An embedded system for evoked biopotential acquisition and processing

Pablo Andrés García*

Laboratorio de Electrónica Industrial, Control e Instrumentación.
Departamento de Electrotecnia, Facultad de Ingeniería,
Universidad Nacional de La Plata,
Calle 48 y 116. CC91. CP 1900, La Plata, Buenos Aires, Argentina
E-mail: pagarcia@ing.unlp.edu.ar
*Corresponding author

Enrique Mario Spinelli and Graciela Mabel Toccaceli

Laboratorio de Electrónica Industrial, Control e Instrumentación,
Departamento de Electrotecnia, Facultad de Ingeniería,
Universidad Nacional de La Plata,
Calle 48 y 116. CC 91. CP 1900, La Plata, Buenos Aires, Argentina
and
Consejo Nacional de Investigaciones Científicas y Técnicas, CONICET,
Av. Rivadavia 1917. CP C1033AAJ, Ciudad Autónoma de Buenos Aires, Argentina
E-mail: spinelli@ing.unlp.edu.ar
E-mail: toccaceli@ing.unlp.edu.ar

Abstract: This work presents an autonomous embedded system for evoked biopotential acquisition and processing. The system is versatile and can be used on different evoked potential scenarios like medical equipments or brain computer interfaces, fulfilling the strict real-time constraints that they impose. The embedded system is based on an ARM9 processor with capabilities to port a real-time operating system. Initially, a benchmark of the Windows CE operative system running on the embedded system is presented in order 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 using two techniques: the fast Fourier transform and stimulus locked inter trace correlation, are also presented.

Keywords: embedded system; Micro2440; latency; stream interface driver; assistive device; brain computer interface; BCI; evoked potential.

Reference to this paper should be made as follows: García, P.A., Spinelli, E.M. and Toccaceli, G.M. (2014) 'An embedded system for evoked biopotential acquisition and processing', *Int. J. Embedded Systems*, Vol. 6, No. 1, pp.86–93.

Biographical notes: Pablo Andrés García received his Engineer in Electronics and MS from Universidad Nacional de la Plata (UNLP), La Plata, Argentina in 2002 and 2008, respectively. Since 2002, he has been with the Facultad de Ingeniería, UNLP, working on instrumentation. He is currently a Full Professor Assistant in Programación, Algoritmos y Estructuras de Datos with the Facultad de Ingeniería, UNLP. His current research interests are brain computer interfaces and embedded systems.

Enrique Mario Spinelli received his Engineer in Electronics, MS and PhD degrees from Universidad Nacional de La Plata (UNLP), La Plata, Argentina, in 1989, 2000 and 2005, respectively. Since 1990, he has been with the Industrial Electronics, Control, and Instrumentation Laboratory (LEICI), Facultad de Ingeniería, UNLP, working on scientific instrumentation. He is currently a Professor of Control Systems with the Facultad de Ingeniería, UNLP. His current research interests are analogue signal processing and brain control interfaces.

Graciela Mabel Toccaceli graduated from the Faculty of Engineering, National University of La Plata, in 1980. She was a Scientific Associate at MicroLab – INFN – International Center for Theoretical Physics (ICTP), Italy. Currently, she is a Full Professor in the Electrical Engineering Department at UNLP. Her primary area of interest is computer simulation applied to renewable energy conversion systems and real-time. Her research interests include real-time embedded system area.

1 Introduction

Evoked potentials (EPs) are neurophysiologic techniques commonly used to record brain or nerves responses to different types of sensory stimuli. The stimuli commonly used are visual, auditory or somatosensory. Brain response is typically recorded using electroencephalography (EEG) because it is a non-invasive practice. EPs can be used in a number of medical applications. Among others can be mentioned:

- Visual EPs to the optic nerve study (Evans and Boggs, 2012).
- Brainstem auditory evoked responses to estimate the threshold of hearing or locate hearing disorders (Strauss et al., 2004).
- Somatosensory EPs to determine the state of sensory pathways (Hu et al., 2002).
- Cognitive EPs such as the P300 to detect the cause of problems such as memory loss or impairment of orientation (Ozaki, 2007).

EPs are also used for the implementation of assistive devices (AD) for handicapped people, as brain computer interfaces (BCIs). Two of the most widely used EPs for BCIs are P300 (Serby et al., 2005) and steady state visual evoked potentials (SSVEPs) (Chi et al., 2012).

However, in diagnostic techniques as in BCI, the EP measurement conditions are very unfavourable. It is common to recover these EP signals embedded in SNR of 3 dB or worse (Wang et al., 2005) by means of statistical processing techniques.

