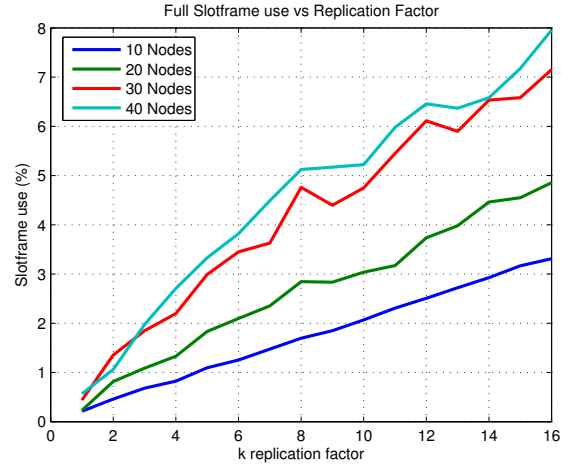


Fig. 5: Full Slotframe occupation vs. k replication factor

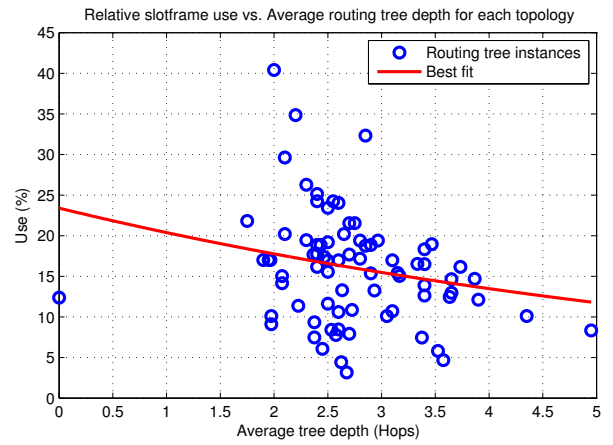


Fig. 6: Relative Slotframe occupation vs. Tree depth

available channels (5 and 6). The relative slotframe usage is calculated with respect to the used (1,2,3,4) channels.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1															
2															
3															
4															
5															
6															

TABLE I: Slotframe with free channels

VIII. CONCLUSIONS AND FUTURE WORK

In this work, we have proposed the queue-based algorithm to allocate TSCH resources on the slotframe in typical industrial scenarios. By evaluating the performance of this algorithm using simulations, we have shown that the absolute occupancy rate is small compared to the standard slotframe size, thus allowing for the use of parallel (non interfering) routing trees, to trigger a RPL routing update or to add more leaf nodes

on the network. We have also shown that there is a linear relationship between the tree depth and the relative occupancy rate, as a consequence, there is an equilibrium point between the configuration of the topology (given spatial limitations) and the efficiency of the slotframe use. We are currently working on the improvement of this algorithm by enabling frequency reuse, include interference considerations and RPL tree adaptation.

IX. ACKNOWLEDGEMENTS

This study was partially supported by Fondecyt (CONICYT, Chile) Project No. 11121475.

REFERENCES

- [1] IEEE 802.15.4 Standards Website. <http://standards.ieee.org/findstds/standard/802.15.4-2011.html>.
- [2] 6TiSCH Working Group site at IETF. <https://datatracker.ietf.org/wg/6tisch/documents/>.
- [3] N. Salman, I.Rasool, A.H.Kemp. *Overview of the IEEE 802.15.4 standards family for Low Rate Wireless Personal Area Networks*, 2010.
- [4] IEEE standard for Information Technology *IEEE std. 802.15.4e, Part. 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer*, April 2012.
- [5] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*, RFC 4944, September 2007.
- [6] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, RFC 6550, March 2012.
- [7] Mark Nixon, *A comparison of WirelessHARTTM and ISA100.11a* July 2012.
- [8] *ISA100, Wireless Systems for Automation*, <http://www.isa.org/Community>.
- [9] *Berkeley's OpenWSN Project Homepage*, <http://www.openwsn.org/>.
- [10] Pister, K., Thubert, P., Dwars, S., and T. Phinney, *Industrial Routing Requirements in Low-Power and Lossy Networks*, RFC 5673, October 2009.
- [11] P. Radmand, A. Talevski, S. Petersen, S. Carlsen. *Comparison of Industrial WSN Standards*, 2010.
- [12] S. Carlsen, A. Petersen, S. Doyle. *Using wireless sensor networks to enable increased oil recovery*, 2008.
- [13] X. Vilajosana, Q. Qang, F. Chraim, T. Watteyne, T. Chang Kristofer S. J. Pister *A Realistic Energy Consumption Model for TSCH*, February 2014.
- [14] M. R. Palatella, N. Accettura, L. A. Grieco, G. Boggia, Mischa Dohler, T. Engel. *On Optimal Scheduling in Duty-Cycled Industrial IoT Applications Using IEEE802.15.4e TSCH*, RFC 5673, October 2009.
- [15] J. Delsing, J. Eliasson, V. Leijon. *Latency and Packet Loss of an Interfered 802.15.4 Channel in an Industrial Environment*, 2010.
- [16] Bachir, Abdelmalik, Mischa Dohler, Thomas Watteyne, and Kin K. Leung. *MAC essentials for wireless sensor networks. Communications Surveys and Tutorials, IEEE 12, no. 2 (2010): 222-248.*
- [17] Huang, Pei, Li Xiao, Soroosh Soltani, Matt W. Mutka, and Ning Xi. *The evolution of MAC protocols in wireless sensor networks: A survey. Communications Surveys and Tutorials, IEEE 15, no. 1 (2013): 101-120.*
- [18] Zhou, Gang, Chengdu Huang, Ting Yan, Tian He, John A. Stankovic, and Tarek F. Abdelzaher. *MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks.* In *Infocom*, vol. 6, pp. 1-13. 2006.
- [19] Wu, Yafeng, John A. Stankovic, Tian He, and Shan Lin. *Realistic and efficient multi-channel communications in wireless sensor networks.* In *INFOCOM 2008. The 27th Conference on Computer Communications.* IEEE, 2008.
- [20] Yu, Qing, Jiming Chen, Yanfei Fan, Xuemin Shen, and Youxian Sun. *Multi-channel assignment in wireless sensor networks: A game theoretic approach.* In *INFOCOM, 2010 Proceedings IEEE*, pp. 1-9. IEEE, 2010.
- [21] Incel, Ozlem Durmaz, Lodewijk van Hoesel, Pierre Jansen, and Paul Havinga. *MC-LMAC: A multi-channel MAC protocol for wireless sensor networks.* *Ad Hoc Networks* 9, no. 1 (2011): 73-94.
- [22] Kim, Youngmin, Hyojeong Shin, and Hojung Cha. *Y-mac: An energy-efficient multi-channel mac protocol for dense wireless sensor networks.* In *Proceedings of the 7th international conference on Information processing in sensor networks*, pp. 53-63. IEEE Computer Society, 2008.
- [23] Borms, Joris, Kris Steenhaut, and Bart Lemmens. *Low-overhead dynamic multi-channel mac for wireless sensor networks.* In *Wireless Sensor Networks*, pp. 81-96. Springer Berlin Heidelberg, 2010.
- [24] Palatella, Maria Rita, Nicola Accettura, Mischa Dohler, Luigi Alfredo Grieco, and Gennaro Boggia. *Traffic Aware Scheduling Algorithm for reliable low-power multi-hop IEEE 802.15. 4e networks.* In *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, pp. 327-332. IEEE, 2012.
- [25] Accettura, Nicola, Maria Rita Palatella, Gennaro Boggia, Luigi Alfredo Grieco, and Mischa Dohler. *Decentralized traffic aware scheduling for multi-hop low power lossy networks in the internet of things.* In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on*, pp. 1-6. IEEE, 2013.
- [26] Dujovne, Diego, Thomas Watteyne, Xavier Vilajosana, and Pascal Thubert. *6TiSCH: deterministic IP-enabled industrial internet (of things).* *Communications Magazine, IEEE 52, no. 12 (2014): 36-41.*
- [27] Vilajosana, Xavier, and Kris Pister. *Minimal 6TiSCH configuration.* (2013).
- [28] Thubert, Pascal. *An Architecture for IPv6 Over Time Slotted Channel Hopping. IETF ID draft-thubert-6tsch-architecture (Work In Progress)* (2015).
- [29] Winter, T. *Routing Protocol for Low-Power and Lossy Networks.* rfc 6550, 6551, 6552. IETF, 2012.
- [30] Palatella, M., P. Thubert, and T. Watteyne. Q. Wang, *Terminology in IPv6 over the TSCH mode of IEEE 802.15. 4e, Internet-Draft draft-ietf-6tischterminology-04*, 2015.
- [31] Wang, Qin, Xavier Vilajosana, and Thomas Watteyne. *6tsch operation sublayer (6top).* Internet-Draft draft-wang-6tisch-6top-sublayer-01 (2015).

FreeRTOS User Mode Scheduler for Mixed Critical Systems

Francisco E. Páez^{1,2}, José M. Urriza¹

¹Depto. de Informática - Facultad de Ingeniería
Universidad Nacional de la Patagonia San Juan Bosco
Puerto Madryn, Argentina

²CONICET

fpaez@unpata.edu.ar, josemurriza@unp.edu.ar

Ricardo Cayssials³, Javier D. Orozco^{2,3}

³Depto. de Ingeniería Eléctrica y Computadoras
Universidad Nacional del Sur
Bahía Blanca, Argentina

Abstract— Real-time systems are applied in the more diverse applications. Most of these applications require specific configurations in order to meet their timing constraints. Real-time Operating Systems offer scheduling policies that produce a certain behavior of the application, difficult to change in order to get a specific and desired dynamic of the system. On the other hand, most real-time applications include tasks with different real-time constraints: from hard real-time tasks to sporadic and soft real-time tasks. Configuring a real-time operating system to get the desired response for each one of these tasks generally involves a deep knowledge of the structure of the kernel and programming it in a privilege mode. In this paper, it is proposed an implementation of a user-level scheduler, easily configurable to adapt the dynamic of the system to the specific features of the application.

Keywords—Real-Time System; Scheduling; Mixed Critical Systems; Application-Defined Scheduling

I. INTRODUCTION

Real-Time Operating Systems (RTOS) are used in the most widely range of applications. The evolution of these applications forces RTOSs to adapt their features to new requirements. In such a way, it is possible to find applications in which the RTOS has to execute, not just a set of *real-time tasks* (RTTs), but also a set of *non-real-time tasks* (NRTTs). These kinds of systems that deal with a *heterogeneous set of tasks* (that involves RTTs as well as NRTTs) are named *Mixed Critical Systems* [1, 2]. When a RTOS has no support for a heterogeneous set of tasks, it may happen that: (1) the timing constraints of the RTTs may not be satisfied, (2) the performance of NRTTs may not be adequate or (3) the system resources may be wasted.

There exist two alternatives to adapt a RTOS in order to support a heterogeneous set of tasks. The first one is to modify the kernel in order to include new scheduling policies for this kind of set of tasks. The second alternative is to provide a *user-level scheduler*, which overrides the kernel scheduler actions.

The former alternative requires changes to the RTOS kernel, which in many cases are complex from several points of view: requires a thorough knowledge of the RTOS structure, and can cause the loss of safety certifications, robustness and

reliability, although modifications of this type lead to a more efficient use of the resources. The second option, unlike the first one, allows the designer to easily adapt the RTOS to the application specific scheduling requirements. The RTOS safety certifications, robustness and reliability are not altered, and it should be just enough to verify the developed application. A survey of previous and related work in user-level schedulers is presented in the next section.

In this paper, a user-level scheduler is developed through a user-level task, called *Heterogeneous Scheduling Task* (HST), in order to support *Mixed Critical Systems* in a very flexible way. An implementation on *FreeRTOS* is presented.

A. Previous and related work

Similar methods are proposed in the literature. An application-defined scheduling for *MaRTE OS* is proposed in [3, 4], and support for user-level hierarchical scheduling policies for *FreeRTOS* are presented in [5, 6]. An application-defined scheduling model and its implementation as a library framework for *RTLlinux* are presented in [7, 8]. A framework for implementing real-time scheduling algorithms, verified with a model checker, is presented in [9]. A general scheduler framework, with implementations for *Linux* and *VxWorks*, called *ExSched*, is developed in [10]. Similar frameworks for prototyping and implementing user-space hierarchical schedulers are *SF3P* ([11]), *FSF* ([12]) and *Hijack* ([13]). A high-level domain specific language for programming user-space scheduling policies, called *Bossa* and a framework for executing it, are developed in [14]. In [15], it is proposed a *CPU broker* that mediates between a RTOS and its RTTs, and provides the means of implementing a new scheduling policy. A system that enables applications to dynamically load and unload scheduling policies into a general-purpose operating system is described in [16].

B. RTOS Selection

Nowadays, there exist several RTOSs used in different kind of applications. *VxWorks*, *QNX*, *μs-OS*, *eCos*, *Windows CE* and *RT-Linux* are just a few RTOSs currently available. The selection of the RTOS to implement the proposed HST was based mainly on: (1) portability of the RTOS among different embedded systems, (2) availability of well

documented source code and (3) license to introduce modifications into the source code. With these criteria in mind we selected *FreeRTOS* which is a small footprint and widely used *RTOS*.

FreeRTOS is an open source *RTOS* distributed under a modified *GPL* license. It is mainly written in *C*, with small sections written in assembler. Its main features are: its small footprint, its low memory and processing requirements and its availability in more than 33 processor architectures.

FreeRTOS implements tasks as threads, each with its own stack of configurable size, and supports processors with Memory Protection Unit (*MPU*). It implements multiple memory profiles, from static to dynamics ones with support of *malloc* and *free* functions.

FreeRTOS provides a preemptive scheduler, with a *FIFO* for each priority level. The scheduler guarantees that, for any given instant, it will execute the highest priority task in the ready list. If multiple tasks with the same priority are ready, a *Round Robin* policy may be enabled among them. The priority of a task is assigned by the designer when created, and can be modified at runtime.

C. Task Model

A *Mixed Critical System* contains two sub-sets of tasks, each one with a specific task model.

1) Critical and Periodic Real-Time Tasks sub-set

The *Critical* and *Periodic RTTs* are considered independent and preemptable. However, tasks may share system resources among them and exclude themselves in critical sections. We assume that the *RTT* sub-set is schedulable under the chosen scheduling policy.

Each periodic task i is characterized by its *worst-case execution time* ($WCET_i$ or C_i), period (T_i), deadline (D_i), and worst case blocking time (B_i), given by the interference that would produce lower priority tasks when inversion priorities takes place. It is assumed that $C_i \leq D_i \leq T_i$. Because of periodicity, each task generates an infinite number of instances. The utilization factor of task i , denoted UF_i , determines its relative utilization of *CPU* processing time, and is defined as C_i/T_i .

2) Non-Real-Time Tasks sub-set

The *NRTTs* sub-set includes tasks with different execution requirements. Tasks belonging to this sub-set range from *aperiodic* and *sporadic* ones that require execution as soon as possible, to tasks that do not have temporal specifications and could be executed when the system became idle. Whilst the scheduler has to satisfy the temporal constraints for *RTTs*, it has to improve as much as possible the design criteria of quality of service for the *NRTTs*.

In a *Mixed Critical System*, the total utilization factor, defined as the summation of the UF_i of all tasks, may be greater than 100%. However, in order to meet all the temporal constraints of the *RTTs*, its utilization factor should be less than 100%. On the other hand, *RTTs* has to leave enough idle processing time to be able to execute the *NRTTs*.

II. SYSTEM ARCHITECTURE

The *Task Control Block (TCB)* defined by *FreeRTOS* does not include attributes to hold particular parameters for generic task models as the proposed in this paper. For this reason, it was defined an *extended TCB*, denoted *eTCB*, based on the attributes that a task in a *Mixed Critical System* may present. Moreover, each *eTCB* has a reference to one *TCB* in order to keep functional the *FreeRTOS* tasks control functions.

The *eTCB* for a task i hold the following parameters:

- Pointer to *FreeRTOS TCB (Task handler)*.
- Task priority.
- Worst Case Execution Time (C_i).
- Task Period (T_i).
- Relative deadline (D_i).
- Absolute deadline (d_i).
- Latest release time (x_i).
- Number of instances (j_i).
- Current executed time at instant t ($c_i(t)$).
- Pointer to additional parameters structure.

By means of this last attribute, the *eTCB* may be extended in order to adequate it to specific requirements of a scheduling policy.

The *eTCBs* are accessed by the *HST* to determine the next task to be executed. Contrary to the only one ready list of the *FreeRTOS*, the *HST* implements two ready lists: one for the *RTT* sub-set and the other for the *NRTT* sub-set.

The *HST* is implemented as a *FreeRTOS* task that is executed with an *HST_Prio* priority. On the other hand, tasks belonging to the *RTT* and *NRTT* sub-sets are implemented as *FreeRTOS* tasks with priorities equal to *HST_TaskPrio*. Both *HST_Prio* and *HST_TaskPrio* are configurable, but they always has to satisfy $\bar{HST_Prio} > HST_TaskPrio$.

The *HST* release takes place on the following events:

1. *Timer tick*: this event takes place with each *CPU* clock interrupt.
2. *Task release*: this event takes place each time a task is released and inserted in the ready list (task is promoted from waiting to ready state).
3. *Task completion*: this event takes places when a task finishes its execution. In this case, the task is suspended until its next release.
4. *Task block*: this event takes place when a task requires a shared resource granted previously to another task.
5. *Task suspension*: this event takes place when the execution of a task is suspended for a certain interval (using *delay* or *sleep* functions)

Each one of these events determines the activation times of the *HST* in which there exists a certain chance to produce a

Context Switching (CS). The Fig. 1 shows the interaction between the *HST*, the scheduled tasks and the *RTOS* kernel.

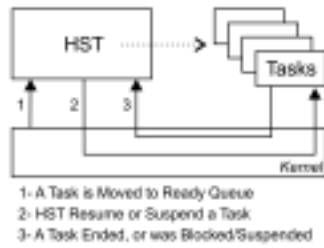


Fig. 1. General interaction among the *HST*, its application-scheduled tasks and the *FreeRTOS* kernel.

The *HST* functionality is implemented by two modules: (1) the *AppSched* module and (2) the *AppSchedLogic* module. These modules are described in the following sections.

A. *AppSched* module

This module is in charge of implementing the scheduling mechanism.

The *AppSched_HST()* function implements the logical behavior of the *HST*, and call the *AppSchedLogic_Sched()* function in the *AppSchedLogic* module to perform the scheduling policy and then it blocks itself.

This module defines an *Application Programming Interface (API)*, consisting of the following functions:

- *AppSched_Init()*: creates the *HST* and its resources and executes the *RTOS* scheduler.
- *AppSched_Create()*: creates a *RTT* or *NRTT*, scheduled by the *HST* instead of the *RTOS* scheduler. This function reserves the memory resources required by the *eTCB* and initialize its parameters. The created task is inserted into the list of tasks managed by the *HST*. Its initial state is changed to suspended, in order to avoid that the *RTOS* scheduler executes it.
- *AppSched_WaitForNextPeriod()*: blocks the calling task until its next period. The *eTCB* holds the values of the period (T_i) and latest release time (x_i) to implement the behavior of a periodic task.

The module also defines the following private functions, which are invoked when certain events take place:

- *AppSched_Tick()*: this function is invoked at every *RTOS* timer tick. As it is executed within the context of an Interrupt Service Routine (*ISR*) its implementation should be efficient. Unblock the *HST* if it is required.
- *AppSched_Ready()*: this function is invoked when the *RTOS* promotes a task from *blocked* or *suspended* to *ready* state. This function should verify that the task was promoted by the *RTOS* instead of the *HST*. In such a case, the *eTCB* is updated as follows: (1) absolute deadline is updated ($d_i = x_i + D_i$), (2) the instance counter is incremented ($j_i = j_i + 1$) and (3) the processing time of the task is set to zero ($c_i = 0$). At the end, the *HST* is released to perform a new scheduling.

- *AppSched_Delay()*, *AppSched_Block()* and *AppSched_Suspend()*: these functions block and unblock the *HST* and they are invoked when a task finishes its execution, it is blocked or it is suspended, respectively.

B. *AppSchedLogic* module

This module implements a specific scheduling policy through the following functions:

- *AppSchedLogic_Init()*: initializes all the data structures required by the implemented scheduling policy.
- *AppSchedLogic_Tick()*: implements the action to be carry out each time that the timer tick interrupt takes place. Returns *true* if it is required to unblock the *HST*.
- *AppSchedLogic_Sched()*: implements the scheduling policy.

III. ADAPTING *FreeRTOS* TO MANAGE MIXED CRITICAL SYSTEMS

This section describes how the modules presented in the previous section are implemented in *FreeRTOS*.

A. *AppSched* module implementation

The *AppSched_Tick()* function is implemented using the *vApplicationTickHook()* callback function of *FreeRTOS*, which is executed at every timer tick interrupt.

The *HST* is created in the *AppSched_Init()* function. The *HST* is implemented as an infinite loop, inside which it executes the scheduling policy calling the *AppSchedLogic_Sched()* function. The *HST* blocks itself at the end of each loop. Then, it is unblocked from one of the events listed in section II.

Trace macros is a *FreeRTOS* feature that offers developer macros in order to expand its functionality. Most of the time, this *trace macros* feature is used to implement code profiling techniques and code debugging strategies.

The *HST* uses these *trace macros* features to invoke the following functions of the *AppSched* module:

- The *AppSched_Delay()* function is associated with the *trace macro traceTASK_DELAY_UNTIL()*. This macro is called within the *vTaskDelayUntil()* function.
- The *AppSched_Block()* function is associated to the *trace macros traceBLOCKING_ON_QUEUE_RECEIVE()* and *traceBLOCKING_ON_QUEUE_SEND()* (semaphores and mutexes are implemented with queues in *FreeRTOS*).
- The *AppSched_Suspend()* function is associated to the *trace macro traceTASK_SUSPEND* called within the *vTaskSuspend()* function.
- The *AppSched_Ready()* function is associated to the *trace macro traceMOVED_TASK_TO_READY_STATE()*.

The synchronization between the *HST* and its scheduled tasks is implemented through a *direct to task notification*. This mechanism allows unblocking a task using a straight

notification that may include a value. In this way, it is possible to replace very effectively the utilization of *binary semaphores* for task synchronization, avoiding the overhead produced by them. The *HST* blocks itself when trying to *take* a notification and it is resumed by a notification issued from the correspondent *traces macros* functions:

- *AppSched_Block()*, *AppSched_Suspend()* and *AppSched_Delay()*: these functions unblock the *HST* sending a notification to it, using the *xTaskNotifyGive()* function. The notification is not sent if the *HST* is the current task in execution. Fig. 2 shows an example of its execution.

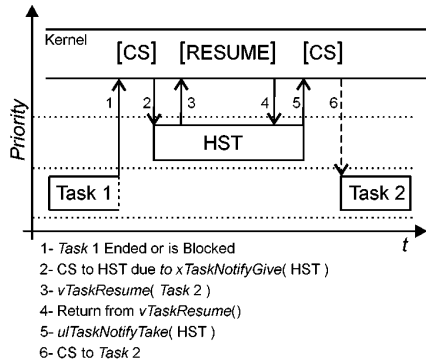


Fig. 2. Example of the *HST* being released after the finalization of the current release of Task 1, and then performing a CS to Task 2.

- *AppSched_Ready()*: this function is called when a task is moved to the ready state, and activates the *HST* by sending a notification to it. If the *AppSched_Ready()* function is executed from an *ISR*, a CS is forced with the *portYIELD_FROM_ISR()* function. Fig. 3 shows an example of its execution from an *ISR*.

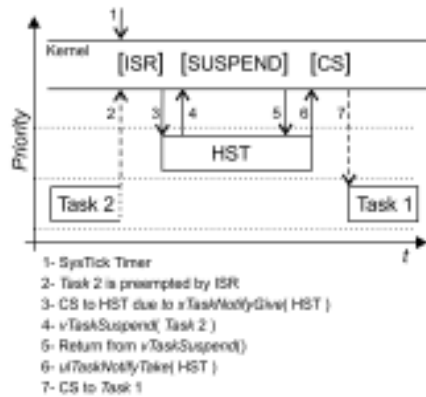


Fig. 3. Example of the *HST* being released from an *ISR*, and performing a CS from Task 2 to Task 1.

The *AppSched* module *API* is implemented as follows:

- *AppSched_WaitForNextPeriod()*: this function executes the *FreeRTOS* *vTaskDelayUntil()* function, using the *x_i* and *T_i* parameters from the *eTCB* as arguments.

- *AppSched_Create()*: creates a task using the *xTaskCreate()* function, with a priority equal to *HST_TaskPrio*. The task is then suspended using the *vTaskSuspend()* function in order to avoid inserting it into the *FreeRTOS* ready list. System memory is reserved for the corresponding *eTCB* by calling the *pvPortMalloc()* function.
- *AppSched_Init()*: creates the *HST* by calling the *vTaskCreate()* function with a priority equal to *HST_Prio*. It calls the *AppSchedLogic_Init()* function and then the *FreeRTOS* scheduler is executed, which starts the *HST*.

Fig. 4 shows the overall architecture of both modules and they interaction with the *FreeRTOS* kernel and a user-level application.

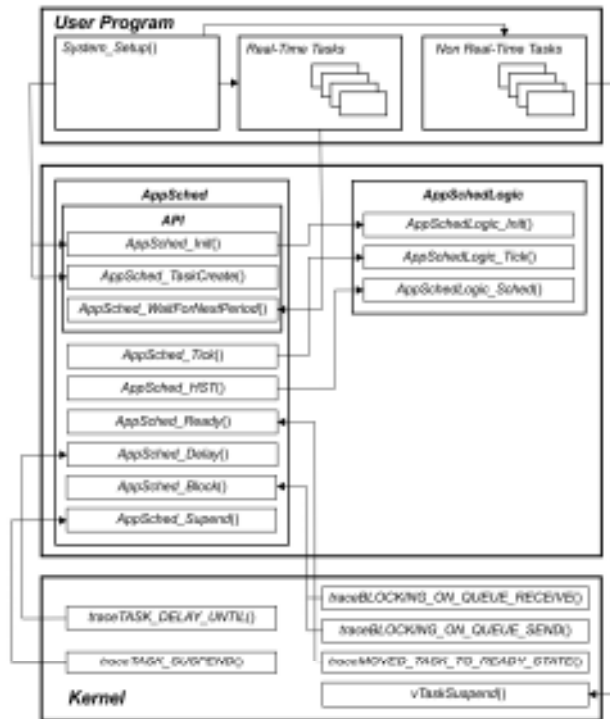


Fig. 4. Interaction between the *HST* modules and the *FreeRTOS* kernel.

B. Blocking and Suspension of tasks

The *HST* scheduler takes into account the blocking of a task when its required shared resources are not available. When a task is blocked, the *AppSched_Block()* function releases the *HST*. Then the *HST* removes the task from its ready list and chooses a new ready task for execution.

When a shared resource is either released or the task blocking timeout expired, *FreeRTOS* inserts the task in the ready list. Then the *AppSched_Ready()* function activates the *HST* and executes the scheduler.

In a similar way, when a task is suspended, the *HST* is released, and removes the task from its ready list. Then the *HST* chooses a new task for execution.

C. AppSched_Logic module

The scheduling policy is implemented with the `AppSchedLogic_Init()`, `AppSchedLogic_Tick()` and `AppSchedLogic_Sched()` functions. The next section shows an example of how these functions may be configured according to specific features of the application.

IV. EXAMPLE: RATE MONOTONIC SCHEDULING WITH A SLACK STEALING METHOD

In a Fixed Priority discipline, a priority is assigned to each task and it remains fixed during runtime. There exist several alternatives to assign different priorities to a set of *RTTs*. *Rate Monotonic (RM)* is a fixed priority scheduling discipline in which tasks with smaller periods are assigned with higher priorities [17]. *RM* is the optimal fixed priority assignment in the sense that if a real-time system (*RTS*) is not schedulable under a *RM* assignment it will be not schedulable under any other fixed priority assignment.

On the other hand, a *Slack Stealing (SS)* method determines the system time that leaves idle the *RTTs* [18, 19, 20, 21]. Adequately determined, this idle time may be used to execute in advance *NRTTs* without jeopardize the *RTT* sub-set schedulability.

In this section, the `AppSchedLogic` module is configured to implement a *RM* discipline for the *RTTs* and a *SS* mechanism to execute the *NRTTs*.

This example implements the *SS* method presented in [22]. The *Available Slack* of a *RTT* is defined as the maximum amount of time that a task i could be delayed without a deadline miss for a particular instant t , and is denoted as $AS_i(t)$. The *Available Slack of the System* for an instant t , $AS(t)$, is defined as the minimum $AS_i(t)$ of the *RTT* sub-set.

The $AS_i(t)$ is stored in a structure called `Task_Slack`, along other parameters required by the *SS* method, such as the *worst case response time (WCRT_i)*, the absolute deadline and maximum delayed finalization instant of its next release. A `Task_Slack` structure is associated to the *eTCB* of each *RTT*.

The module keeps the $AS(t)$ as a global variable. A lower limit AS_{min} is defined, at which the *RTT* should always be executed prior to the *NRTTs*. In this way, AS_{min} determines a safety level for the *RTT* sub-set, in order to avoid jeopardizing its schedulability, at expenses of the *NRTTs* quality of service. Also, a pointer to the current running task is kept.

The `AppSchedLogic` module keeps updated two ready lists: the first one for *RTTs* and the second one for *NRTTs*. These lists are implemented as double-linked ordered lists, using the `List_t` type defined by `FreeRTOS`. Each item in these lists contains a reference to an *eTCB*. These lists are kept priority-ordered, from higher to lower priority. As all *NRTTs* have the same priority, the non-real-time ready list behaves as a *FIFO* queue. To ease the *SS* method implementation, a third list containing all the *RTTs*, ready and suspend, is also kept.

The `AppSchedLogic_init()` function initializes the two ready lists. For each *RTT*, it associates the corresponding `Task_Slack` structure to the *eTCB*, and determines in the *starting time* $AS_i(0)$. Then it calculates $AS(0)$.

The `AppSchedLogic_Tick()` function verifies whether a *NRTT* is executing or the system is *idle*. When either of these two conditions happens, the $AS_i(t_c)$ counter of every *RTT* is decremented. When a *RTT* task is executing increments its processing time ($c_i = c_i + 1$), and decrements the $AS_i(t_c)$ counter of all the higher priority *RTTs*. The function then verifies the *RTTs* deadlines ($t_c \leq D_i$) and call a *hook* function for each task that missed its deadline. Finally, if $AS(t_c) = AS_{min}$ and a *NRTT* is executing, the *HST* is activated. The *HST* will suspend the *NRTT* and run the maximum priority *RTT* ready to execute.

The scheduling algorithm is implemented in the `AppSchedLogic_Sched()` function. It first verifies if the current *RTT* release has ended. In that case the $AS_i(t_c)$ is calculated. If the *RTT* release execution required less computation time than its C_i , this difference is added to all lower priority $AS_i(t_c)$ counters. Then updates the $AS(t_c)$. If $AS(t_c) > AS_{min}$, the first *NRTT* in the ready list, if any, is activated. If $AS(t_c) = AS_{min}$, then the running *NRTT*, if any, is suspended. Otherwise, the maximum priority *RTT* ready to run is executed.

A *NRTT* is released through an *ISR*. From this routine, the *NRTT* is added to the non-real-time ready list, and the *HST* is unblocked.

V. EXPERIMENTAL RESULTS

The example in section IV was implemented using `FreeRTOS` v8.2.1 running on an *mbed* LPC1768 microcontroller (ARM Cortex-M3). A port-specific method for selecting the next task to execute was used, which is more efficient than the generic one. In order to reduce the *CS* overhead, the stack overflow detection methods were disabled.

For the evaluation, real time systems (*RTS*) of 10 *RTTs* each, were generated with *UFs* of 70%, 75%, 80%, 85% and 90%, respectively ([23]). Each *UF* group consists of 100 *RTS* with uniform distributed periods and execution times between 25-1000 *ticks*. The length of each *tick* is 1 *ms*. The *CS* cost for the finalization of a release was measured in *CPU* cycles. The cycles were counted by means of the *clock cycle counter (CYCCNT)*, available as part of the *DWT (Data Watchpoint and Trace)* features of the Cortex-M3. In order to perform a fair comparison with the `FreeRTOS` scheduler, the cost of the *SS* method was not accounted. For each *RTT*, the cost of the first 10 *CS* was averaged. Fig. 5 presents the results.

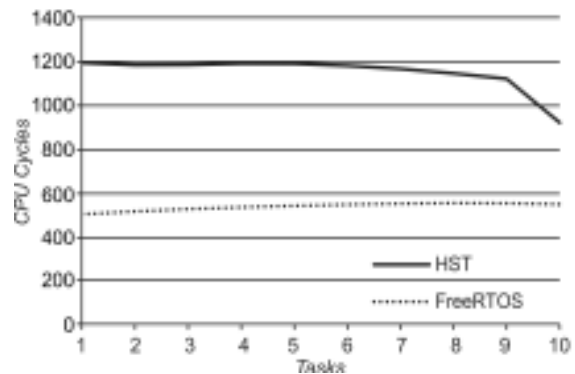


Fig. 5. Average *CS* cost in *CPU* cycles for each task, with and without the *HST*.

As Fig. 5 shows, the average *CS* cost decreases for lower priority tasks when the *HST* is used. This happens because the *HST* suspends any new task arrival with a lower priority than the running task one. For higher priority tasks it is more likely that new arrivals had happened. As the priority of a task decreases, the chances of other arrivals at the end of its execution are lower. In consequence, the *HST* doesn't have to suspend new arrivals as often.

The clock cycles required for the execution of the *HST* after an instance finalization is slightly more than twice of the required by the *FreeRTOS* scheduler alone. Furthermore, it shows little variation with respect the *UF*. In the evaluations the increased *CS* time have not affected the overall real-time response of the tasks.

VI. CONCLUSIONS AND FUTURE WORK

A user-level scheduler support for *FreeRTOS* is presented. It gives the system designer more flexibility for developing new scheduling methods, especially for managing mixed critical systems, without an excessive overhead. As an example of this capability, an implementation of a *RM* user-level scheduler which manages *NRTT* with a *Slack Stealing* mechanism was developed. As future work, the proposed architecture will be extended with support for concurrent user-level schedulers and hierarchy scheduling strategies. Also, new scheduling algorithms will be implemented, evaluating their costs and feasibility within the framework of the *HST*.

AVAILABILITY

Source code and documentation can be downloaded from the UNPSJB *Real Time Systems Group* website: <http://www.rts.g.ump.edu.ar/>.

ACKNOWLEDGMENT

We want to thanks Richard Barry, for his advices in order to improve this work, the anonymous reviewers for their comments, and the ARM University Program, which provided the *mbed* microcontrollers.

REFERENCES

- [1] Alan Burns and Robert Davis, "Mixed criticality systems-a review," 2013.
- [2] S. Vestal, "Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance," in Real-Time Systems Symposium, 2007. RTSS 2007. 28th IEEE International, 2007, pp. 239-243.
- [3] M. A. Rivas and M. G. Harbour, "Posix-compatible application-defined scheduling in marte os," in Real-Time Systems, 2002. Proceedings. 14th Euromicro Conference on, 2002, pp. 67-75.
- [4] Mario Aldea Rivas and Michael González Harbour, "A new generalized approach to application-defined scheduling," in Proceedings of 16th Euromicro Conference on Real-Time Systems (WiP), Catania, Sicily (Italy), 2004.
- [5] R. Inam, J. Maki-Turja, M. Sjodin, S. M. H. Ashjaei and S. Afshar, "Support for hierarchical scheduling in freertos," in Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on, 2011, pp. 1-10.
- [6] R. Inam, M. Sjodin and R. J. Brfi, "Implementing hierarchical scheduling to support multi-mode system," in Industrial Embedded Systems (SIES), 2012 7th IEEE International Symposium on, 2012, pp. 319-322.
- [7] Arnoldo Diaz, Ismael Ripoll, P Balbastre and A Crespo, "A new application-defined scheduling implementation in rtlinux," in Proceedings of the Sixth Real-Time Linux Workshop, 2004.
- [8] A. Diaz, I. Ripoll and A. Crespo, "A library framework for the posix application-defined scheduling proposal," in Electrical and Electronics Engineering, 2005 2nd International Conference on, 2005, pp. 21-26.
- [9] Li Peng, B. Ravindran, S. Suhaib and S. Feizabadi, "A formally verified application-level framework for real-time scheduling on posix real-time operating systems," Software Engineering, IEEE Transactions on, vol. 30, N° 9, pp. 613-629, 2004.
- [10] M. Asberg, T. Nolte, S. Kato and R. Rajkumar, "Exsched: An external cpu scheduler framework for real-time systems," in Embedded and Real-Time Computing Systems and Applications (RTCSA), 2012 IEEE 18th International Conference on, 2012, pp. 240-249.
- [11] A. Gomez, L. Schor, P. Kumar and L. Thiele, "SF3p: A framework to explore and prototype hierarchical compositions of real-time schedulers," in Rapid System Prototyping (RSP), 2014 25th IEEE International Symposium on, 2014, pp. 2-8.
- [12] M. Aldea, G. Bernat, I. Broster, A. Burns, R. Dobrin, J. M. Drake, G. Fohler, P. Gai, M. G. Harbour, G. Guidi, J. J. Gutierrez, T. Lennvall, G. Lipari, J. M. Martinez, J. L. Medina, J. C. Palencia, and M. Trimarchi, "Fsf: A real-time scheduling architecture framework," in Real-Time and Embedded Technology and Applications Symposium, 2006. Proceedings of the 12th IEEE, 2006, pp. 113-124.
- [13] G. Parmer and R. West, "Hijack: Taking control of cots systems for real-time user-level services," in Real Time and Embedded Technology and Applications Symposium, 2007. RTAS '07. 13th IEEE, 2007, pp. 133-146.
- [14] G. Muller, H. Duchesne and J. L. Lawall, "A framework for simplifying the development of kernel schedulers: Design and performance evaluation," in Object-Oriented Real-Time Dependable Systems, 2005. WORDS 2005. 10th IEEE International Workshop on, 2005, pp. 219-227.
- [15] E. Eide, T. Stack, J. Regehr and J. Lepreau, "Dynamic cpu management for real-time, middleware-based systems," in Real-Time and Embedded Technology and Applications Symposium, 2004. Proceedings. RTAS 2004. 10th IEEE, 2004, pp. 286-295.
- [16] George M Candea and Michael B Jones, "Vassal: Loadable scheduler support for multi-policy scheduling," in Proceedings of the 2nd USENIX Windows NT Symposium, 1998, pp. 157-166.
- [17] C. L. Liu and James W. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment," Journal of the ACM, vol. 20, N° 1, pp. 46-61, 1973.
- [18] John P. Lehoczky and Sandra Ramos-Thuel, "An optimal algorithm for scheduling soft-aperiodic tasks in fixed-priority preemptive systems," in IEEE Real-Time Systems Symposium, Phoenix, Arizona, EUA, 1992, pp. 110-123.
- [19] Sandra Ramos-Thuel and John P. Lehoczky, "On-line scheduling of hard deadline aperiodic tasks in fixed-priority systems," in Real-Time Systems Symposium, 1993, pp. 160-171.
- [20] Sandra Ramos-Thuel and John P. Lehoczky, "Algorithms for scheduling hard aperiodic tasks in fixed-prioriys systems using slack stealing," in Real-Time Systems Symposium, 1994, pp. 22-33.
- [21] T. S. Tia, J. W. S. Liu and M. Shankar, "Algorithms and optimality of scheduling soft aperiodic requests in fixed priority preemptive systems," The International Journal of Time-Critical Computing Systems, vol. 10, N° pp. 23-43, 1996.
- [22] José M. Urriza, Francisco E. Paez, Ricardo Cayssials, Javier D. Orozco and Lucas Schorb, "Low cost slack stealing method for RM/DM," International Review in Computers and Software (IRECOS), vol. 5, N° 6, pp. 660-667, 2010.
- [23] Gabriela Olguín, Laura Biscayart and José M. Urriza, "Generación de tareas periódicas y aperiódicas para simulación de sistemas de tiempo real," Journal of Industrial Engineering (IJIE), vol. 3, N° 6, pp. 53-69, 2011.

Author Index

Author	page
Bergen, F. von	25
Bezerra , Teles de Sales	19
Cayssials, Ricardo	37
Dujovne, Diego	31
Eleutério , Saulo Aislan da Silva	19
Farías , Angelo Antonio	31
Giribet, Juan	1,25
Gonzalez, Rodrigo	7
Lutenberg, Ariel	13
Martos, P.	25
Moreno, Patricio	1
Orozco, Javier D.	37
Páez, Francisco E.	37
Patiño, Héctor Daniel	7
Pose, Claudio	1
Rodrigues de Sousa, José Anderson	19
Silva Rocha , Jerónimo	19
Sisterna, Cristian	7
Sutter, Gustavo	7
Urriza, José M.	37
Vargas, Fabian	13
Zacchigna, Federico G.	13



www.sase.com.ar

August 12th-14th, 2015
University of Buenos Aires, Argentina

ISBN 978-987-45523-4-1



9 789874 552341

CAS E 2015

CONGRESO ARGENTINO DE **SISTEMAS EMBEBIDOS**

Libro de Trabajos
Foro tecnológico y Posters

Publicación realizada con el apoyo de:
**Agencia Nacional de Promoción
Científica y Tecnológica**



www.sase.com.ar

Facultad de Ingeniería I UBA |
Buenos Aires, Argentina

ISBN 978-987-45523-3-4

CASE 2015

Libro de Trabajos

Modalidades Foro Tecnológico y Póster

Congreso Argentino
de
Sistemas Embebidos

12 al 14 de Agosto de 2015
FIUBA, Buenos Aires, Argentina



Congreso Argentino de Sistemas Embebidos CASE 2015 : libro de trabajos, foro tecnológico y posters / Diego Javier Brengi, José Lipovetzky , Maximiliano Antonelli, Luciana De Micco, Ariel Lutenberg, ... [et.al.]. - 1a ed. - Ciudad Autónoma de Buenos Aires : ACSE - Asociación Civil para la investigación, Promoción y Desarrollo de Sistemas Eléctricos Embebidos, 2015.

100 p. : il. ; 29x20 cm.

ISBN 978-987-45523-3-4

1. Ingeniería. 2. Actas de Congresos. I. Brengi, Diego Javier
CDD 620.711

Fecha de catalogación: 29/05/2015

Libro de Trabajos
Modalidades Foro Tecnológico y Póster
Congreso Argentino de Sistemas Embebidos - CASE 2015

Editores:

Antonelli, Maximiliano. UNMDP
Brengi, Diego. INTI/UNLaM
De Micco, Luciana. UNMDP/CONICET
García Inza, Mariano. FIUBA
Lipovetzky, José. IB/CNEA/CONICET
Lutenberg, Ariel. FIUBA/CONICET/ACSE

Copyright © 2015. 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 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 SASE busca fomentar esta temática realizando las siguientes actividades:

- CASE: Congreso Argentino de Sistemas Embebidos, presentación trabajos científicos.
- Workshops: Talleres prácticos en la modalidad hands-on.
- Tutoriales: Charlas de capacitación.
- Plenarias: Conferencias y debates abiertos.
- Concurso de proyectos tecnológicos: sobre trabajos finales y materias de grado.
- Concurso de emprendimientos tecnológicos: destinado a promover emprendimientos electrónicos con viabilidad económica.
- Programa de equipamiento para universidades: para transferir a las universidades las donaciones de los auspiciantes.
- Becas de viaje y alojamiento: Ayudas económicas para viaje y estadía a estudiantes de grado, estudiantes de doctorado, docentes e investigadores, de Argentina y Latinoamérica.

Los objetivos que persigue el CASE 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.
- Estimular en los estudiantes universitarios avanzados el interés por la investigación en el área de los S.E.
- Difundir los proyectos de investigación mediante el desarrollo de un sitio web.
- Coordinar y actualizar los contenidos de S.E. de los programas de grado y posgrado de las universidades argentinas.

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

Arquitecturas de Procesadores, Bioingeniería, DSPs, FPGAs, HDLs y ASICs, Implementación de Sistemas Embebidos, Protocolos y Comunicaciones, Robótica, RTOS, Software Embebido, Linux embebido y comunicaciones inalámbricas. Dentro de cada una de estas áreas se permiten las modalidades Artículo, Foro Tecnológico y Póster, según el tipo de trabajo.

Los trabajos presentados al CASE fueron sometidos a un proceso de revisión por pares y posterior corrección. De este modo fueron seleccionados 21 trabajos en la modalidad foro tecnológico y 17 pósters.

Esta publicación se encuentra también disponible en forma online en la página www.sase.com.ar.

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

Atentamente,

Comité Organizador CASE

Auspiciantes Diamond

- ELEMÓN
- EMTECH
- INTEL
- CORADIR
- VICDA ARGENTINA
- CIKA
- ASEMBLI
- ELECTROCOMPONENTES

Auspiciantes Platinum

- Probattery
- Synopsys
- Telit Wireless Solutions

Auspiciantes Gold

- INVAP
- Sur Emprendimientos Tecnológicos
- CADIPÉL
- Dai Ichi Circuitos
- Freescale

Auspiciantes Silver

- Ar-Sat
- L&R Ingeniería
- Digi International

Instituciones Auspiciantes

Asociaciones Científicas

- IEEE Argentina (Institute of Electrical and Electronics Engineers)
- IEEE CASS (Circuits and Systems Society)

Cámaras y Asociaciones Industriales

- ADIMRA (Asociación de Industriales Metalúrgicos de la República Argentina)
- CADIEEL (Cámara Argentina de Industrias Electrónicas, Electromecánicas y Luminotécnicas)
- CAME (Confederación Argentina de la Mediana Empresa)
- CIECCA (Cámara de Industrias Informáticas, Electrónicas y de Comunicaciones del Centro de Argentina)

Instituciones Científicas y Tecnológicas

- CoNAE (Comisión Nacional de Actividades Espaciales)
- CONICET (Consejo Nacional de Investigaciones Científicas y Técnicas)
- INTI (Instituto Nacional de Tecnología Industrial)
- Fundación Sadosky (Investigación y Desarrollo en TIC)
- Fundación Fulgor

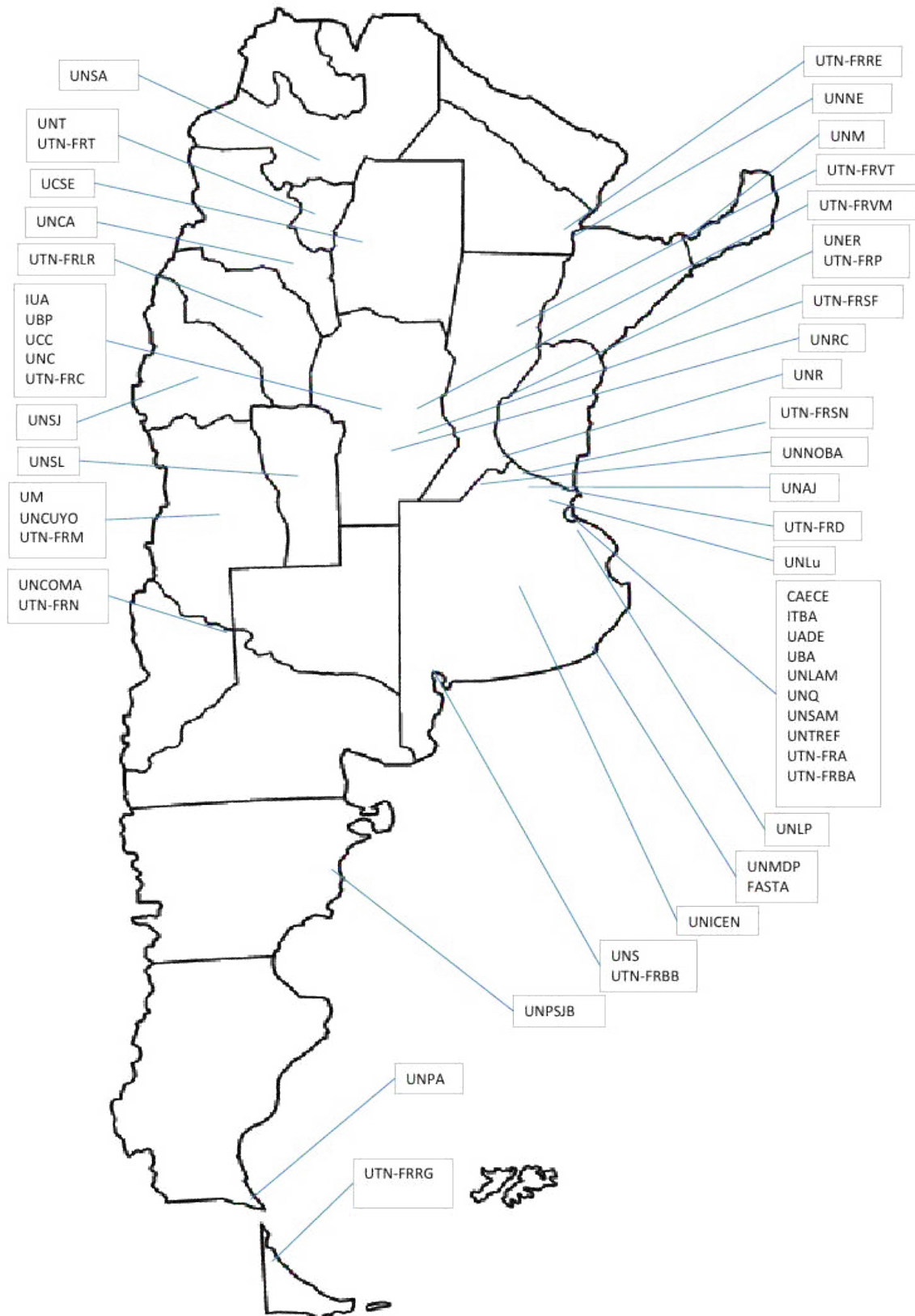
Ministerios Nacionales

- MinCyT (Ministerio de Ciencia, Tecnología e Innovación Productiva)
- ANPCyT (Agencia Nacional de Promoción Científica y Tecnológica)
- Ministerio de Industria

Redes Universitarias

- CONFEDI (Consejo Federal de Decanos de Ingeniería)
- RUSE (Red Universitaria de Sistemas Embebidos)

Universidades Auspiciantes



Coordinación General SASE

- Dr. Ariel Lutenberg (FIUBA/CONICET/ACSE)

Coordinación CASE

- Dr. José Lipovetzky (IB/CNEA/CONICET)
- Dra. Luciana De Micco (UNMDP/CONICET)
- Mg. Diego Brengi (INTI/UNLaM)
- Ing. Maximiliano Antonelli (UNMDP)
- Dr. Mariano García Inza (FIUBA)

Chairs

- Bioingeniería: Ing. Juan Manuel Reta (UNER)
- DSPs: Dra. María Liz Crespo (ICTP)
- Linux Embebido: Ing. Lucas Chiesa (FIUBA)
- Software Embebido: Dr. Ricardo Medel (Ascentio Technologies)
- FPGAs, HDLs y ASIC: Ing. Salvador Tropea (INTI/UTN-FRBA)
- Implementación de Sistemas Embebidos: Msc. Cristian Sisterna (UNSJ)
y Dr. Gustavo Sutter (UAM)
- Arquitectura de procesadores: Ing. Alejandro Furfaro (UTN-FRBA)
- Comunicaciones y protocolos: Ing. Gustavo Mercado (UTN-FRM)
- Robótica: Dr. Luis Canali (UTN-FRC)
- RTOS: Dr. Ricardo Cayssials (UNS)
- Comunicaciones inalámbricas: Dr. Leonardo Rey Vega (FIUBA)

Revisores

Alcalde Bessia, Fabricio	Lutenberg, Ariel
Alessandrini, Gustavo	Marchi, Edgardo
Antonelli, Maximiliano	Martos, Pedro
Arias, Ricardo	Mas, Ignacio
Brenji, Diego	Mata, Walter A.
Briff, Pablo	Medel, Ricardo
Burgos, Enrique Sergio	Melo, Rodrigo
Calarco, Nicolás	Monte, Gustavo
Canali, Luis Rafael	Oliva, Rafael
Carbonetto, Sebastián	Orallo, Carlos Martin
Carrá, Martín	Perez Paina, Gonzalo F.
Cayssials, Ricardo	Perez, Martín
Cogo, Jorge	Perez, Santiago
Crespo, Maria Liz	Petrashin, Pablo Antonio
De Micco, Luciana	Pose, Claudio
Escudero, Gustavo	Pucheta, Julian
Ferrao, Hilda	Puga, Gerardo Ludovico
Ferreyra, Pablo Alejandro	Rey Vega, Leonardo
Filomena, Eduardo	Reyes, Benjamin
Fraire, Juan Andres	Ridolfi, Pablo
Furfaro, Alejandro	Rocha, Fábio
Garcia Inza, Mariano	Romeo, Marcelo
Garcia, Javier	Roncagliolo, Agustín
Gavinowich, Gabriel	Sambuco Salomone, Lucas
Gayoso, Carlos Arturo	Sisterna, Cristian
Giribet, Juan Ignacio	Sofo Haro, Miguel
Grimblatt, Victor	Sutter, Gustavo
Gwirc, Sergio N.	Taffernaberry, Carlos
Hernandez Tabares, Lorenzo	Todorovich, Elías
Juarez, Gustavo	Tropea, Salvador
Leiva, Lucas	Weisz, Rosa María
Lipovetzky, Jose	Zacchigna, Federico G.
Lozano, Clevis	Zaradnik, Ignacio

ÍNDICE

DSP		PAG
Control V/f de un motor trifásico asíncrono		3
Convertor Analógico-Digital Sigma-Delta implementado sobre FPGA		8
Comparison of Isolated Speech Recognition Algorithms with a Cortex-M4 Processor		13
Inversor Sinusoidal con Control Embebido		15
FPGAs, HDLs y ASICs		PAG
Savitzky-Golay Filter design in FPGA described in VHDL		19
FPGA-based embedded system for remote control of home devices		25
Comparison of classical and chaotic sequences for DSSS: hardware implementation		31
Microprocesador embebido en FPGA para el tratamiento de ablación por radiofrecuencia en el hígado		37
Robótica		PAG
Módulo Tacómetro para Motores Brushless		41
Implementación de un Sistema de Navegación a Robots Móviles		46
Implementación de Sistemas Embebidos		PAG
Sistema de alarma de bajo costo controlado remotamente por dispositivos móviles		49
Sistema de Adquisición de Datos Sísmicos		55
Controlador de acceso mediante tarjeta inteligente		61
Equipo de bajo costo para la construcción de circuitos impresos mediante el uso de luz ultravioleta		63
Diseño y Desarrollo de un Podómetro Detector de Cielo para Monitoreo de Ganado Bovino		68
Sistema de transferencia automática para grupos electrógenos		74
Generador de claves automáticas utilizando Smartphone		79
Bascula Inclínada usando MBED		80
Unidad inercial de bajo costo asistida por GPS		81
Geige-Müller counter implementation with an 8-bit microcontroller		82
Diseño de una Tarjeta de Adquisición y Transmisión de Datos Para Comunicación Software-Hardware en Simuladores		83
Sonda de temperatura digital para uso agrícola		85
Implementación práctica de mecanismos de calibración en sistemas embebidos - Caso de medición de curva de potencia de pequeños aerogeneradores		86
Módulo didáctico para medición de presión hidrostática		88
Software Embebido		PAG
Controlador para actuador de posición		93
Digital Sound Meter		99
Real-Time Iris-Tracking Embedded System		104
uPOSIX: Una biblioteca POSIX para microcontroladores		108
Indoor UAV tracking using image processing		113
Generador de Señales de alta frecuencia y Qmetro inductivo		117
Protocolos, comunicaciones y redes inalámbricas		PAG
Sensorística para el control de Plataforma Robótica destinada al estudio de Redes de Sensores Móviles		121
Design of DTMB IP modules		127
Diseño y evaluación de un Sistema de Adquisición y Almacenamiento de Datos Ambientales con Transmisión Satelital		132
Wireless Sensor Network for Monitoring Infrastructure Adjacent to Metro de Medellín Railway System		138
Aplicación de Sistemas Embebidos para Control y Monitoreo Remoto Aplicado a Colectores Solares		144
Sistema de adquisición y postprocesamiento de parámetros de vuelo de vectores		145
Prototype of ZigBee RSSI Measurer Using Arduino		146
Seguimiento de Objetivos en Redes de Sensores Multimedia mediante Estimación Distribuida		147

Foro Tecnológico

Pósters

DSPs

Control V/f de un motor trifásico asíncrono

Alejandro Nuñez Manquez
Departamento de Física
Universidad Nacional de San Luis
San Luis, Argentina
Email: janyo12@unsl.edu.ar

Abstract—En este trabajo se describe la implementación de un control V/f sobre un motor trifásico asíncrono utilizando las librerías de Control Suite de Texas Instrument y la placa TMS320C2000TM Experimenter Kit Overview. Este kit tiene conectada la placa delfino, la cual posee el microcontrolador TMS320F28335 [6].

Keywords—Control, DSP, V/f, microcontrolador.

I. INTRODUCTION

Los motores de inducción con rotor de jaula de ardilla son las principales herramientas de la industria debido a su bajo costo y su construcción resistente. Cuando se opera desde los voltajes de línea, un motor de inducción trabaja casi con una velocidad constante. Sin embargo, por medio de convertidores de electrónica de potencia es posible variar la velocidad de un motor de inducción. Los accionamientos por motor de inducción se clasifican en dos categorías amplias con base en sus aplicaciones:

- 1) **Accionamientos de velocidad ajustable:** Una aplicación importante de estos accionamientos se encuentra en el control de procesos, donde controlan velocidad de ventiladores, compresores, bombas, soldadores, etc.
- 2) **Servoaccionamientos:** Por medio de un control avanzado, los motores de inducción se pueden usar como servoaccionamientos en periféricos de computadoras, máquinas herramientas y robótica[1].

A. Motor de inducción trifásico

Un motor de inducción trifásico está formado por un estator de inducción que consiste en tres devanados de fase distribuidos en las ranuras del estator. Estos tres bobinados están desplazados 120° en espacio uno con respecto del otro. El rotor en jaula de ardilla consiste en una pila de laminados aislados. A través de él se encuentran insertadas barras de conducción eléctrica, cerca de la periferia en el sentido axial, las cuales están en cortocircuito eléctrico en cada extremo del rotor por medio de anillos de extremo, lo que constituye una estructura parecida a una jaula.

En la Fig. 1 se puede observar las características constructivas de un rotor de jaula de ardilla. Y en la Fig. 2 se pueden observar las velocidades de giro de los flujos magnéticos del rotor y del estator, como así también la velocidad de giro del rotor.

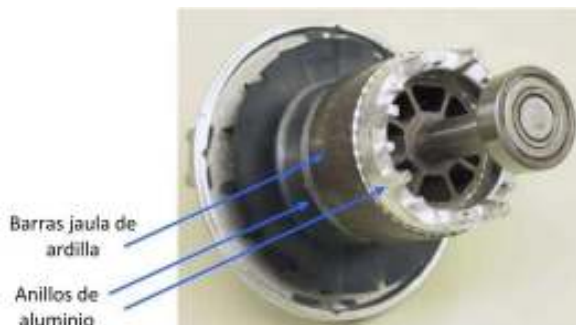


Fig. 1. Estructura del rotor de un motor asíncrono.

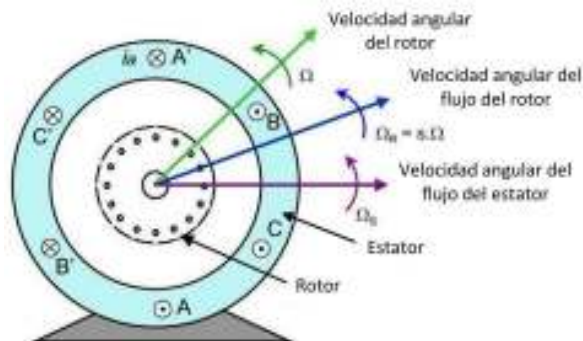


Fig. 2. Velocidad de giro del rotor y de los campos magnéticos en un motor asíncrono.

II. CONTROL VOLTAJE/FRECUENCIA

El control por medio de la variación de frecuencia (y voltaje) permite tener la capacidad de operar el motor no sólo con velocidades inferiores a la velocidad nominal, sino también superiores a la velocidad nominal. Esta capacidad es muy atractiva en muchas aplicaciones, pues la mayoría de los motores de inducción, debido a su robusta construcción, se opera hasta el doble de la velocidad nominal sin ningún problema mecánico. Por debajo de la frecuencia nominal, al mantener la relación V/f constante, se mantiene constante el par del motor. Cuando se supera la frecuencia nominal lo que se mantiene constante es la potencia siendo el torque variable en esta zona de trabajo, Fig. 3.

En este trabajo el sistema de control de velocidad que se ha diseñado trabaja en la zona lineal (región de par de torsión constante), es decir, se mantiene constante la relación entre tensión de alimentación y frecuencia de la señal trifásica de alimentación sin sobrepasar los valores nominales. Esto

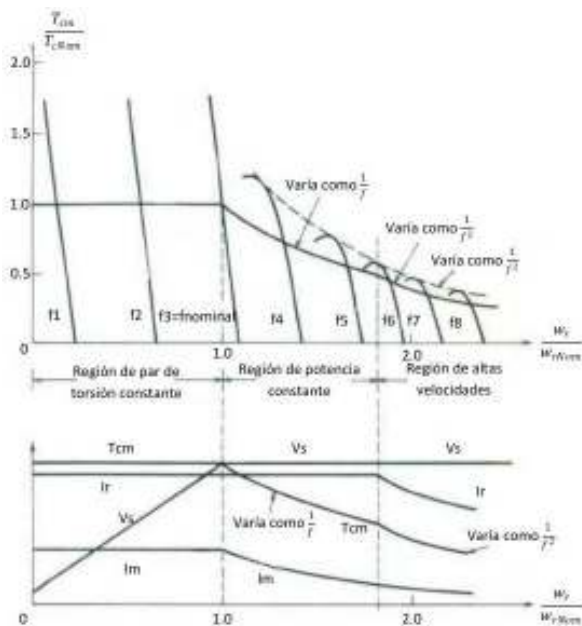


Fig. 3. Curvas características del motor de inducción [1].

permite mantener el flujo magnético constante sin saturar el núcleo magnético. Una justificación matemática se puede deducir de la ley de Faraday, ecuación (1).

$$v(t) = -N \frac{d\phi}{dt} \quad (1)$$

Si se aplica un voltaje $v(t) = V_m \sin(\omega t)$ al núcleo, el flujo ϕ resultante es:

$$\phi(t) = \frac{1}{N_p} \int v(t) dt = \frac{1}{N_p} \int V_M \sin(\omega t) dt \quad (2)$$

$$\phi(t) = -\frac{V_M}{\omega N_p} \cos(\omega t) \quad (3)$$

La frecuencia eléctrica aparece en el denominador de la ecuación 3. Entonces, si se mantiene constante la relación entre la tensión y la frecuencia eléctrica aplicada al estator, el flujo en el núcleo del motor se mantiene constante, al igual que la corriente de magnetización, manteniendo el par motor constante. Esto permite mantener el torque aproximadamente constante entre $0.2 * V_{nominal}$ y $V_{nominal}$. Si se mantiene constante la tensión y se aumenta la frecuencia de alimentación por encima del valor nominal, el torque comienza a disminuir como se observa en la Fig. 3.

III. CONFIGURACIÓN DEL PROYECTO

El proyecto está formado por los distintos módulos que se pueden observar en la Fig. 4, los cuales son: DSP, Inversor, Motor Trifásico, Encoder, Bus VCC, Fuente Conmutada Auxiliar y Adaptación de Señales.

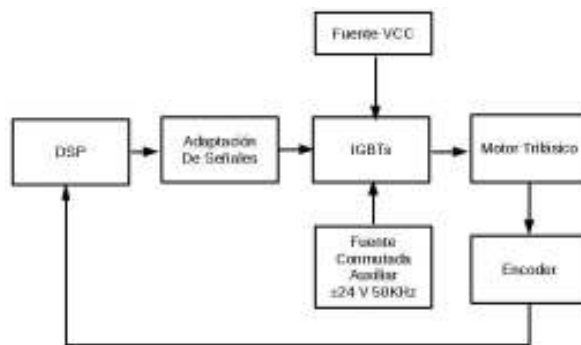


Fig. 4. Diagrama esquemático del sistema.

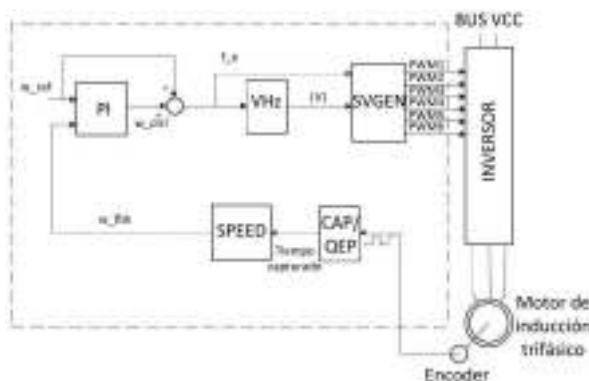


Fig. 5. Diagrama de bloques de la implementación del driver para control V/f de un motor de inducción trifásico

A. Implementación del Control V/f en el DSP

Para la implementación del control V/f se han utilizado las librerías dadas por Texas Instrument a través de su Control Suite [4]. Estas librerías permiten un rápido prototipado y puesta a punto del software que se va a cargar en el DSP. En este trabajo en particular se ha utilizado como guía la nota de aplicación [5] que indica los pasos a seguir para implementar este tipo de control. En la Fig. 5 se puede observar un diagrama de bloques con las librerías usadas perteneciente a esta guía. Para la compilación del código se ha usado Code Composer Studio V6 [7].

Los bloques que están encerrados por la línea punteada (Fig. 5) corresponden a los bloques software con los cuales se ha programado el DSP. Lo que está fuera de la línea punteada corresponde al inversor, al motor trifásico y al encoder utilizados.

- **Bloque PI:** este bloque define un controlador PI y es el encargado de generar una señal que permita ajustar la velocidad de salida del motor con la velocidad seteada. Esta velocidad se calcula internamente en por unidad (pu).
- **Bloque V/Hz:** Este bloque calcula el módulo de la tensión a aplicar al motor en función del valor de velocidad que recibe.
- **Bloque SVGEN:** Este bloque calcula los coeficientes que son utilizados para generar las señales PWM que generarán las tensiones del estator utilizando

la técnica de SVPWM (Space Vector Pulse Width Modulation[1][2]).

- **Bloque SPEED:** Este bloque calcula la velocidad usando como dato el tiempo capturado por el bloque CAP/QEP.
- **CAP/QEP:** Este bloque está conectado directamente con un módulo hardware del DSP que permite capturar las señales del encoder en forma independiente del procesador.

Este sistema trabaja con interrupción de un timer que permite ajustar la frecuencia del PWM, la cual es de 20 KHz. El flujo de ejecución del software es como sigue:

Dentro del main

- 1) Se inicializan los módulos software.
- 2) Se inicializan las bases de tiempo.
- 3) Se habilita la base de tiempo CNT_zero de interrupción del EPWM1 y las interrupciones globales.
- 4) Se inicializan otros sistemas y parámetros de otros módulos en el caso de que sea necesario.
- 5) Se ingresa en un bucle infinito para esperar las interrupciones del EPWM1.

Dentro de la interrupción EPWM1

- 1) Se guarda el contexto y se limpian las banderas de interrupción.
- 2) Se ejecuta el módulo PI.
- 3) Se ejecuta el módulo V/Hz.
- 4) Se ejecuta el módulo SVGEN.
- 5) Se ejecuta el módulo CAP/QEP.
- 6) Se ejecuta el módulo SPEED.
- 7) Se cargan los valores calculados al módulo PWM.
- 8) Se restaura el contexto.
- 9) Se retorna al bucle infinito.

Los bloques mostrados en la Fig. 5 se describe a continuación.

B. Inversor

Un inversor trifásico es un circuito que convierte corriente continua (CC) en corriente alterna (CA) de tres fases. Esto se hace mediante la conmutación de dispositivos semiconductores que lo componen, los cuales actúan como llaves. Actualmente son muchas las topologías que se utilizan. En este trabajo se ha utilizado la topología VSI (*Voltage Source Inverter*)[2], Fig. 6.

La naturaleza de este circuito impone dos reglas que garantizan su correcto funcionamiento, a saber:

- Las llaves de una misma pierna no pueden estar cerradas al mismo tiempo, ya que esto produciría un cortocircuito en la bus de CC.
- Las llaves de una misma pierna no pueden estar abiertas al mismo tiempo, ya que esto dejaría que la tensión en el punto medio de la pierna dependa del sentido de circulación de la corriente por la carga.

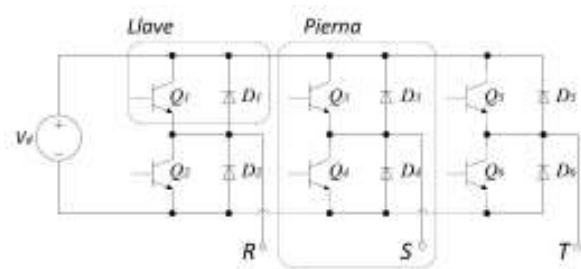


Fig. 6. Diagrama esquemático de un inversor trifásico.

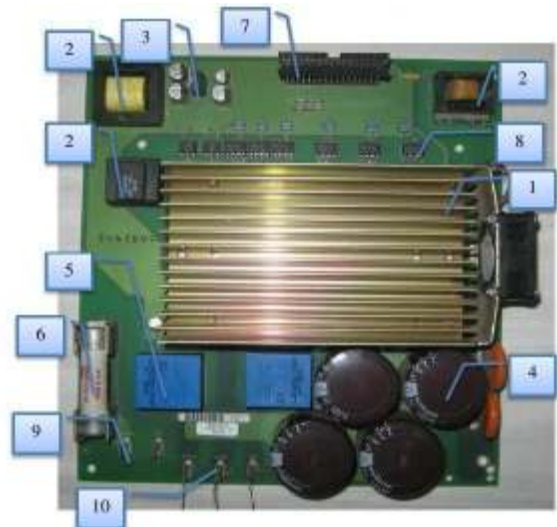


Fig. 7. Placa inversora.

Las reglas propias de estos sistemas ya están contempladas en las librerías que propone Texas Instrument. En este caso es el bloque pwm de la librería "f2833xpwm.h" el que se encarga de que en ningún momento estén cerradas o abiertas al mismo tiempo dos llaves de una misma pierna.

En este trabajo el inversor que se usó se muestra en la Fig. 7, el cual posee un módulo PM25RSK120 marca Mitsubishi, Fig. 8.

En la Fig. 7 se puede ver el disipador (1), debajo del cual se encuentra el módulo PM25RSK120, sus respectivas fuentes de alimentación (2), una fuente de 5 V (3), filtros del bus CC (4), circuitos de monitoreo (5), protección (6), conectores para el control de llaves (7), optoacopladores (8), conectores para el bus de CC (9) y para las líneas de salida de la señal trifásica (10). El módulo PM25RSK120, es un módulo inteligente de potencia compuesto por siete llaves IGBT con sus respectivos circuitos de activación y protecciones.

C. Bus de VCC

Para el bus VCC se ha utilizado una fuente doble puesta en serie, la cual entrega una tensión de 60 V y 3 A máximos. Esto permite realizar las pruebas del proyecto sin correr riesgos.

Como el motor utilizado funciona con una tensión trifásica de 380 V y 50 Hz, se tuvo que calcular la nueva frecuencia máxima de trabajo, suponiendo que el motor va a trabajar sin carga.



Fig. 8. Módulo: PM25RSK120, Marca: Mitsubishi.



Fig. 9. Motor trifásico y encoder acoplado a su eje.

En el caso de este trabajo y debido a que el motor trabaja en vacío la frecuencia máxima de alimentación se fija a 20 Hz.

D. Fuente Auxiliar

La placa inversora necesita para su funcionamiento una fuente que entregue una tensión de onda cuadrada de $\pm 24V$ a 50 KHz.

E. Motor trifásico

Se utilizó un motor trifásico conectado en estrella modelo TE1BFOXO marca WEG tipo W22 [8], como se observa en la Fig. 9.

F. Encoder

Para medir las RPM desarrolladas por el motor trifásico se ha utilizado un encoder óptico incremental ENB-200-3-1. Este encoder tiene como salida las señales A, B y Z en totem pole, con una resolución de 200 pulsos por revolución.

IV. PRUEBAS

Para las pruebas del sistema se usaron las indicaciones de la nota de aplicaciones [5]. La primera de ella fue probar el código a lazo abierto para ver si las señales PWM correspondían con las esperadas. Se graficó la diferencia entre las señales Ta, Tb y Tc que entran al bloque pwm. El resultado que se obtuvo se muestra en la Fig. 10

En la Fig. 10 se puede observar que para una tensión $0.3 * V_{nom}$ la frecuencia es un 30% de la frecuencia nominal. En

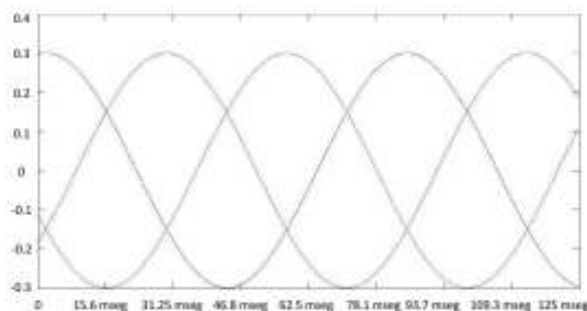


Fig. 10. Señales Ta-Tb, Tb-Tc y Tc-Ta al 30% de la tensión nominal

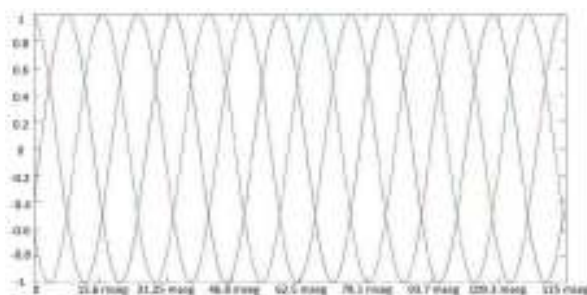


Fig. 11. Señales Ta-Tb, Tb-Tc y Tc-Ta a la tensión nominal

el caso de la Fig. 11 la tensión aplicada es la nominal, por lo tanto la frecuencia es la nominal. Con esto queda demostrado que el sistema cumple con los requerimientos.

A. Ajuste del controlador PI

El módulo PI responde a la ecuación 4

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau \quad (4)$$

en donde

- $u(t)$ es la salida del PID
- $e(t)$ es el error entre la referencia y la retroalimentación
- K_p es la ganancia proporcional del PID
- K_i es la ganancia integral del PID

El módulo PI se ha configurado siguiendo los siguientes pasos:

Paso 1.

Establecer la ganancia integral $K_i = 0$ y la ganancia proporcional $K_p = 1$.

Paso 2.

Mientras se le inyecta una señal escalón a la entrada, ajustar poco a poco la variable de ganancia proporcional K_p , hasta alcanzar el tiempo de subida y overshoot óptimos.

Paso 3.

Si es necesario, aumentar gradualmente la ganancia integral K_i para optimizar el retorno de la salida de estado estacionario nominal. El controlador será muy sensible a este término y se puede volver inestable así que asegúrese de comenzar con un

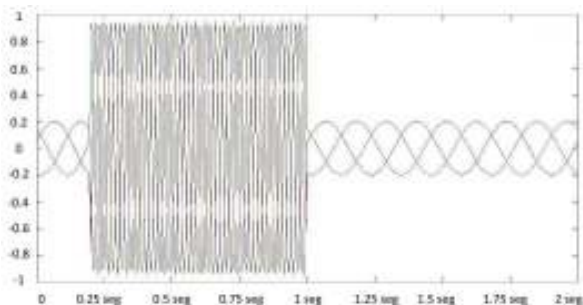


Fig. 12. Señales Ta-Tb, Tb-Tc y Tc-Ta dada una entrada escalón

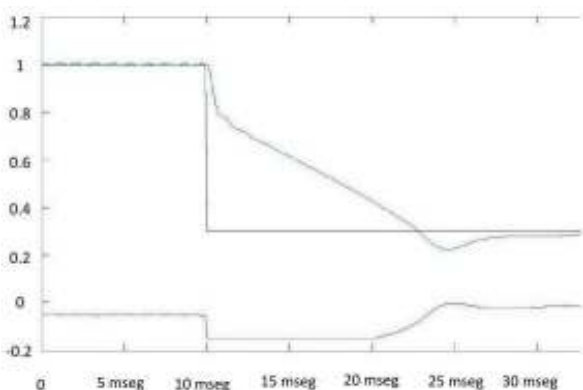


Fig. 13. Velocidad, salida del controlador y señal escalón cuando el escalón es descendente

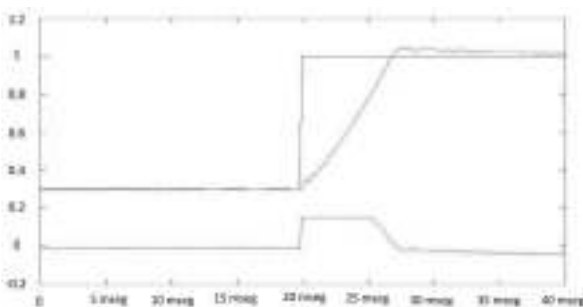


Fig. 14. Velocidad, salida del controlador PI y señal escalón cuando el escalón es ascendente

número muy pequeño. La ganancia integral se traducirá en un aumento de sobreimpulso y de oscilación, por lo que puede ser necesario disminuir ligeramente el término K_p de nuevo para encontrar el mejor equilibrio.

B. Pruebas finales

Para la prueba del sistema se desarrolló una señal tipo escalón que variaba entre 0.3 y 1. Esta señal tiene un periodo de 5 segundos. Los resultados se muestran en las Fig. 12, 13 y 14.

V. CONCLUSIONES

En este trabajo se ha logrado realizar un control V/f usando las librerías propias de Texas Instrument. Se logró modificar las librerías y acondicionarlas para el **DSP TMS320F28335**.

Este DSP puede trabajar directamente con datos tipo float, lo cual permitió modificar las librerías y pasarlas a float. En las pruebas realizadas también se aumentó la carga aplicada al eje del motor y se comprobó que el sistema buscaba ajustar la velocidad de acuerdo con la velocidad de referencia. También se frenó el motor hasta pararlo, y este volvía a su estado de régimen a la velocidad de referencia.

REFERENCES

- [1] Ned Mohan, Tore M. Undeland, William P Robbins. *Electrónica de Potencia. Convertidores, aplicaciones y diseño*. McGraw Hill. Tercera Edición
- [2] Muhammad Rashid. *Electrónica de Potencia. Circuitos, dispositivos y aplicaciones*. Pearson Prentice Hall. Tercera Edición.
- [3] Alfonso Álzate, Duberney Murillo Yarce, Marcela González Valencia. "Control de velocidad mediante relación voltaje-frecuencia". ISSN 0122-1701
- [4] <http://www.ti.com/tool/controlsuite>
- [5] Scalar (V/f) Control of 3-Phase Induction Motors. Application Report. <http://www.ti.com/lit/an/sprabq8/sprabq8.pdf>
- [6] <http://www.ti.com/product/TMS320F28335>
- [7] Code Composer Studio <http://www.ti.com/tool/ccstudio>
- [8] <http://www.weg.net/w22/index-es.php?market=am-latina>

Conversor Analógico-Digital Sigma-Delta implementado sobre FPGA

Andrés Julio Demski

Laboratorio de Procesamiento Digital - DPLAB
Universidad Tecnológica Nacional - FRBA
Buenos Aires, Argentina
ademski@frba.utn.edu.ar

Abstract—El objetivo de este trabajo fue desarrollar un conversor Analógico-Digital Sigma Delta para señales de baja frecuencia de una forma simple y de fácil implementación. Las bases de funcionamiento de este conversor son dos técnicas de procesamiento de señales, *noise shaping* y *oversampling*. Con las cuales se busca disminuir la potencia de ruido en las frecuencias de interés. Tomando en cuenta la gran influencia que tienen las FPGA (*Field-Programmable Gate Array*) en el procesamiento digital de señales, se buscó la forma de integrar la etapa de adquisición dentro de la misma para así poder prescindir de dispositivos externos que realicen esta tarea. Como resultado del trabajo, utilizando una FPGA Spartan 6 (XC6SLX9), se obtuvo un ADC de 8 bits (6.56 efectivos) con una frecuencia de muestreo de 2KHz, rango de 5v y comportamiento prácticamente lineal.

