

Sistemas Embebidos de Tiempo Real con aplicaciones en Bioingeniería



UNIVERSIDAD
NACIONAL
DE LA PLATA



Tesis de Doctorado

Pablo Andrés García

Presentada ante la Facultad de Ingeniería de la Universidad Nacional de La Plata como requisito para la obtención del grado académico de DOCTOR EN INGENIERÍA

Dirección de tesis:

Dr. Enrique M. Spinelli
Ing. Graciela M. Toccaceli

Jurado de Tesis:

Dr. Rubén Acevedo
Dr. Pablo Diez
Dr. Fernando Farfán

Febrero de 2019

La Plata, Buenos Aires, Argentina

.....con orgullo y cariño dedicado a:
mis padres Rubén y Mari
mi esposa Valeria
mis hijos Fran y Mora

Resumen

El avance de la tecnología permite abordar problemas cada vez más complejos con implementaciones cada vez más compactas. Un claro ejemplo de esto son los sistemas embebidos (SE): dispositivos electrónicos compactos y autónomos, con capacidad de cómputo, que realizan procesamiento de datos y/o control sobre variables físicas externas. La mayor diferencia que presenta un SE, respecto de un computador personal (PC), es que un SE está dedicado a una función particular para la cual fue desarrollado, mientras que un PC está concebido para usos múltiples. Además, los recursos de hardware que dispone un SE son generalmente más reducidos y deben afrontar importantes restricciones de consumo.

Ejemplos actuales de equipos que contienen sistemas embebidos son: un router, un teléfono celular, un lavarropas, un equipo reproductor de audio o la unidad central de control de un automóvil, entre muchos otros. También son múltiples y diversas las aplicaciones biomédicas que requieren SE como dosificadores de drogas, monitores de parámetros fisiológicos, interfaces cerebro-computadora y equipos de diagnóstico autónomos. Una característica distintiva de estas aplicaciones es que deben cumplir con estrictas restricciones en los tiempos de respuesta y de ejecución para las tareas que deben realizar, por lo cual exigen sistemas embebidos de tiempo real. En general, las aplicaciones biomédicas requieren la captura de señales, su procesamiento y además de generar estímulos en forma sincronizada con la adquisición de las señales. Estas demandas imponen serias restricciones, tanto en el software como en el hardware de este tipo de equipos.

Esta tesis propone una metodología de diseño de SE para aplicaciones biomédicas. A partir de ejemplos concretos se describe el fraccionamiento de las tareas; en primer lugar entre el procesamiento analógico y el procesamiento digital de señales, y luego entre distintas plataformas sobre las cuales se implementan las tareas de procesamiento digital. Los dispositivos desarrollados en el marco de esta tesis son: un adquisidor de señales de EEG autónomo con transmisión inalámbrica para Interfaces Cerebro-Computadora (ICC) basada en ritmos cerebrales, un equipo para diagnóstico de hipoacusias basado en Oto-emisiones Acústicas (OAE) diseñado para una empresa de audiología que actualmente lo comercializa y una plataforma para implementación de ICC basada en potenciales evocados visuales de estado estacionario (del inglés SSVEP: Steady State Visual Evoked Potential)

Los dispositivos desarrollados, que tienen distintos grados de complejidad y requerimientos particulares, abarcan desde pequeños SE basados en microcontroladores de 8 bits, a plataformas con microprocesadores de 32 bits con sistemas operativos de tiempo real. Cada uno de estos equipos resultaron como soluciones propias y apropiadas a problemas específicos de bioingeniería e incorporan aportes originales en distintos aspectos de los SE.

Índice

Parte 1: Fundamentos.	6
1. Introducción.	7
1.1. <i>Sistemas embebidos para aplicaciones biomédicas.</i>	8
1.2. <i>Descripción de la tesis.</i>	9
1.3. <i>Producido de la tesis.</i>	9
1.3.1. <i>Transferencia de Tecnología.</i>	9
1.3.2. <i>Publicaciones científicas.</i>	9
2. Sistemas Embebidos.	11
2.1. <i>Definición.</i>	11
2.2. <i>Difusión de Sistemas embebidos en Argentina y en el mundo</i>	11
2.3. <i>Hardware de Sistemas Embebidos</i>	13
2.5.1. <i>Microcontrolador</i>	13
2.5.2. <i>Microprocesador</i>	14
2.5.3. <i>DSP</i>	15
<i>SoC</i>	16
2.4. <i>Plataformas para SE</i>	17
2.6.1. <i>Microcontrolador AduC</i>	18
2.6.2. <i>Arduino</i>	18
2.6.3. <i>Placa LPCXpresso</i>	19
2.6.4. <i>Micro2440SDK</i>	20
2.6.5. <i>dsPIC</i>	22
2.6.6. <i>SoC DE10-Nano</i>	24
2.5. <i>Firmware</i>	26
2.6. <i>Sistema Operativo Embebido</i>	26
2.9.1. <i>SO de procesamiento secuencial</i>	27
2.9.2. <i>SO primer plano/segundo plano</i>	28
2.9.3. <i>SO de tiempo real cooperativo y expropiativo</i>	28
3. Interfaces Cerebro Computadora.	30
3.1. <i>Definición.</i>	30
3.2. <i>Evolución y Estado actual.</i>	30
3.3. <i>Estructura clásica de una ICC.</i>	31
3.4. <i>Clasificación de las ICC según la característica del EEG</i>	32
3.5. <i>Potenciales Evocados Visuales.</i>	33
3.6. <i>Interfaz Cerebro Computador basada en SSVEP.</i>	34
3.7. <i>Plataformas de Hardware-Software para ICC.</i>	35
3.7.1. <i>NCAN (National Center for Adaptive Neurotechnologies).</i>	35
3.7.2. <i>Proyecto OpenBCI</i>	35
3.7.3. <i>G.tec</i>	36

3.7.4. <i>OpenVibe</i>	38
3.7.5. <i>Emotiv</i>	39
3.8. <i>Motivación para la implementación de plataformas propias para ICC.</i>	40
Parte 2: Desarrollo e Implementación de Sistemas Embebidos.	42
4. Sistema Embebido configurable para investigación en Interfaces Cerebro Computadora.	43
4.1. <i>Introducción</i>	43
4.2. <i>Esquema general del Sistema embebido.</i>	44
4.3. <i>Adquisición de EEG</i>	45
4.4. <i>Núcleo.</i>	47
4.5. <i>Sistema Operativo Embebido.</i>	47
4.6. <i>Módulo de comunicación.</i>	49
4.6.1. <i>Topología de Red Bluetooth.</i>	49
4.6.2. <i>Topología de Red Zigbee.</i>	52
4.6.3. <i>Comparativa</i>	53
4.7. <i>Resultados experimentales</i>	54
4.8. <i>Conclusión</i>	57
4.9. <i>Sistema embebido monocanal.</i>	57
4.10. <i>Resultados experimentales con SSVEP</i>	58
4.10.1. <i>Estímulo visual (primeros ensayos).</i>	58
4.10.2. <i>Conclusión.</i>	64
5. Interfaz Cerebro – Computadora autónoma basada en SSVEP	65
5.1. <i>Introducción.</i>	66
5.2. <i>Núcleo</i>	66
5.3. <i>Sistema operativo embebido.</i>	66
5.3.1. <i>Latencias de Interrupción</i>	67
5.3.2. <i>Controlador: Stream interface driver</i>	68
5.3.3. <i>Medidas</i>	69
5.4. <i>Primeras pruebas con potenciales evocados visuales</i>	72
5.4.1. <i>Adquisición de EEG</i>	72
5.4.2. <i>Estimulador visual y aplicación de usuario</i>	73
5.4.3. <i>Resultados experimentales con procesamiento off-line.</i>	73
5.4.4. <i>Análisis espectral.</i>	74
5.4.5. <i>Detección en el dominio del tiempo</i>	74
5.4.6. <i>Puesta a punto del algoritmo de recuperación de potenciales evocados</i>	76
5.5. <i>Implementación práctica de ICC: Un deletreador embebido híbrido</i>	78
5.5.1. <i>Introducción.</i>	78
5.5.2. <i>Sistema embebido.</i>	79
5.5.2.1. <i>Núcleo.</i>	79
5.5.2.2. <i>Adquisición de EEG</i>	79
5.5.3. <i>Software de alto nivel.</i>	80
5.5.4. <i>Algoritmo: Densidad Espectral de Potencia</i>	81
5.5.5. <i>Experimentos</i>	83
5.5.6. <i>Resultados</i>	83
5.5.7. <i>Conclusión</i>	85

6.	Sistema Embebido para evaluación de la capacidad auditiva basado en Emisiones Otoacústicas (OAE).	86
6.1.	<i>Breve descripción del Sistema Auditivo.</i>	86
6.2.	<i>Otoemisiones acústicas.</i>	87
6.3.	<i>Determinación de la capacidad auditiva mediante OAE.</i>	88
6.3.1.	<i>SOAE: Spontaneous Otoacoustic Emissions.</i>	88
6.3.2.	<i>TEOAE: Transient Evoked Otoacoustic Emissions</i>	88
6.3.3.	<i>DPOAE: Distortion Product Otoacoustic Emission</i>	88
6.4.	<i>Procedimiento de medida</i>	89
6.5.	<i>Desarrollo del equipo de medida.</i>	90
6.6.	<i>Esquema general del equipo de medida.</i>	90
6.7.	<i>Probeta y Amplificador.</i>	91
6.8.	<i>Núcleo.</i>	91
6.8.1.	<i>Estructura del sistema adquirente/generador de audio.</i>	92
6.8.2.	<i>Selección del Codec de audio.</i>	93
6.8.3.	<i>Selección del DSP.</i>	93
6.8.4.	<i>Generador de audio.</i>	93
6.8.5.	<i>Sistema operativo embebido.</i>	94
6.9.	<i>Módulo de comunicación.</i>	95
6.10.	<i>Computadora.</i>	95
6.10.1.	<i>Algoritmos para la detección de otoemisiones.</i>	95
6.10.1.1.	<i>Promediación coherente</i>	95
6.10.1.2.	<i>Comparación de Algoritmos por simulación.</i>	96
6.10.1.3.	<i>Promediación Coherente en DPOAE</i>	97
6.10.1.4.	<i>Promediación de Espectros en DPOAE</i>	98
6.10.1.5.	<i>Promediación Coherente Vs. Promediación de espectros en DPOAE</i>	99
6.10.2.	<i>Programación en computadora.</i>	99
6.11.	<i>Conclusiones.</i>	100

Parte 3: Resultados. 102

7.	Conclusiones y Líneas Futuras.	103
7.1.	<i>Conclusiones particulares para los dispositivos implementados.</i>	104
7.1.1.	<i>Adquirente de señales de EEG autónomo con transmisión inalámbrica para Interfaces Cerebro-Computadora (ICC).</i>	104
7.1.2.	<i>ICC embebida basada en SSVEP.</i>	105
7.1.3.	<i>Equipo para diagnóstico de hipoacusias basado en Oto-emisiones Acústicas (OAE).</i>	105
7.2.	<i>Conclusiones por escenario.</i>	105
7.2.1.	<i>Escenario 1: SE simple para adquisición y PC para procesamiento e interfaz de usuario.</i>	106
7.2.2.	<i>Escenario 2: SE compuesto para adquisición y estimulación, y PC para procesamiento e interfaz de usuario.</i>	107
7.2.3.	<i>Escenario 3: SE compuesto para adquisición, estimulación, procesamiento e interfaz de usuario</i>	108
7.3.	<i>Conclusiones generales y Líneas Futuras.</i>	109

Parte 1: Fundamentos.

Capítulo 1: Introducción.

En nuestra vida cotidiana convivimos y utilizamos múltiples y diversos sistemas embebidos (SE). Reportes bibliográficos de hace 10 años, indicaban que en una casa típica o en un vehículo podían ya encontrarse más de 100 procesadores embebidos en su interior [Ping, 2008] [Ashrab y Ghazali, 2011], y que actualmente más del 90 % del código que se desarrolla es para sistemas embebidos [Arafa et al., 2017] [Hartenstein, 2004] [Fernandes y Machado, 2007].

Dentro de los SE, aquellos de tiempo real son un caso particular que plantea numerosos problemas a resolver en campos diversos como circuitos digitales y microprocesadores, sistemas operativos de tiempo real [Laplante, 1992], procesamiento digital de señales [Proakis y Manolakis, 1996], procesamiento analógico de señales [Pallás Areny y Webster, 1999], diseño de hardware y desarrollo de software. El tema es de actualidad y está en continuo desarrollo, siendo habituales las publicaciones científicas sobre la implementación de estos sistemas [Stanciu et al., 2017] [Zulzilawat et al., 2017] [García et al., 2014.a] [Short, 2008].

Los procesadores utilizados en SE pueden ser desde un simple microcontrolador de 4 bits como los presentes en juguetes, hasta poderosos procesadores de 64 o 128 bits diseñados a medida. De la misma forma, algunos de estos sistemas ejecutan pequeños programas escritos en lenguaje ensamblador formado por unas pocas líneas y sin sistema operativo, mientras que otros más complejos pueden utilizar sistemas operativos de tiempo real con programas multitarea escritos en C o C++ [Aspencore, 2017]. Actualmente es usual la utilización de versiones modificadas de los sistemas operativos Linux y Windows sintonizados para sistemas embebidos.

Una etapa muy importante, en la cual se integra el procesador embebido, y con él la tecnología de hardware-software necesaria para la implementación, es la etapa de procesamiento. De acuerdo con los requerimientos de la aplicación, el procesamiento digital de señales puede fraccionarse entre circuitos digitales dedicados o programables (FPGA) y sistemas microprocesados. Los primeros pueden ejecutar tareas a muy alta velocidad, pero son rígidos, mientras que los dispositivos que ejecutan programas son flexibles y pueden adaptarse a distintas condiciones, inclusive en tiempo de ejecución. Esto establece las pautas generales para asignar las tareas entre estos bloques: las tareas fijas con alta demanda deben ejecutarse sobre hardware dedicado y aquellas no tan exigentes mediante software.

Cuando se analiza el software que deben portar los Sistemas Embebidos son varios los escenarios posibles en función de los requerimientos de la aplicación. Un primer punto a tener en cuenta es la necesidad de utilizar un SO o programar directamente la aplicación (normalmente conocido como: “programación bare-metal”). Dependiendo del nivel de procesamiento requerido, la interfaz de usuario necesaria, el soporte o no para multitarea, las restricciones de tiempo real, y la plataforma de hardware que se utilice, será conveniente uno u otro escenario.

En vista de la amplia gama de alternativas posibles para la implementación del software, la división del mismo dependerá de los requerimientos particulares de la aplicación embebida. En términos generales se puede plantear la siguiente división:

- *Software de tiempo real duro*: cuando se hace sobre una plataforma que utiliza un SO con soporte para tiempo real este tipo de software formará parte de un controlador (driver) encargado de manejar la tarea. Cuando se hace la programación directa, es decir sin SO, el software de tiempo real duro se programa con un lenguaje de bajo nivel (C o assembler) teniendo especial control sobre los tiempos de ejecución. En otros escenarios de tiempo

real se puede utilizar lógica programable para dividir la solución entre hardware y software.

- *Software de tiempo real blando*: las opciones para implementar este tipo de software son similares a las previas a diferencia que al no ser de tiempo real duro se puede utilizar un SO embebido de propósito general (sin prestaciones de TR) que permita cumplir con las restricciones temporales de la tarea “casi” siempre. Es decir, se considera aceptable que el SE en algún momento no pueda verificar las restricciones temporales.
- *Software de aplicación*: típicamente el objeto de este software es implementar la interfaz de usuario y las tareas de baja prioridad que no presentan restricciones temporales. En las soluciones que utilizan un SO de propósito general es común programar estas aplicaciones en lenguajes de alto nivel usando el paradigma orientado a objetos y reutilizando las librerías disponibles. Cuando no se utiliza un SO se suele programar en lenguaje C, y dependiendo de la plataforma de hardware utilizada y sus herramientas de diseño se reutilizan librerías o bien se programa lo que sea necesario.

1.1. Sistemas embebidos para aplicaciones biomédicas.

Las aplicaciones biomédicas en general requieren captura de señales y la generación de estímulos o acciones de control en correspondencia con ellas, exigiendo SEs de tiempo real. Las soluciones basadas en ordenadores personales (PC) son, por su flexibilidad, una alternativa interesante para ser utilizadas en investigación, pero no para implementaciones finales: el tiempo de arranque, el tamaño y su fiabilidad, no son apropiados para dispositivos de uso continuo.

Los sistemas embebidos generalmente interactúan con su entorno a partir de la captura de señales y de la generación de acciones de control. En las aplicaciones biomédicas, el paciente -o el usuario de un dispositivo- forma parte de este entorno. En algunos casos el SE registra señales que se generan en el usuario en forma espontánea, por ejemplo, su electrocardiograma (ECG), pero en otras las señales capturadas se producen en respuesta a estímulos precisos: son evocadas.

El registro de señales evocadas es un caso frecuente en aplicaciones biomédicas. Dado que estas señales se producen en respuesta a un estímulo, utilizando técnicas de promediación coherente, es posible detectar respuestas que de otro modo quedarían inmersas en el ruido base. Estas técnicas consisten, básicamente, en aplicar repetidamente estímulos y registrar las respuestas a cada uno de ellos. Si es posible considerar que el proceso es estacionario, las señales registradas como respuestas a cada uno de estos estímulos $x_i(t)$ estará compuesta por la respuesta evocada $r(t)$ más componentes de tipo aleatorio debido a aportes de ruido de diversas fuentes. Luego, promediando un número significativo de realizaciones $x_i(t)$, es posible “rescatar” $r(t)$, aunque su amplitud sea inferior al ruido presente.

Un SE para adquirir respuestas evocadas debe ser capaz de generar estímulos en forma sincronizada con la adquisición de las respuestas, lo cual impone restricciones de tiempo real y fuertes interrelaciones entre los distintos elementos de hardware y procesos que conviven en él. Un esquema que describe estas interrelaciones se muestra en la Figura 1.2.

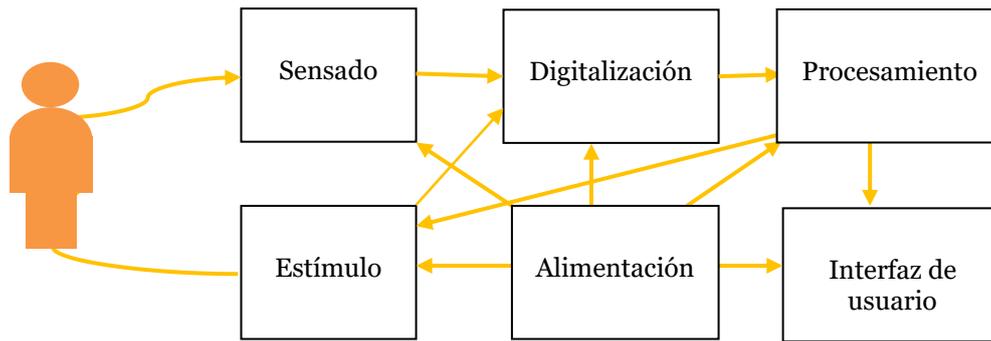


Figura 1.2. Esquema general de las etapas de un equipo para captura de potenciales evocados.

1.2. Descripción de la tesis

En este capítulo se presenta una introducción general a los requerimientos que las aplicaciones biomédicas imponen sobre los sistemas embebidos, los cuales se analizan en el Capítulo 2. En el Capítulo 3 se detallan las características de las Interfaces Cerebro-Computadora (ICC), profundizando en aquellas basadas en potenciales evocados. En los capítulos 4, 5 y 6 se presentan los equipos desarrollados en el marco de esta tesis, que corresponden a distintos escenarios y distintas soluciones tanto de hardware como de software. Finalmente, en el Capítulo 7 se detallan las conclusiones de esta tesis y las líneas a desarrollar en el futuro.

1.3. Producido de la tesis

Esta tesis ha dado lugar a una transferencia de tecnología y a publicaciones científicas.

1.3.1. Transferencia de Tecnología:

El equipo de diagnóstico para hipoacusia por otoemisiones acústicas fue transferido a la empresa de audiología LADIE, que actualmente lo comercializa (www.ladie-audiologia.com). Este desarrollo fue realizado a través de un subsidio FONTAR de la Agencia Nacional de Promoción Científica y Tecnológica y, debido a cuestiones de confidencialidad, aquellos detalles de los circuitos, software embebido y de la implementación de este equipo que permitirían reproducirlo, no se incluyen en la tesis. Sí se consignan la estrategia de diseño, la cual es objeto de esta tesis, y resultados experimentales que han dado origen a la siguiente publicación en conjunto con la empresa solicitante del desarrollo:

- [1]. García Pablo A, Spinelli Enrique, Guaraglia Dardo, Daffunchio Martín, Bonti Horacio, Velis Ariel, Berreta Ricardo. “Desarrollo de un equipo para evaluación de capacidad auditiva basado en Emisiones Otoacústicas (OAE)”, Primeras Jornadas de Investigación y Transferencia, Facultad de Ingeniería, UNLP, La Plata, Argentina. Abril de 2011.

1.3.2. Publicaciones científicas:

- [2]. Pablo A. García, Enrique M. Spinelli and Graciela M. Toccaceli. “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

- [3]. Pablo A. García, Enrique M. Spinelli y Graciela M. Toccaceli, "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.
- [4]. Pablo A. García, Enrique M. Spinelli y Graciela M. Toccaceli. "Interfaz cerebro computadora embebida". VI Congreso Latinoamericano de Ingeniería Biomédica (CLAIB 2014). Paraná, Entre Ríos, Argentina. Octubre de 2014.
- [5]. Pablo A. García, Graciela M. Toccaceli y Enrique M. Spinelli." Interfaz cerebro computadora embebida basada en potenciales evocados visuales de estado estacionario". Segundas Jornadas de Investigación y Transferencia, Facultad de Ingeniería, UNLP, La Plata, Argentina. Abril de 2013.
- [6]. García Pablo A, Toccaceli Graciela M. y Spinelli Enrique M. "Latencias de Interrupción de Windows CE 6.0, Análisis y Medición de su Performance sobre la Plataforma Micro2420", Congreso Argentino de Sistemas Embebidos CASE 2012. Buenos Aires, Argentina. Agosto de 2012.
- [7]. García Pablo A, Spinelli Enrique M, Toccaceli Graciela M, Haberman Marcelo A. "Interfaz Cerebro Computador basada en potenciales evocados visuales de estado estacionario: ensayos preliminares", XVIII Congreso Argentino de Bioingeniería SABI 2011. Mar del Plata, Argentina. Septiembre de 2011.
- [8]. García Pablo A, Haberman Marcelo y Spinelli Enrique. "A Versatile 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.
- [9]. Marcelo Haberman, Graciela Toccaceli, Enrique Spinelli y Pablo A. García. "Arquitectura software de tiempo real para interfaz cerebro-computadora", Anales del XXII Congreso Argentino de Control Automático AADECA 2010. Ciudad de Buenos Aires, 2010.
- [10]. Pablo A. García, Enrique M. Spinelli y Marcelo A. Haberman. 2009. "Plataformas de Hardware para Interfaces Cerebro Computadora". Presentado en II Jornadas Argentinas sobre Interfaces Cerebro Computadora (JAICC 2009).
- [11]. Enrique M. Spinelli, Pablo A. García y Marcelo A. Haberman. 2009. "Paradigmas ICC para validación de Plataformas autónomas". Presentado en II Jornadas Argentinas sobre Interfaces Cerebro Computadora (JAICC 2009).
- [12]. Marcelo A. Haberman, Pablo A. García y Enrique M. Spinelli. 2009. "Comparación de dos amplificadores de EEG para Interfaces Cerebro-Computadora". Presentado en II Jornadas Argentinas sobre Interfaces Cerebro Computadora (JAICC 2009).

Capítulo 2: Sistemas Embebidos.

Resumen

En este capítulo se presenta el estado del arte sobre “Sistemas Embebidos” y se describe el panorama actual de los mismos a nivel mundial y local. Además, se presentan las distintas alternativas para el procesador embebido y se detalla el hardware disponible. Como un caso especial, se describen los sistemas embebidos de tiempo real y se presentan las distintas variantes existentes para la implementación del Sistema Operativo para SEs.

2.1. Definición.

Se denomina sistema embebido (SE) o empotrado a un sistema electrónico con capacidad de cómputo diseñado para ejecutar una o varias tareas bien específicas, a diferencia de una computadora personal que es para usos múltiples. En general los sistemas embebidos poseen menos capacidad de cómputo que las computadoras personales y se diseñan optimizando el tamaño y el consumo de energía. Se denominan embebidos porque forman parte de un dispositivo mayor con partes mecánicas o electromecánicas. Los SE están integrados a nuestro cotidiano. Ejemplos típicos de SE pueden ser: un lavarropas, un router, un teléfono inteligente, el controlador de vuelo de un avión, el control de estabilidad y frenado de un auto, entre muchos otros.

La necesidad de desarrollo constante de estos dispositivos ha dado lugar a una nueva especialización en el ámbito de la electrónica digital conocida como “sistemas embebidos” y es común encontrar distintas organizaciones que enmarcan los tópicos de interés de esta nueva especialización.

2.2. Difusión de Sistemas embebidos en Argentina y en el mundo

Entre los distintos actores a nivel mundial en la electrónica embebida se destaca el grupo de medios “Aspencore”, el cual fue comprado por Arrow Electronics en 2016. Presenta sitios de noticias, productos, diseño y herramientas, entre otros. [Aspencore Media Guide, 2017]. El tema de los SE también está instalado en Argentina, tanto en la comunidad de la industria electrónica como en el ámbito académico, a partir de la realización de talleres, congresos y simposios.

Simposio Argentino de Sistemas Embebidos – Congreso Argentino de Sistema Embebidos:

El Simposio Argentino de Sistemas Embebidos (SASE) es un evento anual que reúne a la comunidad académica y a la industria en torno a la temática de los sistemas embebidos. Como parte del SASE se realizan Tutoriales, Workshops y el Congreso Argentino de Sistemas Embebidos (CASE), donde se presentan cada año numerosos trabajos de investigación y desarrollo, que se publican en distintas categorías en forma gratuita. Además, se realiza un concurso de proyectos estudiantiles, organizado en diferentes categorías, un Programa de equipamiento para universidades y un Programa de becas de alojamiento para estudiantes de la República Argentina y otros países de la región. Todo esto se hace para lograr cumplir con los objetivos del SASE, que son los siguientes:

- Difundir en el ámbito profesional y académico las tecnologías asociadas a los sistemas embebidos.
- Fomentar la interacción industria-academia en temas asociados a los sistemas embebidos.
- Incentivar entre los estudiantes universitarios y los jóvenes profesionales el interés por los sistemas embebidos.

Asociación Civil para la investigación, promoción y desarrollo de Sistemas Electrónicos embebidos (ACSE): La ACSE fue creada en julio de 2011 y es una institución sin fines de lucro cuyos propósitos son:

- Difundir en el ámbito profesional y académico las tecnologías asociadas a los sistemas embebidos.
- Fomentar la interacción industria-academia en temas asociados a los sistemas embebidos.
- Incentivar entre los estudiantes universitarios y los jóvenes profesionales el interés por los sistemas embebidos.
- Generar un espacio de intercambio entre las empresas del sector, que promueva el desarrollo de la industria nacional.

La ACSE comenzó a operar en 2012, siendo sus integrantes profesores universitarios de diferentes ciudades de la República Argentina. Desde 2012 a la actualidad organiza el Simposio Argentino de Sistemas Embebidos, que cada año cuenta con alrededor de 1.500 participantes de todo el país y de la región y diversas actividades. Desde 2013 la ACSE tiene una intensa articulación con la Red Universitaria de Sistemas Embebidos (Red RUSE) del Consejo Federal de Decanos de Ingeniería (CONFEDI) y organiza Talleres de discusión, Escuelas para la formación de docentes, etc. A su vez desde 2013 la ACSE tiene una activa vinculación con la Cámara Argentina de Industrias Electrónicas, Electromecánicas y Luminotécnicas, CADIEEL, con quien coordina el desarrollo de la Computadora Industrial Abierta Argentina (Proyecto CIAA). Este sitio web busca difundir todas estas actividades y fomentar los lazos de la ACSE con instituciones, empresas y personas interesadas en la investigación, promoción y desarrollo de los sistemas electrónicos embebidos.

Red RUSE y Articulaciones

A partir de una propuesta del Consejo Federal de Decanos de Ingeniería (CONFEDI) el 12 de septiembre de 2013 se creó la Red Universitaria de Sistemas Embebidos, que está actualmente integrada por 57 Unidades Académicas de diferentes ciudades de la República Argentina y que permite establecer formalmente articulaciones e iniciativas de interés general.

Los objetivos de la RUSE son:

- Promover la capacitación de los recursos humanos en temáticas relacionadas con los sistemas embebidos y el diseño digital a través de espacios de formación académica como el auspicio de cursos de extensión abiertos a la industria y los profesionales en general, seminarios de capacitación para docentes de instituciones miembros de la Red, talleres, y experiencias de campo.
- Facilitar el intercambio de docentes e investigadores dentro de RUSE como docentes invitados, de acuerdo con las modalidades y reglamentaciones específicas de cada integrante de la red.
- Promover la participación de docentes miembros de la red en el dictado de materias de Posgrado en las Disciplinas relacionadas con los Sistemas Embebidos y el Diseño Digital Avanzado en la codirección de las Tesis respectivas en las instituciones participantes.
- Instrumentar mecanismos de cooperación entre las Unidades Académicas (UAs) para la disposición y uso compartido de equipamiento y software específico y otros recursos de acceso no masivo.
- Instrumentar mecanismos de cooperación entre las Unidades Académicas miembros con organismos públicos y privados para la realización conjunta de proyectos de investigación y transferencia tecnológica.
- Instrumentar mecanismos de cooperación entre las Unidades Académicas miembros de la Red a efectos de elaborar y unificar criterios en relación a planes de estudios, y la adecuada actualización de sus contenidos, acorde con los constantes cambios de las disciplinas involucradas, como así también en planes de mejora que cada Unidad pudiera plantearse. Se

propondrán acciones concretas para lograr el reconocimiento recíproco por equivalencia de actividades curriculares entre diferentes planes de estudio de las UAs integrantes de la red, considerando las restricciones particulares en este aspecto de cada UA.

- Instrumentar mecanismos para que los alumnos de grado de las Unidades Académicas miembros puedan realizar total o parcialmente las Prácticas Profesionales Supervisadas, Proyectos Finales de carrera o de Tesis de grado, o Pasantías en proyectos de investigación y/o desarrollo, en las distintas Unidades Académicas, relacionados con los sectores productivos y/o de servicios que involucren a los sistemas embebidos y al diseño digital avanzado.
- Fomentar el intercambio de alumnos para cursar asignaturas electivas en las distintas Unidades Académicas.
- Difundir, articular, promover, fomentar, e instrumentar mecanismos de cooperación estrecha con la Asociación Civil para la Promoción, Investigación y Desarrollo de los Sistemas Electrónicos Embebidos.
- Articular transitoriamente con los posgrados vigentes en Sistemas Embebidos y en Diseño Digital Avanzado para contribuir a la generación de nuevos posgrados sobre la misma área de vacancia en diferentes puntos geográficos y estratégicos de país.
- Promover en forma conjunta acciones tendientes a difundir, concientizar, motivar y captar interés en la comunidad, en particular la de estudiantes de nivel medio, a los efectos de lograr y sostener una matrícula adecuada a la necesidad de formación de recursos humanos en esta área de vacancia.
- Realizar toda otra actividad académica, tecnológica y científica relacionada con los sistemas embebidos y al diseño digital avanzado, que sea de mutuo interés.

2.3. Hardware de Sistemas Embebidos

El componente fundamental en este tipo de sistemas es su unidad programable o *procesador embebido*, que porta el firmware del SE. Esta unidad programable se puede implementar con distintos dispositivos según la capacidad de procesamiento que se requiera. Las cuatro variantes más usuales, que se describen a continuación, son: Microcontroladores, Microprocesadores, Procesadores Digitales de Señales (DSP: Digital Signal Processors) y “Sistemas en chip” (SoC: Systems on chip).

2.3.1. Microcontrolador:

La alternativa más usual es utilizar microcontroladores de 4 a 32 bits que incluyen memoria principal y secundaria además de periféricos en un único chip. La memoria principal se utiliza para albergar las variables del programa mientras que la memoria secundaria (típicamente FLASH) contiene el firmware. Para microcontroladores de hasta 16 bits suele implementarse programación básica (bare metal), es decir que se programa la aplicación de manera directa sin necesidad de utilizar un SO. Cuando se utilizan microcontroladores de 32 bits, dependiendo de la aplicación, suelen utilizar un SO. El lenguaje de programación que se utiliza normalmente es “C” y se complementa con ensamblador cuando se necesita un control estricto del tiempo de ejecución para alguna tarea [Bertolotti, 2017]. En la Figura 2.1 se presenta un gráfico de los lenguajes de programación más utilizados en embebidos extraído del estudio de mercado realizado por Aspenncore para los años 2015 y 2017 [Aspenncore: 2017 Embedded Market Study, 2017].

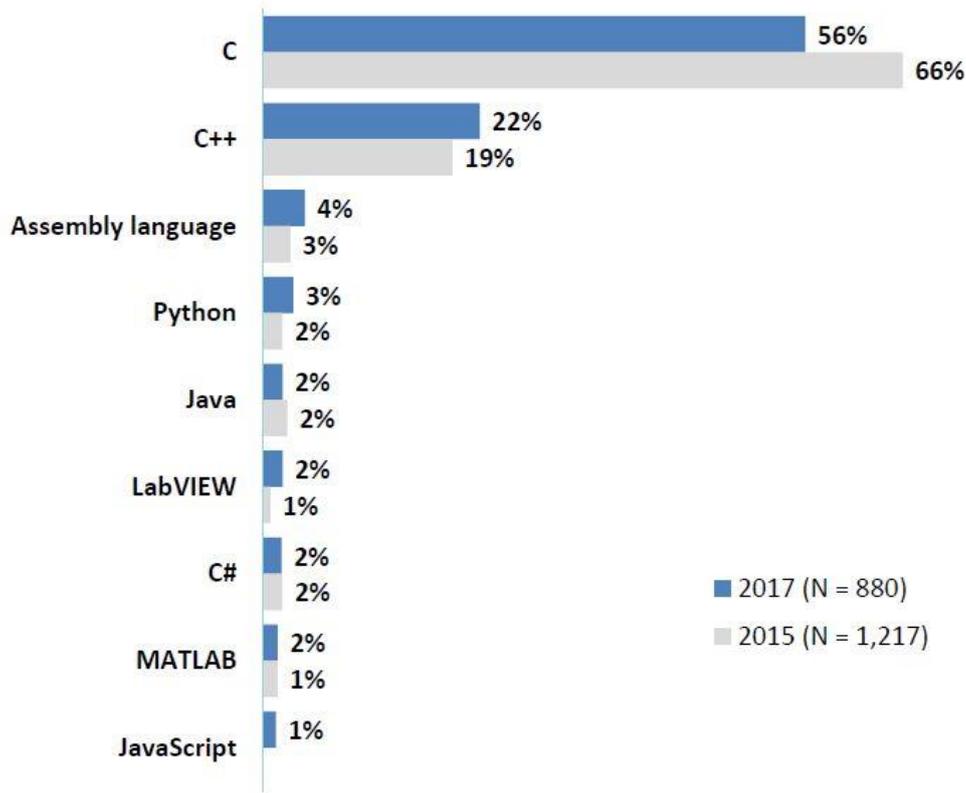


Figura 2.1. Lenguajes de programación más usados en embebidos para 2015 y 2017.

2.3.2. Microprocesador:

Los microprocesadores se han desarrollado fundamentalmente pensando en aplicaciones que requieren una gran capacidad de cálculo, manejo de gran cantidad de memoria y gran velocidad de procesamiento. A diferencia del microcontrolador que internamente incluye los buses, el banco de memoria, el reloj y temporizadores, el microprocesador necesita soporte externo para poder funcionar. Así, es necesario que en una placa madre se ruteen los buses de datos, dirección y control para poder acceder a los bancos de memoria tanto RAM como ROM y los periféricos, que también deben incluirse en la placa madre. Teniendo en claro esta diferencia, es común hacer referencia a la unidad de procesamiento del SE como el procesador embebido indistintamente ya sea que se haya utilizado un microcontrolador o un microprocesador. Típicamente, el uso de una u otra variante vendrá de la mano de la velocidad de procesamiento necesaria y la necesidad de uso de un sistema operativo. En la Figura 2.2 se presenta una gráfica de los procesadores embebidos utilizados en los últimos 6 años en base a un estudio que incorpora más de 1000 empresas y universidades de todo el mundo [Aspencore: 2017 Embedded Market Study, 2017].

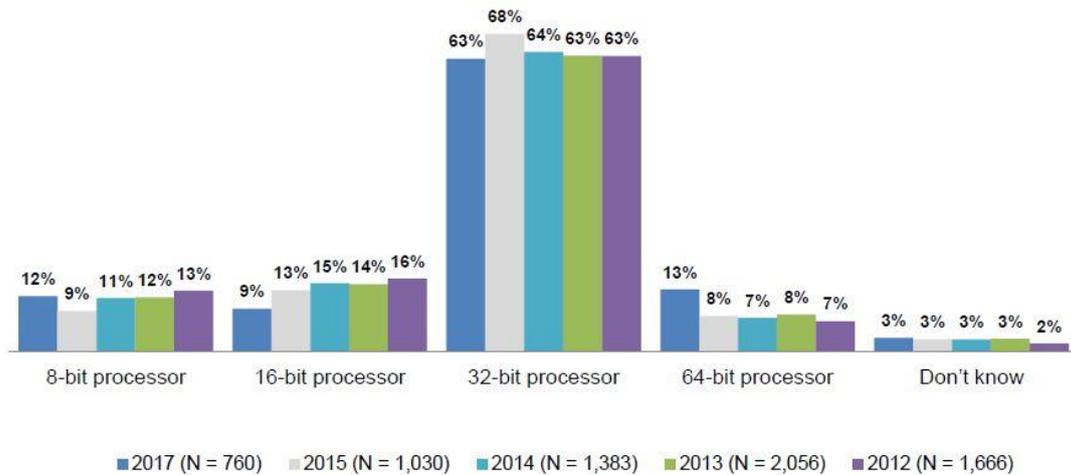


Figura 2.2. Procesadores utilizados en SEs, años 2012-2017.

En este mismo estudio también se presenta un detalle de los procesadores de 32 bits más usados en 2017 (Figura 2.3).

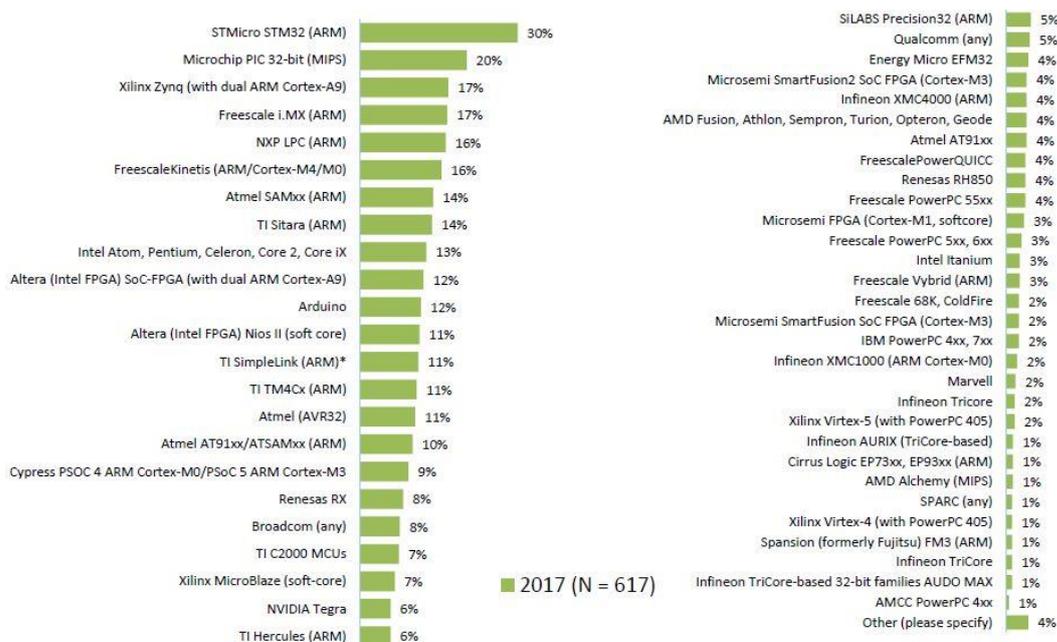


Figura 2.3. Distribución de procesadores de 32 bits utilizados para SE en 2017.

2.3.3. DSP:

Un procesador digital de señales o DSP (por sus siglas en inglés de Digital Signal Processor) es un sistema basado en un procesador optimizado para aplicaciones numéricas a muy alta velocidad. Se utilizan normalmente en aplicaciones que requieran procesamiento en tiempo real como por ejemplo procesamiento de audio y video. Se diseñan optimizando las tareas más habituales del procesamiento digital como sumas, multiplicaciones y almacenamientos en memoria. Comúnmente presentan arquitecturas de memoria que permiten un acceso múltiple para permitir de forma simultánea cargar varios operandos, por ejemplo, una muestra de la señal de entrada y

el coeficiente de un filtro simultáneamente en paralelo con la carga de la instrucción. También incluyen una variedad de modos especiales de direccionamiento y características de control de flujo de programa diseñadas para acelerar la ejecución de operaciones repetitivas. Además, la mayoría de los DSP incluyen en el propio chip periféricos especiales e interfaces de entrada-salida que permiten que el procesador se comunique eficientemente con el resto de los componentes del sistema, tales como convertidores analógico-digitales y distintos tipos de interfaces de tipo serie. La diferencia esencial entre un DSP y un microprocesador es que el DSP tiene características diseñadas para soportar tareas de altas prestaciones, repetitivas y numéricamente intensas [J. Salazar, 2001].

Los DSPs pueden programarse en lenguajes de alto nivel como C o bien en ensamblador. También pueden utilizarse lenguajes de más alto nivel, más sencillos y rápidos de usar como Ladder, LabVIEW o Matlab.

En la Figura 2.4 (extraída de: [Aspencore: 2017 Embedded Market Study, 2017]) se presentan los DSP más usados en los últimos años. Como puede observarse entre las empresas Microchip y Texas Instruments acaparan el 50% del mercado.

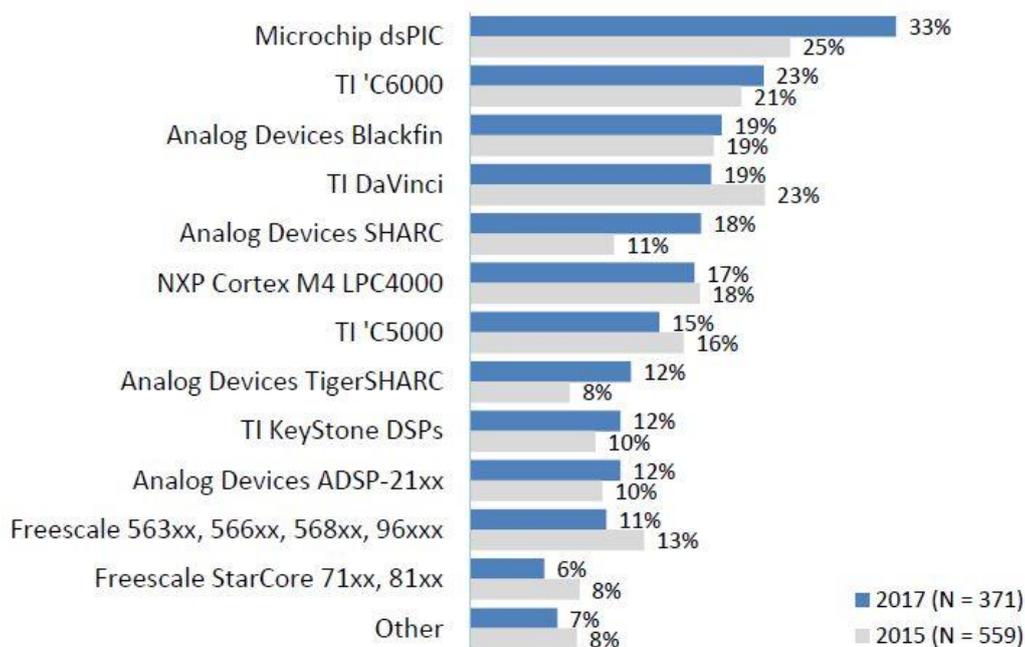


Figura 2.4. DSPs utilizados en SE para años 2015 y 2017.