The most widespread approach is coherent averaging, which consists of applying consecutive stimuli and recording the answer to each of them. Each of these answers is called a trace and through their averaging the signal to noise ratio (SNR) is improved. For such applications, it is essential to have a very good synchronisation between the applied stimuli and the recorded response, thus imposing strong real-time requirements on the hardware-software platform. While desktop computers with general purpose operating systems are usually used for EP diagnosis systems, they are not appropriate for ADs, because these devices require to be implemented in a reduced size and provide portability. Moreover, long boot time (up to a few minutes) also implies a serious drawback for personal ADs.

Many current BCI applications exist exclusively in academic research and are implemented by a front end biopotential acquisition with processing done on computer (Long et al., 2012; Yu et al., 2012). As mentioned by Millán et al. (2010), BCIs 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 needs to be replaced. In this sense, the authors propose the use of an embedded system as an integral solution presenting the implementation and measures to verify the proper operation. This pattern is a general tendency that repeats in other medical applications as mentioned in the work of Alouani et al. (2010).

2 Embedded system

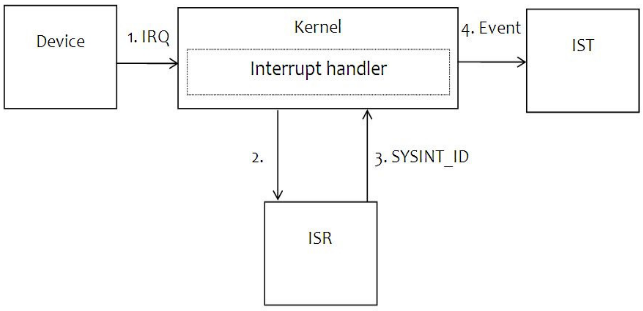
The system must be able to acquire EEG signals and generate stimulus synchronously. Sampling rates of 512 to 1,024 samples/sec are enough for most biomedical applications (Fazel-Rezai et al., 2007).

The system core is a 32 bits processor with high computing power, ability to carry a real-time operating system (RTOS), small size and low cost. The board selected as the platform was the Micro2440, based on the Samsung S3C2440A processor with ARM architecture and a processing speed of 400 MHz (Friendly Arm Project, http://www.friendlyarm.net/products/micro2440). This platform presents different ways for the operating system implementation. The first one is to develop a proprietary OS like the one presented by Schimpf (2013) for the arduino board. The other way is to use a general purpose OS like Android, Linux or Windows CE. In this case, the only one hard real-time OS is Windows CE, but there are no available interrupt latency measurements running on the Micro2440. So, the first step of this work was to characterise the interrupt latencies of the set: Windows CE 6.0 OS + Micro2440.

3 Interrupt latencies

The interrupt handling mechanism implemented in Windows CE is divided into two processes: the interrupt service routine (ISR) in kernel mode, and the interrupt service thread (IST) in user mode. Upon the occurrence of a particular device interrupt, the kernel invokes the ISR to determine the interruption source, mapping it into a SYSINT_ID (system interrupt identification). After identifying the interrupt source, the OS activates the event associated to the IST, where the tasks linked to the interrupt are implemented (Figure 1) (Pavlov and Belevsky, 2008).

Figure 1 Interrupt handling



To measure the latencies of the ISR and IST a stream interface driver was implemented.

3.1 Stream interface driver

This type of controller can be configured to be loaded by the device manager at boot time. It receives commands from applications and also from the device manager using file system calls.

To implement the controller there are ten needed functions that make up its interface and serve as the entry points. These functions are: xxx_Init, xxx_Deinit, xxx_Open, xxx_Close, xxx_IOControl, xxx_Read, xxx_Write, xxx_Seek, xxx_Powerup and xxx_Powerdown, where xxx is the name of the controller.

To measure the latencies, a stream interface driver was implemented to control a microprocessor's timer that generates the interrupts. In this case, the timer 0 was used because others were used by the scheduler and the profiler.

As a starting point for implementing the driver, a Windows CE 6.0 OS project supplied by the manufacturer (BSP: board support package) was used. The steps used during the implementation of the controller were similar to those described by Dong et al. (2010):

- The required code was added to implement the timer 0 driver interface formed by the ten functions previously described.
- A thread was generated within the controller to include the real-time task. It was connected to the timer interrupt by an event.
- The ISR code was modified within the kernel to handle the interrupt generated by the timer 0.
- The driver was added in the list of 'built-in' devices. In this way, it is started by the device manager during system start-up.
- The pin connected to the timer (T0) was activated so as to dispose of the execution time of the interruption as an external reference.
- The state of the general purpose pin (GPB1) was changed within the ISR to measure its latency externally. This pin is set high when entering the ISR.
- The GPB1 pin state was modified within the IST to measure its latency, from the high state imposed by the ISR to the low state.