Keywords—FPGA, ADC, Sigma Delta, Oversampling, Noise Shaping

I. INTRODUCCIÓN

Un FPGA (Field Programmable Gate Array) es un dispositivo digital programable con una arquitectura diseñada especialmente para la implementación de circuitos sincrónicos. Uno de los usos más comunes de esta plataforma es el procesamiento de señales para el cual posee bloques específicos de *hardware* para optimizar esta tarea. Uno de las dificultades que presenta esta plataforma es la manera por la cual son adquiridos los datos. La mayoría de las veces, es necesario el uso de conversores analógico digital (ADC) externos que lo resuelvan, lo cual produce un incremento del costo y complejidad al dispositivo.

Existen distintos tipos de conversores AD y cada uno con sus correspondientes aplicaciones (Figura 1). Como se puede ver, el ADC $\Sigma\Delta$ que se pretende desarrollar comprende las aplicaciones de baja frecuencia como puede ser audio, señales biológicas y metrología. La fortaleza de este conversor va a estar dada por la cantidad de bits con los cuales representa una magnitud.

En este trabajo, se propondrá una primera implementación de un ADC $\Sigma\Delta$, sobre una FPGA prácticamente en su totalidad.

A. Discretización y Cuantización

Teniendo una señal continua $x(t)$, el proceso de discretización consiste en la obtención de muestras de la señal

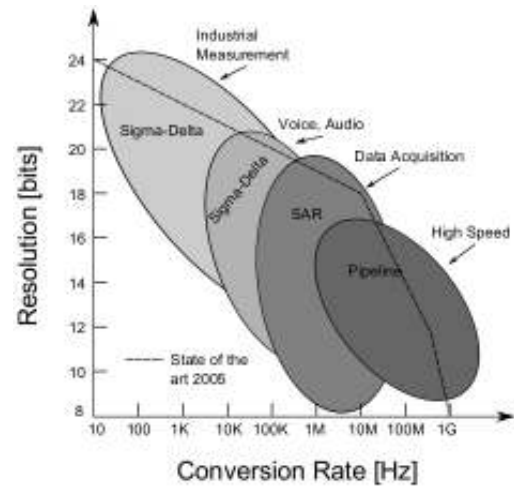


Fig. 1: Tipos de conversores[5]

cada un cierto tiempo de muestreo T . Se puede expresar matemáticamente como:

$$x_s(t) = \sum_{n=-\infty}^{+\infty} x(nT) \cdot \delta(t-nT) \quad (1)$$

Realizando la transformada de Fourier a la señal $x_s(t)$:

$$X_s(\omega) = \sum_{n=-\infty}^{+\infty} X(\omega - n\frac{2\pi}{T}) \quad (2)$$

Como se puede apreciar, el espectro de la señal estará repetido cada $\frac{2\pi}{T}$. De esto, se deduce que: 1) El espectro de la señal debe estar restringido a un cierto ancho de banda finito. 2) La frecuencia de muestreo debe ser del doble de la frecuencia máxima de la señal (teorema de Nyquist). De no cumplirse alguna de las dos condiciones, se produce el efecto *Alias* (solapamiento del espectro).

La señal discretizada en tiempo, también puede ser representada en el dominio discreto:

$$x[n] = x(nT), \forall n \quad (3)$$

Cuando es muestreada una señal, no solo va a estar discretizada en tiempo, también va a estar discretizada en amplitud. A esto

se lo conoce como cuantización[3]. Este fenómeno, produce un error cuya densidad de probabilidad se considerará constante entre $-q/2$ y $q/2$. Siendo q la resolución del cuantizador. Este error se lo modeliza como ruido equidistribuido N_q , incorrelado, con una potencia:

$$e_q^2 = q^2/12 \quad (4)$$

Por perturbaciones externas, la señal también va a estar contaminada con otros ruidos, como por ejemplo el ruido de 50Hz de línea. Luego la señal muestreada sera:

$$x_{q[n]} = x_{[n]} + N_{[n]} + N_{q[n]} \quad (5)$$

B. Oversampling y Decimación

El error de cuantificación nunca se podría eliminar en un sistema digital, pero puede reducirse la potencia del mismo en nuestro ancho de banda de interés. La técnica que se usa para esto se conoce como *oversampling*.

Si se observa la expresión de la potencia del ruido de cuantización, se puede observar que es constante y no depende del tiempo de *sampling* T. Aumentando la frecuencia de muestreo, el nivel ruido se verá disminuido a causa de mantener la misma potencia total y seguir distribuyendo de manera uniforme.

Este método posee varias ventajas: 1) Requiere un filtro Anti-Alias menos exigente, 2) Disminuye el ruido de cuantización en la banda de interés. 3) Permite que el ruido fuera de la banda pueda ser eliminado.

Una gran desventaja que posee este método es la gran capacidad de procesamiento que se necesita y la gran cantidad de memoria que consumiría en el sistema para realizar cualquier procesamiento. Es por ello que, luego de pasar la señal sobremuestreada por un filtro pasa bajo, se realiza un proceso de decimación. Básicamente, consiste en descartar muestras para llevar la señal a una condición mas próxima a la impuesta por el teorema de Nyquist. A la hora de descartar las muestras, hay que tener en cuenta la expansión del espectro (se esta variando implícitamente la frecuencia de muestreo) y que no se produzca el efecto *Alias*, siendo ese el motivo de filtrar previamente la señal.

C. Noise Shaping

Aunque con el método explicado en la sección B se ha eliminado mucho del ruido de cuantización, existe otro método que ayudará a eliminar gran parte del ruido restante. Este método es conocido con el nombre de *noise shaping*. Analizando el diagrama en bloques de la figura 2 con el teorema de superposición, se obtiene la siguiente salida:

$$Y_{(s)}|_{N_{(s)}=0} = X_{(s)} \frac{H_{(s)}}{1 + H_{(s)}} \quad (6)$$

$$Y_{(s)}|_{X_{(s)}=0} = N_{(s)} \frac{1}{1 + H_{(s)}} \quad (7)$$

$$Y_{(s)} = X_{(s)} \frac{H_{(s)}}{1 + H_{(s)}} + N_{(s)} \frac{1}{1 + H_{(s)}} \quad (8)$$

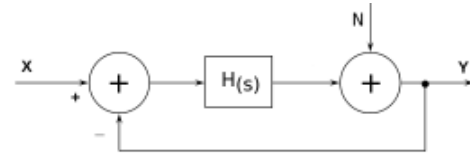


Fig. 2: Diagrama en bloques - Noise Shaping

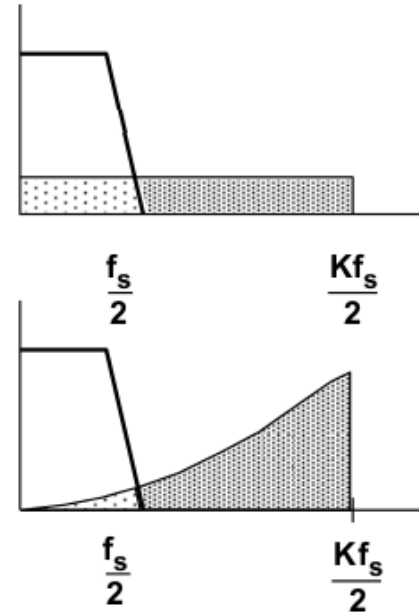


Fig. 3: Noise Shaping [4]

Por ejemplo, considerando $H_{(s)} = \frac{1}{s}$:

$$Y_{(s)} = X_{(s)} \frac{1}{s+1} + N_{(s)} \frac{s}{s+1} \quad (9)$$

Es posible percibir un comportamiento dual del sistema: Se comporta como un pasa bajos para la señal y como pasa altos para el ruido de cuantización. Este efecto se puede distinguir en la Figura 3. Si se desea obtener mejor moldeado del ruido, se pueden usar transferencias de orden superior u otro tipo de realimentaciones.

II. MATERIALES Y MÉTODOS

En la Figura 4 se observa los distintos bloques que componen el ADC en su totalidad. A continuación, se evaluarán las soluciones propuestas para cada uno de estos bloques.

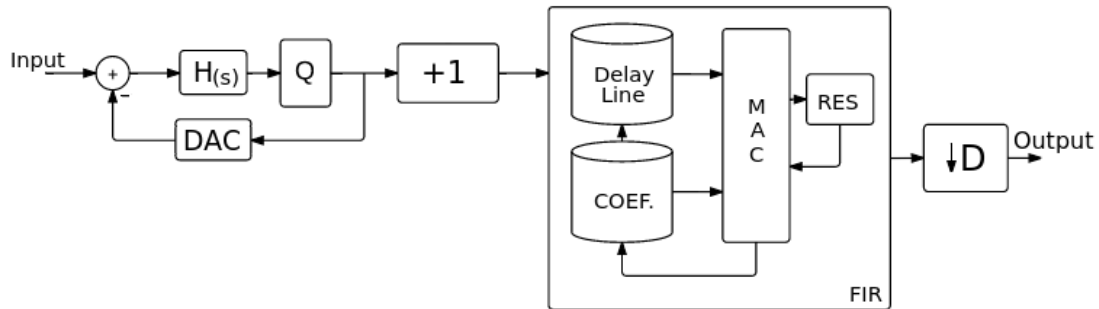


Fig. 4: Diagrama en Bloques - Implementación

A. Etapa de entrada

La primer etapa consta del cuantizador y *noiseshaping* [4, 5, 6, 7], comúnmente conocida como "Modulador $\Sigma\Delta$ ". Para realizarlo, se propone el circuito de la Figura 5. A su salida, vamos a obtener un *bit streaming* el cual, la cantidad de 1's en una ventana de $2^N - 1$ muestras, es proporcional a la tensión aplicada a su entrada, siendo N la cantidad de bits que se desean obtener de esta primer etapa. Es por ello, que se dispone de un sumador serial (Figura 6) el cual suma el valor en su entrada durante la ventana. Actualmente, estas ventanas no están superpuestas (no comparten muestras) realizando implícitamente una decimación con un factor de $2^N - 1$. Siendo la ecuación matemática de la salida la siguiente:

$$y[n] = \sum_{i=0}^{2^N-1} w_{[n \cdot 2^N - i]} \quad (10)$$

Esta solución es fácil de implementar dentro de una FPGA, pero nos impone un limite en cuanto a la relación entre bits y la frecuencia de muestreo. Si se aumenta 1 bit la cuantización, la frecuencia de sampling debe ser el doble de la anterior. Como la frecuencia de muestreo máxima esta dada por la tecnología de la FPGA y el diseño sobre la misma, no se podrá incrementar indiscriminadamente la cantidad de bits. El circuito modulador consta de dos resistores y un capacitor, formando un nodo que realiza la suma de las corrientes de entrada y de realimentación, produciendo entonces la carga y descarga del capacitor ya mencionado. Para poder generar una realimentación negativa sin el uso de un circuito activo, se tomo como tensión de realimentación la salida negada del cuantizador. Esto produce que, cuando la tensión en el capacitor supera el valor de *threshold* de la entrada del *flip-flop*, la salida negada del *flip-flop* se pondrá en 0, produciendo una corriente que descarga el capacitor. Análogamente, si la tensión en el capacitor es menor que el valor de *threshold*, la salida negada del *flip-flop* tomaría el valor lógico 1, produciendo la carga del capacitor. Por simple inspección, se puede ver que esta etapa genera una oscilación dependiente de la tensión de entrada, la frecuencia de muestreo y las constantes de tiempo

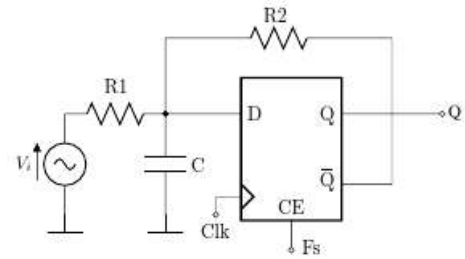


Fig. 5: Implementación - Modulador Sigma Delta

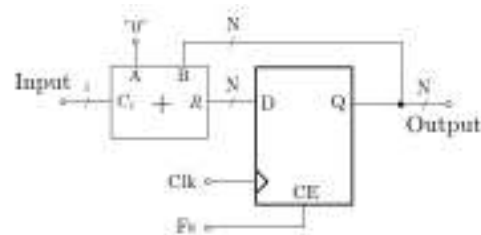


Fig. 6: Sumador Serial

de carga del capacitor. Si la constante de tiempo es mucho menor al tiempo de muestreo, se producirá una secuencia a la salida del *flip-flop* que varía constantemente por cada ciclo de *clock* concluyendo en una cuantización errónea de la señal. De lo contrario, si la constante de tiempo es mayor que la frecuencia de muestreo, a la salida del *flip-flop* tendremos una secuencia en la cual la información de la tensión de entrada va a estar dada por la duración de los estados lógicos. Los valores de tensión que pueden ser medidos, son calculados en el dominio temporal (analizando el régimen permanente). Aplicando el método de superposición y teniendo en cuenta que el *flip-flop* tiene una tensión de *threshold* de $V_{dd}/2$, siendo V_{dd} la tensión de alimentación de la FPGA:

$$V_{dd} \left(\frac{R_1 + R_2}{2R_2} - \frac{R_1}{R_2} \right) \leq V_i \leq V_{dd} \left(\frac{R_1 + R_2}{2R_2} \right) \quad (11)$$

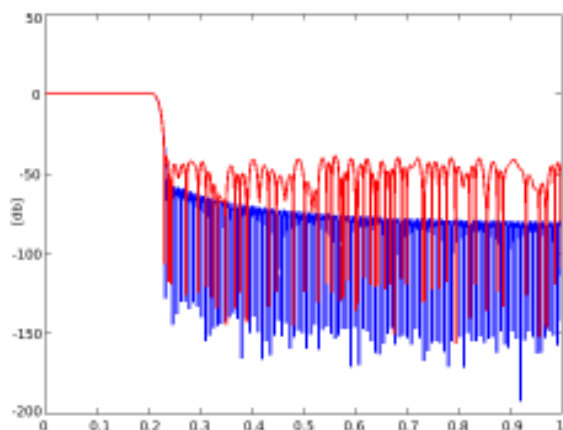


Fig. 7: Respuesta del Filtro. Azul: Double-precision floating-point - Rojo: Fixed Point 1:8

En el caso que R_1 y R_2 sean iguales, el rango del ADC sería $0 \leq V_i \leq V_{dd}$. De no cumplirse esta condición, además de modificarse el rango de conversión, las corrientes incidentes al nodo, provenientes de la entrada y de la realimentación, influenciarían de distinta forma la carga y descarga del capacitor, lo cual causaría un error de offset en la conversión. Actualmente, el conversor implementado es de 8 bits, por lo tanto, la ventana de análisis del sumador es de 255 muestras. Si la tensión de entrada la consideramos constante y de un valor lo suficientemente alto para mantener la salida del cuantizador constante en 1, el valor que se obtendrá a la salida del sumador es de 255 (0xFF), el valor mas grande que puede representarse en 8 bits. De lo contrario, si la tensión es lo suficientemente baja para que el cuantizador no cambie su valor de cero lógico, el sumador obtendrá a su salida el valor cero (0x00).

B. Filtrado y Decimación

Para realizar el filtro, se utilizaron 2 bloques de memoria: una para la *delay line* (muestras) y otra para los coeficientes. También, se desarrollo una MAC (*multiply-accumulate*) en punto fijo 1:8. (1 Bit de Signo/Magnitud y 8 decimales). Se tomó en cuenta los tamaños de los registros del multiplicador (el doble que el de los factores) y acumulador para que no produzca *overflow* ni efectos de saturación.

Para diseñar el filtro, se realizó un script en el software MATLAB. Utilizando la función `fir1` y pasándole como parámetros la frecuencia del corte 0.2π y orden 255, se obtuvieron los coeficientes que luego fueron codificados en punto fijo 1:8. Esto produjo un deterioro significativo del SNR (Signal Noise Ratio) como se puede ver en la Figura 7 (de 70dB a 48dB). Si se realiza el calculo de bits efectivos [3]:

$$ENOB = \frac{SNR - 1.76}{6.02} \quad (12)$$

Se obtiene como resultado que el ADC propuesto tendrá como máximo 7.7 bits efectivos.

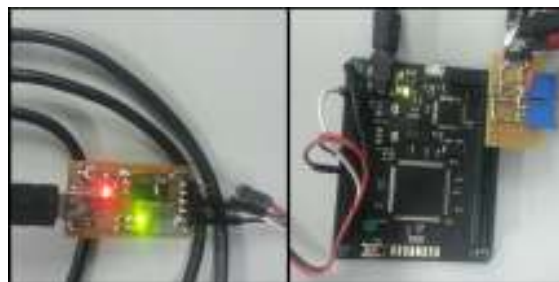


Fig. 8: MOJOv3 y FT232

Luego del proceso de filtrado, se procedió a descartar muestras para realizar la decimación con un factor de 4. Dando una decimación total del sistema de 1024. La frecuencia final de muestreo del ADC es de 2Khz.

C. Hardware y Síntesis

El sistema ha sido implementado bajo la plataforma de desarrollo MOJOv3 de embeddedmicro. Esta, posee una FPGA Spartan 6 XC6SLX9 y un entorno de programación de la misma basada en un microcontrolador Atmel.

Utilizando la herramienta de desarrollo ISE de Xilinx, se describió en VHDL todo el ADC analizado anteriormente y un periférico UART de baudrate 115200, no-parity y un bit de stop, para poder extraer los datos.

La síntesis dio como resultado una frecuencia máxima de 100MHz, siendo el limitante la unidad MAC. Los recursos utilizados de la FPGA son 2262 Slices (18%). Este numero fue muy elevado porque se describió la *delayline* del filtro como una FIFO en vez de como una RAM para facilitar la lógica. Si se realiza con una BlockRam, comparando con la implementación de la memoria con los coeficientes, se estima que ese numero disminuiría a 700 Slices (menos del 8% de la FPGA).

En la Figura 8 se puede ver la MOJOv3 con una pequeña plaqueta conectada, la etapa de entrada. Tiene la posibilidad de calibración con 2 potenciómetros multi-vuelta (R_1 y R_2) y de cambiar el capacitor. Además, en la imagen de la izquierda, se puede ver la placa conversora de USB-UART que se utilizará para la obtención de datos.

D. Mediciones

Para realizar las mediciones, se utilizó de generador de señales de referencia el generador arbitrario Agilent 33600A, para luego adquirir las muestras con un script en Matlab y procesarlas.

Para comprobar la linealidad y monotonía, se tomaron 1024 muestras por cada nivel de tensión dentro del rango del ADC con un paso de 10mV. A estas muestras, se le calcularon la mediana, obteniendo un gráfico Tensión-Muestra (Figura 9). Para analizar su comportamiento, se realizó una regresión lineal a lo obtenido anteriormente, para luego calcular el

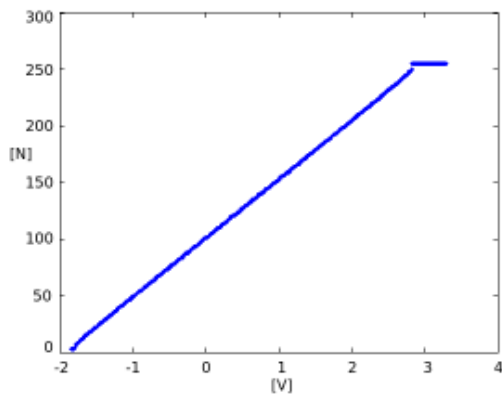


Fig. 9: Linealidad

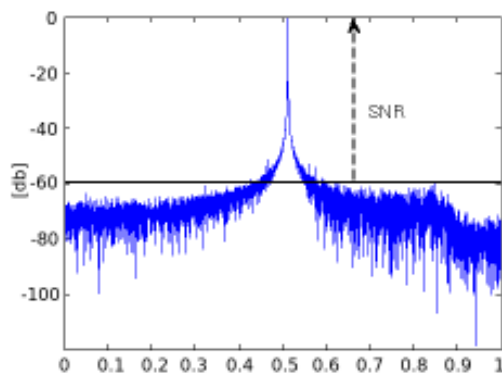


Fig. 10: Relación señal a Ruido

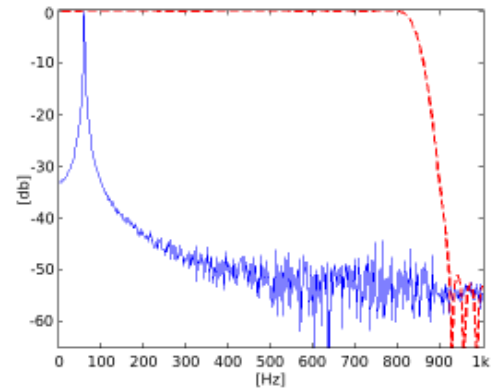


Fig. 11: Efectos del Filtro en el espectro de salida

funciona acorde a lo previsto. Los bits efectivos obtenidos en la práctica dieron un resultado esperado. Habiendo calculado 7.7 bits como máximo, se obtuvieron 6.5 los cuales, para la simpleza de esta implementación, son considerados aceptables.

En próximos trabajos, se propondrá eliminar la relación que vincula la cantidad de bits y la frecuencia de muestreo. Para ello, se están estudiando distintas arquitecturas para la implementación de filtros de decimación como pueden ser los filtros polifásicos y los filtros CIC[8] (*Cascaded integrator-comb filter*). También, se están estudiando otras transferencias para la etapa de entrada, para así, disminuir lo mas posible la potencia de ruido de cuantización [1].

coeficiente de linealidad dando como resultado 99.98%. En la figura 9 se puede ver su respuesta lineal y los niveles de saturación que se obtuvieron (-1,84v y 3.1v) dando un rango de casi 5v.

Posteriormente, se procedió a realizar 160 pruebas ingresando señales senoidales con distintas frecuencia (30Hz, 60 Hz, 100Hz, 250Hz y 500 Hz) y amplitud (4.6v y 2.3v pico a pico), de las cuales se adquirieron 500 muestras en cada caso. A esas muestras, se les realizó la transformada discreta de Fourier y, de manera gráfica, se determinó el SNR como se puede ver en la figura 10. Una vez medido el SNR de los distintos casos, se procedió a calcular la media, dando como resultado SNR=41dB. Con la ecuación 12, se determinaron 6.56 bits efectivos.

III. CONCLUSIÓN

Los resultados obtenidos de este trabajo son satisfactorios con respecto a lo esperado. Se desarrolló una forma sencilla de integrar un conversor analógico digital a cualquier modulo con FPGA que no lo disponga.

Analizando la transformada de Fourier de la salida del ADC y contrastándola con la curva de transferencia del filtro (Figura 11), podemos notar que el filtro anti-alias de decimación

REFERENCES

- [1] J. Hellman. *Implementation of a Low-Cost Analog-to-Digital Converter for Audio Applications Using an FPGA*, Linköpings universitet, 2013.
- [2] H.V. Palagiri, M.L. Makkena, K.R. Chantigari. *Performance Analysis of First Order Digital Sigma Delta ADC*, Computational Intelligence, Fourth International Conference on Communication Systems and Networks, 2012.
- [3] R.G. Lyons. *Understanding Digital Signal Processing*, Third Edition, Prentice Hall, 2004.
- [4] W. Kester. *Analog-Digital Conversion*, Analog Devices, 2004.
- [5] W. Kester. *Which ADC Architecture Is Right for Your Application?*, Analog Dialogue, (39),2005.
- [6] R.W. Stewar. *An Overview of Sigma Delta ADCs and DAC Devices*, IEE Colloquium on Oversampling and Sigma-Delta Strategies for DSP, 1995.
- [7] S. Park. *Principles of Sigma-Delta Modulation for Analog-to-Digital Converters*, Motorola, 1993.
- [8] U. Meyer-Baese *Digital Signal Processing with Field Programmable Gate Arrays*, Tercera Edición, Springer, 2007.

Comparison of Isolated Speech Recognition Algorithms with a Cortex-M4 Processor

Alejandro G. Alvarez and Mariano Marufo da Silva
Electronics department
UTN-FRBA
Buenos Aires, Argentina

Diego A. Evin
Laboratorio de Investigaciones Sensoriales
CONICET-UBA
Buenos Aires, Argentina

Abstract - This paper summarizes the proposals and developments that are part of two theses in electronic engineering. Both theses propose the design and development of real-time isolated speech recognition systems, embedded on microcontroller devices. The first one applies dynamic time warping (DTW) and is implemented on a STM32F4-Discovery board, while the second makes use of support vector machines (SVMs) and is implemented under the EDU-CIAA board (<http://www.proyecto-ciaa.com.ar/>). Despite the differences between boards and acquisition methods, both are based on Cortex-M4 processors and will have the same processing stage, which allows side by side comparison.

I. Introduction

Even though automatic speech recognition (ASR) on embedded platforms has reached performance levels consistent with the requirements of commercial applications, in most cases this performance is achieved at the expense of transferring the complexity of the problem to remote servers. Typical applications run the speech signal processing on the device, send the compressed data to servers thousands of kilometers away, where the recognition process takes place, and finally get the result back to the device [1]. This represents a problem since the system requires an internet connection and its speed is strongly limited by the available bandwidth. The proposed solution is to implement the recognition algorithm directly on the device.

II. Acoustic parameters extraction

Due to the speech signal redundancies, and inspired by the human auditory processing, most of the ASR systems rely on some type of short time spectral information; extracted in a step called the 'front end'.

In this work we use Mel-frequency cepstral coefficients (MFCCs), one of the most popular spectral features in the literature [2]. We implement the front end processing using the DSP libraries available for the Cortex-M4 processors.

III. Word models with HMMs

A Markov model is a finite state machine in which the transition from one state to another is performed according to a probability. Depending on the state in each time unit, it generates a new observation vector from a probability density output function. A hidden Markov model (HMM) is a double stochastic process in which the process that generates the state sequence is not observable (it is hidden) and the only

observable process is the one that generates the observation vectors sequence. Speech production can be represented by these models and in particular, in the isolated speech recognition problem each word from the system vocabulary is represented by its own HMM [3], [4], [5].

IV. Recognition algorithms

Techniques of template matching such as dynamic time warping (DTW) use dynamic programming for time alignment between two utterances in order to derive a meaningful assessment of their similarity [6], [7], [8].

Support vector machines (SVMs) are one of the state of the art supervised learning algorithms used in pattern recognition problems. However, they cannot effectively model the temporal speech dynamic. For that reason, currently SVM/HMM hybrid systems are used, which combine the discriminative power of that method, with the temporal modeling properties of HMMs [9].

The first thesis will apply the DTW technique using a STM32F4-Discovery board [10] and the second one will implement an SVM/HMM system on an EDU-CIAA board.

V. Comparison

Upon the completion of the implementations, the differences in performance for both systems will be evaluated and compared. Even though there are differences between boards and acquisition methods (different models of digital MEMS microphones), both are based on Cortex-M4 processors and will have the same processing stage, which allows side by side comparison.

In that comparison the performance will be assessed in terms of accuracy and speed. Accuracy will be rated with the word error rate (WER), whereas speed will be measured using the real time factor. Furthermore, resources requirements and consumption, such as processor usage, or required memory, will be considered in the evaluation. We expect to have available the results by the end of the current semester.

REFERENCES

- [1] M. Schuster, "Speech Recognition for Mobile Devices at Google," in PRICAI 2010: Trends in Artificial Intelligence. Springer Berlin Heidelberg, 2010, pp. 8-1

- [2] M. Sahidullah and G. Saha, "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition," *Speech Communication* 54, no. 4, 2012, 543-565.
- [3] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE* 77.2, 1989, 257-286.
- [4] L. R. Rabiner and B. Juang, *Fundamentals of speech recognition*, First. Prentice-Hall International, Inc., 1993.
- [5] D. Jurafsky and J. Martin, *Speech & language processing*, Second. 2009.
- [6] B. H. Juang and L. R. Rabiner, "Automatic speech recognition—a brief history of the technology development," Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara, 1, 2005.
- [7] M. Müller, "Dynamic time warping," in *Information retrieval for music and motion*, 2007, p. 318.
- [8] S. Salvador and P. Chan, "FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space," *Intell. Data Anal.*, vol. 11, no. 5, 2007.
- [9] A. Ganapathiraju, J. E. Hamaker and J. Picone, "Applications of support vector machines to speech recognition," *Signal Processing, IEEE Transactions on*, 52(8), 2004, 2348-2355.
- [10] Q. Qu, "Recognition Module Based on STM32," *Int. Symp. Commun. Inf. Technol.*, no. Iscit, pp. 73–77, 2011.

Inversor Sinusoidal con Control Embebido

José María Barrio Chaud

Laboratorio de Electrónica

FMA. Universidad Católica de Santiago del Estero

Santiago del Estero, República Argentina

electrolab@ucse.edu.ar

Daniel Ernesto Saad

Laboratorio de Electrónica

FMA. Universidad Católica de Santiago del Estero

Santiago del Estero, República Argentina

electrolab@ucse.edu.ar

Resumen- En este trabajo se presenta el diseño, desarrollo e implementación de un sistema de conversión de energía eléctrica de corriente continua a corriente alterna (CC-CA) (1) para instalaciones fotovoltaicas, con salida de tensión sinusoidal regulada. El proyecto se basa en un algoritmo embebido en un microcontrolador Cortex-M4 LM4F232 (2), y un puente MOSFET (3) para obtener una modulación de onda SPWM (Modulación Sinusoidal por Ancho de Pulso) (4) y un regulador digital PID (Proporcional Integral Derivativo) (5) para el control de la amplitud de la onda de salida. Los resultados obtenidos permiten disponer de una tensión alterna sinusoidal con especificaciones de 50 Hz, 220 Vrms, y regulación automática de amplitud.

Índice de Términos— Convertidor. SPWM. Fotovoltaico.

I. INTRODUCCIÓN

La generación de energía alterna a partir de un sistema fotovoltaico, requiere de la conversión de la tensión continua a tensión alterna, con especificaciones de 50 Hz, 220 volt RMS, y potencia de salida adecuada a la carga consumidora. A estos fines se presenta el diseño y desarrollo de un ondulador del tipo sinusoidal, con las especificaciones señaladas.

II. DESARROLLO

El sistema se compone de dos bloques principales, un puente de transistores MOSFET, conmutado por un algoritmo SPWM en modo de salida unipolar, y un convertidor de CC-CC elevador (Boost) (6), de alimentación del puente MOSFET, con control de amplitud de la onda sinusoidal de salida por algoritmo PID, la figura 1 muestra el diagrama del sistema desarrollado. El diseño utiliza a la salida transistores MOSFET para una carga máxima de 400 W.

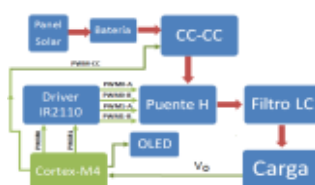


Fig. 1. Diagrama esquemático del sistema

III. RESULTADOS

El desarrollo logrado cumple con las especificaciones de un convertidor sinusoidal a 50 Hz, 220 Vrms, distorsión armónica menor al 5%, rendimiento mínimo del 90 % y potencia de salida máxima de 400 W, en una función de herramienta didáctica para varias disciplinas de la carrera de ingeniería electrónica. Principalmente se utiliza con dos paneles fotovoltaicos de 90 W cada uno y baterías de acumuladores de ácido plomo como fuente primaria de alimentación del convertidor CC-CC.

IV. CONCLUSIONES

Se desarrolló un convertidor CC-CA sinusoidal puro, a partir del uso de software embebido y algoritmos de procesamiento y control digital utilizando tecnología de microcontroladores de 32 bits, con visualización en tiempo real de la onda de salida utilizando un lcd Grafico (OLED) (7). Entre las prestaciones futuras a implementar se están desarrollando el modulo algorítmico de la transformada rápida de Fourier (FFFT) (8), el algoritmo de control de carga de baterías y seguimiento del punto máxima potencia para los paneles fotovoltaicos.

REFERENCIAS

- [1] Daniel W. Hart, *Electronica de Potencia*, Prentice Hall, 2001, pp. 315-358.
- [2] TexasInstruments. http://d1.ourdev.cn/bbs_upload782111/files_49/ourdev_709459MD2LEP.pdf. PB-LM4F-Series-00. 2011.
- [3] Muhammad H. Rashid, *Electronica de Potencia : Circuitos, Dispositivos*, Prentice Hall, 1995, pp. 280-285.
- [4] Ned Mohan, Tore M. Undeland y William P. Robbins I.S. *Electrónica de potencia: Convertidores, aplicaciones y diseño*, 3ra Edición, 2009, pp. 502-521.
- [5] Katsuhiko Ogata. *Ingenieria de Control Moderna*. Prentice Hall, 2010, pp. 567-592.
- [6] Ned Mohan, Tore M. Undeland y William P. Robbins I.S., *Electrónica de potencia: Convertidores, aplicaciones y diseño*, MacGraw-Hill, 3ra Edición, 2009, pp. 176-214.
- [7] Takatoshi Tsujimura, *OLED Display Fundamentals and Applications*, Series Editor's Foreword, 2012, pp. 5-30.
- [8] Roman Kuc. *Introduction to Digital Signal Processing*, SSP BS Publications, 2008, pp. 118-142.

Foro Tecnológico

Pósters

FPGAs, HDLs y ASICs

Savitzky-Golay Filter design in FPGA described in VHDL

L. F. Rocco, L. De Micco and C. A. Gayoso

National University of Mar del Plata

{cgayoso,ldemicco}@fi.mdp.edu.ar, leandro_f_rocco@hotmail.com

Abstract—The present work consists in the implementation of Savitzky-Golay filters described in VHDL. The work includes a complete system able to design and test the filters. In the case of the design, an application that generates parametric IP Cores was developed. The application allows the user to select the desired parameters of the filter and automatically generates the VHDL code. This file is ready to be compiled and programmed in any FPGA with minimum hardware requirements. The testing issue was resolved by a circuit board that was designed and implemented, customized for this case. The board is in charge of solving the inputs and outputs of the FPGA, this is, the analog to digital and digital to analog conversions. The board is also able to add noise to the incoming signal before being digitalized, it also has a clock generator circuit and digital isolators to protect the FPGA against harmful voltages. This platform allows designing and testing the filter in order to implement it in an ASIC.

I. INTRODUCTION

The Savitzky-Golay (SG) filters [1], are digital devices whose topology is similar to that of the FIR filters, with the advantage that all their coefficients are integers. This fact facilitates its implementation in digital electronic devices due to the data format is in two's complement. So, for example, there is no need to implement complex circuits to perform floating point calculations.

The main advantage of these filters is that they preserve the most important characteristics of the incoming signal, such as the maximum and minimum values of the curves and the width of their peaks. These characteristics are generally lost or distorted when using other filtering techniques. What is more, the SG filters not only allow to obtain the filtered input signal, but also their filtered derivatives. This fact makes them extremely useful in most applications related to digital signal processing. For example it is employed for removing high frequency noise components from electrocardiograms (ECGs) [2][3], in algorithms of detection of the QRS complexes of ECG signals. Sherya in [4] proposed a QRS detection algorithm using Savitzky-Golay filter, however, as the implementation of the filter was sequential the real time execution of the algorithm was lost. To our knowledge this is the first time this filter is implemented in a FPGA.

In their original work, Savitzky and Golay have calculated the coefficients, also known as convolution integers, for polynomials of different degrees, number of points and orders of their derivatives. Some corrections can be found in [5].

Here, the calculation of convolution integers was performed using matrix algebra, which facilitated the design of the algorithms used in the application. This application was programmed in *Visual Studio* in C# language. After the user has entered the filter parameters, the application generates the *vhd* file that contains the VHDL code of the SG filter. Once the *vhd* file is created, the user can compile and program it in the FPGA.

The rest of the paper is organized as follows: The calculation of convolution integers is developed in section II. There, the topologies used for the implementation of SG filters in the FPGA are also shown. Section III describes the operation of the testing board and it lists the devices used for its construction. The section IV shows graphs that were obtained during the testing stage of the system. These graphs were obtained using the testing board and the available laboratory instruments. Finally, section V deals with the conclusions.

II. SAVITZKY-GOLAY FILTERS

The SG filters perform a convolution process between the incoming signal, $x[n]$, and the impulse response $h[n]$ of an equivalent LTI (Linear Time-Invariant) system, corresponding to the SG filter. This impulse response is obtained by the definition of a approximation polynomial $p[n]$ that minimizes the average squared error between this polynomial and the $2M + 1$ samples of the input signal. The polynomial $p[n]$ is as follows:

$$p[n] = \sum_{k=0}^N a_k n^k \quad (1)$$

where N is the degree of the approximation polynomial and the a_k are the coefficients to be determined.

The average squared error is obtained as follow:

$$\epsilon_N = \sum_{n=-M}^M (p[n] - x[n])^2 = \sum_{n=-M}^M \left(\sum_{k=0}^N a_k n^k - x[n] \right)^2 \quad (2)$$

which is valid for a set of $2M + 1$ samples of the input signal $x[n]$.

In the bibliography, particularly in [6], matricial algebra is used to perform the mathematical demonstration to obtain the coefficients a_k , this procedure results in the following matrices:

$$\mathbf{a} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{x} = \mathbf{H} \mathbf{x} \quad (3)$$

where the matrix \mathbf{A} is:

$$\mathbf{A} = \begin{pmatrix} (-M)^0 & (-M)^1 & (-M)^2 & \dots & (-M)^N \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (-1)^0 & (-1)^1 & (-1)^2 & \dots & (-1)^N \\ 1^0 & 1^1 & 1^2 & \dots & 1^N \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ M^0 & M^1 & M^2 & \dots & M^N \end{pmatrix}$$

$\mathbf{a} = (a_0, a_1, a_2, \dots, a_N)^T$ is the coefficients matrix of Savitzky-Golay in which each coefficient a_k is a function of the input samples $x[n]$ in the interval of $2M + 1$ samples. The vector \mathbf{x} is:

$$\mathbf{x} = (x[-M], \dots, x[-1], x[0], x[1], \dots, x[M])^T \quad (4)$$

Therefore \mathbf{H} is a $(N + 1) \times (2M + 1)$ matrix that contains the convolution integers:

$$\mathbf{H} = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \dots & h_{1,2M+1} \\ h_{2,1} & h_{2,2} & h_{2,3} & \dots & h_{2,2M+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{N+1,1} & h_{N+1,2} & h_{N+1,3} & \dots & h_{N+1,2M+1} \end{pmatrix}$$

The first row of \mathbf{H} matrix contains the convolution integers of Savitzky-Golay which are necessary to calculate the output $y[n]$. The remaining rows of \mathbf{H} correspond to the convolution integers of the derivatives of the signal $x[n]$.

The procedure for obtaining the output samples $y[n]$ is as follows:

- 1) $2M + 1$ samples of the input signal should be taken and multiplied by their corresponding convolution integers.
- 2) The obtained products must be added and divided by the "Norm". The Norm is an integer that results from calculating the greatest common divisor of the corresponding row of \mathbf{H} matrix. This leads to obtaining the polynomial coefficients a_k .
- 3) Once the a_k coefficients of the polynomial $p[n]$ are determined, it should be evaluated at $n = 0$, obtaining the output $y[0] = a_0$ corresponding to the midpoint of the range $2M + 1$.
- 4) Finally, right shift the interval one sample, and the whole procedure must be repeated until all samples

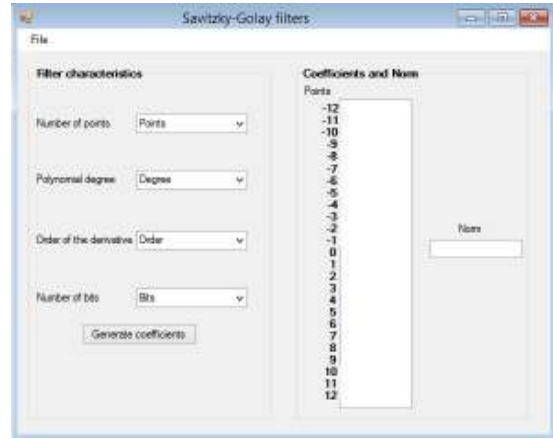


Fig. 1: Application (first window).

are processed.

If it is desired to obtain the i^{th} derivative of the input signal, $p[n]$ polynomial should be derived with respect to n and then evaluate in $n = 0$, and the a_i coefficient is obtained. Then, the output sample is $y[0] = a_i$.

The developed application executes an algorithm which calculates the matrix \mathbf{H} . So, once obtained the convolution integers, the application inserts them in the *vhd* file. Once this is done, the VHDL code can be compiled and programed in the FPGA. This application is shown in Fig. 1, on the left it can be seen the area where the user will introduce the parameters of the filter, such as the number of points, the degree of the approximation polynomial, the order of the derivatives and the number of bits with which the filter will be designed. Once selected these parameters the coefficients appear on the right text box of the application by clicking the "Generate coefficients" button.

The application not only generates the VHDL code corresponding to the SG filter, but also the IP cores that are necessary for the control of the testing board. Furthermore, it also generates the *qsf* file that contains the pin assignment for the DE2¹ board to carry out the tests. For this reason, a second window was designed to explain the functioning of the switches of the DE2 board and also, to allow defining the sample frequency that will be used for the ADC and DAC converters. This second window is shown in Fig. 2.

This second window also allows to access a PDF file that contains all the information about the testing board. It includes a detailed description of the input and output pins. The reason for this is that the pin assignment performed here is for the particular case of using the DE2 board. In the case the user wanted to employ another board, a new pin assignment, adequate for that board, should be performed

¹The board used in this work is the Altera DE2 Development and Education board.

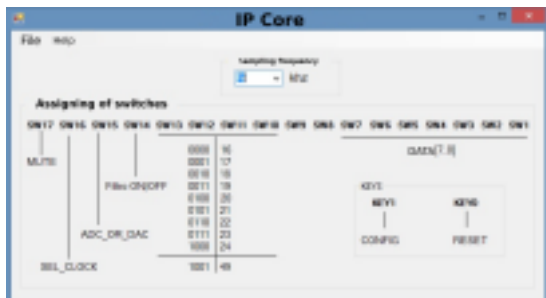


Fig. 2: Application (second window).

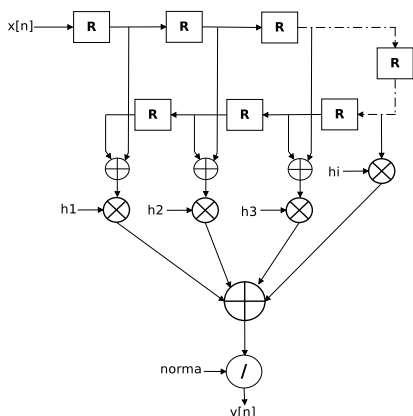


Fig. 3: Symmetrical topology.

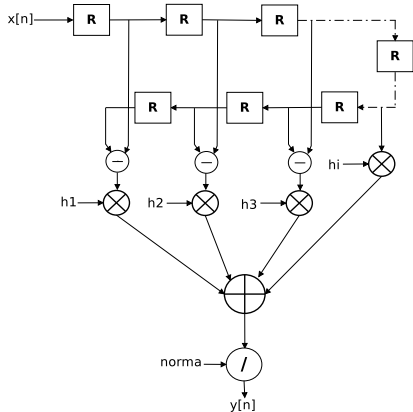


Fig. 4: Asymmetric topology.

with the help of this PDF file.

The implemented topologies are shown in Fig. 3 and 4. Fig. 3 corresponds to the symmetrical topology, in which the convolution integers to the sides of the central point ($M = 0$) have the same sign (this correspond to the filters and the derivatives of second order, fourth, sixth, etc). On the other hand, Fig. 4 corresponds to the asymmetrical topology, in which the convolution integers to the sides of the central point have different sign (derivatives of first order, third, fifth, etc).

The adders, multipliers and the divisor perform the operations in two's complement. The blocks designated with the letter "R" are shift registers that shift the samples that enter the filter. h_i are the convolution integers.

III. TESTING BOARD

The board developed mainly consists in two converters: the PCM1807 [7], an analog to digital converter, and the PCM1789 [8], a digital to analog converter. These devices are high performance converters designed to work in the audio band, with configurable sample frequencies. The data format is 24 bits in two's complement. These devices use the I^2S protocol [9] for transmitting and receiving data, and SPI protocol to configure their internal registers. The use of these converters proved to be advantageous for two reasons: the first is that the format of the samples is two's complement, therefore there is no need to perform any data conversion process. The second reason is that the samples are transmitted in series, which means a saving of data lines between the FPGA and the testing board, simplifying the design and reducing the amount of hardware.

The board has the following electronic circuits:

- *Adder*: performs the sum between the input signal and AWGN (Additive White Gaussian Noise). The operational amplifier used is the OPA2134 [10].
- *Low pass filter*: is a Butterworth second order filter, it is responsible for filtering the signal reconstructed by the DAC to achieve higher conversion efficiency.
- *Isolators*: provide electrical isolation between the FPGA and the testing board. These isolators are ISO7640 and ISO7641 [11].
- *Clock generator*: generates all the clock signals necessary for the operation of the converters. This is the PLL1705 [12]. The user can choose to generate the clock signal on the FPGA or from the testing board using this PLL.
- *Supply circuit*: consists of two voltage regulators that provide analog and digital voltages required for the converters and an inverter that provides negative voltage needed to power operational amplifiers.

A block diagram of the testing board is shown in Fig. 5. This board was designed in *Altium Designer*, taking into account the noise attenuation methods in the design of a PCB.

Fig. 6 shows the circuit of the noise generator. This circuit generates an approximation of AWGN. This is achieved by the reverse polarization of the base-emitter junction of transistor $Q1$.

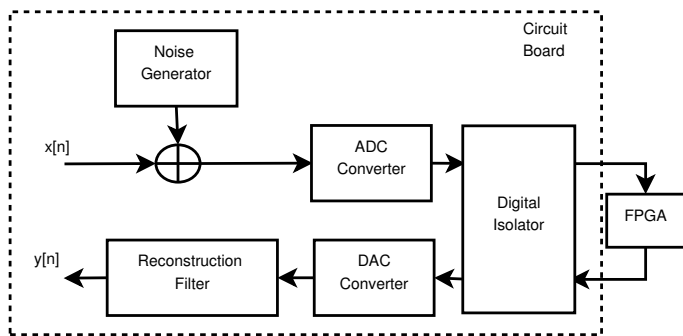


Fig. 5: Block diagram.

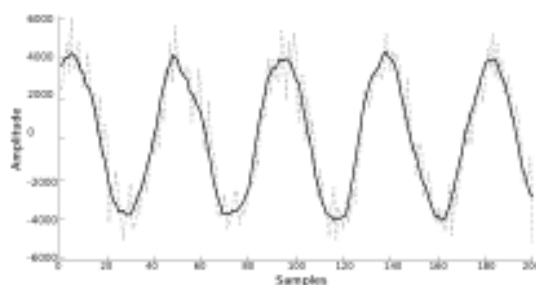


Fig. 8: Filtering 15 points.

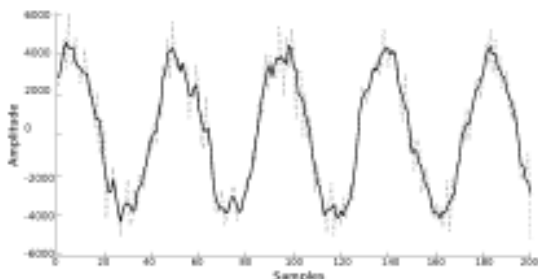


Fig. 7: Filtering 7 points.

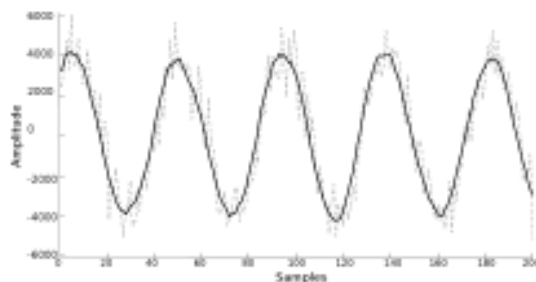


Fig. 9: Filtering 21 points.

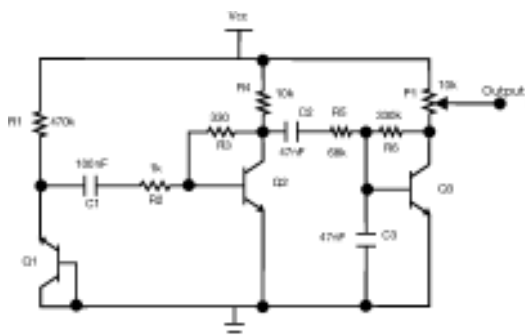


Fig. 6: Noise generator.

IV. RESULTS

SG filters with 7-25 points were implemented to verify proper operation. A 5Khz tone was used as test signal, which was immersed in AWGN. The output samples were acquired with the SignalTap software, included in the Quartus II package of the Altera company. Fig. 7, 8, 9 and 10 show some of the obtained results. These figures clearly show that the higher the number of points of the filter the better the filter behaves.

A derivative filter of 25 points was also performed. The acquired data are shown in the Fig. 11.

Table I shows a comparison of the amount of hardware used in the FPGA and the maximum operating frequency, in relation to the number of points of the SG filter.

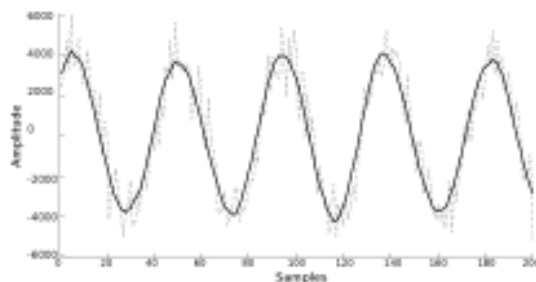


Fig. 10: Filtering 25 points.

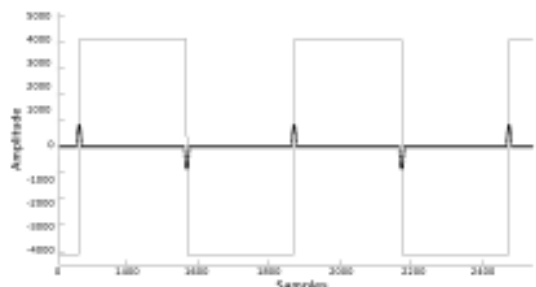


Fig. 11: Derivative filter of 25 points.

In order to illustrate how the SG filters preserve the main characteristics of the signal, Matlab simulations were performed using the algorithm mentioned earlier. For this, a signal similar to the one obtained with an electrocardiograph was created. In this kind of signals the important characteristic to recover are the width and the height of the peaks. This

Filter's points	Combinational functions	Maximum frequency without segmentation (Mhz)
5	984	11.83
7	1095	11.81
9	1413	10.52
11	1671	10.11
13	1517	10.69
15	2043	9.79
17	2055	9.78
19	2396	9.35
21	2553	9.02
23	2540	10.00
25	2906	8.66

Table I: Comparative table of SG filters.

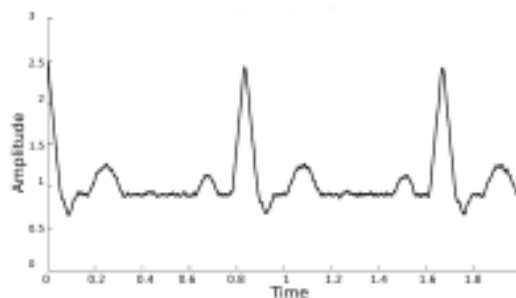


Fig. 14: ECG signal filtered. Cubic polynomial. $M = 75$.

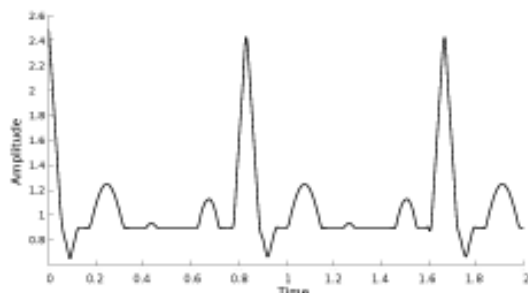


Fig. 12: ECG signal.

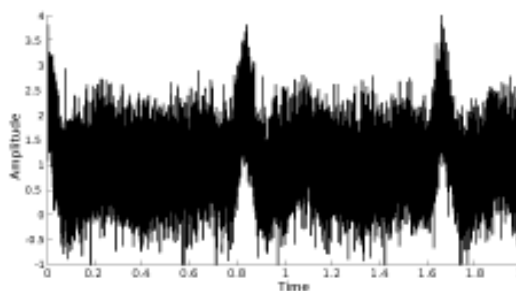


Fig. 15: ECG signal with noise.

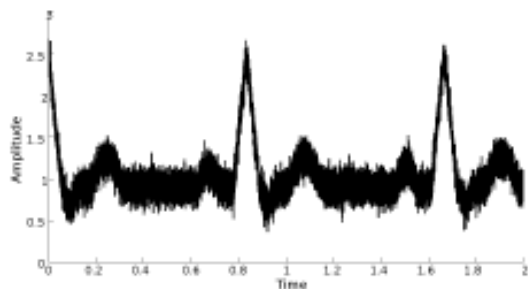


Fig. 13: ECG signal with noise.

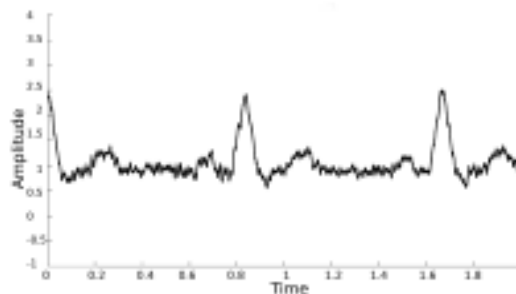


Fig. 16: ECG signal filtered. Cubic polynomial. $M = 75$.

signal is shown in Fig. 12.

Fig. 13 shows the same signal immersed in noise. This signal was filtered using $M = 75$, the obtained output can be seen in Fig. 14.

Fig. 15 and 16 evidence that even with a significantly high noise level the characteristics of the original signal can be partially recovered by using these filters.

A. Filter Effects on Noise

It follows from the calculation of the convolution integers that the same values are obtained for the degrees of polynomials 2 and 3, 4 and 5, 6 and 7, etc. This is also true for the derivatives of even order. However, for odd order derivatives the convolution integers are the same for the

degrees of polynomials 1 and 2, 3 and 4, 5 and 6, etc.

Fig. 17 shows the effect of attenuation of noise relative to the number of points of the filter and the degree of polynomial approximation. As the number of points increases, the *signal/noise* ratio improves. In contrast, increasing the degree of the polynomial approximation the *signal/noise* ratio worsens.

B. Data acquired with testing board

Data acquired with the board constructed in this work are shown in Fig. 18 and 19. At the top of Fig. 18 the input signal embedded in AWGN is shown, and in the bottom of the figure the filtered output signal can be seen. Fig. 19 shows the input signal (top) and the output signal (bottom) of a derivative filter of 25 points.

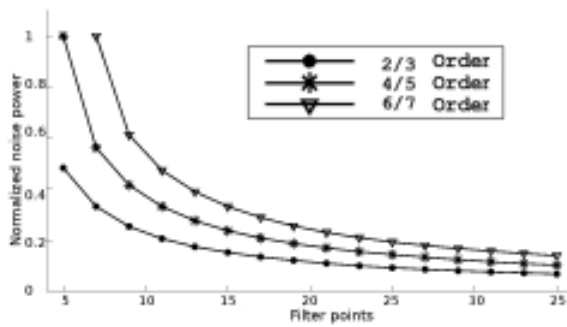


Fig. 17: Normalized noise power based on the number of points.

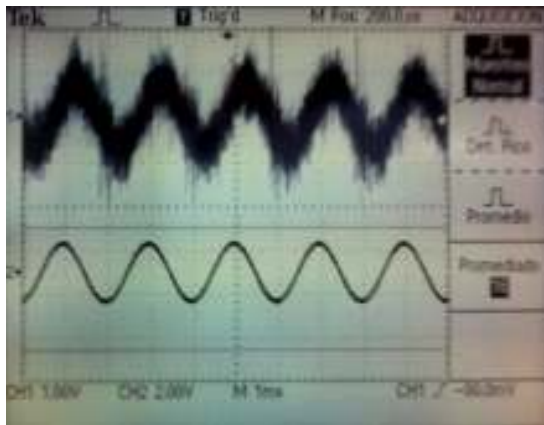


Fig. 18: 25 points filter.

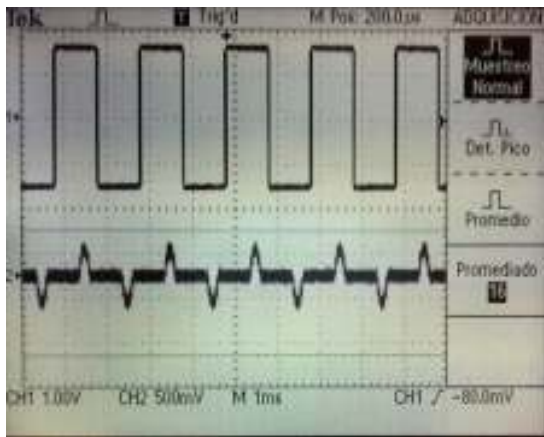


Fig. 19: 25 points derivative filter.

V. CONCLUSION

The most important equations of the Savitzky-Golay procedure have been listed. This procedure was implemented using matrix algebra, which is a different approach to that used classically, and offers several advantages.

Moreover, it has been developed an application which generates all the necessary files for programming a SG

filter in a FPGA. What is more, it has been designed and implemented a board that deals with the adaptation of the input and output signal and also with the testing of the implemented filter. It allows verifying the proper operation of the filters implemented in the FPGA.

The experiments and measurements performed have led to two main interesting conclusions: 1) Increasing the number of filter points produce an improvement in the *signal/noise* ratio with respect to the input signal. While, for the same amount of points, increasing the degree of the polynomial approximation, worsens this ratio. 2) It was also determined that it is not always desirable to increase the number of points of the filter. If the number of samples per cycle of the input signal is small compared with the number of points of the filter, increasing this last produces an attenuation of the input signal that is more significant than increasing the number of points of the filter SG.

The design of a SG filter turns out to be simple and inexpensive with the use of the application developed here. The testing board, along with the application developed in this work proved to be a good tool for academic use, allowing the user to have a complete system ready to run, truly easy to use and very fast.

REFERENCES

- [1] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," in *Analytical Chemistry*, vol. 36, no. 8, 1964, pp. 1627–1639.
- [2] S. Hargittai, "Savitzky-golay least-squares polynomial filters in ecg signal processing," in *IEEE, Computers in Cardiology*, 2005.
- [3] M. R. Bahmanyar and W. Balachandran, "An algorithm for filtering electrocardiograms to improve nonlinear feature extraction," in *Journal of Systemic, Cybernetics and Informatics*, vol. 5, no. 2, 2012.
- [4] S. Das and M. Chakraborty, "Qrs detection algorithm using savitzky-golay filter," in *Aceee International Journal on signal & Image processing*, vol. 3, no. 1, 2012.
- [5] J. Steinier, Y. Termonia, and J. Deltour, "Comments on smoothing and differentiation of data by simplified least square procedure," in *Analytical Chemistry*, vol. 44, no. 11, 1972, pp. 1906–1909.
- [6] R. W. Schafer, "What is a savitzky-golay filter?" in *IEEE SIGNAL PROCESSING MAGAZINE*, 2011, p. 111–117.
- [7] *PCM1807 datasheet*. Texas Instruments, 2005. [Online]. Available: <http://www.ti.com/lit/ds/symlink/pcm1807.pdf>
- [8] *PCM1789 datasheet*. Texas Instruments, 2009. [Online]. Available: <http://www.ti.com/cn/cn/lit/ds/symlink/pcm1789.pdf>
- [9] P. Semiconductors, "I2s bus specification," 1996.
- [10] *OPA2134 datasheet*. Texas Instruments, 1997. [Online]. Available: <http://www.ti.com/lit/ds/symlink/opa2134.pdf>
- [11] (2013) *Iso7640/41 datasheet*. [Online]. Available: <http://www.ti.com/cn/cn/lit/ds/symlink/pcm1789.pdf>
- [12] *PLL1705 datasheet*. Texas Instruments, 2002. [Online]. Available: <http://www.ti.com/lit/ds/symlink/pll1705.pdf>

FPGA-based embedded system for remote control of home devices

M. A. Gómez López, C. B. Goy.

Departamento de Electricidad, Electrónica y Computación
Facultad de Ciencias Exactas y Tecnología - UNT
Tucumán, Argentina
Mgomezlopez@herrera.unt.edu.ar

M. C. Herrera.

Departamento de Bioingeniería
Facultad de Ciencias Exactas y Tecnología - UNT
Tucumán, Argentina

Abstract— This paper proposes the design and implementation of an FPGA-based embedded system which allows the manipulation of different home devices through a selection menu displayed on a monitor that works with the VGA (Video Graphics Adapter) standard and a matrix keyboard. The communication between master and slave devices is performed by means of a wireless communication established with Zigbee protocol. The implementation of the video controller, the logic control and the data packages generation for the wireless communication are explained by means of detailing the performance of each digital block of the logic design.

Keywords—FPGA; wireless; zigbee; video; keyboard

I. INTRODUCTION

The domotic origin back to the 1970s. After much research, the first buildings automation devices appeared. During the following years, the international community showed a great interest in the search of the ideal home starting several tests with advanced appliances and automatic devices for the home. In Argentina, in the early of the 90s, and over the years, the installations of these devices become more frequent and important and began to expand on the market. Home automation is the science that provide the ability to convert a home into a smart home [1], this is accomplished using systems capable of providing energy management, safety, welfare and communication. Such systems are integrated through internal and external communication networks, which are wired or wireless, and whose control is performed from inside or outside the home. The communication networks are implemented using modules that allows the remote operations of appliances and house lights and thanks that all modules are compatible, the user can expand the capabilities of his home automation system whenever he decides.

In this work, a centralized domotic system that allows the manipulation of home devices such as lights, domestic appliances and electric doors is designed, and its control is digitally implemented using a FPGA (Field Programmable Gate Array) device. This technology offers the possibility of developing a comprehensive control system for the use of home devices, as well as its graphical user interface which will be displayed on a computer monitor and will consist in a selection menu to control the devices. In order to get the devices working together, it is necessary that they are connected through an internal network which can be wired or

wireless. In this work a Zigbee based wireless technology is chosen. This technology has advantages over others ones such as low power consumption, topology used, easy integration, ability to coexist with other networks, simple installation and easy operation. These characteristics fit all the current needs of the domotic control in homes. The management of the home devices is done through a wireless communication between a master device and slave actuators [3].

Figure 1 shows the general scheme of the system. It can be observed that each peripheral (or home device) to be controlled has a Zigbee based wireless receiver (Xbee module). The data entry is performed using a 4x3 matrix keyboard. The user interface consists in a selection menu displayed on a monitor. For this, a video controller generates timing signals and RGB signals compatible with the VGA standard. A Xbee module configured as coordinator communicates the system with the Xbee modules (configured as end devices) that are connected to each peripheral establishing a WPAN (Wireless Personal AreaNetwork).

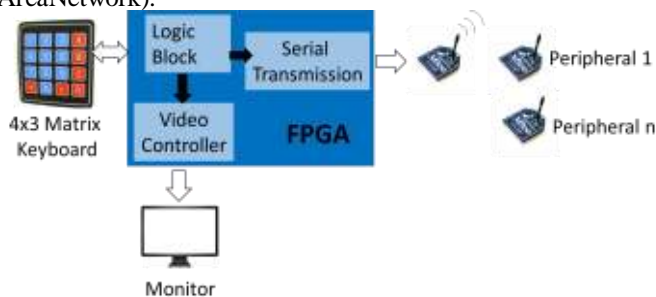


Fig. 1. General scheme of the system

II. MATERIALS AND METHODS

A. Video controller

A video controller is a computer component responsible for sending electrical signals to the monitor so it shows images to the user. It has its own memory and their abilities are measured in bytes. In this paper, a VGA video controller of 720x400 and 640x480 pixels is implemented in text and graphical mode respectively. It has 256 colors and 256 KB memory and sends electrical signals to monitor through a DB15 connector. The VGA standard contains five active signals of which, two are synchronism signals: HS (horizontal synchronism) and VS (vertical synchronism) compatible with TTL logic levels. They

indicate the change of horizontal line and frame (refresh rate) respectively. The other three signals give information about each color: R (red), G (green) and B (blue). They are analog signals whose values range from 0V to 0.7V indicating “no signal” and “peak intensity” respectively. The FPGA device can be programmed to deliver signals of three bits for each color which results in 512 different colors. In this work, the video controller is designed to use only one bit for each RGB channel. So they are only eight possible color combinations [4].

B. ZigBee protocol

An important part of a wireless network design is the need to create an efficient, fast and reliable way to transmit information within an area that covers a person and a radius of approximately 10m around him, whether the person is in motion or not. Thereby, the concept of WPAN networks emerges. In this work, the communication range is not a limiting factor since the data will be transmitted within the home. So, in this work the communication protocol study will be focused to WPAN networks [5], [6].

The IEEE (Institute of Electrical and Electronics Engineers) has developed two international standards with specific characteristics and interests. The first is detailed in the 802.15.1 standard for Bluetooth specifications [7]. The second is described by the 802.15.4 standard [8]. This are the basis adopted by the "Zigbee Alliance" to create a homonymous technology to meet the market needs for a low cost, low power consumption, safe and reliable communication system.

The Zigbee architecture stack is composed by a series of blocks called layers. Each layer performs certain functions necessary for the communication between modules. In this work, the layers model is based on the reference OSI (Open System Interconnection) model. The two bottom layers of the architecture -the **physical layer** (PHY) and **medium access control** (MAC) layer- are defined by the IEEE 802.15.4 standard. The Zigbee Alliance created the specifications for the **network** (NWK) and **application** layers, based on this standard. The application layer provides the functions required by the user, discovers the devices, maintains link tables between them and defines the role they have in the network. The NWK layer is responsible of handling the topology that the network acquires, the use of the protocols necessary to discover available channels and to enable or not communication security mechanisms. The MAC layer recognizes and validates the bit streams sent and the channel access mechanisms, ensures time slots correct management and implements security systems. The PHY is responsible of evaluating the channel to determine the most appropriate one to perform the transmission.

El IEEE standard defines two modes of operation: Full-Function Device (FFD) and Reduced-Function Device (RFD). The first one can operate as a coordinator or router while the latter one can only operates as an end-device. Simultaneously, the network layer defines three types of network topologies: star, tree and mesh depending on how coordinator, router and end-devices are connected. In this paper all the peripheral are set as end-devices and there is a single coordinator. They are connected using a star topology.

C. Technical description of the FPGA device

The system developed is implemented on a EP2C8Q208C8 Cyclone II FPGA device [9]. The chip is soldered on an educational board where the device can be configured with an USB Blaster cable. Also, it has a 50MHz clock (oscillator) and additional elements (e.g. buttons, connectors, LEDs, VGA port). The FPGA device works with the SRAM technology. It has a 622-pin encapsulated, 68.416 LEs (logic elements) and 36 M4K blocks (blocks of embedded memory). Each LE consists of a 4-input lookup table (LUT), a programmable flip flop and a signal used for carry and cascade operations. Each M4K block has a 4.608 bits memory which can be used to create logic functions.

D. Xbee radiofrequency modules

In this paper Xbee PRO Zb Modules [10], [11] (Digi International Inc., Minnesota, EEUU) are used to establish a Zigbee communication networks. The use of these modules decreases system development time because they already have the Zigbee stack and the RF transmitter/receiver incorporated. They operate as configurable modems and they are mounted on a XBoard development board. The modules operate in API (Application Programming Interface) serial interface and use the Remote Command Request (0x17) API frame type. They are externally configured using X-CTU free software [12].

Modules can send and receive messages up to 84 bytes. Before and after each message transmission a 11.5ms guard time should be expected. This is specified by the Zigbee protocol. In this work each message to be transmitted has 20 bytes, in which the direction of each home device (peripheral) wireless module and the action to perform are detailed. Different 21-message are generated and sent, 2 for each peripheral -one to activate it (ON) and other to disable it (OFF)-. The last message has a special command that acts on all the peripherals.

III. SYSTEM DESCRIPTION

Figure 2 shows the complete scheme of the logic design for the domotic system. The logic design is implemented with the Web Edition of the Quartus II 9.0 software using schematic mode and hardware description language (VHDL) [13]. The data entry is performed using a 4x3 matrix keyboard whose rows and columns are connected with the FPGA device. These are decoded using the **4x3_Matrix_Keyboard** logic block. The selection menu is displayed on a VGA monitor. The **Synchronism** logic block generates the HS and VS synchronism signals. The external peripheral control is implemented with **Peripherals** logic block which generates data packages for the activation or disabling of the home devices. Packages are sent to the coordinator module using an UART communication between it and the FPGA device. The coordinator sends the data by RF to each end-device connected to a peripheral. The **Text_gen** logic block takes pixel_x and pixel_y signals from **Synchronism** and generates characters that will be displayed on the screen. The output signal of this block and the output signal of the **Restriction 640x480** logic block enter to an AND gate. This ensures that no character is drawn in the not-visible area. The AND gate output enters to

Color logic block and generates the RGB signals for the colors to be displayed on the monitor. The **Control_Menu** logic block selects the characters to be displayed on the screen and selects peripheral status depending on the selection menu status (ON / OFF). **Control_Menu** block outputs are used by the **Peripherals** logic block to select the data packet that will be sent to the end-device of each peripheral.

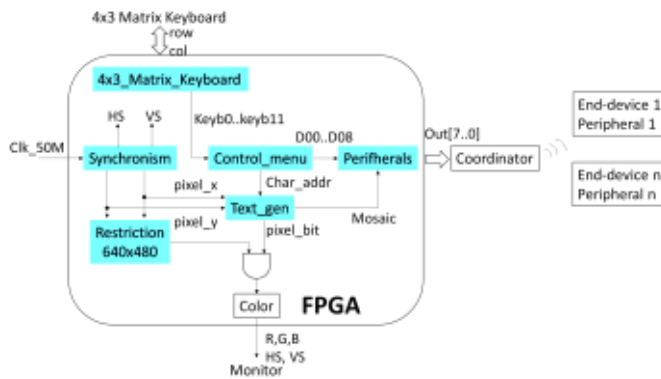


Fig. 2. Scheme of logic design for the complete system.

A. 4x3_Matrix_Keyboard

4x3 matrix keyboard is a device of 4 rows and 3 columns. It consists of conductive rows and columns. There is a mechanical push button where they cross together that establishes an electrical contact when it is pressed. The rows are configured as inputs and the columns are configured as outputs of the FPGA device. The rows have pull-up resistors to have a logic zero at the FPGA input when a column is pressed. The activation of a key may produce mechanical rebounds that are interpreted as repeated activations of the same key. To avoid this, **Debounce** blocks with a duration of 8ms are implemented. Figure 3 shows the parts of the **4x3_Matrix_Keyboard** logic block that indicates which key was pressed.

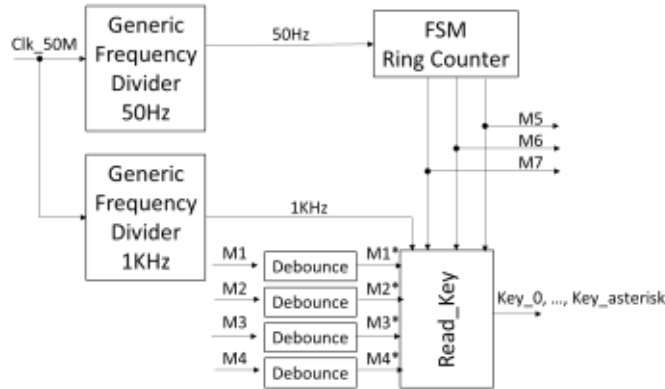


Fig. 3. 4x3_Matrix_Keyboard

The Clk_50M signal comes from the crystal of the educational board and enters to two generic frequency divider blocks (**Generic Frequency Divider**), which are configured to deliver two signals of 50Hz and 1kHz. This block is implemented with a Lpm_counter mega function configured to count up, 1000000 and 50000 module generics to establish

50Hz and 1kHz signals respectively. The 50Hz signal is connected to **FSM Ring Counter** logic block (Figure 3) and allows the transition of a ring counter implemented with a finite state machine (FSM). This machine generates the M5, M6 and M7 outputs that are connected to the columns of the matrix keyboard. **Read_Key** is synchronized with the 1KHz signal and decodes what key was pressed using "If" statements with the input values from the matrix keyboard rows (M1, M2, M3 and M4) and the outputs values from the **FSM Ring Counter** (M5, M6 and M7).

B. Synchronism

Figure 4 displays **Synchronism** logic block. This block has a Clk_50MHz input signal, and two outputs -HS and VS signals- that enter to the monitor; x_axis and y_axis signals represent horizontal and vertical monitor visible area coordinates respectively. HS and VS signals are showed in Figure 5. The Clk_50MHz signal enters to the **Generic frequency Divider** logic block that generates a 25MHz signal necessary for the VGA standard. The time T (25MHz) to display a pixel is 40ns [14].

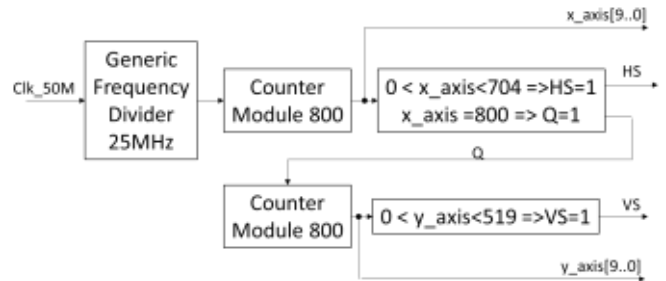


Fig. 4. Synchronism logic block.

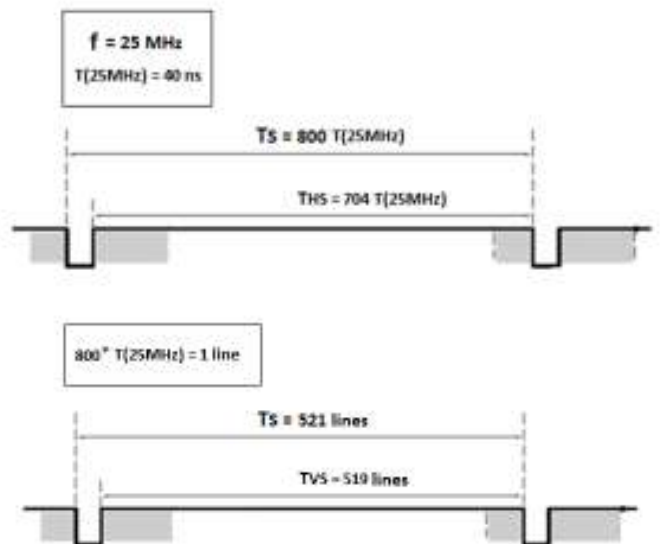


Fig. 5. HS (Up) and VS (Down) signals waveform.

C. Restriction 640x480

Figure 6 shows the monitor screen where there is a 640x480 visible area and a restricted area. The last area corresponds to the delay time where the electron gun of the

monitor should be turned off. This is carried out using “if” statement as follows:

If $x_axis < 48$ or $x_axis > 688$ or $y_axis < 29$ or $y_axis > 509$ then $S=0$ Else $S=1$

Where S signal enables the AND gate in order to draw in the screen only when $S=1$.

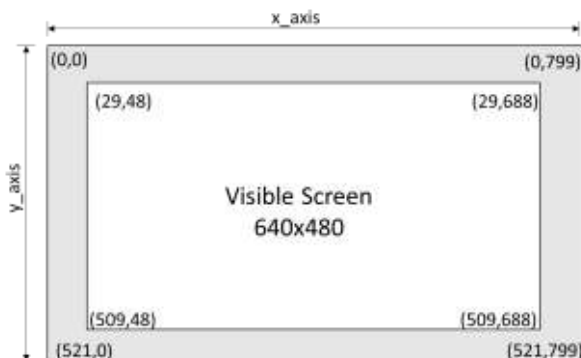


Fig. 6. Visible and restricted area of a monitor screen.

D. Text_gen

The 640x480 visible area of a monitor screen is partitioned in "mosaics" where each one represents a character. The character size is 8 pixels width x 16 pixels height. The screen is mapped into a 30x80 character matrix. Each pixel is associated with one location of the 30x80 characters matrix. The character patterns are stored in a ROM memory (Font ROM) implemented with a Lpm_ROM mega function. This works uses 100 different patterns including numbers, uppercase and lowercase letters, punctuation marks and special characters. The Font ROM has $2^{11} \times 8$ bits. The most significant seven bits (MSB) of the 11-bit address are used to identify the character and the less significant four bits (LSB) are used to identify the row of each character pattern. A logic level 1 or 0 in the Font ROM indicates that a pixel is intended to be showed or not on the screen respectively.

Figure 7 shows **Text_gen** logic block. The x-axis and y-axis signals enter to the **xy_pixel** block, x_pixel and y_pixel are coordinates of the monitor visible area. The seven and five MSB's of the x_pixel and y_pixel signals respectively, provide the x_mosaic and y_mosaic coordinates of a mosaic on the screen. The three and four LSB's of the x_pixel and y_pixel respectively, provide the row (x_rom) and the column (y_rom) within the Font ROM. The x_mosaic and y_mosaic signals enter to the **Control_menu** logic block that select the character to display ($char_addr$) on each mosaic. The $char_addr$ [6..0] are the seven MSB's of the **ROM Memory** address where the beginning of the character pattern to be displayed on the screen is. The $char_addr$ [6..0] signal is concatenated with rom_y [3..0] so the 11-bit address required for the 100 character patterns in the Font ROM is completed. $Font_word$ [7..0] is a 8-bit character pattern and enters to the **Mux** logic block which is a multiplexer that routes each bit with x_rom [2..0] signals. In this way, the pixels are plotted one by one when the electron gun is advancing through the monitor visible area.

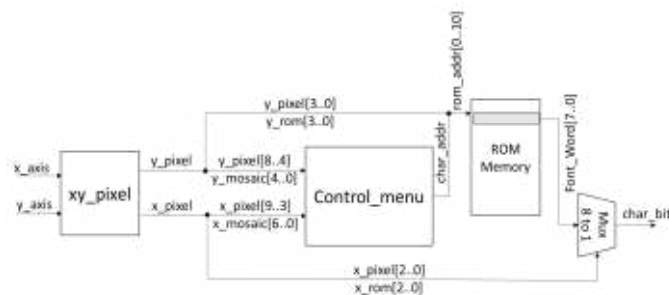


Fig. 7. Text_gen logic block.

E. Control_Menu

The **Control_menu** logic block (Figure 8) is implemented with a FSM. It takes the x and y coordinates of each mosaic and generates $char_addr$ signal. Depending on the FSM state, this block delivers the $char_addr$ signal with the ROM memory address of the character pattern to be displayed. Also, the FSM controls two peripherals in every one of the six home environments (alarm, bathroom, kitchen, bedroom, garage and living room) plus an alarm. Figure 8 show the FSM behavior.

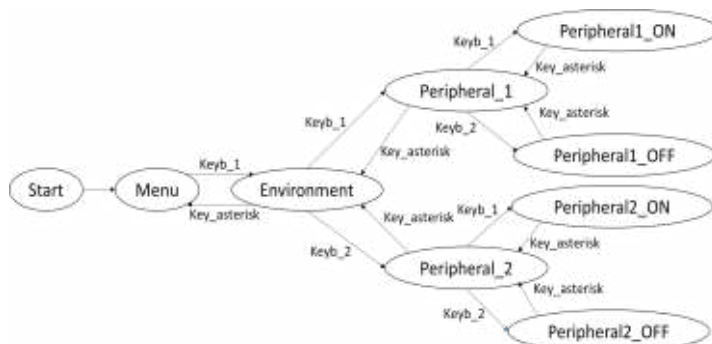


Fig. 8. Control_menu FSM

Control_menu allows the user to select the **Environment** state by means of **Menu** state pressing $keyb_1$ to $keyb_6$ for alarm, bathroom, kitchen, bedroom, garage and living room respectively. Pressing $Keyb_1$ or $Keyb_2$ allows the user to select the **Peripheral_1** or **Peripheral_2** state respectively for each environment. Finally, pressing $Keyb_1$ or $Keyb_2$ allows the user to select the **Peripheral_ON** or **Peripheral_OFF** state respectively for each peripheral. Pressing $key_asterisk$ there is turning back from any state. Transitions between states are synchronized with the 50Mhz clk signal. There is a **wait** state (not drawn in Figure 8) between states. This state allows only one transition when a key is pressed. The **Start** state displays on the screen a title and member names. To exit this state any key can be pressed.

F. Color

Figure 9 shows the **Color** logic block which is implemented through a FSM. Figure 10 shows its logic design. This FSM has $char_color$ and $background_color$ signals as inputs. These signals enter to a **Debounce** module to eliminate noisy signals produced by the rebound effect of the keys. There

is a **wait** state (not drawn in Figure 9) between states. This state allows only one transition to occur when a key is pressed. The FSM outputs (R1, G1, B1, R2, G2, B2) correspond to the RGB color values for the character and the background. The **Mux** multiplexer block captures these signals. S signal comes from **Text_gen** and controls the **Mux** logic block. When S is 0 or 1 a pixel or a background color is displayed on the screen respectively. R_out, G_out, B_out signals enter to monitor connector.