2.3.4. SoC

Los sistemas en chip o SoC (de las siglas en inglés System on Chip) son sistemas de cómputo heterogéneos que integran en un solo chip un procesador, una FPGA (Field Programmable Gate Array) y periféricos de Entrada/Salida.

Como se menciona en [C Pazos, 2013], a medida que las aplicaciones embebidas crecen en complejidad, las arquitecturas de hardware y las herramientas del diseño de sistemas embebidos deben evolucionar para satisfacer esta demanda y también minimizar el tiempo de diseño. En general, los SEs cuentan con un solo procesador embebido, pero para satisfacer las aplicaciones

complejas y demandantes de tiempo real, los diseñadores han tenido que aumentar la frecuencia de operación y/o utilizar arquitecturas multiprocesador. Hoy día, para aplicaciones de alta demanda de procesamiento, los diseños están migrando a sistemas heterogéneos para proporcionar un mejor balance entre rendimiento, latencia, flexibilidad y costo. En estos sistemas heterogéneos, las FPGAs son útiles para manejar operaciones en paralelo de un gran número de canales de datos; y dado que implementan cómputo directamente en hardware, proporcionan una ruta de baja latencia para tareas como disparo personalizado y control en lazo cerrado de alta velocidad. Por otra parte, mejora la flexibilidad de los sistemas embebidos, haciéndolos más fácil de actualizar que los sistemas con lógica fija, permitiéndoles adaptarse a los requerimientos cambiantes de E/S.

Combinar un microprocesador y un FPGA en una arquitectura de computación heterogénea hace posible que en los diseños de sistemas embebidos se utilicen las fortalezas de cada elemento de procesamiento y se cumpla de manera óptima con los requerimientos de aplicación. En este esquema, se pueden resolver las tareas de baja latencia o tiempo real usando la FPGA y el procesador embebido puede ocuparse de portar un SO que resuelva la interfaz de usuario y el resto de las tareas que no requieren baja latencia.

En la Figura 2.5 (extraída de: [Aspencore: 2017 Embedded Market Study, 2017]) se presentan los kits de FPGA más usados en los últimos años.

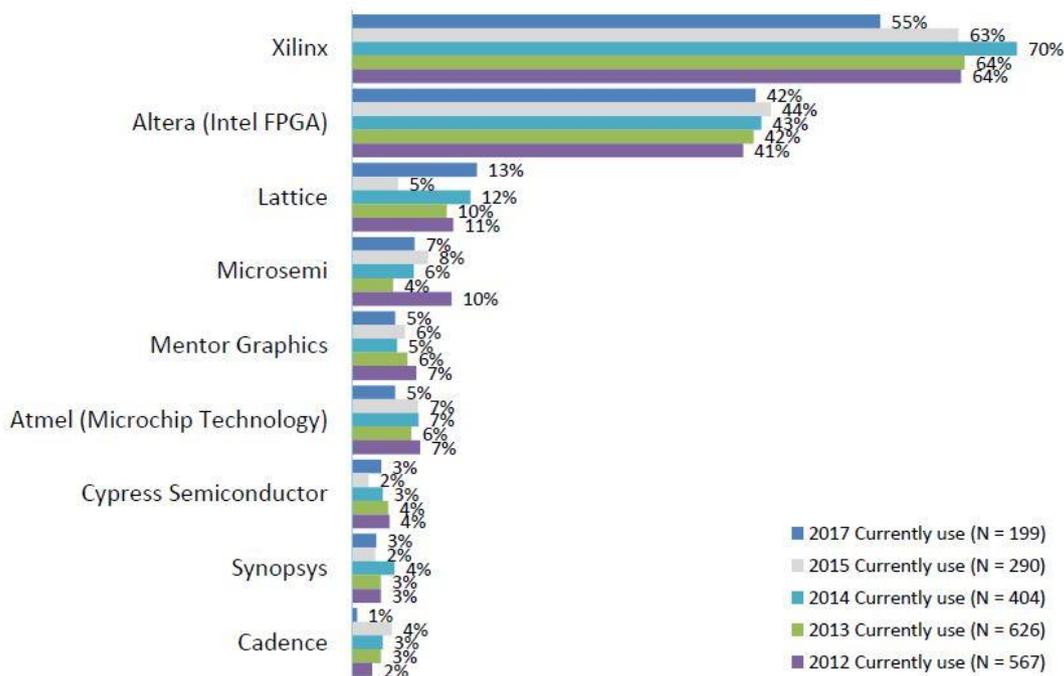


Figura 2.5. Selección de FPGA entre los años 2012-2017.

2.4. Plataformas para SE

Existen diversas plataformas que permiten implementar SE compactos. Estas integran dispositivos de entrada salida digitales y analógicos con capacidad de almacenamiento y

procesamiento. A continuación, se presentan algunas de las plataformas de hardware disponibles más utilizados en el mercado embebido, desde simples plataformas de 8 bits hasta sistemas heterogéneos de SoC-FPGA con alto poder de cómputo y capacidad para portar SO.

2.4.1. Microcontrolador ADuC.

La familia de microcontroladores de gama baja de Analog Devices ha sido pensada para implementar front-ends de transductores integrando conversores ADC y DAC de distintas resoluciones (12-24 bits). Un ejemplo es el ADuC841 que cuenta con 8 canales A/D de 12 bits, 2 canales D/A de 12 bits, un procesador de 8 bits @ 20MHz de la familia 8051 y 64 KBytes de Memoria Flash, todo integrado en un único chip. Esta plataforma es suficiente para implementar SEs de prestaciones moderadas. La diferencia de la familia ADUC, respecto de otros microcontroladores disponibles, es que provee convertidores ADC integrados de alta calidad.

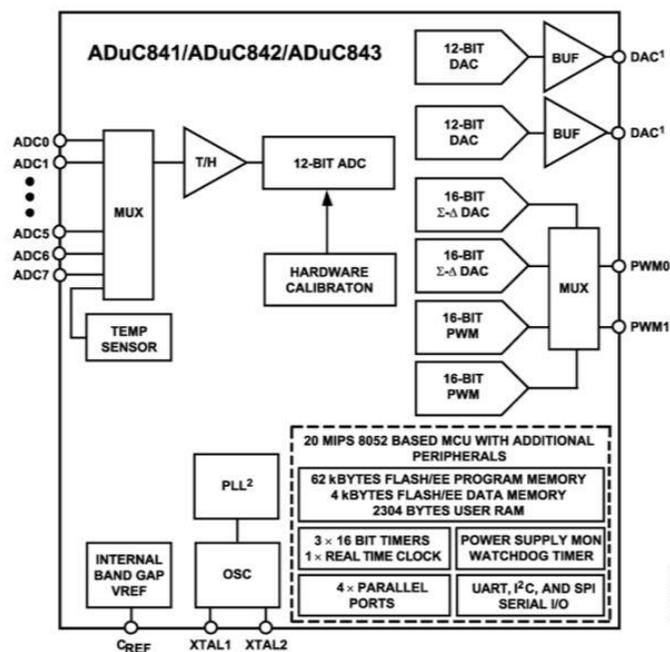


Figura 2.6. Estructura y periféricos integrados del Microcontrolador ADuC84X.

Esta plataforma puede programarse en lenguaje C desde un ambiente de desarrollo integrado, como puede ser el compilador Keil™, y se programa por medio de una aplicación propietaria y gratuita de Analog Devices denominada WSD (por sus siglas en inglés: Windows Serial Downloader). Es muy sencillo implementar y poner en funcionamiento un SE basado en este microcontrolador. Por sus bajas prestaciones, generalmente se programa “bare metal” o bien utilizando algún SO propietario simple basado en interrupciones.

2.4.2. Arduino.

Arduino surgió como una herramienta sencilla de prototipado rápido para estudiantes con pocos conocimientos en electrónica y programación. Ha sido utilizada en objetos comunes de uso diario hasta complejos instrumentos científicos. Cuenta con una amplia comunidad mundial de estudiantes, hobbistas, artistas, programadores y profesionales que han aportado amplia

información alrededor de esta plataforma open-source. Las placas son económicas, cuentan con un IDE multiplataforma que es sencillo de usar y se puede programar en lenguaje C o C++. Si bien no dispone de ADCs de altas prestaciones ni una gran capacidad de cómputo, por su costo y disponibilidad es frecuente encontrar implementaciones de SE basadas en esta plataforma [Tedeschi et al., 2017][Georgitzikis et al., 2012]

Un ejemplo de esta plataforma es la placa Arduino Mega 2560 que se muestra en la Figura 2.7 y sus principales características son: 16 entradas analógicas de 10 bits, 54 pines de entrada/salida digitales, 1 salida PWM, ejecuta 16 MIPS, 256 kBytes de Memoria Flash y 8 kBytes de Memoria RAM.



Figura 2.7. Arduino Mega2560.

El proyecto Arduino dispone de una amplia gama de placas, algunas con menores prestaciones y otras con mayores prestaciones que la que se ha descrito. Cabe destacar que gracias a la amplia comunidad de usuarios colaborando, existen una gran cantidad de “shields” que se pueden anexas a la placa principal y permiten conectar dispositivos como servomotores, pantallas u otros periféricos.

2.4.3. Placa LPCXpresso

Estas plataformas de hardware desarrolladas por NXP y Embedded Artists, son de bajo costo y de rápida puesta en marcha. Estas placas han sido desarrolladas para un prototipado rápido y una puesta en marcha sencilla. La gama de placas disponibles es muy amplia. Una plataforma disponible localmente es la LPC1769, que utiliza un procesador con gran capacidad de cómputo y almacenamiento. Se programa mediante un entorno de desarrollo libre, provisto por el fabricante, basado en la plataforma Eclipse. A continuación, se detallan las principales características de esta plataforma:

- Procesador ARM Cortex M3 de 32 bits.
- Frecuencia de reloj 120 MHz.
- Memoria flash de 512 kB.
- 70 pines de I/O.

- 4 contadores de propósito general de 32 bits.
- 8 conversores analógico-digital de 12 bits.
- 4 puertos UART.
- Entorno de desarrollo libre, desarrollado por NXP basado en Eclipse.
- Placa de desarrollo incorporada con conexión USB a la PC.
- Alimentación 3,3V.

La placa LPC1769 consta de dos partes. Una de ellas, denominada “Target”, contiene el microcontrolador con sus periféricos y el acceso a los diferentes pines; y la otra es el programador denominado “LPC Link”. Este programador es del tipo JTAG/SWD y permite conectarse a la PC mediante USB y a la placa Target mediante 16 pines.

En la Figura 2.8 se observa la placa de desarrollo, con la Target y el LPC Link unidos como vienen de fábrica. Existe la posibilidad de separarlos para usar el mismo LPC Link con diferentes Targets.

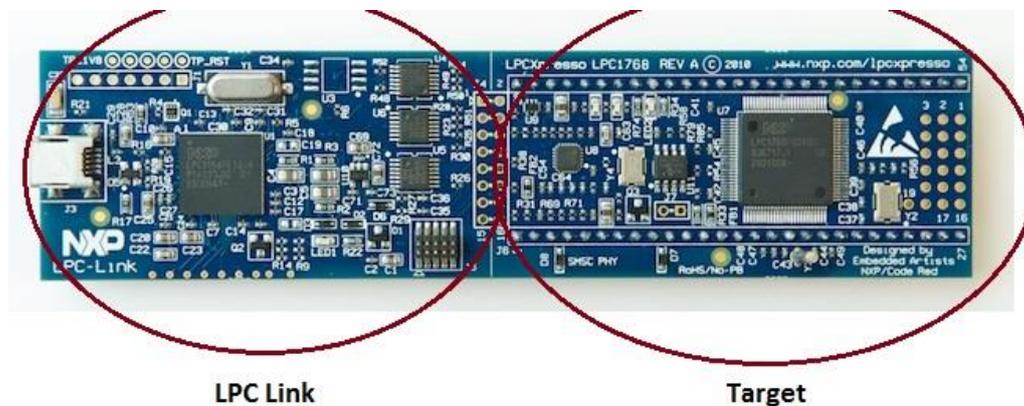


Figura 2.8. Placa de desarrollo LPC1769

El software utilizado para programar esta plataforma es el LPCXpresso, desarrollado por el fabricante. Esta aplicación, que está basada en Eclipse, puede descargarse de la web del fabricante, en la cual se debe registrar para evitar una limitación respecto al tamaño del programar generar y cargar en el microcontrolador. El mismo dispone de varias guías y ejemplos para comenzar a usar el dispositivo, como también de las librerías CMSIS que son propias de ARM las cuales definen etiquetas y asignan direcciones a los registros.

2.4.4. Micro2440SDK

Esta plataforma para SEs desarrollado por “FriendlyArm”, se basa en un microprocesador Samsung S3C2440A de 32 bits con arquitectura ARM920TDI y una velocidad de procesamiento de 400 MHz. Es una plataforma mucho más potente que las presentadas anteriormente y sus prestaciones permiten presentarla como una mini computadora. En la Figura 2.9 se muestra la placa de desarrollo con todas las periféricos que dispone: 4 puertos USB host y 1 USB dispositivo, Ethernet, pantalla LCD con touch-screen, 3 puertos serie RS-232, entrada y salida de audio, zócalo para memoria SD y un reloj de tiempo real, entre otros. En un recuadro amarillo se destaca la placa del procesador que incluye memoria RAM y Flash [http://www.friendlyarm.net, 2017].

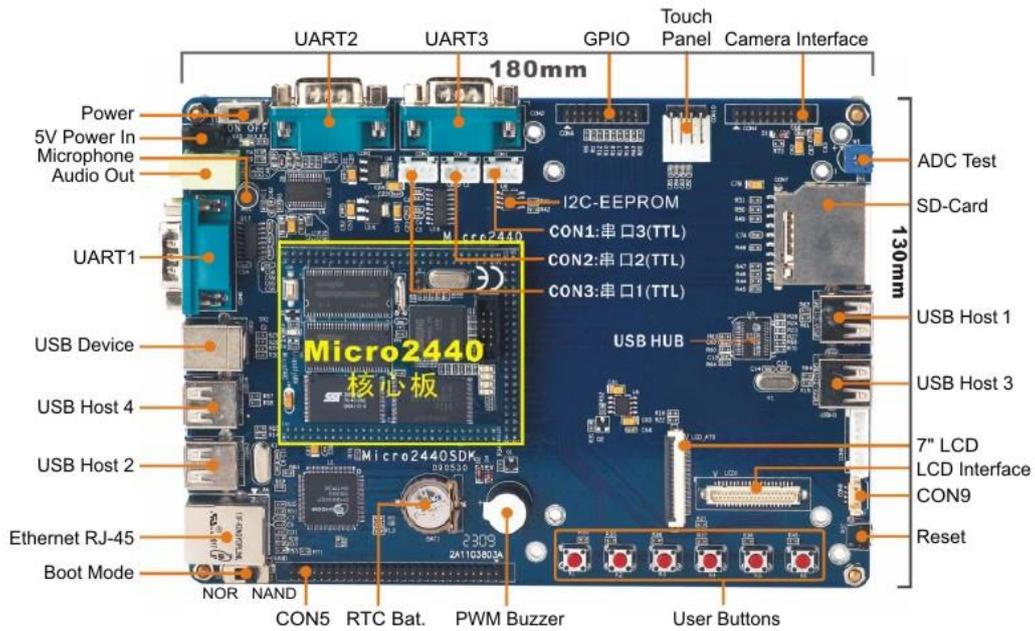


Figura 2.9. Placa de desarrollo Micro2440

En la Figura 2.10 se presenta una imagen de la placa incluyendo una pantalla LCD de 7" que se puede adquirir como un opcional. Como se observa en la imagen, el diseño terminado incluyendo pantalla y placa presenta un formato compacto similar al de las tabletas.



Figura 2.10. Placa de desarrollo Micro2440 con Pantalla LCD de 7"

Esta plataforma puede programarse en forma directa (bare-metal) pero soporta los sistemas operativos Windows CE 5.0 o 6.0 y Linux. El uso de un SO es prácticamente indispensable cuando se requiere utilizar sus capacidades gráficas (Display LCD), de conectividad (WIFI,

Bluetooth) y de almacenamiento (Memorias Flash, SD y RAM). A continuación, se detallan las principales características de la placa que contiene el procesador y de su placa de expansión (SDK):

Placa del procesador:

- Dimensión: 63 x 52 mm
- Microprocesador de 400 MHz Samsung S3C2440A ARM920T
- RAM: 64 MB, 32 bit Bus
- Flash: hasta 1GB NAND Flash y 2 MB NOR Flash con BIOS
- SPI, USB, LCD, CMOS Camera Interface
- Entrada salida analógica
- 4x LEDs de usuario

Placa de expansión (SDK-Board):

- Dimensión: 180 x 130 mm
- EEPROM: 1024 Bytes (I2C)
- Memoria de extensión: SD-Card socket
- Puerto serie: 3x DB9 (RS232)
- USB: 4x USB-A Host 1.1, 1x USB-B Device 1.1
- Salida de audio: 3.5 mm estéreo Jack
- Entrada de audio: 3.5mm jack (mono) + micrófono
- Ethernet: RJ-45 10/100M (DM9000)
- RTC: Reloj de tiempo real con batería (CR1220)
- Beeper: PWM buzzer
- Cámara con interfaz de 20 pines (2.0 mm)
- LCD: conector de 41 pines para el display FriendlyARM
- Panel táctil: 4 pin (resistivo)
- Entradas de usuario: 6x botones y 1x puerto A/D
- Expansión headers (2.0 mm)
- Alimentación: 5V.

Es de destacar el precio de esta plataforma al año 2017 ronda los 160 U\$S con pantalla LCD de 7 pulgadas incluida. La placa del procesador con 1GB de Flash cuesta unos 50 U\$S y con la placa de expansión aproximadamente 80 U\$S [<http://www.friendlyarm.net>, 2017].

2.4.5. *dsPIC*

El dsPIC (de sus siglas en inglés Digital Signal Controller) de Microchip es un controlador embebido que combina los atributos de un microcontrolador con las capacidades de un DSP en un único procesador. Presenta dos familias dsPIC30F y dsPIC33F que ejecutan la mayoría de las instrucciones en un solo ciclo. El dsPIC posee el corazón de un microcontrolador de 16 bits con los periféricos y el ágil manejo de interrupciones de un DSP que maneja las instrucciones más demandantes de cómputo. En este chip, que integra dos mundos, se mantiene un balance entre las prestaciones del micro de 16 bits y la performance esperable para el DSP. A modo de ejemplo de la performance que logran como DSP se presenta a continuación una tabla (Figura 2.11) con los tiempos requeridos para operaciones numéricamente demandantes [dsPIC Digital Signal Controllers, 2005].

DSP Performance				
Function	Cycle Count Equation	Conditions *	Number of Cycles	Execution Time @40 MIPS
Complex FFT**	—	N=64	3739	93.5 μ s
Complex FFT**	—	N=128	8485	212.1 μ s
Complex FFT**	—	N=256	19055	476.4 μ s
Single Tap FIR	—	—	1	25 ns
Block FIR	$53+N(4+M)$	N=32, M=32	1205	30.2 μ s
Block FIR Lattice	$41+N(4+7M)$	N=32, M=32	7337	183.5 μ s
Block IIR Canonic	$36+N(8+7S)$	N=32, S=4	1188	29.7 μ s
Block IIR Lattice	$46+N(16+7M)$	N=32, M=8	2350	58.7 μ s
Matrix Add	$20+3(C+R)$	C=8, R=8	212	5.3 μ s
Matrix Transpose	$16+C(6+3(R-1))$	C=8, R=8	232	5.8 μ s
Vector Dot Product	$17+3N$	N=32	113	2.9 μ s
Vector Max	$19+7(N-2)$	N=32	229	5.7 μ s
Vector Multiply	$17+4N$	N=32	145	3.6 μ s
Vector Power	$16+2N$	N=32	80	2.0 μ s
PID Loop Core	—	—	7	175 ns
*C= # columns, N= # samples, M= # taps, S= # sections, R= # rows				
**Complex FFT routine inherently prevents overflow				
1 cycle = 25 nanoseconds @ 40 MIPS				

Figura 2.11. Performance de los dsPIC como DSP realizando cálculos típicos

A modo de ejemplo se presentan las características de uno de los dsPICs perteneciente a la familia de propósito general: el dsPIC33FJ128GP205.

- Procesamiento de 40 MIPS.
- Memoria flash de 128 KB.
- Memoria RAM de 8 KB
- 53 pines de I/O.
- 9 temporizadores de 16 bits.
- Interfaz para códec de audio (I2S)
- 1 convertor analógico-digital con 18 canales de 12 bits.
- 2 puertos UART, 2 SPI y 1 I2C.
- IDE libre desarrollado por Microchip (MPLAB).
- Alimentación: 3-3.6V.

El ambiente de desarrollo para esta plataforma es por medio del IDE gratuito MPLAB que permite su programación en lenguaje C o ensamblador. Existen otras herramientas como depuradores en circuito y grabadores que se consiguen a bajo precio. Por otro lado, se dispone de una amplia cantidad de librerías con funciones matemáticas, para periféricos e implementación de filtros IIR o FIR, FFT y operaciones con matrices o vectores, entre otros.

En caso de ser necesario implementar un sistema operativo de tiempo real para manejar distintas tareas se puede optar entre cuatro posibilidades: CMX-RTX (un SO multitarea preemptivo), CMX-Tiny+ (versión limitada del RTX OS), MX-Scheduler (un SO gratuito multitarea preemptivo) y una variante de SO basada en OSEK/VDX que admite ejecución multitarea orientado al uso del protocolo CAN (Controller Area Network).

2.4.6. SoC DE10-Nano

Este kit de desarrollo de la empresa terasIC presenta una plataforma de desarrollo robusta alrededor del System-on-Chip (SoC) FPGA, que combina un procesador dual-core Cortex-A9 con lógica programable de alta performance para lograr diseños de alta flexibilidad. En este sentido, los usuarios pueden combinar el poder de reconfigurar lógica con un sistema de procesamiento de bajo consumo y altas prestaciones. El SoC integra un sistema de procesamiento basado en ARM formado por un procesador, periféricos y memoria vinculados a la FPGA por medio de un bus de interconexión de gran ancho de banda. Está equipada con memoria DDR3 de alta velocidad, conversor A/D y Ethernet, entre otros.

En la Figura 2.12 se presenta el diseño y los componentes que conforman la placa. En amarillo se resaltan los componentes pertenecientes al procesador y en verde los pertenecientes a la FPGA. Las principales características de esta plataforma son:

Hardware del procesador:

- Procesador Cortex-A9 dual core de 800MHz.
- 1 GB DDR3 SDRAM (32-bit data bus)
- 1 Gigabit Ethernet PHY conector RJ45.
- Puerto USB OTG (Conector Micro-AB)
- Zócalo para memoria MicroSD.
- Acelerómetro (interface I2C + interrupción)
- UART a USB (Conector Mini-B)
- Botones de warm reset y cold reset
- 1 botón y 1 led de usuario
- Header de expansión LTC 2x7 (Linear Technology)

Hardware en la FPGA:

- Cyclone V SE 5CSEBA6U23I7
- EPCS128: dispositivo de configuración serie.
- USB Blaster II: onboard para programar, Modo JTAG
- 2 botones
- 4 switches
- 8 LEDS verdes
- 3 fuentes de reloj de 50 MHz.
- 2 headers de expansión de 40 pines
- 1 header de expansión para Arduino Uno R3
- 1 header de expansión analógico con 10 pines.
- Conversor A/D SPI de 4 cables
- HDMI Tx compatible con DVI v1.0 y HDCP v1.4

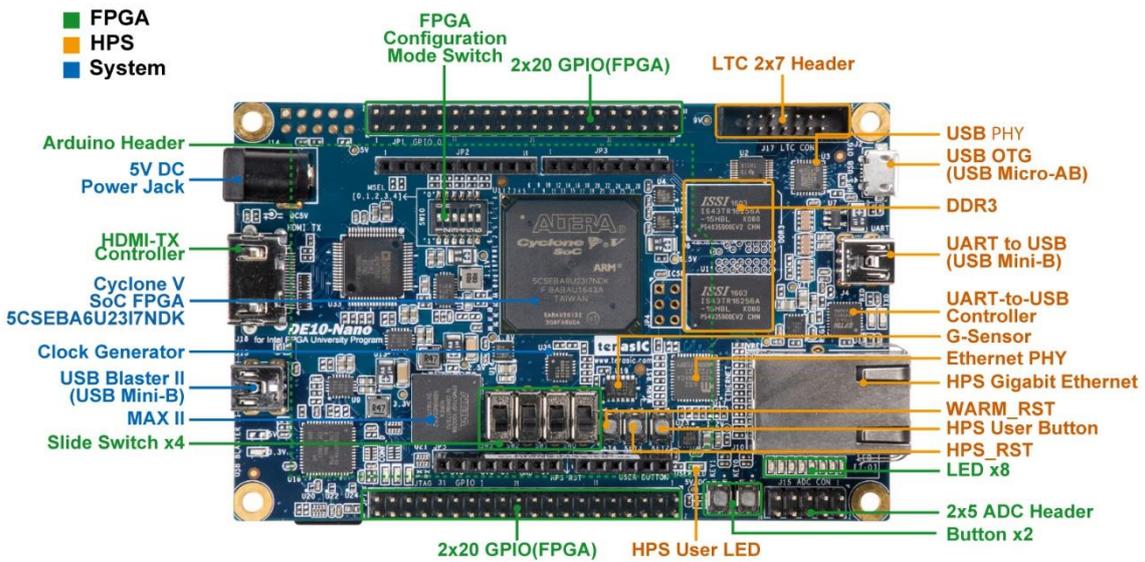


Figura 2.12. Diseño y componentes de la DE10-Nano.

Herramientas de desarrollo:

- Altera Quartus II: provee las herramientas necesarias para trabajar sobre la FPGA.
- Altera SoC Embedded Design Suite (EDS): provee herramientas de desarrollo y aplicaciones de ejemplo para facilitar el desarrollo sobre la plataforma.
- EDS incluye el ARM Development Studio (DS-5) para depurar aplicaciones Linux sobre una conexión Ethernet.

La placa se provee con una imagen del SO Linux compilada que se puede correr desde la memoria micro SD (Figura 2.13).

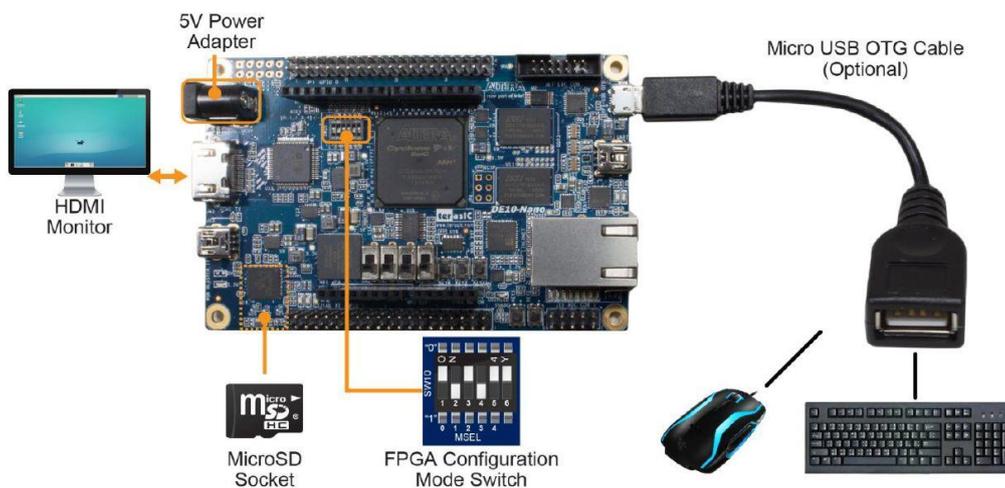


Figure 3-4 Setup for Xfce Desktop

Figura 2.13. Booteo desde Linux con escritorio de la DE10-Nano.

2.5. Firmware

El *firmware* es el conjunto de instrucciones de un programa que se encuentra almacenado en la memoria del sistema embebido. Estas instrucciones fijan la lógica primaria que ejerce el control de los circuitos que conforman el SE.

El firmware forma parte del hardware ya que se encuentra integrado a la electrónica, pero también está considerado como parte del software al estar desarrollado bajo algún lenguaje de programación. Podría decirse que el firmware funciona como el nexo entre las instrucciones que llegan al dispositivo desde el exterior y sus diversas partes electrónicas. Entre sus funciones básicas se destacan:

- Proporcionar al SE las rutinas de funcionamiento y respuesta a las peticiones de usuario.
- Controlar y gestionar el arranque del SE.

2.6. Sistema operativo embebido

Los sistemas embebidos actuales han alcanzado un grado de complejidad en el cual el desarrollo del software exige ejecución multitarea, sincronización entre tareas, sistema de archivos, soporte de red, interfaces gráficas, manejo de múltiples dispositivos periféricos, seguridad y manejo de memoria entre otros. En este escenario, pierde sentido el desarrollo de SO propietarios por el tiempo que demandaría a los programadores tal desarrollo y cobra sentido la utilización de SO que incluyan este tipo de soporte [Yu et. al., 2017][Ridolfi, 2016]. Teléfonos celulares de una década atrás ya contaban con más de 5 millones de líneas de código, y en este escenario claramente conviene pagar la licencia de algunos dólares por dispositivo para utilizar un SO comercial antes que iniciar un desarrollo que demande una gran cantidad de horas hombre [Hamblen, 2007] [Nguyen et. al., 2009].

El sistema operativo con el que cuenta el sistema embebido se denomina *sistema operativo embebido*. Son muchas las alternativas posibles cuando hay que elegir el sistema operativo (SO) a embeber. Hace algunos años, los procesadores embebidos (típicamente de 8 bits) no contaban con gran capacidad de cómputo y los SO típicos eran propietarios del tipo procesamiento secuencial o primer plano/segundo plano (foreground/background en inglés) [Dasgupta, 2008]. Con el avance de los procesadores, los SO también han evolucionado brindando soporte para aplicaciones multitarea con sincronismo entre las mismas, sistema de archivos y soporte para interfaces gráficas. Para lograrlo, estos sistemas utilizan planificadores con esquemas multitarea cooperativos o expropiativos [Nguyen et. al., 2009]. En la Figura 2.14 se presenta un detalle de los SO embebidos utilizados en el mundo para el año 2017 extraído de [Aspencore: 2017 Embedded Market Study, 2017].

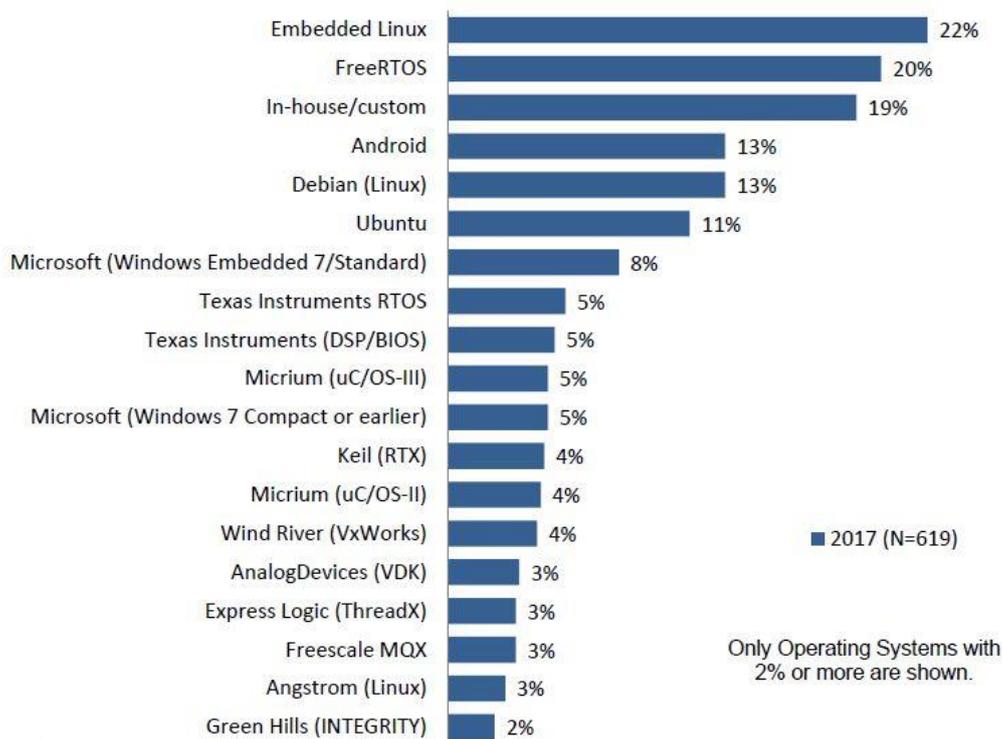


Figura 2.14. Sistemas operativos embebidos (2017).

Un caso particular, pero habitual en aplicaciones biomédicas, es asegurar que el SE responda a sus entradas generando sus salidas en un tiempo acotado, para lo cual se requiere que el SO sea de *tiempo real*. El tiempo de respuesta del sistema, comúnmente conocido como latencia, debe ser determinístico [Nguyen et. al., 2009]. Si el tiempo de respuesta está por encima de la cota se dice que el sistema de tiempo real falla. Un ejemplo de este tipo de sistema es el control del airbag en un automóvil. Cuando el acelerómetro del sistema detecta una colisión se deben disparar los airbags con un tiempo de respuesta no mayor a los 10 ms para evitar lesiones en los ocupantes del vehículo. Este tipo de sistema se conoce como “sistema de tiempo real duro” (hard real time). Otro caso menos exigente son los “sistemas de tiempo real blando” (soft real time). En este tipo de sistemas, el tiempo de respuesta normalmente está dentro de la cota especificada pero no se considera que el sistema ha fallado en caso de que alguno esté por encima. Un ejemplo de este tipo de sistema es un reproductor multimedia: puede ser que esporádicamente el sistema pierda algún paquete de video o alguna muestra de audio y que el usuario ni siquiera lo perciba dado que funciona de manera correcta casi todo el tiempo.

De acuerdo a los requerimientos, restricciones, y a la naturaleza de las tareas a ejecutar, existen distintos tipos de sistemas operativos embebidos.

2.6.1. Sistema operativo de procesamiento secuencial

La técnica de procesamiento secuencial consiste en ejecutar una tarea tras otra de manera consecutiva dentro de un bucle infinito. En este esquema, cada una de las tareas se ejecuta de inicio a fin cuando llega su turno, y es por esta forma de funcionar que no pueden bloquearse a la espera de algún evento externo. Esto es porque los eventos externos son asincrónicos con el funcionamiento del sistema, y de esta forma no se puede establecer un tiempo máximo de ejecución ni para la tarea ni para el bucle, violando la condición fundamental a verificar como SO de tiempo real. En el caso de que todas las tareas eviten bloquearse y que la suma de los tiempos de las tareas sea menor al tiempo de muestreo del sistema, el método de procesamiento secuencial es la forma más sencilla de implementar un SO de tiempo real. Esta técnica tiene como principales

ventajas la sencillez de implementación y la capacidad de compartir información de manera sencilla entre tareas dado que todas se ejecutan de principio a fin y nunca pueden superponerse. Por esto mismo, tampoco es necesario realizar cambios de contexto entre tareas, evitando pérdidas de tiempo propias del funcionamiento del SO que sí se verifican en otro tipo de implementaciones que veremos a futuro. El principal inconveniente de este sistema es la latencia que coincide con el tiempo de ejecución del bucle infinito. En caso de que este tiempo sea insuficiente hay que cambiar a un procesador más rápido o bien utilizar otra técnica más compleja.

2.6.2. Sistema operativo primer plano/segundo plano.

En este tipo de sistema se logra disminuir la latencia de los sistemas secuenciales vinculando las tareas que demandan tiempos de respuesta acotado a interrupciones y dejando el resto de las tareas en un bucle infinito. En este esquema de funcionamiento las tareas vinculadas a interrupciones se denominan de primer plano (foreground) y las tareas dentro del bucle infinito de segundo plano (background). Típicamente las tareas de primer plano están vinculadas a eventos externos que son asíncronos y de esta forma se puede dedicar tiempo a los mismos solo en el instante necesario a su arribo. La principal ventaja de este tipo de sistema es la mejora en la latencia para las tareas de primer plano y que no requiere cambios de contexto que deba implementar el SO, pues esto puede resolverse con el manejador de interrupciones por hardware con el que cuentan los procesadores. Como principal desventaja se destacan la concurrencia entre tareas debido a la naturaleza asíncrona de las interrupciones. Lo que se pretende es que las tareas vinculadas a interrupciones sean lo más rápida posible, dejando para tareas de segundo plano el procesamiento que demora más tiempo. Es en este escenario donde surgen las complicaciones dado que estas tareas deben compartir información y la naturaleza asíncrona de las interrupciones puede provocar errores en la forma de acceder a la información compartida que es lo que se conoce como incoherencia de datos. Los tramos de código en los cuales se puede acceder a esta información compartida se conocen como zonas críticas y se suele inhabilitar las interrupciones en estas zonas para evitar posible incoherencia en los datos.

Como ya se ha mencionado, las tareas asíncronas de primer plano que demoran tiempo deben separarse en dos, dejando la carga de procesamiento para el segundo plano. En este caso, la latencia para esta tarea de segundo plano será la del bucle infinito, limitando el escenario de posibles implementaciones. Típicamente se adopta este tipo de soluciones sencillas cuando las tareas de primer plano se ejecutan rápidamente sin tener que dividir las en dos tareas. Para implementaciones más complejas el escenario de los SO propietarios pierde protagonismo entrando ya en juego los SO de tiempo real con planificadores del tipo cooperativo o expropiativo.

2.6.3. Sistemas operativos cooperativos y expropiativos

El componente fundamental en los sistemas operativos de tiempo real (SOTR) es el planificador, cuya función principal es la de administrar las tareas de segundo plano. En este escenario las tareas de primer plano están vinculadas a las rutinas de servicio de interrupción. Típicamente identificadas por sus siglas en inglés ISR: Interrupt Service Routines. En estos casos, el planificador maneja información en estructuras de datos internas sobre cada una de las tareas del segundo plano que le permiten administrar la tarea a ejecutar en cada momento en función de su prioridad y/o su tiempo para finalizar la ejecución. Como ventaja respecto de los escenarios previos, ahora también las tareas de segundo plano pueden tener prioridades evitando el manejo secuencial.

Existen dos tipos de planificadores: cooperativos (non preemptive en inglés) y expropiativos (preemptive en inglés). En el caso de los planificadores cooperativos, son las tareas de segundo plano las que invocan al planificador cuando han finalizado o bien cuando consideran que han dispuesto de suficiente tiempo del uso del procesador. En ese momento el planificador dará lugar a una tarea que tenga prioridad y esté lista para ejecutarse o bien dará lugar a la tarea que seguía en la secuencia. El principal inconveniente de este tipo de planificador es la falta de control sobre

la latencia de las tareas en segundo plano ya que lo recupera solo cuando la tarea se lo devuelve. En caso de que una tarea no verifique su tiempo máximo, provocará que sea imposible garantizar la temporización al resto de las tareas. Este tipo de planificador demanda al programador de las tareas en segundo plano a situar la cesión del procesador en lugares estratégicos a fin de asegurar la latencia. Una de sus ventajas es que al ser la tarea quien cede el procesador y es el programador quien establece el punto de cesión, no existirá incoherencia de datos entre tareas de segundo plano. Claro está que, si las tareas de segundo plano comparten información con las tareas de primer plano (ahora las ISR), si podrá existir incoherencia de datos por la naturaleza asíncrona de las mismas.

En el caso de los planificadores expropiativos, estos se ejecutan de manera automática en intervalos periódicos de tiempo. Para esto se utiliza un temporizador que a un intervalo de tiempo periódico conocido como ventana de tiempo (del inglés: time slice) genera una interrupción que invoca al planificador. Llegada la interrupción, el planificador asignará una nueva ventana de tiempo a la tarea que siga en el turno. En este tipo de planificador no es necesario que el programador tenga que agregar cesiones explícitas del CPU pues lo hará el planificador cuando se verifique la ventana de tiempo. Si en este caso se pretende ajustar el tamaño de la ventana de tiempo tan chico como para asegurar que todas las tareas de segundo plano verifican su latencia, pero si el mismo es muy pequeño el planificador perderá gran cantidad de tiempo cambiando de contexto sin poder avanzar con las tareas. Tiempos típicos para la ventana de tiempo de sistemas operativos de tiempo compartido como Windows, Linux o Mac OS son del orden de 10 a 20 ms. En el caso de SOTR como QNX o FreeRTOS son comunes tiempos del orden de 4 o 5 ms [Frias, 2008].

Un inconveniente de los planificadores expropiativos es el acceso a recursos compartidos. Un escenario típico podría ser el de dos tareas en segundo plano que comparten el recurso pantalla. Las dos tareas requieren imprimir un mensaje en la pantalla, pero con un planificador expropiativo es muy probable que la impresión resulte una mezcla entre los dos mensajes a imprimir de cada tarea por el cambio de contexto que realiza el planificador en cada ventana de tiempo. Una de las formas de evitar este tipo de inconvenientes es la inhibición de las interrupciones mientras que una tarea está usando el recurso compartido, pero esta inhibición debe durar poco tiempo pues aumentaría la latencia de todas las tareas (1° y 2° plano). Otra alternativa es evitar que el planificador realice cambios de contexto a la espera de que la tarea de segundo plano que maneja el recurso compartido lo libere, pero al igual que antes se empeora la latencia del resto de las tareas en segundo plano. La tercera alternativa es el uso de semáforos sobre el recurso compartido. En este esquema todas las tareas piden acceso al recurso al planificador quien dará acceso en caso de que esté libre o bloqueará la tarea a la espera de que se libere el semáforo.

Capítulo 3: Interfaces Cerebro Computadora

3.1. Definición.

Una de las aplicaciones biomédicas de los SEs son las Interfaces Cerebro-Computadora (ICC). Entre las muchas definiciones de estos dispositivos, la más clara y referenciada fue propuesta por un grupo de referentes de distintos países en el año 2002 [Wolpaw et al., 2002]: “A direct brain-computer interface is a device that provides the brain with a new, nonmuscular communication and control channel”. Es decir, una ICC es un dispositivo que brinda al cerebro un nuevo canal de comunicación y control sin hacer uso de los músculos.

El objetivo original de las ICC es brindar un canal de comunicación a personas con movilidad reducida, que les permita realizar tareas como operar un deletreador, comandar una silla de ruedas, comandar electrodomésticos o manejar un mouse de computadora, entre otras. Hoy en día también están comenzando a ser utilizadas para entretenimiento en video juegos.