3.2 Latency measurements

The timer was configured to generate interrupts at a frequency of 1 kHz. Figure 2 shows the marks generated by the controller to measure the latencies by watching the output pin related to the timer (T0) and GPB1.

Figure 3 shows the measurement setup implemented for determining latencies. This figure shows Micro2440SDK (bottom right), a digital oscilloscope (top right), a 225 MHz universal counter Model HP 53132A (right centre) and a PC screen (left). To record the measurements, the counter was configured to send the data to the PC through the serial port.

Figures 4 and 5 show the histograms of the latency measurements for ISR and IST, with and without additional processing load.

In order to measure latencies under processing load conditions, files were copied from a pen drive to a secure digital memory while other desktop tasks were performed during the test.

Figure 2 ISR and IST latencies

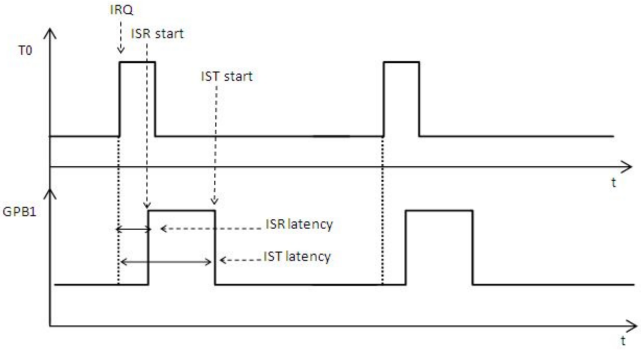
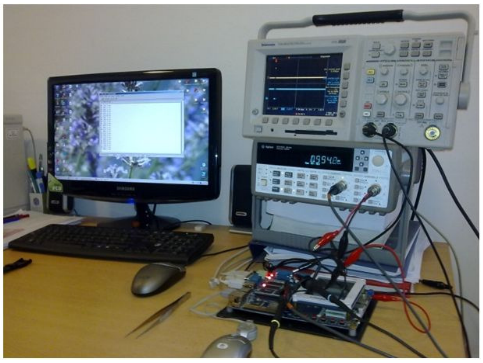


Figure 3 Measurement setup (see online version for colours)



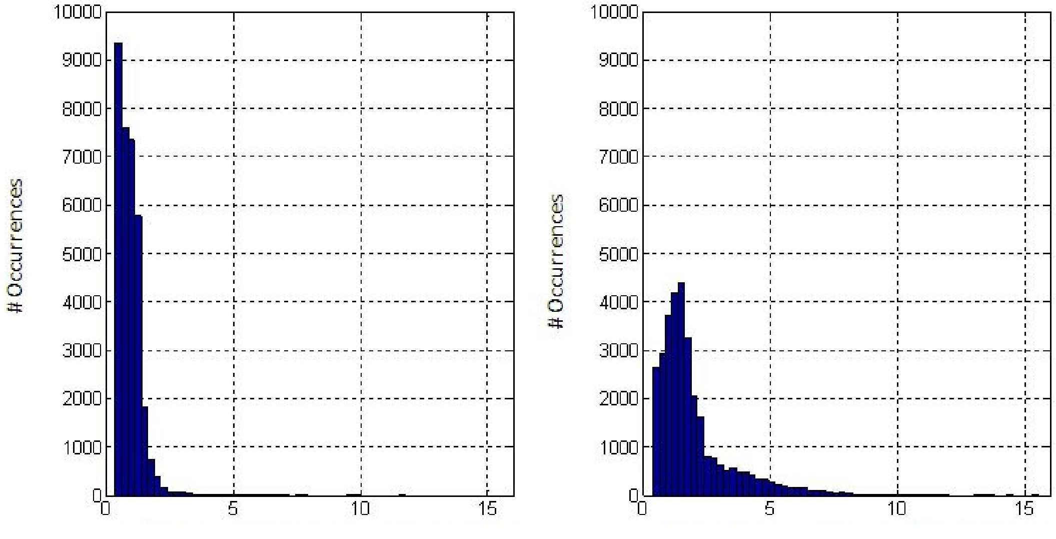
From the histograms of Figures 4 and 5, it can be seen that the maximum latency for ISR in these working conditions does not exceed 15 microseconds and the IST is less than 100 microseconds. Table 1 presents the maximum, minimum and average for each of the measurements.

