System on Chip based Brain-Computer Interface

Pablo A. García
GIBIC-LEICI
Facultad de Ingeniría - UNLP
La Plata, Argentina
pagarcia@ing.unlp.edu.ar

Matías Oliva
GIBIC-LEICI
Facultad de Ingeniría - UNLP
La Plata, Argentina
mati93_o@hotmail.com

Enrique Spinelli
GIBIC-LEICI
Facultad de Ingeniría - UNLP
CONICET
La Plata, Argentina
spinelli@ing.unlp.edu.ar

***Abstract*— This work presents an autonomous embedded system based on a system on chip that implements a brain-computer interface. The brain-computer interface is based on steady-state visual evoked biopotential and the embedded system on a system on chip which combines a dual-core Cortex-A9 embedded processor with programmable logic for design flexibility. The programmable logic side provides a solution for parallel tasks with strict real-time constraints, and the processor includes capabilities to port an operating system with a graphical user interface, network support, and file system, among others. Initially, a verification of the operative system running on the embedded processor sharing data with the logic side is presented, to find out its real-time capability as a set. Finally, a brain-computer interface based on visual evoked potentials is implemented. Results of this application recovering visual evoked potential on the embedded system, are also presented.**

Keywords*— *Embedded System, System on Chip, Assistive Device, Brain-Computer Interface, Steady-State Visual Evoked Potential, Speller.

I. INTRODUCTION

Brain-computer interfaces (BCI) are one of the most important applications of embedded systems (ES). Among the many definitions of these devices, the most referenced and clear was proposed by a group of referents from different countries in 2002 [1]: "A direct brain-computer interface is a device that provides the brain with a new, non-muscular communication and control channel".

The BCI's main objective is to offer a communication and control channel to people with reduced mobility, which gives them the possibility to handle a speller, a wheelchair, a household appliance or a mouse computer, among others [2] [3] [4]. Nowadays they are also being used in rehabilitation [5] and entertainment in video games [6].

Brain-computer interfaces use brain biopotentials to command different types of devices through a computer. The commonly used form to record brain activity is through electroencephalography (EEG) because it is a non-invasive practice that is recorded with electrodes arranged on the scalp. Fig. 1 shows a proprietary BCI prototype with two commands. It can be observed the electrodes on the scalp to register the brain activity.

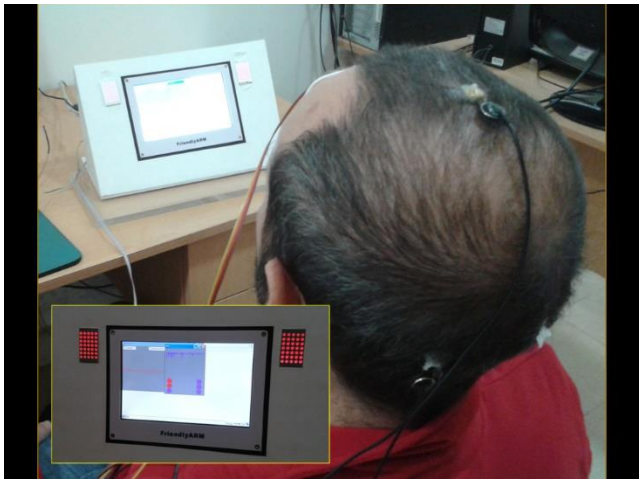

Fig. 1. Brain-computer interface.

There are different types of BCI, based on different potentials. Some of them are event-related synchronization and desynchronization (ERS/ERD) [7], slow cortical potentials [8], brain rhythms (α y β) [9] and steady-state visual evoked potentials (SSVEP) [10].

Because of its simplicity, ease of use and high transfer rate, an SSVEP based BCI is implemented on this work. These evoked potentials reflect brain activity produced by the visual information processing that can be registered in the occipital zone by EEG techniques. These potentials are presented in response to certain visual stimuli, and for this reason, they are called evoked. There are two types of visual evoked potentials: Transient Visual Evoked Potentials (TVEP) and Steady-State Visual Evoked Potentials (SSVEP). TVEPs occurs when visual stimulus rate is less than two stimuli per second, while SSVEPs occur for periodic stimuli with frequencies greater than 6 Hz [11]. In the latter case, evoked potentials overlap forming a steady-state response with components in the stimulus frequency and its harmonics.

Fig. 2 shows an SSVEP based BCI functional model. It comprises a visual stimulator, a biopotential acquisition stage, a processing stage and a command generator that will handle the user assistance device.

