



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



Electroencefalógrafo inalámbrico de bajo consumo

MEMORIA DE PROYECTO PRESENTADA A LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE LA REPÚBLICA POR

Martín Causa, Franco La Paz, Santiago Radi

EN CUMPLIMIENTO PARCIAL CON LOS REQUERIMIENTOS PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO ELECTRICISTA.

TUTOR

Julián Oreggioni Universidad de la República

TRIBUNAL

Gabriel Gómez Universidad de la República

Julio Pérez Universidad de la República

Julián Oreggioni Universidad de la República

Montevideo
domingo 19 noviembre, 2017

Electroencefalógrafo inalámbrico de bajo consumo, Martín Causa, Franco La Paz,
Santiago Radi.

Esta tesis fue preparada en L^AT_EX usando la clase iietesis (v1.1).
Contiene un total de 197 páginas.
Compilada el domingo 19 noviembre, 2017.
<http://iie.fing.edu.uy/>

Agradecimientos

En la vida es casi imposible hacer las cosas solos. Siempre estamos necesitando de ayuda, de apoyo, de alguien con más experiencia que nos pueda guiar; y esto también se refleja en una actividad tan complicada pero interesante como es un proyecto de fin de carrera.

Queremos agradecer de corazón a todas las personas que nos ayudaron directa e indirectamente a realizar este proyecto. Las personas que aportaron su conocimiento para salir más rápido en algunos momentos que estábamos complicados, y también a las personas que sin necesidad de aportar conocimiento, nos dieron el espacio y el tiempo que necesitábamos para continuar avanzando por nosotros mismos.

En esta sección no podemos dejar de mencionar a nuestras familias; nuestro fundamental sustento para llegar hasta este lugar y poder estar hoy realizando el proyecto de fin de carrera. Nos brindaron todo para que nosotros contemos con todas las herramientas y el tiempo necesario para dedicarnos a este proyecto, librándonos de otras responsabilidades.

Nuestras novias y nuestros amigos, otro sustento y apoyo fundamental, que soportaron días en los que estábamos complicados ayudándonos además en lo que estaba a su alcance, aportando sus conocimientos y formando parte de esto que es tan importante para nosotros.

En esta facultad, conocimos muchos amigos, ellos nos ayudaron a lo largo de toda la carrera y algunos perduran hasta el día de hoy. En especial, queremos mencionar a Esteban Risso, que cuando uno de nosotros estuvo sin computadora, le brindó la posibilidad de quedarse trabajando en su casa todos los días hasta las 3 de la mañana.

Algunos de ellos son de la misma carrera, y por tanto, pudieron dar además de una apoyo moral, volcar su conocimiento. En esta categoría queremos agradecer a Nicolás Márquez, quien colaboró en la interfaz de usuario, en consejos para la elección de la radio y otras dificultades menores; a Florencia Blasina quien fue partícipe en dificultades referentes a la placa y al manejo de programa Eagle; a Federico Nin y a Federico Silva, por ser compañeros de una asignatura, cuyo pro-

yecto de fin de curso fue un 50 % del software embebido de nuestro proyecto; y a Federico Favaro, que brindó aportes en muchísimas áreas de nuestro proyecto. Fueron tantos sus aportes que merece un capítulo aparte, pero en resumen, por estar trabajando con nuestros clientes del proyecto brindó información sobre funcionalidades del microcontrolador utilizado, aspectos de velocidad en las comunicaciones del microcontrolador con los periféricos, configuración de la radio y el Router, características del algoritmo de compresión, etc.

Varios docentes del instituto participaron del proyecto de forma directa. Remarcamos agradecimientos para Sebastián Fernández, por ser el encargado de soldar los integrados RHD2132 a la placa que tuvimos que diseñar; a Ignacio Ramírez por realizar la compra de la placa AFE; a Pedro Arzuaga por darnos una charla y brindarnos resúmenes de las normativas que los equipos médicos (como lo es nuestro proyecto de fin de carrera) deben cumplir; a Fernando Silvera y nuevamente a Pedro Arzuaga por ser encargados de la revisión de la placa AFE diseñada por el proyecto; a Leo Steinfeld por colaborar innumerables veces en aspectos asociados al software embebido y también ser un colaborador fundamental en la elección y compra de la radio CC3100, que fue la finalmente utilizada en nuestro proyecto; y a Juan Pablo Oliver, por siempre preocuparse por la evolución de nuestro proyecto y ayudarnos con sus aportes en los momentos necesarios. En especial por permitirnos dedicarle la mayor parte del tiempo, por las recomendaciones asociadas a la elección de la batería, por ser uno de los encargados de la revisión de la placa diseñada y por brindarnos un lugar para trabajar en ciertos momentos.

Por último, aunque no menos importante, a nuestro tutor Julián Oreggioni, que estuvo incansables veces a nuestra disposición más allá de las reuniones establecidas con antelación para ayudarnos. Por brindarnos siempre soluciones o posibilidades acertadas que nos guiaron de manera inmejorable al objetivo final. Sin duda que nos enriquecimos con una muy invaluable experiencia a través de sus consejos.

Para todos ellos dedicamos nuestro trabajo. Todos aportaron su granito de arena para que esto fuese posible, y estaremos eternamente agradecidos.

“Uno puede devolver un préstamo de oro, pero está en deuda de por vida con aquellos que son amables.”

Proverbio popular.

Resumen

El proyecto consistió en desarrollar un electroencefalógrafo inalámbrico de bajo consumo y tamaño reducido, desde la adquisición de la señal hasta su transmisión inalámbrica y visualización en un PC. El hecho de que el sistema sea inalámbrico es para que el paciente pueda moverse libremente por un tiempo razonable en un entorno de corta distancia, permitiendo extender el campo de aplicación de los estudios tradicionales de electroencefalografía (EEG).

El sistema se compone de un módulo remoto y un PC. El módulo remoto se encarga de adquirir las señales de EEG, amplificarlas, filtrarlas, digitalizarlas, procesarlas y enviarlas de forma inalámbrica al PC. El PC recibe los datos transmitidos por el módulo remoto y a través de una interfaz de usuario despliega las señales adquiridas. La interfaz de usuario permite controlar la operación, configurar el módulo remoto y almacenar los datos recolectados.

El módulo remoto consta de un front-end analógico con dos integrados RHD2132 de Intan Technologies, un microcontrolador MSP432 de Texas Instruments y una radio Wi-Fi CC3100 de Texas Instruments. Los dos RHD2132 permiten adquirir, amplificar, filtrar y digitalizar hasta 64 señales biológicas, todas relativas a una referencia común. Posee un filtro con ancho de banda programable, alto rechazo al modo común, alta impedancia y bajo ruido a la entrada, y un conversor A/D de 16 bits. El MSP432 es un microcontrolador ARM de muy bajo consumo (8,5mA en modo activo, 70µA en modo bajo consumo), cuenta con un reloj interno de 48MHz, unidad de punto flotante y se encarga de controlar el funcionamiento del módulo remoto.

A través de una interfaz de usuario implementada desde un PC, la cual se maneja a través de MatLab, se puede programar la frecuencia de muestreo, el ancho de banda del filtro, la cantidad de canales de adquisición o bien incluir hasta 6 canales de sincronismo para estudios de reacción a estímulos.

Se caracterizó el sistema mediante una serie de pruebas, entre ellas, se testeó la comunicación inalámbrica entre el módulo remoto y el PC, se realizaron mediciones de las tareas principales del software embebido en el microcontrolador determinando las limitaciones del sistema, se determinaron las frecuencias máximas a las cuales funciona el sistema corroborando los tiempos medidos en el software,

se midieron los consumos de los distintos componentes en distintos estados y se midió la autonomía del sistema.

Como producto final, se obtuvo un sistema capaz de adquirir hasta 29 señales (fácilmente extendible a 64 señales y 6 señales de sincronismo) durante más de 24 horas, con una frecuencia de muestreo programable entre 92 Hz y 9850 Hz. El sistema admite 4 tipos de configuraciones: los 29 canales, 21 canales predefinidos, 4 canales predefinidos, o 1 canal a elección entre los 29 posibles. Posee un filtro pasabanda programable; la frecuencia de corte inferior puede variar entre 0,1 Hz y 500 Hz y la frecuencia de corte superior puede variar entre 100 Hz y 20 kHz. El bloque analógico del sistema tiene una impedancia de entrada de $1,3 G\Omega$, un ruido intrínseco menor que $2,4 \mu V_{rms}$ y un rechazo al modo común de $82 dB$. El módulo remoto (parte del sistema que se ubica en el paciente) se encuentra dentro de una carcasa de dimensiones $12 cm \times 8 cm \times 5 cm$. La distancia máxima entre el módulo remoto y el PC es 12 metros.

Tabla de contenidos

Agradecimientos	I
Resumen	I
1. Introducción	1
1.1. Motivación y clientes del proyecto	1
1.2. Estado del arte	2
1.3. Características de las señales EEG	3
1.4. Objetivos y especificaciones funcionales del proyecto	5
1.5. Descripción global del funcionamiento	7
1.6. Organización del documento	8
2. Elección de los componentes de hardware	11
2.1. AFE (Analog Front End)	11
2.1.1. ADS1299	11
2.1.2. RHD2132	11
2.1.3. RHD2164	12
2.2. Microcontrolador	13
2.3. Sistema de comunicación inalámbrico (Radio)	14
2.4. Alimentación y estimación de consumo	16
2.4.1. Estimación del consumo del módulo remoto	16
2.4.2. Elección de la batería	17
2.5. Electrodo	18
2.5.1. Electrodo seco	18
2.5.2. Electrodo húmedo	19
3. Diseño del AFE	21
3.1. Funciones a utilizar del chip RHD2132	22
3.2. Comunicación con el microcontrolador	23
3.3. Componentes externos al chip RHD2132	24
3.4. Conexión de la placa AFE con el microcontrolador	25
4. Diseño del software embebido	27
4.1. Descripción general del funcionamiento del Software Embebido	28
4.2. Interrupciones	30
4.3. Estructura en módulos	31

Tabla de contenidos

4.4.	Módulo SPI	33
4.4.1.	Buffers del módulo SPI	35
4.4.2.	Funciones del módulo SPI	35
4.4.3.	Interrupciones del módulo SPI	36
4.5.	Módulo AFE	37
4.5.1.	Funciones de bajo nivel de abstracción	37
4.5.2.	Funciones de alto nivel de abstracción	39
4.6.	Módulo Process	41
4.7.	Módulo main	41
4.8.	Funcionamiento del módulo remoto en configuración	42
4.9.	Funcionamiento del módulo remoto durante un estudio	44
5.	Transmisión y recepción de datos	49
5.1.	Opciones de protocolos	50
5.2.	Problemas con Bluetooth	51
5.2.1.	Bluetooth 4.0 + LE	51
5.2.2.	Bluetooth 4.0 + EDR	51
5.2.3.	Bluetooth 4.0 + HS	52
5.3.	Wi-Fi	52
6.	Interfaz de usuario	55
6.1.	Protocolo de comunicación	55
6.2.	Descripción general del funcionamiento de la interfaz de usuario	57
6.3.	Inicialización de la interfaz y conexión con el módulo remoto	59
6.4.	Funcionamiento menú principal	59
6.5.	Configurar parámetros	60
6.5.1.	Algoritmo de compresión	61
6.5.2.	Frecuencia de muestreo	61
6.5.3.	Cantidad de canales	61
6.5.4.	Señales de sincronismo	61
6.5.5.	Frecuencia de corte superior	62
6.5.6.	Frecuencia de corte inferior	62
6.5.7.	Eliminación de offset	62
6.5.8.	Distorsión absoluta máxima por muestra	62
6.5.9.	Tiempo de ventana a graficar	63
6.5.10.	Guardar configuración	63
6.5.11.	Enviar	63
6.6.	Ventana gráfica	63
6.7.	Respuesta de la interfaz de usuario ante pérdidas del sistema	65
7.	Montaje del módulo remoto	69
7.1.	Electrodos y casco	69
7.2.	Carcasa	70
7.3.	Instalación en el paciente	71

8. Pruebas y análisis de resultados	77
8.1. Pruebas de comunicación	77
8.1.1. Medición del throughput útil en función del tamaño de paquete	77
8.1.2. Medición del throughput en función de la distancia	78
8.1.3. Medición del throughput útil variando el tipo de comunicación entre Router y PC	79
8.2. Medición de tiempos del software embebido	82
8.2.1. Tiempo de envío de palabra de 16 bits por SPI	82
8.2.2. Tiempo de una adquisición	83
8.2.3. Tiempo de envío de datos del microcontrolador a la radio .	86
8.3. Prueba del sistema	87
8.3.1. Frecuencia máxima en función de la cantidad de canales . .	87
8.3.2. Medición del consumo	91
8.3.3. Autonomía del sistema	93
8.3.4. Prueba de punta a punta	95
8.3.5. Testeo del funcionamiento de todos los canales	97
8.4. Resumen de las pruebas	97
9. Conclusiones	101
9.1. Conclusiones generales	101
9.2. Comparación del sistema con otros dispositivos EEG comerciales .	103
9.3. Trabajos futuros	105
9.4. Principales problemas enfrentados y lecciones aprendidas	107
Anexos	109
A. Explicación del no uso de ADS1299	109
A.1. ADS1299 sin amplificación externa	110
A.2. ADS1299 con amplificación externa	111
A.3. Demostración de las ecuaciones A.2 y A.3	111
B. Información complementaria a la elección de componentes	113
B.1. Tablas con todas las opciones de radio manejadas	113
B.2. Router para la comunicación Wi-Fi	114
B.3. Conversores DC/DC	115
B.3.1. Elección del convertor DC/DC para los chips RHD2132 . .	115
B.3.2. Elección de convertor DC/DC para los bloques MCU y Radio	116
B.4. Test de algunas de las baterías	117
B.5. Cable para electrodos	118
B.6. Casco	119
B.7. Carcasa	119
B.8. Cargador de baterías	120

Tabla de contenidos

C. Low Voltage Differential Signal (LVDS)	121
C.1. Funcionamiento de los dos tipos de comunicación SPI	121
C.2. Ventajas del uso de LVDS	123
C.3. Desventajas del uso del LVDS	124
D. Conexión de cada pin del AFE	125
D.1. Conexión de los convertidores DC / DC	128
D.1.1. Conexionado del ADP150	128
D.1.2. Conexionado del TPS736	129
D.2. Lista de componentes a soldar en la placa	130
D.3. Esquemático de la placa AFE	131
E. Funcionamiento del test de impedancia de electrodos	133
E.1. Generación de la corriente AC	134
E.2. Elección de la corriente AC	135
F. Impedancia característica de las pistas	137
G. Información complementaria del diseño del software embebido	139
G.1. Elección de Round-Robin frente a encolado de funciones	139
G.2. Diagramas de flujo de las principales funciones de SPI.h y AFE.h .	140
G.3. Módulo queue	144
G.4. Módulo Timer	147
G.4.1. Timer A0	147
G.4.2. Timer A1	148
G.5. Módulo Radio	148
G.5.1. Radio_init()	148
G.5.2. Radio_receiveData	150
G.5.3. Radio_sendData	150
G.5.4. Radio_close	150
G.6. Módulo CTRLerrores	151
G.7. Cálculo del tamaño de paquete a enviar	154
H. Información complementaria a la interfaz de usuario	155
H.1. Menús interfaz de usuario	155
H.2. Formato de los comandos enviados al módulo remoto	155
H.3. Recomendaciones ante fallas del sistema	157
I. Información complementaria de las pruebas	159
I.1. Pseudocódigo de la medición del throughput en función del tamaño de paquete	159
I.2. Comparación entre el tiempo de conversión (t_{conv}) medido y el es- perado	160
I.3. Dedución de la ecuación 8.1	161
I.4. Otras medidas de tiempos	163
I.5. Deduciones de la ocurrencia de pérdida de datos	163

Tabla de contenidos

I.6. Deducción de la ecuación I.2	165
I.7. Circuitos para la medida de consumo	166
I.8. Deducción de la ecuación 8.4	168
J. Seguridad en equipos médicos	169
Referencias	171
Índice de tablas	175
Índice de figuras	178

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 1

Introducción

1.1. Motivación y clientes del proyecto

El avance de la neurociencia y la electrónica en los últimos años ha hecho posible pensar en dispositivos inalámbricos que midan señales de EEG (electroencefalografía) en un diverso rango de aplicaciones. Existen dispositivos para detectar si una persona se está durmiendo, esto es particularmente importante en personas que recorren largos trayectos conduciendo vehículos ya que muchos accidentes se deben a eventos de este tipo. Por otro lado, existen videojuegos que funcionan con “control mental” usando dispositivos que procesan señales de EEG. Finalmente, es factible pensar que en los próximos años se desarrollen dispositivos EEG inalámbricos para controlar una prótesis o para detectar con antelación si un paciente está por sufrir un ataque de epilepsia. Esto último permitiría que una persona con esta enfermedad pueda realizar actividades que actualmente no puede realizar con seguridad, como por ejemplo conducir un automóvil o nadar. El impacto que genera el avance de la neurociencia y todo el campo en el cual se puede aplicar es una importante motivación de este trabajo.

El Dr. Ángel Caputi, reponsable del Departamento de Neurociencias Integrativas y Computacionales del Instituto de Investigaciones Biológicas Clemente Estable (IIBCE) realiza habitualmente “registros de señales biológicas, en diversos seres vivos. Esto genera una demanda constante de innovación en las técnicas de adquisición, y con ello la necesidad de nuevos y mejores dispositivos que permitan alcanzar dicho fin” [1]. El Dr. Caputi ha tenido inconvenientes con varios equipos comerciales que ha utilizado, como por ejemplo equipos de DataWave Technologies [2], NeuraLynx [3] y Natus [4]. El equipo de DataWave ha generado problemas con inmunidad al ruido e interferencias y el equipo de NeuraLynx no ofrece soluciones inalámbricas, y en general ninguno resuelve adecuadamente el registro simultáneo de señales de estímulo (canales de sincronismo) [5]. Desde hace varios años el Dr. Caputi y el Departamento de Electrónica vienen colaborando en estos temas. En particular, el Dr. Caputi es co-tutor de la tesis de doctorado de Julián Oreggioni, donde se abordan los aspectos antes mencionados y se fabricó un ampli-

Capítulo 1. Introducción

ficador neural integrado. En este contexto mediante un proyecto de fin de carrera, se desarrollase una plataforma de bajo consumo (basada en un microcontrolador), capaz de tomar las señales del amplificador antes mencionado y enviarlas en forma inalámbrica a un PC.

Por otro lado, el grupo de Teoría de la Información (INCO-IIE, Facultad de Ingeniería) y el Departamento de Electrónica (IIE, Facultad de Ingeniería) realizan un proyecto con financiación de la Comisión Sectorial de Investigación Científica de la UdelaR (Proyecto CSIC I+D) cuyo principal objetivo es portar un algoritmo de compresión de señales de EEG (ACEEG)¹ implementado en una plataforma de alto nivel (basada en un PC) a una plataforma de bajo consumo; y era de su interés, que mediante un proyecto de fin de carrera, se pudiera probar su algoritmo en una plataforma de bajo consumo.

Reuniendo todos los intereses antes mencionados, surge la motivación de realizar el proyecto wEEG (Wireless Electroencephalography), que consiste en desarrollar un electroencefalógrafo inalámbrico de bajo consumo y de tamaño reducido. Asimismo, se espera que en una segunda etapa el producto implementado permita incluir el algoritmo ACEEG y el amplificador neural integrado antes mencionado.

1.2. Estado del arte

La adquisición y análisis de señales biopotenciales² han sido de interés para la ciencia desde hace varios siglos. En particular, la adquisición de las señales de EEG tuvo su inicio en 1875, cuando Richard Caton logró medir la actividad cerebral de forma continua y espontánea en perros utilizando un galvanómetro [7] [8]. Luego, el 6 de julio de 1924, Hans Berger realiza el primer registro de actividad eléctrica en el cerebro de un joven de 17 años, a través de una trepanación³, utilizando un galvanómetro de cuerda [8].

De aquí en más, se han logrado múltiples avances tanto en el tipo de registros, así como en la variedad de dispositivos empleados. A partir de los avances en la electroencefalografía y la electrónica, en 1968, el físico David Cohen permite obtener información de la actividad cerebral de forma totalmente no invasiva. La figura 1.1 muestra como se realizan actualmente los estudios de electroencefalografía.

La neurociencia junto a la electroencefalografía se ha expandido, y esto ha llevado al desarrollo del estudio de las emociones, de manipulación de prótesis, e incluso juegos controlados con la mente. Ejemplos de esto último son el Force Trainer de Star Wars [10], el dispositivo creado por el instituto de tecnología del norte

¹Para más información sobre el funcionamiento del algoritmo ACEEG ver [6].

²Es un potencial eléctrico debido a procesos bioquímicos, que puede medirse entre puntos de células, tejidos u organismos.

³Perforación del cráneo con un instrumento quirúrgico.

1.3. Características de las señales EEG



Figura 1.1: Estudios de electroencefalografía actuales (imagen extraída de [9]).

de Alberta (NAIT) en Canadá [11], y los dispositivos creados por VROne [12]. A su vez, la miniaturización de la electrónica y la reducción de costos y de consumo eléctrico, necesarios para este tipo de dispositivos, permitió que pudiesen ser cada vez más accesibles, al punto de existir páginas libres que detallan como construir equipos domésticos capaces de adquirir las señales más estándares de EEG. Ejemplo de estas páginas son “OpenEEG” [13] y “Open Ephys” [14].

Otros antecedentes interesantes son los presentados por NESIA (Neural Signal Acquisition) [1] y los trabajos [15] y [16]. NESIA es una plataforma creada por un proyecto de fin de carrera del IIE en 2013 que adquiere y procesa señales neuronales (entre ellas EEG) en forma inalámbrica. El trabajo [15] es un sistema discreto de registro de señales neurales (no solo para señales EEG) basado en FPGAs, sin compresión. Y el trabajo [16] es un sistema discreto de registro de señales EEG basado en MSP430 y que presenta un esquema de compresión.

Existen actualmente dispositivos inalámbricos de señales de EEG disponibles para su comercialización. La tabla 1.1 muestra alguno de los dispositivos estudiados y sus principales características. El “-” en todas las tablas se deben a que no se encontró una información oficial de cada fabricante. Para mayor información de estos dispositivos visitar: para EPOC/EPOC+ [17], para Insight [18], para ENOBIO [19], para BioRadio [20], para Cognionics [21].

1.3. Características de las señales EEG

Las señales de EEG más comunes varían entre los $10\mu V$ y los $200\mu V$ y tienen un rango de frecuencia entre $0,1Hz$ y $100Hz$. Se clasifican en 5 tipos según su frecuencia:

⁴Analog Front End

Capítulo 1. Introducción

	EPOC/EPOC+	Insight	ENOBIO	BioRadio	COGNIONICS
		Secos	Secos	-	Secos
Características del AFE ⁴	Electrodos	-	-	-	-
	Impedancia de entrada	-	>1 GΩ	500 MΩ	-
	Rango de entrada	-	-	1μV-2V	-
	Ancho de banda	0,2 - 45 Hz (con notch en 50 Hz y 60 Hz)	-	0-125 Hz	0-131 Hz @500 sps 0-262 Hz @1000 sps
	Ruido a la entrada	-	-	<1 μV RMS	0,7 μV RMS 1-50 Hz
	CMRR	-	-	-	100 dB
	Número de canales	14	5	8,20,32	4 diferenciales 8 single-ended 16,24,32 y 64
Sincronismo	0	2	8,20,32	0	8
Características digitales	Tasa de muestreo	128 sps (2048 Hz interno, 16 × 128Hz)	500 sps	Programable entre 250 sps a 16000 sps	500 sps @64 canales 1000 sps @32 canales
	Resolución	0,51μV - 14 bits	0,05 μV - 24 bits	30, 5μV - 16 bits	-
Características generales	Autonomía	12 Hs	16 Hs	8 Hs	6 hs inalámbrico, 10 hs con SD Card
	Dimensiones (LAH)	-	6 × 8,5 × 2cm ³	10 × 6 × 2cm ³	-
	Peso	-	65 g	-	350 g
Características de la comunicación	Protocolo	Propietario de banda en 2,4 Ghz + Bluetooth 4,0 LE	Propietario de banda en 2,4 Ghz + Bluetooth 4,0 LE	Bluetooth 2,1	Bluetooth optimizado para alta velocidad
	Throughput	-	-	200 kbps	-
Rango de transmisión	-	-	-	100 m	-

Tabla 1.1: Estudio de mercado de electroencefalógrafos inalámbricos comerciales.

1.4. Objetivos y especificaciones funcionales del proyecto

Ondas delta: oscilan entre $0,1Hz$ y $4Hz$ y están asociadas a etapas de sueño profundo sin soñar, daño cerebral y coma [22].

Ondas theta: oscilan entre $3,5Hz$ y $7,5Hz$ y están asociadas a las primeras etapas de sueño. Se generan tras la interacción entre los lóbulos temporal y frontal [23].

Ondas alfa: oscilan entre $8Hz$ y $13Hz$ y están asociadas a períodos de relajación, con los ojos cerrados, pero despierto [24].

Ondas beta: oscilan entre $12Hz$ y $30Hz$ y están asociadas a estados de vigilia [25].

Ondas gamma: oscilan entre $25Hz$ y $100Hz$ y podrían estar asociadas a estados de conciencia, aunque no hay un acuerdo unánime con respecto a esto [26].

1.4. Objetivos y especificaciones funcionales del proyecto

El objetivo del proyecto es diseñar, fabricar y testear un electroencefalógrafo inalámbrico: desde la adquisición de la señal hasta su transmisión y visualización en un PC. Se requiere que sea de bajo consumo, tamaño reducido y que permita integrar el algoritmo ACEEG mencionado en la sección 1.1. Que el sistema sea inalámbrico tiene el fin de que el estudio sea cómodo y de que el paciente pueda moverse libremente por un tiempo razonable en un entorno de corta distancia, permitiendo extender el campo de aplicación de los estudios de EEG.

El sistema se compone de un módulo remoto y una interfaz de usuario que corre en un PC.

El **módulo remoto** (ver figura 1.2) se puede localizar en la cabeza o en la cintura del paciente al que se le realizará el estudio y está compuesto a su vez de tres bloques bien diferenciados. Un primer bloque llamado **adquisidor**, consta de electrodos (encargado de la transducción de la señal), de un casco (encargados del sostén mecánico y la ubicación espacial de los electrodos) y el AFE (Analog Front End) que es un circuito de acondicionamiento de las señales analógicas que utiliza filtros, amplificadores operacionales y conversores A/D que digitalizan las señales relevadas. Un segundo bloque llamado **procesador**, se basa en un microcontrolador (MCU) que se encarga del procesamiento de las señales digitalizadas y de procesar comandos provenientes del PC. El procesamiento en una primera etapa será enviar los datos crudos (las muestras adquiridas directamente), dejando previsto una arquitectura que en una segunda etapa permita realizar la compresión de los datos mediante el ACEEG. Esto permitiría transmitir la misma información en menos bytes, reduciendo el consumo. Finalmente, un tercer bloque llamado **radio**, se encarga de la transmisión inalámbrica al PC. El módulo remoto debe ser de

Capítulo 1. Introducción

tamaño reducido y autonomía suficiente.

En el **PC**, corre una interfaz de usuario que permite controlar el estudio, configurar el módulo remoto, almacenar los datos recibidos y mostrarlos en tiempo real.

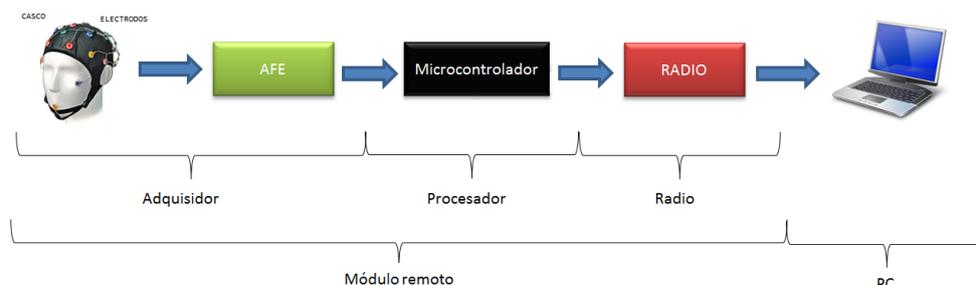


Figura 1.2: Diagrama de los bloques del sistema.

Luego de reuniones con los clientes del proyecto en las que se analizaron las necesidades de cada uno, se determinaron los requerimientos aceptables e ideales a cumplir por el sistema a diseñar. Estos requerimientos se presentan en cinco tablas (de tabla 1.2 a tabla 1.6).

Especificaciones		Caso aceptable	Caso ideal
Número de canales	Mayor o igual que	16	64
Autonomía	Mayor que	12 horas	24 horas
Peso del módulo remoto	Menor que	300 gramos	100 gramos
Tamaño físico de módulo remoto (LAH) ⁵	Menor que	20cm × 15cm × 9cm	13cm × 9cm × 7cm
Retardo de visualización ⁶ sin compresión	Menor que	1 s	500 ms
Tipo de electrodos		Húmedos	Secos
Presupuesto ⁷	Menor o igual que	US\$ 1300	US\$ 1000

Tabla 1.2: Requerimientos generales.

Especificaciones		Caso aceptable	Caso ideal
Frecuencia de corte superior	Programable entre	[100 Hz ; 200Hz]	[100 Hz ; 500Hz]
Frecuencia de corte inferior	Programable entre	0,1 Hz a 10 Hz	0,1 Hz a 30 Hz
Impedancia de entrada	Mayor que	500MΩ	1GΩ
Ruido intrínseco del amplificador	Menor que	5μV _{rms}	3μV _{rms}
CMRR	Mayor que	70 dB	100 dB

Tabla 1.3: Requerimientos para el AFE (64 canales).

⁵LAH significa que las dimensiones están dadas en el orden de Largo, Ancho y Altura.

⁶Se define el retardo de visualización de compresión como el tiempo que transcurre desde que una señal es adquirida por los electrodos, hasta que es visualizada en la interfaz de usuario.

⁷El presupuesto incluye la fabricación de una placa y materiales, menos el costo de electrodos secos.

1.5. Descripción global del funcionamiento

Especificaciones		Caso aceptable	Caso ideal
Frecuencia de muestreo	Programable entre	[200 Hz ; 400 Hz]	[200 Hz ; 1000 Hz]
Bits libres de ruido	Mayor o igual que	8 bits	12 bits

Tabla 1.4: Requerimientos para el conversor A/D (64 canales).

Especificaciones		Caso aceptable	Caso ideal
Memoria FLASH	Mayor que	12 kB	20 kB
Memoria RAM	Mayor que	32 kB	128 kB
FPU		No	Si

Tabla 1.5: Requerimientos para el Microcontrolador.

Especificaciones		Caso aceptable	Caso ideal
Throughput útil	Mayor que	128 kbps	1,12 Mbps
Distancia entre módulo remoto y PC	Mayor que	4 metros	6 metros

Tabla 1.6: Requerimientos para el transmisor/receptor.

A su vez desde la interfaz de usuario se debe poder configurar:

- Frecuencia de corte superior de los 64 canales.
- Frecuencia de corte inferior de los 64 canales.
- Frecuencia de muestreo del sistema.

De manera opcional, estará dentro del alcance del proyecto:

- Incorporar al sistema señales de sincronismo para estudiar la reacción a estímulos, sincronizadas con la señales de EEG (por ejemplo la presión de un pulsador) llamadas en adelante “señales de sincronismo”. Esto permitiría al médico/investigador realizar estudios de reacción a estímulos externos del paciente.
- Permitir que el AFE pueda ser sustituido por un amplificador neural integrado provisto por el tutor.
- Permitir el uso del algoritmo de compresión ACEEG.

1.5. Descripción global del funcionamiento

La figura 1.3 muestra el esquema de la comunicación entre los distintos bloques del sistema mencionados en la sección 1.4.

Al inicio, el usuario selecciona la configuración con la que desea adquirir las señales de EEG desde la interfaz de usuario. Entre las variables que puede configurar, selecciona la cantidad de canales de EEG (N_C) que desea visualizar entre 4 posibles arreglos predefinidos (64, 21 y 4 canales ya predefinidos⁸; o 1 canal cualquiera a elección) y la frecuencia de muestreo f_s con la que desea recibir estos N_C

⁸La elección de estas cantidades de canales se hicieron para ser compatibles con estudios medicos, ejemplo 21 es compatible con el EEG 10-10.

Capítulo 1. Introducción

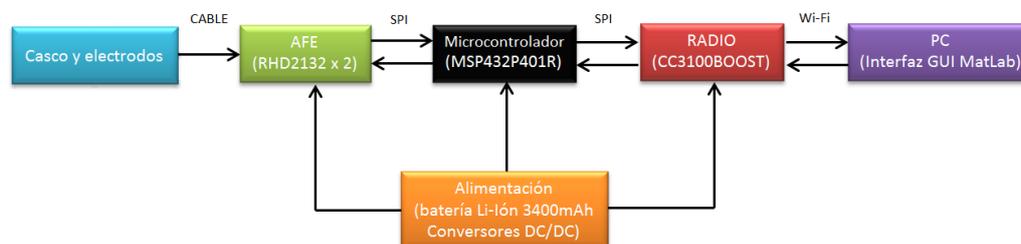


Figura 1.3: Comunicación de los distintos bloques del sistema.

canales.

De esta manera los N_C canales son muestreados periódicamente por el AFE cada $T_s = 1/f_s$ (controlado por el microcontrolador), y los envía por SPI⁹ al microcontrolador, que los envía también por SPI a la radio como se observa en la figura 1.3. La radio los envía por Wi-Fi al PC donde el usuario puede visualizar los canales adquiridos en tiempo real.

El usuario puede detener el estudio (adquisición periódica de los N_C canales) desde la interfaz de usuario cuando desee, e incluso puede configurar el funcionamiento del AFE y el microcontrolador.

El módulo remoto posee un filtro pasabanda programable, lo que le permite al usuario visualizar las señales en el rango de frecuencias que él quiera, lo que le da mayor versatilidad a sus estudios. Además puede medir la impedancia de los electrodos y agregar hasta 6 señales para realizar estudios a reacción de estímulos (canales de sincronismo).

En una segunda etapa, en la cual se integre el ACEEG, desde la interfaz se podrá configurar si se realizará la compresión de los datos, así como setear algunos parámetros del ACEEG en caso de utilizarlo. En el capítulo 6 se detalla el funcionamiento de la interfaz y los comandos disponibles.

1.6. Organización del documento

Este documento se separa en 9 capítulos:

En el **capítulo 1** se realizó la introducción al tema y se dio una breve descripción del funcionamiento del proyecto.

⁹SPI (Serial Peripheral Interface) es un protocolo de comunicación síncrono que se utiliza para la transferencia de datos entre circuitos integrados. En la comunicación uno de los integrados funciona como maestro y otro como esclavo. El maestro decide cuando quiere comunicarse con el esclavo imponiendo el reloj en la comunicación. El Microcontrolador es el maestro en la comunicación tanto con la radio como con el AFE.

1.6. Organización del documento

En el **capítulo 2** se detalla la elección de componentes del sistema, explicando el criterio elegido para la elección de cada uno de ellos.

En el **capítulo 3** se explica el diseño de la placa “AFE”.

En el **capítulo 4** se detalla el funcionamiento del software embebido diseñado. Sus distintos módulos y su implementación.

En el **capítulo 5** se justifica el protocolo de comunicación inalámbrica utilizado para intercambiar datos entre el módulo remoto y el PC.

En el **capítulo 6** se explica el diseño de la interfaz de usuario, qué comandos se pueden enviar al módulo remoto y cómo se utiliza para visualizar las señales adquiridas.

En el **capítulo 7** se habla del montaje del sistema. Cómo se conectan físicamente las distintas partes del módulo remoto, donde se sitúan en el paciente, que tipos de conectores se utilizaron, etc.

En el **capítulo 8** se agregan las pruebas más importantes del sistema, como por ejemplo medida de consumo, frecuencia máxima de funcionamiento, etc. Se hace un análisis de los resultados, justificando algunos criterios del diseño que fueron decididos empíricamente.

En el **capítulo 9** se presentan las conclusiones, trabajos a futuro para mejorar el sistema y tabla de las características finales.

Para no cargar el documento, demostración de algunos resultados y detalles de algunos criterios en el diseño se presentan en anexos. Al final del documento se encuentra la bibliografía consultada.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 2

Elección de los componentes de hardware

2.1. AFE (Analog Front End)

Se buscaron integrados dedicados al tratamiento de señales biológicas, de manera de evitar diseñar un AFE con componentes discretos. Se estudió el uso de tres integrados capaces de realizar esta tarea: ADS1299 de Texas Instruments, RHD2132 de Intan Technologies y RHD2164 de Intan Technologies.

2.1.1. ADS1299

El ADS1299 es capaz de amplificar por distintos valores (1, 2, 4, 8, 12 y 24) y digitalizar a distintas tasas de muestreo hasta 8 señales de EEG con una resolución de 24 bits. Presenta las siguientes desventajas:

- Son necesarias etapas de filtro pasabajo y pasaalto previas al ADS1299.
- Solo es capaz de relevar 8 canales, por lo que para llegar a los 64 canales requeridos es necesario buscar una solución alternativa con MUX analógicos o utilizar 8 integrados ADS1299.
- Los 24 bits no se aseguran libres de ruido, y uno de los requerimientos es que al menos 12 bits sean libres de ruido. Esto descartó el uso del ADS1299 puesto que con amplificación y filtros externos previos al ADS1299 no se logró asegurar 12 bits libres de ruido (ver fundamentación en anexo A).

2.1.2. RHD2132

El RHD2132 es un AFE para señales biológicas fabricado por Intan Technologies que permite muestrear hasta 32 canales.

Ventajas:

Capítulo 2. Elección de los componentes de hardware

- Amplificador, filtro pasaalto y pasabajo programable y conversor digital para 32 canales.
- Ruido referido a la entrada de $2,4\mu V_{rms}$.
- Frecuencia de corte superior variable entre $100Hz$ a $20kHz$.
- Frecuencia de corte inferior variable entre $0,1Hz$ a $500Hz$.
- Capacidad de medir la impedancia de los electrodos en tiempo real.
- Entradas ADC auxiliares para la interconexión de sensores adicionales.
- Conversor A/D de 16 bits libres de ruido con salida SPI a una velocidad de operación de $2,1MS/s$ ($32,8kS/s$ por canal).
- Tamaño reducido de $8,0mm \times 8,0mm \times 0,85mm$ (LAH).
- Posee un encapsulado estándar que permite ser soldado fácilmente.

La tabla 2.1 compara los requerimientos mínimos e ideales impuestos para el sistema con las características analógicas de la familia RHD2000 (en particular el RHD2132). En la misma se observa que los RHD2000 se ajustan a las necesidades de este proyecto.

Especificaciones	Requisitos mínimos	Requisitos deseados	Familia RHD2000
Cantidad de canales	16	64	64 (con 2 RHD2132)
Muestreo de cada canal	$0,2ksp/s/ch$	$1ksp/s/ch$	$32,8ksp/s/ch$
Resolución	$1\mu V_{pp}$	$1\mu V_{pp}$	$390nV_{pp}$
Número de bits libres de ruido	8	12	16
CMRR	$80dB$	$100dB$	$82dB$
Impedancia de entrada	$500M\Omega$	$1G\Omega$	$1,3G\Omega$
Ruido intrínseco a la entrada del amplificador	$5\mu V_{rms}$	$3\mu V_{rms}$	$2,4\mu V_{rms}$
Rango de frecuencia de corte superior	$100Hz$ a $200Hz$	$100Hz$ a $500Hz$	$100Hz$ a $20kHz$
Rango de frecuencia de corte inferior	$0,1Hz$ a $10Hz$	$0,1Hz$ a $30Hz$	$0,1Hz$ a $500Hz$

Tabla 2.1: Cuadro comparativo de las características de la familia RHD2000 con lo requerido por el cliente.

Como desventaja se tiene que la amplificación no es programable (fija de $192V/V$) y el costo es elevado. El costo de cada uno de estos integrados es de $US\$390$, lo que implica un presupuesto de $US\$780$ sólo para el AFE.

2.1.3. RHD2164

El RHD2164 es un AFE para señales biológicas fabricado por Intan Technologies que permite muestrear hasta 64 canales, el costo de este chip es de $US\$435$. Por ser de la familia RHD2000 tiene las características de la tabla 2.1. Sin embargo, no viene encapsulado, lo que hace que su soldado en una placa requiera una infraestructura especial la cual es muy costosa. Para evitar el soldado se lo puede comprar una placa de desarrollo que sale $US\$1785$.

2.2. Microcontrolador

	RHD2164	RHD2132 x 2
Costo	US\$1785 con placa de desarrollo	US\$780
Tamaño/Área (LAH)	7,3mm × 4,2mm × 0,20mm	2 × (8,0mm × 8,0mm × 0,85mm)
Rendimiento	Igual que RHD2132	Igual que RHD2164
Cantidad de canales	64 canales	64 canales
Montaje	Sin placa de desarrollo: Muy caro	Estándar (hay que diseñar placa)

Tabla 2.2: Tabla comparativa entre RHD2132 y RHD2164.

La tabla 2.2 compara a los integrados RHD2132 y RHD2164.

Después de esta comparación se decidió utilizar dos integrados RHD2132, por costo y facilidad de montaje.

2.2. Microcontrolador

El bloque microcontrolador de la figura 1.3 debe de cumplir con las siguientes características:

- Generales: de bajo consumo, liviano y pequeño.
- Compatibilidad con el algoritmo ACEEG: el algoritmo ACEEG necesita de $24kB$ en SRAM ¹ como máximo para realizar la compresión de 59 canales. El tamaño del programa es de aproximadamente $28kB$, lo cual implica un tamaño de memoria FLASH mayor a esa cantidad. Además, la velocidad de procesamiento debe de ser lo suficientemente alta como para que el algoritmo ACEEG procese las muestras más rápido que lo que el AFE demora en adquirirlas. El algoritmo ACEEG requiere además una unidad de punto flotante (FPU), por lo que se busca un microcontrolador con FPU.
- Compatibilidad con el AFE: los integrados RHD2132 (ver sección 2.1) requieren de una comunicación SPI por lo que, el microcontrolador a elegir debe contar con al menos dos unidades SPI. Los RHD2132 envían el muestreo de cada canal como un dato 16 bits, por lo que la velocidad de las unidades SPI debe ser mayor que $1,120MHz = (1kHz \times (64 + 6) \text{canales} \times 16 \text{bits/canal})^2$.

La tabla 2.3 muestra algunas de las opciones evaluadas.

Dado que los modelos MSP430G2xxx no cumplen con los requisitos necesarios para integrar el algoritmo ACEEG, se optó por descartarlos.

Se observó que en general un mayor tamaño de memoria en el microcontrolador se corresponde a un mayor consumo. En función de esto se prefirió reducir la memoria para ganar en consumo, lo que llevó a la elección del MSP432P401R. MSP432 y MSP430 tienen muchas características similares, y con este último han

¹Información provista por el cliente.

²Los 6 canales son las señales de sincronismo.

Capítulo 2. Elección de los componentes de hardware

Microcontrolador	MSP430G2xxx	MSP432P401x	SAM4CM	STM32L486xx	Zynq-7000 (DSP)	
Fabricante	TI	TI	ATMEL	ST	XILINKX	
Memoria	RAM	512 B	64 kB	304 kB	128 kB	256 kB SRAM 512 kB cache
	Flash	2 kB	256 kB	2 MB	1 MB	
Consumo	Activo	5,0mA	8,5 mA	100 mA	300 mA	≈ 500mA
	Dormido	56μA (LPM0)	70μA (LPM 0)	10μA (Backup Mode)	-	-
FPU	No	Sí	Sí	Sí	Sí	

Tabla 2.3: Opciones de microcontroladores a utilizarse para el sistema, todas las opciones cuentan con comunicación SPI.

trabajado dos de los tres integrantes del proyecto.

Se detallan algunas características relevantes del MSP432 (para más información de este microcontrolador ver [27]):

- Memoria: SRAM = 64 kB, Flash = 256 kB.
- Frecuencia máxima del reloj del sistema: 48 MHz.
- Consumo:
 - Activo = 8,5 mA³.
 - LPM 0 ⁴ = 70μA.
 - LPM 3, LPM 4 = 5μA.
- Cantidad de puertos SPI: 8.

2.3. Sistema de comunicación inalámbrico (Radio)

Inicialmente se buscaron radios que soportaran la comunicación Bluetooth. Luego de concluir que Bluetooth no cumplía los requerimientos necesarios⁵ para el sistema, se buscaron radios que soportaran la comunicación Wi-Fi (la discusión entre Bluetooth y Wi-Fi y la elección de Wi-Fi se desarrolla en la sección 5.2). Estas radios deben comunicarse con el microcontrolador por medio de UART o SPI y tener un throughput útil⁶ mayor a 1,120Mbps ($1kHz \times (64 + 6) \text{canales} \times 16 \text{bits/canal}$).

La tabla 2.4 muestran algunas de las opciones manejadas. En el anexo B.1 se muestran más opciones.

³Consumo máximo en el LDO en modo activo (@MCLK = 48MHz), no se encontró información acerca del consumo de los periféricos, por más información ver [28].

⁴En LPM 0 (low power mode) el reloj del sistema pasa a ser 128kHz.

⁵No alcanza el throughput necesario para el sistema ver sección 5.2.

⁶El throughput útil es definido como la velocidad real de transporte de datos a través de una red telemática, el cual normalmente se mide en Mbit/s y siempre será inferior al ancho de banda o bandwidth.

2.3. Sistema de comunicación inalámbrico (Radio)

Requerimientos		ESP8266EX	CC3100	ATSAMW25 MR210PA MR210PB	AMW006 / AMW106
Fabricante		Espressif Systems IOT Team	Texas Instrument	ATMEL	ACKme Networks
Memoria	Ram	<36 kB	700 B	32 kB	AMW006:96 KB AMW106: 128 KB
	Flash	<16 MB (external stack)	7 kB	256 kB	1 MB
Consumo	Tx max	b - 170 mA g - 140 mA n - 120 mA	1 DSSS - 272 mA (L0), 188 mA 6 OFDM - 248 mA (L0), 179 mA 54 OFDM - 223 mA (L0), 160 mA	-	-
	Rx max	b- 50 mA g - 56 mA n - 56 mA	1 DSSS - 53 mA 54 OFDM - 53 mA	-	-
	Otros	80 mA - Operation current 12 mA - I/O máx 15 mA - Modem sleep 10 μ A - Deep sleep	6 mA - High-level source current 450 mA - Peak calibration current 6 mA - Low-level sink current	13 mA - VDD IO H	5,7 mA Microcontrolador Stop mode, Wi-Fi rx 11,4 mA Microcontrolador Stop mode, Wi-Fi tx 320 mA Microcontrolador Stop mode, Wi-Fi tx 5,7mA Active receive @ 1Mbit/s UDP
Configuración ⁷		ATInstruction Set, Cloud Server, Android	Comandos propios	Comandos propios	Drivers provistos por el fabricante
Voltaje de operación (V)		3,0-3,6	2,1-3,6	2,7 - 3,6	3,0-3,6
Precio (U\$S)		9,95 (ESP8266 - 2471) 11,75 (317060018)	31,00 (con Launchpad para MSP432)	42,65 (digikey)	44,00 (digikey)

Tabla 2.4: Opciones de radios Wi-Fi a utilizarse en el sistema. Todas estas opciones utilizan SPI o UART, tienen los protocolos 802.11 b,g y n y cuentan con Stack TCP/IP.

Priorizando costos y dada la información obtenida de consumo, se optó por dos de ellas: ESP8266EX y CC3100.

La radio ESP8266 alcanza un throughput de $300kbps$ (inferior a los 1, 1, $120Mbps$ requeridos), por lo que la radio elegida fue la CC3100, que alcanza un throughput de $13Mbps$ utilizando TCP⁸. La radio CC3100, al ser de Texas Instrument, viene en una placa de desarrollo fácilmente integrable con la placa del MSP432. Además, Texas Instrument brinda al usuario ejemplos de aplicaciones que utilizan la radio, lo que facilita la programación. Se compraron 2 radios CC3100.

⁷Que tipo de comandos se tiene que enviar a la radio cuando se le quiere setear un modo de trabajo.

⁸TCP (Transmission Control Protocol) es un protocolo que se encarga de la comunicación desde un extremo al otro. Las funciones básicas del protocolo son: control de congestión, flujo de paquetes (para que un transmisor rápido no sature a un receptor lento), control de errores, retransmisiones en caso de pérdida de paquetes y secuenciamiento.

2.4. Alimentación y estimación de consumo

Para elegir la batería a utilizar en el módulo remoto fue necesario primero tener un estimativo del consumo del AFE, el microcontrolador y la radio elegidas.

2.4.1. Estimación del consumo del módulo remoto

En la página 34 de la hoja de datos del integrado RHD2132 se presenta el detalle del consumo de los bloques del mismo. En nuestro caso nos interesan:

- Cada integrado consume una corriente de reposo de $200\mu A$ para la alimentación de los amplificadores.
- Cada amplificador consume $7,6\mu A/kHz$ donde el consumo aumenta cuanto mayor sea la frecuencia de corte superior.
- Los ADC del integrado consumen $510\mu A$ de corriente de reposo.
- Los ADC consumen $2,14\mu A/(ksps/s)$, donde el consumo aumenta cuanto mayor sea la frecuencia de muestreo.
- ADC extras consumen cada uno $10\mu A$.

La tabla 2.5 detalla el consumo en el peor caso, esto es, con una frecuencia de corte superior máxima $f_H = 500Hz$ y una frecuencia de muestreo de $= 1ksps/ch$.

Bloques	Consumo detallado	Resultado
Corriente de reposo de los amplificadores	$2 \times 200\mu A$	$400\mu A$
Consumo amplificadores	$2 \times 32 \times 7,6\mu A/kHz \times 0,5kHz$	$243\mu A$
Corriente de reposo ADC	$2 \times 510\mu A$	$1020\mu A$
Consumo ADC	$2 \times 2,14\mu A/(ksps/s)32ksps/s$	$137\mu A$
Total		$1,81mA$

Tabla 2.5: Detalle del cálculo de consumo del RHD2132. Se recuerda que el AFE consta de dos chips RHD2132 cada uno capaz de adquirir 32 señales. Por lo tanto todos los consumos se deben contar dos veces.

Se estima el consumo del microcontrolador, tomando el peor caso, que corresponde al microcontrolador trabajando todo el tiempo en modo activo, por lo que, según lo indicado en la sección 2.2 el consumo será de $8,5mA$.

Para estimar el consumo de la radio, se toma el peor caso, que es que la radio esté todo el tiempo transmitiendo. La tabla 2.4 muestra que el consumo en transmisión es de $272mA$.

El consumo total estimado es de:

$$1,81mA + 8,5mA + 272mA = 282mA$$

2.4. Alimentación y estimación de consumo

2.4.2. Elección de la batería

Se busca una autonomía del módulo remoto de al menos 12 horas por lo que a partir del resultado anterior, la batería tiene que tener una carga superior a:

$$282mA \times 12h = 3384mAh$$

Para determinar la tensión de la batería se observan los rangos de funcionamiento de cada uno de los integrados en la tabla 2.6.

Integrado	Rango de tensión
RHD2132	3,2 V a 3,6 V (recomendado 3,3 V)
MSP432P401r	1,62 V a 3,7 V
CC3100	2,2 V a 4,8 V

Tabla 2.6: Rango de voltaje de funcionamiento de los tres integrados utilizados en el módulo remoto.

Dado que el rango del RHD2132 es el más acotado de los tres y que el valor de 3,3V sirve para los tres integrados se decide trabajar a este nivel de tensión regulando la tensión de salida de la batería para asegurar un correcto funcionamiento. Se buscan baterías en el rango de tensión de 3,0V a 3,7V.

En la búsqueda, se da relevancia a las siguientes características de la batería:

- Que sea una batería recargable.
- Tensión nominal dentro del rango mencionado (3,0V a 3,7V).
- Carga disponible (mayor o igual a 3,4Ah).
- Peso por pila.
- Precio.
- Si cuenta con cargador.
- Tamaño.
- Que se encuentre en stock.
- País de envío.
- Tipo de electrolito.

La tabla 2.7 muestra las mejores opciones manejadas.

Para conocer la calidad de las baterías, se investigaron tests de las tres últimas que eran pilas que se podían encontrar en el mercado interno (los test se presentan en el anexo B.4). Finalmente, se eligió la batería NCR18650B de Panasonic puesto que además de cumplir con los requerimientos necesarios, posee un circuito protector a prueba de descargas, lo que reduce significativamente el riesgo de explosión en una descarga brusca. Es fundamental en nuestro caso contar con este circuito

Capítulo 2. Elección de los componentes de hardware

Pila	Tensión nominal	Carga disponible	Peso por pila	Cantidad de pilas	Precio total	País de envío	Tipo de electrolito
Orbtronic 18650	3,7V	3400mAh	47g	2	US\$35	EEUU	Li-Ion
LiFePO4 26650	3,3V	3200mAh	86g	1	US\$9,91	EEUU	LiFePO4
Foxnovo 18650	3,7V	3400mAh	47g	4	US\$120	EEUU	Li-Ion
BRC 18650	3,7V	5000mAh	150g	4	US\$10,99	EEUU	Li-Ion
UltraFire 18650	3,7V	5000mAh	38,7g	1	\$250	Uruguay	Li-Ion
ICR18650-26F	3,7V	2600mAh	44,8g	1	US\$15	Uruguay	Li-Ion
NCR18650B	3,7V	3400mAh	45g	1	US\$14	Uruguay	Li-Ion

Tabla 2.7: Tabla comparativa de las opciones de baterías a comprar. Ninguna de las pilas de la tabla viene con cargador.

de protección puesto que la batería va a estar conectada en la nuca o en la cintura del paciente.

La tensión de la batería es regulada por convertidores DC/DC. Debido a que los integrados RHD2132 del AFE son más sensibles a variaciones de la alimentación que el microcontrolador y la radio, se decide utilizar dos convertidores DC/DC; uno dedicado únicamente a alimentar a los RHD2132 del AFE, y otro encargado de alimentar al resto del módulo remoto, es decir, a los bloques Microcontrolador y Radio. El ADP150 se utiliza para regular la tensión de los RHD2132 mientras que el TPS736 se utiliza para regular la tensión de los bloques microcontrolador y Radio. En el anexo B.3 se detalla la elección de los mismos.

2.5. Electrodo

En una primera instancia se centró la búsqueda en electrodos secos.

2.5.1. Electrodo seco

La tabla 2.8 resume las características de los electrodos estudiados.

Los electrodos g.SAHARA son activos, esto significa que incluyen componentes electrónicos que requieren alimentación y estos componentes son capaces de filtrar las señales que miden. El filtro tiene una frecuencia de corte superior de $40Hz$ [29] lo que no nos permitiría programar el ancho de banda hasta los $500Hz$ como se planteaba en los requerimientos del sistema.

Tomando en cuenta el costo y que tienen una impedancia mucho menor que la impedancia de entrada del RHD2132, se optó por comprar los dos tipos de electrodos de Cognionics⁹. Los electrodos FLEX SENSORS para utilizarse en la cabeza

⁹El presupuesto de los electrodos no se incluye en el presupuesto de proyecto.