Las interfaces cerebro computadora utilizan los biopotenciales cerebrales para comandar distintos tipos de dispositivos por medio de una computadora. La forma comúnmente utilizada para registrar la actividad cerebral es por medio de la electroencefalografía (EEG) debido a que es una práctica no invasiva que se registra con electrodos dispuestos sobre el cuero cabelludo. En la Figura 3.1 se presenta una imagen de un prototipo de ICC que permite generar dos comandos. En la misma se pueden observar los electrodos, colocados sobre el cuero cabelludo, que registran la actividad cerebral del usuario.

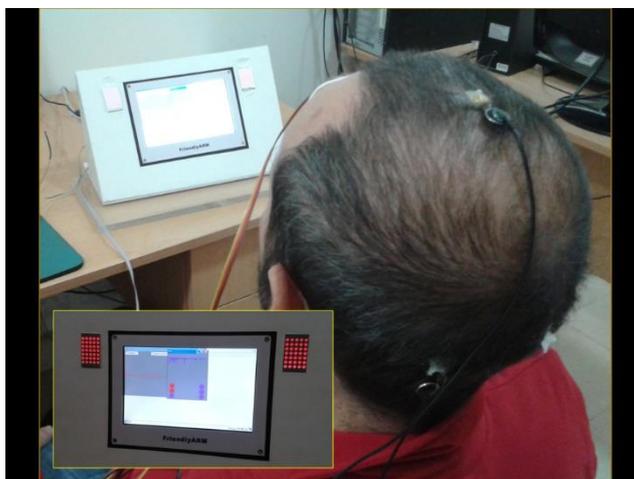


Figura 3.1. Prototipo de ICC construída en el marco de esta tesis.

Existen diversos tipos de ICC basadas en distintos potenciales. Algunos de estos son: sincronización y desincronización relacionada a eventos (ERS/ERD) [Pfurtscheller et al., 1997][García et al, 2010], potenciales corticales lentos [Zhao et al., 2009], P300 [Li et al, 2009], ritmos cerebrales (α y β) [Spinelli et al, 2000] y potenciales evocados visuales de estado estacionario [Gao et al, 2003].

3.2. Evolución y Estado actual.

El primer antecedente de ICC posiblemente sea el artículo publicado por Dewan en la revista Nature [Dewan, 1967], quien propuso utilizar señales de EEG para generar señales en morse.

Más tarde, en 1973, Vidal analizó la posibilidad de utilizar señales cerebrales en una comunicación hombre-máquina al mismo tiempo de desarrollar una herramienta para el estudio de los fenómenos neurofisiológicos que gobiernan la producción y el control en eventos observables [Vidal, 1973]. Desde esos años hasta cerca del año 2000 fueron menos de 10 los centros de investigación que dedicaron recursos a investigar en esta temática. A partir de 1999, cuando el Departamento de Salud del Centro Wadsworth organizó el Primer Encuentro Internacional de ICC en la ciudad de Nueva York, se generó un marcado interés en la temática. Así, hoy en día son más de 100 los centros en los que se investiga en ICC, siendo muy amplia la gama de publicaciones y habituales las competencias y encuentros internacionales sobre el tema [Huggins et al.,2014][Cincotti et al.,2005].

Las ICCs también están instaladas en nuestro ambiente científico. En Argentina se han realizado dos jornadas JAICC (Jornada Argentina sobre Interfaces Cerebro-Computadora), en 2006 y 2009, reuniendo investigadores en ICC del país y del exterior. En nuestro país existen varios grupos que actualmente trabajan en ICCs, como el Laboratorio de Ingeniería en Rehabilitación e Investigaciones Neuromusculares y Sensoriales (LIRINS) de la Facultad de Ingeniería de la Universidad Nacional de Entre Ríos, el Instituto de Investigación en Señales, Sistemas e Inteligencia Computacional Sinc(i) de la Universidad Nacional del Litoral (UNL), el Gabinete de Tecnología Médica GATEME de la Universidad Nacional de San Juan (UNSJ), el Grupo Medios e Interfaces de la Universidad Nacional de Tucumán y el Grupo de Instrumentación Biomédica, Industrial y Científica GIBIC de la Universidad Nacional de La Plata, entre otros.

3.3. Estructura clásica de una ICC.

La estructura general de una ICC y sus etapas se presenta en la Figura 3.2. A continuación se describe la función de cada una de estas etapas.

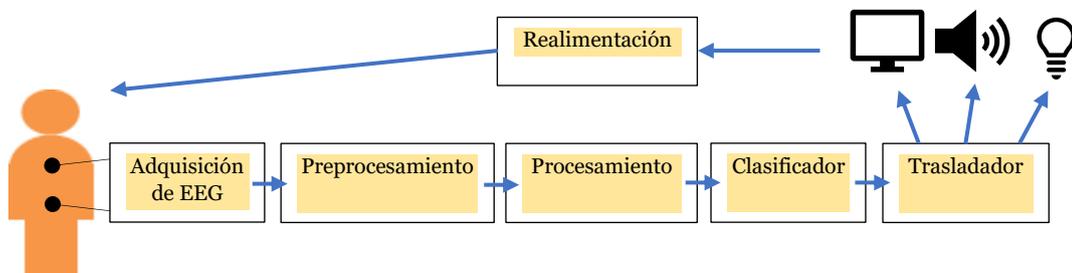


Figura 3.2. Estructura clásica de una ICC.

- Adquisición de EEG: en esta etapa se realiza la amplificación, filtrado y digitalización de la señal de EEG.
- Preprocesamiento: en esta etapa se acondiciona la señal digitalizada.
- Procesamiento: en esta etapa se estiman parámetros o características de la señal, como energía por bandas o coeficientes de un modelo autorregresivo [Rodríguez-Bermúdez et. al, 2012], que son remitidas al clasificador.
- Clasificador: realiza la decodificación de la intención del usuario de la ICC a partir de características de la señal de EEG obtenidas en la etapa previa.

- Traductor: luego de decodificada la intención del usuario por medio del clasificador, el traductor envía comandos al dispositivo objeto de la ICC, como puede ser un mouse de computadora o encender/apagar una luz.
- Realimentación: esta etapa brinda información al usuario del funcionamiento de la ICC que le permite ajustar o mejorar la estrategia que utiliza para generar los comandos.

3.4. Clasificación de las ICC según la característica del EEG

Las ICC utilizan distintos potenciales de EEG que se consignan en la Fig. 3.3. En principio pueden ser espontáneos o evocados. Los potenciales evocados se producen sobre el sujeto en respuesta a un estímulo externo, a diferencia de los potenciales espontáneos que pueden ser generados por el sujeto a voluntad, sin necesidad de estimulación externa.

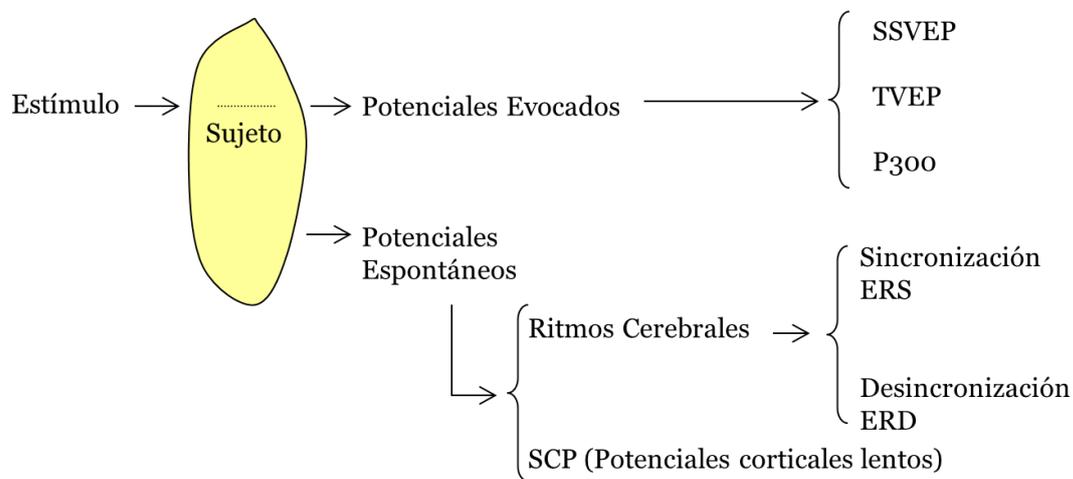


Figura 3.3. Clasificación de ICC en función de la característica EEG utilizada.

Entre los potenciales evocados se destacan:

- Potenciales evocados visuales de estado estacionario (SSVEP: steady state visual evoked potential).
- Potenciales evocados visuales transitorios (TVEP: transient visual evoked potential).
- Potencial evocado P300.

Los potenciales espontáneos más utilizados en ICC son los potenciales corticales lentos y los ritmos cerebrales. En el caso de los SCP, el EEG se registra en el vertex y los usuarios aprenden a controlar este potencial utilizando una realimentación visual del mismo por medio de entrenamiento. Típicamente se ha utilizado para mover un cursor o bien seleccionar uno de dos objetivos en una pantalla de PC [Birbaumer et al., 2000]. Dentro de los ritmos cerebrales suele utilizarse la sincronización o desincronización relacionada a eventos (ERD y ERS: del inglés event related synchronization and desynchronization) [Pfurtscheller et al., 1997][García et al., 2010]

En el ámbito del presente trabajo se ha experimentado con los ERD/ERS y los SSVEP. Estos últimos son de gran interés, dado que pueden ser implementadas con tan solo un par de electrodos, no requieren entrenamiento para su utilización, el procesamiento a realizar es sencillo y se han logrado altas tasas de transferencia de información (70 bits/min) [Gao et al., 2003], [Cheng et al.,

2002]. Además, los SSVEP permiten implementar interfaces del tipo asincrónicas donde el usuario genera el comando cuando lo requiere y no cuando la ICC le asigna una ventana de tiempo.

3.5. Potenciales Evocados Visuales.

Los potenciales evocados visuales reflejan la actividad cerebral producida por el procesamiento de información visual. Estos potenciales se observan especialmente en la zona occipital y se producen en respuesta a ciertos estímulos visuales. Existen dos tipos de potenciales evocados visuales: los transitorios (TVEP) y los de estado estacionario (SSVEP). Los TVEP se producen cuando la frecuencia de cambio del estímulo visual es menor de 2 Hz, mientras que las SSVEP se producen para estímulos con frecuencias mayores a 6 Hz [Gao et al., 2003], [Zhu et al., 2010]. Cuando la frecuencia del estímulo supera los 6 Hz, los potenciales evocados en respuesta al estímulo comienzan a superponerse formando una respuesta de estado estacionario (los SSVEPs) formada por componentes en la frecuencia de excitación y sus armónicos,

En la Figura 3.4 se presenta un esquema por medio del cual se pueden registrar los SSVEP. En la parte superior de la Figura 3.4 puede observarse el usuario en dos estados posibles, observando y sin observar un led de color rojo destellando a una frecuencia de 13 Hz. En la parte inferior de la Figura se presenta la densidad espectral de potencia normalizada aplicada al registro diferencial de EEG capturado sobre las posiciones O1 - O2 del sistema internacional 10-20. En gris se observa el espectro de la señal de EEG cuando el usuario evita mirar el led y en rojo cuando lo mira con atención. Como se observa, existe una marcada diferencia en la amplitud del espectro entre ambas condiciones. En general los potenciales evocados son de muy poca amplitud (μV) y baja relación señal a ruido, por lo cual requieren técnicas de promediación para ser detectados.

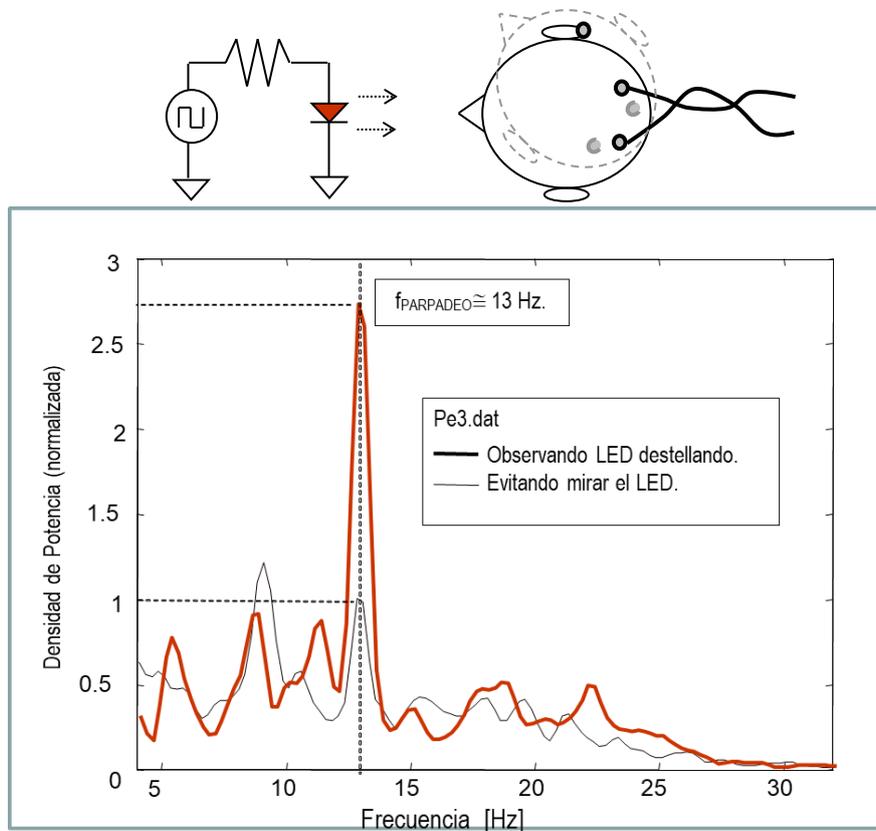


Figura 3.4. Potenciales evocados visuales.

3.6. Interfaz Cerebro Computador basada en SSVEP.

En la Figura 3.5 se presenta un modelo funcional de ICC basada en SSVEP. Como se observa en la Figura 3.5, está compuesto por un estimulador visual, una etapa de adquisición de biopotenciales, una etapa de procesamiento y por último el generador de comandos que manejará el dispositivo de asistencia al usuario.

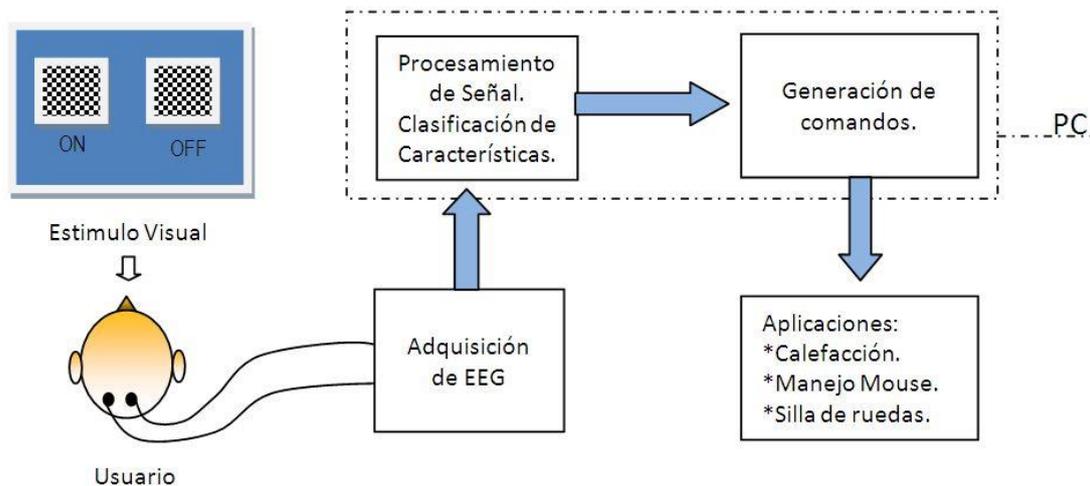


Figura 3.5. Modelo funcional de ICC basada en SSVEP.

La etapa de estimulación visual normalmente se implementa utilizando diodos emisores de luz (LED), monitores de tubo de rayos catódicos (CRT) o displays de cristal líquido (LCD) [Zhu et al, 2010], [Wang et Al, 2008]. En las implementaciones de ICC para investigación, en la etapa de adquisición de EEG se suelen utilizar equipos comerciales y la etapa de procesamiento de señales y generación de comandos se ejecutan sobre un computador personal que comanda el dispositivo objeto de la aplicación.

La forma de operar de este tipo de ICC es la siguiente: se presentan estímulos al usuario destellando a diferentes frecuencias. Cada uno de estos estímulos está asociado a un comando sobre el dispositivo que la aplicación utilice. El usuario debe enfocar su visión en el comando que desea ejecutar, lo cual generará el potencial evocado con componentes en la frecuencia del estímulo y sus armónicos. Luego, por medio de un análisis espectral, se detecta el comando seleccionado. A este tipo de ICC se lo conoce como SSVEP basado en frecuencia ya que la selección del estímulo se realiza entre varias frecuencias de excitación. Con el objeto de aumentar el número de posibles estímulos al usuario se ha reportado un nuevo tipo de ICC conocido como SSVEP basado en detección de fase [Jia et. al, 2011][Shyu et. al, 2010]. En este tipo de dispositivo se presentan al usuario diversos estímulos destellando a la misma frecuencia, pero con distintas fases. Luego, por medio de la representación en el plano complejo de la parte real e imaginaria de la transformada discreta de Fourier (TDF) se pueden detectar los SSVEP producidos por los distintos estímulos.

3.7. Plataformas de Hardware-Software para ICC.

Existen diversas plataformas para implementar ICCs. En algunas de ellas es abierto tanto el hardware como el software y pueden adaptarse a distintos paradigmas de Interfaces Cerebro-Computadora. A continuación, se describen algunas de las plataformas disponibles.

3.7.1. NCAN (National Center for Adaptive Neurotechnologies).

Este centro mantiene activa la plataforma de software BCI2000 que resulta útil para adquisición en tiempo real de bio-señales, análisis y realimentación y que se ha convertido en una tecnología estándar para ICC e investigación en neuro-tecnología. El centro la distribuye sin costo alguno para investigación o clínica médica a través del mundo.

En la Figura 3.6. se presenta una imagen del “Wadsworth BCI Home System” extraída a partir de una video demostración del propio centro [<http://www.neurotechcenter.org/videos/brain-computer-interface-system-setup-training-video>]. En la actualidad este sistema está siendo utilizado por un pequeño número de personas con un estadio avanzado de esclerosis lateral amiotrófica para comunicación básica y otros propósitos domiciliarios. El sistema ICC utiliza los potenciales evocados P300 permitiendo al usuario deletrear notas en un editor de texto, enviar y recibir correos electrónicos y navegar en Internet.

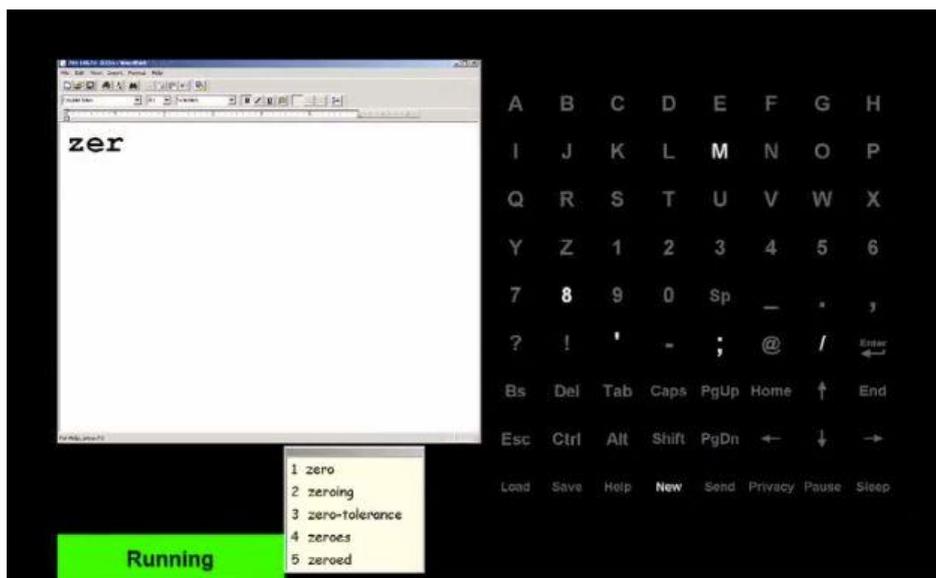


Figura 3.6. Sistema ICC domiciliario del Wadsworth Center.

3.7.2. Proyecto OpenBCI

Es una comunidad formada por investigadores, ingenieros, artistas y diseñadores entre otros, dedicada al desarrollo de ICC libres. Disponen de sistemas de registro de biopotenciales de bajo costo que permiten muestrear actividad cerebral (EEG), actividad muscular (EMG) y actividad cardíaca (ECG), entre otros. Posee un foro por medio del cual personas con distintos perfiles y distintas nacionalidades pueden interactuar y compartir experiencia con el único fin de innovar en ICC. Dentro de la política de código abierto (del inglés open source) presentan algunas placas de hardware y software para su registro y visualización. Entre el hardware que ofrecen se destaca la placa “CYTON” cuyas características pueden observarse en la Figura 3.7.

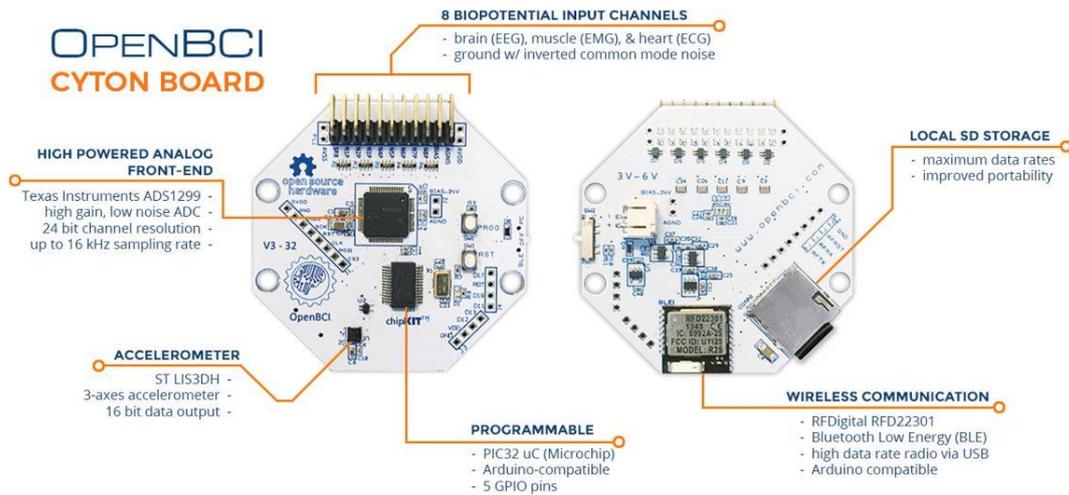


Figura 3.7. Placa CYTON de OpenBCI [Imagen extraída de: openbci.com]

En la Figura 3.8 se presenta una de sus pantallas, que se brinda para visualizar en tiempo y frecuencia las señales registradas acompañado de un mapa con la disposición de los electrodos. Es de destacar la gran utilidad de este sistema dado que brinda a sus usuarios los datos crudos (del inglés Raw data) otorgando más libertad y capacidad de análisis de datos en etapas de investigación.

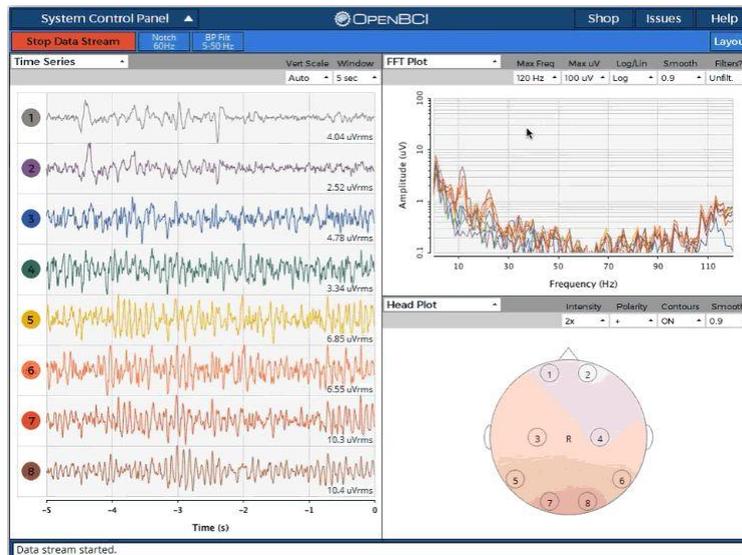


Figura 3.8. Software libre de OpenBCI [Imagen extraída de: openbci.com]

3.7.3. G.tec

Es una empresa en constante crecimiento con sede central en Austria, España y Estados Unidos y distribuidores en todo el mundo. Se dedican a desarrollar hardware y software para ICC, siendo miembros activos en distintos proyectos de investigación internacionales y realizando publicaciones. Fueron quienes desarrollaron el primer sistema ICC comercial en 1999 y hoy en día venden sus sistemas en más de 60 países.

Entre los muchos sistemas y productos que comercializan se destaca el amplificador de biopotenciales denominado “USBamp 3.0” por su amplia difusión en el ambiente científico y su robustez. Es un sistema de adquisición y procesamiento de biopotenciales de alta performance y precisión. Permite la investigación en cerebro, corazón, actividad muscular, movimiento de ojos, respuesta galvánica de la piel y otros parámetros físicos y fisiológicos. Cuenta con conexión USB (del inglés Universal serial bus) a PC y dispone de 16 canales simultáneos de muestreo con 24 bits de resolución. En caso de necesitar más canales se pueden sincronizar más dispositivos. El amplificador cuenta con un rango de entrada de ± 250 mV que permite acondicionar señales acopladas en continua que incluyan el potencial de electrodo sin saturar.

Dispone de soluciones de software de distinto nivel, desde sistemas de registro cómodos basadas en ventanas hasta drivers para procesamiento online sobre MATLAB/SIMULINK y LabVIEW. También presenta soporte para comunidades open source como OpenVibe y BCI2000.

En la Figura 3.9 se presenta una imagen del amplificador junto con sus especificaciones y en la Figura 3.10 una ICC comercial, ambas extraídas de la página de la empresa [<http://www.gtec.at>].

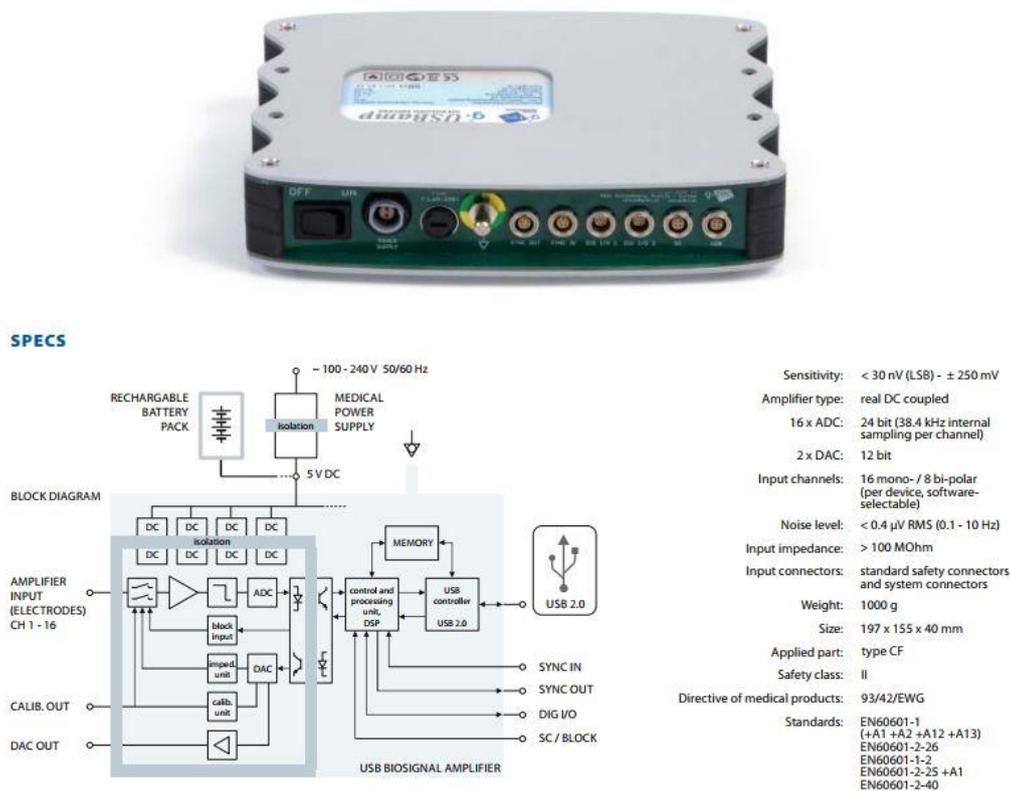


Figura 3.9. USBamp de la empresa g.tec [Imagen extraída de www.gtec.at]



Figura 3.10. ICC comercial de la empresa g.tec

3.7.4. OpenVibe

Es una plataforma de software dedicada al diseño, prueba y utilización de interfaces cerebro-computador. Openvibe permite el procesamiento, clasificación y visualización de las señales cerebrales en tiempo real. Es libre, de código abierto y puede ejecutarse sobre los sistemas operativos Windows y Linux. Ha sido desarrollado en Francia, en el Instituto INRIA (traducido del francés es el Instituto Nacional de Investigación en Informática y Automática). Uno de los puntos fuertes de este software es el soporte para diversas plataformas de hardware (como las dos previamente descritas: OpenBCI Cyton y USBamp, entre otras). El código está escrito en lenguaje C++ y por medio de la integración de distintos bloques el usuario puede implementar distintos escenarios de ICC rápidamente. En la Figura 3.11 se puede observar el ambiente de trabajo para generar aplicaciones de ICC. En la Fig. 3.11 se muestra una pantalla típica extraída de la página del proyecto: <http://openvibe.inria.fr>].

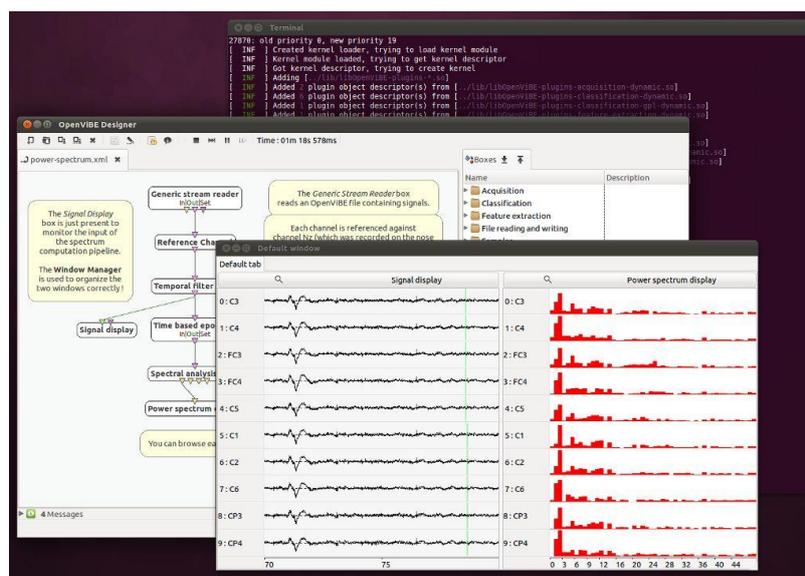


Figura 3.11. Ambiente de desarrollo del OpenVibe [Imagen extraída de <http://openvibe.inria.fr>]

3.7.5. *Emotiv*

Es una empresa especializada en bioinformática con el objeto de potenciar a individuos en el entendimiento de su cerebro usando EEG. Fue fundada en 2011 con un perfil comercial que trataba de acercar la tecnología de EEG a bajo costo a personas interesadas en ICC. Hoy en día se reconoce a esta empresa como pioneros en este mercado y su comunidad de desarrolladores e investigadores se ha difundido a más de 100 países. Posee variantes de hardware/software dependiendo del perfil de usuario y sus intereses. Disponen de variantes para: Educación e investigación sobre el cerebro, Bienestar y performance y Tecnología controlada con el cerebro.

El formato que más se ha difundido es el de auricular (del inglés headset), incluyendo un conjunto de electrodos húmedos en posiciones prefijadas y la electrónica de acondicionamiento con conexión inalámbrica a PC para el envío de datos. Presentan distintos productos según el perfil de usuario. Desde los más sencillos usados por hobbistas hasta los más complejos de uso en investigación. Entre estos últimos se destaca el modelo EPOC+ (Figura 3.12), que se presenta como un dispositivo EEG inalámbrico de 14 canales diseñado para investigación en ICC. El sistema brinda acceso a los datos crudos de EEG por medio de un software (Pure-EEG) basado en suscripción.

A continuación, se detallan sus características:

- 14 canales: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4.
- Muestreo secuencial con un solo ADC.
- Tasa de muestreo de 128 o 256 mps.
- Resolución: 14 bits, 1 LSB=0.51 μ V (ADC de 16 bits con 2 bits descartados por piso de ruido de instrumental).
- Ancho de banda: 0.2 – 43 Hz con filtro digital notch en 50 y 60 Hz.
- Rango dinámico de 8400 μ Vpp
- Acoplamiento en alterna.
- Conectividad Bluetooth o propietaria en la banda de 2.4 GHz.
- Batería de litio de 480mAh.
- Autonomía de hasta 12 horas con Wireless propietario y 6 horas con Bluetooth.



Figura 3.12. Headset EPOC+ de la empresa Emotiv.

3.8. Motivación para la implementación de plataformas propias para ICC.

Como puede observarse en las distintas combinaciones de hardware - software presentadas para la implementación de ICCs, el denominador común es la utilización de una computadora para la etapa de procesamiento de señales y la interfaz de usuario. En algunas implementaciones como OpenBCI o Intendix se utilizan enlaces inalámbricos para el envío de datos a la computadora, tendiendo el diseño de la etapa de adquisición de biopotenciales a ser vestible (del inglés wearable) y de tamaño reducido, pero se mantiene la dependencia de una computadora para procesar. Como propuesta superadora, en el ámbito de esta tesis se implementa una ICC embebida basada en SSVEP (ver Capítulo 5) que prescinde del uso de una computadora, extendiendo el concepto de vestible más allá de la frontera de la adquisición, incluyendo el procesamiento.

Luego de haber realizado un relevamiento de las tecnologías ICC disponibles y con la motivación de contar con tecnología propia, se inició un proceso de implementación gradual hasta alcanzar la ICC embebida. En primera instancia, ajustando a un modelo similar al de OpenBCI e Intendix, se decidió implementar una plataforma de hardware simple con enlace inalámbrico para el envío de datos crudos a una computadora. En base a esta plataforma se ensayaron distintas implementaciones de ICC que permitieron verificar técnicas reportadas en la bibliografía. En primera instancia, se implementó una ICC basada en el ritmo alfa visual que permite comandar (encender/apagar) distintos dispositivos que se presentan al usuario en un menú rotante. Luego se implementó una ICC basada en ritmos motores que utiliza ERD/ERS para comandar el movimiento de un objeto en la pantalla de una computadora y por último se ensayaron técnicas de recuperación de potenciales evocados visuales (ver Capítulo 4).

El uso de esta plataforma de hardware simple permitió ensayar y verificar las distintas técnicas reportadas en la bibliografía utilizando procesamiento off-line en la computadora. A partir de

estos ensayos se logró especificar los requerimientos de hardware – software que la plataforma totalmente embebida para la ICC basada en SSVEP debe verificar:

- Capacidad de cómputo para la implementación de técnicas de promediación coherente y resolución de transformada rápida de Fourier en tiempo real.
- Sistema operativo embebido con prestaciones de tiempo real duro.
- Control de señales a bajo nivel que permita la sincronización entre eventos.
- Tiempo de arranque reducido.
- Interfaz de usuario amigable con salida gráfica.
- Bajo consumo y tamaño reducido.

En el Capítulo 5 se presenta el desarrollo final de la ICC embebida basada en potenciales evocados visuales de estado estacionario.

Parte 2: Desarrollo e Implementación de Sistemas Embebidos

Capítulo 4

Sistema Embebido configurable para investigación en Interfaces Cerebro Computadora

Resumen.

En este capítulo se describe un sistema embebido que consiste en una plataforma flexible para Interfaces Cerebro-Computadora, la cual admite distintas configuraciones en su etapa analógica y distintas opciones para la transmisión de datos. El objetivo de este dispositivo es disponer de un equipo que permite experimentar con distintos tipos y paradigmas de ICC a fin de determinar los requerimientos que estos imponen sobre el hardware y el software. El dispositivo fue ensayado como parte de una ICC basadas en ritmos cerebrales y en la captura de potenciales evocados por estímulos visuales.

4.1. Introducción.

A fin de disponer de un sistema flexible para experimentar con distintos tipos de interfaces, se desarrolló un SE pequeño, vestible, capaz de adquirir señales de EEG y transmitirlos a una computadora personal PC en tiempo real (Fig. 4.1). Sobre este PC se realiza el procesamiento, la visualización de las señales y las realimentaciones visual y sonora al usuario.

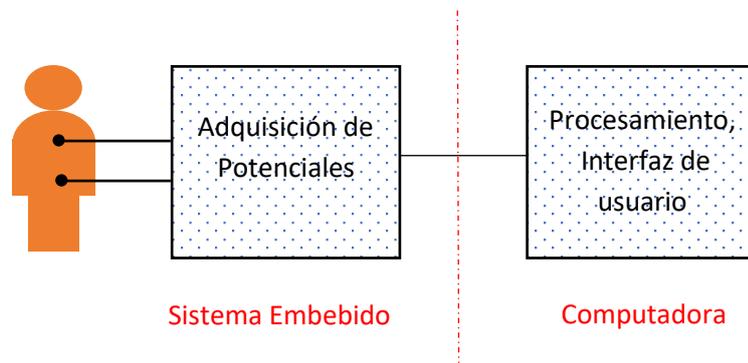


Figura 4.1. Esquema general del escenario 1.

El SE captura los biopotenciales y los transmite en bruto a la PC, dónde pueden procesarse off-line para observar distintas características de las señales o directamente implementar una ICC completa. Sus requerimientos generales son:

- Adquisición de señales de EEG en tiempo real.
- Portabilidad (transmisión inalámbrica)
- Tamaño reducido (vestible).
- Bajo consumo.
- Versatilidad para reutilizar el hardware en distintos escenarios.
- Transmisión en tiempo real de datos crudos a computadora.

En la Figura 4.2 se presenta un esquema con detalle en el SE donde se modulariza la implementación a fin de simplificar el análisis. Está formada por tres módulos: el amplificador de electroencefalografía (EEG), el núcleo, y el módulo de comunicaciones.

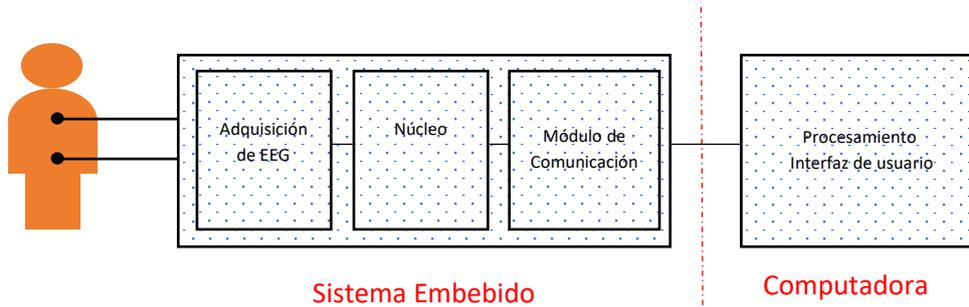


Figura 4.2. Esquema general con detalle en el sistema embebido.

La implementación efectiva de cada uno de estos módulos dependerá de la aplicación, pero en principio se pueden detallar las prestaciones mínimas del hardware a utilizar y el esquema de software necesario. Se necesita de un procesador con capacidad para digitalizar los biopotenciales amplificados y comandar un módulo de comunicación adecuado para enviar los datos crudos a la computadora. Eventualmente realizará algún procesamiento sencillo en tiempo real de la señal previo al envío.

4.2. Esquema general del Sistema embebido.

El esquema general propuesto para la plataforma presenta dos alternativas para el módulo de adquisición de EEG: un amplificador analógico acoplado en alterna con un convertor analógico/digital de 12 bits y un convertor analógico/digital de alta resolución (Σ - Δ de 24 bits) acoplado en continua. En cuanto a la transmisión de las señales a la PC, dispone de tres alternativas: dos inalámbricas (Bluetooth y Zigbee) y otra mediante una fibra óptica (Figura 4.3).

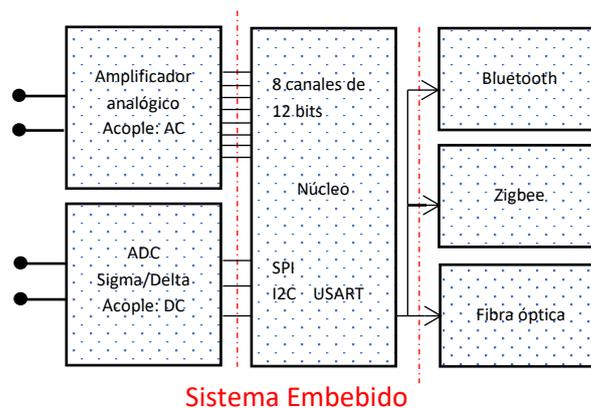


Figura 4.3. Esquema general del SE.

4.3. Adquisición de EEG.

El acondicionamiento y la digitalización de las señales de EEG son tecnologías maduras por lo cual se debe orientar el esfuerzo en reducir el consumo de energía manteniendo las prestaciones [Cincotti, 2006]. Existen dos alternativas: la primera es un amplificador basado en un A/D Σ - Δ de alta resolución (19-24 bits) acoplado en continua y la segunda es un amplificador clásico de bajo consumo acoplado en alterna con un A/D de 12 bits de resolución.

A/D Σ - Δ acoplado en continua:

El uso de un conversor A/D de alta resolución permite adquirir la señal de entrada con su componente de continua, debida a la semi-pila que se genera en la interfaz electrodo-piel conocida como *potencial de electrodo*. Para poder resolver un biopotencial de $\pm 1 \mu\text{V}$ inmerso en un offset de $\pm 300 \text{ mV}$ se necesita un rango dinámico de 110 dB, que exige un A/D con una resolución de más de 19 bits. El A/D seleccionado, que verifica estos requerimientos, es el ADS1256 de Texas Instruments. Para adaptar el nivel de señal al rango de entrada del A/D se utilizó un front-end full diferencial que se muestra en la Figura 4.4. El ADS1256 admite entradas diferenciales de hasta $\pm 5\text{V}$ y la ganancia del front-end se ajustó en 12 veces, con lo cual resulta un rango de entrada de unos $\pm 400 \text{ mV}$.

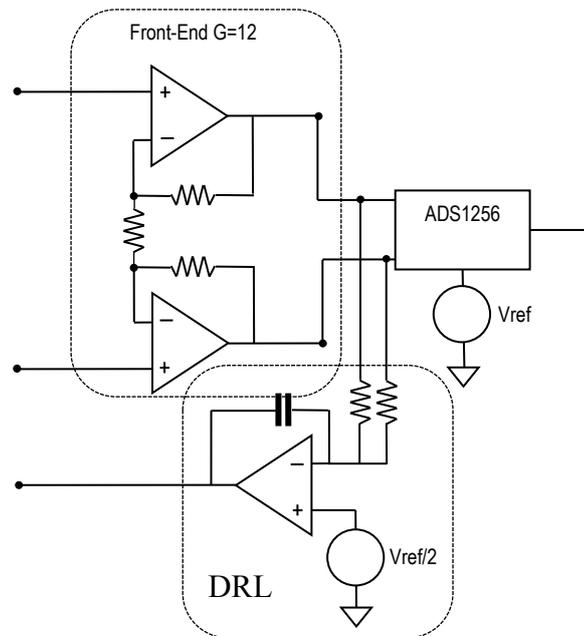


Figura 4.4. A/D Σ - Δ acoplado en continua.

