



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

Curso Académico:

Memoria

AGRADECIMIENTOS

A mi tutor, Julio, por toda su ayuda y por todo lo aprendido y mejorado durante el desarrollo de este proyecto.

A Javier Saiz, por ofrecerme esta oportunidad de trabajo y su ayuda desinteresada en este proyecto.

A mis amigos, por hacer que los momentos duros sean menos duros y los buenos sean todavía mejores.

A mi familia, por su continuo apoyo incondicional, sus ánimos y su confianza.

A Marta, por estar siempre a mi lado y ayudarme siempre que lo he necesitado.

RESUMEN

Las patologías cardíacas son un destacado problema de salud en la actualidad, por ello los instrumentos para estudiar el estado del corazón tienen una gran importancia. En la actualidad existe una tendencia de incorporar dispositivos que puedan monitorizar de forma remota al sujeto, interfiriendo mínimamente en su actividad y obteniendo la mejor calidad de información posible.

Con la aparición de esta tendencia, los fabricantes han desarrollado circuitos integrados que incluyen las etapas fundamentales de la captación de las señales biomédicas. De esta forma se reduce el tamaño, peso y complejidad de estos dispositivos, facilitando su diseño y funcionamiento. Además, mejoran el consumo, maximizando su autonomía y permitiendo obtener más información.

Por todo ello, en este proyecto se estudia el diseño y desarrollo de un dispositivo de monitorización cardíaca utilizando el ADAS1000 que sirve de front-end para el ECG. Asimismo, se analiza cómo integrar y comunicar este dispositivo con otros que puedan servir para estudiar la señal de interés.

El dispositivo empleado en este proyecto, el ADAS1000 de Analog Device, es un circuito integrado que sirve de etapa de entrada (front-end) analógico para la captación de señales bioeléctricas. Este integrado tiene las entradas necesarias para conectar los electrodos o sensores. Dichas entradas pasan a un amplificador diferencial con ganancia programable, un convertidor analógico digital de aproximaciones sucesivas (SAR) de 14 bits y un filtro digital. Este dispositivo también incorpora otras funciones como la detección de electrodos sueltos. La ventaja que presenta es su bajo coste para un sistema complejo (ECG, respiración...), su bajo consumo y sus pequeñas dimensiones.

Palabras Clave: ECG, ADAS1000, Analog Devices, Microchip, Microcontrolador, PIC24FJ128GC006, monitorización remota.

RESUM

Les patologies cardíques són un destacat problema de salut en l'actualitat, per això els instruments per estudiar l'estat del cor tenen una gran importància. En l'actualitat existeix una tendència d'incorporar dispositius que puguin monitoritzar de forma remota al subjecte, interferint mínimament en la seua activitat i obtenint la millor qualitat d'informació possible.

Amb l'aparició d'aquesta tendència, els fabricants han desenvolupat circuits integrats que inclouen les etapes fonamentals de la captació dels senyals biomèdiques. D'aquesta manera es redueix la grandària, pes i complexitat d'aquests dispositius, facilitant el seu disseny i funcionament. A més, milloren el consum, maximitzant la seua autonomia i permetent obtenir més informació.

Per tot això, en aquest projecte s'estudia el disseny i desenvolupament d'un dispositiu de monitorització cardíaca utilitzant el ADAS1000 que serveix de front-end per l'ECG. Així mateix, s'analitza com integrar i comunicar aquest dispositiu amb altres que puguin servir per estudiar el senyal d'interès.

El dispositiu emprat en aquest projecte, el ADAS1000 d'Analog Device, és un circuit integrat que serveix de front-end analògic per a la captació de senyals bioelèctrics. Aquest integrat té les entrades necessàries per connectar els elèctrodes o sensors. Aquestes entrades passen a un amplificador diferencial amb guany programable, un convertidor analògic digital de tipus SAR de 14 bits i un filtre digital. Aquest dispositiu també incorpora altres funcions com la detecció d'elèctrodes per connectar. L'avantatge que presenta és el seu baix cost per a un sistema complex (ECG, respiració...), el seu baix consum i les seues petites dimensions.

Paraules clau: ECG, ADAS1000, Analog Devices, microchip, microcontrolador, PIC24FJ128GC006, monitorització remota.

ABSTRACT

Cardiac pathologies are a prominent health problem in modern life, so the mechanisms to study the state of the heart are of great importance. Nowadays the trend is to incorporate devices that can monitor the patients' heart remotely, minimally interfering with their activity and obtaining the best quality of information possible.

So that, manufacturers have developed integrated circuits that include the fundamental stages of capturing biomedical signals. This reduces the size, weight and complexity of these devices, facilitating their design and functioning. In addition, they improve consumption, maximizing their autonomy and allowing to obtain more information.

Therefore, this project studies the design and development of a cardiac monitoring device using the ADAS1000 that serves as a front-end for the ECG. Furthermore, it also analyzes how to integrate and communicate this device with others that can serve to study the signal of interest.

The device used in this project, the ADAS1000 of Analog Device, is an integrated circuit that serves as an analog front-end for the capture of bioelectric signals. This integrated has the necessary inputs to connect the electrodes or sensors. The inputs pass to a differential amplifier with programmable gain, a digital analog converter of SAR type of 14 bits and a digital filter. This device also incorporates other functions such as the detection of loose electrodes. Its advantages are low cost, low consumption and small dimensions.

Keywords: ECG, ADAS1000, Analog Devices, Microchip, Microcontroller, PIC24FJ128GC006, remote monitoring.

Contenido

1. Introducción	1
2. Objetivos	3
3. Contextualización	5
3.1. El corazón	5
3.2. El electrocardiograma	8
3.3. Los electrodos	10
3.4. Características de la señal	11
3.5. Procesamiento de señal	13
3.6. Equipo de adquisición	15
3.7. Seguridad eléctrica	15
4. Front-end ADAS1000	17
4.1. Estructura y funcionamiento ADAS1000	18
5. Diseño del dispositivo	23
5.1. Estructura	23
5.2. Elección de componentes principales	23
5.3. Diseño de la placa	25
5.2. Consumo	30
5.3. Montaje de la placa	32
5.3.1. Lista de materiales	32
5.3.2. Proceso de montaje	34
6. Firmware	35
6.1. Interfaz PIC-ADAS1000	35
6.2. Firmware PIC y comunicación bluetooth	43
6.3. Funcionamiento del dispositivo	44
6.4. Software e interfaz grafico	45
6.5. Funcionamiento de la interfaz grafico	45
7. Procesamiento digital de la señal	51
8. Resultados	53
8.1 Medidas	56
9. Conclusiones	57
10. Líneas futuras	59
11. Referencias	61
1. Características del proyecto	65

2. Costes.....	67
2.1 Costes de los Componentes y materiales del dispositivo	67
2.2 Horas de trabajo.....	69
2.3 Coste de los materiales utilizados del diseño y montaje del dispositivo	69
2.4 Presupuesto total.....	70

ÍNDICE DE FIGURAS

Figura 1. Polar H10 & Polar Beat [4], medidor de frecuencia cardiaca.....	1
Figura 2. Visualización del sistema global	3
Figura 3. Estructura del corazón [5].	5
Figura 4. Fases del ciclo cardiaco, sístole y diástole [6].	6
Figura 5. Principales canales y bombas de una célula cardiaca [8].....	7
Figura 6. Sistema de conducción eléctrica del corazón [9].	7
Figura 7. Derivaciones periféricas de Einthoven [8]. Triangulo de Einthoven	9
Figura 8. Circuito de las derivadas aumentadas [8].	10
Figura 9. Colocación de los electrodos para las derivaciones precordiales [8].....	10
Figura 10. Modelo equivalente del interfaz piel electrodo [12].	11
Figura 11. Izquierda, actividad eléctrica de cada parte del corazón. Derecha señal de electrocardiograma normal [8].	12
Figura 12. Espectro bioseñales [13].	14
Figura 13. Etapas fundamentales de un ECG.	15
Figura 14. Esquema de funcionamiento del ADAS1000 [14].	17
Figura 15. Esquema propuesto por Analog Devices para el del ADAS1000.....	18
Figura 16. Esquema de la estructura de los canales de los electrodos del ADAS1000 [14].....	19
Figura 17. Estructura modo común del ADAS1000 [14].	19
Figura 18. Estructura centro de terminal de Wilson del ADAS1000 [15].	20
Figura 19. Simplificación de la detección de electrodo suelto en alterna del ADAS1000 [14]. ...	21
Figura 20. Diagrama de flujo del algoritmo de detección de artefactos del ritmo.....	22
Figura 21. Diagrama de bloques funcionales del dispositivo a diseñar	23
Figura 22. Recomendación del fabricante de la colocación del RN42 [16].	26
Figura 23. Diagrama de voltaje para los niveles digitales	26
Figura 24. Esquema de la PCB simplificado. Líneas rojas corresponden a la parte analógica. ...	27
Figura 25. Esquema propuesto por el fabricante para los condensadores de desacoplo del PIC24FJ128GC006 [17].	28
Figura 26. Esquema propuesto por el fabricante del MCP73833 [18].	29
Figura 27. Esquema representativo de una comunicación SPI en grupo.....	30
Figura 28. Placa del dispositivo	34
Figura 29. Visualización general de la interfaz.	45
Figura 30. Bloque de conexión	46

Figura 31. Bloque de resultados.....	46
Figura 32. Mensaje de espera.	47
Figura 33. Visualización de la señal.....	47
Figura 34. Resultados de la captura de datos	47
Figura 35. Guardar resultados.....	48
Figura 36. Derivada 1 con el electrodo RA suelto	48
Figura 37. Derivada 3 con el electrodo RA suelto	49
Figura 38. Mensaje de electrodo suelto.....	49
Figura 39. Diagrama de bloques del funcionamiento del algoritmo de Pan y Tompkins	51
Figura 40. Programa Siow con los datos recibidos del ADAS1000.....	53
Figura 41. Dos canales con la señal cuadrada de la prueba que proporciona el ADAS1000	54
Figura 42. ProSim™ 4, Vital Signs Simulator de FLUKE Biomedical. Vista frontal, panel de control, vista lateral, electrodos que simulan las extremidades.	54
Figura 43. Señal simulada con el ProSim™	55
Figura 44. Señal real, captada realizando movimientos lentos.	55
Figura 45. Señal real, captada realizando un ejercicio de sentadillas.....	56

ÍNDICE DE TABLAS

Tabla 1. Resumen de los componentes del dispositivo	25
Tabla 2. Valores resistencias y condensadores del PIC	28
Tabla 3. Consumo teórico de cada componente principal	31
Tabla 4. Lista de materiales.....	32
Tabla 5. Estructura del registro de entrada al ADAS1000.....	35
Tabla 6. Registro CMREFCTL del ADAS1000.....	35
Tabla 7. Registro FILTCTL del ADAS1000.....	37
Tabla 8. Registro FRMCTL del ADAS1000.....	37
Tabla 9. Registro ECGTL del ADAS1000.....	39
Tabla 10. Registro LOFFCTL del ADAS1000.	41

1. Introducción

La mayor causa de muertes en el mundo durante los últimos 15 años es la cardiopatía isquémica y el accidente cerebrovascular. De hecho, en 2015 causaron 15 millones de muertes en el mundo. Por otro lado, la segunda causa es la enfermedad pulmonar obstructiva crónica (EPOC) causando 3,2 millones de muertes [1].

Aunque la tasa de mortalidad ha disminuido, la prevalencia de factores de riesgo como la hipertensión, la obesidad, hábitos sedentarios y el tabaco siguen siendo un problema. Esto, junto al aumento de la edad de la población y otras enfermedades crónicas, causa una saturación del sistema sanitario a nivel mundial [2].

Para subsanar este problema se debe actuar en los tres niveles de prevención que existen [3]:

- Prevención primaria, es decir, informar de que causa las cardiopatías y promover los hábitos que disminuyen el riesgo.
- Prevención secundaria, es el nivel en el que el proyecto está más centrado, consiste en detectar y aplicar tratamiento a las enfermedades en estados tempranos y tratar la enfermedad desde el principio. Esto se consigue mejorando la tecnología de detección de las cardiopatías.
- Por último, prevención terciaria, es decir, evitar que la enfermedad empeore cuando está instaurada.

Por tanto, la monitorización continua de un paciente puede dar información vital al equipo sanitario para anticiparse en la detección y prevención de cardiopatías, ya que una detección en fases tempranas tiene mejor pronóstico que cuando la enfermedad esta instaurada y los síntomas se manifiestan.

La monitorización del paciente no solamente tiene porque realizarse en un hospital, actualmente existen dispositivos portátiles que monitorizan durante periodos de horas o días, permitiendo además la total movilidad al sujeto. También existen dispositivos para el deporte como, por ejemplo, la marca Polar (Figura 1). Esta marca tiene un dispositivo para entrenamiento de los equipos, su función es mejorar el rendimiento del entrenamiento y también detectar si existe alguna anomalía. En muchos casos una detección precoz de una enfermedad cardiovascular puede ser de vital importancia.



Figura 1. Polar H10 & Polar Beat [4], medidor de frecuencia cardiaca.

2. Objetivos

El objetivo del presente trabajo es el diseño de un dispositivo de adquisición de ECG utilizando un microcontrolador y el circuito integrado ADAS1000 de Analog Devices.

El dispositivo que se va a desarrollar tiene como objetivo mejorar la prevención secundaria de cardiopatías. Es un dispositivo que se utilizará para hacer monitorización continua y pruebas de esfuerzos y poder así estudiar la actividad física de un individuo. Por tanto, también puede ser utilizado para seguir la rehabilitación de un paciente con una cardiopatía.

Este dispositivo se deberá conectar de forma inalámbrica para permitir una completa movilidad al sujeto que utilizará el dispositivo y no intervenir en su actividad física. Todo ello permitirá además un mejor procesamiento y estudio de la señal ya que esta será transferida a otro dispositivo, como un ordenador personal, que tendrá mejores características de computación para procesar la señal.

Los objetivos a seguir en este trabajo son todas las fases del proceso de diseño de un dispositivo de monitorización:

1. Estudiar el funcionamiento del circuito integrado ADAS1000.
2. Diseñar circuito eléctrico, los esquemas eléctricos para el correcto funcionamiento del dispositivo, así como seleccionar los componentes que se utilizarán para obtener los mejores resultados
3. Diseñar la PCB y establecer cómo se montará el dispositivo
4. Programar el firmware, es decir, programar el microcontrolador para poner en marcha el dispositivo
5. Realizar pruebas de funcionamiento, con ello se podrá comprobar que todos los elementos de la placa funcionen y garantizar su correcto funcionamiento global
6. Proponer un procesado digital. La señal enviada por el dispositivo tendrá que ser procesada para su posterior utilización. En esta etapa está previsto que se utilice un ordenador personal, aunque podría ser cualquier dispositivo digital (*smartphone*, tableta electrónica, otro microcontrolador...).

El sistema estará formado por 2 etapas principalmente (figura 2):

1. Firmware y hardware: El sujeto conectado al dispositivo que captará la señal y se enviará por bluetooth.
2. Software: Receptor de esta señal, procesará la señal que servirá para el personal especializado

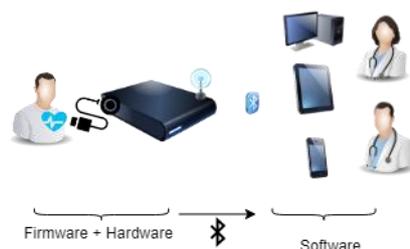


Figura 2. Visualización del sistema global

3. Contextualización

3.1. El corazón

El corazón es el órgano encargado de bombear la sangre que circula por el cuerpo, manteniendo el ritmo y presión adecuada en cada momento. Este órgano se compone de cuatro cámaras (dos superiores o aurículas y dos inferiores o ventrículos) comunicadas dos a dos por las válvulas aurículo-ventriculares (válvula tricúspide o derecha y válvula mitral o izquierda).

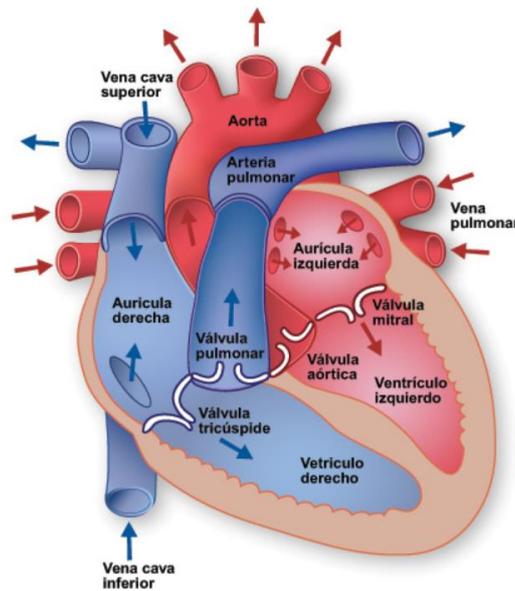


Figura 3. Estructura del corazón [5].

Los ventrículos están separados de las arterias aorta y pulmonar por las válvulas semilunares (aórtica y pulmonar) (figura 3). La sangre fluye ininterrumpidamente hacia las aurículas a través de las venas cava y pulmonares, y sale del corazón a través de la arteria aorta y pulmonar, estableciéndose así un doble circuito llamado sistema cardiovascular.

Existen dos fases principales en el ciclo del corazón: diástole y sístole. La diástole es la fase donde el corazón reposa o está inactivo y se llenan las aurículas. La sístole es la fase donde el corazón se contrae y bombea la sangre al resto del cuerpo (figura 4).