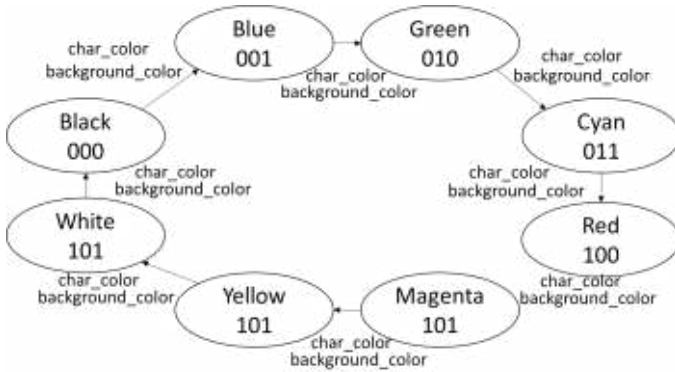


Fig. 9. FSM of the **Color** logic block.

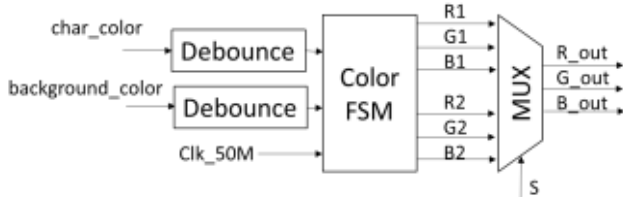


Fig. 10. Color logic block design.

G. Peripherals

Peripherals logic block (Figure 11) operates as a selector switch. It receives a Clk_50M clock signal and a 10-bit signal from the **Control_menu** logic block. It has two outputs: Tx and OK. The Tx signal is necessary for serial communication and the OK signal indicates the end of a transmission. A logic block (**block_D00..D08, all_off**) is selected with each one of the signals (D00 ... D08, all_off). Each logic block generates two data message to be transmitted (ON and OFF message) of 20-bytes each one. Two 20-states FSM are employed and implemented in VHDL respectively. For example, the ZigBee byte sequence for turn ON peripheral 1 is 7E 00 10 17 01 00 13 A2 00 40 3A 4A 2D FF FE 02 44 30 05 C9 and to turn OFF is 7E 00 10 17 01 00 13 A2 00 40 3A 4A 2D FF FE 02 44 30 04 CA. So, for example, if D00 is high level then **block_D00** enables the generation of a 1-byte of the ON message and the generation of an UART interface (Universal Asynchronous Receiver-Transmitter) by means of two FSM respectively. When D00 is low level, **block_D00** resets both FSM and it generates 1-byte of an OFF message in the same way with two FSM. When all_off signal is high level all peripherals are turned off in a consecutive manner excluding **block_D00** that turns on the alarm.

An 11-state FSM generates the Tx output of the UART interface with a 9600 baud rate. An 872Hz synchronism signal enables the UART interface FSM (1x872≈9600).

IV. RESULTS AND FUTURE DEVELOPMENTS

The simulations of the logic blocks and the logic resources used for all the logic design showed the expected results.

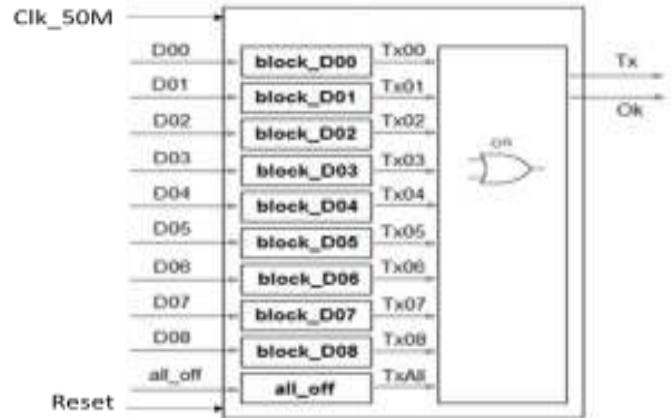


Fig. 11. Peripherals logic block

A. Simulations

In order to simulate the system, Quartus II wave editor is used. Figure 12 shows the waveforms of clk_25Mz, HS and VS. The times obtained are those established by the VGA standard.

The Figure 13 shows the signal waveforms of **Peripherals**. When "D00" is enabled, the message is transmitted by through a "Datos" signal synchronized with the "872Hz" signal. "Listo" signal indicates when a byte was sent. "Datos" and "Listo" signals are incorporated in order to test the system. The "9600Hz" signal is obtained from the "872Hz" signal and synchronizes the "Tx00" signal. Finally, Tx00 allows to generate the Tx signal of the UART interface for each enabled peripheral device. Figure 13 shows the correct system operation.

B. Physical resources.

The physical resources of the EP2C8Q208C8 integrated circuit used in this logic design are: 25 input/output pins, 16383 memory bits (a 10% of the total embedded memory) and 3995 logic elements (a 48% of the total logic elements).

C. Future developments.

The integration of the matrix keyboard and the user interface would be a significant improvement for user comfort.

A functional improvement for the system would be to incorporate the monitoring of certain parameters such as temperature, humidity and light intensity, and to develop a feedback control system for controlling a peripheral. To do this the Xbee pins must be configure as analog inputs and outputs and a PID (Proportional Integral Derivative) control block must be developed in the FPGA device.

V. CONCLUSIONS

[2] Zigbee Alliance. Especificación del protocolo Zigbee. Disponible en:

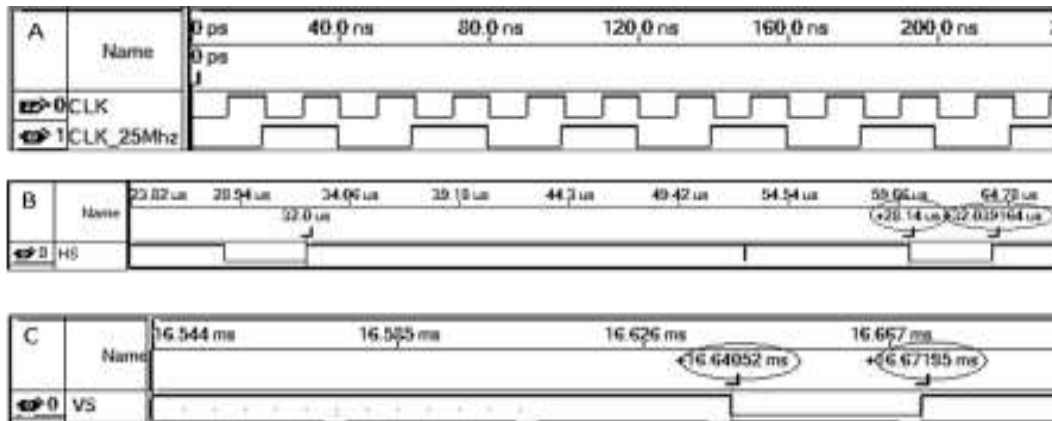


Fig. 12. (A) The 50MHz and 25MHz signals. (B) Horizontal synchronism signal –HS-. (C) Vertical synchronism signal –VS-

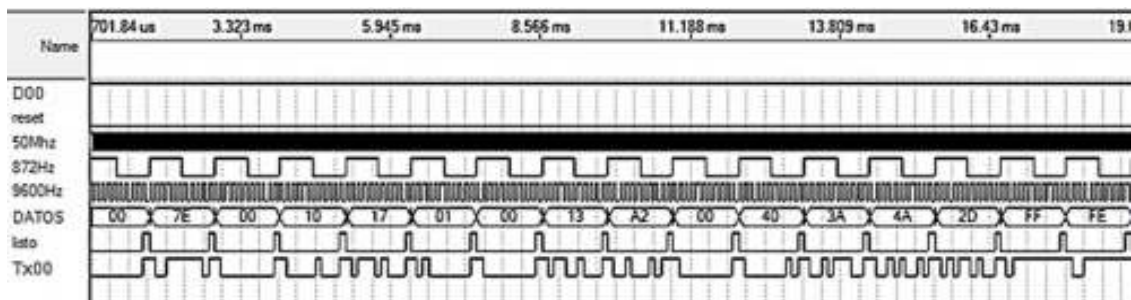


Fig. 13. Signal waveforms of Peripherals logic block

A home devices control system for home use has been successfully developed. The system is implemented on a FPGA device and allows the control of home devices such as lights, appliances and electric doors. The management is done through a wireless communication between a master device and slave actuators.

A video controller has been designed and programmed and successfully implemented using FPGA digital technology. It was designed according to the VGA standard that allows to graph alphanumeric characters on a monitor screen.

A control interface for a ZigBee wireless network has been successfully implemented in a FPGA. The interface generates control signals, builds the packet and sends it to the Zigbee coordinator module. The results show the correct behavior of the digital interface since it allows the communication among Zigbee modules using the specified protocol.

ACKNOWLEDGMENTS

This work was supported by grants from the UNT Research Council (CIUNT), 26/E547.

REFERENCES

[1] Gill, K. ; Shuang-Hua Yang ; Fang Yao ; Xin Lu “A zigbee-based home automation system”, Consumer Electronics, IEEE Transactions on, vol. 55, pp. 422-430, 2009.

www.zigbee.org/
[3] M A Gómez López, C B Goy, P C Bolognini, M C Herrera . “Implementación en FPGA de una interface de control de red inalámbrica ZigBee para señales biomédicas”,2011.
[4] Grupo de Procesamiento Digital de Señales e Imágenes – Pontificia Universidad Católica del Perú. “Muestra de una señal biomédica en tiempo real sobre un monitor VGA con resolución 640x480”. IX Workshop IBERCHIP, 2003.
[5] C. Goy, “Evaluación Tecnológica e Implementación de una Interfaz Inalámbrica”, Diploma Thesis, UNT, Tucumán, Argentina 2010.
[6] F. Pinedo, J.A García, “An experimental analysis of Zigbee Networks”, IEEE 978-1-4244-2413-9/08, 2008
[7] IEEE (Institute of Electrical and Electronics Engineers), Norma IEEE 802.15.1, 2003. Disponible en internet en: <http://standards.ieee.org/getieee802/802.15.html>.
[8] IEEE (Institute of Electrical and Electronics Engineers), Norma IEEE 802.15.4. Disponible en internet en: <http://standards.ieee.org/getieee802/802.15.html>, 2005.
[9] Altera, Cyclone II Device Handbook, 2008.
[10] MAXSTREAM, Datasheet y Manual de usuario de los módulos Xbee y XbeePro 802.15.4, 2009.
[11] XBee S2 Quick Reference Guide Disponible en: www.tunnelsup.com, 2012.
[12] X-CTU Configuration & Test Utility Software, 2008.
[13] F. Pardo, J. A. Boluda, Edición RA-MA Abril 1999. VHDL Lenguaje para Síntesis y modelado de circuitos”.
[14] VGA Timing Information, Página Web: www.epanorama.net/documents/pc/vga_timing.htm

Comparison of classical and chaotic sequences for DSSS

Kevin G. Patat and L. De Micco
Electronic Department, School of Engineering,
Universidad Nacional de Mar del Plata and CONICET,
email: kevinguillermopatat@gmail.com.

Abstract—Pseudo Noise codes (PN-codes) are finite sequences used in Spread Spectrum (SS) communication systems, a widely employed wireless communication technique. The selection of the best set of “PN-codes” (PN-family) is an important issue in Direct Sequence Spread Spectrum (DSSS). Among the advantages of the use of chaotic sequences in DSSS are the availability of a great number of them, the ease of their generation, as well as the inherent improvement in the security they achieve. In this paper a DSSS communication system has been implemented in two Cyclone FPGAs boards and Additive White Gaussian Noise (AWGN) has been added to the transmitted waveform in order to emulate the channel noise. The Bit Error Rate (BER) of the implemented DSSS system was evaluated showing that chaotic sequences present similar performance to the classical ones.

I. INTRODUCTION

The utilization of chaos in communication systems has attracted attention since the introduction of chaos synchronization [1].

Many proposals have been presented [2], in the case of Spread Spectrum communication systems many researches have realized analytical and theoretical analysis [3], [4], [5], [6], [7], [8].

On the other hand some FPGA implementations of non-chaotic DSSS systems were developed for different applications: In [9] direct sequence principle based Code Division Multiple Access (CDMA) transmitter and receiver was implemented in VHDL for applications in ADHOC networks and defense communication links.

Reference [10] describes the design of CDMA modem in FPGA based on DSSS. 8051 microcontroller & FPGA is linked with rf transmitter/receiver, a acknowledgement is being sent to FPGA once the bit is transmitted/receive by the 8051 microcontroller. At the receiver output of 8051 microcontroller is compared with the threshold levels using a comparator and this provides binary output which is the actually transmitted sequence.

This paper follows the analysis performed in [11] where PN classical and chaotic sequences were studied. There, a very involved statistical analysis based on a new global quantifiers was performed. A quantifier based on the Zipping Complexity Measure turns out to be a global optimization quantifier of the whole system. There it was concluded that chaotic generated PN-families were competitive according to the quantifiers used there.

In this work a further analysis was performed, by means of probability binary error rate. A hardware platform was

designed and implemented based on two FPGA boards, one for the DSSS transmitter and the with Gaussian noise generator and the other for the DSSS receiver. To our knowledge this is the first time this issue is systematically studied using a real hardware implementation.

The paper is organized in five sections following this introduction: in Section II the SS technique are reviewed and the multiple-access scheme Code Division Multiple Access (CDMA). In section III, the main characteristics of the different PN-codes used are mentioned. Section IV presents the test bench employed in order to do the measurements, it constitutes of the transmitted and received implemented system, and the AWGN noise generator. Also in this section, the main procedures are explained. Results are reported in Section V. Finally, Section VI deals with conclusions.

II. SPREAD SPECTRUM COMMUNICATION SYSTEM

A Spread Spectrum system is a communication system [12] that produces a transmission signal with a bandwidth much greater than the message bandwidth. A SS system distributes the transmitted signal energy over a wider bandwidth so the signal to noise ratio at the receiver input is low. However, the receiver is capable of successfully recover the message because the waveform that spreads the message bandwidth (pseudorandom or PN code) is actually deterministic, and thus can be reproduced. In this case the receiver can undo the effect of spreading by correlating the received signal with a replica of the pseudorandom code. Applications of this technology include cell phones, wireless data transmission and satellite communications.

All SS systems must satisfy two criteria:

- First Transmitted signal bandwidth must be much larger than the message one.
- Second PN code must be independent of the message and known by the receiver.

The main advantages of SS are:

- Resistance to intentional and unintentional interference, a very important quality when operating in congested areas.
- Ability to eliminate or mitigate the effect of multi-path propagation, which is a major obstacle in urban communications.
- The same frequency band can be share with other users, due to its similarity to a noise signal.

- In any situation all the bandwidth is used.
- Privacy due to unknown random codes, the codes applied are in principle unknown to an unwanted user.
- Possibility of random access, users can start their transmission at any instant of time.

According to the methods used for spreading the data, SS techniques are classified as:

- Direct Sequence.
- Frequency Hopping.
- Time Hopping.
- Chirp.
- Hybrid Methods.

In this work we have employed Direct Sequence technique (DSSS).

The ratio of transmitted bandwidth to information bandwidth is called the processing gain G_p of the DSSS system:

$$G_p = \frac{B_t}{B_i} \quad (1)$$

Where, B_t is the transmission bandwidth and B_i is the bandwidth of the information bearing signal.

As said in a DSSS communication systems, the bandwidth of the signal is commonly expanded to several orders of magnitude, before transmission. In these systems all users transmit in the same bandwidth simultaneously. In a multi-user environment, users can share the same band by using DSSS technique.

As different CDMA users take different PN sequences, each CDMA receiver can discriminate and detect its own signal, by cross correlation of the received signal with each of the possible user code sequences.

III. PN SEQUENCES

PN codes, as said, play a fundamental role in the performance of the DSSS system. A PN-code is a N -bit binary string so there exist 2^N number of possible different strings. This number increases exponentially with the number of bits but most families have a considerable lower number of usable PN-codes (family members) because of the restrictions imposed by the following optimization requirements: [13].

- All the members of a given PN-family must be generated using only one algorithm: this is a hardware restrictions and there are several strategies to produce PN-families reported in the engineering literature. Each strategy gives rise to a different family. Chaotic maps are good candidates because they are very simple to be implemented.
- Availability of a large number of elements: the number of members depends on the family generation method but usually it is much lower than 2^N . This number determines the maximum quantity of users sharing the spectrum. Again chaotic maps are good candidates

because they can provide in principle infinite codes (in fact limited by the employed arithmetic).

- The autocorrelation of each PN-code must be as similar as a delta function: this makes each PN-code easily distinguishable from a circular time-shifted version of itself to make the synchronization step faster. This property is also important for reducing the effect of multipath propagation.
- The cross correlation between any pair of PN-codes must be as lower as possible (ideally a null function): each sequence in the family must be easily distinguishable from (a possibly circular time-shifted version of) every other signal in the same family. This requirement is important during the detection step for minimizing the multiuser interference.
- Spread spectrum: the modulated signal must have a uniform spectrum over the assigned bandwidth. This is a requirement common to all digital systems in order to reduce the Electromagnetic Interference (EMI).

Chaotic families studied in this paper have two important advantages: the big number of sequences generated by a chaotic map and the ease of their generation. In this paper they are compared with the commonly used families in the engineering literature. A brief description of each studied family follows and their main characteristics.

A. Classical PN sequences

Non chaotic PN-codes studied in this paper are:

(a) M-sequences: They are the most widely studied PN-codes and they are the basis for other PN-families. Each family is generated by a m -stage shift register with linear feedback according to a primitive polynomial. The length of each PN-code is $N = 2^m - 1$. The spreading and autocorrelation properties are almost ideal but the cross-correlation between any pair of codes is a periodic function with high peaks. The number of members of each family is small.

(b) Gold-codes: Gold PN-codes are obtained by means of an exclusive OR operation between two preferred M-sequences of the same family. Preferred M-sequences are pairs with minimal cross correlation. Including the preferred pair, a total of $2^m + 1$ Gold Codes can be produced from any m -stage feedback shift register. The cross-correlation between any pair of Gold PN-codes is uniform and bounded. Gold PN-families have a large number of members. The spread spectrum properties are not optimal.

B. Chaotic PN sequences

Chaotic PN-codes studied in this paper are generated by iterating two chaotic maps:

(c) Chaotic TWBM: Sequences generated by Three-Way Bernoulli Map.

(d) Chaotic FWTSM: Sequences generated by Four-Way Tailed Shift Map.

(e) Chaotic TWBMd: Sequences generated by the d th

TABLE I. ILLUSTRATING THE **Skipping** PROCEDURE

Chaotic sequence (\mathbb{R})	Skipped sequence $\equiv M^d$		
	$d = 2$	$d = 3$	$d = 4$
0.010559404			
0.041791613	0.041791613		
0.160180296		0.160180296	
0.538090276	0.538090276		0.538090276
0.994196523			
0.023079185	0.023079185	0.023079185	
0.090186145			
0.328210418	0.328210418		0.328210418
0.881953358		0.881953358	
0.416446528	0.416446528		
0.972075269			
0.10857976	0.10857976	0.10857976	0.10857976
0.387160782			
0.949069244	0.949069244		
0.193347257		0.193347257	
0.62385638	0.62385638		0.62385638

skipping of TWBM map.

(f) Chaotic FWTSMd: Sequences generated by the d th skipping of FWTSM map.

The sequences generated by each map are transformed into a binary code by means of the usual symbolics dynamics rule ($[0, 0.5) \rightarrow 0$, $(0.5, 1] \rightarrow 1$). The number of family members is limited by the periodicity of the pseudo chaotic digital sequence.

Also, chaotic maps were randomized by means of a randomizing procedure namely skipping [14]. This is the case for TWBMD and FWTSMD sequences. Skipping procedure consists on “skip” $(d - 1)$ values of the chaotic sequence to get the TWBMD and FWTSMD sequences. In other words, one employs, instead of the original map M , its d -times iterated one M^d . This randomization technique is routinely (and successfully) used with piecewise linear maps in many applications [15]. In Table I, an example with different values of d is displayed.

Autocorrelation, cross correlation and Spreading Spectrum properties of these codes were previously studied in [13]. Now, the intention is to go one step further by analyzing the error rate obtained in an implemented DSSS system.

IV. DESCRIPTION OF THE IMPLEMENTED SYSTEM

The communication system was implemented in two Altera development boards, one for the transmitter system (Cyclone IV) and one for the receiver system (Cyclone III). Both systems were developed using VHDL.

The transmission system is responsible for spreading the spectrum of data to be sent via DSSS technique. These data enter into the transmitter via an infrared communication.

Meanwhile the receiving system is responsible for decoding and recovering the original message, and then, sending it to a PC via RS232 communication.

The received data is presented in a graphical interface developed in Matlab software was performed as shown in Fig. 1, in which the user can configure the characteristics of the serial communication between the receiver board and PC. In

addition the received data can be stored in a file, with extension txt, xls, or mat, which is selected by the user.

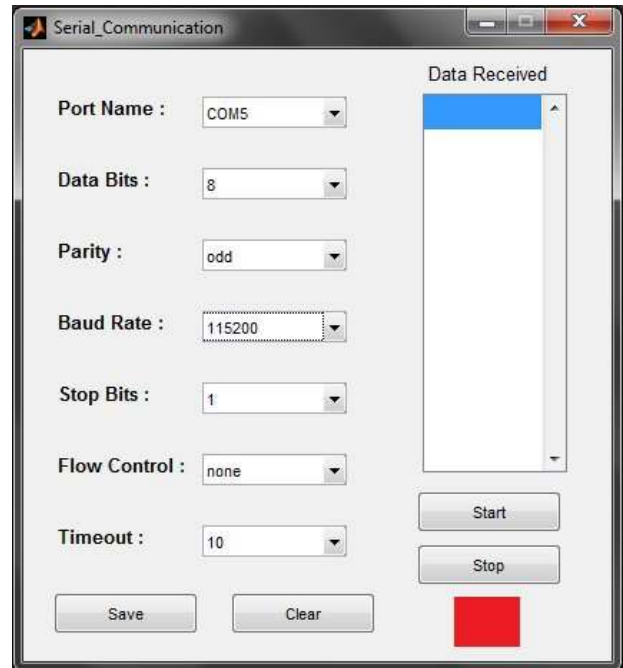


Fig. 1. Graphic interface

A. DSSS transmission system

Fig. 2 shows a block diagram of the DSSS transmission system and also the AWGN generator that is added to the data, all these are implemented in the Cyclone IV board. The AWGN generator implementation is detailed in section IV-C. The transmitter is divided in two main blocks. The first one refers to the entry of data to be transmitted and in the figure corresponds to the IR receiver block. While, the second stage is related to the spreading process of the data.

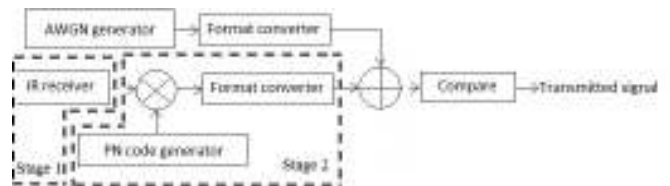


Fig. 2. Block diagram of the circuitry implemented in the Cyclone IV board: transmitter DSSS system plus the addition of AWGN.

In order to load the data to be transmitted, the first stage employs an infrared remote control receiver module that is incorporated into the development board [16]. Thus, each time the remote control button is pressed a frame is emitted, which is identified by IR receiver.

Meanwhile, the data transmission process is subdivided into three blocks. The first one is in charge of sending a preamble, this is, sending a determined number of times the pseudorandom sequence, in order to allow the receiver to achieve the synchronization. The quantity of PN sequences

needed to be send in this preamble is directly related to the number of bits the sequence has. This relationship is shown in equation 2.

$$M = 2(n - 1) + 3 \quad (2)$$

where “M” is the number of times the PN sequence should be send, and “n” is the length of the sequence.

The second block involves the reception of data from the infrared communication and the creation of data packets. Thus each group consists of 10 bits, 8 bits preceded by a start bit and ended by a stop bit.

Finally, regarding the third block, which is composed by an OR and XNOR gates is associated with the final transmission of the system, i.e., it combines the above two blocks, resulting in the signal to be transmitted. Note that the synchronization of the entire system must be perfect, since it is necessary that each period of the pseudorandom sequence is associated with a high level or a low level. Thus, the final waveform represents each data bit by the PN sequence chosen if it is desired to transmit a 1, or the inverted sequence to transmit a 0.

B. DSSS reception system

In Fig. 3 is shown a block diagram of the DSSS reception system. This is composed of three stages: signal acquisition, monitoring or tracking signal and data recovery.

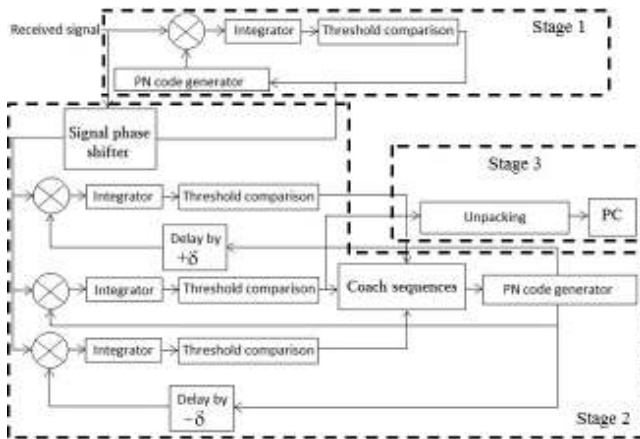


Fig. 3. Block diagram of the DSSS receiver system.

The first stage is composed of a serial acquisition system. It works by comparing the output of the correlation between the PN sequence generated at the receiver system and the received signal with a certain level, through a threshold detector, with the objective of synchronizing the transmitting and receiving clocks. This acquisition process is shown in figure 4.

The second stage is to prevent possible misalignments of the clock that was already synchronized in the acquisition stage. For this, three identical pseudorandom sequences are generated locally. One of these sequences is in phase and the other are one cycle forward and one cycle backward.

The operation consists of correlating the received signal with each of the three sequences and then decide by a threshold

detector, which PN sequence produces the highest correlation result. This sequence will be the one that is synchronized with the received signal. This tracking process is shown in figure 5.

Finally, the third stage is in charge to unpack the information to get the original message, and then send it to the PC via RS232. Then, the received data can be seen by the user through an application programmed with Matlab. Note that the previous two stages have eliminated the spread spectrum effect of the incoming signal, so the recovery stage receives data packets containing only the sent information.

C. Additive Withe Gaussian Noise

Wireless data transmissions are affected by noisy channels. In such environments the power level of the signal rapidly decays and it could become strongly affected by the noise in the channel. Under these circumstances, the performance of the scheme can be measured by the BER. Additive White Gaussian Noise was added in the channel with different signal to noise (S/N) ratios. For this, a core from Open Cores [17] named Gaussian Noise Generator (GNG) was employed. This core is based on 64-bit combined Tausworthe generator and an approximation of the inverse normal cumulative distribution function, which obtains a Probability Density Function that is Gaussian to up to 9.1σ . The generated noise is quantized to 16 bits with 5 bits of integer and 11 bits of fraction. At this stage, the FPGA works with this arithmetic in order to emulate he channel noise, so the DSSS output is represented in the same format. The *Format converter* block of Fig. 2 is in charge of doing this. Once the sum is concluded, the arithmetic returns to be binary. Fig. 6 shows how the noise sequence is added to the signal waveform to be transmitted. These data were collected by the Signal Tap tool provided by ALTERA and plotted in Matlab. Fig. 6 a) shows in black the DSSS output, it corresponds to the output of stage 2 in Fig. 2.

Fig. 6 b) corresponds to the obtained signal after the adder in Fig. 2, the black signal is the resulting waveform of the addition of the DSSS output with the AWGN, finally the gray waveform is the transmitted signal in Fig. 2 after the Compare block. Note that the channel is emulated inside the FPGA board, so the transmitted signal is actually the received signal.

Fig. 7 a), b) and c) show the original and resulting waveform for three different S/N ratios.

V. RESULTS

In order to test the performance of DSSS communications system several experimental transmissions were done. As said, different families of classical and chaotic PN sequence where employed adding different levels of AWGN. Sections III-A and III-B detailed the PN sequences evaluated, all of them having 63 bits of length. For this analysis the bit error probability (BER) is estimated as shown in 3

$$BER = \frac{\text{number of failed bits}}{40000000 \text{ transmitted data}} \quad (3)$$

Main results are shown in Fig. 8 where the BER obtained when transmitting with different signal to noise ratios using classical and chaotic codes are shown. It can be seen that in

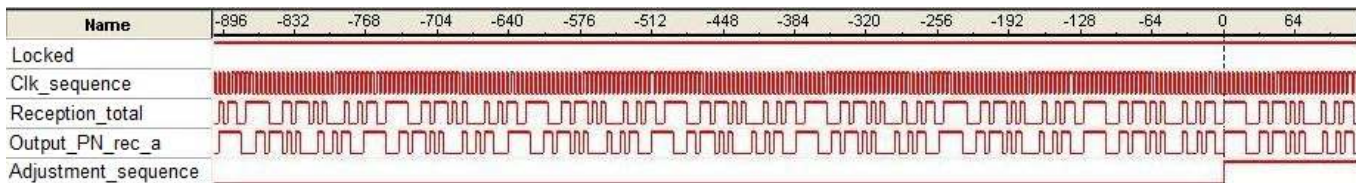


Fig. 4. Acquisition process for M-sequence of 15 bits length.

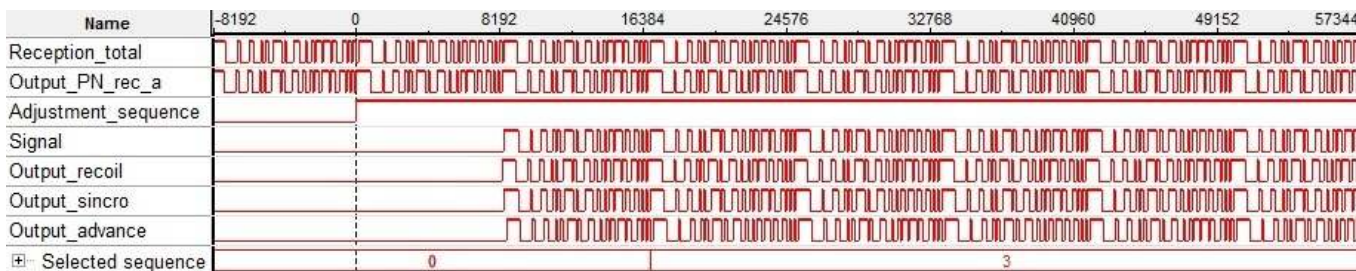
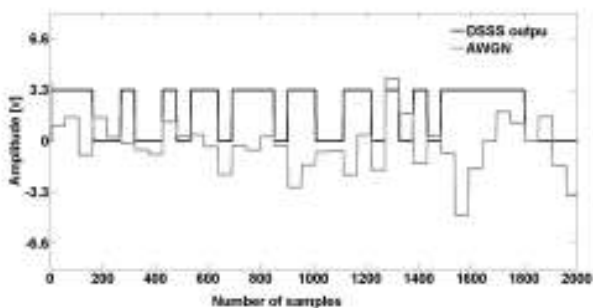
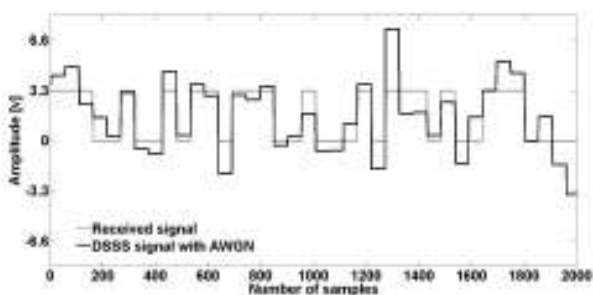


Fig. 5. Tracking process for M-sequence of 63 bits length.



(a) DSSS output (black), AWGN (gray).



(b) Signal received (gray), DSSS output with noise (black).

Fig. 6. Transmitted signal with AWGN



(a) $\frac{S}{N} = -3dB$. (b) $\frac{S}{N} = 1dB$. (c) $\frac{S}{N} = 3dB$.

Fig. 7. Chaotic code affected with Gaussian noise (above) and the same chaotic code without noise (below).

spite of the Gold and m-sequences present slight lower values of BER, the chaotic ones perform well, specially the skipped ones.

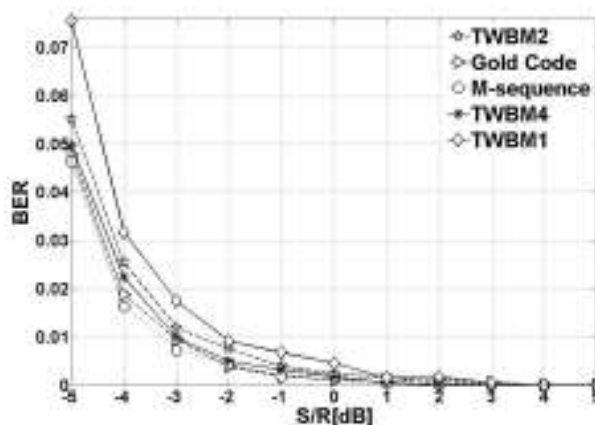


Fig. 8. DSSS system performance

Regarding the hardware implementation of the system Table II shows the resources used in the transmitter and receiver.

TABLE II. MAIN CHARACTERISTICS OF THE FPGA HARDWARE IMPLEMENTATION.

Application	Maximum operation freq.	Logic Elements	Memory bits	Registers
Transmitter	229.31MHz	201	0	150
Receiver	188.39MHz	1140	0	451

VI. CONCLUSION

A FPGA implementation of DSSS transmitter and receiver was developed and used to experimentally evaluate the BER performance of the system. AWGN was added to the transmission in order to explore the system's behavior in different scenarios.

Using this platform classical and chaotic PN-families were studied. Results obtained show that chaotic families behave almost as well as classic ones. Also, it was shown that skipping improves randomness (mixing) of the chaotic sequence, this is also reflected in the BER obtained when using skipped chaotic codes. This is consistent with the fact that they have equivalent correlation and spreading properties than the classical ones (studied in [13]).

This, together with the fact that they can have a higher number of members make them extremely attractive for CDMA applications. Furthermore they can be easily implemented and consequently they are good candidates to be used in real systems.

Our present conclusions are consistent with those of previous work.

ACKNOWLEDGMENT

This work was partially supported by the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina, ANPCyT and UNMDP Argentina. The DE2-115 board used in this project was kindly donated by ALTERA within The Altera University Program.

REFERENCES

- [1] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Phys. Rev. Lett.*, vol. 64, no. 8, pp. 821–824, Febrero 1990.
- [2] M. Kennedy, "Communicating with chaos: State of the art and engineering challenges," in *Proc. NDES*, vol. 96, 1996, pp. 1–8.
- [3] G. Heidari-Bateni and C. D. McGillem, "A chaotic direct-sequence spread-spectrum communication system," *IEEE Transactions on Communications*, vol. 42, no. 234, pp. 1524–1527, 1994. [Online]. Available: <http://dblp.uni-trier.de/db/journals/tcom/tcom42.html/Heidari-BateniM94>
- [4] G. Mazzini, G. Setti, and R. Rovatti, "Chaotic complex spreading sequences for asynchronous cdma - part ii: Some theoretical performance bounds," in *IEEE Transaction on Circuit and Systems I: Fundamental Theory and Applications*, 1998, pp. 496–505.
- [5] X. Yongxiang, S. Xiuming, R. Yong, Y. Xunhe, and L. Feng, "Correlation properties of binary spatiotemporal chaotic sequences and their application to multiple access communication," *Physical Review E*, vol. 64, no. 6, p. 067201, 2001.
- [6] V. Varadan and H. Leung, "Design of piecewise maps for chaotic spread-spectrum communications using genetic programming," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 49, no. 11, pp. 1543–1553, 2002.
- [7] H. Saigui, Z. Yong, H. Jiandong, and B. Liu, "A synchronous cdma system using discrete coupled-chaotic sequence," in *Southeastcon'96. Bringing Together Education, Science and Technology., Proceedings of the IEEE*. IEEE, 1996, pp. 484–487.
- [8] G. Mazzini, G. Setti, and R. Rovatti, "Chaotic complex spreading sequences for asynchronous ds-cdma-part I: System modeling and results," *IEEE Trans. Circuits Sys. I*, vol. 44, no. 10, pp. 937–947, 1997.
- [9] B. Sreedevi, V. Vijaya, C. Rekh, R. Valupadasu, and B. Chunduri, "Fpga implementation of dsss-cdma transmitter and receiver for adhoc networks," in *Computers Informatics (ISCI), 2011 IEEE Symposium on*, March 2011, pp. 255–260.
- [10] A. Tripathi and M. S. Korde, "Dsss based cdma modem using fpga & microcontroller," in *Proceedings of the 4th International Conference on Communications and Information Technology*, ser. CIT'10. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2010, pp. 128–130. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1864098.1864122>
- [11] L. De Micco, C. M. Arizmendi, and H. A. Larrondo, "Zipping characterization of chaotic sequences used in spread spectrum communication systems," *Institute of Physics Conference Proceedings 913*, pp. 139–144, 2007.
- [12] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread spectrum communications handbook*. McGraw-Hill New York, 1994, vol. 2.
- [13] L. De Micco, C. M. Arizmendi, and H. A. Larrondo, "Zipping characterization of chaotic sequences used in spread spectrum communication systems," *American Institute of Physics Conference Proceedings 913*, pp. 139–144, 2007.
- [14] L. De Micco, C. M. González, H. A. Larrondo, M. T. Martin, A. Platinio, and O. A. Rosso, "Randomizing nonlinear maps via symbolic dynamics," *Physica A*, vol. 387, pp. 3373–3383, 2008.
- [15] G. Setti, G. Mazzini, R. Rovatti, and S. Callegari, "Statistical modeling of discrete-time chaotic processes: Basic finite-dimensional tools and applications," *Proceedings of the IEEE*, vol. 90, no. 5, pp. 662–689, Mayo 2002.
- [16] Altera, *Cyclone IV Device Handbook*.
- [17] G. Liu. (2015) Gaussian noise generator (gng). [Online]. Available: <http://opencores.org/>

Microprocesador Embebido en Fpga para el Tratamiento de Ablación por Radiofrecuencia en el Hígado

Oscar Fernando Gaidos Rosero

Departamento de Engenharia Elétrica, Estudante
Universidade de Brasília - UnB
Brasília, Brasil
ogaidos@gmail.com

Suélia Rodrigues Fleury Rosa

Departamento de Engenharia Elétrica, Professora
Universidade de Brasília - UnB
Brasília, Brasil
rodrigues.suelia@gmail.com

El presente trabajo describe el diseño, desarrollo e implementación de un sistema híbrido reconfigurable para el tratamiento térmico de cáncer en el hígado, conocido como ablación por radiofrecuencia (ARF), que es la quema de células indeseadas del cuerpo humano a través de la aplicación de ondas electromagnéticas de radiofrecuencia.

La ablación por radiofrecuencia es una terapia para tumores mínimamente invasiva. En esta intervención un electrodo tipo aguja se inserta dentro del tumor y el tejido entorno de él se calienta debido a la radiación. El objetivo es destruir el tumor con un margen de seguridad, pero con el mínimo de daños para el tejido saludable circundante. Hoy en día, aunque su intervención es realizada en la práctica clínica, su resultado es difícil de pronosticar y los resultados satisfactorios dependen del tipo de control de potencia de salida del generador de radiofrecuencia. La extensión exacta de la zona de necrosis es difícil de planear y controlar. Uno de

los problemas más comunes en electro-cirugía son las quemaduras y heridas por exceso de potencia o altas densidades de corriente en regiones imprevistas como la espalda o las piernas, ya que en estos lugares está localizado un electrodo de retorno del equipo electro-quirúrgico. El diseño de un sistema digital para monitorizar y controlar en tiempo real la potencia que circula por el paciente basado en la placa de desarrollo Stratix III EP3SL150N que tiene un procesador embebido en FPGA, podría dar una solución alternativa a los problemas de ARF con el fin de aumentar la seguridad y la eficacia del equipo médico. Este sistema digital además permitirá la visualización y control de la temperatura de ablación y la impedancia eléctrica del tejido a través de una LCD gráfica con touch screen (modelo AGM12864A-803T). Finalmente estos datos se envían por medio de Bluetooth a un Tablet donde corre un aplicativo para sistema operativo Android.

Foro Tecnológico

Pósters

Robótica

Módulo Tacómetro para Motores Brushless

Gabriel Infante, Jeremías Báez Carballo, Federico Díaz Báez, Cristian David Cavenio
Departamento de Ingeniería Electrónica
Facultad Regional Córdoba
Universidad Tecnológica Nacional
{g.o.infante, jere.baez91, f.diazbaez, cristian.cavenio}@gmail.com

Resumen—En este trabajo se presenta el desarrollo de un sensor para medir la velocidad instantánea de un motor brushless. Este tiene dos aplicaciones: realimentar la velocidad de los motores al autopiloto y servir como instrumento de ensayo. Este sensor está basado en la medición de la señal aplicada a dos fases cualesquiera. Aquí se presenta el sistema de adquisición y acondicionamiento de dicha señal y su posterior procesamiento. Una vez implementado, se realizó una contrastación del sensor utilizando un instrumento patrón, resultando exitosa. También se determinó que las mediciones del sensor presentan una distribución gaussiana y además se muestra el ensayo realizado para determinar la respuesta de un motor a un escalón.

I. INTRODUCCIÓN

En los últimos años, los vehículos aéreos no tripulados (UAV, Unmanned Aerial Vehicle) han ganado gran popularidad en áreas tan diversas como investigación, agricultura, vigilancia, cine, fotografía, etc. En estas aeronaves, la energía es un recurso muy preciado debido a que es almacenada en baterías. Para propulsar estos vehículos es necesario utilizar motores de alta eficiencia, elevado empuje y reducido tamaño. Los motores *brushless* reúnen todas estas características. En la Fig. 1a se muestra un modelo comercial de motor brushless usado en aeromodelismo. Como contrapartida, el control de estos motores no puede realizarse de forma directa como un motor de corriente continua convencional, debido a que son trifásicos. Para ello se utilizan controladores electrónicos, más precisamente inversores, conocidos como ESC (*Electronic Speed Controller*). Estos módulos, cuyo aspecto físico se observa en la Fig. 1b, poseen una gran versatilidad, ya que permiten definir la velocidad deseada del motor tomando como referencia el ciclo de trabajo de una señal de PWM (*Pulse Width Modulation*).

Cuando sobre una aeronave se implementa un sistema de control de orientación y/o posición se utiliza como realimentación señales provenientes de los distintos sensores a bordo (acelerómetro, giróscopo, magnetómetro, etc.); mientras que la salida de este control es una velocidad de rotación de los motores. Esta debe ser expresada en porcentaje de ciclo de trabajo de la señal de PWM, para luego ser aplicada al ESC. Sin embargo, los ESC no realimentan información sobre los motores. Es decir, el autopiloto no tiene manera de determinar en el estado de estos. Además, ante un desperfecto en alguno de los motores, el autopiloto tampoco tiene información de esta situación.

En este trabajo se presenta el desarrollo de un sensor para medir la velocidad instantánea de un motor brushless. Este tiene dos posibles aplicaciones: por un lado, realimentar la velocidad de los motores al autopiloto, y por otro, servir como instrumento de ensayo para este tipo de motores.

Este trabajo surge en el marco del proyecto Quadrotor Autónomo de Arquitectura Abierta (QA3) [?] desarrollado en el Centro de Investigación en Informática para la Ingeniería (CIII), perteneciente a la Universidad Tecnológica Nacional, Facultad Regional Córdoba. En la Fig. 2 se puede observar el aspecto general del QA3.

El presente trabajo se organiza de la siguiente manera: en las secciones II y III se muestra los desarrollos de hardware y firmware,

respectivamente. En la sección IV se muestra la implementación final del sistema y los resultados obtenidos. Finalmente, en la sección V se muestran las conclusiones y trabajos a futuro.

II. ADQUISICIÓN Y ACONDICIONAMIENTO DE SEÑAL

El objetivo de este sensor es determinar la velocidad instantánea de un motor brushless. Para lograr esto existen varios métodos, algunos de los cuales utilizan sensores ópticos o magnéticos (de efecto Hall). En este trabajo, sin embargo, se presenta otro método, basado en la medición de la señal aplicada a dos fases cualesquiera. En esta sección se presenta el sistema de adquisición y acondicionamiento de dicha señal para poder ser interpretada.

La señal que se obtiene entre dos fases del motor es de naturaleza diferencial y se observa en la Fig. 3a. La velocidad a la que el motor rota depende de la frecuencia de la señal de excitación. La expresión



(a) Motor brushless de utilización en aeromodelismo. (b) Modelo comercial de ESC.

Fig. 1: Conjunto motor - ESC



Fig. 2: Quadrotor modelo QA3 Mini.

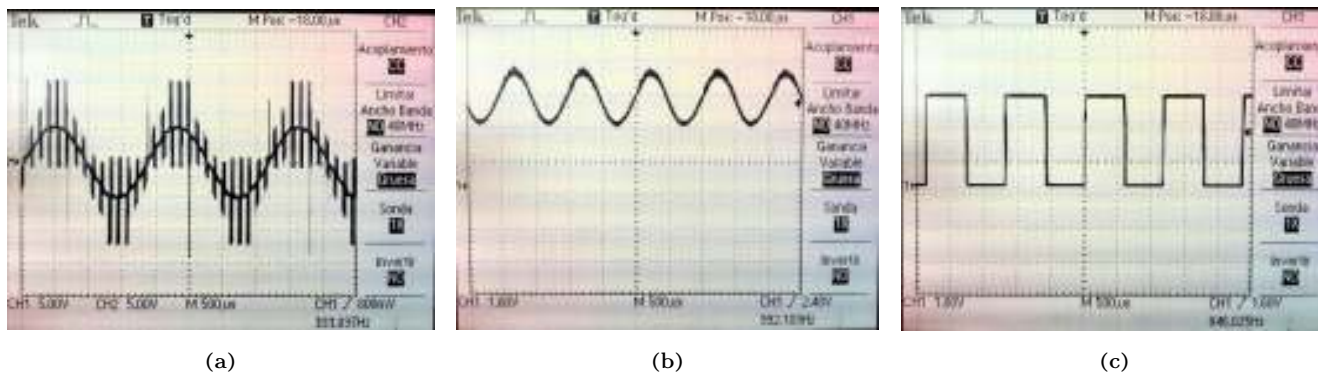


Fig. 3: Capturas sobre las distintas etapas del acondicionador. (a) entrada al circuito, (b) salida del filtro y (c) salida del acondicionador.

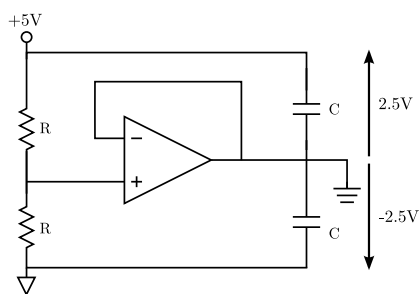


Fig. 4: Esquemático de la fuente de alimentación partida implementada.

(1) relaciona la frecuencia de esta señal (f_c) con la velocidad de rotación (ω).

$$\omega = \frac{60 f_c}{p} \quad (1)$$

donde p es el número de pares de polos; ω y f_c vienen dados en rpm y Hz, respectivamente.

Para poder determinar la frecuencia de la señal de la Fig. 3a es necesario un procesamiento tanto analógico como digital. El primero es llevado a cabo por un circuito acondicionador de señal, mientras que un microcontrolador realiza el segundo. El objetivo del acondicionador es convertir la señal con la que el motor se alimenta en un tren de pulsos de igual frecuencia. Luego, el microcontrolador se encarga de determinar la frecuencia del tren de pulsos y realizar las operaciones necesarias para traducirlo en velocidad de rotación aplicando (1).

En la Fig. 6 se muestra el esquemático del circuito implementado. Este consta de cuatro partes: un atenuador (a), un amplificador diferencial (b), un filtro pasa bajo (c) y un comparador (d). El primero se debe a que las señales presentes en la entrada son de gran amplitud, lo que saturaría el amplificador diferencial. La siguiente etapa cumple la función de transformar la señal obtenida a los bornes del motor (Fig. 3a) en una señal de modo común. En este caso, la etapa tiene ganancia unitaria. Luego, se implementa un filtro pasa bajo de topología Sallen-Key de segundo orden [1]. La función de este filtro es eliminar las componentes de alta frecuencia presentes en la señal de la Fig. 3a, que se deben a la conmutación inherente de los inversores. Para determinar la frecuencia de corte del filtro, se hizo girar el motor a la máxima velocidad y se midió la frecuencia fundamental de la tensión de excitación, resultando ser de 1.3kHz. Al ser esta la frecuencia máxima de interés, el filtro se diseña para esa

frecuencia de corte. La señal obtenida luego del filtrado se observa en la Fig. 3b. En la última etapa, utilizando un comparador, la señal filtrada se convierte en un tren de pulsos de igual frecuencia, como se observa en la Fig. 3c.

Para el correcto funcionamiento de un filtro activo (como el presentado en la Fig. 6c) es necesario que los amplificadores operacionales sean alimentados con una fuente simétrica. Sin embargo, en esta aplicación solo se dispone de una fuente única de 5V. Para cumplir con lo mencionado, se implementa el circuito de la Fig. 4 [2], cuya función es generar un punto de referencia virtual con su respectivo positivo y negativo. En este circuito, el nivel de la referencia virtual es la mitad de la tensión de alimentación. Por lo que, la nueva fuente de alimentación provee al circuito con tensiones de $\pm 2,5V$.

Este sistema puede ser integrado al conjunto de sensores a bordo y también utilizado para realizar diferentes ensayos sobre un motor. Debido a esto, el sistema presenta dos modos de funcionamiento, que hemos decidido llamar: modo *sensor esclavo* y modo *sensor independiente*, respectivamente. En ambos, el hardware utilizado es el mismo. La única diferencia es el canal por el cual se transmite la información obtenida. En el modo sensor esclavo la comunicación con el autopiloto es a través del bus I²C (*Inter-Integrated Circuit*) [3], donde el autopiloto hace las veces de maestro y el sensor, de esclavo. El autopiloto cuenta, a su vez, con otros sensores conectados a dicho bus. Esto hace que sea fácilmente integrable, no solo a este sistema sino también a cualquier otro que cuente con el bus I²C. En el modo sensor independiente, el canal de salida es una UART. Esta se encuentra conectada a un módulo transeceptor inalámbrico XBee [4], que utiliza el estándar Zig-Bee. Mediante estos dispositivos se establece un enlace con la estación terrena, donde se puede visualizar el resultado de cada ensayo.

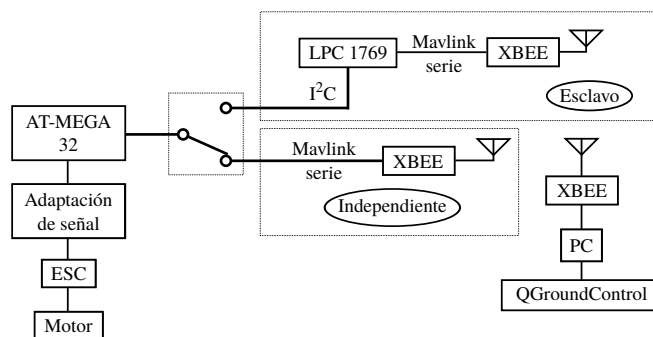


Fig. 5: Esquema de los módulos intervinientes y su coexistencia.

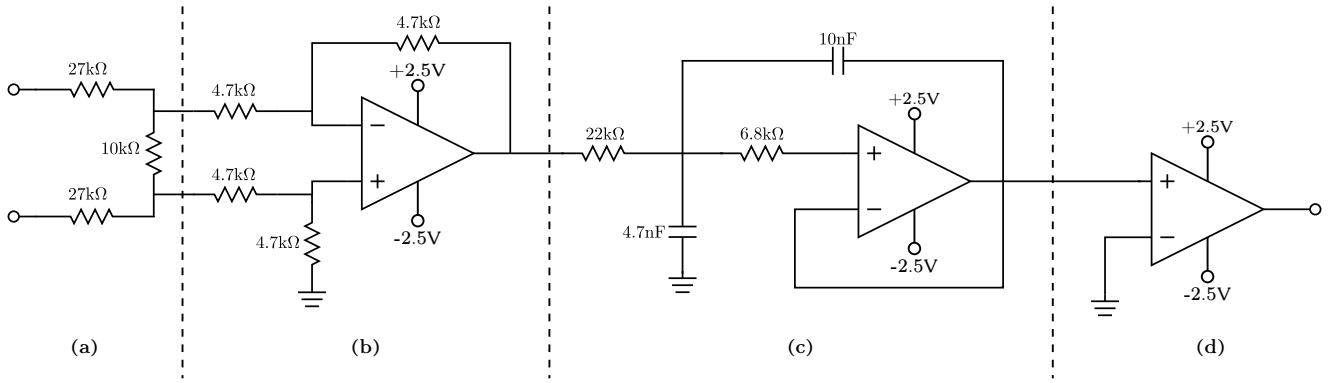


Fig. 6: Esquemático del circuito implementado.

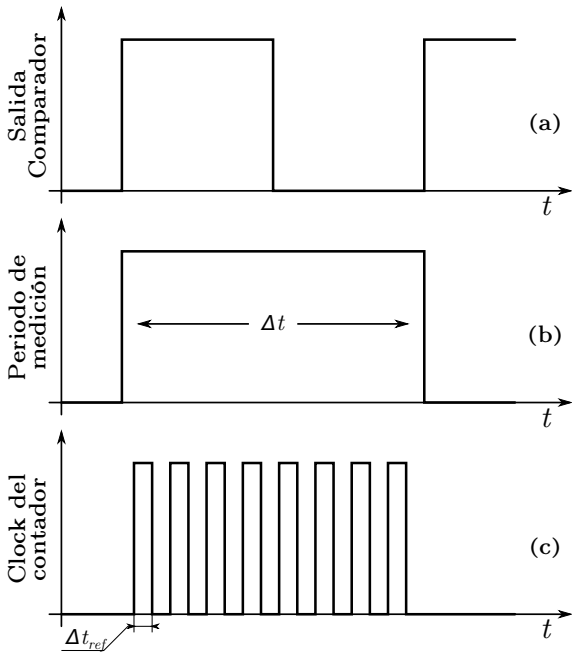


Fig. 7: Esquema de la teoría de medición utilizada.

III. FIRMWARE

En esta sección se muestra el firmware implementado para el desarrollo del sensor. Se encuentra dividido en dos capas. La primera corresponde a las librerías de cada uno de los periféricos del microcontrolador (UART, I²C e interrupción externa y captura). La segunda capa es la aplicación principal, encargada de determinar el modo de funcionamiento y administrar las interrupciones de los periféricos. La plataforma utilizada es el microcontrolador ATmega32 [5].

III-A. Selección de modo

Para determinar el modo de funcionamiento se utiliza una entrada digital del microcontrolador. De acuerdo a su estado, que únicamente es chequeado al inicio, el usuario puede elegir el modo de funcionamiento. Para que un cambio de modo se haga efectivo, se debe

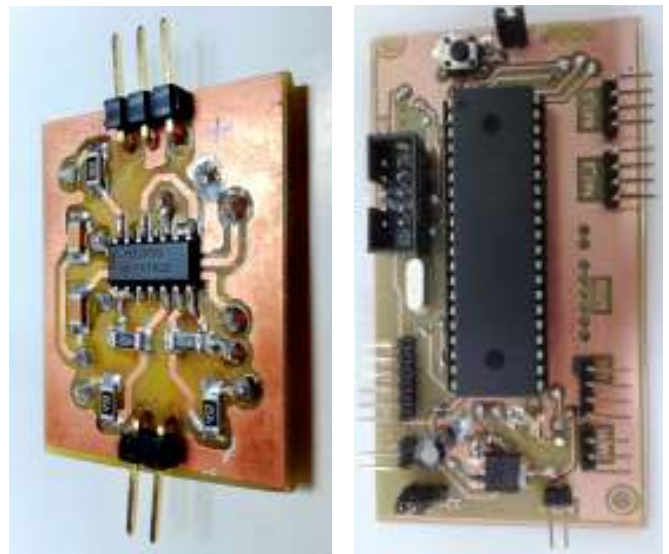
reiniciar el sistema. La razón de esto es evitar un cambio de modo indeseado. Un aspecto general de todo el conjunto operativo se puede observar en la Fig. 5, donde se ha simbolizado con un interruptor el selector de modo.

III-B. Teoría de medición

La teoría de medición es la misma que utilizan los contadores normalizados, aplicando la señal de entrada como ventana de conteo para el timer local. Una demostración gráfica puede observarse en la Fig. 7.

Para el sistema de conteo se configuró el timer de 16 bits que el microcontrolador posee [6], utilizándolo como referencia de tiempo con una resolución de 1μs.

La configuración de las interrupciones externas se hizo considerando los flancos de subida (Fig. 7a), de modo que la ventana considera un período completo de la señal a medir (Fig. 7b).



(a) Acondicionador de señales. (b) Microcontrolador y periféricos.

Fig. 8: Circuitos impresos utilizados en este trabajo.

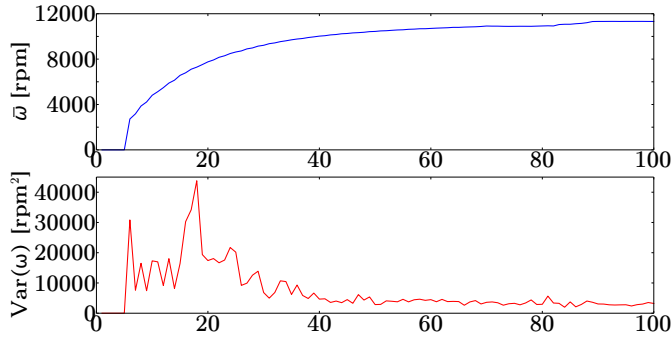


Fig. 9: Medición de la media y la varianza de la velocidad de rotación de un motor comercial para distintos porcentajes de *Throttle* de entrada.

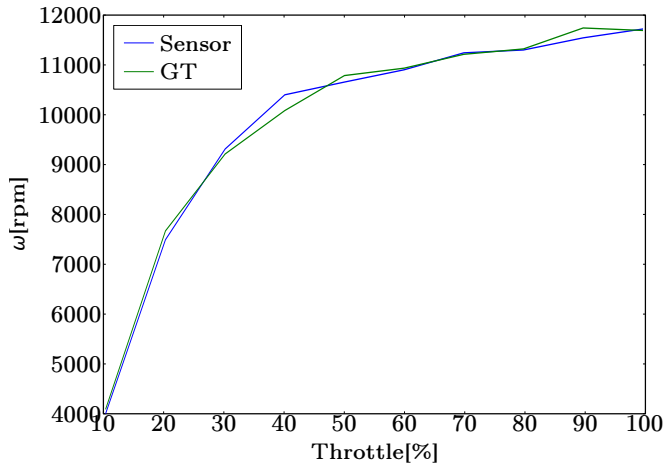


Fig. 10: Contrastación del sensor aquí presentado con un instrumento patrón.

Rescatando el valor de la cuenta del timer en cada flanco (t_1 y t_2) se procede con el cálculo del Δt equivalente al período de la señal utilizando los $N\Delta t_{ref}$ que se capturaron dentro de la ventana y cuyo valor es definido por la configuración del timer (Fig. 7c).

Así, de manera indirecta, se obtiene la frecuencia que es la variable necesaria para la determinación de la velocidad de rotación según (1).

El timer se encuentra configurado en modo de corrida libre, siendo necesario realizar los controles correspondientes para evitar una posible incongruencia entre t_1 y t_2 (Fig. 7a), debido a que durante un período de la señal de entrada podría producirse un desborde del contador. Además, esta configuración permite una mayor escalabilidad en la obtención de distintos Δt cuando sea necesario registrar una mayor cantidad de eventos independientes, tal como se prevé a futuro, midiendo cuatro motores en simultáneo. Debido a que el microcontrolador utilizado solo provee un módulo *capture*, para obtener información de los cuatro motores se utilizan tres interrupciones externas además de este módulo. Al entrar en cada una de las interrupciones externas, se compara el valor actual del contador con el de la última vez que el servicio de interrupción fue solicitado.

Además se fija un umbral de frecuencia mínima para el establecimiento del cero. Según las mediciones realizadas utilizando osciloscopio, se definió un período máximo de 65ms, es decir, un Δt equivalente a un desborde de timer.

III-C. Comunicación

El módulo I²C se inicializa únicamente para el funcionamiento en modo esclavo (Fig. 5). En este protocolo, el maestro realiza peticiones de lectura a los diferentes sensores conectados al mismo bus. Cuando el sensor de velocidad recibe una petición, envía la última medición obtenida. Este dato está almacenado en una variable de 16 bits (tamaño del timer), por lo que durante la transmisión es necesario dividirlo en dos palabras de 8 bits, debido a que el protocolo I²C soporta hasta 8 bits de información. Luego el maestro se encarga de la composición y procesamiento de los datos.

Del mismo modo, la UART se inicializa únicamente cuando el sensor funciona en modo autónomo, configurándose con una trama del tipo 8N1 y una velocidad de transferencia de 115200bps. La información es codificada y enviada utilizando el protocolo MAVLink (*Micro Air Vehicle Link*). Este es un protocolo libre [7] utilizado para la comunicación de vehículos aéreos no tripulados con su estación terrena, que en este caso es una PC. En la PC esta información es visualizada utilizando el entorno QGroundControl. Dicho protocolo MAVLink provee una gran cantidad de mensajes predeterminados. Sin embargo, ninguno de ellos prevé el envío de la velocidad de los motores de un UAV. Dado que es un proyecto de software libre y altamente expansible, se creó un mensaje personalizado para el envío de los datos de interés y se incluyó el empaquetador dentro del microcontrolador y el desempaquetador dentro del entorno QGroundControl.

IV. IMPLEMENTACIÓN Y RESULTADOS

La implementación del circuito acondicionador de señales de la Fig. 6 se observa en la Fig. 8a. El circuito integrado utilizado es el LM324 [8]. Este circuito integrado cuenta con cuatro amplificadores operacionales, tres de los cuales utilizan en el filtro activo y el restante, para generar el punto de referencia virtual. En la Fig. 8b se muestra el circuito impreso que contiene el microcontrolador y lo pertinente para su correcto funcionamiento y fácil conexión.

Una vez ensamblado, se caracterizó el sensor mediante una contrastación de sus mediciones con un patrón (*ground truth*). Este último se construyó utilizando un optoacoplador de reflexión (CYN70) que se montó convenientemente apuntando al rotor, sobre el cual se colocó una marca absorbente de luz para poder medir la frecuencia de rotación utilizando un osciloscopio. La Fig. 10 muestra que las mediciones efectuadas con ambos sensores coinciden en todo el rango. Otra característica importante es la media y la varianza de las mediciones. Para obtenerlas se tomaron muestras por un determinado tiempo a diferentes valores porcentuales de aceleración (usualmente referenciada como *throttle*). La respuesta que se obtuvo se observa en la Fig. 9. Además, se conformaron histogramas para las mediciones a 20% y 60% de throttle verificando el grado de similitud que tiene el

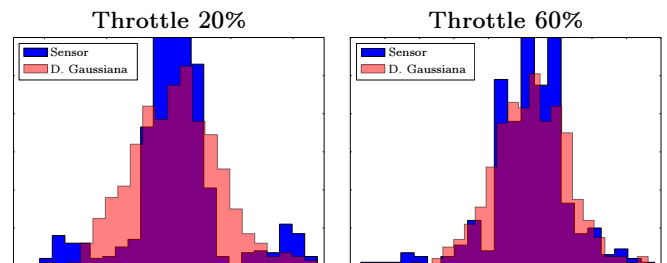


Fig. 11: Histogramas de las muestras obtenidas en el sensor comparadas con una distribución Gaussiana cuya media y desviación estándar son iguales a las de las muestras.

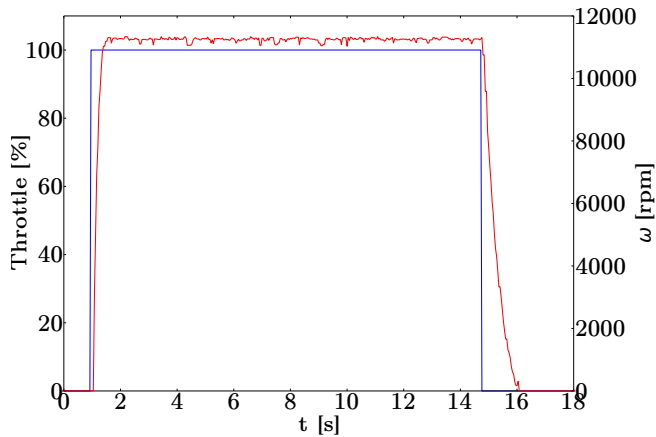


Fig. 12: Respuesta del conjunto ESC-motor ante una estrada escalón.

ruido presente en las mediciones con una distribución gaussiana de la misma media y varianza. De la Fig. 11 se deduce que las mediciones para valores bajos de throttle no se corresponden con una distribución gaussiana; mientras que para valores altos, sí.

Una medición que se realizó fue caracterizar la respuesta al escalón del conjunto ESC-motor, llevando el throttle desde 0% a 100% y luego, nuevamente a 0%. Los resultados de esta prueba se observan en la Fig. 12, donde se aprecia que tiene una constante de tiempo mecánica pequeña, dado que durante el ensayo el motor no tenía acoplada ninguna carga más que su propia inercia.

V. CONCLUSIONES

Se alcanzó el objetivo de desarrollar un sensor capaz de medir velocidades de motores brushless, utilizando únicamente las señales presentes en dos de sus fases. La implementación de la comunicación serial vía UART permite utilizar este sistema como un instrumento virtual, que puede ser utilizado en forma cableada o haciendo uso de un enlace inalámbrico. La ventaja de este último radica en que normalmente se llevan a cabo ensayos utilizando hélices a altas rpm, lo que podría ocasionar lesiones físicas al usuario.

La aplicación de este sensor en modo esclavo utilizando el bus I²C le da la posibilidad al sistema de coexistir junto a otros sensores de

distintas características haciendo uso del mismo bus. Esto se probó con éxito, utilizando el sistema junto a los sensores previamente instalados en la plataforma.

Se ensayó el sensor para distintos porcentaje de throttle de entrada y se determinó que para valores medios y altos (mayores al 30%) de este, las mediciones del sensor presentan una distribución aproximadamente gaussiana. Mientras que para valores bajos, esta distribución no es gaussiana. Debido a que el valor mínimo de throttle para mantener al cuadrotor en vuelo es de aproximadamente 40%, se puede considerar que las mediciones obtenidas con este sensor son gaussianas en todo el rango. Esto es de particular interés debido a que se lo puede utilizar en un estimador de posición y/o orientación que exige que el ruido presente en los procesos sea gaussiano (como por ejemplo un filtro de Kalman).

A futuro se planea utilizar este sensor en el modo independiente en un banco de pruebas para caracterizar motores brushless en distintas condiciones de funcionamiento. Como por ejemplo medición de empuje para distintas velocidades y hélices, parametrización de controladores electrónicos, medición del deslizamiento mecánico a distintas velocidades, modelado dinámico del conjunto motor-hélice para determinar frecuencias de resonancia y vibraciones a distintas velocidades utilizándose junto a un acelerómetro.

En relación al modo esclavo, se planea aumentar el número de acondicionadores presentes en el QA3 para obtener información simultánea del estado de todos los motores. Esto posibilita realizar un lazo de control de empuje, debido a que la relación empuje-velocidad de giro es constante y varía para distintos tipos de hélice.

REFERENCIAS

- [1] R.P. Sallen and E.L. Key. *A Practical Method of Designing RC Active Filters*. Technical report (Lincoln Laboratory). Massachusetts Inst. of Technology, 1954.
- [2] Bruce Carter. *A Single-Supply Op-Amp Circuit Collection*. Technical report, Texas Instruments, 2000.
- [3] NXP. *UM10204 - I2C-bus specification and user manual*, 2004.
- [4] Digi. *XBee/XBee-PRO ZB RF Modules*, March 2012.
- [5] Atmel. *ATmega32(L) Datasheet*, 2011.
- [6] Atmel. *AVR130: Setup and Use the AVR Timers*, 2002.
- [7] Repositorio proyecto MAVlink. <https://github.com/mavlink/mavlink>.
- [8] Texas Instruments. *LMx24-N, LM2902-N Low-Power, Quad-Operational Amplifiers*, 2000.

Implementación de un Sistema de Navegación a Robots Móviles

*Luis Martín, Cornejo Sernaqué
Universidad de Piura
Piura, Perú

*John Paul George, Ancajima Rodríguez
**Ivonne Margareth, Chunga Ramírez
Universidad de Piura
Piura, Perú

El presente proyecto consiste en la implementación de un sistema de localización y navegación a una unidad robótica móvil en un ambiente dinámico, basándose en una programación para la navegación autónoma en ambientes estáticos; con la finalidad de expandir sus aplicaciones industriales, enfocándonos en el transporte de materiales peligrosos y los automóviles autónomos, procurando mantener los procesos de manera eficaz y confiable.

Se utilizaron diferentes tipos de sistemas embebidos, ya sea para el control de los motores o para la lectura e interpretación de los datos extraídos por los sensores. Esta data es procesada como una distribución probabilística con la finalidad de reducir el problema de la incertidumbre en la localización de los robots móviles y obtener un mapeo óptimo. [11]

Se empleó un módulo robótico móvil llamado CoroBot fabricado en el año 2008 por CoroWare. El sistema posee sensores de proximidad IR (Infrarrojos) (SHARP GP2D12 [1], 1101 IR Distance Adapter [2]), voltaje (1117 Voltage Sensor [3]) y fuerza (3104 Interlink Electronic 0.5 [4]), todos estos dispositivos son de la familia Phidgets y son conectados a una tarjeta 1018 Phidgets Interface [5]. Además, dos encoders (1057 PhidgetEncoder High-Speed [6]) en las ruedas delanteras.

En ejecución de los comandos se utiliza una tarjeta SSC-32 ésta envía las señales PWM (Pulse width Modulation) para el control de los motores [7] DC Gearmotor HG37. Todo el proceso de control se realiza en la motherboard del CoroBot Mini ITX 7F2WE1GD5D [10].



Figura 1 Módulo Robótico CoroBot

El cálculo de la posición del móvil está basado en un modelo probabilístico, el cual se corrige constantemente mediante la toma de datos de los sensores infrarrojo; este procedimiento busca converger a una solución aproximada mediante una disminución en la desviación de la distribución.

Según los datos recopilados, el mapa se irá actualizando, comparando entre zonas ocupadas, libres o inciertas [8]. Para la explicitación de la incertidumbre de la posición se ha empleado el filtro extendido de Kalman (EFK) [9].

Se incorporó un ordenador de placa reducida (Banana Pro) el cual incluye un System-on-a-Chip Allwinner A20, Procesador ARM® Cortex™-A7 Dual-Core a 1GHz, 1GB de RAM DDR3, módulo WiFi integrado, entre otras características técnicas [14]; el cual se encargará del algoritmo de control el cual será programado en lenguaje Python mediante el SO de Linux (Lubuntu), reemplazando a la motherboard.

Se Procedió a controlar los cuatro motores DC acoplados a cada una de las ruedas, mediante un PID digital, programado en python. También se elaboró un algoritmo de planificación de trayectoria, el cual busca la ruta más corta con desplazamientos rectos, horizontales y verticales en el plano.

Finalmente se utilizará un sensor URG-04LX-UG01 de la marca Hokuyo Automatic CO LTD, el cual permitirá una mayor precisión y distancia de medición, gracias a su rango de lectura de 20 milímetros a 4000 milímetros con un barrido de 240° [13].

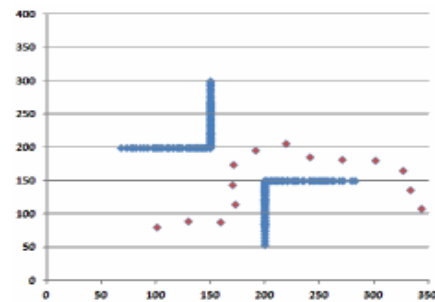


Figura 2 Resultados del algoritmo durante la Simulación. Los puntos azules son los obstáculos y los rojos la trayectoria planteada por el móvil

REFERENCES:

- [1]. Sharp. "GP2D12 Optical Device. Product manual". 2005
- [2]. Phidget s. "1101 – IR Distance Adapter. Product manual". 2011 [3].
- [3]. Phidget s. "1117 – Voltage Sensor. Product manual". 2009
- [4]. Interlink Electronics. "Force sensing resist or integration guide and evaluation parts catalog".
- [5]. Phidget s. "1018 – PhidgetInterfaceKit 8/8/8. Product manual". 2008
- [6]. Phidget s. "1057– Phidget Encoder High-speed. Product manual". 2011
- [7]. Lynxmotion. "SSC-32 Ver 2.0.user's manual". 2005
- [8]. Thrun, S.; Burgard, W.; Fox, D. "Probabilistic Robotics". London, England. 2006.
- [9]. Siegwart, R.; Nourbakhsh, I. "Introduction to autonomous mobile robots". London, England. 2004.
- [10]. J. Hinojosa. "Puesta en operación del robot CoroBot". Tesis de la Universidad de Piura, Piura, Perú. 2008
- [11]. López, R. "Modelo probabilístico para sensor de ultrasonido de bajo costo aplicado a la localización de robots móviles en ambientes dinámicos". Universidad de Piura, Perú. 2011
- [12]. Vilcahuamán, G. "Diseño E Implementación De Un Sistema De Navegación Autónomo Basado En El Filtro Extendido De Kalman En El Módulo Robótico CoroBot Empleando Lenguaje C". Universidad de Piura, Piura, Perú. 2013
- [13]. Hokuyo "Scanning Laser Range Finder URG-04LX-UG01 (Simple-URG) Specifications". 2008
- [14]. Lemaker "Specifications from board manufacturer". Banana Pro.

Foro Tecnológico

Pósters

**Implementación
de
Sistemas Embebidos**

Sistema de alarma de bajo costo controlado remotamente por dispositivos móviles

Melisa Kuzman, Walter Gemin, Juana Fernández, Raúl Rivera, Miguel Revuelta
Facultad de Ingeniería – Departamento de Electrónica
Universidad Nacional de Mar del Plata
Mar del Plata, Argentina
Email: melisakuzman@fi.mdp.edu.ar

Resumen— En este trabajo se presenta un sistema de alarma basado en microcontrolador, de muy bajo costo y fácil instalación, que no requiere display ni teclado. Su simpleza reduce el número de componentes para su construcción, sin limitaciones en sus prestaciones, respecto a las alarmas más modernas. Estas ventajas se basan en el uso de interfaz bluetooth que la hacen compatible con dispositivos móviles, donde una aplicación Android permite controlar su funcionamiento mediante una pantalla de interfaz gráfica de alta calidad.

Palabras claves—bluetooth; alarma; android; remoto

I. INTRODUCCIÓN

Los sistemas de alarma y monitoreo consisten principalmente de una unidad central, compuesta de un procesador que monitorea constantemente el estado de los sensores distribuidos en las zonas protegidas, un teclado que permite una mínima configuración y el ingreso de la clave para activar o desactivar la alarma, y un display alfanumérico que indica el estado de la misma.

Las tecnologías móviles que comprenden a los teléfonos inteligentes y tablets, se consideran el sector de mayor auge dentro del ámbito tecnológico actual, es por ello que un gran número de novedades tecnológicas están relacionadas con estos dispositivos móviles. En este trabajo se los utiliza para reemplazar a los componentes: display, teclado y control remoto, mediante una aplicación Android, reduciendo considerablemente el costo de su implementación.

De esta forma, la pantalla del dispositivo móvil es una representación virtual del display, el teclado de la alarma y el control remoto provisto por la conexión Bluetooth, permitiendo reemplazar componentes hardware por software. Es decir entonces que, mediante líneas de código (programación) se puede definir la funcionalidad y apariencia de la interfaz de usuario de la alarma. Esto representa una sensible mejora en las prestaciones del sistema, dado que está basado en un dispositivo de gran capacidad gráfica, donde se puede presentar mayor cantidad de información, más clara e intuitiva.

En este trabajo se describe el hardware compuesto por la central basada en el microcontrolador PIC16F88[1], la interfaz de RF para el control remoto estándar y la interfaz Bluetooth para la comunicación con el dispositivo móvil. Posteriormente, se describe el software del microcontrolador, con especial énfasis en las funciones de conexión RF y Bluetooth[2], y el software de la aplicación Android[3].

II. DIAGRAMA EN BLOQUES

En la Figura 1 se observa el diagrama en bloques completo del sistema, compuesto por las etapas unidad central (microcontrolador), sensores[4], sirena[5] y los módulos bluetooth y RF que se describen a continuación.

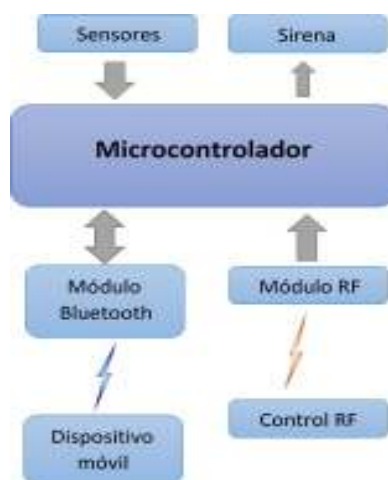


Figura 1: Diagrama en Bloques

Los sensores son dispositivos diseñados para detectar determinadas situaciones, que por lo general desencadenan una serie de acciones para notificar el hecho o actuar sobre el sistema. Activar una sirena es uno de los métodos más simples, rápidos, económicos y eficaces para generar alerta en la zona que rodea al domicilio. La unidad central se encarga

de procesar la información proveniente de los sensores, y con ello determinar qué hacer. La lógica, en este trabajo, es controlada a través de un microcontrolador al cual ha sido cargado con el firmware pertinente, que se detalla en la sección correspondiente.

Por otra parte, el usuario tiene dos interfaces para acceder al sistema. El primer caso es un control remoto RF, en donde cada botón corresponde a una acción determinada. El segundo método corresponde a comunicación bluetooth, donde el usuario puede controlar el sistema a través de una aplicación de Android. La Figura 2, muestra una imagen con los componentes del sistema antes mencionados.



Figura 2: Sistema de Alarma

III. TECNOLOGÍAS BLUETOOTH Y RF

A. Comunicación Bluetooth

Bluetooth es un protocolo de comunicación diseñado para dispositivos de bajo consumo, que requieren corto alcance de emisión y están basados en transceptores de bajo costo.

Los módulos bluetooth están diseñados para establecer una comunicación bidireccional, que permite conectar diferentes dispositivos para el intercambio de información. Su uso ha crecido significativamente en los últimos años, principalmente porque facilita las comunicaciones, elimina conexiones físicas (cables y conectores) y pueden crear pequeñas redes inalámbricas, simplificando la sincronización de datos entre equipos.

En el presente trabajo se utiliza un módulo denominado comercialmente HC-05[6], que puede ser observado en la Figura 3. Sus parámetros son configurados a

través de una interfaz serie RS232, con el uso de comandos AT[7].



Figura 3: Módulo Bluetooth

Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre una PC y un Terminal MODEM, como un protocolo de comunicación para configurarlo o proporcionarle instrucciones, tales como marcar un número de teléfono. Muchos otros dispositivos periféricos adoptaron esta serie de comandos de comunicación. Dentro del computador la comunicación de la CPU con cada interfaz también adopta la modalidad de comandos que asegura compatibilidad entre la interfaz y el Sistema Operativo, independientemente del fabricante de la misma. Los comandos consisten en una secuencia de caracteres ASCII que sintetizan el significado de la operación que se quiere ejecutar. De esta forma asignando un comando a cada operación del repertorio que soporta la interfaz, se puede armar un lenguaje de comunicación que el microcontrolador interpreta y ejecuta devolviendo la información requerida o actuando sobre el entorno.

Las características que son programadas para el desarrollo de la alarma son: el baudrate, el formato de la trama de datos, el nombre de identificación del módulo, la contraseña y su rol.

B. Control remoto RF

Muchos sistemas que se utilizan diariamente están comandados por señales RF, como portones y alarmas de autos (Figura 4). Su practicidad y alcance la hacen una tecnología muy utilizada.

Es por ello que, uno de los medios de acceso a la alarma domiciliaria es a través de un control remoto inalámbrico con una interfaz de recepción. El mismo utiliza una portadora de 433MHz, modulación ASK y 8 bits de codificación. Tiene un alcance de transmisión entre 120 y 150m.