Amplificador analógico acoplado en alterna:

En esta topología se eliminan las componentes de continua, por lo cual es suficiente un rango dinámico (RD) de 54 dB para resolver una señal de $\pm 1 \mu\text{V}$ en un rango de entrada de $\pm 500 \mu\text{V}$. Para alcanzar este rango dinámico es suficiente un A/D con 9 bits de resolución, pero las componentes de baja frecuencia de la señal de EEG ($\approx 0.1 \text{ Hz}$) imponen restricciones al filtro paso alto de acoplamiento dando como resultado tiempos de establecimiento de la línea base de varios segundos.

Se utilizó una red de acoplamiento con una frecuencia de corte de 0.03 Hz [Spinelli, 2003]. Esta red provee de un camino a las corrientes de polarización de los operacionales sin degradar la impedancia de modo común y preservando el rechazo de modo común (CMRR) (Figura 4.5).

El amplificador de instrumentación es el clásico de tres operacionales implementado con operacionales rail-to-rail (TLV2274 de Texas Instruments) con una ganancia de 1000. Su salida

es centrada en medio del rango de entrada del A/D. La etapa siguiente es un inversor con una ganancia de 3.3 veces, resultando una ganancia total de 3300, un rango de entrada de $\pm 378 \mu\text{V}$ y una resolución de $0.18 \mu\text{V}$ considerando un A/D de 12 bits. Antes de digitalizar la señal se filtra con un pasa-bajos en 32 Hz. La salida del amplificador es del tipo Single-Ended, para poder utilizar los A/D que generalmente disponen embebidos los microcontroladores de propósitos generales.

Comparación de las dos alternativas

Ventajas de A/D Σ - Δ acoplado en continua:

- Se evita el acople en alterna evitando lentos transitorios de establecimiento.
- Se reduce el espacio necesario en el circuito impreso por usar menos componentes.
- El filtro antialiasing no es crítico.
- La componente de continua se puede remover por software.
- Tiene mayor inmunidad a los artefactos debido a su mayor rango de entrada.

Y las ventajas para el módulo analógico acoplado en alterna son:

- Utilizando un amplificador por canal, el A/D se puede compartir entre los canales.
- Se requieren menos bits para representar cada muestra, lo cual simplifica el procesamiento digital y reduce el ancho de banda necesario para la comunicación.

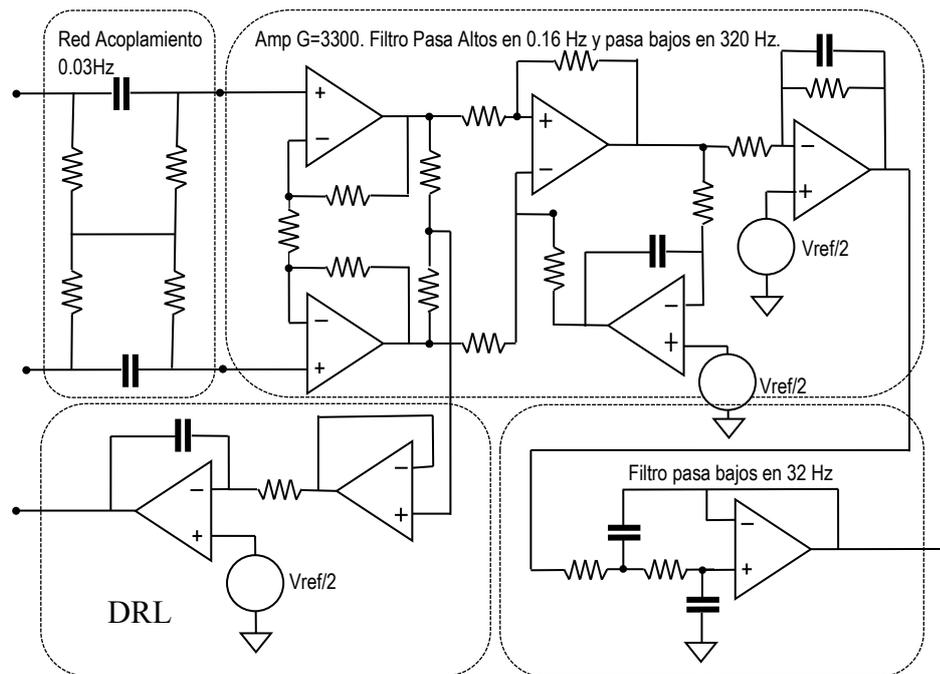


Figura 4.5. Amplificador analógico acoplado en alterna.

La elección de una u otra variante depende de la clase de ICC a implementar. Para una ICC simple y compacta basada en ritmos cerebrales, el módulo acoplado en alterna resulta adecuado porque permite utilizar multiplexado un A/D de resolución moderada (10-12 bits) como los que usualmente disponen los microcontroladores de propósitos generales de gama media. Cuando se utilizan A/D de alta resolución acoplados en continua, se dispone de un amplio rango de entrada para soportar artefactos y transitorios, pero se requiere un convertidor A/D por cada canal y deben enviarse hasta 24 bits por cada muestra de la señal lo cual demanda capacidad del canal con el consiguiente consumo de potencia.

Actualmente existen circuitos integrados comerciales, como el ADS1299, que integran varios

convertidores ADC de alta resolución (24 bits) y circuitos accesorios. Si bien su costo es elevado, permiten implementar un sistema de adquisición de EEG con un único chip y es una muy buena alternativa para ICCs que utilicen varios canales (entre 4 y 8).

4.4. Núcleo.

El núcleo implementado es reutilizable independientemente de la configuración a utilizar. La tarea principal del mismo es comandar el módulo de comunicación para enviar en tiempo real las muestras de EEG a la computadora. Se implementó usando un microcontrolador de gama media de Analog Devices (ADuc841) que incluye un A/D de 8 canales y 12 bits de resolución. La velocidad máxima de operación es de 20 MHz, con lo cual, a futuro se pueden transferir algunas tareas de procesamiento desde la computadora al núcleo.

Para la implementación del sistema operativo embebido se decidió implementar una versión propietaria del tipo primer plano/segundo plano haciendo uso de los recursos de hardware del microcontrolador.

4.5. Sistema Operativo Embebido.

Como ya se ha descrito en la sección 2.6, un sistema operativo multitarea debe implementar mínimamente tres funciones específicas:

1. Un despachante de tareas que tome los recaudos necesarios para iniciar un nuevo proceso en el entorno multitarea.
2. Mecanismos de comunicación y sincronización entre los diferentes procesos que se ejecutan simultáneamente.
3. Un planificador de tareas que determine el orden y la forma en la que se ejecutan las tareas. Como se describe en [Laplante, 1992], para lograr que el SO sea de tiempo real en sistemas multitarea que deben estar en contacto con el mundo real, el planificador más adecuado es del tipo preemptivo basado en prioridades.

Para la implementación del kernel propio sobre un microcontrolador de gama media es conveniente asociar cada una de las tareas con restricciones temporales a interrupciones. Esta asociación nos permite utilizar el hardware de manejo de interrupciones que implementa el microcontrolador para implementar partes del despachante y del planificador de tareas. Así, las tareas con restricciones de tiempo (tareas de tiempo real) se asocian a las interrupciones de hardware (primer plano) y las tareas sin restricciones de tiempo constituyen el lazo de segundo plano del programa, que es constantemente interrumpido por las tareas de primer plano. Esto permite utilizar el manejo de interrupciones del microcontrolador para administrar las tareas a ejecutar simultáneamente.

A continuación, se detallan los recursos que dispone el microcontrolador para lograr implementar el SO:

- Nueve fuentes de interrupción: una interrupción del ADC, tres interrupciones internas de temporizadores, una interrupción del puerto serie, dos interrupciones externas, una interrupción del puerto SPI o I²C y una interrupción del temporizador para contar intervalos de tiempo. Las interrupciones son atendidas por medio de un llamado a subrutina.
- Un registro de manejo de prioridad de dichas interrupciones (IP: del inglés interrupt priority) que permite solamente dos niveles de prioridad. Cuando llega una interrupción

de menor o igual prioridad que la que está siendo atendida, ésta no es descartada, sino que queda pendiente su atención.

- Un registro de habilitación/deshabilitación de interrupciones (IE: del inglés interrupt enable). Incluye un bit (EA) que deshabilita todas las interrupciones a la vez y también permite habilitar o deshabilitar interrupciones de manera independiente.

En este caso, se utiliza como *despachante* de tareas al mecanismo propio del microcontrolador que asocia la ejecución de una subrutina a la llegada de una interrupción. Ante la llegada de una interrupción, el microcontrolador producirá un salto al comienzo de la subrutina correspondiente.

La *comunicación entre tareas* se implementa por medio del uso de variables globales que se pueden acceder desde las distintas tareas del código. El uso de variables globales no es el método más elegante para la comunicación entre tareas, pero es un método rápido y sencillo.

Para la implementación del *planificador preemptivo* basado en prioridades se utilizan los niveles de prioridad de interrupción que se pueden configurar desde el registro IP del microcontrolador. En este caso, se cuenta con tan solo dos niveles de prioridad para las interrupciones. Utilizar solo dos niveles de prioridades puede ser útil en los casos que una de las tareas sea crucial (se le asigna prioridad 1), un grupo de tareas tenga restricciones de tiempo menos estrictas (se les asigna prioridad 0), y un grupo de tareas no tenga restricciones en tiempo (segundo plano). En el caso que se necesiten más niveles de prioridad, la solución es que las propias tareas manipulen constantemente el registro IP en una forma cooperativa, habilitando o deshabilitando la posibilidad de ser interrumpidas por otras tareas. Lo ideal es que se disponga de tantos niveles de prioridad como tareas deban ejecutarse.

Para la versión implementada, en primer plano está operando la actividad que adquiere las muestras de la señal de EEG y genera los paquetes de datos; y en segundo plano está funcionando la tarea que comanda el módulo inalámbrico para enviar los paquetes a la computadora. En la Figura 4.6 se muestran extractos de código en lenguaje C donde se gestionan las tareas. El código que se observa en la Figura corresponde a la implementación de SO utilizada para el front end analógico y utilizando el conversor analógico digital de 12 bits incluido en el microcontrolador. En la Figura se puede observar la tarea en segundo plano que se realiza dentro de la función main (código en ventana lado izquierdo) y la tarea en primer plano que se realiza desde la rutina de interrupción del temporizador 1 (código en ventana lado derecho). También se observa el uso de variables globales para compartir la información entre tareas y la inicialización de las interrupciones por medio de la función `ini_interrupts()`.

El código a implementar para el SO propietario que permite comandar las tareas en el caso de utilizar el front end digital (conversor A/D de alta resolución con 4 canales diferenciales) es similar en cuanto a la distribución de tareas en primer y segundo plano. Cambian algunas funciones porque en este nuevo escenario el ADC es externo al microcontrolador y se accede por medio del puerto SPI. La base de tiempo se sigue manejando por medio de un temporizador que genera las interrupciones donde se muestrea la señal (primer plano) y dentro del módulo principal (main) se continúa enviando la información al módulo inalámbrico.

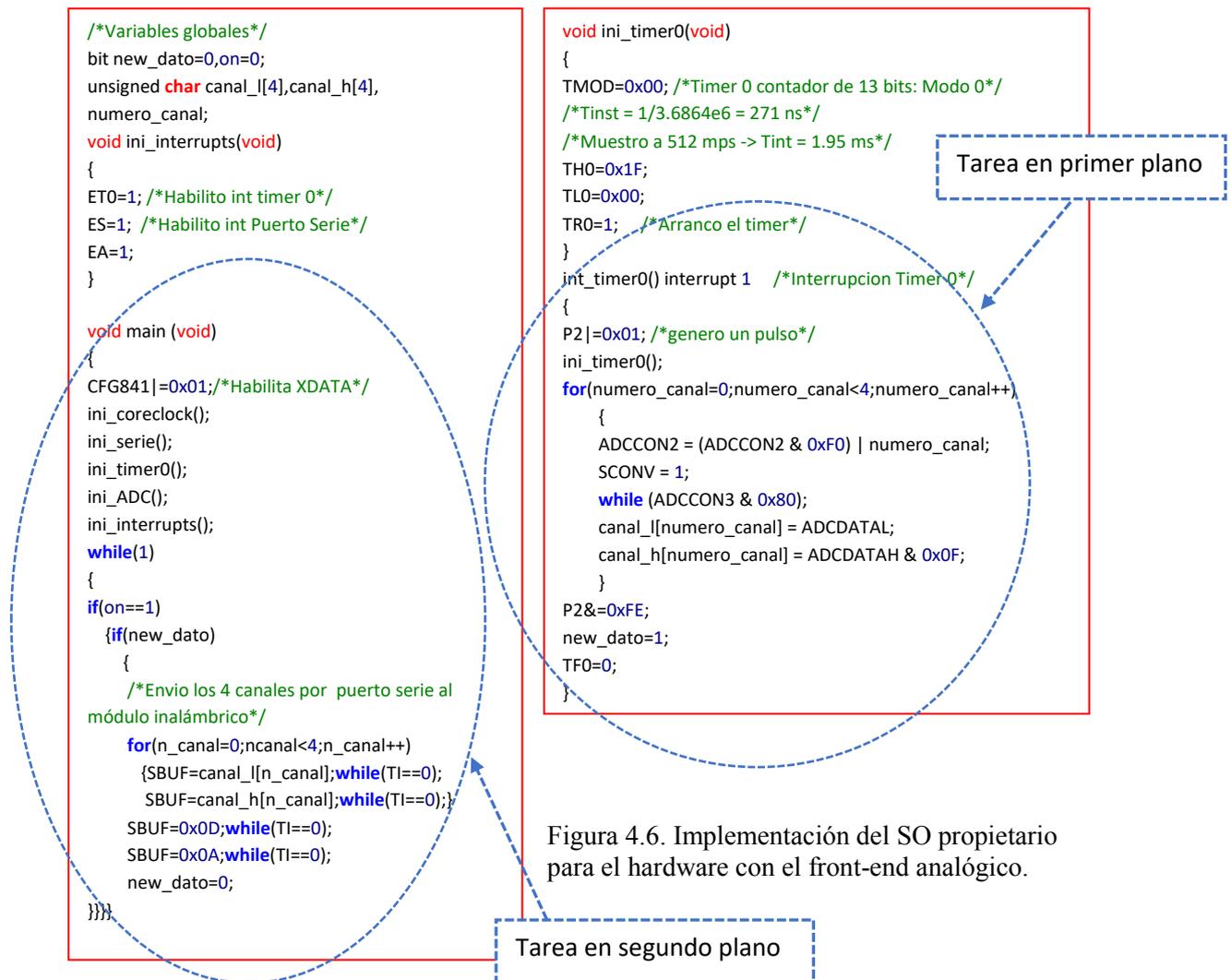


Figura 4.6. Implementación del SO propietario para el hardware con el front-end analógico.

4.6. Módulo de comunicación.

Existen diferentes alternativas para la implementación de este módulo que permiten aislar eléctricamente al usuario de la ICC de la computadora. La fibra óptica es una solución simple, económica y de bajo consumo, pero mantiene al usuario físicamente conectado a la computadora. Para evitar esto, las nuevas tendencias optan por soluciones inalámbricas [He et al., 2016]. Las soluciones implementadas por la empresa G-tec (IntendiX) y por OpenVibe son un claro ejemplo (ver Capítulo 3).

Las dos variantes inalámbricas en la plataforma son los protocolos Bluetooth y Zigbee. El motivo principal para su selección es que son estándares muy difundidos. A continuación, se presenta un análisis de la topología de red para estos dos protocolos detallando los distintos estados en los cuales se pueden encontrar los nodos, con el objeto final de comprender y comparar los distintos tiempos de activación y el consumo.

4.6.1. Topología de Red Bluetooth.

En la Figura 4.7 se presenta una piconet donde cada una de las circunferencias pequeñas (M,S,P,Sb) representa un radio bluetooth en un estado determinado. Los radios bluetooth se conectan formando una piconet. Cada piconet está formada por un maestro y hasta 7 esclavos. Los radios bluetooth pueden operar como maestros o como esclavos indistintamente, y la

disposición de la red se determina cuando se está formando. Normalmente, el radio que inicia la conexión se convierte en el maestro, aunque existe la posibilidad de invertir los roles por medio de una función “swap”.

Los estados en los que pueden estar cada uno de los nodos son: M: Maestro (Master), S: esclavo (Slave), P: estacionado (Park) o bien Sb: Standby.

Para poder formar una piconet, el radio necesita conocer dos parámetros: el patrón de salto en frecuencia del radio con el cual se quiere conectar y la fase dentro del patrón. Cada radio bluetooth posee un “Global ID” único que se utiliza para generar su patrón de salto en frecuencia. A la hora de formar una piconet, el maestro comparte su “Global ID” con otros radios, que luego se convierten en esclavos siguiendo el patrón de salto en frecuencia del maestro. El maestro también comparte el offset de su reloj con los esclavos, por lo cual éstos pueden sincronizarse con el patrón de salto en frecuencia del maestro de manera correcta. Esta información se intercambia por medio de unos paquetes denominados FHS.

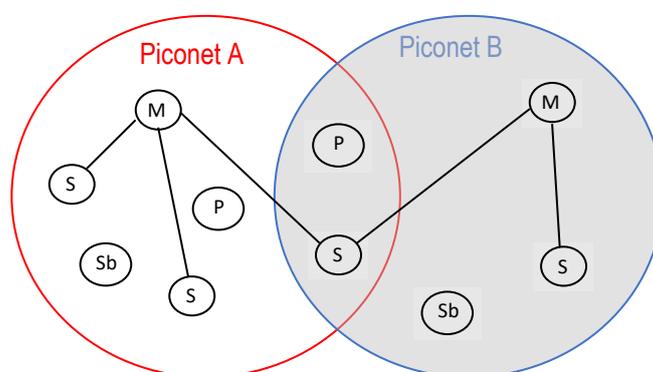


Figura 4.7. Topología de red Bluetooth.

Normalmente, los radios que no están conectados en la piconet se encuentran en el modo “Standby”. En este modo, el radio está esperando que otro radio lo encuentre (“Inquire”) o bien está escuchando por un pedido de unión para formar la piconet (“Page”). Cuando un radio hace público un pedido de “Inquire”, los radios que escuchan responden con su paquete FHS (Global ID y offset del reloj), con lo cual el maestro conoce todos los radios que están en su área de alcance.

Para formar una piconet, el maestro hace un “Page” a otro radio con el “Global ID” del destinatario (obtenido previamente por medio de un “Inquire”). El radio destino del “Page” responde con su “Global ID” y el maestro le pasa su paquete FHS. Luego el radio destino carga el “Global ID” y el offset del reloj del maestro convirtiéndose en esclavo.

Una vez que un nodo se une a la piconet, se le asigna una dirección de miembro activo de 3 bits (AMA) permitiendo que otros miembros de la piconet lo direccionen. Una vez que una piconet tiene 8 miembros activos, el maestro debe pasar alguno de los esclavos al modo “Park”. Este radio se mantiene coordinado con la red pero libera su AMA y obtiene una dirección de miembro pasivo de 8 bits (PMA). La combinación de AMA y PMA permite que 256 miembros residan en la piconet, pero solo 8 con el AMA activo pueden transferir datos.

Los radios que están estacionados (“Parked”) esperan por un intervalo de tiempo señalado para determinar si existe información que se dirija a ellos. De esta forma, el maestro puede hacer un broadcast a todos los esclavos, ya sea activos o estacionados.

Los radios que no están conectados a la piconet se encuentran en el modo “Standby”. Estos radios escuchan por “inquires” o “pages” de otros radios cada 1.25 segundos por medio de métodos de búsqueda denominados “inquire scan” o bien “page scan”.

El proceso “Inquiry” envuelve a dos o más radios, uno realizando el “inquiry” en busca de

radios a su alcance y otros realizando el “inquiry scan”. Este proceso se realiza en una única secuencia de 32 canales. El radio que realiza el “inquiry scan” escucha cada 1.25 segundos en uno de esos 32 canales durante 10ms, luego repite la búsqueda en el canal siguiente. Un radio con el “inquiry scan” habilitado continúa en este estado hasta que se deshabilita.

Una vez que un radio ha sido encontrado (por medio del “Inquiry”) y unido a la piconet (por medio del “page”) ya se encuentra funcional y puede moverse entre los distintos estados de un radio bluetooth, como se observa en la Figura 4.8.

En el estado conectado, el radio posee una dirección de miembro activo de 3 bits (AMA) por medio de la cual puede dirigir datos a distintos dispositivos dentro de la red (el master posee siempre la dirección 0). Se puede hacer un broadcast en la piconet por medio del maestro enviando un paquete a la dirección 0. Los radios se pueden mantener en el estado conectado con la piconet (mantiene el patrón de salto de frecuencia y el offset del reloj) y al mismo tiempo estar en un modo de bajo consumo. Para esto, los radios se pueden poner en el estado “Park”, “Hold” o “Sniff”. En los estados Hold y Sniff, los radios están durmiendo durante ciertas ranuras y se despiertan en intervalos bien determinados. En el estado Sniff el radio puede transferir datos en ese intervalo (por ejemplo, un teclado puede enviar y recibir datos cada 20 ranuras), mientras que en el Hold no puede transferir datos. En el estado “Park”, el radio se va a dormir y se le asigna una dirección de miembro pasivo (PMA). En este estado, el radio va a escuchar durante intervalos señalizados para ver si el maestro ha:

- Preguntado al radio estacionado para convertirse en miembro activo.
- Preguntado si algún dispositivo estacionado desea convertirse en miembro activo.
- Mandado algún dato broadcast.

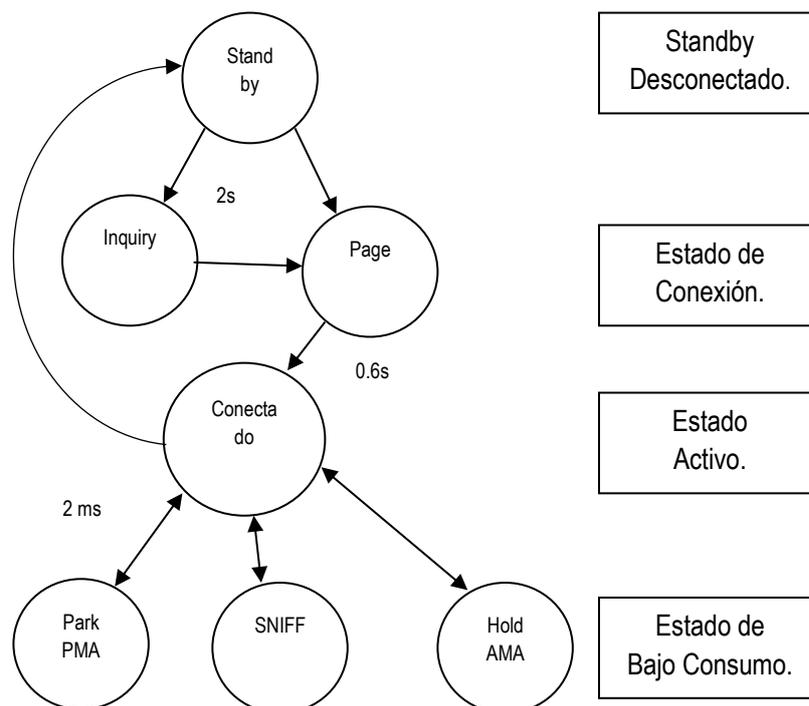


Figura 4.8. Esquema de transición entre estados.

En el estado conectado, los radios bluetooth pueden utilizar dos tipos de paquetes: SCO (Synchronous Connection Oriented) o bien paquetes ACL (Asynchronous Connectionless). Los paquetes SCO están asociados con datos isócronos, es decir, principalmente voz. Típicamente es un paquete simétrico de 1, 2 o 3 ranuras que se reserva para su uso en la piconet. Para poder tener

una conexión SCO, el radio debe haber establecido una conexión ACL anteriormente. Una vez que se agrega un enlace SCO, un maestro o un esclavo pueden enviar paquetes SCO sin recibir un “poll”.

Existen tres esquemas de corrección de errores en la banda base bluetooth: 1/3 FEC, 2/3 FEC y ARQ (Automatic Repeat Request).

En la Figura 4.8 no se ha incluido el proceso de enumeración, que es el que se ejecuta cuando dos dispositivos se emparejan por primera vez. En este momento es cuando se producen los intercambios de identificación y contraseña de validación.

4.6.2. Topología de Red Zigbee.

El protocolo Zigbee soporta las topologías de red en estrella, malla y árbol (Figura 4.9). En la topología en estrella un único coordinador controla la red, siendo responsable de la inicialización de los otros dispositivos (usuarios finales). Estos usuarios finales intercambian la información de manera directa con el coordinador al igual que en las piconet del protocolo Bluetooth. En las topologías en malla y árbol el coordinador es responsable de la inicialización y el manejo principal de la red, con los routers (FFD: del inglés Full Functional Device) que permiten también el ingreso de nuevos dispositivos a la red.

La topología en árbol está formada por subredes que se comunican por medio de los routers. Un ejemplo de esta topología es la domótica en edificios, pudiendo existir muchos dispositivos en distintos pisos. En este caso, el alcance de la señal es mayor que desde un coordinador a un cliente dado que se puede acceder a clientes en los distintos pisos por medio de los routers.

En la topología en malla, la red se vuelve a reconfigurar con la entrada de cada nuevo dispositivo durante el proceso de inicialización con lo cual se trata de optimizar el tráfico de datos. Con esta topología se pueden construir redes más largas y complejas que permiten el control y monitoreo de grandes áreas (por ejemplo, el control de temperatura y humedad en un viñedo).

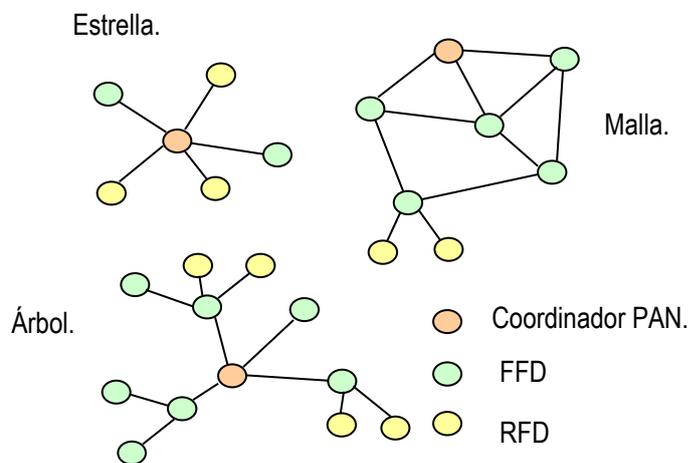


Figura 4.9. Topologías de red Zigbee.

Durante el proceso de formación de este tipo de redes, un dispositivo con capacidad de coordinación detecta un canal relativamente libre de interferencia por medio de un escaneo de energía y luego se autoconfigura como coordinador (coordinador PAN: del inglés Personal Area Network). En la medida que nuevos dispositivos quieran agregarse a la red, deberán emitir pedidos para recibir balizas desde dispositivos que le permitan unirse a la red. Inicialmente solo el coordinador PAN responderá. Cuando se agregue algún router (FFD) a la red, este también podrá emitir balizas para permitir a nuevos dispositivos integrarse. Este proceso se denomina asociación y se intercambian varios mensajes durante su proceso. Un factor importante es la capacidad de cada router de aceptar nuevos dispositivos como sus hijos. En este factor es donde

esta tecnología inalámbrica difiere de otras como por ejemplo Wi-fi, que acepta una estructura padre-hijo solamente. Por ejemplo, un dispositivo que se une a un coordinador podría también ser un coordinador que a su vez permita la unión de nuevos dispositivos hijos. Como resultado podemos lograr múltiples niveles de asociación a la red.

4.6.3. Comparativa

En la tabla 4.1 se presenta un resumen comparativo con los parámetros de interés entre ambos protocolos:

TABLA 4.1
COMPARACIÓN ENTRE BLUETOOTH Y ZIGBEE

Protocolo	Bluetooth	Zigbee
Aplicación	Redes Personales y reemplazo de cables.	Redes de sensores, monitoreo y control.
Rango	1-100 m.	1-100 m.
Ancho de Banda	720 kbps.	250 kbps.
Topología	Estrella (Pico net), híbrida (scatternet)	Estrella, malla o árbol.
Nodos por red	7	65000
Consumo de energía Activo	80 mA.	60 mA
Consumo de energía Dormido	90 uA.	10 uA.
Proceso enumeración.	20 s.	30ms.
Tiempo de activación desde dormido.	3s.	15 ms.
Tiempo de acceso al canal	2 ms.	15 ms.

El dispositivo a implementar se pretende utilizar para ensayos con registros de señales de EEG. Considerando 8 canales de adquisición con muestras en 24 bits y una frecuencia de muestreo de 1000 mps se requiere un ancho de banda de 192 kbps.

$$BW[kbps] = 8[ch] * 24 \left[\frac{b}{m} \right] * 1000 \left[\frac{m}{s} \right] = 192 kbps$$

Ambos protocolos poseen suficiente ancho de banda para transferir la señal de EEG en bruto a la computadora en tiempo real. El parámetro de mayor interés para dispositivos ICC portables es el consumo de energía. Zigbee ha sido desarrollado atendiendo a este requerimiento y su principal ventaja respecto de Bluetooth es la velocidad de activación desde el modo dormido.

Para la implementación práctica se utilizaron dos módulos disponibles de bajo costo, uno para cada protocolo.

Para el protocolo bluetooth se utilizó un módulo BR-SC30N de la empresa BlueRadios. Este es un módulo de comunicación inalámbrica que verifica la norma Bluetooth V1.2. El módulo se puede configurar, comandar y controlar por medio de un enlace bluetooth (via RF) o bien por medio de un puerto serie utilizando comandos AT como un modem estándar. De esta forma, el módulo cuenta con una API bien definida por medio de comandos AT que permiten al usuario comandar el módulo de forma sencilla.

Para el protocolo Zigbee se utilizó un módulo XBee de la empresa DigiRF. Al igual que el módulo bluetooth, se puede configurar rápidamente por medio de comandos AT haciendo uso de un terminal propietario denominado X-CTU.

Como se puede observar en el código de la Figura 4.6, para los primeros ensayos con ambos

protocolos se decidió enviar a la computadora la señal de 4 canales muestreados a 512 mps, muestra por muestra, sin armar paquetes ni utilizar modo de bajo consumo. Con los módulos en estado activo todo el tiempo, sus consumos son similares: 60 mA para Zigbee y 70mA para Bluetooth.

Con el objeto de reducir el consumo se decidió ensayar con los modos dormir (del inglés sleep) de ambos módulos. En la aplicación de interés, el modo dormido para el módulo Bluetooth no se puede utilizar debido a su lento tiempo de activación desde Stand-by (3 segundos), que impide el envío de señales en tiempo real a la computadora. Su utilización requeriría el armado de paquetes de datos muy extensos, del orden de 6 a 10 s, por lo cual se recibiría en computadora una señal muy retardada que pierde sentido para una realimentación al usuario.

El tiempo de activación desde dormido para Zigbee es de solo 15 ms, por lo cual se puede utilizar el modo dormido para ahorrar energía y aun así poder enviar en tiempo real la señal a la computadora. Armando paquetes de datos de 500 ms se logra un retardo aceptable para el arribo de datos a la computadora, con lo cual se puede realimentar la forma de onda al usuario (gráfica temporal o análisis espectral) en tiempo real. Con esta modificación se logró reducir el consumo desde 60 a 35 mA.

Cabe destacar que para el enlace Zigbee, se necesita conectar un módulo en el extremo de la computadora. En este sentido, la ventaja del protocolo Bluetooth es su amplia implementación en teléfonos inteligentes, notebooks y tabletas, que facilita la transferencia de datos usando un dispositivo estándar sin añadir hardware.

Otra variante es transferir el procesamiento al sistema embebido, con lo cual el canal de comunicación se utilizaría solo para enviar comandos reduciendo considerablemente el consumo de energía.

4.7. Resultados experimentales.

El sistema embebido fue utilizado y verificado en la implementación de dos ICC diferentes. La primera utiliza un amplificador analógico de 3 canales acoplado en alterna y un módulo de comunicación Zigbee. La segunda posee un amplificador monocanal acoplado en continua con A/D Σ - Δ y módulo de comunicación Bluetooth (Figura 4.10).

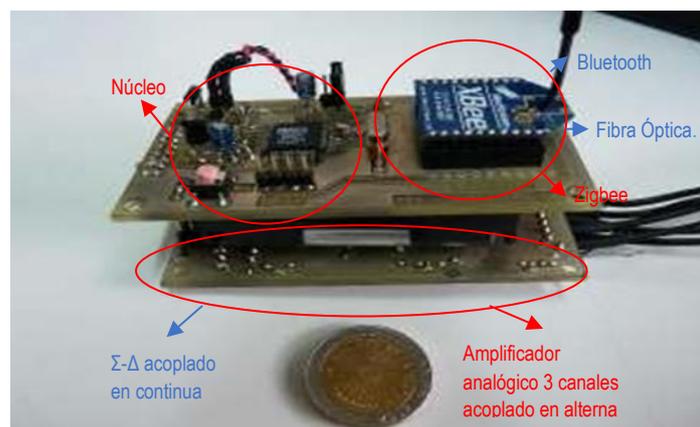


Figura 4.10. Implementación práctica con los distintos módulos resaltados en rojo.

El sistema embebido configurado con un amplificador monocanal acoplado en continua y un módulo Bluetooth se verificó implementando una ICC basada en el ritmo alfa [Pfurtscheller, 1999]. La señal de EEG digitalizada se envía a la computadora en tiempo real, y en esta se ejecuta un algoritmo que estima el nivel de energía en la banda alfa [Spinelli, 2000].

En primer momento, para verificar la correcta operación del sistema se realizó un

procesamiento de la señal off-line usando un programa de matemáticas. En la Figura 4.11 se presenta un registro propio de la señal de EEG capturada en las posiciones O_1-O_2 del sistema internacional 10-20. Como se puede observar en la figura superior, a partir del segundo 44 del registro el usuario cierra los ojos con lo cual aparece una marcada oscilación cercana a los 12 Hz (ritmo alfa). Como se observa en la figura central, la señal se eleva al cuadrado para luego obtener la envolvente decimada (Figura inferior). En función de umbrales móviles sobre la señal envolvente, se pueden producir activaciones cortas o largas.

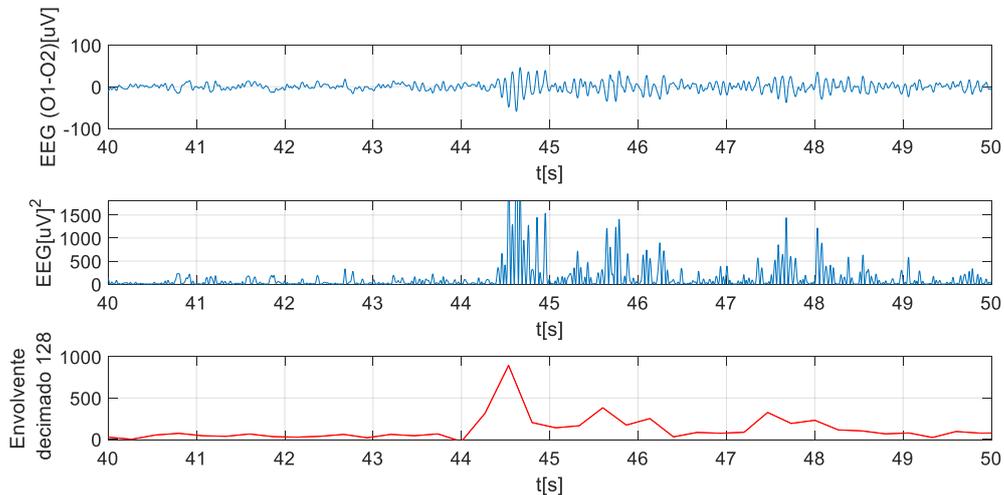


Figura 4.11. Registro y procesamiento de ritmo Alfa visual.

Como segundo paso, junto con el grupo de trabajo y como se describe en [Haberman et. al, 2010], se desarrolló un software de tiempo real con interfaz gráfica de usuario. La ICC presenta un menú rotante en la pantalla de la computadora (Figura 4.12) que permite al usuario seleccionar y comandar distintos dispositivos conectados a la misma. El software que estima el nivel de energía en la banda alfa corre en segundo plano utilizando un umbral adaptivo.

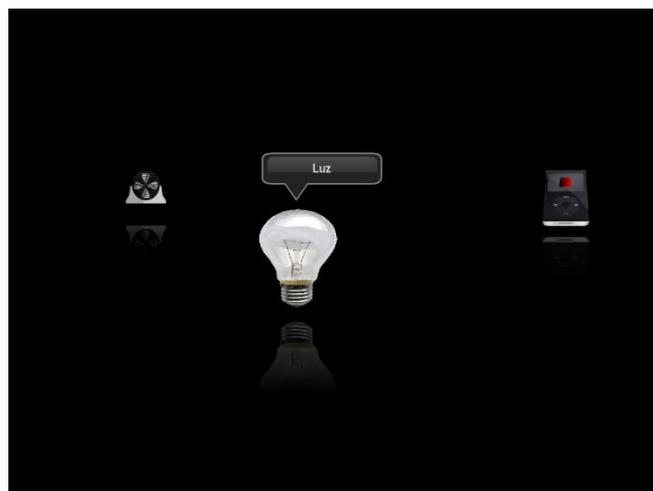


Figura 4.12. Interfaz de usuario para selección y comando de dispositivos.

En la segunda configuración, el sistema embebido cuenta con el amplificador de 3 canales acoplados en alterna y un módulo de comunicación Zigbee. Se verificó digitalizando y enviando a la computadora el ritmo beta relacionado al movimiento de las manos (derecha e izquierda) y el ritmo alfa visual. En la computadora se utilizó un software de tiempo real con interfaz gráfica de usuario que estima los niveles de energía en las bandas beta (16-24 Hz) y alfa (10-14 Hz) [Haberman et. al, 2010]. Comparando estos niveles contra umbrales móviles el usuario puede

comandar el desplazamiento de un objeto en la pantalla de la computadora (Figura 4.13).

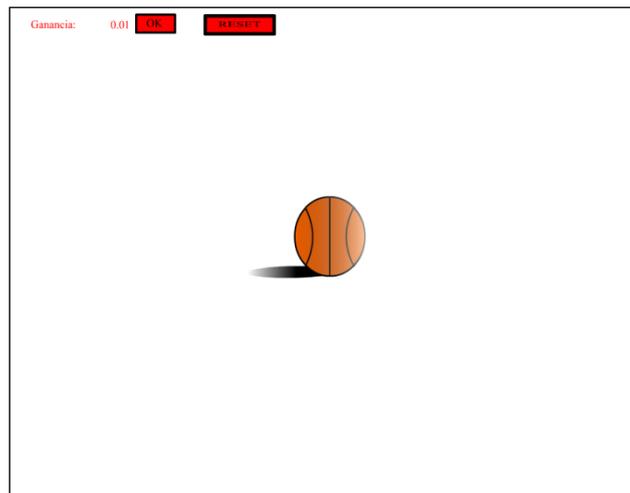


Figura 4.13. Interfaz de usuario para el movimiento en una dimensión de un objeto.

En la Figura 4.14 puede observarse el sistema embebido vestible de tres canales de EEG con los electrodos en las posiciones C3-F3, C4-F4 y O1-O2 del sistema internacional 10-20 para captar los ritmos motores y el ritmo alfa visual.

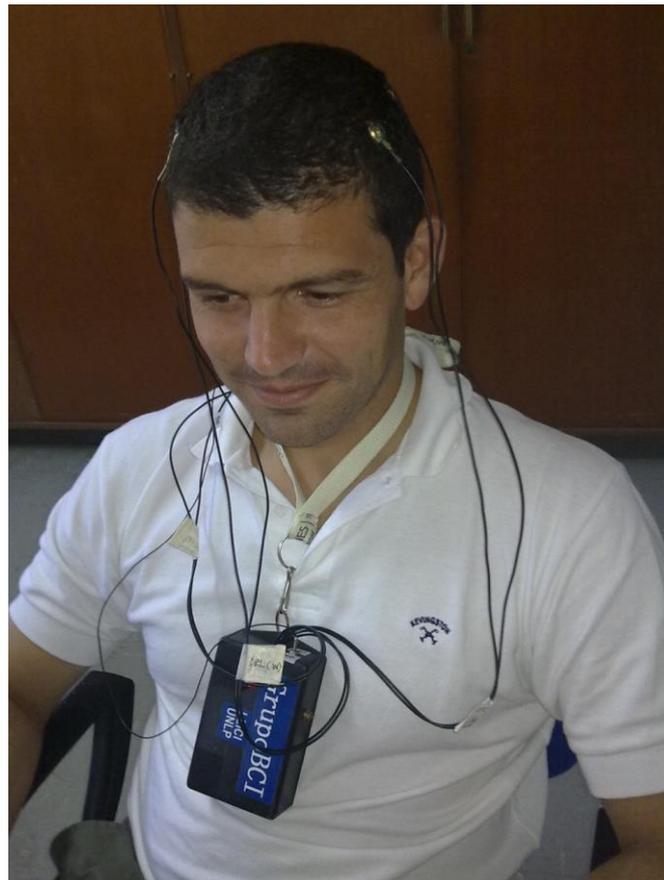


Figura 4.14. Sistema embebido vestible de tres canales de EEG.

En la Figura 4.15 puede observarse el sistema ICC completo con el software de tiempo real

corriendo en la computadora y su interfaz de usuario. En esta aplicación, el usuario de la ICC debe comandar el movimiento de la pelota en una dimensión (derecha-izquierda) por medio de los ritmos motores. Para generar el movimiento a derecha de la pelota, el usuario de la interfaz debe imaginar o realizar el movimiento de su mano derecha. De la misma forma, para mover la pelota hacia la izquierda, el usuario deberá imaginar o bien mover la mano izquierda.



Figura 4.15. ICC completa funcionando con ritmos motores.

4.8. Conclusión.

En este punto se ha presentado la implementación de una ICC versátil basada en un Sistema embebido autónomo, pequeño y configurable. Se exploraron diversas variantes para los módulos que forman el SE que se compararon y verificaron en aplicaciones reales sobre interfaces cerebro computadora.

4.9. Sistema embebido monocanal

A partir de la experimentación con distintas configuraciones de la plataforma flexible se implementó un SE monocanal compacto para captura de biopotenciales en tiempo real. Para esta aplicación se eligió el amplificador acoplado en continua, un conversor analógico/digital de alta resolución (Σ - Δ de 24 bits) y transmisión de datos mediante bluetooth. Esta configuración requiere un reducido número de componentes y presenta una resolución de $0.05\mu\text{V}$ en un rango de entrada de $\pm 450\text{ mV}$ siendo apropiada para señales de EEG, ECG y EMG. La transmisión mediante el protocolo bluetooth permite recibir las señales en la PC sin requerir agregar hardware. En la Figura 4.16 se muestra un esquema de su estructura.

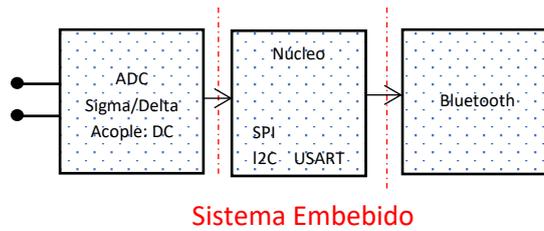


Figura 4.16. Esquema del SE.