Table 1 Interrupt latencies for different measure conditions

Measure condition	*Minimum*	*Average*	*Maximum*
ISR without load	0.5 µs	0.9 µs	11.6 µs
ISR with load	0.5 µs	1.9 µs	15.4 µs
IST without load	3.4 µs	6.5 µs	87.2 µs
IST with load	3.4 µs	21.4 µs	100.1 µs

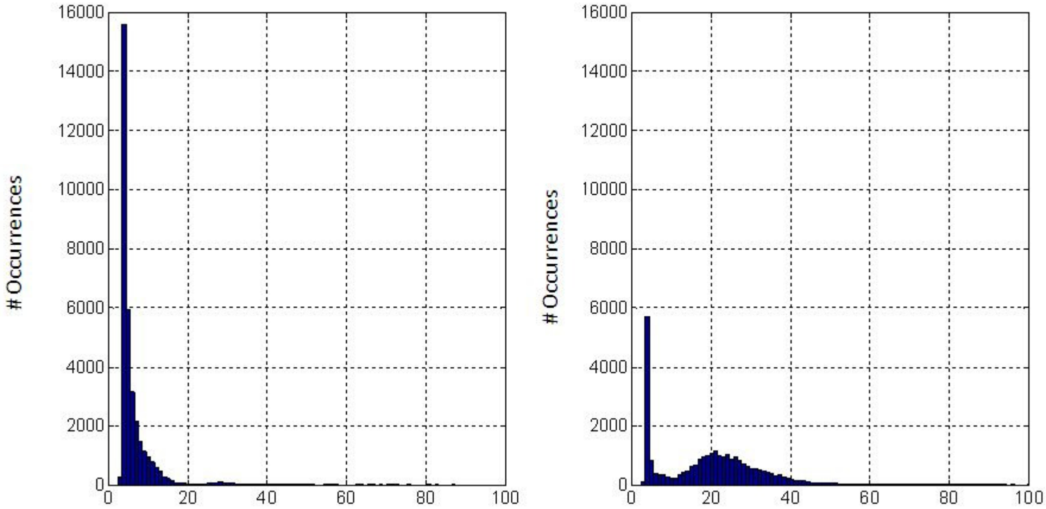
Based on these measurements, it can be concluded that the setup could be used to implement a real-time biopotential acquisition system digitising at a frequency rate of 1 kHz, with a synchronisation between external interrupt and stimulus generation better than 100 µs.

Figure 4 ISR latency, (a) ISR latency without processing load [μs] (b) ISR latency with processing load [μs] (see online version for colours)



(a) (b)

Figure 5 IST latency, (a) IST latency without processing load [μs] (b) IST latency with processing load [μs] (see online version for colours)



(a) (b)

4 Practical implementation

In order to test the proposed embedded system and its real-time capabilities, a SSVEP-based BCI was implemented.

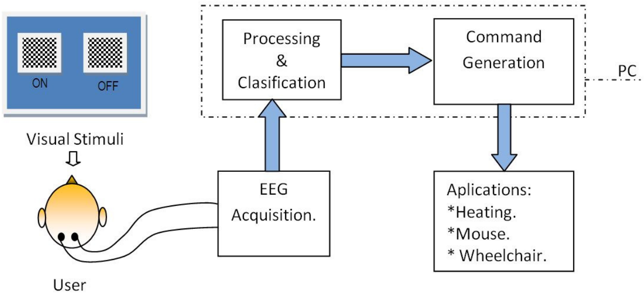
The EPs reflect brain activity produced by the visual information processing that can be registered in the occipital zone by EEG techniques. These potentials are presented in response to certain visual stimuli, and for this reason they are called evoked. There are two types of visual EPs: transient visual evoked potentials (TVEP) and SSVEPs. TVEPs occur when the visual stimulus rate is less than two stimuli per second, while SSVEPs occur for periodic stimuli with frequencies greater than 6 Hz (Gao et al., 2003). In the latter case, EPs overlap forming a steady state response with components in the stimulus frequency and its harmonics.

41 BCI based on SSVEP

Figure 6 shows a SSVEP-based BCI functional model. It is comprised of a visual stimulator, a biopotential acquisition stage, a processing stage and a command generator that handles the user assistance device.