ISBN: 9789874685926

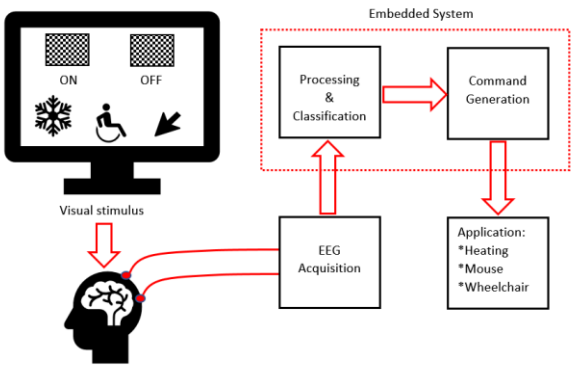

Fig. 2. SSVEP based BCI.

During operation, visual stimuli are presented to the user blinking at different frequencies. Each of these stimuli (frequency) is associated with a command. When the user focuses the vision on a command the brain's visual information processing will generate an evoked potential with the associated stimulus frequency and its harmonics). Then, through a spectral or time domain analysis the selected command is detected from the EEG signal recorded over the occipital zone.

Many current BCI applications exist exclusively in academic research and a few commercial brands. They are commonly implemented by a front end biopotential acquisition with processing done on a computer [12]. Brain-computer interfaces have proved to be useful and it is time to migrate the deployment from a laboratory prototype to a portable device reduced in size. In this scheme, the computer cannot be part of the interface and need to be replaced.

In this sense, the natural replacement is an embedded system. There are different variants for the implementation of the ES that can be dimensioned according to the temporary restrictions to be verified. A first variant consists of using an ES with good computing capacity, real-time features, and the ability to implement a friendly user interface.

Fig. 3 shows a diagram of this variant where it can be seen that the processor must port an RTOS that allows it to implement, on the one hand, a friendly user interface and communication ports; and on the other hand, ability to implement controllers to acquire real-time samples and handle the visual stimulator.

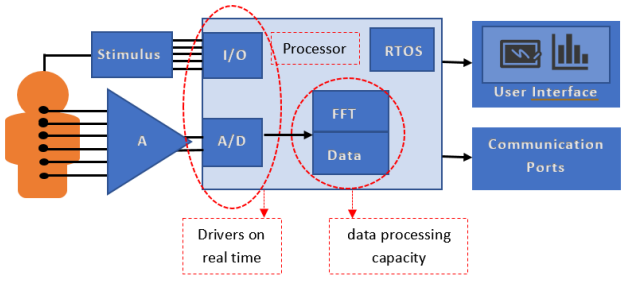

Fig. 3. Embedded SSVEP based BCI scheme.

In [13] a BCI platform that verifies this operation scheme is presented. It verifies real-time features, with interruption latencies on the order of 100 µs for the real-time controllers. These real-time features are enough for BCI implementations, but when it is required to manage greater parallelism and to handle lower interruption latencies, it is necessary to migrate to other schemes. In this sense, the authors propose the use of a system on chip (SoC) based embedded system as an integral solution.

II. System on Chip

The SoC systems that integrate both architectures (processor and FPGA) in a single device have a higher level of integration, lower consumption, smaller board size and a greater communication bandwidth between the processor and the FPGA. They integrate the high-level functionality of a processor with the real-time operation, complex data processing and interface functions of an FPGA.

In these heterogeneous systems, FPGAs are ideal for handling parallel operations of many data channels; and because they implement computation directly in hardware, they provide a low latency path for tasks such as custom triggering and high-speed closed-loop control. On the other hand, they improve the flexibility of embedded systems, making them easier to update than systems with fixed logic and allowing them to adapt to changing I/O requirements.

Combining a microprocessor and an FPGA in a heterogeneous computing architecture allows the use of the strengths of each processing element, to best meet the application requirements. In this scheme, the designers can solve the tasks of low latency or real-time using the FPGA, while the embedded processor takes care of porting an operating system that solves the user interface and the rest of the tasks that do not require low latency.

Before implementing the BCI using the SoC, it was decided to evaluate its real-time performance, to determine all possible implementations in which it may be useful.

It was decided to experiment with the low-cost DE10-Nano board from Terasic (Fig. 4).