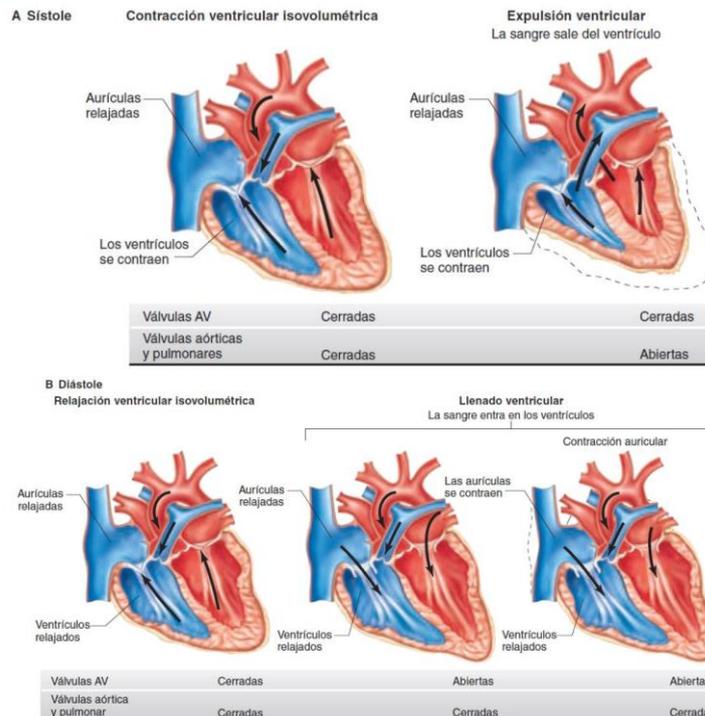


Figura 4. Fases del ciclo cardiaco, sístole y diástole [6].

Al inicio de la diástole, las válvulas AV se relajan dejando pasar la sangre por la diferencia de presión que hay entre ellas y llenando el ventrículo. Éste no se llena del todo, se termina de llenar con la sangre que entra directamente de las venas (diástasis). Por último, las aurículas se contraen para que el resto de sangre pase a los ventrículos.

La siguiente fase es la sístole. La sístole empieza con una contracción ventricular isométrica, es decir sin variación del volumen, a la vez que las válvulas A-V se cierran. Con las válvulas cerradas y la contracción ventricular, la presión aumenta hasta alcanzar un nivel suficiente como para abrir las válvulas aórtica y pulmonar.

Una vez abiertas las válvulas el ventrículo empieza a vaciarse, la sangre se mantiene saliendo por la contracción ventricular (protodiástole). Por último, las fibras ventriculares se relajan, por eso la sangre vuelve al ventrículo ya que la presión en él es inferior a la de la arteria en ese momento. Finalmente, las válvulas semilunares se cierran impidiendo el retorno de la sangre y concluyendo el ciclo cardiaco [7].

Estas contracciones del tejido cardiaco se consiguen gracias a que está formado por células cardiacas, llamadas miocitos, que se activan por una secuencia de despolarización-repolarización de la célula. Esta despolarización-repolarización ocurre por un efecto químico-eléctrico de las células.

Desde un punto de vista fisiológico, se pueden diferenciar dos medios: el medio intracelular y el medio extracelular. Estos medios están compuestos por muchos componentes, entre ellos iones cargados eléctricamente. Estos iones tienen concentraciones diferentes entre el interior de la célula y el exterior. La membrana celular es impermeable y actúa de dieléctrico impidiendo el paso libre de los iones y por tanto genera una diferencia de potencial.

Dependiendo del tipo de células este potencial puede variar, pero en todas las células excitables el potencial de reposo es negativo.

Los iones pueden moverse a través de la célula mediante unos canales que permiten el paso de ciertos iones (figura 5). Además, estos canales tienen un sistema de apertura y cierre que puede depender de muchos factores como el voltaje u otras moléculas del metabolismo del cuerpo como el ATP, entre otros. El movimiento de iones a través de estos canales genera cambios del potencial de acción. Este potencial de acción se puede propagar a la célula siguiente generando más cambios y de esa forma el potencial de acción y la actividad eléctrica se propaga por el corazón.

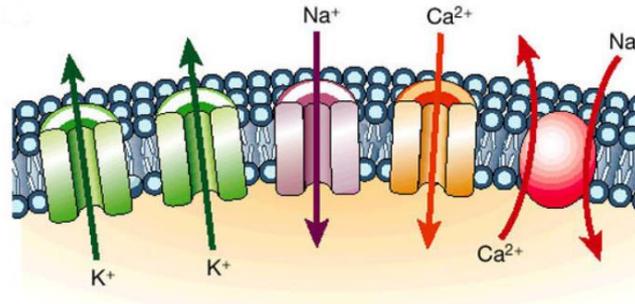


Figura 5. Principales canales y bombas de una célula cardiaca [8].

La mayoría de las células no son autoexcitables, necesitan un estímulo externo para excitarse. Existe un grupo de células especiales en el nodo sinoauricular con un funcionamiento distinto, éstas sí son autoexcitables y son las encargadas de empezar el ritmo cardiaco y controlarlo. Este nodo se sitúa en la pared superior de la aurícula derecha. El nodo SA (figura 6) genera su propio potencial funcionando como marcapasos natural del corazón. Además, se ajusta a la frecuencia que se necesita en cada momento pudiendo variar la frecuencia de latidos del corazón si es necesario.

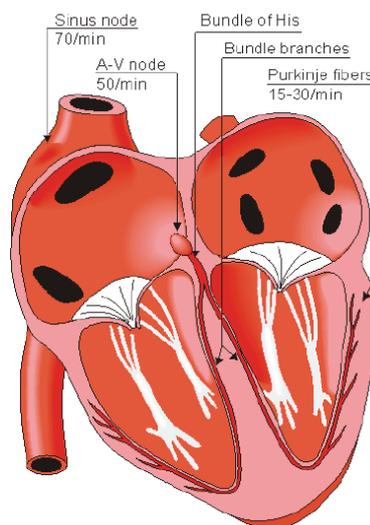


Figura 6. Sistema de conducción eléctrica del corazón [9].

El potencial generado en el nodo SA se propaga hacia las aurículas a través de unas fibras especializadas, las fibras de Purkinje (figura 6), que están especializadas en el transporte de impulsos eléctricos en el ventrículo [10]. Este potencial también se propaga a los ventrículos a través de las Vías Internodales hasta llegar al nodo aurículo-ventricular (nodo A-V), donde hay un retraso en el potencial para provocar un desfase de tiempo entre las sístoles auricular y ventricular. Con ello la contracción de los ventrículos no comienza hasta que las aurículas hayan sido totalmente vaciadas.

El potencial que pasa por el nodo A-V se conecta con el Haz de Hiss, que está formado por un tejido de células de conducción especializadas. El Haz de Hiss se separa en dos ramas que se distribuyen por las paredes del septum. De estas dos ramas aparecen las fibras de Purkinje, la función de estas células especializadas es hacer llegar el potencial a todas las fibras musculares del miocardio simultáneamente provocando la contracción del corazón [7].

3.2. El electrocardiograma

La actividad eléctrica del corazón que se explica en el apartado anterior se puede captar desde el interior o el exterior del cuerpo, esta señal es la que se llama electrocardiograma (ECG). El objetivo de la electrocardiografía, es captar la actividad eléctrica del corazón de forma no invasiva, mediante el uso de unos electrodos en la superficie de la piel, y observar el funcionamiento eléctrico del corazón. Gracias a esta información no sólo es posible detectar ciertas patologías, sino también medir el ritmo cardiaco en otros ámbitos, como el deporte.

Un concepto que guarda gran relación con el ECG es el dipolo cardíaco. El dipolo cardiaco es una expresión que permite calcular el potencial del corazón en cualquier punto. Es un sistema de diagnóstico de interés clínico como la vectocardiografía. Si referimos los dipolos cardiacos correspondientes a todos los instantes de tiempo a un origen común, sus extremos describirán una curva alabeada que recibe el nombre de vectocardiograma. Ésta representa la dirección y magnitud de las fuerzas eléctricas generadas por la actividad del corazón, correspondientes a un ciclo cardiaco. Mediante unos electrodos en la superficie de la piel, se puede captar la proyección del vectocardiograma y por tanto visualizar el vectocardiograma desde distintas perspectivas [7].

Dependiendo de la posición de los electrodos se puede captar distintas posiciones del vector electrográfico y por tanto poder observar distintas actividades del corazón explicadas en el apartado 3.1. Existen una gran variedad de posiciones para colocar los electrodos a lo largo del cuerpo sin embargo sólo se utilizan un número establecido de estas.

Se pueden dividir en tres grupos:

- Las 3 derivaciones estándar: I, II y III (Triángulo de Einthoven) (figura 7).
- Las 3 derivaciones aumentadas: aVR, aVL y aVF (figura 8).
- Las 6 derivaciones precordiales: V1, V2, V3, V4, V5 y V6 (figura 9).

Las tres derivaciones estándar son las más antiguas, fueron las primeras en utilizarse por W. Einthoven a principios del siglo XX. Las derivadas aumentadas utilizan los mismos terminales positivos que las de Einthoven y las de terminal negativo utilizan el punto medio entre las dos extremidades opuestas.

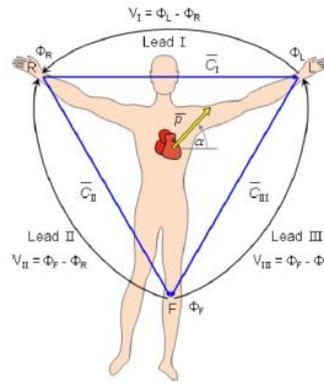


Figura 7. Derivaciones periféricas de Einthoven [8]. Triángulo de Einthoven

Las derivaciones estándar son el resultado de medir el potencial extracelular que se registra en la superficie corporal mediante la combinación de 3 electrodos, uno situado en el brazo izquierdo, otro en el derecho y otro en el pie izquierdo.

De esta forma, cada derivación se puede definir como:

- Derivación I: Es el potencial diferencial entre el electrodo del brazo derecho (negativo) y el brazo izquierdo (positivo).
- Derivación II: Es el potencial diferencial entre el electrodo del brazo derecho (negativo) y el pie izquierdo (positivo).
- Derivación III: Es el potencial diferencial entre el electrodo del brazo izquierdo (negativo) y el pie izquierdo (positivo).
- Derivación aVR: Es el potencial diferencial registrado entre el electrodo del brazo derecho (positivo) y un electrodo posicionado en el “infinito”. Como no se puede situar en el infinito, se coge la tensión promediada del pie izquierdo y del brazo izquierdo.
- Derivación aVL: Es el potencial diferencial registrado entre el electrodo del brazo izquierdo (positivo) y un electrodo posicionado en el “infinito”. Como no se puede situar en el infinito, se coge la tensión promediada del pie izquierdo y del brazo derecho.
- Derivación aVF: Es el potencial diferencial registrado entre el electrodo del pie izquierdo (positivo) y un electrodo posicionado en el “infinito”. Como no se puede situar en el infinito, se coge la tensión promediada del brazo derecho y del brazo izquierdo.

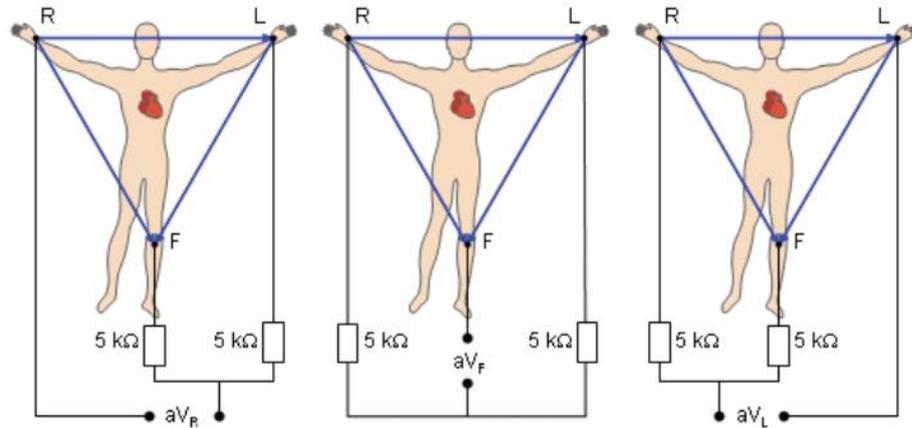


Figura 8. Circuito de las derivadas aumentadas [8].

Las derivaciones precordiales son el potencial diferencial entre el electrodo situado en el tórax (positivo), también llamados V1, V2, V3, V4, V5 y V6 en sus posiciones normalizadas (figura 9), y el terminal negativo estaría en el infinito. En realidad, se coge la tensión promediada del pie izquierdo, del brazo derecho y del brazo izquierdo, punto que se conoce como CTW (Central terminal de Wilson). El CTW sirve de referencia independiente para las derivadas precordiales (V1, V2, V3, V4, V5 y V6).

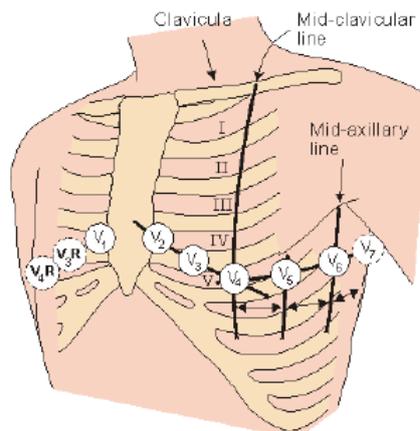


Figura 9. Colocación de los electrodos para las derivadas precordiales [8].

3.3. Los electrodos

Para representar el ECG, es necesario transformar las corrientes iónicas en corriente eléctrica para que el dispositivo pueda procesar la señal, esto se logra mediante electrodos. Los electrodos son conductores eléctricos que se ponen en contacto con las soluciones iónicas acuosas del cuerpo.

La interacción entre los electrones de los electrodos y los iones en el cuerpo puede afectar en gran medida el rendimiento de los electrodos y por lo tanto hay que tenerlo en cuenta a la hora de hacer una media. La interfaz entre un electrodo y una solución iónica provoca

reacciones redox (reducción - oxidación), lo que induce un desplazamiento de carga entre electrodo y solución.

Un metal en contacto con una solución iónica, provoca un cambio local en la concentración de los iones de la solución próxima a la superficie del metal. Esto hace que la neutralidad de carga localmente no se mantenga, modificando el potencial eléctrico del electrolito que rodea el metal. Por tanto, se establece una diferencia de potencial, conocida como el potencial de media celda, entre el metal y la mayor parte del electrolito.

En consecuencia, cuando no fluye corriente entre electrodo y solución iónica se podrá observar el potencial de media celda. Sin embargo, cuando hay una corriente, este potencial se ve alterado. La diferencia entre el potencial de media celda y el potencial medido con flujo de corriente es el resultado de la alteración de la distribución de carga en la solución iónica en contacto con el electrodo. Este efecto se conoce como la polarización del electrodo.

Los electrodos perfectamente polarizados admiten una corriente entre el electrodo y la solución iónica por el cambio de la distribución de cargas dentro de la solución cerca del electrodo. Por lo tanto, no hay una corriente que cruza el interfaz electrodo-electrolito. Sin embargo, electrodos no polarizados, permiten que la corriente pase libremente a través de la interfaz electrodo-electrolito, sin cambiar la distribución de carga en la solución cerca del electrodo.

Aunque estos tipos de electrodos se pueden describir teóricamente, no se pueden fabricar en la práctica. Para la mayoría de las mediciones se prefieren los electrodos no polarizados, ya que este tipo de electrodos no están tan afectados por artefactos de movimiento como los electrodos polarizados [11].

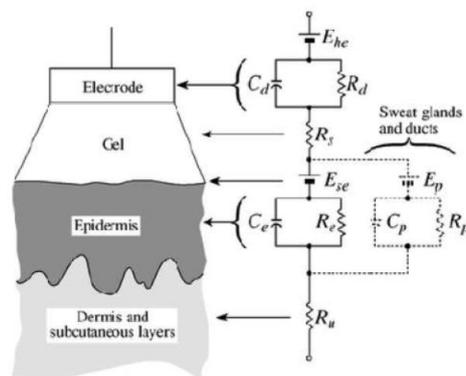


Figura 10. Modelo equivalente del interfaz piel electrodo [12].

Para entender el funcionamiento de un electrodo e incluirlo en el sistema eléctrico se crea el modelo eléctrico equivalente (figura 10). Con este modelo equivalente es más fácil estudiar su comportamiento dentro de un sistema eléctrico.

3.4. Características de la señal

La señal de ECG se puede separar en “trozos” de onda, llamados lóbulos. Cada lóbulo representa la actividad eléctrica de un momento del ciclo cardiaco. No sólo se estudia la forma de una onda, sino también las duraciones entre intervalos.

La señal de ECG, es una señal fisiológica muy característica y similar entre personas. A diferencia de otras Señales como el EEG (electroencefalograma), esta señal varía mucho entre sujetos, incluso entre el mismo sujeto en diferentes situaciones, por lo que es una señal difícil de estudiar. Sin embargo, como la señal de ECG normal mantiene las mismas características en todas las personas (figura 11), en caso de presencia de alguna patología, la forma de la onda varía y difiere de un ECG normal, lo que facilita su detección y estudio.

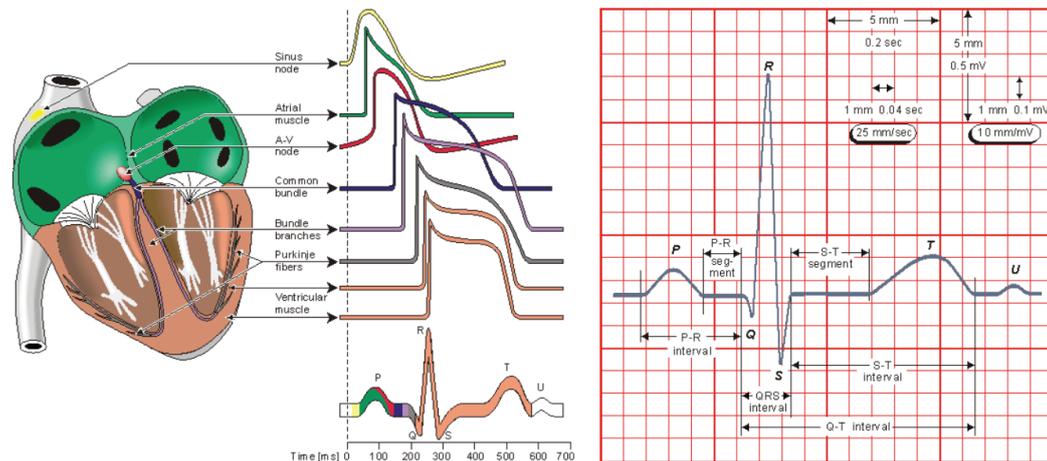


Figura 11. Izquierda, actividad eléctrica de cada parte del corazón. Derecha señal de electrocardiograma normal [8].