Figura 4: Control remoto RF y módulo receptor

El control remoto RF emite 4 señales diferentes, cada una corresponde a un botón en particular. Estas cuatro señales identifican a solo una de las salidas que tiene el módulo receptor. En estado de reposo las líneas de salida del mismo se encuentran en alto (utilizando lógica positiva). Cuando alguno de los pulsadores es presionado, la salida correspondiente cambia de estado, manteniéndose en bajo mientras el botón esté accionado. Estas líneas de salida son conectadas como entradas digitales al microcontrolador, las cuales son previamente programadas como líneas de entrada de interrupción por flanco descendente. Así, cada vez que se detecte un cambio, existe una rutina de atención en el microcontrolador para realizar las acciones necesarias (En este caso activar o desactivar la alarma).

C. Unidad Central

La unidad central de la alarma está basada en el microcontrolador PIC16F88 de la empresa MICROCHIP. Su arquitectura es de 8 bits, un encapsulado de 18 pines, que, dentro de las prestaciones más relevantes se pueden destacar:

- 1 UART
- 256 Bytes EEPROM
- 7 KBytes FLASH
- 368 Bytes RAM
- Oscilador interno de 8MHz
- 5 Entradas para interrupciones
- 2 Timers de 8 bits
- 1 Timer de 16 bits

IV. DESCRIPCIÓN DEL SOFTWARE

Para responder a eventos, los microcontroladores cuentan con un recurso conocido como interrupciones. Las interrupciones son señales que se generan externa o internamente al microcontrolador que detienen la ejecución normal del programa para atender alguna subrutina de respuesta al evento. Una vez ejecutada, continúa el programa en el punto en que se encontraba antes de generarse la interrupción. En este trabajo se utilizan una serie de interrupciones que permiten configurar el comportamiento del microcontrolador por eventos, dado que mantiene ejecutando

el lazo principal del programa y atiende al puerto de entrada/salida serie cuando este lo requiere. Las interrupciones programadas en este sistema son las de cambio de estado en entradas, el arribo de un carácter al puerto serie o el cumplimiento de un periodo de tiempo fijado por un timer o temporizador.

A. Comandos Bluetooth

El módulo bluetooth se conecta a las entradas correspondientes a la UART del microcontrolador. Así, a través de la interrupción serie, se captan las tramas provenientes del dispositivo móvil para que luego sean analizadas por el microcontrolador. En la Figura 5 se observa el diagrama de flujo de atención a la interrupción serie. La decodificación del mensaje se realiza a través de un intérprete de comandos.

Cada comando está compuesto por una secuencia de seis caracteres en mayúscula que lo identifican, que representan los parámetros de la acción completa que deberá ejecutarse. Finaliza siempre en un carácter especial ($\backslash n$) de nueva línea, que sirve para indicar el fin del comando. Por ejemplo, al recibir la secuencia ACTIVA $\backslash n$, el microcontrolador decodifica el mensaje, y ejecuta las acciones necesarias para activar la alarma. En el caso que el dato no sea válido, se notifica el error a través de la interface serie y se descarta.

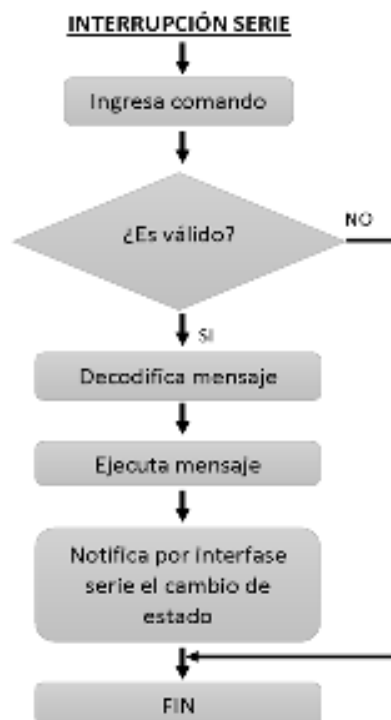


Figura 5: Atención de la interrupción serie

B. Comandos RF

El sistema de alarma tiene, además de la comunicación bluetooth con una aplicación android, una conexión a través de un dispositivo de radiofrecuencia.

Para utilizar el control remoto RF, el módulo de recepción está conectado a la entrada del microcontrolador, en un pin de entrada digital, configurado como interrupción externa.

En la Figura 6 se observa cómo el microcontrolador atiende la interrupción externa y ejecuta la serie de acciones necesarias para completar el comando. Además, se encarga de enviar una trama a través de la interfaz serie para notificar dicho cambio al dispositivo móvil, en caso que hubiera una conexión establecida.

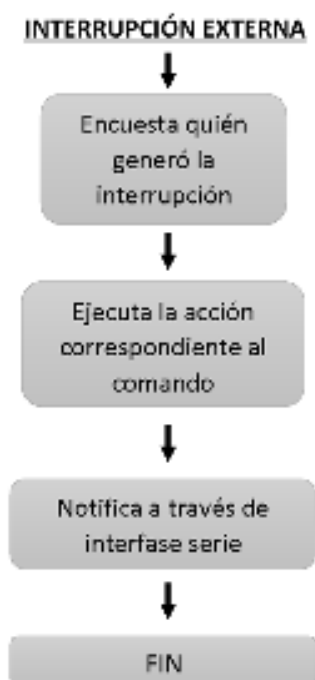


Figura 6: Atención de interrupción externa

C. Aplicación Android

Android comenzó siendo un SO para telefonía móvil, difundiendo luego a las tablets. La principal ventaja de su uso es que disponemos de una gran cantidad de dispositivos y aplicaciones compatibles.

La principal diferencia entre Android y el resto de sistemas operativos para dispositivos móviles es su condición de software libre, basado en Linux y de código abierto [8] [9].

La aplicación para comunicarse con la alarma domiciliaria está desarrollada en el entorno “**Basic4android**”, similar a Visual Basic. Algunas de las características de Basic4android son:

- Una herramienta RAD (Rapid Application Development) para apps nativas de Android.
- Un entorno (IDE) y un lenguaje de programación enfocado en el desarrollo de Android.
- Lenguaje de programación orientado a objetos.
- Depurador rápido.
- Editor visual WYSIWYG para Android. Se soportan pantallas múltiples y resoluciones.
- Soporta todos los dispositivos Android desde la versión 1.6 hasta la versión 4.x.

Basic4android posee diversas librerías que facilitan la interacción con los datos. En este proyecto en particular, se utilizan:

- **RandomAccessFile:** Posee diversos objetos para trabajar con cadenas de datos, tales como lectura y escritura de tramas, con y sin encriptación, monitorear en tiempo real dicho proceso, etc..
- **Serial:** Es una librería que facilita el uso del adaptador bluetooth del dispositivo móvil, y posee funciones propias para una comunicación serie.
- **ByteConverter:** Posibilita la conversión de las tramas en diferentes representaciones numéricas.

A modo de ejemplo se detalla un caso particular. Dentro de la librería Serial se encuentra un objeto denominado BluetoothAdmin. Usando el mismo se puede habilitar o deshabilitar el adaptador de bluetooth, monitorear su estado y buscar dispositivos para generar la conexión. Los eventos que le pertenecen a este objeto, y que pueden ser aplicados son:

- StateChanged (NewState As Int, OldState As Int)
- DiscoveryStarted
- DiscoveryFinished
- DeviceFound (Name As String, MacAddress As String)

Así, por ejemplo, “DeviceFound” es un evento que se ejecuta cuando un nuevo dispositivo bluetooth es encontrado, brindando los datos del nombre y MAC del mismo. Este evento siempre es posterior a “DiscoveryStarted”, donde comienza el proceso de búsqueda de móviles.

En la Figura 7 se pueden observar tres diagramas de flujos de diferentes eventos, todos correspondientes al desarrollo de la aplicación en Android. En la parte superior se encuentra el “Evento Conexión Serie” que representan qué acciones se llevan a cabo en función del éxito de la conexión entre el dispositivo móvil y la central de la alarma. El “Evento Click” es invocado cuando algún botón de la aplicación es presionado. Siempre y cuando el enlace se encuentre establecido, se envía la trama de control pertinente (un comando para apagar si la alarma se encuentra activa, o para encenderla en el caso contrario). Por último, en la figura inferior se representa el “Evento Recepción Serie” de atención

al puerto serie, que interpreta la trama recibida y en función a su contenido notifica qué acción fue llevada a cabo en la alarma. Este evento corresponde al ingreso de nuevas tramas y permite verificar si el sistema ha cambiado de estado, ya que es el microcontrolador el que envía un mensaje informando la acción ejecutada. Por otro lado, posibilita monitorear en tiempo real diferentes acontecimientos, como el cambio del estado de operación del sistema a través del control RF, o si alguno de los sensores ha sido accionado, notificando dicho suceso en la pantalla.

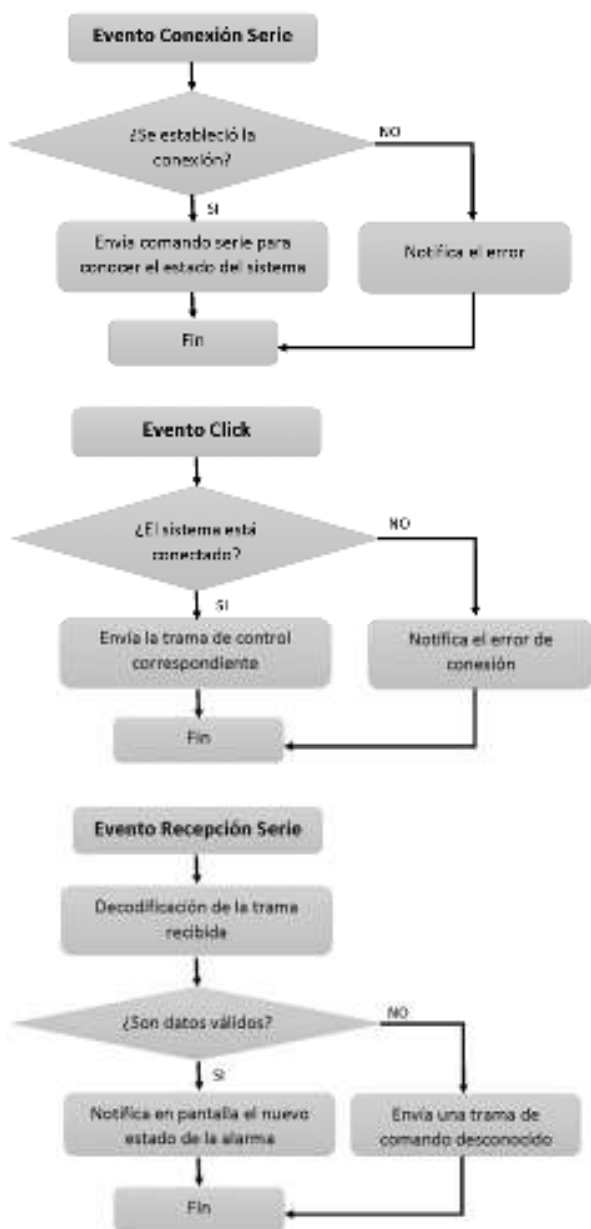


Figura 7: Aplicación Android

En la Figura 8 se pueden observar dos capturas de pantallas de la aplicación, diseñada para dispositivos con sistema operativo Android. En ambos casos son visibles 3 objetos bien diferenciados: una leyenda de texto (Label), que indica con palabras el estado de la alarma; una imagen (ImageView) representativa del estado de la alarma; y un botón (Button) que se utiliza para activar o desactivar la alarma, según corresponda. Así, se refleja la idea de desarrollar un sistema que se adapte a las nuevas tecnologías, intuitivo y fácil de utilizar.



Figura 8: Capturas de la aplicación

V. CONCLUSIONES

Este trabajo demuestra como las nuevas tecnologías pueden reemplazar y ofrecer mejoras respecto de otras existentes. Esto lo llevan a cabo no solo cumpliendo las mismas funciones que los sistemas anteriores, sino también dotándolos de nuevas características.

En este caso en particular, se muestra un sistema de alarma que integra una tecnología de uso frecuente, y otra que está en pleno auge, un control RF y una aplicación Android, respectivamente. Así, se logra reemplazar el teclado alfanumérico de una alarma, por una aplicación con una interfaz gráfica de alta calidad, versátil e intuitiva, logrando reducir de manera significativa los costos de la alarma.

En las pruebas realizadas por varios usuarios sobre distintos modelos de dispositivos móviles, se verificó que este sistema es accesible, intuitivo y confiable.

Referencias

- [1] Microchip, "PIC16F87/88 Data Sheet", Microchip Technology Inc., 2005.
- [2] Bluetooth, <https://www.bluetooth.org/en-us>.

- [3] Android, <https://www.android.com/>.
- [4] <http://www.alonsohnos.com/content/download/130/1756/version/9/file/Install-sp-ir-700-1-13-web.pdf>
- [5] <http://www.alonsohnos.com/content/download/151/1818/file/install-smp-120-12-12-web.pdf>
- [6] http://www.tec.reutlingen-university.de/uploads/media/DatenblattHC-05_BT-Modul.pdf
- [7] http://eskimon.fr/wpcontent/uploads/2014/10/commandes_ATHC05.pdf
- [8] Kevin Purdy, "The Complete Android Guide", Lulu.com, 2010.
- [9] Nicolas Gramlich, "Handbook - Android Programming", anddev.org, 2008.

Sistema de Adquisición de Datos Sísmicos

Ing. Ricardo Gabriel Sifón
Departamento Laboratorio Sismológico
Instituto Nacional de Prevención Sísmica
San Juan, Argentina
rsifon@inpres.gov.ar

Mg. Héctor Riso
Especialidad en Sistemas Embebidos
Instituto Universitario Aeronáutico
Córdoba, Argentina
hriso@iua.edu.ar

Resumen— El presente trabajo tiene como objeto adquirir señales sísmicas asociadas a la sismicidad local.

Para cumplir con tal fin se desarrolló un Sistema de Adquisición de Datos Sísmicos sobre una placa computadora reducida (Single Board Computer, SBC) Raspberry Pi, integrándose el hardware necesario para su conexión con el instrumento sensor (sismómetro), encargado de percibir los movimientos sísmicos. Permite la comunicación directa con un software de procesamiento y colección de datos, haciendo uso del difundido protocolo estándar en sismología SEEDLINK.

Palabras clave: Raspberry Pi, sismología, SEEDLINK, EARTHWORM, procesamiento, adquisición, protocolo, Sistema.

I. INTRODUCCIÓN.

En la mayoría de los países, diversos organismos se ocupan de monitorear la actividad sísmica. Sobre el territorio de la República Argentina, el Instituto Nacional de Prevención Sísmica (INPRES) es el encargado de llevar a cabo esta tarea. En particular la *sismicidad local* es un área de estudio e interés que está en curso en esta Institución.

Los sismos locales son aquellos eventos que ocurren dentro de un radio de hasta 100 km desde donde se ubica el instrumento sensor (sismómetro) y tienen como característica un valor de frecuencia que oscila entre 0 y 100 Hz [10]. A pesar de que algunos de estos sismos son de poca magnitud en la escala Richter [1], y no son percibidos en la vida cotidiana de las personas, su estudio es de gran importancia puesto que la mayoría de los sismos de mayor intensidad siempre están precedidos de sismos de menor intensidad e imperceptibles para las personas. Además su registro y análisis sirve para un mejor diseño futuro de estructuras, prospección y análisis de suelo, y elaboración de sistemas de alerta.

El INPRES cuenta con una Red Nacional de Estaciones Sismológicas (RNES) distribuida en todo el país. Básicamente cada una de las Estaciones Sismológicas, cuentan con un sismómetro que registra los eventos y con un Sistema de Adquisición de Datos (Data Acquisition System, DAS).

Cada Estación Sismológica transmite los datos registrados hacia el INPRES en tiempo real, colectados mediante un software de procesamiento y colección de datos de código abierto, denominado EARTHWORM [2].

El establecimiento de la comunicación entre EARTHWORM y los distintos DAS con los que cuenta el INPRES debe realizarse a través de un software propietario (provisto por cada empresa fabricante de los DAS) que hace las veces de intermediario entre EARTHWORM y cada uno de los DAS.

Por lo expresado, se hace necesario disponer de un Servidor dedicado, para alojar y ejecutar el software propietario de cada DAS.

Por sus características propias los organismos sismológicos (como INPRES) están bajo modificación, ampliación y actualización constante de sus redes y estaciones. Al depender de software propietario para obtener los datos, se presentan dificultades ante una modificación que se pretenda efectuar al software EARTHWORM. Aumentando la complejidad, mantenimiento y operación de la RNES. El presente trabajo busca conseguir un Sistema de adquisición capaz de adquirir las señales sísmicas como también flexibilidad, conexas e independencia de terceros.

II. REQUERIMIENTOS.

A continuación se describen los requerimientos necesarios para este trabajo:

1. Desarrollar un Sistema que permita la comunicación directa con EARTHWORM sin necesidad de utilizar un software intermediario, ni el desarrollo de un módulo adicional para EARTHWORM.
2. Los datos registrados deben ser transmitidos en tiempo real.
3. Los sismómetros a utilizar en este trabajo deben ser aptos para sismicidad local. El INPRES dispone para tal fin sismómetros de período corto, de la firma Teledyne Geotech, modelo S-13[8].
4. El Sistema debe ser capaz de registrar en forma óptima sismos locales que se encuentren dentro del rango de magnitud 2 a 4 en la escala Richter.
5. La señal de tensión analógica entregada por el sismómetro debe ser previamente acondicionada para así posibilitar el registro de sismos locales.
6. Las estaciones sismológicas se ubican en lugares geológicamente estratégicos, que por lo general son

zonas alejadas de las grandes ciudades, lo que dificulta la operación del DAS en forma local por parte de un operario en cualquier instante. Por lo cual resulta imprescindible poder configurar los parámetros esenciales del DAS y operar el mismo en forma remota.

III. PROPUESTA DE DESARROLLO.

En base a los requerimientos planteados en la sección anterior, se propone realizar un Sistema de Adquisición de Datos que permita enviar directamente los datos capturados en tiempo real al Sistema Colector de Datos EARTHWORM (Figura 1).

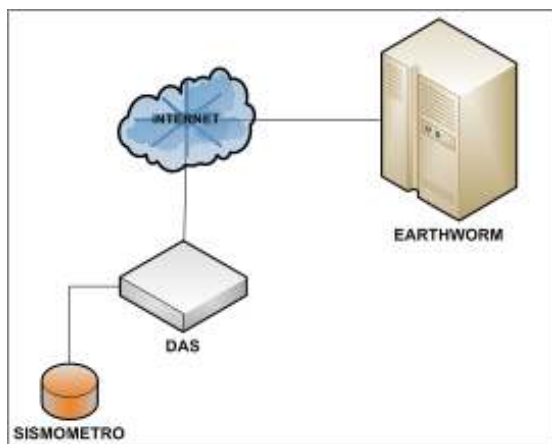


Figura 1. Propuesta de Desarrollo.

El dispositivo tendrá instalado un Web Server embebido, que posibilitará al usuario realizar la configuración y puesta en marcha del mismo.

La realización del Sistema propuesto implica el desarrollo de dos partes claramente diferenciables, una concerniente al Hardware y otra al Software.

En la siguiente sección se detallará cada una de ellas.

IV. DESARROLLO DEL SISTEMA.

A. DESARROLLO DEL HARDWARE.

A1. Plataforma de Desarrollo.

Entre los requerimientos citados en la sección II, se encuentran dos que son los que llevan a optar por la plataforma de desarrollo a utilizar. El primero de ellos, motivación principal para la elaboración de este trabajo, es que el Sistema pueda establecer una comunicación directa con EARTHWORM sin implementar un módulo de Software adicional. El segundo, transmitir los datos registrados por el sismómetro en tiempo real.

EARTHWORM cuenta con un módulo denominado SLINK2EW, que permite comunicarse en forma directa con él, a través del uso del protocolo SEEDLINK [3] (basado en el protocolo TCP). Este protocolo está escrito en Lenguaje ANSI C y es empleado de forma masiva por las más

importantes instituciones sismológicas del mundo para el intercambio de datos sísmicos.

De lo mencionado en el párrafo anterior, se desprende el escoger una plataforma cuya arquitectura provea conectividad sobre un stack TCP/IP y que posibilite la implementación del protocolo SEEDLINK.

Entre las distintas alternativas al momento de escoger una plataforma para la elaboración del Sistema propuesto, se opta por escoger la SBC Raspberry Pi Modelo B [4], la cual cuenta con conectividad sobre un stack TCP/IP, requisito para la implementación del protocolo SEEDLINK.

El protocolo de comunicación escogido debe ser implementado bajo Sistema Operativo Linux. La SBC Raspberry Pi ejecuta Raspbian [5], distribución de GNU/Linux basada en Debian.

Es importante destacar que sobre Raspbian, las librerías necesarias para implementar el protocolo SEEDLINK fueron utilizadas con éxito a través del software SEISCOMP [6].

A2. Acondicionamiento de la señal de entrada.

La señal que registra el sismómetro S-13 puede alcanzar la frecuencia de 100 Hz [8], mientras que las señales sísmicas de interés para este trabajo tienen un valor máximo de 10 Hz [21]. Por ello es necesario acondicionar las señales obtenidas por el sismómetro. Esto se logra amplificando y filtrando la señal. Luego la SBC Raspberry Pi debe procesar y transmitir los registros hacia EARTHWORM, por lo cual se deben digitalizar los datos obtenidos por el sensor.

A2.1. Amplificación.

La señal de salida del sismómetro S-13 es una tensión diferencial y sensible al ruido producido por fuentes externas, provenientes de la red eléctrica (50 Hz de frecuencia). El máximo valor de tensión que entrega el sismómetro ante un sismo de magnitud 4 en la escala de Richter es de 2 Vpp [21].

Teniendo en cuenta lo antes mencionado, fue tomada la decisión de implementar un amplificador de instrumentación [18]. Estos dispositivos tienen una ganancia diferencial precisa y estable, cuyo valor puede controlarse mediante una resistencia externa (Gain Resistor, RG). Presentan una Relación de Rechazo de Modo Común (Common Mode Rejection Ratio, CMRR) elevada, lo que posibilita tener una buena relación señal a ruido en todo el rango de frecuencia de interés. Además su alta impedancia de entrada, permite aislar del sismómetro el resto del circuito.

A partir de las características mencionadas en el párrafo anterior fue elegido para implementar el circuito integrado AD622ANZ de Analog Devices [9].

El valor de ganancia será escogido de acuerdo al máximo valor de tensión que tenga la entrada de la etapa de digitalización, la cual será desarrollada más adelante.

A2.2. Filtrado.

La magnitud de la señal sísmica a registrar por el sismómetro debe estar entre los valores 2 y 4 de magnitud en la escala Richter (ítem 4 de la sección Requerimientos). Debido a esto, el ancho de banda de interés se encuentra en el rango de 0 a 10 Hz [21].

Además, como fue mencionado anteriormente, la señal entregada por el sismómetro debe ser digitalizada para ser procesada por la SBC Raspberry Pi. Según el Teorema de Nyquist [19], para poder replicar con exactitud la forma de una onda es necesario que la frecuencia de muestreo (Sampling Frequency, FS) sea igual o superior al doble de la máxima frecuencia a muestrear. Si esto no se cumple aparece un fenómeno denominado Aliasing [19], que hace que la banda de frecuencia de la señal muestreada se repita (tenga un alias) a múltiplos de la frecuencia de muestreo. Es por ello que, si no es escogida la frecuencia de muestreo correcta, el alias puede quedar solapado con la banda de la señal base produciendo solapamiento u overlapping.

Por este motivo, debe ser implementado un filtro analógico Anti-Aliasing antes de proceder a digitalizar la señal. Para eliminar las frecuencias por encima de un cierto umbral, debe ser un filtro pasa bajos, que permita limitar el ancho de banda de la señal y evitar la aparición del efecto de solapamiento. Además, su respuesta debe ser máximamente plana, tanto en la banda de paso como en la banda de atenuación. La topología que presenta estas características es la de Butterworth, en configuración Sallen Key [20].

La frecuencia de corte para el diseño del filtro, a fines prácticos, fue elegida en 20 Hz (el doble del valor de interés), para que en el rango de 0 a 10 Hz el desplazamiento de fase introducido por el filtro analógico Anti-Aliasing [20] sea lineal (retraso de grupo constante) [19]. En una etapa posterior, luego de tener en forma digital la señal sísmica, se procederá a aplicar un filtro digital por Software, como será explicado más adelante.

La frecuencia de muestreo utilizada en sismología está en el rango comprendido entre 1 y 200 Hz [10]. Para el desarrollo del Sistema propuesto se utilizará un valor de 100 Hz, el cual es el escogido por el INPRES para todas sus estaciones [17].

El valor máximo de atenuación de la señal en la banda de atenuación es obtenido calculando la Relación Señal a Ruido (Signal Noise Relation, SNR) [23], que depende del número de bits utilizados para cuantificar [22] en el proceso de digitalizar una señal.

$$SNR = 6.02 * N + 1.76 \text{ dB} \quad (1)$$

En la etapa de digitalización será utilizado un Conversor Analógico a Digital (Analog to Digital Converter, ADC) [22]. La cantidad de bits del ADC a utilizar está directamente asociada a la magnitud del sismo que se quiere registrar. Un sismo de magnitud 2 registrado a 100 km de distancia del sismómetro presenta el número de cuentas de 200 (límite inferior si la señal necesita ser registrada con un valor razonable de SNR), y si el ADC utilizado es de 16 bits [10].

Este dispositivo presenta en su salida 32768 cuentas (teniendo en cuenta que su entrada acepta valores positivos y negativos), con un rango dinámico [22] igual a:

$$\text{Rango Dinámico} = \frac{32768}{200} \cong 164 \quad (2)$$

Suponiendo que la magnitud se incrementa con el logaritmo de la amplitud, la magnitud máxima a registrar con un ADC de 16 bits es de:

$$\text{Magnitud}_{MAX} = 2.0 + \log_{10} 164 \cong 4.2 \quad (3)$$

Con lo cual, un ADC de 16 bits, cumple con los requisitos solicitados para el registro de sismos locales de magnitud entre 2 y 4 en la escala de Richter. Teniendo definido el número de bits en el proceso de digitalización de la señal, el valor de SNR, utilizando la ecuación 1, es de 98dB.

El filtro obtenido fue de orden 8, conformado con 4 etapas tipo Butterworth. Su ganancia en la banda de paso es de 0 dB, ya que la ganancia del Sistema fue establecida en la etapa de amplificación. El amplificador operacional que fue escogido fue el LM324 de Texas Instruments [24].

A2.3. Digitalización de la señal.

La señal (analógica) registrada por el sismómetro es digitalizada a través de un ADC de 16 bits de resolución. La señal digitalizada será enviada a la SBC Raspberry Pi, la cual cuenta con un bus de Interfaz de Periféricos Serie (Serial Peripheral Interface, SPI) [7] para la conexión de dispositivos externos.

Teniendo en cuenta lo mencionado en el párrafo anterior y que el filtro Anti-Aliasing presenta una tensión de salida cuyo valor máximo es de 10 Vpp, el ADC seleccionado fue el ADS8507 de Texas Instruments [11]. Este dispositivo acepta 10 Vpp de tensión analógica en sus entradas, lo que permite establecer el valor de ganancia en la etapa de Amplificación. Al ser de 2 Vpp la máxima tensión a registrar por el sismómetro para un sismo de magnitud 4 en la escala de Richter, para aprovechar todo el rango del ADC, fue establecido un valor de ganancia igual a 10.

El detalle de cada una de las etapas antes mencionadas se ha incluido en la documentación del Sistema [35].

B. DESARROLLO DEL SOFTWARE.

B1. Transmisión de datos.

Una vez realizado el acondicionamiento de la señal sísmica registrada por el sismómetro, es necesario llevar a cabo la transmisión de la misma hacia EARTHWORM.

Tal como fue mencionado en la sección IV, se hará uso de la SBC Raspberry Pi, como plataforma de desarrollo, de Raspbian como Sistema Operativo, y del protocolo SEEDLINK, para establecer una comunicación directa con EARTHWORM.

Establecer una comunicación entre la SBC Raspberry Pi y EARTHWORM, haciendo uso del protocolo SEEDLINK,

debe llevarse a cabo a través de una arquitectura Cliente-Servidor. EARTHWORM cuenta un módulo que actúa como Cliente SEEDLINK, denominado SLINK2EW.

En la SBC Raspberry Pi se optó por instalar la aplicación RINGSERVER [16], que hace las veces de Servidor SEEDLINK, y fue desarrollada por el Incorporated Research Institutions For Seismology (IRIS) para posibilitar el intercambio de datos entre diversas instituciones sismológicas. RINGSERVER permite evitar la pérdida de datos ante una interrupción en la comunicación con EARTHWORM. Cuando esto sucede, la transmisión de datos inicia en el punto en el que se encontraba al momento de ocurrir el corte. Los datos que RINGSERVER envía a cada Cliente son registros Mini-SEED [14] de 512 bytes.

De acuerdo a lo mencionado anteriormente, fue desarrollada una aplicación que, a partir de los datos registrados por el sismómetro, acondicionados y posteriormente digitalizados por el ADC (ADS8507), crea registros Mini-SEED con el fin de ser interpretados por RINGSERVER.

Debido a que fue necesario utilizar librerías externas para establecer una comunicación con el ADC, crear y enviar registros Mini-SEED hacia RINGSERVER, y todas ellas están escritas en Lenguaje ANSI C, este lenguaje de programación fue utilizado para realizar la aplicación antes mencionada.

Cada una de las tareas que realiza la aplicación se detallan a continuación:

B1.1. Lectura de datos del ADC.

La lectura de los datos desde el ADC por parte de la SBC Raspberry Pi se realiza a través de un bus SPI, haciendo uso de la librería Wiring Pi [25], que posee Raspbian, cada 10 ms (correspondiente a una frecuencia de muestreo de 100 Hz).

La frecuencia con que se procede a la lectura del bus SPI para obtener los datos provenientes del ADC, es obtenida a partir de la señal SIGALRM [26] del que dispone el Lenguaje ANSI C para su uso en Sistemas Operativos Unix/Linux.

B1.2. Filtrado Digital.

Cada dato obtenido del ADC es filtrado digitalmente, como fue mencionado en la sección IV. La utilización de un filtro digital tiene como ventajas: inmunidad a ruido fuerte, mucha exactitud, fácil modificación de las características del filtro, entre otras [19].

Fue escogido un filtro del tipo Respuesta Finita Impulsiva (Finite Impulse Response, FIR) [29] pasa bajos. Estos son siempre estables y capaces de tener una fase lineal (retraso constante de grupo), lo que conlleva a una ausencia total de distorsión. Su valor de frecuencia en la banda de paso es de 0 a 10 Hz, y de 10 a 20 Hz en la banda de transición. En la banda de paso fue escogido un valor de atenuación de 1dB y en la banda de rechazo 98 dB (valor obtenido del cálculo de la SNR en la sección anterior).

El diseño del filtro fue realizado haciendo uso del toolbook de MATLAB denominado Filter Design and Analysis Tool (FDATool) [30]. El número de coeficientes obtenidos fue de 31.

B1.3. Creación de registros Mini-SEED.

La creación de los registros Mini-SEED se realiza a través del uso de la librería LIBMSEED [15]. Un registro Mini-SEED de 512 bytes está compuesto por los siguientes campos:

1. Una cabecera de 48 bytes de longitud, cuyos campos más relevantes son: un identificador para cada registro, detalle de la Estación Sismológica (Nombre, Código y Canal) e información respecto a la tasa de muestreo.
2. Dos campos denominados “blockettes”, de 8 bytes de longitud cada uno. La información más importante se ubica en el primer “blockette” y hace referencia al formato de codificación utilizada en el campo de datos [14].
3. Un campo de datos con los valores registrados por el sismómetro, de 448 bytes de longitud. La cantidad de datos que se almacenan en cada registro depende del formato de codificación y de la frecuencia de muestreo escogida.

Para el desarrollo del presente trabajo fue elegido un valor de 300 datos [28], para almacenar en cada registro Mini-SEED, y codificados según el formato de compresión STEIM2 [27].

Cada vez que finaliza la creación de un registro Mini-SEED, se procede a enviarlo hacia RINGSERVER. Dicho procedimiento será descrito a continuación.

B1.4. Envío de datos a RINGSERVER.

Cada registro Mini-SEED creado deberá ser enviado a RINGSERVER para que éste, a su vez, haciendo uso del protocolo SEEDLINK, los transmita a EARTHWORM. Esto será posible gracias a la implementación de la librería LIBDALI [13], la cual permite transferir registros Mini-SEED (a través de TCP/IP) por medio del protocolo DATALINK [12] hacia RINGSERVER.

El diagrama de conexión entre la aplicación (Cliente) y RINGSERVER se observa en la figura 2.

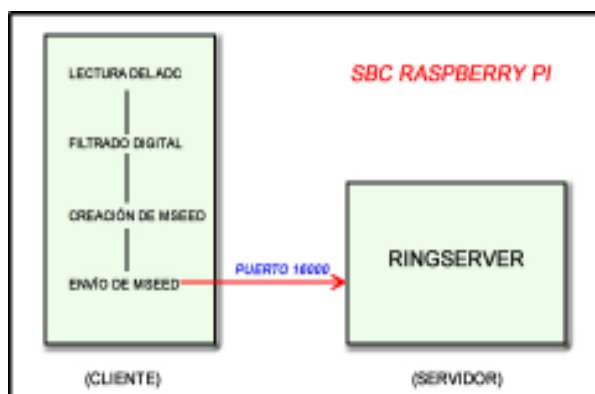


Figura 2. Envío de registros Mini-SEED hacia RINGSERVER.

B2. Configuración y Operación del DAS.

Uno de los requerimientos solicitados al momento de desarrollar este trabajo fue la posibilidad de configurar (parámetros esenciales) y operar el DAS en forma remota. Para cumplir con ello, será implementado un sitio Web dinámico desarrollado en Lenguaje PHP (Hypertext Preprocessor) [31]. Raspbian trae incorporado el Servidor Web Apache [32], el cual soporta el Lenguaje PHP.

El sitio Web implementado, permitirá a través de una interfaz gráfica, observar y modificar diversos parámetros del Sistema de Adquisición de Datos desarrollado (Figura 3). El acceso al mismo, será posible a través de un browser, mediante una dirección IP asignada por defecto.



Figura 3. Modificación de parámetros del DAS.

A través de este sitio Web, el operario también podrá iniciar, detener, reiniciar y apagar el Sistema de Adquisición de Datos en forma remota.

V. VALIDACIÓN & VERIFICACIÓN

Se procedió a revisar unitariamente cada una de las partes que conforman el Sistema, a fin de verificar que cada una cumpla con los requisitos propuestos.

Se realizaron mediciones en la entrada y salida de las etapas de Amplificación y Filtrado (acondicionamiento de la señal), utilizando como señal de referencia, la respuesta del sismómetro S-13 tras ser excitado por una señal sinusoidal de frecuencia variable [34]. El filtro Anti-Aliasing desarrollado, presentó un desplazamiento lineal con la frecuencia en la banda de interés (0 a 10 Hz), cumpliendo con las especificaciones planteadas en la etapa de diseño (figura 4), evitando de esta forma distorsión de la señal.

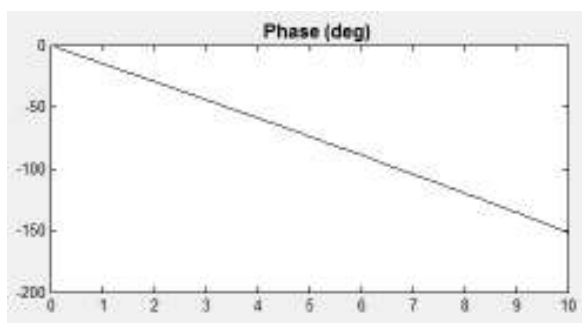


Figura 4. Respuesta de fase del filtro Anti-Aliasing.

Sobre la banda de paso del filtro la atenuación fue nula. En la banda de rechazo, ante una señal de 100 Hz, la atenuación fue superior a los 98 dB, valor planteado al momento del diseño del filtro.

Se realizó también de forma unitaria, la verificación de la frecuencia de lectura del ADC (100 Hz) a través del bus SPI por parte de la SBC Raspberry Pi, haciendo uso de la señal SIGALRM.

Luego se procedió a integrar el Hardware, encargado de realizar el acondicionamiento de la señal, a la SBC Raspberry Pi, para efectuar la verificación del filtro digital implementado por Software, como así también la transmisión de datos hacia EARTHWORM. Para efectuar estas pruebas fue necesario implementar un Servidor con EARTHWORM para verificar la comunicación del Sistema, y el uso del Software SWARM [33] en una Computadora Personal (Personal Computer, PC) utilizada como cliente, para la comprobación de los datos transmitidos.

Para finalizar, se verificó la configuración y operación del DAS a través del sitio Web implementado para tal fin.



Figura 5. DAS, sensor e instrumentos de medición en laboratorio.

El detalle de cada una de las pruebas realizadas y sus resultados se han incluido en la documentación del Sistema [35].

La prueba del Sistema de Adquisición de Datos en un terreno con abundante actividad sísmica durante un período considerable de tiempo ha quedado para una etapa posterior del presente trabajo y no ha podido llevarse a cabo para ser presentado en esta publicación.

VI. CONCLUSIONES

El presente trabajo plasma el desarrollo de un Sistema de Adquisición de Datos Sísmicos capaz de comunicarse en forma directa al Sistema Colector de Datos EARTHWORM en tiempo real, permitiendo enviar a éste, los datos capturados de un sismómetro (sensor) sin la necesidad y complejidad que conlleva el utilizar un Servidor que haga las veces de intermediario, como sucede al adquirir de terceros, un DAS de prestaciones similares.

Durante del proceso de desarrollo se utilizaron prácticas de Ingeniería de Software, como requerimientos, diseño por etapas y verificación, poniendo de manifiesto la utilidad de estas técnicas en el desarrollo de Sistemas Embebidos. De igual forma, hay un alto grado de contenido en este trabajo de técnicas de procesamiento de señales, para el tratamiento

de la señal sísmica en varias de las etapas que conforman el Sistema.

El Sistema de Adquisición logrado hace uso del difundido protocolo en sismología para el intercambio de datos SEEDLINK, lo que permite su integración a cualquier red sísmológica.

La configuración y operación del DAS a través de una interfaz Web, da la posibilidad de utilizar el mismo en lugares de difícil acceso, donde normalmente se encuentran ubicadas las Estaciones Sísmológicas.

Tras los resultados obtenidos en laboratorio, queda expuesto, en principio, que es viable encarar el desarrollo de un Sistema, como el presentado en esta publicación, para instituciones como INPRES, logrando flexibilidad, conexasión e independencia de terceros.

REFERENCIAS

- [1] Bolt, B. "Earthquakes and geological discovery". Scientific American Library. W.H. Freeman & Co, 221 p. New York, USA. 1993.
- [2] EarthWorm Central. Instrumental Software Technologies, Inc. <http://www.earthwormcentral.org/> [Consulta: Enero de 2014]
- [3] Andres Heinloo. "Networked Seismographs: GEOFON Real-Time Data Distribution (SEEDLINK)". Orfeus Newsletter, Volumen 2 N° 3, Diciembre 2000. <http://www.orfeus-eu.org/organization/Organization/Newsletter/vol2no3/geofon.html> [Consulta: Febrero de 2014]
- [4] Raspberry Pi Foundation. University of Cambridge's Computer Laboratory, 2014. <http://www.raspberrypi.org/> [Consulta: Marzo de 2014]
- [5] Raspbian Operating System. Developers: Mike Thompson and Peter Green. <http://www.raspbian.org/> [Consulta: Marzo de 2014]
- [6] SeisComp. Seismological Software. GEOFON Program. <http://www.seiscomp3.org/> [Consulta: Marzo de 2014]
- [7] "Serial Peripheral Interface (SPI) & Inter-IC (IC2) (SPI_I2C) Application Note" - Renesas, 2003. http://documentation.renesas.com/doc/products/region/rtas/mpumcu/apn/spi_i2c.pdf [Consulta: Abril de 2014]
- [8] "Operation and Maintenance Manual Portable Short-Period Sesimometer, Model S-13". Teledyne Geotech, 1983.
- [9] Analog Devices AD622ANZ Data Sheet. http://www.analog.com/static/imported-files/data_sheets/AD622.pdf [Consulta: Marzo de 2014]
- [10] Jens Havskov and Gerardo Alguacil. "Instrumentation In Earthquake Seismology". 1st ed. Springer. Netherlands. 2004.
- [11] Texas Instruments ADS8507 Data Sheet. <http://www.ti.com/lit/ds/symlink/ads8507.pdf> [Consulta: Marzo de 2014]
- [12] Protocolo de Comunicaciones DATALINK - Chad Trabant, IRIS Data Management Center, 2013. <https://seiscode.iris.washington.edu/svn/orb2ringserver/tags/release-1.0/libdali/doc/DataLink.protocol> [Consulta: Febrero de 2014]
- [13] LIBDALI, DataLink Client Library. Chad Trabant, IRIS Data Management Center, 2013. <http://www.iris.edu/pub/programs/ringserver/libdali-1.6.tar.gz> [Consulta: Febrero de 2014]
- [14] Incorporated Research Institutions for Seismology (IRIS) "SEED Reference Manual". 3rd ed. 2006. Appendix G - Data Only SEED Volumes (Mini-SEED).
- [15] LIBMSEED, The Mini-SEED library. Chad Trabant, IRIS Data Management Center, 2013.
- [16] RINGSERVER, A generic ring buffer and a SeedLink Server - Chad Trabant, IRIS Data Management Center, 2013. <https://seiscode.iris.washington.edu/projects/ringserver/> [Consulta: Febrero de 2014]
- [17] Parámetro establecido por el Departamento Laboratorio Sísmológico del INPRES.
- [18] Milton Kaufman y Arthur H. Seidman. "Manual para Ingenieros y Técnicos en Electrónica". 2da ed. McGraw-Hill. 1992.
- [19] Ashok Ambardar. "Procesamiento de Señales Analógicas y Digitales". 2da ed. Thomson Learning. 2002.
- [20] Bonnie C. Baker. "Anti-Aliasing, Analog Filters for Data Acquisition Systems". Application Note 699. Microchip Technology Inc. 1999.
- [21] Valores registrados por el Departamento Laboratorio Sísmológico, en base a al registro de sismos locales con epicentro a una distancia de 100 km de la ubicación del sísmómetro.
- [22] Steven W. Smith. "The Scientist and Engineer's Guide to Digital Signal Processing". 2nd ed. Analog Devices. California Technical Publishing. San Diego, California, USA. 1999.
- [23] Enrique Bueno Gimeno. "Diseño de un Convertidor Analógico-Digital de Aproximaciones Sucesivas de bajo consumo y área reducida". Proyecto de Fin de Carrera. Austriamicrosystems AG. Universidad Politécnica de Valencia. 2010.
- [24] Texas Instruments LM324 Data Sheet. <http://www.ti.com/lit/ds/symlink/lm124-n.pdf> [Consulta: Agosto 2014]
- [25] Wiring Pi. GPIO Interface library for the Raspberry Pi. <http://wiringpi.com/> [Consulta: Marzo de 2014]
- [26] W. Richard Stevens, Stephen A. Rago. "Advanced Programming in the UNIX Environment". 2nd ed. Addison Wesley Professional. 2005.
- [27] Shu-Fang Newman. "Seismographic Data Compression". Institute of Technology, University of Washington, Tacoma. 2006.
- [28] Valor establecido de acuerdo al formato de compresión utilizado y a la frecuencia de muestreo escogida, según reomendación de Chad Trabant (Autor de la librería LIBMSEED).
- [29] Emmanuel C. Ifeachor and Barrie W. Jervis. "Digital Signal Processing: A Practical Approach". 1st ed. Addison-Wesley Publishing Company. USA. 1993.
- [30] Filter Design and Analysis Tool (FDATool). MATLAB Documentation, 2014. <http://www.mathworks.com/help/signal/ug/overview.html#br179zi-4> [Consulta: Agosto de 2014].
- [31] PHP Official Web Site. <http://php.net/> [Consulta: Mayo de 2014]
- [32] Apache. HTTP Server Project. <http://httpd.apache.org/> [Consulta: Mayo de 2014]
- [33] SWARM. Seismic Wave Analysis and Real-time Monitoring. Peter Cervelli, United States Geological Survey (USGS), 2014. <http://volcanoes.usgs.gov/software/swarm/index.php> [Consulta: Mayo de 2014]
- [34] Parámetro establecido por el Departamento Laboratorio Sísmológico del INPRES para la verificación de cada uno de sus Sistemas de Adquisición de Datos.
- [35] Documentación técnica. Validación & Verificación. http://www.inpres.gov.ar/docs/Doc_Sistema_de_Adquisicion_de_Datos_Sismicos.pdf

Controlador de acceso mediante tarjeta inteligente

Fernando Minciocchi
Universidad Tecnológica Nacional
Facultad Regional Bahía Blanca
Bahía Blanca, Argentina
Email: fer.minciocchi@hotmail.com

Resumen—Se presenta un sistema electrónico que permite controlar el acceso al laboratorio de electrónica de la Universidad, por medio de una tarjeta inteligente. El sistema controla la identidad de quien ingresa, y el número de tarjeta. Cuando un usuario ingresa, el sistema guarda la fecha y hora del ingreso, junto con el nombre de usuario, es decir, que posee un historial del acceso de las personas al laboratorio. Funciona en conjunto con una computadora por comunicación USB para poder registrar una tarjeta, dar de baja una tarjeta o mirar el historial de ingreso, sin embargo, no es necesaria una computadora para el funcionamiento normal. El sistema está compuesto por un lector de tarjetas, un kit de desarrollo embebido, un pendrive, un adaptador, una cerradura eléctrica, un dispositivo sonoro y un display.

Palabras clave—controlador de acceso, tarjeta inteligente, sistema electrónico.

I. INTRODUCCIÓN

Hoy en día es muy común el uso de estas tarjetas para los parquímetros, el transporte público, controladores de acceso, entre otras aplicaciones. Una de las razones por las que es conveniente el uso de estos sistemas, es que los usuarios pueden interactuar de forma rápida, otra razón es porque poseen una ventaja en cuanto a la seguridad, por ejemplo, si una persona extravía su tarjeta el sistema puede invalidar la tarjeta para que otro usuario no pueda ingresar con ella.

El sistema que se va a presentar permite que solo los profesores y alumnos del departamento de electrónica, que cuenten con una tarjeta de identificación, puedan ingresar al laboratorio.

A continuación se realizará una definición del problema a resolver, así como también una descripción del sistema, y sus principales características.

II. DEFINICIÓN DEL PROBLEMA

El laboratorio de electrónica posee instrumentos de medición, los cuales deberían ser utilizados por los alumnos y profesores del departamento de electrónica. La puerta de entrada debe ser controlada por el sistema y cada acceso al laboratorio debe ser registrado.

III. DESCRIPCIÓN DEL SISTEMA

El sistema representado en la Figura 1 está compuesto por un sistema embebido STM32f4 Discovery (STM32F407VG) el cual controla los periféricos y es la unidad central de

procesamiento. El lector de tarjetas es un módulo Mifare RC522 que utiliza comunicación SPI para la transmisión de datos con el sistema embebido, y RFID para la comunicación inalámbrica entre la tarjeta y el lector. Un display que muestra la hora y fecha, y da la bienvenida al usuario cuando ingresa, un control para cerradura eléctrica, una bocina que se activa cuando se lee una tarjeta, un pendrive que es la memoria primaria y un adaptador de comunicación RS232 a USB.

Estos componentes se interconectan entre sí con una placa madre diseñada específicamente. El sistema embebido posee un reloj propio el cual es alimentado por una pila externa de 3Volts, y se necesita de una computadora y de la red de alimentación más precisamente de una fuente de alimentación de 220Volts de corriente alterna a 12Volts de corriente continua conectada a la red. Funciona con cualquier sistema operativo que posea un programa de emulación de terminal serie. Los sistemas operativos más utilizados poseen esta característica.

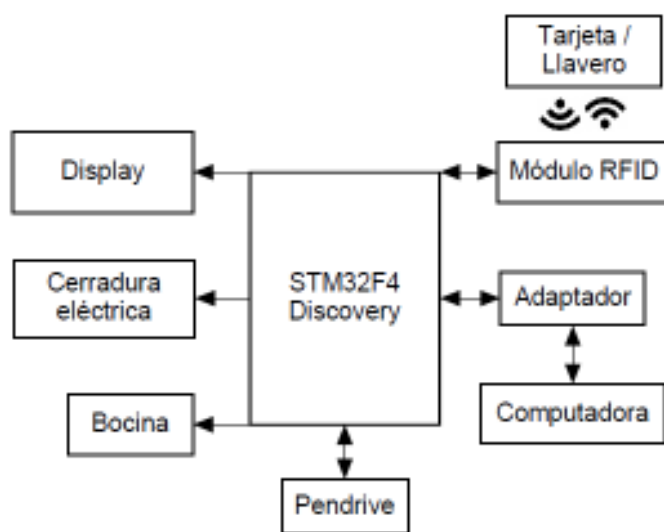


Figura 1. Diagrama en bloques del sistema completo.

Comparado con otros controladores de acceso existentes, se puede destacar su bajo costo, y la opción de un display.

IV. OPERACIÓN DEL SISTEMA

El diagrama de flujos de la figura 2 explica el funcionamiento del sistema.

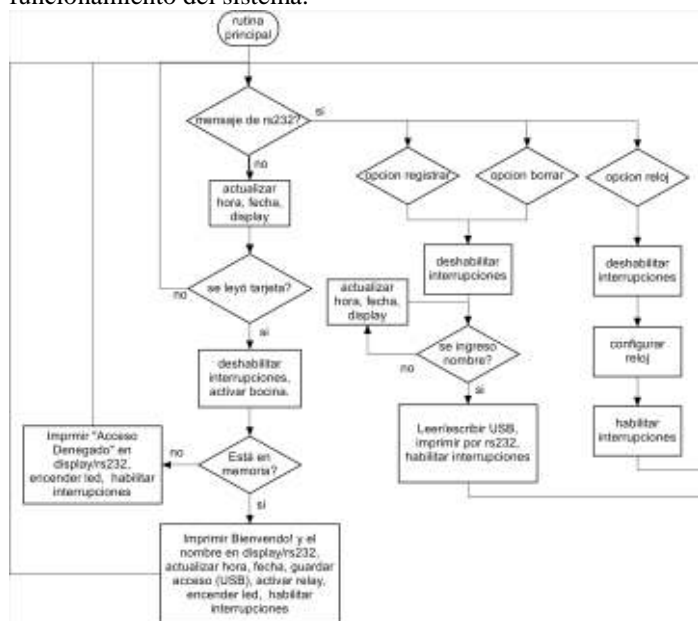


Figura 2. Diagrama de flujos

V. SOFTWARE DEL MICROCONTROLADOR

El entorno de desarrollo para el software del microcontrolador del sistema embebido es "CooCox CoIDE", Version: 1.7.3. El lenguaje de programación utilizado es C/C++.

VI. CARACTERISTICAS DEL SISTEMA

La cantidad de usuarios y cantidad de registros es prácticamente ilimitada ya que se almacenan en archivos de textos. Si se cuenta con decenas de miles de usuarios el sistema puede demorar en encontrar una coincidencia, esto sucede aproximadamente por encima de los 100.000 usuarios.

VII. CONCLUSIONES Y FUTURO DEL PROYECTO

Podemos concluir que el dispositivo resuelve el problema planteado, es fácil de introducir sin modificaciones en las instalaciones, y el costo de adquisición es bajo respecto a otros controladores de acceso.

En un futuro, se tiene en mente reemplazar la comunicación USB por una comunicación inalámbrica Wifi (estándar 802.11), esto permitiría conectar de ser necesario más de un dispositivo lector en distintas ubicaciones además de eliminar la conexión por cable con una computadora.

AGRADECIMIENTOS

Se desea agradecer a los profesores por compartir sus conocimientos e ideas, su disposición, y su apoyo a la realización del proyecto.

REFERENCIAS

- [1] "The C Reference Manual" by Dennis M. Ritchie, a version of which was published in the C programming Language by Brian W. Kernighan and Dennis M. Ritchie, Prentice-Hall, Inc., (1978).
- [2] "The C++ Programming Language" by Bjarne Stroustrup, Fourth Edition, PEARSON.
- [3] Reference Manual RM0090 by STMicroelectronics. STM32F405xx/07xx, STM32F415xx/17, STM32F42xxx and STM32F43xxx advanced ARM-based 32-bit MCUs.
- [4] Programming manual PM0214 by STMicroelectronics. STM32F3xxx and STM32F4xxx Cortex-M4.
- [5] Datasheet – production data by STMicroelectronics. STM32F405xx and STM32F407xx.
- [6] Product data sheet MFRC522 Standard 3V MIFARE reader solution by NXP Semiconductors. Rev. 3.7.

Equipo de bajo costo para la construcción de circuitos impresos mediante el uso de luz ultravioleta

Gelosi Iván Exequiel, Casadei Manuel, Calcagno Emanuel, Martin Nicolás, Uriz Alejandro José, Agüero Pablo
Laboratorio de Comunicaciones, Facultad de Ingeniería
Universidad Nacional de Mar del Plata, Buenos Aires, Argentina
egelosi@fi.mdp.edu.ar, manucasadei@gmail.com, emalc90@fi.mdp.edu.ar, nico12martin@gmail.com,
ajuriz@fi.mdp.edu.ar, pdaguero@fi.mdp.edu.ar

Abstract—En este trabajo se presenta un equipo capaz de realizar un prototipado rápido de circuitos impresos mediante luz ultravioleta. Este método permite desarrollar diseños de circuitos impresos con mayor precisión que los métodos comúnmente conocidos. Este trabajo está motivado en la necesidad de alumnos de la Cátedra Medios de Transmisión de la carrera de Ingeniería Electrónica de la Universidad Nacional de Mar del Plata. En el marco de este curso se les exige a los alumnos implementar un trabajo final de la cátedra donde apliquen técnicas de diseño basadas en *microstrip*, para las cuales se requiere una implementación con muy baja tolerancia. Cabe destacar además, que el equipo desarrollado permite llevar a cabo un prototipado de calidad y muy bajo costo de producción.

Keywords—Circuitos impresos, Luz ultravioleta, Insoladora.

I. INTRODUCCIÓN

EN el campo de la electrónica, uno de los aspectos más importantes a la hora de ensamblar un sistema, es el circuito impreso o PCB (de su traducción del inglés: *Printed Circuit Board*). Un PCB cumple el rol de unir mecánica y eléctricamente los distintos componentes de un sistema electrónico. En la actualidad, existen circuitos impresos de muy alta complejidad, debido a que existen componentes electrónicos con alta densidad de pines. Además, se suele requerir una gran cantidad de pistas (conexiones) que pueden ser de tamaños pequeños. Incluso, debido a la alta densidad de pistas, se requiere una separación muy baja entre las mismas. Es por ello que se requiere que las técnicas de construcción de circuitos impresos requieran una muy buena resolución y una muy baja tolerancia de construcción.

Debido a que los sistemas utilizados para la realización de circuitos impresos de calidad industrial son de alto costo para estudiantes y los plazos de construcción suelen ser de al menos dos semanas, dió lugar a la búsqueda de métodos más económicos y con la mayor precisión posible. Luego de observar los distintos métodos existentes, se llegó a la conclusión que el que se hallaba con los requerimientos buscados era por medio de luz ultravioleta.

Entre los métodos para la realización de circuitos impresos se pueden hallar:

Circuitos impresos elaborados con tinta indeleble: Esta manera de producir placas de circuito impreso, es la más económica a baja escala que existe, ya que solo es necesario

tinta indeleble, la placa donde se plasma el diseño y el agente que se encarga de corroer la superficie de cobre no deseada.

Este método se realiza mediante el dibujo manual de las pistas del circuito, razón por la cual resulta muy difícil llegar a obtener trabajos de mediana complejidad, además de carecer de calidad de impresión.

Circuitos impresos elaborados con logotipo: La elaboración mediante logotipo es muy similar a la mencionada anteriormente. Esta solo difiere en la forma de impresión ya que en el procedimiento anterior se dibuja a mano el circuito sobre la placa con la tinta indeleble.

Esta técnica consiste en colocar sobre la placa logotipos (calcomanías) que tienen diversas figuras: pistas y terminales de componentes. Las calcomanías tienen la característica de que inhiben sobre la superficie cubierta la acción corrosiva. De esta forma se logra mejor calidad que con el procedimiento anterior, aunque no deja de ser una forma artesanal de producción. De la misma manera resulta muy difícil llegar a obtener diseños de circuito impresos de mediano y gran tamaño.

Circuitos impresos elaborados con la técnica de serigrafía: Esta técnica tiene la ventaja de obtener trabajos de buena calidad a un precio razonable. Además permite la realización de varias copias del mismo diseño una vez que se ha revelado en la seda, lo que nos lleva a una producción en serie de tarjetas impresas. Aunque no deja de ser un proceso manual, es válida y permite obtener trabajos con la suficiente calidad y presentación para la realización de prototipos electrónicos y/o aplicaciones específicas de la Industria.

Circuitos impresos elaborados con la técnica de luz ultravioleta: El método fotográfico para la elaboración de circuitos impresos se lleva a cabo a partir de un fotolito negativo, ya sea de un dibujo manual en papel o de un diseño por computadora impreso. Esta técnica posee la ventaja de obtener una buena calidad de trabajo para mediana y alta complejidad, y si bien no es económico como las otras técnicas mencionadas anteriormente, lo es en relación a la calidad de trabajo obtenido.

Circuitos impresos elaborados por fabricantes: Este es el método más efectivo dado que en él los circuitos poseen una excelente definición. Si bien es la mejor opción para la realización de un circuito impreso, entre las técnicas mencionadas anteriormente, es la de mayor costo. Además tienen un tiempo de fabricación de al menos dos semanas.

Dada la comparación anterior, para el caso objeto de este

trabajo es notable que la mejor técnica a realizar es mediante luz ultravioleta, por lo que se realizó un dispositivo que permitiera la realización de dicho método.

Al iniciar la búsqueda de elementos UV se tuvieron en cuenta la utilización de LED o lámparas de tubo. Dado que el consumo de los LED es menor, y posee mayor facilidad de utilización, fue seleccionado para la realización del trabajo.

Considerando que la luz ultravioleta (UV) causa efectos perjudiciales para la salud es necesario que la ubicación donde se encuentre actuando no esté en contacto con los usuarios. Por dicho motivo se tomaron las precauciones necesarias para evitar el contacto directo, colocando sistemas de seguridad que se mencionarán en la siguiente sección. El trabajo se organiza de la siguiente manera: en la Sección 2 el desarrollo del equipo, con los cálculos realizados, en la Sección 3 el proceso para realizar los circuitos impresos mediante el método propuesto y por último, en la Sección 4, las conclusiones del trabajo junto con los objetivos a futuro.

II. DESARROLLO DEL EQUIPO

Al momento de iniciar el desarrollo del equipo, una vez seleccionada la tecnología UV a utilizar, se necesitaba un sistema de control para evitar la sobre-exposición a la luz UV, dado que en dicho caso se afectaría la impresión realizada sobre el cobre. Para ello se dispuso de un contador programable, que permite una cuenta regresiva de 0 a 99 segundos. El objetivo del mismo es establecer el funcionamiento del equipo durante el intervalo de tiempo necesario. Dicho contador posee botones para aumentar y disminuir el tiempo de funcionamiento de acuerdo al nivel de exposición requerido para cada caso, un botón de inicio y uno de finalización. Para el desarrollo del temporizador, se utilizó un microcontrolador PIC18F2550 de Microchip[1], dado que éste dispone de la arquitectura necesaria para la implementación del contador y la interfaz de control del equipo.

Como se mencionó anteriormente, para evitar la exposición del usuario a los rayos UV, se colocó un sensor, de modo que en el caso que la tapa se encuentre abierta se desactive la placa de LED's.

Para la utilización de los LED's, se necesita que la iluminación sea homogénea, por lo que se debieron realizar una serie de cálculos. De este modo, se aseguró dicha homogeneidad, y a su vez se colocó un vidrio esmerilado, de modo que éste se comporte como un difusor y realizar una mejor distribución de la luz. Un diagrama de bloques del sistema se aprecia en la Figura 1.



Fig. 1. Diagrama en bloques

A. Cálculos

Para los cálculos, primero se debió analizar la distancia a la cual se colocarían, y la separación entre los mismos. Para ello se analizó el esquema de la Fig. 2.

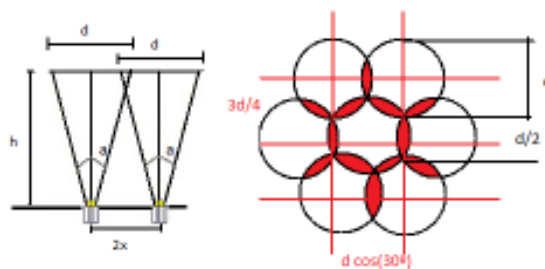


Fig. 2. Esquema. A la izquierda vista lateral de distribución de LED's, a la derecha distribución desde vista superior.

Siendo $d\cos(30)$ la distancia entre leds, h la altura entre los leds y la ubicación del vidrio esmerilado para colocar las placas, y a el ángulo de apertura de la luz. Se decidió utilizar la disposición mostrada en la figura para lograr que la distribución de luz sea homogénea. Considerando las Ecuaciones (1), (2) y (3) se pueden obtener los valores de distancias.

$$d_{entre.filas} = \frac{d}{2} + \frac{d}{4} = \frac{3d}{4} \quad (1)$$

$$x = \frac{d}{2} \times \cos(30) \quad (2)$$

$$d_{entre.leds} = 2 \times x = d \times \cos(30) \quad (3)$$

III. PROCESO DE REALIZACIÓN DE CIRCUITOS IMPRESOS

La realización del circuito impreso mediante esta técnica consta de 7 simples pasos, en los cuales se debe asegurar al comienzo de tener los siguientes elementos dado que en el caso de detener el proceso, podría no obtenerse la calidad deseada. Los elementos son:

- Percloruro Férrico (El percloruro no férrico también puede ser utilizado).
- Film fotosensible (el utilizado es el que posee 3 capas y será explicado a continuación).
- Pistola de calor.
- Impresión en papel transparente del circuito impreso a realizar (en negativo).
- Revelador.
- Removedor.

Para el primer paso, se debe tomar la placa en la cual se desea realizar el circuito y limpiarle la suciedad de modo uniforme (Para realizar este proceso puede ser utilizado una esponja de acero fina o una esponja con jabón), una vez eliminada toda la suciedad sobre la superficie limpiarla con agua limpia.

Antes de continuar, debe asegurarse que el ambiente en el cual se realice el procedimiento posea una iluminación tenue (para no revelar el film antes de completar el proceso. No es necesario un cuarto oscuro).

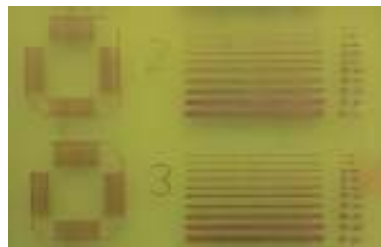
Una vez terminado el primer paso, se debe recortar un trozo de film fotosensible del tamaño de la placa, en el cual sobresalga pequeños extremos. Se pueden observar que el film tiene 3 capas, de las cuales la dos exteriores una es opaca y otra brillante. Primero se debe remover la opaca, es posible notarlo dado que esta es mas gruesa que la otra. Para removerlo de modo simple, se debe colocar un adhesivo sobre la capa que se desea retirar y luego lentamente retirarla y se desprenderá de las dos capas restantes. Antes de colocar el film se debe rociar con agua (solo que humedezca la placa), dado que el agua ayudará al colocar el film a poder moverlo y colocarlo en la posición mas centrada. Una vez colocado el film, presionar con un paño toda la superficie hasta que no queden rastros ni de burbujas ni de agua. Luego de pasar el paño a toda la superficie calentar el film con la pistola de calor para que de este modo el agua restante sea evaporada (por un corto plazo, dado que sino puede perder la adhesión el film). Repetir alternadamente con el paño y la pistola de calor, entre 3 o 4 veces hasta notar la perfecta adhesión del film.

Una vez adherido, como se presenta en la Fig. 3.a se debe dejar reposar entre 5 ó 10 minutos de modo que se enfríe la placa para poder pasar al siguiente paso.

Una vez que la placa se enfría hasta la temperatura ambiente, se debe colocar sobre ella el circuito impreso que se desea transferir a la misma (recordar que debe utilizar papel transparente e imprimir utilizando colores en negativo). La placa con la impresión se colocan sobre el vidrio esmerilado el cual iluminará con luz UV durante el tiempo necesario. Este tiempo dependerá del espesor y la separación de las pistas del circuito a construir. Con fines de calibrar este parámetro, se realizaron pruebas donde se expusieron pistas de similares características



(a) Placa con film



(b) Prueba de tiempos



(c) Placa luego de ser insolada



(d) Placa luego de la sustancia removedora



(e) Placa Finalizada

Fig. 3. Imágenes placa, de arriba a abajo: (a) Placa con el film adherido. (b) Experimentación de tiempos de exposición. (c) Placa luego de ser insolada. (d) Placa una vez completada la etapa removedora. (e) Placa finalizada.

a luz UV durante distintos períodos de tiempo. El resultado de dicho experimento se aprecia en la Fig. 3.b

Una vez finalizado el tiempo de exposición, se debe remover el segundo film autoadhesivo. Una vez quitado este segundo film se coloca la placa sobre revelador, en el cual debe permanecer un breve período. Con la ayuda de un pincel se debe ir removiendo los excesos (porciones expuestas a la luz UV. Es decir, la porción que debe ser removida por el ácido). En la Fig. 3.d se muestra a modo de ejemplo el resultado de este proceso.

Luego, se debe colocar al PCB dentro del percloruro férrico para que este remueva todo el cobre indeseado dentro de la placa. Una vez finalizado este paso, solo queda pasar la placa por el removedor para que quite los restos del film que quedaron sobre la placa. Para ello, se deposita la placa dentro de la sustancia removedora basada en soda cáustica. Finalmente, luego de un cierto tiempo se obtendrá el resultado final de la técnica mostrado en la Fig. 3.e.

Como puede verse, el equipo desarrollado permite obtener circuitos impresos de una forma simple y económica. Cabe destacar que utilizando esta técnica se han desarrollado también circuitos impresos de dos capas. Siendo necesario para este caso, transferir las impresiones en ambas caras de la placa antes de sumergirlas en removedor. A continuación se describen las conclusiones del trabajo. Cabe destacar que mediante esta técnica ha sido posible construir circuitos impresos con pistas de un espesor inferior a las 10 milésimas de pulgada y separación entre pistas con una resolución de hasta 8 milésimas de pulgada. De esta forma, se dispone de una tecnología capaz de realizar prototipado rápido de PCBs para montar circuitos integrados 64TQFP[2] ó 32VQNF[3] por ejemplo.

IV. CONCLUSIÓN

En este trabajo se presentó un equipo capaz de realizar prototipado rápido de circuitos impresos, basado en un PIC18F2550 de Microchip. Además, se desarrolló el proceso utilizado para la elaboración de un circuito impreso utilizando esta técnica. Este equipo permite construir circuitos impresos con pistas de un espesor de hasta 7 milésimas de pulgada y con una separación entre pistas de hasta 8 milésimas de pulgada. Cómo puede ser observado en la Fig. 5. El costo del equipo es de aproximadamente U\$S 100 siendo una herramienta de interés para cátedras cuyos alumnos deban realizar prácticas que requieran circuitos impresos para sus trabajos prácticos. En el futuro se desean seguir estudiando técnicas para mejorar la calidad y precisión de los circuitos impresos obtenidos. Para ello se analizarán técnicas como la mascara antisoldante, el estañado de pistas, la tipografía entre otras. En la Fig. 4 se puede observar el prototipo del equipo en funcionamiento.



(a) Equipo con la tapa cerrada



(b) Equipo con la tapa abierta sin funcionar



(c) Equipo con la tapa abierta funcionando

Fig. 4. Imágenes del equipo. De arriba a abajo: (a) Equipo con la tapa cerrada. (b) Equipo con la tapa abierta sin iniciar su funcionamiento. (c) Equipo con la tapa abierta una vez iniciado su funcionamiento.



Fig. 5. En la figura se puede observar en comparación la placa finalizada a la derecha y a la izquierda la placa insolada antes de completar el proceso. Aquí se observa la definición lograda por el equipo.

REFERENCES

- [1] MICROCHIP TECHNOLOGY INC., *Microprocesador PIC18F2550*, 2006, <http://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf>.
- [2] ST MICROELECTRONICS, *64 LEAD Thin Quad Flat Package*, 2003, http://pdf.datasheetcatalog.com/datasheets/134/306806_DS.pdf.
- [3] TEXAS INSTRUMENTS, *Low-Power Stereo Codec with 6 Inputs, 6 Outputs, Speaker/HP Amp and Enhanced Digital Effects*, 2007, <http://www.ti.com/product/tlv320aic3101>.

Diseño y Desarrollo de un Podómetro para Detección de Celos y Monitoreo de Ganado Bovino

Anderson Julca Vasquez, Leopoldo Yabar Escribanel

Universidad Tecnológica del Perú

Facultad de Ingeniería Sistemas y Electrónica

E-mail: andersonjulcav@gmail.com, lyabar@grupoutp.edu.pe

Resumen—El presente artículo muestra el diseño y desarrollo de un Podómetro detector de celos para monitoreo de ganado bovino basándose en el hecho de que una vaca incrementa su actividad física cuando se encuentra en celo. Para ello se utiliza un acelerómetro digital y un microcontrolador de bajo consumo, juntos registran la cantidad de pasos dados, luego a través de un transmisor, se envían dichos registros hacia un módulo receptor. Así mismo, se recurrieron a técnicas de programación y diseño de hardware orientado al bajo consumo y a la vez una interfaz gráfica para observar las señales obtenidas. El prototipo desarrollado fue aplicado sobre un ejemplar vacuno, lográndose recolectar datos sobre su aceleración al andar, con el fin de determinar un umbral de aceleración óptimo, e introducirlo como parámetro en el algoritmo de detección de pasos por umbral dinámico

Palabras clave—Podómetro, estado de celo, detección de celos, hato ganadero, ultra bajo consumo, acelerómetro, transmisor sub 1 GHz.

I. INTRODUCCIÓN

La ganadería intensiva consiste en la crianza del ganado bajo condiciones climáticas y ambientales favorables creadas artificialmente, utilización de alimentos enriquecidos e introducción de mejoramiento genético dentro del hato, con el fin de incrementar la producción. Específicamente dentro de la ganadería lechera, se requiere alcanzar un parto por año, para dicho propósito, la detección de celos es un factor determinante. Para detectar el celo existen varios métodos, siendo el uso de podómetros uno de ellos, debido a que las vacas incrementan notablemente su actividad física durante éste periodo.

La utilización de podómetros dentro de la ganadería no es novedad, a tan sólo tres años de creado el primer podómetro comercial (1965), en 1968 T. L. Powell utiliza podómetros mecánicos para medir la distancia recorrida por ovejas [1], en 1977 D. M. Anderson utiliza podómetros digitales para monitorear el recorrido de vacas [2], por otro lado en Israel S.A.E. Afikim (Hoy Afimilk), lanza su sistema computarizado de monitoreo en 1984 [3]. Años después el 5 de mayo de 2008, H. Flexer y E. Heimlich presentan la patente de un sistema que determina si una vaca está descansando o no [4], el cual sería integrado al nuevo “Pedometer Plus” de Afimilk e introducido al mercado en 2011 en los Estados Unidos.

La situación en Latinoamérica ha evolucionado a buen ritmo en los últimos años. En 2006, estudiantes argentinos desarrollaron detectores de monda por presión [5]; a su vez R. Murray ha obtenido muy buenos resultados integrando sincronización con monitoreo de actividad (sistema denominado “4-20”) para ganado semiestabulado en Argentina [6]. Científicos brasileños han logrado integrar algoritmos de lógica difusa a la detección de celos [7]. Para 2010 en Colombia, se ha presentado un sistema que detecta celos, envía SMS y almacena resultados en la web [8]. En Perú la introducción de podómetros al mercado ha sido lenta pero ascendente [9].

En el Perú aún se sigue utilizando la detección de celos por simple observación, realizada por una o varias personas según esté programado. Lo cual resulta ineficiente debido a la gran variación en los indicadores de celos visuales del animal -en época lluviosa aumentan a 32.7% los celos anovulatorios [10]-, además si el hato crece, la observación se vuelve complicada. Se ha demostrado que una pobre detección de celos provoca considerables bajas en la producción de leche, disminuyendo de 17.0 a 12.2 L la leche producida por vaca/día [10]. Por otro lado el manejo de tecnología y el nivel de capacitación técnico-empresarial de los ganaderos peruanos es bastante pobre. Más del 80% del ganado se encuentra en manos de productores con un nivel económico y cultural muy bajo. Según el último Censo Nacional Agropecuario realizado por el INEI, tan sólo el 1.6% de los productores del sector agropecuario cuentan con internet y más alarmante aún es que el 30% de estos productores no cuentan con ningún medio tecnológico [11].

Teniendo en cuenta que la Organización de las Naciones Unidas para la Alimentación y la Agricultura, recomienda un consumo per cápita de leche de 130 litros anuales [12], es necesario aumentar nuestra producción, según el Ministerio de Agricultura, nuestro promedio ha venido incrementándose, lográndose en 2013 un consumo de 80 litros per cápita, sin embargo, esta cifra es inferior si nos comparamos con países vecinos, Uruguay, Argentina, Brasil e Incluso Chile, ya han superado el índice recomendado [13]. Según Rolando Piskulich, presidente de la Asociación de Industriales Lácteos en el Perú (ADIL), la demanda de consumo de leche impone un gran reto, requiriéndose una visión de mediano y largo plazo,

demandando 114,000 nuevas hectáreas destinadas a ganadería, y aumentar la producción actual de 2 millones 400 mil toneladas de leche a 3 millones de toneladas para el año 2021 [12]. Es justificable por ende, el desarrollo de proyectos similares, lo cuales impulsen el desarrollo de este sector para lograr las metas tanto de producción como de consumo, y esto sólo es posible introduciendo tecnología [12]. De esta manera nos enfocamos primero en aumentar la producción y ganancias del ganadero poniendo a su disposición herramientas tecnológicas, permitiéndoles (con la debida capacitación) crecer como empresa y alcanzar las metas propuestas.

Por lo expuesto anteriormente, se fijó como objetivo diseñar y desarrollar un podómetro para monitorear la actividad física del ganado lechero, con lo cual se podrá determinar el estado de celo en el mismo. Para ello se buscó:

- Diseñar y construir un sistema embebido de ultra bajo consumo para adquirir los datos de aceleración.
- Desarrollar un algoritmo de detección de pasos para ganado bovino, basado en los datos obtenidos.
- Poder determinar el estado de celo a partir de los datos ofrecidos por nuestro podómetro.
- Lograr una autonomía de más de 5 años sin cambiar o recargar la batería.

II. METODOLOGÍA Y CRITERIOS DE DISEÑO

Para atacar el problema se propuso e implementó un sistema de medición de pasos (podómetro) conformado por tres bloques fundamentales, sensor, microcontrolador y transmisor. Como se observa en la Fig. 1. las señales de aceleración generadas por el animal a la hora de caminar son captadas por el acelerómetro y enviadas en intervalos regulares de tiempo hacia el microcontrolador, quien se encarga de procesarlas aplicándoles nuestro algoritmo de detección de pasos. Si el resultado es positivo se incrementa el registro contador. El contenido de dicho registro es transmitido a través del transceiver cada vez que se requiera. Del otro lado los datos son recibidos y decodificados por el módulo receptor, quien luego se los transmitirá a una PC, la cual tendrá instalado un software que ayude y permita interpretar los datos recibidos, como también deberá tener la capacidad de almacenar un historial con la información recibida.

A. Algoritmo de Detección de Pasos

Podemos mencionar tres tipos de algoritmos para detectar pasos:

- Algoritmo Pan-Tompkins, este tipo de algoritmo aplica una serie de filtros a la señal de aceleración, hasta que esta sea fácil de procesar, e.g. una señal con valores discretos +1 -1, donde +1 representa un paso.
- Algoritmo de detección por plantilla, en este tipo de algoritmo se tiene una señal plantilla, la cual representa la aceleración típica de un paso, luego se va comparando la señal recibida con dicha plantilla, se valida el paso cuando se encuentra una coincidencia.
- Algoritmo de cruce por umbral, en este algoritmo se configura cierta amplitud de aceleración como umbral

para definir si se ha dado un paso o no.

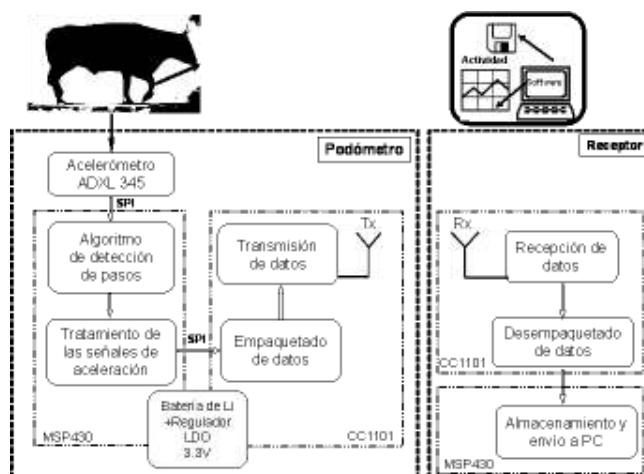


Fig. 1. Diagrama de bloques del sistema propuesto.

Tal cual se observa en la Fig. 2. Cada paso dado por el animal presenta una aceleración elevada por encima del nivel de reposo. Permittiéndonos implementar un algoritmo de detección por umbral sin requerir sofisticadas técnicas de procesamiento digital de señales. Sólo es necesario conocer un umbral de referencia (± 4 g según Fig. 2.) y una ventana de tiempo realista dentro de la cual podría ocurrir el fenómeno (i.e. 5 pasos por segundo sería un absurdo). Dados dichos parámetros, el flujo del algoritmo sería el mostrado en la Fig. 3.

Cabe resaltar que para no tener restricciones en cuanto a la posición del podómetro, podemos dinamizar el proceso seleccionando para cada ventana de tiempo el eje que presente mayor variación en amplitud (siempre y cuando supere al umbral de referencia), de esta forma el umbral dinámico se define según (1).

$$Umbral = \frac{muestra\ máxima + muestra\ mínima}{2} \quad (1)$$

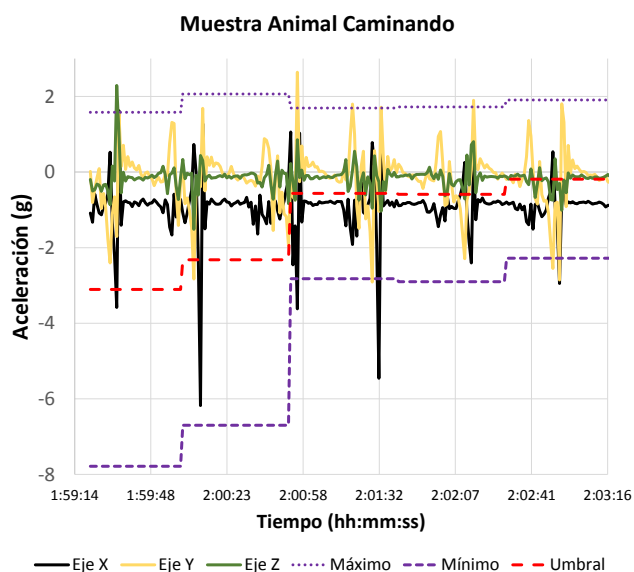


Fig. 2. Aceleración del animal al andar, se observa la adaptación del umbral.

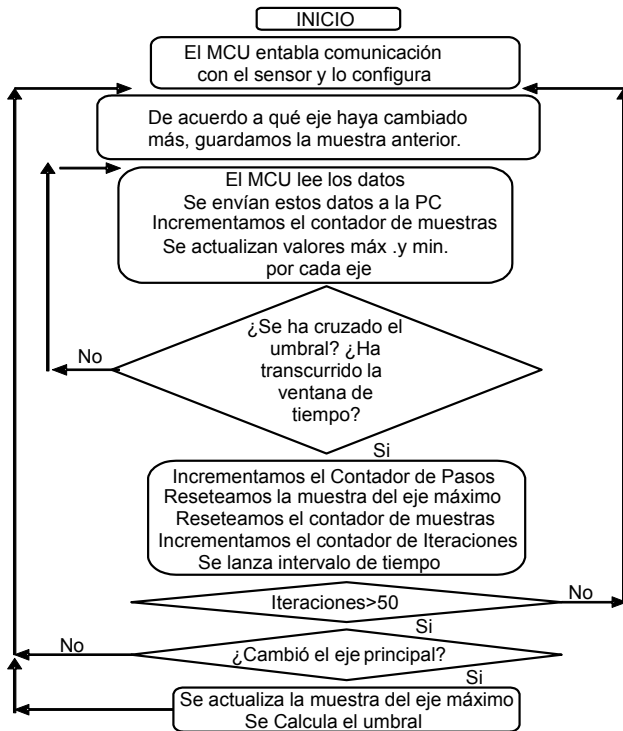


Fig. 3. Diagrama de flujo del algoritmo de detección de pasos.