4.10. Resultados experimentales con SSVEP.

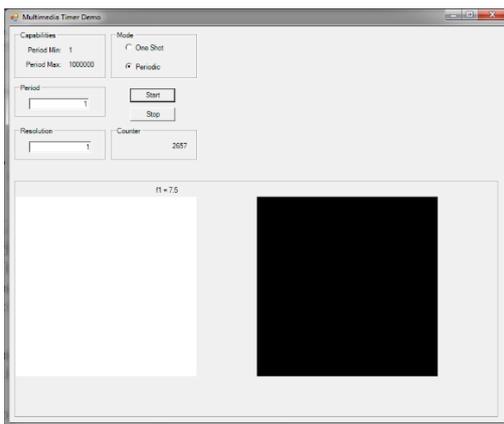
Para validar el hardware se realizaron algunos ensayos preliminares sobre la utilización de potenciales evocados visuales de estado estacionario en interfaces cerebro computadora. Este tipo de potenciales, descritos en el Capítulo 3, son muy utilizados en ICC dado que requieren de tan solo un par de electrodos de EEG, no es necesario entrenamiento para su utilización, el procesamiento requerido es sencillo y se logran altas tasas de transferencia.

4.10.1. Estímulo visual (Primeros ensayos)

Con el objetivo de testear las distintas variantes normalmente utilizadas para implementar el estimulador visual se ensayaron distintas configuraciones utilizando un monitor LCD y otras utilizando diodos emisores de luz (LED).

En la Figura 4.17 se presentan imágenes de los primeros ensayos realizados con potenciales evocados usando como estimulador visual:

- (a). Un patrón blanco-negro alternante en un monitor LCD.
- (b). Un LED color rojo.
- (c). LEDs color azul dentro y fuera de una caja para modificar la iluminación de fondo.



(a)



(b)



(c)

Figura 4.17. Estimuladores utilizados (primeras pruebas). (a): Monitor LCD
(b) y (c): LED con distinta iluminación ambiental.

Los potenciales evocados son notorios para los distintos tipos de estimuladores, pero se debe tener en cuenta que el número de estímulos que se pueden generar en un monitor es menor que la cantidad que puede generarse utilizando LEDs, ya que los mismos deben ser submúltiplos de la frecuencia de barrido del monitor (60, 75 o 140 Hz).

Por otro lado, el potencial evocado utilizando LEDs tiene mayor amplitud que el evocado con un monitor para las mismas condiciones de entorno [Zhu, 2010][Wang, 2008]. Este es uno de los motivos reportados en la bibliografía por los cuales se logran mejores tasas de transferencia en ICCs con estimuladores de LEDs. En [Zhu, 2010] se presenta la tasa de transferencia media lograda por distintos grupos de investigación siendo los mismos:

- Estimulación con LEDs: 42 bits/min.
- Gráficos simples en monitor: 35 bits/min.
- Inversión de patrón en monitor: 26 bits/min.

Por otro lado, existen muchos factores a tener en cuenta que modifican la amplitud del potencial evocado tales como: iluminación ambiental, contraste, color y tamaño del estímulo, distracción visual, etc [Bieger ,2010][van Hemert ,2009]. Los SSVEP tienen mayores amplitudes en la banda de 6 a 24 Hz y son más simples de detectar, pero también se utilizan frecuencias mayores que producen menor fatiga visual [Diez et. al, 2011]. La discriminación en frecuencia es muy alta, pueden detectarse estímulos cuyas frecuencias difieran en 0.2 Hz o menos.

Se realizaron un conjunto de medidas variando las condiciones del potencial registrado. En una primera instancia se registró el nivel base, es decir la diferencia de potencial entre las posiciones O1 y O2 del sistema internacional 10-20 sin ningún estímulo visual. En la Figura 4.18 se presenta un segmento de señal de 10 segundos y su respectivo espectro obtenido mediante la transformada rápida de Fourier (FFT). La señal fue muestreada a 480 mps, con lo cual el tamaño de ventana es de 4800 muestras y se obtiene una resolución en frecuencia de 0.1 Hz.

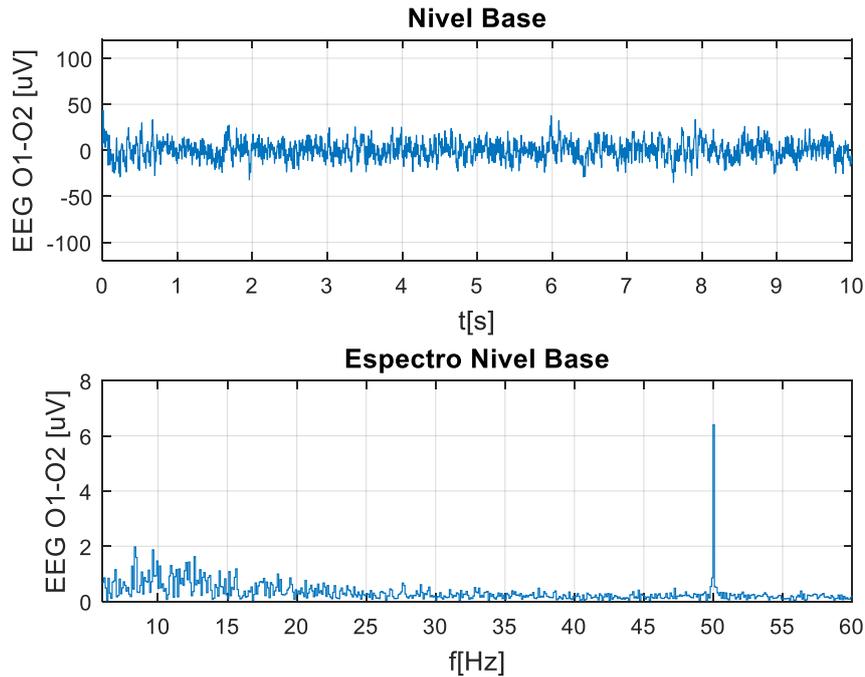


Figura 4.18. Nivel Base.

En segundo lugar, se registró el ritmo alfa. Como ya se ha mencionado, este es un ritmo muy utilizado que se manifiesta en condiciones de relajación visual. La forma más típica de conseguirlo es manteniendo los ojos cerrados, aunque también puede conseguirse evitando fijar la vista o manteniendo la visión desenfocada. Como se puede observar en la Figura 4.19, a partir del segundo 3 el usuario cierra los ojos y medio segundo después se produce el ritmo alfa durante 1,5 segundos aproximadamente.

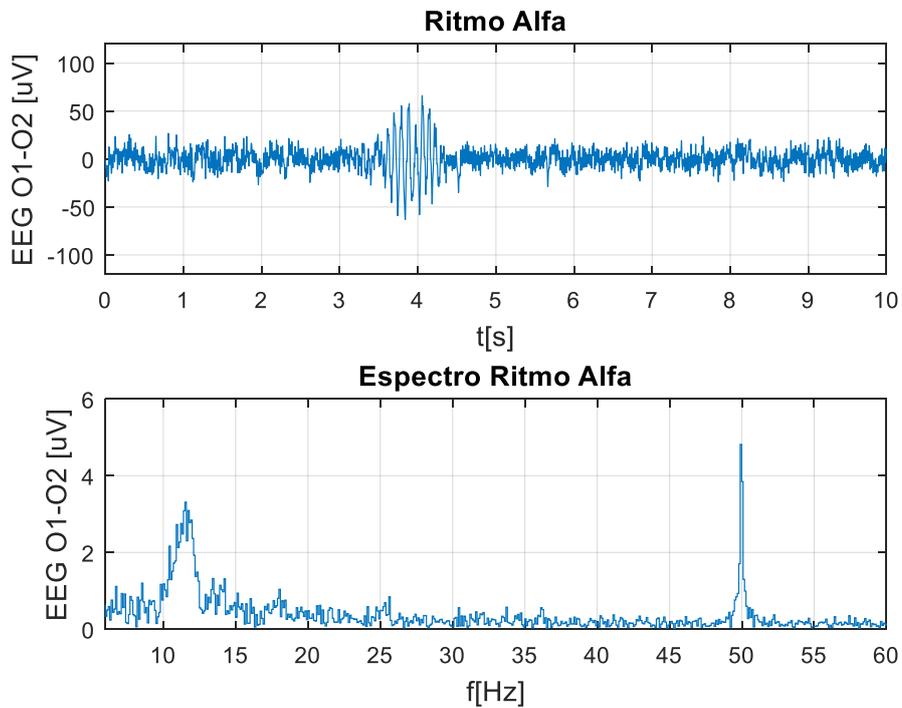


Figura 4.19. Ritmo Alfa.

Orientando las medidas hacia el desarrollo de una ICC embebida y dado que el potencial que evocan es de mayor amplitud, se decidió realizar ensayos utilizando el estimulador basado en LEDs en la banda de 6-24 Hz. En la Figura 4.20 se presenta un segmento de 10 segundos de señal y su espectro, para un potencial evocado por un estímulo en 18 Hz generado con un LED azul. Se observan claramente las componentes en la frecuencia de estímulo y su primer armónico.

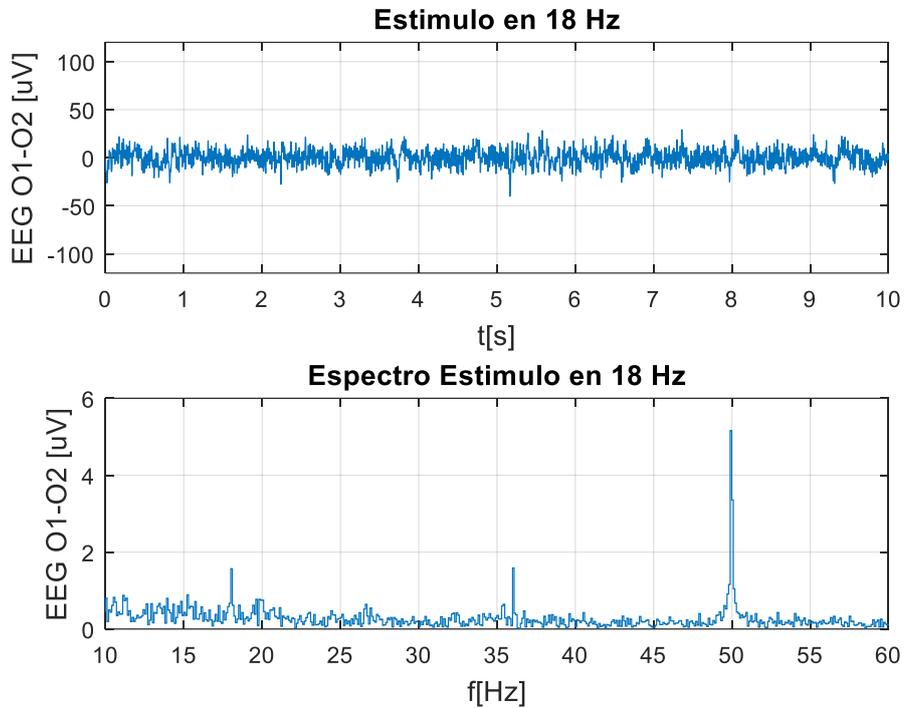


Figura 4.20. Estímulo con led en 18 Hz.

La resolución en frecuencia reportada en la literatura para este tipo de sistemas es de 0,2 Hz [Gao et. al, 2003]. Se realizaron un conjunto de medidas separando los estímulos en 0,2 Hz en varias frecuencias. En la Figura 4.21 se presentan los resultados para un par de medidas con estímulos centrados en 16 Hz (rojo) y 16.2 Hz (azul). De la misma forma que en los registros previos se utilizaron ventanas de registros de 10 segundos, por lo cual la resolución en frecuencia es de 0.1 Hz.

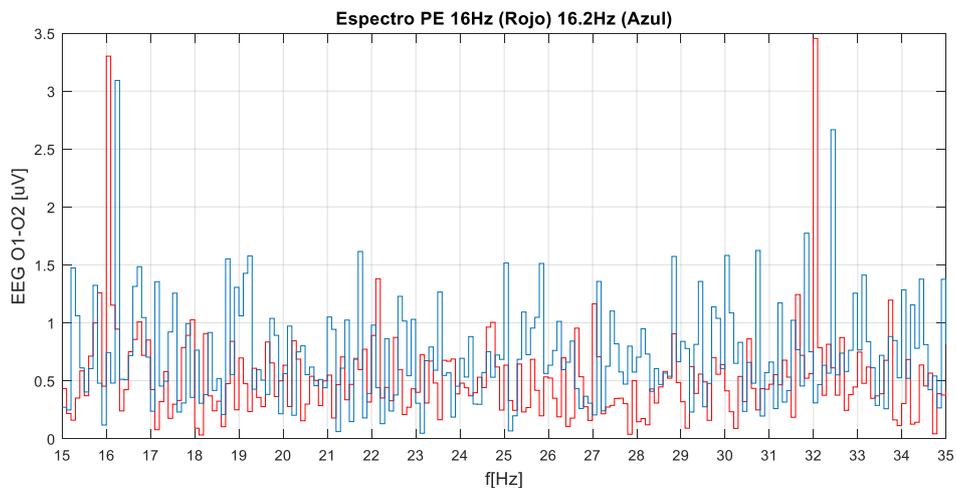
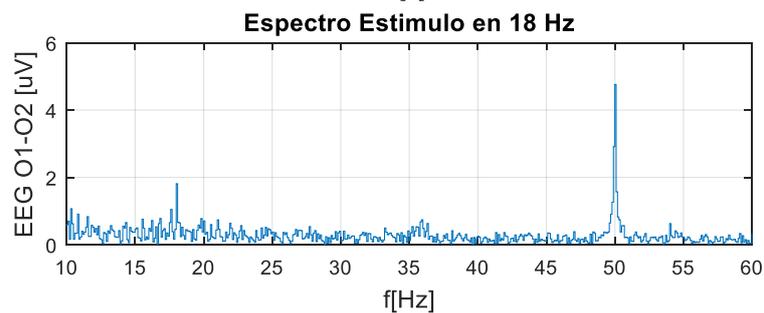
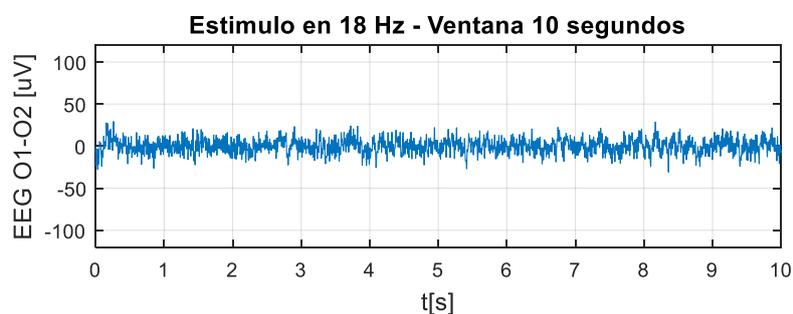


Figura 4.21. Espectro de potenciales evocados por estímulos en 16 Hz (Rojo) y 16.2 Hz (Azul).

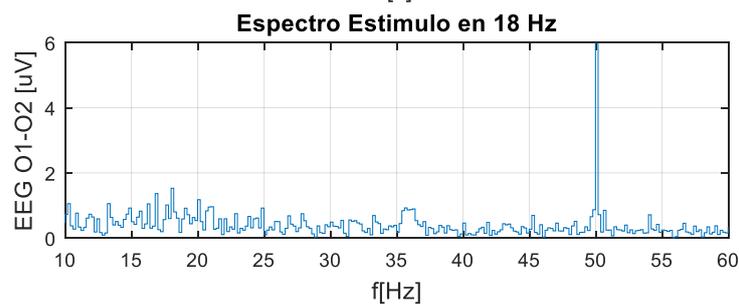
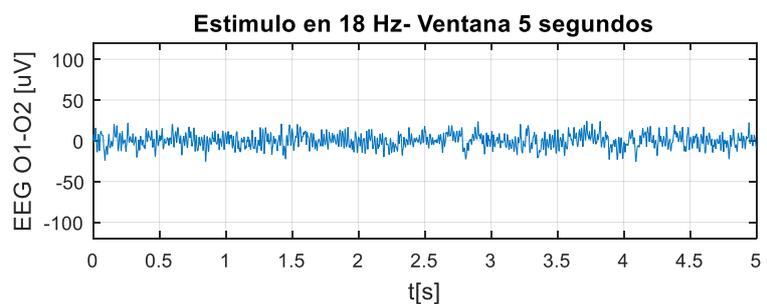
Como se observa en la Figura, los estímulos pueden ser claramente diferenciados entre sí y además poseen una muy buena relación señal-ruido. A diferencia de las medidas previas, estos ensayos fueron realizados sin luz ambiente, lo que se ve reflejado en el aumento de la amplitud del potencial evocado para la misma ventana de análisis.

Existe una importante relación entre la frecuencia de muestreo, el tamaño de ventana y la frecuencia de estímulo que permite optimizar la recuperación de los SSVEP.

Tamaño de ventana de análisis: en los ensayos previos se observa que con ventanas de análisis de 10 segundos resulta sencillo recuperar los potenciales evocados. A medida que aumenta el tamaño de ventana, también lo hace la relación señal a ruido, en caso que la señal esté siempre presente y el ruido sea aleatorio. Para este tipo de implementaciones el desafío consiste en recuperar los SSVEP en el menor tiempo posible, es decir, con ventanas de tiempo tan cortas como se pueda. A continuación, se presenta un análisis de la recuperación de potenciales en base a un SSVEP en 18 Hz, variando el tamaño de ventana (Figura 4.22):



(a)



(b)

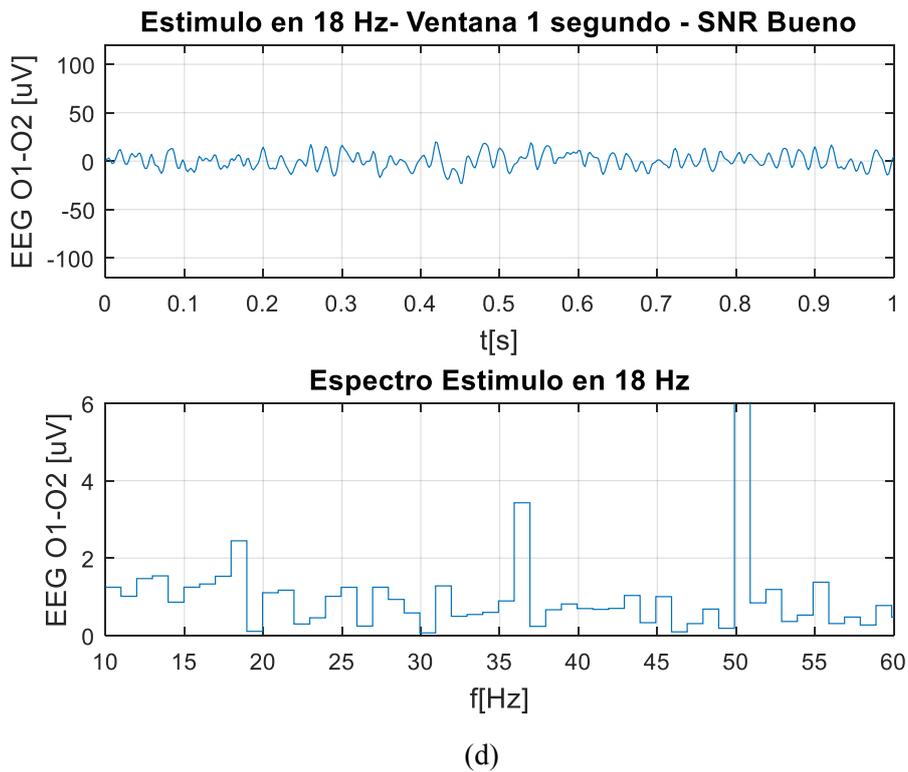
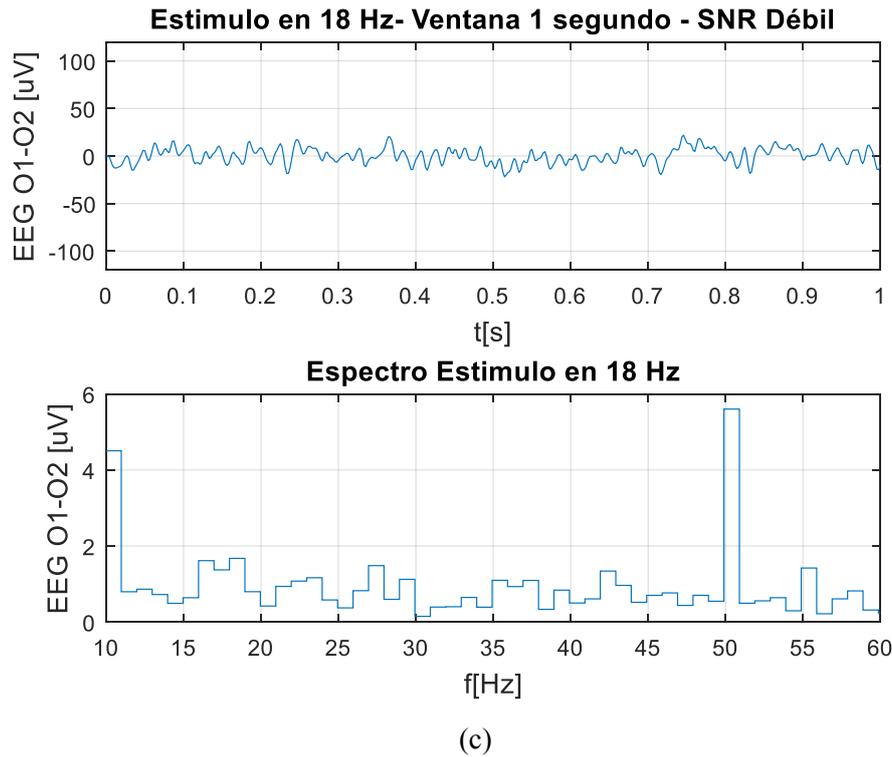


Figura 4.22. Recuperación de SSVEP con distintas ventanas.

Como se observa en la Figura 4.22, a medida que disminuye el tamaño de ventana resulta más difícil recuperar el SSVEP. En las partes (c) y (d) de la figura, se han procesado distintos segmentos del registro con ventanas de 1 segundo, estando presente el estímulo en 18 Hz para ambos segmentos. En la parte (c) resulta imposible recuperar el SSVEP, mientras que en la parte (d) es fácil de recuperar. En este sentido, se pretende experimentar para lograr la mayor cantidad de segmentos con buena SNR.

Relación entre tamaño de ventana y frecuencia de estímulo: en la Figura 4.23 se presenta una simulación usando una frecuencia de muestreo de 1024 Hz con ventanas de 0.5s (512 muestras). Se presentan distintas señales sinusoidales a distintas frecuencias: 14, 14.5 y 15 Hz con su correspondiente TDF (calculada usando el algoritmo FFT). Como se puede observar en la Figura, la señal de 14 Hz presenta toda su energía en un solo beam de la FFT, a diferencia de las otras señales (14.5 y 15 Hz) que presentan su energía distribuida en más de un beam de la FFT. La razón para esta diferencia es que el período de la señal de 14 Hz entra un número entero de veces en la ventana de análisis. Como conclusión, es más sencillo recuperar señales que verifican esta condición.

Se seleccionaron los estímulos:

- $f_1=14$ Hz (7 períodos por ventana)
- $f_2=16$ Hz (8 períodos por ventana)
- $f_3=18$ Hz (9 períodos por ventana)
- $f_4=20$ Hz (10 períodos por ventana)
- $f_5=22$ Hz (11 períodos por ventana)

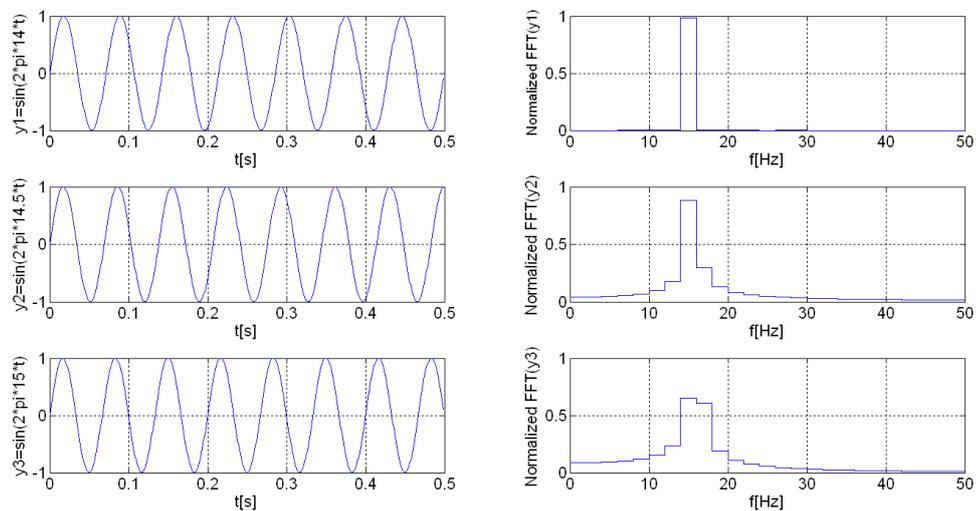


Figura 4.23. Relación entre tamaño de ventana y frecuencia de estímulo para optimizar la recuperación de potenciales evocados.

4.10.2. Conclusión.

Una estructura con un SE para adquisición y una computadora para procesamiento de señales resulta cómoda y útil para procesar señales off-line y para implementar ICCs completas. En particular, se han presentado dos ICCs basadas en ritmos cerebrales y algunos ensayos preliminares sobre potenciales evocados visuales de estado estacionario en ICC. Se han testeado dos tipos de estimuladores: uno usando un monitor del tipo CRT y otro basado en LEDs. Ambos tipos de estimuladores pueden ser utilizados en ICC, presentando potenciales evocados de mayor amplitud y una mayor cantidad de posibles frecuencias de excitación el estimulador basado en LEDs. Además, se han realizado medidas para verificar la resolución en frecuencia de este tipo de sistema, detectando estímulos separados en 0,2 Hz sin inconveniente. Se han realizado varias medidas que permitieron verificar la correcta generación y detección de los potenciales evocados en concordancia con lo reportado en la bibliografía y constituyen el punto de partida para la implementación de una ICC embebida prescindiendo del uso de una computadora que se presenta en el siguiente capítulo.

Capítulo 5

Interfaz Cerebro-Computadora autónoma basada en SSVEP

En tareas de investigación sobre ICC, realizar el procesamiento de las señales sobre una computadora es una alternativa apropiada, que permite ensayar distintas alternativas y paradigmas en forma ágil, modificar estrategias y depurar errores. Este tipo de dispositivo presenta algunos inconvenientes en una implementación final, como un tiempo de arranque considerable, un alto consumo y una baja confiabilidad: características heredadas de la PC que los soporta.

Cuando el desarrollo de la ICC está maduro, puede implementarse en un sistema embebido autónomo, que prescindiera del uso de una PC e integre la captura de las señales, su procesamiento y la interfaz de usuario (Figura 5.1).

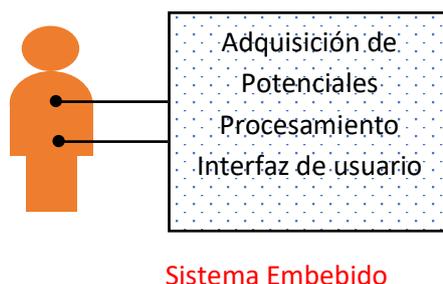


Figura 5.1. Esquema general de la tercera implementación.

Actualmente existen plataformas para sistemas embebidos con gran capacidad de memoria, de cómputo y conectividad a bajo costo. Así, por ejemplo, se puede observar la amplia difusión de sistemas embebidos como teléfonos inteligentes y tabletas. Este avance ha permitido que estos sistemas embebidos puedan resolver actividades de procesamiento que antes eran exclusivas de las computadoras y también que presenten interfaces de usuario más amigables basadas en ventanas. Estos avances han traído aparejada la necesidad de migrar desde sistemas operativos propietarios dedicados, a versiones de sistemas operativos más complejos que resuelvan gran parte del hardware asociado al procesador como interfaces gráficas y puertos de conexión. Existen variantes similares a los SO de propósito general para computadora adaptadas a los embebidos como Windows mobile o CE, Linux embebido o Android, que tienen amplia difusión.

A continuación, se presenta el desarrollo de un SE con este esquema que fue pensado para la adquisición y procesamiento de potenciales evocados. El trabajo completo fue dividido en varias etapas que luego de resueltas fueron permitiendo avanzar hacia el objetivo final: una interfaz cerebro computador embebida basada en potenciales evocados visuales de estado estacionario que implementa un “deletreador”.

5.1. Introducción.

Las restricciones de diseño iniciales sobre el hardware del SE a implementar fueron pensadas para lograr la mayor versatilidad posible en los distintos escenarios en los cuales el grupo de investigación trabaja normalmente. En este sentido, se consideró que un SE que permita digitalizar 8 canales de EEG diferenciales muestreando a 1000 mps en tiempo real cubre la mayoría de las aplicaciones biomédicas. Por otro lado, se pretende que el SE disponga de un procesador embebido que permita implementar técnicas de promediación coherente y análisis espectral en tiempo real para resolver el procesamiento, y que pueda portar un sistema operativo con prestaciones de tiempo real que resuelva la interfaz gráfica de usuario y sus puertos de conexión. En la Figura 5.2 se presenta un esquema general del SE con sus partes principales.

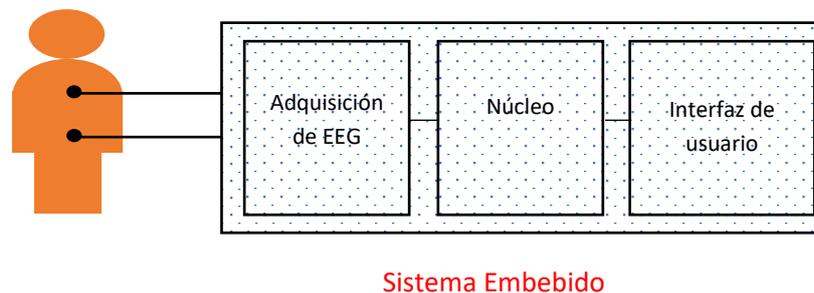


Figura 5.2. Esquema general con detalle en el sistema embebido.

5.2. Núcleo

La tarea de tiempo real prioritaria para el SE será la adquisición de hasta 8 canales diferenciales de EEG. Para la mayoría de las aplicaciones biomédicas frecuencias de muestreo del orden de 512 a 1024 muestras por segundo son suficientes [Fazel-Rezai et al., 2007]. Se seleccionó como núcleo de SE un microprocesador con arquitectura ARM de 32 bits (Samsung S3C2440) con velocidad de procesamiento de 400 MHz y posibilidad de portar SO. El mismo integra parte de la placa de desarrollo Micro2440 disponible a bajo costo y con tamaño reducido. [Friendly Arm Project, <http://www.friendlyarm.net/products/micro2440>]. La selección de este tipo de procesador con capacidad para portar un SO simplifica de gran manera la implementación de la interfaz de usuario y las tareas de baja prioridad dado que las mismas se pueden resolver por medio de una aplicación de alto nivel, en algunos casos con programación concurrente, usando los recursos y abstracción que brinda el SO. Para las tareas de tiempo real se puede utilizar un procesador esclavo dedicado que asegure el determinismo o bien resolver esta tarea haciendo uso del procesador principal. Para esta segunda alternativa, es común tener que implementar un controlador de dispositivo (driver) que permita manejar la tarea de alta prioridad con determinismo.

5.3. Sistema operativo embebido

El SE presenta distintas alternativas para la implementación del SO. La primera de ellas es implementar un SO propietario como el presentado por [Schimpf, 2013]. Otra posibilidad es utilizar un SO de propósito general como Android, Linux o Windows CE. Este tipo de placas disponen normalmente de un paquete de soporte (BSP: board support package) por medio del cual

se dispone de imágenes de SO listas para cargar o bien el código abierto de los mismos para poder modificarlos agregando o sacando controladores de dispositivos y luego volviendo a compilar.

Las distribuciones de Linux y Android con soporte de controladores para el procesador seleccionado no brindan prestaciones de tiempo real. En Linux, el proceso a seguir para agregar prestaciones de tiempo real consiste en emparchar (del inglés patch) la distribución. Como se describe en [Barbalace et. al, 2008] existen extensiones libres de Linux como RTAI y Xenomai que se pueden portar para el hardware necesario agregando determinismo al SO.

Para el microprocesador seleccionado, la única alternativa de SO disponible con prestaciones de tiempo real duro es Windows CE, pero no se dispone de la información de latencias de interrupción para el conjunto a utilizar: SO Windows CE 6.0 + Micro2440. La primer parte del trabajo consistió en realizar la medición de estas latencias a fin de asegurar su posible utilización en el registro de señales de EEG.

Windows CE se encuentra entre los SO de propósitos generales para computadoras de escritorio y los pequeños SO de tiempo real clásicamente utilizados en los sistemas embebidos. En este sentido, captura las mejores prestaciones de cada uno de ellos resultando en una alternativa atrayente. Su Kernel es un sistema multitarea con desalojo basado en prioridad, que además incorpora numerosas funciones de sincronización, señalización y eventos. Este SO está disponible para varios procesadores de 32 bits del tipo CISC o RISC, como Intel i486, Intel Pentium, MIPS R4000, ARM, StrongARM, PowerPC e Hitachi SH-3 y 4. El SO es configurable por módulos, generando imágenes que van desde los 540 KBytes hasta los 40 MBytes. En su versión más básica (540KBytes) incluye únicamente lo indispensable para su funcionamiento, sin incluir ningún elemento del catálogo de Windows CE 6.0 ni soporte para pantalla. La versión más completa (40MBytes), ofrece una interfaz gráfica, así como una versión adaptada de “Office” con soporte para sincronización con Exchange, reproductor Windows Media Player, mensajería instantánea y soporte para VoIP.

Los desarrolladores de aplicaciones disponen del entorno de la interfaz de programación basada en las API de Microsoft Win32 además de la facilidad y versatilidad de los lenguajes de secuencias de comandos o scripts.

Windows CE también ofrece soporte de serie para servicios multimedia, Internet, redes de área local, comunicaciones móviles y seguridad y provee la capacidad de implementar controladores para hardware propietario por medio de los “stream interface drivers”, que se describen más adelante.

Además de ser un sistema operativo de tiempo real nativo (sin parches), se eligió este SO por la familiaridad de los usuarios con el entorno visual de Windows basado en ventanas. Este es un conocimiento disponible en un gran universo de usuarios que facilita el manejo del dispositivo y reduce el tiempo de aprendizaje.

5.3.1. Latencias de Interrupción

El SO Windows CE divide en dos procesos el manejo de interrupciones: La rutina de servicio de interrupción (ISR: interrupt service routine) en modo Kernel, y el hilo de servicio de interrupción (IST: interrupt service thread) en modo usuario. Cada vez que ocurre una interrupción de un dispositivo particular, el kernel invoca a la ISR para determinar la fuente de interrupción haciendo un mapeo de identificación (SYSINT_ID: system interrupt identification). Luego de identificar la

fuente de interrupción el SO dispara el evento asociado al IST donde se implementa la tarea vinculada a la interrupción (Figura 5.3) [Pavlov y Belevsky, 2008].

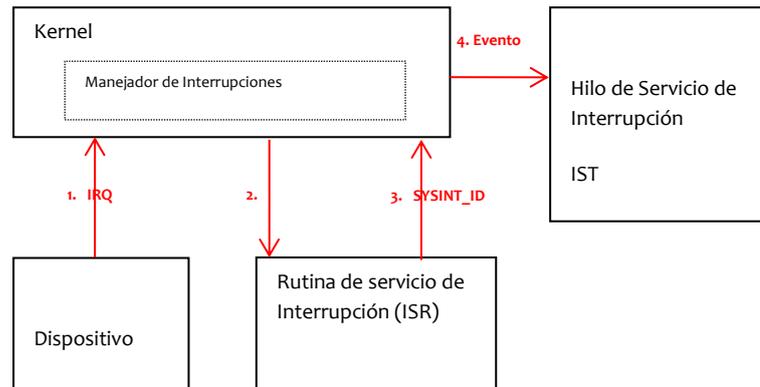


Figura 5.3. Manejo de Interrupciones

Para poder medir las latencias de interrupción tanto en la ISR como en la IST fue necesario implementar un controlador (driver) para un temporizador que permita generar marcas de tiempo medibles externamente. El SO presenta un modelo de controlador conocido como interfaz de flujo (stream interface driver) que permite resolver el problema.

5.3.2. Controlador: Stream interface driver

En la Figura 5.4 se presenta la arquitectura de un controlador del tipo “Stream Interface”. Este tipo de controladores puede ser configurado de tal forma que sea cargado por el administrador de dispositivos en tiempo de arranque. Como se puede observar en la Figura, las aplicaciones se comunican con el administrador de dispositivos y el hardware haciendo uso del stream interface driver por medio de las API de archivo. Este tipo de controladores recibe comandos desde el administrador de dispositivos y las aplicaciones por medio de llamadas al sistema de archivos.

Para implementar el controlador se necesitan 10 funciones que conforman su interfaz y son su punto de entrada. Estas funciones son: XXX_Init, XXX_Deinit, XXX_Open, XXX_Close, XXX_IOControl, XXX_Read, XXX_Write, XXX_Seek, XXX_Powerup y XXX_Powerdown, donde XXX es el nombre del controlador.

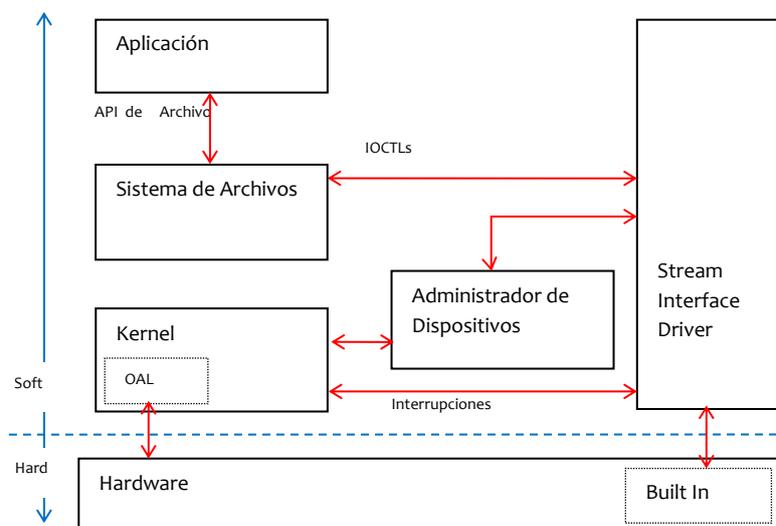


Figura 5.4. Arquitectura de un Stream Interface Driver

Para poder medir las latencias se implementó un controlador que maneja uno de los temporizadores libres del microprocesador. En este caso se utilizó el temporizador 0, dado que los restantes son utilizados por el scheduler y el profiler.

Como punto de arranque para la implementación del mismo se utilizó un proyecto de SO suministrado por el fabricante de la placa que incluye soporte para la misma (BSP). Los distintos pasos seguidos durante la implementación del controlador fueron:

- Se agregó el código necesario para implementar el controlador del temporizador 0 formado por las 10 funciones que forman su interfaz.
- Se generó un hilo dentro del controlador vinculado a la interrupción del temporizador por medio de un evento, de tal forma de poder incluir a futuro dentro del mismo la tarea de tiempo real.
- Se modificó el código de la ISR dentro del Kernel de tal forma que la misma sea capaz de manejar la interrupción generada por nuestro temporizador.
- Se agregó el controlador dentro del listado de dispositivos “*built-in*” de tal forma que el administrador de dispositivos dispare su inicio durante el arranque del sistema.
- Se activó el pin de salida vinculado al temporizador (T0) de tal forma de disponer en el exterior del instante de ejecución de la interrupción como punto de referencia.
- Dentro de la ISR se modificó el estado del pin GPB1 disponible en el exterior para poder medir su latencia. Este pin se fija en nivel alto al ingresar a la ISR.
- Dentro de la IST se modificó el estado del pin GPB1 para medir su latencia, pasando del estado “1” impuesto por la ISR al estado “0”.

5.3.3. Medidas

En la Figura 5.5 se presentan las marcas generadas por el controlador que permiten medir desde el exterior las latencias de la ISR y la IST a través de los pines T0 y GPB1.

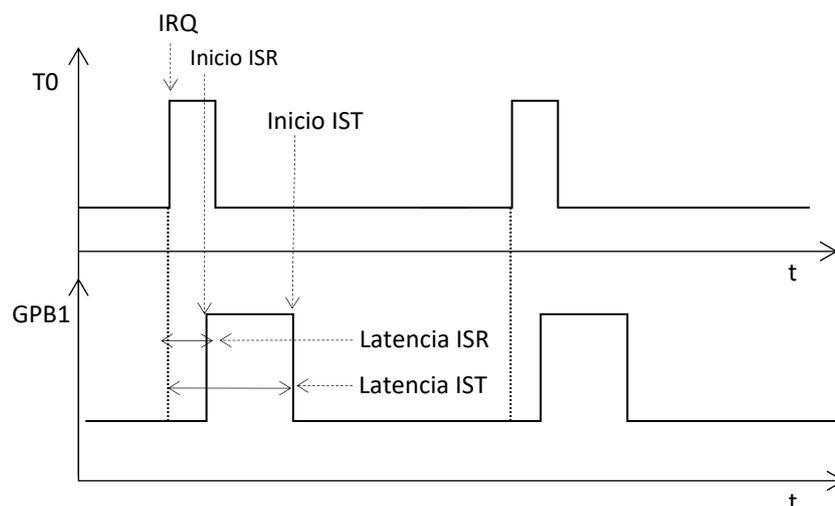


Figura 5.5. Latencias de ISR e IST.

En la Figura 5.6 se muestra el setup de medida implementado para la determinación de las latencias. En esta figura se puede observar la placa Micro2440SDK (derecha abajo), un osciloscopio digital (derecha arriba), un contador universal Modelo HP 53132A de 225 MHz (derecha centro) y la pantalla de una PC (izquierda). Para registrar las mediciones se configuró el contador para enviar las medidas realizadas a la PC por puerto serie.

Dado que las latencias a medir son variables y fuertemente dependientes de la carga de operaciones del sistema operativo se midieron las mismas con y sin carga durante períodos de tiempo de media hora.

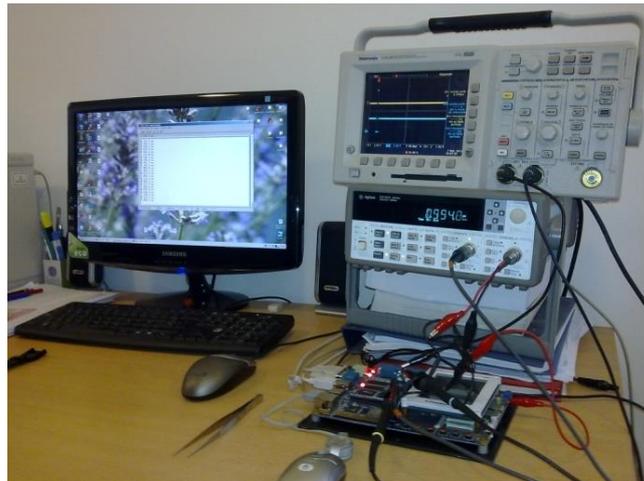


Figura 5.6. Setup de medida.

En las Figuras 5.7 y 5.8 se presentan los histogramas correspondientes a las medidas de latencia para la ISR y la IST. En cada una de las Figuras se puede observar un histograma sin carga de operaciones (izquierda) y otro con carga (derecha).