2.5. Electrodo

Nombre	g.SAHARA	DRYTRODE	DRYPAD SENSORS	FLEX SENSORS
Fabricante	g.TEC	Neuroelectrics	Cognionics	Cognionics
Precio por unidad	-	50 USD	20 USD	20 USD
Offset	Depende del contacto con la piel	Depende del contacto con la piel	<100mV	-
Impedancia de entrada	Baja (No especifica)	Baja (No especifica)	10-100k Ω	100-2000k Ω
Rango de frecuencia	0,1 a 40 Hz	Depende de la electrónica	Depende de la electrónica	Depende de la electrónica
Pasivo/Activo	Activo	Pasivo	Pasivo	Pasivo
Material	-	Ag/AgCl	Ag/AgCl	Ag/AgCl
Casco	Sí	No lo venden	No lo venden	No lo venden
Cable con el electrodo	Sí	Se compra aparte	Cualquiera	Cualquiera
Diámetro electrodo	19 mm	-	19 mm	19 mm
Observaciones	Requiere hardware de adaptación	-	Para piel	Para pelo

Tabla 2.8: Tabla de electrodos analizados para realizar la compra. Información obtenida por contacto con los proveedores.

(cuero cabelludo) y los electrodos DRYPAD SENSORS para colocar en zonas sin cabello como en la nariz, por debajo de las orejas, etc. y medir señales de referencia.



(a) Electrodo de contacto con el cuero cabelludo (FLEX SENSORS).



(b) Electrodo de contacto sin cabello (DRYPAD SENSORS).

Figura 2.1: Electrodo seco de Cognionics Inc..

2.5.2. Electrodo húmedo

Debido a que la llegada de los electrodos secos se atrasó, en una segunda instancia se decidió comprar electrodos húmedos con el fin de no atrasar las pruebas. Las dos opciones evaluadas junto a los componentes más económicos que se en-

Capítulo 2. Elección de los componentes de hardware

contraron para cada opción se muestra en la tabla 2.9.

Opción	Componente	Empresa	Modelo	Cantidad	Precio
Opción 1 Electrodos EEG	Electrodos de disco (cable incluido)	Química Cenit S.A.	H503A (NE-113A)	26 unidades	\$9183
	Pasta EEG	Química Cenit S.A.	F510 (Z-401CE)	400 g x 3 unidades	\$580
Opción 2 Electrodos ECG	Electrodos ECG	S.A. Medical	4036-SWARO -V50-500-6000	50 unidades	\$500

Tabla 2.9: Electrodo húmedos en el mercado local.

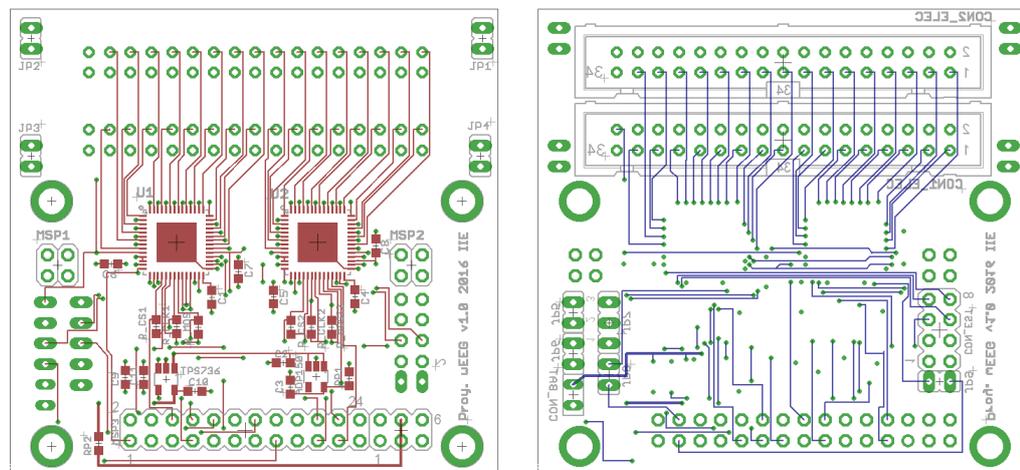
Analizando estos precios, se observó que el precio por unidad de la opción 1 (\$353 por unidad) es muy costoso en comparación con los electrodos de ECG de la opción 2 (\$10 por unidad).

Como estos electrodos se compran con el fin de adelantar las pruebas, se debe de verificar que se pueden adquirir señales EEG a partir de estos. Como trabajo a futuro se realizará esta verificación.

Capítulo 3

Diseño del AFE

A continuación se presenta el diseño de la placa AFE en la que se soldaron los integrados RHD2132 y los convertidores DC/DC ADP150 y TPS736. Utilizando el programa EAGLE se realizó el esquemático y el layout de la placa. Las figuras 3.1a y 3.1b muestran el layout de la misma con sus correspondientes conexiones.

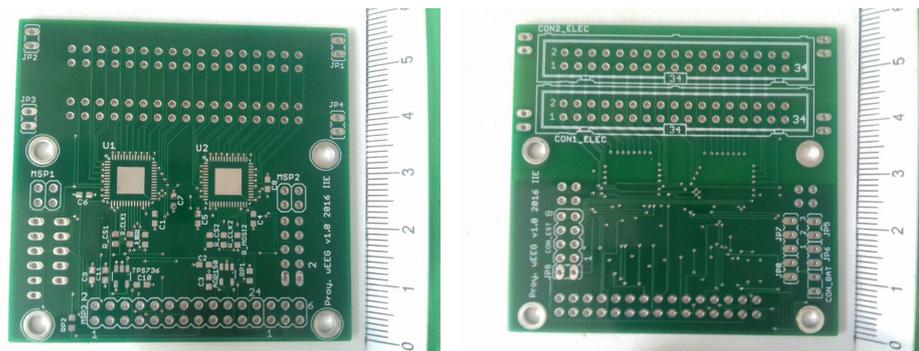


(a) Layout capa superior de la placa del AFE.

(b) Layout capa inferior de la placa del AFE.

Figura 3.1: Layout capas superior e inferior de la placa del AFE

La placa fue fabricada por “PCBCART”. Por recomendación del fabricante de los integrados RHD2132 (página 35 de [30]) se fabricó con 4 capas según el siguiente esquema: las capas superior e inferior fueron utilizadas para ruteo, mientras que las capas intermedias fueron utilizadas para planos de GND y VDD (GND en la capa más próxima a la capa superior). En la figura 3.2 se observa el resultado final.



(a) Vista superior de la placa.

(b) Vista inferior de la placa.

Figura 3.2: Vistas superior e inferior de la placa.

3.1. Funciones a utilizar del chip RHD2132

El integrado RHD2132 permite muestrear 32 señales y realizar otro tipo de medidas como test de impedancia, test de voltaje de alimentación, etc. Es capaz de recibir instrucciones (palabras de configuración) para setear variables como la frecuencia de corte superior e inferior de los filtros pasabanda que posee, la frecuencia del filtro para la eliminación de offset, etc. A continuación se mencionan las funciones más relevantes para el sistema. Para más información ver [30].

- **Frecuencias de corte superior e inferior del filtro pasabanda:** en la etapa de amplificación analógica de las señales adquiridas, el RHD2132 posee un filtro pasabanda de frecuencias de corte programable. La frecuencia del polo de corte inferior es de primer orden y puede variar en el rango de $0,1Hz$ a $500Hz$ y la frecuencia del polo de corte superior es un Butterworth de tercer orden y puede variar en el rango de $100Hz$ a $20kHz$. Para más información ver la página 26 de [30].
- **Filtro para eliminación de offset:** el RHD2132 posee un filtro digital pasaalto de primer orden para eliminar el voltaje residual de offset DC asociado a los amplificadores analógicos. La frecuencia de corte de este filtro es programable y además varía proporcionalmente en función de la frecuencia de muestreo. Para más información ver la página 32 de [30].
- **Test de impedancia de electrodos:** se elige un canal de los 32 disponibles en el RHD2132 y se mide la impedancia vista hacia atrás, lo que permite medir la impedancia del electrodo que está conectado. Para la medida, se inyecta en el canal una corriente conocida, que es definida seteando registros del RHD2132 y se realiza la adquisición del canal obteniendo valores de voltaje. Con el voltaje y con el valor de corriente conocido, se calcula la impedancia del electrodo. La corriente se puede inyectar de dos maneras: que el mismo RHD2132 inyecte una señal de corriente y sense la diferencia de voltaje que se genera, o que a partir de uno de sus pines (pin `elec_test`) se inyecte una señal externa. En el caso de que el integrado inyecte la señal,

3.2. Comunicación con el microcontrolador

es necesario que el pin `elec_test` quede sin conectar. Para más información ver anexo E.

- **Sensor del voltaje de alimentación:** dado que el RHD2132 es sensible a variaciones de alimentación (la misma no puede ser menor que 3,2V ni mayor a 3,6V para evitar daños), medir el valor de la alimentación VDD permite tener control sobre este voltaje minimizando la probabilidad de daños. Esta funcionalidad no agrega componentes extras en el exterior del chip. Para más información ver la página 32 de [30]. Queda como trabajo a futuro la implementación del software necesario para usar el sensor.
- **Activación de electrodos:** se puede utilizar el chip RHD2132 para aplicar voltajes DC a sus pines de entrada de manera de generar la activación de los electrodos. Esta tensión debe ser proporcionada por el pin `elec_test` (ver página 31 de [30]).
- **Señales de sincronismo:** cada chip RHD2132 presenta 3 entradas analógicas `auxin1`, `auxin2`, y `auxin3`. Estas entradas serán utilizadas en el sistema para realizar los estudios de reacción a estímulos externos (canales de sincronismo). El integrado RHD2132 muestrea sus 32 canales de señales a la vez que muestrea sus 3 entradas “auxin” y las envía por SPI. Dado que se tiene dos integrados RHD2132, se tendrán entonces 6 señales de sincronismo. Se debe tener en cuenta que las señales analógicas a ingresar por los pines “auxin” deben estar en el rango de 0,1V a 2,45V y que en caso de no ser usados deben ser conectados a VDD para minimizar el consumo de energía. Esto se resuelve permitiendo al médico/investigador que de forma externa pueda conectar VDD; para ello, en la placa AFE, se dejan accesibles dos pines VDD en un conector tipo “header” para que en caso de no usar los canales de sincronismo, se cortocircuiten estos pines con los pines VDD mediante un “jumper”. Para más información de estas entradas analógicas auxiliares ver la página 5 de [30].
- **Calibración de los conversores A/D:** el RHD2132 posee un conversor A/D que debe ser calibrado antes de iniciar una adquisición. Para la calibración, el chip utiliza señales conocidas para setear los parámetros del conversor.

3.2. Comunicación con el microcontrolador

La comunicación de los integrados RHD2132 es por SPI de cuatro hilos. El microcontrolador actúa de maestro en la comunicación, activando el chip select (\overline{CS}) y determinando la velocidad del reloj (CLK) dentro del rango que el microcontrolador y el RHD2132 soportan.

El RHD2132 tiene la capacidad de realizar la comunicación SPI utilizando Low-Voltage Differential Signaling (LVDS). Esta comunicación requiere hardware adicional, mientras que la SPI de cuatro hilos es suficiente para el sistema y está integrada en el microcontrolador. Se deja previsto LVDS para agregar en el futuro,

Capítulo 3. Diseño del AFE

ver anexo C para más detalle.

Para la comunicación SPI sin LVDS existen dos posibilidades como se muestra en las figuras 3.3a y 3.3b.

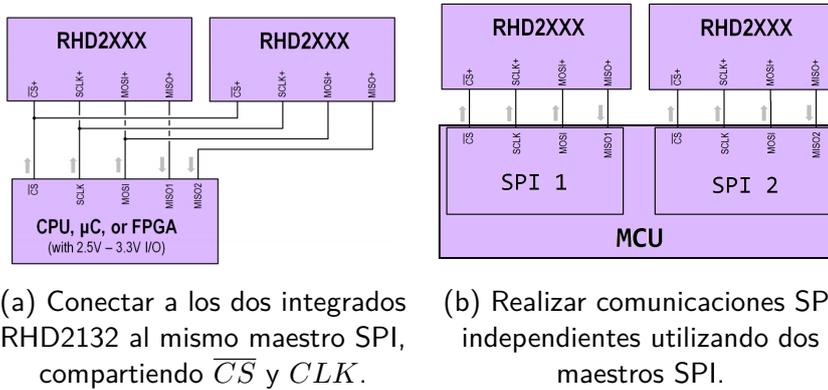


Figura 3.3: Posibles configuraciones para comunicar los integrados RHD2132 al microcontrolador MSP432.

La figura 3.3a es la recomendada por el fabricante de los integrados RHD2132 cuando se desea conectar dos de estos al mismo microcontrolador (ver página 14 de [30]). Esta forma de conexión presenta sin embargo una desventaja importante en el sistema, y es que no permite configurar a los integrados de maneras diferentes, puesto que, al recibir las mismas instrucciones de configuración, siempre se configuran ambos de la misma manera. Por esta razón se decidió independizar las comunicaciones de los chips trabajando con dos puertos SPI del maestro por separado como muestra la topología de comunicación en la figura 3.3b.

3.3. Componentes externos al chip RHD2132

La figura 3.4 muestra cómo debe conectarse cada integrado RHD2132 a cada puerto SPI del microcontrolador MSP432, y qué componentes externos deben utilizarse¹.

- El pin 26 (VDD) de cada chip RHD2132 lleva conectado un condensador de 100nF que debe estar a menos de 1cm del mismo. El condensador debe tener un dieléctrico X5R o X7R, un empaque mayor o igual a 0402 y soportar un voltaje mayor o igual a 16V (ver página 27 de [30]). Estos son los condensadores C6 y C5 en la placa AFE.
- Si no se usa LVDS se debe colocar otro condensador de 100nF a menos de 1cm de cada chip, conectado en el pin 31 (VDD)(ver página 27 de [30]). Estos son los condensadores C7 y C8 en la placa AFE.

¹Por errata de [30], en el pin 31 debería de mostrarse un condensador de 100nF. Este sí se indica en el resumen posterior a la imagen y en la página 27 de [30].

Capítulo 3. Diseño del AFE

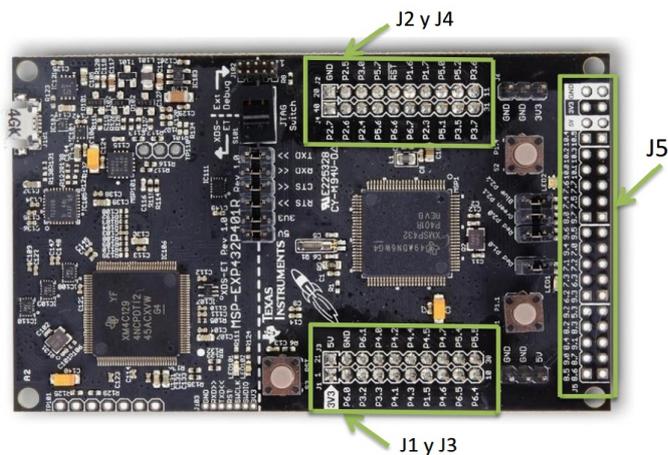


Figura 3.5: Disposición de los pines del MSP432 y su conexión con el resto de los bloques del módulo remoto.

Algunos de los pines de los integrados RHD2132 no son utilizados en el sistema, pero se dejan accesibles para futuras versiones que utilicen esta placa. En el anexo D se detalla cómo se conectan todos los pines de los RHD2132 y que componentes se utilizaron para realizar la placa.

Capítulo 4

Diseño del software embebido

Como se mencionó en la sección 1.4, el módulo remoto posee un microcontrolador que se encarga del procesamiento y de comunicarse con la interfaz de usuario que corre en un PC. Esto requiere de un software embebido capaz de:

- Recibir los datos de las señales EEG digitalizadas, provenientes de los dos integrados RHD2132.
- Procesar estos datos y enviarlos a la radio, que se encarga de transmitirlos de forma inalámbrica al PC.
- Recibir comandos desde el PC para modificar, tanto el funcionamiento del AFE, como del microcontrolador. Algunos de estos comandos deben de enviar una respuesta al PC.
- Prever la incorporación del algoritmo de compresión brindado por el grupo de teoría de información.

La figura 4.1 muestra un esquema donde se resume qué tipo de datos comparten el microcontrolador con el AFE y el microcontrolador con la radio.



Figura 4.1: Tipo de datos que comparte el microcontrolador con el resto de los bloques del módulo remoto.

En cuanto a la arquitectura del programa principal (main), se estudió las posibilidades de Round-Robin y encolado de funciones. Por razones que se exponen en el Anexo G.1, se decide utilizar Round-Robin.

En esta sección se utilizarán algunos términos especiales que son definidos a continuación:

Capítulo 4. Diseño del software embebido

Módulo: son las distintas partes en las que se subdivide el código de un software para englobar una determinada funcionalidad. Cada módulo tiene un determinado objetivo y en él se implementan las funciones necesarias que permiten lograrlo. No confundir a estos módulos de software con el módulo remoto de hardware definido en la sección 1.4.

Estudio: se define como una tarea que comienza desde que se recibe en el microcontrolador un comando *START* proveniente del PC hasta que se recibe desde el PC un comando (distinto de *START*). Esta tarea no tiene una duración fija (termina cuando se reciba un comando distinto de *START*) e involucra el envío de los datos de señales EEG digitalizados al PC.

Tirada: se define una tirada como los datos resultantes del muestreo de N_C canales de EEG y N_S señales de sincronismo. Como se mencionaba en la sección 1.5, N_C puede ser 64, 21, 16, 4 ó 1 y la cantidad de señales N_S puede variar de 0 a 6 dependiendo de la configuración elegida.

Adquisición: se define una adquisición como la acción del muestreo y registro de una tirada. Esta definición no incluye el envío de las tiradas por la radio. Las tiradas se adquieren de forma periódica con frecuencia f_s .

Instrucción: se define una instrucción como la palabra de configuración que se utiliza para controlar los chips RHD2132 del AFE.

Dentro de un mismo estudio pueden adquirirse muchas tiradas. Por ejemplo, si el estudio dura 1 hora a una frecuencia de muestreo de 500 Hz, se adquirirán 1,8 millones de tiradas ($1\text{hora} * 60\text{min}/\text{hora} * 60\text{seg}/\text{min} * 500\text{Hz}$).

4.1. Descripción general del funcionamiento del Software Embebido

A continuación se indican las tareas que se realizan desde el microcontrolador ordenadas por orden de prioridad:

Adquirir periódicamente las tiradas a través del AFE: durante un estudio, esta es la tarea con mayor prioridad, ya que es importante mantener el sincronismo de muestreo de las señales EEG. Esta tarea comienza con la interrupción del *Timer_A0* en donde se comienza con la solicitud de la lectura del primer canal.

Gestión de comandos del PC: comienza con la interrupción del *Timer_A1* donde se enciende periódicamente una bandera que se consulta periódicamente por polling desde el loop principal consultando por la llegada de comandos y ejecutándolos en caso afirmativo.

4.1. Descripción general del funcionamiento del Software Embebido

Gestión de pérdidas de paquetes: se realiza mediante polling en el loop principal, consultando si hubo pérdidas por desbordamiento del buffer de transmisión de paquetes a la radio (`buffer_packet`). Este es el único buffer que se puede desbordar, en la sección 4.9 se explica.

Envío de paquetes: se realiza mediante polling en el loop principal, consultando si hay paquetes para enviar y siempre que no haya habido pérdidas de paquetes.

Reposo: en caso de que no haya tareas pendientes, el microcontrolador entra en este estado.

La figura 4.2 muestra el diagrama de estados del software embebido de forma resumida.

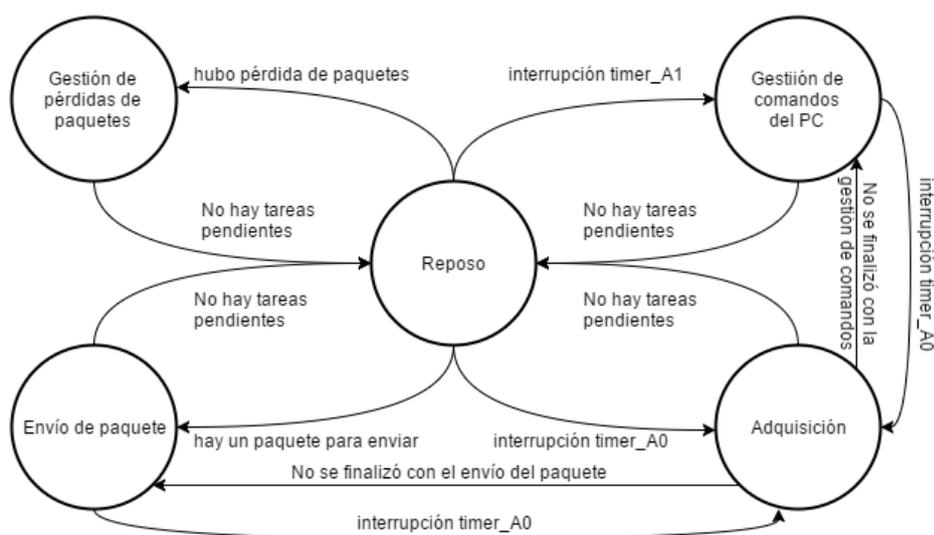


Figura 4.2: Diagrama de estados básico del software embebido.

Al comienzo, el microcontrolador inicializa los módulos. Luego ingresa al loop principal (ver sección 4.7). El espacio de memoria para los buffer se reserva en tiempo de compilación. El microcontrolador utiliza dos timers. El timer A1 interrumpe periódicamente cada 250 ms, y con esta interrupción enciende la bandera de `commandCheckFlag`. Con esta bandera el microcontrolador entra al estado de recepción de comandos y pregunta a la radio si hay algún comando proveniente del PC. En caso de que no hayan, el microcontrolador continúa con la tarea que se encontraba realizando, pero si llega algún comando, debe identificarlo entre todos los posibles. Los comandos se separan en dos grupos:

- Si el comando está asociado al comienzo o fin de un estudio.

Capítulo 4. Diseño del software embebido

- Si el comando está asociado a la configuración de parámetros o consulta del estado actual.

Si el comando es del segundo tipo, el módulo remoto actúa en consecuencia, configurando o respondiendo lo que debe, volviendo al loop principal. Si el comando es un START, comienza el estudio habilitando las interrupciones del Timer_A0 para comenzar a adquirir de forma periódica (de período T_s controlada por el timer A0). Dentro de un estudio, se tiene a ambos timers interrumpiendo. El timer A0 mantiene el período de muestreo T_s mientras que el timer A1 continúa consultando por la llegada de comandos provenientes del PC.

Las tiradas que se van adquiriendo, se van guardando en un paquete, estos paquetes se van guardando en un buffer (llamado `buffer_packet`) esperando a ser enviados. Cuando hay por lo menos un paquete completo, se entra al estado de envío de paquetes.

Notar que tanto en el estado de recepción de comandos como en envío de paquetes, se puede ir a adquisición con una interrupción del timer A0, volviendo al estado en que se encontraba al finalizar.

En caso de que dentro de un estudio se generen pérdidas causadas por el desbordamiento de `buffer_packet`, el sistema entra en estado de gestión de pérdidas, donde notifica al PC de lo ocurrido, limpia todos los registros involucrados con la adquisición de tiradas y reanuda el estudio.

El funcionamiento del sistema cuando recibe un nuevo comando desde el PC se explica en la sección 4.8. El detalle de como se realiza el estudio y como se envían los datos de las señales procesadas se detalla en la sección 4.9.

En caso de que no hayan tareas pendientes, el microcontrolador entra en el estado de reposo, donde entra en LPM0 (Low Power Mode) que es un estado de bajo consumo, y se sale de él cuando llega una nueva interrupción.

4.2. Interrupciones

A continuación se listan los periféricos que interrumpen en el sistema.

- Timer A0.
- Timer A1.
- Periférico SPI asociado al chip 0 (SPI0).
- Periférico SPI asociado al chip 1 (SPI1).
- Puerto I/O digital asociado a la radio (SPI).

4.3. Estructura en módulos

La interrupción **timerA0** se habilita al comenzar un estudio. En la rutina de atención a esta interrupción (*TA0_0_IRQHandler()*) se comienza a adquirir una tirada y solicitando la conversión del primer canal. En caso de que se ejecute esta rutina antes de que se haya finalizado con la adquisición de la tirada anterior, se envía un mensaje de fallo al PC, este tipo de fallas se llaman de aquí en más “fallas por superposición en la adquisición de tiradas consecutivas”.

La interrupción **timerA1** se habilita en la inicialización del módulo remoto. En la rutina de atención a esta interrupción (*TA1_0_IRQHandler()*) se enciende la bandera *commandCheckFlag* para consultar por la recepción de comandos desde el loop principal. Para ahorrar energía se deshabilita esta interrupción al entrar en LPM0, habilitándola cuando llegue un nuevo comando con la rutina de atención a la interrupción de la radio como se detalla más adelante.

La interrupción **SPIO** se habilita en la inicialización del módulo remoto. Interrumpe cada vez que tiene lista la muestra de un canal o de una señal de sincronismo del chip 0. Desde la rutina de atención a esta interrupción (*EUS-CIA3_IRQHandler()*) se registra el dato del pin solicitado y se pide la conversión del siguiente canal (del chip 0 o del chip 1).

La interrupción **SPI1** se habilita en la inicialización del módulo remoto. Interrumpe cada vez que tiene lista la muestra de un canal o de una señal de sincronismo del chip 1. Desde la rutina de atención a esta interrupción (*EUS-CIB3_IRQHandler()*) se registra el dato del pin solicitado y pide la conversión del siguiente canal. A diferencia de la rutina anterior, al finalizar con la conversión de canales, se guarda la tirada en *buffer_packet*.

Finalmente, la interrupción del **Puerto I/O de la radio** se habilita en la inicialización del sistema e interrumpe por dos razones, para avisar al microcontrolador que llegó un comando y en una transmisión para habilitar la copia de un paquete a la radio. En la rutina de atención a esta interrupción (*PORT2_IRQHandler()*) se habilitan las interrupciones del timer A1 para comenzar la lectura periódica de comandos desde el PC al salir de LPM0.

4.3. Estructura en módulos

La figura 4.3 muestra la estructura en módulos del software embebido.

A continuación se detalla la funcionalidad de cada uno de los módulos de la figura 4.3:

SPI: se encarga de implementar la comunicación mediante el protocolo SPI entre los dos integrados RH2132 del AFE y el microcontrolador. Para mayor información de este módulo ver la sección 4.4.

permite realizar tanto cola de datos (chars, int, string) como de variables más complejas (arreglos, strings, funciones). Pueden acceder a ella todos los módulos que precisen instanciar colas o buffers. Las funciones de este módulo se detallan en el anexo G.3.

Timer: se encarga de controlar el funcionamiento de los timers a utilizar en el sistema (timer_A0 y timer_A1). El timer A0 se encarga de la frecuencia de muestreo f_s la cual es elegida desde el PC. Este timer solo interrumpe durante un estudio, su funcionamiento se detalla en la sección 4.9. El timer A1 se utiliza para preguntar periódicamente si se recibió un comando proveniente del PC. El tiempo del timer A1 no es seteable por el usuario. En el anexo G.4 se detalla las funciones de este módulo.

Radio: se encarga del envío y la recepción de los datos desde el PC. Resuelve la comunicación con la radio CC3100 y permite establecer una conexión TCP/IP con el PC abriendo un puerto en modo servidor. Implementa funciones de transmisión y recepción de datos de forma inalámbrica que son utilizadas para enviar al PC las muestras procesadas, recibir comandos del PC y enviar mensajes de error en caso de haber fallos en el sistema. Para controlar a la radio CC3100 desde el MSP432 se utiliza la librería SimpleLink. Simplelink es una librería de Texas Instrument que provee funciones a los microcontroladores MSP43x para controlar a los integrados CC31xx / CC2xx, encargados de la comunicación Wi-Fi. Para mayor información de la librería SimpleLink ver [31]. En el anexo G.5 se describe el funcionamiento del módulo Radio.

CTRLerrores: se encarga de enviar mensajes al PC cuando ocurren eventos que no corresponden al curso normal del funcionamiento. En el anexo G.6 se detalla los mensajes de error que son enviados.

4.4. Módulo SPI

Como se mencionaba en la sección 3.2, se utiliza una comunicación SPI de cuatro hilos para cada uno de los integrados RHD2132. Esto significa que la comunicación necesita de cuatro señales digitales por cada integrado RHD2132 (ver figura 3.3b):

- \overline{CS} : la señal de Chip Select (\overline{CS}) es manejada por el microcontrolador y se activa cuando el MSP432 quiere enviar o recibir datos del integrado RHD2132. Esta señal es activa por nivel bajo.
- CLK : la comunicación SPI es síncrona por lo que requiere de un reloj (CLK) para que la transferencia ocurra.
- $MISO$: por este pin el RHD2132 envía datos al microcontrolador ($MISO$ son las siglas de master input - slave output).

Capítulo 4. Diseño del software embebido

- *MOSI*: por este pin el microcontrolador envía datos al RHD2132 (*MOSI* son las siglas de master output - slave input).

Los puertos SPI del MSP432 elegidos son: A3 para comunicarse con el chip 0 y B3 para comunicarse con el chip 1¹. Los pines del MSP432 a los que estos puertos corresponden se indican en la tabla D.2.

Los integrados RHD2132 utilizan una comunicación SPI de 16 bits con \overline{CS} activo por nivel bajo. Los bits se leen en flanco de subida del CLK y se cambian en flanco de bajada del CLK (CLK es activo por nivel alto). De la palabra de 16 bits, se debe enviar primero el bit más significativo (MSB). La figura 4.4 muestra como envía y recibe palabras de 16 bits el integrado RHD2132.

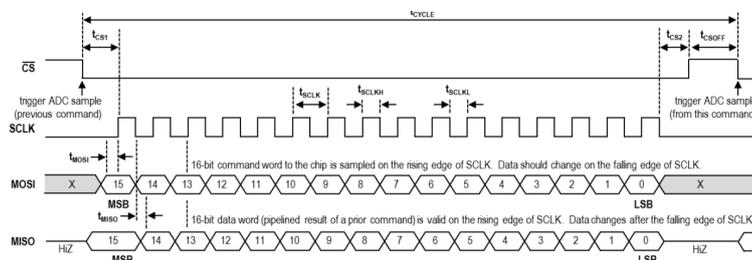


Figura 4.4: Funcionamiento deseado de la comunicación SPI para los integrados RHD2132 (imagen extraída de [30]).

Debido a que el MSP432 solo puede realizar comunicaciones SPI de 8 bits, dejando un retardo entre dos palabras de 8 bits, los fabricantes del RHD2132 recomendaron [32] realizar el envío como muestra la figura 4.5.

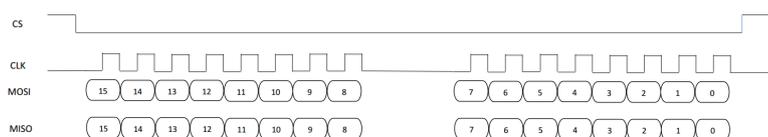


Figura 4.5: Funcionamiento de la comunicación SPI de 16 bits.

En este caso, se descompone cada palabra de 16 bits en dos palabras de 8 bits, y se envía la parte más significativa primero, y tiempo después la segunda parte de 8 bits, pero sin subir el \overline{CS} entre estas. Se utilizó la mayor velocidad posible en la comunicación SPI que permite el MSP432 (12 MHz).

Los integrados RHD2132 contestan al comando enviado por SPI dos ciclos de lectura de SPI después, como se puede apreciar en la figura 4.6. Esto ocurre

¹La enumeración de los RHD2132 de chip 0 y chip 1 se corresponde con la numeración mostrada en el anexo D.

porque, primero lo recibe, en el tiempo siguiente lo interpreta y lo ejecuta, y en el tiempo siguiente envía la respuesta. Esto implica que en casos en los que se necesite esperar la respuesta, es necesario enviar dos comandos **DUMMY**².

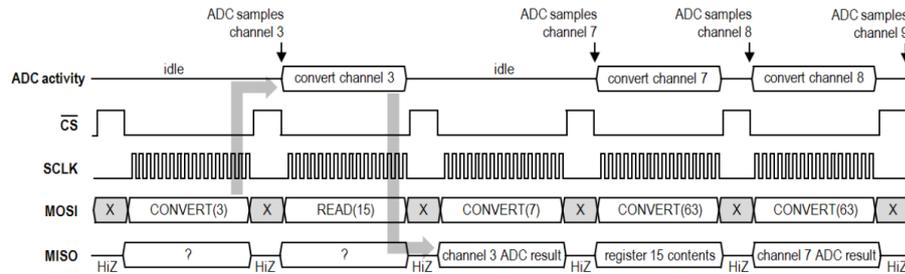


Figura 4.6: Diagrama de tiempos de instrucciones para el RHD2132 (imagen extraída de [30]).

4.4.1. Buffers del módulo SPI

Como muestra la figura 4.7, para la comunicación con los dos integrados RHD2132 se utilizan dos buffers circulares de 16 bits para la recepción y dos buffers circulares de 16 bits para la transmisión. **buffer_out_SPI** es un arreglo de buffers de transmisión. El `buffer_out_SPI[0]` corresponde al buffer para transmitir datos al chip 0 y `buffer_out_SPI[1]` corresponde al buffer para transmitir datos para el chip 1. De manera análoga, **buffer_in_SPI** es un arreglo de buffers de recepción. Los `buffer_in_SPI[0]` y `buffer_in_SPI[1]` corresponden a los buffers para recibir los datos de los chips 0 y 1 respectivamente. En estos buffers pueden almacenarse hasta 100 palabras de 16 bits.

El funcionamiento de estos buffers circulares se hace utilizando las funciones del módulo que se muestran en el anexo G.3. En los buffers de transmisión se agregan todas las instrucciones que se desean mandar, y el mismo módulo se encarga de enviar todo lo que esté en los buffers hasta que no queden más datos para transmitir. Es necesario utilizar una de las funciones del módulo SPI para iniciar el envío de estos datos.

4.4.2. Funciones del módulo SPI

El módulo SPI posee dos funciones que se utilizan para enviar datos de dos maneras distintas:

- **void send_word(uint8_t destinatario)**

²Se define como DUMMY a una instrucción que solo se envía al RHD2132 con el propósito de generar ciclos de lectura de SPI, para obtener la respuesta a un comando enviado previo al comando DUMMY. Se eligió como comando DUMMY, la lectura de un registro del integrado RHD2132 que simplemente envía la letra 'I' en código ASCII.

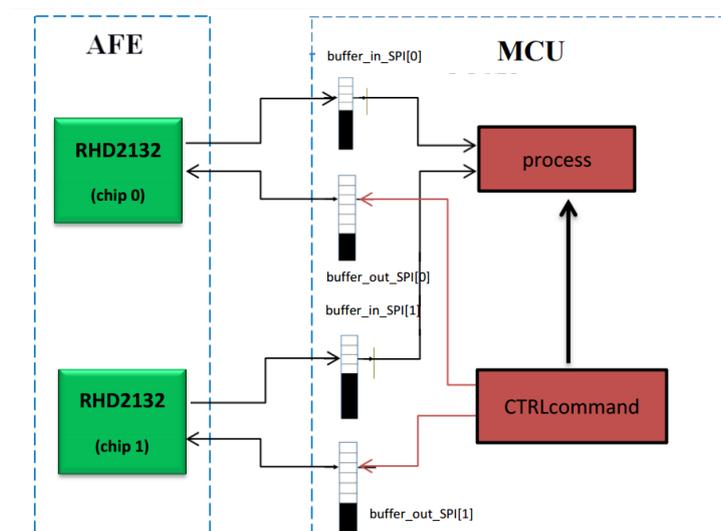


Figura 4.7: Diagrama en bloques de la transmisión y recepción de la comunicación SPI del lado del microcontrolador.

Esta función envía la parte alta del primer dato del `buffer_out_SPI` al chip indicado en destinatario. Se utiliza para comenzar el envío de los comandos que están en `buffer_out_SPI[destinatario]`. El parámetro destinatario tomará los valores 0 y 1 en función del chip al que se le quiera enviar (el diagrama de flujo de esta función se muestra en el anexo G.2 (ver figura G.1).

- `send_and_wait(uint16_t comando, uint8_t destinatario, uint8_t ciclos)`

Envía la palabra de 16 bits *comando* al chip *destinatario*, con la particularidad de que bloquea la ejecución del microcontrolador hasta que no se reciba una cantidad *ciclos* + 1 de palabras de 16 bits. Esta función se utiliza para el caso en que se necesite esperar la respuesta al comando enviado, como se mencionaba al final de la sección 4.4.1. La figura G.3a muestra el diagrama de flujo de la función *send_and_wait*.

4.4.3. Interrupciones del módulo SPI

Se tendrá una rutina de interrupción por cada puerto de comunicación SPI, aunque estas tendrán la misma estructura. El diagrama de flujo presentado en el anexo G.2 (ver figura G.2), resume el funcionamiento de cualquiera de ellas.

Las interrupciones se separan en dos tramos. Un primer tramo es la recepción de la palabra enviada por el RHD2132, y el otro tramo es la transmisión de un nue-

vo dato³. Como la transmisión y recepción SPI del microcontrolador es de 8 bits, pero las palabras a enviar a los integrados RHD2132 son de 16 bits, es necesario dos interrupciones SPI para tener la palabra completa. Por lo tanto, en cada caso es necesario tener claro si lo que se envía o recibe es la parte baja o la parte alta de la palabra de 16 bits. Primero se recibe la parte alta, se guarda en memoria y a la siguiente interrupción cuando se recibe la parte baja, se guarda toda la palabra en el buffer `in_SPI` del chip correspondiente. Lo mismo ocurre con la transmisión.

La rutina de interrupción para ambos chips es similar, salvo que se esté dentro de un estudio, en cuyo caso, los integrados RHD2132 trabajan por turnos. Primero muestrea todos los canales el chip 0 y cuando termina le da la orden al chip 1. Cuando el chip 1 termina, se llama a la función `transfer_channel()` que se encarga de guardar la tirada adquirida. En la sección 4.9 se detalla el funcionamiento del sistema en un estudio.

4.5. Módulo AFE

Este módulo implementa funciones de alto nivel que envían instrucciones a los integrados RHD2132. Separamos las funciones de este módulo en dos grandes grupos en función de su nivel de abstracción.

Funciones de bajo nivel de abstracción: generan la instrucción de 16 bits a enviar a los RHD2132, envían y esperan la respuesta a una instrucción determinada. En este grupo entran funciones que permiten modificar o leer registros del RHD2132, convertir un determinado canal, etc.

Funciones de alto nivel de abstracción: utilizan las funciones de bajo nivel de abstracción para realizar una determinada funcionalidad. Por ejemplo, funciones que permiten modificar o leer las frecuencias de corte superior e inferior, convertir canales en las 4 configuraciones posibles, etc.

4.5.1. Funciones de bajo nivel de abstracción

El RHD2132 responde únicamente a 5 instrucciones (información extraída de [30]). En todos los casos, el RHD2132 entrega una respuesta:

- **CONVERT(C):** se muestrea y digitaliza la señal analógica del canal C. Se puede optar por aplicarle un filtro pasaalto para eliminar el offset de la señal a la salida del ADC. La respuesta es el valor medido representado en formato digital por desplazamiento.
- **CALIBRATE():** realiza la auto-calibración del ADC. Deben enviarse 9 comandos DUMMY luego de CALIBRATE() para que el integrado realice la calibración. La respuesta es una palabra fija que permite verificar que el comando se recibió correctamente.

³Se recuerda que en SPI, por cada transmisión, hay una recepción simultánea.

Capítulo 4. Diseño del software embebido

- **CLEAR()**: limpia los parámetros de calibración generados por el comando CALIBRATE(). La respuesta es una palabra fija que permite verificar que el RHD2132 recibió este comando.
- **WRITE(R,D)**: escribe el dato D en el registro R del RHD2132. La respuesta tiene en la parte alta un identificador de que recibió este comando, y en la parte baja devuelve que valor el RHD2132 escribió en el registro R.
- **READ(R)**: hace una lectura del registro R del RHD2132. La respuesta es en la parte alta un identificador que recibió este comando y en la parte baja la lectura del registro R solicitada.

En el módulo AFE se implementan las funciones *get_word_CALIBRATE*, *get_word_READ*, *get_word_WRITE*, *get_word_CLEAR* y *get_word_CONVERT* con sus correspondientes parámetros, que generan la palabra que el integrado RHD2132 entiende para ejecutar los comandos anteriores:

- **uint8_t WRITE(uint8_t registro_RHD2132, uint8_t dato, uint8_t destinatario)**

Escribe *dato* en el registro *registro_RHD2132* del chip *destinatario*. Esta función retorna 1 si el registro se escribió correctamente o 0 en el caso contrario.

- **uint8_t READ(uint8_t registro_RHD2132, uint8_t destinatario)**

Lee el registro *registro_RHD2132* del chip *destinatario*. Esta función retorna el valor de la lectura realizada.

- **void CALIBRATE(uint8_t destinatario)**

Realiza la calibración al chip indicado en *destinatario*.

- **void CLEAR(uint8_t destinatario)**

Limpia los parámetros de calibración del chip indicado en *destinatario*.

- **uint16_t CONVERT(uint8_t channel, uint8_t elim_offset)**

Se realiza la conversión de un solo canal y se espera a recibir la respuesta. Si el canal elegido está entre 0 y 31, la instrucción se envía al chip 0, si el canal elegido está entre 32 y 63, la instrucción se envía al chip 1. *elim_offset* indica si a la señal a muestrear se le aplica eliminación de offset. Esta función retorna el resultado de la digitalización de dicho canal.

Todas las funciones anteriores se basan en la función *send_and_wait* descrita en la sección 4.4.2. Debido a que no hay restricciones de tiempo en el momento de atender un comando de configuración del sistema, se utilizan estas funciones de espera de respuesta.

Todas las funciones esperan 2 tiempos de SPI por lo mencionado en la sección 4.4, salvo la función CALIBRATE que espera 9 tiempos de SPI.

4.5.2. Funciones de alto nivel de abstracción

A partir de las funciones de baja abstracción se pueden crear funciones que realicen una determinada funcionalidad del sistema. Las funciones que consideramos de alta abstracción del AFE son:

- **void CONVERTburst()**

La función CONVERT descrita en 4.5.1 espera la respuesta. Esto implica que por cada comando CONVERT(C) que se envía, se envían dos comandos DUMMY. En términos de velocidad, usar CONVERT no nos permite aprovechar al máximo los recursos del integrado RHD2132, puesto que se desechan 2 de cada 3 respuestas enviadas. Por tanto, se implementa la función CONVERTburst() que permite enviar **ráfagas** de comandos CONVERT(C), esto es, enviar de forma continua comandos CONVERT(C) para que los integrados RHD2132 muestreen y digitalicen los canales que se les indica, sin uso de comandos DUMMY entre medio.

La función CONVERTburst() se encarga de cargar en los buffer_out_SPI de cada chip los comandos CONVERT(C) necesarios para recibir el muestreo de los canales, según la configuración de canales establecida desde el PC. En caso de que corresponda, se agrega en los buffer_out_SPI instrucciones para convertir la cantidad de señales de sincronismo que se indique. Finalmente, inicia la comunicación SPI.

La sección 4.9 detalla el funcionamiento del sistema en un estudio. En el anexo G.2 (ver figura G.4) se observa el diagrama de flujo de CONVERTburst().

- **void set_sup_cutoff_freq(int freq)**

Lee y sobrescribe los registros necesarios de los dos integrados RHD2132 para configurar la frecuencia de corte superior en el valor indicado en *freq*. La frecuencia de ambos integrados siempre tiene que ser igual.

- **void set_inf_cutoff_freq(int freqx100)**

Lee y sobrescribe los registros necesarios de los dos integrados RHD2132 para configurar la frecuencia de corte inferior en el valor indicado en *freqx100*. Se envía el valor de frecuencia f_{inf} multiplicado por de 100, para evitar el uso de números no enteros en los paquetes de configuración. La frecuencia de ambos integrados siempre tiene que ser igual.

▪ **void offAmps(uint8_t NumCH_to_convert)**

Los integrados RHD2132 tienen 4 registros cada uno, que pueden ser seteados para apagar los amplificadores que no se estén utilizando. Esto permite reducir el consumo del RHD2132. La función *offAmps*, permite apagar todos los amplificadores que no se usen en función de la configuración de canales que se estén utilizando. Solo en el caso de *NumCH_to_convert* = 1, se mantiene encendido todos los amplificadores, puesto que a priori, el canal a utilizar puede ser cualquiera de los 64 disponibles.

▪ **void config_sincronismo(uint8_t sincronismo)**

Setea los registros necesarios para habilitar o deshabilitar el uso de las señales de sincronismo.

▪ **int get_sup_cutoff_freq()**

Solicita a los dos integrados RHD2132 la frecuencia de corte superior y comprueba que sea la misma. En caso de haber diferencias, se reporta el error al PC (ver anexo G.6). En caso contrario, se retorna el valor de la frecuencia de corte superior de ambos chips.

▪ **int get_inf_cutoff_freq()**

Solicita a los dos integrados RHD2132 la frecuencia de corte inferior y comprueba que sea la misma. En caso de haber diferencias, se reporta el error al PC (ver anexo G.6). En caso contrario, se retorna el valor de la frecuencia de corte inferior de ambos chips.

▪ **void AFE_init()**

Se encarga de setear todos los registros del RHD2132 utilizando funciones de bajo y alto nivel de abstracción. Algunos registros deben setearse como se indica en [30]. Para otros, se tomó un criterio para inicializar el sistema que se describe a continuación:

- Se setea la frecuencia de corte del filtro pasaalto para la eliminación de offset. Como se indica en la página 33 de [30], la frecuencia f_{eo} de este filtro es proporcional a la frecuencia con la que se envían comandos para conversión de canales ($f_{eo} = k_{frec_{DSP}} f_{sample}$). Esta constante $k_{frec_{DSP}}$ se inicia en el valor $3,886 \times 10^{-5}$.
- Se inicia con el testeo de impedancia de electrodos desactivado.
- Para el filtro pasabanda programable se setea la frecuencia de corte superior f_H en 250Hz y frecuencia de corte inferior f_L en 0,1Hz.
- Se encienden los amplificadores de los dos integrados RHD2132.
- Se prepara al sistema para iniciar con 64 canales.
- Se setean los registros para empezar sin señales de sincronismo.

Finalmente se envía CALIBRATE a ambos chips.

4.6. Módulo Process

Este módulo se encarga del procesamiento de las tiradas y su posterior almacenamiento. Los datos procesados se almacenan dentro de un registro llamado **paquete**. Este registro constituye un conjunto de tiradas que se agrupan para transmitirse al PC. Cada paquete tiene al principio un encabezado. El encabezado será de dos bytes. El primer byte corresponde al identificador de paquete (que es una “S” en código ASCII para indicar al PC que el paquete recibido corresponde a la llegada de un conjunto de tiradas) y el siguiente indica el número de paquete enviado, lo que le permite al PC identificar si hubo pérdida de paquete.

Cada paquete a su vez, se guarda dentro de un buffer esperando a ser enviado al PC. Este buffer, llamado *buffer_packet* tiene una capacidad máxima para 40 paquetes de 1460 bytes. El tamaño de *buffer_packet* fue elegido para alcanzar el valor óptimo de throughput útil (ver fundamentación en 8.1.1).

Se considera que *buffer_packet* deja de estar vacío cuando se llena un paquete, quedando habilitada la transmisión de paquetes desde el loop principal.

Si se utiliza el algoritmo ACEEG, previo a copiar los datos adquiridos a *buffer_packet*, se debe utilizar el módulo *compress*.

4.7. Módulo main

El programa principal tiene arquitectura Round-Robin como se muestra en la figura 4.8.

La comunicación SPI y las interrupciones habilitadas deben ejecutarse antes de *AFE_init()* puesto que como se mencionó en la sección 4.5.2, *AFE_init()* necesita la comunicación SPI para escribir en los registros de los integrados RHD2132. También *Radio_init()* debe estar antes de *Timer_init()*, para evitar que el timer A1 interrumpa sin que esté establecida la comunicación con el PC.

La frecuencia de muestreo con la que inicia el modulo remoto por defecto es $f_s = 500Hz$ la cual se setea en *Timer_init()*.

En el loop principal se realizan 3 tareas:

- **Envío de paquetes a la radio:** si no hay pérdidas y no está vacío *buffer_packet*, se envían datos a la radio (ver sección 4.6 para más información de *buffer_packet*).

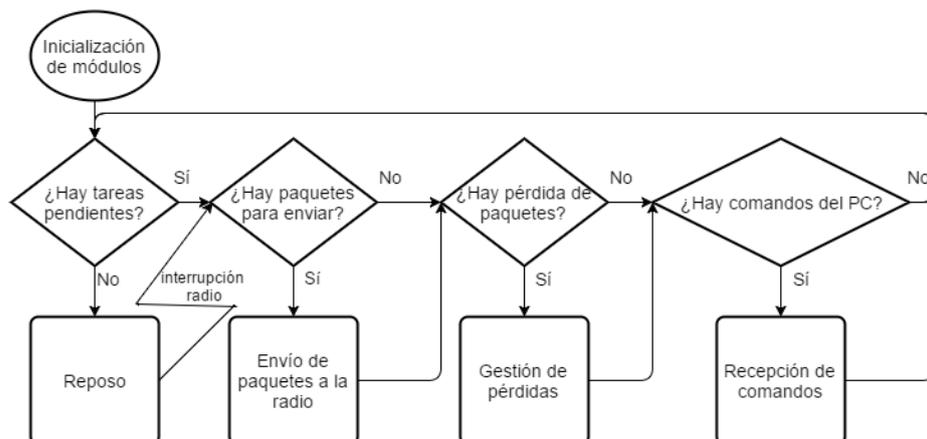


Figura 4.8: Diagrama de flujo del módulo principal con arquitectura Round-Robin con interrupciones.

- **Gestión de pérdidas:** en caso de que se llene *buffer_packet*, se corta momentaneamente la adquisición de muestras y se ejecuta esta tarea, la cual se ocupa de enviar al PC un paquete que indique que hubo pérdida de datos, esta tarea limpia los buffers de comunicación SPI y *buffer_packet* reanudando con el estudio.
- **Recepción de comandos:** si la bandera *commandCheckFlag* está activa (*commandCheckFlag* se enciende en cada interrupción del timer A1) se ejecuta la función *commandCheck()* que consulta por la llegada de algún comando desde el PC. En caso afirmativo se atiende el comando, enviando una respuesta al PC en caso de que el comando recibido no implique iniciar o detener el estudio.

En caso que no haya ninguna tarea para realizar en el loop principal, el microcontrolador entra en modo de bajo consumo (LPM0). Se elige el modo de bajo consumo LPM0 puesto que es el de mayor ahorro de consumo para los requerimientos del sistema. El siguiente modo de bajo consumo apaga los timers, por lo que no se podría consultar por la llegada de comandos del PC.

4.8. Funcionamiento del módulo remoto en configuración

En el momento en que llega un comando desde el PC, desde la función *commandCheck()* se lee el comando recibido y se ejecuta *commandRead()* donde se toma la acción correspondiente. Desde el PC se pueden enviar 9 comandos distintos:

- **Configuración (CONFIG):** modifica la frecuencia de muestreo con la que se va a trabajar, la frecuencia de corte superior e inferior del filtro programable de los integrados RHD2132, con cuántas señales de sincronismo se va

4.8. Funcionamiento del módulo remoto en configuración

a trabajar, si se va a utilizar el algoritmo ACEEG ⁴.

- **Comenzar un estudio (START)**: se inicia el estudio con la configuración de más de un canal. En este comando se indica a su vez con qué configuración de canales se va a trabajar y si se va a usar eliminación de offset.
- **Comenzar estudio con un solo canal (ONE_CHANNEL)**: se inicia el estudio con la configuración para un solo canal. En este comando se indica a su vez qué canal se debe muestrear y si se va a utilizar eliminación de offset.
- **Detener un estudio (STOP)**: con este comando el PC le indica al módulo remoto que se va a terminar con el estudio en proceso.
- **Estado actual (EA)**: Al recibir este comando, se devuelve el estado actual del sistema, indicando frecuencia de muestreo, frecuencia de corte superior, frecuencia de corte inferior, cantidad de señales de sincronismo, si se está trabajando con el algoritmo ACEEG.
- **Test de impedancia de electrodos (TIE)**: Realiza el test de impedancia de electrodo en dos canales, uno por cada integrado RHD2132.
- **Calibrar AFE (CALIBRATE)**: calibra los conversores A/D de los dos integrados RHD2132.
- **Salir (SALIR)**: resetea al módulo remoto para establecer una nueva comunicación.

La figura 4.9 muestra el diagrama de flujo del módulo remoto en configuración. Se recuadran los comandos para mayor claridad del diagrama. Para complementar el diagrama de flujo se agregan los siguientes comentarios:

- Para el comando *CONFIG*, previo a realizar las modificaciones, se revisa que el módulo remoto ya no esté configurado de esa manera, lo que ahorra re-configuraciones.
- *config_sincronismo()* se utiliza solamente si se pasa de ninguna señal de sincronismo a una cantidad no nula, o viceversa. Esto es porque estas entradas se apagan o encienden en conjunto.
- *flag_compress* es la bandera que indica si se utiliza el algoritmo ACEEG.

El funcionamiento del sistema durante un estudio se detalla en la sección 4.9.

⁴En caso de utilizar el algoritmo ACEEG se setea el valor de distorsión absoluta máxima por muestra.

Capítulo 4. Diseño del software embebido

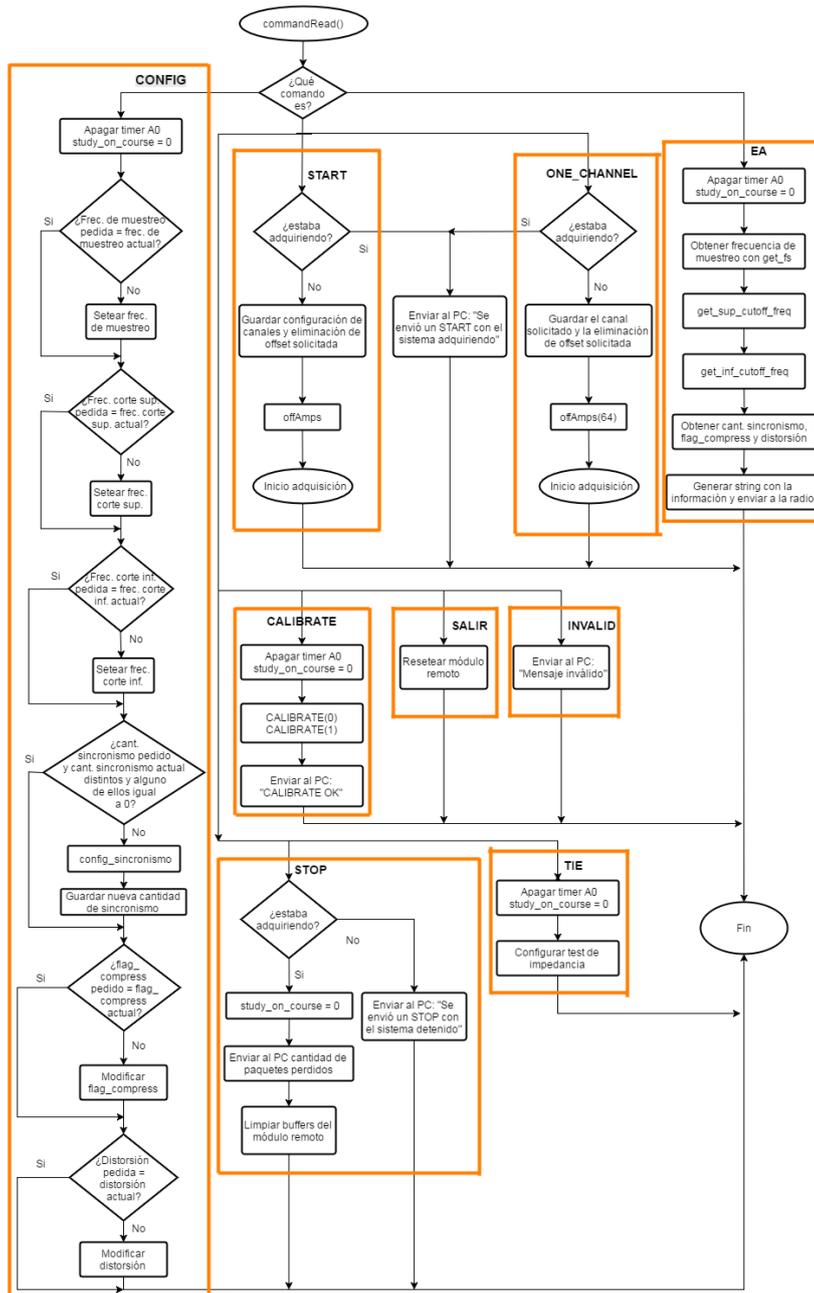


Figura 4.9: Diagrama de flujo para interpretar un comando proveniente del PC.