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

Figure 6 Functional model of SSVEP-based BCI (see online version for colours)



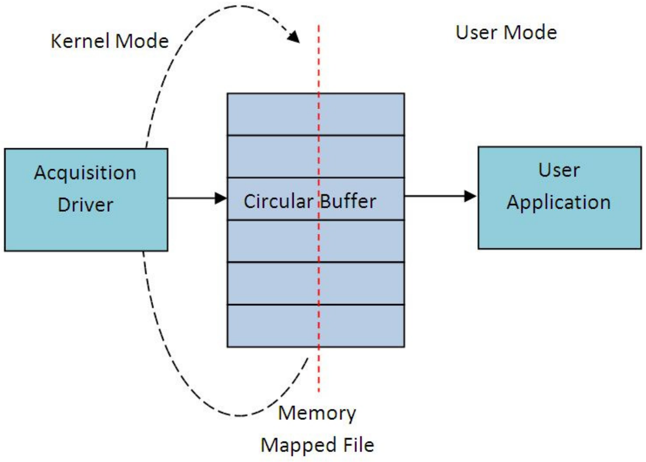
4.2 *Front-end*

To acquire the EEG signal a front-end based on the integrated circuit ADS1298 of Texas Instruments was used. It provides eight acquisition channels with one 24 bit Σ-Δ ADC converter per channel, which delivers data through a serial peripheral interface (SPI). Power supply and data communication were provided through an isolation barrier implemented by an integrated isolator ADUM6401 of analogue devices (Guerrero et al., 2013).

4.3 *Processing*

A stream interface driver was implemented to handle the information generated during the acquisition stage. To achieve real-time performance it should be included in the operating system image before compilation. The driver runs in kernel mode and implements a circular buffer of 1,024 samples per channel (24,576 bytes corresponding to 1,024 samples of eight channels with 3 bytes/sample). Through the management of memory mapped file resource, different user applications can manipulate the data generated by the driver (Figure 7).

Figure 7 Management of data generated by the driver (see online version for colours)



4.4 *Stimuli*

As visual stimulators, white LEDs arrangements flashing at two different frequencies were used. To maintain synchronism between the stimuli and the biopotential acquisition, the sampling frequency (1 kHz) was used as base time. The stimuli used were $f_1 = 14.28$ Hz and $f_2 = 20$ Hz, both generated directly from de sampling frequency divided by 70 and 50, respectively.

5 Experimental results

Some tests were made in order to check the possibility of implementing coherent averaging techniques to recover EPs. One differential channel in position F_z-O_z of the 10–20 international system with and without visual stimuli was recorded, preserving the stimulation information for synchronisation.

5.1 *Spectral analysis*

Figure 8 shows the spectral analysis performed on the recorded signal in three different moments.

The fast Fourier transform (FFT) on Figure 8(a) was performed in real-time on 1 second segments of signal (trace) when the user was not looking at the stimulus. Ten different traces of the signal can be seen in light blue and its average in black.

The FFT in Figure 8(b) corresponds to the condition of the user looking at the stimulus flashing at 14.28 Hz. In this case, the EP, in response to the stimulus at the fundamental frequency (14.28 Hz) with higher amplitude on the first harmonic frequency, can be distinguished. As mentioned (Wang et al., 2008), the amplitude of the first harmonic may be higher than the fundamental frequency on some subjects. The FFT in Figure 8(c) was performed when the user was looking at the stimulus flashing at 20 Hz. In this case, in accordance with the previous, the EP can be distinguished in response to the stimulus at the fundamental frequency (20 Hz) and its first harmonic.

5.2 *Time domain detection*

It was decided to test a novel time domain detection method known as stimulus locked inter trace correlation (SLIC) using EEG time-locked to stimulus onsets presented by Luo and Sullivan (2012), and Manyakov et al. (2010).

Figure 9 shows the temporal analysis performed on the recorded signal using white stimuli in three different moments.

The SLIC graphic of Figure 9(a) was performed to segments of signal time locked to the 14.28 Hz stimulus (T = 70 msec) when the user was not looking at it. The different traces of the signal can be seen in light blue, and in black, the average corresponding to the same time interval used in FFT. The SLIC graphic in Figure 9(b) was performed when the user was looking at the stimulus flashing at 14.28 Hz. In this case, it can be distinguish the EP in synchronism with the stimuli. In accordance with the spectral analysis, the first harmonic of the fundamental frequency is clearly observable.