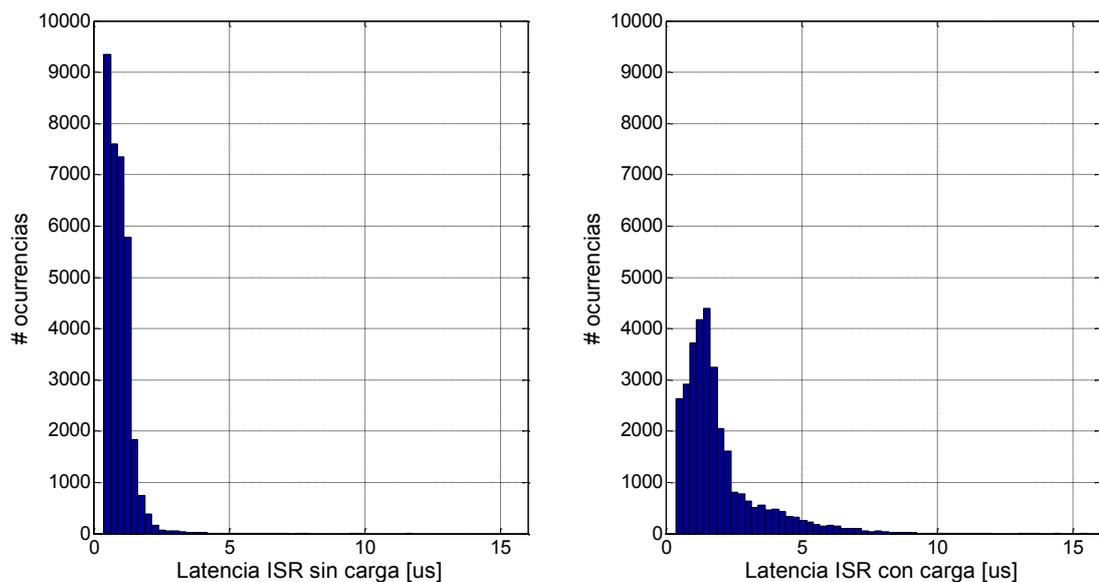


Figura 5.7. Latencia ISR.

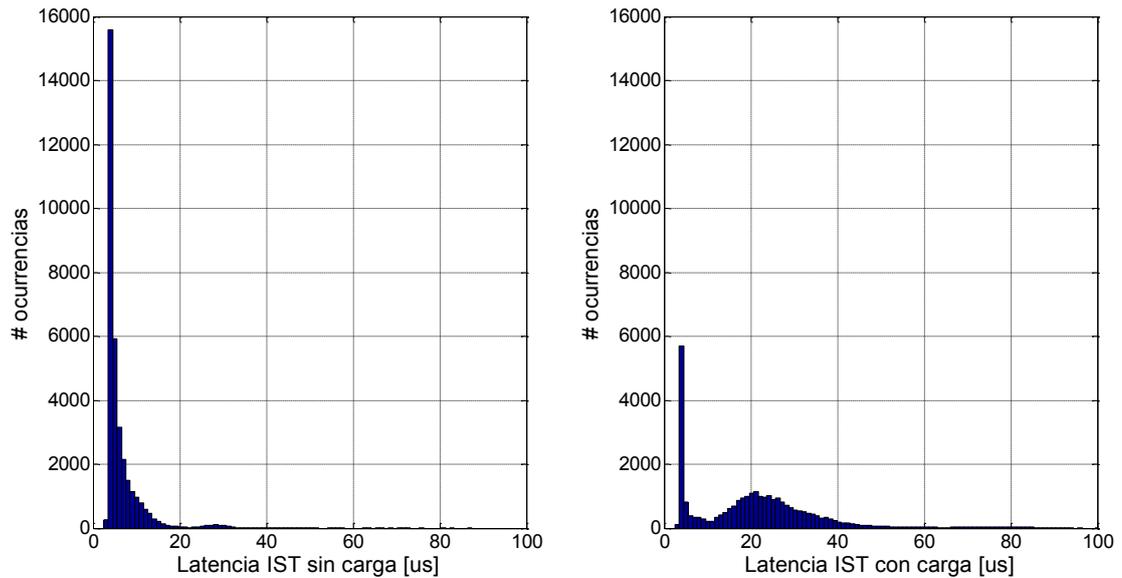


Figura 5.8. Latencia IST.

Para las mediciones “con carga” se registraron las latencias mientras el SO realizaba copias de directorios de gran tamaño desde un pendrive hacia una memoria del tipo SD en forma continua paralelamente con otras tareas de escritorio.

A partir de los histogramas, se puede observar que la latencia máxima para la ISR en estas condiciones de trabajo no supera los 15 μs y para la IST no supera los 100 μs . En la Tabla 5.1 se presenta la información con los valores máximo, mínimo y medio para cada una de las medidas realizadas.

Tabla 5.1. Latencias.

Medición de Latencias			
<i>Condición de Medida</i>	<i>Valor Mínimo</i>	<i>Valor Medio</i>	<i>Valor Máximo</i>
ISR sin carga.	0.5 μs	0.9 μs	11.6 μs
ISR con carga.	0.5 μs	1.9 μs	15.4 μs
IST sin carga.	3.4 μs	6.5 μs	87.2 μs
IST con carga.	3.4 μs	21.4 μs	100.1 μs

A partir de la latencia máxima medida (100 μs) para esta configuración de hardware y software se puede afirmar que el SE puede ser utilizado en una gran cantidad de aplicaciones de tiempo real. Los resultados obtenidos superan ampliamente las expectativas iniciales del trabajo en cuanto a las prestaciones del sistema operativo de tiempo real.

La implementación del controlador y su depuración hasta conseguir un correcto funcionamiento no es un proceso complejo, pero si complicado por la falta de información organizada.

Habiendo conseguido caracterizar el SE y capitalizado la metodología de implementación de controladores en este entorno se concluye que el conjunto puede ser utilizado en gran cantidad de aplicaciones de tiempo real. En particular, es de nuestro interés la digitalización y procesamiento de potenciales evocados visuales.

5.4. Primeras pruebas con potenciales evocados visuales

5.4.1. Adquisición de EEG

Para las primeras pruebas con potenciales evocados visuales se decidió utilizar un front-end basado en el circuito integrado ADS1298 de Texas Instruments. Este integrado presenta 8 canales de adquisición diferenciales con un conversor A/D de 24 bits del tipo Σ - Δ . Posee una interfaz SPI (Serial Peripheral Interface) para su configuración y extracción de muestras, por lo cual hubo que reacondicionar el driver desarrollado para poder capturar las muestras digitalizadas por el A/D atendiendo a una interrupción generada por el mismo. En esta etapa de desarrollo se decidió proveer alimentación y comunicación al front-end por medio de una barrera de aislación utilizando el aislador integrado ADUM6401 de Analog Devices [Guerrero, 2013].

El controlador opera en modo Kernel implementando un buffer circular de 1024 muestras por canal (24576 bytes correspondientes a 1024 muestras de 8 canales con 3 bytes/muestra).

A través del recurso de archivo mapeado en memoria se permite que distintas aplicaciones en modo usuario puedan acceder a la información generada por el controlador (Figura 5.9).

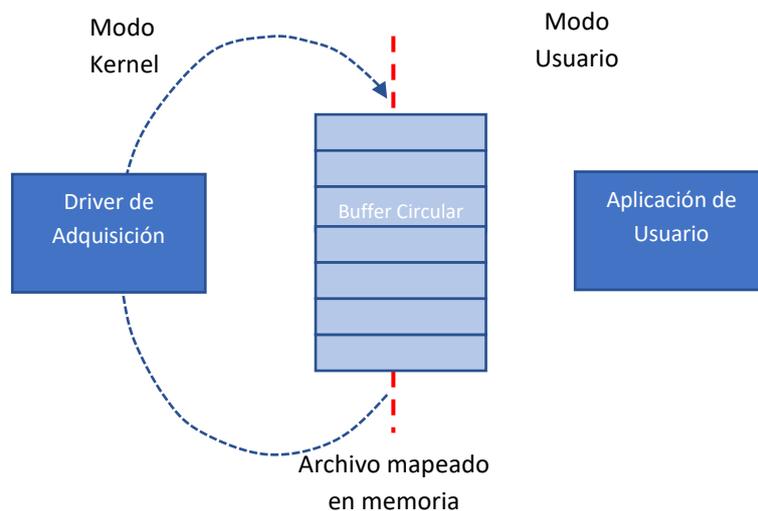


Figura 5.9. Manejo de los datos generados por el controlador.

5.4.2. Estimulador visual y aplicación de usuario

Como estimuladores visuales se utilizaron arreglos de LEDs en color blanco destellando a dos frecuencias distintas. Para mantener sincronismo entre el estímulo y la adquisición se utilizó la frecuencia de muestreo a 1 kHz como base de tiempo. Los estímulos utilizados están en $f_i=14.28$

Hz y $f_2=20$ Hz, ambas generadas directamente desde la frecuencia de muestreo dividiendo por 70 y 50 respectivamente.

Como aplicación de usuario se implementó un programa sencillo que accede a la memoria compartida, grafica la forma de onda de un canal diferencial y genera un archivo de texto para poder procesar la información digitalizada off-line haciendo uso de una computadora (Figura 5.10).



Figura 5.10. SSVEP, primeras pruebas.

5.4.3. Resultados experimentales con procesamiento off-line.

Como se observa en la Figura 5.10 se realizaron algunos ensayos con el objetivo de verificar la opción de implementar técnicas de promediación coherente para detectar potenciales evocados visuales. Se realizaron registros en archivo de potenciales en la posición Fz-Oz del sistema internacional 10-20 enfocando y desenfocando la vista en los estimuladores visuales. Se adjuntó a estos registros la información de los estimuladores por sincronismo.

5.4.4. Análisis espectral

En la Figura 5.11 se presenta el resultado del análisis espectral realizado off-line usando una computadora sobre la señal de EEG capturada en tres momentos distintos. La transformada discreta de Fourier (TDF), calculada utilizando la transformada rápida de Fourier (FFT), que se presenta en la parte (a) de la Figura corresponde a segmentos de 1 segundo de la señal con el usuario sin enfocar la vista en los estímulos visuales. Se pueden observar 10 segmentos de señal en color celeste y su promedio en negro. La FFT en (b) corresponde a la condición del usuario enfocando la vista en el estímulo destellando a 14.28 Hz. En este caso se puede observar el potencial evocado en su frecuencia fundamental con mayor amplitud en el primer armónico. En concordancia con lo reportado por [Wang, 2008] en algunas condiciones y sobre algunos usuarios, el potencial evocado en el primer armónico puede resultar de mayor amplitud que el fundamental. La FFT en la parte (c) de la figura corresponde al usuario enfocando la vista al estímulo en 20 Hz. En este caso, al igual que en el anterior, se puede distinguir el potencial evocado en la frecuencia fundamental y su primer armónico.

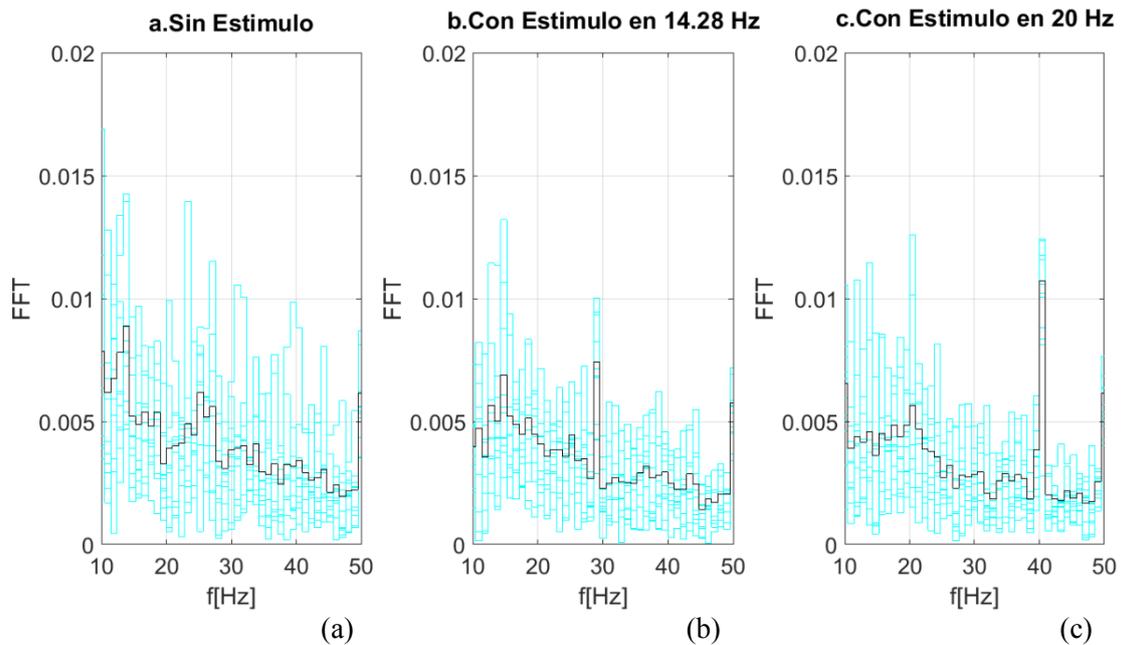


Figura 5.11. PE estimadores blancos (a) sin estímulo (b) estímulo en 14.28 Hz (c) estímulo en 20 Hz

5.4.5. Detección en el dominio del tiempo

Se decidió testear otro método de detección de potenciales evocados en el dominio del tiempo conocido como “stimulus locked inter-trace correlation” (SLIC) [Luo y Sullivan, 2012] [Manyakov, 2010]. Este método utiliza la información temporal de los estímulos para poder realizar promediación coherente en el dominio del tiempo entre señales sincronizadas con los estímulos.

La Figura 5.12 presenta este análisis temporal sobre la señal capturada en tres momentos distintos. El gráfico en la parte (a) corresponde a segmentos de señal sincronizados en tiempo con el estímulo de 14.28 Hz ($T=70$ ms) cuando el usuario no está enfocando la vista en ningún estímulo. En celeste se pueden observar los distintos tramos de señal y en negro el promedio de los mismos para el mismo intervalo de tiempo que en la FFT (10 segundos). El gráfico de la parte (b) corresponde al momento en el cual el usuario está enfocando la vista en el estímulo de 14.28 Hz. En este caso se puede distinguir el potencial evocado en sincronismo con el estímulo. Al igual que en el análisis en frecuencia, se observa claramente la primer armónica. El gráfico en la parte (c) corresponde al usuario enfocando la vista en el estímulo destellando a 20 Hz. En este caso no se puede distinguir el potencial evocado haciendo uso de la ventana de tiempo correspondiente al estímulo en 14.28 Hz.

La Figura 5.13 muestra el mismo escenario pero usando una ventana de tiempo sincrónica al estímulo en 20 Hz ($T=50$ ms). En este caso, es notorio el potencial evocado recuperado en la parte (c) de la Figura.

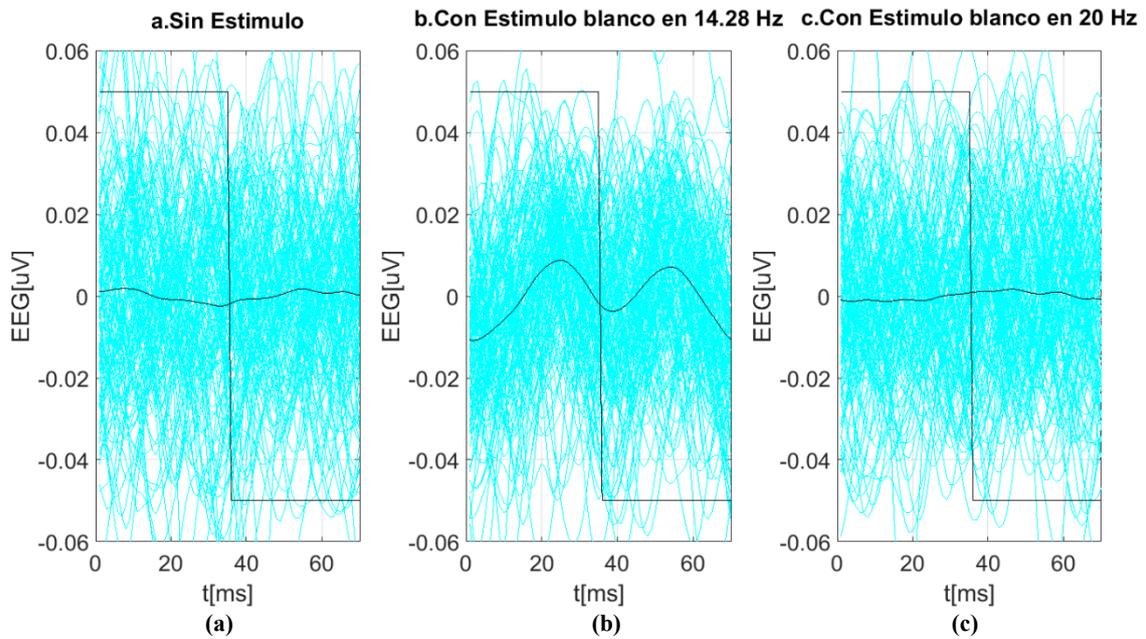


Figura 5.12. PE recuperado usando ventanas de 70 ms (a) sin estímulo (b) estímulo en 14.28 Hz (c) estímulo en 20 Hz

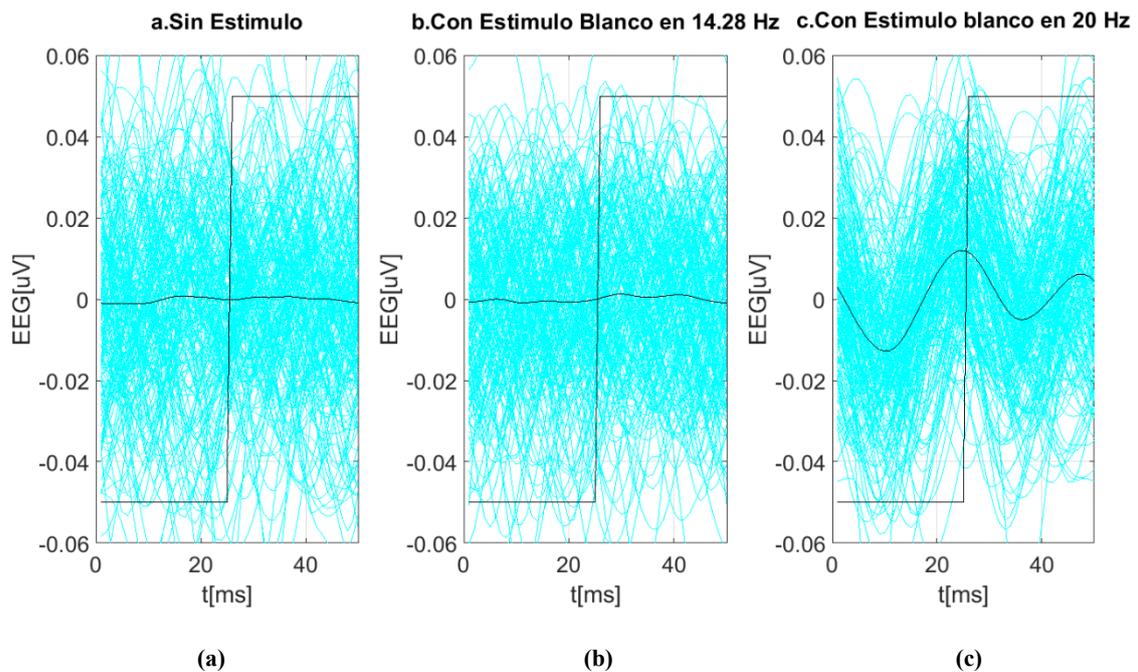


Figura 5.13. PE recuperado usando ventanas de 50 ms (a) sin estímulo (b) estímulo en 14.28 Hz (c) estímulo en 20 Hz

Estos resultados experimentales permiten afirmar que el sistema embebido implementado digitaliza y almacena de manera correcta la señal de EEG y que es posible recuperar los potenciales evocados visuales por medio de cualquiera de estas dos técnicas: TDF o SLIC.

En esta etapa de desarrollo se pretendía verificar la correcta operación del sistema embebido en las tareas de adquisición de datos, representación gráfica de los mismos y almacenamiento en archivo para su posterior utilización. De esta forma, las técnicas de recuperación de potenciales evocados se testearon off-line en computadora.

5.4.6. Puesta a punto del algoritmo de recuperación de potenciales evocados

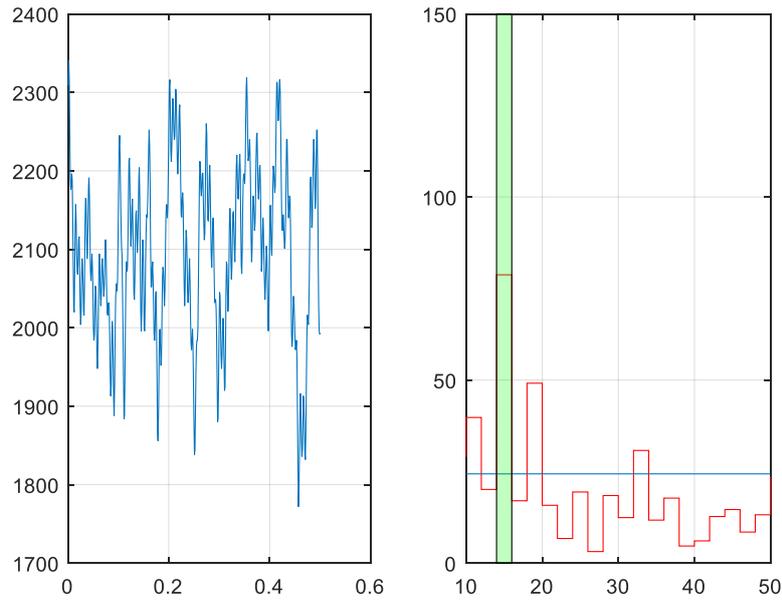
Con el objeto de poner a punto el proceso de recuperación de SSVEP se realizaron una serie de medidas utilizando 5 estimuladores visuales (Figura 5.14) y optimizando las condiciones de medida según el análisis descrito en el Capítulo 4, es decir,

Frecuencia de muestreo: $f_s=1024$ mps
Tamaño de ventana: $N= 512$ muestras (0,5 seg)
Posición de los electrodos: O_1-O_2 del sistema internacional 10-20
Frecuencia estímulo 1: $f_{est1}=14$ Hz
Frecuencia estímulo 2: $f_{est2}=16$ Hz
Frecuencia estímulo 3: $f_{est3}=18$ Hz
Frecuencia estímulo 4: $f_{est4}=20$ Hz
Frecuencia estímulo 5: $f_{est5}=22$ Hz

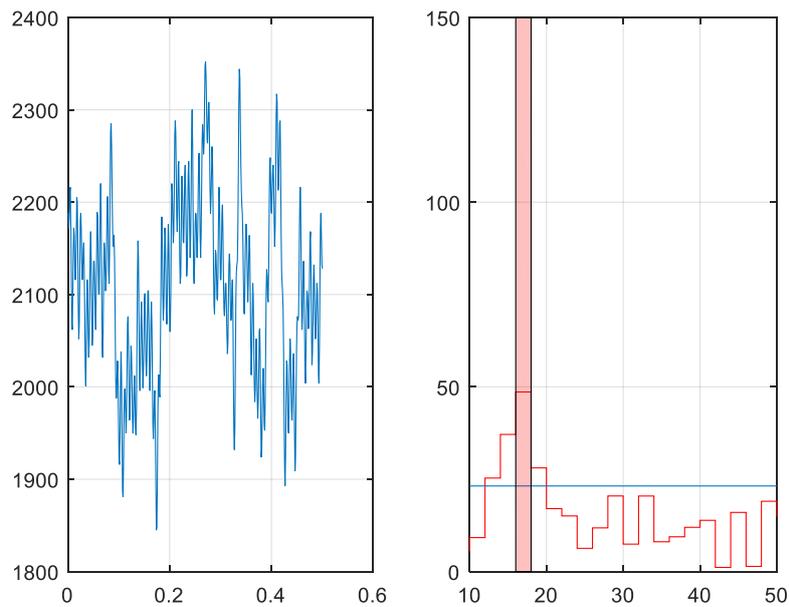


Figura 5.14. Configuración de estimulador visual ensayada.

Se realizaron 10 registros de señal (entre 30-60 segundos de duración c/u) enfocando la vista en los distintos estímulos a razón de 2 registros por estímulo. Con procesamiento off-line realizado en computadora se programó un script que parte en ventanas de 512 muestras cada registro y comparando estímulos entre sí y contra la media decide por uno de ellos como ganador. El programa acumula en 5 contadores los totales acumulados para cada estímulo. Como puede observarse en la Figura 5.15 (a), el script también muestra de forma gráfica resaltando con color verde la componente buscada, en caso de que haya resultado la de mayor amplitud. Caso contrario resalta en rojo la componente de mayor amplitud no buscada (Figura 5.15 (b)). En ambos espectros también se grafica en color azul la media calculada para esa ventana y el tramo de señal en el dominio del tiempo. En ambos dominios las señales se muestran en unidades arbitrarias.



(a)



(b)

Figura 5.15. Script de recuperación de SSVEP (a). Estimulo recuperado en 14 Hz correctamente. (b) Error. Ante estímulo en 14 Hz la banda con mayor potencia resulta la de 16Hz.

En la Tabla 5.2 se presenta el resultado del procesamiento para los 10 registros. La primera columna indica el nombre de archivo, la segunda columna el número de ventanas que contiene el registro y de las tercera a la séptima columna los totales de los contadores para cada estímulo. En la última columna se indica el porcentaje de ventanas en las que logró recuperarse el SSVEP esperado.

Tabla 5.2.

Test de recuperación de SSVEP Off-line							
Nombre de Archivo	# de Ventanas	C14	C16	C18	C20	C22	% éxito
Estimulo_en_14_a	59	43	3	2	4	1	72.8%
Estimulo_en_14_b	58	48	2	1	2	2	82.7%
Estimulo_en_16_a	93	9	59	5	3	5	63.4%
Estimulo_en_16_b	77	5	62	0	3	0	80.5%
Estimulo_en_18_a	97	17	16	48	5	3	49.5%
Estimulo_en_18_b	100	14	10	56	5	1	56%
Estimulo_en_20_a	89	18	12	8	22	15	24.7%
Estimulo_en_20_b	85	5	9	7	45	4	52.9%
Estimulo_en_22_a	100	12	19	14	9	31	31%
Estimulo_en_22_b	100	12	12	6	9	48	48%

En función de estas medidas se pueden extraer algunas conclusiones:

- La distribución espacial de los estimuladores propuesta no presenta inconvenientes para la correcta recuperación de los SSVEP, es decir, aunque el resto de los estimuladores visuales están presentes en el campo de visión del usuario, el mismo puede filtrarlas enfocando la vista en el estímulo deseado.
- Resulta notorio el número de ventanas en las cuales puede recuperarse el SSVEP deseado.
- En los estímulos más altos (20 y 22 Hz) existe un menor porcentaje de éxito en concordancia con la respuesta en frecuencia que se reporta en la bibliografía para los SSVEP.

El siguiente paso en el desarrollo fue migrar este procesamiento al SE para lograr un dispositivo autónomo que prescinde del uso de computadora. La implementación práctica sobre la cual se decidió avanzar es una ICC embebida basada en potenciales evocados visuales de estado estacionario y el ritmo alfa. En particular se implementó un deletreador (speller).

5.5. Implementación práctica de ICC: Un deletreador embebido híbrido

5.5.1. Introducción

En los últimos años varios grupos de investigación han comenzado a explorar una nueva variante de ICC, las híbridas. En este caso se combinan múltiples modalidades en una única ICC para mejorar la performance de la misma [Pfurtscheller et. Al., 2010].

Aun cuando las ICCs basadas en SSVEP son las que presentan mejores prestaciones, no pueden ser apagadas o prendidas de manera voluntaria por el usuario debido a su dependencia exclusiva en los potenciales evocados. En la implementación híbrida propuesta se pretende utilizar potenciales espontáneos para permitir al usuario encender a apagar los estimuladores visuales evitando fatiga o falsos comandos.

Se pretende implementar una ICC simple, robusta y de inicio rápido. Debe operar fuera del laboratorio usando la menor cantidad de electrodos posible. Trabaja a demanda utilizando el ritmo alfa visual para ejecutar comandos de control. Por ejemplo, el usuario de la ICC puede prender o apagar los estimuladores visuales por medio de activaciones generadas con el ritmo alfa codificado en tiempo.

Como se menciona en [Huggins et. al, 2014] y en [Millán et. al, 2010] las ICC han demostrado ser útiles y es tiempo de migrar el desarrollo desde prototipo de laboratorio a un dispositivo embebido portable de tamaño reducido. Para poder acercar la tecnología ICC al usuario final es necesario reemplazar la computadora por un dispositivo más compacto y confiable. Se propone el uso de un SE con una interfaz de usuario amigable y familiar para el usuario con rápido tiempo de arranque. Cualquier ICC o dispositivo de asistencia debe estar operacional tan rápido como sea posible y no debe necesitar calibración para ser útil. Un tiempo de arranque lento puede inclinar al usuario a optar por otro canal de comunicación.

5.5.2. Sistema embebido

La ICC propuesta es un SE en un único dispositivo. Implementa un deletreador que permite seleccionar entre 25 caracteres usando 5 estímulos visuales (en la configuración presentada en 5.4.3) y el ritmo alfa para instrucciones de control. Puede ser operado por el usuario de manera autónoma sin asistencia externa o supervisión.

5.5.2.1. Núcleo

Para implementar el SE se reutilizó la placa principal previamente descrita (Micro2440) portando el SO Windows CE 6 y los controladores previamente desarrollados.

5.5.2.2. Adquisición de EEG

Para la adquisición de biopotenciales se decidió implementar un nuevo amplificador de biopotenciales analógico dedicado para un canal diferencial (Figura 5.16), prescindiendo del uso del convertidor sigma-delta de 8 canales y 24 bits.

Es un amplificador de bajo ruido acoplado en alterna, que se alimenta desde una fuente simple de 5V. El esquema se completa con un circuito “Driven Right Leg” (DRL) que fija al paciente a un potencial de continua VREF de 1.25 V respecto de la masa del amplificador según [Spinelli et. al, 2001]. Para el DRL se utilizaron dos electrodos independientes que se colocaron en la frente, evitando así problemas de fijación por el pelo. En este primer prototipo se utilizó un sólo canal de adquisición que presenta una ganancia de 5832 veces dividida en dos etapas de 5.25 y 1111. El amplificador se montó en una pequeña caja plástica que puede fijarse a la ropa del usuario mediante un broche (Figura 6.17). Para digitalizar las muestras e imponer la tasa de muestreo se utilizó un microcontrolador ADuc812 de la empresa Analog Devices (procesador esclavo dedicado). La información digitalizada se transfiere a la placa principal por medio de un puerto SPI aislado por un convertidor DC/DC modelo ADuM6401, de la misma empresa.

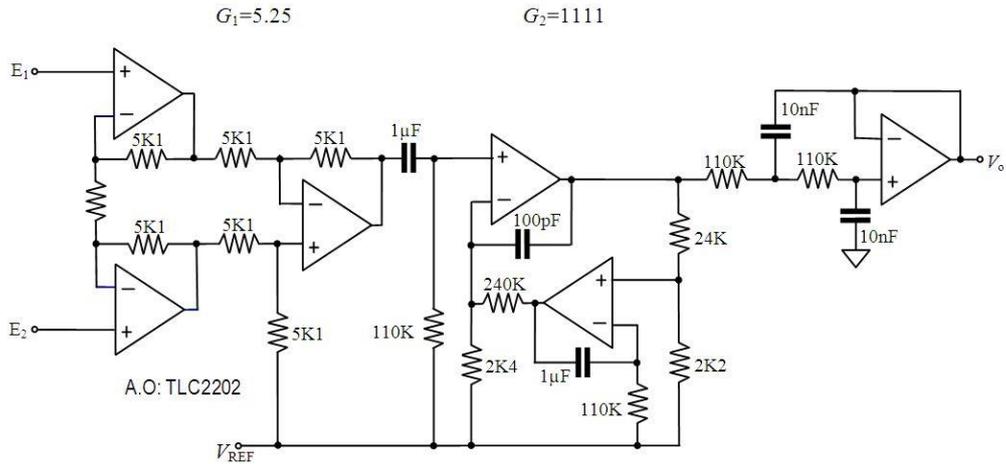


Figura 5.16. Esquemático del Amplificador de EEG.



Figura 5.17. Amplificador de EEG implementado.

El conversor A/D impone la base de tiempo del sistema por medio de su frecuencia de muestreo (1024 mps). El microcontrolador que contiene el A/D notifica a la placa principal el evento de “dato listo” por medio de una interrupción externa que también está aislada por el ADuM.

5.5.3. Software de alto nivel

Por medio del recurso de archivos mapeados en memoria, cualquier aplicación de alto nivel que corre en modo usuario puede acceder a la información de EEG generada por el driver en modo kernel. Las aplicaciones de alto nivel se pueden desarrollar bajo el paradigma de la programación orientada a objetos. Se seleccionó el lenguaje C# para programar la aplicación visual que implementa el deletreador.

Como herramientas previas al desarrollo de la aplicación se realizaron distintas clases propias, siendo las de mayor interés la clase “graficadora” y la “FFT”. La primera realiza la gráfica en tiempo real de la señal digitalizada por un determinado canal de adquisición y la segunda presenta el módulo del espectro calculado mediante la transformada rápida de Fourier para un tamaño de ventana de 512 muestras (0.5 s) (Figura 5.18). El uso de estas herramientas fue de gran utilidad durante las primeras pruebas para optimizar la forma de estimular visualmente y recuperar los potenciales de manera óptima.

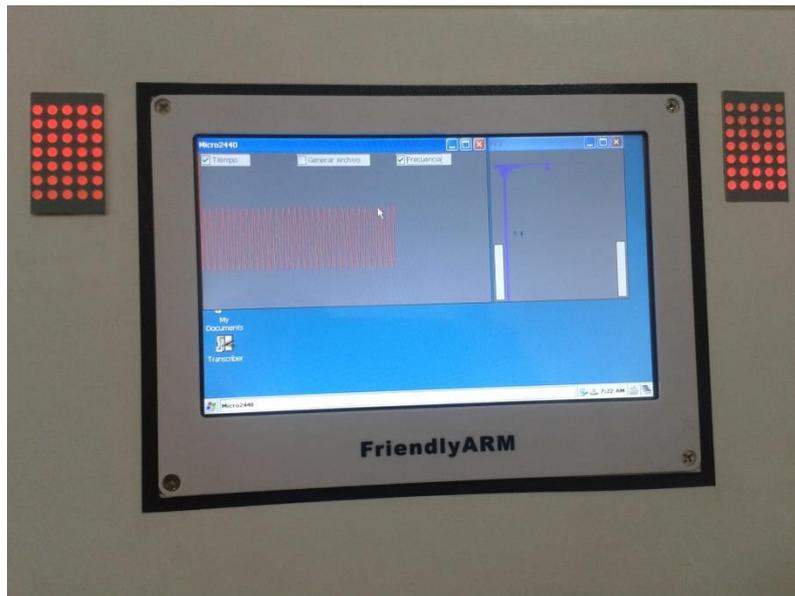
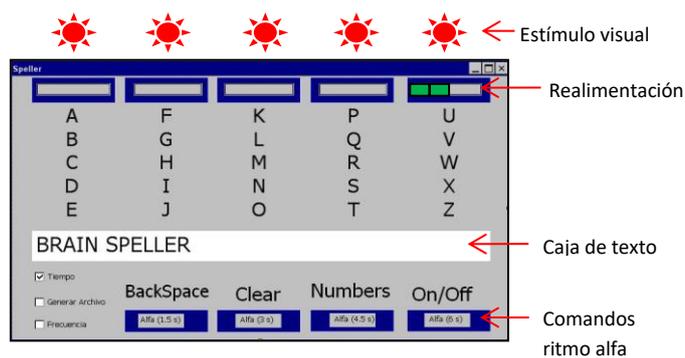


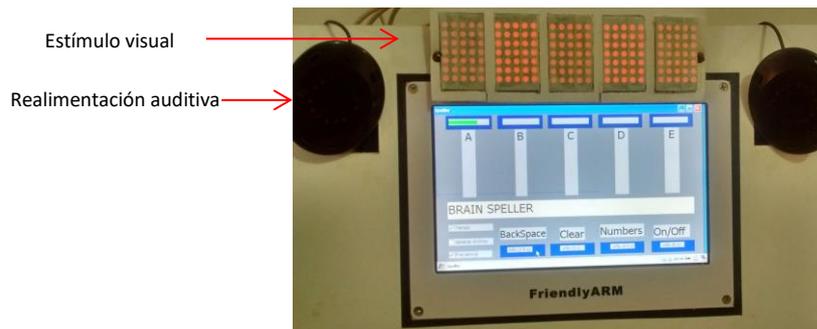
Figura 5.18. Herramientas para representación en dominio del tiempo y frecuencia.

En la Figura 5.19 se presenta la interfaz gráfica de usuario del deletreador. La ventana principal presenta 25 caracteres agrupados en 5 columnas con 5 letras en cada una. Cada uno de los 5 estimuladores visuales está en la parte superior de cada columna. Los estimuladores se implementaron con arreglos de 7x5 led (Figura 5.19 (b)).

Para seleccionar una letra el usuario debe enfocar la vista en el estímulo que se encuentra en la parte superior de la columna donde está la letra deseada. La ICC detectará el potencial evocado y redibujará las 5 letras de la columna seleccionada pero ahora cada letra en una columna (ver Figura 5.19 (b): luego de haber seleccionado la primera columna las 5 letras están cada una en una columna). Finalmente, el usuario debe enfocar la vista en el estímulo visual correspondiente a la letra objetivo.



(a)



(b)

Figura 5.19. GUI. (a) Ventana principal del deletreador con estímulos visuales sobre cada columna de letras. (b) Fotografía del deletreador mostrando la segunda pantalla luego de haber seleccionado la primera columna.

Por medio de la transformada rápida de Fourier el algoritmo de la ICC detecta la frecuencia correspondiente al estímulo en el cual el usuario está enfocando la vista. Una vez que la ICC detecta un estímulo en particular activa uno de tres segmentos en una barra de progreso que representa la realimentación visual para el usuario. El comando se dispara luego de tres segmentos consecutivos de detección para un mismo estímulo. Es de destacar la utilidad de esta realimentación visual. En tiempo de operación resulta muy ventajoso contar con una realimentación en tiempo real para reforzar o cambiar la forma de generar el potencial evocado correcto.

En la parte inferior de la ventana se pueden observar los comandos: “backspace”, “clear”, “numbers” y “On/Off”. Estos comandos se generan por medio del ritmo alfa visual. Los cuatro comandos se diferencian por la duración en tiempo del ritmo generado:

- Backspace: 1.5s.
- Clear: 3s.
- Numbers: 4.5s.
- On/Off: 6s.

Cada uno de estos estímulos posee una realimentación auditiva para notificar al usuario su selección. Por ejemplo, un segundo y medio después de haber cerrado los ojos el usuario recibe la notificación sonora del backspace. Si este es el comando que el usuario pretende ejecutar abre los ojos finalizando el comando. Caso contrario, mantiene los ojos cerrados hasta escuchar la realimentación sonora correspondiente al comando que pretende ejecutar.

El comando más importante permite al usuario encender o apagar los estimuladores visuales. El comando debe ser robusto para que no ocurran activaciones indeseadas. Para esto el usuario debe generar un ritmo alfa de al menos 6 segundos.

El comando “numbers” permite visualizar un segundo teclado con números y caracteres inusuales. Se mantiene el mismo esquema con 25 caracteres.

5.5.4. Algoritmo: Densidad Espectral de Potencia

Para calcular la densidad espectral de potencia se utiliza la clase de desarrollo propio “FFT” que retorna una transformada rápida de Fourier de 512 puntos. El algoritmo implementado calcula en tiempo real la FFT de 512 puntos cada 0,5 s. Para detectar la intención del usuario el algoritmo busca la banda de frecuencia con mayor energía comparando contra el resto de los estímulos y contra un umbral que representa el valor medio del espectro. La decisión del algoritmo puede inclinarse por uno de los 5 estímulos o ninguno de ellos en caso de que el usuario no esté enfocando la visión en ningún estímulo. Como ya se ha mencionado, se necesitan tres decisiones consecutivas sobre el mismo estímulo para disparar el comando.

5.5.5. Experimentos

Para verificar y caracterizar el deletreador se realizaron varios experimentos usando los electrodos ubicados en la posición O₁-O₂ del sistema internacional 10-20. Se usaron dos electrodos autoadhesivos en la frente para el circuito DRL. La señal de EEG se adquiere a 1024Hz luego de ser filtrada entre 1 y 100 Hz por el front-end (Figura 5.20).

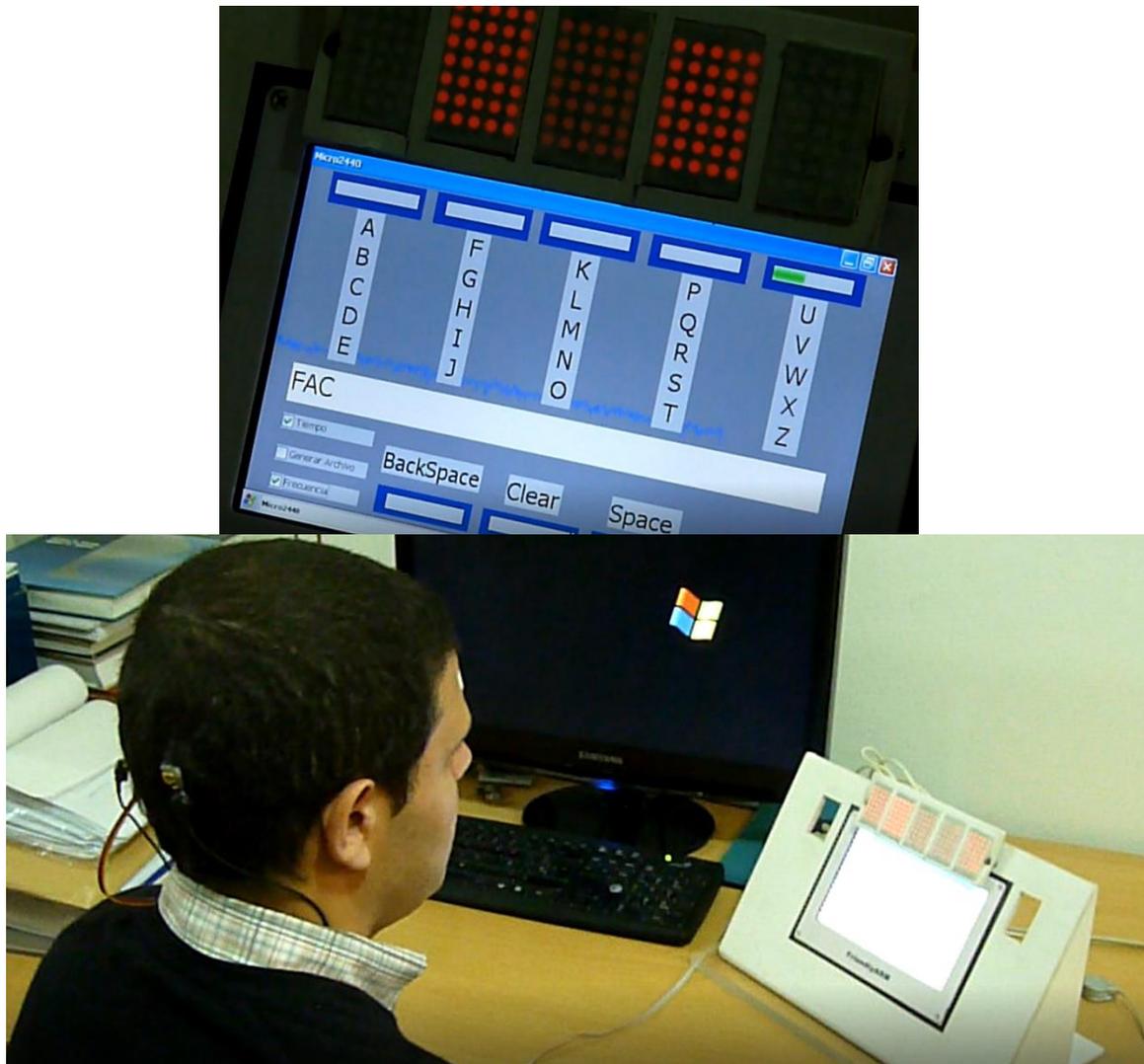


Figura 5.20. Ensayos sobre el deletreador embebido.