Con lo expuesto anteriormente ya podemos definir qué tipo de hardware podemos utilizar para implementar el sistema. Disponibilidad de módulos de desarrollo en el mercado, capacidad de procesamiento necesaria para manejar el algoritmo, y prestaciones en bajo consumo, fueron los pilares de elección.

B. Acelerómetro.

Se decidió utilizar un acelerómetro triaxial y digital, eliminando restricciones de orientación y obtenemos mejor performance en bajo consumo respectivamente. De los módulos disponibles en el mercado nos inclinamos por utilizar el ADXL345 debido a las características [14] expuestas en la Tabla I.

TABLA I
COMPARACIÓN DE ACELERÓMETROS

Dispositivo Características	ADXL 345	MPU 6050	MMA 7455L	MMA 8451Q
I _{DD} Activo (µA)	45	140	400	14
I _{DD} Standby (µA)	0.1	NO	2.5	1.8
Interfaces	SPI/I2C	I2C	SPI/I2C	I2C
Tiempo Off-On (ms)	21.1	NO	20	41
Modo Bajo Consumo	SI	SI	NO	SI
Interrupciones	SI	SI	SI	SI
FIFO	SI	SI	NO	SI

I_{DD} medida a frecuencia de muestreo de 50Hz en modo de bajo consumo, excepto MPU6050 (40 Hz), MMA7455L frecuencia fija a 125 Hz

A pesar de que el MMA8451Q presenta un consumo en modo activo tres veces menor que el ADXL345, el consumo en standby y el tiempo necesario para la transición de este estado al de medición es mucho mayor, lo cual es relevante teniendo en cuenta que se espera un mayor tiempo en standby. Por otro lado la comunicación SPI es preferible en diseños de bajo consumo, debido a que podemos alcanzar frecuencias de varios MHz, contra los 400 kHz del I2C, el cual además presenta pérdidas de corriente ocasionadas por las resistencias pull-up.

C. Microcontrolador.

Varias familias de microcontroladores orientadas al bajo consumo fueron revisadas, todas ellas ofrecen kits de desarrollo disponibles en el mercado, el microcontrolador presente en dichos kits fue elegido para la comparación de la Tabla II. Cabe resaltar que aquellos de 32 bits son ARM Cortex M0. Básicamente el firmware implementado requiere tres estados para el MCU, modo activo para procesar los datos, un modo de bajo consumo del cual pueda salir sin requerir interrupciones externas (Timer interno), y por último un modo de bajo consumo en el cual sólo se retengan los datos de los registros y el estado de los pines (retención de RAM)

TABLA II
COMPARACIÓN DE MICROCONTROLADORES

Dispositivo Características	MSP430 G2553	PIC24F 16KA102	STM32 L100RC	EFM32 ZG222	MKL02Z 32VFM4
	Texas Ins.	Microchip	STM	S. Labs	Freescale
Bus de Datos	16 bits	16 bits	32 bits	32 bits	32 bits
Rango de Voltaje (V)	1.8 - 3.6	1.8 - 3.6	1.65 - 3.6	1.98 - 3.8	1.71 - 3.6
Número de E/S	16	24	51	37	28
I _{DD} Activo (mA)	2.3	3.05	1.93	1	2.8
I _{DD} Retención RAM (µA)	0.1	0.045	0.435	0.02	0.3
I _{DD} Timer Interno (µA)	0.63	0.8	1.5	0.9	86
Tiempo Off-On (µs)	1	1	58	163	95
Número de SPI's	2	1	3	1	1

V_{DD} = 3.3 V, F_{OSC} = 8 MHz, T_A = 25 °C

Nuestro Sistema requiere de dos interfaces SPI, una para el acelerómetro, y otra para el transceiver, por lo tanto el MSP430G2553, cumple con tal requerimiento sin sobredimensionar el sistema como lo haría el STML100RC, además presenta un mejor tiempo de transición desde un estado de bajo consumo a modo activo [15]. Por otro lado es recalable el hecho de que existe mucha mayor información disponible para MSP430, hojas de datos mejor detalladas, gran número de notas de aplicación y soporte por parte de Texas Instruments.

D. Transceiver

Diversas tecnologías para transmisión inalámbrica a bajo consumo fueron consultadas. Tecnologías que operan en la banda de los 2.4 GHz ofrecen hoy en día excelentes prestaciones, combinando bajo consumo y altas tasas de transferencia, tal es el caso del Bluetooth de baja energía (BLE por sus siglas en inglés). Sin embargo nuestro sistema no transmite grandes cantidades de datos, lo que se buscó fue bajo consumo, confiabilidad de enlace, precio y disponibilidad en el mercado. Los dispositivos consultados, así como características relevantes para el sistema se resumen en la Tabla III.

TABLA III
COMPARACIÓN DE TRANSCEIVERS

Dispositivo Características	CC1101	NRF 24L01	CC2540	CC2541
Fabricante	TI	Nordic	TI	TI
Tecnología	Sub 1 GHz	2.4 GHz	BLE	BLE
Frecuencia (MHz)	315/433/868/915	2400	2400	2400
Interfaz de Com.	SPI	SPI	SPI	SPI
I _{DD} Tx (mA)	16	11.3	27	18.2
I _{DD} Rx (mA)	17.1	12.3	22.1	20.2
I _{DD} Standby (µA)	0.2	0.9	0.4	0.5
T. de Calibración (ms)	2	5.3	2	2

V_{DD} = 3 V, T_A = 25 °C, I_{DD} Rx en alta ganancia, I_{DD} Tx a 0 dBm (CC1101 a 433 MHz 250 kbps)

Las frecuencias por debajo de 1 GHz atraviesan con mayor facilidad las paredes, además de poder alcanzar rangos hasta cuatro veces (a 433 MHz) el de una transmisión a 2.4 GHz utilizando la misma potencia. A ello podemos sumarle el hecho de que el espectro de los 2.4 GHz se encuentra saturado [16]. Es entonces por ello que el CC1101 fue elegido para nuestro sistema, prestaciones en bajo consumo [17] y menor precio comparado contra tecnología BLE primaron para la elección.

E. Fuente de Alimentación

El objetivo de mantener el sistema funcionando por más de 5 años sin sufrir cambio y/o recarga de batería alguno demanda además de un buen diseño, un buen criterio para escoger el tipo de batería a utilizar. Es aquí donde las baterías de Litio destacan del resto, su alta capacidad para almacenar energía, amplio rango de temperatura de operación, resistencia a golpes y vibraciones, sumado a su baja auto descarga [18], las hacen ideales para este tipo de aplicaciones. Sin embargo el Litio tan sólo representa el ánodo de la batería, el cátodo puede estar representado por una alta variedad de compuestos, dentro de los cuales destacan el Óxido de Manganeso (MnO₂) y el Cloruro de Tionilo (SOCl₂), con un voltaje de celda de 3 V y 3.6 V respectivamente. Para nuestro proyecto se optó por baterías de Litio - Óxido de Manganeso (Li-MnO₂), ello debido a que son más baratas, fáciles de conseguir y además no son tan contaminantes o tóxicas como las que contienen SOCl₂.

F. Manejo Dinámico de Periféricos y los Estados del Sistema

El manejo dinámico de periféricos consiste en encender cada uno de los componentes sólo cuando estos se requieran, i.e.

mientras estamos procesando las tramas de aceleración, tanto el acelerómetro como el transmisor deberían permanecer en modo standby. En la Fig. 4. puede observarse mejor este proceso:

- Al iniciar, el MCU configura tanto al acelerómetro como al transmisor, dejándolos en modo standby. Luego solicita al acelerómetro un set de muestras. MCU en Activo 1.
- Una vez enviada la solicitud de muestras, mientras el acelerómetro pasa a modo activo, coloca las muestras en el registro FIFO el MCU pasa a modo de bajo consumo 4 (LPM4).
- Una vez terminada la escritura en el FIFO, el ADXL345 genera una interrupción externa al MCU, avisando que las muestras están listas, con ello el MCU las lee, inmediatamente pone en standby al acelerómetro y pasa a procesar el set de muestras aplicándoles el algoritmo detector de pasos. MCU en Activo 2.
- Una vez procesadas las muestras, se espera hasta solicitar muestras de nuevo y empezar el ciclo nuevamente.
- Este ciclo se repite hasta completar el período de la ventana de tiempo, después de la cual se actualizan los registros de umbral y contador de pasos.
- Transcurrido mucho más tiempo (2 horas en algunos sistemas comerciales), se proceden a transmitir los datos recolectados.

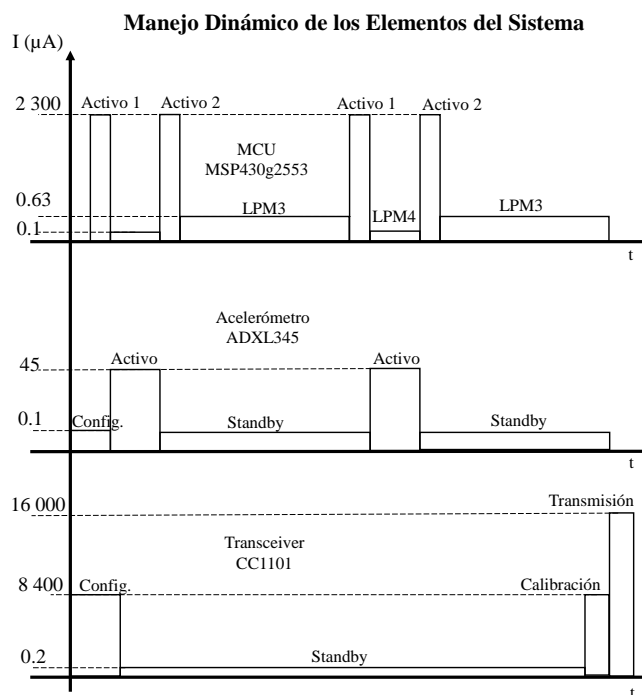


Fig. 4. Estados de cada uno de los elementos del sistema durante la ejecución del programa.

G. Programación en C Orientada al Bajo Consumo.

Para la realización del software, se utilizaron algunas técnicas de programación, las cuales nos ayudaron a optimizar nuestro código, con el objetivo de ahorrar hasta el último ciclo de reloj posible, se mencionan los más relevantes:

- Se utilizaron temporizaciones únicamente con

temporizadores y mientras se realizaba cada conteo, se colocó al MCU en el modo de más bajo consumo posible.

- Se utilizó polling únicamente para revisar los flags del módulo SPI que indican el término de una transmisión o recepción de trama, se pudo haber utilizado interrupción pero hubiera tomado más ciclos de reloj.
- Se usaron variables locales en lugar de globales según convenga, las variables locales se crean y almacenan en los registros de uso general del CPU, los cuales físicamente se encuentran muy cerca de éste, por lo tanto son más rápidos y eficientes comparados contra el acceso a ubicaciones dentro de la RAM o FLASH.
- Para variables locales que no presentan cambios en el tiempo fue conveniente utilizar el modificador static con el fin de que dicha variable se almacene en la memoria flash como parte de la función misma, minimizando la cantidad de código necesario para colocar y re-inicializar esta variable cada vez que se llame a la función.
- En cuanto al uso de funciones, se utilizó la instrucción inline y el pragma “#pragma FUNC_ALWAYS_INLINE (Nombre_de_Funcion)” con el fin de evitar las instrucciones de salto en cada llamada a función. Ahorrando ciclos de reloj al tener un programa con partes que se repiten pero sin instrucciones de salto.
- La ventana de tiempo se obtiene contabilizando cierta cantidad de muestras, aquí se utilizó una iteración for con cuenta descendente, de esa forma el valor actual de iteración se compara directamente con cero, ahorrando ciclos de reloj.

III. MEDICIONES Y RESULTADOS

A. Bajo Consumo

El prototipo desarrollado contiene pines entre la línea de alimentación de cada uno de los componentes, con lo cual podemos medir la corriente en cada uno de ellos o en todo el conjunto. Para medir el modo activo del microcontrolador fue necesario implementar un bucle infinito con la instrucción while(1), ello debido a que la frecuencia con la que el MCU cambia de estados es elevada y el multímetro no puede medirla directamente. Operación similar se realizó para medir la corriente en el acelerómetro y en el transmisor. Los resultados pueden observarse en la Tabla IV.

TABLA IV
CONSUMO POR DISPOSITIVO

Dispositivo	Modo Activo (µA)	Modos de Bajo consumo (µA)
Microcontrolador MSP430G2553	2580	LPM3: 1.2 LPM4: 1
Acelerómetro ADXL345	85	65
Transceiver CC1101	~16000	1

Mediciones realizadas utilizando un multímetro digital BK PRECISION 2890, VDD = 3 V, T_A=25°C

Como se observa en la Tabla IV. el consumo presentado por el acelerómetro difiere demasiado con la hoja de datos. Ello debido a que se utiliza un voltaje diferente al estipulado en la misma. Además el módulo adquirido tiene un regulador de voltaje integrado en el PCB, lo cual estaría provocando un offset de corriente.

B. Prototipo

Para tener una mejor visualización de los datos de aceleración, se desarrolló una pequeña aplicación utilizando LabView, en la cual podemos observar la aceleración en los tres ejes, además de los valores máximos y mínimos de aceleración en el eje con mayor variación, también se visualiza como el umbral dinámico va cambiando en el tiempo. Ello puede observarse en la Fig. 5. Dicho prototipo fue colocado en un ejemplar vacuno con el fin de obtener nuestro umbral de referencia. Según los datos presentados en la Fig. 1. el umbral de referencia podríamos definirlo en ±4 g, descartando los falsos pasos que tienen aceleraciones un poco menores.

Se buscó monitorear la aceleración que se presenta mientras el animal descansa, sin embargo este aplastó el prototipo, no se causaron daños, pero se interrumpió la transmisión de inmediato.

C. Software.

Luego de aplicar la instrucción inline para evitar los saltos entre las llamadas a funciones, y el uso de la instrucción static se registraron los cambios mostrados en la Tabla V.

TABLA V
VARIACIÓN EN USO DE MEMORIA RAM Y FLASH

Memoria	(%) Usado antes de Optimización	(%) Usado luego de Optimización
RAM	29	25
FLASH	11	12

Mediciones realizadas utilizando Code Composer Studio 6.1

IV. CONCLUSIONES

Se ha desarrollado satisfactoriamente un prototipo detector de pasos para ganado bovino, habiendo obtenido valiosos datos luego del experimento. Ello ha sido posible gracias a la interfaz gráfica desarrollada para observar nuestra señal y parámetros en tiempo real, de ello se concluye que el uso de dichas interfaces acelera el tiempo de desarrollo de un sistema de este tipo. Además se ha concluido que existen valiosas mejoras por introducir en cuanto a la detección de otros parámetros presentes en la actividad física del ganado vacuno.

Se ha conseguido un bajo consumo de corriente para cada uno de los elementos que conforman el sistema, el manejo dinámico de cada uno de los estados activos y de bajo consumo, nos ayuda a reducir significativamente el consumo total; pudiéndose aún introducir mayores mejoras de hardware. Sin embargo también es necesario contar con mejores equipos y técnicas de medición.

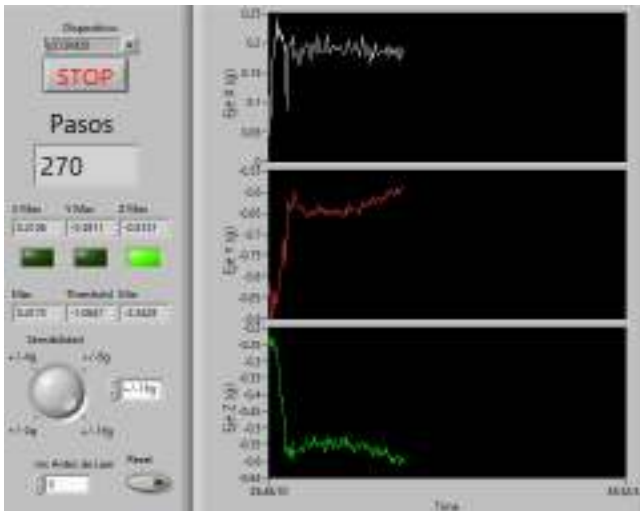


Fig.5. Aplicación desarrollada en LabVIEW con el fin de observar las señales y diversos parámetros en tiempo real.

En aplicaciones de bajo consumo, la reutilización de código por el uso de funciones puede resultar en el uso de ciclos de reloj innecesarios. Si se dispone de suficiente memoria de programa es deseable no tener saltos de código. Para dicho análisis es necesario conocer a profundidad el modo en que el compilador genera las instrucciones en lenguaje ensamblador para cada instrucción en lenguaje de alto nivel.

Debido a las bajas corrientes que se manejan, es necesario contar con mejores equipos de medición, un multímetro de precisión y un amplificador de instrumentación, este último de vital importancia, ya que podríamos observar la onda de corriente en un osciloscopio, con dicha onda podríamos calcular con mayor precisión los tiempos para cada uno de los estados, y con ellos calcular la capacidad de la batería necesaria para una autonomía de más de 5 años. Debido a que la batería representa buen porcentaje del precio del sistema, también podríamos aproximar un costo total.

Existen dos mejoras de hardware planificadas. La primera consiste en el uso de un regulador de voltaje lineal de baja caída de tensión LDO (Low Dropout Voltaje), específicamente uno de la familia TPS780xx [19], los cuales pueden ser ajustados a dos voltajes de salida, controlados según el estado lógico de uno de sus pines. Con lo cual podríamos aplicar la técnica del Escalamiento Dinámico de Voltaje, ya que por ejemplo, tendríamos la capacidad de alimentar con tan sólo 2.2 V a cualquiera de nuestros dispositivos cuando se encuentren en modo de bajo consumo y con ello disminuir la corriente de alimentación, luego al pasar a modo activo, elevar el voltaje de alimentación a 3 V o 3.3 V, tensión requerida para mayores frecuencias de trabajo. Por ende, teniendo en cuenta que nuestro sistema se encuentra la mayor parte del tiempo en estado de bajo consumo, podemos concluir que es viable una disminución adicional en el consumo de energía.

Como trabajo futuro se pretende captar la aceleración producida al montar otro animal, con ello podría obtenerse una plantilla y aplicar un algoritmo Pan-Tompkins. Durante el

experimento el ejemplar no se encontraba en celo, impidiendo dicha medición. Además es necesario diseñar un case con materiales de mejor resistencia mecánica, que sea impermeable y que no atenúe la señal transmitida.

REFERENCIAS

- [1] T. L. Powel. (2006, Apr 27). Pedometer Measurements of the Distance Walked by Grazing Sheep in Relation to Weather. Grass and Forage Science. [Online]. 23(1). Available: <http://onlinelibrary.wiley.com/doi/10.1111/j.1365-2494.1968.tb00558.x/citedby>
- [2] D. M. Anderson. (1977, Jul). Monitoring animal travel with digital pedometers. Journal of Range Management. [Online]. 30(4). Available: <https://journals.uaiz.arizona.edu/index.php/jrm/article/view/6736/6346>
- [3] Afimilk. (2012). Afimilk (SAE Afikim) - More than 30 years of advanced computerized solutions... Afimilk. Kibbutz, Israel. [Online]. Available: <http://www.afimilk.com/knowledge-center/articles/afimilk-sae-afikim-more-30-years-advanced-computerized-solutions>
- [4] Device, system and method for monitoring animal posture pattern, by Hagai Flexer. (2008, May 5). Patent US 8111166 B2 [Online]. Available: <http://www.google.com/patents/US8111166>
- [5] F. Terreno, "Sistema Detector de Celo," unpublished [Online]. Available: http://www.edutecne.utn.edu.ar/cyral_frmv/CyTAL_2006/Archivos/TF18%20Sistema%20de%20Detector%20de%20Celo.pdf
- [6] R. Murray, "Nuevo sistema 4-20," unpublished [Online]. Available: <http://rodolfomurray.com.ar/ARTICULOS%20DE%20INTERES/nueva%204-20.htm>
- [7] L. Brunassi. (2010, Oct). Improving detection of dairy cow estrus using fuzzy logic. Sci. agric. [Online]. 67(5). Available: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-90162010000500002
- [8] O. Segura, "Un mensaje de texto para saber cuando su vaca esté en celo," unpublished [Online]. Available: <http://www.contextoganadero.com/cronica/un-mensaje-de-texto-para-saber-cuando-su-vaca-este-en-celo>
- [9] A. Pérez. (2014, Jul). PROGENEX PERU SAC. Available e-mail: PROGENEXPERU@GMAIL.COM
Message: get PODÓMETROS – En busca de Información
- [10] C. Arana. (2006, Dic). Factores que afectan el intervalo parto-primer servicio y primer servicio-concepción en vacas lecheras del Valle del Mantaro durante la época lluviosa. Rev. investing. vet. Perú. [Online]. 17(2). Available: http://www.scielo.org.pe/scielo.php?pid=S1609-91172006000200004&script=sci_arttext
- [11] INEI. (2012). IV Censo Nacional Agropecuario 2012. INEI. Lima, Perú. [Online]. Available: <http://censos.inei.gob.pe/cenagro/tabulados/>
- [12] Perulactea. "Peruanos ya Consumen 80 Litros de Leche al Año, aunque FAO Recomendaba 130," unpublished [Online]. Available: <http://www.perulactea.com/2014/05/28/peruanos-ya-consumen-80-litros-de-leche-al-ano-aunque-fao-recomienda-130/>
- [13] Ministerio de Agricultura. (2009). Boletín Mensual de Leche. Ministerio de Agricultura. Lima, Perú. [Online]. Available: [http://www2.congreso.gob.pe/sicr/cendocbib/con3_uibd.nsf/8A16F586B87652D50525798000775A32/\\$FILE/410.pdf](http://www2.congreso.gob.pe/sicr/cendocbib/con3_uibd.nsf/8A16F586B87652D50525798000775A32/$FILE/410.pdf)
- [14] Analog Devices. "3-Axis,+2g/+4g/+8g/+16g Digital Accelerometer," ADXL345 datasheet, 2013.
- [15] Texas Instruments. "Mixed Signal Microcontroller," MSP430G2x53 MSP430G2x13 datasheet, Apr. 2011 [Revised May. 2013].
- [16] Infineo. "Make your Application Wireless," Sub 1 GHz RF Solutions brochure, Oct. 2014.
- [17] Texas Instruments. "Low-Power Sub-1 GHz RF Transceiver," CC1101 datasheet, Nov. 2013 [Revised Nov. 2013].
- [18] DURACELL. "Lithium/Manganese Dioxide," Technical Bulletin. [Online]. Available: http://ww2.duracell.com/media/en-US/pdf/gtcl/Technical_Bulletins/Lithium%20Technical%20Bulletin.pdf
- [19] B. Moyer, "Low-Power Design for Embedded Processors," Proc. IEEE, vol. 89, Nov. 2001, pp. 1576-1587.

Sistema de transferencia automática para grupos electrógenos

Turconi, Diego^{1,2}; Agüero, Jorge¹; Campero, Ignacio¹; Farina, Juan¹; Lupi, Daniel²; Zaradnik, Ignacio²
Cátedra de Proyecto Final¹, Laboratorio de Inteligencia Ambiental², Departamento de Ingeniería e Investigación Tecnológica
Universidad Nacional de la Matanza. Buenos Aires, Argentina.

diego.turconi@gmail.com

Abstract—Los avances que se vienen dando en las Tecnologías de la Información y Comunicación (TICs), entre las que podemos citar las Redes Eléctricas Inteligentes e Internet de las Cosas (IoT), así como en las energías renovables, como ser la fotovoltaica, la eólica, y la geotérmica, permiten administrar mejor los recursos energéticos, favorecer la protección del medio ambiente y responder a los requerimientos cada vez más exigentes de calidad de servicio y producto. Sin embargo todas estas nuevas tecnologías deben ser, en última instancia, respaldadas por sistemas clásicos como ser un grupo electrógeno, el cual permitiría continuar las actividades socio-productivas.

En el presente trabajo se pretende explicar el desarrollo de un sistema de transferencia automática para grupos electrógenos. Se detalla la selección del hardware, la implementación del firmware asociado, y las pruebas y ensayos.

Keywords- Electrificación rural, grupos electrógenos, microcontroladores .

I. INTRODUCCIÓN

La electrificación rural, muchas veces no rentable para las empresas de distribución de energía eléctrica, posee una gran importancia ya que permite la integración de los sectores rurales al desarrollo económico nacional, frena la migración rural-urbana y mejora el nivel socio cultural de los habitantes entre otras. Es por estos motivos que a nivel país existen políticas de electrificación rural en distintos países de Latinoamérica [1][2]. En particular en Argentina podemos nombrar el proyecto de electrificación rural para el desarrollo pecuario del departamento de 25 de mayo, en la provincia de San Juan [3], el cual está bajo el Programa de Servicios Agrícolas Provinciales (PROSAP), dependiente del ministerio de Agricultura, Ganadería y Pesca.

Dentro de estos programas está contemplado el uso de energías renovables, eólica, fotovoltaica, biomasa, etc., ejemplos del uso de energía eólica se pueden ver en el proceso de electrificación del espacio rural del Partido de Tandil, proceso llevado adelante por la Cooperativa Rural Eléctrica de Tandil-Azul Limitada (CRETAL) [4], así como ejemplos de energía fotovoltaica para la agricultura y el desarrollo sostenible se pueden ver en [5].

Toda fuente de energía renovable debe contar con un sistema de almacenamiento de energía, el cual le permita almacenar la energía generada y disponer de ella cuando sea necesaria. En función del dimensionamiento de dicho sistema se podrá contar, por un determinado tiempo, con energía aun cuando las condiciones ambientales no sean las favorables para la generación. Pero como las condiciones ambientales están fuera de nuestro alcance es importante contar con un sistema de respaldo, en este caso un grupo electrógeno.

En muchas aplicaciones productivas, donde el personal no tiene acceso permanente a las instalaciones, y se necesita un suministro de energía constante, es necesario contar con un sistema de transferencia automática. El mismo, permite el constante monitoreo de los parámetros de la fuente de suministro, y se encarga de hacer de forma automática la transferencia de energía a la carga, ya sea de la línea al grupo electrógeno, en caso de corte de luz o disminución de la energía almacenada, o del generador a la línea, en caso contrario.

II. ESTUDIO DE MERCADO

Luego de realizada una búsqueda de este tipo de sistemas en el mercado, nos encontramos con productos de empresas multinacionales como el Sistema de Control Modular EPIC de Caterpillar [13], los automatismos BA y UA de Schneider Electric [14] y el sistema SENTRON ATC5300 de Siemens [15]. Empresas más pequeñas ofrecen una gran variedad de implementaciones como el sistema AUT-MP12 de la empresa Electra Molins [16] que se basa en un módulo programable con tres microprocesadores, el sistema de transferencia automática con contactores de la empresa Velázquez Ingenieros Asociados LTDA basado en el sistema de transferencia automática ITACMC [17] y el sistema de la empresa Control Para la Industria basado en PLC Omron de la familia CJ1[18]. También se han encontrado gran variedad de trabajos realizados en esta temática [19][20][21], en cuyos casos el sistema implementado se basa en un controlador de lógica programable (PLC).

Los sistemas presentados pueden ser separados en dos grandes grupos: los sistemas dedicados y los sistemas basados en PLC. Los primeros suelen estar pensados para cubrir las demandas estándar de la aplicación, siendo poco flexibles a adaptarse a casos particulares del usuario. Los sistemas basados en PLC, si bien son más flexibles, esta flexibilidad

suele estar asociada a la incorporación de módulos extra, lo que genera un aumento de los costos finales del sistema.

Debido a estos motivos, se planteó el desarrollo de un sistema de transferencia automática para grupos electrógenos de bajo costo y flexible.

III. DESCRIPCIÓN GENERAL

El trabajo que se presenta es parte de un sistema más complejo que involucra la medición de parámetros de consumo eléctrico, de generación de energías renovables y climáticos, su transmisión a un nodo central, su concentración y posterior transmisión a un servidor para su análisis.

El sistema toma muestras de cada una de las fases (R S y T) tanto de línea como del grupo electrógeno para controlar los valores de tensión y frecuencia con la finalidad de mantener los valores dentro de los parámetros deseados. Si alguno de los valores de la fuente de energía aplicada a la carga no está dentro los parámetros preestablecidos conmuta automáticamente a la otra fuente. Los valores muestreados son a la vez transmitidos a una estación colectora [6] para su retransmisión y análisis. La figura 1 muestra el diagrama en bloques del sistema, mientras que la figura 2 muestra el diagrama de conexión del sistema a la línea eléctrica.

El sistema de transferencia automática cuenta con los siguientes módulos:

- Medidor de parámetros de suministro de línea: con el cual se monitorea la tensión de línea y las corrientes de la carga.
- Medidor de parámetros de suministro de grupo: con el cual se monitorea la tensión de grupo y las corrientes de la carga.
- LCD: es el módulo donde se visualizan los datos.
- Módulo Serie/USB: es el encargado de la comunicación Serie/USB.
- Módulo GSM/GPRS: es el encargado de hacer la comunicación GSM y GPRS con el usuario.
- Relé contactores: es el módulo que hace la habilitación de cada contactor.
- Relé solenoide, arranque, alarma: es el que habilita el paso de combustible, hace el arranque y hace sonar una alarma externa en caso de haber algún error.

A. Hardware

Como base de nuestro desarrollo se tomó un microcontrolador basado en el núcleo Cortex-M4 de ARM. El producto seleccionado fue el STM32F407ZET6 de ST [7]. La elección se fundamentó en:

- La disponibilidad de conversores analógicos digitales para la medición de parámetros.

- Gran numero de interfaces seriales (I2C, UART, SPI), que posibilita la comunicación con sensores y distintos módulos OEM.
- Interfaz Ethernet
- Capacidad de procesamiento adecuada para trabajar con una pila TCP/IP.

Otro factor que influyó en la selección de este microcontrolador, es la posibilidad de que el sistema pueda actuar como estación colectora y si bien el desarrollo actual de la estación colectora [6] se basó en el ATSAM3X y se había planteado una evolución natural al ATSAM4E, dicha opción se descartó por la disponibilidad y costo de los microcontroladores y los kits de evaluación en el mercado local. Siendo el STM32F4xx y el ATSAM4E de similar desempeño se consideró adecuado el cambio.

Los bloques medidores de parámetros de suministro de línea y grupo, encargados de monitorizar las tensiones y corrientes de los mismos, están formado por el integrado ATM90E36A [8] de la firma ATMEL. Dichos dispositivos se encuentran en la parte de alta tensión del sistema. Para aislar los mismos del microcontrolador se utilizó los integrados de aislación digital de Silabs, Si8641 [9], que poseen aislación de 5kV y una velocidad de transferencia máxima de 150Mbps. La medición de corriente se realiza a través de un transformador de corrientes, AC1020 de la firma Talema [10].

En la figura 3 se puede ver una parte del circuito impreso del sistema. En el mismo se pueden apreciar tres bloques, dos de ellos correspondientes a la parte de alta tensión, en los cuales se encuentran los ATM90E36A y el tercero al de baja tensión, en el cual se encuentra el microcontrolador. Entre ambos bloques de pueden apreciar la existencia de circuitos integrados, los cuales son los aisladores Si8641.

B. Entorno de desarrollo integrado(IDE)

Para la realización del proyecto se utilizo el IDE de IAR Workbench ARM versión 7.3

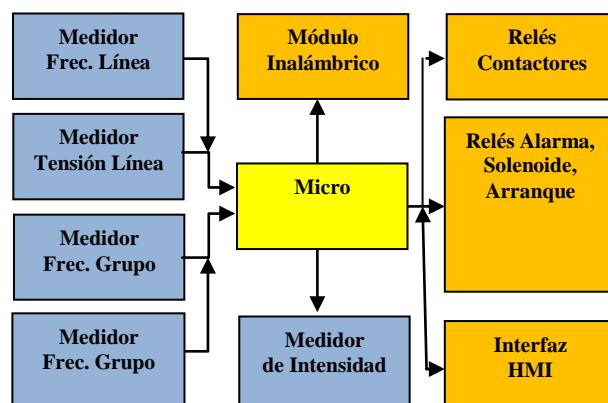


Figura 1. Diagrama en bloques del sistema.

IV. FIRMWARE

El firmware del microcontrolador del sistema se encuentra estructurado de la siguiente forma: inicialización de periféricos, configuración de los dispositivos de medición (ATM90E36A) y un ciclo infinito en el cual se verifica los valores eléctricos y actúa en función de ellos. La acción a realizar depende del estado del grupo electrógeno (Detenido, Arranque, Marcha), esta se representa en el diagrama de flujo en el bloque “Alarmas”, figura 4. A continuación se presentan los valores medidos en el display de la interfaz hombre máquina (HMI), se verifica la recepción de algún comando de configuración y se envían los datos por el puerto serie para ser presentados en el software de PC.

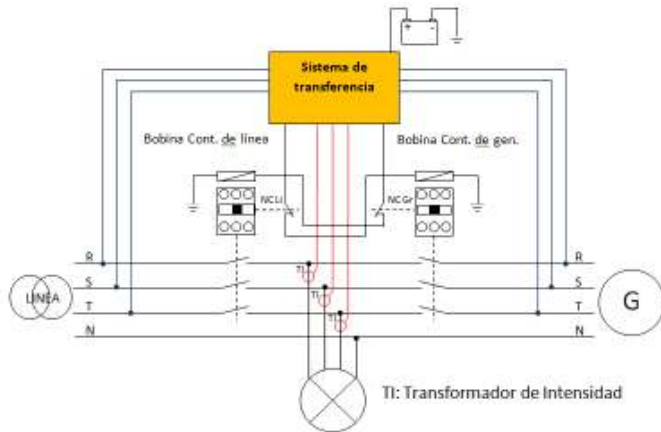


Figura 2. Diagrama de conexión del sistema a la línea eléctrica.

La principal virtud del firmware implementado radica en la rápida y transparente configuración de los parámetros del dispositivo de medición. El ATM90E36A cuenta con más de 100 registros para la lectura de los valores y la configuración de todas las funcionalidades del mismo [8].

La figura 5 presenta una pantalla del software desarrollado, a través de la cual se puede configurar la ganancia y el offset para cada una de las fases y el neutro conectados al ATM90E36A, en este caso particular los asociados a la medición del grupo electrógeno. Una vez configurados los valores y presionado “enviar”, los mismos son almacenados en los registros correspondientes del dispositivo de medición.

A continuación se detalla el código de inicialización de los dispositivos de medición:

// Reset chip Generator /Line

```
Atm90e36a_SetDataG(0x0000,0x789A);
Atm90e36a_SetDataL(0x0000,0x789A);
Delay_us(120000);
```

// Lee los parametros de los chips medidores

```
Atm90e36a_Read_ParamG();
Atm90e36a_Read_ParamL();
```

// Configura los chips medidores

```
Atm90e36a_Conf_updateG();
Atm90e36a_Conf_updateL();
```

//Setea los parametros de los chips medidores

// Calc Checksum 2

```
Atm90e36a_SetDataG(0x0050,0x5678);
ATM90E36A_GEN.HarmStart=Atm90e36a_GetDataG(0x0050);
ATM90E36A_GEN.CS2=atm90e36a_checksum(&ATM90E36A_GEN.POffsetAF,6);
Atm90e36a_SetDataG(0x0057,ATM90E36A_GEN.CS2);
ATM90E36A_GEN.CS2=Atm90e36a_GetDataG(0x0057);
ATM90E36A_GEN.SysStatus0=Atm90e36a_GetDataG(0x0001);
Atm90e36a_SetDataG(0x0050,0x8765);
```

// Calc Checksum 3

```
Atm90e36a_SetDataG(0x0060,0x5678);
ATM90E36A_GEN.AdjStart=Atm90e36a_GetDataG(0x0060);
ATM90E36A_GEN.UgainA=0x8B2F;
Atm90e36a_SetDataG(0x0061,ATM90E36A_GEN.UgainA);
ATM90E36A_GEN.UgainB=0x8B2F;
Atm90e36a_SetDataG(0x0065,ATM90E36A_GEN.UgainB);
ATM90E36A_GEN.UgainC=0x8B2F;
Atm90e36a_SetDataG(0x0069,ATM90E36A_GEN.UgainC);
ATM90E36A_GEN.IgainA=0x1AF5;
Atm90e36a_SetDataG(0x0062,ATM90E36A_GEN.IgainA);
ATM90E36A_GEN.IgainB=0x1AF5;
Atm90e36a_SetDataG(0x0066,ATM90E36A_GEN.IgainB);
ATM90E36A_GEN.IgainC=0x1AF5;
Atm90e36a_SetDataG(0x006A,ATM90E36A_GEN.IgainC);
ATM90E36A_GEN.IgainN=0x1AF5;
Atm90e36a_SetDataG(0x006D,ATM90E36A_GEN.IgainN);
ATM90E36A_GEN.CS3=atm90e36a_checksum(&ATM90E36A_GEN.UgainA,14);
Atm90e36a_SetDataG(0x006F,ATM90E36A_GEN.CS3);
ATM90E36A_GEN.CS3=Atm90e36a_GetDataG(0x006F);
ATM90E36A_GEN.SysStatus0=Atm90e36a_GetDataG(0x0001);
Atm90e36a_SetDataG(0x0060,0x8765);
```

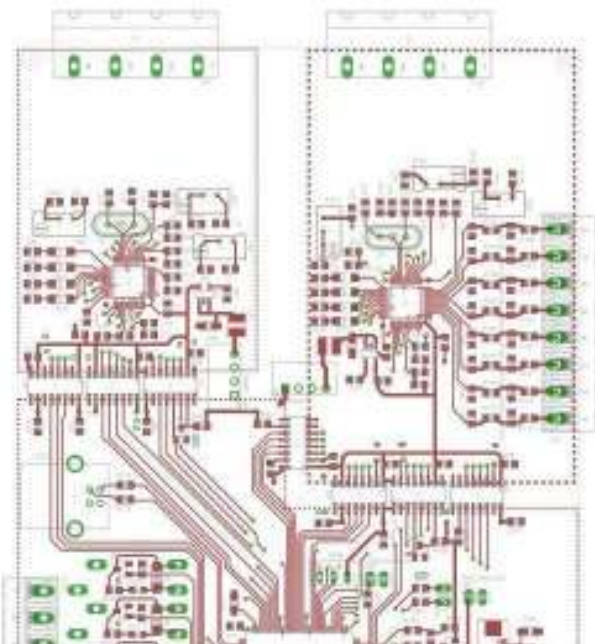


Figura 3. Imagen parcial del circuito impreso.

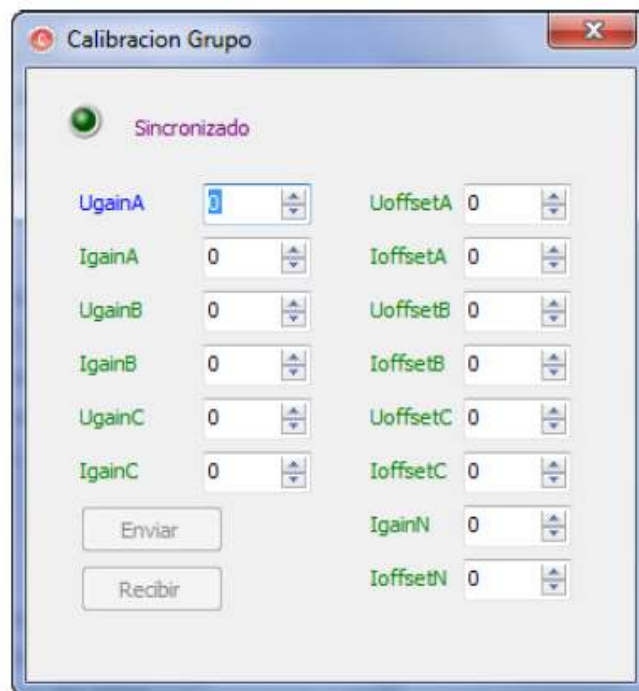


Figura 4. Pantalla de calibración.

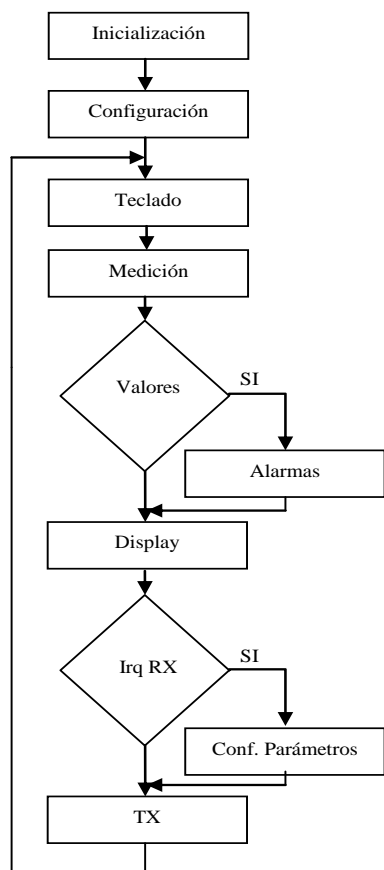


Figura 4. Diagrama de flujo del firmware.

V. ENSAYOS

Finalizados los chequeos de hardware, firmware y software el sistema debe ser calibrado para la correcta medición de tensiones y corrientes. Esto se puede realizar por medio del software de PC, previamente nombrado, y/o de unos potenciómetros en la placa.

La calibración, tanto para la corriente como para la tensión, se debe realizar de a una línea por vez, es decir que si se quiere calibrar la medición de tensión de la fase A en el grupo se debe cortocircuitar la fase B y C. La misma se debe llevar a cabo inyectando una tensión conocida y comparándola con la medición que entrega el equipo en la pantalla de la PC. Desde la pantalla se puede modificar tanto la ganancia como el offset para lograr que ambas mediciones coincidan en su valor. Si la calibración no se lograra mediante el software de PC, el cual modifica los parámetros de los chips medidores, se puede ajustar la ganancia vía los potenciómetros ubicados en la placa para tal fin.

Se realizaron mediciones de tensión y corriente luego de la calibración de los dispositivos, los valores obtenidos se pueden ver en la tabla N°1.

Para la calibración y las posteriores mediciones se utilizaron la pinza UT208 y el multímetro digital UT71A ambos de la firma UNI-T [11][12].

Se observó que los dispositivos son capaces de medir con muy buena exactitud los valores de entrada y que sobrepasaron las expectativas planteadas en el comienzo del proyecto, ya que no es necesaria tanta exactitud en nuestra

aplicación planteada. La figura 5 muestra una imagen del sistema.

Tabla N°1. Comparación de valores reales y medios.

Tensión Entregada [V]	Tensión Medida [V]
223,67	223,45
219,49	219,32
113,82	113,67
110,49	110,26
Corriente Consumida[A]	Corriente Medida[A]
3,678	3,669
7,902	7,89
10,615	10,601

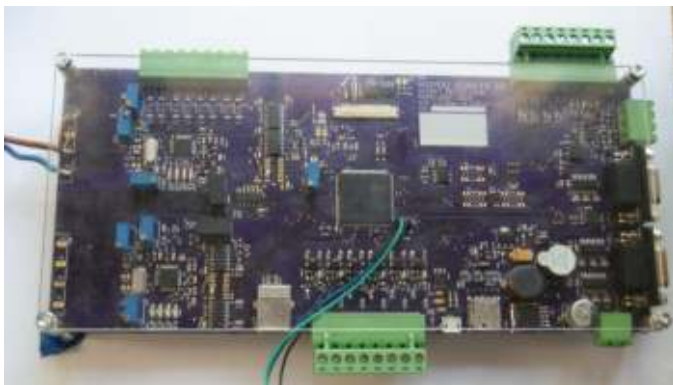


Figura 5. Imagen del sistema.

VI. CONCLUSIONES

Se ha logrado el desarrollo de un sistema capaz de medir con exactitud los valores eléctricos de la línea y los generados por un grupo electrógeno, y de realizar la conmutación entre ambos sistemas a fin de mantener el suministro de energía sobre la carga.

Para los prototipos desarrollados (dos) se realizó un gasto de aproximadamente U\$S 200 por unidad, costo el cual se estima puede llegar a menos de la mitad en producción. Siendo dicho costo un valor competitivo.

La exactitud de valores medidos se ha logrado gracias a la amigable interfaz de PC, que permite fácilmente configurar los parámetros del dispositivo de medición.

La eficiencia de esta conmutación, es decir la estabilidad de la energía entregada a la carga, el tiempo de respuesta, la suavidad de la transición, no se han llegado a probar, quedando dichos ensayos como uno de los próximos pasos a realizar.

El sistema desarrollado posee un gran número de prestaciones que no se encuentran implementadas, como ser la comunicación Ethernet, la medición de parámetros asociados a la calidad de la señal eléctrica, el uso de un display inteligente como interfaz local. Se tiene como objetivo la pronta implementación de dichas características, lo que permitirá al sistema poder trabajar en redes eléctricas inteligentes.

REFERENCIAS

- [1] http://dger.minem.gob.pe/ArchivosDger/PNER_2015-2024/F1-PNER-2015-2024.pdf, última visita 01/05/2015.
- [2] <http://www.subdere.gov.cl/documentacion/programa-de-electrificaci%C3%B3n-rural-contrato-de-pr%C3%A9stamo-1475oc-ch>, última visita 01/05/2015.
- [3] <http://www.prosap.gov.ar/docs/SJuan-Electrificacion25DeMayo-PAA.pdf>, última visita 01/05/2015.
- [4] Guillermina P. Jacinto, María Luciana Nogar, “Electrificación rural, desarrollo territorial y pequeñas localidades. El caso de Tandil (provincia de Buenos Aires, Argentina)” http://www.filo.unt.edu.ar/rev/ieg/ieg_21/Breves%20201_electrificacion_jacinto_nogar.pdf, última visita 01/05/2015.
- [5] B. van Campen, D. Guidi y G. Best, “Energía solar fotovoltaica para la agricultura y desarrollo rural sostenibles”, Documento de Trabajo sobre Medio Ambiente y Recursos Naturales, No.3 FAO, Roma, 2000, <http://www.fao.org/uploads/media/Solar%20photovoltaic%20for%20SARD%20ES.pdf>, última visita 01/05/2015.
- [6] Canziani, Mónica; Gomez, Rodrigo; Lupi, Daniel; Nassipián, Verónica; Slawiski Javier; Turconi, Diego; Zaradnik, Ignacio, “Plataforma de conexión de Redes Eléctricas Inteligentes a Internet de las Cosas”, CASE 2014, Agosto 2014.
- [7] ST, “STM32F405xx STM32F407xx, ARM Cortex-M4 32b MCU+FPU, 210DMIPS, up to 1MB Flash/192+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 15 comm. interfaces & camera”, <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/DM00037051.pdf>, última visita 01/05/2015.
- [8] ATMEL, “Atmel M90E36A, Enhanced Poly-Phase High-Performance Wide-Span Energy Metering IC”, <http://www.atmel.com/Images/Atmel-46004-SE-M90E36A-Datasheet.pdf>, última visita 01/05/2015.
- [9] Silabs, “Si8640/41/42/45, LOW-POWER QUAD-CHANNEL DIGITAL ISOLATOR”, <https://www.silabs.com/Support%20Documents/TechnicalDocs/Si864x.pdf>, última visita 01/05/2015.
- [10] Talema, “AC1020 - 20 Amp Current Transformer” <http://www.nuvotem.com/en/products/pdf/AC-1020%20Jun-06.pdf>, última visita 01/05/2015.
- [11] <http://uni-trend.com/UT208.html>, última visita 01/05/2015.
- [12] <http://uni-trend.com/UT71A.html>, última visita 01/05/2015.
- [13] <http://latinamerica.cat.com/cda/files/2061850/9/LSXE0007-00.pdf>, última visita 01/05/2015.
- [14] <http://www.schneider-electric.cl/documents/local/catalogos/de/cap4.pdf>, última visita 01/05/2015.
- [15] http://www.editores-srl.com.ar/revistas/ie/255/sistema_de_transferencia_automatico, última visita 01/05/2015.
- [16] <http://www.electramolins.es/AUT-MP12.aspx?MenuSup=7>, última visita 01/05/2015.
- [17] http://www.velasquez.com.co/catalogo/transferencia_automatica_con_contactores.pdf, última visita 01/05/2015.
- [18] <http://cpi.com.ar/soluciones/sistema-de-transferencia-automatica/>, última visita 01/05/2015.
- [19] Montatixe Almachi, Walter Patricio; Pillajo Guano, Aníbal Geovanny “Construcción de un tablero de transferencia automática de energía eléctrica para la central telefónica de Echandía de Andinatel S.A.” <http://bibdigital.epn.edu.ec/handle/15000/2082>, última visita 01/05/2015.
- [20] Carabajal Gutierrez José Gualberto, “Control automático de transferencia de energía eléctrica” <http://es.slideshare.net/PedroChavez1/control-automatico-de-transferencia-de-energia-elctrica>, última visita 01/05/2015.
- [21] Hans Eslava Maldonado, Nicolás Franco Franco, “Automatización de una planta de emergencia para cargas no mayores a 10kw”, <http://itzamna.bnct.ipn.mx/dspace/bitstream/123456789/10105/1/6.pdf>, última visita 01/05/2015.

Generador de claves automáticas utilizando Smartphone

Prototipo para Agencias Bancarias

Oscar Fernando Gaidos

Professor Assistente, Engenharia Mecânica
Centro Universitário do Distrito Federal – UDF
Brasília, Brasil
ogaidos@gmail.com

Felipe da Silva Maeda

Escola de Engenharia, Engenharia Mecânica
Centro Universitário do Distrito Federal – UDF
Brasília, Brasil
maeda.felipe@gmail.com

Este trabajo presenta el desarrollo de un sistema digital para generar claves automáticas utilizando un microcontrolador PIC y un aplicativo para Smartphone con S.O. Android. Este prototipo podrá reemplazar las tradicionales claves impresas en papel por claves generadas en un aplicativo de celular. Para esto fue construido un sistema embebido utilizando el microcontrolador PIC16F877A y un módulo Bluetooth HC06 para comunicación inalámbrica con Smartphone en el cual existen dos aplicativos para Android, los cuales fueron desarrollados por medio de la herramienta App Inventor.. En la figura 1 es posible observar el prototipo hardware del generador de claves automáticas.



Fig. 1. Placa del generador de claves automáticas

Después del montaje de la placa del generador de claves, fue desarrollado en el compilador MikroC el firmware del microcontrolador, donde fueron elaboradas rutinas de monitorización, escrita en la pantalla LCD, además del envío y recepción de datos por medio del módulo serial del microcontrolador (USART), el cual enviara estos datos directamente al módulo Bluetooth y este los enviara al Smartphone.

El programa del microcontrolador incluye también una rutina responsable por realizar el conteo de las claves separándolas según el criterio de clave “normal” o clave “preferencial” esta última utilizada para personas mayores de edad, mujeres embarazadas, personas con deficiencia, personas con bebés y otros casos específicos. Además se tiene una rutina responsable por la configuración inicial del sistema donde son definidos los parámetros de clave inicial, clave final, código para la clave “normal” y código para la clave “preferencial”.

Después de la creación de las rutinas básicas del microcontrolador, fue creado el aplicativo del celular por medio de la herramienta App Inventor donde fueron creados dos aplicativos: Uno destinado a los clientes (aplicativo de usuario) y otro destinado a los funcionarios (aplicativo para la parte administrativa) del establecimiento que utiliza el sistema. En la figura 2 se muestra la pantalla principal de los aplicativos.

El aplicativo de usuario es responsable por solicitar una clave (normal o preferencial) al módulo generador de claves y monitorizar el progreso de la fila de espera avisando al usuario cuando llega su turno de ser atendido.

El aplicativo del administrador posee una función de enviar al módulo generador de claves la información de que existe una taquilla libre para ser atendido y está listo para realizar un nuevo atendimento. Además de permitir la reinicialización remota del sistema a partir de la clave inicial.

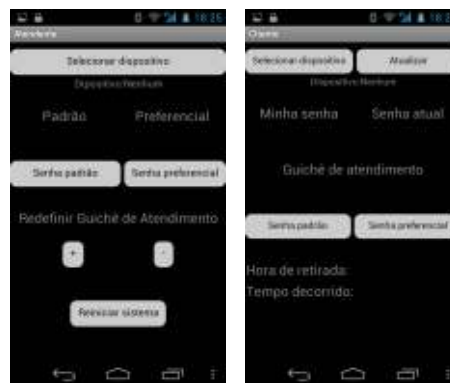


Fig. 2. Aplicativo de Administrador y usuário

El prototipo del generador de claves automáticas presenta potencial para el desarrollo de un dispositivo más robusto y posiblemente comercial. Una opción de mejorar el desempeño del prototipo, en futuros proyectos, estaría en la implementación de la tecnología WIFI en el sistema, aumentando así el alcance del módulo generador de claves.

Báscula Inclínada usando Mbed

Aplicación en una cosechadora de caña de azúcar

Raul Emilio Melo Sevilla*
Universidad Santiago de Cali- Cali, Colombia
raulmelo@usc.edu.co

El cultivo de la caña de azúcar ocupa un renglón importante en la economía del Valle del Cauca (Colombia); en otros países como Brasil se considera como uno de los productos más competitivos, representando un 8% de la agricultura [1].

La agricultura de precisión es una técnica moderna donde se utilizan una gran cantidad de datos [2], en ella se usan muchos sensores para llevarlos a un sistema de información, este permite almacenar [3], procesar y gestionar la información de una manera rápida y precisa.

Los sistemas de información usados, están conformados por varios componentes [3]: el de sensado y adquisición de datos, el de posicionamiento global (GPS), el de control de motores, válvulas, etc. [4] y finalmente el de visualización.

En este trabajo se pretende mostrar el desarrollo de una parte del sistema de información, un circuito electrónico basado en los microcontroladores MBED (LPC1768) que permite capturar la información de una báscula inclinada en una cosechadora de caña de azúcar, ver figura 1, para determinar la productividad en toneladas de caña de azúcar por hectárea (TCH).



Figura 1. Cabezote cosechadora de caña de azúcar

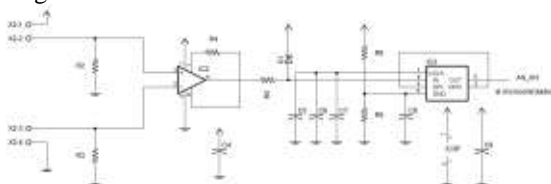


Figura 2. Diagrama esquemático de una entrada analógica

En la figura 2, se muestra el diagrama esquemático para una de las entradas analógicas y la implementación de un filtro pasa bajas usando el MAX291, la celda de carga es de 100Kg, Teda Huntleigh Modelo 1022, en la figura 3, se muestra el diseño esquemático del módulo MBED, con la comunicación I2C del inclinómetro AXDL345, la entrada digital, que usa un sensor inductivo marca Autonics referencia PR18-8DP, así

como la salida de comunicación CAN (Controlled Area Network) para la comunicación con un computador y su posterior visualización de datos. La Figura 4 muestra el circuito impreso desarrollado para el sistema.

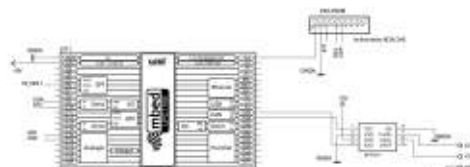


Figura 3. Diagrama esquemático del módulo MBED

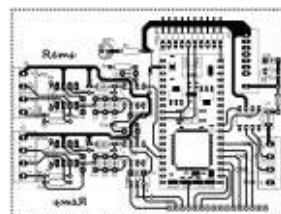


Figura 4. Circuito impreso desarrollado.

Referencias

- [1] P. S. G. Magalhães and D. G. P. Cerri, "Yield Monitoring of Sugar Cane," *Biosyst. Eng.*, vol. 96, no. 1, pp. 1–6, Jan. 2007.
- [2] A. Camilli, C. E. Cugnasca, A. M. Saraiva, A. R. Hirakawa, and P. L. P. Corrêa, "From wireless sensors to field mapping: Anatomy of an application for precision agriculture," *Comput. Electron. Agric.*, vol. 58, no. 1, pp. 25–36, Aug. 2007.
- [3] R. Nikkilä, I. Seilonen, and K. Koskinen, "Software architecture for farm management information systems in precision agriculture," *Comput. Electron. Agric.*, vol. 70, pp. 328–336, 2010.
- [4] J. V. Stafford, "Implementing Precision Agriculture in the 21st Century," *J. Agric. Eng. Res.*, vol. 76, no. 3, pp. 267–275, Jul. 2000.

* Raúl Emilio Melo Sevilla, Ingeniero Electricista, Especialista en redes de comunicación, Profesor tiempo completo de la Facultad de ingenierías de la Universidad Santiago de Cali, Cali - Colombia. Líder del grupo de investigación en instrumentación Electrónica Industrial y Ambiental (GIEIAM) categorizado en C por Colciencias.

Unidad inercial de bajo costo asistida por GPS

Arístide Silvestris, Mauricio Principi, Damián Primo
Grupo Sistemas de Tiempo Real (GSTR)
Facultad de Ingeniería – UNRC
Río Cuarto, Argentina
asilvestris@ing.unrc.edu.ar

Ariel Principi, Diego Fusari, Diego Badino
Centro I+D Tecnologías Aeronáuticas (CITeA)
Fuerza Aérea Argentina - Área Material Río Cuarto
Las Higueras, Argentina

Se plantea un sistema destinado a aplicaciones aeronáuticas que brinda la posibilidad de mejorar la calidad de los datos entregados por las unidades de medición inercial (IMU), gracias a la utilización de una serie de elementos comerciales off the shelf (COTS). Estos se encuentran desarrollados para aplicaciones industriales de bajo costo, como lo son los micro-controladores, pudiendo ser adaptados fácilmente a los requerimientos aeronáuticos.

En la actualidad los sistemas de medición inercial han sido reemplazados en gran medida por IMU's basadas en la tecnología Micro-electro mechanical system (MEMS), sin partes móviles, de costos relativamente bajos y buenas prestaciones, si bien la exactitud de estos sistemas no posee la calidad de los costosos sistemas inerciales lásericos, estos pueden compensar sus falencias mediante la integración de varios tipos de sensores tales como barómetro, tubo pitot y sistemas de posicionamiento global (GPS).

La presente publicación pretende contrastar el comportamiento de un equipo certificado para aplicaciones aeronáuticas contra un diseño modular propio.

En lo referente al diseño propio se ha trabajado en la implementación mediante una placa prototipo para desarrollo de la firma Texas Instruments que cuenta con un micro-controlador TM4C123G y una serie de puertos adicionales que facilitan la conectividad con periféricos externos.

Para la codificación de software se ha trabajado bajo un entorno de desarrollo propietario que cuenta con las herramientas necesarias para programación y depuración de cada uno de los registros.

Con respecto al sensor inercial, se ha trabajado con el MPU9150 que contiene sensores de movimiento en 9 ejes contando con 3 acelerómetros, 3 giróscopos y 3 magnetómetros, se posee además un barómetro BMP180 y varios sensores de variables físicas como pueden ser humedad, luz y temperatura.

El módulo de GPS fue seleccionado para que pueda efectuarse una interfaz compatible con el sistema de medición inercial, trabajando en conjunto con un filtro implementado por software ejecutado por el micro-controlador de manera que compense el típico error de deriva que poseen los sistemas inerciales.

De esta manera mediante la implementación del citado filtro, la posición correcta puede actualizarse una vez por segundo desde el GPS en recorridos rectos y nivelados, mientras que en maniobras que involucren rolidos o giros la información más confiable es puramente proveniente de la IMU.

El destino de dicho módulo es el de ser incorporado en aplicaciones aeronáuticas ya sea como parte constitutiva en el desarrollo de un prototipo autónomo y de bajo costo de EFIS (Electronic Flight Instrument System) en donde se visualice la orientación angular de la aeronave que lo use o como EGI (Embedded GPS in Inertial) en UAS (unmanned aerial vehicle).

REFERENCIAS

- [1] PERFORMANCE OF AUTOMOTIVE-GRADE MEMS SENSORS IN LOW COST AHRS FOR GENERAL AVIATION, Lance Sherry, Chris Brown, Ben Motazed, Dave Vos, Athena Technologies Inc., Fairfax, Virginia.
- [2] DEVELOPMENT OF A LOW-COST INTEGRATED GPS/IMU SYSTEM. Xiufeng He and Yongqi Chen The Hong Kong Polytechnic University and Jianye Liu Nanjing University of Aeronautics and Astronautics.
- [3] GPS FAULT DETECTION WITH IMU AND AIRCRAFT DYNAMICS. T. S. Bruggemann, D. G. Greer, R. A. Walker, Member, IEEE Queensland University of Technology
- [4] GPS/IMU DATA FUSION USING MULTISENSOR KALMAN FILTERING: INTRODUCTION OF CONTEXTUAL ASPECTS. François Caron a, Emmanuel Duos A), Denis Pomorski B), Philippe Vanheegehe A) .A) LAGIS UMR 8146 Ecole Centrale de Lille Cité Scientifique BP 48 F59651 Villeneuve d'Ascq Cedex, France. B) LAGIS UMR 8146 - Bat. P2 Université Lille I - F59655 Villeneuve d'Ascq Cedex, France
- [5] FUSION FILTER ALGORITHM ENHANCEMENTS FOR A MEMS GPS/IMU. Jose A. Rios, Crossbow Technology, Inc. Elecia White, Crossbow Technology, Inc.

Geiger-Müller counter implementation with an 8-bit microcontroller

Husain, Ignacio Santiago
School of Engineering
University of Buenos Aires, AR
ihusain@fi.uba.ar

García-Inza, Mariano
DevicePhysics-Microelectronics Lab
Engineering School – UBA, AR
mgarciaianza@fi.uba.ar

Lipovetzky, José
Low Temperature Div.
Bariilche Atomic Center – CNEA, AR
lipo@ib.edu.ar

This work presents a simple and low cost system for measuring ionizing radiation by means of a Geiger-Müller (GM) tube. The main objective was to build and test a prototype based on an 8-bit Atmel's® AT89S8253 microcontroller. The validation was made comparing counts per second (CPS) measurements against the ones obtained by a homologated commercial GM counter, using a partially shielded ⁹⁰Sr radiation source. We tested the system with two different tube types, evaluating their response as a function of distance to the source and biasing voltage. Results show that the prototype is suitable for low dose detection applications as personnel dosimetry, and that the response to different fluxes proves that is comparable to the commercial detector.

The embedded system has an LCD for displaying information to the user, an UART interface for serial data communication, a keypad for configuration, an alarm that sounds when there is an excess of measured radiation in the environment for personnel protection (threshold configurable through keypad), and a GM-tube interface for its biasing using a charge-and-pump circuit. The system basically measures the amount of current pulses produce by the tube over a fixed, but configurable, amount of time, called *window size*. The tubes used in the experiments were the SBM-20 and the Si-3bg. According to their datasheets, their working voltages are in the range of 350V to 475V, with a recommended operating voltage of 400V. The Plateau length for both is about 100V.

Two types of experiments were held in order to assess the performance of the designed GM counter, so it could be compared with the GammaSys®Galert 500 P+, a commercial Pancake type GM counter. This one has an effective diameter of 4.5cm and is able to detect alpha, beta, and gamma (X-ray) radiation types. As the radioactive source didn't produced more than 300 CPS, a window size of 10 seconds was used in order to achieve the lowest error possible in the measurements.

A shielded 100mCi ⁹⁰Sr source was used. The decay of ⁹⁰Sr produces beta particles---high energy electrons---and a small fraction---0.02% of 1.76MeV gamma photons. The shielding of the source is designed to completely stop the beta particles, but allows the transmission of a high fraction of the gamma photons, which are the ones we detect with the GM Tube.

Fig. 1 shows the CPS as a function of distance to the radioactive source. We approximated the data with an inverse square law $1/r^2$ showing that at the measured distances, the radioactive element can be seen as a spherical source. With the Si-3bg glass tube, only 1 CPS was measured when the counter was at 1cm from the radioactive source, so it's wasn't

considered. The relationship between the areas of the tube and pancake sensors according to their datasheets is 0.57, indicating why one obtains less sensitivity in the CPS using the GM tube. The average relationship between measured CPS for the SBM-20 tube and measured CPS for the commercial counter is 0.3789, allowing us to make a rough calibration of our system with respect to the commercial one.

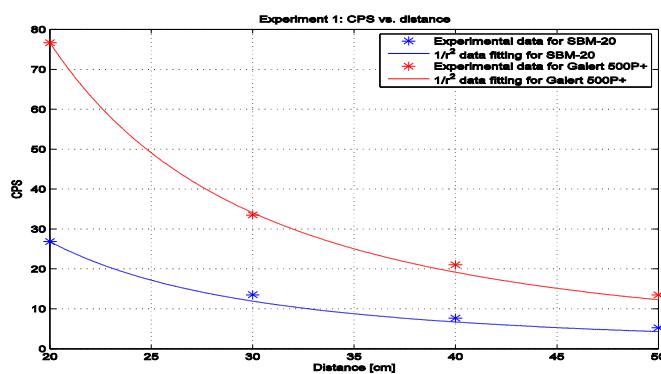


Figure 1

In order to determine the Plateau length, Fig. 2 shows the measured CPS using the SBM-20 metal tube at 20cm from the radioactive source when the biasing voltage was swept from 150V to 400V with a step of 50V was. It's seen that the average CPS measurements are not affected by the tube's biasing voltage.

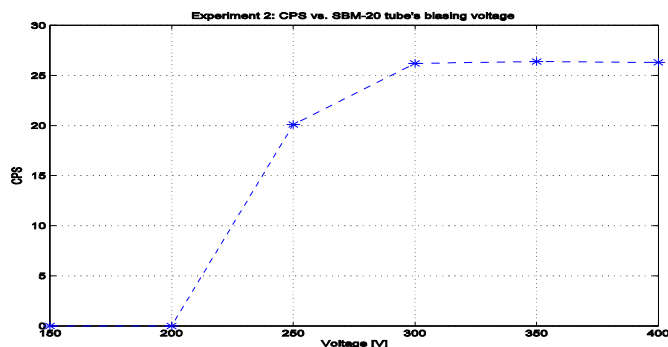


Figure 2

REFERENCES

- [1] SBM-20 Metal Geiger Muller Tube 2398 Datasheet [Online] Available: <http://www.gstube.com/data/2398/>
- [2] Si3bg Geiger Muller Tube 2486 Datasheet [Online] Available: <http://www.gstube.com/data/2486/>
- [3] GammaSys Detector Manual [Online] Available: <http://www.gammasy.com.ar/product5.html>

Diseño de una Tarjeta de Adquisición y Transmisión de Datos Para Comunicación Software-Hardware en Simuladores

Carlos Arturo Gomez Jimenez
Departamento de Investigación
CODALTEC
Villavicencio, Colombia
cgomez@codaltec.com

José Luis Ramos Fernández
Departamento de Investigación
CODALTEC
Villavicencio, Colombia
jramos@codaltec.com

Dayan Velasquez Parrado
Departamento de Investigación
CODALTEC
Villavicencio, Colombia
dvelasquez@codaltec.com

Alexander Cucaita Gómez
Facultad de Ciencias Básicas e Ingeniería
Universidad de los Llanos
Villavicencio, Colombia
alexandercucaita@yahoo.com

La investigación y diseño de una tarjeta de adquisición y transmisión de datos ATD, surge al observar que en el desarrollo de simuladores de vehículos terrestres y aéreos requieren de una gran variedad de componentes de hardware que cumplen diferentes tareas específicas, por ejemplo: la configuración del panel de cambios de una aeronave para modificar la posición de los elevadores y del rudder [1], permitiendo el cambio de rumbo de la aeronave. Cada uno de estos componentes de hardware resulta en múltiples periféricos de E/S tanto análogas como digitales dificultando su integración con el software de simulación dado a que se encuentran a una gran distancia de la CPU y además presentan diferentes interfaces de comunicación.

En el mercado se encuentra un gran número de tarjetas de ATD, pero la gran mayoría no son para el desarrollo del hardware de simulación, como en el caso de las tarjetas estudiadas: NI USB-6341, NI USB-6353, NI USB-6356, NI PCI-6143, NI PCI-6704, NI PCI-6289 y NI PCI-6519 [2], los escasos sistemas que cumplen con las características requeridas, están atadas al software proporcionado por su fabricante impidiendo una ágil integración con el motor de simulación. Otros sistemas para adquirir datos son los PLC pero debido a su arquitectura modular como el caso del SIMATIC S7-1200 [3] resulta ser una solución costosa y poco flexible.

Al no encontrar un sistema que cubra los requerimientos de cada hardware de simulación desarrollado, se opta por diseñar una tarjeta ATD que brinde adaptabilidad, flexibilidad, rápidos tiempos de integración con el motor de simulación y una administración óptima de los recursos.

Luego de realizar el levantamiento de información de cada simulador diseñado por CODALTEC, se observó que el hardware de simulación se compone de una variedad de

elementos que se acoplan a múltiples tarjetas ATD, que administran periféricos digitales o análogos. De acuerdo a los requisitos levantados se decide integrar un microcontrolador ARM Cortex M4 [4] con frecuencia de 80MHz que administrara los recursos de la tarjeta y una FPGA Spartan 3e [5] que brindará flexibilidad al describir hardware que se adapte a las necesidades de cada simulador.

Es importante garantizar el acceso a los recursos por cada una de las tareas programadas en la tarjeta, por tal razón se hizo el estudio de dos Sistema Operativos en Tiempo Real RTOS como el FreeRtos [6] y el SYS/BIOS [7], este último cuenta con una interfaz que agiliza el desarrollo del firmware.

También se investigó e implementó la interfaz JTAG para la programación de la FPGA [8] y del microcontrolador seleccionados, además se investigaron diferentes tipos de protección contra interferencias y sobrevoltajes. Por último se realiza el diseño impreso de la tarjeta ATD en Altium Design [9].

Con la implementación de la tarjeta ATD se espera reducir costos y tiempo en la integración Hardware – Software de simulación.

REFERENCES

- [1] John Anderson, Introduction to Flight, McGraw-Hill, New York, Fourth Edition, 2000.
- [2] National Instruments. (2014, 9 25). Adquisición de Datos Multifunción. Retrieved 10 5, 2014, from <http://www.ni.com/data-acquisition/multifunction/esa/>.
- [3] Siemens AG. (2015). Basic Controller SIMATIC S7-1200, Basic Panels y TIA Portal. Retrieved 7 2015.
- [4] TEXAS INSTRUMENTS. (2013, 11). Getting Started with the Tiva TM4C123G LaunchPad Workshop. Retrieved 9 2014.

- [5] XILINX. (2013, 07, 19). Spartan-3E FPGA Family Data Sheet, Retrieved 7 2015.
- [6] Frías, J. D. (2009, Febrero). Sistemas Empotrados en Tiempo Real. Retrieved 10 2014.
- [7] TEXAS INSTRUMENTS. (2014, 4). Intro to the TI-RTOS kernel Workshop. Retrieved 6 2015.
- [8] XILINX. (2009, 10 26). Spartan-3 Generation Configuration User Guide. Retrieved 6 2015.
- [9] Altium. (2009). Altium Designer Module 5: Multi-sheet Design. Retrieved 6 2015.

Sonda de temperatura digital para uso agrícola

Guillermo Grünwaldt, Matías Pecchia, Gabriel Pereira, Ana Laura Diedrichs, Germán Tabacchi, Matías González González, Gustavo Mercado

Laboratorio GridTICs, Dpto. Electrónica, Universidad Tecnológica Nacional Facultad Regional Mendoza
Rodríguez 273, Ciudad de Mendoza, Mendoza, Argentina
{guillermo.gruenwaldt, matias.pecchia, gabriel.pereira, ana.diedrichs, german.tabacchi, matias.gonzalez, gustavo.mercado}@gridtics.frm.utn.edu.ar

Este documento presenta el diseño y caracterización de una sonda de temperatura utilizada en la Red de Sensores Inalámbricos para Investigación Agronómica (SIPIA), desarrollada por el grupo GridTICs y detallada en [1], cuyo principal objetivo es el monitoreo de temperatura en campo para caracterizar eventos de heladas, que es un fenómeno de escala micro-climático, cuya disminución brusca de la temperatura ocasiona graves daños a los cultivos en épocas de floración y cuaje de frutos, afectando su producción futura. Como requisito de sensado, es aceptable un rango de operación de la sonda de temperatura de $-13\text{ }^{\circ}\text{C}$ a $48\text{ }^{\circ}\text{C}$, según registro histórico de temperatura en el Oasis Norte de Mendoza [2] y una exactitud de $\pm 1,0\text{ }^{\circ}\text{C}$.

La sonda de temperatura es un sistema embebido de escala pequeña compuesta por: un sensor de temperatura TC1047A [3], un microcontrolador PIC12F683 [4], una referencia de tensión MCP1525 [5], un condensador de tantalio de $1\text{ }\mu\text{F}$, tal como indica el fabricante de la referencia de tensión, y un condensador cerámico de 100 nF para filtrado de la tensión en el microcontrolador. Como se explica en [2], la sonda implementa comandos SDI-12, un protocolo maestro/esclavo por consulta, lo que aumenta la interoperabilidad con otros sistemas que utilizan dicho protocolo.

Se realizaron ensayos para determinar la precisión, exactitud (medida como el error absoluto), consumo promedio y tiempo de respuesta, que es el tiempo para que la respuesta de un instrumento se eleve de un 10 % a un 90 % de su valor final al aplicarle una función escalón, con un escalón aplicado de $\Delta T = 1\text{ }^{\circ}\text{C}$. Tanto para la determinación de precisión y

exactitud se utilizaron cámaras térmicas y freezer para testear en valores extremos. Como instrumento de referencia de medición de temperatura utilizamos un termómetro [6] con dos termopares [7].

Los resultados mostraron que la sonda de temperatura funcionó adecuadamente en el rango de $-15\text{ }^{\circ}\text{C}$ hasta $60\text{ }^{\circ}\text{C}$, coincidente con el intervalo de operación de la aplicación. De los ensayos realizados, se concluye que la sonda diseñada cuenta con una precisión de $0,05\text{ }^{\circ}\text{C}$; un tiempo de respuesta de 4 min para alcanzar un 90 % del valor final desde el 10 %, y 7 min para llegar al 99 %, una aproximación mayor hacia la asíntota del valor real. Durante una noche de helada pueden observarse descensos de temperaturas de $1,0\text{ }^{\circ}\text{C}$ en media hora, por ello el tiempo de respuesta obtenido es adecuado a la aplicación. El mayor error absoluto, obtenido en el intervalo de medición, es $1,1\text{ }^{\circ}\text{C}$. El consumo promedio de la sonda de temperatura, suponiendo que la sonda sea consultada cada 5 min, es de $57\text{ }\mu\text{A}$. Pudimos observar que mientras la tensión de alimentación de la sonda se mantuvo mayor a $2,6\text{ V}$ operó normalmente en todo el rango de medición.

REFERENCIAS

- [1] Dirección de Agricultura y Contingencias Climáticas, website: http://www.contingencias.mendoza.gov.ar/web1/agrometeorologia/datos_estadisticos_anuales.php
- [2] A. L. Diedrichs, G. Tabacchi, G. Grünwaldt, M. Pecchia, G. Mercado, F. González Antivilo, "Low-power wireless sensor network for frost monitoring in agriculture research," in Biennial Congress of Argentina (ARGENCON), IEEE, 2014, p. 525-530.
- [3] "TC1047/TC1047A Precision Temperature-to-Voltage Converter," Datasheet, Microchip Technology Inc., 2001-2012. <http://www.microchip.com/downloads/en/DeviceDoc/21498D.pdf>
- [4] "PIC12F683 Datasheet 8-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology," Microchip Technology Inc., 2007. <http://www.microchip.com/downloads/en/DeviceDoc/41211D.pdf>
- [5] "MCP1525/41 2.5V and 4.096V Voltage References," Microchip Technology Inc., 2001-2012. <http://ww1.microchip.com/downloads/en/DeviceDoc/21653C.pdf>
- [6] "Termómetro Fluke 54IIB. Fluke 51-54 Series II Thermometer Product Overview," Fluke Corporation, 1999-2011. http://media.fluke.com/documents/5154___poeng0200.pdf
- [7] "Termopares K Fluke 80PK-1. Fluke Instruction Sheet 80PK-1," Fluke Corporation, 1989-2002. http://media.fluke.com/documents/80pk1___iseng0300.pdf

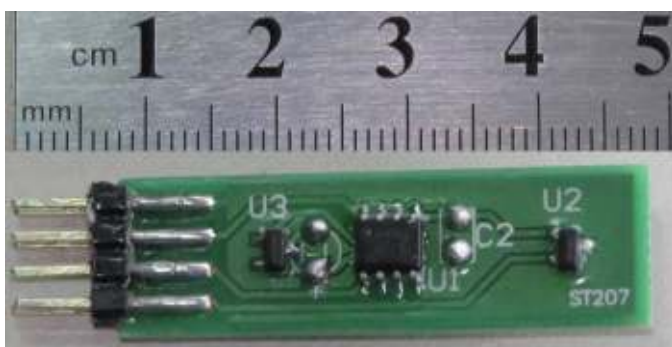


Fig. 1. Fotografía de la sonda de temperatura.

Implementación práctica de mecanismos de calibración en sistemas embebidos

Caso de medición de curva de potencia de pequeños aerogeneradores

Rafael B. Oliva

Universidad Nacional de la Patagonia Austral -
Área Energías Alternativas y L&R Ingeniería
Río Gallegos, Argentina
roliva@lyr-ing.com

Este trabajo presenta consideraciones sobre calibración y su implementación práctica en registradores automáticos de construcción local, orientados a mediciones de potencia eléctrica de corriente continua. Se presentan asimismo técnicas que resuelven en firmware (en lenguaje C) el procedimiento de calibración. Cada registrador es un sistema embebido basado en un controlador ATmega1284 y dos controladores PSoC CY8C29466 [3],[4], y es utilizado para la evaluación de curva de potencia en pequeños aerogeneradores llevada adelante desde 2012 por el Instituto Nacional de Tecnología Industrial (INTI) en Neuquén [6] (Figura 1). El firmware ha evolucionado para acomodar calibración en varios niveles, y almacenamiento de las respectivas constantes de calibración.



Fig. 1. Sistema PWRC2 de registro de curva de potencia [6].

Puede definirse calibración como "un conjunto de operaciones que establecen, bajo condiciones especificadas, la relación entre valores de cantidades indicadas por un instrumento (o sistema de medición), y los valores correspondientes producidos por patrones estándar" [1], y se la vincula crecientemente con los conceptos de incertidumbre y trazabilidad [2]. Si la linealidad del conjunto sensor mas acondicionamiento de señal está ensayada, la calibración por el método de dos puntos [3] permite minimizar los errores promedio de lectura del sistema con un procedimiento sencillo que se incorpora al firmware del sistema, interactuando con el operador en este caso a través de un terminal RS232/LAN. Se

supone una aplicación típica con un ADC de n bits, y que el sistema embebido puede hacer uso de números de punto flotante (sea por software o a través de una FPU). La expresión U.I. indica magnitudes físicas en Unidades de Ingeniería. El circuito de entrada (Figura 2) [5] incluye un sensor con una cierta relación lineal de conversión G_T (V/U.I.), un amplificador con una ganancia A_v , un sumador de tensión de bias V_B , un filtro, un multiplexor ideal, el conversor A/D de n bits y una tensión de referencia FSV. La ecuación general para la conversión del valor numérico (cuenta) ADC_{counts} a la salida del conversor A/D en UI es:

$$U.I. = \left(\frac{ADC_{counts} * FSV}{2^n - 1} - V_{bias} \right) * \frac{1}{G_T * A_v} \quad (1)$$

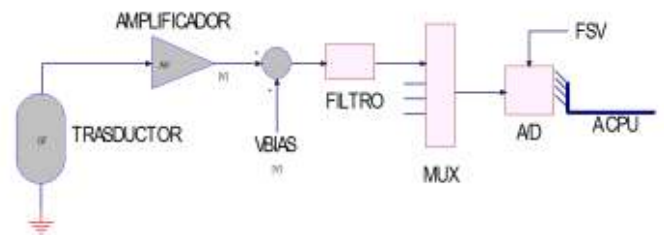


Fig. 2. Etapa de entrada - adquisición de datos, notación de Labrosse [5].

A partir de la (eq. 1), se pueden agrupar convenientemente los valores "por defecto" en dos coeficientes C_{off} y G_{conv} , y definir dos coeficientes de corrección: a) la corrección en cuentas C_{cal} , con un valor inicial 0, y b) la corrección en ganancia G_{cal} , con un valor 1.0 inicial, según (2):

$$U.I. = (ADC_{counts} + C_{offset} + C_{cal}) * G_{conv} * G_{cal} \quad (2)$$

El método de los dos puntos involucra realizar dos mediciones de incertidumbre muy baja en puntos separados utilizando la (eq. 2), y despejar de las ecuaciones los valores de G_{cal} , C_{cal} . Asimismo, este método puede realizarse considerando una única cadena sensor-sistema de adquisición o separarse en distintos niveles. Las constantes son almacenadas en memoria EEPROM no volátil, y son cargadas a RAM durante el siguiente arranque del sistema.