4.9. Funcionamiento del módulo remoto durante un estudio

Como se mostraba en la sección 4.8, cuando se envía un comando START desde el PC, en él se indica la cantidad de canales que se desea adquirir, y si se desea utilizar la eliminación de offset o no. La cantidad de canales de sincronismo

4.9. Funcionamiento del módulo remoto durante un estudio

y la frecuencia de muestreo f_s se deben determinar antes con un comando de configuración, puesto que una vez recibido el comando START se enciende el timer A0 con el valor de f_s que el microcontrolador tiene cargado.

A continuación se definen las banderas que indican distintos estados dentro de un estudio:

- **study_on_course**: indica que se está realizando un estudio (activa por alto). Vale 1 desde que se termina de procesar un comando START hasta que se recibe un comando distinto de START.
- **is_acquiring**: indica que se está adquiriendo una tirada (activa por alto), vale 1 cuando el timer interrumpe para adquirir una nueva tirada y se baja a 0 cuando se termina de copiar la tirada en *buffer_packet*.
- **buffer_packet_overflow**: indica el estado de *buffer_packet*. Vale 1 cuando este buffer se llenó causando pérdida de datos. En este estado se debe avisar al PC de la falla. Una vez realizado el aviso, se vacían todos los buffers involucrados y *buffer_packet_overflow* vuelve a 0 reanudándose el estudio.

La figura 4.10 ilustra la diferencia entre estas dos banderas.

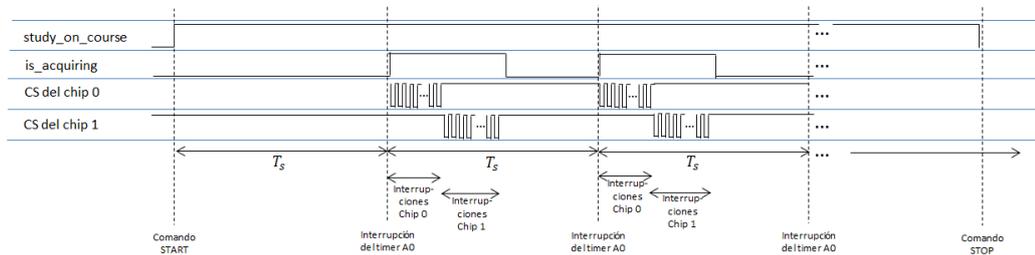


Figura 4.10: Diagrama de tiempo correspondiente a un estudio.

Cuando el timer A0 interrumpe se setea la bandera *is_acquiring* y se llama a la función *CONVERTburst()*, quien se encarga de cargar los comandos necesarios para iniciar una nueva tirada.

En todas las configuraciones se reparte la conversión de canales de forma equitativa, de manera que el chip 0 y el chip 1 muestreen la misma cantidad de canales⁵. Esto se hizo con la intención de paralelizar el trabajo, aunque por cuestiones de velocidad que se exponen en la sección 8.2.2, terminó siendo más conveniente que los integrados RHD2132 trabajen por turnos, es decir, que primero el chip 0 muestree todos los canales que se le piden y luego el chip 1 haga lo mismo. Este balance de trabajo también se hace con los canales de sincronismo. Si se trabaja con una

⁵En el caso de 21 canales el chip 1 muestrea un canal más que el chip 0.

Capítulo 4. Diseño del software embebido

cantidad par de canales de sincronismo, cada chip tendrá que muestrear la mitad del total. Si la cantidad a convertir es impar, el chip 0 siempre tiene que muestrear uno más que el chip 1.

A través de la rutina de interrupción de los dos puertos SPI se completa el muestreo de todos los canales solicitados. Una vez que esto ocurre, se utiliza la función *transfer_channel()* que se encarga de cargar los datos muestreados a *buffer_packet*. Una vez realizada esta tarea, se setea *is_acquiring* a 0 y se vuelve al loop principal.

La tirada debe guardarse ordenadamente en *buffer_packet* para su posterior procesamiento en la interfaz de usuario. Se guardan primero todos los canales convertidos del chip 0, luego todos los canales del chip 1, y luego los canales de sincronismo en orden de su numeración (ver figura 6.2). Esto se repite cada vez que ocurre una interrupción del timer A0. Al completarse un paquete, en el loop principal, se envía a la radio. Si $N_{tiradas}$ es la cantidad de tiradas que entran en un paquete, se envía un paquete a la radio cada $N_{tiradas}T_s$.

La figura 4.11 muestra el diagrama de flujo correspondiente a un estudio. Para no cargar el diagrama, el flujo de las funciones *CONVERTburst()* y *transfer_channel()* se presentan en el anexo G.2 (ver figuras G.4 y G.3b). El flujo de las rutinas de interrupción de los puertos SPI es presentado en el anexo G.2 (ver figura G.2).

En el funcionamiento normal de un estudio, los tres tipos de interrupciones ocurren en tiempos distintos, es decir, primero ocurre la interrupción del timer A0, luego todas las interrupciones de SPI del chip 0 y luego todas las interrupciones de SPI del chip 1. En la siguiente tirada el ciclo se repite. Esto se puede observar en la figura 4.10.

Si ocurre una falla por superposición en la adquisición de tiradas consecutivas, es decir, una interrupción del timer_A0 antes de que *is_acquiring* vuelva a 0, se descartan estas dos tiradas, enviando al PC dos tiradas nulas en su lugar (se carga el valor lógico correspondiente a 0V en todos los canales y señales de sincronismo), se limpian los buffers de la comunicación SPI, y se continúa normalmente con la adquisición. Cuando ocurre este tipo de fallas, el contador *sampleOverlap* se incrementa en 2. El valor de *sampleOverlap* se envía al finalizar el estudio reportando el total de tiradas perdidas.

Otro inconveniente que puede surgir es que se llene *buffer_packet*. Esto puede ocurrir si la velocidad a la que se adquieren las tiradas es mayor a la velocidad a la que se vacía *buffer_packet* al enviar los paquetes a la radio desde el loop principal. En este caso, se limpian todos los buffers del módulo remoto y se envía un paquete especial que avisa al PC que ocurrió este conflicto. Este paquete especial comienza con una "P" para indicar la pérdida. En el diagrama de la figura 4.11, este paquete

4.9. Funcionamiento del módulo remoto durante un estudio

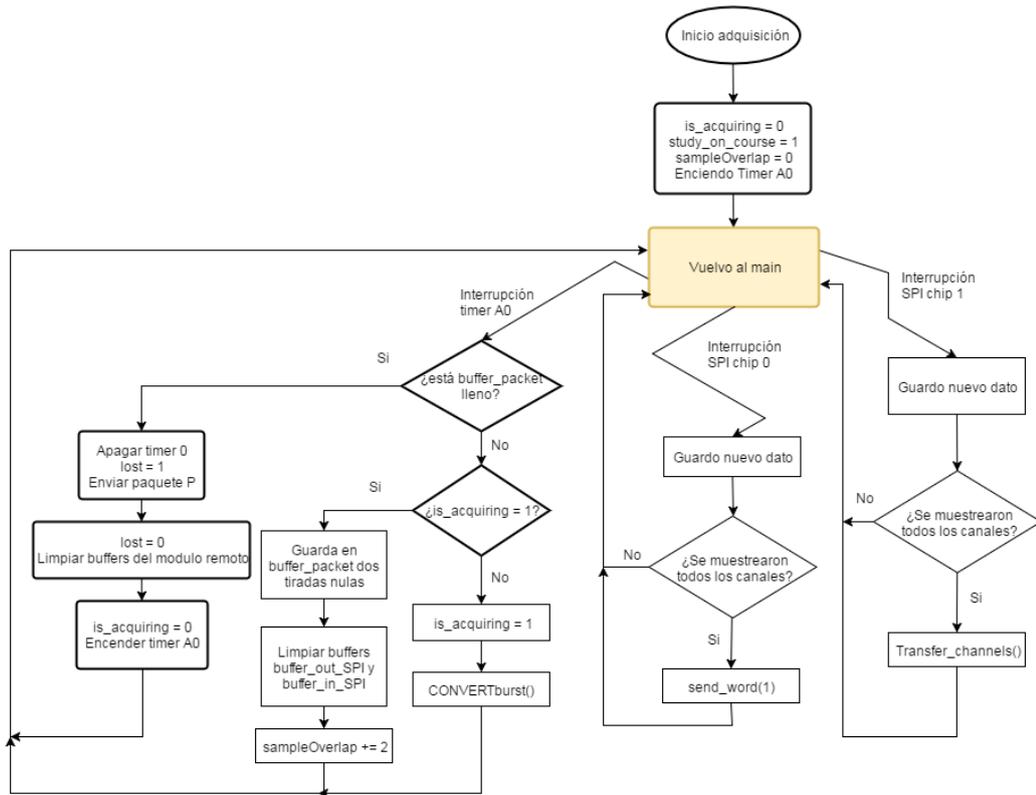


Figura 4.11: Diagrama de flujo del módulo remoto en medio de un estudio. Se recuerda que la función `send_word(1)` se utiliza para comenzar la adquisición con el chip 1 y que la función `transfer_channel()` se encarga de copiar la tirada al buffer de transmisión de paquetes a la radio (`buffer_packet`).

se identifica con el nombre de paquete P.

La gestión de pérdidas solo se preocupa del desbordamiento de `buffer_packet` puesto que para el otro tipo de falla el arreglo es interno y el usuario lo observa en la pantalla cuando se grafican todos los canales como una señal nula por un lapso de tiempo. En la sección 8.3.1 se estudia las causas de que ocurran estos tipos de pérdidas.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 5

Transmisión y recepción de datos

El objetivo de que el electroencefalógrafo sea inalámbrico es que se puedan realizar estudios con el paciente en movimiento dentro de una sala de aproximadamente $36m^2$, brindando mayor versatilidad al producto.

El protocolo de comunicación se elige considerando los siguientes compromisos:

- Bitrate¹ necesario (indicado en la Tabla 1.6 y 5.1).
- Bajo consumo de energía.
- Distancia máxima a la que se asegura transmisión de datos.
- Trabajar en una banda de frecuencia libre.

Especificaciones		Caso aceptable	Caso ideal
Throughput útil	Mayor que	128 kbps	1,12 Mbps
Distancia entre módulo remoto y PC	Mayor que	4 metros	6 metros

Tabla 5.1: Requerimientos para el transmisor/receptor.

El bitrate indicado en la tabla 5.1 se obtiene de considerar el caso de mayor transferencia de datos. En el peor caso se envían 64 canales más 6 canales de sincronismo disponibles, muestreados a $1kHz$ y codificando cada muestra con 16 bits, lo que implica un bitrate de:

$$bitrate_{ideal} = (64 + 6)canales \times 1000 \frac{muestras}{canal} \times 16 \frac{bit}{muestra} = 1120kbps$$

Para el caso aceptable se utilizan 16 canales, muestreados a $1000Hz$ y codificando cada muestra con 8 bits, lo que implica un bitrate de:

$$bitrate_{acceptable} = 16canales \times 1000 \frac{muestras}{canal} \times 8 \frac{bit}{muestra} = 128kbps$$

¹Tasa de bits por unidad de tiempo.

5.1. Opciones de protocolos

Se estudiarán principalmente el uso de tres protocolos de comunicación: Bluetooth, Wi-Fi y ZigBee. A continuación se resume las características que nos interesa analizar de cada uno de ellos.

■ ZigBee

- Tiene un consumo de $30mA$ mientras transmite y de $3\mu A$ en reposo [33].
- Opera en banda libre de 2,4 GHz [34].
- Bajo ciclo de trabajo lo que proporciona larga duración de la batería [34].
- Alcanza una velocidad en el aire de 250 kbps [33].
- Rango de transmisión máximo oscila entre 10 metros y 75 metros según el entorno a una potencia de 0 dBm. [33].

■ Bluetooth

- Tiene un consumo de $40mA$ mientras transmite y de $200\mu A$ en reposo [33].
- Opera en banda libre de 2,4 GHz.
- Dependiendo del tipo de protocolo Bluetooth, alcanza velocidades en el aire entre 1 Mbps y 3 Mbps [35]. La transferencia de 1 Mbps es alcanzada por ejemplo por Bluetooth Low Energy, que es un tipo de protocolo Bluetooth que optimiza al Bluetooth clásico en consumo de energía.
- Rango de transmisión máximo de $10m$ a una potencia de 4 dBm [35].

■ Wi-Fi

- Tiene un consumo en transmisión de $272mA$ y de $6mA$ en reposo (ver tabla 2.4).
- Opera en banda libre de 2,4 GHz.
- Alcanza velocidades en el aire superiores a 11 Mbps. Con el estándar IEEE 802.11b se alcanza la velocidad de 11 Mbps, con el estándar IEEE 802.11g se alcanza una velocidad de 54 Mbps y finalmente con el estándar IEEE 802.11n se alcanza una velocidad de 300 Mbps [36].
- Rango de transmisión máximo de $50m$. [37].

Lo primero a destacar es que ZigBee no alcanza la velocidad de transmisión de datos requeridas, puesto que su velocidad en el aire es de 250 kbps y se necesita un bitrate útil de al menos 1120 kbps para cubrir el caso ideal.

5.2. Problemas con Bluetooth

Se analizan tres posibles versiones de Bluetooth: Bluetooth 4.0 + EDR, Bluetooth 4.0 + HS y Bluetooth 4.0 + LE también conocido como Bluetooth Low Energy (BLE).

5.2.1. Bluetooth 4.0 + LE

Esta versión de Bluetooth optimiza al Bluetooth clásico en consumo de energía (para más información ver [38]). Como se mencionó en la sección 5.1, la velocidad en el aire es de 1 Mbps, sin embargo, este valor no es la carga útil de datos, sino que este valor tiene en cuenta toda la transferencia de datos incluyendo los encabezados del protocolo de Bluetooth, retransmisiones, etc.

Si el usuario utiliza el protocolo BLE transmitiendo datos en una sola dirección en condiciones óptimas, tendrá 270 kbps disponibles. A su vez, por limitaciones de Hardware y condiciones del ambiente en que se tiene interferencia de otros protocolos, que trabajan en el mismo ancho de banda, se ve aún más limitado en el bitrate efectivo. En una condición de ambiente normal, utilizando BLE entre dos módulos que no envían paquetes de reconocimiento (paquetes ACK), se logra un bitrate de 60 kbps [39]. En [40] se calcula el bitrate máximo entre varios dispositivos que utilizan BLE a partir de la información entregada por cada uno de los fabricantes.

En conclusión, el throughput del protocolo BLE se encuentra muy por debajo de los requerimientos del sistema.

5.2.2. Bluetooth 4.0 + EDR

Esta versión de Bluetooth optimiza a la versión Bluetooth clásica en velocidad en el aire de transferencia de datos. Mientras que Bluetooth clásico logra 1 Mbps de velocidad en el aire, Bluetooth 4.0 + EDR (Enhanced Data Rate) alcanza una velocidad en el aire de 3 Mbps [35].

Al igual que ocurre con BLE, la velocidad en el aire no se corresponde con el throughput disponible para el usuario. [41] calcula que el throughput en Bluetooth 4.0 + EDR es de 2,1 Mbps. Si bien el sistema necesita un bitrate de 1,12 Mbps para alcanzar el caso ideal, el throughput calculado por [41] no considera ningún tipo de reconocimiento de error, de corrección de error o de retransmisiones. Esto no se acerca a las características del sistema puesto que es necesario asegurar que no hay pérdida de paquetes.

Analizando algoritmos de corrección de error o de retransmisiones se encontró que en el mejor caso, utilizando FEC 2/3, se llega a un throughput de 477,8 kbps [42] que no alcanza el bitrate requerido para el caso ideal.

5.2.3. Bluetooth 4.0 + HS

Bluetooth 4.0 + HS (high speed) es una versión de Bluetooth que soporta velocidades de transferencia de hasta 24 Mbps. Ocurre sin embargo, que esta versión no es únicamente Bluetooth, puesto que utiliza enlaces Bluetooth para la negociación y el establecimiento mientras que para el tráfico de datos de alta velocidad realiza el enlace a través de Wi-Fi [35]. Esta versión por tanto tiene las desventajas del consumo de Wi-Fi en transmisión y además termina siendo un protocolo mixto que no es tan estándar como lo es Bluetooth clásico o Wi-Fi.

5.3. Wi-Fi

Wi-Fi es un protocolo de comunicación que entre otras bandas, opera en la banda libre de 2,4GHz. El ancho de banda que tiene disponible en esta banda se divide en 11 canales que se pueden superponer como se puede ver en la figura 5.1. Por ejemplo si se tiene una comunicación en el canal 1, esta tiene interferencia de otras comunicaciones que estén en los canales cercanos (2, 3, 4 ó 5).

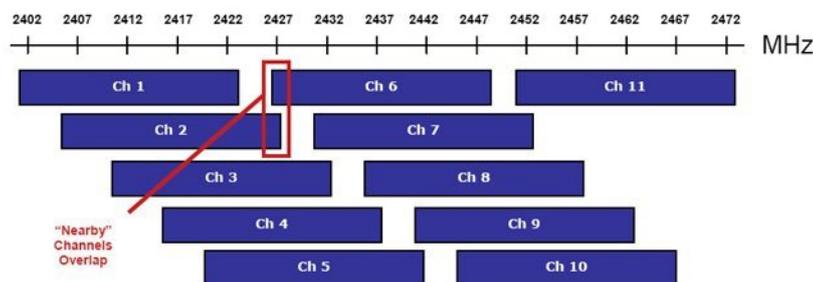


Figura 5.1: Canales Wi-Fi dentro de la banda de 2,4GHz (imagen extraída de [43]).

Esta desventaja no se da solo con otras comunicaciones en un canal cercano, sino que si hay otra comunicación en el mismo canal, puede producir colisiones o que por momentos no se pueda usar el canal porque otra comunicación lo está usando.

Trabajar con Wi-Fi implica las siguientes ventajas con respecto a Bluetooth 4.0 + HS:

- El throughput alcanzable con Wi-Fi es igual o en algunos casos mayor que el de Bluetooth 4.0 + HS.
- Es un protocolo muy difundido y utilizado, por lo cual muchos problemas ya fueron detectados y solucionados.
- Tiene mayor rango de transmisión.

Trabajar con Wi-Fi implica las siguientes desventajas:

- Tiene un consumo mayor que Bluetooth.
- Imposibilidad de usar un canal si otro dispositivo lo está usando en ese momento.
- Interferencia entre canales cercanos.

Se decidió utilizar Wi-Fi por ser un protocolo muy difundido, utilizado y que el throughput útil es mayor al requerido. No es un protocolo de bajo consumo y esto impacta en la autonomía. En la sección 8.3.2 se puede ver que el sistema tiene una autonomía de más de un día que era el objetivo ideal.

La red es creada por un router, ya que el router da varias facilidades. Por ejemplo permite elegir qué canal de Wi-Fi usar y configurar una IP fija para la radio CC3100, lo cual es importante para utilizar el código de SimpleLink² en el cual hay que setear la IP que va a tener la radio antes de compilar.

Dado que la comunicación entre el router y el módulo remoto es Wi-Fi, es recomendable que la conexión entre el router y el PC sea Ethernet³, por la imposibilidad de usar un canal si otro dispositivo lo está usando en ese momento. En la sección 8.1.3 se desarrolla el efecto de esta desventaja y como influye en la comunicación.

Wi-Fi tiene varios canales para transferencia de datos. En algunos casos existen canales con otras comunicaciones con una potencia considerable utilizando el canal, llamados en adelante “canales compartidos”, y “canales libres” a los canales en los cuales no hay otra comunicación.

Como protocolo de transporte se va a usar TCP y no UDP. Esto se debe a que TCP resuelve varios problemas, que si utilizáramos UDP se tendrían que resolver en capa de aplicación, por ejemplo que los datos lleguen ordenados y se repitan en caso de pérdidas.

²Simplelink es una librería con drivers que se utiliza para controlar la radio CC3100.

³Ethernet es un estándar de comunicación, cableado.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 6

Interfaz de usuario

Para la implementación de la interfaz de usuario, se realizó un programa en MatLab para que el usuario sea capaz de controlar al sistema a partir del PC. La interfaz de usuario debe ser capaz de:

- Recibir los datos de las señales de EEG digitalizadas provenientes del módulo remoto.
- Mostrar dichos datos en tiempo real a partir de una ventana gráfica.
- Guardar un estudio en un archivo para poder visualizarlo y procesarlo cuando el usuario desee.
- Cargar archivos de estudios anteriores.
- Configurar tanto la interfaz de usuario como el módulo remoto.
- Iniciar o detener un estudio.

La interfaz tiene varios comandos y parámetros para controlar al sistema. La tabla 6.1 resume los comandos con los que el usuario puede configurar al módulo remoto a través del PC. En el anexo H.2 se detallan algunos de estos comandos y el formato del paquete que se envía al módulo remoto para que los interprete.

En la interfaz de usuario se elige también:

- Identificador del módulo remoto.
- El ancho de ventana gráfica.

6.1. Protocolo de comunicación

Para que la comunicación se inicie, es necesario que el PC esté conectado a la red que genera el router del sistema cuyo nombre es “wEEG”. La red cuenta con

Capítulo 6. Interfaz de usuario

Comando	Descripción del comando
START	Solicita el inicio de un estudio de más de un canal
STOP	Solicita la detención del estudio en curso
ONE_CHANNEL	Solicita el inicio un estudio de un canal
CONFIG	Envía parámetros de configuración al módulo remoto
EA	Solicita que el módulo remoto envíe una respuesta mostrando su configuración actual
CALIBRATE	Solicita la calibración de los conversores A/D del AFE
TIE	Solicita al módulo remoto que inicie un test de impedancia
RESET	Solicita el reinicio del módulo remoto

Tabla 6.1: Descripción de los comandos disponibles desde la interfaz de usuario que configuran al módulo remoto.

contraseña y WAP2¹. El PC es el cliente TCP, encargándose de iniciar la comunicación con el módulo remoto que actúa como servidor.

Dado que se tiene disponibles dos radios CC3100, el router setea la IP de cada uno de los módulos remotos. Al inicio, el usuario debe elegir con cuál módulo remoto conectarse. De esta manera MatLab sabe a que IP se tiene que conectar.

Una vez establecida la comunicación, la interfaz de usuario envía comandos de configuración al módulo remoto, dejando al sistema en una configuración por defecto como se menciona en la sección 6.3. En caso de éxito, ya se puede comenzar con el estudio o modificar las condiciones iniciales.

Al enviarse un comando, la interfaz introduce un tiempo muerto de un segundo, para dar el tiempo necesario al módulo remoto de procesar el comando. En caso de que se deba recibir una respuesta, la interfaz queda pendiente por ésta hasta ser recibida. En cualquier caso, si el usuario da la orden de enviar un nuevo comando, este se enviará luego de pasado el tiempo muerto y de recibida la respuesta en caso de ameritarlo.

Una vez que se inicia el estudio, se crean dos archivos: en uno se guarda la configuración del estudio y en otro los datos adquiridos. El nombre de estos archivos será la fecha y la hora en la cual se inició el estudio. Al salir de la interfaz, se envía el comando “SALIR” hacia el módulo remoto, reiniciándolo de manera que se pueda establecer una nueva comunicación.

Los paquetes recibidos del módulo remoto pueden tener 4 tipos de datos:

- Muestras procesadas.
- Señalización de fallas por desbordamiento de buffer.
- Señalización de fallas en la configuración del módulo remoto.

¹Wi-Fi Protected Access 2 (WAP2), es un sistema para proteger las redes inalámbricas.

6.2. Descripción general del funcionamiento de la interfaz de usuario

- Respuesta de comandos.

Los paquetes del primer tipo se reciben periódicamente durante el estudio. Estos paquetes comienzan con la letra “S”, seguida de un entero de 8 bits que indica el número de paquete (timestamp) y a continuación las muestras procesadas. Los paquetes de señalización de fallas por desbordamiento de buffer comienzan con la letra “P” (de pérdida), en la sección 6.7 se detalla la respuesta de la interfaz de usuario ante estas pérdidas. Los paquetes que señalizan las fallas en la configuración del módulo remoto comienzan con la letra “W” desplegando la causa de la falla en la línea de comandos de MatLab y deteniendo el estudio a diferencia de las fallas por desbordamiento de buffer. Por último, los paquetes que se reciben como respuesta de comandos llegan luego de enviado el comando en cuestión como se mencionó en el párrafo anterior. El formato de la respuesta se menciona en el anexo H.2.

6.2. Descripción general del funcionamiento de la interfaz de usuario

La figura 6.1 muestra el diagrama de flujo de la interfaz de usuario de manera resumida.

Al comienzo, se inicializa la interfaz de usuario seteando la configuración por defecto, luego se intenta comunicar con el módulo remoto. Cuando se logra la comunicación, se despliega el **menú principal** en el cual el usuario puede modificar la configuración del sistema o iniciar un nuevo estudio con los parámetros actuales de la interfaz de usuario.

El usuario puede modificar la configuración, de los 10 parámetros que le brinda el sistema (ver figura 6.1). Si el usuario presiona “Enviar” se envía un comando CONFIG al módulo remoto con los nuevos valores de los parámetros modificados en el menú “Configuración”, seguido por un EA para determinar si la modificación fue exitosa. Al presionar “Iniciar un nuevo estudio”, se envía un CONFIG con los valores actuales de la interfaz, seguido por un EA del mismo modo que en el caso anterior. En los dos casos se verifica que la configuración enviada y la respuesta recibida sean coherentes. En el caso de que no sea así, se vuelven a enviar CONFIG y EA. Si al tercer intento no se configuró correctamente el módulo remoto, se despliega en pantalla un mensaje de error, pudiéndose reconfigurar nuevamente los parámetros.

Si se corrobora que el módulo remoto está bien configurado se abre la **ventana gráfica**. En esta ventana el usuario puede iniciar o detener un estudio, graficar cuántos y cuáles canales desee y enviar comandos al módulo remoto. Cuando se abre la ventana gráfica, el usuario tiene disponibles la ventana gráfica y el menú

Capítulo 6. Interfaz de usuario

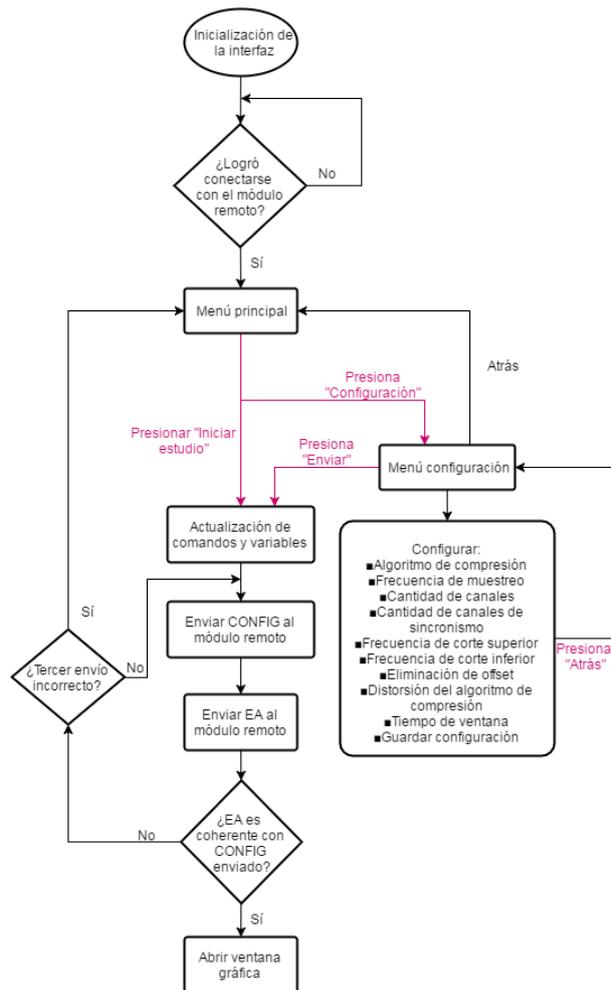


Figura 6.1: Diagrama de flujo básico de la interfaz de usuario.

principal, pudiendo usar cualquiera de las dos simultáneamente.

Al presionar “Salir” se cierra la interfaz de usuario y se resetea el módulo remoto. Al presionar “X”², va a depender de la ventana en la que se encuentra la acción que se ejecuta. En algunos casos presionar “X” cierra la interfaz de usuario y en otros casos “X” funciona como el botón “Atrás”, más adelante, en la descripción de cada ventana, se especifica para cada ventana en concreto.

²La “X” hace referencia al botón de cerrar ventana.

6.3. Inicialización de la interfaz y conexión con el módulo remoto

En la inicialización se elige una configuración por defecto, de manera que si el usuario decide iniciar un estudio antes de configurar, el sistema la inicia de la siguiente manera:

- Cantidad de canales: 64.
- Cantidad de canales de sincronismo: 0.
- Frecuencia de muestreo f_s en 500 Hz.
- Frecuencia de corte superior del filtro pasabanda f_H en 250 Hz.
- Frecuencia de corte inferior del filtro pasabanda f_L en 0,1 Hz.
- Algoritmo ACEEG desactivado.
- Tiempo de ventana para visualizar las muestras de 1 segundo.

Una vez elegido el módulo remoto con el menú mostrado en el anexo H (ver figura H.1), el PC intenta comunicarse con el módulo remoto. Mientras no se logre la comunicación se despliega la cantidad de intentos de conexión que se tuvo. Se hace un nuevo intento de conexión cada 5 segundos por tiempo indefinido. En el anexo H.3 se explica las posibles causas de la falla de la conexión. Si se logra la comunicación, se despliega el menú principal.

6.4. Funcionamiento menú principal

El menú principal se observa en la figura 6.2.

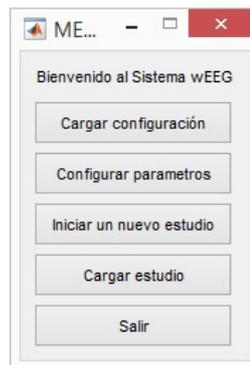


Figura 6.2: Menú principal de la interfaz de usuario.

En menú principal el usuario tiene 4 opciones distintas:

Cargar configuración: se carga la última configuración guardada en el archivo "Parametros.mat". En caso de que el archivo se haya eliminado o nunca se

Capítulo 6. Interfaz de usuario

haya creado, la interfaz ingresa automáticamente a “Configurar parámetros”. En caso de que exista el archivo se actualizan las variables y se intenta configurar al módulo remoto como se muestra en la figura 6.1.

Configurar parámetros: se cambia el valor a distintos parámetros del sistema. En la sección 6.5 se detalla las distintas opciones de variables a configurar.

Iniciar nuevo estudio: se intenta configurar al módulo remoto y se abre la ventana gráfica como se muestra en la figura 6.1.

Cargar estudio: se carga un estudio guardado anteriormente para visualizarlo y realizar el tratamiento deseado.

Salir: cierra la interfaz de usuario y reinicia al módulo remoto. Al presionar “X” se logra el mismo efecto.

6.5. Configurar parámetros

La figura 6.3 muestra el menú de configuración de parámetros. En él se observan los distintos parámetros que se pueden setear y otros botones que se detallan a continuación.

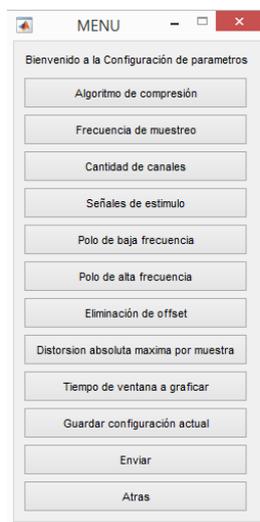


Figura 6.3: Menú principal de la interfaz de usuario.

Al presionar “Atrás”, vuelve al menú principal. Al presionar “X” se cierra la interfaz de usuario y se reinicia el módulo remoto.

6.5.1. Algoritmo de compresión

Se despliega un menú en el cual el usuario puede elegir si desea utilizar el algoritmo de compresión ACEEG o no. Luego puede apretar “Atrás” o “X” para volver al menú “Configurar parámetros”.

6.5.2. Frecuencia de muestreo

Para cambiar la frecuencia de muestreo, el usuario debe escribir en la ventana de comandos de MatLab, un valor de frecuencia entero positivo entre 100 Hz y 20000 Hz y presionar ENTER para confirmar. En caso de que ingrese un número que no esté en el rango o que escriba caracteres que no sean números (letras y/o símbolos), el valor de frecuencia no es modificado y el usuario tiene la oportunidad de ingresar otro número. En caso de que no ingrese nada, no se modifica la frecuencia de muestreo y vuelve al menú de configuración de parámetros.

La figura 6.4 muestra la ventana de comandos donde debe escribirse la frecuencia de muestreo.

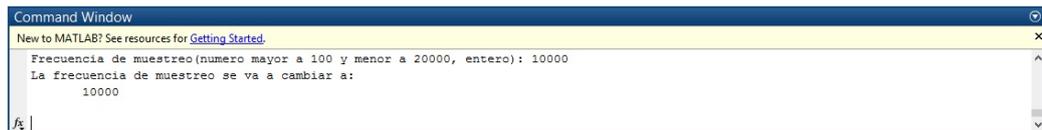


Figura 6.4: Menú para indicar la configuración de canales para el próximo estudio a realizar.

6.5.3. Cantidad de canales

Se despliega un menú en el cual puede elegir una de las 4 configuraciones posibles de canales: 64, 21, 4 ó 1. Si elige la configuración de 1 canal, se debe indicar en la ventana de comandos de MatLab, el número de canal que desea adquirir según la tabla 7.1. La forma de proceder al ingresar un número es la misma que en la frecuencia de muestreo. Luego de elegir en qué condiciones quiere trabajar, puede apretar “Atrás” o “X” para volver al menú “Configurar parámetros”.

6.5.4. Señales de sincronismo

Se despliega un menú que permite elegir cuántas señales de sincronismo desea muestrear (hasta 6). Luego de elegir en qué condiciones quiere trabajar, puede apretar “Atrás” o “X” para volver al menú “Configurar parámetros”.

Si se configura para utilizar N_S señales de sincronismo, se utilizan las primeras N_S según la tabla 6.2. Cada uno de los pines para señales de sincronismo se encuentran en el conector CON_EST de la placa AFE (ver anexo D) y tiene un número asociado.

Capítulo 6. Interfaz de usuario

Número de la señal sincronismo	Pin en el conector CON_EST
1	pin 8
2	pin 5
3	pin 7
4	pin 4
5	pin 6
6	pin 3

Tabla 6.2: Correspondencia de las señales de sincronismo con los pines del conector CON_EST en la placa AFE.

6.5.5. Frecuencia de corte superior

El usuario puede elegir el valor de la frecuencia de corte superior del filtro pasabanda del AFE que permite filtrar las señales de EEG. La frecuencia de corte superior puede tomar 17 valores distintos, siendo 100 Hz el más bajo y 20 kHz el más alto. En el anexo H (ver figura H.6) se muestra el menú que se despliega. Luego de elegir en qué condiciones quiere trabajar, puede apretar “Atrás” o “X” para volver al menú “Configurar parámetros”.

6.5.6. Frecuencia de corte inferior

Permite elegir el valor de la frecuencia de corte inferior del filtro pasabanda del AFE que permite filtrar las señales de EEG. La frecuencia de corte inferior puede tomar 25 valores distintos, siendo 0,1 Hz el más bajo y 500 Hz el más alto. En el anexo H (ver figura H.6) se muestra el menú que se despliega. Luego de elegir en qué condiciones quiere trabajar, puede apretar “Atrás” o “X” para volver al menú “Configurar parámetros”.

6.5.7. Eliminación de offset

Despliega una ventana que permite elegir si utilizar un filtro pasaalto extra para eliminar el offset que agregan los amplificadores del AFE. Este filtro agrega un polo simple en $f_{eo} = k_{frecDCO} f_{sample}$ siendo f_{sample} la frecuencia con la que se envían instrucciones para la conversión de canales y $k_{frecDCO}$ una constante determinada desde el módulo remoto cuyo valor es $3,886 \times 10^{-5}$. Luego de elegir en qué condiciones quiere trabajar, puede apretar “Atrás” o “X” para volver al menú “Configurar parámetros”.

6.5.8. Distorsión absoluta máxima por muestra

El usuario podrá elegir un valor entero de distorsión entre 0 y 16. El parámetro de distorsión es un parámetro del algoritmo de compresión. Luego de elegir en qué condiciones quiere trabajar, puede apretar “Atrás” o “X” para volver al menú “Configurar parámetros”.

6.5.9. Tiempo de ventana a graficar

Para modificar el tiempo de visualización de la ventana gráfica debe escribir en la ventana de comandos de MatLab un número positivo que indica el tamaño de ventana que desea visualizar. La precisión del número será de 0,01, es decir, si el número que escribe es T , el tiempo de ventana es seteado en $\frac{\text{round}(100 \times T)}{100}$ ³. Debe presionar ENTER para confirmar. La forma de proceder al ingresar un número es la misma que en la frecuencia de muestreo.

6.5.10. Guardar configuración

Guarda (o sobrescribe) los valores actuales de los parámetros en el archivo “Parametros.mat”. Esto puede ser útil para casos en los que se realiza el mismo estudio varias veces y el usuario necesita tener guardada la configuración para otros estudios similares.

6.5.11. Enviar

Se envía un comando CONFIG al módulo remoto con los valores actuales de los parámetros como muestra la figura 6.1. Seguido por un EA para determinar si la modificación fue exitosa. Luego se verifica que la configuración enviada y la respuesta recibida sean coherentes. En el caso de que no sea así se vuelven a enviar CONFIG y EA. Si al tercer intento no se configuró correctamente el módulo remoto, se despliega en pantalla un mensaje de error, pudiéndose reconfigurar nuevamente los parámetros. En caso de que se configuró bien se abre la ventana gráfica como se muestra en a figura 6.1.

6.6. Ventana gráfica

En la ventana gráfica, el usuario visualizará las señales de EEG y de sincronismo en tiempo real (ver figura 6.5). La ventana gráfica también permite enviar comandos al módulo remoto o configurar a la interfaz con los botones a la izquierda de la gráfica.

En el recuadro de la parte superior, a la izquierda se muestra la configuración actual del sistema. Debajo, en los cuadros de canales, se indica qué canales de EEG se van a visualizar en la pantalla y por debajo de esto cuáles canales de sincronismo se van a visualizar. El usuario puede escribir directamente en ambos cuadros para quitar o agregar canales que desea visualizar dentro del rango de la configuración de canales actual. Cabe destacar que si se está en medio de un estudio y se modifica los canales que se desea visualizar, es necesario que se detenga el estudio y se comience de vuelta para que el cambio tenga efecto. La lista de canales en estos cuadros debe estar numéricamente ordenada (no está permitido escribir “2 1”).

³La función *round* redondea al valor entero mas cercano.

Capítulo 6. Interfaz de usuario

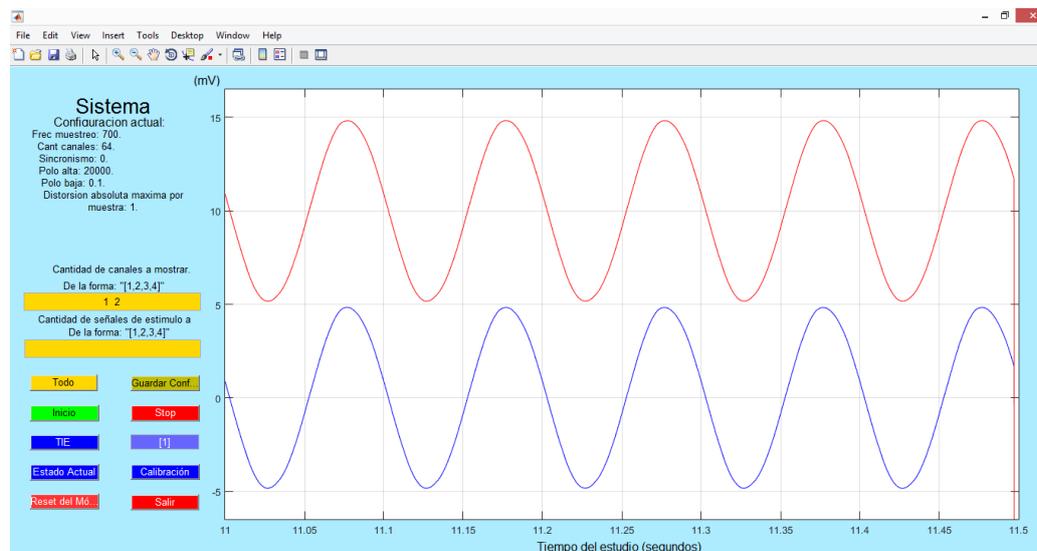


Figura 6.5: Ventana gráfica en medio del estudio de una señal sinusoidal de 10Hz y 10mVpp. Como se observa en el extremo superior izquierdo de la imagen, la configuración es de 64 canales con una frecuencia de muestreo de 700Hz.

Los 10 botones por debajo se detallan a continuación:

Todo: se grafican todos los canales (de EEG y de sincronismo) según la configuración actual. Si se está por ejemplo en el caso de 64 canales con 6 señales de sincronismo, se mostrarán en la gráfica 70 señales distintas.

Guardar configuración: se brinda en la ventana gráfica la misma función que la presentada en la sección 6.5.10.

Inicio: se inicia un estudio con los parámetros actuales de la interfaz de usuario. El estudio puede ser con las configuraciones de 1, 4, 21 o 64 canales, o puede ser el test de impedancia de electrodos (ver comando TIE). A medida que se grafican los canales deseados en la ventana gráfica para visualizarlos en tiempo real, se guardan los datos adquiridos en archivos que MatLab es capaz de abrir.

Stop: si se estaba en medio de un estudio, el comando STOP hace que se detenga. Cuando el usuario detiene un estudio en curso, en la ventana de comandos de MatLab se muestra el porcentaje del tiempo que hubo pérdida de datos. Esta pérdida de datos se explica en la sección 6.7.

TIE: el usuario debe elegir un canal donde realizar el test de impedancia de electrodos. El valor debe incluirse en el cuadro de texto que se encuentra a la derecha del botón asociado a este comando. El número a ingresar debe de estar entre 1 y 32. El test se realiza simultaneamente sobre 2 canales (canal x y canal $x + 32$) (ver anexo E). En caso de que se desee realizar el test con un canal del

6.7. Respuesta de la interfaz de usuario ante pérdidas del sistema

chip 1 (por ejemplo, canal 54), se debe de restar 32 a la posición del canal (en el ejemplo se debería de ingresar 22). Una vez elegido el canal, se configura la interfaz de usuario para realizar el test, y al presionar el botón “Inicio” se envía al módulo remoto el comando TIE con el canal elegido. Al comenzar el estudio se muestran en pantalla la señal del pin elegido de ambos chips (en el ejemplo se graficarían los canales 22 y 54). Como la corriente es conocida, se puede calcular a partir de aquí la impedancia de los electrodos en estos dos canales (en el anexo E se detalla el funcionamiento del test de impedancia de electrodos).

Estado actual: se envía el comando EA al módulo remoto y se despliega el resultado de la configuración actual del mismo en la parte superior izquierda de la ventana gráfica. En el anexo H.2 se describe como es la respuesta del módulo remoto a un comando EA.

Calibración: se envía el comando CALIBRATE al módulo remoto. Cuando el módulo remoto recibe este comando calibra a los conversores A/D del AFE.

Reset del algoritmo: este botón se presiona si al utilizar el algoritmo ACEEG, se observa que los datos que llegan a la interfaz de usuario son señales sin sentido, que puede deberse a una des-sincronización del algoritmo. Con este comando se limpian los parámetros de la compresión (en el módulo remoto) y de la descompresión (en la interfaz de usuario) pero no se detiene el estudio. Si se presiona el botón y no se está utilizando el algoritmo, la interfaz de usuario no permite el envío de este comando al módulo remoto.

Salir: al presionar este botón se cierra la interfaz de usuario y se reinicia el módulo remoto independientemente de si se está o no en un estudio.

En lo que refiere a las señales que se grafican en la ventana gráfica, las mismas están distanciadas de 10 mV como se observa en la figura 6.5.

Al presionar “X” en la ventana gráfica, sólo se cierra dicha ventana y la interfaz de usuario vuelve al menú principal. En este caso no se detiene el estudio por lo que si se cierra esta ventana sin detener el estudio, la interfaz va a seguir graficando y no se va a poder detener. Si sucede esto, debe enviar un comando “Salir” (disponible en el menú principal) para cerrar la comunicación y reconectar el sistema.

6.7. Respuesta de la interfaz de usuario ante pérdidas del sistema

Como se mencionó en la sección 4.9 las pérdidas de datos en el sistema pueden deberse a 2 tipos de falla.

Capítulo 6. Interfaz de usuario

- Superposición en la adquisición de tiradas consecutivas.
- Desbordamiento del buffer de transmisión de paquetes del microcontrolador a la radio (buffer_packet).

Una falla de superposición en la adquisición de tiradas consecutivas ocurre cuando se comienza a adquirir una tirada sin haber finalizado con la adquisición de la tirada anterior. En este caso, como se menciona en la sección 4.9, se envían dos tiradas nulas las cuales la interfaz de usuario grafica normalmente.

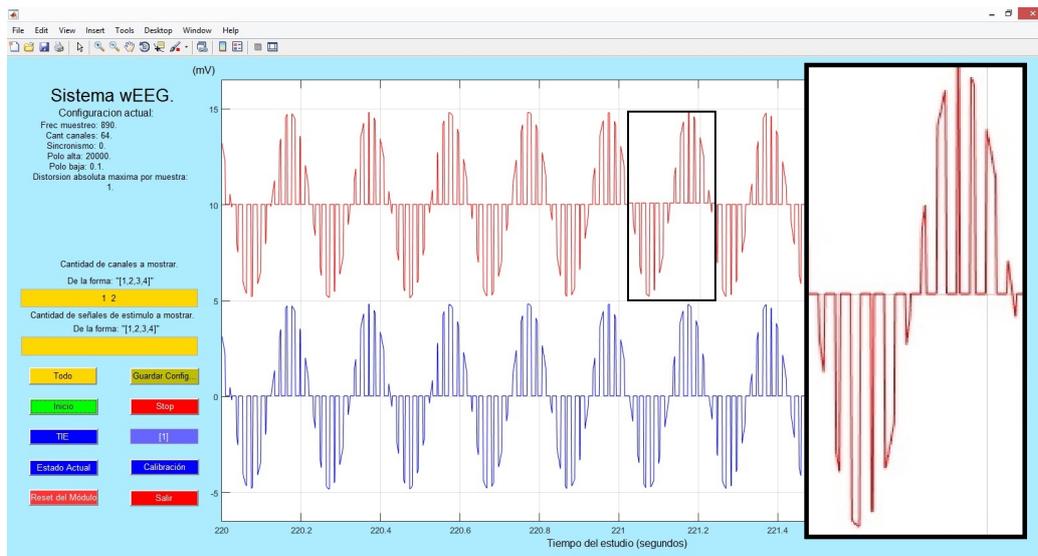


Figura 6.6: Recepción de una señal sinusoidal de 5Hz y 10mVpp con fallas debidas a superposición de adquisición de tiradas consecutivas. La configuración de canales es 64 y la frecuencia de muestreo es 890Hz.

Cuando ocurre una falla por desbordamiento del buffer entre el microcontrolador y la radio, el módulo remoto envía un paquete que comienza con la letra “P” (en código ASCII) como se menciona en la sección 6.1. Cuando la interfaz de usuario identifica este paquete, se realiza una adaptación temporal a partir del reloj interno del PC y seteando todos los valores de tirada desde la última recibida en 0V.

6.7. Respuesta de la interfaz de usuario ante pérdidas del sistema

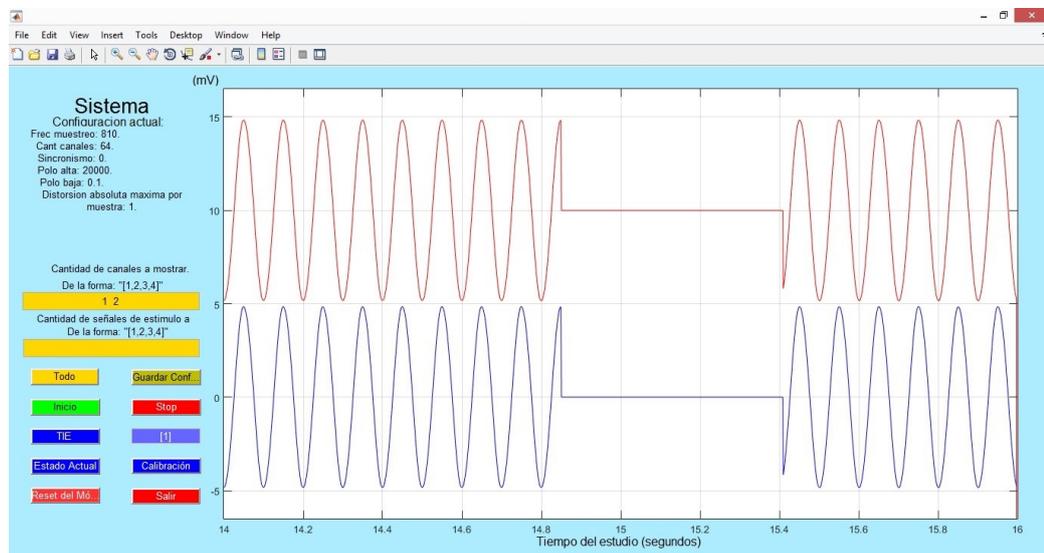


Figura 6.7: Recepción de una señal sinusoidal de 10Hz y 10mVpp con pérdidas debidas a desbordamiento de buffer. Las pérdidas ocurren entre los 14,8s y 15,4s en los dos canales que se grafican. La configuración de canales es 64 y la frecuencia de muestreo es 810Hz.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 7

Montaje del módulo remoto

7.1. Electrodo y casco

Se utilizaron electrodos húmedos de ECG (Ag/AgCl) debido a la oferta existente en el mercado. Si bien uno de los objetivos que se plantearon fue utilizar electrodos secos, la gestión y compra de los mismos se demoró a tal punto de que no fue posible utilizarlos. Los electrodos húmedos (ver figura 7.1) son adhesivos y tienen dimensiones de: $36 \times 50\text{mm}$ el adhesivo y 10mm de diámetro el conductor. Estos electrodos son mas grandes que los de EEG y ocupan mucha área para usar en la cabeza. Debido a esto, es difícil adquirir señales en zonas del cuero cabelludo y trabajar con muchos canales, aunque se podrían recortar, perdiendo su efectividad adhesiva.



Figura 7.1: Electrodo húmedo de cloruro de plata (Ag/AgCl).

La fabricación del casco está incompleta, como trabajo a futuro se piensa terminar el diseño, haciendo los orificios donde irán los electrodos en las posiciones correctas. Un diseño parcial del casco se observa en las figuras 7.2a y 7.2b.



(a) Diseño del casco, vista frontal.



(b) Diseño del casco, vista de perfil.

Figura 7.2: Diseño parcial del casco sin los orificios de colocación de los electrodos.

7.2. Carcasa

Los bloques “AFE”, “Microcontrolador” y “RADIO” de la figura 1.3 se corresponden con tres placas conectadas como se muestra en la figura 7.3. A la izquierda de estas se marca la posición del portapilas que alimenta a las tres placas del módulo remoto.



Figura 7.3: Diagrama de la conexión de las placas en el módulo remoto.

Para el “Microcontrolador” se utilizó la placa de desarrollo MSP-EXP432P401R de Texas Instruments y para la radio la placa de desarrollo CC3100BOOST de Texas Instruments. Los integrados RHD2132 junto con los convertidores DC/DC ADP150 y TPS736 están soldados en la placa “AFE” cuyo diseño fue explicado en el capítulo 3.

El módulo remoto se encuentra dentro de una carcasa que protege mecánica y eléctricamente a las tres placas. Las figuras 7.4a y 7.4b muestran al módulo remoto dentro y fuera de la carcasa.

7.3. Instalación en el paciente

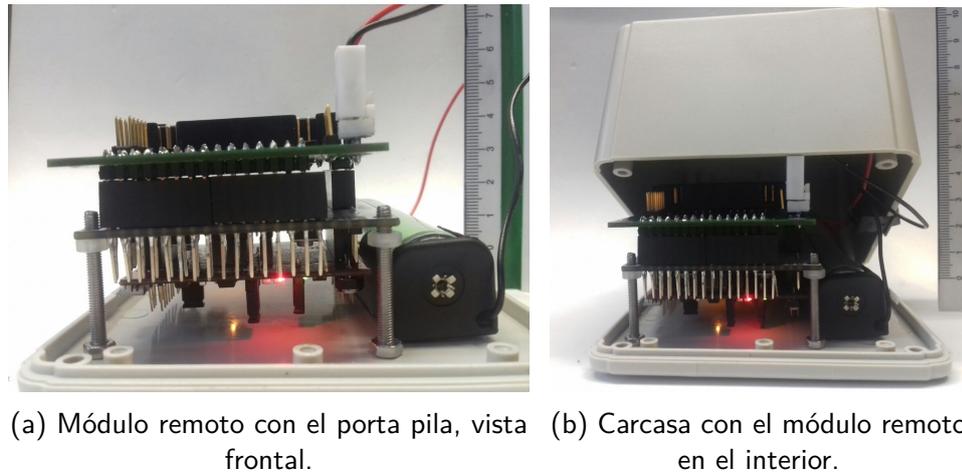


Figura 7.4: Carcasa y su contenido.

La carcasa posee en una de sus caras laterales un interruptor que permite cortar la alimentación del módulo remoto.

7.3. Instalación en el paciente

El módulo remoto puede configurarse para 64, 21, 4 y 1 canal. Para todas las configuraciones, el sistema mide las señales contra un mismo electrodo de referencia que se coloca en zonas que presentan un potencial estable. Este se utiliza con el fin de medir las señales de forma diferencial ¹, eliminando las señales en modo común del paciente, como pueden ser potenciales asociados a movimientos oculares, latidos del corazón, que no son señales provenientes de la actividad cerebral del paciente.