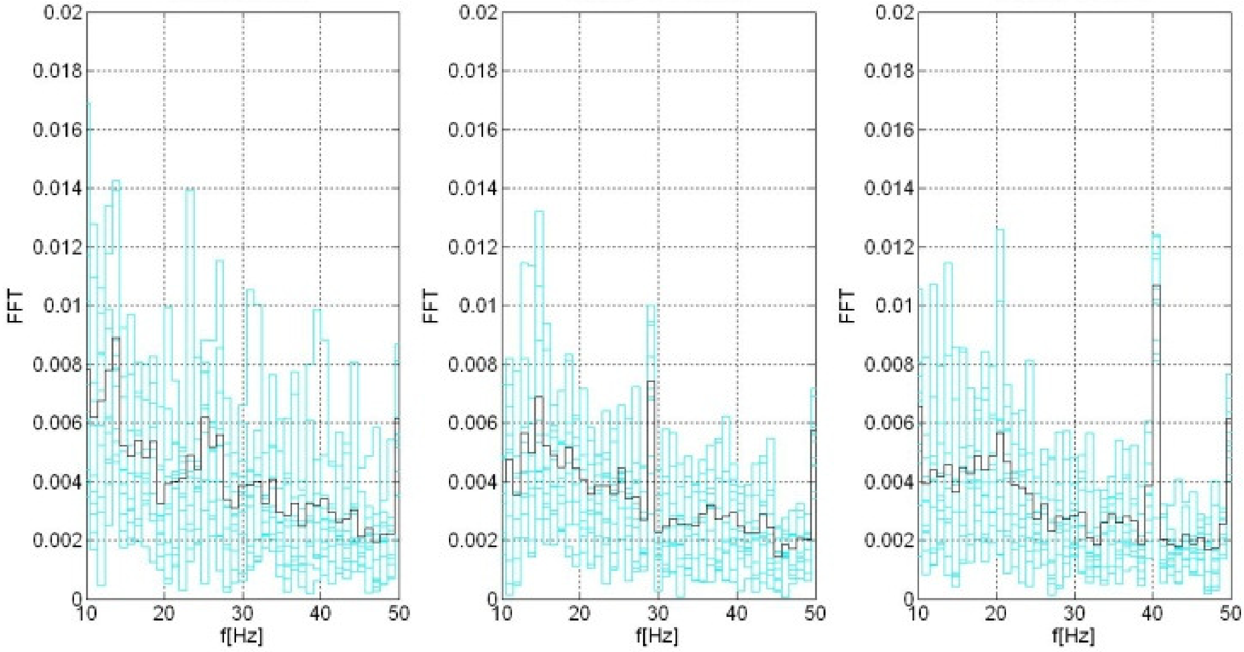
The SLIC graphic in Figure 9(c) corresponds to the user looking at the stimulus flashing at 20 Hz. In this case, the

EP cannot be distinguished using the time window corresponding to 14.28 Hz.

Figure 10 shows the same scenario, but using the analysis window time-locked to the 20 Hz stimuli (T = 50 ms).

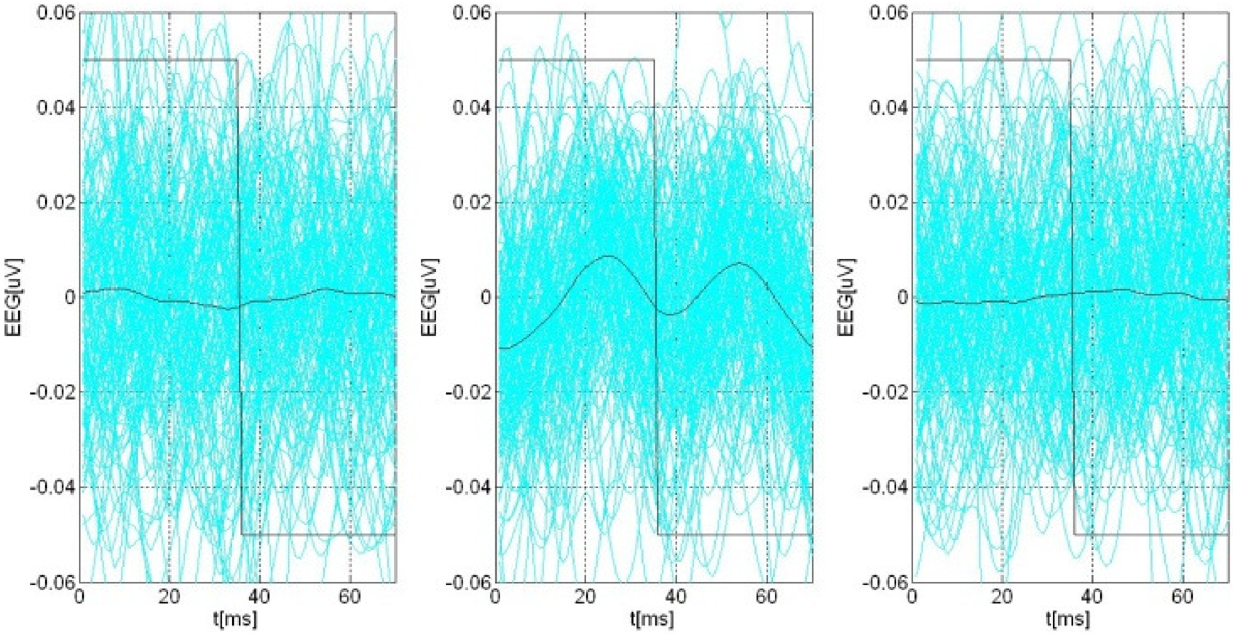
In this case, on the SLIC graphic in Figure 10(c), the visual EP time-locked to the 20 Hz stimuli can be seen.