REFERENCIAS

- [1] ISO/IEC Guide 99, (2007) "International vocabulary of metrology -- Basic and general concepts and associated terms (VIM)" International Organization for Standardization (ISO), 1st Ed , Geneva, Suiza, 1993.
- [2] Philips, S.D.; Estler, W.T.; Doiron, T; Eberhardt, K. and Levenson, M. (2001) "A careful consideration of the calibration concept", Journal of the NIST, Vol. 106, N° 2, pp.371-379, ISSN: 2165-7254.
- [3] Oliva, R. (2012) "ESTACIÓN METEOROLÓGICA DE CONSTRUCCION MODULAR ORIENTADA A LA PROSPECCION EÓLICA EN ARGENTINA", Tesis - Maestría en ER, Un.Nac. de Salta (Defendida 03/10/2014).
- [4] Oliva, R. (2014) "Evaluación de incertidumbre en mediciones de potencia eléctrica en registradores automáticos - Caso de implementación para medición de curva de potencia de pequeños aerogeneradores", Libro de Trabajos - Modalidades Foro Tecnológico y Póster, Congreso Argentino de Sistemas Embebidos – CASE 2014, ISBN 978-987-45523-2-7, pp.43-48
- [5] Labrosse, J. (2000) Embedded System Building Blocks, 2nd.Ed., , Ch. 10. R&D Books, ISBN 0-87930-604-1
- [6] A. Zappa, R. Oliva, J. Duzdevich, G. Martín, (2013) EVALUACIÓN DE CURVA DE POTENCIA EN PLATAFORMA DE ENSAYO PARA AEROGENERADORES DE BAJA POTENCIA, Acta de Asociación Argentina de Energías Renovables y Medio Ambiente - Vol. 1, pp. 06.89-06.98, 2013. ISBN 978-987-29873-0-5

Módulo Didáctico para medición de Presión Hidrostática

Nestor J. Cortez

Departamento de Física

Universidad Nacional de la Patagonia Austral (UNPA)

Lisandro de la Torre 1070 - 9400 Río Gallegos - Santa Cruz

TE 02966 442317/19, email: nesjaco@gmail.com

Rafael B. Oliva

Área Energías Alternativas,

Universidad Nacional de la Patagonia Austral (UNPA)

Lisandro de la Torre 1070 - 9400 Río Gallegos - Santa Cruz

email: micro-en@unpa.edu.ar

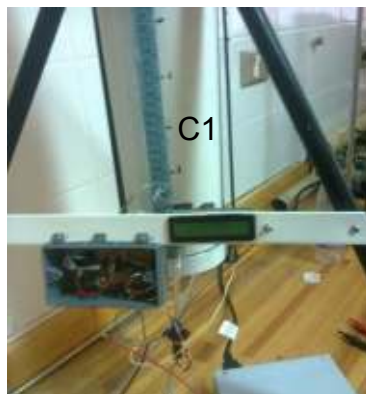
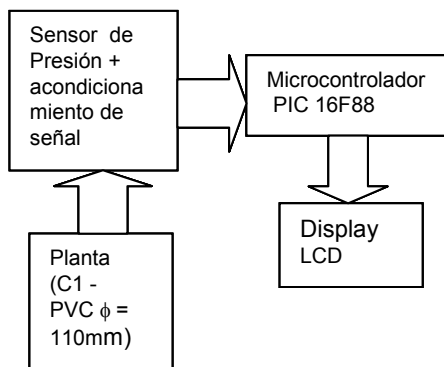
Resumen

La construcción y desarrollo de este prototipo se decidió para complementar y fortalecer el equipamiento didáctico (Medidor de Venturi) existente en el laboratorio de Física de la Universidad Nacional de la Patagonia Austral, Unidad Académica Río Gallegos.

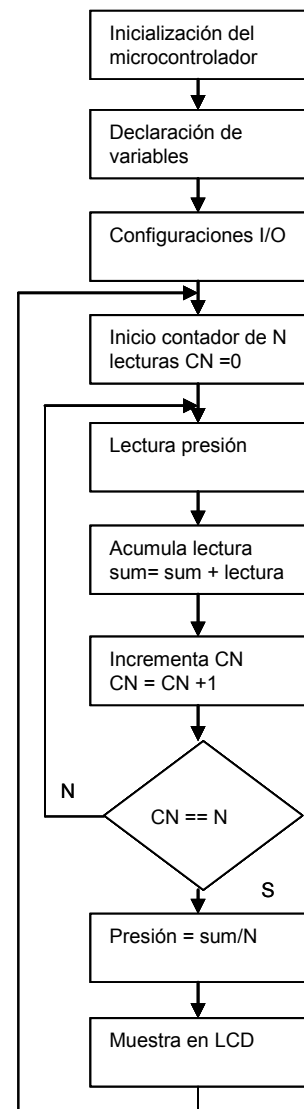
Mecánicamente, el prototipo está constituido por un segmento de cañería (C1) de PVC de $\phi=110$ mm, y una longitud de 1,20 m. La variación de altura en la columna de agua genera una presión en su base que es captada por el sensor MPX10DP de Freescale [1]. Se trata de un sensor de presión piezorresistivo que provee una señal de salida lineal de 20 mV a 50 mV a 25 °C para un rango de presiones diferenciales de 0 a 10 kPa. Se puede alimentar con tensiones de 3 Vcc a 6 Vcc. La salida del sensor llega a un amplificador de instrumentación diferencial de precisión INA122A. Una resistencia externa conectada a este circuito provee la ganancia de amplificación adecuada para la conexión de la salida del INA122A al microcontrolador. Se utiliza un PIC 16F88 (Microchip) que posee canales A/D de 10bits. Después de realizar el procesamiento de la señal se muestra la presión manométrica en un display LCD con una resolución de una décima de kPa. Su lectura máxima será de 9,8 kPa correspondiente a un metro de altura de columna de agua. Las lecturas de este dispositivo se corroboran con las obtenidas teóricamente mediante la expresión (1).

$$P-P_0 = \rho gh \quad (\text{Presión manométrica}) \quad (1)$$

A continuación se presenta un diagrama en bloques del Módulo didáctico, y una foto del prototipo.



El sistema se programa utilizando un compilador C de CSS [3] adquirido para un proyecto de medición similar [2], y el diagrama de flujo correspondiente al programa contenido en el mismo es:



REFERENCIAS

- [1] http://cache.freescale.com/files/sensors/doc/data_sheet/MPX10.pdf
- [2] Cortez, N. J. ; Oliva, O. "DISEÑO E IMPLEMENTACIÓN DE UN REGISTRADOR DE POTENCIA Y ENERGÍA DE BAJO COSTO PARA EMPLAZAMIENTOS AISLADOS - AVANCES" ASADES 2009 (COMUNICACIONES - ISSN 0329-5184) U.N. Rio Cuarto, Vol. 13 Pág. 819 (2009).
- [3] http://www.ccsinfo.com/product_info.php?products_id=PCM_full.

Foro Tecnológico

Pósters

Software Embebido

Controlador para actuador de posición

Diseño, desarrollo e implementación

Ariel Dalmas Di Giovanni¹, Daniel A. Pastafiglia¹, Raúl Cristian Bruña¹, Edgardo A. Comas¹

Laboratorio de Técnicas Digitales-Departamento de Electrónica Aplicada
Instituto de Investigaciones Científicas y Técnicas para la Defensa (CITEDEF)
Villa Martelli, Buenos Aires, Argentina.

¹{adigiovanni,dpastafiglia,cbruna,ecomas}@citedef.gob.ar

Abstract— Se presenta el diseño, desarrollo e implementación de un controlador simple de bajo costo de cómputo para un actuador de posición, utilizando una plataforma de hardware basada en un microcontrolador, con un *firmware* modular y escalable.

Keywords—Controlador, vehículos no tripulados, microcontrolador.

I. INTRODUCCIÓN

Actualmente hay un vasto campo de trabajo en las áreas de navegación, guiado y control, relacionado con los vehículos no tripulados, como ser UAVs (*del inglés: unmanned aerial vehicles*), UGVs, (*del inglés: unmanned ground vehicles*), los cuales incorporan unidades electrónicas autónomas: autopilotos. Son éstas las encargadas de resolver los algoritmos que determinan las maniobras adecuadas del vehículo para que cumpla con un objetivo o misión. Una maniobra se conforma de un conjunto de acciones coordinadas que deben cumplir los actuadores del vehículo.

En muchos casos lo único que se pretende del actuador es que alcance una posición determinada, como ser el caso de un timón de cola de un avión. De esta manera será necesario asegurar mediante un sistema de control que el actuador alcanzó el punto de consigna, la referencia, indicada por el autopiloto. Usualmente este tipo de actuadores se resuelve mediante un servomotor de corriente continua.

De esta manera se puede pensar en una estructura donde hay una unidad autopiloto, responsable del control y guiado de la aeronave que genera las referencias para las unidades responsables del control de los actuadores.

Desde el enfoque de la implementación de algoritmos de control, hoy se tiende a poder implementar el control mediante controladores digitales [1] [4], ya sea con un microcontrolador, DSP, FPGA, según la necesidad.

En el presente trabajo se presenta el diseño, desarrollo e implementación de un controlador para un actuador de posición. Utilizando una plataforma microcontrolada, en la cual se puso el foco en la implementación del *firmware* tal que pueda ser escalable a más actuadores, modular y con la capacidad de implementar otros algoritmos de control.

Otro objetivo perseguido fue la de generar una capacidad en el área de implementación de sistemas de control, orientado a los actuadores.

La estructura del trabajo es la siguiente: en el capítulo II se presenta el actuador, sus características y las estrategias de control a implementar. En el capítulo III se presenta el prototipo de hardware, con un detalle del *firmware*. Finalmente, en el capítulo IV se finaliza con los resultados obtenidos que dan lugar a las conclusiones del trabajo presentadas en el capítulo V.

II. DESARROLLO

A. El Actuador

Para poder implementar y evaluar el controlador fue necesario contar con algún tipo de actuador, a tal fin se utilizó un actuador de posición disponible en el Instituto. El cual se compone de un servomotor de corriente continua, una caja reductora, y un sensor de posición acoplado con el eje del motor. Este último es un potenciómetro de precisión.

El rango de variación del actuador está limitado por dos topes mecánicos dispuestos uno a cada lado del eje, que limitan la posición angular máxima a alcanzar. Es decir, el rango de variación es de $\pm 15^\circ$, sin embargo en los extremos el actuador se bloquea sin posibilidad de revertirlo, es por tanto que se trabajó dentro de un rango de seguridad de $\pm 14^\circ$ y es una restricción en el diseño del controlador.

A los fines de analizar el actuador con la teoría de control clásica se considerará como planta al conjunto actuador – sensor.

B. Obtención de un modelo aproximado

Para tener una dimensión del comportamiento de la planta fue necesario obtener un modelo aproximado del actuador. A tal fin se utilizó la técnica de modelización de un sistema mediante la caracterización de la respuesta al escalón [2] [3]. Para lo cual se excitó al actuador con un pulso de tensión, señal $r(t)$, lo suficientemente “largo” tal que pudiendo así suponer que el pulso se asemeja a un escalón ideal. El pulso es de amplitud fija, 12V de tensión continua.



Fig. 1. Planta a controlar que pueda recorrer todo el rango de posición angular.

Mediante el uso de un sistema de adquisición de datos, se adquirió la señal de excitación $r(t)$, y la respuesta del actuador $c(t)$ en valores de tensión, como resultado de la transducción posición-tensión dada por el sensor, como se muestra en la Fig. 1. En la Fig. 2, se presenta uno de los ensayos, donde se puede ver que la respuesta del actuador es prácticamente lineal.

La respuesta $c(t)$ se la puede modelizar como:

$$c(t) = m \cdot (t - td) + bo \quad t > to \quad (1)$$

Siendo, m : la pendiente de la recta de la respuesta, td : tiempo de retardo de la salida desde el estímulo, bo : valor inicial de la respuesta para $t = to$, y to : Valor temporal donde se produce el estímulo de la planta.

Consideramos que la respuesta del sistema es independiente del valor inicial, es decir un nivel de continua, por tanto despreciamos el término bo , y entonces a la respuesta se la puede expresar como:

$$c(t) = m \cdot (t - td) \quad t > to \quad (2)$$

Haciendo uso de la transformada de Laplace en (2), y considerando que dentro del rango de trabajo: $r(t)$ puede asumirse como un escalón ideal, y la respuesta $c(t)$ como una recta; se obtiene la función transferencia de la planta $G(s)$, a saber:

$$C(s) = \frac{m \cdot e^{-tds}}{s^2} = R(s) \cdot G(s) \quad (3)$$

$$R(s) = \frac{1}{s} \Rightarrow C(s) = \frac{1}{s} \cdot G(s) = \frac{m \cdot e^{-tds}}{s} \cdot \frac{1}{s} \quad (4)$$

$$G(s) = \frac{m \cdot e^{-tds}}{s} \quad (5)$$

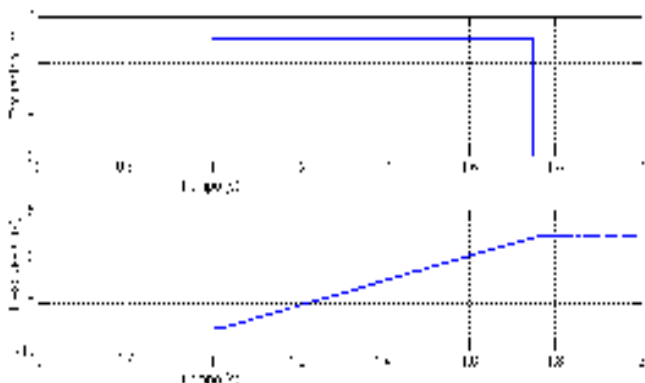


Fig. 2. Ejemplo de respuesta al escalón.

Donde Kp es la ganancia de lazo, coincidente con m . Además, dado que el valor del retardo (td) obtenido es del orden de 8ms, para una primera aproximación se lo desprecia.

Por tanto la transferencia aproximada del actuador $Gp(s)$ es:

$$Gp(s) = \frac{Kp}{s} \quad (6)$$

C. Estrategias de control

El esquema general de un lazo de control se presenta en la Fig. 3, donde: r es referencia del sistema, e es la señal de error, u es la fuerza o acción de control, e y es la salida del sistema.

Dada la naturaleza fuertemente integradora de la planta se plantearon dos estrategias de control, una primera estrategia del tipo On – Off [4] y otra estrategia de control proporcional.

1) Control On-Off:

Según el diagrama de la Fig. 3, se puede definir:

$$e = r - y \quad (7)$$

En este tipo de control se actúa cuando la señal de error es distinta de cero. En particular la acción de control podrá tomar valores de “+1”, “0”, “-1”, es decir: avance, detenido, y retroceso en la posición angular, respectivamente. Para hacer al controlador menos sensible a los errores ocasionados por el sistema de medición, que puedan generar oscilaciones de la señal u , se define una zona de no actuación Δu que mitigue este fenómeno

Ésta estrategia es simple de implementar, pero presenta las desventajas de generar una acción correctora máxima [4].

2) Control Proporcional:

Una forma de modificar el accionar del control On–Off es la de hacer proporcional la acción de control al error, esto se logra definiendo una constante de proporcionalidad K , según la ley de control:

$$u = K \cdot e \Rightarrow u = K \cdot (r - y) \quad (8)$$

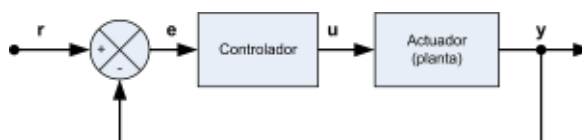


Fig. 3. Sistema de control.

Además, dado que la amplitud de la acción de control está fija en 12V, el parámetro a controlar es el tiempo que ésta permanece activa.

Por otro lado, para implementar la estrategia de control se usará un sistema digital y por tanto se define un tiempo de actuación t_a . Es decir, cada cuánto el controlador tiene oportunidad de modificar la fuerza de control u . El valor mínimo admisible para este parámetro se extrajo de los ensayos mencionados en el apartado B sobre la planta, en concordancia con el tiempo de retardo t_d , de valor 8ms. Por tanto, se definió el tiempo t_a en un valor de 5ms, es decir un poco más rápido que el tiempo de retardo. Este tiempo fue corroborado experimentalmente, observando que por debajo de este valor el actuador no respondía.

A los fines de la implementación digital, el tiempo de activación de la acción de control u se lo puede interpretar como al producto entre un número de pulsos, u_p , y el tiempo de actuación t_a , como se expresa en la siguiente ecuación:

$$u = u_p \cdot t_a \quad (9)$$

En suma, si además se consideran las características del actuador, y siendo e_m la señal de error digitalizada, K se la define como:

$$u_p = K \cdot e_m \quad (10)$$

$$K = \frac{T_A}{RANG \cdot t_a} \quad (11)$$

Siendo T_A : tiempo máximo del actuador para cubrir el rango de posición definido por $RANG$.

De esta manera dada una posición determinada, el controlador deberá actuar en un tiempo proporcional a la relación establecida por T_A , $RANG$, t_a y la señal de error.

III. IMPLEMENTACIÓN

A. Introducción

Para implementar las estrategias de control, es decir el algoritmo del controlador, se diseñó y desarrolló una plataforma basada en un microcontrolador, con capacidad para adquirir, procesar, controlar y actuar.

A su vez, otro elemento rector del diseño fue el de generar un conjunto hardware - *firmware* tal que pueda ser expandido, reutilizado, o sirva como base para futuras aplicaciones y desarrollos en lo que respecta al control de actuadores y temas afines.

Es por tanto que en el caso del hardware se dimensionó más allá de lo estrictamente necesario para la aplicación puntual. En igual sentido, el *firmware* se estructuró de forma tal de lograr modularidad, escalabilidad, reusabilidad y facilidad de comprensión del mismo.

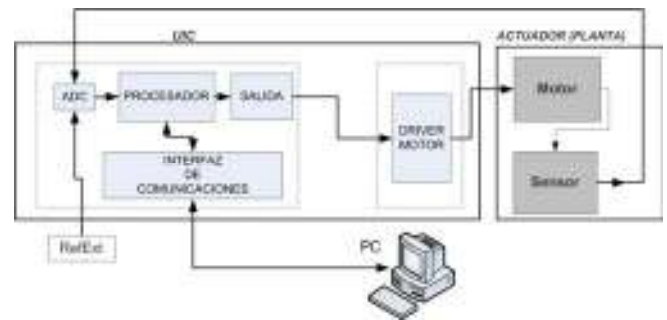


Fig. 4. Esquema de bloques de la plataforma de hardware.

En los siguientes apartados se detallan los principales aspectos del hardware y el *firmware* desarrollados para la implementación del controlador del actuador de posición.

B. Diseño y desarrollo de la plataforma de hardware

A la plataforma de hardware se la puede separar en dos bloques funcionales, uno que se encarga del manejo de las señales de baja potencia y el procesamiento, y otro que es el responsable de manejar la potencia sobre el actuador, es decir el *driver*. En la Fig. 4 se muestra un esquema del actuador, y los dos bloques funcionales antes detallados que forman la plataforma a la cual se la denominó “unidad de instrumentación y control” (UIC).

La UIC incorpora un conversor analógico – digital de cuatro entradas, que permite digitalizar la señal del sensor de posición del actuador, una unidad de procesamiento, donde se implementó principalmente el controlador y una interfaz de salida de baja potencia para conectarse con el *driver* del motor. Adicionalmente se incorpora una interfaz de comunicaciones basada en el estándar EIA-232, para configurar la UIC y poder monitorear su operación.

Las referencias se incorporan al sistema desde:

- La interfaz de comunicaciones, desde un terminal remoto, por ejemplo una PC, un autopiloto, etc.
- Una entrada analógica, *RefExt* en Fig. 4, por ejemplo proveniente desde un Joystick analógico.

La fuente de referencias es configurada al momento de iniciar el sistema y sólo una de éstas estará activa a la vez.

El principal componente de la UIC es un dsPIC33 [8] con el que se resolvió el bloque ADC, el del procesador, y la puerta de comunicaciones, implementada con una UART. Se trata de un microcontrolador de una velocidad de procesamiento que alcanza los 40MIPS, con un ADC de 10 bits que permite el muestreo de cuatro entradas en simultáneo, y conversión secuencial, lo cual es muy apropiado para aplicaciones de control.

El muestreo de cada canal se realizó a una frecuencia de 10kHz, donde se aprovecharon las capacidades de DMA del microcontrolador, para adquirir en 400µs un total de cuatro muestras por cada canal y generar un promedio de cada uno, disminuyendo el ruido blanco de las señales analógicas entrantes. El resultado de cada promedio será el valor utilizado por el controlador. Por tanto, se puede considerar

que la frecuencia de muestreo efectiva se reduce en cuatro veces, es decir a 2,5kHz.

En este caso se puede ver que se ha sobredimensionado la adquisición con la intención que quede como base para otros desarrollos futuros. Un criterio similar es el que se realizó en la elección del microcontrolador, que puede considerarse sobredimensionando para la velocidad requerida de actuación (5ms), pero dota a la UIC de una buena unidad de procesamiento de señales pensando en futuras aplicaciones.

Para el manejo o *driver* del motor se utilizó un “puente H”, haciendo uso del integrado L293D [9], que resuelve el puente de manera fácil y abstracta para el usuario, incorpora internamente los elementos de protección, y las entradas son compatibles para una conexión directa con el microcontrolador.

Como interfaz de comunicaciones se usa una UART del microcontrolador conectada a un FTDI232, para conexión con una computadora vía USB. En el caso de querer vincularse con otro dispositivo se puede prescindir de la interfaz USB y conectarlos en forma directa.

A través de este puerto de comunicaciones, y siguiendo un protocolo de mensajes propietario, la UIC:

- Recibe mensaje de parámetros de configuración.
- Recibe mensaje de referencias para el actuador.
- Envía un mensaje con la información de los canales analógicos adquiridos.
- Envía un mensaje de estados del sistema (Por ejemplo: errores, comando recibido desconocido, etc.)

C. Prototipo de hardware

La implementación de la UIC se realizó mediante un prototipo de hardware formado por una placa de tipo kit desarrollada en el Laboratorio con el fin de ser una plataforma de prototipado rápido que incorpora espacio para soldado de integrados TQFP 64/80/100 y pines para conexionado rápido, una interfaz EIA-232 y FTDI232, una fuente de tensión de corriente continua ajustable, circuitos de reset y oscilador para microcontroladores.



Fig. 5. Prototipo de la UIC.

El *driver* del motor se incorporó en una placa auxiliar, conectada con el kit mediante los pines de conexionado rápido.

En la Fig. 5 se observa una vista del prototipo, donde a la izquierda de la imagen se destaca el kit, el dsPIC33 soldado y rodeado por los pines. A la derecha de la misma se aprecia la placa auxiliar con el *driver*.

D. Firmware

El *firmware* está concebido con la técnica conocida como *superloop* o también *Foreground/Background* [5], que es una forma de trabajo cooperativo entre las distintas tareas a implementar, evitando lazos bloqueantes. El programa principal o *main* se compone de una instancia de inicialización y luego de un lazo infinito que recorre todas las tareas y sólo se ejecutan aquellas que tienen una necesidad puntual. A tal fin cada tarea tiene un *flag* de activación y cada tarea se comunica con otras mediante *flags* y/o colas de datos.

A su vez, hay servicios de interrupciones que atienden y administran periféricos de *hardware* liberando a la CPU de este trabajo. De esta forma se utilizan los siguientes periféricos:

- *Convertor analógico digital*: Se implementa el *Timer 3* como base de tiempos del periférico; a su vez se lo vincula con el canal DMA0 para adquirir cuatro canales y cuatro muestras de cada uno en simultáneo. Esto permite alargar cuatro veces el tiempo de interrupción del periférico, es decir a 0,4ms.
- *Timer 1*: se utiliza como base de tiempo del sistema. Mediante variables asociadas a diferentes tareas que actúan como divisores de tiempo. La interrupción se produce cada 0,1ms.
- *Recepción UART*: se implementa para la recepción carácter a carácter.
- *Watchdog*: en el caso de no haber procesamiento después de un tiempo determinado fuerza el *reset* del microcontrolador.

En la Fig. 6 se presenta el flujo del lazo principal *main*, las tareas principales, los servicios de interrupción y módulos de hardware autónomos. Éstos últimos se representan a la izquierda del esquema como bloques sin conexión.

Cada tarea detallada en el esquema está basada en una máquina de estados finitos.

A continuación se detallará una operación normal del programa y las tareas involucradas, a saber:

La inicialización del programa establece y configura todos los periféricos, variables, tareas y la fuente de referencias.

La tarea *Mensajes* recibe los mensajes vía la UART, ya decodificados, extrae la carga útil y los coloca en la cola de datos correspondiente según el mensaje. Por ejemplo recibe el mensaje de referencias y extrae el valor de ángulo.

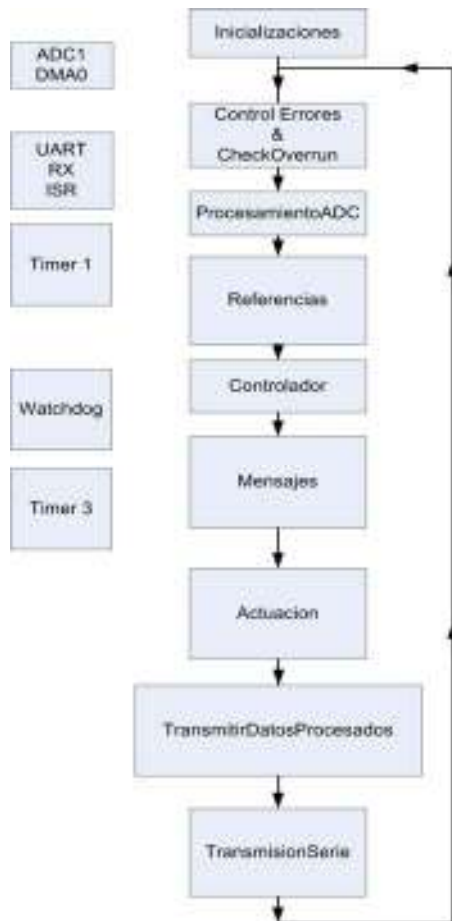


Fig. 6. Esquema del Firmware.

La tarea *ProcesamientoADC*, procesa los datos provenientes del conversor ADC y los coloca en una cola de datos.

La rutina *Referencias* procesa según sea la fuente de referencias establecida en la inicialización, un dato proveniente de un canal analógico (Modo referido anteriormente como *RefExt*), o de la tarea *Mensajes* si el dato proviene de un dispositivo externo.

Los datos provenientes de *Referencias* y *ProcesamientoADC* son tomados por la tarea *Controlador* que implementa las estrategias de control detalladas en el capítulo II, apartado B, incluido el anti-saturador para evitar que se bloquee el actuador. La salida de esta tarea, que es básicamente el tiempo que permanece la salida activa, alimenta a la tarea de *Actuacion* que es la encargada de actualizar las salidas cada t_a unidades de tiempo de actuación, con la polaridad apropiada que necesita el *driver*, para establecer el sentido de giro correspondiente.

Además, cada 2 ms se envía un mensaje con la información adquirida, lo cual es responsabilidad de las tareas *TransmisionSerie* y *TransmitirDatosProcesados*.

Este flujo de diseño permitió incorporar y depurar las tareas una a una sin perjuicio del funcionamiento de las otras.

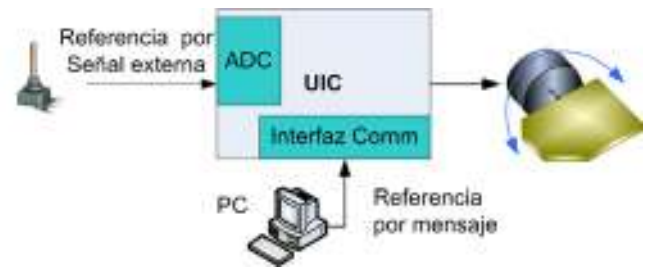


Fig. 7. Esquema del contexto de ensayos.

A su vez el proyecto de *firmware* está enteramente documentado con *Doxygen*, lo que posibilita la generación de documentación automática y una estructura de comentarios de código muy utilizada en el ambiente, que facilita la comprensión del código fuente.

E. Software

Para poder interactuar con la UIC desde una computadora se desarrolló una aplicación de software de alto nivel, en lenguaje *Delphi*, que permite el envío de mensajes de referencia, la recepción de los mensajes de datos adquiridos por la UIC, con almacenamiento en disco para post-procesamiento, y mensajes de estado.

F. Metodología de ensayo

Para evaluar el comportamiento del conjunto UIC-actuador, se implementaron las dos estrategias de control, de forma tal de establecer una comparativa entre ambas. Además, se evaluó el uso de los dos tipos de referencias: por mensajes y en modo seguimiento de una entrada analógica, utilizando un potenciómetro como dispositivo de comando. En la Fig. 7 se muestra un esquema del contexto de ensayo.

IV. RESULTADOS

A. Estrategia de control On-Off

Los resultados responden a la estrategia presentada en el apartado C.1 del capítulo II, On-Off. En la Fig. 8 se presentan los resultados de un ensayo con referencia por mensaje, donde en el gráfico superior se presenta la actividad del *driver* en unidades de tensión. En el gráfico inferior se observa la evolución del actuador, expresado en grados.

B. Estrategia de control proporcional

Los resultados responden a la estrategia presentada en el apartado C.2 del capítulo II, control proporcional. En la Fig. 9 se presentan los resultados de un ensayo con referencia por mensaje, donde en el gráfico superior se presenta la actividad del *driver* en unidades de tensión. En el gráfico inferior se observa la evolución del actuador, expresado en grados.

En la Fig. 10 se observa el resultado de un ensayo exigente utilizando la entrada de referencia analógica externa. El gráfico superior indica la actividad del *driver*, el inferior posee la referencia del sistema, y la respuesta, en colores azul y verde respectivamente. Se podrá notar que en

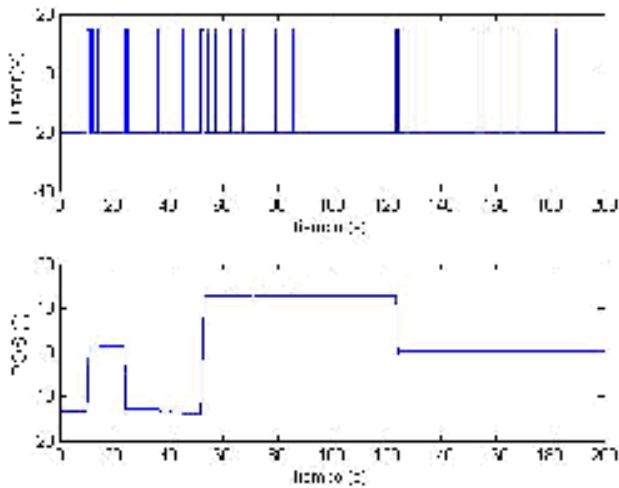


Fig. 8. Estrategia On-Off, referencia por mensaje.

los casos en que las transiciones son muy rápidas, el sistema de control no puede seguir a la referencia, efecto atribuible a la lentitud del actuador y a la estrategia de control implementada. Es importante aclarar que este modo de referencia y la velocidad de variación de la misma está fuera del objetivo pretendido para este trabajo, simplemente se realizó la evaluación con el fin de conocer mejor las características del controlador implementado.

C. Comparativa

A modo de analizar los resultados antes presentados, se puede notar que con la estrategia On-Off el controlador fuerza al *driver* a pesar de estar estable la posición, efecto predecible por tratarse de una acción correctora máxima. Dicho fenómeno se observa en la Fig. 8, en los intervalos comprendidos entre 60s a 100s y 140s a 200s.

En el caso del controlador proporcional, ante situaciones similares, cuando la referencia es estable - Fig. 9 intervalos 40s a 60s y 100s a 120s-, permanece sin acción correctora.

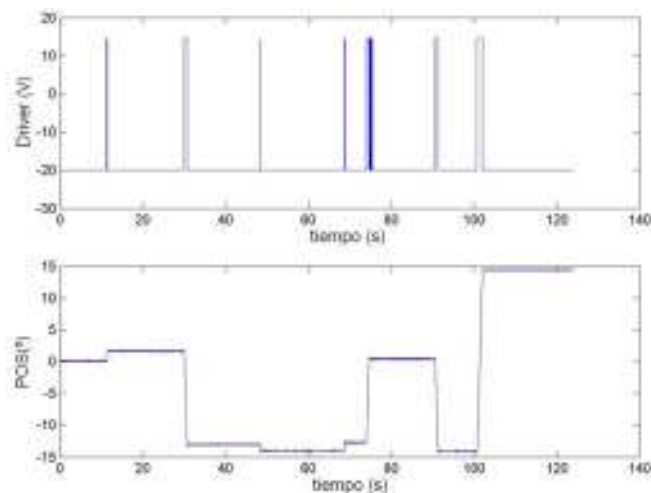


Fig. 9. Estrategia proporcional, referencia por mensaje.

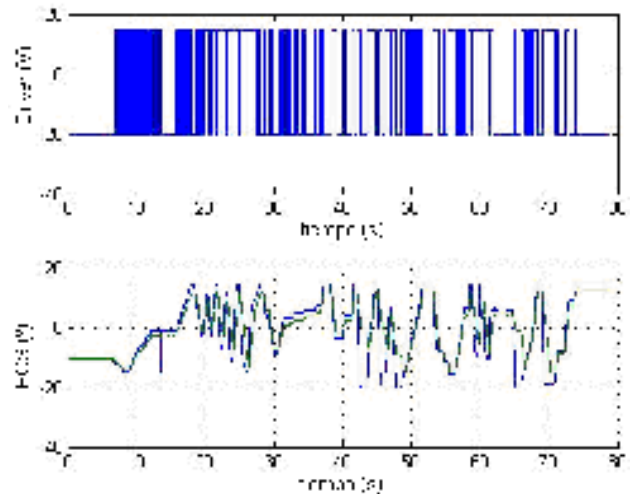


Fig. 10. Estrategia proporcional, referencia por señal externa.

V. CONCLUSIONES Y COMENTARIOS FINALES

Además de haber cumplido con los objetivos planteados, queda demostrado que con el debido conocimiento de una planta es posible diseñar un controlador apropiado, con un adecuado balance de recursos. En este caso, se logró un controlador simple y de bajo costo de cómputo en comparación con otros controladores más difundidos como pueden ser: el PID [3] [4] [7], o un controlador basado en espacio de estados [3] [6].

De todas formas, como trabajos futuros y para ampliar las capacidades de la UIC, se considera incorporar:

- Un controlador PID que posibilite el control de plantas más complejas.
- Módulos de filtrado digital de las señales adquiridas, aprovechando las características de hardware de procesamiento que posee el dsPIC.

Referencias

- [1] Katsuhiko Ogata, "Sistemas de control en Tiempo Discreto". Pearson Educación, 2da Edición, 1996.
- [2] Edward W. Kamen, Bonnie S. Heck. "Fundamentos de señales y sistemas usando la web y Matlab®". Pearson Educación, Tercera Edición, 2008.
- [3] Katsuhiko Ogata, "Ingeniería de control moderna". Pearson Educación, 4da Edición, 2003.
- [4] Kart J. Åström, Tore Hägglund. "Control PID avanzado". Pearson Educación. 2009.
- [5] Gustavo Galeano, "Programación de Sistemas Embebidos en C", Alfaomega Grupo Editor, julio, 2009.
- [6] Tajrin Ishrat, Hasib Bin Liakat. "DC Motor Position Control Drive State-Space Design". Canadian Journal on Electrical and Electronics Engineering Vol.2, N°11. November 2011.
- [7] Tim Bucella. "AN 523 - Servo Control of DC-Brush Motor". Microchip Technology Inc. 1997.
- [8] Microchip Inc, "dsPic33FJXXXGPx06_x08_x10 datasheet". 2009.
- [9] Texas Instruments, "L293 datasheet". Noviembre, 2004.

Digital Sound Meter

Agustn Alba Chicar*, Gabriel Marol†, Lucas Paglia‡, Javier Sankowicz§, Maximiliano Vega¶

Universidad Tecnologica Nacional

Buenos Aires, Argentina

Email: * ag.albachicar@gmail.com , † mgabo_55@hotmail.com , ‡ lucaspaglia@gmail.com ,

§ javi_sank@hotmail.com , ¶ vegamaximiliano0@gmail.com

Abstract—Throughout this document, development and construction of a sound meter is explained. This device consists of a plane response microphone, an acquisition system under standard IEC 61.672 and a microcontroller, which runs algorithms in order to process the acquired signals and display the measurement.. This sort of instrument is widely used in work environments of many kinds in order to check the allowed noise levels according to current laws that guarantee the safety of people in work time.

This project has been carried out by Electronic Engineering students at UTN FRBA within the framework of the classes "Circuit's Theory 2" and "Electronic Measurements 1".

Keywords—

Digital filters, analog filters, spectral power, embedded system.

I. INTRODUCTION

The purpose is to build a device capable of measuring acoustic pressure levels within the frequency band between 31.5 Hz and 8.5kHz, as specified by the IEC 61.672 standard [9], complying third class specs in fast measurement mode. The working principle of this device is based on the spectral power density measurement using Parseval's equation, and then unit conversions prior to being displayed. In order to achieve this, audio needs to be acquired, amplified, filtered, sampled and then digitally processed to obtain the final displayable value.

The sound meter finds its field of action among industry and research. The former, for preventive purposes towards the employee, allowing thereby the characterization of the work environment in which the employees work, avoiding ear injuries. Regarding research, this device is an essential tool for environmental impact fieldwork, allowing the measurement of sound pollution inside a certain environment.

This document details the constructive procedure of a sound meter with commercial characteristics which complies with one of the current regulations, and shows the design decisions taken during building it. This piece of equipment reveals the constructive and technical bases to build a better device, with a better competitive cost. There is a business objective behind the development of the device due to the lack of national manufactured sonometers in the market.

June 30, 2015

II. THEORETICAL BACKGROUND

The sonometer design has some important parts which determines the quality of the device. These are involved with the acquisition and processing of the sound wave. The more important parts are:

- A. Anti-aliasing filter.
- B. DFT algorithms.
- C. Isophonic curves weighting.
- D. Spectral power calculation.

A. Anti-aliasing filter

The anti-aliasing filter is essential in any instrumentation before the sampling block. This filter has been designed based on the Modern Filter Theory using active circuits. Different active structures have been selected, such as Multiple Feedback , Sallen-Key, Ackerberg-Mossberg and State Variable [1], so as to make a sensitivity analysis. In the end, the Inverse Chebyshev in a pass band configuration has been implemented. Studies have been done to compare the structures to obtain the selectivity Q factor and the w_o frequency (among them) which are less sensitive to variance in the component's values. There is a specific problem with capacitors, because of their poor manufacturing accuracy which makes the parameters vary a lot. The State Variable structure has been selected, with the UAF42 integrated circuit because of their high accuracy (1%) in their integrated capacitors which makes it ideal. This characteristic of the ICs is quite important, they let the transfer function parameters keep close to the previously calculated one reaching high Q factors with active structures that do not oscillate.

B. Discrete Fourier Transform Algorithms.

First of all, sound pressure measurement is carried out in the frequency domain. This allows an ideal cutoff frequency, according to the one established by the standards. Hence, all power beyond the working frequency band is deprecated.

Different DFT algorithms such as DFT [2], FFT Radix 2 implemented in DSP Lib, CMSIS [3] provided by NXP, FFT Radix 2 proposed in, and FFT Radix 2 evaluated for real signals [4], all of them with a float numeric base implemented under either ANSI C programming language or Assembler for the ARM Cortex M3 (LPC1769 stick) architecture have been tested (all of them using an input length of 4096 samples). Among all of them, the last one was selected due to the fact that it shows substantial saving of memory and the use of floating point samples in a FPU-lacking architecture does not show a significant cost computationally speaking.

This algorithm uses half the memory, taking advantage of the spectral symmetry of signals which are real in the time domain. Since measured signals by our device qualify as real, this algorithm ensures that the calculated output is accurate from a physical point of view.

C. Isophonic Weighting Curves.

Once all the 4096 samples have been acquired, a scalar multiplication in time domain between the input signal vector and a Flat Top window is carried out (the Flat Top window coefficients have been previously acquired using Matlab). The purpose of using such function is to minimize spectral leakage which can lead to spectral power loss beyond the frequency band set by the standards.

After windowing, the samples are weighted using either the A, B or C isophonic curve. This allows the measurement to resemble the human ear response, which is not plane (like the microphone response). The weighting is also carried out in the frequency domain because its linear nature order. This makes the whole process linear order and it improves the time consumed in the calculation.

Different curves are used depending on the work environment in which the measurement is carried out. The most usual ones are shown below in fig. 1:

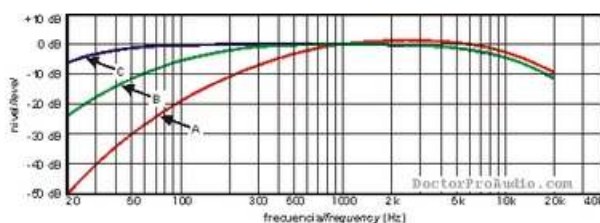


Fig. 1. Isophonic curves.

The weighting curves have been implemented using digital systems. The digitization of the transfer functions, which model the isophonic curves, were originally defined in the Laplace domain. The results have been discrete transfer functions with poles dangerously close to the oscillation zone. These functions are implemented using discrete convolution IIR filters, a large numerical precision is required so that the system does not oscillate.

In order to eliminate the impact of finite precision on weighting, the former is carried out in the frequency domain using scalar multiplication between the spectrum of the acquired signal already windowed and the weighting isophonic curve in the frequency domain.

From an algorithmic point of view, through frequency filtering it was possible to work comfortably on our signals, without any risk of overflow due to numerical format approximation. The digitalization of such signals was carried out using Matlab bilinear transform in order to get from the continuous Laplace frequency domain S to the discrete Z domain.

Coefficients in the frequency domain (bins) were extracted from Matlab directly taking their absolute value, and then used for the weighting process.

D. Spectral power calculation.

The spectral power calculation is made using the well known Parseval relation. It proposes the integration of the square sample vector which derives into the DFT weighted modulus. This operation has linear order and its expression is

the following:

$$P = \sum_{i=1}^n [X(f_i)]^2 \quad (1)$$

III. MATERIALS AND METHODS.

This section is intended to show the details in selected the microphone, its instrumentation, the antialias filter, the ADC selection and the uncertainty analysis, the measurement and calibration procedure. First, a block diagram of the device is shown in fig. 2.

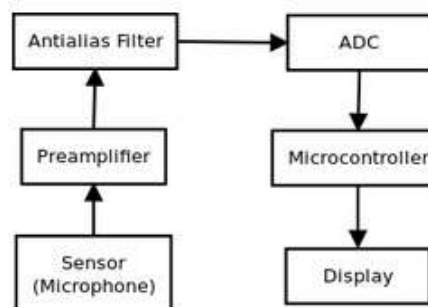


Fig. 2. Device block diagram.

A. Block diagram description.

- 1) Microphone: this is the sensor which converts the sound signals into electric waves. It is capacitive and has a flat frequency response in audible band and high dynamic range.
- 2) Preamplifier: this block is in charge of the amplification of the signal level so as to effectively use the ADC dynamic range and improve the signal to noise ratio of the system.
- 3) Anti-aliasing filter: the active filter block is in charge of adjusting the frequency content of the sound signal in the band of 31.5Hz to 8500Hz. This band is described in the IEC 61.672 standard [9].
- 4) ADC: the analog to digital converter. It should have high dynamic range and resolution following the proposed device characteristics.
- 5) Microcontroller: it will be the device executing the signal processing and of the digital sound signal. It weighs the signal given the different isophonic curves, it calculates the FFT to the sample vector and computes the sound pressure level in dB SPL.
- 6) Display: the visualization part of the device.

B. Sensor selection.

A capacitive flat frequency response microphone has been selected. It also has a high dynamic range and resolution. The model is ECM-8000 from Behringer. Its spatial response is shown in the figure -3- and the frequency response in the fig. 3.

The microphone has the following specifications:

- Type electret condenser, omni-directional.
- 200 Ohms of impedance.
- Sensitivity -70 dB/Pa.

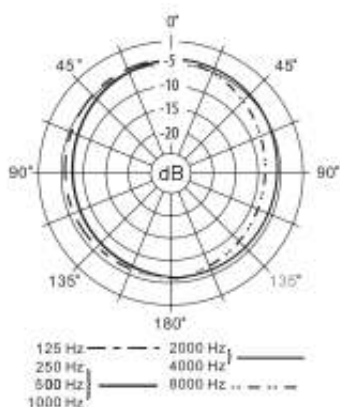


Fig. 3. Microphone polar diagram.

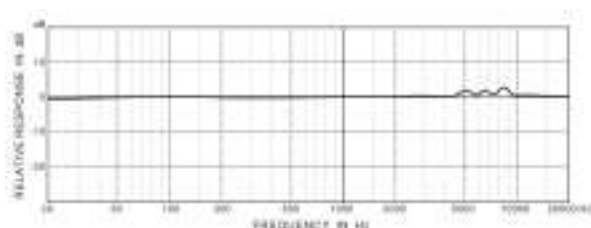


Fig. 4. Microphone frequency response.

- Frequency response in the band of 15 Hz to 20 kHz.
- XLR connector.
- Phantom power supply between +15 V a +48 V
- Weight 120 g

C. Sensor instrumentation circuit.

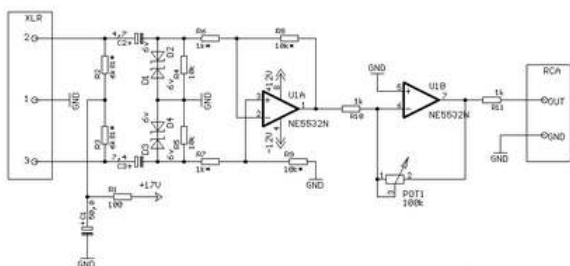


Fig. 5. Pre-amplifier circuit.

The microphone instrumentation is shown in the Fig. 5. We have also implemented a +17V phantom power line (Fig. 6) which polarizes the sensor internal circuit and through the same power lines it delivers the output differential signal. This way, mounting the output signal over the power lines as a differential wave, it improves the noise rejection of the system and makes a better use of the ADC precision. This circuit is capable of transforming a millivolts voltage signal of the sensor into a volts signal at the input of the ADC.

This kind of power supply is commonly used in professional audio applications with high fidelity constraints. It is because it gives to the output the acquired signal as a differential wave with a simple power instead a common

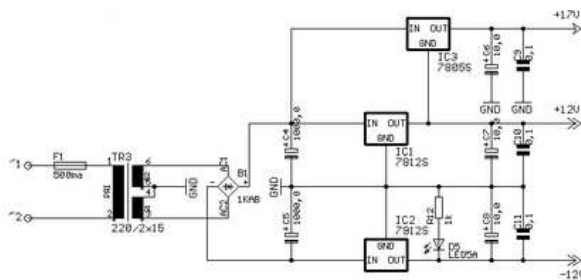


Fig. 6. Microphone power circuit.

mode output.

D. Antialias filter.

The antialias filter is a tenth order pass band filter. It is made by a sixth order low pass filter concatenated with a fourth order one. The lower cut frequency is 31.5Hz and the upper one is 8500Hz. Both points are defined for a 1dB of attenuation. The high order filter is sustained on the sampling period is very relaxed which causes the Nyquist frequency to be close to the upper band restriction standard.

E. ADC.

The diagram in fig. 7 shows the implemented analog circuit block diagram.



Fig. 7. Analog circuit block diagram.

The more important amplification is given by the instrumentation block, indicated with A_{FBP} . The antialias filter in its band pass zone does not amplify the signal though it is indicated with AFBP. The total system amplification is controlled via a trimmer and a linear relationship between two resistors. The following expression is the pressure versus the sensor output.

$$P = 1 - V_{MIC} \frac{R}{1K\Omega} 10^{\frac{-70}{20}} [Pa] \quad (2)$$

In eq. 2, V_{MIC} is the microphone differential output signal. The constant 10 belongs to the instrumentation amplifier gain relation. And the last factor is related to the microphone sensitivity. It is measured in $\frac{Pa}{V}$, so a simple multiplication with the signal gives the differential pressure. The pressure expression is convenient to work with the voltage. Thereby the calculations in the microcontroller will work with the voltage measured by the ADC, and in the end a constant will be applied to get the right units. The input ADC voltage is between -10V to 10V, and this range is divided into 2^{16} -1 steps (for a 16 bit converter) giving $305,18\mu V$ each LSB.

We may transform the pressure expression to get the dynamic range with the previous value of the LSB:

$$RD = 20 \log (2^{16} - 1) = 90.3dB \quad (3)$$

We get a better dynamic range than the one offered by the microphone with a 16 bit converter. The next step explains the verification of the effective bits of the ADC to confirm the selection.

A successive approximations ADC is chosen. The model is ADS7825-P [6] from Texas Instruments. Characteristic are:

- Range +/-10V with single ended power supply 5V.
- Stable internal reference of 2.5V.
- Maximum INL +/-2LSB, see fig. 8.
- Maximum conversion time of 25uS.
- Parallel (8 bit) or serial SPI communication.

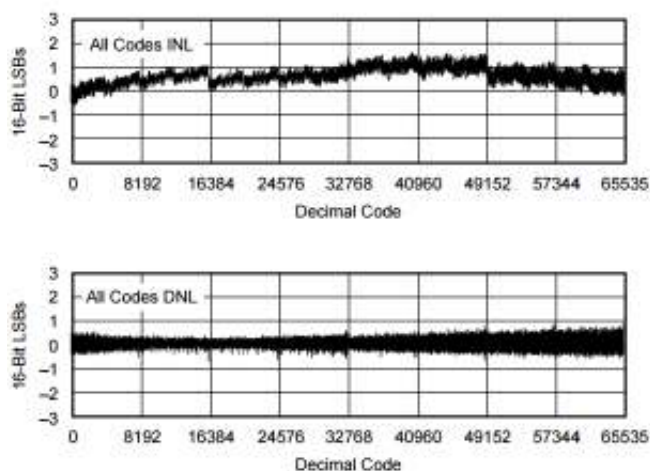


Fig. 8. DNL variation in the ADS7825 IC.

After the datasheet analysis we have determined the following errors:

- Transition Noise +/-0.8LSB.
- Full Scale Error +/-0.5LSB.
- INL +/-2LSB.

Finally we have got 12.7 effective bits, so ADC is appropriate.

F. Measurement and calibration procedures.

The measuring procedure consists of several steps:

- Data acquisition along the time window established by the standards.
- Subsequent windowing using a Flat-Top, which improves significantly the accuracy of the spectral power density measurement.
- Obtaining the FFT of the already windowed real signal and the output vector in the frequency domain.
- Weighting in the frequency domain and subsequent power calculations using Parseval's identity expression [7].
- Multiplying by a scale conversion factor in order to achieve a value with pressure units.
- Multiplying by a measurement correction factor and showing the results in dB_SPL by applying the logarithmic function.

For better understanding, fig. 10 shows the procedure graphically.

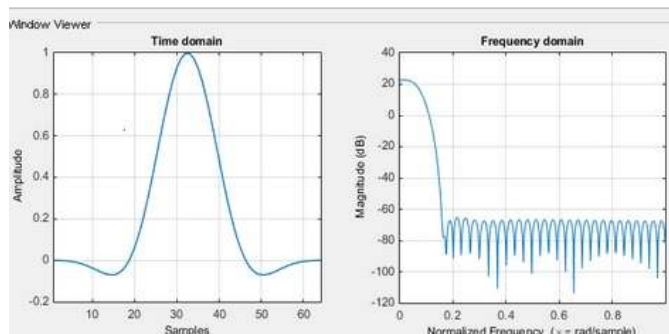


Fig. 9. Flat Top Window. Time domain, left, and its Fourier transform, right.

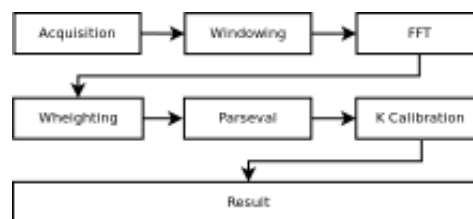


Fig. 10. MeasuringProcedure

The calibration procedure follows the same structure as the measurement procedure. In this case, 100 measurements are carried out one after another. Then the average value is calculated. This is done in order to achieve a lower variance in the measured value. After that, the average value is compared to (divided by) the one that the device is supposed to measure for a standard sound source calibrated at 94 dB_SPL weighted using the A curve.

Finally, after comparing the above mentioned, a constant value called the calibration constant is obtained, which is stored and will be used in later measurements to achieve the measured value specified by the standards, thus accomplishing a minimum user intervention in the calibration process and using only a calibrated sound source as a pattern.

IV. RESULTS

Once built, a series of measurements and tests have been carried out on the device so as to validate its characteristics. These tests allow us to check the right behavior of the device, characterize it and calibrate it. At the beginning, a series of measurements in the work environment let us determine that the device's precision is $318 \mu Pa$:

$$e_{precision} = 318 \mu Pa \quad (4)$$

Since it was logistically impossible for us to test the device using only one sound source with a known frequency inside a soundproof room, we replaced the transducer (the microphone), with a sine wave generator of frequency 1000Hz and amplitude 2 V peak to peak.

The purpose of such a test was to check accuracy, precision, and repeatability of the device, as well as the performance of the analog band pass filter and the weighting algorithms emulating the sound source.

The first test consisted of ten measurements using the sine wave generator, which had been previously set up as described above. The results are shown in table I.

TABLE I. INSTRUMENT'S RELEVANT SPECIFICATIONS.

Average	93.969dB_SPL
Deviation	0.005dB_SPL
Expanded uncertainty	0.003dB_SPL

Later, preserving the signal's amplitude, a few tests have been carried out at different frequencies, within and out of the measurement range in order to check the behavior of the filter and the A weighting curve. Table II has the frequencies and sound pressure levels measured with A weighting curve.

TABLE II. INSTRUMENT'S CURVE A MEASUREMENTS.

Frequency[Hz]	Measurement[dBA]
5	34.00
15	34.00
31	54.79
100	75.34
200	82.58
500	89.76
1000	93.97
2000	95.16
5000	94.54
10000	73.97
11000	61.19
11500	34.00
12000	34.00
15000	34.00
20000	34.00

The 34dB value shown by the instrument in some of the measurements represents to us absolute silence, since 34dB is the minimum measurable noise level. Precise performance can be observed within the measuring range due to the weighting algorithm. It is important to remember that the test signals amplitude was not modified along the test, and that the displayed value is attenuated according to its frequency.

Moreover, in frequencies such as 10kHz and 11kHz, attenuation in the order of 20dB and 30dB respectively can be appreciated due to the analog filtering.

Concerning the analog filtering, amplitude fluctuations of 0.5 dB can be found along the pass band. Furthermore, the cutoff frequencies are within a 1% error margin compared to the original design. No firmware corrections were carried out to correct these frequency errors.

A further advantage of frequency filtering is the implementation of ideal filters (brick wall). This allows us to separate accurately the frequencies which belong to our range from the ones that do not. This result would be impossible to achieve in the time domain because the filters would turn out to be anti-causal.

V. CONCLUSIONS

Different problems in different areas have been found along the development of this project. For instance, filter oscillation, spectral power loss, dynamic range adjustment in ADC, aliasing, and so on.

Most of these problems tend to be usual in this sort of projects. However, we have managed to measure acoustic pressure as planned at the beginning, with uncertainty rates comparable to any commercial device. Moreover, we have

checked our measurements with calibrated pattern devices and calibrated sound sources, which validate our devices calculations.

In comparison with commercial soundmeters, this has little less dynamic range because of the sensor involved, though the filtering and amplification stages support a higher sensitivity. However, repeatability, accuracy and precision is higher than commercial devices.

Future work on this project will be centered in redesigning hardware. The power source of this device is quite important given the fact that every amount of noise it may have will be directly transmitted to the microphone. In our design, noise is stopped by the instrumentation because it rejects significantly and inherently the common mode component of the signal.

Nevertheless, the filtering stage is not balanced, resulting in a worse signal quality if the filtering stage is noisy. The noisiness of the filtering stage is directly proportional to the amount of components used in the analog circuit. As a consequence, high order analog filters will be noisier. Provided the fact that the antialiasing band pass filter has a high order, low noise figure filters are required for such design. An alternative solution to this problem could be designing a lower order antialias analog band pass filter, increasing the sampling frequency and then decimate the samples. This sort of alternatives are currently being evaluated for future designs.

REFERENCES

- [1] *Universal Active Filter UAF42 Application Note*. pgs 6-7.
- [2] *Richard G. Lyons, Understanding Digital Signal Processing, 1st ed., Prentice Hall PTR*, pgs. 130-132.
- [3] *DSPLib CMSIS*. <http://www.lpcware.com/content/faq/lpcpresso/cmsis-dsp-library>.
- [4] *William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, Numerical Recipes in C, 2nd ed., Cambridge University Press.*, pgs.510521.
- [5] *UM10360, LPC176x/5x User manual, April 2 2014, NXP*.
- [6] *Measurement Condenser Microphone ECM8000 Technical Specifications*.
- [7] *ADS7825, BURR - BROWN, DataSheet*.
- [8] *Richard G. Lyons, Understanding Digital Signal Processing, 1st ed., Prentice Hall PTR*, pgs. 80-87.
- [9] *Electroacoustics - Sound level meters - Part 1: Specifications, IEC 61672-1 ed2.0*.

Real-Time Iris-Tracking Embedded System

Cristian Ordoñez*, Juan I. Pastore*[†], Virginia Ballarin* and Eduardo L. Blotta*

*Laboratorio de Procesamiento Digital de Imágenes, Facultad de Ingeniería, U.N.M.D.P., Mar del Plata

[†]Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

ordonez@fi.mdp.edu.ar

Abstract—This paper proposes a real-time iris tracking system using a low cost board based on a ARM M4F microcontroller and a camera. This work is the first step of a system that aims to allow handicapped people with severely reduced mobility to drive a wheelchair using the iris position. The digital image processing embedded system is based on Hough Transform in order to identify the iris position. The design seeks portability and low cost.

Index Terms—Real time, Digital Image Processing, Hough Transform, Iris tracking.

I. INTRODUCTION

ONE interesting application of embedded technology is the assistance of handicapped people. Depending on the degree of disability, some patients experiment a severe limitation to control their extremities, unable to perform tasks such as driving a wheelchair controlled by a joystick. Nowadays, it can be found a lot of developments in this field. Some examples are the chair control by tongue movement presented in [1] or the hands-free control systems based on Electromyography (EMG) signals [2] [3]. Despite these kinds of designs have a good performance they require expensive technology to “interface” human brain to the device.

Most eye tracking methods use computer vision based techniques. In these methods, a camera is set to focus on one or both eyes and record the eye movement. Several different approaches have been proposed and used to implement different algorithms for these technologies -pattern recognition, corneal reflection points, bright pupil effect via infrared illumination, eye models, hybrid techniques, etc- [12]. Although this methods have good results they require advanced processing systems. Making a really difficult task to present methods which can be used in portable and low cost applications to detect and track eye accurately.

The use of computing devices to process images in real time requires performing high complexity mathematical calculations in a very short time, being the Digital Signal Controllers (DSC) one of the tools that most flexibility provides to give high performance solutions to these problems. The DSCs incorporate CPUs with highly advanced architecture, including floating point units (FPU) and digital signal processing (DSP), providing great potential for signal and image processing.

In this work we present a portable and low cost Real Time Iris Tracking Embedded System based on Hough Transform intended to be applied to the wheelchair control in the future.

Iris recognition can be done in several ways [12] but Hough Transform has been chosen due to its satisfactory noise

performance [6]. The Hough transform is a technique used to segment parametric curves in an image. The basic idea is to find a curve that could be parameterized as straight lines, circles or ellipses [9].

The proposed system runs on a Discovery Board. A camera, attached to a pair of glasses, captures images of the eye of the wheelchair driver continuously, then the software process and determines four possible movement states: idle, turn left, turn right and go forward. A display attached to the kit allows showing the results of the processed image, Fig. 6.

II. MATERIALS AND METHODS

A. Hough Transform

When extracting shape information of an object in an image, the most relevant information is often contained within the boundaries or edges of the object [4]. Therefore the edges are used as a discriminative property. The Hough Transform (HT) is a global processing technique that allows to identify and locate objects whose shape is parameterized by the equation $g(v, c) = 0$, where v is a vector of coordinates and c is a vector of coefficients. This technique is robust to noise presence and also to incomplete edges. HT turns a segmentation problem in the image space, in a peak detection problem in the parameters space. Paul Hough developed the HT in 1962 to detect straight lines [5]. Subsequently, Duda and Hart improved the Hough technique extending it to detect other types of parametrical curves [6]. Merlin and Faber [7] probed that the HT can be generalized to detect arbitrary shapes with a certain scale and orientation. Ballard eventually extended the scope of the work, formulating the Generalized Hough Transform (GHT) to efficiently detect arbitrary shapes with any orientation and scale [4] [8].

Starting from the premise that both, the iris and the pupil are perfectly circular, in this paper we propose to apply the Circle Hough Transform (CHT) in order to detect the iris position. Although the premise of iris circularity is not always satisfied the CHT algorithm is good enough for our purpose. CHT is similar to the standard or linear HT, only the space parameters are different, corresponding in this case to the analytical model of the circle [9]:

$$(x - c_1)^2 + (y - c_2)^2 = c_3 \quad (1)$$

where (c_1, c_2) are the coordinates of the center of a circle of radius c_3 passing through (x, y) , on a Cartesian coordinate system.



Fig. 1: Discovery Kit: main board, base board, camera and display

It is usual to use the parametric representation of the circle to implement the CHT [9]:

$$x = c_1 + c_3 * \cos(\theta) \quad (2)$$

$$y = c_2 + c_3 * \sin(\theta) \quad (3)$$

where $\theta \in (0, 2\pi]$

B. Hardware Description

The proposed system runs on a *Discovery* kit, see Fig. 1, based on a *STM32F407VGT6* Micro Controller Unit (MCU). A base board, connected to the *STM32F4* Discovery provides a micro SD Card slot and extension connectors for the LCD and camera boards. A digital camera board featuring a *1.3Megapixel* CMOS sensor and a *3"5* LCD board, with touch screen capability, Fig. 6, compose the whole system. More details on [11].

III. IMPLEMENTATION

The proposed iris segmentation method can be divided in three stages:

1) Data acquisition: One goal of this work is to achieve a video processing system in real time. Therefore, data acquisition is a crucial stage in system performance because it requires the transfer of a large amount of data -each frame is composed of $320 \times 240 \text{ pixels}$, i.e. 76800 pixels - in a very short time.

The MCU has a General Purpose Direct Memory Access Controller -called DMA- which allows data transfers to take place in the background, without the intervention of the processor. During this operation, the main processor can execute other tasks and it is only interrupted when a whole data block is available for processing. Thus, large amounts of data can be transferred with no major impact on the system performance.

2) Thresholding and edge detection: The attached camera, see Fig. 2, is set to operate in a reduced color space called RGB565. Therefore, each image pixel it is represented by a 16 bits word. This is done due to RAM memory limitations. The first 5 bits represent the red channel information, the following



Fig. 2: Camera detail

6 bits the green one and the last 5 bits the blue one. Such limited color space and the continuous environment lighting changes implies that choosing the right threshold level is a really difficult task. We propose to assign a weight to each RGB channel and then add them to get a dataset ready for thresholding, see Ec. 4.

$$G_{(x,y)} = kr * R + kg * G + kb * B \quad (4)$$

where $G_{(x,y)}$ represent the intensity of a generic (x,y) image pixel and kr , kg and kb the weights of the R, G and B channels, where kr , kg and $kb \in [0, 1]$. This conversion improves the threshold selection. Taking into account that, due to its wavelength, the red channel provides more information regarding the human eye than the others channels, we assign a weight of 1.0 to the red channel and a weight of 0.8 to both green and blue channels. After this operation we obtain a gray level image, with a bimodal histogram, which allow us to select an appropriate threshold level. At last, we detect the edges by applying the Laplacian operator [9].

3) Iris candidates: At this stage we seek those points where the probability of having a center circle is maximized. As detailed in Section II-A, this algorithm is based on the localization of the local maximum in the space of parameters, i.e. the 3-dimensional Hough accumulation matrix. The local maximum corresponds to the position of the accumulation matrix where all the possible circumferences satisfy the equation. This involves an intensive use of memory requiring a significant amount of calculations because each frame is composed by 76800 pixels represented in two bytes each.

In order to reduce the computational cost, a slight modification to the Hough Transform algorithm is proposed. Initially, we define a range of radius r where the HT will be applied. Also, the angular space of the infinite circles passing through a point is discretized. Then, through the edge image, we calculate the center of the n circumferences passing through each pixel, where n is equal to the number of discretization points multiplied by the number of possible radios. Each of these circumferences will generate an increase in the Hough accumulation matrix in the c_1 and c_2 parameters which corresponds to a possible circumference center. The local maxima of the accumulation matrix will correspond to those where the probability of finding a pupil is maximum.

To avoid performing complex calculations, such as sine and cosine functions, we defined two vectors that already contain the values of these functions for each discretized point. These

values are scaled to minimize the calculations with floating point variables, which require longer processing time. Finally, the iris center is determined by finding the point where accumulation matrix is maximal.

IV. RESULTS

Figs. 3 and 4 show the processing results of two different eye positions, becoming in turn-right and go-forward commands, while Fig. 5 shows the processing results of a partially hair covered woman eye. The Figures show on the left (a), the images of the captured frames and on the right (b), the LCD images of the processed frames. The cross represent the iris center detected. The test images were taken from UBIRIS database [13].

Processing times of these tests oscillated between 100 and 150ms for each frame, involving data capture, processing and displaying. System clock frequency was set to 144MHz.

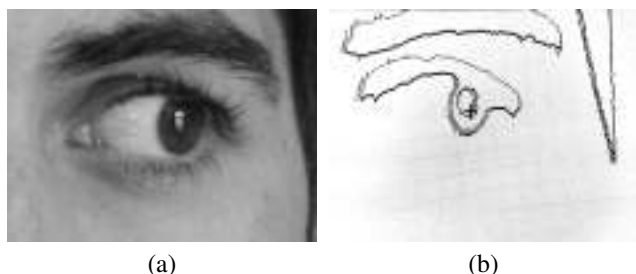


Fig. 3: Subject eye pointing right: (a) Original frame (b) Processed frame.

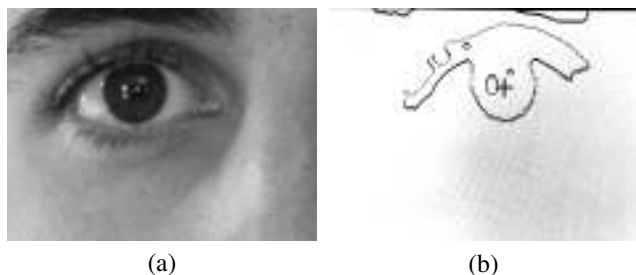


Fig. 4: Subject eye pointing forward: (a) Original frame (b) Processed frame.



Fig. 5: Subject eye partially covered with hair: (a) Original frame (b) Processed frame.



Fig. 6: LCD Display detail

V. CONCLUSIONS

This paper proposes a real-time iris tracking system implemented with a low cost board. Processing times are between 100 and 150 ms for each frame, that is fast enough to give an order of movement direction of a wheelchair if it is considered that six orders per second is more than necessary in a real application. We conclude that the system running on a Discovery Board is able to control a wheelchair through the movement of the iris by an ad-hoc image processing software. The proposed system has the advantage of being minimally invasive for the potential user, who would need a short training process. For future work we will improve the system performance under different environment conditions and we will develop a fully operational system able to control a wheelchair.

REFERENCES

- [1] Lund, M.E.; Christensen, H.V.; Caltenco, H.A.; Lontis, E.R.; Bentsen, B.; Struijk, J.J., "Inductive tongue control of powered wheelchairs", Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, vol., no., pp.3361,3364, Aug. 31 2010-Sept. 4 2010. doi: 10.1109/IEMBS.2010.5627923
- [2] Tsui, C.S.L., Pei Jia; Gan, J.Q., Huosheng Hu; Kui Yuan, "EMG-based hands-free wheelchair control with EOG attention shift detection, Robotics and Biomimetics", 2007. ROBIO 2007. IEEE International Conference on, vol., no., pp.1266,1271, 15-18 Dec. 2007. doi: 10.1109/RO-BIO.2007.4522346
- [3] J. D. Simeral, S. P. Kim, M.J. Black, J.P. Donohue and L.R. Hochberg, "Neural control of cursor trajectory and click by a human with tetraplegia 1000 days after implant of an intracortical microelectrode array", J. Neural Eng., vol. 8, no. 2, April 2011.
- [4] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes", Pattern Recognition, vol 13(2), pp. 111-122, 1981.
- [5] P.V.C. Hough, "Method and means for recognizing complex patterns", U. S. Patent 3069654, 1962.
- [6] R.O. Duda, P.E. Hart, "Use of the Hough transform to detect lines and curves in pictures", Comm. Assoc. Computing, vol. 15, pp. 1-15, 1972.

- [7] P.M. Merlin, D.J. Farber, "A parallel mechanism for detecting curves in pictures", IEEE Trans. Comput., C-24, pp. 96-98, 1975.
- [8] J.S. Bruno, "Reconocimiento de burbujas en formularios inteligentes aplicando la Transformada de Hough", FRBA-UTN, Buenos Aires, Argentina, 2005.
- [9] Rafael C. Gonzalez y Richard E. Woods. "Digital Image Processing". Prentice Hall 2nd Edition 2002.
- [10] Daugman, John, "How iris recognition works", IEEE 2004.
- [11] STM32F4DISCOVERY Expansion Boards Datasheets, www.element14.com/stm32f4-expansion.
- [12] Miad Faezipour y Amer Al-Rahayfeh, "Eye Tracking and Head Movement Detection: A State-of-Art Survey". University of Bridgeport, Bridgeport, CT 06604, USA, 6 November 2013.
- [13] UBIRIS database, <http://iris.di.ubi.pt/>

μPOSIX: Una biblioteca POSIX para microcontroladores

Pablo Ridolfi, Leandro Kollenberger
Laboratorio de Procesamiento Digital
Departamento de Ingeniería Electrónica
Universidad Tecnológica Nacional - Facultad Regional Buenos Aires
Buenos Aires, Argentina
Email: {p.ridolfi,l.kollenberger}@frba.utn.edu.ar

Abstract—Actualmente el mercado de microcontroladores ofrece una gran variedad de arquitecturas con configuraciones de periféricos y memoria muy diferentes entre sí, ya sea que se trabaje con un mismo o diferentes fabricantes de circuitos integrados. Esta marcada variabilidad de hardware resulta en una escasa capacidad de portabilidad del firmware o software embebido que el desarrollador debe enfrentar, muchas veces diseñando su propia Capa de Abstracción de Hardware (HAL, por sus siglas en inglés), y por lo tanto desperdiciando tiempo que podría dedicarse a la implementación de su aplicación propiamente dicha. En este trabajo se propone la utilización del estándar POSIX para la implementación de una HAL orientada a su uso en microcontroladores, que provea una interfaz estándar y unificada para el manejo de periféricos y dispositivos, así como el desarrollo de aplicaciones multihilo de tiempo real a partir del uso de las funciones Pthread (POSIX Threads).

Keywords—POSIX, microcontrollers, API, HAL, real-time, pthread

I. INTRODUCCIÓN

El mercado actual cuenta con una gran cantidad de sistemas operativos *real-time* (tiempo real) orientados al uso en microcontroladores, aunque la mayoría de éstos están orientados a mantener portabilidad a través de varias arquitecturas y fabricantes. Esto provoca que la gran mayoría carezcan de un diseño optimizado al hardware apropiado, por lo cual se pierde no solo rendimiento debido a la gran capa de abstracción involucrada, sino también el soporte de hardware especial de algunos microcontroladores.

El contraste a esto es el uso de las API (*Application Programming Interface*, Interfaz de Programación de Aplicación) nativas de microcontroladores implementadas por los mismos fabricantes y diseñadores del hardware. Los modelos actuales de estas API no se encuentran estandarizados, es decir que el usuario debe estudiar un nuevo modelo cada vez que migra de plataforma. A su vez múltiples sistemas operativos *real-time* con diferentes API no permiten la portabilidad de código desarrollado por el programador entre diferentes arquitecturas de microprocesadores existentes en el mercado, lo cual aumenta el costo de migración a una nueva plataforma, dificultando y extendiendo el tiempo de desarrollo (Fig. 1).

μPOSIX propone un modelo con un acceso estandarizado a los recursos del sistema, como periféricos y servicios del sistema operativo, también implementando los *drivers* (controladores) de dispositivo necesarios y maximizando la portabilidad de las aplicaciones de usuario desarrolladas para diferentes



Fig. 1. Modelo actual de APIs para microcontroladores.

plataformas. Este nuevo modelo introduce una capa de abstracción sobre las librerías externas del fabricante, exponiendo una API familiar para el programador similar a la del estándar POSIX[1] (Fig. 2).

μPOSIX fue creado originalmente para la enseñanza académica del funcionamiento de un *scheduler*, del acceso de bajo nivel a interfaces de I/O (entrada/salida), de administración dinámica de memoria (*malloc*), de programación multihilo (*threads*), y de comunicación entre hilos (*pipes*) en sistemas embebidos; aunque también puede ser utilizado en otras plataformas con requerimientos similares.

II. CARACTERÍSTICAS GENERALES

μPOSIX incluye una interfaz para operaciones con archivos a nivel usuario (con *file descriptors*, descriptores de archivo), un administrador de memoria dinámica basado en el algoritmo TLSF (Two Levels Segregate Fit)[2], un *scheduler* (planificador) orientado a aplicaciones de tiempo real, una implementación de *pipes* compatible con POSIX, y un desarrollo parcial de la librería de POSIX *Threads*, utilizando el *scheduler* incorporado. El diseño de dicho *scheduler* se publicó en [3].

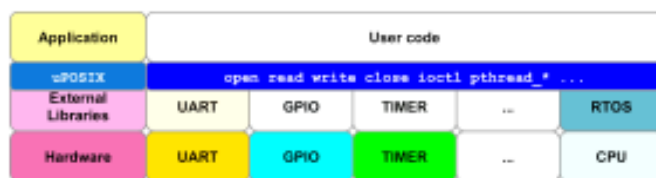


Fig. 2. Modelo propuesto por μPOSIX de APIs para microcontroladores.

Además se incorpora un diseño práctico para la implementación de nuevos *drivers* de otros fabricantes, microcontroladores y arquitecturas, incluyendo ya el código específicamente diseñado para los microcontroladores LPC1769[4] y LPC4337[5] de NXP Semiconductors. El primero es utilizado

ampliamente por la comunidad académica argentina y el segundo fue elegido para el desarrollo de la Computadora Industrial Abierta Argentina[6].

Siguiendo las especificaciones del estándar POSIX, implementa varias funciones características de un sistema operativo, pero orientado a los escasos recursos presentes en los micro-controladores.

Gracias a la naturaleza modular de μ POSIX, la migración del sistema a una nueva arquitectura es simple, basta con agregar los *drivers* específicos, mapear los dispositivos de I/O a nuevos archivos de interfaz (ubicados en /dev según el estándar POSIX) y agregar el código específico necesario para la inicialización del chip.

La modularidad de μ POSIX está dada por la organización seccional del código fuente, respetando una separación del código propio de μ POSIX del código de proyectos externos, y del código dependiente de la arquitectura soportada. El detalle de la organización jerárquica del código fuente está representada en detalle en la Figura 3. Puede observarse el código fuente en [7].

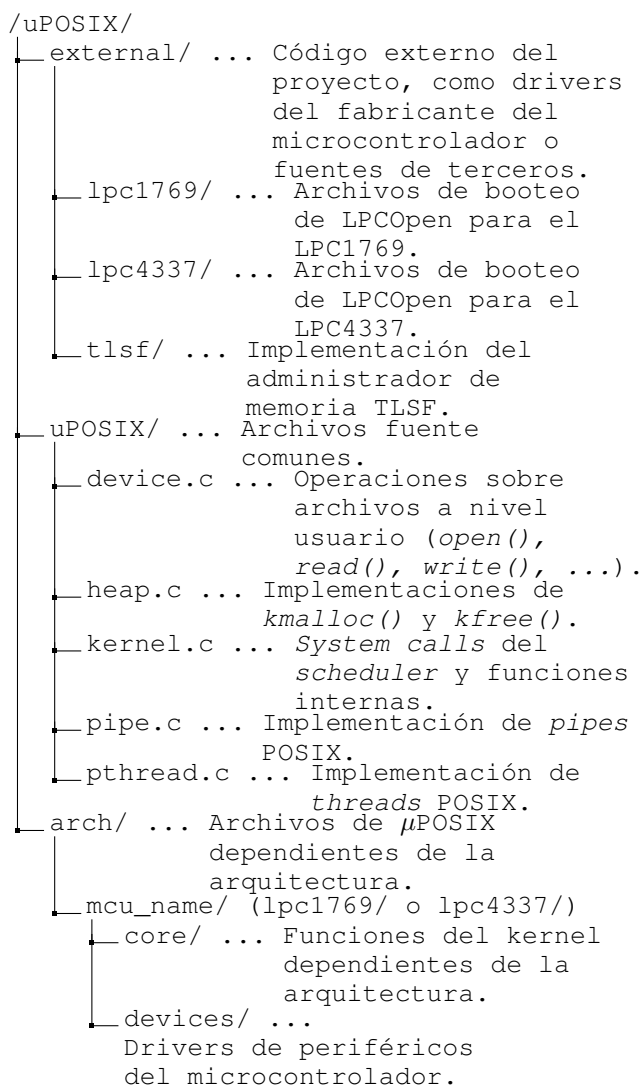


Fig. 3. Organización jerárquica del código fuente de μ POSIX.

III. SCHEDULER Y SISTEMAS OPERATIVOS REAL-TIME

El *scheduler* de μ POSIX exhibe el comportamiento de un sistema operativo *real-time*[8]. Esto se debe a que hay aplicaciones donde el tiempo de respuesta a un estímulo del sistema es un parámetro crítico. Estas aplicaciones son comunes en las instalaciones de sistemas embebidos de control.

En los sistemas *real-time* es un requisito que la respuesta del sistema se ubique dentro de una ventana de tiempo. Respuestas demasiado tempranas o demasiado tardías pueden ser indeseables, por eso un RTOS provee además de confiabilidad un aumento del determinismo del sistema.

El *scheduler* de μ POSIX pone a disposición del programador recursos que permiten asegurar la respuesta del sistema, con cierta tolerancia dada por la aplicación. Estos recursos son:

- Tareas con prioridad definida, reconfigurable en tiempos de ejecución.
- Semáforos de tipo *mutex* (exclusión mutua).
- API preparada para su uso en rutinas de atención a interrupciones (no se requiere una API especial).

En los sistemas multitarea, se define a la tarea como la mínima unidad de rutina de ejecución. También conocidas como *threads*, normalmente sólo se encuentra una única tarea en ejecución (RUNNING) mientras que las demás esperan su turno para ejecutarse. La Figura 4 ilustra los diferentes estados que una tarea puede tomar.

Al llamarse a *pthread_create()*, se piden recursos de memoria al sistema para el almacenamiento del estado de ejecución del nuevo *thread*, sus parámetros, argumentos y *stack* (pila). Seguido de esto, se setea en estado READY, significando que la tarea está en la cola de ejecución esperando acceder a recursos de CPU. Desde ese momento, el *scheduler* toma el control del CPU y selecciona la próxima tarea a ser ejecutada cambiando su estado a RUNNING. El *scheduler* podrá pausar esta tarea para resumirla más tarde según sea necesario y según lo dicte su política de *scheduling*. Esta tarea podrá pasar al estado BLOCKED si en algún momento requiere el acceso a algún recurso bloqueante, tales como una *systemcall*, un recurso de I/O, un periférico, o un *mutex*. En este caso el *scheduler* le asignará el CPU a otra tarea que esté en la cola de tareas en estado READY, buscando a partir de las de prioridad más alta, hasta que la tarea anterior en estado BLOCKED consiga el acceso a su recurso pedido, caso en el que volverá a estado READY y a la cola de espera antes de pasar a ejecución.

El *thread* seguirá un ciclo de ejecución, pausa y/o bloqueo hasta que finalice, o sea cancelado externamente por otro *thread*. Si el *thread* finaliza, llamará internamente a la función *pthread_exit()*, quien se ocupará de marcar el *thread* como ZOMBIE, y esperar un *pthread_join()* de otro *thread* para pasarle su valor de retorno (si es necesario), para luego pasar a estado DETACHED. Si se requiere, se puede llamar a la función *pthread_detach()*, que pasará a la tarea a este último estado sin la necesidad de devolver un parámetro.

Por último, el *scheduler* se encargará de liberar los recursos de los *threads* en estado DETACHED en su próximo barrido de la lista de tareas, para dar por finalizada la ejecución de esta unidad.