- (1) Espacio P-R (200 ms)
- (2) Intervalo Q-T (310 ms)
- (3) Segmento P-R (100 ms)
- (4) Complejo QRS (70 ms)
- (5) Intervalo S-T (240 ms)
- (6) Segmento S-T (140 ms)

Datos sacados de [7].

La primera onda que aparece en el ECG normal es la onda P, causada por la contracción de la aurícula. Es más pequeña debido a que la aurícula es pequeña comparada con el ventrículo. La duración típica es de 100 ms.

Aproximadamente 100 ms después del final de la sístole auricular, aparece el segmento P-R. Este periodo corresponde con la repolarización de la aurícula y el comienzo de la despolarización del ventrículo. Dicha actividad eléctrica se limita a la conducción del impulso por el Haz de His hacia las fibras de Purkinje. Como son pocas fibras la actividad eléctrica que se capta es prácticamente nula.

La contracción de las fibras ventriculares produce tres ondas en el registro de derivación I del ECG (figura 11):

- La primera de ellas es la onda Q, de muy baja amplitud y polaridad negativa.
- La segunda es la onda R, de gran amplitud frente a las otras ondas del ECG y de polaridad inversa a la anterior.

- Por último, la onda S, semejante a la onda Q pero de amplitud ligeramente mayor.

Estas ondas forman el complejo QRS, cuya duración típica es de 70 ms.

Aproximadamente 210 ms después de comenzar la despolarización ventricular, comienza repolarización. Esto provoca la aparición de la onda T. El último tramo del ECG forma una línea horizontal denominada línea base que corresponde a las fases isoelectricas del ciclo cardiaco, es una fase en la cual no hay actividad eléctrica.

Para describir los registros del ECG correspondientes a las derivaciones utilizadas se utiliza esta nomenclatura (ondas P, Q, R, S, T y U) para entender que secuencia se está siguiendo y analizando. Las derivadas captadas son proyecciones del vectocardiograma por eso algunas ondas en algunas derivadas se pueden ver negativas o algunos complejos pueden no aparecer [7].

3.5. Procesamiento de señal

La señal de ECG puede proporcionar gran información. Ésta puede ser extraída mediante distintos procedimientos matemáticos como el algoritmo de Pan-Tompkins, explicado más adelante, que sirve para extraer el ritmo cardiaco, o pasar la señal al dominio de la frecuencia y poder detectar patologías. De estos algoritmos lo importante es extraer parámetros para que el personal clínico sepa interpretar de forma sencilla los resultados.

Como paso previo, la señal debe ser procesada, ya que existen muchas fuentes de ruidos e interferencias en la señal. Las fuentes de ruido, comprenden un conjunto de señales que no interesa medir y aparecen junto a la señal de interés debido al método de adquisición. En muchos casos no se pueden evitar, por tanto, se necesita un procesamiento posterior para eliminarlas o atenuarlas. Ejemplos de este tipo de señales son:

- Otras señales fisiológicas: Existen muchas señales fisiológicas bioeléctricas (figura 12), con diferentes magnitudes de amplitud y frecuencia. Son señales bioeléctricas, pero en esta aplicación (ECG) son consideradas ruido ya que son señales que no interesan e interfieren en la señal principal.

Algunas señales se superponen en amplitud y frecuencia dando a entender que se interferirían (electroretinograma) pero por la colocación de los electrodos no se afectan mutuamente. Por otro lado, tenemos el electromiograma, esta señal procede de la actividad muscular. En este caso no se puede evitar ya que esta señal está presente prácticamente en todo el cuerpo. Por todo ello es importante filtrar la señal, ya que las frecuencias no se superponen totalmente.

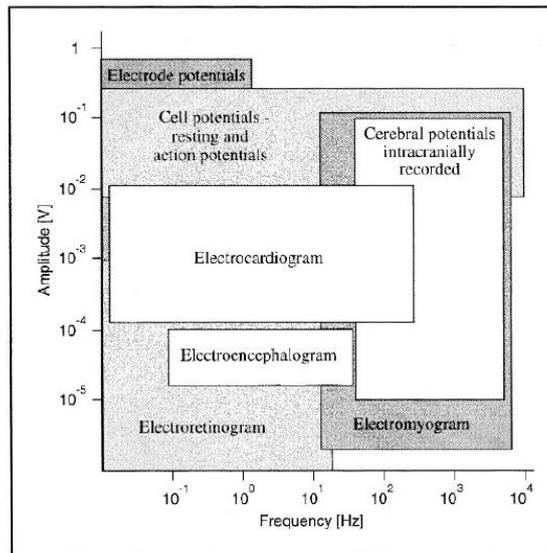


Figura 12. Espectro bioseñales [13].

- Contacto de electrodos: los electrodos, como se ha explicado en el punto 3.3, tienen un modelo equivalente donde aparecen unos condensadores. Estas capacitancias pueden generar un nivel de continua o fluctuaciones en la línea base que no aportan información y pueden ser perjudiciales para el procesamiento de la señal. Esto puede ser reducido mediante electrodos no polarizados con potenciales de media celda bajos como los electrodos de Ag/AgCl, alta impedancia en los amplificadores y la reducción de la impedancia de la fuente mediante gel electrolítico.
- Movimiento del paciente: el movimiento del paciente no solo genera señales bioeléctricas, sino también cambia la estructura corporal como la respiración. Estos movimientos se traducen a la señal del electrocardiograma, afectando a la línea base como frecuencias muy bajas.

Las fuentes de interferencia, son perturbaciones que pueden afectar a la señal y aparecen posteriormente a su captación:

- Interferencia de la red eléctrica: el cuerpo puede actuar como una antena y captar ruido procedente de la señal de red eléctrica. Esta señal puede ser compensada mediante protección contra interferencias electromagnéticas. Además, es una señal senoidal de frecuencia conocida (50/60 Hz). Es posible minimizar esta interferencia mediante determinados filtros.
- Interferencia de otros equipos médicos: el electrocardiógrafo puede estar actuando junto a otros sistemas médicos que pueden introducir interferencia. Por ejemplo, el electrobisturí, que interfiere de forma importante sin tener posibilidad de filtrar o atenuar esta señal.
- Interferencia por los cables de los electrodos: la corriente de los cables puede inducir interferencias en otro cable. Este efecto se logra minimizar trenzando los cables.

3.6. Equipo de adquisición

La amplitud de la señal no es suficiente para estudiar la señal por eso existen unas etapas previas para tratarla. En la figura 13 se representa un diagrama con las etapas de adquisición de un ECG.



Figura 13. Etapas fundamentales de un ECG.

La señal antes de ser amplificada se filtra mediante un filtro paso alto para eliminar el nivel de continua, que no aporta información, y evitar que el sistema sature. La etapa de amplificación debe amplificar la señal sin añadir ruido a la misma. El último filtro utiliza un filtro paso bajo para utilizarlo de antialiasing y eliminar todas las frecuencias altas no deseadas. En esta etapa también se puede integrar en serie un filtro pasa banda para eliminar la interferencia del ruido de red. Este filtrado no es necesario y puede ser procesado posteriormente mediante otros filtros digitales más potentes. La última etapa es la conversión analógica digital, antes de enviarla al dispositivo encargado de procesar la señal.

El ADAS1000 es un circuito integrado el cual devuelve la señal ya digitalizada cuando se conectan los electrodos. Más adelante se explica su funcionamiento con más detalle.

3.7. Seguridad eléctrica

La corriente eléctrica cuando recorre el cuerpo humano puede suponer un peligro para el individuo dependiendo de la magnitud, de la intensidad, de la frecuencia, del tiempo de duración y de la zona que afecte. Por eso es importante que existan medidas de seguridad eléctricas.

En relación a la magnitud se pueden diferenciar distintos grupos [7]:

- Nivel de percepción (0.5 a 5 mA), es el nivel de intensidad a la que un individuo entorno a los 70 kg notaría su efecto recorriendo el cuerpo.
- Contracciones involuntarias (10 mA), con este nivel de intensidad el individuo no podría controlar las contracciones del músculo, llegando a ser doloroso si se mantienen durante un periodo prolongado de tiempo.
- Parálisis respiratoria (20 mA), a este nivel de intensidad las contracciones musculares son tan fuertes que se pierde el control de los músculos respiratorios, causando, por tanto, parálisis respiratoria.
- Fibrilación ventricular (80 a 400 mA), el corazón como se comenta en el apartado 2.1, es susceptible a la actividad eléctrica por tanto si esta corriente parte llega a pasar en parte por el corazón ésta resulta ser suficiente como para excitar parte del musculo del corazón, la actividad del corazón normal es afectada y por tanto éste puede aumentar el ritmo cardiaco a 300 ppm debido a las ondas que llegan al ventrículo, esto puede causar que el corazón se pare causando la muerte en pocos minutos.

Dependiendo de la zona que afecte se pueden diferenciar dos grupos:

- Microshock: la corriente entra desde el exterior y cruza el cuerpo entero, en esta situación cerca del 5-10% de la corriente cruza el corazón.

- Macroshock: la corriente entra desde el exterior y toda fluye por un catéter a través corazón.

Según los tipos de protección se pueden agrupar en equipos de distintas clases:

- Equipo Clase I:

- Aislamiento básico.
- Partes metálicas accesibles conectadas a tierra.

- Equipo Clase II:

- Aislamiento reforzado o doble aislamiento.
- No conectado a tierra.

- Equipo Clase III:

- Alimentación a muy baja tensión (hasta 24 V AC o 50 V DC).

- Equipo con alimentación interna:

- No requiere conexión a la red eléctrica (equipos a baterías).
- No puede funcionar si se conecta a la red (por ej., para cargar la batería):

El dispositivo de este proyecto está dentro de la categoría de equipo con alimentación interna. Además, los electrodos conectados al circuito integrado ADAS1000 no dejan el paso de corrientes.

4. Front-end ADAS1000

En el diseño de este dispositivo se utilizará el circuito integrado ADAS1000, del que previamente al diseño se estudiará su funcionamiento. El ADAS1000, es un circuito integrado diseñado para simplificar la adquisición y garantizar la calidad de la señal de ECG. Este proporciona un sistema de adquisición de pequeños datos y baja potencia para aplicaciones de biopotenciales.

Tiene características auxiliares que ayudan a la adquisición de la señal de ECG de mejor calidad. Éstas incluyen (figura 14): una derivada multicanal promediada, una referencia seleccionable, rápida recuperación de sobrecarga, circuito de respiración flexible que devuelve información de magnitud y fase, algoritmo de detección de ritmo interno que opera en tres derivaciones y la opción de detección de electrodo desconectado en AC o DC.

Tiene varias salidas digitales que aseguran la flexibilidad cuando se monitorice y se analice la señal. El post-procesamiento cardíaco se hace externo con un DSP, microprocesador o FPGA.

Debido a que los sistemas de ECG abarcan diferentes aplicaciones, cuentan con una arquitectura de escalada potencia/ruido donde el ruido puede reducirse a expensas de aumentar el consumo de energía. Por un lado, los canales de adquisición de señal se pueden cerrar para ahorrar energía, así como reducir las velocidades de datos.

La estructura de entrada es un amplificador diferencial, lo que permite a los usuarios una variedad de opciones de configuración que mejor se adapten a su aplicación.

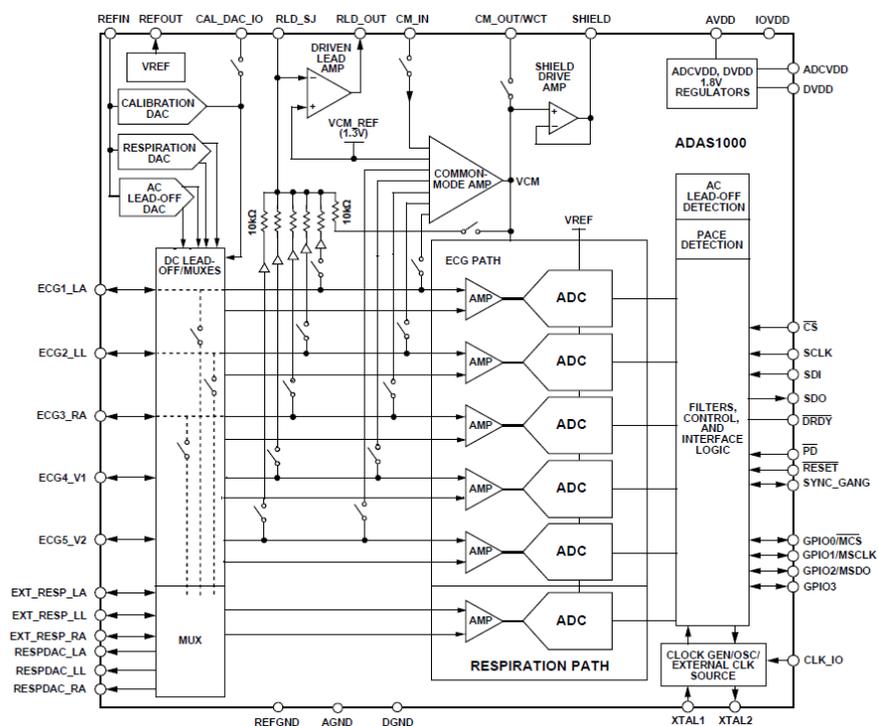


Figura 14. Esquema de funcionamiento del ADAS1000 [14].

El ADAS1000 tiene 5 entradas (LA, LL, RA, V1 y V2) para la adquisición de los biopotenciales, también tiene otras 6 entradas para medir el ritmo de respiración pero que en ese proyecto no se van a usar. Además, cuenta con una unidad de referencia (RLD), con distintos módulos analógicos y con una unidad de cálculo.

Por último, hay que tener en cuenta que el ADAS1000 opera con una frecuencia de 8.192 MHz proporcionada por un oscilador externo conectado al generador de reloj externo.

4.1. Estructura y funcionamiento ADAS1000

Analog Devices propone un ejemplo de esquema eléctrico para el correcto funcionamiento (figura 15) [15]. Este esquema sigue de guía para el diseño y asegurarse su funcionamiento.

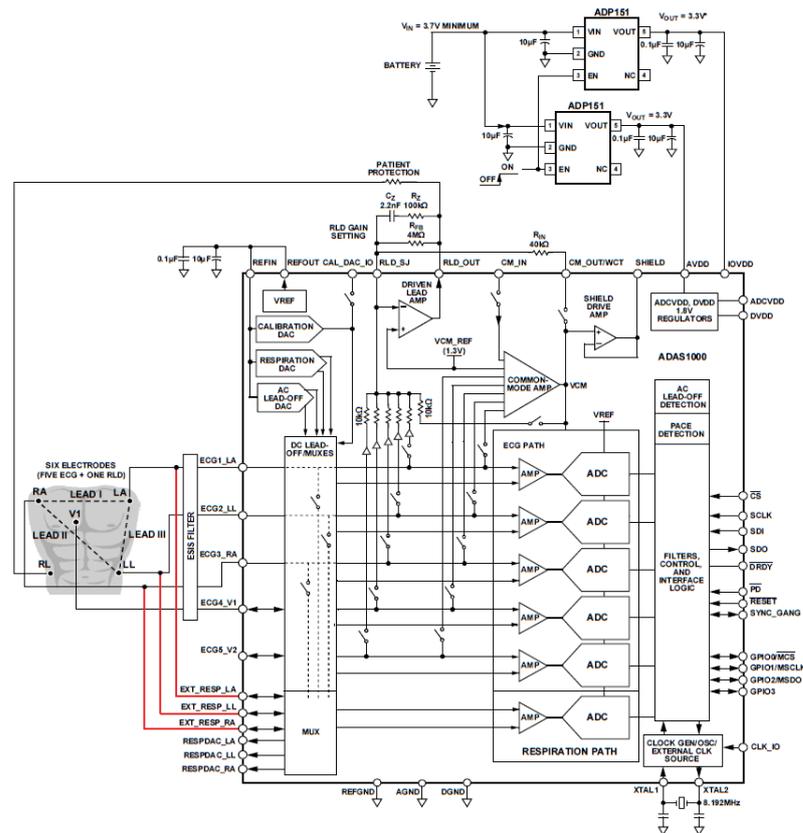


Figura 15. Esquema propuesto por Analog Devices para el del ADAS1000.

1) Entrada electrodos

Los electrodos se conectan directamente a las entradas correspondientes. Normalmente entre los cables y la entrada al dispositivo, se coloca una protección externa contra desfibrilación. Consiste en colocar entre el cable y masa unas bombillas de neón, las cuales no conducen a tensiones bajas y en caso de desfibrilación, donde las tensiones son elevadas, los neones conducen, la corriente se deriva por ellos y no entra en el dispositivo el cual podría verse afectado.

Por razones de tamaño del dispositivo y, ya que el dispositivo no se prevé que este en contacto con un desfibrilador, no se implementara la protección contra desfibrilaciones y los electrodos se conectaran directamente al ADAS1000.

2) Estructura de cada canal

Cada canal de ECG está formado por (figura 16), en primer lugar, unos diodos que sirven para proteger sobre tensiones y corrientes elevadas (positivas o negativas) para el sistema o el sujeto. Éstos trabajan a menor tensión que los neones comentados en el punto anterior. Posteriormente se conecta a un amplificador operacional de bajo ruido con ganancia programable, a un filtro que sirve de filtro *antialiasing* con ganancia fija, a un buffer de datos y al convertor analógico digital (ADC), este es de 14 bits, pero con un sobre-muestreo completa la información de 24 bits que posteriormente es enviada.

Cada entrada se conecta al terminal no inversor del amplificador operacional de ganancia programable y en el terminal inversor se puede conectar otro electrodo o el *WTC*. Para seleccionar estas opciones consta de unos interruptores configurables, el *WTC* proviene del propio integrado y se explica más adelante.

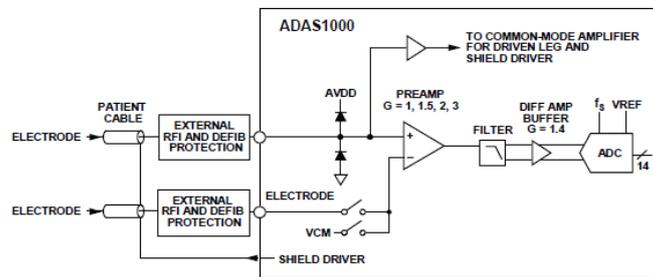


Figura 16. Esquema de la estructura de los canales de los electrodos del ADAS1000 [14].