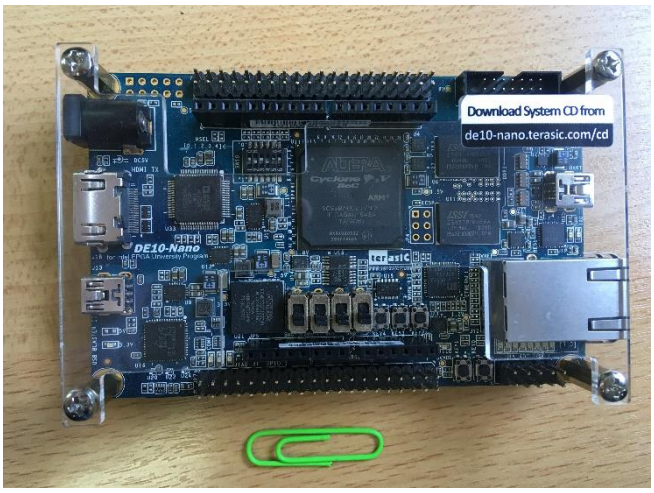

Fig. 4. DE10-Nano / SoC.

The system combines a Cyclone V FPGA with an ARM-Cortex A9 processor which port a Linux embedded operating system. As a first test, it was decided to implement a data sampling and storage system, using the analog to digital converter (ADC) LTC2308 integrated on the board. This is a successive approximations converter with 8 channels, 12 bits resolution and 500 k samples per second.

To optimize the computing speed, the sampling and storage system was totally wired on the FPGA side, preserving the embedded processor to solve the user interface.

Fig.5 shows the system's scheme implemented to register data continuously. In this scheme, the HPS processor manages two memories, reading one of the memories while the other is still filling. It implements a double circular buffer like the one in [13]. The advantage is that it allows the continuous sampling of the data, which is not limited by the capacity of the FIFO memory used since the data of one can be filled while the system continues writing in the other. The frequency of the data reading clock is higher than that of the writing clock.

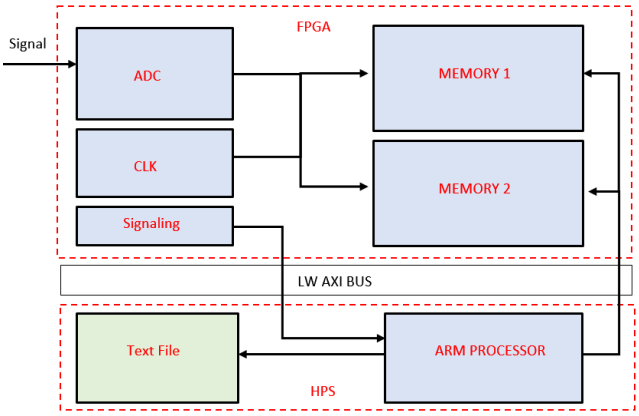

Fig. 5. SoC-based sampling implementation.

The FPGA hardware was programmed on HDL (Hardware Description Language) using the Quartus II software provided by Altera. In conjunction with this software, Altera provides Intellectual property (IP) blocks which implement different functionalities, like the ADC controller, memory blocks and even software processors.

The HPS module is an ARM-Cortex A9 processor, which ports an embedded Linux operating system, that is loaded from a micro SD card.

An FPGA to HPS bridge must be included in the design to give the embedded processor access to the memory zones of the FPGA. The most basic bridge, a lightweight Advanced eXtensible Interface (AXI) bus bridge, was used.

Once defined the necessary configuration of the FPGA, the operating system must be programmed to load it when Linux is being booted. In this way, the sampling runs completely by the FPGA and the user should not worry more than to start it and then store the obtained samples.

The program that runs on the ARM processor was written in C. It must start sampling, wait for full memory signaling from the FPGA and then request the reading of the memory successively until it is empty. While the data is being read, it is saved in a text file on the micro sd memory card, making it easily available to the user. The user can start a new reading window when necessary.

The text file generated by the HPS was processed off-line on a computer verifying the correct functioning of the entire system.

III. SoC-BASED BCI

In this implementation, it is intended to solve the real-time processing with the FPGA side. In particular, the generation of data through the integrated A/D converter and, on the other hand, performing the calculation of the fast Fourier transform (FFT) to recover SSVEP by dedicated hardware in the FPGA. Information can be shared between the FPGA and the processor through an internal interconnection. This way, the processor can run an OS without temporal restrictions to solve the user interface and communication ports. This board offers support through drivers (BSP) for a Linux embedded distribution (Fig 6).