Figure 8 EP with white stimuli, (a) without stimuli (b) stimuli in 14.28 Hz (c) stimuli in 20 Hz (see online version for colours)



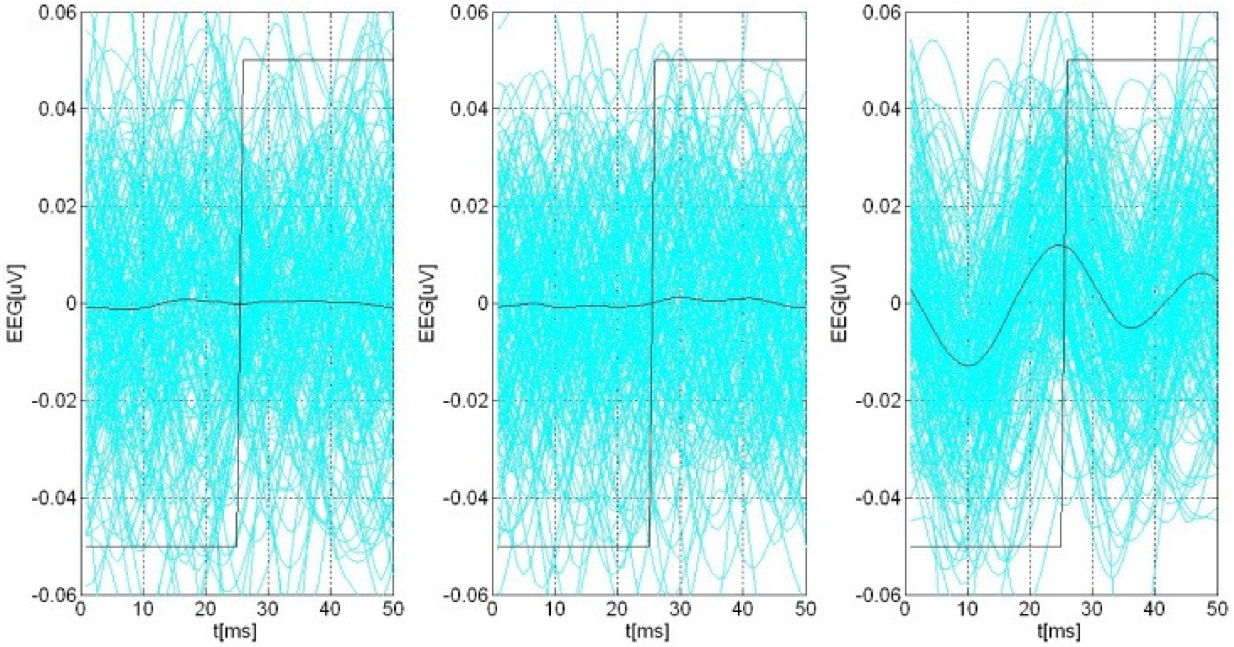
(a) (b) (c)

Figure 9 EP with white stimuli recovered using a 70 msec windows, (a) without stimuli (b) with white stimuli 14.28 Hz (c) with white stimuli 20 Hz (see online version for colours)



(a) (b) (c)

Figure 10 EP with white stimuli recovered using a 50 msec windows, (a) without stimuli (b) with white stimuli 14.28 Hz (c) with white stimuli 20 Hz (see online version for colours)



(a) (b) (c)

6 Conclusions

It has been demonstrated that a low cost general purpose portable embedded system can be used for evoked biopotential acquisition and processing. The embedded system implemented presents latency times of 15 µs for code in ISR and 100 µs for code in IST. The system is compact, portable and does not rely on a personal computer, presenting a booting time below 10 sec. A prototype was built and tested in real conditions, implementing a BCI based on steady state visual EPs with two stimuli. Experimental results show correct visual EP recoveries by means of two different methods: FFT and SLIC.

Acknowledgements

This work is supported in part by the projects: PIP-112-2009-0100253 from CONICET (‘Analog and digital processing of signals. biomedical applications in brain computer interfaces’) and I-167, UNLP (‘Procesamiento de señales e instrumentación electrónica en aplicaciones de física experimental y bioingeniería’).

References

Alouani, A., Elkeelany, O. and Abdallah, M. (2010) ‘Stand-alone portable digital body sound data acquisition device’, *Int. Journal of Embedded Systems*, Vol. 4, Nos. 3/4, pp.292–297.