Se realizaron medidas sobre dos sujetos para verificar el sistema. El experimento se inicia escribiendo palabras sueltas de manera similar a [Cecotti, 2010]. El usuario debe escribir ocho palabras: cinco palabras fijas y tres variables. Las palabras fijas son: BCI, BRAIN, SPELLER, INTERFACE y COMPUTER. Las palabras variables son: nombre, apellido y equipo de futbol. Cada palabra debe ser escrita de manera correcta sin errores. El usuario puede hacer correcciones por medio del comando alfa “backspace”. El usuario puede optar por descansar entre palabras apagando los estimuladores visuales si así lo desea.

5.5.6. Resultados

El parámetro que se utiliza para medir y contextualizar a las ICCs es la tasa de transferencia de información (ITR: Information transfer rate) [Wolpaw et. Al. 2002][Gao et. Al., 2003] definida como:

$$ITR = \frac{\text{Numero de comandos} * B}{T} \quad (1)$$

Donde B es la capacidad del canal, dada por:

$$B = \log_2 N + P \log_2 P + (1 - P) \log_2 \left(\frac{1-P}{N-1} \right) \quad (2)$$

P es la probabilidad de detectar un comando correctamente, N es el número de comandos y T es el tiempo requerido para producir un cierto número de comandos. En esta implementación con N=5 el mejor B obtenible es de 2.32 bits/comando. Con cualquier otro P<1 la capacidad del canal decrece. Idealmente, en la mejor situación, el usuario podrá ejecutar un comando en 1.5 s y la ITR correspondiente con P=1 será de 92.87 bpm. En este escenario, la velocidad del deletreador sería de 20 letras/minuto (2 comandos por letra).

Las tablas 5.3 y 5.4 presentan los resultados medidos para los sujetos 1 y 2 respectivamente.

Tabla 5.3. Exactitud y tasa de bits para Sujeto 1

Palabra	#Comandos	Tiempo[s]	Exactitud [%]	ITR[bpm]	Letras/min
BCI	6	20	100	41,76	9
BRAIN	10	34	100	40,94	8,82
SPELLER	14	59	100	33,03	7,12
INTERFACE	18	90	100	27,84	6
COMPUTER	16	85	100	26,20	5,64
PABLO	10	45	100	30,93	6,67
GARCIA	12	53	100	31,52	6,79
BOCA	8	40	100	27,84	6

Tabla 5.4. Exactitud y tasa de bits para Sujeto 2

Palabra	#Comandos	Tiempo[s]	Exactitud[%]	ITR[bpm]	Letras/min
BCI	6	28	100	29,83	6,43
BRAIN	10	36	100	38,67	8,33
SPELLER	15	134	93,3	12,22	3,35
INTERFACE	18	125	100	20,04	4,32
COMPUTER	16	185	100	12,04	2,59
ENRIQUE	14	120	100	16,24	3,50
SPINELLI	16	105	100	21,21	4,57
PINCHA	12	64	100	26,10	5,62

5.5.7. Conclusión

Se ha logrado implementar un deletreador híbrido embebido que brinda un posible canal de comunicación a personas discapacitadas. El sistema es robusto, simple de usar y portable. Prescinde del uso de una computadora y presenta un rápido tiempo de encendido (~15s). Se han utilizado arreglos de LED en color rojo con un film difusor para generar los estímulos visuales. Se ha implementado el concepto de “brain switch” usando ritmo alfa que permite al usuario encender o apagar los estímulos visuales. Por otra parte, el sistema implementa una realimentación visual por medio de una barra de progreso que resulta muy útil para mejorar la recuperación de potenciales evocados; y una realimentación auditiva para los comandos alfa.

Los resultados experimentales han validado la correcta operación del sistema y la posibilidad de transferirlo a su usuario. Aunque las pocas pruebas realizadas muestran una buena performance, los resultados experimentales sugieren que el entrenamiento puede mejorar notablemente la velocidad del deletreador. Con muy poco entrenamiento las velocidades de escritura promedio para los dos sujetos son de 7 y 4.3 letras/minuto.

Finalmente cabe destacar que se ha logrado la implementación completa del deletreador embebido, siendo el sistema completamente funcional y representando un buen precedente para la implementación requerida por un usuario final. El hardware puede ser modificado fácilmente para implementar otros dispositivos de asistencia como controlador de silla de ruedas o controlador de prótesis, entre otros.

Capítulo 6

Sistema Embebido para evaluación de la capacidad auditiva basado en Emisiones Otoacústicas (OAE).

La función del oído es percibir sonidos, pero en 1978 Kemp [Kemp, 1978] observó que un oído sano tiene además capacidad de producirlos. Estos débiles sonidos, que se denominan “Emisiones Oto-Acústicas (del inglés OAE: Otoacoustic Emission), que deben captarse en el canal auditivo, permiten realizar un diagnóstico objetivo del sistema auditivo. El esquema general del dispositivo necesario para el registro de OAE es similar al SE descrito en el Capítulo 5: requiere la generación de estímulos (en este caso auditivos) y registrar las OAE en forma sincronizada. Tratándose de señales de audio, las OAE requieren tasas de muestreo de decenas de kHz.

Este trabajo ha sido desarrollado en el ámbito de la Unidad de Investigación y Transferencia Gitec perteneciente a la Facultad de Ingeniería de la UNLP. Es un desarrollo que ha sido transferido al medio y surgió a requerimiento de una empresa del ámbito local que desarrolla y produce tecnología médica para audiología desde hace más de 30 años (LADIE: Laboratorio de Aplicación y Desarrollo de Instrumental Electrónico). El principal objetivo es la sustitución de equipamiento importado (de origen europeo o norteamericano), que trae aparejado un muy alto costo de inversión inicial y mantenimiento, como así también demoras en las reparaciones.

Por cuestiones de confidencialidad, en este capítulo no se describen detalles que permitan la reproducción del equipo. El contenido de este capítulo se centra en las estrategias de diseño adoptadas para el sistema embebido, objeto de esta tesis, tomando el desarrollo de este dispositivo como un ejemplo de especificaciones de un SE. Además, este proyecto ha sido aprobado por el FONTAR (Fondo Tecnológico Argentino -Ministerio de Ciencia, Tecnología e Innovación Productiva - Agencia Nacional de promoción Científica y Tecnológica).

La utilización de OAE es relativamente reciente y su campo de aplicación se concentra principalmente en neonatos para la detección temprana de hipoacusias. En nuestro país, a partir de la promulgación en el año 2001 de la ley 25415 del Programa Nacional de detección de Hipoacusia se ha incluido dentro del plan de estudios a realizar a los recién nacidos.

En este capítulo se describen los principios básicos de las OAE y el desarrollo de un equipo para evaluar la capacidad auditiva basado en emisiones otoacústicas. Estos equipos permiten evaluar la capacidad auditiva de un sujeto en forma objetiva, es decir, sin requerir respuestas por parte del paciente. Estas técnicas permiten detectar problemas de audición en pocos minutos con sólo introducir una sonda en el canal auditivo.

6.1. Breve descripción del Sistema Auditivo.

El sistema auditivo percibe los sonidos producidos en el exterior y los transforma en impulsos eléctricos que son enviados al cerebro a través del nervio auditivo (Figura 6.1). Este complejo proceso se desarrolla en varias etapas. La presión sonora ingresa por el canal auditivo hasta llegar al tímpano, donde se transforma en una vibración mecánica. Esta vibración se transmite a través de los osículos (martillo, yunque y estribo) hacia la cóclea o caracol. Es en la cóclea donde las

vibraciones se convierten en impulsos eléctricos que llegarán al cerebro a través del nervio auditivo.

A partir de las vibraciones impuestas por el estribo en la ventana oval, se genera una onda viajera en el líquido que contiene la cóclea. Dentro de ésta se encuentra la membrana basilar donde se distribuyen células ciliadas. El movimiento de la membrana basilar se transfiere a estas células generando los impulsos eléctricos que serán transmitidos al cerebro. La membrana basilar es una estructura cuyo espesor y rigidez no es constante: cerca de la ventana oval, la membrana es delgada y rígida, pero a medida que se acerca hacia el extremo de la cóclea (helicotrema) se vuelve más gruesa y flexible [Flanagan, 1960]. De este modo, las altas frecuencias excitarán fundamentalmente la primera parte, cercana a la ventana oval, mientras que los efectos de las bajas frecuencias se concentrarán hacia el extremo de la cóclea. Como cada zona de la cóclea está asociada a una banda de frecuencias en particular, se suele decir que este órgano oficia de “analyzer de espectro”. Los desplazamientos de las células ciliadas correspondientes a cada zona son codificados como impulsos eléctricos y transmitidos al cerebro que recibe esta información a través del nervio auditivo.

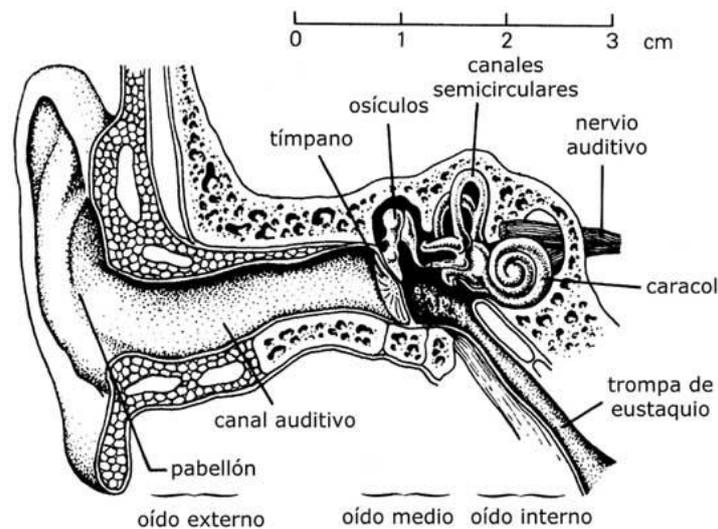


Figura 6.1. Corte del Sistema Auditivo

6.2. Otoemisiones acústicas.

En 1978 Kemp observó experimentalmente un fenómeno en cierta manera recíproco al anterior: las células ciliadas de un oído sano presentan también la capacidad de moverse espontáneamente y de generar sonidos factibles de ser captados en el canal auditivo [Kemp,1978]. Es decir, las células ciliadas se mueven espontáneamente. Este movimiento se transmite al líquido coclear y por medio de éste, a los oscículos que imprimen una vibración en el tímpano. Finalmente, este último produce una presión acústica en el conducto auditivo, que puede captarse utilizando un micrófono. Este fenómeno recibe el nombre de emisión otoacústica.

Las OAEs son un fenómeno natural y su observación fue el punto de partida de diversas técnicas de diagnóstico que permiten determinar la capacidad auditiva de una persona. A partir de este descubrimiento, varias firmas audiológicas tal como Grason-Stadler y OtoDynamics (dirigida por el Dr. Kemp) comenzaron a producir instrumentos basados en OAE.

6.3. Determinación de la capacidad auditiva mediante OAE.

Existen varias formas de poner de manifiesto este tipo de emisiones, las cuales suelen asociarse a distintos tipos de OAE. Las más utilizadas son las emisiones otoacústicas espontáneas (SOAE: del inglés Spontaneous Otoacoustic Emissions), las emisiones otoacústicas evocadas transitorias (TEOAE: del inglés Transient Evoked Otoacoustic Emissions) y las emisiones otoacústicas por producto de distorsión (DPOAE: del inglés Distortion Product Otoacoustic Emission). Cada uno de los tipos de OAE da origen a distintas técnicas de medida y de diagnóstico.

Para captar las OAE se introduce en el canal auditivo una sonda que se instala en forma hermética. Esta sonda posee un conducto que se conecta a un micrófono que capta las OAE y uno o dos conductos adicionales que se utilizan para inyectar estímulos auditivos.

El nivel de presión sonora, conocido normalmente por sus siglas en inglés, SPL (Sound Pressure Level) se mide en decibelios (dB) y varía entre 0 dB que es el umbral de audición y 120 dB que es el umbral del dolor. Debido al amplio margen que existe entre la sonoridad más intensa (200 Pa) y la más débil (20uPa) se utiliza el dB en escala logarítmica referido a 20uPa en lugar del Pascal.

6.3.1. SOAE: Spontaneous Otoacoustic Emissions.

Este es el tipo de OAE que observó Kemp en 1978. Se trata de señales de muy bajo nivel (-10 a 15 dB SPL) con frecuencias entre 1 y 4 kHz, las cuales se generan espontáneamente en la cóclea sin aplicar estímulo alguno. La ausencia de SOAE no es un buen indicador de pérdida de audición, dado que este tipo de OAE se observa en aproximadamente el 50% de la población. Esto, sumado al bajo nivel de las señales a capturar, que las hace muy vulnerables al ruido ambiente, lleva a que las SPOAE no se utilicen habitualmente en equipos de diagnóstico.

6.3.2. TEOAE: Transient Evoked Otoacoustic Emissions

Las TEOAE se producen en respuesta a un estímulo de corta duración. El procedimiento para observar TEOAEs consiste en aplicar estímulos auditivos, en general “clicks” o tonos pulsados con duraciones de aproximadamente 100µs y amplitudes del orden de 40 dB SPL. En forma sincronizada con estos estímulos se registran las OAE captadas mediante el micrófono y luego, mediante técnicas de promediación estadística, similares a las utilizadas en ABR (Auditory Brainstem Response), se estiman las señales de OAE. Las TEOAE presentan componentes de hasta unos 5kHz.

6.3.3. DPOAE: Distortion Product Otoacoustic Emission

El método de DPOAE es posiblemente la técnica más difundida actualmente para estudios diagnósticos y la que fue utilizada en el SE desarrollado. Estas OAE se producen por un proceso no-lineal activo que ocurre en la cóclea. Se trata de OAE evocadas, es decir, resultan de una respuesta a un estímulo. En este caso, el estímulo consiste en la combinación lineal de dos tonos.

Debido a la característica no-lineal de la cóclea, al aplicar dos tonos de frecuencias f_1 y f_2 , las OAE evocadas presentaran componentes en las frecuencias f_{DP} que verifiquen:

$$f_{DP, n, m} = n \cdot f_1 + m \cdot f_2 \quad n, m \text{ enteros} \quad (1)$$

Estas OAE se denominan productos de distorsión (DPOAE). Si bien estas OAE se producen en todas aquellas frecuencias que verifiquen (1), la de mayor amplitud [Craigmile, 2004] y la más utilizada es el producto de distorsión cúbico dado por:

$$f_{DP} = 2 \cdot f_1 - f_2 \quad (2)$$

Este producto de distorsión es el más robusto y estable en el tiempo [Craigmile, 2004]. Además, experimentalmente se encontró que el producto cúbico de distorsión es máximo si las frecuencias de los tonos verifican:

$$f_2 = 1.2 f_1 \quad (3)$$

Las DPOAE se producen en el rango de 1 a 6 kHz y es usual realizar los estudios en 2, 3 y 4 kHz. La amplitud de estas OAE crece con la amplitud de las señales primarias inyectadas f_1 , f_2 y por eso es deseable utilizar la mayor amplitud posible; aunque hay algunas restricciones que deben cumplirse. Si la amplitud de las primarias supera los 75 dB SPL pueden observarse DPOAE que resultan de reflexiones en la membrana basilar y no son útiles para determinar la capacidad auditiva del paciente [Kinght, 2000] [Kemp, 1997]. Por lo tanto, generalmente se utilizan amplitudes primarias f_1 , f_2 entre 60 y 70dB SPL.

6.4. Procedimiento de medida

Antes de comenzar el estudio propiamente dicho se aplican los estímulos f_1 , f_2 en el canal auditivo y se verifica, mediante el micrófono, que el nivel de éstas sea el deseado. De no ser así se modifica la amplitud de los tonos hasta conseguirlo. Este procedimiento permite verificar el correcto ajuste de la sonda en el canal auditivo, la presencia de cerumen o la obstrucción de los conductos de la sonda. Estas falencias se detectan porque se requiere incrementar más allá de lo habitual la potencia de f_1 y f_2 para conseguir el nivel nominal en el canal auditivo.

Una vez asegurado un buen ajuste de la sonda, se aplican los tonos f_1 , f_2 , se adquiere la OAE mediante el micrófono y se estima su amplitud. La amplitud de las DPOAE es muy baja, entre -10 y 15dB SPL, pero por tratarse de un potencial evocado, producto de un estímulo que está disponible (las primarias f_1 , f_2), existe una gran colección de técnicas clásicas y emergentes para rescatar las DPOAE del ruido de fondo. Las técnicas más utilizadas son la promediación coherente y la transformada discreta de Fourier (DFT), pero también se están introduciendo técnicas de filtrado adaptivo lineal, no-lineal y métodos de ajuste (“fitting”) [Schwartz, 1975]. Como criterio general, se considera que la OAE está presente si su SNR (relación señal a ruido) es mejor de 6 dB y su amplitud supera los 0 dB SPL.

Según se infiere de la documentación de los instrumentos comerciales, éstos trabajan a nivel de tramo o “frame”. Es decir, toman segmentos de corta duración de la señal, a partir de los cuales van obteniendo una estimación de la OAE que mejora con el número de frames acumulados. Este modo de trabajar la señal por tramos, permite rechazar aquellos que contengan artefactos o un alto nivel de ruido, los cuales corromperían la estimación de las OAE. Algunos equipos proveen información de la cantidad de frames rechazados para que el operador tome medidas. Por ejemplo, si la proporción de frames rechazados es muy alta, puede intentarse reducir el ruido ambiente o verificar la correcta colocación de la sonda.

6.5. Desarrollo del equipo de medida.

No existe una norma ni un procedimiento preciso de las funciones de un equipo para OAE, pero algunas marcas hegemónicas han impuesto sus estrategias de medida como estándares. Por este motivo, se optó como estrategia de desarrollo, seguir los lineamientos de estas marcas e implementar una primera versión del hardware que permita generar estímulos y adquirir las respuestas OAE. El equipo básico debe cumplir con las siguientes funciones:

1. Generación de tonos entre 1 y 5 kHz con intensidades entre 40 y 80 dB SPL
3. Captura de señales de audio con un ancho de banda de 6 kHz.

Además, las OAE deben capturarse en forma sincronizada con los estímulos aplicados y no se requiere la captura continua de las OAE, sino que puede hacerse por intervalos (frames).

6.6. Esquema general del equipo de medida.

Los requisitos que debe cumplir el equipo para DPOAE son similares a los de la ICC basada en SSVEP descrita en el capítulo previo: se requiere generar estímulos, adquirir la respuesta en forma sincronizada y calcular su transformada de Fourier. La diferencia sustancial entre estos SEs es la velocidad requerida: mientras que en SSVEP los estímulos y las respuestas son señales de algunos Hz, en DPOAE son de algunos kHz. Esto impacta en las frecuencias de muestreo, en los tiempos de respuesta que el dispositivo debe asegurar y en consecuencia en el diseño del SE, particularmente en la división de las tareas y en su estructura.

Distribución de tareas y estructura del hardware

El dispositivo a desarrollar posee dos partes fundamentales: un sistema embebido que permite la adquisición de potenciales y el procesamiento en tiempo real, y la computadora para el procesamiento sin restricciones temporales y la interfaz de usuario (Figura 6.2).

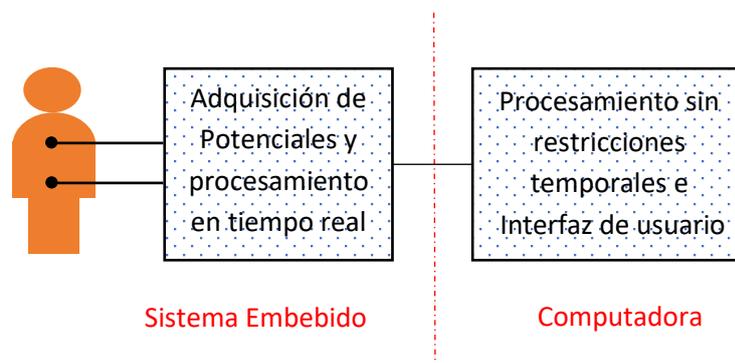


Figura 6.2. Esquema general del equipo.

En la Figura 6.3 se presenta un diagrama en bloques del SE. En líneas generales, el equipo dispone de una probeta que aloja un micrófono y dos receptores (parlantes), un preamplificador de bajo ruido, una unidad de adquisición, un generador de estímulos programable y un módulo de comunicaciones que transfiere las señales capturadas a un ordenador donde se realiza el procesamiento de las mismas. El sistema embebido estará del lado del paciente, por lo cual su tamaño debe ser reducido. Por otro lado, por seguridad del paciente debe operar con baterías, lo cual exige al diseño un bajo consumo y una autonomía mínima de 8 horas para lograr finalizar un día de trabajo al profesional a cargo. La función principal del SE es generar los estímulos y

registrar la respuesta del micrófono en sincronismo con la excitación. Como ya se ha mencionado, la señal se procesará por tramos (frames), con lo cual, por medio del módulo de comunicación los mismos se irán enviando a la computadora a medida que se ven registrando.

En la computadora, se deberá desarrollar un programa que permita visualizar el avance del estudio al profesional a cargo y de la misma forma gestionar una base de datos conformada por los estudios de sus pacientes.

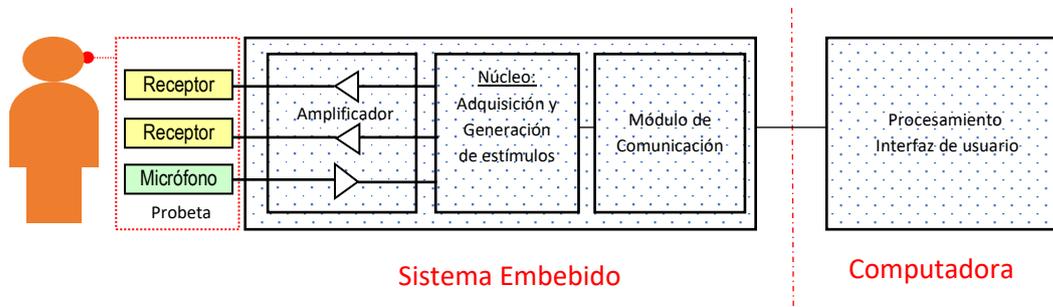


Figura 6.3. Diagrama en bloques de equipo de OAE.

6.7. *Probeta y Amplificador.*

La probeta debe contener un micrófono de bajo ruido y dos receptores capaces de generar niveles sonoros de 65dB SPL en el canal auditivo. Ambos deben operar en un rango de trabajo de 1-6 kHz. El micrófono debe resolver pequeñas OAE (del orden de -10dB SPL en presencia de estímulos de gran amplitud (65dB SPL): su rango dinámico (DR), en conjunto con el canal de adquisición de audio, debe ser superior a los 75dB.

$$DR_{MIC} > 80 \text{ dB}$$

En la Figura 6.4 se presentan la probeta de un equipo comercial (Grason-Stadler) y la primera probeta desarrollada localmente por la empresa Ladie Audiologia.



Figura 6.4. Probetas

6.8. *Núcleo.*

La placa adquisidora debe permitir capturar las señales en un rango de frecuencias de 1 a 6 kHz resolviendo OAE de -10dB SPL en presencia de tonos (f_1 , f_2) de 65 dB SPL, lo cual exige un

Rango Dinámico (DR) de alrededor de 80 dB. Para cumplir con estos requerimientos es suficiente utilizar una frecuencia de muestreo f_s de 50kHz y una resolución de 16 bits (idealmente DR=96dB).

$$f_s = 50\text{kHz} ; 16 \text{ bits}$$

El instrumento debe generar estímulos sinusoidales entre 1 y 6 kHz (tonos con niveles sonoros de hasta 65 dB SPL. Estos niveles deben poder ser ajustados de acuerdo a las características del canal auditivo con una precisión de al menos 1 dB SPL. Las armónicas y los productos de distorsión propios del generador deben estar por debajo de -10 dB SPL en presencia de primarias f_1, f_2 de 65 dB SPL (-75 dB por debajo), para no generar señales espurias que puedan confundirse con las OAE. En la Tabla 6.1 se consignan las principales especificaciones del generador de estímulos.

Tabla 6.1. Especificaciones del sistema de generación de estímulos.

Frecuencias:	1-6kHz
Nivel sonoro:	ajustable entre 55 y 65 dB SPL
Resolución:	mejor que 1 dB SPL
Distorsión:	mejor que -75 dB.

6.8.1. Estructura del sistema adquireedor/generador de audio.

La estructura general del circuito se muestra en la Figura 6.5. El mismo está basado en un códec de audio estéreo (convertidores ADC y DAC), controlado mediante un DSP que está conectado a un PC mediante un enlace inalámbrico. El prototipo debe generar señales de audio en forma continua y enviar frames de 20 ms de duración de la señal capturada por el micrófono.

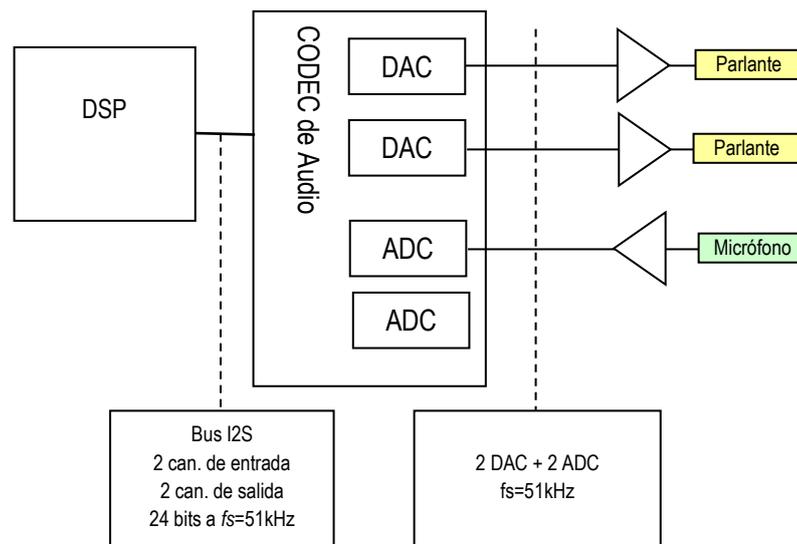


Figura 6.5. Esquema general del sistema adquireedor/generador de audio.

6.8.2. Selección del Codec de audio.

Para poder adquirir las débiles señales de OAE (0-20 dB SPL) en presencia de estímulos de gran amplitud (65 dB SPL), se requiere un rango dinámico de al menos 80-90 dB, es decir conversores ADC y DAC de unos 15 bits libres de ruido. Para cumplir estas especificaciones se seleccionó un códec de audio de 24 bits que también puede operar con una resolución de 16 bits.

6.8.3. Selección del DSP.

El procesamiento de las señales de OAE se realizará en la PC. La principal función del DSP, núcleo del SE, es generar los estímulos a partir de tablas almacenadas, capturar las señales entregadas por el ADC y enviar frames de 20ms de duración al PC. Estos frames deben adquirirse en forma sincronizada con los estímulos generados.

Si bien las tareas que debe realizar el DSP son simples, éstas deben hacerse de forma muy eficiente por el gran volumen de datos en juego. Considerando 2 canales de audio de 24 bits a una frecuencia de muestreo de 51.2kHz, resulta una tasa de datos de 2.5Mbits/seg entre el DSP y el codec. Para conseguir esta velocidad de transferencia es conveniente utilizar un DSP con interfaz I2S: un standard habitual en los sistemas de audio de alta calidad.

La elección del DSP no es crítica, basta con un dispositivo que disponga de la interfaz I2S, al menos 64KB de Memoria Flash (para almacenar código y tablas) y 4 KB de memoria RAM. Se optó por un DSP que la empresa LADIE ya estaba utilizando en sus diseños a fin de reducir la variedad de componentes de sus productos.

6.8.4. Generador de audio.

Este generador debe reproducir señales almacenadas en memoria a una frecuencia de muestreo de 51.2kHz. En el modo DPOAE, debe generar 2 tonos de audio en forma simultánea, de las frecuencias f_1 , f_2 que se indican en la Tabla 6.2 y con intensidades ajustables entre 50 y 70 dB SPL en pasos de 1 dB SPL o menores. En forma sincronizada con la generación de estos tonos, se debe adquirir la señal entregada por el micrófono.

Tabla 6.2. Frecuencias e intensidades de las primarias f_1 , f_2

f_{nominal} [Hz]	f_1 [Hz] 65 dB SPL	f_2 [Hz] 55 dB SPL
2000	1750	2150
3000	2550	3100
4000	3300	3950
5000	4150	5000
6000	5050	6150

Para generar estos tonos se confeccionaron 10 tablas con las muestras correspondientes a cada una de las primarias (5 para f_1 y 5 para f_2). En primer lugar, se evaluó la resolución necesaria en la representación de las muestras considerando que con estímulos de 65 dB SPL todas las componentes espurias deben estar por debajo de -10dB SPL, es decir se requiere un rango dinámico de más de 75dB. En teoría, esto puede conseguirse con un DAC “ideal” de unos 13 bits.

Cómo primera aproximación se consideró utilizar los DACs con una resolución de 16 bits; dejando como reserva la posibilidad de incrementarla a 24 bits en caso de ser necesario. En la Figura 6.6 se muestra el contenido espectral, obtenido por simulación, para el caso de generar un único tono de 65 dB SPL con un DAC ideal de 16 bits. Se observa aquí que las componentes espurias se encuentran por debajo de -40dB SPL, lo cual cumple holgadamente con los requerimientos. Las tablas con las primarias f_1 , f_2 se confeccionaron con una amplitud igual al fondo de escala, a fin de aprovechar el rango dinámico del DAC. La amplitud de las señales de salida, que serán aplicadas a los receptores, se controlará mediante los atenuadores embebidos en el códec.

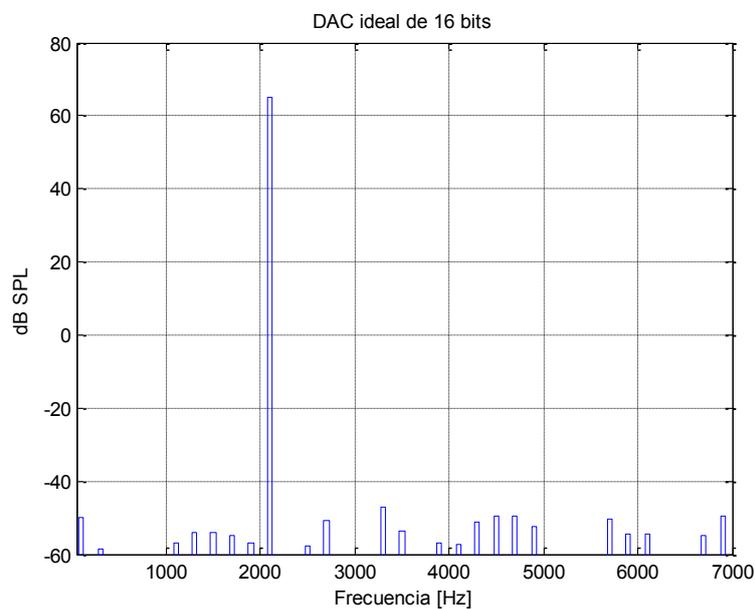


Figura 6.6. Contenido espectral de una sinusoide generada con una resolución de 16 bits a una tasa de muestreo de 51.2 kHz.

6.8.5. Sistema operativo embebido.

Para la implementación del sistema operativo embebido es suficiente la implementación de una versión propietaria del tipo primer plano/segundo plano haciendo uso de los distintos niveles de prioridad que implementa el DSP en su manejador de interrupciones. En primer plano estará operando la actividad que adquiere las muestras de la señal del micrófono en sincronismo con los tonos generados por tabla y genera los paquetes de datos; y en segundo plano estará funcionando la tarea que comanda el módulo inalámbrico para enviar los paquetes a la computadora.

Las formas de onda a generar utilizando los DAC del convertor (f_1 y f_2) fueron generadas utilizando un programa de matemática y almacenadas en vectores estáticos dentro de archivos de cabecera. De esta manera, las formas de onda se almacenan en memoria de programa liberando memoria RAM, y cuando se requiere utilizarlas se recorren por medio del uso de punteros.

6.9. Módulo de comunicación.

El sistema de comunicaciones debe transmitir frames desde el adquisidor hacia el ordenador que procesará las señales. Como referencia se tomó un equipo comercial (Grason & Stadler) que procesa alrededor de 400 frames en 30 segundos, es decir aproximadamente 14 frames/s. Cada uno de estos frames tiene una duración de 20ms, que a una frecuencia de muestreo aproximada de 50kHz y una resolución de 16 bits corresponden a 16kbits (1 frame=16 kbits). Entonces, para transmitir 15 frames por segundo se requiere de un canal de comunicaciones con una capacidad C de al menos 220 kbits/s.

$$C > 220 \text{ kbits/s}$$

Para implementar la etapa de comunicación entre el sistema embebido y la computadora se utilizó el protocolo de comunicación inalámbrico "Bluetooth". Uno de los principales motivos por los cuales se ha seleccionado este protocolo es por su condición de estándar, lo cual asegura la compatibilidad con otros dispositivos en el mismo área de trabajo. Por otro lado, es un protocolo muy difundido en notebooks, netbooks y teléfonos celulares por lo cual la etapa de comunicación en la computadora se puede resolver de manera directa. Otra ventaja es que, al estar tan difundido, se asegura la disponibilidad de hardware por un largo plazo de tiempo.

Otros posibles protocolos estándar disponibles en el mercado actual son: Wi-fi y Zigbee. El protocolo Wi-fi verifica los requerimientos de ancho de banda (BW), rango de alcance, disponibilidad en el mercado e integración en netbooks, pero se desestimó por su alto consumo de energía. Como contraparte, el protocolo Zigbee presenta un consumo de energía optimizado, pero se descartó porque su ancho de banda no verifica los requerimientos del equipo de OAE.

Se omite el detalle de las principales características del protocolo Bluetooth y el módulo de radiofrecuencia utilizado dado que ya fueron presentados en el capítulo 4.

6.10. Computadora.

El principal objetivo de esta etapa es la implementación de la interfaz de usuario y el procesamiento de los frames recibidos desde el sistema embebido por Bluetooth. Este tipo de implementación se puede resolver haciendo uso de cualquier computadora, pero se considera conveniente utilizar una notebook que incluya de serie el protocolo Bluetooth permitiendo una mayor seguridad para el paciente y comodidad para el profesional de la fonoaudiología.

6.10.1. Algoritmos para la detección de otoemisiones.

Las otoemisiones acústicas presentan bajas relaciones señal/ruido (SNR) las cuales difícilmente superan el 0dB. Para mejorar esta SNR y realzar las OAE frente al ruido, es habitual utilizar técnicas de promediación estadística. Si bien existen diversas formas de abordar este problema, las técnicas más utilizadas son la promediación coherente y la promediación de espectros.

6.10.1.1. Promediación coherente

La promediación coherente es un método simple y eficaz para rescatar señales obtenidas como respuestas a un estímulo disponible. Esta técnica consiste en adquirir tramos, realizaciones o "frames" de la respuesta en forma sincronizada con el estímulo y luego, promediando un gran

número de estos frames, se obtiene una estimación de la respuesta. En la Figura 6.7 se muestra un ejemplo con 10 frames y el resultado obtenido al promediarlos.

Cómo puede observarse en la figura 6.7, el método de promediación coherente requiere una precisa sincronización entre los frames, para que al promediarlos resulte realizada la respuesta buscada. Esto exige, por un lado, que la adquisición de los frames sea realizada en forma sincronizada con el estímulo aplicado y por otro, que las respuestas estén sincronizadas con el estímulo. La primera restricción es sólo un requerimiento para la electrónica de adquisición, mientras que la segunda está relacionada con la fisiología del fenómeno a estudiar y limita el campo de las posibles aplicaciones de este método.

El método en sí se reduce a la promediación de un cierto número de realizaciones del fenómeno, pero su implementación práctica requiere de algunas consideraciones adicionales. Por ejemplo, es importante rechazar aquellos frames que contienen artefactos u otras señales espurias de gran amplitud, porque de otro modo estas señales no deseadas “destruirán” el promedio y se requerirá un gran número de realizaciones para recuperar la señal deseada. El rechazo automático de frames es un punto importante del algoritmo, que requiere fijar criterios y definir estrategias específicas de cada aplicación. Para esto último es fundamental adquirir experiencia y familiaridad con las señales.

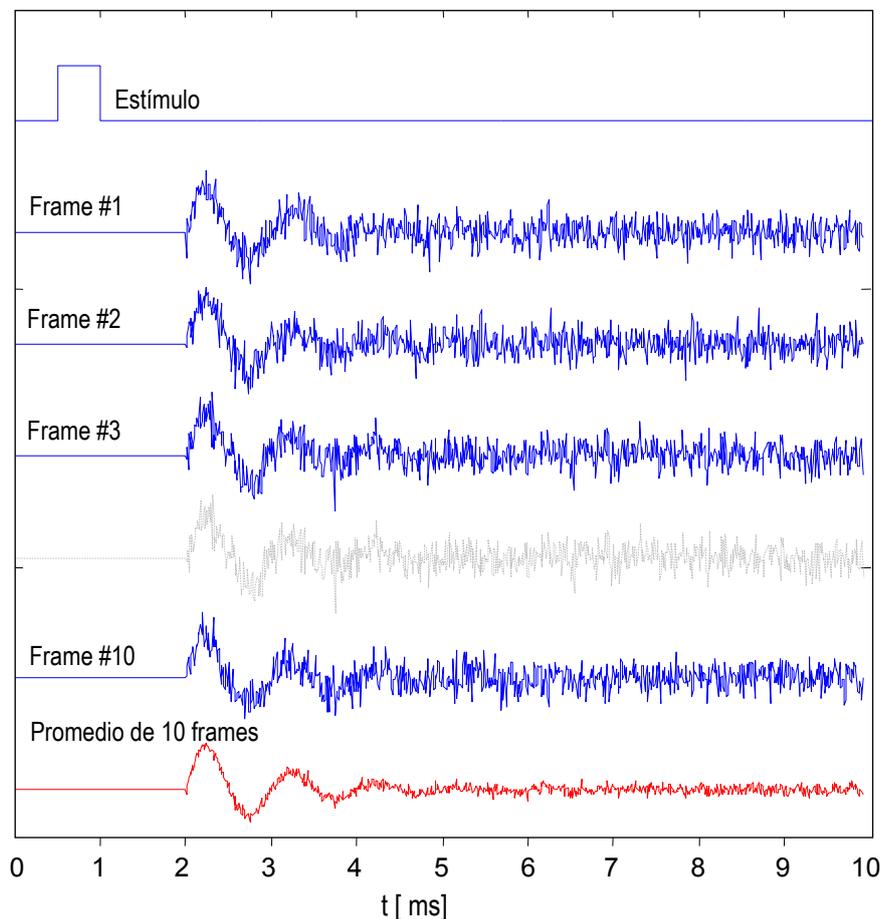


Figura 6.7. Mejora de la SNR por promediación coherente

6.10.1.2. Comparación de Algoritmos por simulación.

A continuación, se presenta, mediante un ejemplo, la implementación de algoritmos simples para estimación de DPOAE.

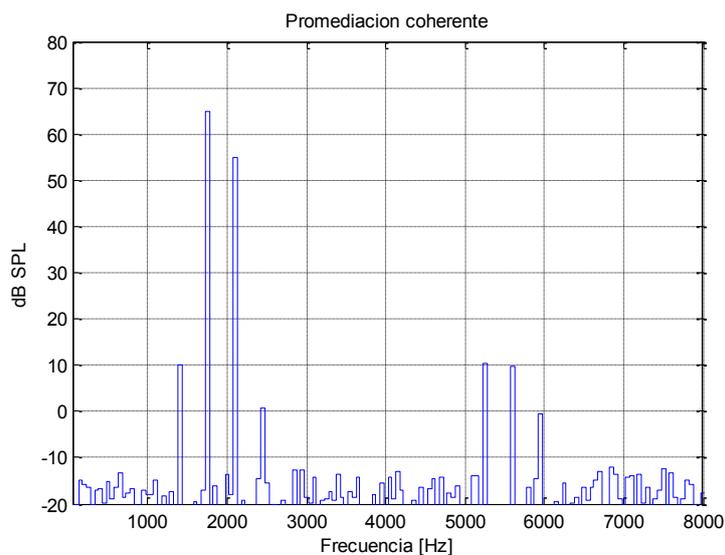


Figura 6.9. Espectro estimado mediante promediación coherente. Se adicionó ruido blanco de 30 dB SPL y se promediaron 128 frames.

6.10.1.4. Promediación de Espectros en DPOAE

En las mismas condiciones del punto previo, es decir un nivel de ruido de 30dB SPL y promediando 128 frames, resulta el espectro que se muestra en la Figura 6.10. Se observa aquí que las amplitudes estimadas para las primarias y de las OAE coinciden con las obtenidas por promediación coherente, pero existe un nivel base de ruido por debajo del cual no es posible detectar OAEs. En este caso, el mencionado nivel base corresponde a los 30dB SPL distribuidos en el ancho de banda de 0-25.6kHz.

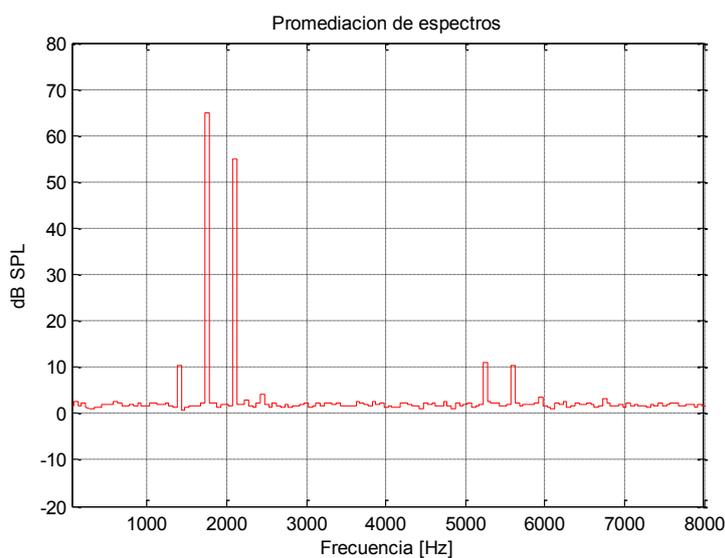


Figura 6.10. Espectro estimado mediante promediación de espectros. Se adicionó ruido blanco de 30 dB SPL y se promediaron 128 frames.