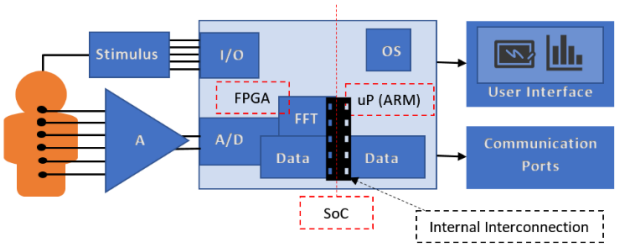

Fig. 6. SoC-based BCI scheme.

Fig. 7 shows the different parts of the BCI system implemented, which are described continuously.

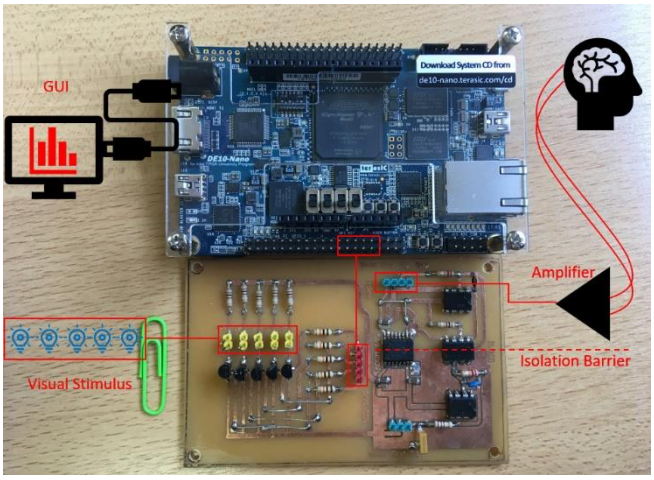

Fig. 7. SoC-based BCI parts.

A. Amplifier

To acquire the EEG signal, it was decided to implement an analog amplifier with one channel and differential input presented in [14]. It is a low noise AC coupled amplifier energized from a single 5V source. The setup is complemented with a "Driven Right Leg" (DRL) circuit to fix the patient's potential to 1.25V referenced to the amplifier's ground.

For the DRL, two independent electrodes placed on the forehead were used, avoiding fixation problems by the hair. The single-channel amplifier presents a gain of 5832, divided into two stages of 5.25 and one of 1111. The amplifier was mounted in a small plastic box that can be attached to the user's clothes by a snap (Fig. 8).

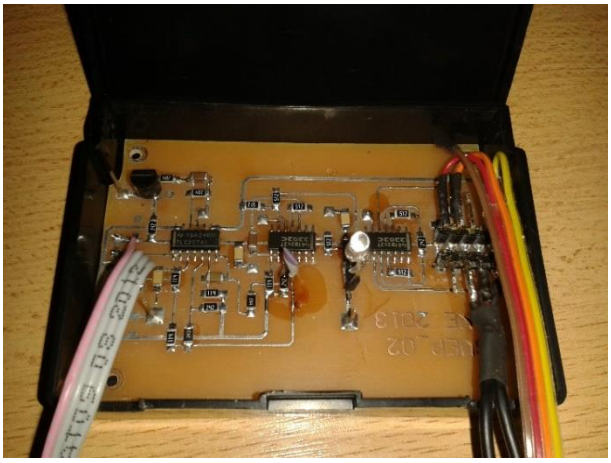

Fig. 8. Amplifier.

B. *Isolation Barrier*

Power supply to the amplifier was provided through an isolation barrier from the SoC, implemented by an integrated isolator ADUM6401 of Analog Devices.

It was implemented a photovoltaic isolation amplifier using the IL300 optocoupler to isolate the signal.

C. *Visual Stimulus*

As visual stimulators, red LEDs arrangements flashing at five different frequencies were used (Fig. 11). To maintain synchronism between the stimuli and the biopotential acquisition, the sampling frequency (1024 Hz) was used as base time. The stimulus used were generated using logic on the FPGA and a transistor-based power circuit to drive the led arrays.