Las figuras 7.5, 7.6 y 7.7 muestran las conexiones de los electrodos para 64, 21 y 4 canales. Desde la interfaz de usuario se utiliza la numeración de canales del 1 al 64. En las tablas 7.1, 7.2 y 7.3 se detalla la relación entre la numeración utilizada por la interfaz y la posición de los electrodos de las figuras. Si se decide trabajar con un canal, el usuario podrá elegir cualquiera de los 64 canales de la configuración de la figura 7.5.

Cabe destacar que el número con que se asocia cada electrodo es un cambio en la interfaz de usuario, y no debe realizarse ninguna reconexión, salvo en el caso de T9 a A1 si se pasa de 64 canales a 21 canales (ver figuras 7.5 y 7.6).

¹A la entrada del módulo remoto se cuenta con 64 amplificadores diferenciales. Todas las entradas inversoras de los amplificadores están conectadas al electrodo de referencia y las entradas no inversoras están conectadas a los distintos electrodos que miden las señales neuronales.

Capítulo 7. Montaje del módulo remoto

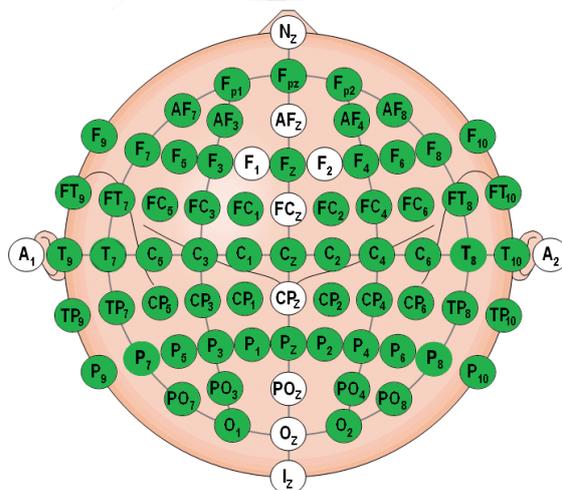


Figura 7.5: En color se muestra las posiciones que se utilizan con la configuración de 64 canales (imagen extraída de [44] y luego modificada).

N°	Pos	N°	Pos	N°	Pos	N°	Pos
1	Fp1	17	C1	33	Cz	49	C6
2	AF7	18	TP9	34	Pz	50	T8
3	AF3	19	TP7	35	Fp2	51	T10
4	F9	20	CP5	36	AF4	52	CP2
5	F7	21	CP3	37	AF8	53	CP4
6	F5	22	CP1	38	F4	54	CP6
7	F3	23	P9	39	F6	55	TP8
8	FT9	24	P7	40	F8	56	TP10
9	FT7	25	P5	41	F10	57	P2
10	FC5	26	P3	42	FC2	58	P4
11	FC3	27	P1	43	FC4	59	P6
12	FC1	28	PO7	44	FC6	60	P8
13	T9	29	PO3	45	FT8	61	P10
14	T7	30	O1	46	FT10	62	PO4
15	C5	31	Gnd	47	C2	63	PO8
16	C3	32	Fz	48	C4	64	O2

Tabla 7.1: Relación entre la numeración de canales de la interfaz de usuario y la posición en la cabeza para la configuración de 64 canales.

7.3. Instalación en el paciente

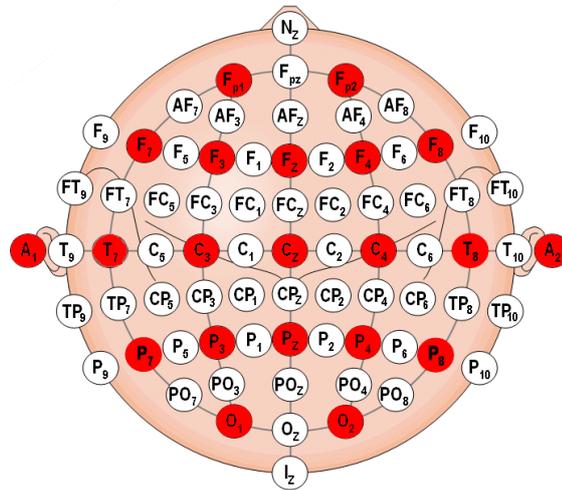


Figura 7.6: En color se muestran las posiciones que se utilizan con la configuración de 21 canales (imagen extraída de [45] y luego modificada).

N°	Pos	N°	Pos
1	Fp1	12	Pz
2	A1	13	Fp2
3	F7	14	F4
4	F3	15	F8
5	T7	16	C4
6	C3	17	T8
7	P7	18	P4
8	P3	19	P8
9	O1	20	A2
10	Fz	21	O2
11	Cz		

Tabla 7.2: Relación entre la numeración de canales de la interfaz de usuario y la posición en la cabeza para la configuración de 21 canales.

Capítulo 7. Montaje del módulo remoto

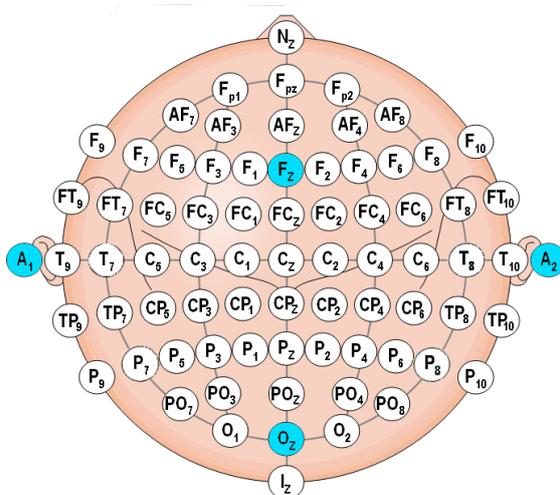


Figura 7.7: En color se muestran las posiciones que se utilizan con la configuración de 4 canales (imagen extraída de [46] y luego modificada).

N°	Pos	N°	Pos
1	A1	3	Oz
2	Fz	4	A2

Tabla 7.3: Relación entre la numeración de canales de la interfaz de usuario y la posición en la cabeza para la configuración de 4 canales.

En la figura 7.8 se muestra un esquema de conexión de los electrodos a la carcasa.

La conexión entre los electrodos y la carcasa se hace en dos tramos para tener flexibilidad en la ubicación del módulo remoto. Un primer tramo (de aproximadamente 40cm) se compone de cada uno de los cables que van desde un electrodo hacia un conector IDE que se encuentra en la nuca, y un segundo tramo que consta de un cable plano de aproximadamente 85cm que va desde el conector IDE a la carcasa. El uso de cables IDE fue avalado por el desarrollador de los integrados RHD2132, indicando que a la frecuencia de las señales con las que vamos a trabajar, las mismas no van a sufrir deterioro por interferencia o crosstalk ².

El armado de este cable no pudo completarse todavía. Como no se pudo encontrar a nivel local cables para electrodos EEG a un costo accesible, la implementación del primer tramo se pensó de dos formas, la primera fue utilizar cables

²El efecto de crosstalk refiere a la interferencia que presenta una señal por tener otra señal cercana. Es común en telecomunicaciones y su efecto es notorio en señales con alta frecuencia [47].

7.3. Instalación en el paciente

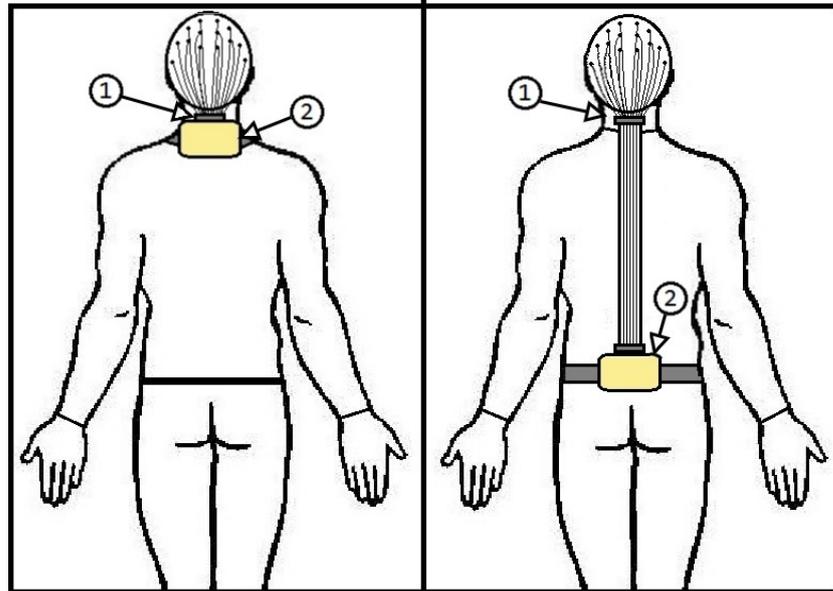


Figura 7.8: Conexión de los electrodos con la carcasa, a la izquierda se muestra la opción de conectar los electrodos directamente. A la derecha se muestra la opción de extender la conexión mediante un cable plano, de forma de dejar la carcasa en la cintura del paciente.

Referencias: 1-Conector IDE, 2-Carcasa.

diseñados para electrodos de ECG (figura 7.9a), la segunda fue utilizar broches de 10mm de diámetro (figura 7.9c), soldados y cubiertos con silicona (ver figura 7.9b) que se conecten al cable IDE. Si el costo lo permite, se opta por la primer opción.

Capítulo 7. Montaje del módulo remoto

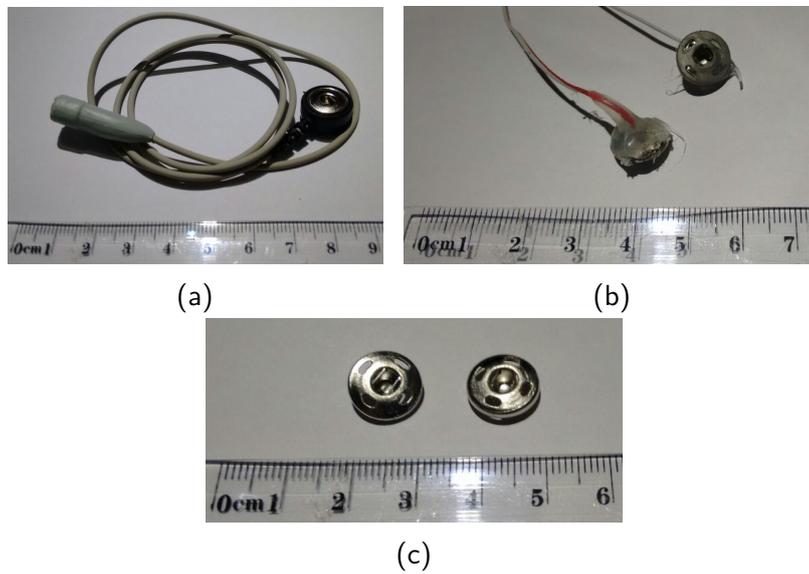


Figura 7.9: Distintos diseños de la unión de los electrodos con la carcasa. En (a) se muestra un cable comercial para electrodos de ECG, en (b) se muestra un cable fabricado artesanalmente, en (c) se muestran los broches utilizados en el diseño artesanal que sujetan los electrodos.

Capítulo 8

Pruebas y análisis de resultados

8.1. Pruebas de comunicación

8.1.1. Medición del throughput útil en función del tamaño de paquete

Esta prueba consiste en determinar el throughput útil en función del tamaño de paquete (*TAM_PAQUETE*) que se envía desde el módulo remoto al PC. Se varía el tamaño del paquete entre 128 y 2048 bytes, cada 128 bytes. El contenido del paquete que se envía es siempre el mismo.

Esta medición se realiza enviando un paquete de tamaño *TAM_PAQUETE* dentro de un loop desde el módulo remoto, por un período de 120 segundos. Desde el PC, se cuentan la cantidad de paquetes obtenidos durante el tiempo de la adquisición y al finalizar se calcula el throughput. El código del software que implementa esta prueba se presenta en el anexo I.1. Todas las medidas se realizaron en las siguientes condiciones de trabajo: la distancia entre el módulo remoto y el router es de 6 metros y sin obstáculos, se utiliza un canal de Wi-Fi libre, y la comunicación entre el router y el PC es Ethernet.

El throughput útil se calcula como:

$$\text{throughput}(bps) = \frac{8(\text{bits/byte}) \times \text{TAM_PAQUETE}(\text{bytes}) \times \text{paquetes_transmitidos}}{\text{duracion_adquisicion}(s)}$$

Resultados

La figura 8.1 muestra los resultados obtenidos. Se observa que el throughput útil máximo vale 5,48Mbps y se da para un tamaño de paquete de 1460 Bytes. Este valor coincide con el valor máximo de paquete permitido para el protocolo TCP/IPV4 estándar. El MTU¹ es de 1500 Bytes y como los encabezados de IP y

¹Maximum Transfer Unit - Es la cantidad de bytes máxima por paquete, sin que se tenga que fragmentar.

Capítulo 8. Pruebas y análisis de resultados

Tamaño de paquete (Bytes)	Paquetes transmitidos	Throughput (Mbps)
28	221984	1,89
256	214477	3,66
384	194800	4,99
512	152131	5,19
640	123571	5,27
768	104250	5,34
896	90077	5,38
1024	79207	5,41
1152	70735	5,43
1280	63915	5,45
1408	58278	5,47
1460	56268	5,48
1536	52026	5,33
1664	48249	5,35
1792	44962	5,37
1920	42140	5,39
2048	39631	5,41

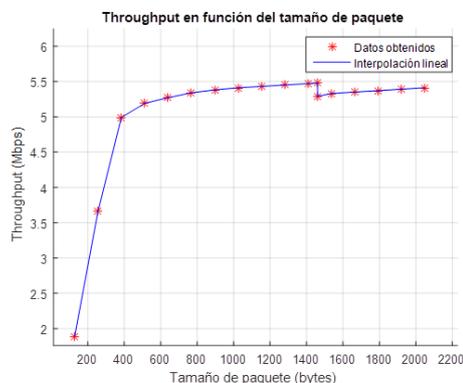


Figura 8.1: Throughput útil en la transmisión desde el módulo remoto al PC en función del tamaño de paquete.

TCP son de 40 Bytes en su forma más reducida quedan 1460 Bytes. Utilizando un tamaño de paquete mayor a 1460 Bytes, en la comunicación con el router se va a fragmentar en más de un paquete, disminuyendo la velocidad de transmisión como se observa en la tabla de la figura 8.1. Esto explica el salto que se da, para un tamaño de paquete de 1460 Bytes, en la figura 8.1.

Conclusiones

El throughput nominal de la radio CC3100 es de $13Mbps$ [48]. Buena parte de la diferencia entre el throughput útil y nominal se debe a que la frecuencia del reloj de SPI de la radio que alcanza los $13Mbps$ es de $20MHz$ y la máxima frecuencia de reloj SPI del MSP432 es de $12MHz$. Esta prueba determinó que el paquete óptimo sea de 1460 Bytes. De aquí en más se trabajará con este valor. Se recuerda que el requerimiento de throughput útil del sistema es de $1,12Mbps$, por lo que el valor obtenido para el tamaño de paquete de 1460 Bytes supera el requerimiento.

8.1.2. Medición del throughput en función de la distancia

Esta prueba consiste en determinar el throughput útil en función de la distancia entre el router y el módulo remoto. Las medidas se realizaron dentro de un salón amplio, variando la distancia desde $1m$ hasta $10m$, cada $1m$. Todas las medidas se realizaron en las siguientes condiciones de trabajo: tamaño de paquete igual a 1460 Bytes, sin obstáculos, con conexión entre router y PC Ethernet, canal de Wi-Fi libre y duración del estudio de 120 segundos.

8.1. Pruebas de comunicación

Resultados

A partir de la figura 8.2 se observa que dentro de un radio de 10mts, el throughput útil permanece inalterado. Realizando la misma prueba a una distancia de 12mts con una pared hueca de obstáculo, tampoco se observaron cambios en el throughput útil, lo que refleja firmemente su independencia con la variación de la distancia.

Distancia (metros)	Paquetes transmitidos	Throughput (Mbps)
1	56280	5,48
2	56290	5,48
3	56245	5,47
4	56379	5,49
5	56338	5,48
6	56304	5,48
7	56238	5,47
8	56308	5,48
9	56283	5,48
10	56311	5,48
12	56254	5,48

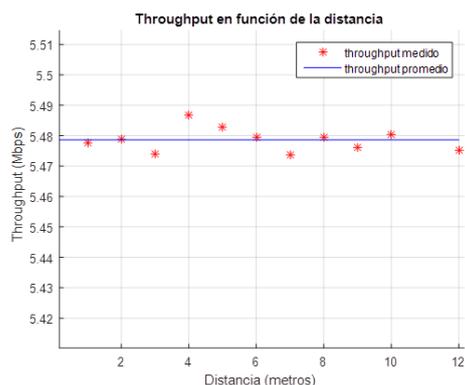


Figura 8.2: Throughput en la transmisión desde el módulo remoto al PC en función de la distancia entre el módulo remoto y el router.

Conclusiones

Los requerimientos del sistema establecen un throughput útil mayor a 1, 12Mbps con una distancia menor a 6 metros. A partir de la prueba realizada se puede concluir que el throughput útil en transmisión es independiente de la distancia (para distancias menores a 12 metros). Por tanto, cumple con el requerimiento. Si bien no se midió el throughput para distancias mayores a 12 metros, el sistema muestra potencial de tener un alcance mayor.

8.1.3. Medición del throughput útil variando el tipo de comunicación entre Router y PC

La comunicación entre el Router y el PC puede ser cableada (Ethernet) o inalámbrica (Wi-Fi). A su vez, se tienen varios canales de Wi-Fi para transferencia de datos. En algunos canales existen otras comunicaciones que interfieren (canal que llamaremos compartido) y en otros no (canal que llamaremos libre). Para mayor información ver la sección 5.3).

Esta prueba consiste en estudiar las variaciones en el throughput para los siguientes casos:

- Comunicación Router-PC: Ethernet. Canal Wi-Fi: Libre.
- Comunicación Router-PC: Wi-Fi. Canal Wi-Fi: Libre.

Capítulo 8. Pruebas y análisis de resultados

- Comunicación Router-PC: Ethernet. Canal Wi-Fi: Compartido.
- Comunicación Router-PC: Wi-Fi. Canal Wi-Fi: Compartido.

Condiciones

- Tamaño de paquete: 1460 Bytes.
- Distancia del módulo remoto al router: 6 metros.
- Sin obstáculos.
- Duración de la adquisición: 120 segundos.
- Canal libre: 1 y Canal compartido: 11.

Para medir la potencia de otras redes en el canal donde se hacen las medidas se utilizó el programa *Acrylic Wi-Fi Home*.

Resultados

La figura 8.3 muestra las potencias de las redes de Wi-Fi disponibles. Se observa que la red “wEEG” es la de mayor potencia seguidas de “iie” y “wifing” que se encuentran en el canal 11. En el canal 1 la siguiente red en potencia es “teldejuan.b” con una potencia de $-80dBm$.

La tabla 8.1 muestra los resultados obtenidos cuando se midió el throughput útil en las 4 casos mencionados. Se observa que en el caso de usar un canal libre, el throughput útil es prácticamente igual en los 2 casos, siendo un poco mejor si se usa Ethernet en vez de Wi-Fi como comunicación entre el router y el PC.

En el caso de un canal compartido, el throughput útil varía mucho con el uso del canal por otros dispositivos y el tipo de comunicación que se use entre el router y el PC. En momentos de mucho tráfico en el canal, el throughput útil se ve disminuido, como consecuencia, no se llega al throughput útil requerido de $1,12Mbps$ con estabilidad. En la figura 8.3, el canal 11 es un buen ejemplo de canal compartido. Si se trabaja en el canal 1 este fenómeno se ve disminuido debido a que la siguiente red que le sigue en potencia es de una potencia baja, en este caso el canal 1 puede considerarse como libre.

Comunicación Router-PC	Estado del canal	Paquetes transmitidos	Throughput (Mbps)
Ethernet	Libre	56288	5,48
Wi-Fi	Libre	55855	5,44
Ethernet	Compartido	14845	1,44
Wi-Fi	Compartido	2784	0,27

Tabla 8.1: Throughput en la transmisión desde el módulo remoto al PC en función del tipo de comunicación router-PC.

8.1. Pruebas de comunicación

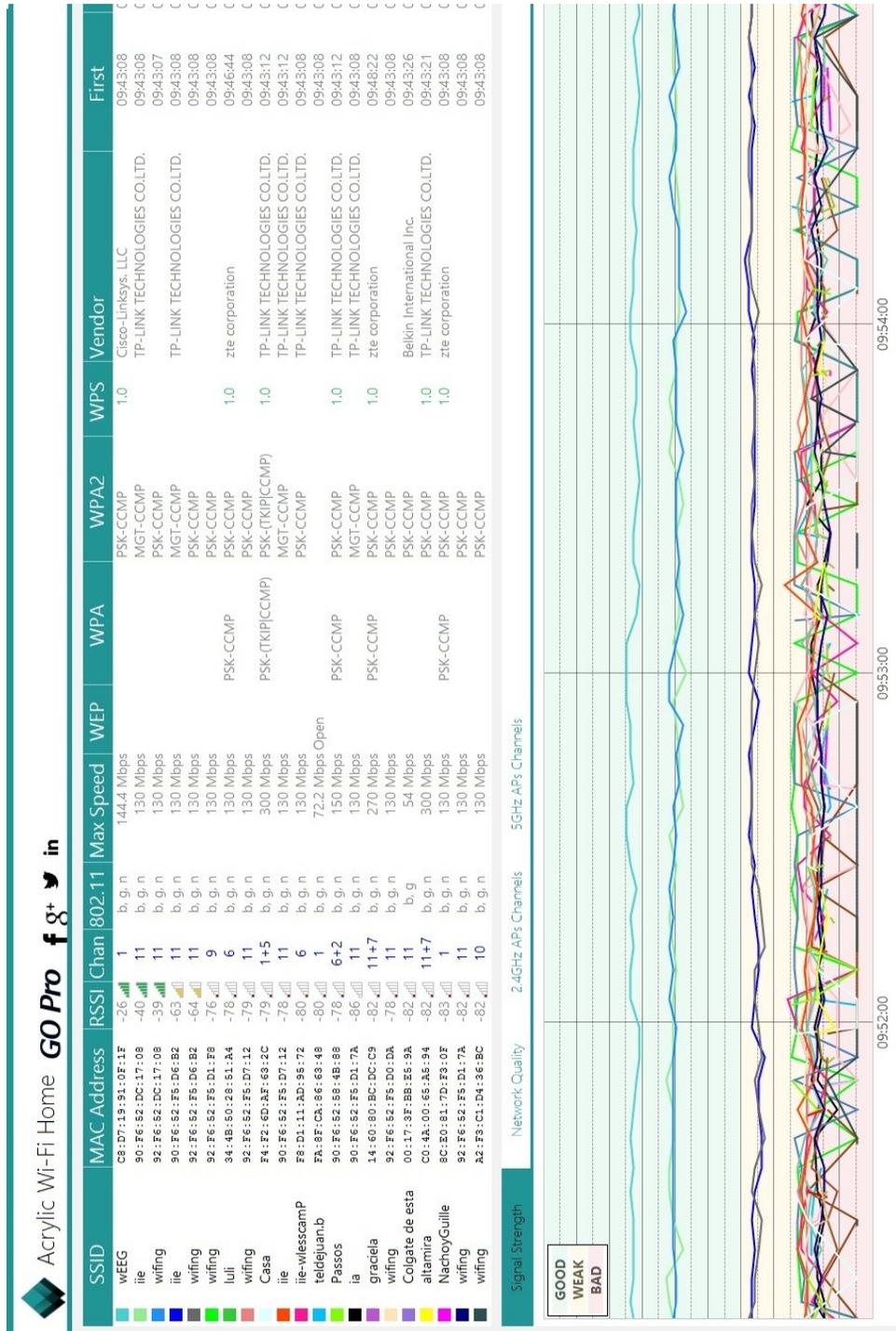


Figura 8.3: Redes de Wi-Fi disponibles. En la tabla se indica el nombre de la red, la potencia en dBm (columna RSSI), canal de Wi-Fi en la que se encuentra (columna Chan) y máxima velocidad de la red (columna Max Speed). Por debajo se muestra la variación de la potencia de cada señal en función del tiempo.

Conclusiones

A partir de la prueba realizada se puede concluir que se tiene que usar un canal libre y que recomendablemente la comunicación entre el router y el PC tiene que ser Ethernet. De aquí en más se trabaja en estas condiciones.

8.2. Medición de tiempos del software embebido

8.2.1. Tiempo de envío de palabra de 16 bits por SPI

La figura 4.5 muestra como van a enviarse y recibirse palabras entre el AFE y el microcontrolador. Se midieron las señales del reloj de SPI (CLK) y el chip select (\overline{CS}) del chip 0 para obtener:

- (1) Tiempo activo de \overline{CS} (t_{CS}).
- (2) Tiempo previo al envío o la recepción de la parte alta ($t_{inicio_{SPI}}$).
- (3) Tiempo de envío o recepción de cada palabra de 8 bits ($t_{palabra_{SPI}}$).
- (4) Tiempo entre envío o recepción de parte alta y parte baja ($t_{intermedio_{SPI}}$).
- (5) Tiempo posterior al fin del envío o la recepción de la parte baja ($t_{final_{SPI}}$).

La figura 8.4 muestra las señales \overline{CS} (arriba) y CLK (abajo) medidas con el osciloscopio.



Figura 8.4: \overline{CS} (arriba) y CLK (abajo) medidos. Las números indican: 1-Tiempo activo de \overline{CS} (t_{CS}). 2-Tiempo previo al envío o la recepción de la parte alta ($t_{inicio_{SPI}}$). 3-Tiempo de envío o recepción de cada palabra de 8 bits ($t_{palabra_{SPI}}$). 4-Tiempo entre envío o recepción de parte alta y parte baja ($t_{intermedio_{SPI}}$). 5-Tiempo posterior al fin del envío o la recepción de la parte baja ($t_{final_{SPI}}$).

Los tiempos registrados se muestran en la tabla 8.2.

El tiempo de envío y recepción de una palabra ($t_{palabra_{SPI}}$) es coherente con la frecuencia de la SPI indicada en la sección 4.4. Teóricamente $t_{palabra_{SPI}} = \frac{8}{12MHz} = 667ns$.

8.2. Medición de tiempos del software embebido

Medida	Tiempo (μs)
Tiempo activo de \overline{CS} (1)	4,46
Tiempo previo a la recepción de la parte alta (2)	0,22
Tiempo de envío o recepción de una palabra de 8 bits (3)	0,68
Tiempo entre recepción de parte alta y parte baja (4)	2,68
Tiempo posterior al fin de la recepción de la parte baja (5)	0,20

Tabla 8.2: Tiempos de las distintas tareas que se realizan en la adquisición de la señal de un canal.

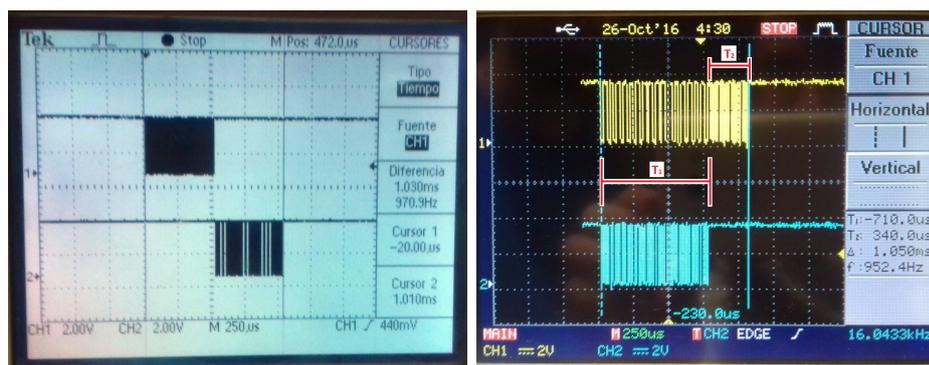
El tiempo $t_{intermedios_{SPI}}$ es grande y esto suponemos se debe a que el microcontrolador guarda y saca instrucciones de los buffers de entrada y salida del puerto SPI, actualiza los índices de los buffers, etc.

8.2.2. Tiempo de una adquisición

Para adquirir una tirada, el microcontrolador debe comunicarse con los dos integrados RHD2132. Se analizaron dos posibles configuraciones:

- Solicitar todos los canales al chip 0 y luego todos los canales al chip 1 (comunicación en serie).
- Solicitar paralelamente a los dos chips (configuración en paralelo).

Se estudió el tiempo que lleva cada caso para la configuración de 64 canales (sin señales de sincronismo). Las figuras 8.5a y 8.5b muestran las señales \overline{CS} de los dos puertos SPI.



(a) Configuración serie. La señal de arriba corresponde al \overline{CS} del chip 0 y la señal de abajo corresponde al \overline{CS} del chip 1.
 (b) Configuración en paralelo. La señal de arriba corresponde al \overline{CS} del chip 1 y la señal de abajo corresponde al \overline{CS} del chip 0.

Figura 8.5: Posibles configuraciones para la adquisición de señales.

Capítulo 8. Pruebas y análisis de resultados

Para la configuración de la figura 8.5a, se observa que el tiempo de la adquisición de los 64 canales más las cuatro instrucciones DUMMY ² es de 1,03ms y para la configuración de la figura 8.5b el tiempo de la adquisición es de 1,05ms. Se observó que los tiempos entre los dos tipos de configuración son casi iguales. Si bien a priori es de imaginar que la configuración en paralelo sea más rápida, ocurre que como se tiene un único procesador, no puede atender las dos interrupciones a la vez, y elige atender primero a la interrupción del chip 0 por tener mayor prioridad. Se puede observar en la figura 8.5b que la SPI del chip 1 se ve demorada por las interrupciones en el chip 0 (tiempo T_1), pero cuando el chip 0 termina, las interrupciones del chip 1 son igual de rápidas que las interrupciones en el caso de la configuración en serie (tiempo T_2).

Debido a que los tiempos son similares se decide utilizar la configuración serie, puesto que permite un manejo de software embebido más sencillo y ordenado.

Se midieron los tiempos asociados a una adquisición (tiempo desde que se muestrean los canales hasta que son guardados en un buffer):

- Tiempo total de adquisición de una tirada t_{adq} (medido a través de la bandera *is_acquiring*).
- Tiempo anterior a la adquisición del primer canal. Es el tiempo que necesita el microcontrolador para inicializar los buffers y cargar las instrucciones a enviar a los RHD2132 (t_{CB}). Este tiempo depende principalmente del tiempo que demora en ejecutarse la función `CONVERTburst()`³.
- Tiempo que demora el AFE en convertir los canales (t_{conv}).
- Tiempo entre la adquisición del último canal y el copiado de la tirada a `buffer_packet` (t_{BP})⁴.
- Tiempo entre adquisiciones de canales consecutivas ($t_{entreCS}$).

Se midieron en los siguientes casos de configuraciones de canales:

- 64 canales con 6 señales de sincronismo.
- 64 canales sin sincronismo.
- 21 canales con 6 señales de sincronismo.
- 21 canales sin sincronismo.
- 4 canales con 6 señales de sincronismo.

²En la sección 4.4 se definió instrucción DUMMY a una palabra que solo se envía al RHD2132 con el propósito de generar ciclos de lectura de SPI, para obtener la respuesta a una instrucción enviada previo al DUMMY.

³Se recuerda que la función `CONVERTburst()` carga en el buffer de salida de SPI las palabras a enviar a los integrados RHD2132 y no incluye la adquisición.

⁴`buffer_packet` es el buffer en el MSP432 que guarda los paquetes de tiradas adquiridas. Este buffer se encarga de acumular las tiradas para luego enviárselas a la radio. Una descripción más detallada se encuentra en la sección 4.6.

8.2. Medición de tiempos del software embebido

- 4 canales sin sincronismo.
- 1 canal sin sincronismo.

La figura 8.6 muestra a *is_acquiring* (arriba) y a los \overline{CS} de los dos puertos SPI (abajo). La tabla 8.3 muestra el valor de los tiempos registrados para cada configuración de canales.

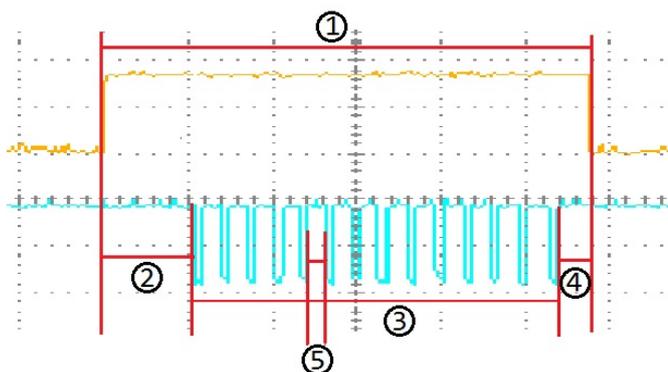


Figura 8.6: Adquisición de 4 canales con 6 señales de sincronismo. Arriba se muestra la bandera *is_acquiring* y abajo una señal auxiliar que indica por nivel bajo cuando el \overline{CS} de alguno de los puertos SPI está activo. Las referencias indican: 1- Tiempo de *is_acquiring* activo (t_{adq}). 2- Tiempo anterior a la adquisición del primer canal (t_{CB}). 3- Tiempo neto de la adquisición de una tirada (t_{conv}). 4- Tiempo entre la adquisición del último canal y el copiado de la tirada a *buffer_packet* (t_{BP}). 5- Tiempo entre adquisiciones de canales consecutivos ($t_{entreCS}$).

Configuración	f_s (Hz)	t_{adq} (1) (μs)	t_{CB} (2) (μs)	t_{conv} (3) (μs)	t_{BP} (4) (μs)
64 canales c/sinc	500	1270	64	1172	34
64 canales s/sinc	700	1096	40	1030	26
21 canales c/sinc	1500	564	52	488	24
21 canales s/sinc	2000	440	31	391	18
4 canales c/sinc	2000	288	54	214	20
4 canales s/sinc	2000	165	33	118	14
1 canal s/sinc	10000	77	23	54	0

Tabla 8.3: Tiempos de la adquisición de una tirada para distintas configuraciones de canales. Los casos de configuración con sincronismo implica muestrear los 6 canales de sincronismo. Los índices de tiempo hacen referencia a la figura 8.6. El tiempo (5), $t_{entreCS} = 11\mu s$ en todos los casos.

Para todos los casos se realizó el estudio de tiempos a una frecuencia de muestreo f_s para la cual no se generaban pérdidas, lejos de la frecuencia máxima del sistema. La segunda columna indica a qué frecuencia de muestreo se realizó la medida según la configuración de canales.

En todos los casos se observó que el tiempo entre adquisiciones de canales consecutivos ($t_{entreCS}$) era siempre igual y correspondía a $11\mu s$. El tiempo t_{conv} es coherente con lo esperado. El análisis se encuentra en el anexo I.2.

Capítulo 8. Pruebas y análisis de resultados

En cuanto al tiempo t_{CB} se puede observar que es menor para el caso de 21 canales que para el caso de 4 canales. Esto se debe a que para el caso de 21 canales, el microcontrolador posee en memoria un buffer con las instrucciones al RHD2132 que debe enviar. Este buffer se crea y carga en `AFE_init()` y luego queda disponible para su uso. En el caso de 4 canales, las instrucciones que deben mandarse a los integrados RHD2132 se crean cada vez que se inicia una nueva tirada desde la función `CONVERT_burst()`. Tener las palabras de 21 canales en memoria hace más rápido al configuración de 21 canales que a la configuración de 4 canales, en lo que refiere al tiempo t_{CB} .

Finalmente, en el tiempo t_{BP} se observa que el caso de 4 canales con sincronismo (que implica pasar a `buffer_packet` 10 conversiones) es más lento que el caso de 21 canales sin sincronismo (que implica pasar a `buffer_packet` 21 conversiones). Esto se debe a que los canales de sincronismo se guardan de una manera más lenta en `buffer_packet`, porque tienen que enviarse a la interfaz de usuario de manera ordenada. En la sección 4.9 se detalla el orden en el que se guarda la tirada en el paquete.

8.2.3. Tiempo de envío de datos del microcontrolador a la radio

Se estudió el tiempo de envío de un paquete desde el microcontrolador a la radio para distintos valores de la frecuencia de muestreo f_s (período de muestreo T_s). Si bien la prueba se realizó con 64 canales y sin señales de sincronismo, esto es indiferente puesto que el tamaño de los paquetes es similar para todos los casos de configuración, y en esta prueba solo importa el tamaño de paquete que se envía.

Se define t_{radio} al tiempo de envío del paquete desde el microcontrolador a la radio. Si $t_{adq} + t_{radio} < T_s$, se logra enviar el paquete de `buffer_packet` antes de que ocurra otra interrupción del timer A0. Si esto no ocurre, entonces el microcontrolador no termina de enviar el paquete a la radio cuando debe muestrear una nueva tirada. Esto genera que el envío del paquete se vea interrumpido para ocuparse de muestrear todos los canales, reanudando el envío al finalizar con la adquisición, lo que provoca que el tiempo t_{radio} termine siendo mayor. Si T_s es muy rápido, el envío de los paquetes a la radio puede verse atrasado por varias tiradas. El límite de la cantidad de tiradas que puede demorarse esta tarea debe ser menor que la cantidad de tiradas que entran en un paquete, ya que si esto no ocurre, se van a intentar agregar más datos de los que se sacan provocando pérdidas en el sistema. Se define $t_{radio_{min}}$ como el tiempo de envío del paquete a la radio sin que ocurran adquisiciones de tiradas en el medio.

La relación entre las variables definidas en el párrafo anterior viene dada por:

$$t_{radio} \approx t_{radio_{min}} + Ct_{adq} \quad (8.1)$$

donde C indica la cantidad de tiradas que ocurren mientras el envío no terminó (ver apéndice I.1). El tiempo $t_{radio_{min}}$ es un valor que depende únicamente

8.3. Prueba del sistema

del tamaño de paquete. Cuanto mayor es el tamaño de paquete, mayor $t_{radio_{min}}$, aunque su relación no es lineal.

La tabla 8.4 muestra el tiempo de envío (t_{radio}) para distintos valores de frecuencia indicándose cuantas tiradas ocurren durante el envío y si se registraron pérdidas. Con $f_s = 200Hz$, el envío por la radio no se ve interrumpido por el inicio de una nueva tirada. De esta manera, se obtiene que $t_{radio_{min}} = 2,14ms$. Este valor es fijo para la configuración con la que se realizaron estas medidas (porque solo depende del tamaño de paquete, que para las medidas está fijado en 1460 bytes, valor determinado como óptimo en la sección 8.1.1).

Frecuencia (f_s) en Hz	t_{radio} medido (ms)	Cantidad de tiradas adquiridas durante el envío (C_{medido})	¿Se registran pérdidas?
200	2,14	0	No
400	3,18	1	No
600	5,36	3	No
700	8,7	6	No
760	12	9	No
770	12,9	10	No
780	14	11	Sí

Tabla 8.4: Tiempo de envío de paquete a la radio CC3100 en función de la frecuencia de muestreo. Las medidas fueron realizadas con una configuración de 64 canales sin sincronismo. La última columna muestra el valor del tiempo de envío de paquete esperado.

Dado que 770Hz es la máxima frecuencia a la cual no se registran pérdidas, el valor del throughput real será 790 kbps ($770Hz \times 64ch \times 16bits/ch$)

8.3. Prueba del sistema

8.3.1. Frecuencia máxima en función de la cantidad de canales

Este test se realizó para encontrar la frecuencia máxima a la que se puede trabajar sin que haya pérdidas. Para ello se fue variando la frecuencia, realizando búsqueda binaria hasta encontrar el mayor valor de frecuencia donde no se tiene pérdidas. Para cada configuración de trabajo se realizó una adquisición de 10 minutos con la frecuencia máxima y luego otra adquisición de 10 minutos con una frecuencia 10 Hz mayor a la máxima y se calculó el porcentaje de pérdidas para este segundo caso. La frecuencia máxima fue relevada con una precisión de 10 Hz.

Las configuraciones a las que se mide la frecuencia máxima son:

- 64 canales con 6 señales de sincronismo.
- 64 canales sin sincronismo.

Capítulo 8. Pruebas y análisis de resultados

- 21 canales con 6 señales de sincronismo.
- 21 canales sin sincronismo.
- 4 canales con 6 señales de sincronismo.
- 4 canales sin sincronismo.
- 1 canales sin sincronismo.

Condiciones

- Distancia del módulo remoto al router: 6 metros.
- Conexión Router-PC: Ethernet.
- Canal Wi-Fi: Libre.
- Graficando todos los canales de la configuración elegida.
- Ancho de ventana de interfaz: 0,5 seg.
- Duración de la adquisición: 10 min.

Resultados

Para realizar pruebas en paralelo, las medidas de tiempos realizadas se hicieron con un microcontrolador (microcontrolador de tiempos o MCU_T) y las medidas de frecuencia presentadas en la tabla 8.6 se hicieron con otro (microcontrolador de frecuencia o MCU_F). Luego de realizadas las medidas de la tabla 8.6, se descubrió que las frecuencias máximas que alcanzan los distintos microcontroladores eran distintas, la razón de esta diferencia viene dada porque el reloj MCLK (Master Clock) del MSP432 varía hasta en un 10% entre los distintos microcontroladores según indica su hoja de datos. Para no descartar las medidas se realizó un estudio para comparar la diferencia entre los dos relojes MCLK. Para ello se midió el timer A0 configurado de la misma manera en ambos microcontroladores y se analizó la diferencia. La tabla 8.5 muestra los resultados obtenidos.

Frecuencia esperada	Frecuencia MCU_T	Frecuencia MCU_F
1000 Hz	957 Hz	1000 Hz

Tabla 8.5: Comparación entre la frecuencia de dos de los microcontroladores disponibles en el proyecto (MCU_T hace referencia al microcontrolador utilizado para medir los tiempos y MCU_F el utilizado para medir la frecuencias máxima). Se observa que uno de ellos tiene una variación de 4,3% que lo hace más lento e inexacto.

Como se observa, el MCU_T es 4,3% más lento que el MCU_F. Dado que la frecuencia del timer divide a la frecuencia del reloj MCLK (en los valores de la tabla $f_{timer} = \frac{f_{MCLK}}{48000}$ para ambos chips), se obtiene que la frecuencia del MCLK del MCU_T es de 45,94MHz. Esta variación en la base de tiempos impacta de la

8.3. Prueba del sistema

misma manera sobre la frecuencia máxima del sistema.

En la tabla 8.6 se muestran los valores máximos de frecuencia de muestreo registrados, el porcentaje de pérdidas a una frecuencia de 10Hz mayor.

Configuración		Frecuencia medida (en Hz)	Tiempo perdido (%)
64 CH	s/sinc	810	2,12
		800	0
	c/sinc (6)	740	1,13
		730	0
21 CH	s/sinc	2170	1,06
		2160	0
	c/sinc (6)	1700	1,32
		1690	0
4 CH	s/sinc	5980	5,2
		5970	0
	c/sinc (6)	3390	3,8
		3380	0
1 CH	s/sinc	10300	0

Tabla 8.6: Frecuencia máxima alcanzable para cada configuración. Para frecuencias mayores comienzan a registrarse pérdidas periódicas. *Frecuencia medida* corresponde a las medidas realizadas con el microcontrolador de frecuencia.

Análisis de la frecuencia máxima

El tipo de pérdidas que ocurren en la frecuencia máxima son debidas al desbordamiento del buffer de transmisión de paquetes del microcontrolador a la radio. Este tipo de pérdidas ocurren principalmente si el tiempo en que el microcontrolador demora en enviar un paquete a la radio es menor al tiempo en que el microcontrolador adquiere la cantidad de muestras que entran en un paquete ($t_{vaciado\ de\ buffer} < t_{llenado\ de\ buffer}$). Por lo tanto, para que no ocurran este tipo de pérdidas se debe cumplir la relación:

$$N_{tiradas}T_s \leq t_{radio} \quad (8.2)$$

La frecuencia máxima se obtiene entonces si $C = N_{tiradas}$ y si se da la igualdad en 8.2. Imponiendo la ecuación 8.1 y despejando f_s de la ecuación 8.2 se obtiene:

$$f_{s_{max}} = \frac{1}{T_{s_{max}}} = \frac{N_{tiradas}}{t_{radio_{min}} + N_{tiradas}t_{adq}} \quad (8.3)$$

con $t_{radio_{min}} = 2,14ms$ (medido en la sección 8.2.3).

La tabla 8.7 compara la frecuencia máxima obtenida con el MCU.T y la esperada a partir de la ecuación 8.3. Se observa que el análisis teórico es coherente con

Capítulo 8. Pruebas y análisis de resultados

Configuración	$N_{tiradas}$	t_{adq} (ms)	$f_{s_{max}}$ medida (Hz)	$f_{s_{max}}$ esperada (Hz)	Desviación relativa (%) ⁵
64 canales c/sinc (6)	10	1270	730	674	7,7
64 canales s/sinc	11	1150	800	775	3,1
21 canales c/sinc (6)	27	564	1690	1555	8,0
21 canales s/sinc	34	440	2160	1988	8,0
4 canales c/sinc (6)	72	288	3380	3147	6,9
4 canales s/sinc	182	165	5970	5657	5,2
1 canal	729	77	10300	12510	-21,5

Tabla 8.7: Comparación entre la frecuencia máxima medida y la esperada. El cálculo de $N_{tiradas}$ fue explicado en la sección 4.6, $N_{tiradas} = \frac{729}{N_c + N_s}$

lo obtenido, salvo para el caso de 1 canal que no se logró determinar la explicación.

Como se observa en la tabla 8.7, para las configuraciones de 64, 21 y 4 canales, la frecuencia medida muestra una desviación relativa del 6,5 % en promedio. Esto es coherente con el 4,3 % de variación en los MCLK de los dos microcontroladores usados para las medidas. En caso de que se utilice otro MSP432 para el sistema, se recomienda trabajar a una frecuencia por debajo del 90 % de la frecuencia máxima medida. En el caso de la configuración de un canal, se observa que la limitante en la frecuencia máxima no es causada por el llenado del buffer de transmisión de paquetes del microprocesador a la radio, a frecuencias mayores, el sistema presenta inestabilidad, perdiendo la comunicación al cabo de unos segundos de comenzado el estudio.

La principal limitante en la frecuencia máxima alcanzable viene dado por el tiempo de adquisición de tiradas. La tabla 8.8 muestra el porcentaje del tiempo en que el microcontrolador se encuentra adquiriendo. Para casi todas las configuraciones, se usa más del 90 % del tiempo en adquirir muestras, dejando menos de un 10 % para realizar otras tareas, principalmente el envío de paquetes a la radio. Por lo tanto, reduciendo el tiempo de adquisición t_{adq} se lograría reducir el porcentaje de tiempo en adquisición para aprovecharlo en otras tareas.

Conclusiones

La frecuencia máxima para cada configuración se ve limitada principalmente por el tiempo de adquisición, en la tabla 8.8 se observa que la mayor parte de la adquisición se utiliza en conversión de canales, por lo que para aumentar la frecuencia máxima se debe disminuir el tiempo de conversión de canales. Este

⁵Desviación relativa = $100\% \times \frac{f_{s_{max}} medida - f_{s_{max}} esperada}{f_{s_{max}} medida}$.

⁶Porcentaje del tiempo de adquisición = $100\% \times \frac{t_{adq}}{T_s}$.

8.3. Prueba del sistema

Configuración	t_{adq} (ms)	$f_{s_{max}}$ (Hz)	$T_s = \frac{1}{f_{s_{max}}}$ (ms)	% del tiempo en adquisición ⁶
64 canales c/sinc (6)	1,270	730	1,370	92,7
64 canales s/sinc	1,096	800	1,250	87,7
21 canales c/sinc (6)	0,564	1690	0,592	95,3
21 canales s/sinc	0,440	2160	0,463	95,0
4 canales c/sinc (6)	0,288	3380	0,296	97,3
4 canales s/sinc	0,165	5970	0,168	98,5
1 canal	0,077	10300	0,097	79,3

Tabla 8.8: Limitaciones de la frecuencia de muestreo alcanzable por el tiempo de adquisición.

problema termina limitando el throughput real. En la sección 9.3 se menciona una solución para disminuir t_{adq} .

8.3.2. Medición del consumo

En esta medición se determinaron los consumos de los distintos componentes del módulo remoto durante un estudio y en inactividad (sin realizar tareas) con el fin de determinar los consumos asociados a las tareas principales. Las medidas que se realizaron fueron las siguientes:

- Consumo de:
 - Módulo remoto durante un estudio y en inactividad.
 - AFE en inactividad (sin adquirir).
 - Microcontrolador en inactividad (sin adquirir ni transmitir datos).
 - Radio en inactividad.
- Duración de la transmisión y recepción de la radio.

A partir de estas medidas se calcularon los consumos de la radio en transmisión y recepción. El detalle de como se realizaron las medidas se da en el anexo I.7. La tablas 8.9 y 8.10 muestran los valores de estas medidas. Se observa que el consumo tanto del microprocesador, como del AFE en reposo y en adquisición no cambian sustancialmente. Quién demanda el consumo en la adquisición es la radio, que se encarga de copiar los paquetes a sus buffers internos, transmitir y recibir información proveniente del PC.

El consumo del AFE es de $23,4mA$ cuando el estimado era menor a $2mA$ según lo mencionado en la sección 2.4.1. La principal causa a la diferencia entre el

Capítulo 8. Pruebas y análisis de resultados

Consumo del AFE en inactividad	23,4 mA
Consumo del microcontrolador en inactividad	8,2 mA
Consumo de la radio en inactividad	23,4 mA
Consumo del módulo remoto en inactividad	55 mA

Tabla 8.9: Consumo en inactividad de los bloques del módulo remoto.

Consumo promedio en transmisión de la radio (extra al consumo del módulo remoto en inactividad)	240 mA
Duración de la transmisión de la radio	0,2 ms
Períodos de ocurrencia de los picos de transmisión @770 Hz, 64CH s/sinc	15,4 ms
Períodos de ocurrencia de los picos de transmisión @500 Hz, 64CH s/sinc	19,8 ms
Consumo promedio en recepción de la radio (extra al consumo del módulo remoto en inactividad)	60 mA
Duración de la recepción de la radio @200 Hz, 4CH s/sinc	920 ms
Consumo del módulo remoto en adquisición @770 Hz, 64CH s/sinc	115 mA

Tabla 8.10: Medidas del módulo remoto durante un estudio.

consumo medido y el consumo esperado teóricamente se asocia a que uno de los integrados RHD2132 está dañado y puede tener un consumo por fuera de lo normal.

En cuanto al consumo del microcontrolador el valor estimado ($8,5mA$) es coherente con el valor medido. Se puede mejorar el consumo apagando periféricos del microcontrolador o configurando sus pines que no se usan como pines de entrada, para lograr un consumo de $5,0mA$.

En cuanto a la radio, el consumo en transmisión estimado ($272mA$) es coherente con el consumo en transmisión medido ($240mA + 23,4mA = 263mA$).

Se pudo determinar que el pico de $240mA$ corresponde a cuando la radio envía paquetes al PC. Este pico de transmisión tiene una duración de $200\mu s$ y ocurre cada $15,4ms$, que es coherente con el tiempo con el que se envía un paquete desde el microcontrolador a la radio ($\frac{N_{tiradas}}{f_s} = \frac{11}{770}s = 14,3ms$). Si se considera el caso de 500 Hz de frecuencia de muestreo, entonces $\frac{N_{tiradas}}{f_s} = \frac{11}{500}s = 22ms$, que se asemeja al valor $19,8ms$ mostrado en la tabla 8.10.

La figura 8.7 muestra el consumo del módulo remoto en una adquisición de 4 canales sin sincronismo a 200 Hz. En la figura se observan picos de corriente de poca duración que corresponden a la transmisión de un paquete al PC. Los pulsos cuadrados de la imagen tienen un valor promedio de $115mA$ ($60mA$ más que el consumo del módulo remoto en reposo), una duración de $74ms$ y ocurren aproximadamente cada $920ms$, que es coherente con el tiempo en el que se genera un paquete en el microcontrolador ($\frac{N_{tiradas}}{f_s} = \frac{182}{200}s = 910ms$).

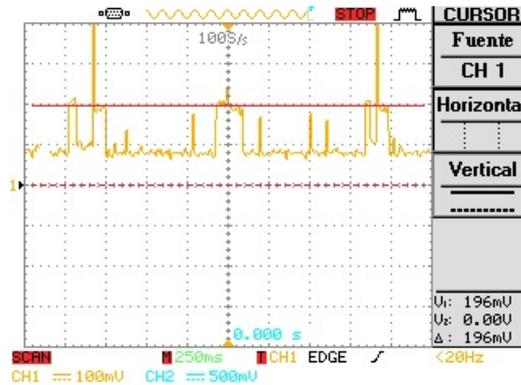


Figura 8.7: Consumo del módulo remoto para una adquisición de 4 canales sin sincronismo a una frecuencia de muestreo de 200 Hz.

Si la frecuencia f_s de muestreo aumenta, el período con el cual ocurren los pulsos cuadrados de $115mA$ disminuye, a tal punto en el cual, tendremos una frecuencia de trabajo donde el consumo es constante en $115mA$. Dado que la duración de los pulsos es de $74ms$ cualquiera sea la configuración (porque el tamaño de paquete que se envía es siempre de 1460 bytes), la frecuencia f_s a la cual ocurre este fenómeno es: $\frac{N_{tiradas}}{f_s} = 74ms$, es decir, $f_s = \frac{N_{tiradas}}{74ms}$. La tabla 8.11 muestra las frecuencias a la que esto ocurre para la distintas configuraciones de canales. $N_{tiradas}$ depende de la configuración de canales utilizada (ver tabla 8.7).

Configuración	Frecuencia de consumo de recepción constante (Hz)
64 canales sin sincronismo	149
64 Canales + 6 de sincronismo	135
21 canales sin sincronismo	459
21 canales + 6 de sincronismo	365
4 canales sin sincronismo	2459
4 CH + 6 canales de sincronismo	973
1 canal	9851

Tabla 8.11: Frecuencia a la cual el consumo debido al copiado de paquetes al buffer de transmisión del CC3100 comienza a ser constante.

El consumo extra de $60mA$ se puede comparar con el consumo que tiene la radio al recibir que es de $53mA$ (ver tabla 2.4). Los picos de corriente más pequeños en la figura 8.7 también están asociados al consumo en recepción de la radio.

8.3.3. Autonomía del sistema

Para el estimativo de la autonomía del sistema, fue necesario conocer el voltaje mínimo de alimentación para el cual el módulo remoto funciona. Este valor se midió utilizando un generador como voltaje de entrada y bajando su valor hasta que

Capítulo 8. Pruebas y análisis de resultados

el sistema dejó de adquirir. Con esta prueba se obtuvo que el valor de voltaje mínimo es $3,2V$, ya que si la alimentación está por debajo de este valor, los integrados RHD2132 dejan de funcionar. El valor obtenido empíricamente, es coherente con la tabla 2.6, en la cual se muestran los rangos de voltaje según las hojas de datos de cada integrado del módulo remoto.