6.10.1.5. Promediación Coherente Vs. Promediación de espectros en DPOAE

Utilizando promediación coherente es teóricamente posible reducir el ruido en forma ilimitada incrementando el número de frames promediados; pero para que esta reducción de ruido implique una mejora en la SNR es necesaria una perfecta sincronización de las respuestas registradas. De lo contrario, la estimación de las OAE se diluirá conjuntamente con el ruido.

El método de promediación de espectros es más robusto, no requiere de una precisa sincronización de las respuestas OAE pero, aun incrementando ilimitadamente el número de frames promediados, no permite discriminar OAE por debajo de un valor umbral relacionado con el nivel de ruido en el canal auditivo.

Si existe una buena sincronización entre las respuestas OAE, la promediación coherente es la mejor alternativa, pero si existen variaciones en el tiempo de latencia, esta técnica puede conducir a medidas erróneas.

Si el nivel de ruido no es muy elevado, la promediación de espectros es una solución simple y robusta. Este método proporciona además medidas confiables del nivel de las primarias f_1 , f_2 y del nivel de ruido en el canal auditivo.

6.10.2. Programación en computadora.

Como se ha analizado, el sistema de procesamiento debe permitir realizar la promediación coherente de las señales adquiridas, calcular su transformada de Fourier utilizando el algoritmo FFT y la amplitud de las distintas componentes espectrales expresadas en dB SPL. Esto último requiere incluir la respuesta propia del micrófono y posiblemente también un proceso de calibración. Un punto importante dentro del algoritmo a utilizar es fijar estrategias y criterios para rechazar aquellos frames que posean un alto nivel de ruido.

La interfaz gráfica debe guiar al usuario en la operación del instrumento, debe permitir ajustar los niveles sonoros de estimulación, el tipo (DPOAE/ TOAE), las frecuencias de estos estímulos y presentar en pantalla la amplitud de las distintas componentes en dB SPL.

En la Figura 6.11 se presenta una captura de pantalla del software desarrollado como interfaz con el usuario del equipo de OAE. En la misma puede observarse que el equipo cuenta con todas las prestaciones de los instrumentos que en la actualidad se están utilizando para realizar este tipo de estudios.

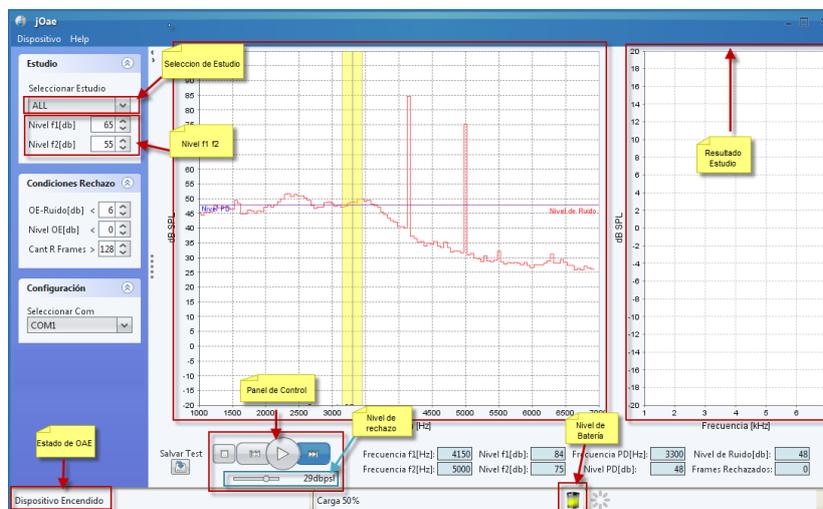


Figura 6.11. Captura de pantalla del software de OAE.

6.11. Conclusiones.

En este capítulo se ha presentado el proceso de desarrollo de un equipo para la detección temprana de hipoacusias. El desarrollo ha sido generado como un trabajo de transferencia a pedido de una empresa del ámbito local. No se conocen desarrollos similares en América del Sur y existen muy pocas empresas en el mundo que han accedido a esta tecnología.

La versión final del equipo es totalmente funcional y supera ampliamente las expectativas planteadas al inicio del proyecto. Como se puede observar en la Figura 6.12 se ha logrado integrar todos los módulos que integran el SE en una única placa resultando un equipo compacto y robusto. Se ha transferido la información y el conocimiento al personal de la empresa de tal forma de lograr independencia y dar por finalizado el convenio.



Figura 6.12. Versión final del prototipo transferido.

El equipo desarrollado es actualmente comercializado por la empresa LADIE Audiología. En la Figura 6.13 se presenta el catálogo utilizado actualmente para la publicidad y comercialización del producto.

Analizador de Otoemisiones Acústicas OE-500

Bluetooth

Visualización en tiempo real de la evolución de la prueba con gráficos y valores de DPOAE, ruido, frecuencia, etc.

Base de datos para pacientes, con todas las pruebas realizadas

Protocolos por defecto disponibles y funcionalidad para generar protocolos personalizados

Completo Informe de resultados personalizable

LADIE
Audiología
Laboratorio de Aplicación y Desarrollo de Instrumental Electrónico

Figura 6.13. Folleto actual del equipo OE-500 comercializado por Ladie audiológica.

Parte 3: Resultados.

Capítulo 7

Conclusiones y Líneas Futuras.

Los requerimientos que las aplicaciones biomédicas imponen sobre los sistemas embebidos son muy específicos y disímiles; por ello en esta tesis se analizaron distintas situaciones y se propusieron arquitecturas y soluciones para cada uno de estos escenarios. A partir de ejemplos concretos se describió el fraccionamiento de las tareas; en primer lugar, entre el procesamiento analógico y el procesamiento digital de señales y luego entre distintas plataformas sobre las cuales se implementan las tareas de procesamiento digital.

Los dispositivos desarrollados, que tienen distintos grados de complejidad y requerimientos particulares, abarcan desde pequeños SE basados en microcontroladores de 8 bits, hasta plataformas con microprocesadores de 32 bits con sistemas operativos de tiempo real. Cada uno de estos equipos resultaron como soluciones propias y apropiadas a problemas específicos de bioingeniería e incorporan aportes originales en distintos aspectos de los SE.

Los sistemas desarrollados en esta tesis cubren gran parte de las aplicaciones biomédicas comúnmente implementadas tanto en el ambiente científico como en la clínica médica y presentan escenarios muy disímiles con requerimientos temporales de distinto tipo. Típicamente, este tipo de sistemas deben verificar tareas en tiempo real, con sincronización entre estímulos y sus respuestas evocadas.

En algunas aplicaciones, donde se pretende registrar algún biopotencial de manera continua y no es necesaria la sincronización entre estímulo y respuesta, es suficiente la utilización de un escenario con un sistema embebido simple que se complementa con una computadora para el postprocesamiento de las señales. En este caso, el SE contará con un sistema operativo embebido (SOE) del tipo primer plano/segundo plano propietario que registre las muestras en primer plano y envíe a la computadora los paquetes de datos en segundo plano. El sistema descrito en el capítulo 4 es un ejemplo de este escenario de implementación: un adquisidor de señales de EEG autónomo con transmisión inalámbrica para Interfaces Cerebro-Computadora (ICC), que también fue reutilizado como plataforma de inicio para experimentar con SSVEP.

En el caso de equipos autónomos, que deben prescindir del uso de una computadora, pero que requieran una interfaz gráfica de usuario, soporte para red y sistema de archivos, entre otras prestaciones, se debe migrar a la utilización de un SE más complejo que resuelva estos requerimientos por medio de un sistema operativo que pueda portar. En estos casos, conviene desarrollar los controladores necesarios, reutilizando la solución que el SOE implementa para la interfaz de usuario y el sistema de archivos entre otros. Será la responsabilidad del diseñador seleccionar un SO adecuado y desarrollar los controladores de dispositivo necesarios para verificar las prestaciones de tiempo real que la implementación demande. El sistema descrito en el Capítulo 5 es un ejemplo de este escenario: una plataforma para implementación de ICC embebida basada en SSVEP. En este caso se seleccionó el SO Windows CE 6 que presenta una interfaz gráfica amigable y conocida por los usuarios, con soporte para red y sistema de archivos. En este caso, fue necesario desarrollar un controlador de dispositivo (driver) para lograr digitalizar

las muestras de EEG en sincronismo con los estímulos aplicados al usuario. El tiempo de latencia de atención a interrupciones resultó por debajo de 100 μ s, siendo el sistema útil para muestrear señales por debajo de 10 kHz sin pérdida de muestras, con prestaciones de tiempo real duro.

Cuando es necesario muestrear señales a mayor velocidad, del orden de cientos de kHz, y manteniendo aún el sincronismo con los estímulos, es conveniente migrar a un escenario que incluya un SE con particiones de hardware dedicado para tareas específicas. Una posibilidad es la inclusión de un procesador digital de señales (DSP) dedicado a la resolución del muestreo en sincronismo con el estímulo de alta velocidad, dejando el procesamiento de señal para una segunda etapa que podrá resolverse en una computadora o un SE similar al descrito previamente. El sistema descrito en el Capítulo 6 de esta tesis es un ejemplo de este tipo de escenario: un equipo para diagnóstico de hipoacusias basado en oto-emisiones acústicas. En este caso, el DSP se encarga de comandar un codec de audio dedicado a generar los tonos de estímulo en sincronismo con la digitalización de la señal del micrófono operando a 51,2 kHz. El mismo cuenta con un SO del tipo primer plano/segundo plano propietario. En primer plano se comanda el codec de audio y en segundo plano se envían paquetes de datos a la computadora para su procesamiento.

A continuación, se detallan algunas conclusiones particulares para los distintos dispositivos desarrollados y luego algunas conclusiones generales que presentan lineamientos para el fraccionamiento de tareas.

7.1. Conclusiones particulares para los dispositivos implementados.

7.1.1. Adquisidor de señales de EEG autónomo con transmisión inalámbrica para Interfaces Cerebro-Computadora (ICC):

Esta implementación que se ajusta al modelo de sistema embebido simple para la adquisición de potenciales y una computadora para el procesamiento y la interfaz de usuario; es de gran utilidad en el ambiente académico cuando se comienza a trabajar con algún potencial desconocido. En este sentido, adquirir los biopotenciales con el sistema embebido portátil y transmitir la información en bruto a la computadora resulta de gran utilidad. Esto permite realizar un procesamiento offline de la señal usando programas de cálculo matemático para extraer información de utilidad y comparar distintas técnicas que permitan detectar características en la señal analizada.

La flexibilidad de esta estructura resulta cómoda para experimentos iniciales, por ejemplo, en la detección de potenciales evocados visuales de estado estacionario como adquisidor de señales de EEG. Este sistema de medida, al igual que el ejemplo previo, verifica el escenario compuesto por un SE simple para adquisición y una computadora para procesamiento. Esta plataforma permitió realizar ensayos preliminares sobre la utilización de potenciales evocados visuales de estado estacionario en ICC. A partir de estos resultados se definieron las especificaciones de una ICC autónoma, que resulta apropiada para una implementación compacta y autónoma, pero resulta algo rígida para realizar ensayos exploratorios. En estos ensayos se probaron dos tipos de estimuladores: uno utilizando un monitor de tubo de rayos catódicos y otro basado en diodos emisores de luz (LEDs). Se observó que ambos tipos de estimuladores pueden ser utilizados en ICC, pero la estimulación mediante LEDs produce potenciales evocados de mayor amplitud con buena resolución espectral, lo cual permite utilizar una mayor cantidad de frecuencias de estimulación simultáneas.

Estas medidas, permitieron verificar la correcta generación y detección de los potenciales evocados en concordancia con lo reportado en la bibliografía y constituyeron el punto de partida para la implementación de una ICC embebida basada en estos potenciales.

7.1.2. ICC embebida basada en SSVEP:

En Interfaces Cerebro-Computadora y dispositivos de asistencia para usuarios finales en general, es deseable evitar la dependencia de computadoras de propósitos generales en la implementación del equipo, al menos en sus funciones esenciales. Por un lado, un ordenador personal o una notebook tiene capacidades gráficas y de conectividad, pero tienen tiempos de arranque significativos y son poco robustos. Por el otro, implementar soporte gráfico y de red en un SE requiere un gran esfuerzo y trabajo de desarrollo. En los casos en que el dispositivo requiere interfaz gráfica y soporte de red, la solución adecuada es utilizar plataformas de propósitos generales que provean el hardware necesario y un sistema operativo embebido que lo resuelva. Esta estructura se utilizó en el desarrollo de un deletreador híbrido embebido que brinda un posible canal de comunicación a personas con discapacidad.

La ICC desarrollada es totalmente autónoma, prescinde del uso de una computadora y presenta un rápido tiempo de encendido de unos 10 segundos. Se utilizaron arreglos de LED en color rojo con una película difusora para generar los estímulos visuales y se implementó el concepto de “brain switch” usando el ritmo alfa visual que permite al usuario encender o apagar los estímulos visuales, para evitar que perturben al usuario cuando no utiliza la ICC. Además, el sistema provee una realimentación visual por medio de una barra de progreso que resulta muy útil para mejorar la recuperación de potenciales evocados; y una realimentación auditiva para los comandos alfa. Los resultados experimentales validaron la correcta operación del sistema y la posibilidad de transferirlo a un usuario. Si bien las pruebas realizadas no han sido extensivas, muestran una buena performance, y los resultados experimentales sugieren que el entrenamiento puede mejorar notablemente la velocidad del deletreador. Con muy poco entrenamiento las velocidades de escritura promedio para los dos sujetos con los que se experimentó fueron de 7 y 4.3 letras/minuto. Finalmente cabe destacar que se ha logrado la implementación completa del deletreador embebido, siendo el sistema completamente funcional y un buen precedente para la implementación requerida por un usuario final. El hardware puede ser modificado fácilmente para implementar otros dispositivos de asistencia como un controlador de silla de ruedas o un controlador de prótesis, entre otros.

7.1.3. Equipo para diagnóstico de hipoacusias basado en Oto-emisiones Acústicas (OAE):

En esta implementación se presentó el proceso de desarrollo de un equipo para la detección temprana de hipoacusias. El desarrollo fue generado como un trabajo de transferencia a pedido de una empresa del ámbito local. No se conocen desarrollos similares en América del Sur y existen muy pocas empresas en el mundo que han accedido a esta tecnología.

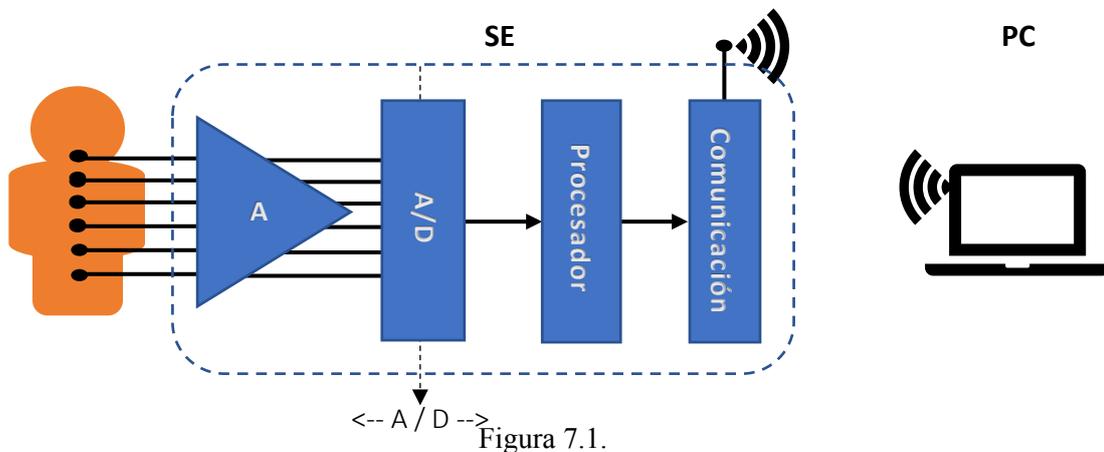
Este proyecto fue una transferencia directa de los conocimientos producidos y adquiridos en el desarrollo de esta tesis hacia el sector productivo y ratificó los buenos resultados obtenidos en cuanto a la interacción entre profesionales del ámbito privado y personal del ámbito académico. La versión final del equipo es totalmente funcional y supera ampliamente las expectativas planteadas al inicio del proyecto. Se logró integrar todos los módulos que conforman el SE en una única placa resultando un equipo compacto y robusto. Se transfirió la información y el conocimiento al personal de la empresa de tal forma de lograr independencia y dar por finalizado el convenio. El equipo desarrollado se encuentra en proceso de comercialización.

7.2. Conclusiones por escenario.

A continuación, se detallan algunos lineamientos generales a seguir a la hora de implementar algún dispositivo biomédico que enmarca en alguno de los tres escenarios descritos a lo largo de la tesis. Se considera que los tres escenarios analizados cubren gran parte de las aplicaciones biomédicas comúnmente implementadas tanto en el ambiente científico como en la clínica médica.

7.2.1. Escenario 1: SE simple para adquisición y PC para procesamiento e interfaz de usuario:

En la Figura 7.1 se presenta un esquema general de este tipo de escenario compuesto por un SE y una PC. El SE está compuesto por un amplificador de biopotenciales, un convertor A/D, un procesador y un módulo de comunicaciones.



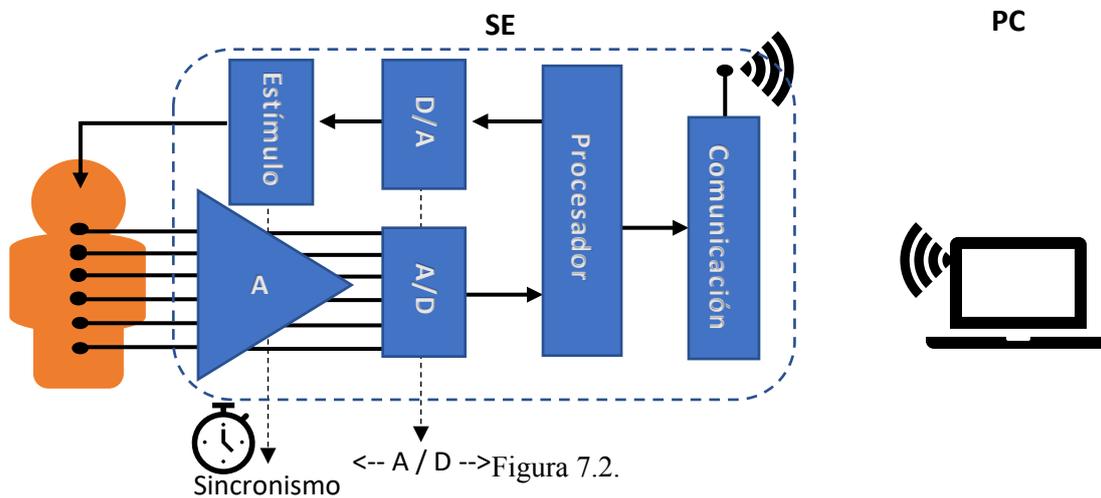
Lineamientos:

- El SE está del lado del paciente, por lo cual debe operar con baterías, tener bajo consumo y tamaño reducido.
- El amplificador de biopotenciales (A) deberá tener ganancia acorde al biopotencial a manejar, el número de canales que la aplicación requiera y buen rechazo a la interferencia electromagnética.
- El procesador debe contar con la interfaz de comunicación que el A/D seleccionado utilice. Típicamente SPI (del inglés Serial Peripheral Interface) o I2C (del inglés Inter Integrated Circuit).
- El procesador deberá cumplir con dos tareas principales: comandar el A/D para digitalizar señales y formar paquetes para enviar a la PC por medio del módulo de comunicación los datos crudos. En la gran mayoría de las aplicaciones será suficiente un microcontrolador de 8 bits de gama media/baja para cumplir con estas tareas.
- En cuanto al SO embebido es suficiente con una versión propietaria del tipo primer plano/segundo plano que utilice las interrupciones del procesador y sus prioridades para manejar las tareas. En primer plano estará muestreando y en segundo plano armando paquetes de datos y enviándolos a la PC.

- El módulo de comunicación debe ser inalámbrico para mantener al paciente aislado y contar con un ancho de banda acorde a la información a transmitir.
- En la PC se pueden utilizar programas de matemática para procesar la información off-line o bien desarrollar aplicaciones de usuario que procesen on-line la información recibida e implementen una interfaz de usuario.

7.2.2. *Escenario 2: SE compuesto para adquisición y estimulación, y PC para procesamiento e interfaz de usuario:*

En la Figura 7.2 se presenta un esquema general de este tipo de escenario compuesto por un SE y una PC. El SE está compuesto por los mismos módulos que el escenario anterior con el agregado de un módulo de conversión D/A y un módulo de estimulación. Este tipo de escenario, que a diferencia del anterior agrega la etapa de estimulación, típicamente necesita verificar sincronismo entre la señal estímulo aplicada y las respuestas evocadas a digitalizar.



Lineamientos:

- Se mantienen gran parte de los lineamientos presentados para el escenario previo, detallando solamente los que registran diferencias.
- El D/A deberá contar con tantos canales como señales se necesiten generar para comandar el estimulador, una resolución, rango dinámico y frecuencia adecuadas. En algunos casos, como cuando se procesa audio (OAE), puede resultar conveniente el uso de A/D y D/A integrados en un mismo chip (código de audio) a fin de simplificar la implementación y verificar el sincronismo.
- El procesador debe contar con la interfaz de comunicación que el A/D y D/A seleccionados utilicen. En el ejemplo de OAE donde se utilizó un código de audio la interfaz que se utiliza es I2S (del inglés Integrated Interchip Sound). Este tipo de interfaz está comúnmente disponible en DSPs o microcontroladores específicos.

- El procesador deberá cumplir con tres tareas principales: comandar el D/A para generar señales y controlar el estimulador, comandar el A/D para digitalizar los potenciales evocados en sincronismo con el estímulo generado y formar paquetes para enviar a la PC los datos crudos, por medio del módulo de comunicación. En los casos donde se manipulen señales de baja frecuencia y baja resolución puede ser suficiente un microcontrolador de 8 bits de gama media/baja para cumplir con estas tareas. En casos donde se manipulen señales de mayor frecuencia y resolución (como por ejemplo audio: frecuencias de muestreo de 50 kHz y resolución de 16 o 24 bits) es conveniente el uso de un DSP.
- En cuanto al SO embebido es suficiente con una versión propietaria del tipo primer plano/segundo plano que utilice las interrupciones del procesador y sus prioridades para manejar las tareas. En primer plano estará manejando el D/A y el A/D en sincronismo y en segundo plano armando paquetes de datos y enviándolos a la PC.

7.2.3. *Escenario 3: SE compuesto para adquisición, estimulación, procesamiento e interfaz de usuario:*

En la Figura 7.3 se presenta un esquema general de este tipo de escenario compuesto tan solo por un SE, prescindiendo del uso de PC. El SE está compuesto por los mismos módulos que el escenario anterior, pero en este esquema, será el propio SE quien deba procesar la información e implementar la interfaz de usuario.

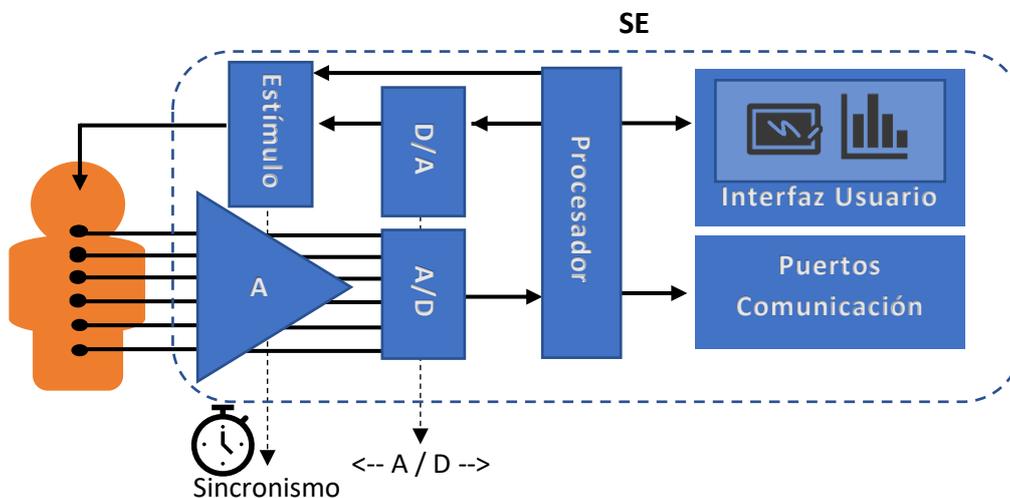


Figura 7.3.

Lineamientos:

- Se mantienen gran parte de los lineamientos presentados para el escenario previo, detallando solamente los que registran diferencias.
- En algunas aplicaciones puede prescindirse del uso del D/A, siendo la entrada al estimulador una señal digital generada directamente desde el procesador (Ejemplo SSVEP).

- El procesador debe contar con la interfaz de comunicación que el A/D y D/A seleccionados utilicen, además de puertos de E/S de propósito general para comandar el estimulador de manera directa.
- El procesador deberá cumplir con todas las tareas del escenario previo y además deberá implementar la interfaz de usuario y procesar las señales en tiempo real. En caso de que la interfaz de usuario sea gráfica, el procesador deberá controlar adicionalmente un display o monitor. Por otro lado, deberá contar con puertos de comunicación que permitan al SE interactuar con el exterior. Para lograr cumplir con esta demanda de tareas se puede utilizar un microcontrolador/microprocesador de 32 bits o un SoC que integre una FPGA y un microprocesador en un único chip.
- En cuanto al SO embebido es conveniente utilizar alguno que implemente los controladores para comandar el display gráfico y los puertos de comunicación por medio de su BSP. Por otro lado, es conveniente que verifique prestaciones de tiempo real duro a fin de no perder información de las señales a muestrear.
- El SO embebido deberá permitir el desarrollo de aplicaciones de usuario para programar la interfaz gráfica deseada y poder ensayar distintas técnicas de procesamiento sobre las señales de entrada.

7.3. Conclusiones generales y Líneas Futuras.

En esta tesis se presentó el desarrollo de algunos SE para aplicaciones biomédicas que son de utilidad tanto en el ambiente académico como en el clínico. Los dispositivos presentados tienen diferencias notorias en cuanto a las prestaciones que brindan y los requerimientos de hardware necesario para su implementación. Esto permitió experimentar con procesadores embebidos de distintas gamas. Desde microcontroladores de 8 bits sin SO, pasando por procesadores digitales de señal de 16 bits (DSP), hasta microprocesadores de 32 bits que incluyen unidad de manejo de memoria (MMU) y pueden portar SO como Linux Embedded, Windows Embedded o Android. Algunos de estos con prestaciones de tiempo real duro.

A futuro se pretende estudiar las plataformas conocidas como SoC (del inglés System On Chip) que integran la funcionalidad de alto nivel de un procesador con la operación en tiempo real, el procesamiento de datos complejos y las funciones de interfaz de una FPGA en un único dispositivo.

Estos dispositivos que integran ambas arquitecturas (procesador y FPGA) en un único dispositivo presentan mayor nivel de integración, menor consumo, tamaño de placas más pequeño y un mayor ancho de banda de comunicación entre el procesador y la FPGA.

En la actualidad se encuentran disponibles tres SoC de marcas hegemónicas con buena documentación: Altera SoC, Xilinx Zynq 7000 EPP y Microsemi SmartFusion2. Todas con prestaciones similares. En particular, se dispone del kit de desarrollo DE10-Nano de la empresa TerasiC, que incluye un SoC de Altera, obtenido por medio de un programa académico de la empresa al precio de US\$ 100. Como primer escenario de prueba se prevé migrar la implementación del delectador híbrido basado en SSVEP implementado en esta tesis (Figura 7.4) a una nueva versión utilizando un SoC (Figura 7.5).

Referencias.

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.

Arafa Pansy, Solomon Daniel, Navabpour Samaneh and Fischmeister Sebastian (2017) “Debugging Behaviour of Embedded-Software Developers: An Exploratory Study”. *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*.

Ashraf Mahmood and Ghazali Masitah (2011) “Investigating Physical Interaction Complexities in Embedded Systems”. *2011 5th Malaysian Conference in Software Engineering (MySEC)*.

Apencore (2017) “2017 Embedded Markets Study, Integrating IoT and Advanced Technology Designs, Application Development & Processing Environments”, April 2017. EETimes/embedded.com

Barbalace A., Luchetta A., Manduchi G., Moro M., Soppelsa A. and Taliercio C., (2008). “Performance Comparison of VxWorks, Linux, RTAI, and Xenomai in a Hard Real-Time Application”. *IEEE transactions on nuclear science*, vol. 55, no. 1, february 2008.

Berger H. (1929) “Über das elektrenkephalogramm des menschen”. *European Archives of Psychiatry and Clinical Neuroscience*, 87(1):527–570.

Bertolotti Ivan Cibrario (2017) “RTOS support in C-language toolchains”, *2017 IEEE International Conference on Industrial Technology (ICIT)*.

Bieger Jordi, Garcia- Molina Gary, Zhu Danhua (2010) “Effects of Stimulation Properties in Steady State Visual Evoked Potential Based Brain-Computer Interfaces”, *32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Buenos Aires, Argentina, August 31- September 4, 2010.

Birbaumer N., Ghanayim N., Hinterberger T., Iversen I., Kotchoubey B., Kübler A., Perelmouter J., Taub E. and Flor H. (1999), “A speeling device for the paralysed,” *Nature*, 398:297-298.

Birbaumer N. (2006), “Breaking the silence: Brain-Computer-interfaces (BCI) for communication and motor control”, *Psychophysiology*, vol. 43, pp. 517-532, 2006.

Blankertz Benjamin, Dornhege Guido, Krauledat Matthias, Müller Klaus-Robert, Kunzmann Volker, Losch Florian, and Curio Gabriel (2006), “The Berlin Brain-Computer Interface: EEG-Based Communication Without Subject Training,” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 14, no. 2, June 2006.

Cecotti H., (2010), “A Self-Paced and Calibration-Less SSVEP-Based Brain-Computer Interface Speller”, *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 2, pp. 127-133, 2010.

Cheng Ming, Gao Xiaorong, Gao Shangkai and Xu Dingfeng (2002) “Design and Implementation of a Brain-Computer Interface With High Transfer Rates”, *IEEE transactions on biomedical engineering*, Vol. 49, N°. 10, October 2002.

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.

Cincotti F., Bianchi L., Birch G., Guger C., Mellinger J., Scherer R., Schmidt R., Yañez Suarez O. and Schalk G. (2006), “BCI meeting 2005 - workshop on technology: Hardware and software,” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 14, no. 2, June 2006.

Craigmile P.F and King, W.M (2004), “Periodogram based tests for distortion product otoacoustic emissions,” *The Journal of the Acoustical Society of America*, Vol. 116, No.1, pp. 442-451.

Dasgupta Ranjan (2008) “Anatomy of RTOS and Analyze the Best-Fit for Small, Medium and Large Footprint Embedded Devices in Wireless Sensor Network”, *IEEE Computer Society, The Second International Conference on Sensor Technologies and Applications*, 2008.

Dewan, E. M. (1967). Occipital alpha rhythm eye position and lens accommodation. *Nature*, 214(5092), 975.

Diez Pablo F, Mut Vicente A, Avila Perona Enrique M and Laciár Leber Eric (2011) “Asynchronous BCI control using high frequency SSVEP”. *Journal of NeuroEngineering and Rehabilitation*, 2011 8:39.

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*.

Fernandes João M and Machado Ricardo J. (2007), “Teaching Embedded Systems Engineering in a Software-Oriented Computing Degree”, *37th ASEE/IEEE Frontiers in Education Conference*, October 10-13, 2007, Milwaukee, WI.

Flanagan J. L. (1960) “Models for approximating basilar membrane displacement”, *The Bell System Technical Journal*, Volume: 39, Issue:5, pages: 1163-1191.

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

Gao Xiaorong, Xu Dingfeng, Cheng Ming, and Gao Shangkai (2003) “A BCI-Based Environmental Controller for the Motion-Disabled”, *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 11, N°. 2, June 2003.

García P. A., Spinelli E. M., Vignoni R. y Guaraglia D. (2007) “Nodo programmable para redes de sensores distribuidos con aplicación en biomedicina”, XVI Congreso Argentino de Bioingeniería, V Jornadas de Ingeniería Clínica, San Juan, Argentina, 26 al 28 de Septiembre de 2007.

García P. A. (2008) “Redes de sensores distribuidos”, Thesis submitted for the master's degree in engineering. La Plata, Argentine, April 2008.

García P. A., Haberman M. and Spinelli E. M. (2010) “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.

García P.A, Spinelli E.M. y Toccaceli G.M., (2014.a), "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. a

Georgitzikis V., Akribopoulos O. and Chatziannakis I. (2012), “Controlling Physical Objects via the Internet using the Arduino Platform over 802.15.4 Networks”, *IEEE LATIN AMERICA TRANSACTIONS*, VOL. 10, NO. 3, APRIL 2012

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.

Haberman M., Toccaceli G, Spinelli E. y García P.A. (2010). “Arquitectura software de tiempo real para interfaz cerebro-computadora”, Anales del XXII Congreso Argentino de Control Automático AADECA 2010. Ciudad de Buenos Aires, 2010.

Hamblen James O. (2007) “INTRODUCTION TO EMBEDDED SYSTEMS USING WINDOWS EMBEDDED CE”, School of Electrical and Computer Engineering, Georgia Institute of Technology.

Hartenstein R (2004) “The Digital Divide of Computing”, *1st Conference on Computing Frontiers (CF '04)*, ACM Press, 2004, pages 357-62.

He Wei, Zhao Yue, Tang Haoyue, Sun Changyin and Fu Wei (2016), “A Wireless BCI and BMI System for Wearable Robots”, *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS*, VOL. 46, NO. 7, JULY 2016

Huggins J. E., Guger C., Allison B., Anderson C. W., Batista A., Brouwer A-M., Brunner C., Chavarriaga R., Fried-Oken M., Gunduz A., Gupta D., Kübler A., Leeb R., Lotte F., Miller L. E., Müller-Putz G., Rutkowski T., Tangermann M. & Thompson D. E., (2014), “(2014) Workshops of the Fifth International Brain-Computer Interface Meeting: Defining the Future, Brain-Computer Interfaces”, 1:1, 27-49, DOI: 10.1080/2326263X.2013.876724.

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.

Ishikawa Y., Takata M. and Joe K. (2012) “Constitution and phase analysis of alpha waves”, Biomedical Engineering International Conference (BMEiCON), 2012, DOI: 10.1109/BMEiCon.2012.6465482.

Jia Chuan, Gao Xiaorong, Hong Bo and Gao Shangkai (2011) “Frequency and Phase Mixed Coding in SSVEP-Based Brain–Computer Interface”, *IEEE transactions on biomedical engineering*, Vol. 58, N° 1, January 2011.

Kemp, D.T. (1978) “Stimulated acoustic emissions from within the human auditory system,” *The Journal of the Acoustical Society of America*, Vol. 64, No. 5, pp. 1386-1391.

Kemp, D.T (1997) “Understanding and Using Otoacoustic Emissions,” Otodynamics Ltd.

Knight R.D. and Kemp, D.T (2000) “Indications of different distortion product otoacoustic emission mechanisms from a detailed f1, f2 area study”, *The Journal of the Acoustical Society of America*, Vol. 107, No. 1, pp. 457-473.

Kuo-Kai Shyu, Po-Lei Lee, Ming-Huan Lee, Yun-Jen Chiu (2010) “The Low-Cost Implement of a Phase Coding SSVEP-Based BCI System”, *Electronics, Circuits, and Systems (ICECS), 2010, 17th IEEE International Conference*, Digital Object Identifier: 10.1109/ICECS.2010.5724573, Publication Year: 2010 , Page(s): 559 – 562.

Laplante Phillip A (1992) “Real-Time Systems Design and Analysis,” *IEEE Press, USA*. ISBN 0-7803-0402-0.

Leeb R., Settgast V., Fellner D., Pfurtscheller G. (2007) “Self-paced exploration of the Austrian National Library through thought”, *International Journal of Bioelectromagnetism* , Vol. 9, No.4, pp. 237 - 244, 2007 .

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

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.

Marshall D., Coyle D., Wilson S., and Callaghan M. (2013) “Games, Gameplay, and BCI: The State of the Art”, *IEEE Trans. on computational intelligence and ai in games*, vol. 5, no. 2, pp. 82-99, 2013.

Millán J. R., Renkens F., Mouriño J. and Gerstner W. (2004) “Noninvasive Brain-Actuated Control of a Mobile Robot by Human EEG,” *IEEE transactions on biomedical engineering*, vol. 51, no. 6, June 2004

Millán J., Rupp R., Müller-Putz G., 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*, DOI: 10.3389, 2010.

Muñoz Frías José Daniel (2008) “Sistemas Empotrados de Tiempo Real, una introducción basada en FreeRTOS y el microcontrolador ColdFire MCF5282”.

Neumann N. and Kubler A. (2003) “Training locked-in patients: a challenge for the use of brain-computer interfaces”, *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 11, issue 2, pp. 169-172, 2003.

Nguyen Tran, Anh Bao and Tan Su-Lim (2009) “REAL-TIME OPERATING SYSTEMS FOR SMALL MICROCONTROLLERS”, IEEE Computer Society.

Ortner R., Allison B. Z., Korisek G., Gaggli H. and Pfurtscheller G. (2011) “An SSVEP BCI to Control a Hand Orthosis for Persons with Tetraplegia”, *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 19, issue 1, pp. 1-5, 2011.

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

Pallás Areny Ramón, Webster John G (1999) “Analog Signal Processing”, Wiley-Interscience, John Wiley & Sons, Inc. 1999. ISBN 0-471-12528-8.

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.

Pazos Carlos (2013) “Understanding a Key Trend in Embedded System Design”, National Instruments, Q1 2013 issue of Instrumentation Newsletter, September-2013.

Perelmouter J. and Birbaumer N. (2000) “A binary spelling interface with random errors”, *IEEE Trans. Rehab. Eng.*, vol. 8, pp. 227–232, June 2000.

Pfurtscheller G., Neuper Ch., Flotzinger D., Pregenzer M. (1997) “EEG-based discrimination between imagination of right and left hand movement”, *Electroencephalography and clinical Neurophysiology*, 103 (1997) 642-651, Accepted for publication: 7 July 1997

Pfurtscheller G., Lopes da Silva F.H. (1999) “Event-related EEG/MEG synchronization and desynchronization: basic principles,” *Clinical Neurophysiology*, 110 (1999) 1842-1857, Accepted May 1999.

Pfurtscheller G., Allison B., Brunner C., Bauernfeind G., Solis-Escalante T., Scherer R., Zander T., Müller-Putz G., Neuper C., and Birbaumer N. (2010) “The hybrid BCI,” *Frontiers in Neuroscience*, vol. 4, art. 42, pp. 1-11.

Postelnicu C. and Talaba D. (2012) “P300-Based Brain-Neuronal Computer Interaction for Spelling Applications”, *IEEE Trans. on Biomedical Engineering*, vol. 60, issue 2, pp. 534 – 543, 2012.

Proakis John G, Manolakis Dimitris G (1996) "Digital Signal Processing, Principles, Algorithms and applications", Prentice Hall International Inc, ISBN 0-13-394338-9.

Ping Wang (2008) "Research on the Embedded System Teaching", *2008 International Workshop on Education Technology and Training & International Workshop on Geoscience and Remote Sensing*.

Ridolfi Pablo O (2016) "Extension of the FreeOSEK RTOS for Asymmetric Multiprocessor Systems", *2016 IEEE Biennial Congress of Argentina (ARGENCON)*.

Robinette, M.S. and Glatke T.J. (2007) "Otoacoustic emissions: clinical applications," *Thieme Medical Publishers, Inc, New York, USA, 2007, ISBN 10:1-58890-411-3*.

Rodríguez-Bermúdez G, García-Laencina PJ, Roca-González JL, Roca-González J, Roca-Dorda J (2012) "Diseño de sistemas BCI adaptativos mediante una selección eficiente de características para discriminantes lineales". XXX Congreso Anual de la Sociedad Española de Ingeniería Biomédica (CASEIB 2012).

Salazar Jordi (2001) "Procesadores digitales de señal (DSP), Arquitecturas y criterios de selección", *Mundo electronico, n° 314, pp. 46-57*.

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*.

Schwartz M. and Shaw L (1975) "Signal Processing," McGraw Hill, ISBN 0070556628.

Short Michael (2008) "Development Guidelines for Dependable Real-Time Embedded Systems", *2008 IEEE/ACS International Conference on Computer Systems and Applications*.

Shyu Kuo-Kai, Lee Po-Lei, Lee Ming-Huan, Chiu Yun-Jen (2010) "The Low-Cost Implement of a Phase Coding SSVEP-Based BCI System", *Electronics, Circuits, and Systems (ICECS), 2010, 17th IEEE International Conference, Digital Object Identifier: 10.1109/ICECS.2010.5724573, Publication Year: 2010, Page(s): 559 - 562*

Spinelli E. M., Mayosky M. A. y Martinez N. H. (2000) "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.

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

Spinelli E., Pallàs-Areny R., Mayosky M. (2003) "AC-Coupled Front-End for biopotential measurements," *IEEE Transactions on Biomedical Engineering. Vol. 50, No. 3, pp. 391-395, 2003*.

Stanciu Alexandra, Balan Titus, Sandu Florin and Gerigan Carmen (2017) "Reconfigurable platform for embedded systems teaching". *2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME)*.

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.

Tedeschi Antonio, Calcaterra Stefano and Benedetto Francesco (2017) “Ultrasonic Radar System (URAS): Arduino and Virtual Reality for a Light-Free Mapping of Indoor Environments”, *IEEE SENSORS JOURNAL*, VOL. 17, NO. 14, JULY 15, 2017.

Tianyou Y., Jun X., Fangyi W., Zhang R., Zhenghui G., Cichocki A. and Yuanqing L. (2015) “Enhanced Motor Imagery Training Using a Hybrid BCI with Feedback”, *IEEE Trans. on Biomedical Engineering*, vol. 62, issue 7, pp. 1706–1717, 2015.

Van Hemert C.A. (2009) “The impact of visual distractions in SSVEP-based BCI”, 11th Twente Student Conference on IT, Enschede 29th June, 2009, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.

Vidal J.-J. (1973) “Toward direct brain-computer communication”. *Annual review of Biophysics and Bioengineering*, 2(1):157–180.

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.

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.

Wolpaw J. R., Birbaumer N., McFarland D. J., Pfurtscheller G., and Vaughan T. M. (2002) “Brain-computer interfaces for communication and control,” *Clin. Neurophysiology*, vol. 113, pp. 767–791.

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

Yu Chengjing, Ma Xudong, Fang Fang, Qian Kun, Yao Shun and Zou Yanping (2017) “Design of Controller System for Industrial Robot Based on RTOS Xenomai”, *12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*.

Zhao Hai-bin, Wang Hong, Li Chun-sheng and Li Yun-gong (2009) “Brain-Computer Interface Design Based on Slow Cortical Potentials using Matlab/Simulink”, *Proceedings of the 2009 IEEE International Conference on Mechatronics and Automation*, August 9 - 12, Changchun, China.

Zhu Danhua, Bieger Jordi, Garcia Molina Gary and Aarts Ronald M. (2010) “A Survey of Stimulation Methods Used in SSVEP-Based BCIs”, *Hindawi Publishing Corporation Computational Intelligence and Neuroscience*, Volume 2010, Article ID 702357, 12 pages, doi:10.1155/2010/702357.

Zulzilawati Jusoh, Hasnorhafiza Husni, Syila Izawana Ismail, Suziana Omar and Rina Abdullah (2017) “Implementation of Embedded System Design in Student’s Final Year Project Using Problem Based Learning Approach”. *2017 IEEE 9th International Conference on Engineering Education (ICEED)*, Kanazawa, Japan, November 9-10.