Stimulus frequency 1: f_{st1}=14 Hz
Stimulus frequency 2: f_{st2}=16 Hz
Stimulus frequency 3: f_{st3}=18 Hz
Stimulus frequency 4: f_{st4}=20 Hz
Stimulus frequency 5: f_{st5}=22 Hz

D. *Data acquisition*

Data acquisition was solved using the ADC included on the SoC. The ADC is managed by an IP block implemented with logic on the FPGA side. It was configured at a frequency of 1024 sps. The sampling is initiated by the HPS processor through the WR enable line. On each rising edge of its clock, the ADC controller makes available the samples to the memory block. This memory uses a FIFO (First in first out) scheme and is implemented in two blocks of 512 samples (1024 Bytes). In this scheme, the HPS processor manages two memories, reading one of the memories while the other is still filling. It allows the continuous sampling of the data, which is not limited by the capacity of the FIFO memory used since the data of one can be filled while the system continues writing in the other (Fig. 9).

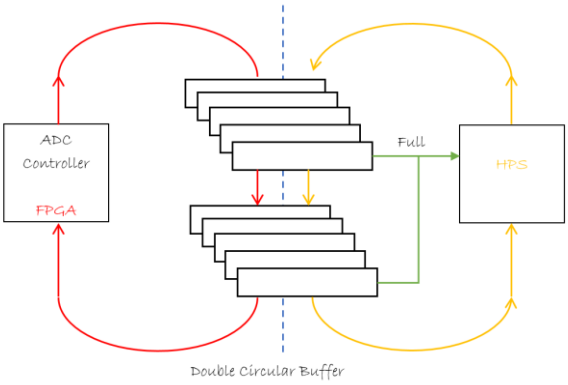

Fig. 9. Management of acquired data.

E. *Data processing.*

To access the data on the double circular buffer on real-time using the Linux OS, the HPS processor must open a special file on the directory "/dev" were the samples and signalization are mapped. This file, "/dev/mem", is an interface to the memory controller that gives access to the data written by the FPGA. This was implemented with a program on C# language using the Mono open-source implementation of the .NET Framework.

As a first step, before a BCI practical implementation, it was developed a visual tool to visualize and analyze the EEG signal recorded. In Fig. 10 is presented the GUI of the developed tool. This allows observing the signal in the time and frequency domain.

The signal is continuously plotted on the time domain and processed on frames of 512 samples to calculate the discrete Fourier transform (DFT) through the fast Fourier transform (FFT). With this frame size and a 1024 sps sampling frequency, the frequency resolution on the FFT is 2 Hz per beam in accordance with the frequency separation on the visual stimulus selected.

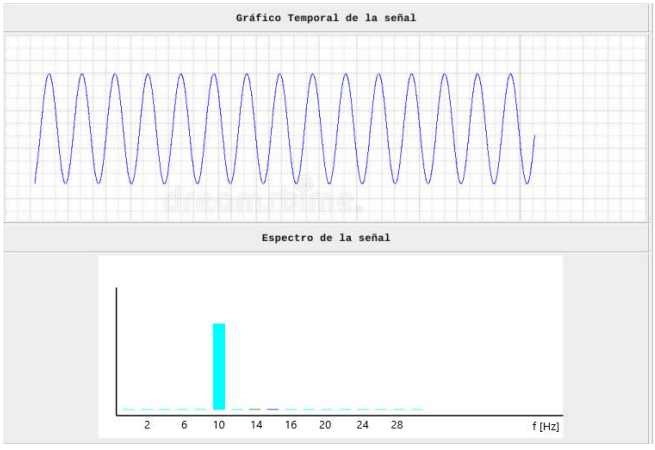

Fig. 10. EEG analysis tool.

The usage of this tool allows us to realize the first experiments on SSVEP recuperation as a previous step to the BCI practical implementation. There were realized various SSVEP registers with the electrodes placed on the O1-O2 position of the 10-20 international system.

IV. Practical Implementation

To test the embedded system proposed and its real-time capabilities, an SSVEP based speller was implemented.

A. Speller GUI

In this implementation, the SSVEP registers of different subjects are acquired. In each epoch (512 samples or 0.5 seconds) the magnitude of the signal's FFT is calculated in the frequencies of interest (14, 16, 18, 20 y 22 Hz), and compared with each other and with the mean, selecting the one that has a higher magnitude. Each visual stimulus is associated with an option in the screen of the SoC's OS. If a certain frequency is selected in three consecutive epochs, that is to avoid false positives, then the option corresponding to that frequency is selected. Using this system, the subject can, looking at the five stimuli, select between five different options. This way, the user can select between 25 different options in two consecutive stages.