Chi, Y.M., Wang, Y-T., Wang, Y., Maier, C., Jung, T-P. and Cauwenberghs, G. (2012) ‘Dry and noncontact EEG sensors for mobile brain computer interfaces’, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, March, Vol. 20, No. 2, pp.228–235.

Dong, X., Jianqun, L. and Jirong, W. (2010) ‘Research on real-time control of embedded NC system based on Windows CE 5.0’, *Mechanic Automation and Control Engineering (MACE)*, Issue Date: 26–28 June, pp.31–35, Print ISBN: 978-1-4244-7737-1.

Evans, A. and Boggs, J. (2012) ‘Clinical utility of evoked potentials’, *Medscape Reference, Drugs, Diseases & Procedures* [online] http://emedicine.medscape.com/article/1137451-overview#awaab6b3 (accessed 10 June 2013).

Fazel-Rezai, R., Pauls, M. and Slawinski, D. (2007) ‘A low-cost biomedical signal transceiver based on a Bluetooth wireless system’, *Proceedings of the 29th Annual International Conference of the IEEE EMBS*, Cité Internationale, Lyon, France, 23–26 August.

Friendly Arm Project, Micro2440SDK [online] http://www.friendlyarm.net/products/micro2440 (accessed 10 June 2013).

Gao, X., Xu, D., Cheng, M. and Gao, S. (2003) ‘A BCI-based environmental controller for the motion-disabled’, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, June, Vol. 11, No. 2, pp.137–140.

Guerrero, F.N., Spinelli, E. and Haberman, M. (2013) ‘Dispositivo Multicanal Para Adquisición De Biopotenciales’, *Segundas Jornadas de Investigación y Transferencia*, Facultad de Ingeniería, Universidad Nacional de La Plata, La Plata, 16 al 18 de abril de 2013.

Hu, Y., Luk, K.D., Lu, W.W. and Leong, J.C. (2002) ‘Comparison of time-frequency analysis techniques in intraoperative somatosensory evoked potential (SEP) monitoring’, *Computers in Biology and Medicine*, Vol. 32, No. 1, pp.13–23.

Long, J., Li, Y., Wang, H., Yu, T., Pan, J. and Li, F. (2012) '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*, September, Vol. 20, No. 5, pp.720–729.

Luo, A. and Sullivan, T.J. (2010) 'A user-friendly SSVEP-based brain-computer interface using a time-domain classifier', *Journal of Neural Engineering*, Vol. 7, No. 2, p.026010, 10pp.

Manyakov, N.V., Chumerin, N., Combaz, A., Robben, A. and Van Hulle, M.M. (2010) 'Decoding ssvep responses using time domain classification', *International Conference on Fuzzy Computation and 2nd International Conference on Neural Computation*, pp.376–380.

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. (2010) 'Combining brain-computer interfaces and assistive technologies: state-of-the-art and challenges', *Frontiers in Neuroscience*, Review Article published 7 September, DOI: 10.3389 fnins.2010.00161.

Ozaki, I. (2007) 'Changes in the primary sensory cortex during cognitive tasks', *IEEE/ICME: International Conference on Complex Medical Engineering, CME 2007*.

Pavlov, S. and Belevsky, P. (2008) 'Windows® Embedded CE 6.0 fundamentals', 11 August, ISBN-10: 0735626251.

Schimpf, P.H. (2013) 'ARTK: a compact real-time kernel for arduino', *Int. Journal of Embedded Systems*, Vol. 5, Nos. 1/2, pp.106–113.

Serby, H., Yom-Tov, E. and Inbar, G.F. (2005) 'An improved p300-based brain-computer interface', *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, March, Vol. 13, No. 1, pp.89–98.

Strauss, D.J., Delb, W. and Plinkert, P.K. (2004) 'Analysis and detection of binaural interaction in auditory evoked brainstem responses by time-scale representations', *Computers in Biology and Medicine*, September, Vol. 34, No. 6, pp.461–477.

Wang, B.Y., Gao, X., Hong, B., Jia, C. and Gao, S. (2008) 'Brain computer interfaces based on visual evoked potentials – feasibility of practical system designs', *IEEE Engineering in Medicine and Biology Magazine*, September/October, Vol. 27, No. 5, pp.64–71.

Wang, Y., Wang, R., Gao, X. and Gao, S. (2005) 'Brain-computer interface based on the high-frequency steady-state visual evoked potential', *2005 First International Conference on Neural Interface and Control Proceedings*, Wuhan, China, 26–28 May.

Yu, Y-C., Nawroj, A., Wang, S. and Gabel, L. (2012) 'Mobile robot navigation through a brain computer interface', *Signal Processing in Medicine and Biology Symposium (SPMB), 2012 IEEE*.