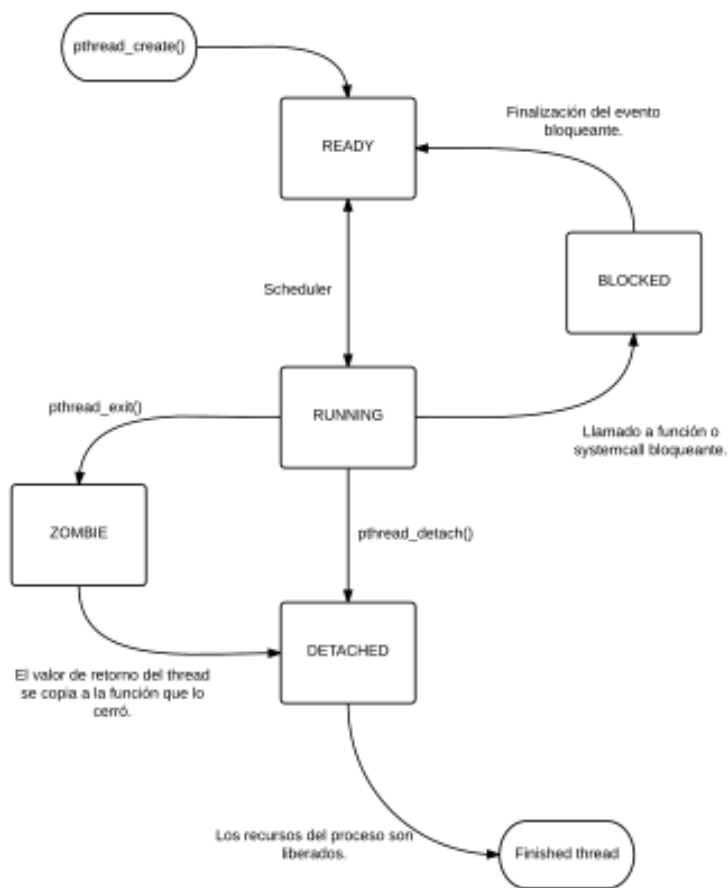


Fig. 4. Diagrama de flujo del funcionamiento de un scheduler tipo compatible con POSIX.

Particularmente en los sistemas *real-time* a cada tarea se le asigna una prioridad. En función de dicha prioridad se determina la tarea a ejecutarse y su intervalo determinado de ejecución, lo cual es decidido por el *scheduler*.

Muchas veces es necesario bloquear una tarea hasta que ocurra un determinado evento. El sistema operativo es el que provee mecanismos que brindan esa funcionalidad, a través de ciertas herramientas disponibles al desarrollador, tales como:

- *Delays* (Demoras)
- Semáforos
- Colas de mensajes

En otros casos es necesario evitar accesos concurrentes a ciertos recursos compartidos por varias tareas. En este caso se suelen utilizar semáforos especiales que bloquean de forma mutuamente los recursos, denominados *mutex*. [9]

El *scheduler* de un sistema operativo debe seguir una cierta política de *scheduling*, de las cuales podemos destacar dos tipos:

- Round Robin (SCHED_RR): Las tareas listas para ejecución (en estado READY) se ordenan en listas según su prioridad y el *scheduler* deberá ir seleccio-

nando tareas desde la lista de mayor prioridad, a la de menor prioridad. Mientras haya tareas disponibles para ejecutar en una lista de prioridad N, las tareas de prioridad N-1 no serán ejecutadas. Las tareas de igual prioridad serán ejecutadas alternadamente, con una ventana de tiempo o *time-slice* (intervalo de tiempo) determinado por el *clock* (reloj) del sistema. En el caso de μ POSIX dicha ventana es de 1 ms.

- FIFO (SCHED_FIFO): Similar a Round Robin, salvo que las tareas de igual prioridad serán tratadas de manera cooperativa, de modo que la tarea será la encargada de ceder el CPU.

En el caso de μ POSIX se utiliza un algoritmo Round Robin porque se considera más estable al momento de implementar una aplicación con requerimientos de tiempo real, ya que no requiere que los hilos en ejecución retornen el control del CPU al *scheduler*. De todas formas, la política FIFO está contemplada para ser implementada a futuro, a fin de lograr compatibilidad con otros RTOS como los basados en el estándar OSEK-VDX[10].

Los *schedulers* de sistemas en tiempo real deben evitar dos problemas comunes en el ámbito multitarea, los dos causados por bloqueo exclusivo de acceso a un recurso. Estos problemas son:

- *Deadlock* o bloqueo mutuo: Ocurre cuando dos o más hilos de ejecución toman recursos utilizados por el otro hilo. Esto puede provocar que ambos hilos esperen la liberación del recurso tomado por el otro y se bloqueen indefinidamente.
- Inversión de prioridades[11]: Ocurre cuando dos hilos de diferente prioridad comparten el acceso a un recurso, que es obtenido por la tarea de más baja prioridad. Al ejecutarse la de más alta prioridad y ésta intenta acceder a ese recurso, se bloqueará, resultando en una inversión de prioridades de las tareas. Este comportamiento es indeseable debido a que puede afectar el tiempo de respuesta de la tarea de alta prioridad.

Los sistemas operativos de tiempo real deben proveer mecanismos adecuados para evitar o minimizar los efectos de estos inconvenientes. En el caso de μ POSIX se implementa un mecanismo de *priority ceiling* (elevación de prioridad) en el cual una tarea de baja prioridad N que adquiere un recurso compartido con otra de mayor prioridad N+P, obtendrá mientras tenga el recurso tomado la prioridad N+P+1. De esta manera se minimiza el tiempo de bloqueo de la tarea de prioridad N+P y se reduce el efecto de la inversión de prioridades.

IV. MANEJO DE PERIFÉRICOS

El acceso a los periféricos de I/O de las plataformas soportadas por μ POSIX son completamente transparentes al usuario, siendo envueltas por una capa de abstracción que provee una implementación similar a la estandarizada por POSIX.

En los sistemas compatibles con dicho estándar, la API vincula a un nodo del sistema de archivos (para los periféricos

se toma como base la ruta `/dev`) con un número entero denominado *File Descriptor*, el cual es un indicador utilizado para acceder de manera práctica y amigable para el programador a diferentes recursos como por ejemplo un archivo, un *pipe*, una conexión remota de red, o en el caso más frecuente visto en los microcontroladores, una interfaz I²C o GPIO.

Siguiendo este concepto, μ POSIX implementa cinco funciones estandarizadas de acceso a periféricos y *pipes* comunes a los sistemas simil-UNIX:

- `open(path_to_device, mode)` Abre o crea un archivo o dispositivo.
- `read(fd, buffer, len)` Lee de un archivo o dispositivo.
- `write(fd, buffer, len)` Escribe a un archivo o dispositivo.
- `close(fd)` Cierra el acceso a un archivo o dispositivo.
- `ioctl(fd, request, data)` Controla un dispositivo, permitiendo enviar comandos específicos.

V. POSIX THREADS

El estándar POSIX especifica un juego de interfaces (funciones y archivos cabecera) para programación multihilo más comúnmente denominada *POSIX Threads* ó *Pthreads*[12].

Un único proceso puede contener múltiples hilos, todos ejecutando el mismo programa. Estos hilos comparten el mismo espacio global de memoria para variables globales, pero cada uno tiene su propia pila (variables locales y contexto).

μ POSIX incluye una implementación parcial de las librerías de POSIX Threads utilizando su *scheduler*, orientada al uso en microcontroladores y otras aplicaciones de bajo costo y limitados recursos de procesamiento. Las funciones implementadas corresponden a las de uso más frecuente y se agregarán más en próximos *releases* (lanzamientos). Actualmente el desarrollador cuenta con:

- `pthread_create()` crea un nuevo *thread*.
- `pthread_exit()` termina el *thread*.
- `pthread_join()` espera la finalización de un *thread* y recibe su valor de retorno.
- `sched_yield()` cede explícitamente el CPU.
- `pthread_setschedparam()` setea la política y parámetros de *scheduling* de un *thread*.
- `pthread_getschedparam()` lee la política y parámetros de *scheduling* de un *thread*.
- `pthread_detach()` marca un *thread* como DETACHED y cuando éste finaliza sus recursos son liberados inmediatamente. No puede hacerse *join* sobre un hilo en estado DETACHED.

Las operaciones de sincronización de *threads* disponibles son las siguientes:

- `pthread_mutex_init()` inicializa un *mutex*.
- `pthread_mutex_destroy()` destruye un *mutex*.
- `pthread_mutex_lock()` toma un *mutex*. Si el *mutex* fue

tomado previamente, bloquea el *thread*.

- `pthread_mutex_unlock()` libera un *mutex* y al próximo *thread* que haya sido bloqueado por el mismo.
- `pthread_mutex_trylock()` es una función similar a `pthread_mutex_lock()` pero no bloquea al *thread* si el *mutex* ya fue tomado.
- `usleep()` pausa la ejecución de un *thread* por un intervalo determinado de microsegundos (se le asigna el estado BLOCKED).

Las operaciones de comunicación entre *threads* pueden llevarse a cabo a través de *pipes* con la API propuesta:

`pipe(pipefd[2])[13]` crea el canal de datos unidireccional que puede ser usado para comunicación entre hilos. Utiliza dos *file descriptors*:

- `pipefd[0]` refiere al puerto de lectura. Se utiliza junto con `read()`.
- `pipefd[1]` refiere al puerto de escritura. Se utiliza junto con `write()`.

Los datos escritos al puerto de escritura del *pipe* son almacenados por el *kernel* (núcleo del SO) hasta que sean leídos por su puerto de lectura.

VI. IMPLEMENTACIÓN EJEMPLO

A continuación se muestra un ejemplo de implementación simple (comúnmente denominado *Hola Mundo*) que enciende y apaga un LED de manera intermitente cada 500 ms. Nótese la API POSIX utilizada, y la simplicidad de acceso a las interfaces GPIO.

```
static void mySysTickCallback(void);
static int fd_gpio;
static int fd_systick;
static devGPIO_pin_t ledStick;

static void mySysTickCallback(void) {
    // Esta función será llamada una vez cada
    // milisegundo por el SysTick, por lo tanto
    // usamos un contador para cambiar el estado
    // del LED cada 500ms.
    static int count;

    count++;
    if (count >= 500) {
        ioctl(fd_gpio, devGPIO_REQ_TOGGLE_BIT,
            &ledStick);
        count = 0;
    }
}

int main(void) {

    // Inicializa librerías LPCOpen
    open("/dev/lpcopen", 0); // [1]

    // Inicializa GPIO, recibe file descriptor
    fd_gpio = open("/dev/gpio", 0);

    // Setea estructura de pines con salida
    deseada
```

```
ledStick = devGPIO_LPCXpresso1769_LED;

// Setea salida para P0.22
ledStick.value = 1;
ioctl(fd_gpio, devGPIO_REQ_WRITE_DIR,
      &ledStick);

// LED off
ledStick.value = 0;
ioctl(fd_gpio, devGPIO_REQ_WRITE_BIT,
      &ledStick);

// Inicializa SysTick
fd_systick = open("/dev/systick", 0);

// Setea el callback creado por el usuario
// con el SysTick
ioctl(fd_systick,
      devSysTick_REQ_SET_CALLBACK, (void
      *)mySysTickCallback);

// Loop infinito
while(1);
} // [1] En este caso el valor de retorno de
// la función open() es innecesario, ya que
// únicamente inicializa las librerías de
// LPCOpen para el microcontrolador.
```

VII. CONCLUSIÓN Y PRÓXIMOS PASOS

Si bien μ POSIX fue creado para la demostración académica del desarrollo de un pequeño sistema operativo, puede ser utilizado en aplicaciones de producción, ya que su amplia capa de abstracción de *hardware* incluida hace posible su portabilidad a otros sistemas y acelera sustancialmente el desarrollo de cualquier aplicación embebida, sin el consumo de recursos de *hardware* que un sistema operativo completo conlleva. Esto puede traer efectos secundarios indeseables, en especial en aplicaciones *real-time* desarrolladas en ambientes de *hardware* reducidos, como es el caso de los microcontroladores. Actualmente las aplicaciones desarrolladas por el programador son compiladas junto con el *kernel*; a su vez la vinculación o *linking* genera un único archivo binario que es cargado en la memoria del microcontrolador.

Una propuesta de continuación del proyecto μ POSIX propone implementar la carga de binarios de aplicaciones dinámicamente en la memoria de código y ejecutarlos según sea necesario, acercando este modelo de aplicación embebida a uno de propósito general manteniendo las características de tiempo real ya disponibles.

Continuando la misma propuesta, se puede implementar un intérprete por línea de comandos compatible con *sh* a través de la interfaz estándar ya implementada por UART, o de forma remota mediante la implementación del protocolo SSH.

El control de forma remota del sistema embebido es otra rama del proyecto pensada para el futuro. Actualmente muchos microcontroladores incluyen una conexión Ethernet y además existen muchas interfaces de conectividad remota compatibles con las interfaces incluidas, por ejemplo, dispositivos Wi-Fi a través de SPI o UART.

Por supuesto el estándar POSIX contiene muchas más funciones que las implementadas en μ POSIX, por eso se irán agregando nuevas a medida que sean requeridas por los diferentes usuarios y sus aplicaciones. Lo mismo aplica a la política de *scheduling* SCHED_FIFO.

Gracias a la modularidad y rápida implementación de *drivers* que provee μ POSIX, todas estas ideas pueden ser plasmadas en realidad de manera muy simple.

Finalmente, se prevee el desarrollo de un *testbench* (pruebas de rendimiento de aplicaciones realizadas con esta plataforma) para la comparación de rendimiento de μ POSIX frente a otros sistemas operativos *real-time* orientados a sistemas embebidos ya existentes en el mercado.

AGRADECIMIENTOS

Los autores de esta publicación agradecen a:

Esp. Ing. Alejandro Furfaro, Director del Departamento de Ingeniería Electrónica de la UTN-FRBA, por su ayuda y acompañamiento permanente para el desarrollo de este proyecto en el marco del Laboratorio de Procesamiento Digital.

Mg. Ing. Mariano Cerdeiro, en su rol de Responsable de Firmware del Proyecto CIAA, por su desinteresado y valioso aporte sobre conceptos de desarrollo de aplicaciones para sistemas embebidos así como la realización de pruebas unitarias y funcionales.

Dr. Ing. Ariel Lutenberg, en su rol de Coordinador General del Proyecto CIAA, por facilitar el acceso a las herramientas y plataformas de evaluación sobre las cuales se hicieron las primeras pruebas de μ POSIX.

REFERENCIAS

- [1] POSIX.1-2008 IEEE Standard, pubs.opengroup.org/onlinepubs/9699919799
- [2] TLSF Memory Allocator, www.gii.upv.es/tlsf
- [3] P. Ridolfi et al., "Implementación de un Kernel de Tiempo Real para Arquitectura ARMv7-M", CASE 2013, Libro de Trabajos - Foro tecnológico y Pósters, pp 189-194.
- [4] UM10360 LPC176x/5x User manual. NXP Semiconductors, http://www.nxp.com/documents/user_manual/UM10360.pdf
- [5] UM10503: LPC43xx ARM Cortex-M4/M0 multi-core microcontroller user manual. NXP Semiconductors, [nxp.com/documents/user_manual/UM10503.pdf](http://www.nxp.com/documents/user_manual/UM10503.pdf)
- [6] Proyecto CIAA: Computadora Industrial Abierta Argentina, proyecto-ciaa.com.ar
- [7] Repositorio GIT μ POSIX, <http://github.com/pridolfi/uPOSIX>
- [8] Renesas RES05B0008-0100: RTOS General Concepts, Enero 2010.
- [9] P. Ridolfi, "Introducción a los Sistemas Operativos de Tiempo Real", Cátedra Técnicas Digitales II, UTN-FRBA, 2014, no publicado.
- [10] Estándar OSEK-VDX, <http://www.osek-vdx.org>
- [11] L. Sha et al., "Priority Inheritance Protocols: An Approach to Real-Time Synchronization", IEEE Transactions on Computers, vol. 39, no. 9, September 1990.
- [12] pthreads(7) - Linux man pages
- [13] pipe(3P) - POSIX Programmer's Manual

Indoor UAV tracking using image processing

Augmented reality techniques for attitude estimation

Pedro Ignacio Martos

Grupo de Procesamiento de Señales, Identificación y Control (GPSIC)
& Laboratorio de Sistemas Embebidos (LSE)
Universidad de Buenos Aires – Facultad de Ingeniería
pmartos@fi.uba.ar

Abstract— UAV’s absolute attitude in an indoor environment is difficult to estimate. This can be done using different techniques; one of them is ambient image processing to estimate the pose (orientation). In this work we present a method for attitude estimation from ambient images using augmented reality techniques, doing it in two steps. First we make a rough estimate of the position of a fiduciary marker, and then we get a smaller image with the marker inside, which is used to estimate the UAV attitude. This approach uses less image processing computational resources for the estimation of the marker projection, and is able to run in a Beagleboard XM board.

Keywords—UAV attitude estimation, augmented reality, marked based tracking, pose estimation.

I. INTRODUCTION

The vast majority of unmanned aerial vehicles (UAVs) use Global Navigation Satellite System (GNSS) technologies, like GPS and GLONASS, and short range and/or inertial sensors (radar, barometric altimeter, sonar, accelerometers, etc.) to estimate their position, attitude, and altitude relative to the level of the sea and/or ground. But in indoor locations the GNSS signal is not available because there is no line of sight to the satellites, and the inertial sensors measure changes in the attitude, but not the absolute orientation, so the UAV onboard navigation system can’t depend only on those signals to estimate its own position and orientation.

There are different techniques to overcome those limitations, ones based on radiofrequency signals, others based on magnetic fields, and others based on image processing. The radio frequency signal techniques use signals available in the environment, like cellular networks [1], wifi [2], RFID [3] or 802.15.4 wireless sensor networks (WSN) [4] for UAV position and attitude estimation. Techniques based on magnetic sensors assume that a map of the magnetic field of the environment is known and, based on vehicle’s magnetometers measurements the vehicle’s position and attitude can be estimated [5]. Techniques based on image processing can be subdivided in the ones that the UAV takes pictures and estimate its pose based on information from the image, like Xu Guili et al [6]; and techniques where a picture of the area (with the UAV inside) is taken and the pose of the UAV is estimated from the pose of the camera, like Redding et al [7]. As a side note, “attitude” means the orientation of the UAV respect to an inertial frame of reference or other entities (the celestial sphere, nearby objects, etc); and “pose” means

the position and/or orientation of the UAV relative to some coordinate system; when this coordinate system is a inertial frame of reference, attitude and pose are nearly the same.

In this work a camera with a known pose takes ambient pictures and the UAV pose (attitude) is estimated from the pictures using augmented reality (AR) techniques. The image processing system hardware is a Beagleboard XM module, so we can evaluate its performance for this task. This method requires the use of some ad-hoc infrastructure, like indoor cameras with known pose, but those cameras are readily available due to security issues.

II. AUGMENTED REALITY TECHNIQUES: CAMERA POSE ESTIMATION

Augmented Reality (AR) is a variation of Virtual Environments (VE), or Virtual Reality (VR) as it is more commonly called. VE technologies completely immerse a user inside a synthetic environment. While immersed, the user cannot see the real world around him. In contrast, AR allows the user to see the real world, with virtual objects superimposed upon or composited with the real world. Therefore, AR supplements reality, rather than completely replacing it. AR can be thought of as the “middle ground” between VR (completely synthetic) and telepresence (completely real) [8]. In Fig.1 there is an example of an AR Application.

The AR is relevant in this work because is the main investigation area about identification of fiduciary marks in environmental images and estimation of the relative pose between the fiduciary mark and the camera. This estimation will be used in this work to estimate the UAV pose from pictures taken from an indoor environment.



Fig.1: AR Example: a windshield with the path and driving information superimposed to the road

As described in Abásolo et al book [9], to merge the virtual and real elements, we start with a real scene with real objects in different poses, and a camera with its own pose relative to the scene. The camera takes a picture and generates an image that is the projection of the scene in the camera, so we can define three 3D coordinate systems: the “local systems”, which are fixed to the objects and references the points of them, and are local to each object; the “world system” which references the positions of all the objects of the scene in a unified coordinate system; and the “camera system” which has its origin in the optical center of the camera. It is possible to pass from one system to other using geometric transformations. Finally, the points in the image are represented in a 2D coordinate system with its axis aligned to the borders of the image, and they are the result of a projective transformation from the 3D points of the scene that are seen by the camera. So, having a point “Pm” in world system coordinates (Xm, Ym, Zm), it could be expressed in camera system coordinates using a geometric transformation “Tg” and then projected to get the 2D point “Pi” (Xi, Yi) in the image using a projective transformation “Tp”. Those transformations can be expressed as successive matrix multiplications, as seen in (1). Note that Zc is the distance from the object to the camera lens.

$$\begin{matrix} P_i & T_p & T_g & P_m \\ \begin{bmatrix} X_i \\ Y_i \end{bmatrix} / Z_c \begin{bmatrix} z_c X \\ z_c Y \\ z_c \end{bmatrix} = \begin{bmatrix} f \cdot m_x & 0 & o_x & 0 \\ 0 & f \cdot m_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} r_{11} & r_{21} & r_{31} & d_x \\ r_{21} & r_{22} & r_{32} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad (1) \end{matrix}$$

To get these matrices is necessary to know the pose of the camera respect to a real object. There are different techniques to do this, one of them is to use a fiduciary mark as the real object: a “marker” with a known 2D shape, and detect its projection in the image from the camera. This technique is known as “Marker Tracking” [9]. With this technique it is possible to get the camera’s pose relative to the marker. As it is a relative reference, it can be used when the marker is fixed and the camera is moving, so we get the pose of the camera relative to the world system coordinates, or when the camera is fixed and the marker is moving, so we get the pose of the marker relative to the world system coordinates, as in this work, which uses a camera with known pose and a marker in the UAV to get the pose (attitude) of the UAV.

III. RELATED WORK

The “object tracking” is defined in [11] as the estimation of the position of a specific object inside an image. That estimation is done looking for specific characteristics of the object, like its shape or color. In her B.S. Thesis, S.A. Currie [12] did the estimation of the position of a miniature helicopter using the light emitted from LEDs on it. Using image processing techniques from indoor pictures, it isolated the characteristic light emitted from the LEDs and based on their position in the image it can be possible to estimate the position of the helicopter in 2D.

Besides that, the use of markers in UAVs is common as an aid for automatic pilots. In Xu et al [6], a marker with a “T” shape placed in the floor is used by the navigation, guidance & control system (NG&C) to estimate its pose for an automatic landing (the marker is fixed and the camera is moving). In this work we investigate the use of fiduciary markers in the UAV to estimate its pose from images taken by fixed indoor cameras (the camera is fixed and the marker is moving).

IV. ESTIMATION METHOD

The UAV pose estimation using markers is a very heavy computational process, because many filters are applied to the image for marker’s borders recognition and projection estimation. To optimize this process, initially we make a coarse estimation of the position of the UAV inside the image. This is done looking for the light from the LEDs in the UAV from a standard resolution image taken by the camera. Once the UAV position is estimated, we take a new image in high resolution and get a subarea from this new image in the zone where we expect to find the marker (near the LED). This subarea will be processed to get the marker pose, which is the UAV pose.

A. Initial Marker Position Estimation

We start with a RGB (Red, Green, Blue) image at standard resolution (640x480). Then we change the color space from RGB to HSV (Hue, Saturation, Value), because in that color space the color information is independent from the intensity, so a HSV image is better for color detection in presence of lighting changes. A Range Filter is applied to the HSV image, (this filter is applied on each component), so we get a binary image, where the white pixels are the pixels similar to the HSV representation of the LED light color. The other pixels in the binary image are black. This binary image is the input of a characteristics detector, which gets the 2D coordinates of the white areas in the binary image; those areas are candidates to be near the markers in the original image. To draw the rectangles in the original image, we use the output of the characteristics detector, which was applied to the output of the range filter, so we get the 2D coordinates of the white areas. In Fig.2 we see the original image with rectangles in the candidate area (the one that would be near to the marker). All the Figures are close-up and with the microhelicopter stopped for clarity.



Fig.2: Original image with rectangle on the candidate area (the LED in the front)

B. Pose Estimation

Once we have the candidate areas, the camera switches to high resolution (1280x720), takes a new picture, and the candidate areas are extracted, generating subimages of smaller size. To identify the marker and its pose, we use the library ArUCO [13], which integrates the marker generation, marker identification and camera pose estimation.

The subimage processing works as follows: An adaptive thresholding filter is applied, so we obtain an image with borders enhanced. These borders not only have the real marker borders, it contains also a lot of unwanted borders/edges, so is necessary to wipe out them. To do this, first the borders with a small number of points are removed (small areas that can't contain a marker). Then a polygonal approximation of the contours is made. We keep the concave contours with exactly 4 corners (candidates to be the projection of the outer rectangle of the marker). After that, the rectangles that are too close are removed, because the adaptive threshold normally detects the internal and the external part of the outer rectangle of the marker's border. Then, using homography, the projection of the rectangles are removed, so a frontal view of the rectangle is finally obtained; this process can be seen in Chum et al [14].

After that, the Otsu algorithm [15] is applied; this algorithm calculates the optimum level of thresholding to generate a binary image from the grayscale one, under the assumption that there are two types of pixels (foreground and background). If the resulting rectangle is the marker, it has an internal code, so the marker is divided in a 6x6 grid, of which the internal 5x5 cells contains ID information. The rest correspond to the external black border. First, the presence of the external black border is checked. Afterwards, the internal 5x5 cells are checked to see if they provide a valid code (it might be required to rotate the 5x5 cell to get a valid one). When the code is identified, the marker is recognized and its projection is used to estimate the pose of the marker relative to the camera. In Fig.3 we see an identified marker with the pose superimposed.



Fig.3: A marker and its pose.

V. EXPERIMENTAL RESULTS

In this work we used a microhelicopter Syma S107 with the markers attached to the sides. With this arrangement the X axis of the coordinate system of the marker is aligned with the horizontal axis of the vehicle and the Y axis is aligned with the vertical axis of the vehicle. We created a software application with OpenCV and the library ArUCO. The images were captured with a standard web cam (Genius FaceCam 1000). The software also had a brightness and contrast control, but the LED detection was unaffected by their variations. This is expected, because the detection is done in HSV color space, which should be immune to lighting variations. The range filter values (`cv::inRange()`) were obtained experimentally because they depend on the ambient characteristics. To do the LED detection we used a characteristics detector (`cv::simpleBlobDetector()`) filtering by area. The maximum LED detection range was around 3 to 5 meters, depending on ambient conditions, with a image resolution of 640x480 pixels. Each candidate area was processed by ArUCO library for marker detection, and when a marked is detected, the 3-axis orientation is superimposed over the original image, as seen in Fig.4.

The maximum marker detection range was around 1 to 2 meters depending on ambient conditions. This distance was independent of the image resolution and marker pose. This happens because there is a detection limit inside the library, who uses an estimator for the position and orientation of the fiduciary mark that looks for fiduciary marks with information inside (like a QR code). So the library's fiduciary mark projection estimator looks for a "frame" which contains the fiduciary mark inside, to get that information. This search for the fiduciary mark's frame needs a high resolution picture and/or a short distance between the camera and the fiduciary marker. Besides that, to get the attitude, one could use a simpler fiduciary marker, for example a "T" shape. But when we were developing the experimental setup, we had to choose between developing our own detector and projection estimator for the T shape; or use the ArUCO library to test the concept, knowing that this approach will have limitations.

Besides that, the software application running in a PC (AMD FX8120/WinXPSP3) processed 20 to 30 frames per second (FPS); when the software application was running in a Beagleboard-XM (Ubuntu 14.04), the performance was around 10 to 15 FPS.

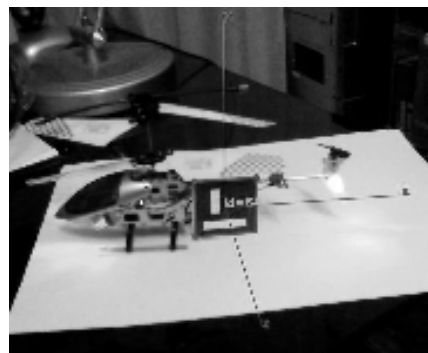


Fig.4: The microhelicopter with its marker and the pose/attitude estimation

VI. CONCLUSIONS AND FUTURE WORK

It was possible to estimate the UAVs attitude in an indoor environment with fiduciary markers using embedded system hardware. To get a better estimation, it would be necessary to integrate this information with the one generated from the UAV's inertial sensors, as described in [14], but this needs a data link between the UAV and the embedded system; this data link will introduce a delay that the algorithm should take into account. We can also use a simpler marker, like the "T" used in [6]. Other improvements are that the parameters of the range filter were adaptive to the ambient conditions, and to use a camera with higher resolution (i.e.1080p). Using a higher resolution camera will require a pan/tilt system to control the pose of the camera also.

ACKNOWLEDGMENT

To Juan Giribet, for his support and suggestions for this work.

REFERENCES

- [1] "Navigation systems based on global system for mobile communications" V. M. Sineglazov, S. S. Shildskiy, *Електроніка та системи управління* Volume 4. (2014)
- [2] "Signal strength based indoor geolocation. In Communications" Chen, Y., & Kobayashi, H. 2002 IEEE International Conference on Communications Volume 1. (2002)
- [3] "An indoor localization mechanism using active RFID tag." Jin, G. Y., Lu, X. Y., & Park, M. S. 2006 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing Volume 1. (2006)
- [4] "UAV real-time location using a Wireless Sensor Network" Rullan-Lara, J; Centre de Recherches de Royallieu, Heudiasyc-UTC, Compiègne, France ; 8th Workshop on Positioning Navigation and Communication – WPNC. (2011)
- [5] "Global indoor self-localization based on the ambient magnetic field" J. Haverinen. *Robotics and Autonomous Systems* Vol. 57, Issue 10. 5th International Conference on Computational Intelligence, Robotics and Autonomous Systems (2009)
- [6] "Use of land's cooperative object to estimate UAV's pose for autonomous landing". Xu, Guili; Qi, Xiaopeng; Zeng, Qinghua; Tian, Yupeng; Guo, Ruipeng; Wang, Biao. *Chinese Journal of aeronautics* Volume 26, Issue 6. (2013)
- [7] "Vision based target localization from a fixed-wing miniature air vehicle" Redding, J. D., McLain, T. W., Beard, R. W., & Taylor, C. N. 2006 American Control Conference. (2006)
- [8] "A survey of Augmented Reality" Azuma, R. *Presence* Volume 6, Issue 4. (1997)
- [9] "Realidad Virtual y Realidad Aumentada. Interfaces Avanzadas" Abasolo, Manresa Yee, Sansó, Venero. 1era Edicion. Editorial UNLP. (2011)
- [10] "Marker tracking and hmd calibration for a video-based augmented reality conferencing system" Kato, H., & Billinghurst, M. *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality – IWAR'99*. (1999)
- [11] "Object tracking: A survey" Yilmaz, A., Javed, O., and Shah, M.. *ACM Computing Surveys – CSUR* 38 (2006)
- [12] "Auto-Pilot: Autonomous control of a remote controlled helicopter" Currie S.A. (B.S. Thesis) Rhodes University (2012)
- [13] "Automatic generation and detection of highly reliable fiducial markers under occlusion" S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, M.J. Marín-Jiménez. *Pattern Recognition* Volume 47, Issue 6 (2014)
- [14] "The Geometric Error for Homographies". O. Chum and T. Pajdla and P. Sturm. *Computer Vision and Image Understanding* Volume 97, Issue 1. (2005)
- [15] "A threshold selection method from gray level histograms" Nobuyuki Otsu. *IEEE Transactions on Systems, Man and Cybernetics* Volume 9, Issue 1 (1979)
- [16] "UAV Position and Attitude Estimation using IMU, GNSS and Camera" Angelino, C. V., Baraniello, V. R., & Cicala, L. 15th International Conference on Information Fusion (FUSION 2012) (2012)

Generador de señales de alta frecuencia y Q-metro para inductores de fuentes conmutadas

Basado en STM32F4 Discovery

Alín Ramón, Rolán
Universidad Tecnológica Nacional
Facultad Regional Bahía Blanca
Bahía Blanca, Argentina

Prof. Ing. Adrián Laiuppa
Universidad Tecnológica Nacional
Facultad Regional Bahía Blanca
Bahía Blanca, Argentina

Este artículo describe una aplicación concreta del kit de desarrollo STM32F4 DISCOVERY, así como las técnicas utilizadas, cálculos, circuitos esquemáticos y el entorno de programación como final de la materia TECNICAS DIGITALES II de la Facultad Regional Bahía Blanca de la UNIVERSIDAD TECNOLIGICA NACIONAL.

Basado en la placa STM CORTEX M4 DISCOVERY y el modulo AD9850 de ANALOG DEVICE, este instrumento de laboratorio proporciona un Q-metro inductivo y un Generador de Señales Cuadradas y Senoidales que van desde los 10Hz a 1MHz y 35Mhz respectivamente. Ofreciendo las funciones básicas que un Generador convencional tales como: variación de frecuencia, amplitud y Offset.

La función Q-metro ensaya inductores que se encuentren dentro del ancho de banda de los 40KHz, basado en un circuito resonante logra la medición de éstos para su aplicación en fuentes conmutadas.

El objetivo de este desarrollo es la de obtener un instrumento de laboratorio económico y con alta performance, de menor peso y tamaño que los analógicos; mejor respuesta en frecuencia y mayor rango de operación.

El Q-metro es una solución concreta para el diseño de bobinas para las fuentes conmutadas, puesto que no son muy comunes estos instrumentos para tan baja frecuencia de operación, además de ser costosos y onerosos.

El generador de funciones otorga las dos formas de onda más utilizadas a nivel académico (senoidal y cuadrada) con lo cual se pueden realizar diversos ensayos con una gran gama de frecuencias disponibles tanto con circuitos analógicos como digitales.

Las funciones que gobiernan el funcionamiento del dispositivo se incluyeron en el diseño para que otorguen la mayor funcionalidad al instrumento.

La función principal o Función Q-metro y Generador de funciones [1], para el generador de funciones están las de control de escalado en frecuencia [2], control de barrido en frecuencia [3], control de amplitud de onda [4] y control de offset [5].

Basado en los amplificadores realimentados en corriente OPA 683, se emplean 2 etapas, una de ganancia 2 y otra de ganancia 5. A partir de las especificaciones dadas por la hoja de datos.

La fuente de alimentación consta de reguladores de tensión tanto para tensiones positivas como negativas. Esta suministra la tensión requerida para todo el sistema, +5V para el kit Discovery, el modulo AD9850, y tensiones duales de +-5V para la amplificación y control de offset.

Para las mediciones del Q-metro inductivo se calcula la frecuencia de resonancia en 40 KHz, se elige un valor práctico comercial de capacitor para iniciar el cálculo de la bobina; este debe ser de un alto factor de calidad, con lo cual no se ven afectadas las mediciones posteriores ni se reducirá la caída de tensión por las pérdidas que pueda presentar.

REFERENCIAS

- [1] OPA683, OPA842, LM7805, LM7905, TL082, LM317K, LM337K : www.datasheetcatalog.com
- [2] Calculo de inductores: material brindado por la cátedra de Tecnología Electrónica (FRBB UTN).
- [3] Teoría de Resonancia: material brindado por la cátedra de Teoría de los Circuitos I (FRBB UTN).
- [4] Teoría de semiconductores: material brindado por la cátedra de Electrónica Aplicada I (FRBB UTN).
- [5] Teoría de diseño de amplificadores realimentados: material brindado por la cátedra de Electrónica Aplicada II (FRBB UTN).
- [6] Material brindado por la cátedra Tecnicas Digitales II (FRBB UTN).
- [7] Definitive Guide to ARM Cortex-M3: Joseph Yiu, Second Edition.

Foro Tecnológico

Pósters

Protocolos, comunicaciones y redes inalámbricas

Sensorística para el control de Plataforma Robótica destinada al estudio de Redes de Sensores Móviles

Trasobares Fernando, Martín Griffa, Leandro Yoaquino, Cristhian Falco
Universidad Tecnológica Nacional - Facultad Regional Córdoba
Centro de Investigación en Informática para la Ingeniería
{trasobaresfernando, martingriffacba, lyoaquino}@gmail.com

Resumen—En el presente trabajo se describe el diseño y desarrollo de la Sensorística y Control de una Plataforma Robótica Móvil, implementada con microcontroladores de la familia ARM Cortex M0, para ser utilizada en el estudio de Redes de Sensores Inalámbricos Móviles (WSNs). La unidad procesadora utilizada fue la placa FRDM-KL46Z de Freescale [1], [2]. Se realiza una pequeña síntesis sobre cada uno de los módulos que forman la plataforma, justificando su elección y diseño. Se utiliza el concepto de campos potenciales artificiales como algoritmo de comando del robot. Para finalizar, se presenta el concepto de modularización de software aplicado al código de control del dispositivo.

I. INTRODUCCIÓN

Una red de sensores inalámbricos (WSN, Wireless Sensor Network [3]) consiste de un conjunto de dispositivos generalmente idénticos, llamados nodos, desplegados sobre un región geográfica de interés, que se usa para la medición y monitoreo de diversos fenómenos físicos, o para la detección y seguimiento de eventos, en forma cooperativa y coordinada. Los elementos que conforman la red pueden estar interconectados entre sí mediante una estructura de malla, o bien siguiendo un esquema jerárquico con estructura de árbol con nodos cabecera que reciben la información sensorial de otros, información que luego remiten hacia un nodo central que procesa toda la información sensorial. Las WSN se utilizan en una gran cantidad de aplicaciones, como son agricultura, monitoreo industrial y/o ambiental, automoción, tracking y domótica.

Dentro del campo de estudio de las Redes de Sensores Inalámbricas, se identifican aquellas de naturaleza Móvil, en las cuales los nodos tienen la capacidad de desplazarse dentro de un área de interés. Este tipo de esquema de Red es beneficiosa en situaciones donde los sensores se encuentran desplegados sobre medios cambiantes, como son ríos y mares, lo que conlleva a que las posiciones de los nodos pueden ser dinámicas. Otro tipo de aplicación es en situaciones en las que se requiere hacer una reubicación de los elementos de la red para mejorar el área de sensado. Para ello, los robots móviles pueden desplazar los nodos sensores según criterios preestablecidos como por ejemplo una mejor señal, ganancia de información, etc.

Para poder evaluar los algoritmos y aplicaciones de las redes de sensores inalámbricas móviles (WSN) es necesario contar con un banco de pruebas o plataforma de experimentación. La mayoría de los bancos de pruebas son sistemas complejos donde además de los sensores, existe toda una infraestructura por detrás para la distribución de energía, y para programar o depurar el funcionamiento de cada nodo. Esto resulta en una estructura que además de ser costosa resulta poco flexible en cuanto a la modificación de la topología de la red, como su aplicación o actualización.

Por otro lado, existen también bancos de pruebas para redes de sensores móviles en las cuales surgen dos consideraciones importantes: primero que la infraestructura de apoyo (para alimentación,

programación y depurado) resulta poco viable y segundo, que en una red móvil es necesario definir de alguna forma la movilidad de cada nodo de la red. Si bien esta movilidad dentro de la plataforma de testeo puede ser provista por vehículos móviles, su costo puede ser crítico a la hora de aumentar la escalabilidad de la misma.

La Plataforma Robótica aquí presentada es un dispositivo móvil controlado creado bajo la premisa de que sea pequeño, económico y fácil de programar. En aplicaciones de investigación, es necesario disponer de este tipo plataformas robóticas sencillas, de forma que el investigador pueda centrar su atención en su proyecto, evitando encontrarse con problemas de programación. De allí que realizamos el diseño de un robot que cuente con distintos grupos de sensores que le permitan realizar una gran diversidad de tareas.

Actualmente, se encuentran disponibles dispositivos open source, similares al que se presenta en este artículo. Uno de ellos es [4], que utiliza sensores IR y cámaras para el control del robot. En [5] también utilizan cámaras como elemento de control, acompañadas con sensores de ultrasonido. Finalmente, uno de los proyectos más llamativos debido a su diminuto tamaño es [6], que basa su sistema en sensores IR.



Fig. 1: Estructura utilizada para la construcción de la plataforma.

El desarrollo de Sensorística y Control presentado en este artículo, encuentra sus bases en el proyecto “Soluciones tecnológicas para el monitoreo distribuido mediante redes inalámbrica de sensores”. Este es un proyecto de cooperación entre el Laboratorio de Comunicaciones Digitales (LCD) de la UNC (Universidad Nacional de Córdoba), Insus (Ingeniería Sustentable) de la Incubadora de empresas de la UNC y el C.I.I.I. (Centro de Investigación en Informática para la

Ingeniería). Esta cooperación se enmarca dentro de los “Proyectos de Asistencia Exportadora Manuel Belgrano”, que tienen la finalidad de impulsar las capacidades exportadoras de las Pymes mediante el trabajo conjunto con las Universidades públicas. Como parte de este proyecto, se desarrolló una estructura robótica que se observa en la Fig. 1, basada en trabajos previos del C.I.I.I como el RoMAA [7].

La estructura robótica desarrollada fue hecha con materiales económicos, que pudieran ser cortados a laser o con una fresa, para facilitar su reproducción. Además, se colocaron dos servomotores, modificados para ser utilizados como motores de corriente continua controlados por una señal de PWM. Por último, se colocaron baterías con cargadores externos, obteniéndose una Plataforma completamente simple y equipada.

Sobre dicha Plataforma fue realizado el desarrollo de Sensorística y Control presentado en este artículo, orientado al estudio de Redes de Sensores Inalámbricas Móviles. Se persiguió el objetivo de proveer módulos sensoriales y de control, para que la Plataforma sea capaz de transportar el nodo que será estudiado, así como de proveerle alimentación, posibilidad de programación y depurado. Otro de los pilares fundamentales de nuestro proyecto fue la modularización del software utilizado, para permitir un control sencillo de la plataforma con programación de alto nivel. Este proyecto fue realizado como trabajo integrador del 4° año de la carrera de grado Ingeniería Electrónica.

La organización de este artículo se presenta de la siguiente manera. La sección II. *Motivación del Proyecto* refuerza los objetivos planteados al inicio del proyecto. III. *Descripción de módulos y circuitos* justifica la elección de cada uno de los módulos y describe el diseño de cada uno de ellos. La sección IV. *Esquema de Programación* esboza la organización modular del software. Finalmente, en V. *Conclusiones y trabajo a futuro* se hace un balance de los resultados obtenidos en el desarrollo del proyecto.

II. MOTIVACIÓN DEL PROYECTO

El objetivo planteado a la hora del diseño del proyecto fue desarrollar e implementar el control de una Plataforma Robótica que tuviera la capacidad de desenvolverse de forma favorable en ambientes adversos, mediante el agregado de módulos sensoriales. El dispositivo debe ser capaz de esquivar obstáculos y de realizar trayectorias que, o bien respondan a programaciones preestablecidas, o se basen en la toma de decisiones en tiempo real, en respuesta a las lecturas de los sensores.

Otra de las metas perseguidas fue la modularización del código de programa que controla la actividad de la Plataforma. De esta manera, se realizaron funciones que permiten utilizar el dispositivo mediante una programación de alto nivel con sentencias sencillas.

Estos objetivos fueron planteados en base a una necesidad específica, presente en las investigaciones actuales de sistemas de sensores. Cuando se estudia el comportamiento de distintos dispositivos sensoriales que integran una red o sistema, se necesita que los sensores se muevan en trayectorias conocidas (Fig. 2), para estudiar la interacción entre nodos, mientras se comunican con centrales, que recolectan la información, y con el resto de los sensores. De esta manera, el investigador puede analizar los datos recibidos y obtener importantes conclusiones sobre la interacción de los sensores.

En resumen, se busca proveer un dispositivo que pueda transportar los sensores de la red que se está estudiando y que cuente a su vez

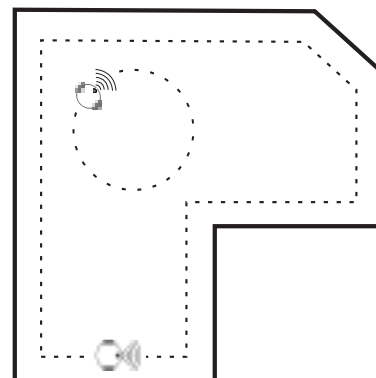


Fig. 2: Aplicación de la Plataforma.

con los módulos sensoriales y de control para desplazarse de forma segura en un espacio reducido según consignas preestablecidas. Todo este sistema se monta sobre la estructura robótica presentada en la Fig. 1. Por otro lado, se realizó la programación necesaria mediante funciones, de forma que el usuario del robot pueda controlarlo con un sencillo programa en lenguaje C.

III. DESCRIPCIÓN DE MÓDULOS Y CIRCUITOS

La estructura robótica utilizada cuenta con dos servomotores como elemento motriz, debido a su simplicidad de control de velocidad mediante señales PWM. Por otro lado, la utilización de motores implica una circulación de corrientes que podría comprometer la integridad del microcontrolador, por lo que se incluyó un circuito de protección.

Finalmente, se eligieron dos sistemas sensoriales para el control del funcionamiento del robot, uno basado en sistemas infrarrojos de medición de distancia y otro en la detección y seguimiento de una fuente de luz. En la Fig. 3 se observa un diagrama en bloques de la Plataforma.

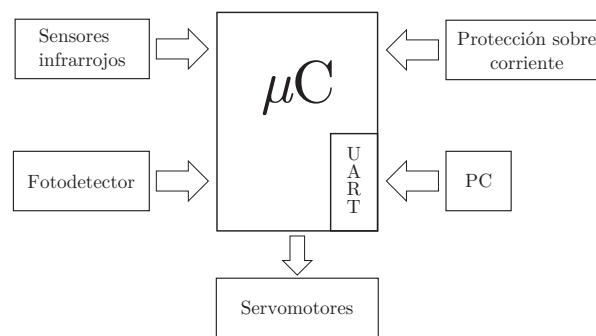


Fig. 3: Diagrama en bloques del sistema.

En la Fig. 4 se puede apreciar una fotografía de la Plataforma terminada en la que puede distinguirse los distintos bloques estructurales que se mencionan en la descripción y en el diagrama. Por otro lado, en la Fig. 5 se aprecia la placa que contiene todas las etapas de procesamiento de señales y el control del microcontrolador.

A continuación, se detalla cada uno de los módulos circuitales implementados en la Plataforma.



Fig. 4: Fotografía del robot terminado.



Fig. 5: Fotografía de la placa del robot.

III-A. Placa controladora

Para realizar todo el procesamiento de la información adquirida por los sensores, y para controlar el funcionamiento de la Plataforma, se utilizó la placa FRDM-KL46Z de Freescale, que es una plataforma de desarrollo de bajo costo perteneciente a la familia de microcontroladores KL4x de la serie Kinetis L construido con el procesador ARM® Cortex™-M0+ de 32 bits que funciona a una frecuencia de 48MHz. Dentro de las principales características de esta placa se encuentran un sencillo acceso a las puertos I/O del microcontrolador, built-in debug y operaciones de bajo consumo.

A continuación se presenta una lista detallada de las características de la placa:

- Freescale KL46Z Kinetis KL4 MCU (MKL46Z256VLL4)
 - Núcleo ARM® Cortex™-M0+
 - 48MHz, 32KB RAM, 256KB FLASH
 - USB (Host/Device)
 - SPI
 - I2C
 - I2S
 - UART
 - PWM
 - ADC
 - DAC (1x 6bit, 1x 12bit)

- Touch Sensor
- GPIO
- LCD Controller
- FRDM-KL46Z Sensores Onboard
 - MMA8451Q - Acelerómetro tridimensional
 - Sensor Táctil Capacitivo
 - MAG3110 - Magnetómetro
 - LCD-S401M16KR - LCD de 4 dígitos y 8 segmentos
 - Sensor de Luz - ALS-PT19-315C/L177/TR8

III-B. Circuito protector de sobrecorriente

La utilización de servomotores implica grandes ventajas en lo referido a simplicidad de control de la velocidad y sentido de giro, sin embargo, la ausencia de encoders hace imposible conocer si el motor está bloqueado o no. En caso de que los motores se traben, el microcontrolador no tiene forma de detectar la situación e interrumpir la alimentación del servomotor, por lo que el mismo seguirá incrementando su corriente para librarse del bloqueo. El consumo del motor con el rotor enclavado puede alcanzar los 1,5A, suficiente para producir fallas, e incluso destruir el resto de los componentes y circuitos que se encuentran en el robot. Es por esta razón que deben tomarse los cuidados necesarios para trabajar con corrientes que se encuentren dentro de los límites establecidos.

Como se observa en la Fig. 6, se diseñó un circuito en el que toda la corriente consumida por el robot es sensada por una resistencia de potencia. De esta manera, se transforma la corriente en una tensión, que puede ser amplificada y comparada con una tensión de referencia, que nos permita conocer si el consumo de nuestro sistema es excesivo. En caso de necesitar interrumpir el funcionamiento de la Plataforma para resguardar la integridad del dispositivo, un relay controlado por un transistor desconecta la alimentación de la placa.

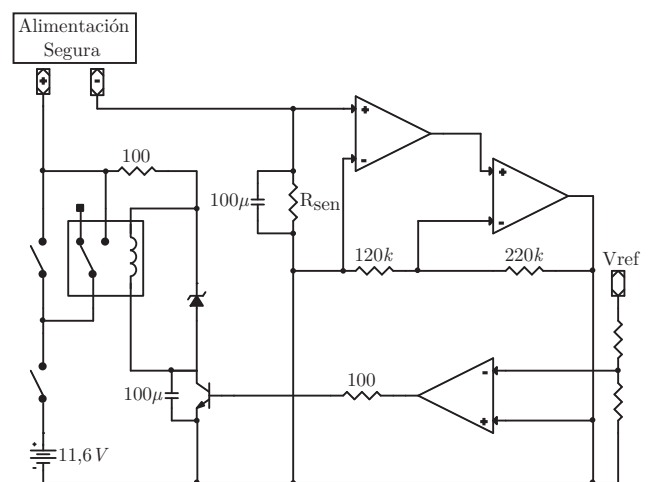


Fig. 6: Circuito protector de sobrecorriente.

III-C. Sistema seguidor de luz

Tener la capacidad de detectar una fuente lumínica, es una funcionalidad importante para cumplir nuestro objetivo de lograr que la Plataforma Robótica desarrolle trayectorias conocidas. Para este fin, se dispone de una luz led en la parte posterior, permitiendo definir

una trayectoria con una Plataforma y que otras puedan seguirlo en cadena. Si bien estos conceptos parecen muy específicos, se debe recordar que la idea era realizar un robot versátil que permita al investigador evaluar un amplio abanico de posibles situaciones.

El sistema desarrollado está basado en la utilización de 4 fotoresistencias o LDR que integran un sistema digital detector de luz (Fig. 7).

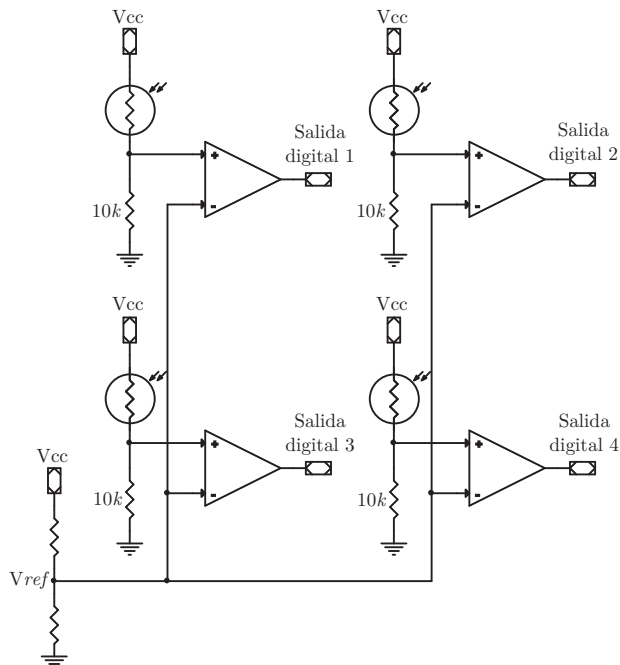


Fig. 7: Circuito controlador de los LDR.

La información que llega al microcontrolador es si cada fotoresistor recibe o no recibe luz. Por lo tanto, con 4 LDR, se tienen 16 combinaciones posibles que permiten detectar la procedencia de la fuente lumínica. Conociendo la disposición de los sensores, el usuario puede realizar un código que modifique la velocidad de los motores dependiendo de la combinación binaria que indique el sistema de fotoresistores. De esta manera, puede lograrse que el robot siga la luz que lo está excitando o, de lo contrario, se aleje de ella.

En la implementación, se construyó una base para los sensores que abarca un rango de 180° (Fig. 10) en donde cada LDR cubre un campo de 45° . Mediante distintos experimentos realizados en ambientes controlados (habitaciones con poca luz, que permitieran concentrar al máximo la fuente que se deseaba medir) se logró que la Plataforma detectara y siguiera luces que se encontraban alejadas hasta dos metros de distancia.

III-D. Sistema infrarrojo para medición de distancia

Es fundamental para este tipo de sistemas automáticos, tener la capacidad de detectar objetos que puedan interponerse en su trayectoria. Mejor aún si se puede conocer la distancia a la que se encuentra el obstáculo, para tomar diferentes decisiones en base a esta información.

Por estas razones, se eligieron los sensores infrarrojos medidores de distancia GP2Y0A02YK0F [8] y GP2Y0A21YK0F [9], los cuales se muestran en la Fig. 9. El primero de ellos tiene un rango de

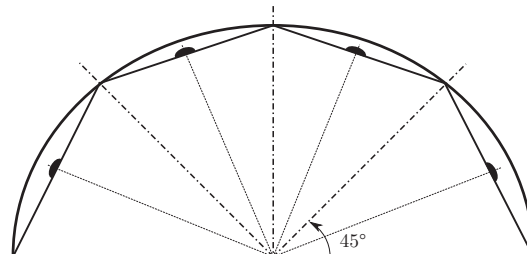
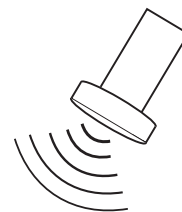


Fig. 8: Disposición de los LDR en el robot.

detección de 20cm a 150cm; se utilizaron dos de estos dispositivos, uno en la parte frontal y otro en la posterior. Por otro lado, el sensor GP2Y0A21YK0F tiene un alcance de 10cm a 80cm, menos preciso que el anterior, por lo que se usaron cuatro de ellos, colocados al costado del robot.



(a) GP2Y0A02YK0F



(b) GP2Y0A21YK0F

Fig. 9: Sensores utilizados en la implementación.

Lo que se buscó fue generar una estructura con los sensores distribuidos en los 360° del robot, para poder evitar choques en todas las direcciones. Si bien no puede cubrirse toda la circunferencia de la Plataforma porque los sensores son direccionales, se distribuyen de forma de optimizar la detección de obstáculos. Para cumplir esta condición, se utiliza una estructura hexagonal (Fig. 10), colocando, como ya hemos mencionado, los sensores de mayor rango en el frente y en la parte posterior, y completando los costados con los sensores de menor alcance. Esta elección fue hecha por una limitación en los costos del proyecto; lógicamente sería mejor utilizar la mayor cantidad de sensores, con el mayor rango y precisión posibles.

Con el fin de contar con una interfaz gráfica, se implementó un script en python que permite conocer la distancia medida por cada uno de los sensores infrarrojos. Básicamente, el script recibe los datos enviados por la UART de la placa, y los representa en tiempo real en un diagrama polar utilizando Matplotlib [10], tal como se observa en

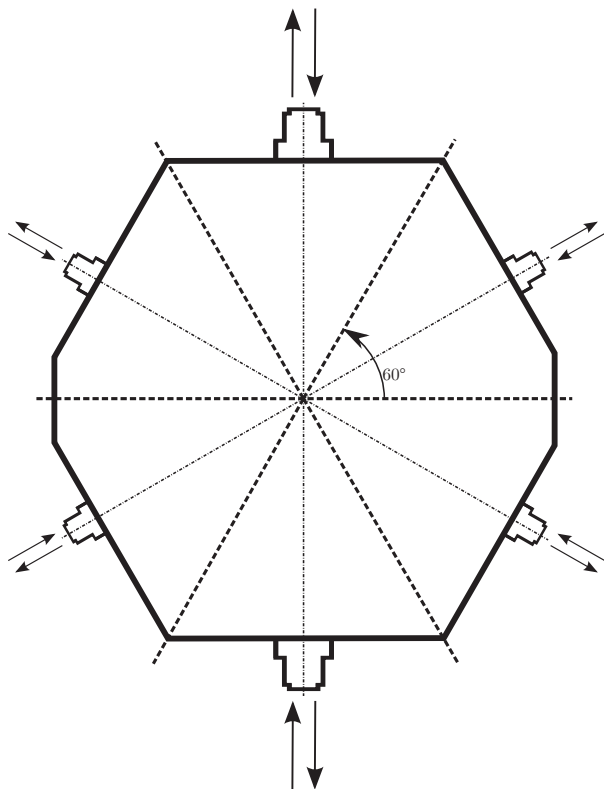


Fig. 10: Disposición de los sensores en la carcasa del robot.

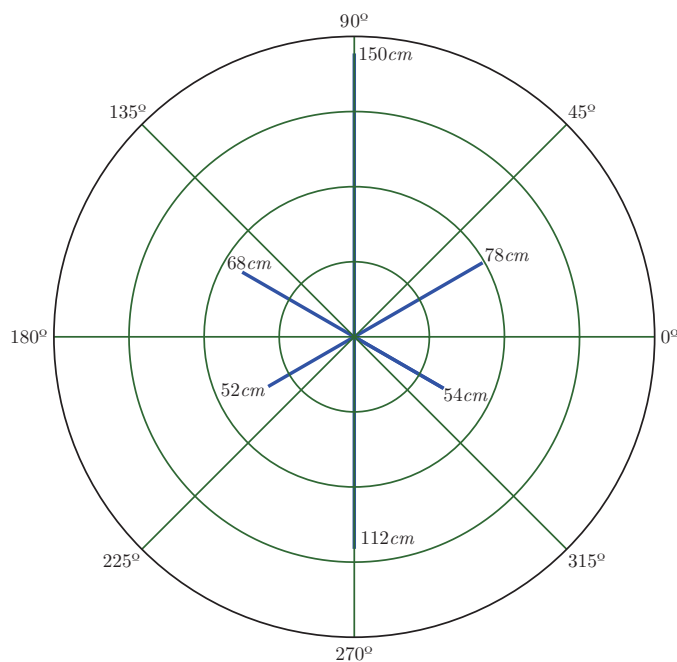
la Fig. 11a.

El algoritmo de control encargado de la detección y evasión de obstáculos está basado en el concepto de campos potenciales artificiales. Para que el robot pueda moverse libremente por el entorno esquivando obstáculos, se lo consideró como una carga inmersa en un campo potencial. Sabemos que dos cargas de la misma polaridad experimentan una fuerza de repulsión, inversamente proporcional al cuadrado de la distancia que las separa, de acuerdo a la Ley de Coulomb:

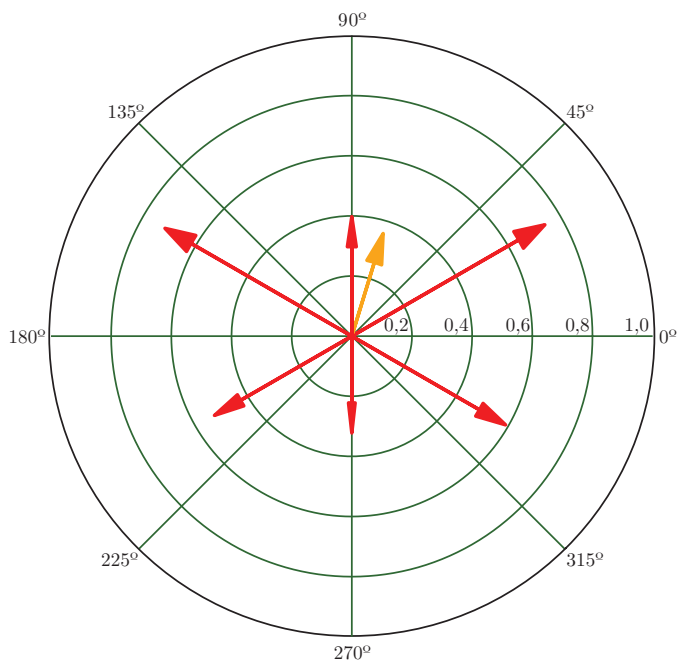
$$F = k \frac{q_1 \cdot q_2}{d^2}$$

Por lo tanto, si se considera que el robot y su entorno, están cargados en forma hipotética por cargas de la misma polaridad, puede suponerse que los obstáculos ejercerán una cierta fuerza de repulsión sobre el robot. Este método, denominado comunmente de campos potenciales, le permite al robot girar y avanzar sin colisionar. Para ello, con cada una de las distancias leídas de los sensores infrarrojos, se calcula una fuerza de repulsión artificial, en la dirección del sensor. Una vez estimadas las fuerzas individuales, se obtiene la fuerza resultante que moverá al robot. La Fig. 11b, muestra como nuestro script grafica las fuerzas mencionadas. En rojo se observan las distintas fuerzas producidas por cada sensor, mientras que en naranja se presenta la resultante.

La fuerza resultante puede interpretarse como un par de fuerzas, una fuerza lateral, aplicada transversalmente al robot que lo hace rotar, y una componente a lo largo de la dirección del movimiento que lo hace avanzar o retroceder.



(a) Distancia del robot a los objetos detectados.



(b) Vectores potenciales artificiales y su resultante.

Fig. 11: Interfaz de la aplicación desarrollada.

IV. ESQUEMA DE PROGRAMACIÓN

A la hora de realizar la programación, se buscó organizar el código en bibliotecas vinculadas entre sí de acuerdo al esquema de

la Fig. 12. Allí se representa la jerarquía que forman las bibliotecas, generando distintas capas de abstracción. Esto permite que el usuario de la Plataforma pueda realizar una aplicación sencilla y en alto nivel, sin preocuparse por configuraciones propias del dispositivo.

En la parte superior del esquema encontramos nuestro software de aplicación que puede ser, por ejemplo, un programa que le ordene a la Plataforma seguir el contorno de una habitación delimitada por paredes, como se muestra en la Fig. 2. Este es el más alto nivel de programación y es el único que compete al usuario, por lo que debe ser simple y fácil de programar.

En la segunda capa se encuentran las bibliotecas programadas durante el desarrollo del proyecto, las cuales configuran los módulos presentes en la Plataforma y realizan un procesamiento primario de la información recolectada por los sensores. De esta forma el usuario recibe, por ejemplo, la distancia a la que se encuentra el objeto detectado por los sensores infrarrojos, en lugar de un simple número, resultado de la conversión analógica/digital de la tensión entregada por el sensor, que sólo tiene significado alguno para quién conoce las especificaciones del sensor.

La siguiente capa incluye las bibliotecas CMSIS y MKL46Z4, que intervienen directamente en el mapeo de la memoria del microcontrolador, y facilitan la programación del mismo en lenguaje C. Aquí, se declaran cada uno de los registros y banderas, y se implementan macros para la configuración de los mismos. Estas bibliotecas formaron la base de nuestro proyecto, facilitando la programación de las capas superiores.

Finalmente, nos encontramos con la capa de hardware que representa la comunicación del microcontrolador con los distintos bloques físicos que integran la Plataforma.

El objetivo de separar el código en bibliotecas es, en primer lugar, individualizar la función de cada uno de los circuitos exteriores al micro, y posteriormente formar una capa de abstracción que le facilita al usuario acceder directamente a la medición de los sensores y al control de los motores.



Fig. 12: Esquema de bibliotecas implementadas.

V. CONCLUSIONES Y TRABAJO A FUTURO

La Plataforma Robótica Móvil se encuentra actualmente disponible en los laboratorios del C.I.I.I. Su bajo costo de producción permite que sea tenida en cuenta en el estudio de grandes redes de sensores móviles que precisan de un gran número de plataformas móviles para el transporte de sus nodos sensoriales. Los materiales baratos, la disponibilidad de los mismos, sumados a la simplicidad de construcción, permiten la reproducción de esta Plataforma sin mayores esfuerzos.

Por otro lado, distintos tipos de ensayos pusieron bajo prueba su capacidad de desplazarse en espacios reducidos, esquivando obstáculos y describiendo distintas trayectorias programadas en el dispositivo. Estos ensayos otorgaron resultados positivos que demuestran el cumplimiento de las condiciones que se impusieron al comienzo del proyecto.

Nuevos objetivos han sido planteados con el fin de mejorar el funcionamiento y desempeño de la plataforma. Una de las nuevas propuestas consiste en dotar al dispositivo con un módulo de comunicación inalámbrica, como por ejemplo los módulos que utilizan el protocolo ZigBee, de forma de poder consultar constantemente el estado de los sensores disponibles en la plataforma. Además, se planea mejorar el desempeño de los algoritmos que procesan la información proveniente de los sensores infrarrojos y las fotorresistencias, con el objetivo de aumentar la precisión de las acciones de control que se basan en estos datos recolectados.

VI. AGRADECIMIENTOS

Se agradece a los estudiantes Daniel Marcheti, Fernanda Suarez y Javier Ramos por el diseño y construcción de la estructura del robot y por su trabajo sobre los servomotores, que formaron la base de nuestro proyecto. Además, agradecemos a los estudiantes Cristian Cavenio, Gabriel Infante, Federico Diaz Baez, Jeremías Baez Carvallo y a los ingenieros Diego Gonzalez Dondo y Gonzalo F. Perez Paina, por su ayuda para la construcción y corrección de este paper.

REFERENCIAS

- [1] *FRDM-KL46Z User's Manual*, Freescale Semiconductor Inc., Microcontroller Solutions Group.
- [2] *KL46 Sub-Family Reference Manual*. Document Number: KL46P121M48SF4RM, Rev. 3, July 2013. Freescale Semiconductor Inc., Microcontroller Solutions Group.
- [3] Gonzalez Dondo, D.; Toloza, J. H.; Canali, L. R., *Aplicación de un filtro de partículas distribuido para el seguimiento de objetivos en el espacio mediante múltiples observaciones angulares*. Actas de la VIII Jornadas Argentinas de Robótica (JAR), 2014.
- [4] Mondada, F.; Bonani, M.; Raemy, X.; Pugh, J.; Cianci, C.; Klapotcz, A.; Magnenat, S.; Zufferey, J.-C.; Floreano, D. and Martinoli, A. (2009) *The e-puck, a Robot Designed for Education in Engineering*. Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions, 1(1) pp. 59–65.
- [5] <http://veterobot.com/>
- [6] <http://swarmrobot.org/>
- [7] Perez Paina, G.F.; Araguas, R.G.; Gaydou, D.A.; Steiner, G.M.; Canali, L.R., *RoMAA-II, an Open Architecture Mobile Robot*. Latin America Transactions, IEEE (Revista IEEE America Latina) , vol.12, no.5, pp.915,921, Aug. 2014
- [8] *Sensor Infrarrojo medidor de distancia, GP2Y0A02YK0F Datasheet*. Sharp
- [9] *Sensor Infrarrojo medidor de distancia, GP2Y0A21YK0F Datasheet*. Sharp
- [10] <http://matplotlib.org/>

Design of DTMB IP Modules

Ernesto Fontes Pupo¹, Reinier Díaz Hernández², Leandro Cruz Reygada³

LACETEL

No. 34515, Independencia Ave., km 14, 1ro de Mayo Area, Boyeros, Havana, Cuba

¹ fontes@lacetel.cu

² reinier@lacetel.cu

³ leandroc@movitel.co.cu

Abstract—Design and implementation of IP modules for modulators equipment according to the DTMB standard is a key element to achieve the technological independence. This article focuses on the design of two IP Modules: ASI-SPI Converter LCT51263 and Frame Header Generator LCT63125, and therefore was obtained a new version of the IP Module DT-1008-A1 which performs the channel coding and modulation according to DTMB standard. The technological assimilation of the original DT-1008-A1 included a deep study of the characteristics of Digital Terrestrial Television (DTT), DTMB standard and Altera's devices and design tools. These results contribute to achieve a national solution for a DTMB Modulator. The implementation of the proposed results was made over FPGA technology (Cyclone IV) using Verilog as Hardware Description Language and the Quartus II and ModelSim software as design tools. All results were validated through the analysis of the Compilation Results and the Physical Implementation in a DTT system.

Keywords— Intellectual Property Modules, FPGA Cyclone IV, Verilog, DTMB.

I. INTRODUCTION

In Cuba does not exist a national solution for the design and implementation of DTMB (Digital Terrestrial Multimedia Broadcasting) modulators for Digital Terrestrial Television (DTT). Therefore LACETEL, Research and Development Telecommunications Institute have as a goal the development of a national solution. As results of previous works, have been developed several blocks of the DTMB modulator as: Scrambler, Channel Coding, Interleaving, Symbol Mapping and OFDM (Orthogonal Frequency Division Multiplexing) modulation.

To contribute to this previous research and advance towards the conformation of a prototype is necessary to work in the remaining blocks of the DTMB modulator structure.

Through a donation from Chinese government LACETEL has the IP (Intellectual Property) Module DT-1008-A1. This describes the operation of DTMB modulator and is programmed in Verilog® Hardware Description Language. The needs to understand this code emerges as fundamental step in the process of technological assimilation. Thus the objective of this research is: To assimilate the technologies for the design and implementation of IP modules: ASI-SPI Converter and Frame Header (see Figure 1.), using Verilog® Hardware Description Language (HDL) and FPGA (Field

Programmable Gate Array) technology Altera Cyclone IV and associated tools.

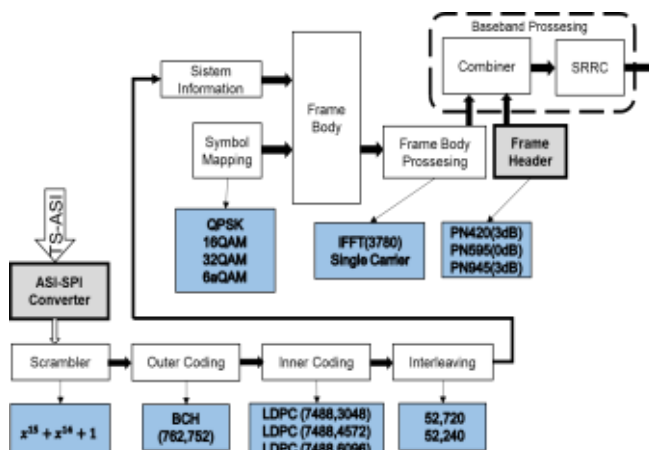


Fig. 1. General scheme of the DTMB Standard Modulator.

II. TECHNOLOGICAL ASSIMILATION

A. Functional block diagram of DT-1008-A1IP Module

The design is described in Verilog at Behavioral level. It is implemented over a Cyclone IV EP4CE115 Altera FPGA employing its libraries to make several functions.

Figure 2 shows general scheme of the IP module, seen to the left and right the inputs and outputs respectively. The ASI (Asynchronous Serial Interface) signal is a serial inputs of MPEG-2 (Moving Picture Experts Group) data. It is processed before pass through FPGA by CY7B933 (Reception block) integrate circuit which realize serial to parallel conversion, 27 MHz clock generation and generation of important signals for the next blocks.[1]

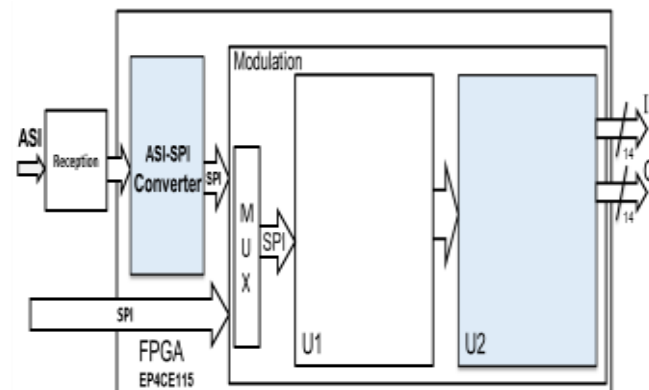


Fig. 2. General scheme of the Modulator.

The FPGA also includes an SPI (Synchronous Parallel Interface) input that does not require previous processing. Modulation block was designed to receive in SPI format so

that, is necessary ASI-SPI Converter block. It has a multiplexer (MUX block) wish select the respective inputs.

The main modulation functions are made in the U1 and U2 blocks. The functions of U1 are: Bitrate Adaptation, Scrambler, Channel Coding, Time and Frequency Interleaving, Symbol Mapping, OFDM Modulation and System Information insertion. The U2 Module is responsible to generate the Frame Header through different Pseudo Noise (PN) Sequences, conform the Signal Frame by multiplexing in time domain the Frame Header and Body (generated in U1), may insert two pilots to help in the tuner at the receiver and process the signal in baseband using SRRC (Square Root Raised Cosine) filter to adjust the output signal to a 6 MHz band width. Figure 3 show its blocks diagram. The outputs of the system are two 14 bits signals calls I (In phase) and Q (In Quadrature), these are the result of the modulation process.

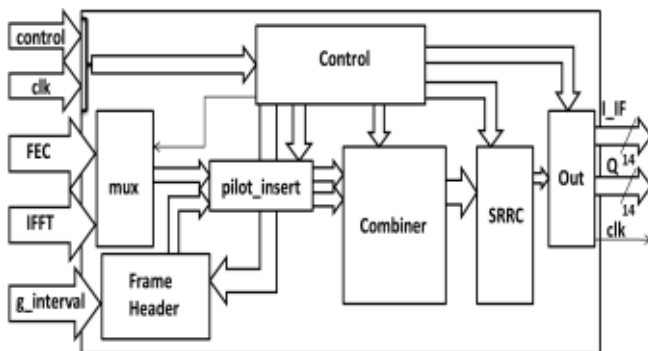


Fig. 3. U2 block diagram.

The multiples inputs of the U2 block were grouped together by functionality in order to facilitate the understanding of the diagram. FEC signal include data and control signals corresponding to single carrier modulation modes. IFFT signal also include data and control signals corresponding to the multicarrier modulation modes. Both signals have 13 bits with symbol rate of 5.67 Msps for 6 MHz channel bandwidth. The clk signal includes three synchronous references at 5.67 MHz, 11.34 MHz (2x) and 22.68 MHz (4x). They are employed by different functions of the module.

III. RESULTS

After the technological assimilation process of the DT-1008-A1, were designed two IP modules: ASI-SPI Converter LCT51263 and Frame Header Generator LCT63125.

A. ASI-SPI Converter- LCT51263

LCT51263 convert Continuous mode ASI data to SPI format. Provides a quick sync with the MPEG-2 data in accordance with ISO/IEC 13818-1 standard. This module accepts 188 and 204 bytes packets and its compatible whit SFN (Single Frequency Network) networks. The main applications of these IP module are: DTT Modulators, Decoders and Multiplexers of MPEG-2-TS signals. Its block diagram is shown in Figure 4.

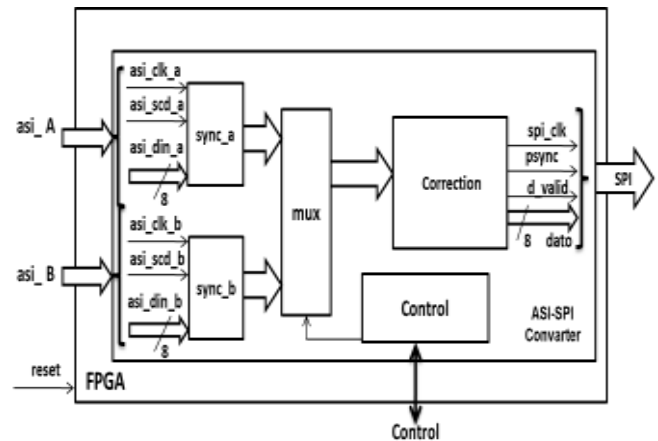


Fig. 4. LCT51263 block diagram.

It has two independents inputs: ASI-A and ASI-B, both with their signals preconditioned (by CY7B933). ASI interface has a constant transmission speed of 270 Mbps and their symbols are encoded in 10 bits for a ratio of 1:10 [2], consequently the symbols clock: asi_clk_a and asi_clk_b has 27 MHz frequency. The signals asi_scd_a and asi_scd_b indicates when there is any special character in the data signals asi_din_a and asi_din_b. The sync_a block handles this signals and ensure the correct synchronism with the structure of the MPEG2-TS package.

The out3 block work with Continuous mode ASI data and in SFN and MFN networks. Its main functions are adjust the duty cycle of the output SPI clock to ensure that be almost 50%, adjust the clock and data to corresponds with real bitrate and generate the four signals of the SPI format (Figure 5, 6 and 7 [2]).

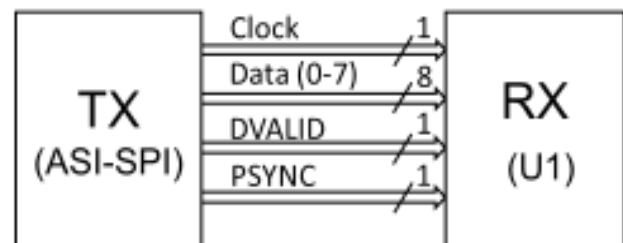


Fig. 5. System for SPI transmission.



Fig. 6. Transmission format with 188 bytes packets.

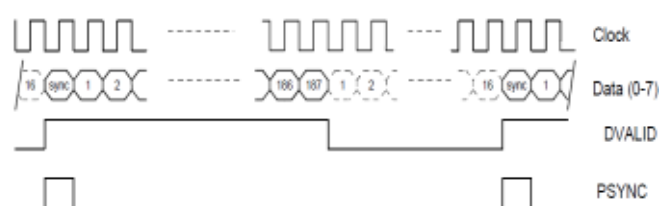


Fig. 7. Transmission format with 204 bytes packets (188 data bytes and 16 dummy bytes).

Figure 8 shows the interfaces of LCT51263.

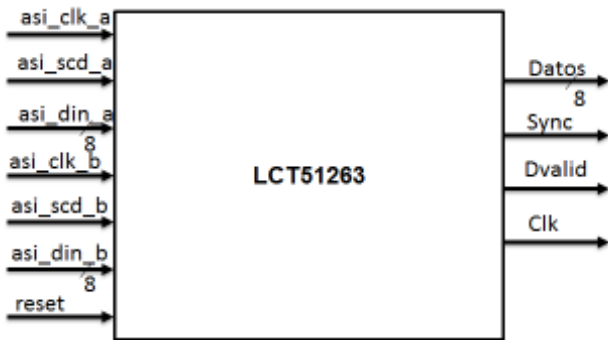


Fig. 8. Interfaces of LCT51263.

B. Frame Header Generator- LCT63125

LCT63125 was designed for DTMB standard. It has three LFSR (Linear Feedback Shift Register) to generate three different PN sequences (420, 595 or 945 symbols) with BPSK modulation. The generated Frame Headers are multiplexed in time with the Frame Body information in accordance with DTMB standard. The PN sequences of length 420 and 945 can generate the same header in all signal frames within a Super Frame or always different. Work at frequencies of 5.67 MHz, 6.615 MHz and 7.56 MHz and has 13 bits to the output. The main applications of these IP module are: DTMB Modulators. Its block diagram is shown in Figure 9.

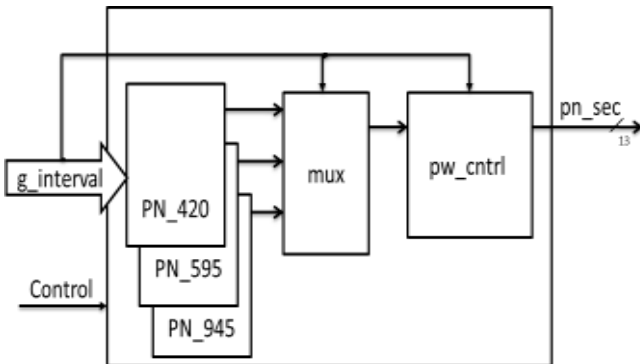


Fig. 9. LCT63125 block diagram.

The signal *g_interval* includes all the information to conform the Frame Header. For example, select one of three different Frame Headers of the standard (PN-420, PN-595, and PN-945), select if its level will be equal or 3 dB bigger than Frame Body signal. Select static or variable initial condition of the PN sequence.