With this configuration, a speller can be implemented. In the first stage, the subject selects between five, five letter groups and in a second stage, selects between the letters of that group (Fig 11). The letter "Y" was omitted due to its low occurrence in the Spanish language, and the limitation of 25 symbols. To provide feedback to the user, the GUI also implements progress bars related to the three successive selections.

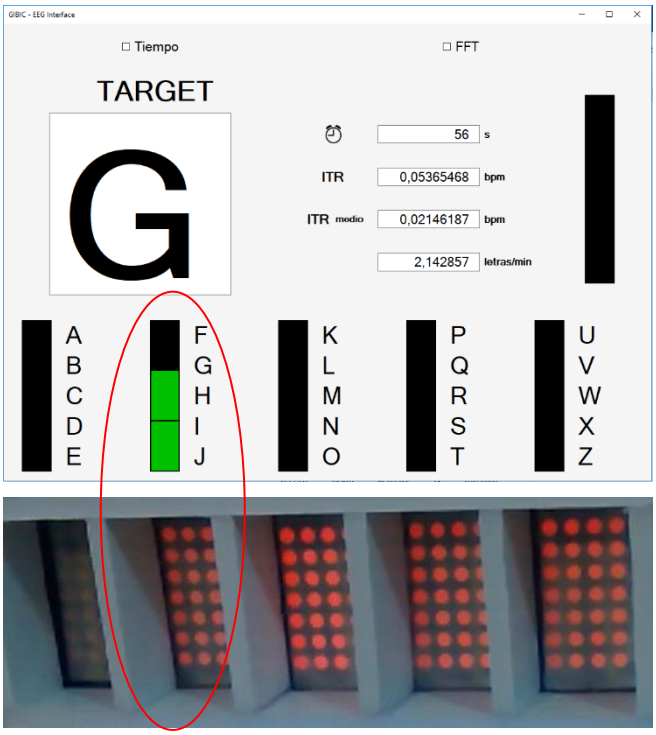

Fig. 11. Speller GUI Interface and visual stimuli.

B. ITR

To measure the performance of the BCI embedded system the information transfer rate was calculated. ITR is a general evaluation metric designed for BCIs and it allows comparison of the performance of systems which have a different number of tasks. ITR was defined in [15], by simplifying the mutual information calculation from Shannon's channel theory. The equation for ITR is:

$$B = \frac{1}{c} * (\log_2 N + P \log_2 P + (1-P) * \log_2 \frac{1-P}{N-1})$$

where B is ITR in bits per second, c is the time per selection, N is the number of possible choices, and P is the probability that the desired choice will be selected, which is estimated during the experiment.

In this implementation with N=5, the best ITR we can obtain is 92.87 bpm when the user produces a command in 1,5 s (3 consecutive frames) and P=1. To avoid false positives, the BCI was adjusted with high thresholds and P results typically in 1.

V. Experimental Results

Some tests were made with three users, to test the BCI proposed and obtain our ITR.

To test the functionality of the system the subjects were presented with 50 aleatory objective letters and the time it takes to select them from the 25 possible letters was measured. With this data, the mean quantity of letters per minute that the subject can select was measured, and the mean ITR of the process was calculated.

Table 1 resumes the results of these experiments for the three users.

TABLE I.

User	# commands	# Letters	ITR_{mean} [bpm]	Letter/min (mean)
1	100	50	34.8	7.5
2	100	50	19.9	4.3
3	100	50	23.2	5

VI. Conclusion

It has been demonstrated that a low-cost general-purpose SoC-based embedded system can be used for visual evoked biopotential acquisition and processing. The system is compact, portable and does not rely on a personal computer, presenting a booting time below 10 sec. The use of a system on chip (SoC) based embedded system expands the horizon of possible applications. The logic side inclusion allows us to manage greater parallelism and to handle lower interruption latencies, keeping the user interface resolved with an operative system on the processor side.

A prototype was built and tested in real conditions, implementing a brain-computer interface based on steady-state visual evoked potentials with five stimuli. Experimental results show correct visual evoked potential recoveries.

The implemented BCI is a speller with a friendly user interface and very easy to use. It was tested with three users without training, working correctly. The best of the users could write 7.5 letters/min, while the slowest one reached 4.3 letters/min. The faster user had more training with the BCI than the others, which allows us to conclude that the ITR improves with training.

ACKNOWLEDGMENT

This work is supported in part by the projects:

- PICT-2015-2257 from FONCYT. ("Control, Electrónica e Instrumentación: Aplicaciones en Energías Alternativas y Bioingeniería")
- I-219 from UNLP. ("Procesamiento analógico y digital de señales. Aplicaciones en instrumentación científica e industrial").