3) Selección modo común y promediado

La señal de modo común se puede derivar de cualquier combinación de una o más de las entradas de los electrodos (figura 17), de la referencia interna fija de modo común o de una fuente externa conectada al terminal correspondiente.

La opción de referencia interna es útil en la medición de las señales de calibración del ADAS1000 o mientras se pretende recoger una señal de ECG mediante la colocación de sólo dos electrodos. La flexibilidad de la generación del modo común permite el control completo por parte del usuario de los canales que contribuyen.

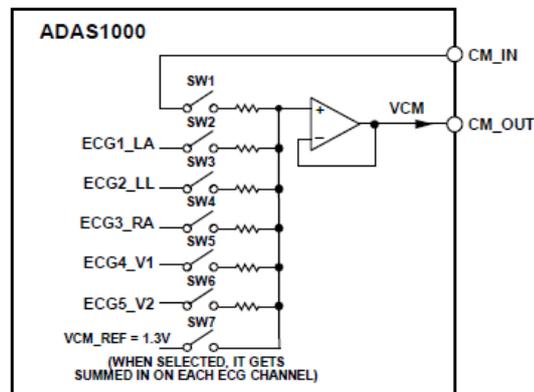


Figura 17. Estructura modo común del ADAS1000 [14].

4) Centro terminal de Wilson

En el punto 3.2 ya se ha introducido qué es el CTW. Los electrodos para el CTW se eligen mediante unos interruptores programables. Cada canal conectado se junta en el terminal inversor de un amplificador operacional. Del terminal inversor (RLD_SJ en la figura 18) y del terminal de salida del amplificador operacional (RLD_OUT en la figura 18) operacional se sacan dos conexiones del ADAS1000 para conectarse a una resistencia y unos condensadores que sirven de filtrado para extraer la señal y conectarla al pin correspondiente con el CTW.

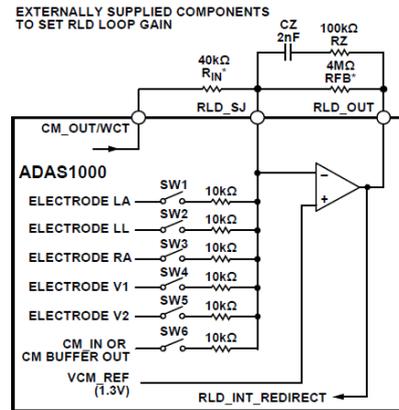


Figura 18. Estructura centro de terminal de Wilson del ADAS1000 [15].

5) Derivada de referencia o derivada de la pierna derecha

El amplificador operacional del módulo de la unidad de la pierna derecha se utiliza como parte del circuito de realimentación. Con él, se fuerza una tensión de modo común en el paciente lo más cerca posible al nivel interno de voltaje de referencia del ADAS1000. Esto proporciona el máximo rango dinámico en los canales de entrada.

También ayuda a disminuir el ruido de las interferencias de fuentes externas tales como luces fluorescentes u otros instrumentos conectados al paciente. Las corrientes inyectadas por el módulo de respiración también vienen absorbidas por el operacional de la unidad de la pierna derecha.

6) Detector de electrodo suelto

Para poder detectar si un electrodo está desconectado o mal colocado, el ADAS1000 tiene implementados dos métodos: la detección mediante corriente alterna y la detección mediante corriente continua. Los dos métodos son completamente independientes uno del otro y pueden ser activados por separado o actuar a la par. Si el ADAS1000 detecta un electrodo suelto lo comunica en un registro vía SPI.

a) Detección de electrodo suelto en continua:

Este método inyecta una pequeña corriente continua de magnitud programable en cada canal de entrada. Si el electrodo está conectado correctamente, la corriente fluye directamente hacia el canal de referencia, produciendo así un cambio mínimo en el potencial del canal. Si el electrodo no está conectado correctamente la corriente provoca un incremento del potencial rápido que es detectable por los comparadores de cada canal. La corriente de inyección es

programable a través de un registro SPI. Se pueden seleccionar corrientes entre 10 nA y 70 nA en intervalos de 10 nA.

b) *Detección de electrodo suelto en alterna:*

Este método de detección se basa en la inyección de corrientes alternas en cada canal y la medición de las amplitudes de las tensiones resultantes, funcionando de manera análoga al método que trabaja con corriente continua. El sistema utiliza una frecuencia portadora fija ligeramente por encima de los 2 kHz que se elimina fácilmente con los filtros incluidos en el ADAS1000, sin introducir artefactos en la fase o la amplitud en la señal a medir. Figura 19 muestra el esquema de funcionamiento del detector de electrodo suelto.

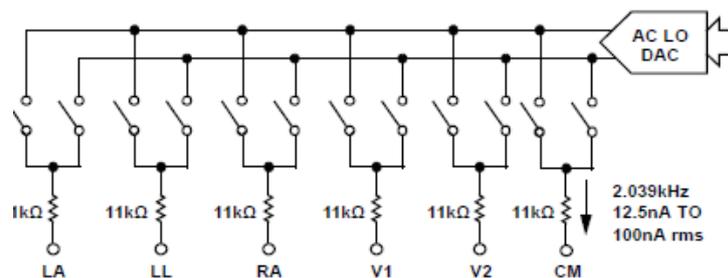


Figura 19. Simplificación de la detección de electrodo suelto en alterna del ADAS1000 [14].

6) Respiración

El ADAS1000, también tiene unos pines destinados a medir la respiración que utilizan una estructura distinta a la del ECG. Este proyecto no está orientado al estudio ni medición de la respiración.

7) Función de detección de espícula de marcapasos

La función de detección de espícula de marcapasos detecta los potenciales artefactados por un marcapasos y mide la anchura de los pulsos válidos. Estos datos son guardados en un registro y además trabajan en paralelo al ECG.

Estas operaciones se realizan mediante una máquina de estados que trabaja a 128 Hz y utiliza datos de 16 bits que provienen de la señal diezmada de ECG principal. Antes de pasar a la detección de espícula, la señal de ECG es diezma y si no está perfectamente alineada, estas diferencias son deterministas por tanto pueden ser compensadas. El algoritmo que sigue esta explicado en la figura 20.

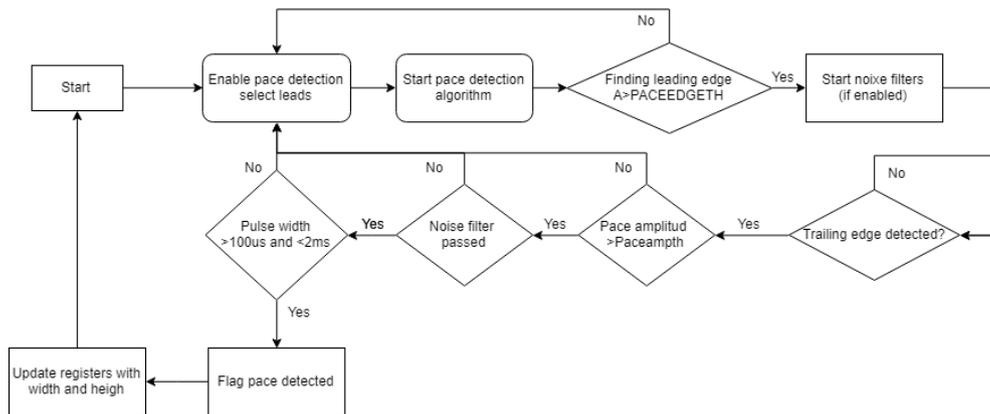


Figura 20. Diagrama de flujo del algoritmo de detección de artefactos del ritmo

8) Filtrado

El dispositivo tiene una etapa de filtrado antes de enviar la señal, estos filtros disponibles son filtros digitales. La frecuencia de muestreo puede ser de 1024 MHz (bajo rendimiento) y 2048 MHz (alto rendimiento).

Existen 4 filtros paso bajo con frecuencia seleccionable por el usuario. Las frecuencias seleccionables son 40 Hz, 150 Hz y 250 Hz, el cuarto filtro es usado para calibración y tiene una frecuencia de corte de aproximadamente 7 Hz.

5. Diseño del dispositivo

Para diseñar un dispositivo es fundamental en primer lugar definir la estructura que va seguir el dispositivo para poder planificar un correcto desarrollo del dispositivo, para reducir los posibles percances y garantizar los mejores resultados.

5.1. Estructura

La parte fundamental y central del dispositivo es el microcontrolador. Éste es el que se encargará de controlar y hacer funcionar el dispositivo. Recibe la señal desde el ADAS1000, circuito integrado a estudiar en este trabajo, y envía los datos por bluetooth o los almacena en una memoria externa para su posterior estudio.

Todos los componentes electrónicos y toda la electrónica necesaria para su correcto funcionamiento deben estar conectados a una alimentación. Uno de los objetivos de este dispositivo es la utilización en el deporte y por tanto requiere de conexiones inalámbricas, por lo que necesita una batería que alimente toda la electrónica. La estructura del dispositivo está definida por el diagrama de bloques de la figura 21.

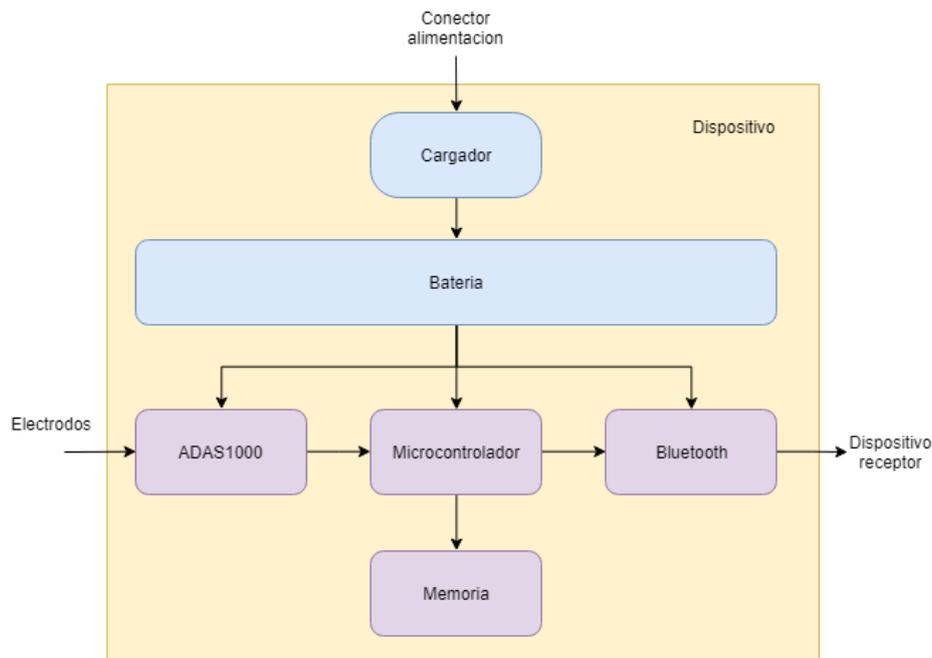


Figura 21. Diagrama de bloques funcionales del dispositivo a diseñar

5.2. Elección de componentes principales

- **Electrodos**, son los que se encargan de captar la señal directamente de la superficie de la piel. El dispositivo estará diseñado para cualquier tipo de electrodo.
- **Batería**, el dispositivo necesita de una fuente de energía para poder funcionar, este dispositivo estará conectado de forma inalámbrica para permitir al usuario total libertad de movimiento. Por tanto, para garantizar esta libertad necesita una batería con autonomía suficiente para el funcionamiento del dispositivo durante el tiempo necesario.

La batería elegida es una batería de litio polímero, su relación peso-energía es muy buena para este diseño ya que interesa que el dispositivo sea ligero y no interfiera en la actividad del sujeto. Además, esta batería no presenta histéresis en la carga y descarga y tiene gran una, apta para este diseño.

- **Circuito de carga**, la batería utilizada es recargable por tanto se necesita de un circuito que permita la carga de esta batería y no se necesite cambiarla. El circuito debe detectar cuando la batería ha llegado a su máxima capacidad para evitar sobrecargas y sobrecalentamientos.

Utilizando un termistor de 10k que modifica su valor con la temperatura y por tanto cuando llega a su capacidad máxima evita que haya sobrecalentamiento. El circuito elegido utiliza esta familia: circuitos delta V. Pero la elección de este circuito en particular es debido a que cumple estas prestaciones y por su tamaño reducido. Para alimentar el circuito de carga y cargar la batería se ha utilizado un conector USB. Éste permite cargar el circuito y además programar el microcontrolador, reduciendo así el número de conectores y permitiendo que el dispositivo tenga un tamaño reducido.

- **PIC24f128GC006**, es el microcontrolador. Un microcontrolador es un circuito integrado programable que contiene un sistema microprocesador completo, es decir, una memoria, una unidad central de procesamiento y periféricos de entrada/salida. Es la parte central de la placa, encargado de controlar los distintos elementos de la placa.

Este dispositivo se encarga de recibir la señal desde el ADAS1000 y enviarla por bluetooth o almacenarla en una tarjeta de memoria. Además, es el microcontrolador el que manda los registros necesarios para configurar y controlar el ADAS1000. Este microcontrolador cumple las necesidades del sistema, ya que dispone de puertos de comunicación *I2C*, *SPI* y *UART*, así como varios temporizadores independientes.

Además, dicho microcontrolador dispone de gran cantidad de memoria RAM para el almacenamiento temporal de datos. El controlador facilita el diseño de sistema ya que dispone de puertos periféricos seleccionables, lo que incrementa la eficiencia de la configuración de los pines de comunicación.

Asimismo, este componente es ideal para sistemas de baja potencia y aplicaciones con múltiples conexiones. Este componente ha sido elegido para esta aplicación por su facilidad gracias a la gran disponibilidad general y por su flexibilidad para crear aplicaciones de gran alcance con las bibliotecas preexistentes,

- **Memoria μ SD**, es un medio de almacenamiento de información no volátil, es decir la información no es borrada después de desconectar de la alimentación. Deriva de las memorias EEPROM, mejorando su funcionamiento y rendimiento. Es el tipo de memoria utilizado en las memorias USB o las tarjetas SD (*Secure Digital*). Para programar con un microcontrolador las memorias SD son las más utilizadas por que son más sencillas. Además, existen muchas librerías por su uso extendido.

Por otro lado, por su tamaño reducido es el más adecuado para este diseño. En concreto, dentro de las tarjetas SD se engloban las tarjetas micro SD, de tamaño más inferior y actualmente de gran capacidad, por ello este tipo es el utilizado para este proyecto. Esta

memoria no volátil extra permite que el dispositivo pueda ser utilizado como Holter y tener una memoria que pueda durar días.

- **RN-42**, es un módulo bluetooth encargado de recibir datos y enviarlos a otro dispositivo bluetooth. Se comunica con el microcontrolador mediante una conexión serie UART. El RN-42 es una versión 2.0 muy utilizada y sencilla. Por tanto, existe una gran variedad de librerías y ejemplos que facilita su uso.

- **ADAS1000**, Este integrado es el *front-end* del ECG, tiene todos los bloques necesarios para el procesado y digitalización de la señal de ECG, se ha elegido para estudiar su funcionamiento y evaluar si cumple las necesidades de un sistema de ECG.

A continuación, se muestra una tabla a modo de resumen de los principales componentes del dispositivo desarrollado:

Tabla 1. Resumen de los componentes del dispositivo

Electrodos	Cualquier tipo de electrodo
Batería	Buena relación peso/energía, recargable y autonomía suficiente
Circuito de carga	Funcionamiento adecuado, tamaño reducido y poca pérdida de calor
PIC24FJ128GC006	Uso extendido y dedicado para aplicaciones de baja potencia
Memoria flash	Memoria no volátil de tamaño reducido
RN-42	Tamaño reducido, uso extendido y sencillo
ADAS1000	Circuito integrado para ECG completo

5.3. Diseño de la placa

Distribución del espacio:

El primer paso para el diseño de la placa es conocer las limitaciones y condiciones que presentan cada componente. Los elementos que tienen ciertas limitaciones son el RN-42 y el ADAS1000.

El RN-42, en la hoja de datos el fabricante recomienda que la antena del bluetooth no esté cubierta por la PCB u otro componente y, si es posible, que esté alejada del resto de componentes electrónicos (figura 22). Por eso se ha diseñado la placa con una hendidura en el medio para situar la antena del RN-42, evitando las interferencias de otros componentes.

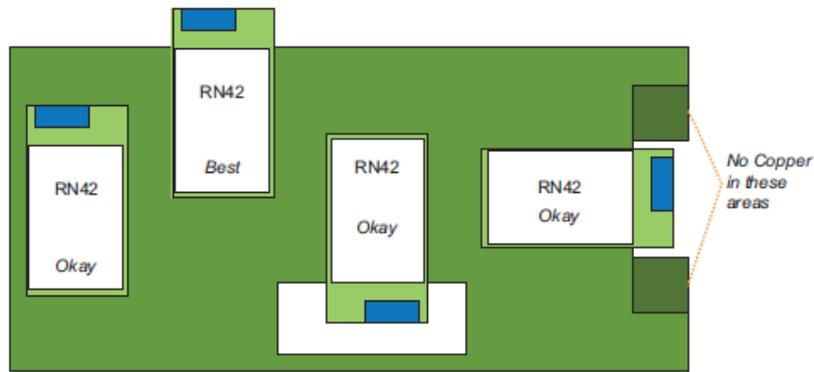


Figura 22. Recomendación del fabricante de la colocación del RN42 [16].

Otro condicionante importante en el diseño de la placa es el ADAS1000. En este la señal analógica se convierte a digital para enviarla al microcontrolador. Este proceso es delicado ya que la parte digital puede introducir fácilmente ruido en la parte analógica y afectar a la calidad de la señal. La señal analógica es más sensible al ruido, esto es debido a que, al tratarse de una señal fisiológica, puede tomar infinitos valores y valores continuos, por lo que pequeñas variaciones pueden afectar a la señal original superponiéndose a la misma.

La señal digital toma valores finitos que se codifica usando solo dos valores (nivel bajo (0) o nivel alto (1)), normalmente se utiliza 0 V para nivel bajo y 3.3 V para nivel alto). Los dispositivos digitales detectan o generan señales digitales dentro de dos márgenes (figura 23), por tanto, cualquier valor entre esos dos valores el dispositivo lo clasificara como nivel alto o nivel bajo. Estos márgenes son del orden de voltios, por ello es una señal robusta a pequeñas interferencias.