La figura 8.8 muestra el voltaje que entrega la pila elegida para alimentar el módulo remoto según la cantidad de carga que se descargó hasta ese momento.

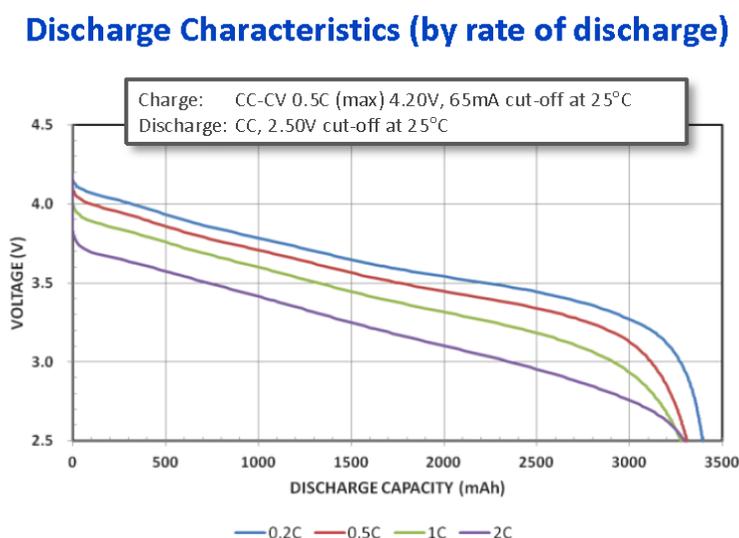


Figura 8.8: Gráfico del voltaje entre bornes de la pila en función de su descarga (imagen extraída de [49]).

Como se observa al comienzo, la pila entrega $4,2V$ pero luego el voltaje que entrega baja. Si bien la carga de la pila es $3400mAh$ como fue indicado en la tabla 2.7, podremos utilizar la pila hasta que el voltaje llegue al valor $3,2V$.

La figura 8.8 presenta cuatro curvas según la velocidad de descarga de la pila. $C = 3400mAh$ es la carga de la batería y si I es la corriente media que se consume entonces $\alpha = \frac{I}{C}$ y por tanto debe mirarse la curva αC o la que mejor se aproxima al caso del sistema [50].

Para la autonomía se realizó el estudio bajo las siguientes condiciones:

- Distancia del módulo remoto al router: 0,5 metros.
- Conexión Router-PC: Ethernet.
- Canal libre.
- Se grafican 2 canales en MatLab.
- Ancho de ventana de interfaz: 2 seg.

8.3. Prueba del sistema

- Configuración de canales: 64 canales sin sincronismo.
- Frecuencia de muestreo: 770 Hz (frecuencia máxima para esa configuración).

Para estas condiciones, el consumo de corriente medio es $115mA$ según lo indicado en la tabla 8.10, por lo que $\alpha = 0,03$ y la curva a considerar es “0,2C”. La carga disponible será entonces $3200mAh$ puesto que luego de esto, el voltaje que entrega la pila es inferior a $3,2V$. Se debe considerar además que el conversor tiene un rendimiento de $\eta = \frac{3,3V}{3,7V} = 89\%$, por lo que la corriente que recibe el módulo remoto es $\eta 3200mAh = 2850mAh$.

La prueba mostró una autonomía del sistema de **24 horas, 21 minutos y 32 segundos**.

Para verificar este resultado, se calcula el consumo del módulo remoto con los datos obtenidos en la sección 8.3.2. El tiempo teórico de duración de la batería es:

$$T_{total} = 2850mAh \frac{N_{tiradas}T_s + T_{transmision}}{T_{transmision}(I_{base} + I_{transmision}) + I_{base}N_{tiradas}T_s} \quad (8.4)$$

En el anexo I.8 se hace la deducción de la ecuación 8.4. Para esta prueba: $T_{transmision} = 200\mu s$, $N_{tiradas} = 11$, $T_s = 1,3ms$, $I_{transmision} = 180mA$ e $I_{base} = 115mA$, lo que lleva a que:

$$T_{total} = 24hs, 50min$$

Se observa que el modelo estimado de autonomía es coherente con lo obtenido experimentalmente, puesto que la diferencia es solo de 30 minutos, que corresponde a menos de un 2% del valor medido.

8.3.4. Prueba de punta a punta

En esta sección se prueba la capacidad de adquisición del sistema. Se inyectaron varias señales conocidas a distintas configuraciones. En la tabla 8.12 se listan las señales de entrada junto a las condiciones de cada medición.

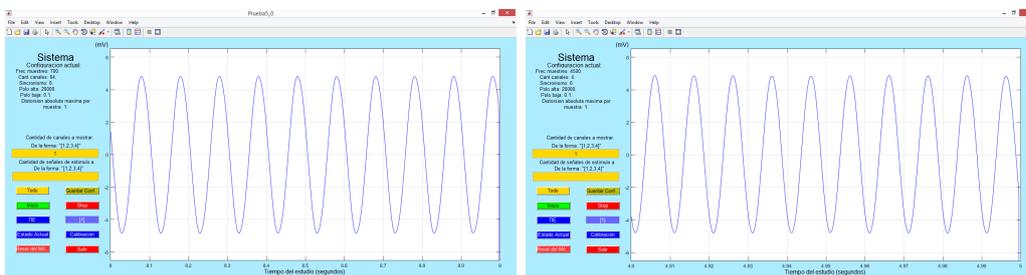
Todas las señales medidas tienen una amplitud de $10mV_{pp}$ y las frecuencias de corte superior e inferior fueron configuradas a $20kHz$ y $0,1Hz$ respectivamente. Los resultados obtenidos a partir de las mediciones se muestran en las imágenes de la figura 8.9.

En las figuras 8.9a y 8.9b se mide la amplitud de cada señal y se observa que la amplitudes de las sinusoides de $10Hz$ y $100Hz$ valen $9,64mV_{pp}$ y de $9,63mV_{pp}$ respectivamente. En la figura 8.9c se observa el efecto del muestreo de una sinusoide de frecuencia del orden de la frecuencia de muestreo ($f_{sinusoide} = 1kHz$ y $f_s = 9kHz$) debido a la unión de los puntos en MatLab, lo correcto sería realizar una reconstrucción a partir de lo muestreado. En la figura 8.9d se muestra una sinusoide

Capítulo 8. Pruebas y análisis de resultados

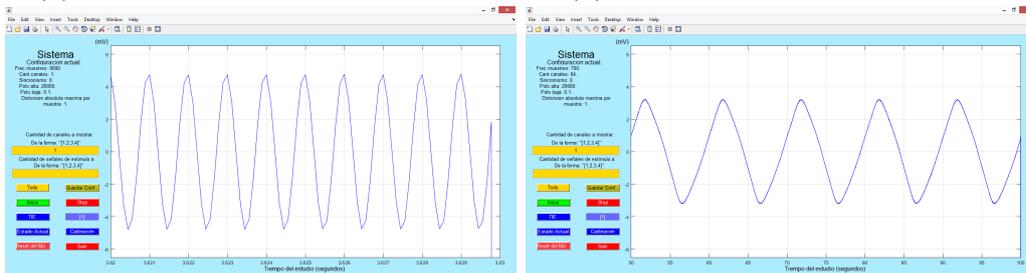
	Señal de entrada	Frecuencia de la señal (Hz)	Configuración	Frecuencia de muestreo (Hz)
a	Sinusoide	10	64 CH/ sin sinc	700
b	Sinusoide	100	4CH/ sin sinc	4500
c	Sinusoide	1000	1 CH	9000
d	Sinusoide	0,1	64CH/ sin sinc	700
e	Diente de sierra	50	21CH/ sin sinc	1500
f	Diente de sierra	4	21CH/ sin sinc	1500

Tabla 8.12: Listado de las señales inyectadas junto a la configuración utilizada. Todas las señales de entrada tiene una amplitud de $10mV_{pp}$.



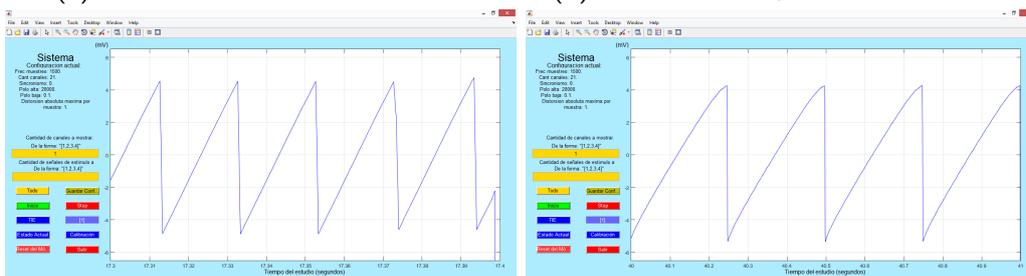
(a) Sinusoide de 10Hz de frecuencia.

(b) Sinusoide de 100Hz frecuencia.



(c) Sinusoide de 1kHz frecuencia.

(d) Sinusoide de 0,1Hz de frecuencia.



(e) Diente de sierra 50Hz de frecuencia.

(f) Diente de sierra 4Hz de frecuencia.

Figura 8.9: Señales medidas para la prueba de punta al punta del sistema, las señales tienen una amplitud de $10mV_{pp}$ y las frecuencias de corte superior e inferior fueron configuradas a $20kHz$ y $0,1Hz$ respectivamente.

de amplitud de $6,40mV_{pp}$; vemos que $\frac{V_{out}}{V_{in}} = \frac{6,40}{9,63} = -3,55dB$, lo cual es coherente con el hecho de estar a la frecuencia del polo simple que posee el AFE para filtrar

8.4. Resumen de las pruebas

señales de baja frecuencia.

Finalmente en las figuras 8.9e y 8.9f se muestran dos dientes de sierra, en el diente de sierra de $4Hz$ se observa un amortiguamiento en los extremos.

En conclusión, el sistema trabaja bien en el rango del ancho de banda programable, observándose una atenuación.

8.3.5. Testeo del funcionamiento de todos los canales

En esta sección se testeó el funcionamiento de todos los canales con una señal conocida (sinusoide de $50Hz$ de frecuencia y $10mV_{pp}$ de amplitud) en una configuración de 64 canales a $f_s = 700Hz$. Para esta prueba, no se incluyeron las señales de sincronismo.

En la figura 8.10 se observan las señales de los 64 canales (el canal 1 corresponde a la señal inferior y el canal 64 a la superior), en la ventana gráfica se observan 29 sinusoides lo cual implica que hay una falla en 35 canales. Las 32 señales que se observan en la parte superior de la figura corresponden al chip 1, esto indica que este chip no responde bien a las instrucciones enviadas desde el microcontrolador, esto puede deberse a un problema al soldar el chip o algún componente entre este y el conector con el microcontrolador. De las 32 señales inferiores, se observan fallas en 3 de ellas, los canales 24, 28 y 32 no muestran respuesta, esto se debe a un problema al resoldar el conector IDE.

Las configuraciones de 64, 21 y 4 canales se ven afectadas directamente debido a que todas utilizan el chip 1. De todos modos, los canales utilizados en las configuraciones de 21 y 4 canales pueden adaptarse para que funcionen en su totalidad, utilizando únicamente los canales del chip 0 que funcionan.

8.4. Resumen de las pruebas

A continuación se resumen las características finales del sistema en la tabla 8.13.

Capítulo 8. Pruebas y análisis de resultados

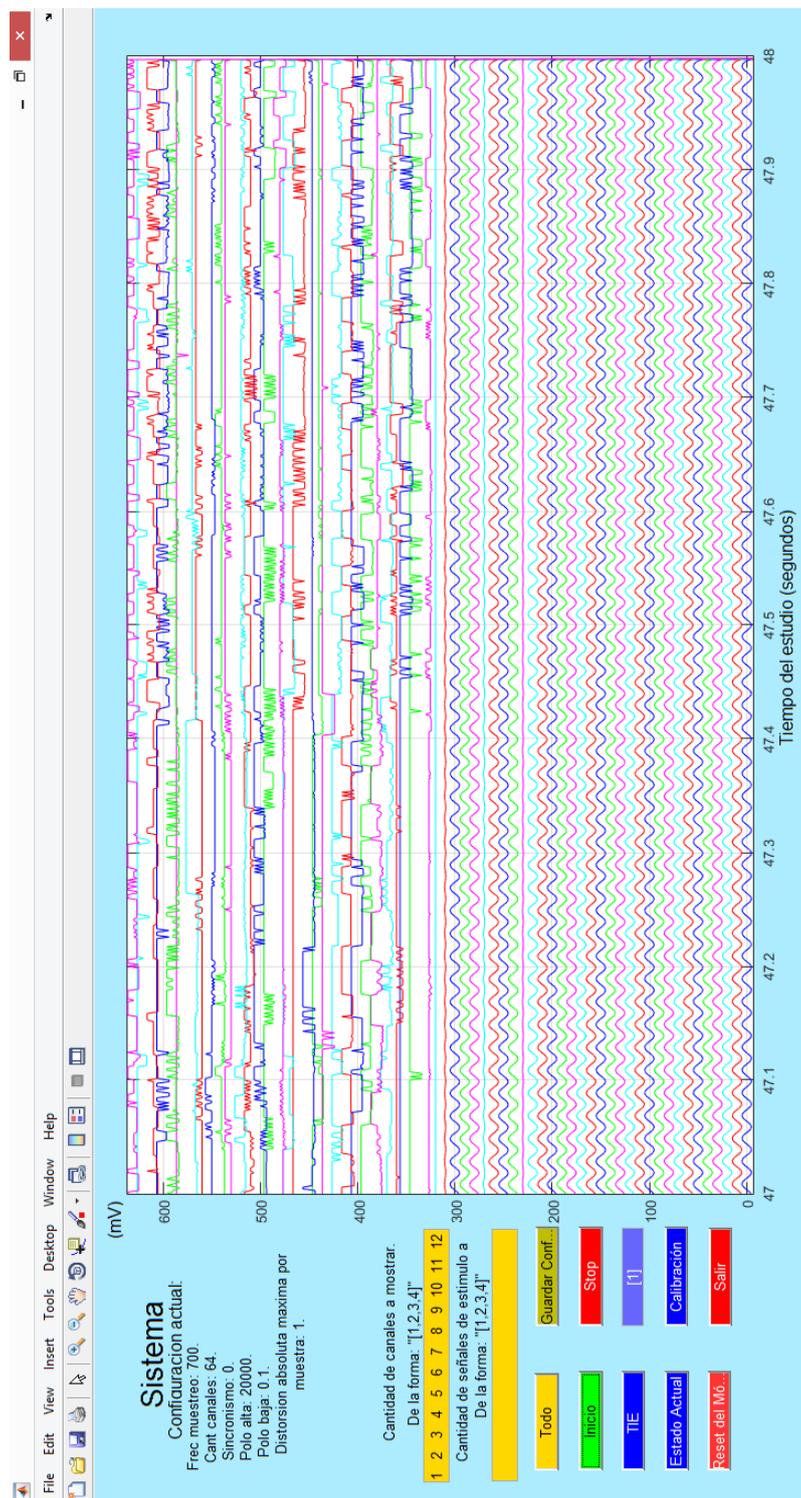


Figura 8.10: Estudio con todos los canales testeando su funcionamiento.

8.4. Resumen de las pruebas

Especificaciones	Dato del sistema	Comentarios
Número de canales	29	Sistema diseñado para soportar 64 canales pero por roturas solo funcionan 29 canales. Admite hasta 4 configuraciones (64,21,4 predefinidos y 1 canal a elección entre los 64 disponibles)
Número de señales de sincronismo	6	Se agregan a las anteriores mencionadas. Se puede elegir una cantidad entre 0 y 6.
Autonomía	24h 22m	-
Peso del módulo remoto	270 g	-
Tamaño físico del módulo remoto	11,70cm × 7,88cm × 4,76cm	Dimensiones en largo, ancho y altura
Tipos de electrodos	Secos o húmedos	-
Consumo módulo remoto	115 mA @64 ch, 770 Hz	-
	55 mA @reposo	-
Frecuencia de corte superior	Programable entre 100Hz y 20kHz	-
Frecuencia de corte inferior	Programable entre 0,1 Hz y 500Hz	-
Impedancia de entrada	1,3GΩ	-
Ruido intrínseco del amplificador	< 2,4μVrms	-
CMRR	82 dB	-
Frecuencia de muestreo	Programable entre	-
	92 Hz y 770 Hz	@64 canales sin sincronismo
	92 Hz y 690 Hz	@64 canales con hasta 6 señales de sincronismo
	92 Hz y 2060 Hz	@21 canales sin sincronismo
	92 Hz y 1610 Hz	@21 canales con hasta 6 señales de sincronismo
	92 Hz y 5710 Hz	@4 canales sin sincronismo
	92 Hz y 3230 Hz	@4 canales con hasta 6 señales de sincronismo
	92 Hz y 9850 Hz	@1 canal sin sincronismo
Throughput útil	5,48 Mbps	-
Throughput real	790 kbps	-
Distancia máxima entre módulo remoto y PC	12 metros	-
Costo de fabricación y componentes	US\$ 3353	El valor indicado incluye el costo de los electrodos secos comprados, que no fueron incluidos en el presupuesto de proyecto como se indicaba en la tabla 1.2. Sin el costo de los electrodos, el presupuesto del proyecto fue de US\$ 1753

Tabla 8.13: Características finales del sistema

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 9

Conclusiones

9.1. Conclusiones generales

Se obtuvo un sistema capaz de adquirir hasta 29 señales (fácilmente extendible a 64 señales y 6 señales de sincronismo) durante más de 24 horas, con una frecuencia de muestreo programable entre 92 Hz y 9850 Hz. El sistema admite 4 tipos de configuraciones: los 29 canales, 21 canales predefinidos, 4 canales predefinidos, o 1 canal a elección entre los 29 posibles. Posee un filtro pasabanda programable; la frecuencia de corte inferior puede variar entre 0,1 Hz y 500 Hz y la frecuencia de corte superior puede variar entre 100 Hz y 20 kHz. El bloque analógico del sistema tiene una impedancia de entrada de 1,3 GΩ, un ruido intrínseco menor que 2,4 μV y un CMRR de 82 dB. El módulo remoto (parte del sistema que se ubica en el paciente) se encuentra dentro de una carcasa de dimensiones 12 cm × 8 cm × 5 cm. La distancia máxima medida entre el módulo remoto y el PC es 12 metros (no se realizaron pruebas a mayor distancia). En la tabla 8.13 se resumen las características del sistema.

En este proyecto se utilizó una amplia diversidad de conceptos. Desde el punto de vista de la ingeniería eléctrica se trabajó con temas de diseño y testeado de un PCB, adquisición y tratamiento de señales a través de un AFE, diseño y programación de un software embebido para procesamiento de señales, transmisión de datos de forma cableada e inalámbrica y el diseño y uso de una interfaz de usuario en un PC. Se aprendieron técnicas de elección de componentes y análisis para elegir los dispositivos más adecuados para el sistema que se desea implementar. Se trabajó en otras áreas como realizar el montaje del sistema, medición y análisis de pruebas, normativa de seguridad para dispositivos médicos, vinculación con clientes, médicos, etc.

Los objetivos planteados inicialmente fueron alcanzados. De la tabla 9.1 a la tabla 9.5 se muestra el nivel de alcance de estos.

Capítulo 9. Conclusiones

Especificaciones	Aceptable	Ideal	wEEG	Alcance
Numero de canales	16	64	29	Aceptable
Autonomía	12hs	24hs	24hs y 21min	Ideal
Peso	300 g	100 g	270 g	Aceptable
Tamaño	20 × 15 × 9 cm	13 × 9 × 7 cm	11,70 × 7,88 × 4,76 cm	Ideal
Retardo de visualización	1s	500ms	> 500ms ¹	Aceptable
Tipo de electrodos	Húmedos	Secos	Húmedos	Aceptable
Costo ²	US\$ 1300	US\$1000	US\$1753	No logrado

Tabla 9.1: Requerimientos generales alcanzados.

Especificaciones	Aceptable	Ideal	wEEG	Alcance
Frecuencia de corte superior	de 100Hz a 200Hz	de 100Hz a 500Hz	de 100Hz a 20000Hz	Ideal
Frecuencia de corte inferior	de 0,1Hz a 10Hz	de 0,1Hz a 30Hz	de 0,1Hz a 500Hz	Ideal
Impedancia de entrada	500 MΩ	1GΩ	1,3GΩ	Ideal
Ruido intrínseco del amplificador	5μVrms	3μVrms	2,4 μVrms	Ideal
CMRR	70dB	100dB	82dB	Aceptable

Tabla 9.2: Requerimientos alcanzados del AFE.

Especificaciones	Aceptable	Ideal	wEEG	Alcance
Frecuencia de muestreo	de 200Hz a 400Hz	de 200Hz a 1000Hz	de 92 Hz a 770Hz @64canales de 92 Hz a 2060Hz @21canales	Aceptable
Bits libres de ruido	8bits	12bits	16bits	Ideal

Tabla 9.3: Requerimientos alcanzados para el convertor A/D.

Especificaciones	Aceptable	Ideal	wEEG	Alcance
Memoria FLASH	12kB	20kB	12kB	Aceptable
Memoria RAM	32kB	128kB	64kB	Aceptable
FPU	No	Sí	Sí	Ideal

Tabla 9.4: Requerimientos alcanzados para el microcontrolador.

Se logró cumplir con el criterio ideal de éxito en muchos aspectos y se alcanzó el criterio aceptable en casi todos, dejando un muy buen punto de partida para lograr un sistema que cumpla no solo con los objetivos ideales planteados, sino también para superarlos en algunos casos, encontrando caminos por los cuales continuar

¹El retardo de visualización depende de la configuración de canales, la frecuencia de muestreo y del PC. Si la frecuencia de muestreo es menor y la cantidad de canales es menor, el retardo de visualización será mayor. El típico es apenas superior a 500 ms, por eso se logró el aceptable.

²No incluye el costo de los electrodos secos

9.2. Comparación del sistema con otros dispositivos EEG comerciales

Especificaciones	Aceptable	Ideal	wEEG	Alcance
Throughput útil	128kbps	1,12Mbps	5,48Mbps	Ideal
Distancia	4mts	6mts	12mts	Ideal

Tabla 9.5: Requerimientos alcanzados para la radio.

explorando y dando una idea de como deben implementarse para mejorar.

En cuanto a los objetivos opcionales del proyecto, se logró incorporar al sistema señales de sincronismo (los dos integrados para el bloque AFE que se compraron son capaces de incorporar hasta 3 señales de sincronismo cada uno). Si bien se escribió el código para implementar señales de sincronismo a la vez que se muestrean canales de EEG, por alguna causa que no se pudo determinar, no se pudo realizar el muestreo de ambos tipos de canales a la vez. Queda como trabajo a futuro revisar el código y encontrar el error que no permite que las señales de sincronismo funcionen correctamente.

Queda también como trabajo a futuro, dar al sistema la capacidad de sustituir el AFE actual por un amplificador neuronal provisto por el tutor e integrar el algoritmo ACEEG para comprimir las señales, aunque para este último objetivo opcional, fue estudiado que si se utiliza DMA (Direct Memory Access) para delegar a un periférico la comunicación SPI del microcontrolador con el AFE, sí se puede integrar el algoritmo ACEEG (ver sección 8.3.1).

En lo que respecta al montaje del sistema, faltó fabricar el casco y los cables planos que comunican el casco con la carcasa. Principalmente, no se llegó a realizar pruebas en seres vivos.

9.2. Comparación del sistema con otros dispositivos EEG comerciales

En la tabla 9.6 se compara el sistema wEEG con otros dispositivos de EEG disponibles en el mercado. El sistema diseñado presenta muchas ventajas frente a otros electroencefalógrafos. Entre estas se observan las características del AFE y la autonomía alcanzada, que pese a utilizar el protocolo Wi-Fi presenta una mayor autonomía que los demás sistemas. El throughput es ampliamente mayor. De todos modos, aún presenta carencias en comparación a los otros sistemas, como puede ser en tamaño y tipo de electrodos. Así mismo, como ya se mencionó, se puede trabajar en la mejora de estos aspectos a futuro, logrando un producto final que pueda superar muchas de las especificaciones de los electroencefalógrafos en el mercado.

Capítulo 9. Conclusiones

	EPOC/EPOC+	Insight Secos	ENOBIO Secos	BioRadio	COGNIONICS Secos	wEEG
Características del AFE	Electrodos	-	-	-	-	Húmedos
	Impedancia de entrada	-	>1 GΩ	500 MΩ	-	1, 3GΩ
	Rango de entrada	-	-	1μV-2V	-	±5mV
	Ancho de banda	0,2 - 45 Hz (con notch en 50 Hz y 60 Hz)	-	0-125 Hz	0-131 Hz @500 sps 0-262 Hz @1000 sps	Programable entre 0,1 Hz y 20kHz
	Ruido a la entrada	-	<1 μV RMS	-	0,7 μV RMS 1-50 Hz	< 2, 4μV _{rms}
	CMRR	-	-	100 dB	-	82 dB
	Número de canales	14	5	4 diferenciales 8 single-ended	16,24,32 y 64	1,4,21,64
	Sincronismo	0	2	8,20,32	8	6
Características digitales	Tasa de muestreo	128 sps (2048 Hz interno, 16 × 128Hz)	128 sps por canal	Programable entre 250 sps a 16000 sps	500 sps @64 canales y 1000 sps @32 canales	770 sps @ 64 canales 2060 sps @21 canales
	Resolución	0,51μV - 14 bits	-	30,5μV - 16 bits	-	391mV _{pp} - 16 bits
Características generales	Autonomía	12 Hs	-	8 Hs	6 hs inalámbrico, 10 hs con SD Card	>24hs
	Dimensiones (LAH)	-	-	10 × 6 × 2cm ³	-	12 × 8 × 5cm ³
	Peso	-	-	65 g	350 g	270 g
Características de la comunicación	Protocolo	Propietario de banda en 2,4 Ghz + Bluetooth 4,0 LE	Propietario de banda en 2,4 Ghz + Bluetooth 4,0 LE	-	Bluetooth optimizado para alta velocidad	Wi-Fi
	Throughput	-	-	200 kbps	-	5,48 Mbps
	Rango de transmisión	-	-	100 m	-	>12 m

Tabla 9.6: Comparación del sistema wEEG con electroencefalógrafos inalámbricos comerciales.

9.3. Trabajos futuros

Probar el sistema con electrodos secos y húmedos: se llegaron a comprar electrodos húmedos en el mercado local y se compraron electrodos secos en el exterior a través de la empresa Cognionics. Queda como trabajo futuro probar señales conocidas usando electrodos secos y húmedos, y estudiar la diferencia de lo medido por ambos. Además, medir la impedancia de los dos tipos de electrodos, medir señales con ellos en humanos con distintos tipos de cuero cabelludo y analizar la diferencia que haya entre las medidas. También se compraron electrodos húmedos de ECG, por lo que queda como trabajo a futuro adquirir señales de EEG con estos electrodos y comparar con los anteriores.

Casco: se logró diseñar una gorra para usar como casco. Como trabajo futuro queda hacer los agujeros donde deben conectarse los electrodos y darle rigidez al casco para medir señales utilizándolo. En su defecto, se puede comprar un casco ya hecho para medir señales neuronales. En el proyecto se analizó esta posibilidad y se observó que las empresas de electrodos que venden cascos, exigen por ellos un costo elevado.

Montaje del módulo remoto: si bien se compraron los componentes para armar los cables no dio el tiempo para armarlos. Queda como trabajo futuro también la sujeción de la carcasa al paciente. Se pensó en utilizar un cinto donde colocar la carcasa sobre el paciente, pero esta idea no se llevó a cabo.

Pruebas en humanos: queda como trabajo futuro medir señales en humanos.

Placa AFE: la placa diseñada y en la que fueron soldados los integrados RHD2132 sufrió muchos daños. Como trabajo a futuro se recomienda mudar todos los componentes a otra placa de AFE.

Accesibilidad de las señales de sincronismo: actualmente la caja no permite el acceso a las señales de sincronismo. Esto se hace solamente cortando la caja en el lugar adecuado para que el acceso a estos pines sea realizable.

Funciones extras del RHD2132: se recomienda estudiar a fondo las funcionalidades de este integrado e implementar muchas funciones que no se implementaron en esta versión del sistema. Ejemplos de estas son el test de impedancia de electrodos, que si bien está todo implementado no se probó su funcionamiento, el sensor de voltaje de alimentación, que permitiría determinar cuando se está agotando la batería que alimenta al módulo remoto y desplegarle un mensaje al usuario desde la interfaz para que la cambie y la activación de electrodos, que aplica un voltaje DC a los mismos para lograr su activación.

Sustituir a la placa AFE por un ASIC: como trabajo a futuro queda sustituir el AFE actual por un amplificador neural integrado, provisto por el tutor.

Capítulo 9. Conclusiones

En principio, el AFE actual puede ser cambiado directamente por otro AFE si el mismo puede conectarse a la placa MCU y se comunica por SPI con palabras de 16 bits como el sistema actual. Otra posibilidad es cambiar el AFE y los módulos de software embebido encargados de la comunicación con el mismo (módulo SPI.h y AFE.h).

LVDS: utilizar LVDS permite alcanzar velocidad mayores de comunicación SPI a las alcanzadas en este proyecto, por lo que un adelanto para la mejora de la velocidad de este sistema es utilizar un microcontrolador que sea capaz de comunicarse utilizando LVDS.

MCLK más estable: para evitar las variaciones en el MCLK reportadas en la sección 8.3.1, se debió utilizar el reloj externo HFTX que se provee en la placa de desarrollo del MSP432. Esta modificación es sencilla pero debido a falta de tiempo en el proyecto no se implementó.

Integrar al microcontrolador el algoritmo ACEEG: queda como trabajo a futuro integrar el algoritmo ACEEG. Para esto es necesario una memoria mayor (ya que el sistema actual ocupa toda la memoria del MSP432) y además mejorar las prestaciones de tiempos del sistema, dado que actualmente no dan los tiempos en el microcontrolador para agregar un procesamiento. El software embebido, está diseñado para que se pueda integrar el módulo que contenga el algoritmo ACEEG. Desde la interfaz, ya están implementados los comandos que permitan configurar si se va a usar el algoritmo ACEEG y los comandos que permiten setear algunos parámetros del mismo. Queda como trabajo a futuro verificar que dan los tiempos en la interfaz de usuario para realizar la descompresión y probar la compresión/-descompresión en el sistema completo.

Uso de DMA: La duración de 1 ms en la adquisición de una tirada puede mejorarse sustancialmente utilizando DMA para las comunicaciones SPI con el AFE. Si esto se hace, la adquisición de una tirada demoraría $500\mu s$ (@64 canales sin sincronismo) y se podría paralelizar el envío por la radio mientras se realiza la adquisición, lo que genera que las pérdidas por *desbordamiento de buffer de transmisión del microcontrolador a la radio* ya no sean la limitante, sino que lo sean las pérdidas por *superposición de adquisiciones de tiradas consecutivas*. Esta mejora permitiría que el sistema alcance una frecuencia de muestreo máxima de $2kHz$ (@64 canales sin sincronismo). Otra ventaja es que si se mejora la demora de los tiempos en el microcontrolador, se podría agregar el algoritmo ACEEG, o reducir el consumo ya que el sistema estaría mas tiempo en LPM. Como trabajo futuro se recomienda implementar el DMA y verificar que el análisis teórico resumido anteriormente permite esta mejora.

Adquisición de un solo canal: si bien actualmente funciona, se analizaron soluciones para que sea más rápido aunque no se implementaron. Por cada conversión de un canal se envían dos comandos DUMMY para obtener la respuesta de

9.4. Principales problemas enfrentados y lecciones aprendidas

forma inmediata. Esto se puede mejorar si se envía a convertir el canal sin enviar los comandos DUMMY, lo que podría llevar a que el sistema sea tres veces más rápido para un canal, llegando a una frecuencia máxima de 30kHz. Además para el caso de un solo canal, se deja encendido a todos los amplificadores del sistema. Esto podría mejorarse apagando todos los amplificadores que no se usan cuando se está adquiriendo solo un canal, lo que mejoraría el consumo del bloque AFE. El apagado de canales que no se usan, sí se realiza con las configuraciones de 21 canales y 4 canales, porque los canales a utilizar están preestablecidos.

4 canales personalizados: actualmente la configuración de 4 canales elige cuatro canales fijos como indica la figura 7.7. El sistema podría ser más versátil y permitir al usuario elegir con qué 4 canales realizar el estudio.

IP dinámicas: actualmente la IP se configura de forma estática en el código tanto del software embebido como en la interfaz de usuario, porque se sabe que IP estática les va a asignar el router. Como trabajo a futuro se debe hacer que la IP del módulo remoto se setee de forma dinámica y que la interfaz de usuario detecte cuál es. Esto permitiría que la interfaz de usuario se conecte con cualquiera de las radios y no sea necesario al comienzo de la interfaz indicar con que radio se debe conectar.

Pruebas de seguridad: una vez resumidas las normativas de equipos médicos, queda como trabajo a futuro realizar las pruebas de seguridad al sistema y verificar que se cumplen las normas necesarias para que el mismo pueda colocarse en pacientes y utilizarse para estudios clínicos.

9.4. Principales problemas enfrentados y lecciones aprendidas

A lo largo del proyecto se presentaron varios problemas y se cometieron varios errores, de los cuales se aprendieron valiosas lecciones, no solo de temas de ingeniería sino de formas de trabajar en general.

Hubo dificultades para buscar información en general, lo que llevó a varios errores y atrasos en el proyecto. Esto dificultó la elección de componentes como también la búsqueda de soluciones de problemas que ya habían sido reportados. Esto llevó a inconvenientes, por ejemplo, con la elección del protocolo de comunicación. En principio se eligió Bluetooth 4.0 pero este protocolo no cumplió con las expectativas en el throughput útil, perdiendo mucho tiempo, y teniendo que cambiar el protocolo y la radio a usar, por lo cual se aprendió la importancia de que en un proyecto final de carrera es necesario hacer una búsqueda exhaustiva, si es que se quiere obtener un throughput útil muy cercano al que el fabricante dice que el protocolo alcanza. Normalmente este valor es el que se obtiene en el aire y no el throughput real. Esto llevó a la compra de una radio Bluetooth que luego no

Capítulo 9. Conclusiones

fue utilizada en el proyecto.

Debido a que a veces no se trabajó en condiciones apropiadas y de forma prolija, se cometieron errores tales como quemar dos MSP432 (sin detectar la causante del problema), y soldar mal un conector en la placa AFE, que al desoldar de forma no apropiada, causó la pérdida de algunas pistas y como resultado la pérdida de algunos canales. Estos errores mostraron la importancia de trabajar de forma ordenada, en un ambiente apropiado y tener responsabilidad al realizar una tarea delicada como soldar. A pesar de este error se aprendió a soldar componentes pequeños en una placa utilizando un microscopio, y se vio cómo es el proceso de soldar un integrado por el método de reflow y los problemas que pueden surgir al utilizar esta técnica sobre una placa que ya tiene componentes soldados. Como consecuencia de estos errores, se quemó uno de los integrados RHD2132 y se perdieron tres canales del otro, terminando con un sistema que si bien está listo para muestrear 64 canales, solo funcionan 29 de ellos.

El sistema se dividió en distintos bloques (AFE, MCU, Radio, PC, etc), los cuales luego fue un desafío integrar en su totalidad para poder formar el sistema. En esta integración surgieron muchos problemas, algunos muy difíciles de solucionar, como por ejemplo el problema de canales libres y canales compartidos de Wi-Fi, el cual no sólo nos llevó mucho tiempo detectar, sino que nos obligó a hacer un sistema robusto a las pérdidas, ya que en principio estas pérdidas se producían sin explicación, y sin importar en las condiciones de frecuencia de muestreo y cantidad de canales que se trabajara.

Se aprendió a utilizar programas durante la realización del proyecto tales como Eagle (para el diseño de la placa AFE), Code Composer Studio (para el diseño del Software embebido), Acrylic (para conocer la potencia de las señales Wi-Fi en el aire), MatLab (para el desarrollo de la interfaz de usuario) y Latex (para realizar la documentación). Se aprendió de los integrados que componen el sistema, principalmente del integrado RHD2132 y del MSP432.

En cuanto a las pruebas el principal inconveniente fue la gran diferencia entre los MCLK (master clock) de dos microcontroladores de la misma familia. Esto como se indicó en el capítulo 8, llevó a realizar una interpolación de las medidas no deseada. Se observó que cuanto mayor sea el MCLK, mejor va a ser el desempeño del sistema.

Apéndice A

Explicación del no uso de ADS1299

En la sección 2.1.1 se mencionaron las desventajas del uso de este integrado:

- Son necesarias etapas de filtro pasabajo y pasaalto previas al ADS1299.
- Solo es capaz de relevar 8 canales, por lo que para llegar a los 64 canales requeridos es necesario buscar una solución alternativa con MUX analógicos o utilizar 8 integrados ADS1299.
- Los 24 bits que se indica en la hora de datos no aseguran ser libres de ruido, y uno de los requerimientos es que al menos 12 bits sean libres de ruido. Esto implica realizar un tratamiento extra previo al ADS1299.

En cuanto al segundo ítem se observó que la solución más óptima en cuanto a minimización de área y costo era agregar 8 MUX analógicos de 8:1 que permitirían utilizar solo un integrado ADS1299. Esto dejaba abierto a dos posibilidades en cuanto a la ubicación de los filtros programables como se muestra en la figura A.1.

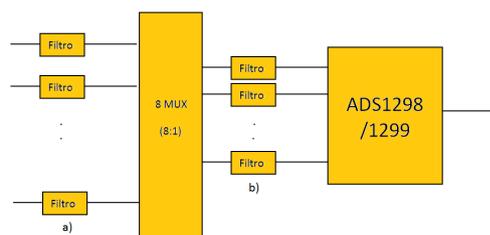


Figura A.1: Opción *a)*, colocar 64 filtros programables a la entrada del AFE. Opción *b)*, colocar 8 filtros programables a la salida del MUX analógico.

La opción *b)* tenía como inconveniente que se desconocía el ancho de banda de la señal luego del MUX y no era tan simple el filtrado tanto pasabajo como pasaalto, pese a utilizar 56 filtros menos que la opción *a)*. A su vez, por interferencia intersimbólica cada señal iba a ver afectado su valor en el punto de muestreo. La opción *a)* sin embargo, permitía aproximar el ancho de banda de la señal luego del MUX. Si B_S era el ancho de banda de cada señal ya filtrada previo del MUX, el

Apéndice A. Explicación del no uso de ADS1299

ancho de banda B luego del MUX sería $B \approx 8B_S$.

Dado que el ancho de banda de trabajo de cada señal (B_S) será de entre $50Hz$ y $100Hz$, la frecuencia de muestreo en el ADS1299 según el teorema de Nyquist tendrá que ser $f_s \geq 2B \approx 16B_S$.

A.1. ADS1299 sin amplificación externa

Pese a que su resolución era de 24 bits, era necesario asegurar al menos 12 bits libres de ruido. El ruido introducido por el integrado era creciente con f_s por lo que, a modo de verificar la validez del uso de esta topología de circuito se calculó la cantidad de bits contaminados de ruido para la frecuencia mínima de muestreo. $f_s \geq 16B_S = 800Hz \Rightarrow f_s = 1kHz$.

Ganancia	Bits libres de ruido	Bits contaminados	μV_p de ruido
1	17,50	6,50	24,27
2	17,43	6,57	12,76
4	17,31	6,69	6,94
6	17,32	6,68	4,60
8	17,14	6,86	3,90
12	16,84	7,16	3,21
24	16,04	7,96	2,79

Tabla A.1: Resumen del ruido agregado por el ADS para $f_s = 1kHz$ por canal (más información en las páginas 6 y 7 de la hoja de datos del integrado ADS1299).

La tabla A.1 muestra el ruido intrínseco que introduce el integrado a la entrada del mismo. Asumimos que los filtros y el MUX analógico no agregan ruido comparable con estos dos.

Se puede demostrar (ver anexo A.3) que para que el ruido del integrado no influya en la señal de entrada se tiene que cumplir que:

$$v_n \leq \frac{v_{pp}}{2^{b_u+1}} \quad (\text{A.1})$$

donde v_n es el ruido intrínseco del integrado ADS1299 cuyo valor se observa en la última columna de la tabla A.1, v_{pp} es la amplitud pico a pico de la señal de entrada ($v_{pp} = 200\mu V$), y b_u es la cantidad de bits libres de ruido que se desean ($b_u = 12$).

A.2. ADS1299 con amplificación externa

$\frac{v_{pp}}{2^{b_u+1}} = 24,41nV$, por lo tanto no hay valor de ganancia de la tabla A.1 que asegure cumplir la condición de la ecuación A.1.

A.2. ADS1299 con amplificación externa

Para minimizar la influencia del ruido intrínseco del integrado, se considera agregar una etapa de amplificación previa al MUX analógico. Si v'_n es el ruido a la entrada de este amplificador se puede demostrar (ver sección A.3) que se debe cumplir ahora que:

$$v'_n \leq \frac{v_{pp}}{2^{b_u+1}} \quad (A.2)$$

$$(G_{ext}v'_n)^2 \gg v_n^2 \quad (A.3)$$

donde v'_n es el ruido a la entrada del amplificador de bajo ruido y G_{ext} es la ganancia agregada por dicho amplificador.

Por tanto el amplificador a agregar debe tener un ruido menor a $24,41nV$. Dado que no se pudo encontrar amplificadores que tuviesen un ruido tan bajo en una banda de frecuencia de $500Hz$ y pudiesen contemplar otras características importantes del circuito se buscó otro tipo de solución.

A.3. Demostración de las ecuaciones A.2 y A.3

En esta sección se demuestra el resultado de las ecuaciones A.2 y A.3 mostradas en el anexo A.2.

Se definen algunos términos que serán importantes:

- v_n ruido de pico a la entrada del ADS1299 (previo a la amplificación del mismo).
- b_u resolución libre de ruido a utilizar.
- G_{ADS} ganancia del ADS (que va de 1 a 24).
- G_{ext} ganancia externa del amplificador a agregar.
- v_{pp} señal pico a pico de entrada al AFE proveniente de los electrodos (máximo $200\mu V$).
- V_{ref} valor máximo del convertor del ADS1299.
- v'_n ruido de pico a la entrada del amplificador que es necesario agregar.
- v_{nt} ruido de pico a la entrada del ADS debido a los dos efectos de ruido.
- b_{nt} bits de ruido asociado a los dos efectos de ruido.

Apéndice A. Explicación del no uso de ADS1299

- b bits utilizados del ADS ($b = \lceil b_{nt} \rceil + b_u \leq 24^1$).

Dado que los ruidos v_n' y v_n son independientes se tiene que $v_{nt} = \sqrt{(G_{ext}v_n')^2 + v_n^2}$.

A su vez, la cantidad de bits de ruido se obtiene como:

$$2v_{nt} = \frac{V_{ref}}{2^{24}G_{ADS}} 2^{b_{nt}}$$

Se agrega un 2 en el primer término puesto que se debe considerar el ruido pico a pico.

La ganancia externa está acotada por la siguiente ecuación:

$$v_{pp}G_{ext}G_{ADS} \leq \frac{V_{ref}2^b}{2^{24}}$$

en la cual, para optimizar el uso de los bits disponibles, se busca que:

$$v_{pp}G_{ext}G_{ADS} \approx \frac{V_{ref}2^b}{2^{24}}$$

Dado que la función techo cumple que $b_{nt} \leq \lceil b_{nt} \rceil$, entonces:

$$\begin{aligned} 2v_{nt} &= \frac{V_{ref}2^{b_{nt}}}{2^{24}G_{ADS}} \leq \frac{V_{ref}2^{\lceil b_{nt} \rceil}}{2^{24}G_{ADS}} = \frac{V_{ref}2^{\lceil b_{nt} \rceil + b_u}}{2^{24}G_{ADS}2^{b_u}} = \frac{1}{G_{ADS}2^{b_u}} \frac{V_{ref}2^b}{2^{24}} = \frac{v_{pp}G_{ext}G_{ADS}}{G_{ADS}2^{b_u}} = \frac{v_{pp}G_{ext}}{2^{b_u}} \Rightarrow \\ v_{nt} &\leq \frac{v_{pp}G_{ext}}{2^{b_u+1}} \end{aligned} \quad (\text{A.4})$$

El caso del anexo A.1 es un caso particular, en el cual $G_{ext} = 1$ y $v_n' = 0V$. De esta manera $v_{nt} = v_n$ y la ecuación A.4 se transforma en la ecuación A.1.

Si por el contrario, se quiere que el ruido del amplificador externo domine como término en el ruido total a la entrada del ADS1299, se impone que:

$$(G_{ext}v_n')^2 \gg v_n^2 \Rightarrow v_{nt} \approx G_{ext}v_n' \Rightarrow G_{ext}v_n' \leq \frac{v_{pp}G_{ext}}{2^{b_u+1}}$$

Que dan paso a la ecuación A.2.

¹ $\lceil x \rceil$ representa a la función techo.

Apéndice B

Información complementaria a la elección de componentes

B.1. Tablas con todas las opciones de radio manejadas

Requerimientos	ESP8266EX	CC3100	ATSAMW25 MR210PA MR210PB	AMW006 / AMW106	
Fabricante	Espressif Systems IOT Team	Texas Instrument	ATMEL	ACKme Networks	
Memoria	Ram	<36 kB	700 B	32 kB	
	Flash	<16 MB (external stack)	7 kB	256 kB	
Consumo	Tx max	b - 170 mA g - 140 mA n - 120 mA	1 DSSS - 272 mA (L0), 188 mA 6 OFDM - 248 mA (L0), 179 mA 54 OFDM - 223 mA (L0), 160 mA	No aparece data	No aparece data
	Rx max	b- 50 mA g - 56 mA n - 56 mA	1 DSSS - 53 mA 54 OFDM - 53 mA	No aparece data	No aparece data
	Otros	80 mA - Operation current 12 mA - I/O máx 15 mA - Modem sleep 10µA - Deep sleep	6 mA - High-level source current 450 mA - Peak calibration current 6 mA - Low-level sink current	13 mA - VDD IO H	5,7 mA MCU Stop mode, Wi-Fi rx 11,4 mA MCU Stop mode, Wi-Fi tx 320 mA MCU Stop mode, Wi-Fi tx 5,7mA Active receive @ 1Mbit/s UDP
Interfaz Serie	SPI, UART, otros...	SPI, UART	SPI, UART	SPI, UART	
Configuración (Plug & play, AT, otra)	ATInstruction Set, Cloud Server, Android	Comandos propios	Comandos propios	Drivers provistos por ellos stream mode	
Tipos de protocolos 802.11	b,g,n	b,g,n	b,g,n	b,g,n	
Voltaje de operación (V)	3,0-3,6	2,1-3,6	2,7 - 3,6	3,0-3,6	
Stack TCP/IP	ok	ok	ok	ok	
Precio (U\$S)	9,95 (ESP8266 - 2471) 11,75 (317060018)	31,00 (con Launchpad para MSP432)	42,65 (digikey)	44,00 (digikey)	

Tabla B.1: Opciones de radios Wi-Fi a utilizarse en proyecto.

Apéndice B. Información complementaria a la elección de componentes

Requerimientos	RN1810/RN1810E	ISM43362-M3G-L44	MRF24WB0MA/B, MRF24WG0MA/B for MLA v5	RN -131	
Fabricante	Microchip	Inventek Systems	Microchip	Roving Networks	
Memoria	Ram	No aparece data	No aparece data	No aparece data	128 kB
	Flash	No aparece data	1 MB	No aparece data	8 MB
Consumo	Tx max	246 mA at 18 dBm (typical)	No indica en tabla	115 mA - IDD core, TX on, +0 dBm 154 mA - IDD core, TX on, +10 dBm	210 mA
	Rx max	64 mA (typical)	No indica en tabla	85mA - IDD core, RX on, Receive @-2 Mbps modulated signal at antenna port	40 mA
	Otros	0,012 mA - Sleep (typical) 20 mA - High and Low-Level Output	Typical current Wi-Fi connected 110 mA. During tx current 340 mA 5ms.	15,4 mA - IOH	24 mA - I/O 15 mA Standby
Interfaz Serie	UART	SPI, UART	SPI	SPI, UART (up to 1 Mbps for the UART)	
Configuración (Plug & play, AT, otra)	wiFly	AT	wiFly	wiFly	
Tipos de protocolos 802.11	b,g,n	b,g,n	b,g,n	b,g	
Voltaje de operación (V)	3,15-3,45	3,0-3,6	2,7-3,6	3,0-3,7	
Stack TCP/IP	ok	ok	ok	ok	
Precio (US\$)	49,95 (digikey)	90,00 (digikey)	49,99 (digikey)	-	

Tabla B.2: Opciones de Radios Wi-Fi a utilizarse en proyecto.

B.2. Router para la comunicación Wi-Fi

Debido a que se utilizará Wi-Fi, existe la necesidad de crear una red inalámbrica como se expone en la sección 5.3. Esta red inalámbrica puede ser creada por un Access Point¹ o por un Router². Dado a que existen en el mercado local, se buscaron Routers o Access Point disponibles en el país. Se priorizó la búsqueda en:

- Velocidad máxima del Router.
- Costo.

La tabla B.3 muestra una lista de algunos de los routers o access point que fueron investigados en la búsqueda.

Como se observa en la tabla B.3, los Access Point son más costosos y alcanzan una mayor velocidad, sin embargo que para las necesidades de nuestro proyecto nos alcanza con las velocidades que manejan los Routers.

¹El Access Point es un dispositivo de red que interconecta equipos de comunicación inalámbricos de la misma red.

²El Router es un dispositivo que intercomunica distintas redes [51].

B.3. Conversores DC/DC

Tipo	Marca	Modelo	Velocidad	Costo (U\$S)	Comentario
Access Point	TP-LINK	EAP 120	1000 Mbps	99	
Access Point	TP-LINK	CPE210	300 Mbps	70	Diseño para exterior
Router	TP-LINK	TL-WR720N	150 Mbps	17	2 antenas internas
Router	TP-LINK	TL-WR845N	300 Mbps	29	3 antenas exrternas
Router	LINKSYS	E900	300 Mbps	43	2 antenas internas
Router	LINKSYS	WRT54G2	54 Mbps	23	1 antena interna

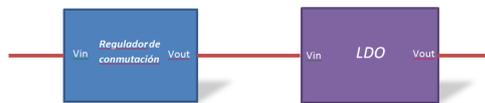
Tabla B.3: Posibles Routers a utilizar en el proyecto.

Se decidió utilizar el Router CISCO LYNKSYS E900, puesto que tiene la velocidad necesaria y porque permite setear una IP fija para los dispositivos que se comunican (esto último es necesario, en la versión actual del sistema, para que la interfaz sepa que IP tiene el módulo remoto y poder iniciar la comunicación).

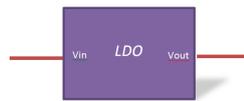
B.3. Conversores DC/DC

B.3.1. Elección del convertor DC/DC para los chips RHD2132

El fabricante del RHD2132 recomendó la utilización de un convertor lineal (LDO)³ previo a la alimentación del chip, evaluando entre dos posibles configuraciones como muestran las figuras B.1a y B.1b.



(a) Configuración con un regulador de conmutación seguido de un LDO.



(b) Configuración solo con un LDO.

Figura B.1: Posibles configuraciones para la regulación de la tensión de la batería.

La eficiencia de cualquier LDO a utilizar será $\eta = \frac{V_{out}}{V_{in}} = \frac{3,3V}{3,7V} = 89\%$ puesto que como se indicó en la sección 2.4 se utilizará una pila de $3,7V$. Como la tensión

³Un LDO es un tipo de convertor DC/DC que puede regular cuando las tensiones de salida y de entrada son similares. Su característica de lineal hace que la eficiencia del mismo sea independiente de la corriente de salida en un amplio rango de voltaje, siendo su cálculo de eficiencia como $\eta = \frac{V_{out}}{V_{in}}$.

Apéndice B. Información complementaria a la elección de componentes

de salida deseada ($3,3V$) es muy cercana a la tensión de entrada ($3,7V$) no es necesario colocar un convertor extra además del LDO ⁴.

Por la razón anterior se centralizó la búsqueda en convertidores LDO. El criterio de búsqueda se centró en las siguientes características:

- Rango de tensiones de entrada que admite.
- Cantidad de componentes externos necesarios.
- Costo.
- Voltaje de dropout ⁵.

La corriente de salida no fue un requerimiento fundamental en este caso, debido al bajo consumo de los integrados RHD2132 (ver sección 2.4.1). La tabla B.4 muestra la comparación de algunos LDO investigados.

	Número de componentes externos	V_{in}	V_{out}	Costo	Dropout Voltage
NCV8535	5	12V - 3,7V	3,3V	US\$ 1,8	110 mV @ 50 mA
MAX15006	4	40V - 4V	3,3V	–	–
ADP150	2	5,5V - 3,35V	3,3V	US\$ 1	105 mV @ 150 mA
MIC5301	3	5,5V - 2,3V	3,3V	US\$ 1,5	40 mV @ 150 mA

Tabla B.4: Comparación de algunos de los convertidores LDO estudiados, en todos los componentes la eficiencia viene dada por η .

En la tabla B.4 se muestra la comparación de los distintos convertidores. Se elige el ADP150 porque optimiza en costo y cantidad de componentes externos, a la vez que regula en el rango de la pila elegida en la sección 2.4.2 y tiene el voltaje de dropout más chico.

B.3.2. Elección de convertor DC/DC para los bloques MCU y Radio

A partir de la tabla 2.4 se tiene que, en transmisión, el CC3100 puede alcanzar picos de $272mA$ por lo que para este convertor DC/DC sí era necesario considerar la corriente de salida capaz de entregar.

La tabla B.5 compara una lista de algunos de los convertidores DC/DC observados.

⁴El caso de utilizar dos convertidores en serie es justificado cuando las tensiones de entrada y salida difieren mucho más; como ejemplo, si $V_{in} = 5V$ y $V_{out} = 3,3V$.

⁵El voltaje de dropout es la mínima diferencia que puede haber entre el voltaje de entrada y el voltaje de salida del convertor DC/DC de manera que el convertor DC/DC asegure el voltaje de salida deseado.

B.4. Test de algunas de las baterías

Convertor	I_{LOAD}	Costo	¿es LDO?	Dropout Voltage
ADP150	150 mA	US\$ 1	Sí	105 mV @ 150 mA
NCV8535	500 mA	US\$ 1,8	Sí	230 mV @ 300 mA
MIC5301	150 mA	US\$ 1,5	Sí	40 mV @ 150 mA
TPS62740	300 mA	US\$ 1,06	No	-
Torex ETR2802-006	400 mA	US\$ 1,23	No	-
RECOM R1S	303 mA	US\$ 7,5	No	-
Murata LXDC2HN	300 mA	US\$ 1,02	No	-
TSP736	500 mA	US\$ 0,84	Sí	75 mV @ 500 mA

Tabla B.5: Comparación de algunos de los convertidores LDO estudiados.

A partir de la comparación se decide elegir el convertor DC/DC TSP736 de Texas Instruments por ser el de menor costo y que es capaz de entregar la corriente deseada.

B.4. Test de algunas de las baterías

La figura B.2 muestra el test para la pila UltraFire 18650.