REFERENCES

[1] Wolpaw J. R, Birbaumer N., McFarland D. J., Pfurtscheller G., and Vaughan T. M. "Brain-computer interfaces for communication and control". Clin. Neurophysiology, vol. 113, pp. 767–791, 2002.

[2] Chabuda Anna, Durka Piotr and Zygierewicz Jarosław, "High Frequency SSVEP-BCI With Hardware Stimuli Control and Phase-Synchronized Comb Filter", IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, VOL. 26, NO. 2, FEBRUARY 2018

[3] Vo Kha, Pham Thuy, Nguyen Diep N., Kha Ha Hoang, and Dutkiewicz Eryk, "Subject-Independent ERP-Based Brain–Computer Interfaces" IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, VOL. 26, NO. 4, APRIL 2018.

[4] Long Jinyi, Li Yuanqing, Wang Hongtao, Yu Tianyou, Pan Jiahui, and Li Feng, "A Hybrid Brain Computer Interface to Control the Direction and Speed of a Simulated or Real Wheelchair", IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, VOL. 20, NO. 5, SEPTEMBER 2012.

[5] Ang Kai Keng and Guan Cuntai, "EEG-Based Strategies to Detect Motor Imagery for Control and Rehabilitation", IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, VOL. 25, NO. 4, APRIL 2017

[6] Kaplan Alexander Y., Shishkin Sergei L, Ganin Ilya P., Basyul Ivan A., and Zhigalov Alexander Y., "Adapting the P300-Based Brain–Computer Interface for Gaming: A Review" IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, VOL. 5, NO. 2, JUNE 2013.

[7] García P. A., Haberman M. and Spinelli E. M. "A Versatil Hardware Platform for Brain Computer Interfaces", *32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Buenos Aires, Argentina. ISBN 978-1-42444124-2. 2010.

[8] Li Yi, Zhang Jianhui, Su Yu, Chen Weidong, Qi Yu, Zhang Jicai, Zheng Xiaoxiang "P300 Based BCI Messenger", Complex Medical Engineering, 2009. CME. ICME International Conference, 9-11 April 2009, Print ISBN: 978-1-4244-3315-5. 2009.

[9] Spinelli E. M., Mayosky M. A. y Martinez N. H. "Interfaz adaptiva para comunicaciones mediante EEG basadas en el ritmo alfa visual", XVIII Congreso de la Sociedad Española de Ingeniería Biomédica, pp. 161-163, Cartagena, Spain. 2000.

[10] García P. A., Spinelli E. M. and Toccaceli G., "An Embedded Hybrid BCI Speller". VII Congreso Latinoamericano de Ingeniería Biomédica (CLAIB 2016). Bucaramanga, Colombia. Octubre de 2016. IFMBE Proceedings, eBook ISBN: 978-981-10-4086-3, DOI: 10.1007/978-981-10-4086-3. 2016.

[11] Gao Xiaorong, Xu Dingfeng, Cheng Ming, and Gao Shangkai,"A BCI-Based Environmental Controller for the Motion-Disabled", IEEE Transactions on neural systems and rehabilitation engineering, vol. 11, No. 2, June 2003.

[12] Millán J. d. R., Rupp R., Müller-Putz G. R., Murray-Smith R., Giugliemma C., Tangermann M., Vidaurre C., Cincotti F., Kübler A., Leeb R., Neuper C., Müller K.-R. and Mattia D., "Combining brain–computer interfaces and assistive technologies: state-of-the-art and challenges", Frontiers in neuroscience, Review Article published 07 September 2010 doi:10.3389 fnins.2010. 00161

[13] García P. A., Spinelli E. M. and Toccaceli G. , "An Embedded System for Evoked Biopotential Acquisition and Processing", *International Journal of Embedded Systems (IJES).* Publication date: may, 2014. Volume:6, Issue:1, pages 86-93.

[14] Spinelli E. M., Martinez N. y Mayosky M., "A Single Supply Biopotential Amplifier". Medical Engineering and Physics, ISSN 1350-4533, Vol. 23/3, pp. 235-238, 2001.

[15] Wolpaw J. R, Ramoser H, McFarland D J and Pfurtscheller G, "EEG-based communication: improved accuracy by response verification". IEEE Trans. Rehabil. Eng. 6 326–33. 1998.