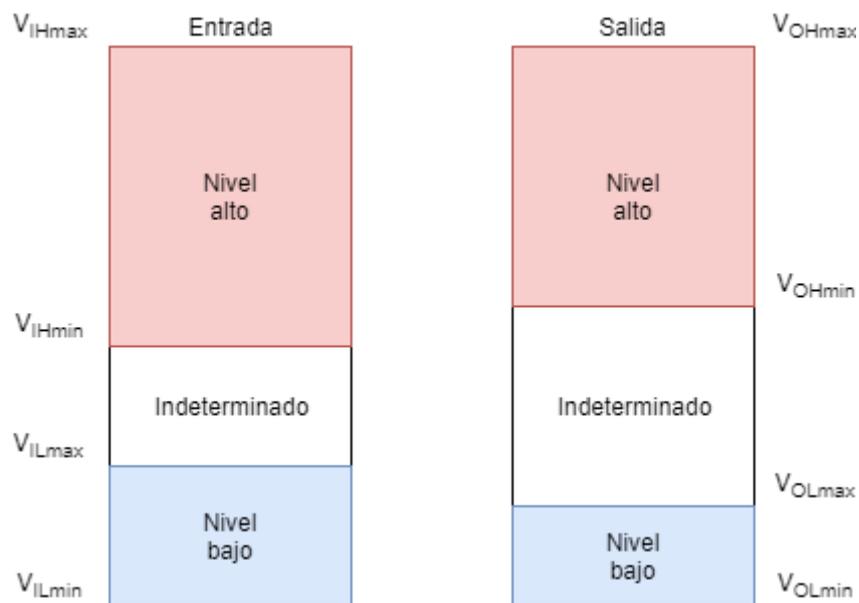


Figura 23. Diagrama de voltaje para los niveles digitales

Además, cabe tener en cuenta que la electrónica utilizada por un integrado digital (reloj de la CPU y otros componentes) puede interferir en la parte analógica. Por eso es importante mantener lo más alejadas posibles la parte analógica de la digital.

Teniendo ello en cuenta, para que ninguna corriente de la parte digital interfiera en la parte analógica, lo ideal sería tener también dos masas y dos alimentaciones. Como eso no es posible, la alternativa implementada consiste en separar dos planos de masas (analógico y digital) que sólo se conecten en un punto y esté, además, alejado de la parte analógica.

Para la alimentación la idea es la misma: en la salida de la fuente de alimentación se separa la alimentación digital y alimentación analógica para que no se interfieran, ya que éste es el punto más alejado que se puede conseguir.

También se ha distribuido el espacio para que los componentes y pistas analógicas no se superpongan a las digitales, ya que, de nuevo, las corrientes digitales pueden inducir corrientes e interferir en las pistas analógicas. Como se ve en la figura 24, la parte analógica, marcada con líneas rojas, está separada del resto de elementos sin superponer ninguna pista para reducir las interferencias.

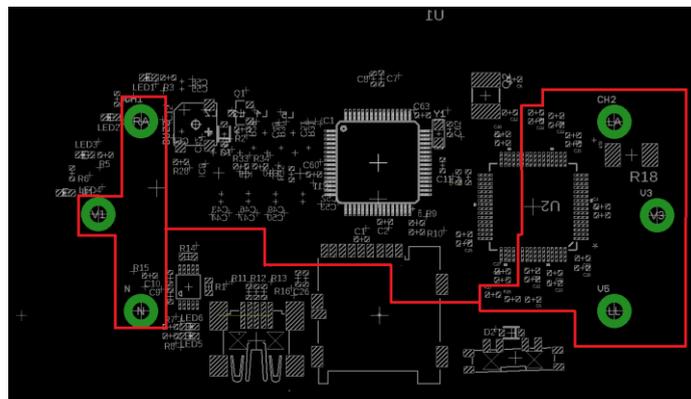


Figura 24. Esquema de la PCB simplificado. Líneas rojas corresponden a la parte analógica.

Los componentes digitales como el microcontrolador y el control de la batería, se sitúan en la parte central de la placa. Por los alrededores y en un lado se sitúan las pistas de los electrodos y el ADAS1000 para no interferir.

El ADAS1000, tiene parte digital y analógica, está diseñado de tal forma que la parte digital queda en un lado y la analógica en otro. La parte digital se orienta hacia el centro de la placa, donde están el resto de componentes digitales y la parte analógica hacia el otro, donde se sitúan los componentes de la parte analógica.

Colocación PIC24FJ128GC006:

El microcontrolador dispone de muchos pines que pueden ser programables con diversas funciones (comunicación: SPI, I2C... entradas/salidas digitales, entradas/salidas analógicas, salidas PWM...). Pero algunos de ellos tienen una función fija y son necesarios para su funcionamiento, como los pines de alimentación y los pines para programar el microcontrolador (PGC, PGD y reset). En la hoja de datos explica estas conexiones básicas de estos pines (figura 25), para su correcto funcionamiento.

El microcontrolador debe conectarse a la alimentación. La alimentación está dividida en 3 tipos: alimentación digital, alimentación analógica y alimentación del transceptor USB (*transceiver*). El microcontrolador necesita distintas alimentaciones ya que, como se ha explicado, las partes analógicas y digitales deben estar separadas para evitar interferencias

En cada alimentación se debe conectar un condensador entre el terminal de alimentación y masa, lo más cercano posible al pin. Estos condensadores, llamados condensadores de desacoplo, sirven para conseguir que la alimentación sea lo más estable posible, por ello filtran los ruidos de la alimentación.

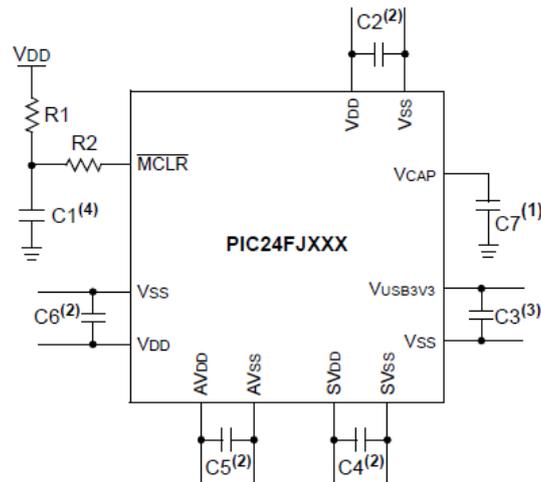


Figura 25. Esquema propuesto por el fabricante para los condensadores de desacoplo del PIC24FJ128GC006 [17].

Los valores de los condensadores de la figura 25 están resumidas en la siguiente tabla:

Tabla 2. Valores resistencias y condensadores del PIC

C1	1 uF	C2	1 uF	C3	1 uF	C4	1 uF
C5	1 uF	C6	1 uF	C7	1 uF	R1	10 kΩ

El MCLR, master clear pin, tiene 3 funciones específicas: reiniciar, programar el dispositivo y debugging (controlar el microcontrolador paso a paso desde el ordenador y entender su funcionamiento).

Colocación ADAS1000:

En información adicional que proporciona el fabricante [15], existe un ejemplo de cómo montar el integrado (figura 15). Además de estos componentes se deben colocar una serie de condensadores de desacoplo importantes para las alimentaciones.

Circuito de carga: La hoja de datos propone un esquema de funcionamiento que es el que ha sido utilizado para el diseño, ya que es el que fabricante garantiza mejor funcionamiento (figura 26).

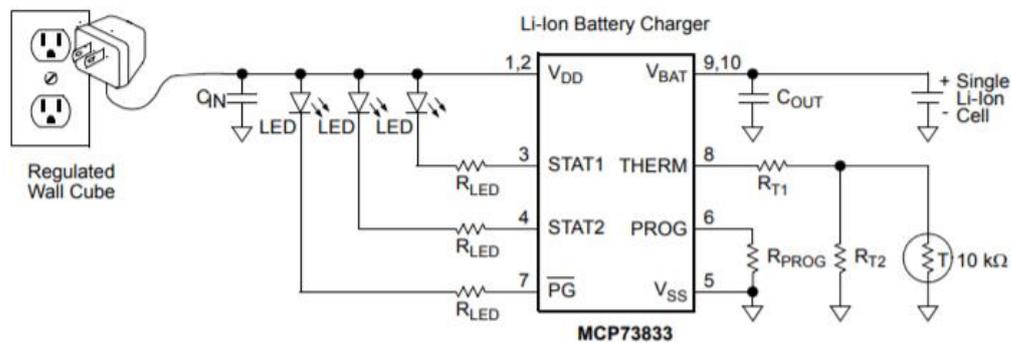


Figura 26. Esquema propuesto por el fabricante del MCP73833 [18].

RN-42: El dispositivo RN-42 necesita un condensador de desacoplo en la alimentación. Por otro lado, los pines de comunicación UART (RX y TX) están conectados directamente a los pines correspondientes del microcontrolador. Finalmente, un tercer pin es utilizado para indicar cuándo el dispositivo está comunicando. A la alimentación se le debe poner un condensador de desacoplo.

Tarjeta SD: La memoria externa requiere de 4 pines para realizar la comunicación SPI, una comunicación que permite transmitir datos utilizando solo 4 cables. En este tipo de comunicación se asignan dos papeles: esclavo y maestro, siendo maestro el que lleva el ritmo de la comunicación (figura 27):

- **MISO:** (master input, slave output), por donde la señal es enviada desde el dispositivo esclavo al dispositivo maestro.
- **MOSI:** (master output, slave input), por donde la señal es enviada desde el dispositivo maestro al dispositivo esclavo.
- **Reloj:** la comunicación se realiza al ritmo de reloj marcado por el maestro, por tanto, el esclavo no requiere de información sobre qué velocidad se va a transmitir.
- **CS o SS:** (chip selector o slave selector), se transmite la señal del maestro al esclavo, indica que el dispositivo maestro va a enviar una señal y el esclavo se prepara para recibir. Esto sirve para utilizar diversos dispositivos como esclavo, manteniendo los mismos cables de reloj, MISO y MOSI y no inferirse entre ellos. Sólo los dispositivos con el CS activado actuarán ante la señal enviada del maestro.
- **Ready:** En la estructura general de comunicación SPI está incluido, pero se utiliza para al dispositivo esclavo. El pin indica cuándo está listo para emitir y el dispositivo maestro empieza a leer. Se utiliza para garantizar una buena sincronía entre ambos dispositivos.

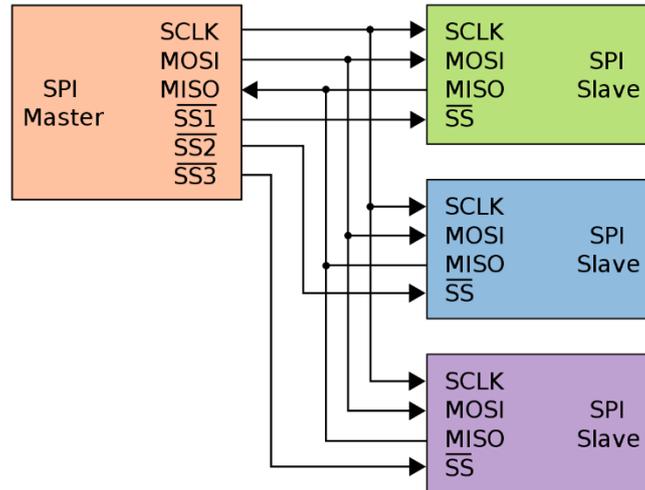


Figura 27. Esquema representativo de una comunicación SPI en grupo.

Alimentación: Se utilizará un integrado siguiendo la información proporcionada por la hoja de datos. Además, se utilizarán 5 circuitos para generar 5 alimentaciones distintas, ya que no todos los componentes funcionarán a la vez y de esa forma se separará la alimentación de la parte analógica de la digital.

Conexión USB: El conector USB sirve para cargar (alimentación y masa) y programar el micro. Los pines PGC, PGD y RST, necesarios para programar, se conectan directamente desde los pines de datos del USB. Finalmente, los pines de masa y alimentación se conectan directamente al integrado, encargado de alimentar la batería.

5.2. Consumo

El consumo del dispositivo es una etapa importante ya que de ella depende la duración de la batería, la cual es fundamental para un dispositivo de monitorización inalámbrico. Éste deberá estar junto al paciente sin conexión a ninguna fuente de alimentación excepto la propia batería. En primer lugar, se calcula el consumo de forma teórica para ver si la batería elegida es adecuada y posteriormente se evalúa si realmente este cálculo se corresponde con la realidad.

La siguiente tabla (tabla 3) desglosa los consumos típicos y máximos que la hoja de datos proporciona de cada componente principal:

Tabla 3. Consumo teórico de cada componente principal

Componente	Consumo típico (mA)	Consumo máximo (mA)
ADAS1000	12.5	15.3
PIC	20	32
RN-42	25	50
MCP 73833	0.05	0.1
NCP551	0.004	0.008
TOTAL	57.554	97.408

La capacidad de la batería es de 800 mAh, por tanto, la duración de la batería puede rondar desde los 13.9 a 8.2 horas, suficientes para realizar las tareas que está diseñado el dispositivo y garantizando una autonomía suficiente.

En el diseño del dispositivo se ha incorporado una resistencia de 1 ohmio en la alimentación, de tal forma que mediante un medidor de corriente o el propio microcontrolador se pueda monitorizar el consumo y estimar la duración del dispositivo.

5.3. Montaje de la placa

5.3.1. Lista de materiales

Tabla 4. Lista de materiales

Componente	Cantidad (unidades)	Ref.	Métrica	Cantidad Mínima	Precio (€)	Total (€)	Total unidad (€)
Condensador 1 uF	21	2495145	402	21	0,128	2,688	0,128
Condensador 10 uF	4	2218854	402	10	0,21	2,1	0,21
Condensador 0.1 uF	12	9402047	402	12	0,12	1,44	0,12
Condensador 4,7 uF	7	2309027	402	10	0,271	2,71	0,271
Condensador 2,2 uF	4	2346879	402	10	0,321	3,21	0,321
Condensador 15 pF	2	1758945	402	10	0,0076	0,076	0,0076
Condensador 12 pF	2	2611891	402	10	0,0229	0,229	0,0229
Resistencias 1K3	1	2324261	402	10	0,0165	0,165	0,0165
Resistencias 1K	2	2447120	402	10	0,0029	0,029	0,0029
Resistencias 560	3	2059223	402	10	0,0164	0,164	0,0164
Resistencias 220	2	2078857	402	10	0,0111	0,111	0,0111
Resistencias 100 k	2	1469671	402	10	0,0573	0,573	0,0573
Resistencias 47	3	2078852	402	3	0,0054	0,0162	0,0054
Resistencias 10 K	3	2447096	402	10	0,0029	0,029	0,0029
Resistencias 4 M	1	2340632	2010	5	1,17	5,85	1,17
Resistencias 40 K	1	1652794	402	10	0,0396	0,396	0,0396
Resistencias 1 M	3	1469667	402	10	0,0573	0,573	0,0573
Resistencias 2.2 K	1	2447148	402	10	0,0033	0,033	0,0033
Resistencias 8.2 K	1	2140851	402	10	0,0573	0,573	0,0573
Resistencias 100	1	2302528	402	10	0,0299	0,299	0,0299

Resistencias 1	1	2078841	402	10	0,0208	0,208	0,0208	
LED 1 y 5 (verde)	2	1525570		2	0,255	0,51	0,255	
LED 2 (azul)	1	2217978		2	0,104	0,208	0,104	
LED 3 y 6(rojo)	2	2507530		5	0,93	4,65	0,93	
LED 4 (blanco)	1	2251493		5	0,27	1,35	0,27	
Transistor 1	1	1653622		5	0,119	0,595	0,119	
Cristal PIC	1			1		0	0	
Conector bateria	1			1		0	0	
KSSG33J12	1	2215087		5	2,01	10,05	2,01	
Diodo (D1)	1	1056502		5	0,331	1,655	0,331	
NCP551SN30	4	2318526		5	0,647	3,235	0,647	
MCP73833	1	1332160		1	0,799	0,799	0,799	
PIC24FJ128GC006	1	2355355		1	4,96	4,96	4,96	
ADAS1000BSTZ	1	2213574		1	49,76	49,76	49,76	
Mini usb b type	1	1125348		1	0,587	0,587	0,587	
USD	1	2134442		1	4,13	4,13	4,13	
Bobinas 4,7 uH	5	2215636		1	0,124	0,124	0,124	
Switch 3 posiciones	1			1		0	0	
Bluetooth RN42	1	2143310		1	15,58	15,58	15,58	
Cristal 8,192 MHz	1	810-2514		1	1,338	1,338	1,338	
						Total:	121,0032	84,5152

5.3.2. Proceso de montaje

El primer paso es la impresión de la PCB, para eso se envía el diseño hecho con Eagle, a una empresa especializada en impresión de PCB.

Con la PCB, utilizando un soldador convencional del tamaño adecuado para los componentes utilizados se ha hecho todo el proceso de soldado.

En la figura 28 se ve la distribución y el resultado final de como es la placa, abajo a la izquierda se ve el conector mini USB, en el medio la ranura para la tarjeta micro SD y a la derecha el interruptor para elegir el modo. El circuito integrado del medio es el PIC24FJ128GC006 y a la derecha el ADAS1000. Por atrás asoma la antena del bluetooth y también se sitúa la batería, por la derecha esta toda la etapa del control de la alimentación y la batería.