Name	UltraFire TR18650 5000mAh (Blue)					
Cell						
Supplier	Ebay: fastpostitem88				Date:	4-2014
Size	Weight:	38.7 g	Length:	66.0 mm	Diameter:	18.2 mm
Info	Top:	SB	Bottom:	metal	Rated A:	
Test condition	Charge voltage:			4.2	Termination current:	
				0,1		
Test current (A)	0,2	0,5	1	2	3	5
Measured capacity (Ah)	1,124	1,123	1,095	1,052	0,955	0,626
Measured energy (Wh)	4,259	4,287	4,096	3,790	3,312	2,024
PCB protection trip current (A)	NA					
Calculated internal resistance (ohm)	0,49					

Figura B.2: Test de la pila UltraFire 18650 5000 mAh. La figura fue extraída de [52]. Para más información del test visitar el link en la referencia.

Como se puede observar, la carga que realmente entrega es cinco veces menor a la carga nominal de la pila, por lo que su rendimiento es solo un 20 % del esperado.

Las figuras B.3 y B.4 muestran los tests para las pilas Samsung ICR18650-26F y Panasonic NCR18650B.

Se observa que las pilas fabricadas por empresas más reconocidas en el mercado como son Panasonic y Samsung, dan una mayor confiabilidad en sus números con respecto a lo que ocurre con la pila UltraFire.

Apéndice B. Información complementaria a la elección de componentes

Name	Samsung ICR18650-26F 2600mAh (Pink)					
Cell	Samsung ICR18650-26F					
Supplier	Akkuteile.de				Date:	6-2014
Size	Weight:	44.8 g	Length:	64.8 mm	Diameter:	18.3 mm
Info	Top:	flat	Bottom:	metal	Rated A:	5.2
Test condition	Charge voltage:		4.2	Termination current:		0,1
Test current (A)	0,2	0,5	1	2	3	5
Measured capacity (Ah)	2,590	2,580	2,549	2,518	2,511	2,519
Measured energy (Wh)	9,762	9,648	9,418	9,102	8,900	8,601
PCB protection trip current (A)	NA					
Calculated internal resistance (ohm)	0,09					

Figura B.3: Test de la pila Samsung ICR18650-26F 2600 mAh. La figura fue extraída de [53]. Para más información del test visitar el link en la referencia.

Name	Panasonic NCR18650B 3400mAh (Green)					
Cell	Panasonic NCR18650B					
Supplier	cn qualti goods				Date:	6-2012
Size	Weight:	45.9 g	Length:	65 mm	Diameter:	18.3 mm
Info	Top:	flat	Bottom:	metal	Rated A:	
Test condition	Charge voltage:		4.2	Termination current:		0,1
Test current (A)	0,2	0,5	1	2	3	5
Measured capacity (Ah)	3,326	3,270	3,197	3,128	3,103	3,080
Measured energy (Wh)	12,142	11,856	11,472	11,014	10,727	10,296
PCB protection trip current (A)	NA					
Calculated internal resistance (ohm)	0,11					

Figura B.4: Test de la pila Panasonic NCR18650B 3400 mAh. La figura fue extraída de [54]. Para más información del test visitar el link en la referencia.

B.5. Cable para electrodos

Como se menciona en la sección 7.3, la conexión entre los electrodos y el AFE se puede hacer en dos tramos. Para el segundo tramo (nuca-AFE) se utiliza un cable IDE. El costo del mismo fue de US\$ 14,70 y trae 3 metros de cable.

Para el cable que debe conectar los electrodos con el cable plano se pensaron las siguientes opciones:

- Comprar los electrodos de disco que vienen con cable incluido.
- Comprar cables para electrodos ECG.
- Diseñar el cable con un cable plano y botones de mercería.

La opción de comprar los electrodos de disco tiene un costo por unidad de \$933 incluyendo el costo de la pasta. Los cables para electrodos de ECG de Química Cenit tienen un costo de \$338 por unidad. Para el diseño con cable plano, se tiene en cuenta que se aprovecharía el cable utilizado para el diseño del tramo nuca-AFE, estando solo en el costo las conexiones con los electrodos que se realizan con botones de mercería.

Debido a la diferencias de costos, se optó por diseñar el cable como se menciona en el tercer punto. Como el diseño del mismo puede introducir ruido, se decidió

por comprar 4 cables para electrodos de ECG para comparar.

B.6. Casco

Para la compra del casco se consultaron precios de distintas empresas encargadas de diseñar dispositivos de EEG. Estos precios eran elevados para el presupuesto que se tiene para hacer el sistema.

Por esto, se optó por diseñar el casco. El diseño del casco fue realizado por una modista, a quien se le proporcionó los materiales. El costo de los materiales y mano de obra se detallan en la tabla B.6.

	Precio(\$)
Tela Modal	99
Mano de obra de la modista (por 3 gorras)	600
Hebillas de plástico x 6	90
Total	789

Tabla B.6: Precios de materiales y mano de obra de la modista.

B.7. Carcasa

Para proteger de forma eléctrica y mecánica al módulo remoto se utiliza una carcasa. Se consideran dos posibles maneras de ubicar los componentes del módulo remoto que son las placas y el portapila; las mismas se muestran en la figura B.5, indicando en cada caso las dimensiones mínimas que debe tener la carcasa.

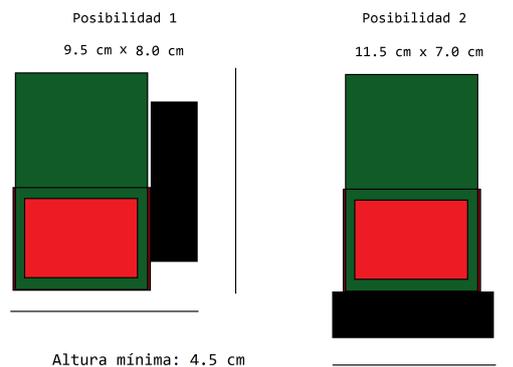


Figura B.5: Dos configuraciones posibles de los componentes del módulo remoto.

Para el diseño del sistema interesan las siguientes características de la carcasa:

Apéndice B. Información complementaria a la elección de componentes

- Bajo costo de la carcasa.
- Dimensiones efectivas que permitan ingresar a todo el módulo remoto⁶.
- Material de la carcasa.

Con respecto al material de la carcasa, nos interesa principalmente que no sea conductivo, de manera que no se forme una jaula de Faraday, impidiendo la comunicación de la radio con el exterior. Se analizaron más de 50 carcasas, la tabla B.7 muestra alguno de los resultados estudiados.

Numero de parte en digikey	Dimensiones (LAH)	Tamaño efectivo (LAH)	Precio
377-1216-ND	11,70cm x 7,88cm x 4,76cm	10cm x 8,30cm x 4,50cm	USD 7,6
377-1220-ND	12,88cm x 9,07cm x 4,04cm	10,43cm x 6,41cm x 3,31cm	USD 7
377-1121-ND	11,50cm x 9,0cm x 5,5cm	6,80cm x 9,00cm x 3,00cm	USD 10,35
CU-3282-MB-ND	11,71cm x 11,71cm x 5,98cm	10,00cm x 11,11cm x 5,73cm	USD 11,6
377-1542-ND	11,5cm x 9,0cm x 5,5cm	7,70cm x 6,81cm x 4,17cm	USD 14,6
HM935-ND	11,20cm x 9,0cm x 6,0cm	95,88cm x 68cm x 51cm	USD 12,61

Tabla B.7: Tabla de algunas de las carcasas estudiadas. Todas las opciones mostradas en la tabla son de plástico.

Se elije la carcasa “377-1216-ND”, porque optimiza en área y costo.

B.8. Cargador de baterías

Fue necesario comprar también un cargador y un portapila del tamaño 18650⁷. El cargador elegido (el TrustFire Multifuncional Charger) puede cargar 2 pilas por separado a dos diferentes voltajes 4,2V y 3,0V. Viene con el cable para cargar en el auto y con el cable para cargar a 230V.

⁶Nos referimos con dimensión efectiva, al espacio en el interior de la carcasa, descontando el espacio que ocupan los tornillos.

⁷El tamaño 18650 es un valor estándar que corresponde a una pila con dimensiones de 65 mm de largo y un diámetro de 18 mm (extraído de [55]).

Apéndice C

Low Voltage Differential Signal (LVDS)

LVDS (low voltage differential signal) es un sistema de transmisión de señales a alta velocidad, cuya diferencia con el formato SPI estándar es que envía las señales por duplicado pero con diferente voltaje (una señal es transmitida con signo positivo mientras que la otra es transmitida con signo negativo). Cuando la señal llega al microcontrolador, éste se encarga de comparar las señales y comprobar cual de las dos tiene mayor potencial y asignar el valor lógico que corresponda. [56].

El chip RHD2132 permite trabajar con LVDS o con una comunicación SPI estándar. A continuación se evalúa el funcionamiento de cada uno de ellos:

C.1. Funcionamiento de los dos tipos de comunicación SPI

El funcionamiento con SPI estándar implica el uso de 4 pines para cada integrado RHD2132:

- $\overline{CS+}$: Activa al RHD2132 por flanco de bajada.
- $CLK+$: Reloj del sistema.
- $MOSI+$: Entrada para enviar señales al RHD2132 desde el microcontrolador.
- $MISO+$: Salida para enviar señales al microcontrolador desde el RHD2132.

Los pines $\overline{CS-}$, $SCLK-$, $MOSI-$ y $MISO-$ no se utilizan y deben dejarse sin conectar. (ver figura 3.4).

Si se desea utilizar LVDS deben conectarse 8 pines para la transmisión de cada chip:

- $\overline{CS+}$.
- $\overline{CS-}$.

Apéndice C. Low Voltage Differential Signal (LVDS)

- SCLK+.
- SCLK-.
- MOSI+.
- MOSI-.
- MISO+.
- MISO-.

Los pines “+” funcionan igual que en SPI estándar y los pines “-” son una réplica de la señal en los pines “+” pero con el signo opuesto.

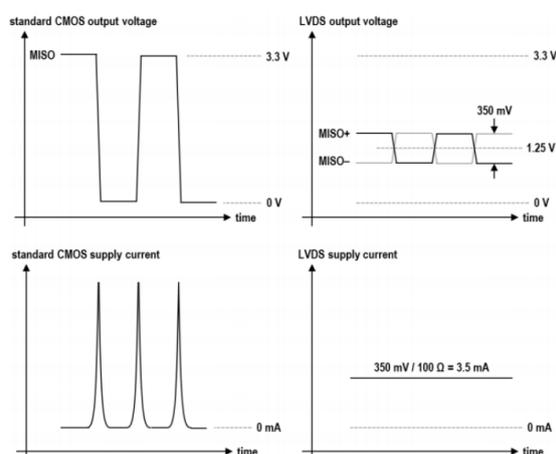


Figura C.1: Formas de onda del voltaje y corriente para funcionamiento con LVDS o sin LVDS (imagen extraída de la página 11 de [30]).

La figura C.1 muestra las formas de onda con o sin LVDS para la comunicación con el microcontrolador. Como se observa, cuando se envía una señal por SPI estándar, un '1' lógico se corresponde con una tensión de VDD (3,3V) y un '0' lógico con GND (0V). La corriente en este caso será muy cercana a cero cuando se mantenga el valor lógico y tendrá picos de corriente cuando la salida cambia de estado, cargando o descargando las capacidades parásitas del pin de salida. Estos golpes de energía introducen ruido de alta frecuencia que afectan de forma adversa los niveles de ruido. Cuanto mayor sea la tasa de muestreo, los picos de corriente ocurrirán en períodos de tiempo más cortos, por lo que su perjuicio en el ruido será mayor. El valor nominal del ruido a la entrada del chip RHD2132 es de $2,4\mu V_{rms}$; operando a 350kS/s, incluso con cables cortos, el incremento en el ruido es de al menos un 10 % del valor nominal ($2,64\mu V_{rms}$). [30] recomienda que para aplicaciones en las que interesa operar a bajo ruido, se trabaje con una tasa de muestreo menor a 175 kS/s (5kS/s por canal).

Cuanto mayor es la tasa de muestreo del ADC mayor será la potencia disipada (el cálculo de la misma se indica en la página 11 de [30]) y mayor serán las reflexiones que interfieren con la señal transmitida (a frecuencias altas el sistema

C.2. Ventajas del uso de LVDS

empieza a mostrar inconvenientes de radiofrecuencia). Para solucionar esto último se plantea en [30] una solución imperfecta que es adaptar en paralelo con una impedancia Z_0 ; esta solución sin embargo absorbe las primeras reflexiones pero no es óptima a altas velocidad o con cables de transmisión largos.

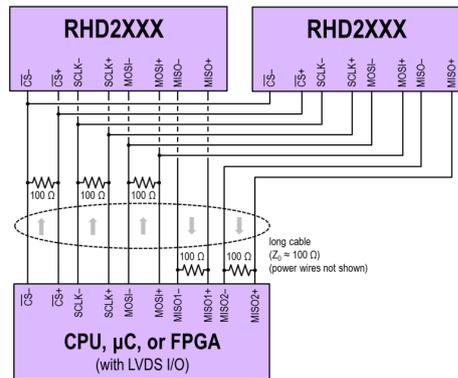


Figura C.2: Diagrama de la comunicación de dos integrados RHD2132 con un microcontrolador si se utiliza LVDS (imagen extraída de la página 14 de [30]).

Si se observa el LVDS en la figura C.1, se tiene que las señales son enviadas con una tensión en modo común de $1,25V$, y debido a la presencia de una resistencia de 100Ω entre los pines “+” y “-” (como muestra la figura C.2) y a que la corriente se mantiene forzada a ser $3,5mA$ en una dirección u otra, se logra crear una diferencia de potencial de $\pm 350mV$ que deberá ser interpretado como un ‘1’ ($1,25V + 0,350V$) o un ‘0’ ($1,25V - 0,350V$).

Que la corriente se mantenga constante en $3,5mA$ hace que no haya picos que aumenten el ruido a la entrada del chip RHD2132. Tener resistencias de 100Ω conectadas entre los pines “+” y “-”, reduce las reflexiones por línea de transmisión, siendo por tanto más adecuado para altas tasas de muestreo y cables largos. Con LVDS se logra minimizar también la interferencia electromagnética.

Se resumen a continuación las ventajas y desventajas de utilizar LVDS por encima de SPI estándar.

C.2. Ventajas del uso de LVDS

- Al enviar la señal dos veces (redundancia) se permite eliminar errores en la transmisión de datos. Si bien las distancias son cortas y con SPI la probabilidad de error suele ser baja, debe tenerse en cuenta que la comunicación entre el RHD2132 y el MSP432 implica tener que cambiar de placa por conectores machos y hembras que pueden generar deterioro de la señal enviada y aumentar la probabilidad de error.

Apéndice C. Low Voltage Differential Signal (LVDS)

- La comunicación con SPI estándar aumenta el ruido a la entrada del RHD2132. A altas tasas de muestreo y utilizando cables largos, se podría llegar a empeorar el ruido a la entrada en más de 30% del valor nominal ($2,4\mu V_{rms} \times 1,3 = 3,12\mu V_{rms}$).
- Se logra disipar menos potencia con LVDS que con SPI estándar.
- Se minimiza la interferencia electromagnética en comparación con SPI estándar.
- Al tener conectado resistencias entre los pines “+” y “-”, reduce las reflexiones por línea de transmisión, siendo más adecuado para altas tasas de muestreo y cables largos en comparación con SPI estándar.
- Según lo indicado en la página 27 de [30], si no se utiliza LVDS, debe ser agregado un condensador de desacople adicional de 100nF en el pin 31 (uno de los pines de VDD del chip).

C.3. Desventajas del uso del LVDS

- El MSP432 no tiene la capacidad de manejar una comunicación con LVDS. La implementación de dicho sistema debería hacerse con un traductor LVDS a SPI que debe ubicarse cercano al microcontrolador. Esto no es posible puesto que el MSP432 viene ensamblado en una placa de desarrollo. Utilizar LVDS implicaría realizar una placa en la cual se encuentren este traductor y el MSP432.
- Por defecto, el MSP432 trabaja con niveles lógicos de 3,3V para un '1' y 0V para un '0'. Esto hace que la comunicación entre el RHD2132 y el MSP432 sea directa y simple con SPI estándar puesto que los dos funcionan con los mismos niveles.
- Como se observa en la figura C.1, la comunicación con LVDS maneja rangos de tensión para los valores lógicos distintos a los de SPI, por lo que los umbrales del MSP432 no leería los valores lógicos correctamente.
- Utilizar la comunicación LVDS implica un consumo extra de $5,7mA$ por cada RHD2132 en la placa AFE, por lo que en el caso del proyecto, al utilizar dos RHD2132, el consumo extra con respecto a SPI estándar sería de $11,4mA$. Esta cantidad es significativa con respecto al consumo del chip puesto que sin utilizar LVDS, el consumo sería de aproximadamente $1,81mA$ (ver sección 2.4.1), pero no es tan relevante en el consumo total del módulo remoto (AFE + μP + Radio) que sin LVDS, tiene un consumo estimado de $282,85mA$ (ver sección 2.4.2).
- Como se indica en la sección C.1, si la aplicación es de bajo ruido, conviene usar una tasa de muestreo por canal menor a 5kS/s. En este proyecto, ninguna de las configuraciones a utilizar superan esta cantidad por canal. Esto implica que los valores de tasa de muestreo no llevarían a un deterioro del nivel de ruido a la entrada utilizando SPI estándar.

Por todos estos motivos es que decide trabajar con SPI estándar.

Apéndice D

Conexión de cada pin del AFE

Se detalla a continuación todos los pines de los integrados RHD2132, se da una breve descripción de su función y se indica cómo se los conecta en la placa AFE y con qué propósito. El nombre de cada pin es coherente con [30] y con la figura 3.4.

- $\overline{\text{CS}}+$, $\text{CLK}+$, $\text{MISO}+$, $\text{MOSI}+$ permite la comunicación SPI Standard.
- $\overline{\text{CS}}-$, $\text{CLK}-$, $\text{MISO}-$, $\text{MOSI}-$ permite la comunicación SPI utilizando LVDS.
- **LVDS_en** se utiliza para activar o desactivar el LVDS. Este pin es una entrada digital y en el sistema permanecerá conectado a un '0' lógico porque no se implementará la comunicación LVDS (ver anexo C).
- **VESD** debe estar la mayoría del tiempo a tierra pero debe tomar el valor VDD mientras se realiza la activación del electrodo (ver página 31 de [30]). En la placa AFE se colocan tres pines: el pin del medio tendrá la señal VESD de los dos chips, uno de los pines de los extremos se conectará a un conector al que se podrá tener acceso desde otra placa y el pin restante estará conectado a tierra. En caso de no utilizar la función de activación de electrodos, debe cortocircuitarse VESD y tierra utilizando un jumper. En caso de utilizarse la función de activación de electrodos, debe cortocircuitarse VESD con el otro de los pines. Ver la figura D.1 para mayor interpretación.
- **elec_test** se utiliza para generar una onda externa al chip que permita medir la impedancia de los electrodos. Es una entrada digital y se conectará al exterior mediante un jumper. Dado que en el sistema se utilizará el generador de ondas interno del RHD2132, en el funcionamiento normal, ese jumper no se conectará y el circuito se va a mantener abierto, puesto que es el estado recomendado por el fabricante del RHD2132 cuando el pin `elec_test` no se utiliza para su fin.
- **auxout** es una salida brindada por el RHD2132. En la placa AFE se conecta este pin a un conector mediante un jumper que permita que una placa externa pueda tener acceso. En esta versión del sistema, este jumper permanecerá abierto.

Apéndice D. Conexión de cada pin del AFE

- **auxin1, auxin2 y auxin3** son entradas analógicas que permitirán realizar los estudios con señales de sincronismo.



Figura D.1: Jumper para tener disponible la función de activación de electrodos. En esta versión del sistema no se va a realizar la activación de electrodos, por lo tanto este jumper va siempre conectado donde indica la figura. Si se desea implementar la activación de electrodos, el valor de VESD se impone con el microcontrolador a través del pin 14 del conector MSP3.

Se utilizan 16 conectores en esta placa para permitir la comunicación con el microcontrolador. La tabla D.1 resume información relevante sobre los mismos.

Nombre del conector	Tipo	Número de pines	Propósito	Capa de la placa en la que se ubica
CON1.ELEC	Macho	34	Conexión de los electrodos a la placa	Capa inferior
CON2.ELEC	Macho	34	Conexión de los electrodos a la placa	Capa inferior
MSP1	Macho	4	Rigidez mecánica al conectar con MSP432	Capa superior
MSP2	Macho	4	Rigidez mecánica al conectar con MSP432	Capa superior
CON.EST	Macho	8	Conexión señales de sincronismo al módulo remoto	Capa inferior
CON.BAT	Hembra	2	Voltaje y tierra de la batería para ser rectificadas en la placa	Capa inferior
MSP3	Macho	30	Conexión entre los chips RHD2132 y el MSP432	Capa superior
JP1, JP2, JP3 y JP4	Macho	2	Soporte para un escudo de tierra contra interferencia de señales de la radio	Capa superior
JP5	Macho	2	Jumper para las distintas conexiones de ref_elec	Capa inferior
JP6	Macho	2	Jumper para las distintas conexiones de elec_test	Capa inferior
JP7	Macho	3	Jumper para las distintas conexiones de VESD	Capa inferior
JP8	Macho	2	Jumper para las distintas conexiones de AUXOUT del chip 0	Capa inferior
JP9	Macho	2	Jumper para las distintas conexiones de AUXOUT del chip 1	Capa inferior

Tabla D.1: Resumen de los conectores de la placa y su funcionalidad.

La figura 3.1 muestra la ubicación de cada uno de los conectores.

A continuación se detalla las señales que se conectan a los conectores MSP3 y CON_EST. Se numeran los integrados RHD2132 como sigue: “chip 0” corresponde al RHD2132 situado a la izquierda y “chip 1” corresponde al RHD2132 situado a la derecha (ver figura 3.1).

Pin RHD2132	Tipo	Conector/Pin placa AFE	Pin MSP432
CS+ del chip 0	Entrada digital	MSP3 pin 10	9.4
SCLK+ del chip 0	Entrada digital	MSP3 pin 11	9.5
MOSI+ del chip 0	Entrada digital	MSP3 pin 13	9.7
MISO+ del chip 0	Salida digital	MSP3 pin 12	9.6
CS+ del chip 1	Entrada digital	MSP3 pin 20	10.0
SCLK+ del chip 1	Entrada digital	MSP3 pin 19	10.1
MOSI+ del chip 1	Entrada digital	MSP3 pin 22	10.2
MISO+ del chip 1	Salida digital	MSP3 pin 21	10.3
CS- del chip 0 ¹	Entrada digital	MSP3 pin 6	7.3 (no se usa desde MSP432)
SCLK- del chip 0	Entrada digital	MSP3 pin 7	7.2 (no se usa desde MSP432)
MOSI- del chip 0	Entrada digital	MSP3 pin 8	7.1 (no se usa desde MSP432)
MISO- del chip 0	Salida digital	MSP3 pin 9	7.0 (no se usa desde MSP432)
CS- del chip 1	Entrada digital	MSP3 pin 17	7.7 (no se usa desde MSP432)
SCLK- del chip 1	Entrada digital	MSP3 pin 24	10.4 (no se usa desde MSP432)
MISO- del chip 1	Salida digital	MSP3 pin 18	7.6 (no se usa desde MSP432)
MOSI- del chip 1	Entrada digital	MSP3 pin 23	10.5 (no se usa desde MSP432)
AUXIN 1 del chip 0	Entrada analógica	CON_EST pin 8	
AUXIN 2 del chip 0	Entrada analógica	CON_EST pin 7	
AUXIN 3 del chip 0	Entrada analógica	CON_EST pin 6	
AUXIN 1 del chip 1	Entrada analógica	CON_EST pin 5	
AUXIN 2 del chip 1	Entrada analógica	CON_EST pin 4	
AUXIN 3 del chip 1	Entrada analógica	CON_EST pin 3 y MSP3 pin 2	5.3
AUXOUT del chip 0	Salida digital	MSP3 pin 4	6.2 (no se usa desde MSP432)
AUXOUT del chip 1	Salida digital	MSP3 pin 3	9.3 (no se usa desde MSP432)
VESD.CON de ambos chips	Entrada digital	MSP3 pin 14	8.0 (no se usa desde MSP432)
LVDS.EN de ambos chips	Entrada digital	MSP3 pin 16	7.4 (no se usa desde MSP432)
ELEC.TEST.CON de ambos chips	Entrada digital	MSP3 pin 15	7.5 (no se usa desde MSP432)

Tabla D.2: Conexión de los pines de CON_EST y MSP3.

Con la intención de medir retardos en el AFE y la calidad de los convertidores A/D del RHD2132 y del MSP432, se conecta el pin 8 del conector CON_EST al pin 2 del conector MSP3. En la placa de del MSP432, el pin 2 del conector MSP3 se

¹Se recuerda que los pines CS-, CLK-, MOSI-, MISO-, VESD, elec_test, LVDS_en y AUXOUT no serán utilizados en esta versión del sistema. Simplemente se permite que un microcontrolador distinto al MSP432 que permita implementar la comunicación SPI con LVDS pueda acceder a estos pines en un futuro.

Apéndice D. Conexión de cada pin del AFE

corresponde con uno de los conversores A/D que posee el MSP432. Además debe conectarse toda la zona debajo de cada chip RHD2132 a tierra. Esta recomendación viene dada en la hoja de datos del RHD2132 para lograr blindaje eléctrico (ver página 36 de [30]).

D.1. Conexión de los conversores DC / DC

El convertor ADP150 se utiliza para rectificar la tensión de los integrados RHD2132, mientras que el convertor TPS736 se utiliza para rectificar la tensión de las placas MCU y Radio (ver sección 2.4.2).

D.1.1. Conexión del ADP150

La figura D.2 resume cómo debe conectarse el convertor DC/DC ADP150.

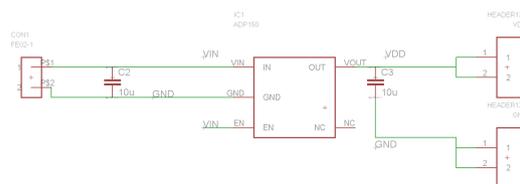


Figura D.2: Conexión del convertor DC/DC ADP150.

El siguiente punteo desarrolla lo mostrado en la figura D.2:

- VIN se conecta a la pila que suministra corriente al módulo remoto, a través de un conector en la placa AFE.
- VOUT será la tensión VDD (3,3V rectificado) que se conecta a los chips RHD2132.
- Entre VIN y GND se conecta un condensador de $10\mu F$ lo más cercano posible a VIN (ver página 6 de [57]).
- Entre VOUT y GND se conecta un condensador de $10\mu F$ lo más cercano posible a VOUT (ver página 6 de [57]).
- EN se conecta a VIN (ver página 6 de [57]).
- NC no se conecta (ver página 6 de [57]).

Para asegurar que no se necesita el uso de un disipador, se estimó la temperatura de juntura en el ADP150. Para ello, se utilizaron las siguientes ecuaciones brindadas en la hoja de datos por el fabricante [57]:

$$T_j = T_A + P_D \theta_{jA}$$
$$P_D = (V_{in} - V_{out}) I_{LOAD} + V_{in} I_{GND}$$
$$T_j < T_{jmax}$$

D.1. Conexión de los convertores DC / DC

donde $V_{in} = 3,7V$ y $V_{out} = 3,3V$. La corriente I_{LOAD} corresponde a la corriente de salida del convertor DC/DC que debido a que alimenta a los integrados RHD2132, la misma se estimó como $I_{LOAD} = 1,81mA$ (ver sección 2.4.1). La corriente de tierra se obtuvo de la página 7 de [57] y se estimó como $I_{GND} = 110\mu A$. Se consideró $T_A \approx 30^\circ C$. Con todos estos datos se obtuvo $P_D = 1,07mW$.

En la página 14 de [57] se indica que para el TSOT con ancho de cobre mínimo para la conexión del ADP150, $\theta_{jA} = 170^\circ C/W$ por lo que:

$$T_j = 30,2^\circ C$$

lejos del valor de temperatura de juntura máxima que es $T_{jmax} = 125^\circ C$.

D.1.2. Conexión del TPS736

La figura D.3 resume cómo debe conectarse el convertor DC/DC ADP150.

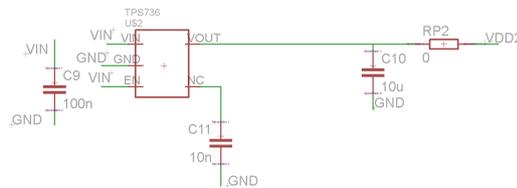


Figura D.3: Conexión del convertor DC/DC TPS736.

El siguiente punteo desarrolla lo mostrado en la figura D.3.

- VIN se conecta a la pila que suministra corriente al módulo remoto, a través de un conector en la placa AFE.
- VOUT será la tensión VDD (3,3V rectificado) que alimenta a las placas MCU y Radio de la figura 7.3.
- Entre VIN y GND se conecta un condensador de $100nF$ lo más cercano posible a VIN para mejorar la estabilidad contra efectos adversos de que la entrada esté lejos del convertor (ver página 16 de [58]).
- Entre NC y GND se conecta un condensador de $10nF$ lo más cercano posible a NC. Esto viene dado con la intención de disminuir el ruido a la salida. La gráfica de la figura 20 página 9 de [58] muestra que cuanto mayor sea este condensador, menor será el ruido a la salida.
- EN se conecta a VIN.
- Entre VOUT y GND se agrega un condensador de $10\mu F$ con el fin de mejorar la respuesta transitoria de la salida.

Para asegurar que no se necesita el uso de un disipador se estimó la temperatura de juntura en el TPS736. Para ello se utilizaron las siguientes ecuaciones brindadas en la hoja de datos por el fabricante:

Apéndice D. Conexión de cada pin del AFE

$$\begin{aligned}T_j &= T_A + P_D \theta_{jA} \\P_D &= (V_{in} - V_{out}) I_{LOAD} \\T_j &< T_{jmax}\end{aligned}$$

donde $V_{in} = 3,7V$ y $V_{out} = 3,3V$. La corriente I_{LOAD} corresponde a la corriente de salida del convertor DC/DC que debido a que alimenta a las placas MCU y Radio, la misma se estimó como $I_{LOAD} = 281mA$ (ver sección 2.4.2). Se consideró $T_A \approx 30^\circ C$. Con todos estos datos se obtuvo $P_D = 112,4mW$.

En la página 5 de [58] se indica que para el TSOT $\theta_{jA} = 221,9^\circ C/W$ por lo que:

$$T_j = 55^\circ C$$

lejos del valor de temperatura de juntura máxima que es $T_{jmax} = 125^\circ C$.

Finalmente cabe destacar que, para poder tener acceso a una medida de consumo, se agregó a la salida en los dos convertidores DC/DC resistencias de 0Ω .

D.2. Lista de componentes a soldar en la placa

- Conectores (todos con un pitch² de 2,54 mm):
 - 5 conectores macho de 2x1 (JP1, JP2, JP3, JP4 y JP7).
 - 2 conectores IDE macho con polaridad para board de 2x34 (CON1_ELEC y CON2_ELEC).
 - 2 conectores macho de 2x2 (MSP1 y MSP2).
 - 1 conector macho de 1x4 (JP3 y JP4).
 - 1 conector macho de 1x5 (JP5 y JP6).
 - 1 conector hembra de 1x2 (CON_BAT).
 - 1 conector macho de 2x4 (CON_EST).
 - 1 conector macho de 2x15 (MSP3).
- Componentes pasivos (todos SMD 0603):
 - Condensadores:
 - 3 de $10\mu F$.
 - 5 de $100nF$.
 - 3 de $10nF$.
 - Resistencias:
 - 2 de 0Ω .

²Pitch corresponde a la distancia entre pines de un conector.

D.3. Esquemático de la placa AFE

- 6 de 100Ω .
- 2 chips RHD2132.
- 1 conversor DC/DC ADP150.
- 1 conversor DC/DC TPS736.

En el anexo D.3 se presenta el esquemático de la placa para una mayor dete-
nimiento de las conexiones en la placa AFE.

D.3. Esquemático de la placa AFE

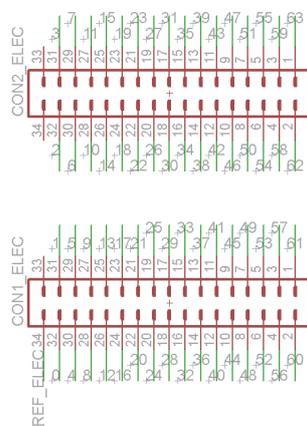


Figura D.4: Esquemático de los conectores CON1.ELEC y CON2.ELEC. Los números del 0 al 63 a los que se conecta corresponden a las señales de EEG a medir.

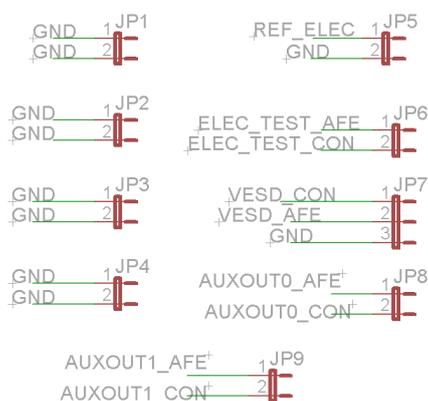


Figura D.5: Esquemático de los conectores JP1 a JP9.

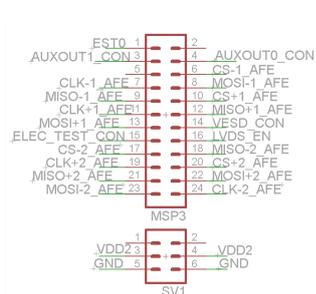


Figura D.6: Esquemático del conector MSP3.

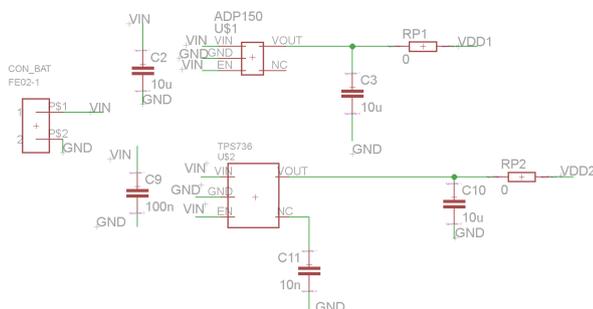


Figura D.7: Esquemático de los conectores JP1 a JP9.

Apéndice D. Conexión de cada pin del AFE

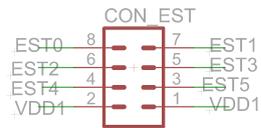


Figura D.8: Esquemático de los dos integrados RHD2132.

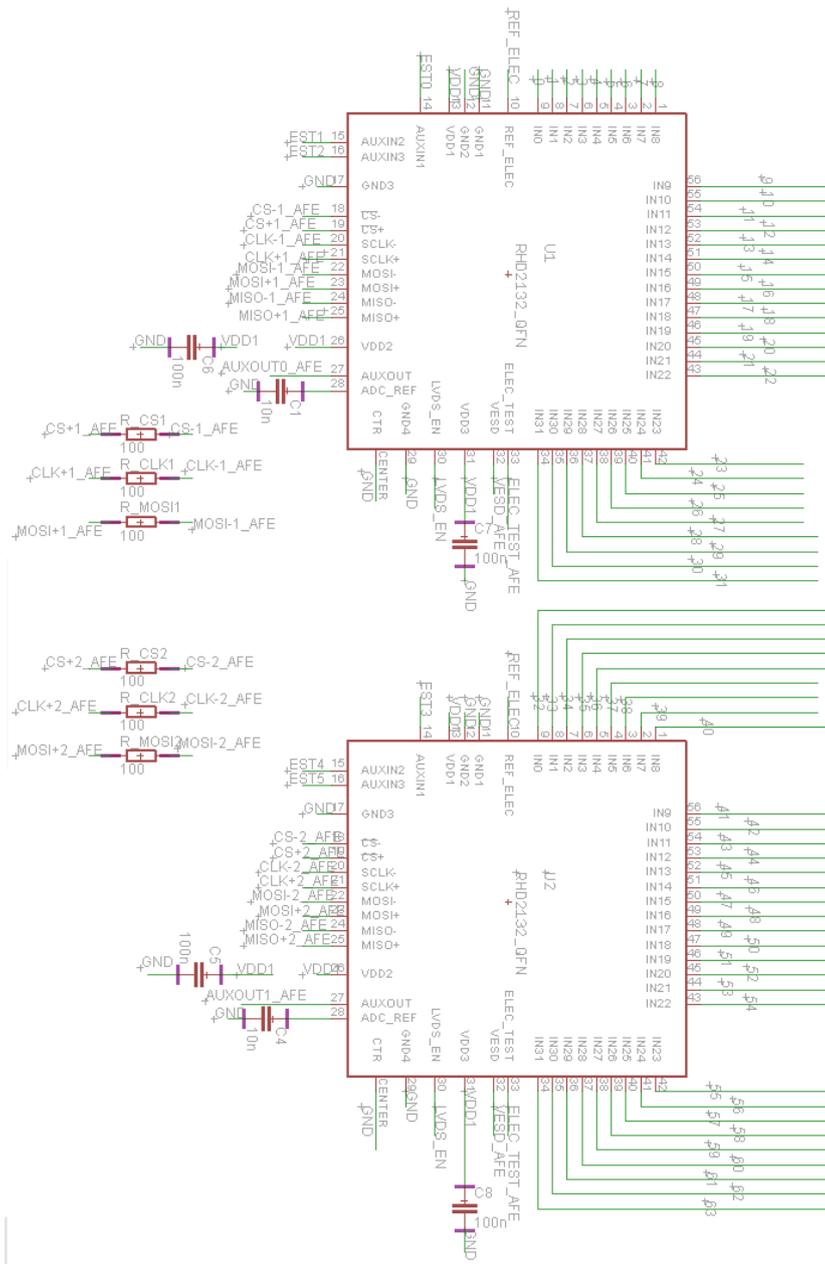


Figura D.9: Esquemático de los dos integrados RHD2132.

Apéndice E

Funcionamiento del test de impedancia de electrodos

Los integrados de la familia RHD2000, como el RHD2132 tienen la capacidad de medir la impedancia de los electrodos. Para ello, proveen un circuito adicional dentro del chip que genera una corriente AC. También es posible generar la corriente de forma externa utilizando el pin `elec_test` [30].

La figura E.1 muestra el esquemático detallado del circuito para la medida de impedancia de electrodos.

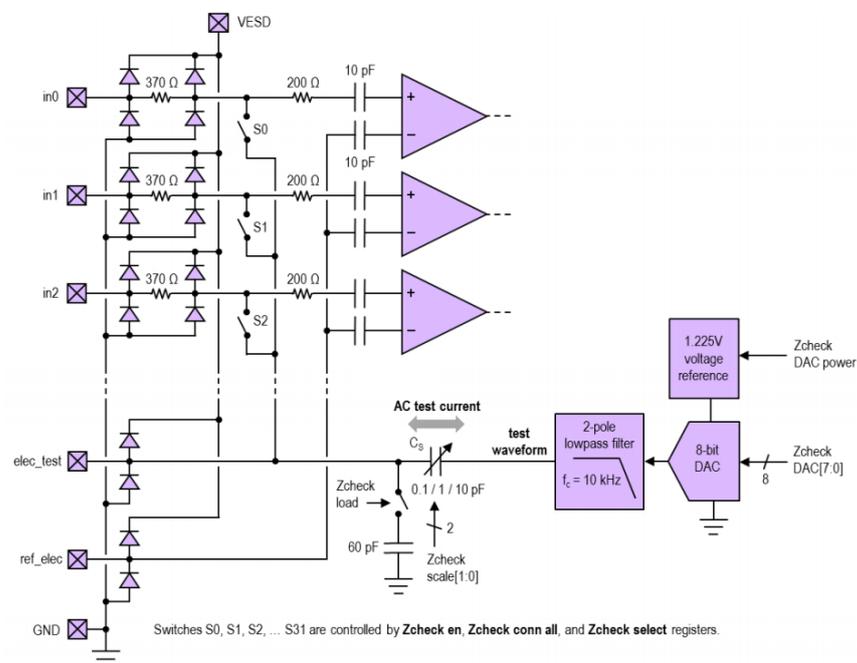


Figura E.1: Esquemático del circuito para la medida de impedancia de electrodos [30].

Apéndice E. Funcionamiento del test de impedancia de electrodos

Si se desea medir la impedancia del canal c , se debe cerrar la llave Sc . Esto se logra escribiendo en algunos registros del RHD2132 que se desea cerrar esa llave y que se desea medir la impedancia del canal c . Luego se genera la corriente con el conversor D/A que se observa en la figura E.1. Con esta corriente inyectada, muestreando el canal c con comandos CONVERT(c) se puede visualizar la forma de onda de voltaje inducida debido a la corriente que se aplica al electrodo. Como la corriente es conocida, se calcula la impedancia como el pico de voltaje dividido el pico de corriente. Esta técnica requiere de pequeñas corrientes puesto que los amplificadores tiene un rango de voltaje de entrada entre $\pm 5mV$.

Se destaca también que cualquier medición de impedancia incluirá la capacitancia de entrada de los amplificadores (10 pF), la de los diodos de protección ESD (0,4 pF), y aproximadamente 1,6 pF de capacidades parásitas asociadas con el adaptador de conexión y el paquete QFN [30].

E.1. Generación de la corriente AC

El conversor D/A del RHD2132 es capaz de generar un voltaje de continua entre 0V y 1,220V ($= \frac{255}{256} \times 1,225V$). Seteando el valor del registro **ZcheckDAC** con un valor α entre 0 y 255, se puede generar un voltaje continuo de " $\frac{\alpha}{256} \times 1,225V$ ". Variando el valor de este registro de manera periódica se puede generar una onda cuasi-sinusoidal como se muestra en la figura E.2.

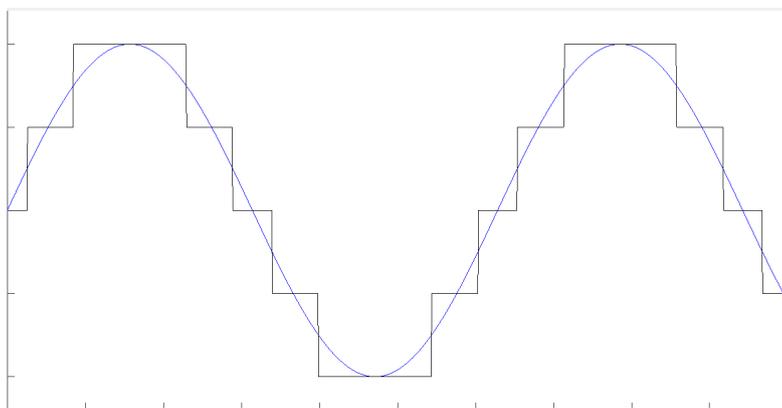


Figura E.2: Onda de voltaje generada variando el valor del registro ZcheckDAC.

Una vez generada esta forma de onda de voltaje, llamada $v_{DAC}(t)$, se aplica un filtro pasabajo para suavizar la señal (ver figura E.1) y luego, con el condensador C_S se convierte la forma de onda de voltaje $v_{DAC}(t)$ en una onda de corriente $i_{DAC}(t)$. Se cumple que:

$$i_{DAC}(t) = C_S \frac{dv_{DAC}(t)}{dt}$$

E.2. Elección de la corriente AC

Dado que se generará una señal de la forma $v_{DAC}(t) = V_A \sin(2\pi ft) + V_0$ se tiene que que:

$$i_{DAC}(t) = 2\pi f C_S V_A \cos(2\pi ft)$$

El valor de C_S se puede elegir entre tres valores posibles: $0,1pF$, $1pF$ y $10pF$ según se quiera ajustar [30].

E.2. Elección de la corriente AC

Se eligen los parámetros de $i_{DAC}(t)$ de manera de medir impedancias entre $Z_{min} = 10k\Omega$ y $Z_{max} = 2000k\Omega$. Se opta este rango debido a la tabla 2.8, en el cual se observa que los dos tipos de electrodos elegidos (DRYPAD SENSORS y FLEX SENSORS) tiene un valor de impedancia dentro del rango mencionado.

Es claro que el pico de corriente es $I_A = 2\pi f C_S V_A$. Evaluamos a continuación los rangos de variación de cada uno de los parámetros que componen a I_A .

Como se mencionaba en el anexo E.1, $0V \leq v_{DAC}(t) \leq 1,220V$ por lo que el mayor valor de V_A se encuentra tomando $V_0 = \frac{128}{256} \times 1,225V = 0,6125V$ y $V_A = 0,6077V$.

C_S puede tomar los valores $0,1pF$, $1pF$ y $10pF$.

f debe ser periódico por lo que se utiliza el timer_A0 (ver anexo G.4). El valor de f está acotado superior e inferiormente por limitaciones en el microcontrolador. Según lo indicado en el anexo G.4, el valor hasta el que debe contar el timer es $N = \frac{6MHz}{f}$. Este valor se guarda en un registro de 16 bits por lo que $N < 2^{16}$ lo que lleva a que $f > \frac{6MHz}{2^{16}} = 91,56Hz$. Por otro lado, para medir el test de impedancia de electrodos, es necesario enviar dos palabras de 16 bits a los integrados RHD2132 cada interrupción del timer A0, uno para modificar el registro ZcheckDAC y otro para mostrar el canal c . Según se midió, el envío de dos palabras es aproximadamente $31\mu s$. Agregando unos $9\mu s$ extras para operaciones del microcontrolador sobre el nuevo valor del registro ZcheckDAC, se obtiene que $\frac{1}{f} > 40\mu s$ y por tanto $f < 25kHz$. En resumen, $92Hz \leq f \leq 25kHz$.

El valor de pico del corriente máximo se obtiene con Z_{max} y está acotado por el rango de los amplificadores, por lo que, para abarcar todo el rango se elige que $I_A Z_{max} = 5mV$, esto lleva a que $I_A = 2\pi f C_S V_A = 2,5nA$.

Para que el timer_A0 interrumpa lo menos posible, se decide trabajar con un valor bajo de frecuencia f , lo que implica trabajar con C_S y V_A máximos. Se toma entonces $C_S = 10pF$ y $V_A = 0,6077V$, de manera que $f = 65,47Hz$. Como este

Apéndice E. Funcionamiento del test de impedancia de electrodos

valor no está dentro del rango se elige $C_S = 1pF$ y entonces $f = 654Hz$.

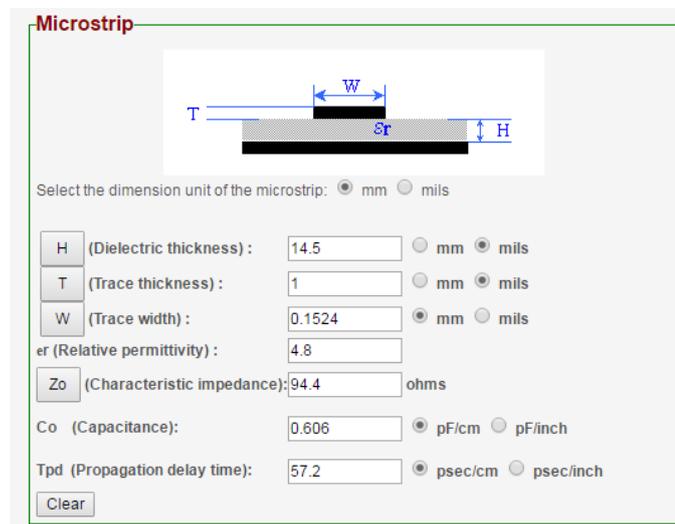
En resumen se trabaja con $V_O = 0,6125V$, $V_A = 0,6077V$, $C_S = 1pF$ y $f = 654Hz$.

Apéndice F

Impedancia característica de las pistas

Una vez conocido el fabricante y que se utiliza un modelo de 4 capas como fue indicado en la sección 3, se utilizó una calculadora de impedancias [59] con el modelo de microstrip (ver figura F.1).

El material utilizado por el fabricante de PCB es FR-4 que tiene un $\epsilon_r = 4,8$ [60]. El ancho de pista utilizado fue $W = 0,1524mm$, el espesor de cobre $T = 1mils$, y el espesor del FR-4 es $H = 14,5mils$. La impedancia característica obtenida es $Z_O = 94,4\Omega$, por lo que, las resistencias que se sueldan serán de 100Ω .



The image shows a software interface titled "Microstrip" for calculating characteristic impedance. At the top, there is a diagram of a microstrip on a dielectric substrate with labels for width (W), thickness (T), and dielectric thickness (H), and a note for relative permittivity (ϵ_r). Below the diagram, there are several input fields and radio buttons for unit selection. The calculated values are displayed next to the input fields.

Parameter	Value	Unit
Select the dimension unit of the microstrip:	<input checked="" type="radio"/> mm <input type="radio"/> mils	
H (Dielectric thickness)	14.5	<input type="radio"/> mm <input checked="" type="radio"/> mils
T (Trace thickness)	1	<input type="radio"/> mm <input checked="" type="radio"/> mils
W (Trace width)	0.1524	<input checked="" type="radio"/> mm <input type="radio"/> mils
ϵ_r (Relative permittivity)	4.8	
Z_o (Characteristic impedance)	94.4	ohms
Co (Capacitance)	0.606	<input checked="" type="radio"/> pF/cm <input type="radio"/> pF/inch
Tpd (Propagation delay time)	57.2	<input checked="" type="radio"/> psec/cm <input type="radio"/> psec/inch

Clear

Figura F.1: Calculadora de impedancia característica.

Como muestra la figura F.1, la calculadora de impedancia también da valor de la capacidad por unidad de longitud de cada pista.

Esta página ha sido intencionalmente dejada en blanco.

Apéndice G

Información complementaria del diseño del software embebido

G.1. Elección de Round-Robin frente a encolado de funciones

Inicialmente se optó por trabajar con encolado de funciones por las siguientes razones:

- Permite definir prioridades en las tareas a realizar.
- Programa principal mas simple e intuitivo.

Sin embargo, en el momento de utilizarlo, surgieron problemas que generaban conflictos en el MSP432:

Velocidad: el tiempo de encolar una función a una cola circular, y luego desencolarla, era mayor que el tiempo de setear banderas y revisarlas en el programa principal.

Fallas del MSP432: desencolar una función de una cola en el programa principal, y que a su vez una interrupción encole una función en la misma cola, generaba fallas internas que llevaban al MSP432 a un estado de *fault*. Se estudió, pero no se logró determinar la causa de este problema.

Encolar varias veces una misma función: funciones con poca prioridad, como ver si hay un nuevo comando proveniente del PC se encolan y pasa mucho tiempo antes de que sean atendidas. Dado que el timer_A1 interrumpe periódicamente, esto implica que se encola muchas veces la misma función, llenando esa prioridad de la cola. Este inconveniente saturaba en algún momento al módulo remoto, independiente del tamaño de la cola que se creó.

Apéndice G. Información complementaria del diseño del software embebido

Si bien en la teoría, utilizar encolado de funciones tenía mas ventajas que Round-Robin, otro problemas que ocurrieron en la práctica terminaron descartando la posibilidad¹.

G.2. Diagramas de flujo de las principales funciones de SPI.h y AFE.h

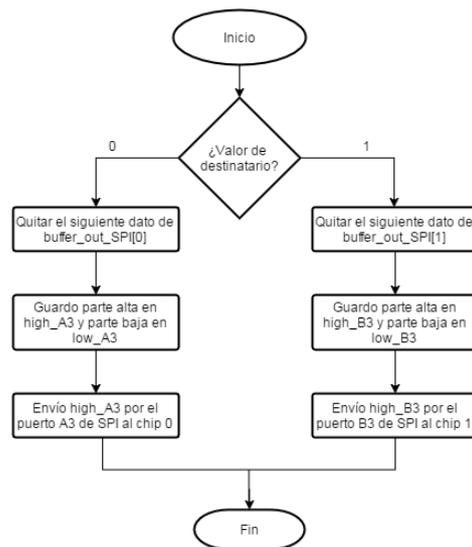


Figura G.1: Diagrama de flujo de la función send_word(destinatario).

¹Para mayor información de las arquitecturas de Round-Robin y encolado de funciones visitar [61].

G.2. Diagramas de flujo de las principales funciones de SPI.h y AFE.h

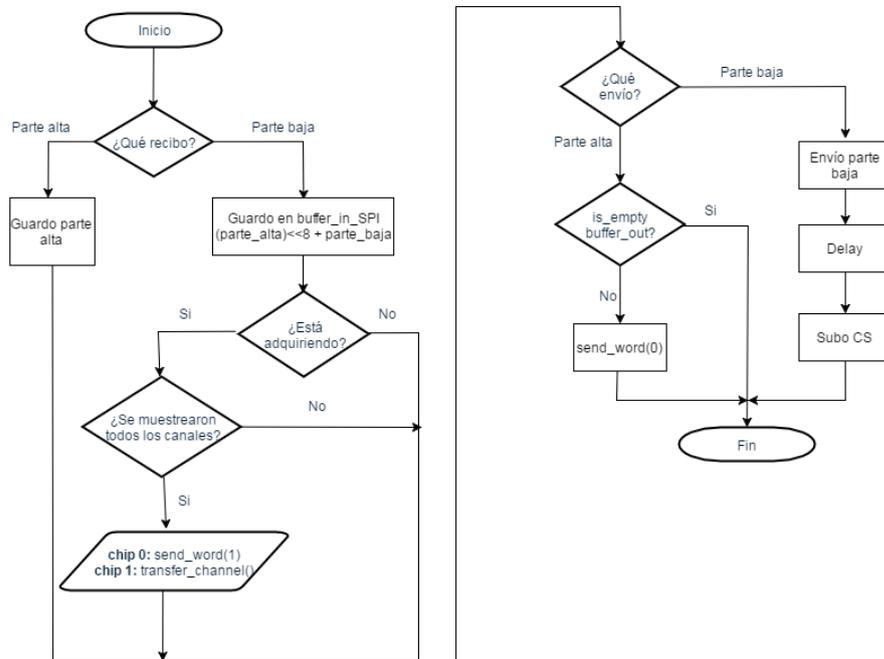
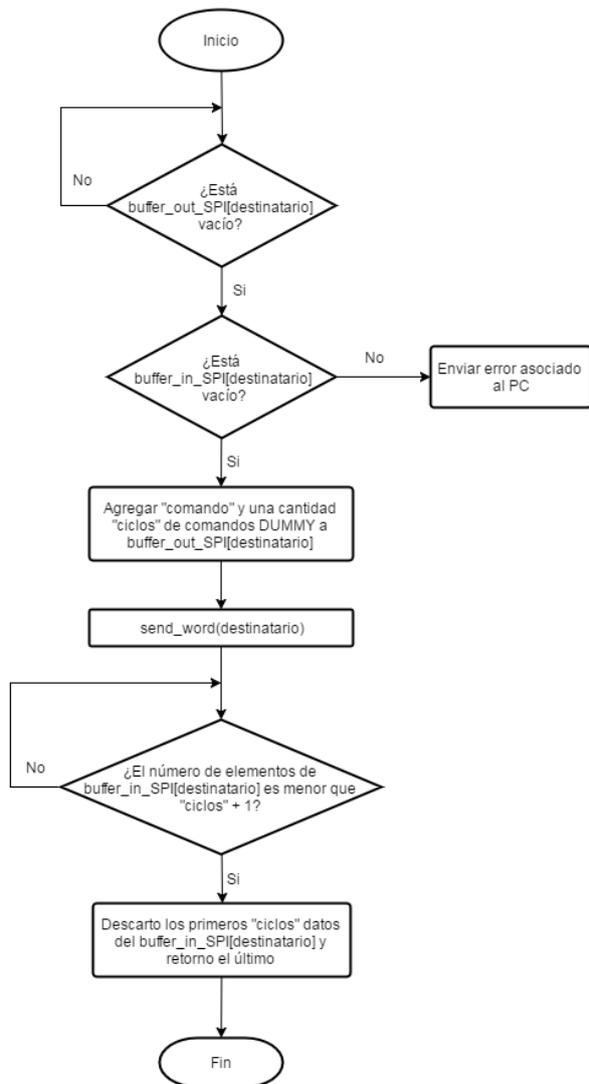
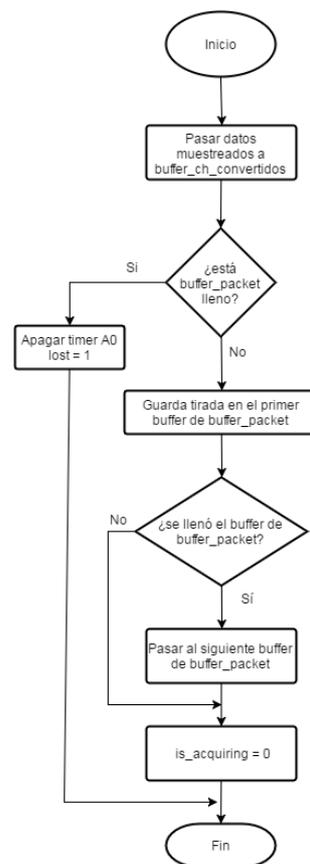


Figura G.2: Diagrama de flujo de las rutinas de interrupción de los puertos SPI que se comunican con el AFE. El paralelogramo indica que la acción en el chip 0 y el chip 1 es distinta según el caso.