Figure 10 shows the internal block diagram of the PN_420 block. The main difference between the blocks PN_420, PN_595 and PN_945 is the internal LFSR block. Equation 1, 2 and 3 shows the LFSR polynomial for each case. [3]

For PN_420:
 $G255(x) = 1 + x + x^5 + x^6 + x^8 \dots \dots \dots (1)$

For PN_595:
 $G1023(x) = 1 + x^3 + x^{10} \dots \dots \dots (2)$

For PN_595:
 $G511(x) = 1 + x^2 + x^7 + x^8 + x^9 \dots \dots \dots (3)$

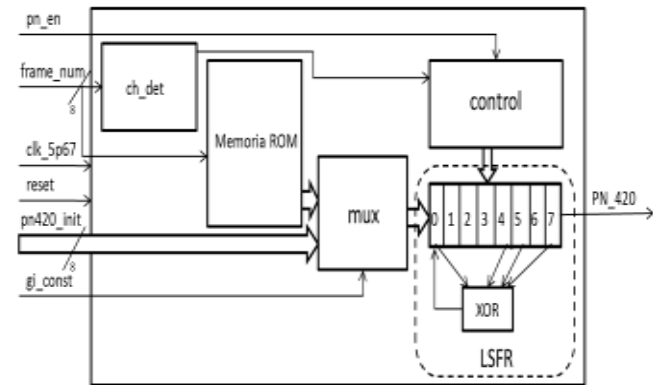


Fig. 10. PN_420 block diagram.

The structure of the three PN sequences are in accordance with DTMB standard.

The control of the average power ratio between frame header and frame body is accomplished by pw_cntrl block. It transform the one symbol bit of PN sequence to 13 bits symbols, matching with the long of Frame body symbols. After that, this block may transmit the PN sequence with the same average power of Frame body or twice as much [3]. All this functions are doing in accordance with DTMB standard.

Figure 11 shows the interfaces of LCT63125.

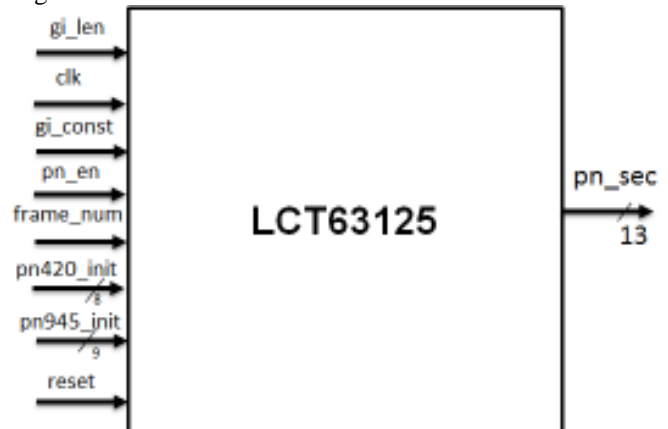


Fig. 11. Interfaces of LCT63125.

IV. VALIDATION OF THE RESULTS

To validate the results two main methods are applied. Compilation Results Analysis: FPGA Resource Utilization, Timing Reports and Power Consumption Reports. The second method is Physical Implementation by integrating to a practical system of DTT and Spectral and Temporal Analysis.

During the application of these validation methods a comparative analysis were made between the original design (IP module DT-1008-A1) and the proposed IP module DT-1008-A1 with all our results embedded.

A. Compilation Results

The FPGA Resource Utilization specifies the quantity of EP4CE115 Cyclone IV FPGA resources that have been used. This test tells us whether the proposed design can be implemented within the available device. A comparison between the total resource usages by the different functional blocks is shown in TABLE I. The tests demonstrate the validation of the results in terms of resource utilization.

TABLE I
COMPARISON BETWEEN THE TOTAL RESOURCE USAGE ON
FPGA AND BETWEEN FUNCTIONAL BLOCK.

Design	Dedicated Registers		Memory Bits		Combinational Functions	
	O	P	O	P	O	P
Total	25207	23785	905796	762932	54062	52325
ASI-SPI	1553	147	139264	0	1834	238
U1	12270	12264	607248	607248	29693	29679
U2	7509	7500	9248	5648	17323	17207

O: Original P: Proposed

The Timing Report shows the maximum frequency for each base system clock. This lets us know if the design is capable of operating in the necessary works frequencies. The maximum frequency is calculated taking into account the paths where sources, registers and destination ports are handled by the same clock. Paths with different clocks, including generated clocks are ignored in the calculation [4, 5]. A comparison is shown in TABLE II in terms of maximum frequencies. The values are higher in all cases than those required for the system clocks.

TABLE II
MAXIMUM FREQUENCIES COMPARISON.

Clock	Maximum Frequency (MHz)	
	original	Proposed
clk_7_56	33.44	34.27
clk_15_12	54.74	55.92
clk_30_24	70.0	65.76
clk_60_48	79.43	81.45

Power consumption Report allows to estimate the power consumption of the implemented design to ensure that the thermal and power consumption requirements of the FPGA are not violated. The accuracy of the results is $\pm 20\%$. Therefore the manufacturer Altera does not recommend using these results as specifications, the purpose of the assessment is help to establish a reference to the power requirements of the design. It is important to check the power consumption during operation of the device to take into account environmental conditions. [6]

There are two types of analysis: thermal analysis, which gives the energy dissipated as heat from the FPGA and power analysis, which gives the current required to operate the FPGA. [6]

TABLE III shows a comparison of dissipated energy as heat in the FPGA. The results show similar values for both designs demonstrating that the thermal parameters are not altered with the changes.

TABLE IV shows a comparison of the current consumed by the FPGA. The required minimum current is less than 500 mA that is capable to deliver the selected source. The analysis of the results shows a decrease in current demand of the proposed design which validates its implementation.

TABLE III
COMPARISON IN TERM OF THERMAL ANALYSIS.

Design	Thermal Power (mW)	Dynamic Power (mW)	Interconnection Power (mW)	Static Power (mW)
Original	101.97	11.91	22.30	67.76
Proposed	99.92	10.78	22.68	66.28

TABLE IV
COMPARISON IN TERM OF POWER ANALYSIS.

Source	Design	Total current (mA)	Dynamic current (mA)	Static current (mA)	Minimum current (mA)
DH1715A-5 (500 mA)	O	18.32	3.19	15.14	18.32
DH1715A-5 (500 mA)	P	18.01	2.87	15.14	18.01

O: Original P: Proposed

B. PHYSICAL IMPLEMENTATION

The goal to perform the Physical Implementation of the proposed design is to validate their correct operation in the practice. For that reason Spectral Analysis, Temporal Analysis and integration to TDT system were performed.

The integration to a transmission system allows us to verify that the proposed design is completely functional and compatible with the rest of the system. Figure 12 show a block diagram of the measurements set employed. The computer (PC) generates a video encoded in MPEG-2 format and transmits it through DVB-ASI interface. This signal is received by the development board where the FPGA is programmed with the proposed design. The UP-converter receives the signal delivered by the FPGA and moves it to the desired frequency band. A commercial STB decodes the digital signal to be displayed by the television (TV). With the correct representation of the signal was shown that the proposed design works right.

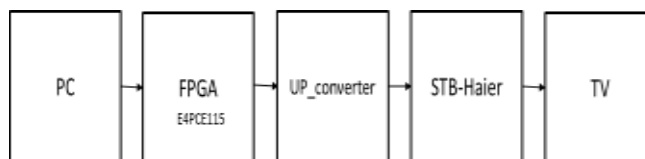


Fig. 12. Block diagram of the Measurement Set.

For Spectral and Temporal analysis was measured the output of the UP-converter with U3771 Advantest Spectrum Analyzer and with MSO7104B Agilent Technologies Oscilloscope. It has obtained the results shown in Figures 13 and 14 respectively. This test was made following the Standard GB 20600-2006 and the Recommendation ITU-R BT.1206-2. [3, 7]

The aim of the spectral analysis is test that the DTMB spectral signal generated by the proposed version of DT-1008-A1 fallow the requirements of the DTMB standard.

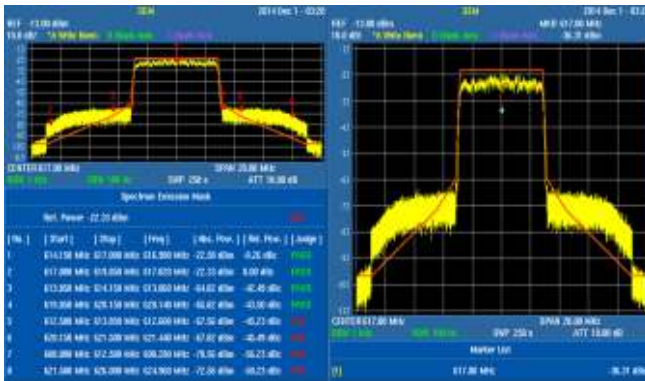


Fig. 13. Spectral Analysis.

The aim of the temporal analysis is test that the DTMB signal generated by the proposed version of DT-1008-A1 follow the requirements of the DTMB standard for time domain. Proving that the generated PN sequence of the IP module is correct.

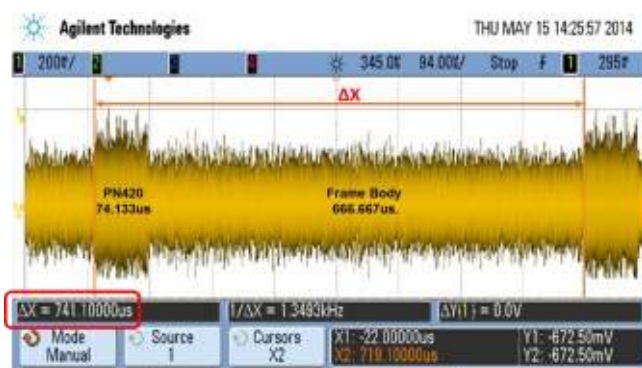


Fig. 14. Timing Analysis.

CONCLUSIONS

The present paper describes the technological assimilation of IP Module DT-1008-A1 which describes the operation of DTMB modulator. As result of that two IP modules ASI-SPI Converter LCT51263 and Frame Header

LCT63125 were designed and inserted into the original DT-1008-A1. Therefore a new version of the DT-1008-A1 Module was archive with better resource utilization and power consumption. The operation of the proposed design was validated by analyzing the Compilation Results and by Physical Implementation.

The main result of this research was the understanding of the internal work of DTMB modulator. Allowing introduce change into the internal performance of it and in the future may have a national solution of the DTMB modulator.

Later works of this research will be focuses on the rest of internal modules to the DT-1008-A1.

REFERENCES

- [1] T. UNIVERSITY, "DTMB (GB20600-2006) Modulator FPGA Code DT-1008-A1" 2012.
- [2] E. STANDARD, "Cable network for television signals, sound signals and interactive services," ed, 2002.
- [3] C. N. Standard, "GB 20600-2006," ed. China, 2007.
- [4] A. Corporation, "Timing Analysis Overview," ALTERA, 2013.
- [5] A. Corporation, "The Quartus II TimeQuest Timing Analyzer," ALTERA, 2013.
- [6] A. Corporation, "PowerPlay Power Analysis and Optimization FAQ " ALTERA, 2012.
- [7] ITU-R, "BT.1206-2: Spectrum limit masks for digital terrestrial television broadcasting," ed, 2014.
- [8] I. STANDARD, "ISO/IEC 13818-1," ed, 2000.
- [9] IEEE, "IEEE Standard Verilog® Hardware Description Language," ed, 2001.
- [10] IUT-R, "Spectrum occupancy measurement," 2011.
- [11] C. Zhang, "Technical Review for DTMB Standard and System," 2014.

Diseño y evaluación de un Sistema de Adquisición y Almacenamiento de Datos Ambientales con Transmisión Satelital

Bruno Díaz*, Emanuel Díaz*,
Hugo Pereyra*, Ana Diedrichs*

* GridTICs – Grupo UTN de I&D en Tecnologías de
la Información y las Comunicaciones
Departamento de Electrónica
UTN Facultad Regional Mendoza
Rodríguez 273, Capital – Mendoza
ana.diedrichs@gridtics.utn.edu.ar

Gustavo Mercado*§

§ CONAE - Comisión Nacional de Actividades Espaciales
Belgrano Oeste 210 - Capital - Mendoza
mercado@conae.gov.ar

RESUMEN – Se describe el diseño, implementación y evaluación de un sistema de adquisición automática de datos ambientales con transmisión satelital. El sistema ha sido desarrollado para su uso en ambientes agrestes, de difícil acceso y con escasos medios de comunicación tradicional.

Los datos ambientales se digitalizan a través de un circuito específico cercano al sensor y se envían mediante el protocolo bus serie inspirado en SDI-12 a la plataforma colectora de datos (DCP). La comunicación digital permite extender la longitud de los cables de los sensores más de 10 veces. El sistema cumple la función de un data logger con almacenamiento en tarjeta de memoria SD y a través del sistema DCS (Data Collection System) se envían los datos al satélite Argentino de observación climática y oceanográfica SAC-D de la CONAE. Esta característica, permite utilizar al sistema en lugares alejados sin la necesidad de presencia de un operador y/o investigador. El sistema está preparado para servir de ayuda a la investigación biológica del bosque patagónico austral por investigadores del Instituto Argentino de Nivología, Glaciología y Ciencias Ambientales (IANIGLA).

Palabras Clave - Remote sensing - Data logger - Data Collection System - SDI-12 - Satélite SAC-D Aquarius

I. INTRODUCCIÓN

Actualmente la colecta de datos ambientales en zonas remotas carentes de conectividad se realiza frecuentemente con “data loggers” importados, de alto costo y con almacenamiento in-situ. Los datos son recuperados luego de un período de tiempo, dependiendo del acceso al lugar remoto, en forma manual. El proyecto viene a dar una solución a los problemas planteados, ya que se pretende desarrollar un producto nacional que permita cumplir con las funciones de un “Sistema de Adquisición de Datos” (DAQ) comercial y al mismo tiempo contar con la posibilidad de recuperar los datos al momento de su producción. Esto último se realiza por medio de un enlace

de comunicaciones satelital que mejora la disponibilidad temporal, permitiendo adelantar estudios y análisis ambientales.

El desafío es diseñar un sistema tan robusto como los extranjeros y acondicionar los sensores provistos por IANIGLA (Instituto Argentino de Nivología, Glaciología y Ciencias Ambientales) [1].

El conjunto de sensoramiento del sistema consta de un medidor de radiación solar (piranómetro), medidor de velocidad del viento (anemómetro) y medidor de dirección de viento (veleta), La información recolectada se digitaliza por medio de un circuito cercano al sensor y se transmite, al sistema central, mediante el protocolo SDI-12 (Serial Digital Interface - 1200 bps). Este método permite ampliar la longitud del cableado a más de 10 veces su longitud real, que para estos sensores es de 2,5 m, y permite además que la unidad central detecte qué sensor se ha conectado como se muestra en la figura 1. En el módulo de adquisición de datos (DAQ) debe recibir los datos de los sensores y almacenarlos en una memoria flash SD en un archivo de formato Excel (.xls) indicando la fecha, hora - proporcionados por un Real Time Clock (RTC)- y valor de la muestra. Luego estos datos se pueden visualizar mediante la memoria SD en cualquier computadora.

Para la transmisión satelital se usa el Sistema Satelital Argentino de Recolección de Datos Ambientales (SSARDA) [2], que es un sistema en el que se recolectan datos ambientales transmitidos por plataformas autónomas denominadas DCP (Data Collection Platforms). Éstas pueden ser fijas o móviles; pueden encontrarse en la superficie terrestre, sobre boyas en los océanos y ríos, en globos, entre otros. Los mensajes transmitidos a intervalos regulares son procesados en vuelo por el receptor DCS (Data Collection System Instrument) el cual forma parte de la misión SAC-D (Satélite Argentino de Aplicaciones Científicas D) de la CONAE [3], Los datos extraídos son almacenados junto con la telemetría y finalmente transmitidos a la Estación Terrena Córdoba (ETC), para su posterior procesamiento y distribución.

Para poder transmitir los datos satelitalmente, se debe establecer una comunicación entre la plataforma transmisora de datos (PTT) y el DAQ, para lo cual es necesario la adaptación de niveles de señal y el establecimiento de una sesión de comunicación mediante un protocolo ad-hoc [2].

Las DCPs transmiten un mensaje a intervalos prefijados, sin ningún tipo de interrogación por parte del satélite. Todas las DCP pueden acceder al canal sin ningún tipo de coordinación entre sí. Teniendo en cuenta que la tasa de bit es de 400 bps, que el largo de los datos de ciencia varía entre 32 bits (mínimo) y 256 bits (máximo) y que además se transmite un protocolo de inicialización para sincronización e identificación, la duración del mensaje transmitido varía entre 360 ms y 920 ms. Debido a la alta sensibilidad del receptor en vuelo, la potencia media de salida de los transmisores de las plataformas es de 30 dBm.

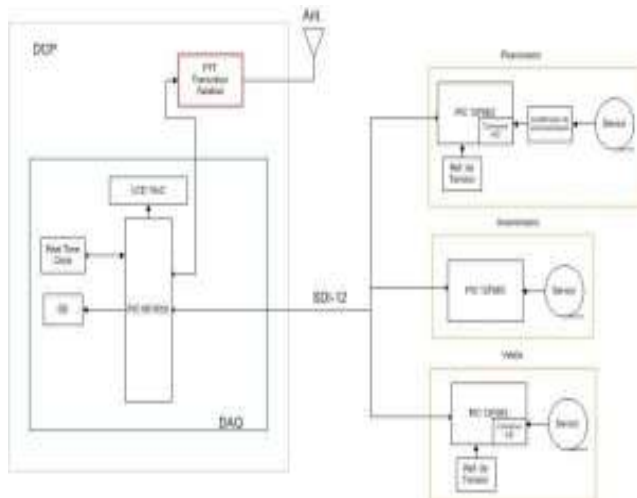


Fig. 1. Grafico esquemático de Sistema de Adquisición y Almacenamiento de Datos

II. EL SISTEMA

a. Objetivo Principal

Implementar sistema de colecta de datos ambientales [4] de bajo consumo, robusto y de altas prestaciones para uso como herramienta de investigación de fenómenos ambientales y biológicos.

b. Objetivos Secundarios

- Estudiar, adaptar e implementar interface de sensores inteligentes basada en norma SDI-12
- Implementar interface con protocolo DCP del sistema DCS.
- Estudiar e implementar procedimientos de bajo consumo.
- Realizar pruebas de verificación y validación a campo

c. Descripción de los Sensores

Piranómetro [5] (Star Pyranometer Mod3040-A WeatherTronic): Medición de radiación solar total ó radiación difusa (W/m²), Internamente compuesto por 72

termocuplas de Cu-Constatán.

Veleta (Micro Response Vane Mod 2020 Weathertronic): Medición del ángulo de desplazamiento horizontal del aire debido a diferencia de presiones atribuido a diferencia de temperaturas.

Anemómetro [6] (Micro Response Anemometer Mod 2030 WeatherTronic): Medición de la velocidad del viento (km/h). Internamente, el instrumento posee una rueda ranurada solidaria al eje de rotación con 30 ranuras y medición mediante fototransistor. La frecuencia de salida del anemómetro puede ser determinada por (1).

$$FHZ = 6.33 (VEL [KM/H] - 0.52) \quad (1)$$

d. Interface SDI-12

SDI-12 [7] es un protocolo estándar para interconectar registradores de datos con sensores basados en microprocesadores. Sus siglas significan interface serial/digital en 1200 baudios. Está diseñado para aplicaciones que requieran operación a batería con consumo mínimo, que sea de bajo costo, que use solo registrador de datos con múltiples sensores en un solo cable (bus) y que pueda tener una distancia de hasta 50 metros entre un sensor y un registrador de datos. Otra ventaja, es la posibilidad de extender el módulo a futuro añadiendo más funcionalidades de SDI-12 para interoperabilidad con sensores comerciales.

Para esta aplicación se implementaron un sub-conjunto básico de comandos que permiten consultar cada sensor por su dirección y obtener las mediciones. También se adaptaron algunas otras especificaciones para la aplicación: como la tensión de la línea (5 V) y la velocidad de transmisión (9600 baudios).

e. Interface Satelital DCS

El DAQ debe establecer una interface con la estación de transmisión satelital PTT. La comunicación se establece a través de una línea RS 232.

Los datos que se quieren transmitir al satélite deben organizarse en tramas específicas que cumplan con las normas [8] de CONAE para este tipo de comunicación. La máxima cantidad de información soportado por el DCS es de 256 bits por trama. El SAC-D asegura la recepción de dos tramas por día, por lo que se deberá organizar la información temporal para cubrir un rango de 12 horas.

Para nuestro sistema se implementa una estrategia de tomar datos cada 4 horas de todos los sensores. Cada trama contiene las últimas 12 horas adquiridas.

f. Almacenamiento is-situ

Para el almacenamiento de datos de la Plataforma Adquisidora de Datos se utilizó una memoria flash del tipo Secure Digital (SD). A esta memoria se accede por medio del estándar de comunicación serial SPI (Serial Peripheral Interface). Elegimos, para nuestra aplicación, una memoria SD Clase 4, con un rango de temperatura de operación de -25 °C a 85 °C, recordando que el dispositivo estará ubicado

a la intemperie. Se seleccionó una memoria SanDisk clase 4, que de acuerdo al fabricante ofrece características robustas tales como: a prueba de agua, golpes y de rayos X.

g. Microcontrolador

Para este sistema se utilizaron dos microcontroladores PIC (Microchip Technology Inc. [9]), uno de ellos para manejo de las interfaces de los sensores y otro para el adquisidor de datos.

h. Procesador para la Interface de Sensores.

Para todas las interfaces de los sensores se utilizó el microcontrolador PIC 12F683 [10], con un encapsulado superficial tipo soic y que posee las siguientes características más importantes: arquitectura de 8 bits, 6 pines de entrada/salida, 1na @ 2.0 V de consumo de energía en standby y 100 μ a @ 1 MHz, 2.0 V en operación típica, tensión de operación de 2 V a 5.5 V, conversor a/d de 10 bits de resolución y temperatura de operación de -40 a +125 °C.

Este micro-controlador tiene la tarea de medir una variable eléctrica proveniente de un transductor, aplicar una conversión matemática para obtener el valor de ingeniería de a la variable climática en cuestión y enviar esta medición mediante el protocolo serial SDI-12 a la plataforma adquisidora de datos, cuando se le sea solicitado.

Es necesario un pin (vref) para conectar una referencia de tensión externa para mejorar la precisión del conversor a/d en el caso de los sensores que hagan uso del mismo. (Piranómetro y veleta) más adelante se amplía información sobre la referencia de tensión utilizada.

La resolución del conversor analógico digital – 10 bits - es suficiente para obtener una adecuada resolución/precisión y rango dinámico para la digitalización de los valores tanto del piranómetro como de la veleta. Para las mediciones del piranómetro es necesario amplificar la fem generada por las termocuplas. Esto es realizado por el amplificador diferencial de instrumentación AD-627 [11]

A las razones antes citadas hay que agregar que debido a que el sistema completo estará alimentado con baterías es muy importante que sea del menor consumo posible, por lo que el micro-controlador elegido posee tecnología nanowatt.

i. Procesador para el Adquisidor de datos

Para el adquisidor de datos se utilizó el micro-controlador PIC18F8722 [12] con encapsulado smd (superficial), con las siguientes características: arquitectura de 8 bits, frecuencia de reloj máxima de 40 MHz, tamaño de memoria del programa de 128 KB, tamaño de ram de datos de 4 KB, 70 pines de entrada/salida, tensión de operación de 4.2 V a 5.5 V, conversor a/d de 10 bits de resolución y 16 canales y temperatura de operación de -40 a +125 °C.

Este micro-controlador se seleccionó por poseer una memoria flash de tamaño suficiente para contener el código objeto. Como características especiales se pueden mencionar también la salida pwm, de control de brillo de la

pantalla y 5 temporizadores.

j. Sistema de Alimentación

El sistema se diseña para mantener la operación del aparato en zonas sin fuente de energía externa. Por lo que se selecciona el uso de paneles solares con fuente principal de energía. El sistema se complementa con el uso de baterías, para la provisión de energía en situación nocturna y/o con niveles de radiación reducidas. El sistema se completa con un elemento cargador de batería,

k. Panel Solar

El panel solar utilizado es un Módulo Fotovoltaico Policristalino de alto Rendimiento de la empresa SOLARTEC modelo KS12T [13] que tiene una potencia nominal de 12 W, suficiente para alimentar a todo el sistema.

Para que este panel pueda cargar la batería en forma correcta se utiliza un Regulador de Carga marca SOLARTEC modelo SR04 [14] con las siguientes características: tensión de trabajo de 12 V, corriente de carga de 4 A, compensación por temperatura, protecciones contra corto circuito (fusible eléctrico), sobre tensión y corriente inversa, rango de temperatura de operación de -40 a +50 °C.

El conjunto panel solar-regulador de carga se encarga de mantener el nivel de carga de la batería para que el equipo pueda operar en los momentos que no hay sol o que la radiación solar es muy baja debido a la nubosidad del ambiente.

Para cargar la batería cuando el equipo no se encuentra instalado en campo se seleccionó un cargador de baterías marca Rippleless, el cual posee corte automático y evita accidentes o mal uso relacionados al exceso de tiempo de carga. Este cargador posee las siguientes características: tensión de alimentación de 220 Vca y 50 Hz, potencia de 15 W, tensión de salida de 13.7 Vcc y corriente de salida de 1 A.

l. Batería

Se seleccionó una batería de gel de 12 v 4.5 Ah de la marca Probattery [15], cuyas características principales son: tensión nominal de 12 V, tasa de descarga nominal de 20 horas, rango de temperatura de descarga de -15 °C a 50 °C.

El objetivo de esta batería es el de mantener una reserva de energía suficiente para que el equipo pueda operar en aquellos momentos en los que el panel solar no pueda proveerla; por ser de noche o porque el día se encuentra nublado.

Para el régimen más exigente, es decir, tomando muestras cada 1 minuto se calcula el consumo de A/h.

El consumo promedio total es:

$$C.P. = \frac{(480 \text{ s} \times 45 \text{ mA/h}) + \left(40 \text{ s} \times 300 \frac{\text{mA}}{\text{h}}\right) + (3080 \text{ s} \times 26 \text{ mA/h})}{3600 \text{ s}} = 31,57 \text{ mA/h}$$

Donde:

Consumo de adquisición (Co-Adqu): Para 60 muestras en una hora, con una duración 8 s por muestra, el consumo es de 45 mA.

Consumo de transmisión (Co-Tx): El PTT envía tramas cada minuto y medio, es decir 40 transmisiones en una hora con una duración de 1 s c/u a un consumo de 300 mA.

Consumo stand-by (Co-Sby): El tiempo restante (3080 s) el sistema está en modo stand-by con un consumo de 26 mA.

El cálculo de la autonomía de batería se calcula según la ecuación de Peukert [16].

$$t = \frac{20}{\left(\frac{0,03157 \times 20}{4,5 Ah}\right)^{1,1}} = 173,47 h$$

Considerando el escenario de mayor consumo, con intervalos de toma de muestras cada 1 minuto (máximo tiempo de muestreo requerido), se obtiene un tiempo de funcionamiento con baterías de 7 días 5 horas y 30 minutos, Por lo que se asegura que el sistema operará adecuadamente durante todo el periodo nocturno, que es el momento donde no se recibe carga del panel solar.

m. Descripción del Firmware

El firmware del sistema se divide en el producido para el sensor SDI-12 y el del DAQ. Cada interfaz de sensor posee un firmware especializado que se encarga de tomar la medición del sensor correspondiente, por medio de un convertor A/D, para luego calcular la ecuación de conversión y así obtener el valor de ingeniería y finalmente realizar el protocolo SDI-12 (lado sensor) para enviar estos datos hacia el DAQ. Este parte sigue los pasos del protocolo y se encarga de dar respuesta a éste. Los datos que se envían es un valor numérico de 0 a 1023 que corresponde a un convertor analógico digital de 10 bits. Este valor será procesado en el registrador de datos.

El firmware del DAQ se encarga de implementar el protocolo SDI-12 (lado DAQ) que lo comunica con los sensores. La tarea es direccionar el sensor y requerir los datos correspondientes, en una secuencia acorde al tiempo de muestreo requerido. Los valores sensados son almacenados en la memoria SD. Además despliega un menú en el display LCD donde el usuario puede configurar el periodo de muestreo, hora, fecha, obtener una lectura instantánea para el chequeo del correcto funcionamiento de los sensores y también permite observar el nivel de carga de la batería.

Finalmente este firmware también se encarga de controlar, mediante el puerto RS-232, al transmisor satelital para generar y cargar las tramas de datos para enviar al satélite. En la Fig. 2 se puede observar el diagrama de flujo del firmware del DAQ.

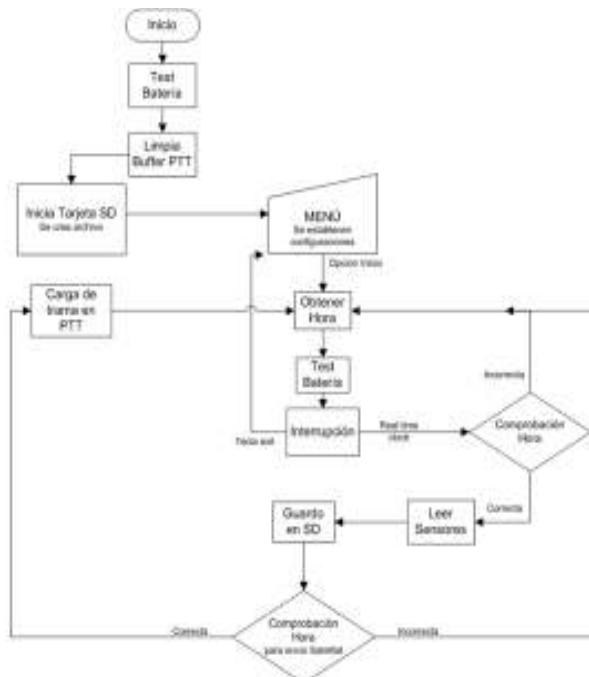


Fig. 2: Diagrama de flujo de software del DAQ

n. Recuperación de datos

El sistema consta de dos formas de recuperar la información, uno vía memory stick SD y la otra por medio del satélite SAC-D.

En el primer caso, el sistema graba la información recolectada en la memoria SD en formato Excel. Para obtener la información se deberá extraer la SD y procesarla en un PC. Si bien la maniobra de recuperación parecería inconveniente, la información es grabada en SD a la máxima velocidad de muestreo, por lo que resulta de mayor regularidad que la enviada satelitalmente. En la Fig. 3 se observa una muestra de los datos almacenados en la memoria SD.

	A	B	C	D	E
1	SADASAT				
2	Fecha	Hora	Velocidad [km/h]	Dirección [°]	Radiación [W/m2]
3	22/12/2013	13:25:57	5206 S		977
4	22/12/2013	13:26:57	4093 E		975
5	22/12/2013	13:27:57	5110 E		974
6	22/12/2013	13:28:57	3124 S		971
7	22/12/2013	13:29:57	13222 S		971
8	22/12/2013	13:30:57	3136 S		972
9	22/12/2013	13:31:57	5348 N		975

Fig. 3: Tabla de datos almacenados en memoria SD

o. Sistema de Recuperación de datos Satelitales

Una vez que el sistema DCS embarcado en el satélite Aquarius SAC/D recibe los datos enviados por las plataformas terrestres, los almacena en la memoria masiva. Cuando el SAC/D entra en contacto con la estación terrena de Falda del Carmen (ETC) “baja” los datos del DCS para su posterior procesamiento y distribución.

La CONAE provee a los usuarios del sistema DCS un acceso remoto a sus bases de datos por medio de protocolo ftp. En tales bases de datos se encuentran las “tramas de datos” recolectadas y correspondiente a cada estación del usuario. Los datos recuperados serán, entonces, los mismos que los enviados por las plataformas y recibidas por el satélite. Solo se le agrega la marce de tiempo (OBT-On Board Time) [8], que indica el instante en que la trama fue recepcionada a bordo.

Se ha diseñado una aplicación que toma las tramas recuperadas y transforma los datos en valores de ingeniería. También tiene la función de detectar y eliminar tramas erradas y duplicadas y velar por la consistencia de los datos.

III. RESULTADOS

Es sistema se ha desarrollado y construido completamente. También se mantuvo en laboratorio para hacer pruebas funcionales y verificar que se cumplen las especificaciones de diseño.

Para las pruebas de campo, el DAQ y los sistemas sensores fueron montados cajas tipo IP-65, para su protección ambiental. Luego el conjunto – DAQ, sensores, panel solar y antena DCS- fue montado en trípode adecuado para separar los sensores del suelo, de acuerdo a las especificaciones del fabricante.

El sistema fue montado en sitio de prueba al aire libre, provisto y fiscalizado por el IANIGLA, donde se ha operado, sin suministro de energía externa, en forma continua por el espacio de varios meses y bajo distintas condiciones climáticas. Estas pruebas han sido satisfactorias por lo que se cumple el requisito de robustez ante condiciones adversas. En la Fig. 4 se muestra dicho montaje.

También se verificó el funcionamiento del transmisor satelital (PTT), enviando tramas de datos, que luego fueron recuperadas del sitio ftp de CONAE.

El resultado de la recuperación de las tramas, luego de la traducción y proceso de graficación, podemos observarlas en las Fig. 5 y 6.

En los gráficos se observa la estrategia de adquisición de datos para los datos satelitales. Se han establecido horas de muestreo en horarios específicos, estos son: 0:00 – 4:00 – 8:00 – 12:00 – 16:00 – 20:00 h.



Fig. 4: Sistema montado en campo de prueba

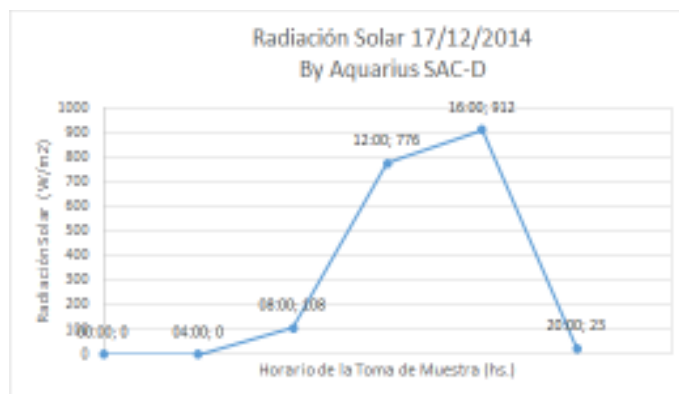


Fig. 5: Grafica de Radiación Solar (W/m^2)

Es decir que se toman 6 muestras diarias de radiación solar, de velocidad y de dirección de viento.

Esta estrategia se basa en la limitada cantidad de bits que posee la trama del DCS y la frecuencia de pasada del SAC-D.



Fig. 6: Grafica de Velocidad de Viento (Km/h)

Otro tipo de gráfica interesante es la de Fig. 7, donde se grafica la radiación solar a las 12:00 h desde el día 17/12/14 al 07/01/15. Estos tipos de gráficas pueden servir por ejemplo para comparar la radiación solar en las diferentes estaciones del año.

Una forma de aumentar la cantidad de datos adquiridos y por lo tanto la frecuencia de muestreo a más de seis tomas diarias, dentro de las restricciones del SAC-D, es realizar compresión de datos. En versiones posteriores del firmware se estudiará e implementará un método de compresión de datos, para la tramas DCS.

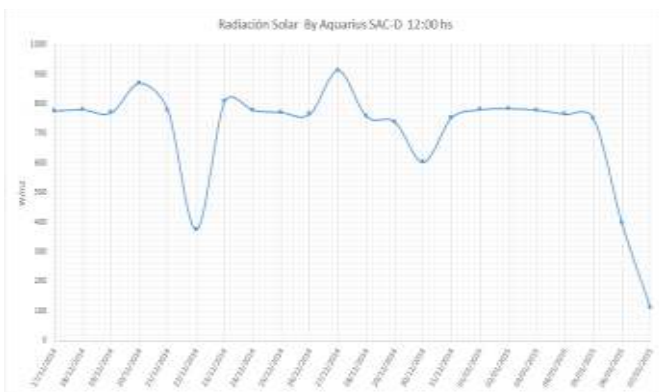


Fig. 5: Grafica de Radiación Solar (W/m²) mensual a mediodía.

IV. AGRADECIMIENTOS

El presente trabajo es realizado por el grupo UTN de I&D GridTICS [17] y está inserto en el proyecto de investigación acreditado por la Universidad Tecnológica Nacional código PID 25/J085 denominado “SIPIA - Estudio a campo de red de sensores inalámbricas en investigaciones agronómicas y biológicas” Se cuenta con la colaboración del Instituto Argentino de Nivología, Glaciología y Ciencias Ambientales (IANIGLA) [1] del Centro Científico Tecnológico de Mendoza CONICET. El IANIGLA contribuye con la disposición de diversos sensores y con los estudios de campo. La Comisión Nacional de Actividades Espaciales (CONAE) [3] contribuye con el dispositivo de transmisión satelital y la antena respectiva.

Por lo que se ha firmado un acuerdo de uso del Satélite SAC-D Aquarius entre la CONAE y la FRM de la UTN.

El proyecto cuenta con el financiamiento de la Secretaría de Políticas Universitarias del Ministerio de Educación por intermedio del programa de “Universidad, Diseño y Desarrollo Productivo” Primera convocatoria 2013 [18].

Se agradece la participación de la Lic Clara Pissolito, de IANIGLA, por la valiosa colaboración en la definición, especificación y pruebas del sistema.

V. REFERENCIAS

- [1] “Instituto Argentino de Nivología, Glaciología y Ciencias Ambientales (IANIGLA),” wiki.mendoza-conicet.gov.ar/index.php/IANIGLA, última visita: 25/03/2015
- [2] A Carlotto, J Juárez, G Sager, G Mercado, “Receptor para el sistema satelital argentino de recolección de datos ambientales” anales del ARANDUCON 2012 (Institute of Electrical and Electronic Engineers - IEEE, Sección Paraguay), 28, 29 y 30 de Noviembre de 2012 - Asunción Paraguay
- [3] “Comisión Nacional de Actividades Espaciales (CONAE),” www.conae.gov.ar, última visita 26/03/2015
- [4] H.R. Taylor “Data Acquisition for Sensor Systems”, Chapman & Hall, 2010
- [5] Piranómetro - es.wikipedia.org/wiki/Piranómetro, última visita 10/04/2015
- [6] Anemómetro - es.wikipedia.org/wiki/Anemómetro, última visita 10/04/2015
- [7] “SDI-12 A Serial-Digital Interface Standard for Microprocessor-Based Sensors Version 1.3”, SDI-12 Support Group, (Technical Committee), Online, http://www.sdi-12.org/current%20specification/SDI-12_version1_3%20January%202013.pdf , Last visit 13/04/2015
- [8] Adrián Carlotto, José María Juárez, Gerardo Sager, Gustavo Mercado, “SAC - D Data Collection System: Requerimientos Detallados e Interfaces de las PTT (Platform Transmitter Terminal)”, CONAE Document: SD-475-6326-CD RELEASE: V1, October 30, 2011
- [9] Microchip Technology Inc, <http://www.microchip.com/>, última visita 12/03/2015
- [10] “PIC12F683 Data Sheet”, 2007 Microchip Technology Inc.
- [11] “Micropower, Single- and Dual-Supply, Rail-to-Rail Instrumentation Amplifier” Data Sheet AD627, Analog Devices, 2013
- [12] “PIC18F8722 Family Data Sheet”, 2008 Microchip Technology Inc.
- [13] “KS12T - MODULO FOTOVOLTAICO POLICRISTALINO DE ALTO RENDIMIENTO”, SOLARTEC.
- [14] “Reguladores de Carga series SR, SRX y SC”, SOLARTEC
- [15] Probattery, <http://www.probattery.com.ar/>, última visita: 25/03/2015
- [16] Peukert's law - en.wikipedia.org/wiki/Peukert%27s_law, última visita 12/4/2015.
- [17] gridTICS UTN FRM, www.gridtics.frm.utn.edu.ar, última visita 12/4/2015.
- [18] Ministerio de Educación - Secretaría de Políticas Universitarias (SPU), “Programa de Universidad, Diseño y Desarrollo Productivo - Primera convocatoria 2013”, Resolución 4016 SPU, 9 Dic 2013.

Wireless Sensor Network for Monitoring Infrastructure Adjacent to Metro de Medellín Railway System

G.A. Meneses

Faculty of Engineering
University of San Buenaventura Medellín
Medellín, Colombia
gustavo.meneses@usbmed.edu.co

Abstract—Structural Health Monitoring (SHM) systems using Wireless Sensor Networks (WSNs) are common nowadays in infrastructure such as tunnels, buildings and highways. Information gathered from in-situ deployed sensors is useful for maintenance purposes, evaluating infrastructure response before particular situations and preventing catastrophic events, among others. The design and implementation of a wireless sensor network for monitoring infrastructure adjacent to Metro de Medellín railway system is presented. A wireless network constituted by a set of nodes involving 8-bit microcontrollers uses the IEEE 802.15.4-compliant MiWi Protocol for communications within the Personal Area Range. Measurement data can be retransmitted over longer distances, within the local area or metropolitan area range, through communication gateways connected to network coordinator.

Keywords—Structural Health Monitoring; Wireless Sensor Networks; MiWi Protocol; 8-bit Microcontroller; IEEE 802.15.4 standard

I. INTRODUCTION

Wireless sensor networks exhibit features that greatly favor the requirements of structural health monitoring systems. Continuous measurement derived from the sensors deployed over infrastructure works such as tunnels, railways, bridges, and buildings, among other, can be achieved by means of wireless networks grouping sensor nodes according to the selected topology [1]. Connecting wireless personal area networks with other communication networks, both wired and wireless, covering the range of local area, metropolitan area or wide area network by using gateway elements provides the reliability needed in order to provide a continuous service.

The railway system of the line A of Medellín Metro runs along a distance of about 20 kilometers in the Valley of Aburrá. Most of the line A of this metro system goes parallel to Medellín River. In order to monitor infrastructure adjacent to railway system to prevent landslide or similar events which could result in risk for metro passengers or stopping operation of this massive transportation system, a pilot project is currently under execution. This project is aimed at complementing current monitoring systems by installing a set of sensor nodes in points considered to be critical according to

previous analysis obtained from site inspections and historical data register analysis.

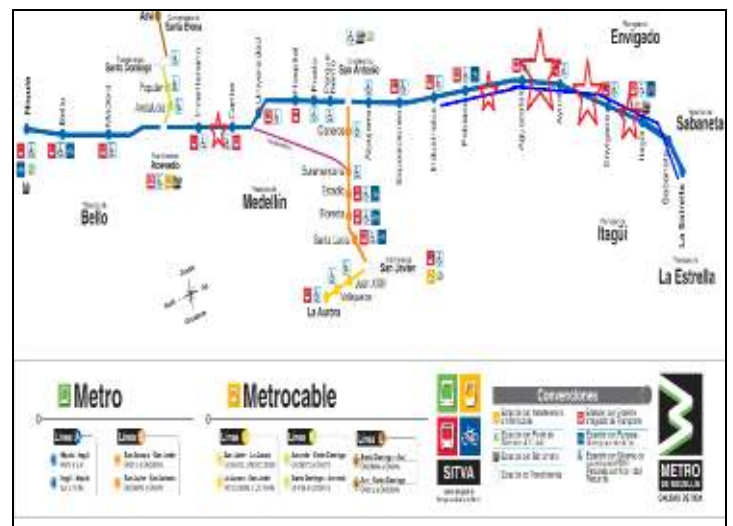


Fig. 1. Metro de Medellín line-A railway system (blue) and selected structural health monitoring points (red stars).

This structural health monitoring application requires unattended operation of nodes over long periods of time because of the conditions of places to be monitored and the restrictions to enter the areas adjacent to railway system during normal operation of trains. Therefore commercial off-the-shelf elements such as batteries and photovoltaic modules are to be used in order to provide maintainability and scalability to the proposed solution. Sensors and instrumentation boxes containing nodes circuitry are to be placed outdoors, therefore design considerations must be observed to fulfill these requirements. According to the performed analysis both wall inclinometers and borehole inclinometers will be installed to monitor the retaining walls adjacent to line A railway system and the ground surrounding them.

Fig. 2 shows the site of the line A selected to install the first nodes in order to conduct monitoring tests.



Fig. 2. Inspecting a selected monitoring place before installing the sensors for the structural health monitoring application under development.

This work is ordered as follows: first MiWi protocol is presented, then the proposed architecture for this particular application is described, network nodes design and code development issues are also described. Finally we discuss preliminary results and present conclusions.

II. THE MIWI PROTOCOL

A. Protocol Overview

The MiWi protocol is proprietary from Microchip [2], this protocol is IEEE 802.15.4-compliant and among their main features we have their ease of implementation on 8,16 or 32 bit microcontrollers, their light footprint regarding to program memory usage and the topology and variants (MiWi P2P, MiWi pro) offered to the developers of applications. MiWi variants like MiWi P2P offer different choices to application developers related to network topologies, maximum number of nodes and routing mechanisms.

B. MiWi P2P

This variant of MiWi protocol works in star and peer-to-peer topologies [3]. MiWi P2P function in single hop networks, therefore the maximum distance for connecting nodes is determined by the range of transceivers. Similarly to IEEE 802.15.4 Standard, we have in MiWi protocol Reduced Function Devices (RFD) and Full-Function Devices (FFD). The PAN Coordinator must be a FFD and, depending on the MiWi P2P topology, end devices can be either RFD or FFD. In star topology the end devices only can establish communication with the PAN Coordinator and this can establish with all the network nodes. Peer-to-Peer topology in MiWi P2P is more flexible than star topology, end devices can establish connection with other devices as shown in Fig. 3.

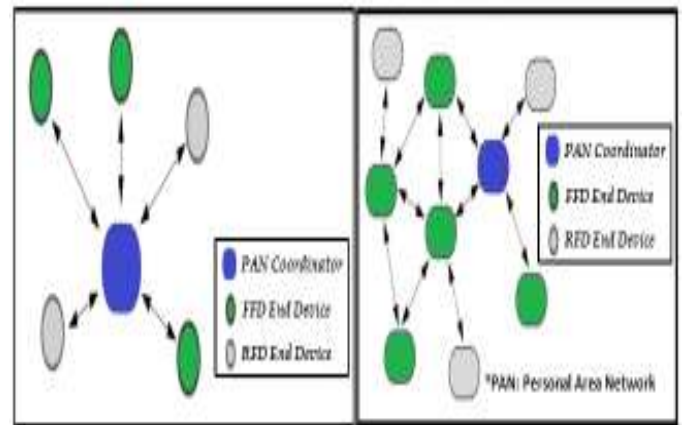


Fig. 3. Star (left) and Peer-to-Peer (right) topologies under MiWi P2P protocol

C. Network Addressing and Message Format in MiWi P2P

For networks under MiWi P2P protocol we have eight-field 64-bit addresses, starting with the number one for the PAN coordinator (for example 11-22-33-44-55-66-77-01), end devices continues the numeration (11-22-33-44-55-66-77-02, and so forth). The personal area network also have an identifier called PAN ID, the default is 1234. Short addressing is used only for broadcast messages [3].

III. PROPOSED ARCHITECTURE

A three layer architecture is proposed in order to develop an application of continuous structural health monitoring. This architecture is conceived to allow accessing collected data at any of the layers: at the lower layer, which is called Field Datasink, at the intermediate level (Local Datasink) or at the upper level (Global Datasink). Accessing data at any of the three layers makes easier conducting tests while implementing the network, network maintenance becomes also easier because datasink are not centralized and provides some flexibility to the monitoring application in aspects such measurement visualization, storage, analysis and processing.

The lower layer, called Sensor Node Layer, is based on a wireless sensor network which uses MiWi P2P to collect data from the field sensors deployed on the infrastructure adjacent to railway system. The PAN coordinator belongs to the intermediate level, it combines the management of sensor node communications with linking connection towards the so-called Ground Station which will be located at a distance around 1 to 2 kilometers. This network coordinator uses gateways which transfer on-field collected data to other data sinks by using other personal area networks protocols, such as IEEE 802.15.1, or Local Area Network protocols (IEEE 802.11 or M2M wireless protocols). At the upper level connection to metropolitan, local or wide area network is available by using GSM modems, TETRA standard radio modems [4], Ethernet gateways or similar devices.

Fig. 4 shows an overview of the proposed architecture. Details are given regarding communication protocols and technologies used at each layer.

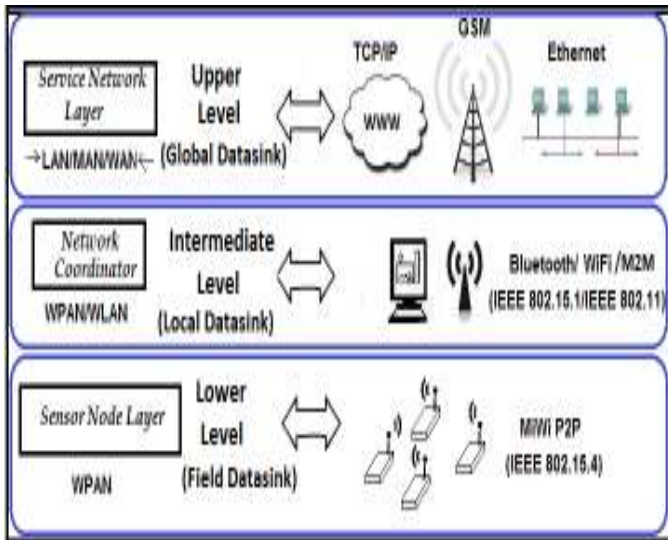


Fig. 4. Three-layer proposed architecture for developing an application of structural health monitoring over infrastructure adjacent to railway system.

Two major concerns arise when planning this application: energy consumption and outdoor long-term functioning.

A. Energy consumption

Long term energy autonomy is a required performance feature when developing an application like the above described. However achieving low consumption of sensor nodes depends on many factors like circuit design, electronic component specifications and firmware programming strategies. For the proposed architecture energy consumption is critical at the lower and at the intermediate layer, therefore traditional energy harvesting means like solar photovoltaic support are to be used in order to increase sensor nodes and coordinator operating time.

B. Outdoor long-term functioning

Outdoor applications involving wireless sensor network pose additional challenges for designers. Weather, environmental and external agent (human beings or animals) influence must be considered in order to achieve continuous and reliable operation over long term time periods. Sun, rain, dust, moisture, vibration and similar factors could affect node performance. Circuit component performance could become degraded because the influence of temperature changes, furthermore rain and similar weather conditions could affect communications, mainly for sensor nodes operating in the 2.4 GHz ISM band. Sensor placement and enclosure design must be conceived considering extreme conditions and favoring maintainability issues and potential external agent damage or operation interference. This issue mainly is crucial for communication elements belonging to the lower and intermediate layer of the proposed architecture.

IV. NODE DESIGN

The nodes for the personal area network are based on Microchip's PIC18F4620 microcontroller [5]. The

PIC18F4620 is an 8-bit microcontroller which can execute the code to perform MiWi communications and additionally perform data logging/acquisition tasks.

A. Network Coordinator

Because of the number of nodes to be deployed over the field star topology has been adopted, therefore the coordinator will initialize the network and sensor nodes only can communicate with it. The coordinator operates continuously in order to collect the data sent from sensor nodes, eventually transmit alarms and report data to the ground station.

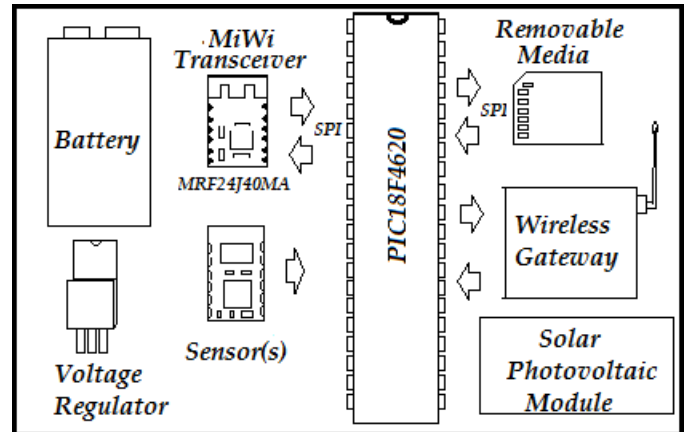


Fig. 5. Elements constituting the network coordinator to be deployed on the measurement field.

The elements constituting the network coordinator are listed in Table I. Actual costs in Colombian local market are shown with the price expressed in United States dollars (\$USD).

TABLE I. NETWORK COORDINATOR ELEMENTS

Element	Main Features	Cost (USD)
MiWi transceiver MRF24J40MA [6]	Operates in the channels 11 to 26 of the 2.4GHz band. Range of about 120 m with line of sight. SPI driven	10
Battery	12 V rechargeable dry battery	16
Voltage Regulator [7]	3.3V dc-dc converter step down (high efficiency)	16
Sensors	Wall/borehole Inclinometers	258/895
Removable Media (plus holder)	SPI driven SD card	8
Wireless Gateway [8]	RS232 interface radio modem	185
Solar Photovoltaic Module (plus charger controller)	12 V, 55 watt	126
TOTAL COST		\$1541 USD

Fig 6. shows some of the elements of the network coordinator



Fig. 6. Wireless gateway and other elements of the network coordinator.

B. Network nodes

On the field we will have sensor nodes and the PAN Coordinator (Network Coordinator). Sensor nodes have a structure similar to that shown in Fig. 5, but they do not include the wireless gateway. The solar photovoltaic module and the removable media could be present or not depending of particular application requirements. Sensors like inclinometers are common in structural health monitoring applications [9][10]. We also show in Fig. 7 the mediating node that provides connectivity towards service networks. At the ground station we have the opportunity to use 120 Vac power supply and eventually connection points to IEEE 802.11-based networks.

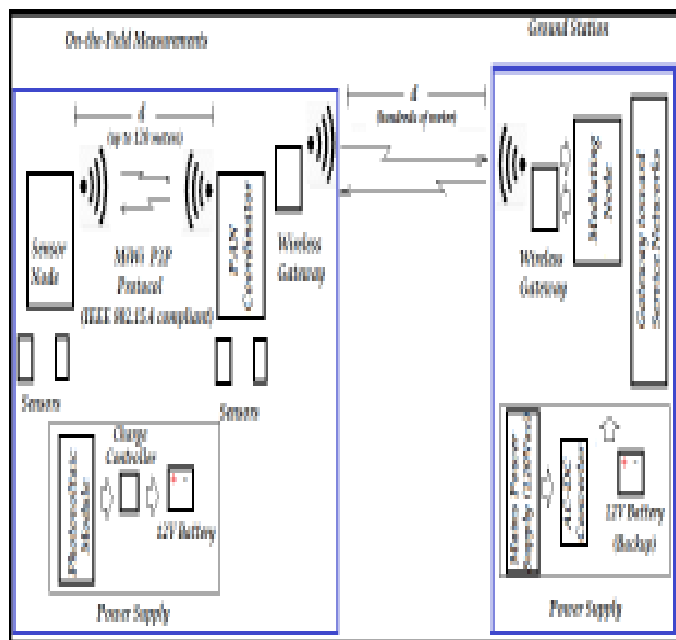


Fig. 7. Overview of the network nodes to be deployed on the field and of the network node at the intermediate layer.

V. CODE DEVELOPMENT AND CONDUCTED TESTS

Two MPLAB projects were developed in order to perform tests involving at least a sensor node and the network coordinator. Microchip's MiWi Application Programming Interface (MiApp) [11] functions were included into sensor node firmware in order to get better performance results from energy consumption point of view. Sleep mode was enabled for both transceiver and microcontroller and the payload was better used because original function MiAppWriteData() only transmitted one character each time. Application developer can also use Microchip MiWi Media Access Controller (MiMAC) [12] functions to access special features at a lower level.

In order to illustrate the overall structure of the code executed by the network coordinator, Fig. 8 shows the flowchart of the source file containing the main routine in the MPLAB project.

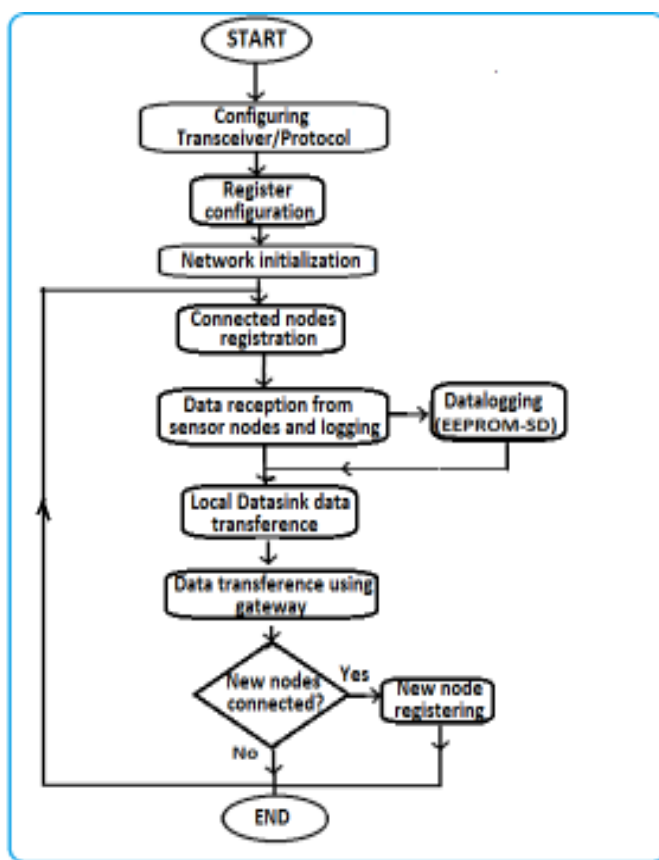


Fig. 8. Flowchart of the source file containing the main routine for the Network Coordinator.

Programming nodes for this structural health monitoring application was successfully achieved by customizing MiWi stack reference libraries and files provided by Microchip. Complementary for-free available resources like MPLAB IDE and C18 compiler from this manufacturer were also used for developing the code and compiling the firmware to be executed by the microcontroller governing the nodes.

TABLE III. MIWI TRANSCEIVER COMPARISON




MiWi Transceiver	Advantages	Constraints
MRF24J40MA 	Widely available in local market Economical (10 dollar approx.)	Range is limited to about 120 meter
MRF24J40MB 	Range is about 1.2 kilometers Printed antenna	Discontinued (end of life declared by manufacturer) Unstable functioning when paired with MRF24J40MA
MRF24J40MC 	Range is about 1.2 kilometers	Higher cost Require external antenna Higher power consumption



Fig. 9. Sensor node and the mains-powered specially developed PAN coordinator to conduct continuous operation tests.

Three types of batteries were tested while a sensor node periodically sent accelerometer data during several hours until packet transmission stopped because of battery run out. The sensor node during the tests only received power from the tested batteries and the network coordinator was powered from mains supply, as shown in Fig. 9, in order to continuously count packet reception. The sensor node tested had a current consumption of about 30mA during transmissions and 6.1mA during standby periods when both, transceiver and microcontroller, were put into sleep state.

TABLE II. CONTINUOUS OPERATION CONDUCTED TESTS

Tested Battery	Packets received	Hours of operation
12V rechargeable dry battery	10800	150
3.7V li-ion rechargeable battery (a pair of batteries connected in series was used)	296	5.5
8.4 V Ni-MH rechargeable battery	570	9

Range tests were also conducted in order to establish the maximum distance covered by the transceivers chosen for this application. MRF24J40MA was the choice made for covering wireless personal area network communication at the lower layer of the proposed architecture. Other transceiver options like MRF24J40MB and MRF24J40MC were considered but performance, technical and economic issues were not enough advisable taking into account project particularities.

Tests conducted in the University with two nodes having MRF24j40MA allowed reaching links with ranges of about 100 meters. The range of the MRF24J40MA transceivers were also tested in the field. A maximum distance of about 100 meters was also reached. As shown in Fig. 12, two persons verified the transmitted packets arrival. Passing trains do not seem to have influence on packet reception while the performed tests.



Fig. 10. Range measurement test, for MRF24J40MA transceiver, conducted on one of the field places selected for structural health monitoring.

Currently this project is under development. All the transducers to be installed and the communication gateways to be used are defined. We are now testing the nodes with the elements to be used on the field, such as the wall inclinometer

for the sensor node, as shown in Fig. 11, and the wireless 900 MHz modem for the network coordinator.

The conducted tests will allow better knowing node's circuitry performance and improving design issues regarding the switching of high-energy consumption elements and resource multiplexing (like SPI interfaces). Additionally, temperature variations will show their influence on battery discharge characteristics [13].



Fig. 11. Basic sensor node element arrangement to operate on the field.

Enclosures to be installed on the field are under study and design adaptations will be done. Details related to data accessing at the different data sink levels are under adjustment.

VI. CONCLUSIONS

We have presented our approach for developing an application of structural health monitoring of infrastructure adjacent to line-A railway system of the Metro de Medellín. Our approach is based on the adoption of the IEEE 802.15.4-compliant MiWi protocol which can be implemented on 8-bit microcontroller without requiring paying royalty fees for network functioning. We also propose a three-layer architecture, with a wireless sensor network at the lower level, for covering application communications and data sink requirements.

After the conducted tests power consumption of network nodes continues to be a major concern. This because of the consumption of the special transducers to be installed on the field and those of complementary elements, such as the wireless gateway to be used to transmit wireless area network data to ground station. Further testing must be conducted with the full system deployed over the field and designs adjusts must be needed.

A continuous operation of about six days and six hours was achieved when used the traditional valve-regulated lead-acid 12V rechargeable battery. On the field performance must be evaluated when installed inclinometer sensor on the field

because conditions will vary because of the current consumption of these devices.

An approximate cost estimation has been performed in order to project the overall cost of developing network nodes for this monitoring application. Sensor node cost is lower than estimated network coordinator cost. Installation and maintenance costs are to be estimated as the project evolves.

ACKNOWLEDGMENT

We wish to thank to the Metro de Medellín Ltda. and to the University of San Buenaventura Medellín for supporting this project, code CN20150040. The engineer Valerio Quintero Arboleda of the Research and Development Department of Metro de Medellín Ltda. supported much of this work.

REFERENCES

- [1] Fraser M., Elgamel A., He X., y Conte J. Sensor Network for Structural Health Monitoring of a Highway Bridge. *Journal of Computing In Civil Engineering Asce/ January/February 2010 / 11.*
- [2] Microchip Technology inc. Microchip MiWi™ Wireless Networking Protocol Stack [Online]. Available: <http://ww1.microchip.com/downloads/en/AppNotes/AN1066%20-%20MiWi%20App%20Note.pdf>
- [3] Microchip Technology inc. Microchip MiWi P2P Wireless Protocol [Online]. Available: <http://ww1.microchip.com/downloads/en/AppNotes/01204B.pdf>.
- [4] M. Smolnikar, A. Hrovat, M. Mohorčič, I. Ozimek, T. Celcer, G. Kandus, and I. J. Stefan, "Telemetry and Telecontrol over Tetra Network," *Inf. MIDEEM*, vol. 38, pp. 61–68, 2008.
- [5] Microchip Technology inc. PIC18F2525/2620/4525/4620 Data Sheet 28/40/44-Pin Enhanced Flash Microcontrollers with 10-Bit A/D and nanoWatt Technology [Online]. Disponible: <http://ww1.microchip.com/downloads/en/DeviceDoc/39626e.pdf>
- [6] Microchip Technology inc. MRF24J40MA Data Sheet 2.4 GHz IEEE Std. 802.15.4™ RF Transceiver Module [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/70329b.pdf>
- [7] TracoPower. DC/DC-Konverter TSR-1 Serie, 1 A [Online]. Disponible: <http://www.adafruit.com/datasheets/tsr1.pdf>
- [8] Laird Technologies. ConnexLink™ Wireless Cable Replacement System [Online]. Available : <http://www.mouser.com/ds/2/223/ConnexLink%5b1%5d-201574.pdf>
- [9] C. Muriá, D. Huerta, C. Sanchez, A. Camargo, J. Cruz, "Seismic Monitoring on the Curved Portion of an Elevated Railway," in *Tenth U.S. National Conference on Earthquake Engineering Frontiers of Earthquake Engineering*, 2014.
- [10] P. J. Bennett, K. Soga, I. Wassell, P. R. a. Fidler, K. Abe, Y. Kobayashi, and M. Vanicek, "Wireless sensor networks for underground railway applications: case studies in Prague and London," *Smart Struct. Syst.*, vol. 6, no. 5, pp. 619–639, 2010.
- [11] Microchip Technology inc. Microchip Wireless (MiWi™) Application Programming Interface–MiApp[Online].Disponible: <http://ww1.microchip.com/downloads/en/AppNotes/01284A.pdf>
- [12] Microchip Technology inc. Microchip Wireless (MiWi™) Media Access Controller – MiMAC [Online]. Disponible: <http://ww1.microchip.com/downloads/en/AppNotes/01283a.pdf>
- [13] Chulsung Park; Lahiri, K.; Raghunathan, A., "Battery discharge characteristics of wireless sensor nodes: an experimental analysis," *Sensor and Ad Hoc Communications and Networks*, 2005. IEEE SECON 2005. 2005 Second Annual IEEE Communications Society Conference on , vol., no., pp.430,440, 26-29 Sept., 2005

Aplicación de Sistemas Embebidos para Monitoreo Remoto Aplicado a Colectores Solares

Palacios Curay, Marco; Atoche Sánchez, Edgardo; Chunga Saavedra, Arturo;
Chumacero Timaná, Pablo; López Monzón, Robinson.
Universidad de Piura
Piura, Perú

En este proyecto se ha desarrollado un sistema embebido con la finalidad de medir variables necesarias para validar el modelo de transferencia de calor y control en un colector solar aplicado al secado de cacao, respondiendo a la urgencia local de implementar tecnologías en los procesos postcosecha, optimizando calidad, productividad y manteniendo estándares industriales.

Para exponer el sistema de monitoreo se mencionarán a continuación cada elemento que interviene en el proceso, especificando las características técnicas correspondientes.

El Colector Solar aprovecha la radiación solar y el efecto invernadero para elevar la temperatura del aire mediante la transferencia de energía térmica. El flujo de aire contenido es orientado convenientemente para el sistema de secado. [1]

El sistema posee un microcontrolador de la familia PSoC (Programmable System-on-chip) el cual se ha servido de un sensor de tipo RTD (Resistance Temperature Detector) para la adquisición de datos de temperatura. Se utilizó una resistencia de referencia para medir la corriente real suministrada por el PSoC, un ADC Delta Sigma con 12 bits de resolución y OPAMP's a la entrada para el acondicionamiento de la señal (todos estos módulos son integrados y reconfigurados dentro del sistema PSoC); adicionalmente se cuenta con un módulo integrado que contiene las aproximaciones polinómicas de temperatura, brindando mayor exactitud en el cálculo. [2][3]

Para la comunicación se utilizó tecnología de radio frecuencia, que permite enviar datos mediante conexión inalámbrica. Los dispositivos instalados fueron módulos XBee, fabricados por la empresa Digi, los cuales fueron configurados tomando en cuenta las bases teóricas del protocolo IEEE 802.15.4. La data registrada por el PSoC es enviada a través de los XBee a un módulo Raspberry Pi. [4][5]

De acuerdo con la tendencia del Internet de las Cosas, se han usado los API's del servicio Ubidots, plataforma web mediante la cual se registra información en tiempo real sincronizado con el huso horario mundial. [6]



Fig. 1. Colector Solar con Sensores de Temperatura

El Raspberry Pi es un ordenador de placa reducida (SBC) incluye un System-on-a-chip Broadcom BCM2835 y contiene un procesador central (CPU) ARM1176JZF-S a 700 Mhz. Este dispositivo se encarga de adquirir los datos, procesarlos mediante filtros Kalman y enviarlos a un servidor en la nube. El sistema completo dispone de una interfaz gráfica desarrollada en Visual Python que permite la supervisión y el almacenamiento para una mejor interacción con el usuario.

Esta propuesta para las lecturas de la temperatura mediante RTD's tiene menor costo en comparación con las alternativas del mercado local, además, permitirá reducir los tiempos de secado del cacao, mejorando su calidad e impulsando su plan de expansión productivo, así como la transferencia tecnológica para otras cadenas de producción.



Fig. 2. Esquema: Monitoreo Remoto en Tiempo Real

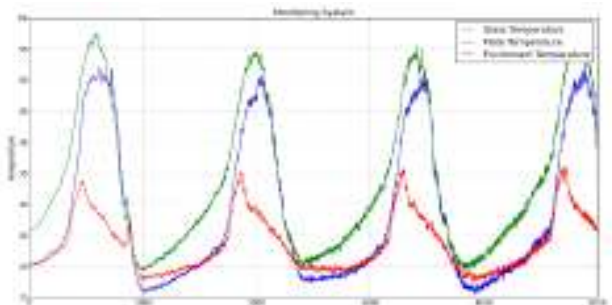


Fig. 3. Gráfica de Resultados (Temperatura vs Tiempo)

REFERENCES

- [1] D. Jain, "Performance evaluation of an inclined multi-pass solar air heater with in-built thermal storage on deep-bed drying application." *Journal of Food Engineering*, 2004, vol. 65, pp. 497 - 509.
- [2] R. Ashby, "Designer's Guide to the Cypress PSoC (Embedded Technology)," Newnes. Boston: Elsevier, August 2005.
- [3] M. Nidhim, "AN84783 - Accurate Measurement Using PSoC3 and PSoC 5LP Delta-Sigma ADCs," 2012.
- [4] A. Titus, "Two-Way Communications with XBee Modules," *The Hands-on XBEE Lab Manual*, Amsterdam: Elsevier, 2012.
- [5] S. Abraham, X. Li, "A Cost-effective Wireless Sensor Network System for Indoor Air Quality Monitoring Applications," *Procedia Computer Science*, 2014, vol. 34, pp. 165-171.
- [6] B. Siddhartha, M. Anjana, S. Kanak, "A Review of Designing and Implementing an Embedded System Using Client Server and Web Technology for Monitoring and Controlling of Physical Parameters." *India: Gauhati University*, 2010, pp.217-226.

Sistema de adquisición y postprocesamiento de parámetros de vuelo de vectores

F. Escobar, J. Oviedo, J. P. Rumie Vittar. E. Giugge.

Centro de Investigación y Desarrollo de Tecnologías
Aeronáuticas. (CITeA).
Fuerza Aérea Argentina.
Las Higueras, Córdoba, Argentina.
fmartinescobar@gmail.com

P. Solivellas, D. Primo, D. Diaz.
Grupo de Sistemas de Tiempo Real. Fac. Ingeniería.
Universidad Nacional de Río Cuarto
Río Cuarto, Córdoba, Argentina.

El presente trabajo describe el diseño, desarrollo e implementación del hardware y el software de un sistema embebido, registrador y autónomo de datos de navegación y actitud de vuelo aplicado a un vector; con la finalidad de adquirir datos de varios sensores, como unidades inerciales, sensores de temperatura, presión barométrica y humedad interna.

El sistema de software, se compone de dos subsistemas: El sistema de Adquisición de Datos Vuelo (SADV) y el sistema de Análisis y Visualización (SAE). El SADV se encarga de la adquisición de parámetros en vuelo, de eventos, etc. para posteriormente almacenarlos en una memoria de estado sólido y finalmente, su análisis post vuelo. Por otro lado, el SAE es el responsable de recibir los datos provenientes del el Vector con el objeto de mostrar la evolución del ejercicio tanto en 2D como en 3D[1].

El sistema de hardware, consta en primera instancia en un gabinete de aluminio reducido en tamaño y peso (55x80x185mm de aluminio 2024 de 2mm) según los requerimientos impuestos Centro de Investigaciones Aplicadas (CIA). El mismo contiene un procesador ARM Cortex A8[2] con un sistema operativo embebido[3] que procesa y almacena la información proveniente de los sensores con los que está equipado[4].

Es importante destacar que todos los componentes utilizados en el desarrollo de hardware son del tipo COTS.

A la fecha se cuenta con un prototipo íntegramente desarrollado y testeado, listo para ser integrado en carácter de módulo invitado en el Vector Experiencia Centenario desarrollado por la Fuerza Aérea y Vector Gradicom del Ministerio de Defensa[5].

REFERENCES

- [1] D. Díaz; F. Escobar; J. Oviedo; J. P. Rumie Vittar; P. Solivellas. "Sistema para Evaluación de Adiestramiento de Combate (SEACO)". CASE 2014. Buenos Aires, Argentina., 2014.
- [2] ARM "Cortex-A8 Technical Reference Manual" Revision r3p2. 7 May 2010. ARM DDI0344K (ID060510).

- [3] Hessel, F.; da Rosa, V.M.; Reis, I.M.; Planner, R.; Marcon, C.A.M.; Susin, A.A., "Abstract RTOS modeling for embedded systems," Rapid System Prototyping, 2004. Proceedings. 15th IEEE International Workshop on, vol., no., pp.210,216, 28-30 June 2004 doi: 10.1109/IWRSP.2004.1311119
- [4] M. Escobar, D. Diaz, P. Solivellas, J. Oviedo y J.P. Rumie Vittar. "Datalink between aircraft and ground station with an embedded RTOS in a PIC18x". Centro de Investigación y Desarrollo de Tecnologías Aeronáuticas (CITeA) – Dirección General de Investigación y Desarrollo – FAA – Las Higueras – Argentina, 2013.unpublished
- [5] RTCA "DO-160D. Environmental Conditions and Test Procedures for Airborne Equipment" 2010. Washington DC, EEUU.

Prototype of ZigBee RSSI Measurer Using Arduino

Teles de Sales Bezerra, José Anderson Rodrigues de Souza and Saulo Aislan da Silva Eleutério
Federal Institute of Education Science and Technology
Campina Grande - Paraíba, Brazil
teles at ieee dot org,
{andersonrodriguespb, sauloaislan20} at gmail dot com

Jerônimo Silva Rocha
Federal Institute of Education, Science and Technology
Campina Grande - Paraíba, Brazil
jeronimo at iecom dot org dot br

The Wireless Sensor Networks (WSN) became a trend in the last years due to advances in wireless communications, such as new information technologies and electronic attributes developed for these technologies [1]. One can say that the WSNs are one from the more promising technologies from this generation, as they have great usability due to their implementation on industrial control systems, and also their low cost combined with multi-functional sensors that do surveillance functions and day-to-day activities control. Its great versatility for being used in several application fields made the WSNs attract an increasingly interest in the last few years [2]. In the last two decades, particularly, extensive researches have induced to the definition of a new wireless systems generation, capable of extending even more the WSN's application fields. There are applications in health care, home automation and automation in general. A relevant topic while researching in wireless systems is the characterization of how the radio signals' range varies in indoor and outdoor environments, because some ambient conditions can easily make a transmission hard to do due to interferences [3].

ZigBee Sensor networks are performed by RSSI values. RSSI (Received Signal Strength Indicator) is a measure from the received radio signal's power. It's a metric used to estimate the transmission quality between two nodes as the distance between them is varied, implemented under the IEEE 802.11 standard [4]. It works by using the distance between transmitter and receiver to designate the quality from the received signal, even considering variations on signal strength by comparing the received signal level with probability distributions and localization measures based on statistic analysis. This research we propose the development of a RSSI measurer, which is utilized to collect data on outdoor environments. This data will be used as a case study, whereupon we've isolated the relevant factors on a RSSI measurement, and have shown radio signals' propagation. The methodology adopted during the assembly from a RSSI measurer were made under the following step's schedule.

First Step: The used device had a direct-access terminal on a specific pin to read the RSSI. On this pin it was read a PWM (Pulse Width Modulation) modulated signal. This signal is treated as an analogical output, but as matter of fact it's a digital output that generates an alternating signal low and high digital levels).

Second Step: Compatibility tests were made between the used Arduino Uno R3 platform and the XBee modules, for the purpose of test basic trigger circuits with the modules, and verify the whether there was correct communication maintenance between the devices. On this stage, the prototypes were assembled in assembly boards as shown in Figure 1.



Figure 1 - Measurer prototype using Arduino Platform

Third Step: On this stage it was written the source-code executed by the prototype. The programming language used is called Wiring, that includes several on-the-box applications allowing an easy development of various input-output operations. That's one of the reasons whereby it's the standard development language for Arduino projects. The analogical pin (the one who provides the RSSI value) was read through the pulseInt() function, made for occasions like this one. This function reads a high or low pulse on the pin and then returns its duration in milliseconds. Thus, it measures the PWM pulses length.

REFERENCES

- [1] F. Yahaya, Y. Yusoff, R. Rahman and N. Abidin. "Performance analysis of Wireless Sensor Network". In 5th International Colloquium on Signal Processing Its Applications, 2009. CSPA 2009, pp. 400-405.
- [2] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. "A survey on sensor networks". IEEE Communications Magazine, vol. 40, no. 8, pp. 102-114, Aug 2002.
- [3] R. Pellegrini, S. Persia, D. Volponi and G. Marcone. "RF propagation analysis for ZigBee Sensor Network using RSSI measurements". In 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology, 2011, pp. 1-5, Feb 2011.
- [4] Benkic, Karl, et al. "Using RSSI value for distance estimation in wireless sensor networks based on ZigBee." Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on. IEEE, 2008.

Seguimiento de Objetivos en Redes de Sensores Multimedia mediante Estimación Distribuida

Diego González Dondo y Julio H. Toloza
Centro de Investigación en Informática para la Ingeniería.
Universidad Tecnológica Nacional, Facultad Regional Córdoba
Córdoba, Argentina
dgonzalez@scdt.frc.utn.edu.ar

Una red de sensores consiste de un conjunto de dispositivos generalmente idénticos, llamados nodos, desplegados sobre un región geográfica de interés, que se usa para la medición y monitoreo de diversos fenómenos físicos, o para la detección y seguimiento de eventos, en forma cooperativa y coordinada [1], [2], [3].

El procesamiento de la información sensorial puede realizarse genéricamente de dos maneras: centralizada o distribuida. En una red centralizada los nodos solamente proveen información que es transmitida a una unidad de procesamiento central. Por el contrario, en el procesamiento distribuido los nodos no sólo proveen la información sensorial sino que también realizan parte del procesamiento de la misma en forma cooperativa, usando el poder computacional de cada nodo. La suposición básica es que cada nodo en la red puede explotar la información recibida para optimizar las futuras acciones de censado y manejar los limitados recursos de procesamiento y comunicación eficientemente.

Un ejemplo típico de aplicación usando procesamiento cooperativo de la información en redes inalámbricas de sensores puede ser localización y seguimiento de objetos móviles en forma colaborativa.

Los nodos desplegados geoespacialmente colectan señales desde un objetivo en movimiento u otra entidad dentro de su área de cobertura y comunica la información sobre el mismo con otros nodos vecinos para ubicar al objeto. Dependiendo del campo de aplicación el objeto a seguir puede ser animales, vehículos, o personas. Los sensores utilizados en tales aplicaciones son generalmente cámaras (sensores de imágenes).

En aplicaciones de seguimiento, la formulación del problema implica la medición y observación del estado del evento cuya evolución temporal/espacial puede ser modelada como un proceso estocástico, que dependiendo de los sensores a utilizar, puede ser no lineal y no gaussiano. Un modelo válido para las observaciones es el empleo de mediciones angulares relativas del objetivo a seguir, respecto de la posición de cada nodo, lo implica una ecuación no lineal en la observación y una distribución de probabilidad a posteriori no gaussiana. Bajo estas consideraciones la estimación de estado puede ser realizada mediante técnicas de filtrado bayesianos, más específicamente implementaciones secuenciales de Monte Carlo, como filtro de partículas [4].

En el trabajo se presentan simulaciones de una red inalámbrica de sensores multimedia, mas precisamente red de cámaras para el seguimiento de objetivos móviles basado en la implementación de un filtro de partículas en forma distribuida.

En este trabajo, consideramos a una red descentralizada de cámaras para realizar tareas de estimación a través de procesamiento y comunicaciones locales. Las estimaciones distribuidas son realizadas bajo el esquema de nodos líderes [5]. Estos son designados dentro del conjunto de nodos que están llevando a cabo las observaciones en un determinado instante de tiempo, y se encargan de realizar las estimaciones mediante un filtro de partículas, colectando la información de los nodos vecinos. En base a la estimación obtenida, se designa el próximo líder, transfiriéndole el procesamiento y el estado previo. Luego este se encargará de coleccionar nuevas observaciones y realizar una nueva estimación.

Las simulaciones llevadas a cabo comprenden el despliegue de cámaras, con una densidad de cobertura de un nodo cada $10m^2$ y generando trayectorias aleatorias para el objetivo a seguir, dentro del área de cobertura y empleando 200 partículas en el filtro. Los resultados muestran luego de 100 corridas, un error medio de 20cm, con una desviación de 35cm en la estimación de la posición del objetivo.

REFERENCIAS

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *Communications Magazine*, IEEE, vol. 40, no. 8, pp. 102–114, aug 2002.
- [2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [3] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960, 2007.
- [4] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing*, IEEE Transactions on, vol. 50, no. 2, pp. 174–188, feb 2002.
- [5] O. Hlinka, F. Hlawatsch and P. Djuric, "Distributed Particle Filtering in Agent Networks: A Survey, Classification, and Comparison" *IEEE Signal Processing Magazine*, 30(1), pages 61 - 81, 2013.



www.sase.com.ar

12 al 14 de agosto

Facultad de Ingeniería | UBA
Buenos Aires, Argentina

ISBN 978-987-45523-3-4



9 789874 552334