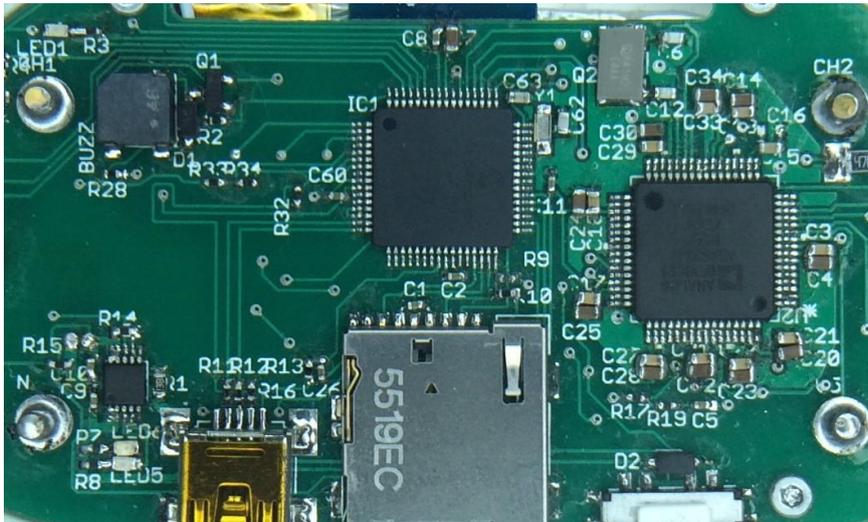


Figura 28. Placa del dispositivo

6. Firmware

El firmware es el programa informático que establece la lógica de más bajo nivel, es el que se encarga de controlar la electrónica de un dispositivo. Por tanto, al software de un microcontrolador se le llama firmware.

El firmware de este proyecto se puede dividir en tres partes, la interfaz entre el PIC y el ADAS1000 (para recibir la señal del ECG), el firmware necesario del PIC (si procesa la señal o que hace con los datos) y la comunicación del PIC con el dispositivo externo que al que enviará la señal.

6.1. Interfaz PIC-ADAS1000

La interfaz entre el microcontrolador y el ADAS se hará mediante comunicación SPI. Éste es un bus serie de comunicaciones, siendo el microcontrolador el maestro y el ADAS1000 el esclavo.

El ADAS1000, se controla mediante una serie de registros. Estos registros son de 32 bits. La estructura que siguen los bits es la siguiente:

Tabla 5. Estructura del registro de entrada al ADAS1000

Bit 31	[Bit 30: Bit 24]	[Bit 23: B 0]
Bit de lectura o escritura	Bits de dirección	Bits de dato (MSB primero)

Siguiendo esta estructura el ADAS1000 interpreta a qué registro va dirigido (bits de dirección) y qué tiene que hacer con él (bits de lectura o escritura). Para configurar el ADAS1000, se utilizan una serie de registros que indican la configuración del integrado. Para el objetivo de este proyecto no es necesario utilizar todos, los registros necesarios son: CMREFCTL, FILTCTL, FRMCTL, ECGCTL y LOFFCTL. Los demás registros no son necesarios para la función de este dispositivo.

Estos registros están definidos en la hoja de datos su estructura y función:

CMREFCTL:

Modo común, referencia y registro de control del *shield Drive*

Dirección: 0x05 [Bit 30: Bit 24]

Tabla 6. Registro CMREFCTL del ADAS1000.

Bit	Nombre	Función
23	LACM	Modo común, seleccionar electrodos
22	LLCM	Cualquier combinación de estos 5 electrodos se puede utilizar para crear la señal de modo común. 1= Contribuye al modo común 0= No contribuye al modo común
21	RACM	
20	V1CM	
19	V2CM	

18:15	Reservados	Reservados y asignados a 0
14 13 12 11 10 9	LARLD LLRLD RARLD V1RLD V2RLD CERLD	RLD junta de sumas. Estos interruptores no se abren automáticamente, el usuario debe deshabilitarlos utilizando estos registros. 0 (predeterminado)= No contribuye a la entrada RLD 1 = Si contribuye a la entrada RLD
8	CEREFEN	Referencia de electrodo común 0 = Desactivado 1 = Activado
7:4	RLDSEL [3:0]	Seleccionar derivada de electrodo de referencia 0000 = RLD_OUT 0001 = LA 0010 = LL 0011 = RA 0100 = V1 0101 = V2 0110 a 1111 = Reservados
3	DRVCM	Salida modo común 0 = Desactivado 1 = Activado
2	EXTCM	Seleccionar fuente de modo común 0 = Modo común interno 1 = Modo común externo
1	RLDSEL	Activar referencia de pierna derecha 0 = Desactivado 1 = Activado
0	SHLDEN	Activar "Shield Drive" 0 = Desactivado 1 = Activado

FILTCTL:

Registro de control del filtro.

Dirección 0x0B [Bit 30:Bit 24]

Tabla 7. Registro FILTCTL del ADAS1000

Bit	Nombre	Función
23:6	Reservados	Bits reservados, ajustados a 0
5	MN2K	2 Khz notch bypass para SPI maestro 0 = Notch filter bypassed 1= Notch filter presente
4	N2KBP	2 Khz notch bypass 0 = Notch filter bypassed 1= Notch filter presente
3:2	LPF [1:0]	00 = 40 Hz 01 = 150 Hz 10 = 250 Hz 11 = 450 Hz
1:0	Reservados	Bits reservados, ajustados a 0

FRMCTL:

Registro de control de la cabecera de datos

Dirección 0x0A [Bit 30:Bit 24]

Tabla 8. Registro FRMCTL del ADAS1000

Bit	Nombre	Función
23	LEAD 1	Incluir o excluir de la palabra de datos del ECG.
22	LEAD 2	0 = Incluir en la cabecera
21	LEAD 3	1 = Excluir de la cabecera
20	V1DIS	
19	V2DIS	
18:15	Reservado	Reservado, ajustado a 1111

14	PACEDIS	Detección de ritmo 0 = Incluir en la cabecera 1 = Excluir de la cabecera
13	RESPMDIS	Magnitud de la respiración 0 = Incluir en la cabecera 1 = Excluir de la cabecera
12	RESPPHDIS	Fase de la respiración 0 = Incluir en la cabecera 1 = Excluir de la cabecera
11	LOFFDIS	Estado del electrodo suelto 0 = Incluir en la cabecera 1 = Excluir de la cabecera
10	GPIODIS	GPIO palabra desactivado 0 = Incluir en la cabecera 1 = Excluir de la cabecera
9	CRCDIS	CRC palabra desactivado 0 = Incluir en la cabecera 1 = Excluir de la cabecera
8	Reservado	Reservado, ajustado a 0
7	ADIS	Excluir automáticamente palabras PACEDIS, RESPMDIS, LOFFDIS si no están configurados 0 = Ajustar formato 1 = Auto desactivar palabras
6	RDYRPT	Repetir el "Ready" 0 = Siempre enviar toda la cabecera 1 = Formato de electrodo
5	Reservado	Reservado, ajustado a 0
4	DATAFMT	Configurar formato de los datos de salida: 0 = Formato digital derivada / vector 1 = Electrodo format

3:2	SKIP	Intervalo Skip, sirve para diezmar los datos 00 = Salida para cada dato 01 = Salida cada dos datos 1x = salida por cada cuatro datos
1:0	FRMRATE	Ajustar la velocidad de datos de salida: 00 = 2 kHz 01 = 16 kHz 10 = 128 kHz 11 = 31.25 Hz

ECGCTL:

Registro de control.

Dirección 0x01 [Bit 30:Bit 24]

Tabla 9. Registro ECGTL del ADAS1000.

Bit	Nombre	Función
23	LAEN	Activar canal de ECG, apaga la potencia del canal, la entrada se convierte en alta impedancia 0 = desactivar canal de ECG 1 = Activar canal de ECG
22	LLEN	
21	RAEN	
20	V1EN	
19	V2EN	
18:11	Reservados	Reservados y ajustados a 0
10	CHCONFIG	Configura el "front end" analógico 0 = Entrada "Single-endend" 1 = Entrada diferencial
9	GAIN [1:0]	Ganancia preamplificador y filtro antialiasing 00 = x1.4 01 = x2.1 10 = x2.8 11 = x4.2

7	VREFBUF	Buffer VREF activado 0 = Desactivado 1 = Activado
6	CLKEXT	Uso de un reloj externo en vez del cristal oscilador 0 = XTAL es la fuente 1 = CLK_IO es la fuente de reloj
5	Master	En modo pandilla, selección que es el maestro 0 = Esclavo 1 = Maestro
4	Gang	Activa modo pandilla 0 = modo de canal único 1 = modo pandilla
3	HP	Selecciona el modo ruido/potencia 0 = 1 MSPS, baja potencia 1 = 2 MSPS, alta rendimiento/bajo ruido
2	CNVEN	Conversión activada 0 = idle 1 = conversión activada
1	PWREN	Potencia activada 0 = potencia apagada 1 = potencia activada
0	SWRST	Reinicio por software 0 = NOP 1 = reset

LOFFCTL:

Registro de control de la detección de electrodo suelto.

Dirección 0x02 [Bit 30:Bit 24]

Tabla 10. Registro LOFFCTL del ADAS1000.

Bit	Nombre	Función
23	LAPH	Fase del electrodo suelto en AC 0 = en fase 1 = 180º fuera de fase
22	LLPH	
21	RAPH	
20	V1PH	
19	V2PH	
18	CEPH	
17	LAACLOEN	Activación individual de detección de electrodo suelto en AC 0 = AC detección de electrodo desactivado 1 = AC detección de electrodo activado
16	LLACLOEN	
15	RAACLOEN	
14	V1ACLOEN	
13	V2ACLOEN	
12	CEACLOEN	
[11:9]	Reservado	Reservado, ajustado a 0
[8:7]	ACCURRENT	Ajustar la corriente para AC detección de electrodo suelto 00 = 12.5 nA rms 01 = 25 nA rms 10 = 50 nA rms 11 = 100 nA rms
[6:5]	Reservado	Reservado, ajustado a 0
[4:2]	DCCURRENT	Ajustar niveles de corriente para DC detección de electrodo suelto 000 = 0 nA 001 = 10 nA 010 = 20 nA 011 = 30 nA 100 = 40 nA

		101 = 50 nA 110 = 60 nA 111 = 70 nA
1	ACSEL	Este registro si LOFFEN está a 0 no influye Si LOFFEN está a 1: 0 = DC detección de electrodo suelto activado (según los its [17 :12]) 1 = DC detección de electrodo suelto desactivad. AC detección de electrodo suelto activado.
0	LOFFEN	Activa la detección de electrodo suelto 0 = Detección desactivada 1 = Detección activada

Para este proyecto la información que debe proporcionar el ADAS1000 es la siguiente:

- Los 5 electrodos de ECG.
- Ganancia de x1.4
- Modo común con usando todos los electrodos
- Centro terminal de Wilson interno.
- Filtro paso bajo de 40 Hz (esta aplicación esta orientada al deporte por tanto el ancho de banda es inferior)
- Detección de electrodo suelto

El ADAS1000 envía datos de 32 bits, estos datos siguen la misma estructura que los registros para configurar el ADAS1000 (tabla 2).

Esta comunicación se hace mediante SPI. Para ello se utiliza a veces otro cable que envía el dispositivo esclavo para indicar que el dato a enviar está listo. Este pin se llama READY ('listo' en inglés). El ADAS1000 está configurado para muestrear a 250 Hz, además es el que marca el ritmo de la comunicación con el pin de READY.

El microcontrolador por tanto debe leer cada vez que lea cambios en el pin donde está conectado el READY. Existen dos posibilidades, configurar toda la lectura de datos dentro de un bucle comprobando en cada iteración si ha habido cambio o actuar por interrupción.

Una interrupción es un evento que hace que el microcontrolador deje de ejecutar la tarea que está realizando para atender dicho acontecimiento, y luego regrese y continúe la tarea que estaba realizando antes de que se presentara la interrupción. Por tanto, mediante el uso de interrupciones se mejora el funcionamiento del microcontrolador, ya que no está en todo momento atendiendo a ver si ha habido un cambio y puede dedicar tiempo a otras tareas.

El compilador utilizado (*CCS compiler*) tiene una librería para comunicaciones, donde las funciones SPI están incluidas y la lectura y envío se hace mediante el uso de estas funciones. Esta librería hay que configurarla según las necesidades de la comunicación

La función utiliza para configurar el SPI es `# spi()`. A esta función se le deben pasar todos los parámetros que se quieran cambiar. Si a un parámetro no se le dice cuánto tiene que valer, toma el valor por defecto. Los parámetros que se deben pasar en este caso son:

- Entrada de datos
- Salida de datos
- Señal de reloj
- Selección de chip
- Velocidad de la comunicación
- Tamaño de datos (6.1 los registros son de 32 bits)
- Flanco de reloj que se pasara el bit, en este caso flanco de subida

6.2. Firmware PIC y comunicación bluetooth

Como se ha explicado en el apartado anterior, el ADAS1000 envía 7 datos de 32 bits. Estos datos siempre siguen el mismo orden, por tanto, se realiza 7 lecturas del canal de SPI para adquirir estos datos.

El módulo bluetooth sólo envía datos de 8 bits. Por tanto, los datos se envían byte por byte. Estos bytes se envían en orden para que posteriormente se puedan volver a juntar y obtener toda la información enviada.

La división de byte por byte se realiza mediante el uso de punteros. El puntero fija el dato en una posición fija de la memoria, se lee este registro byte por byte y se envía mediante la función `putc()`. Esta función, que proporciona *CCS compiler*, envía solo un byte.

Todas las funciones del micro deben poder realizarse con una frecuencia de al menos 250 Hz, ya que es la frecuencia en la que el ADAS1000 está captando la señal. Si esto no se cumpliera se estaría ralentizando la captación y obteniendo una frecuencia de muestreo inferior, pudiendo afectar a la correcta captación de la señal y a la consiguiente pérdida de información.

El bluetooth se ha configurado de tal forma que, cuando está transmitiendo información, se desconecta hasta que se le manda la instrucción de enviar de nuevo. De esta forma, el bluetooth envía siempre los datos siguiendo la secuencia programada y no desde el punto en que se conecte. Esto facilita la captación de la información ya que conociendo el orden de llegada de los datos es más fácil juntar de nuevo los bytes.

Esta frecuencia de muestreo se traduce en un periodo de 2.5 ms (Ecuación 1)

$$T_m = \frac{1}{F_m} \rightarrow T_m = \frac{1}{150} = 2.5 \text{ ms} \quad (1)$$

El bluetooth está configurado a una velocidad de 115200 baudios, es decir 115200 bits por segundo. Por tanto, el tiempo que se tarda a enviar los 7 canales (ecuación 2) es de 2 ms (ecuación 3).

$$: N_{canales} * Tama\tilde{n}o_{dato} = Numero\ bits \quad (2)$$

$$7 * 32 = 224\ bits$$

$$\frac{Numerobits}{Velocidad} = Tiempo \quad (3)$$

$$\frac{224}{115200} = 1.94 * 10^{-3} s = 1.94\ ms \cong 2\ ms$$

El canal de SPI se ha configurado a 4 MHz, por tanto, el tiempo que necesita para leer el dato es 80 veces menor.

Esto da un total de 2 ms. Este valor es inferior al máximo permitido, lo que significa que el microcontrolador podrá realizar la tarea sin problemas. En el caso de que no hubiera tiempo suficiente, el microcontrolador tiene una función llamada DMA (*Direct Memory Access*, acceso directo a memoria). Esta función permite realizar tareas en paralelo al CPU, permitiendo realizar funciones mientras se atiende a otras funciones. Por tanto, se podrían leer los datos del canal SPI mientras se envían los datos por bluetooth vía DMA, ganando tiempo y por tanto velocidad de muestreo.

6.3. Funcionamiento del dispositivo

El dispositivo está diseñado para dos modos de funcionamiento, modo *holter* y modo *real time* (solo ha sido implementado el modo *real time*).

Este dispositivo cuenta con un interruptor de 3 posiciones. En la posición central el dispositivo está apagado, en la posición de la izquierda entra en el modo *real time* y en la posición de la derecha en modo *holter*.

Modo *real time*:

Cuando se inicia el dispositivo se hace una comprobación de todos los leds (estos se encenderán en orden) y se escuchara un pitido. Luego se comprobará el estado de la batería y una luz roja parpadeará hasta un máximo de 4 veces indicando el nivel de batería:

- 1 parpadeo la batería esta entre 0 y 25% de su capacidad.
- 2 parpadeos: la batería esta entre el 25% y el 50% de su capacidad.
- 3 parpadeos: la batería esta entre el 50% y el 75% de su capacidad.
- 4 parpadeos: la batería esta entre el 75% y el 100% de su capacidad.

Una vez finalizado las comprobaciones se inicia el bluetooth. El led azul parpadeante indica que el bluetooth está esperando a ser conectado, el dispositivo ahora puede ser vinculado si

no lo está ya. El led verde parpadeante indica que el ADAS1000 está listo para enviar datos, tanto éste como el dispositivo funcionan correctamente.

Una vez vinculado e iniciada la transmisión de datos se apagan los led azul y verde y se enciende el led blanco. Este led indica que se están transmitiendo datos.

Modo holter:

Este modo no está implementado, por tanto, si se inicia, se encienden los leds de iniciación y el dispositivo se pone en modo dormir sin realizar ninguna tarea. Este modo se tiene contemplado y es una función que se implementará en el futuro.

Cargar el dispositivo:

El dispositivo se carga mediante un cable con conector mini USB. Se conecta a la entrada para este tipo de conector mediante un cargador de 5 V y 1 A. Un led rojo al lado del conector se enciende hasta que la batería está cargada. El tiempo de carga estimado es de 50 minutos.

6.4. Software e interfaz grafico

El Software utilizado para procesar la información enviada por el microcontrolador ha sido Matlab. Es un programa matemático con gran capacidad para procesar información, aunque la parte gráfica tiene ciertas limitaciones.

En esta etapa la información recibida por *bluetooth* se ordena siguiendo los pasos inversos que ha hecho el microcontrolador para enviar los datos y recuperar la información. En este caso se ignora el byte que indica la dirección del dato, ya que la información llega en orden y no es necesario conocer de qué dato se trata. Una vez se obtiene la información se guarda en una matriz con todas las derivadas para utilizar posteriormente. Además, se puede elegir y visualizar una derivada y elegir si se quiere ver el ritmo cardiaco.

6.5. Funcionamiento de la interfaz grafico

En este proyecto se ha desarrollado una aplicación en forma interfaz gráfica de usuario para visualizar la información recibida por la señal. Esta interfaz se ha diseñado utilizando la herramienta GUI de Matlab (figura 29).