Apéndice G. Información complementaria del diseño del software embebido



(a) Diagrama de flujo de la función send_and_wait.



(b) Diagrama de flujo de transfer_channel().

Figura G.3: Diagramas de flujo de CONVERTburst() y transfer_channel().

G.2. Diagramas de flujo de las principales funciones de SPI.h y AFE.h



Figura G.4: Diagrama de flujo de la función CONVERTburst().

G.3. Módulo queue

Considerando que se quiere generar una nueva cola con varias prioridades distintas y que se puedan almacenar una cantidad arbitraria de elementos de cada prioridad distinta, se crea una matriz como la de la figura G.5 que tiene *TAMCOLAS* filas y *CANTPRIO* columnas. Por comodidad se busca además que todas las colas sean circulares. Para ello es necesario tener un par de índices por cada columna, uno que indique cual es el próximo elemento a sacar de la columna (*ind_out*) y otro que indique cual es el siguiente lugar para guardar un dato (*ind_in*).

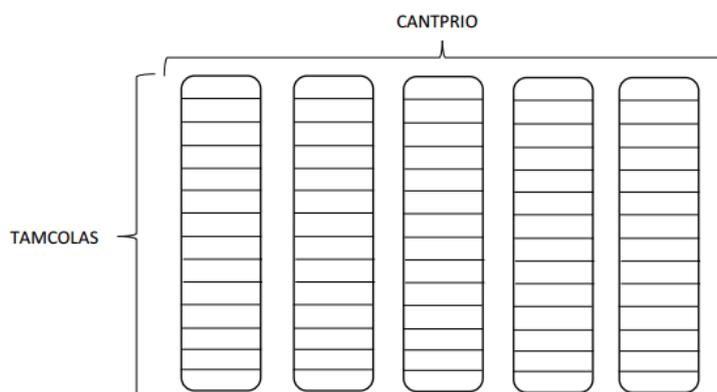


Figura G.5: Estructura de una cola circular con *CANTPRIO* prioridades de tamaño *TAMCOLAS*.

El módulo queue, se encarga de controlar este tipo de colas circulares con prioridades para cualquier tipo de dato. Los módulos que necesiten utilizar buffers o colas circulares podrán instanciar un objeto de tipo especial que se define como *obj.t*. Como se muestra en la figura G.6, las variables de tipo *obj.t* son estructuras de tres tipos de datos. La constante *TAMCOLAS* indica el tamaño que tiene cada prioridad (todas las prioridades tienen el mismo tamaño), la constante *CANTPRIO* indica la cantidades de prioridades distintas que pueden manejarse (ver figura G.5) y *punt_array_ind* tendrá la información de los pares de índices de entrada y de salida de las *CANTPRIO* prioridades que se desean controlar.

La variable *punt_array_ind* es un puntero a un arreglo de *CANTPRIO* elementos, y cada elemento es una estructura cuyo tipo de dato se define como *queue.t*. Cada estructura posee los índices *ind_in* e *ind_out* que son los pares de índices mencionados anteriormente, que controlan a una determinada prioridad. El valor de estos índices es un valor entero entre -1 y *TAMCOLAS* -1. Si el valor que toma es mayor o igual a 0, su valor hace referencia a uno de los lugares de la prioridad a la que controla. La manera de saber si la prioridad se llenó o está vacía se utiliza con el valor especial -1. Si *ind_in* = -1 esto significa que la prioridad asociada a ese índice está llena. Si *ind_out* = -1 esto significa que la prioridad asociada a ese índice está vacía.

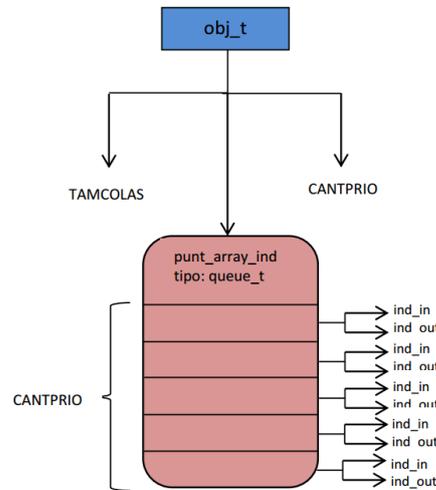


Figura G.6: Estructura de un tipo de dato `obj_t` que se encarga de controlar la cola circular de la figura G.5.

Se puede crear la cantidad de colas con prioridades que se desee y que todas sean independientes. Cada cola creada debe tener su propia estructura de control de tipo `obj_t`. Por lo tanto, crear una cola circular con prioridades para ser manejada con las funciones de este módulo implica la declaración previa de tres variables:

- Una matriz `cola` de tamaño $TAMCOLAS \times CANTPRIO$, donde guardar los datos.
- Un arreglo `array_ind` de tamaño `CANTPRIO` y tipo de dato `queue_t` que permita almacenar los pares de índices de control para cada prioridad.
- Un objeto `object` de tipo de dato `obj_t` que compacte toda la información de control sobre la cola creada.

Se tendrá por tanto `CANTPRIO` prioridades distintas. La prioridad más alta será 0 y la prioridad más baja será `CANTPRIO - 1`. Si lo que desea es simplemente tener un buffer, basta con considerar el caso `CANTPRIO = 1`. En algunas ocasiones, puede ser útil tener varios buffers del mismo tamaño. Para ello, se puede considerar `CANTPRIO` mayor a 1 de manera de englobar a todos los buffers en una misma cola. En estos casos, no se desea respetar las reglas de prioridades, es decir, no siempre el elemento a sacar corresponde al primero de mayor prioridad, sino que se desea elegir de que buffer sacarlo. El módulo `queue` también provee funciones que permiten sacar datos sin respetar la ley de prioridades.

Las funciones de este módulo se detallan a continuación:

```
obj_t object_create(unsigned char cantprio, unsigned char tamcolas,
queue_t* array_ind)
```

Apéndice G. Información complementaria del diseño del software embebido

Crea una estructura de control para una cola de tamaño $TAMCOLAS \times CANTPRIO$.

unsigned char get_numberPrio(obj_t object)

Indica cuantas prioridades distintas tiene la cola asociada al objeto *object*.

unsigned char get_Tamcolas(obj_t object)

Indica el tamaño de cada prioridad en la cola asociada al objeto *object*.

unsigned char is_Full(obj_t object, unsigned char prio)

Indica si la prioridad *prio* está llena.

unsigned char is_emptycprio(obj_t object)

Indica si la cola está vacía. Será 1 si todas las prioridades están vacías.

unsigned char is_emptyprio(obj_t cola, unsigned char prio)

Indica si la prioridad *prio* está vacía.

char add_data(obj_t object, unsigned char prio)

Para agregar un dato a una cola de manera de hacerlo transparente al tipo de dato que se utiliza, se optó por realizar una función que dada la prioridad a la que se desea agregar un dato, devuelva el índice al cual se debe guardar, y que además modifique el índice *ind_in* asumiendo que se agrega un dato. Agregar un dato utilizando la siguiente función se hace de la siguiente manera:

```
cola[prio][add_data(object, prio)] = dato;
```

El módulo que crea la cola conoce el tipo de dato que maneja y puede realizar la instrucción anterior sin inconvenientes.

void get_data_cprio(obj_t cola, int* indices_salida)

Carga en *indices_salida* la posición en cola del próximo dato a sacar respetando la ley de prioridades, y actualiza automáticamente los índices asumiendo que el dato fue quitado.

Para quitar un dato de manera de hacerlo transparente al tipo de dato que se utiliza, se opta por devolver los índices donde se encuentra el próximo a dato a sacar, y que el módulo que creó la cola (que conoce el tipo de dato que maneja) se encargue de quitarlo. La instrucción que permite quitar un dato utilizando esta función es:

```
get_data_cprio(object, indices_salida);  
saco = cola[indices_salida[0]][indices_salida[1]];
```

char get_data_sprio(obj_t cola, unsigned char prio)

Retorna la posición del elemento a sacar y actualiza automáticamente los índices asumiendo que el dato fue quitado.

Esta función se utiliza en los casos que se quiere sacar un dato sin respetar la ley de prioridades, por ejemplo, cuando se trabaja con varios buffers. Ejemplo de uso:

```
saco = matrix[prio][get_data_sprio(object, prio)];
```

unsigned int number_elements(obj_t object, unsigned char prio)

Indica la cantidad de elementos que tiene esa prioridad.

void clear_prioobject(obj_t object, unsigned char prio)

Reinicia los índices *ind_in* e *ind_out* de la prioridad *prio*. Esto es equivalente a borrar los datos de esa prioridad.

void clear_object(obj_t object)

Reinicia los índices *ind_in* e *ind_out* de todas las prioridades. Esto es equivalente a borrar todos los datos de la cola asociada a *object*.

void update_index(obj_t object, unsigned char prio, int8_t ind_in, int8_t ind_out)

Existen casos donde para mejorar la velocidad, se desea agregar o quitar datos a una determinada prioridad sin utilizar las funciones del módulo. Esto está permitido aunque es necesario luego actualizar los valores de los índices *ind_in* e *ind_out* de forma manual para que la estructura de control puede continuar con su labor. Esta función permite modificar manualmente el valor de los índices en caso que se crea necesario. Cabe destacar que cargar datos a una determinada prioridad sin el uso de las funciones de este módulo puede afectar el funcionamiento general si no se hace con las precauciones necesarias.

G.4. Módulo Timer

Este módulo controla los dos timers del proyecto mencionados en la sección 4.3.

G.4.1. Timer A0

Este timer interrumpe cada período f_s indicando que debe muestrearse una nueva tirada. Las interrupciones de este timer solo están encendidas si se está realizando una adquisición.

Apéndice G. Información complementaria del diseño del software embebido

Se configura el reloj en up mode ² a una frecuencia de 6 MHz.

Se proveen dos funciones para este timer:

void set_fs(int f_s)

Setea la frecuencia de interrupción del timer. Se calcula el valor hasta el que debe contar el timer como $\frac{6MHz}{f_s}$.

int get_fs()

Retorna el valor actual de f_s , obteniendo la información directamente del registro del MSP432.

G.4.2. Timer A1

Se utiliza para preguntar periódicamente si se recibió un mensaje proveniente del PC. Su rutina de interrupción solamente setea la bandera *commandCheckFlag* en caso de que no haya sido seteada.

G.5. Módulo Radio

Este módulo se encarga del envío y la recepción de los datos desde el PC. Para controlar a la radio CC3100 desde el MSP432 se utiliza la librería SimpleLink ³. Este módulo implementa el protocolo de comunicación a nivel TCP/IP con la PC. A continuación se explica el funcionamiento de las cuatro funciones de este módulo.

G.5.1. Radio_init()

Esta función inicializa la configuración de la radio, creando un servidor TCP.

En esta función se realizan las siguientes tareas:

- Configura Simplelink al estado por defecto (con la función *configureSimpleLinkToDefaultState()*).
- Establece una comunicación con el punto de Acceso (con la función *establishConnectionWithAP()*).
- Abre un puerto TCP en modo servidor (con la función *BsdTcpServer(PORT_NUM)*).

²El modo up mode implica que el timer cuenta hasta un valor elegido y reinicia su conteo automáticamente. Variando el valor al que cuenta el timer se puede setear el período del timer.

³Simplelink es una librería de Texas Instrument que provee funciones a los microcontroladores MSP43x para controlar a los integrados CC31xx / CC2xx, encargados de la comunicación Wi-Fi. Para mayor información de la librería SimpleLink ver [31].

Esta última función abre un socket⁴ TCP en el modo Listen, bloqueando el sistema hasta que desde el PC, el usuario se conecte al sistema, estableciendo así la conexión TCP.

La configuración del estado por defecto de Simplelink incluye:

- Seteo en modo ESTACIÓN.
- Habilitación de DHCP⁵.
- Seteo la potencia de transmisión al máximo.
- Remoción de todos los filtros.

La figura G.7 muestra el diagrama de la inicialización de la radio.

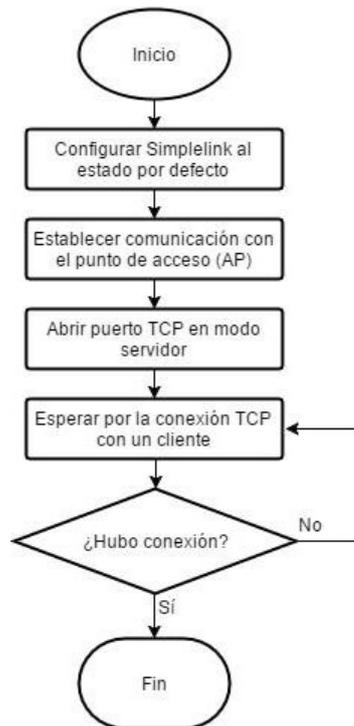


Figura G.7: Diagrama de flujo de la función *radio_init()*.

⁴Socket refiere al puerto por el cual dos sistemas pueden intercambiar cualquier flujo de datos de manera fiable y ordenada [62].

⁵DHCP, cuyas siglas en español significan “protocolo de configuración dinámica de host”, es un servidor que usa un protocolo de red de tipo cliente/servidor en el cual el servidor posee una lista de direcciones IP y las va asignando a los clientes a medida que éstas van quedando libres [63].

G.5.2. Radio_receiveData

Esta función consulta por el estado del buffer de recepción de la radio CC3100 retornando la cantidad de bytes en este buffer y vaciando su contenido en la variable *dataReceive* en caso de que se hayan recibido datos. Esta función es utilizada en el programa principal para consultar si hay nuevos comandos para procesar. El diagrama de la figura G.8 muestra el funcionamiento de esta función.

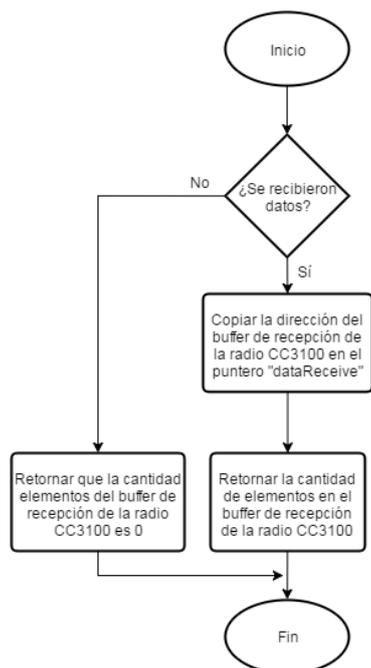


Figura G.8: Diagrama de flujo de la función *radio_receiveData*.

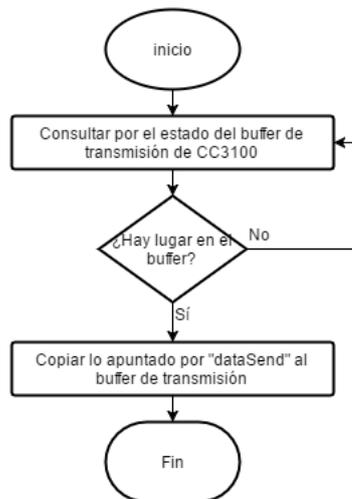
G.5.3. Radio_sendData

Esta función copia lo apuntado por *dataSend* de largo *lengthDataSend* en el buffer de transmisión de la radio CC3100 y retorna la cantidad de bytes que copió. En caso de éxito, este número debe de coincidir con *lengthDataSend*.

El loop que se observa en el diagrama G.9 se realiza para garantizar la transmisión exitosa del paquete a la radio, esto se hace con el fin de cumplir con el protocolo TCP/IP, ya que este protocolo asegura la llegada del paquete a destino.

G.5.4. Radio_close

Esta función se encarga de cerrar el puerto de comunicación. Se utiliza en el módulo *commandCTRL* cuando desde la interfaz se envía un comando SALIR (ver sección 4.8).

Figura G.9: Diagrama de flujo de la función *radio_sendData()*.

G.6. Módulo CTRLerrores

Este módulo se utiliza para notificar al usuario a través del PC que ocurrió un evento inesperado en el módulo remoto. Los mensajes de error son:

Warning: Se intentó ingresar a un buffer no válido

Este error ocurre en el módulo que se ejecuta cuando se pretende entrar a un conjunto de buffers, pero el mismo no se encuentra dentro del rango. Su ocurrencia no depende del usuario, puesto que el ingreso a los buffers es transparente desde el PC. En este caso, el microcontrolador se resetea, lo que ocasiona una inicialización de todo el módulo remoto.

Warning: El buffer que se intenta llenar alcanzó su límite

Este error ocurre en el módulo que se ejecuta cuando se intenta agregar un dato en un buffer sin revisar previamente si el mismo está lleno. Su ocurrencia se debe a un mal manejo del software embebido y no proviene de un mal manejo del PC. En este caso, el microcontrolador se resetea, lo que ocasiona una inicialización de todo el módulo remoto.

Warning: Se intenta usar la función `send_and_wait` pero el buffer de entrada de SPI no está vacío

Este error ocurre en el módulo SPI. Para utilizar la función *send_and_wait* es necesario que el *buffer_in_SPI[destinatario]* esté vacío; de lo contrario, cuando se descarta las primeras *ciclos* respuestas, el siguiente no va a ser el dato esperado (ver diagrama de flujo de esta función en la figura G.3a). Este error no depende del usuario, puesto que el uso de esta función es transparente desde el PC. En este caso, el microcontrolador se resetea, lo que ocasiona una inicialización de todo el módulo remoto.

Warning: La frecuencia de corte superior de los dos chips son distintas o Warning: La frecuencia de corte inferior de los dos chips son distintas

Algunos de estos errores ocurre en el módulo AFE (en las funciones `get_sup_cutoff_freq()` o `get_inf_cutoff_freq()`) y puede deberse a que los integrados RHD2132 no quedaron igual configurados por algún error anterior, o que alguno de los RHD2132 no está contestando de manera correcta, porque la batería del módulo remoto está agotada o porque se dañó alguno de los chips. En este caso el módulo remoto continúa su funcionamiento y devuelve como respuesta el valor de la frecuencia de corte del chip 0.

Warning: La frecuencia de corte superior que se desea escribir no esta en el rango o Warning: La frecuencia de corte inferior que se desea escribir no esta en el rango

Algunos de estos errores ocurre en el módulo AFE cuando se intenta configurar a los integrados con una frecuencia de corte superior (o inferior) que los RHD2132 no admiten. Este error depende de la configuración enviada desde la interfaz de usuario. Si ocurre, el módulo remoto continúa su funcionamiento sin configurar la frecuencia de corte en la que ocurrió el error.

Warning: La configuracion de canales no es valida

Este error ocurre en el módulo AFE (en las funciones `CONVERTburst` y `offAmps`) cuando se intenta iniciar una adquisición con una configuración de canales no válida. Este error depende del comando enviado desde la interfaz de usuario. Si ocurre, el módulo remoto continúa su funcionamiento sin enviar ningún comando a los integrados RHD2132. Esto implica que por ejemplo, todas las adquisiciones serán vacías.

Warning: El testeo de impedancia de electrodos se desea realizar en un canal fuera del rango

Este error ocurre en el módulo AFE cuando se pretende realizar el test de impedancia en un canal que no está entre 0 y 31. Este error depende del mensaje enviado desde la interfaz de usuario. Si ocurre, se cancela el test de impedancia de electrodos.

Warning: Mensaje inválido

Este error ocurre en el módulo `CTRLCommand` cuando el comando enviado no se corresponde con ninguno de los esperados. Este error depende del mensaje enviado desde la interfaz de usuario. Si ocurre, se descarta el mensaje inválido.

Warning: Se envió un START con el sistema adquiriendo

Este error ocurre en `CTRLCommand` cuando se envía un `START` con el sistema adquiriendo. Este error depende del usuario. Si ocurre, se descarta el comando `START` recibido y se continúa con la adquisición del sistema.

Warning: Se envió un STOP con el sistema detenido

Este error ocurre en CTRLCommand cuando se envía un STOP con el sistema adquiriendo. Este error depende del usuario. Si ocurre, se descarta el comando STOP recibido.

G.7. Cálculo del tamaño de paquete a enviar

El tamaño del paquete a enviar depende de la cantidad de canales y señales de sincronismo que se utilice. Este nunca supera el tamaño máximo de paquete (1460 bytes). La ecuación que permite calcular el tamaño de paquete es:

$$tam_{paquete}(bytes) = tam_{maxpaquete} - resto\left(\frac{tam_{maxpaquete} - tam_{encabezado}}{tam_{tirada}}\right) \quad (G.1)$$

En la ecuación G.1 el tamaño de paquete efectivo viene expresado en bytes, $tam_{maxpaquete}$ indica el tamaño máximo de paquete (1460 bytes) y la función $resto(a/b)$ da como resultado el resto de la división entera entre a y b . $tam_{encabezado} = 2$ y es la cantidad de bytes que ocupan el identificador y el número de paquete. El tamaño de cada tirada se calcula como:

$$tam_{tirada}(bytes) = 2(N_c + N_s) \quad (G.2)$$

En la ecuación G.2, N_c indica la cantidad de canales y N_s la cantidad de señales de sincronismo que se adquieren, ambos elegidos por el usuario (desde el PC), como cada muestra adquirida se representa con 2 bytes, se debe de multiplicar por 2 para su conversión a bytes. Por ejemplo, si se utiliza 64 canales y ninguna señal de sincronismo, el tamaño del paquete será 1410, puesto que si se intenta agregar otra tirada, se supera el tamaño de 1460. La cantidad de tiradas que entran en este caso es 11.

La cantidad de tiradas ($N_{tiradas}$) que entran en un paquete se puede calcular en función de lo mencionado anteriormente. $N_{tiradas}$ es el máximo entero positivo que cumple:

$$2 + 2 \times (N_c + N_s) \times N_{tiradas} < 1460$$

Despejando se obtiene:

$$N_{tiradas} = \lfloor \frac{729}{N_c + N_s} \rfloor \quad (G.3)$$

donde $\lfloor x \rfloor$ representa a la función piso (función trunc).

En el ejemplo de 64 canales sin sincronismo, $N_{tiradas} = \lfloor \frac{729}{64} \rfloor = 11$.

Apéndice H

Información complementaria a la interfaz de usuario

H.1. Menús interfaz de usuario

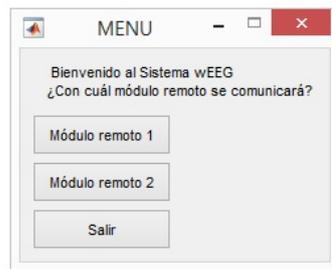


Figura H.1: Menú para indicar al PC con que módulo remoto se debe iniciar la comunicación.

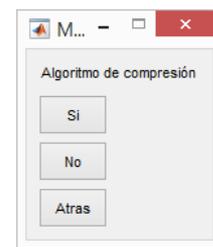


Figura H.2: Menú para setear el uso del algoritmo de compresión.

H.2. Formato de los comandos enviados al módulo remoto

Para que el módulo remoto interprete los comandos de la tabla 6.1, es necesario definir el formato de los mismos. En todos los casos “ \emptyset ” es un espacio, caracter 32 en ASCII.

START: el PC envía un comando con el formato “START \emptyset x \emptyset y”, donde x es 1, 4, 21 ó 64 en función de la configuración de canales e “y” indica con un 1 ó 0 si se utiliza eliminación de offset. Este comando no tiene respuesta.

STOP: se envía el comando “STOP”. Este comando tiene respuesta y es “STOP \emptyset OK”.

Apéndice H. Información complementaria a la interfaz de usuario



Figura H.3: Menú para indicar la configuración de canales para la próxima adquisición a realizar.

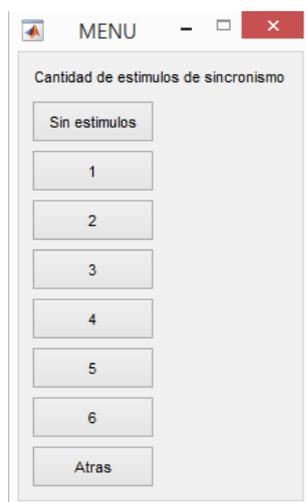


Figura H.4: Menú para indicar cuantas señales de sincronismo se desean muestrear en la próxima adquisición a realizar.

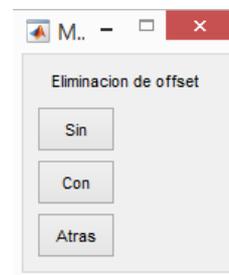


Figura H.5: Menú para elegir el uso de la eliminación de offset.

ONE_CHANNEL: el PC envía el comando con el formato “ONE \emptyset x \emptyset y” donde x es un número entero positivo entre 1 y 64 que indica que canal se quiere adquirir e “y” indica con un 1 ó 0 si se utiliza eliminación de offset. Este comando no tiene respuesta.

CONFIG: el PC envía el comando con el formato “C \emptyset a \emptyset b \emptyset c \emptyset d \emptyset e \emptyset f” donde “a” es un entero positivo que indica el valor de la frecuencia de muestreo, “b” es un entero positivo que indica el valor de la frecuencia de corte superior, “c” es un entero positivo que equivale a cien veces el valor de frecuencia de corte inferior, “d” es un número entero entre 0 y 6 que indica cuantos canales de sincronismo se desean utilizar, “e” puede valer 1 o 0 e indica si se quiere activar el algoritmo ACEEG o no y “f” es un valor entero entre 0 y 16 e indica el valor del parámetro de distorsión. Este comando no tiene respuesta.

EA: se envía el comando “EA”. Este comando tiene respuesta y es “EA \emptyset a \emptyset b \emptyset c \emptyset d \emptyset e \emptyset f” donde “a”, “b”, “c”, “d”, “e”, y “f” se entregan en el mismo orden que cuando el módulo remoto recibe un comando CONFIG.

CALIBRATE: se envía el comando “CAL”. Este comando tiene respuesta y es “CAL \emptyset OK”.

TIE: se envía desde el PC al módulo remoto el comando “TIE \emptyset x” donde x es un número entre 1 y 32 que indica que va a realizarse el test de impedancia de electrodos en los canales “x” y “x + 32”. Este comando no tiene respuesta.

H.3. Recomendaciones ante fallas del sistema



Figura H.6: Menú para elegir la frecuencia de corte superior.



Figura H.7: Menú para elegir la frecuencia de corte inferior.



Figura H.8: Menú para elegir el valor de distorsión para el algoritmo de compresión ACEEG.

RESET: se envía el comando “RESET”. Este comando no tiene respuesta.

H.3. Recomendaciones ante fallas del sistema

A continuación se consideran algunas posibles fallas y como resolverlas:

El PC no se conecta al módulo remoto

- Verifique que el router esté encendido.
- Verifique que el PC está conectado a la red “wEEG”.
- Verifique que la radio elegida en la interfaz de usuario sea la que está conectada al módulo remoto.
- Revise que el interruptor en la carcasa esté encendido.

Apéndice H. Información complementaria a la interfaz de usuario

- Revise alimentación del módulo remoto y carga de la batería.
- Reinicie el módulo remoto y vuelva a iniciar la interfaz de usuario.

Los comandos CONFIG y EA no son coherentes

- Puede ocurrir que no se conectó el PC con el módulo remoto de forma adecuada. Si la última vez que se utilizó el módulo remoto no fue reiniciado, la conexión entre el PC y el módulo remoto no se realiza. En este caso, reinicie el módulo remoto a través del interruptor en la carcasa y reinicie la comunicación.
- Debido al daño de uno de los integrados RHD2132 (chip 1), la respuesta proveniente del mismo no es la esperada algunas veces. Esto lleva a errores cuando el sistema embebido compara la configuración de los dos integrados RHD2132 (ver sección G.6). Este error repetido lleva la falla de que la configuración solicitada no pudo verificarse. En este caso, reinicie el módulo remoto a través del interruptor y si el problema persiste, comuníquese con los fabricantes del sistema.
- Revise que el interruptor en la carcasa esté encendido.
- La batería que alimenta al módulo remoto está por debajo del voltaje mínimo necesario para su funcionamiento. En este caso, cambie la batería e intente conectarse de nuevo al sistema.

La señal que se recibe tiene pérdidas de forma constante

- Ocurre si se configura al sistema a una frecuencia de muestreo muy rápida para la cantidad de canales que se quiere adquirir. En este caso, detenga la adquisición actual y reconfigure la frecuencia de muestreo o la cantidad de canales. Para más información sobre la frecuencia máxima para cada configuración de canales, ver la sección 8.3.1.
- No está conectado el router al PC. Esto puede reducir significativamente la frecuencia máxima de muestreo en cada configuración de canales.

Todas o algunas de las señales que se adquieren son ruidosas

- Algunos o todos los electrodos no están bien conectados.
- La conexión de los electrodos realizada no es coherente con la configuración de canales elegida. Revise la conexión de los electrodos y la configuración elegida utilizando las tablas 7.1, 7.2 o 7.3 según corresponda, para realizar la revisión.
- La impedancia de los electrodos es muy alta y es comparable con el AFE. Realice el test de impedancia de electrodos para los canales ruidosos y verifique que la misma es mucho menor a $1,3G\Omega$.

Apéndice I

Información complementaria de las pruebas

I.1. Pseudocódigo de la medición del throughput en función del tamaño de paquete

El código utilizado en el microcontrolador tiene la siguiente estructura:

```
paquete = ARREGLO[TAMPAQUETE];  
while(1){  
    bytesEnviados = 0;  
    do{  
        bytesEnviados = Radio_sendData(paquete ,  
                                        TAMPAQUETE);  
    } while(bytesEnviados == 0);  
}
```

La función *Radio_sendData(paquete, TAM_PAQUETE)* retorna un valor positivo cuando se envía el paquete. El control de flujo se explica en el anexo G.5.3.

El código en MatLab tiene la siguiente estructura:

Apéndice I. Información complementaria de las pruebas

```
1 %tiempo de la adquisicion en segundos
2 duracion_adquisicion = 120;
3
4 tic
5 tiempo = 0;
6
7 while(tiempo < duracion_adquisicion)
8     paquete = [];
9     while (esVacio(paquete))
10        paquete = leerPuertoTCP
11    end
12    tiempo = toc;           %Se carga el tiempo actual
13    paquetes_transmitidos = paquetes_transmitidos + length(
14        paquete);
end
```

tic se utiliza para comenzar a contar el tiempo desde MatLab y *toc* para conocer el valor actual del tiempo.

I.2. Comparación entre el tiempo de conversión (t_{conv}) medido y el esperado

La tabla I.1 compara t_{conv} con $t_{conv_{esperado}}$. Se omite el caso de un canal porque se adquiere distinto que el resto de los canales.

Configuración	$N_C + N_S$	t_{conv} medido (μs)	t_{conv} esperado (μs)
64 canales c/sinc	70	1172	1133
64 canales s/sinc	64	1030	1040
21 canales c/sinc	27	488	468
21 canales s/sinc	21	391	376
4 canales c/sinc	10	214	205
4 canales s/sinc	4	118	113

Tabla I.1: Comparación entre el tiempo que demora el AFE en convertir los canales teórico y medido.

Si N_C es la cantidad de canales, N_S la cantidad de canales de sincronismo y t_{CS} el tiempo que el CS está activo, se puede deducir a partir de la figura 8.6 que:

$$t_{conv_{esperado}} = (N_C + N_S + 4)t_{CS} + (N_C + N_S + 3)t_{entreCS}$$

Donde $t_{conv_{esperado}}$ indica el tiempo de conversión calculado a partir de los tiempos registrados en la tabla 8.2. La cantidad de palabras que se envían al AFE es la cantidad de canales que se desea adquirir ($N_C + N_S$) más la cantidad de comandos DUMMY que se envían (4 más) y la cantidad de tiempos entre CS es uno menos que la cantidad de mensajes. Como ejemplo de esto se puede ver la figura 8.6. Según lo calculado en la sección 8.2.1, $t_{CS} = 4,46\mu s$.

I.3. Deducción de la ecuación 8.1

La tabla I.1 muestra que lo esperado teóricamente se corresponde con lo medido. Las variaciones se deben especialmente a que los tiempos t_{CS} y $t_{entreCS}$ no son constantes y varían entorno a los valores mencionados en la sección 8.2.2.

I.3. Deducción de la ecuación 8.1

Se define t_{radio} al tiempo de envío del paquete desde el microcontrolador a la radio. Si $t_{adq} + t_{radio} < T_s$, se logra enviar el paquete de *buffer_packet* antes de que ocurra otra interrupción del timer_A0. Si esto no ocurre, entonces el microcontrolador no termina de enviar el paquete a la radio cuando debe muestrear una nueva tirada. Esto genera que el envío del paquete se vea interrumpido para ocuparse de muestrear todos los canales, reanudando el envío al finalizar con la adquisición, lo que provoca que el tiempo t_{radio} termine siendo mayor. Si T_s es muy rápido, el envío de los paquetes a la radio puede verse atrasado por varias tiradas. El límite de la cantidad de tiradas que puede demorarse esta tarea debe ser menor que la cantidad de tiradas que entran en un paquete, ya que si esto no ocurre, se van a intentar agregar más datos de los que se sacan provocando pérdidas en el sistema.

Se define $t_{radio_{min}}$ como el tiempo de envío del paquete a la radio sin que ocurran adquisiciones de tiradas en el medio. Teóricamente, se espera que si C es la cantidad de tiradas que ocurren mientras el envío no terminó, entonces $t_{radio_{teo}} \approx t_{radio_{min}} + Ct_{adq}$.

Frecuencia (f_s) en Hz	t_{radio} medido (ms)	Cantidad de tiradas adquiridas durante el envío (C_{medido})	¿Se registran pérdidas?	t_{radio} esperado (ms)
200	2,14	0	No	2,14
400	3,18	1	No	3,24
600	5,36	3	No	5,43
700	8,70	6	No	8,72
760	12,0	9	No	12,0
770	12,9	10	No	13,1
780	14,0	11	Sí	14,2

Tabla I.2: Tiempo de envío de paquete a la radio CC3100 en función de la frecuencia de muestreo. Las medidas fueron realizadas con una configuración de 64 canales sin sincronismo. La última columna muestra el valor del tiempo de envío de paquete esperado.

La tabla I.2 permite comparar el valor de t_{radio} medido con el valor $t_{radio_{teo}}$ para todos los casos de frecuencia. Se puede observar que siempre se cumple que $t_{radio} < t_{radio_{teo}}$. Esto se debe a que, mientras no ocurran interrupciones, el microcontrolador vuelve al programa principal (ver figura 4.11) en la cual aprovecha para seguir adelantando la tarea del envío del paquete a la radio. Nuestra ecuación de $t_{radio_{teo}}$ considera que no se adelanta en el envío de la radio mientras no haya terminado la adquisición, lo que en la realidad es falso. Esto es lo que lleva a una sobreestimación teórica.

Apéndice I. Información complementaria de las pruebas

Para la deducción teórica de C , se observa que el tiempo que demora el micro-controlador en enviar un paquete a la radio, si no hay interrupciones es $t_{radio_{min}}$. Si comienza a adquirir y no terminó el envío, el mismo se ve interrumpido un tiempo t_{adq} por cada tirada que se realiza en el medio del envío. Las figuras I.1a y I.1b muestran lo explicado anteriormente.

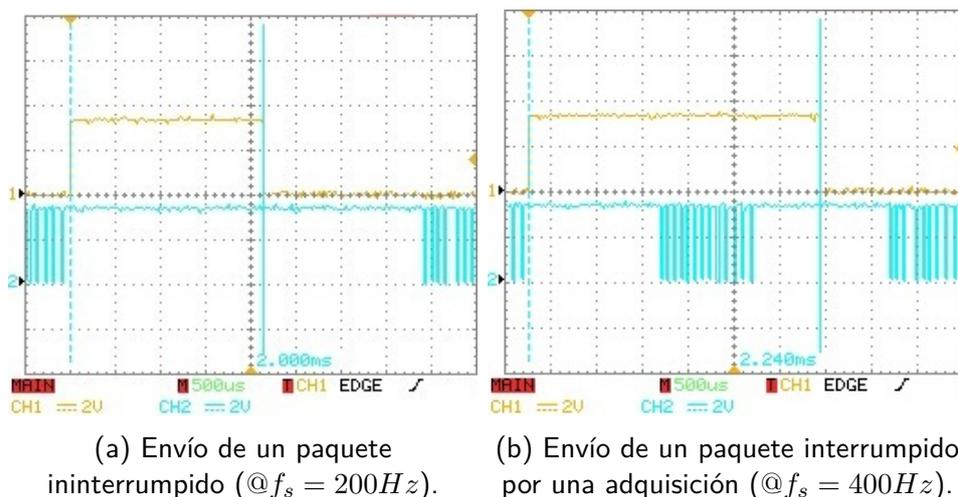


Figura I.1: Envío de un paquete a la radio a una configuración de 64 canales sin sincronismo. En ambas figuras, la señal superior muestra el estado de la transmisión de un paquete hacia la radio (1- Enviando, 0- No enviando), en la señal de abajo se muestra CS.

De la misma se puede deducir que $t_{radio_{min}} < C(T_s - t_{adq})$ con C el mínimo entero positivo que verifica esta inecuación, y por lo tanto:

$$C = \left\lfloor \frac{t_{radio_{min}}}{T_s - t_{adq}} \right\rfloor \quad (I.1)$$

donde $\lfloor x \rfloor$ representa al función piso (función trunc).

La tabla I.3 compara C_{teo} con el valor de C_{medido} de la tabla 8.4.

Frecuencia (f_s) en Hz	C_{medido}	C_{teo}
200	0	0
400	1	1
600	3	3
700	6	6
760	9	9
770	10	10
780	11	11

Tabla I.3: Comparación entre C_{teo} y C_{medido} .

I.4. Otras medidas de tiempos

Se midieron los tiempos de permanencia en las rutinas de atención a la interrupción y el tiempo de consulta de comandos (ejecución de la función *readCommand()*), estos tiempos se muestran en la tabla I.4.

	Tiempo (μs)	Condiciones
Tiempo en timer A0	45,2	700 Hz 64CH s/sinc
Tiempo en timer A1	< 1	700 Hz 64CH s/sinc
Tiempo en PORT2_IRQHandler	< 1	700 Hz 64CH s/sinc
Tiempo en leerComando	2800	TimerA0 off

Tabla I.4: Tiempos que el microcontrolador se encuentra en las rutinas de interrupción y en la función leerComando().

La interrupción PORT2_IRQHandler es usada por la radio para avisar al microcontrolador que la comunicación SPI entre estos dos está lista para usarse o que acaba de llegar un paquete proveniente del PC. Esta interrupción es rápida y no lleva al microcontrolador más de $1\mu s$ realizarla. Lo mismo ocurre con la interrupción del timer A1, que solo setea una bandera para ejecutar una función desde el programa principal. La función que se ejecuta desde el programa principal es leerComando() que es utilizada para determinar si hay un mensaje proveniente desde el PC. Esta función demora 2,8 ms si es ejecutada de forma ininterrumpida (timer A0 apagado). Si se está adquiriendo (timer A0 encendido) la función leerComando() es interrumpida de la misma manera que el envío de los datos del microcontrolador a la radio (ver sección 8.2.3). Esto lleva a que se acumulen algunos paquetes mientras se ejecuta esta función.

Por último, la interrupción del timer A0 ocurre solo cuando se está realizando un estudio y en ella se ejecuta la función CONVERTburst() que como se mencionaba en la sección 4.5.2 se encarga de inicializar las variables y de cargar los mensajes a enviar a los integrados RHD2132 para iniciar la adquisición de una nueva tirada. Este tiempo, se corresponde con el tiempo (2) de la figura 8.6 que para el caso de una frecuencia de muestreo de 700Hz con 64 canales sin sincronismo tiene un valor de $40\mu s$ según muestra la tabla 8.3. Que en este caso sea de $45,2\mu s$ ($5,2\mu s$ mayor) está asociado a que necesita un tiempo extra para ingresar a la rutina de interrupción y ejecutar algunas instrucciones antes de entrar a la función CONVERTburst().

I.5. Deducciones de la ocurrencia de pérdida de datos

Por lo analizado en la sección 4.9 existen dos tipos de pérdidas denominadas **superposición de adquisiciones de tiradas consecutivas** y **desbordamiento del buffer de transmisión de paquetes del microcontrolador a la radio** en la sección 6.7. Las de superposición de adquisiciones de tiradas consecutivas ocurren si no se terminó de adquirir una tirada cuando debe empezarse otra. En términos de los tiempos de la sección 8.2.3, ocurre cuando $t_{adq} > T_s$ siendo T_s el

Apéndice I. Información complementaria de las pruebas

período de muestreo. Las de desbordamiento del buffer de transmisión de paquetes del microcontrolador a la radio, ocurren cuando se llena *buffer_packet*. Esto va a ocurrir si *buffer_packet* se llena más rápido de lo que se envían los paquetes a la radio y está asociado a los tiempos que se estudiaron en la sección 8.2.3. Si la cantidad de tiradas que entran en un paquete es $N_{tiradas}$ y la cantidad de tiradas que ocurren mientras el envío de un paquete del microcontrolador a la radio no terminó es C , por lo visto en la sección 8.2.3, la condición necesaria y suficiente para que haya pérdidas por desbordamiento de *buffer_packet* es que el tiempo de adquirir $N_{tiradas}$ sea menor que el tiempo que demora en enviar el paquete a la radio (tiempo t_{radio}), es decir que, $N_{tiradas}T_s < t_{radio}$. En la sección 8.2.3 se mostró que $t_{radio} \approx t_{radio_{min}} + Ct_{adq}$.

Observación: Las pérdidas por desbordamiento de *buffer_packet* siempre ocurren antes que las pérdidas de superposición de adquisiciones de tiradas consecutivas. Para deducir esto, se demuestra que si se cumple la condición para pérdidas de superposición de adquisiciones de tiradas consecutivas, también se cumple la condición de pérdidas de desbordamiento de *buffer_packet*. En efecto, si ocurren las pérdidas de superposición de adquisiciones de tiradas consecutivas, el sistema no termina de adquirir una tirada cuando ya debe empezar con otra, por lo que, no hay espacio de tiempo para que el microcontrolador envíe un paquete a la radio, lo que lleva a que trivialmente se cumpla que $N_{tiradas} < C$. Luego como ocurren las pérdidas de superposición de adquisiciones de tiradas consecutivas, $T_s < t_{adq}$ y entonces $N_{tiradas}T_s < N_{tiradas}t_{adq} < t_{radio_{min}} + N_{tiradas}t_{adq} < t_{radio_{min}} + Ct_{adq} = t_{radio}$, condición necesaria y suficiente para las pérdidas por desbordamiento de *buffer_packet*.

A partir de esto, se deduce que para encontrar la frecuencia máxima solo se necesita analizar bajo que condiciones ocurren las pérdidas por desbordamiento de *buffer_packet*.

Análisis del porcentaje de pérdidas por desbordamiento de *buffer_packet*

En el anexo I.6 se deduce que las pérdidas por desbordamiento de *buffer_packet* son periódicas de período $T_{perdidas} + \Delta T_{perdidas}$ y que el porcentaje de este tipo de pérdida se calcula como:

$$\%_{perdidas} = 100\% \times \frac{\Delta T_{perdidas}}{T_{perdidas} + \Delta T_{perdidas}} \quad (I.2)$$

donde $T_{perdidas}$ es el tiempo que demora en generarse una pérdida y $\Delta T_{perdidas}$ corresponde en segundos a la cantidad de datos que se perdieron. A su vez $T_{perdidas} = \frac{40}{N_{tiradas}T_s - t_{radio}}$ y $\Delta T_{perdidas} = 40N_{tiradas}T_s$.

La tabla I.5 compara las pérdidas obtenidas en la tabla 8.6 con la esperada teóricamente a partir del análisis anterior. Para el caso teórico se tomó como valor de frecuencia de muestreo $[f_{s_{max}} + 10 \times 0,957]$ para que la comparación tenga coherencia con el valor medido.

I.6. Dedución de la ecuación I.2

Configuración	$N_{tiradas}$	C	t_{radio} en (ms)	f_s (Hz)	$T_{perdidas}$ (s)	$\Delta T_{perdidas}$ (s)	% $_{perdidas}$ teórico	% $_{perdidas}$ medido
64 canales c/sinc	10	11	16,11	683	6,4	0,585	8,36	1,13
64 canales s/sinc	11	11	14,20	784	48,2	0,561	1,15	2,12
21 canales c/sinc	27	28	17,93	1564	18,5	0,691	3,59	1,32
		30	19,06	1574	6,9	0,686	9,09	7,93
21 canales s/sinc	34	35	17,54	1997	23,2	0,681	2,84	1,06
		36	17,98	2007	11,7	0,678	5,46	4,59
4 canales c/sinc	72	74	23,45	3156	33,5	0,913	2,65	3,80
4 canales s/sinc	182	186	32,83	5666	59,5	1,285	2,11	5,20
1 canal s/sinc	729	186	32,83	5666	59,5	1,285	2,11	5,20
$t_{radio_{min}} = 2,14ms$								

Tabla I.5: Comparación entre el porcentaje teórico y el porcentaje de pérdidas medido.

Se observa que el porcentaje de pérdidas teórico es coherente con el medido. Las diferencias entre estos están asociadas a la sensibilidad de la ecuación I.2, especialmente del término $T_{perdidas}$.

La razón por la que se registran pérdidas viene dada por que la copia de tiradas a los paquetes de `buffer_packet` se hacen en una interrupción y el envío de paquetes en la radio se hace en el loop principal. En este caso, mientras más frecuentes sean las interrupciones por el timer A0 y su posterior procesamiento por SPI, menor es el tiempo en que se le pueda dedicar al vaciado del buffer y envío hacia el buffer de transmisión del CC3100.

I.6. Dedución de la ecuación I.2

Si se está en un caso de pérdidas, se limpia `buffer_packet` y se inicia la adquisición nuevamente¹.

Si se está en un caso de pérdidas, se cumple la condición de la ecuación 8.2. Observar que $N_{tiradas}T_s$ corresponde al tiempo de llenado de `buffer_packet` y t_{radio} al tiempo de vaciado de `buffer_packet`, y que la condición de pérdidas es tener un flujo de entrada mayor a un flujo de salida. Desde un punto de vista de caudales, podemos entender que las pérdidas de *desbordamiento del buffer*, ocurren si se llena todo el `buffer_packet`, que tiene capacidad para 40 paquetes como se indicaba en la sección 4.6. El tiempo que demora en ocurrir la pérdida se expresa como:

¹La adquisición se inicia nuevamente desde el punto de vista del módulo remoto. No es necesario que desde la interfaz de usuario se envíe un comando START.

Apéndice I. Información complementaria de las pruebas

$$T_{perdidas} = \frac{40}{\frac{1}{N_{tiradas}T_s} - \frac{1}{t_{radio}}}$$

Luego de ocurrida la pérdida se limpia `buffer_packet` y se continúa con el curso normal del estudio, pero como las condiciones son las mismas, tiempo $T_{perdidas}$ después ocurre otra pérdida y así sucesivamente. Como se observa en la figura I.2, la ocurrencia de pérdidas es periódica.

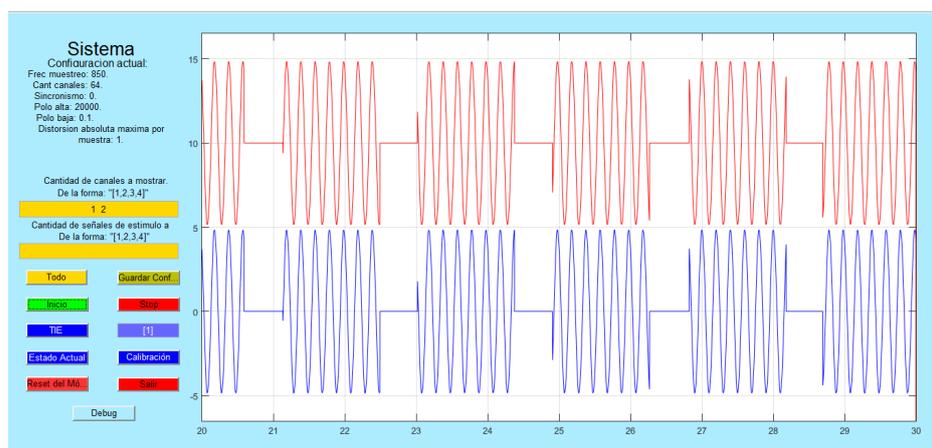


Figura I.2: En la figura se muestra una senoide de $5Hz$ de frecuencia y $5mV_{pp}$ de amplitud, muestreada a $850Hz$ con configuración 64 canales y sin sincronismo, con un tiempo de ventana de $10s$. En la misma se muestra el patrón de pérdidas periódico debido al problema de control de flujo en `buffer_packet`. Las pérdidas corresponden al tiempo de las señales graficadas se mantienen constantes en $0V$.

El tiempo de la pérdida es igual a la cantidad de datos que son descartados. Si llamamos $\Delta T_{perdidas}$ a este tiempo, entonces:

$$\Delta T_{perdidas} = 40N_{tiradas}T_s$$

El porcentaje de pérdidas es:

$$\%_{perdidas} = 100\% \times \frac{\Delta T_{perdidas}}{T_{perdidas} + \Delta T_{perdidas}}$$

Según lo indicado en la sección 8.2.3, $t_{radio} = t_{radio_{min}} + Ct_{adq}$ y $C = \lfloor \frac{T_{radio_{min}}}{T_s - t_{adq}} \rfloor$.

I.7. Circuitos para la medida de consumo

Para medir el consumo del módulo remoto se realizó el circuito de la figura I.3.

La corriente que pase por la resistencia será la corriente que consume el módulo remoto. Haciendo la resta entre los canales 1 y 2 del osciloscopio se puede ver la forma de la caída de voltaje en R_{shunt} , que es proporcional a la corriente que se

I.7. Circuitos para la medida de consumo

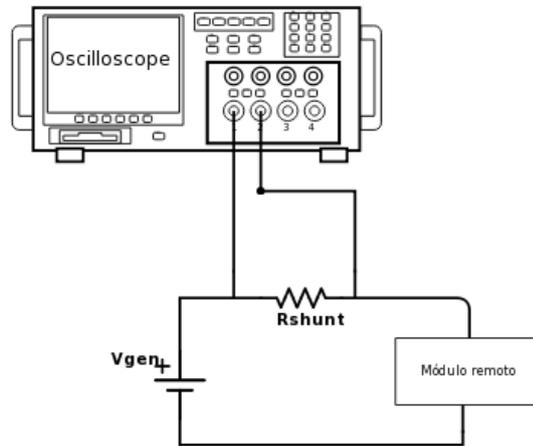


Figura I.3: Circuito para medir el consumo del módulo remoto.

desea medir. La resistencia R_{shunt} y el voltaje V_{gen} fueron elegidos teniendo en cuenta que:

- El voltaje en el canal 2 es el voltaje con el cual se alimenta el módulo remoto. Este voltaje no puede ser menor que $3,3V$ para el correcto funcionamiento de los convertidores DC/DC que alimentan a los tres bloques del sistema (AFE, MCU y Radio).
- La caída de voltaje en la resistencia debe ser lo suficientemente grande como para poder visualizar perfectamente la forma de onda en el osciloscopio. Se elige tener una precisión mayor a $1mV$ por mA de corriente.
- La potencia de la resistencia elegida debe ser la necesaria, según la corriente que se estima va a pasar por ella. Se limita que la potencia que disipe la resistencia a utilizar no supere los $0,25W$.
- Para evitar riesgo de fallas, no se va a utilizar un voltaje en el generador mayor a la capacidad máxima de los convertidores DC/DC. Se limita V_{gen} a un valor no superior a $5V$.

En términos de ecuaciones debe cumplirse que:

$$V_{gen} - R_{shunt}I > 3,3V ; R_{shunt} > \frac{1mV}{1mA} = 1\Omega ; R_{shunt}I^2 < 0,25W ; V_{gen} < 5V.$$

En la sección 2.4.1 se estimaba que el consumo del módulo remoto era a lo sumo $282,85mA$, por lo que, considerando que $I < 300mA$ se decide elegir $R_{shunt} = 2\Omega$ y $V_{gen} = 4,1V$.

Luego se midió el consumo de la radio y el microcontrolador quitando la placa AFE y alimentando al microcontrolador a través de sus pines de potencia y midiendo la corriente de la misma manera que en la figura I.3. De la misma manera

Apéndice I. Información complementaria de las pruebas

se midió el consumo del microcontrolador + el AFE y luego el del microcontrolador solo. El consumo del AFE solo y de la radio sola, se obtuvieron restando el consumo del microcontrolador en las medidas anteriores.

I.8. Deducción de la ecuación 8.4

La figura I.4 muestra una aproximación teórica de la corriente por el módulo remoto en función del tiempo para la configuración de canales elegida.

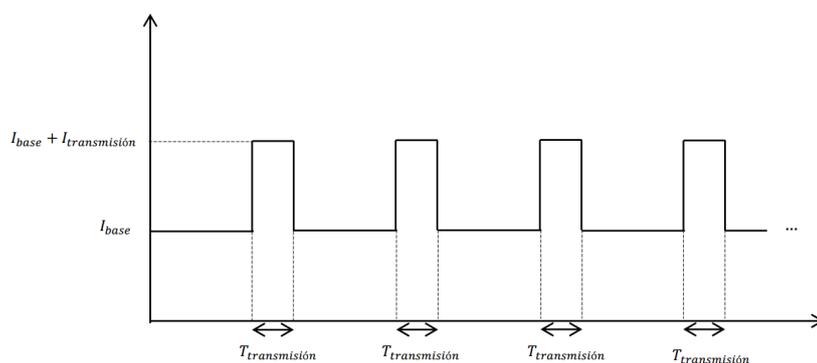


Figura I.4: Modelo teórico de la corriente que consume el módulo remoto para la configuración de 64 canales sin sincronismo.

Si T_{total} es el tiempo que el módulo remoto funciona, se cumple que:

$$\int_0^{T_{total}} i(t)dt = T_{total}I_{base} + N_{picos}T_{transmision}I_{transmision} = 3200mAh$$

Donde N_{picos} es la cantidad de transmisiones que se realizan hasta el tiempo total.

$$N_{picos} \approx \frac{T_{total}}{N_{tiradas}T_s + T_{transmision}}$$

Despejando T_{total} se tiene que:

$$T_{total} = 3200mAh \frac{N_{tiradas}T_s + T_{transmision}}{T_{transmision}(I_{base} + I_{transmision}) + I_{base}N_{tiradas}T_s}$$

Apéndice J

Seguridad en equipos médicos

Como el sistema diseñado es un equipamiento médico que va a estar en contacto con personas, el mismo debe cumplir ciertos requerimientos generales para una seguridad básica de quien lo opere. En el caso del proyecto, como no se trata de un sistema implantable, esta normativa es la 60601-1 IEC 2005 disponible de forma gratuita (ver [64]).

Es muy importante definir cuáles son las partes que van a estar en contacto con el paciente y con el investigador en condiciones normales (operación normal sin ninguna falla). El sistema está dentro de una caja que impide que una persona entre en contacto salvo con los electrodos y cables IDE, estas son las partes que pueden entrar en contacto con el paciente. Las posibles fallas que puede tener el sistema son:

- Rotura de la protección de la batería.
- Cortocircuito que provoque un incendio en el módulo remoto.
- Lesiones en el paciente por ahorcamiento con los conectores.

En condiciones normales, para registrar la fiabilidad del sistema ante fallas se deben tener en cuenta:

Alimentación: el sistema es alimentado por una batería y no debe usarse el sistema con la batería mientras la batería se carga. Esta batería cuenta con un circuito protector que limita la corriente máxima que entrega, evitando sobrecalentamientos y explosiones. Para evitar poner en riesgo la vida de las personas que utilicen este sistema, no se debe cambiar la batería con el equipo funcionando.

Instalación: en la instalación del módulo remoto en el cabeza del paciente hay que tener especial cuidado en como se colocan los cables de los electrodos, de manera que no lastimen ni ahorquen al usuario.

Apéndice J. Seguridad en equipos médicos

Corriente por el paciente: en la norma se hace especial cuidado con la corriente que puede pasar por el paciente en condiciones normales y si hay una sola falla. La tabla J.1 indica los valores máximos de corriente de continua y alterna que pueden pasar por el paciente.

Tipo de corriente	En condiciones normales (NC)	Con una sola falla (SFC)
Corriente continua	$10\mu A$	$50\mu A$
Corriente alterna	$100\mu A$	$500\mu A$

Tabla J.1: Corriente máxima que puede circular por el paciente en condiciones normales y si hay una sola falla. Información extraída de la página 86 de [64].

Para este caso, se debe medir la corriente que pasa por los electrodos en condiciones normales y para una falla simple. Esta medida no fue realizada para verificar si el sistema cumple con las normas de corriente por el paciente. El sistema en adquisición o realizando el test de impedancia de electrodos, parecen ser las tareas más riesgosas puesto que las dos involucran que circule corriente a través de los electrodos.

Referencias

- [1] E. Cilleruelo, A. Nacelle, and G. Robert, “NESIA,” *Proyecto de fin de carrera de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay*, 2013.
- [2] DataWave Technologies, “Wireless headstages.” <http://www.dwavetech.com/products/HdstgWireless.htm> , Consultado: 18/10/16.
- [3] Neuralynx, “Headstages pre amplifiers.” Consultado: 18/10/16.
- [4] Natus, “Nicolet EEG Wireless Amplifier.” Consultado: 18/10/16.
- [5] Grupo wEEG, “Entrevistas realizadas por los autores del presente proyecto a los clientes,” 2016.
- [6] I. Capurro, Federico Lecumberry, Alvaro Martín, Ignacio Ramírez, Eugenio Rovira, and Gadiel Seroussi, “Efficient sequential compression of multi-channel biomedical signals,” *Facultad de ingeniería, Universidad de la República, Montevideo, Uruguay*, 2016.
- [7] Wikipedia, “Richard Caton.” Consultado: 18/10/16.
- [8] P. Gloor, *Hans Berger on the Electroencephalogram of Man*. Elsevier Publishing Company, 1969, 1969.
- [9] Ciencias de Joseleg, “El electroencefalograma.” Consultado: 18/10/16.
- [10] Uncle Milton Industries, “Star Wars Science.” Consultado: 18/10/16.
- [11] Edmonton metro, “NAIT students help create mind-controlled video game.” Consultado: 09/11/16.
- [12] VROne. Consultado: 09/11/16.
- [13] SourceForge.net, “OpenEEG.” Consultado: 18/10/16.
- [14] Open Ephys, “Open Ephys.” Consultado: 18/10/16.
- [15] M. Sawan, Fellow, IEEE, Muhammad T. Salam, Jerome Le Lan, Amal Kasab, Sébastien Gelinas, Phetsamone Vannasing, Frédéric Lesage, Maryse Lassonde, and Dang K. Nguyen, “Wireless Recording Systems: From Noninvasive EEG-NIRS to Invasive EEG Devices,” *PubMed*, 2013.

Referencias

- [16] Alexander J. Casson, David C. Yates, Shelagh J.M. Smith, John S. Duncan, and Esther Rodriguez Villegas, “Wearable Electroencephalography,” *PubMed*, 2010.
- [17] Emotiv, “EPOC/EPOC+.” Consultado: 20/10/16.
- [18] Emotiv, “Insight.” Consultado: 20/10/16.
- [19] Neuroelectronics, “ENOBIO.” Consultado: 20/10/16.
- [20] Great Lakes NeuroTechnologies, “BioRadio.” Consultado: 20/10/16.
- [21] Cognionics, “HD-72.” Consultado: 20/10/16.
- [22] Wikipedia, “Ondas delta.” Consultado: 09/11/16.
- [23] Wikipedia, “Ondas theta.” Consultado: 09/11/16.
- [24] Wikipedia, “Ondas alfa.” Consultado: 09/11/16.
- [25] Wikipedia, “Ondas beta.” Consultado: 09/11/16.
- [26] Wikipedia, “Ondas gamma.” Consultado: 09/11/16.
- [27] Texas Instruments, “MSP432xx Family - Technical Reference Manual.” Consultado: 21/10/16.
- [28] Texas Instruments, “MSP432P401x Mixed-Signal Microcontrollers.” Consultado: 21/10/16.
- [29] g.Tec Medical Engineering, “g.SAHARA Active dry electrode system.” Consultado: 21/10/16.
- [30] Intan Technologies, “RHD2216 - RHD2132.” Consultado: 22/10/16.
- [31] Texas Instruments, “CC3100/CC3200 SimpleLink Wi-Fi Internet-on-a-Chip.” Consultado: 06/11/16.
- [32] La Paz, Franco, “SPI connection with two RHD2132,” Consulta realizada el 25/05/16.
- [33] Wikipedia, “ZigBee.” Consultado: 21/08/15.
- [34] Monografías, “Zigbee: El nuevo estándar global para la domótica e inmótica.” Consultado: 29/10/16.
- [35] Wikipedia, “Bluetooth.” Consultado: 21/08/15.
- [36] Wikipedia, “Wi-Fi.” Consultado: 21/08/15.
- [37] CCM, “Introducción a Wi-Fi (802.11 o WiFi).” Consultado: 29/10/16.
- [38] Wikipedia, “Bluetooth de baja energía.” Consultado: 29/10/16.

- [39] J. Rowberg, “Bluegiga Forums/Knowledgebase/Bluetooth Smart.” Consultado: 29/10/16.
- [40] K. Karami, “Maximizing BLE Throughput on iOS and Android.” Consultado: 29/10/16.
- [41] G. Kewney, “High speed Bluetooth comes a step closer: enhanced data rate approved.” Consultado: 29/10/16.
- [42] Laird, “SPP Throughput Analysis.” Consultado: 29/10/16.
- [43] Xataka Móvil, “Qué son los canales Wi-Fi y cómo escoger el mejor para nuestra red.” Consultado: 16/11/16.
- [44] Geocities, “Sensor placmente for EEG.” Consultado: 31/10/16.
- [45] Bem, “Electroencefalografhy.” Consultado: 31/10/16.
- [46] Brainm, “Electrode Positions.” Consultado: 31/10/16.
- [47] Wikipedia, “Diafonía.” Consultado: 06/03/17.
- [48] Texas Instruments, “CC3100 SimpleLink.” Consultado: 30/11/16.
- [49] Panasonic, “Panasonic NCR18650 Lithium Ion.” Consultado: 26/11/16.
- [50] Corrosion Doctors, “Nominal Capacity and Discharge Current.” Consultado: 26/11/16.
- [51] Wikipedia, “Router.” Consultado: 25/10/16.
- [52] Lygte, “UltraFire TR18650 5000mAh (Blue).” Consultado: 21/10/16.
- [53] Lygte, “Samsung ICR18650 2600mAh (Pink).” Consultado: 21/10/16.
- [54] Lygte, “Panasonic NCR18650B 3400mAh (Green).” Consultado: 21/10/16.
- [55] Hong King Tac, “18650-4S batería de ión de litio cilíndricas 12.8 v 2200mAh para instrumentos electrónicos.” Consultado: 25/10/16.
- [56] Wikipedia, “Senal diferencial de bajo voltaje.” Consultado: 14/03/16.
- [57] Analog Devices, “ADP150 Datasheet.” Consultado: 23/10/16.
- [58] Texas Instrument, “TPS736 Datasheet.” Consultado: 23/10/16.
- [59] Hughes Circuit, “Impedance calculator.” Consultado: 30/03/16.
- [60] Wikipedia, “FR-4.” Consultado: 30/03/16.
- [61] Facultad de Ingeniería, Udelar, “Arquitecturas de software.” Consultado: 30/10/16.
- [62] Wikipedia, “Socket de Inernet.” Consultado: 14/05/17.

Referencias

- [63] Wikipedia, “Dynamic Host Configuration Protocol.” Consultado: 14/05/17.
- [64] IEC (International Electrotechnical Commission, “Medical electrical requirement (60601-1 IEC 2005,” *International Organization for Standardization (ISO)*, 2005.

Índice de tablas

1.1.	Estudio de mercado de electroencefalógrafos inalámbricos comerciales.	4
1.2.	Requerimientos generales.	6
1.3.	Requerimientos para el AFE (64 canales).	6
1.4.	Requerimientos para el convertor A/D (64 canales).	7
1.5.	Requerimientos para el Microcontrolador.	7
1.6.	Requerimientos para el transmisor/receptor.	7
2.1.	Cuadro comparativo de las características de la familia RHD2000 con lo requerido por el cliente.	12
2.2.	Tabla comparativa entre RHD2132 y RHD2164.	13
2.3.	Opciones de microcontroladores a utilizarse para el sistema, todas las opciones cuentan con comunicación SPI.	14
2.4.	Opciones de radios Wi-Fi a utilizarse en el sistema. Todas estas opciones utilizan SPI o UART, tienen los protocolos 802.11 b,g y n y cuentan con Stack TCP/IP.	15
2.5.	Detalle del cálculo de consumo del RHD2132. Se recuerda que el AFE consta de dos chips RHD2132 cada uno capaz de adquirir 32 señales. Por lo tanto todos los consumos se deben contar dos veces.	16
2.6.	Rango de voltaje de funcionamiento de los tres integrados utilizados en el módulo remoto.	17
2.7.	Tabla comparativa de las opciones de baterías a comprar. Ninguna de las pilas de la tabla viene con cargador.	18
2.8.	Tabla de electrodos analizados para realizar la compra. Información obtenida por contacto con los proveedores.	19
2.9.	Electrodos húmedos en el mercado local.	20
5.1.	Requerimientos para el transmisor/receptor.	49
6.1.	Descripción de los comandos disponibles desde la interfaz de usuario que configuran al módulo remoto.	56
6.2.	Correspondencia de las señales de sincronismo con los pines del conector CON_EST en la placa AFE.	62
7.1.	Relación entre la numeración de canales de la interfaz de usuario y la posición en la cabeza para la configuración de 64 canales.	72

Índice de tablas

7.2. Relación entre la numeración de canales de la interfaz de usuario y la posición en la cabeza para la configuración de 21 canales.	73
7.3. Relación entre la numeración de canales de la interfaz de usuario y la posición en la cabeza para la configuración de 4 canales.	74
8.1. Throughput en la transmisión desde el módulo remoto al PC en función del tipo de comunicación router-PC.	80
8.2. Tiempos de las distintas tareas que se realizan en la adquisición de la señal de un canal.	83
8.3. Tiempos de la adquisición de una tirada para distintas configuraciones de canales. Los casos de configuración con sincronismo implica muestrear los 6 canales de sincronismo. Los índices de tiempo hacen referencia a la figura 8.6. El tiempo (5), $t_{entreCS} = 11\mu s$ en todos los casos.	85
8.4. Tiempo de envío de paquete a la radio CC3100 en función de la frecuencia de muestreo. Las medidas fueron realizadas con una configuración de 64 canales sin sincronismo. La última columna muestra el valor del tiempo de envío de paquete esperado.	87
8.5. Comparación entre la frecuencia de dos de los microcontroladores disponibles en el proyecto (MCU_T hace referencia al microcontrolador utilizado para medir los tiempos y MCU_F el utilizado para medir la frecuencias máxima). Se observa que uno de ellos tiene una variación de 4,3% que lo hace más lento e inexacto.	88
8.6. Frecuencia máxima alcanzable para cada configuración. Para frecuencias mayores comienzan a registrarse pérdidas periódicas. <i>Frecuencia medida</i> corresponde a las medidas realizadas con el microcontrolador de frecuencia.	89
8.7. Comparación entre la frecuencia máxima medida y la esperada. El cálculo de $N_{tiradas}$ fue explicado en la sección 4.6, $N_{tiradas} = \frac{729}{N_c + N_s}$	90
8.8. Limitaciones de la frecuencia de muestreo alcanzable por el tiempo de adquisición.	91
8.9. Consumo en inactividad de los bloques del módulo remoto.	92
8.10. Medidas del módulo remoto durante un estudio.	92
8.11. Frecuencia a la cual el consumo debido al copiado de paquetes al buffer de transmisión del CC3100 comienza a ser constante.	93
8.12. Listado de las señales inyectadas junto a la configuración utilizada. Todas las señales de entrada tiene una amplitud de $10mV_{pp}$	96
8.13. Características finales del sistema	99
9.1. Requerimientos generales alcanzados.	102
9.2. Requerimientos alcanzados del AFE.	102
9.3. Requerimientos alcanzados para el conversor A/D.	102
9.4. Requerimientos alcanzados para el microcontrolador.	102
9.5. Requerimientos alcanzados para la radio.	103
9.6. Comparación del sistema wEEG con electroencefalógrafos inalámbricos comerciales.	104

A.1. Resumen del ruido agregado por el ADS para $f_s = 1kHz$ por canal (más información en las páginas 6 y 7 de la hoja de datos del integrado ADS1299).	110
B.1. Opciones de radios Wi-Fi a utilizarse en proyecto.	113
B.2. Opciones de Radios Wi-Fi a utilizarse en proyecto.	114
B.3. Posibles Routers a utilizar en el proyecto.	115
B.4. Comparación de algunos de los conversores LDO estudiados, en todos los componentes la eficiencia viene dada por η	116
B.5. Comparación de algunos de los conversores LDO estudiados.	117
B.6. Precios de materiales y mano de obra de la modista.	119
B.7. Tabla de algunas de las carcasas estudiadas. Todas las opciones mostradas en la tabla son de plástico.	120
D.1. Resumen de los conectores de la placa y su funcionalidad.	126
D.2. Conexión de los pines de CON_EST y MSP3.	127
I.1. Comparación entre el tiempo que demora el AFE en convertir los canales teórico y medido.	160
I.2. Tiempo de envío de paquete a la radio CC3100 en función de la frecuencia de muestreo. Las medidas fueron realizadas con una configuración de 64 canales sin sincronismo. La última columna muestra el valor del tiempo de envío de paquete esperado.	161
I.3. Comparación entre C_{teo} y C_{medido}	162
I.4. Tiempos que el microcontrolador se encuentra en las rutinas de interrupción y en la función leerComando().	163
I.5. Comparación entre el porcentaje teórico y el porcentaje de pérdidas medido.	165
J.1. Corriente máxima que puede circular por el paciente en condiciones normales y si hay una sola falla. Información extraída de la página 86 de [64].	170

Esta página ha sido intencionalmente dejada en blanco.

Índice de figuras

1.1. Estudios de electroencefalografía actuales (imagen extraída de [9]).	3
1.2. Diagrama de los bloques del sistema.	6
1.3. Comunicación de los distintos bloques del sistema.	8
2.1. Electrodo seco de Cognionics Inc..	19
3.1. Layout capas superior e inferior de la placa del AFE	21
3.2. Vistas superior e inferior de la placa.	22
3.3. Posibles configuraciones para comunicar los integrados RHD2132 al microcontrolador MSP432.	24
3.4. Conexión de los componentes externos para un chip RHD2132 (imagen extraída de la página 14 de [30]).	25
3.5. Disposición de los pines del MSP432 y su conexión con el resto de los bloques del módulo remoto.	26
4.1. Tipo de datos que comparte el microcontrolador con el resto de los bloques del módulo remoto.	27
4.2. Diagrama de estados básico del software embebido.	29
4.3. Estructura de módulos del microcontrolador. Las flechas salientes de un módulo indica que usa funciones de otro módulo. Ejemplo: Timer.h utiliza funciones de AFE.h.	32
4.4. Funcionamiento deseado de la comunicación SPI para los integrados RHD2132 (imagen extraída de [30]).	34
4.5. Funcionamiento de la comunicación SPI de 16 bits.	34
4.6. Diagrama de tiempos de instrucciones para el RHD2132 (imagen extraída de [30]).	35
4.7. Diagrama en bloques de la transmisión y recepción de la comunicación SPI del lado del microcontrolador.	36
4.8. Diagrama de flujo del módulo principal con arquitectura Round-Robin con interrupciones.	42
4.9. Diagrama de flujo para interpretar un comando proveniente del PC.	44
4.10. Diagrama de tiempo correspondiente a un estudio.	45

Índice de figuras

4.11. Diagrama de flujo del módulo remoto en medio de un estudio. Se recuerda que la función <code>send_word(1)</code> se utiliza para comenzar la adquisición con el chip 1 y que la función <code>transfer_channel()</code> se encarga de copiar la tirada al buffer de transmisión de paquetes a la radio (<code>buffer_packet</code>).	47
5.1. Canales Wi-Fi dentro de la banda de 2,4GHz (imagen extraída de [43]).	52
6.1. Diagrama de flujo básico de la interfaz de usuario.	58
6.2. Menú principal de la interfaz de usuario.	59
6.3. Menú principal de la interfaz de usuario.	60
6.4. Menú para indicar la configuración de canales para el próximo estudio a realizar.	61
6.5. Ventana gráfica en medio del estudio de una señal sinusoidal de 10Hz y 10mVpp. Como se observa en el extremo superior izquierdo de la imagen, la configuración es de 64 canales con una frecuencia de muestreo de 700Hz.	64
6.6. Recepción de una señal sinusoidal de 5Hz y 10mVpp con fallas debidas a superposición de adquisición de tiradas consecutivas. La configuración de canales es 64 y la frecuencia de muestreo es 890Hz.	66
6.7. Recepción de una señal sinusoidal de 10Hz y 10mVpp con pérdidas debidas a desbordamiento de buffer. Las pérdidas ocurren entre los 14,8s y 15,4s en los dos canales que se grafican. La configuración de canales es 64 y la frecuencia de muestreo es 810Hz.	67
7.1. Electrodo húmedo de cloruro de plata (Ag/AgCl).	69
7.2. Diseño parcial del casco sin los orificios de colocación de los electrodos.	70
7.3. Diagrama de la conexión de las placas en el módulo remoto.	70
7.4. Carcasa y su contenido.	71
7.5. En color se muestra las posiciones que se utilizan con la configuración de 64 canales (imagen extraída de [44] y luego modificada).	72
7.6. En color se muestran las posiciones que se utilizan con la configuración de 21 canales (imagen extraída de [45] y luego modificada).	73
7.7. En color se muestran las posiciones que se utilizan con la configuración de 4 canales (imagen extraída de [46] y luego modificada).	74
7.8. Conexión de los electrodos con la carcasa, a la izquierda se muestra la opción de conectar los electrodos directamente. A la derecha se muestra la opción de extender la conexión mediante un cable plano, de forma de dejar la carcasa en la cintura del paciente. Referencias: 1-Conector IDE, 2-Carcasa.	75
7.9. Distintos diseños de la unión de los electrodos con la carcasa. En (a) se muestra un cable comercial para electrodos de ECG, en (b) se muestra un cable fabricado artesanalmente, en (c) se muestran los broches utilizados en el diseño artesanal que sujetan los electrodos.	76

8.1.	Throughput útil en la transmisión desde el módulo remoto al PC en función del tamaño de paquete.	78
8.2.	Throughput en la transmisión desde el módulo remoto al PC en función de la distancia entre el módulo remoto y el router.	79
8.3.	Redes de Wi-Fi disponibles. En la tabla se indica el nombre de la red, la potencia en dBm (columna RSSI), canal de Wi-Fi en la que se encuentra (columna Chan) y máxima velocidad de la red (columna Max Speed). Por debajo se muestra la variación de la potencia de cada señal en función del tiempo.	81
8.4.	\overline{CS} (arriba) y CLK (abajo) medidos. Las números indican: 1-Tiempo activo de \overline{CS} (t_{CS}). 2-Tiempo previo al envío o la recepción de la parte alta ($t_{inicio_{SPI}}$). 3-Tiempo de envío o recepción de cada palabra de 8 bits ($t_{palabra_{SPI}}$). 4-Tiempo entre envío o recepción de parte alta y parte baja ($t_{intermedios_{SPI}}$). 5-Tiempo posterior al fin del envío o la recepción de la parte baja ($t_{final_{SPI}}$).	82
8.5.	Posibles configuraciones para la adquisición de señales.	83
8.6.	Adquisición de 4 canales con 6 señales de sincronismo. Arriba se muestra la bandera <i>is_acquiring</i> y abajo una señal auxiliar que indica por nivel bajo cuando el \overline{CS} de alguno de los puertos SPI está activo. Las referencias indican: 1- Tiempo de <i>is_acquiring</i> activo (t_{adq}). 2- Tiempo anterior a la adquisición del primer canal (t_{CB}). 3- Tiempo neto de la adquisición de una tirada (t_{conv}). 4- Tiempo entre la adquisición del último canal y el copiado de la tirada a buffer_packet (t_{BP}). 5- Tiempo entre adquisiciones de canales consecutivas ($t_{entreCS}$).	85
8.7.	Consumo del módulo remoto para una adquisición de 4 canales sin sincronismo a una frecuencia de muestreo de 200 Hz.	93
8.8.	Gráfico del voltaje entre bornes de la pila en función de su descarga (imagen extraída de [49]).	94
8.9.	Señales medidas para la prueba de punta al punta del sistema, las señales tienen una amplitud de $10mV_{pp}$ y las frecuencias de corte superior e inferior fueron configuradas a $20kHz$ y $0,1Hz$ respectivamente.	96
8.10.	Estudio con todos los canales testeando su funcionamiento.	98
A.1.	Opción a), colocar 64 filtros programables a la entrada del AFE. Opción b), colocar 8 filtros programables a la salida del MUX analógico.109	
B.1.	Posibles configuraciones para la regulación de la tensión de la batería.115	
B.2.	Test de la pila UltraFire 18650 5000 mAh. La figura fue extraída de [52]. Para más información del test visitar el link en la referencia. 117	
B.3.	Test de la pila Samsung ICR18650-26F 2600 mAh. La figura fue extraída de [53]. Para más información del test visitar el link en la referencia.	118

Índice de figuras

B.4. Test de la pila Panasonic NCR18650B 3400 mAh. La figura fue extraída de [54]. Para más información del test visitar el link en la referencia.	118
B.5. Dos configuraciones posibles de los componentes del módulo remoto.	119
C.1. Formas de onda del voltaje y corriente para funcionamiento con LVDS o sin LVDS (imagen extraída de la página 11 de [30]). . . .	122
C.2. Diagrama de la comunicación de dos integrados RHD2132 con un microcontrolador si se utiliza LVDS (imagen extraída de la página 14 de [30]).	123
D.1. Jumper para tener disponible la función de activación de electrodos. En esta versión del sistema no se va a realizar la activación de electrodos, por lo tanto este jumper va siempre conectado donde indica la figura. Si se desea implementar la activación de electrodos, el valor de VESD se impone con el microcontrolador a través del pin 14 del conector MSP3.	126
D.2. Conexión del conversor DC/DC ADP150.	128
D.3. Conexión del conversor DC/DC TPS736.	129
D.4. Esquemático de los conectores CON1_ELEC y CON2_ELEC. Los números del 0 al 63 a los que se conecta corresponden a las señales de EEG a medir.	131
D.5. Esquemático de los conectores JP1 a JP9.	131
D.6. Esquemático del conector MSP3.	131
D.7. Esquemático de los conectores JP1 a JP9.	131
D.8. Esquemático de los dos integrados RHD2132.	132
D.9. Esquemático de los dos integrados RHD2132.	132
E.1. Esquemático del circuito para la medida de impedancia de electrodos [30].	133
E.2. Onda de voltaje generada variando el valor del registro ZcheckDAC.	134
F.1. Calculadora de impedancia característica.	137
G.1. Diagrama de flujo de la función <code>send_word(destinatario)</code>	140
G.2. Diagrama de flujo de las rutinas de interrupción de los puertos SPI que se comunican con el AFE. El paralelogramo indica que la acción en el chip 0 y el chip 1 es distinta según el caso.	141
G.3. Diagramas de flujo de <code>CONVERTburst()</code> y <code>transfer_channel()</code> . . .	142
G.4. Diagrama de flujo de la función <code>CONVERTburst()</code>	143
G.5. Estructura de una cola circular con <i>CANTPRIO</i> prioridades de tamaño <i>TAMCOLAS</i>	144
G.6. Estructura de un tipo de dato <code>obj.t</code> que se encarga de controlar la cola circular de la figura G.5.	145
G.7. Diagrama de flujo de la función <code>radio_init()</code>	149
G.8. Diagrama de flujo de la función <code>radio_receiveData</code>	150
G.9. Diagrama de flujo de la función <code>radio_sendData()</code>	151

H.1.	Menú para indicar al PC con que módulo remoto se debe iniciar la comunicación.	155
H.2.	Menú para setear el uso del algoritmo de compresión.	155
H.3.	Menú para indicar la configuración de canales para la próxima adquisición a realizar.	156
H.4.	Menú para indicar cuantas señales de sincronismo se desean muestrear en la próxima adquisición a realizar.	156
H.5.	Menú para elegir el uso de la eliminación de offset.	156
H.6.	Menú para elegir la frecuencia de corte superior.	157
H.7.	Menú para elegir la frecuencia de corte inferior.	157
H.8.	Menú para elegir el valor de distorsión para el algoritmo de compresión ACEEG.	157
I.1.	Envío de un paquete a la radio a una configuración de 64 canales sin sincronismo. En ambas figuras, la señal superior muestra el estado de la transmisión de un paquete hacia la radio (1- Enviando, 0- No enviando), en la señal de abajo se muestra CS.	162
I.2.	En la figura se muestra una senoide de $5Hz$ de frecuencia y $5mV_{pp}$ de amplitud, muestreada a $850Hz$ con configuración 64 canales y sin sincronismo, con un tiempo de ventana de $10s$. En la misma se muestra el patrón de pérdidas periódico debido al problema de control de flujo en <code>buffer_packet</code> . Las pérdidas corresponden al tiempo de las señales graficadas se mantienen constantes en $0V$	166
I.3.	Circuito para medir el consumo del módulo remoto.	167
I.4.	Modelo teórico de la corriente que consumo el módulo remoto para la configuración de 64 canales sin sincronismo.	168

Esta es la última página.
Compilado el domingo 19 noviembre, 2017.
<http://iie.fing.edu.uy/>

wEEG: Electroencefalógrafo inalámbrico de bajo consumo

Proyecto de grado de Ing. Eléctrica

Integrantes: Martín Causa, Santiago Radi, Franco La Paz

Tutor: Julián Oreggioni



Figura 1 - Adquisición de un EEG con el sistema wEEG

Especificaciones

- 64 canales diferenciales
- 6 canales de sincronismo
- Frecuencia de muestreo desde 100Hz hasta 10kHz
- Alcance amplio (en una sala de 5x4mts)
- Duración > 12 Hs
- Liviano y cómodo para transportar
- Electrodo secos

Adquisición

- 2 chips RHD2132 de *Intan Technologies*
- Implementan el muestreo, filtrado, amplificación, digitalización de las señales EEG (32 CH c/u)
- Se diseñó y fabricó una placa (PCB) de 4 capas. Que cumple con los exigentes requerimientos del sistema

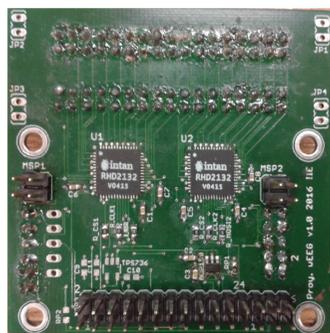


Figura 3 - PCB de la placa adquisidora de señales EEG.

Resultados

Como producto final, se obtuvo un prototipo capaz de:

Especificaciones	Datos del sistema
Autonomía	24hs, 22min
Peso	270 g
Tamaño	11,70 x 7,88 x 4,76 cm
Consumo	115 mA @ 64CH, 770Hz - 55 mA @reposito
Frecuencia de muestreo mínima/máxima	100 Hz/770 Hz @64CH - 9850 Hz @1CH
Alcance	>12mts

Este proyecto fue parcialmente financiado por ANII y CSIC.

Características generales

¿Qué es un EEG?

Un electroencefalograma es un registro de la actividad fisiológica cerebral. Este registro es utilizado en medicina, por ejemplo para el estudio del sueño o en pacientes epilépticos.

¿Qué ventajas presenta wEEG?

- Conexión inalámbrica
- Bajo consumo
- Cómodo para ser usado

Estas ventajas permiten:

- Mayor comodidad
- Libertad de movimiento al paciente/usuario
- Estudios más prolongados
- Mayor diversidad de estudios con EEG
- Nuevas aplicaciones con EEG

Diseño

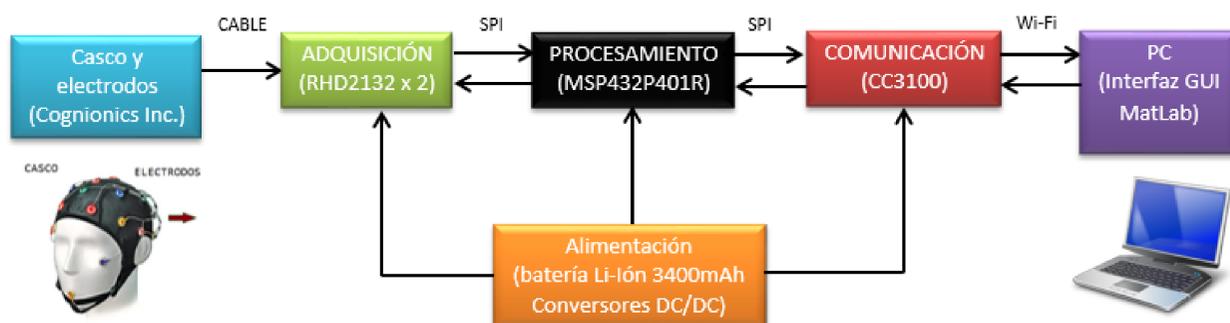


Figura 2 - Diagrama de bloques del sistema wEEG

Procesamiento

- Microcontrolador MSP432p401r de *Texas Instruments*
- Se encarga del procesamiento y control de flujo de la información

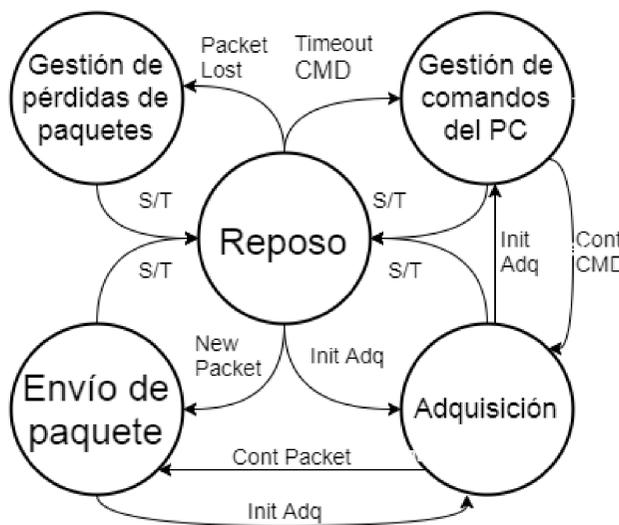


Figura 4 - Diagrama de estados del sistema embebido en el microcontrolador MSP432P401R.

Comunicación

- Radio Wi-Fi CC3100 de *Texas Instruments*
- Implementa el protocolo TCP-IP
- Se comunica por Wi-Fi con el PC

Interfaz de usuario

- Diseñada en *MATLAB*
- Visualización a gusto de señales de interés
- Envío de comandos de configuración al equipo ubicado en el paciente

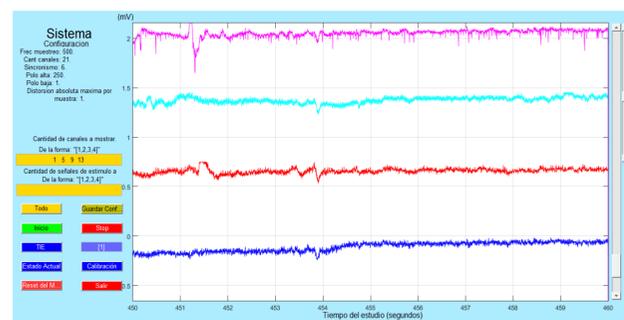


Figura 5 - Ventana gráfica de la interfaz de usuario wEEG.



Figura 6 - Montaje del módulo remoto.

A 64-channel wireless EEG recording system for wearable applications

Martín Causa, Franco La Paz, Santiago Radi, Juan P. Oliver, Leonardo Steinfeld and Julián Oreggioni.

Abstract—A wireless 64-channel EEG (Electroencephalography) recording system for wearable applications is presented. The aim of this system is to allow the patient to move freely for a reasonable time in a short distance environment, in order to extend the field of application of traditional electroencephalography studies. The system consists of a wireless module located in the patient and a user interface that runs on a PC (Personal Computer). The wireless module is responsible for acquiring EEG signals (amplify, filter and digitize), processing and sending them wirelessly to the PC.

The system is capable of acquiring up to 64 EEG signals and 6 synchronism signals for more than 24 hours, with a programmable sampling frequency between 100 Hz and 10 kHz. The system supports four types of system configuration (1, 4, 21 or 64 channels), with a programmable bandpass filter where high-pass frequency can vary between 0.1 Hz and 500 Hz and the low-pass frequency can vary between 100 Hz and 20 kHz. The analog front-end has an input impedance of $1.3G\Omega$, an input-referred noise of $2.4 \mu V_{rms}$ and a common mode rejection ratio of 82 dB. The maximum distance measured between the wireless module and the PC was 12 meters.

I. INTRODUCTION

The advance of neuroscience and electronics in recent years has made it possible to develop wireless devices for recording EEG signals in a diverse range of applications outside the medical clinic. For instance, fatigue monitoring [1], mind-controlled video games [2], market research [3], among others. This makes it feasible to imagine the development of new EEG (Electroencephalography) devices for a large number of novel applications.

A wireless 64-channel EEG recording system for wearable applications is presented. The aim of our work is to allow the patient to move freely for a reasonable time (about 1 day) within a short distance (about ten of meters), in order to extend the field of application of traditional EEG. This work is part of a bigger project that consist of implementing a compression algorithm [4] in the recording system presented here.

There were and there are multiple efforts in this direction. For example [5], [6] present systems that attain very low power consumptions using very specialized custom hardware, and at the expense of other features we designed for, such as transmission range, robust and secure transmission protocols and mobility, among others. On the other hand,

This work was partially funded by CSIC-UDELAR (Comisión Sectorial de Investigación Científica, Universidad de la República, Uruguay), ANII (Agencia Nacional de Investigación e Innovación, Uruguay) and CAP-UDELAR (Comisión Académica de Posgrado, Universidad de la República, Uruguay).

Martín Causa, Franco La Paz, Santiago Radi, Juan P. Oliver, Leonardo Steinfeld and Julián Oreggioni are with the Facultad de Ingeniería, Universidad de la República. Montevideo, Uruguay.

there are works such as [7] where a system based on standard hardware is presented, and there are several commercial systems [8]–[11]. A comparison of the latter with our proposal is presented at the end of this work.

II. WIRELESS EEG RECORDING SYSTEM

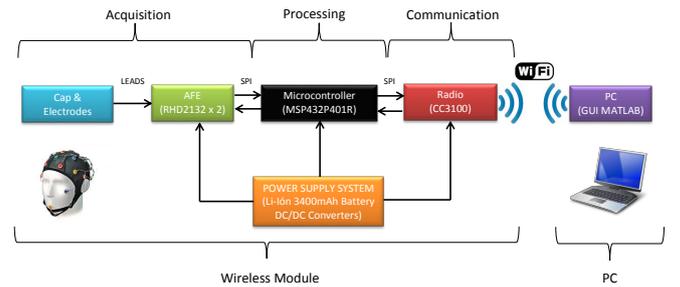


Fig. 1. Block diagram of the proposed system.

The proposed solution, depicted in Fig. 1, consists of a wireless module located in the patient and a graphical user interface (GUI) that runs on a PC (Personal Computer). The wireless module is composed of four blocks. The first block implement the EEG signal acquisition, the second one is in charge of the data processing and the third one is responsible for the wireless communication with the PC. Finally, a power supply system block that comprises a 3400 mAh 3.7 V Li-ion rechargeable battery and two dc/dc converters. One linear dc/dc converter power the analog part of the system and one switched dc/dc converter power the rest of the wireless module. The output of both dc/dc converters is $V_{DD} = 3.3 V$.

At the beginning of the test the user selects the desired system configuration, which implies setting the high-pass (f_{HP}) and low-pass (f_{LP}) frequency, the sampling frequency (f_S), the number of channels to be sampled (N_C , that can be selected from four predefined standard electrode arrangement: 1, 4, 21 or 64 channels) and the number of synchronism channels (N_S , up to 6). The synchronism channels allow to record the reaction to a stimulus synchronized with the EEG signals (for example the pressing of a button in front of a visual stimulus to analyze the reaction time). Next, the user gives the command to start in the GUI and the EEG test begins. The data acquisition block samples $N_C + N_S$ channels every $T_S = 1/f_S$ and sends them (via a Serial Peripheral Interface, SPI) to the data processing block, which in turn processes the data and sends them (via SPI) to the data communication block. Finally, this block sends the acquired data (via WiFi) to the PC, where the user can view them in real time.

A. Data acquisition block

The data acquisition block comprises electrodes, a cap, an analog front-end (AFE), and an analog-to-digital converter (ADC). Dry electrodes from Cognionics were used: Flex Sensor to acquire through hair and Drypad Sensor to acquire in skin. The AFE and ADC were implemented using two off-the-shelf RHD2132 chips from Intan Technologies. Each RHD2132 chip is in charge of acquire, amplify, digitize and transmit via a SPI up to 32 channels at 30 kbps each. The RHD2132 chip features an input impedance of $1.3\text{ G}\Omega$, a common mode rejection ratio (CMRR) of 82 dB, low input referred noise ($2.4\mu\text{V}_{\text{rms}}$), programmable bandwidth and low power operation. For instance, the high-pass frequency can be set between 0.1 Hz and 500 Hz and the low-pass frequency can vary from 100 Hz to 20 kHz. In addition, the total current consumption of the two chips to acquire 64 channels at 500 sps/ch is 1.8 mA and at 1 kbps/ch it is 2.1 mA.

A four-layer Printed Circuit Board (PCB) supporting the RHD2132 chips and the dc/dc converters was designed and fabricated. The top and bottom layers of this PCB (first and fourth layers) were used for signal routing. The second layer was a ground plane and the third layer was a V_{DD} plane.

B. Data processing block

The data processing block is based on the MSP432P401R microcontroller (MCU) from Texas Instruments. This MCU is a 32-bit ARM Cortex-M4F microcontroller with a maximum clock frequency of 48 MHz, with 256 kB of Flash and 64 kB of RAM memory. This chip features a typical power consumption of 4.6 mA in Active Mode and offers several modes of low-power operation, called Sleep Mode, where its power consumption can be as low as hundreds of nanoamperes. In addition, this MCU includes a rich set of peripherals including several SPI ports and timers.

The main functions of the MCU embedded software are: receive the sampled data from the RHD2132 chips, process this data and forward them to the WiFi radio module. In addition, the MCU is responsible for parsing commands received from the PC.

A round-robin with interrupts architecture is adopted, where interrupt service routines (ISR) are extensively used to exchange (transmit and receive) data, and keep the MCU in Sleep Mode while no processing is needed. The ISR in turn use flags to signal in the main loop whether extra processing is needed. If no further processing is needed, the MCU is put in Sleep Mode.

The MCU acts as master in the SPI communication with the RHD2132 chips and the WiFi radio module. MCU activates the chip select signal (\overline{CS}) to select the chip and provides the clock frequency (CLK). Two different SPI ports of the MCU are used to communicate with the RHD2132 chips. Firstly, to parallelize the data flow, and also to configure the two chips separately. An additional SPI port is used to communicate the MCU with the WiFi radio module.

Two MCU timers are used. One is employed to set the sampling frequency f_S . When this timer expire, the MCU

triggers a new acquisition by strobing commands to the RHD2132 chips according to the system configuration and stores the received data into an input buffer. The second timer is used to periodically poll the WiFi radio module input buffer to check for incoming commands from the PC.

C. Data communication block

The data communication block is implemented with a WiFi radio module based on the CC3100 chip from Texas Instruments. The WiFi radio module sends the acquired data to the PC, receive commands from the PC as well as exchange configuration parameters. The SimpleLink library provided by the vendor (Texas Instruments) is used to communicate with the WiFi radio module.

The communication technology was selected considering the system requirements, specially maximum data rate, power consumption and communication range to ensure connectivity within the required area. The maximum effective throughput required by our application correspond to acquiring all channels ($N_C = 64$, $N_S = 6$) at the highest data sample rate ($f_S = 1\text{kHz}$) and with the maximum resolution (16 bits per sample), resulting in 1120 kbps.

WiFi was chosen, which despite of being a technology with relative high power consumption, it meets our requirements and gives us the chance to scale. In addition, WiFi is widely adopted enabling an almost straightforward integration. The application was build over TCP since it provides a reliable data stream.

D. User interface

The software on the PC side runs an application and the corresponding GUI, which allows the user to perform an EEG test. The software was developed in Matlab and is in charge of configuring the wireless module, storing the received data and displaying them in real time. The GUI allows the user to configure: high-pass and low-pass frequency of all channels, the system sampling frequency, the number of channels to be sampled and the number of synchronism channels.

When the user starts the application, it tries to establish a connection to the wireless module, loads the default system configuration and shows the main menu. The user may modify the system configuration selecting new parameter values. Then it sends the new configuration to the wireless module, followed by a command to get the current configuration to verify that the wireless module received it correctly. Finally it shows a new window that allows the user to perform the EEG test (visualize the data and so on).

E. Installation

Fig. 2 shows how the system is mounted on the body of the patient. The electrodes attached to the head are connected directly to a case fixed on the patient's neck (left), or through an extension cable if the case is attached to the patient's waist (right).

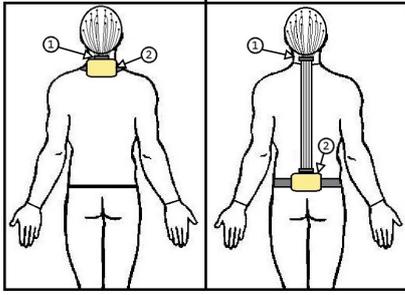


Fig. 2. Installation of the system in the patient. References: (1) electrodes connection (left: directly, right: though an extension cable), (2) electronic case (yellow box).

III. SYSTEM TEST

This section shows and analyzes the results of the several test performed to the proposed system.

A. Wireless communication

The system requirements imposes a minimum throughput for the payload data of 1120 kbps (see Subsection II-C). The maximum measured data throughput, at 12 m of distance between the wireless module and the router, was 5480 kbps. Although the data throughput was not measured for distances greater than 12 meters, the system has the potential to operate in a greater range.

B. Acquisition timing

In order to start an acquisition the MCU must establish a communication with both RDH2132 chips, which we will call AFE0 and AFE1. Two possible configurations were analyzed: request all channels to AFE0 first, and then all channels to AFE1 (serial configuration); or, request all channels in parallel to both chips (parallel configuration).

The measured acquisition time for 64 channels ($N_C = 64$ and $N_S = 0$) in the serial configuration was 1.03 ms, and in the parallel configuration was 1.05 ms. Although at first glance it is expected that the parallel configuration would be faster, since there is only one processor, it can not attend to both interruptions at the same time, and it attends first the interruption of the AFE0 because it has higher priority. It can be observed in Fig. 3 that during T_1 both SPI buses are active in interleaved form, but AFE0 is more busy because it has higher priority, and ends before. Then, during T_2 only AFE1 is attended until it finished its remaining channels.

Due to this similarity between the serial and parallel configuration delays, it is decided to use the serial one, since it allows a simpler and more controlled management of the embedded software.

C. System Tests

1) Maximum sampling frequency vs. number of channels:

The maximum frequency f_S for which the system works without data losses for different system configurations was measured carrying out a 10-minute test. The Table I shows the recorded frequency values.

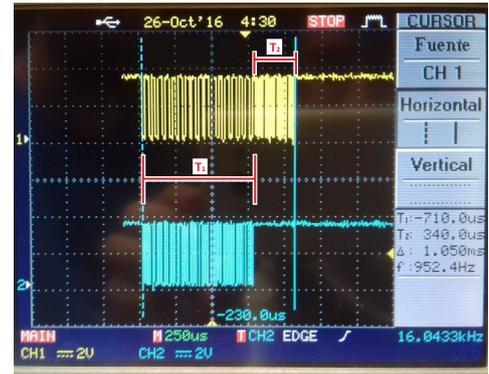


Fig. 3. Parallel configuration. Above (in yellow): \overline{CS} of AFE1. Below (in sky-blue): \overline{CS} of AFE0.

TABLE I
MAXIMUM SAMPLING FREQUENCY.

N_C	N_S	$f_{S_{max}}$ (Hz)
64	0	800
64	6	730
21	0	2160
21	6	1690
4	0	5970
4	6	3380
1	0	10300

The main limitation on the maximum attainable frequency is given by the time of acquisition of a *run* (*run* refers to the sampling of $N_C + N_S$ channels, the data of a *run* correspond to the same instant of time). For almost all system configurations it is used more than 90% of the time to acquire samples, leaving less than 10% to perform other tasks, mainly sending packages to the WiFi radio module. Therefore, a reduction of the acquisition time of a *run* would improve the maximum frequency currently reached by the system.

2) *Battery run-time measurements*: The wireless module was powered with a 3400 mAh Li-ion rechargeable battery. A test with this parameters: $N_C = 64$, $N_S = 0$ and $f_S = 770$ Hz, was left running until the battery could not maintain the minimum voltage required by the RHD2132 chips (3.2 V). The test lasted 24 hours and 22 minutes.

3) *Full-system test*: At the inputs of the AFE, known signals were injected for different system configurations all with an amplitude of 10 mV_{PP}, and choosing $f_{LP} = 20$ kHz and $f_{HP} = 0.1$ Hz, and compared with the registered data.

In Fig. 4 the signal is within the amplifier band-pass and an attenuation of 3.7% is observed (the amplitude of the signal is 9.63 mV_{PP}). This small attenuation can be explained by the fact that the input signal amplitude equals the maximum input linear range of the AFE, which is 10 mV_{PP}.

In the Fig. 5 a sawtooth with a damping at the ends is observed, this is due to the action of the filter in the AFE.

Fig. 6 presents an actual EEG test performed with our system.

IV. CONCLUSIONS

The wireless EEG recording system developed is capable of acquiring up to 64 EEG signals and 6 synchronization

TABLE II
COMPARISON WITH COMMERCIAL EEG WIRELESS RECORDING SYSTEMS (DATA EXTRACTED FROM [12]).

System	Our system	Nicolet	g.Mobilab+	eego rt	HD-72	NIRS-EEG
Number of channels	64	64	8	64	64	8
Weight (grams)	270	800	360	500	350	800
Autonomy (hours)	24	12-24	25-100	5	6	33
Effective data rate per channel (samples/s)	770	4k	256	2k	500	320
Input-referred noise	$2.4\mu V_{rms}$	$2\mu V_{PP}$	-	$1\mu V_{rms}$	$0.7\mu V_{rms}$	-
Communication protocol	WiFi	WiFi	Bluetooth 2.0	WiFi	Bluetooth	Bluetooth
Manufacturer	-	Natus	g.tec	ANT Neuro	Cognionics	-
Reference	-	[8]	[9]	[10]	[11]	[7]

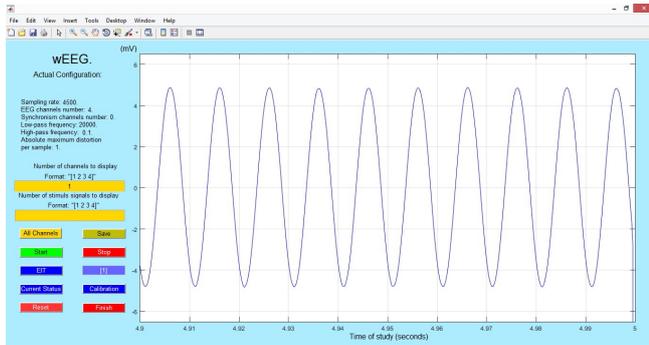


Fig. 4. Input: sinusoid of 100 Hz.

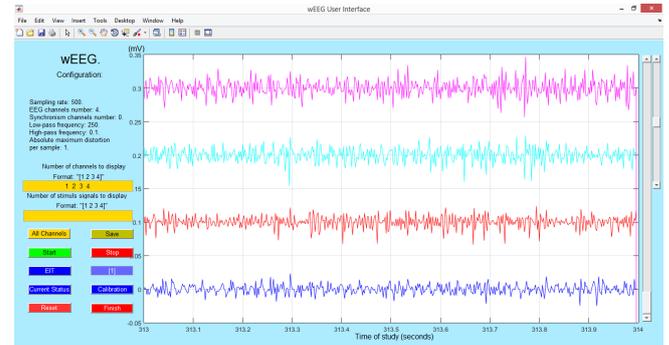


Fig. 6. EEG test performed with our system.

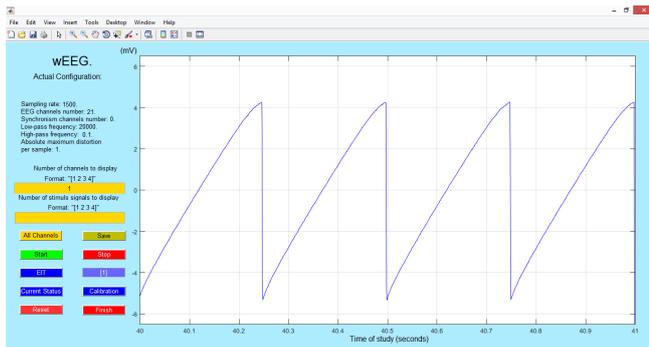


Fig. 5. Input: sawtooth of 4 Hz.

signals, with a programmable sampling frequency between 100 Hz and 10 kHz, and with an autonomy of more than 24 hours. The system supports four types of system configuration (1, 4, 21 or 64 channels), with a programmable bandpass filter where high-pass frequency can vary between 0.1 Hz and 500 Hz and the low-pass frequency can vary between 100 Hz and 20 kHz. The analog front-end has an input impedance of $1.3 G\Omega$, an input-referred noise of $2.4 \mu V_{rms}$ and a CMRR of 82 dB. The maximum distance measured between the wireless module and the PC was 12 meters (but the system has the potential to operate in a greater range).

Table II shows a comparison between our system and some available commercial EEG recording devices. It is observed that our system compares favorably, being the only solution that offers the acquisition of 64 channels with an autonomy greater than 24 hours.

REFERENCES

[1] Smartcap, "Life: where fatigue is the problem, Life is the solution."

<http://www.smartcaptech.com/life-smart-cap/>, Oct 2017. [On line].

- [2] Uncle Milton Industries, "THE FORCE TRAINER II: HOLOGRAM EXPERIENCE." <http://starwarsscience.com/product/the-force-trainer-ii-hologram-experience/>, Nov 2017. [On line].
- [3] Neurosky, "EEG Biosensors." <http://neurosky.com/biosensors/eeg-sensor/>, Nov 2017. [On line].
- [4] G. Dufort, F. Favaro, F. Lecumberry, A. Martin, J. P. Oliver, J. Oreggioni, I. Ramirez, G. Seroussi, and L. Steinfeld, "Wearable EEG via lossless compression," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1995–1998, Aug 2016.
- [5] S. D. Pascoli, D. Puntin, A. Pinciarioli, E. Balaban, and M. Pompeiano, "Design and implementation of a wireless in-ovo EEG/EMG recorder," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 7, pp. 832–840, Dec 2013.
- [6] Y. Zhang, F. Zhang, Y. Shakhsher, J. D. Silver, A. Klinefelter, M. Nagaraju, J. Boley, J. Pandey, A. Shrivastava, E. J. Carlson, A. Wood, B. H. Calhoun, and B. P. Otis, "A batteryless 19 μW MICS/ISM-band energy harvesting body sensor node SoC for ExG applications," *IEEE Journal of Solid-State Circuits*, vol. 48, pp. 199–213, Jan 2013.
- [7] M. Sawan, M. T. Salam, J. L. Lan, A. Kassab, S. G elinas, P. Vannasing, F. Lesage, M. Lassonde, and D. K. Nguyen, "Wireless recording systems: From non-invasive EEG-NIRS to invasive EEG devices," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 7, pp. 186–195, April 2013.
- [8] Natus Medical Inc., "Nicolet EEG wireless amplifier. Rev 05." <https://photos.medwrench.com/equipmentmanuals/5989-3568.pdf>, Oct 2017. [On line].
- [9] g.tec, *Advanced biosignal acquisition, processing and analysis. Product catalogue*, ch. g.MOBilab: wireless biosignal acquisition system in your pocket, pp. 25–28. Oct 2017. [On line].
- [10] ANT Neuro, "eego rt: high-density EEG solution with real-time data access.." http://www.ant-neuro.com/sites/default/files/150112_older_eego_rt_A4.pdf, Oct 2017. [On line].
- [11] Cognionics, "72-Channel dry EEG Headset system, (HD-72)." <http://www.cognionics.com/index.php/products/hd-eeg-systems/72-channel-system>, Oct 2017. [On line].
- [12] J. Oreggioni, A. A. Caputi, and F. Silveira, "Biopotential monitoring," *In Press. Encyclopedia of Biomedical Engineering. Elsevier's Biomedical Sciences Reference Module*, 2018.