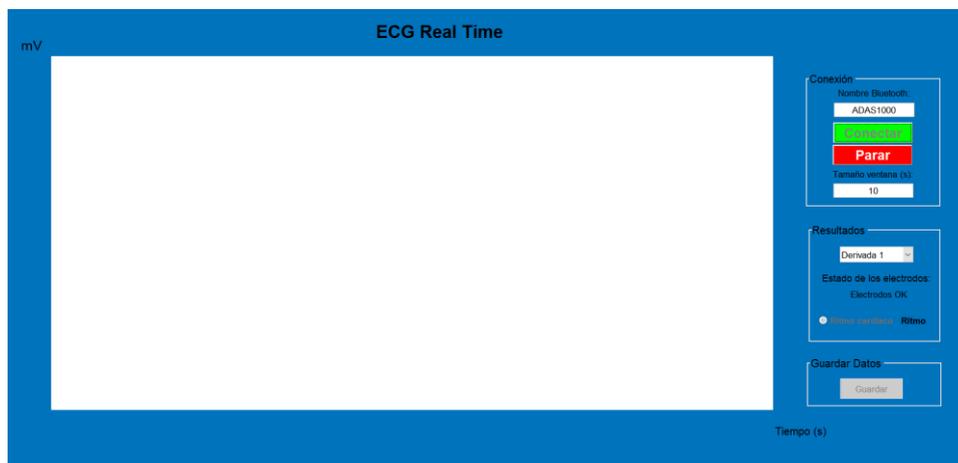


Figura 29. Visualización general de la interfaz.

Funcionamiento de la interfaz:

En primer lugar, el dispositivo debe ser vinculado al ordenador. Una vez conectado se introduce el nombre del dispositivo a vincular en la barra de texto del bloque de Conexión (figura 30). En este caso se ha introducido el nombre del dispositivo "ADAS1000".

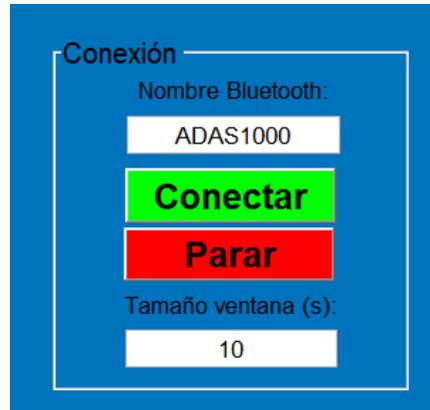


Figura 30. Bloque de conexión

Se tiene la opción de elegir si se quiere ver el ritmo cardiaco o no, esta opción no se puede activar mientras la visualización de datos esté en marcha. Para activar la detección de ritmo cardiaco, en el cuadro de resultados, hay una casilla que se puede marcar y desmarcar. Con la casilla marcada se activa la detección de ritmo cardiaco que se podrá visualizar sobre el texto de la derecha 'Ritmo' (figura 31). Las unidades del ritmo cardiaco en este caso son de latidos por minuto (lpm).



Figura 31. Bloque de resultados

En el bloque de resultados (figura 31), se puede elegir que derivada visualizar, se puede elegir en cualquier momento, antes de la conexión, durante la visualización y después. Las derivadas se eligen en el menú extensible dando a elegir las 5 posibles derivadas que este dispositivo permite: derivada 1, derivada 2, derivada 3, precordial V1 y precordial V2. Antes de iniciar la visualización, en el bloque de conexión, se debe elegir el tamaño de la ventana para visualizar la señal (figura 30). Se debe introducir el tamaño que se quiere visualizar en segundos.

Después de elegir qué derivada se quiere, si se desea visualizar el ritmo cardiaco se puede pulsar la tecla de conectar para conectarse al dispositivo del que se ha introducido el nombre previamente (figura 30). Mientras se espera a que se conecte al dispositivo y se inicie la transmisión, aparece un mensaje indicando que la conexión se está realizando (figura 32).

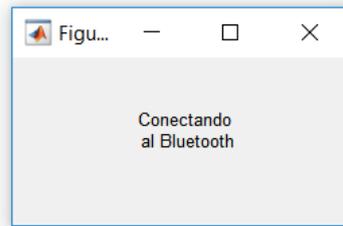


Figura 32. Mensaje de espera.

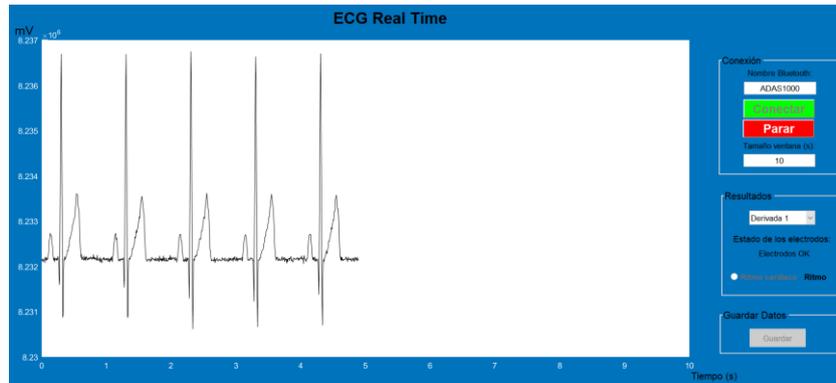


Figura 33. Visualización de la señal

Cuando la conexión se establece, la señal se empieza a visualizar en la pantalla (figura 33). Para parar y desconectar el dispositivo se pulsa la tecla de Parar en el bloque de conexiones (figura 30).

Cuando se para la señal se muestran todas las derivadas a la vez que han sido grabadas, en esta gráfica se puede poner zoom y observar las zonas que se deseen (figura 34).

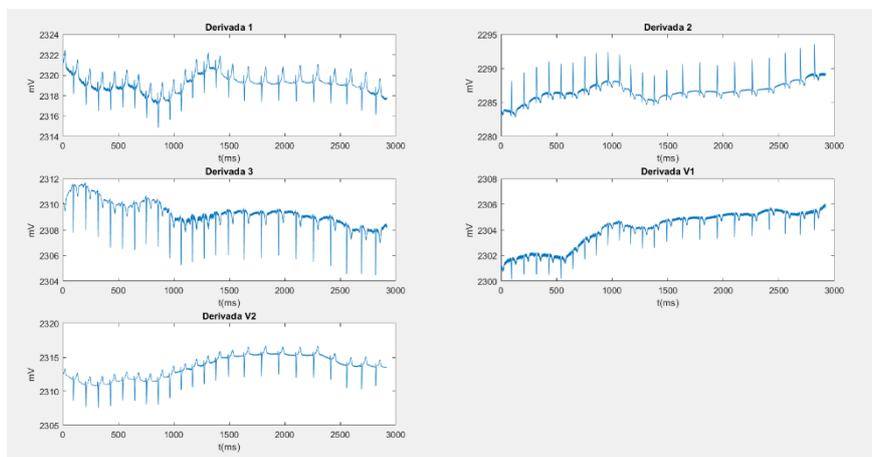


Figura 34. Resultados de la captura de datos

A continuación, se puede guardar en una hoja de cálculo Excel los resultados de las gráficas. Para ello se debe pulsar el botón de guardar. Este botón abre una ventana que permite elegir la ruta en la que se desea guardar el archivo (figura 35). Una vez elegida la carpeta se pulsa "elegir carpeta" y la matriz de datos con las 5 derivadas se guarda automáticamente en una hoja de Excel llamada "ECG_resultados".

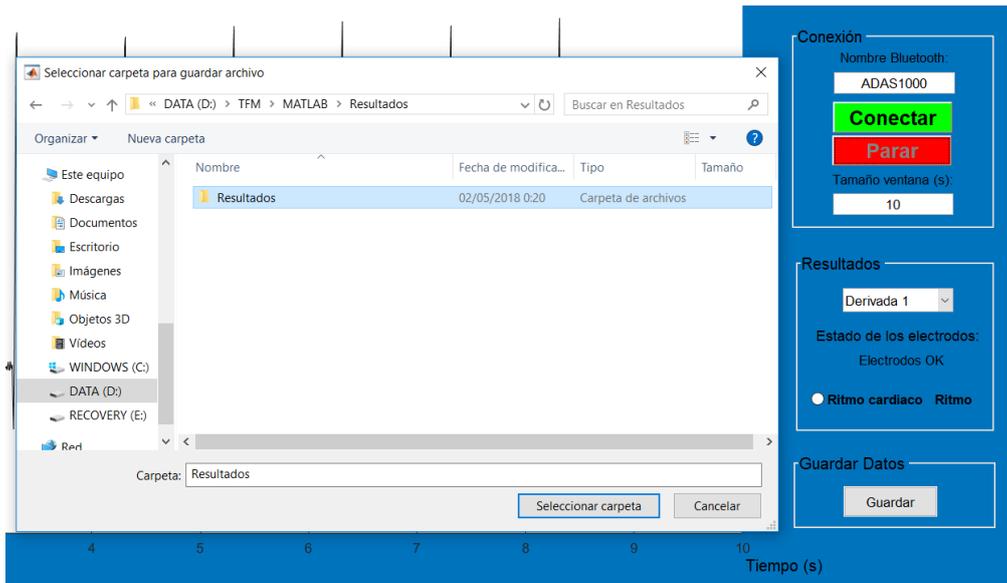


Figura 35. Guardar resultados.

Los botones que no deben ser pulsados cuando no se realiza ninguna acción son bloqueados, por tanto, no dan lugar a error en el funcionamiento. En caso de fallo de conexión suena un sonido de error y el programa debe cerrarse y volverse a abrir para volver a intentar establecer la conexión.

En el panel de resultados se pueden visualizar el estado de los electrodos, mostrando distintos mensajes dependiendo del estado:

- Todos los electrodos conectados: 'electrodos OK' (figura 31).
- Si algún electrodo está desconectado se indica cuál. En la figura 36 y figura 37 se muestra el electrodo RA desconectado, también se puede observar como el electrodo suelto RA afecta a la derivada 1 (figura 36) (RA-LL) pero no a la derivada 3 (figura 37) (LA-LL). Este mensaje puede variar dependiendo de los electrodos que se encuentren mal conectados (figura 38).

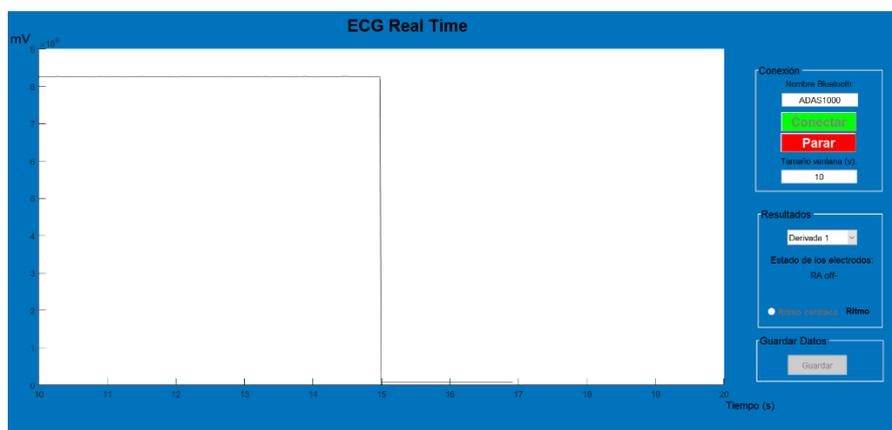


Figura 36. Derivada 1 con el electrodo RA suelto

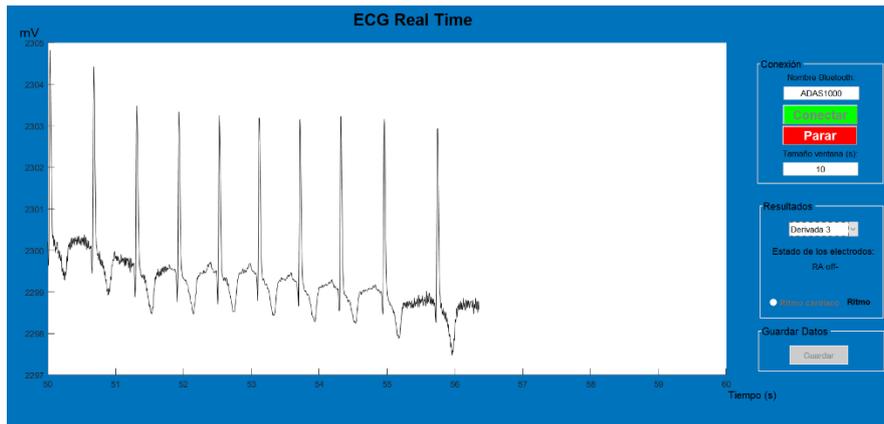


Figura 37. Derivada 3 con el electrodo RA suelto

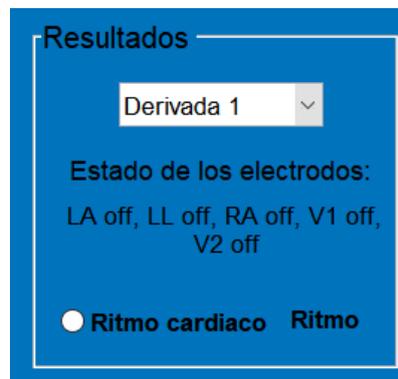


Figura 38. Mensaje de electrodo suelto

7. Procesamiento digital de la señal

El primer paso para el procesamiento de la señal, es juntar los datos, ya que estos son enviados byte por byte. El orden de llegada es el mismo que el de envío, primero se envía el bit de mayor peso (MSB) que corresponde al dato de cabecera, luego los 5 canales de ECG y por último la señal de electrodo suelto. Por tanto, el dato que proporciona informaciones, es un dato de 24 bits.

Los datos son agrupados en el orden correspondiente y ya se tiene los datos listos para el procesamiento ya análisis de la señal.

La señal proveniente del microcontrolador no está debidamente filtrada, ya que los filtros programados son insuficientes para esta tarea. Por ello, mediante un hardware con más capacidad para procesar datos, como es un ordenador o un Smartphone, se pueden procesar estos datos para obtener una señal clara como se ha marcado en el objetivo del proyecto.

Un filtro digital es un sistema discreto que transforma una secuencia de entrada, $x(n)$, en una secuencia de salida $y(n)=\sum h(k)*x(n-k)$, es invariable en el tiempo, lineal, estable y causal. Existen dos tipos de filtros digitales:

-FIR: respuesta impulsional finita (siempre y pueden ser de fase lineal)

-IIR: respuesta impulsional infinita.

Los filtros necesarios son los siguientes:

- Filtro paso bajo (150 Hz ó 40 Hz si es para aplicaciones como el deporte), elimina las interferencias de alta frecuencia.
- Filtro paso alto (0.1 Hz), elimina los movimientos de la línea base.
- Filtro pasa banda (50 Hz), elimina la interferencia de red.

El ADAS1000 tiene estos filtros encargados de procesar la señal y aplica un filtro paso bajo de 40 Hz. Más adelante, en los resultados, se evaluará si estos filtros son suficientes.

Para detectar el ritmo cardiaco se ha utilizado el algoritmo de Pan-Tompkins. Este algoritmo está compuesto por distintas etapas (figura 39).

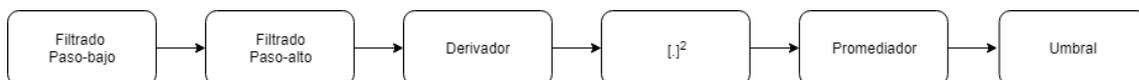


Figura 39. Diagrama de bloques del funcionamiento del algoritmo de Pan y Tompkins

- Filtrado pasa-banda: compuesto por un pasa-alto y pasa-bajo en cascada, que atenúa las bajas frecuencias características de las ondas P y T, así como de las oscilaciones basales, y también atenúa las frecuencias altas del EMG y la interferencia de red.
- Derivación: destaca las pendientes rápidas de la señal, que coinciden en el caso del ECG con el QRS.
- Transformación no lineal: eleva al cuadrado la señal, convirtiéndola en positiva antes de la integración, y acentuando también las frecuencias altas, que corresponden al QRS.

- Promediación: puesto que complejos anormales con amplitudes y pendientes iguales o mayores que los QRS, como es el caso de los PVC, pueden ser detectados como QRS normales, se utiliza un promediador de ventana móvil para obtener la duración del complejo detectado. Esto es porque los PVC suelen presentar duraciones mayores que los QRS normales. La ecuación del promediador viene dada por:

$$y(n) = [x(n - (N - 1)) + x(n - (N - 2)) + \dots + x(n)]/N \quad (4)$$

donde N es el número de muestras de la ventana. La anchura de la ventana debe tomarse de manera que contenga los complejos QRS normales o anómalos, sin llegar a incluir la onda T. El algoritmo fija este valor en 150 ms.

Se definen dos tipos de umbrales, que permiten diferenciar si la detección corresponde a un QRS o a ruido (onda T, EMG, etc.). Cada vez que se detecta un QRS o ruido, los correspondientes umbrales se actualizan con el nuevo valor, por lo que el algoritmo se adapta a los cambios del ECG para cada paciente en particular.

Además, si se produce un fallo de detección (un QRS no es detectado porque el umbral en ese instante es demasiado alto), el algoritmo implementa una técnica de "búsqueda atrás". Para ello, se debe mantener información de la separación normal entre pulsos consecutivos y, si se supera ésta, se vuelve a analizar el tramo anterior con un umbral menor. El algoritmo utiliza un histórico del valor promedio de los últimos 8 intervalos RR normales, y cada vez que se detecta un nuevo QRS se compara el último RR con el promedio. Si está dentro de rango, se actualiza el promedio. Si está fuera de rango, se activa la búsqueda atrás.

Este algoritmo ha sido implementado en Matlab utilizando la función proporcionada por Hooman Sedghamiz [19][20].

8. Resultados

Una vez realizado el montaje, se debe programar el microcontrolador para probar su funcionamiento. Para ello se ha utilizado un programador Pickit 3 conectado a un cable USB que se conecta a la placa. El software utilizado para programar el microcontrolador es el Mplab V8.98.

Para evaluar el funcionamiento del ADAS1000 se ha cargado un programa donde se programa el ADAS1000 para realizar el TESTTONE. El TESTTONE es un registro proporcionado por el ADAS1000 para evaluar su funcionamiento. Utilizando las referencias internas y los convertidores analógicos digital, se utiliza para enviar señales de prueba conocidas:

- Señal senoidal de 150 Hz
- Señal senoidal de 10 Hz
- Onda cuadrada de 1 Hz

Para la prueba se ha elegido la onda cuadrada de 1 Hz y 1 mV de amplitud.

Para recibir los datos se ha utilizado el software Siow (figura 40), este programa recibe y muestra los datos directamente en pantalla. Estos datos pueden estar en ASCII o en hexadecimal, para esta prueba interesa que estén en hexadecimal para juntarlos posteriormente.

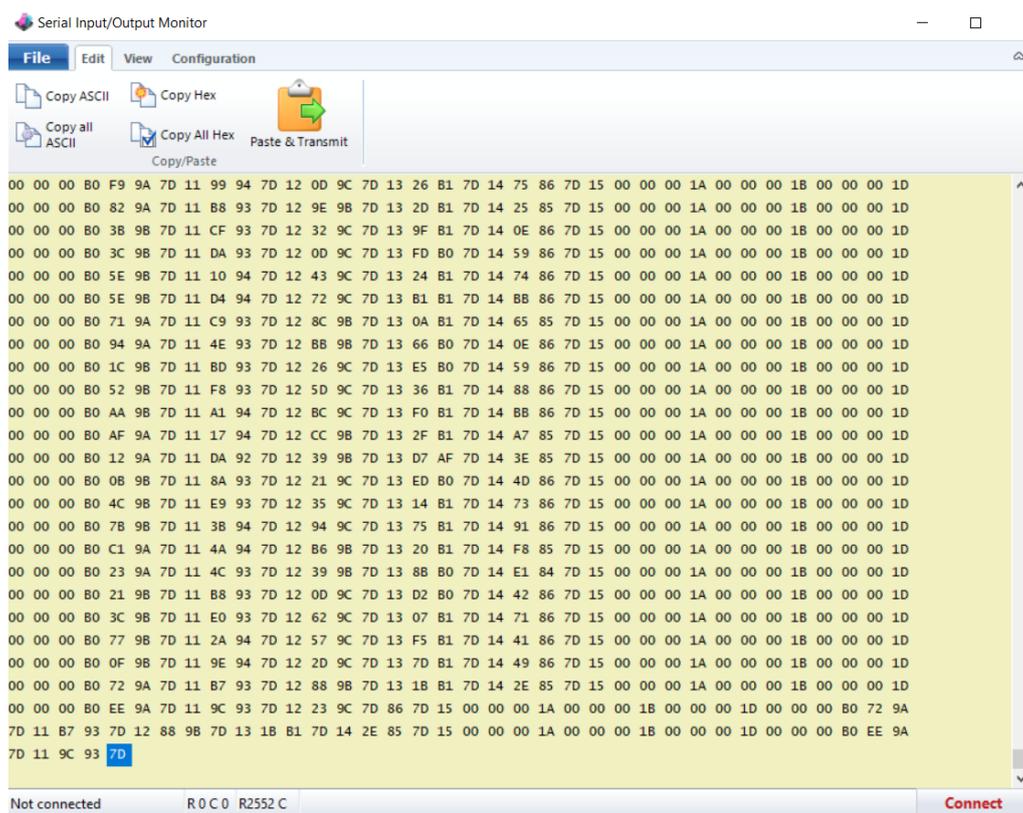


Figura 40. Programa Siow con los datos recibidos del ADAS1000

Estos datos se han copiado a una hoja de cálculo Excel para visualizar si los datos recibidos corresponden a la señal deseada.

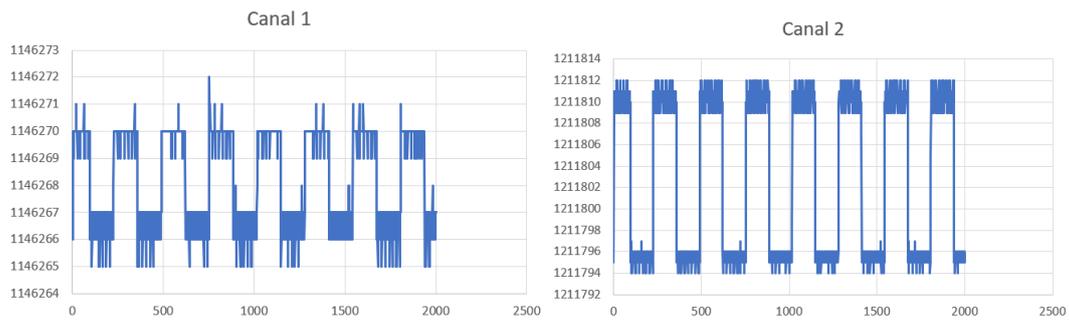


Figura 41. Dos canales con la señal cuadrada de la prueba que proporciona el ADAS1000

Por los canales del ECG se puede visualizar la señal. Esta señal no está escalada por tanto en el eje Y es el valor del bit y en el eje X el número de muestras.

Una vez verificado con el TESTTONE que el ADAS1000 funciona correctamente (figura 41), se prueba con una señal de ECG real.

En primer lugar, esta señal es simulada mediante un paciente virtual. Éste permite un mejor estudio del funcionamiento, ya que el paciente virtual no añade las interferencias que añadiría un sujeto humano. Además, también garantiza una señal estable.

El simulador utilizado es ProSim™ 4, Vital Signs Simulator de FLUKE Biomedical (figura 42). Este dispositivo cuenta con 10 salidas que simulan las derivadas típicas del ECG (las extremidades RA, RL, LA y LL y las precordiales V1, V2, V3, V4, V5 y V6).



Figura 42. ProSim™ 4, Vital Signs Simulator de FLUKE Biomedical. Vista frontal, panel de control, vista lateral, electrodos que simulan las extremidades.

En la figura 43 se muestra los resultados de utilizar el simulador de paciente y la interface, se confirma que funciona correctamente el dispositivo y el interface.

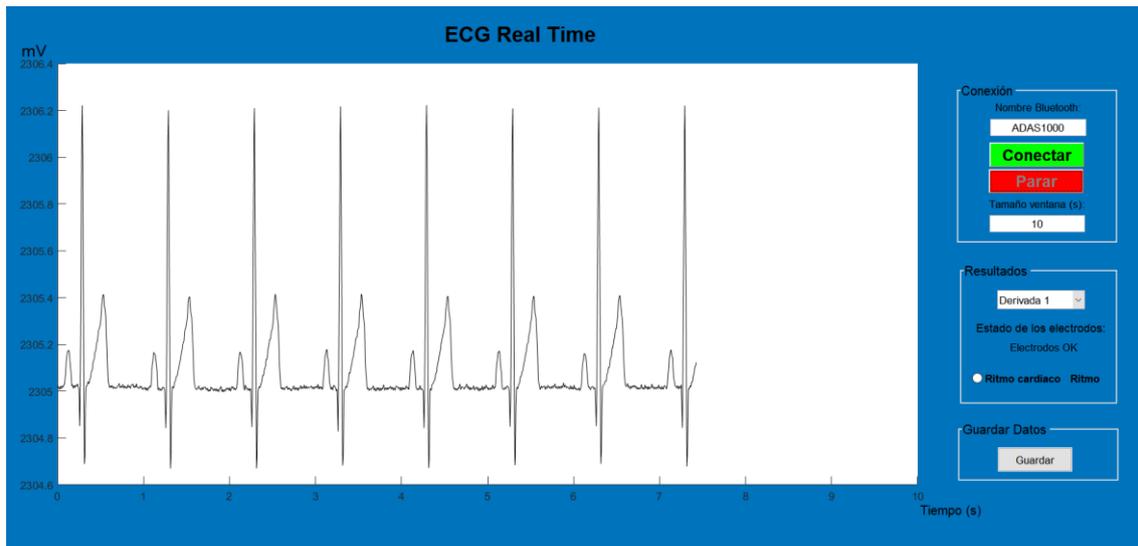


Figura 43. Señal simulada con el ProSim™

Segundo, una vez verificado con el paciente virtual, se evalúa su funcionamiento real, conectando el dispositivo a unos electrodos y estos conectados a un sujeto. Se realizaron dos capturas, una realizando movimientos lentos (figura 44) como andar y otra realizando ejercicio, en este caso sentadillas (figura 45).

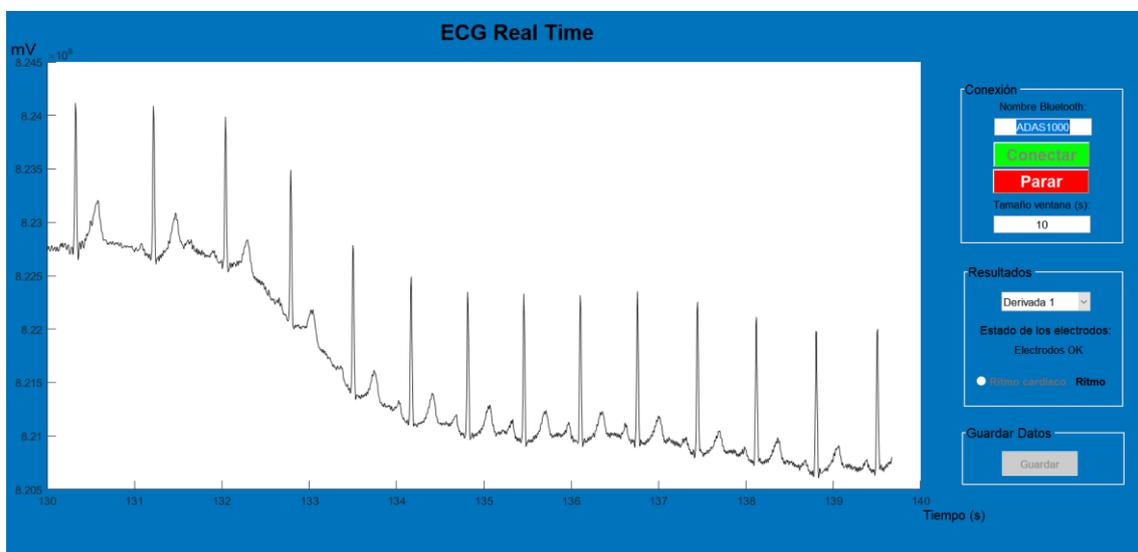


Figura 44. Señal real, captada realizando movimientos lentos.

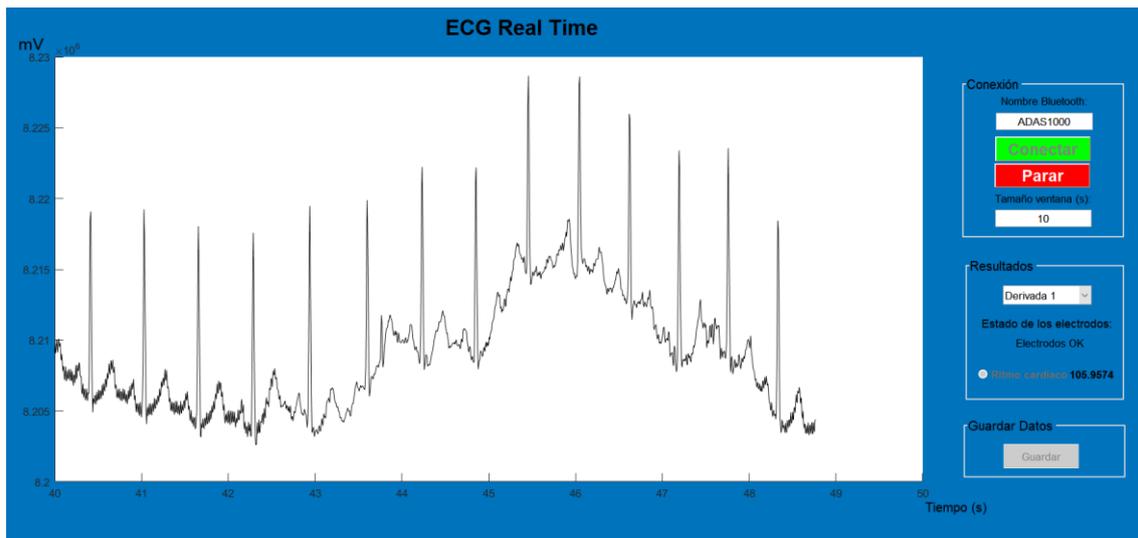


Figura 45. Señal real, captada realizando un ejercicio de sentadillas.

Los resultados muestran que efectivamente se capta la señal de ECG y se identifica el ritmo cardíaco. En las señales más lentas y más estables los filtros del ADAS1000 son suficientes, mientras que en la señal de actividad más elevada la señal cambia más bruscamente para ser filtrada. Aun así, la captación de la señal es correcta.

8.1 Medidas

La resistencia de 1 ohmio situada después de la batería sirve para medir el consumo, la intensidad de circula por esta resistencia es el consumo total de ese instante del dispositivo.

Mediante el uso de un multímetro se ha medido la corriente que circula con el dispositivo encendido y transmitiendo datos. El consumo medido es de 55 mA, este resultado concuerda con el consumo teórico (tabla 3). Siendo este un poco inferior al valor típico estimado. Por tanto la estimación de la batería es la calculada en el apartado 5.2.

9. Conclusiones

El principal objetivo de este trabajo era desarrollar y diseñar un dispositivo de monitorización remota de ECG. Para ello, el diseño se ha llevado a cabo entorno al circuito integrado ADAS1000, un *front end* analógico para el ECG.

En este proyecto se ha diseñado el circuito, los esquemas eléctricos, así como los componentes necesarios para el desarrollo del dispositivo. Asimismo, se ha diseñado la PCB y se han establecido los pasos necesarios para el montaje del mismo. A continuación, se ha programado el microcontrolador para su puesta en marcha y se han realizado diferentes pruebas para garantizar su correcto funcionamiento. Finalmente, se ha realizado un postprocesado de la señal para obtener valores de interés como el ritmo cardíaco.

El diseño se ha hecho acorde a los objetivos y limitaciones existentes, consiguiendo un dispositivo inalámbrico mediante el uso de una batería de litio y mediante la transmisión de la información por vía bluetooth. Por todo ello, este dispositivo es capaz de captar la señal del sujeto a tiempo real y enviarla de forma inalámbrica, evitando en la medida de lo posible la intervención en la actividad del sujeto.

Los resultados se han obtenido monitorizando la señal recibida desde el dispositivo. En esta etapa existen muchas más posibilidades que no se han incluido en los objetivos por la extensión del proyecto, pero sí están explicadas en el apartado de *Líneas Futuras*.

Estos resultados son interesantes para captar el ritmo cardíaco y observar la actividad del corazón de un sujeto en búsqueda de alguna patología. Sin embargo, para su uso el ámbito clínico debería superar una serie de normativas y exigencias más detalladas que las nombradas en este proyecto.

En definitiva, la monitorización de la actividad cardíaca en tiempo real de un paciente es una tarea de gran relevancia en los últimos años debido al incremento considerable de muertes a causa de patologías cardiovasculares. Este proyecto ha intentado dar solución a este problema mediante el dispositivo propuesto, no obstante, la complejidad del problema hace que pueda incluso extenderse a proyectos de investigación de mucha más envergadura.

10. Líneas futuras

Este proyecto ha sido diseñado dentro de las limitaciones existentes. Por eso quedan posibilidades de mejora y funciones pendientes de implementar como las que se enumeran a continuación.

En este proyecto no se han abarcado todas las posibilidades del ADAS1000. Éste tiene capacidad también para medir el ritmo de la respiración y detectar artefactos. En el diseño de referencia [15] se explica su funcionamiento.

Para el software, se ha utilizado Matlab, este programa tiene gran potencia de cálculo, pero la parte gráfica es mejorable. Por ello no se ha podido implementar una interfaz que visualice todas las señales de forma simultánea sin que ralentizase el proceso. Por tanto, el desarrollo desde 0 con un lenguaje de programación dedicado a este tipo de funciones lograría un software más eficiente para esta tarea.

Finalmente, el consumo del dispositivo es elevado debido al módulo *bluetooth*. El *bluetooth* supone el 43% al 51% del consumo total (tabla 3), por tanto, un módulo *bluetooth* u otro tipo de comunicación inalámbrico más apto para sistemas de bajo consumo lograría la extensión de la autonomía del dispositivo. Además, la implementación del modo holter, supondría tener el *bluetooth* apagado y por tanto una autonomía más larga para poder realizar esta tarea que requiere días.

11. Referencias

- [1] OMS, 12 de enero de 2017. Las 10 principales causas de defunción. Disponible en: <http://www.who.int/mediacentre/factsheets/fs310/es/> (consultado el 14 de marzo del 2018)
- [2] Heart Disease and Stroke Statistics, a Report From the American Heart Association. Disponible en: <http://circ.ahajournals.org/content/125/1/e2> (consultado el 14 de marzo del 2018)
- [3] *Diccionario Médico Roche*. Barcelona: Doyma; 1993.
- [4] Polar, pulsímetros. <https://www.polar.com/es/productos> (consultado el 10 de abril del 2018)
- [5] Texas heart institute. Anatomía del corazón. Disponible en: <https://www.texasheart.org/heart-health/heart-information-center/topics/anatomia-del-corazon/> (consultado el 20 de marzo del 2018)
- [6] Barret. 2010. Ganong: *Fisiología Médica*. McGraw Hill: México
- [7] José M. Ferrero Corral, F. Javier Saiz Rodriguez y Antonio Arnau Vives. *Bioelectrónica, señales biológicas*.
- [8] © 2002 Nature Publishing Group Marbán, E. canalopatías cardiacas. *Nature* 415, 213-218 (2002)
- [9] J. Malmivuo and R. Plonsey, *Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields*. Oxford University Press, 1995.
- [10] Medline Plus. Sistema de conducción cardíaco. Disponible en: <https://medlineplus.gov/spanish/ency/anatomyvideos/000021.htm> (consultado el 12 de marzo del 2018)
- [11] J. D. Bronzino and N. R. Michael. *Medical Devices and Systems*.
- [12] Menneses Arévalo, Alexis & Toloza Cano, Daissy. (2012). *Elaboración de las etapas que conforman el circuito de amplificación de los potenciales evocados auditivos*. 10.13140/RG.2.1.4896.8084.
- [13] J. D. Bronzino and N. H. Joachim, "52. biopotential amplifiers," in *Medical Devices and Systems*. CRC Press, Apr. 2006.
- [14] Analog Devices, ADAS1000, Low power, five electrode, electrocardiogram (ECG) Analog Front End.
- [15] Analog Devices, Reference design ADAS1000. <http://www.analog.com/en/design-center/reference-designs/hardware-reference-design/circuits-from-the-lab/cn0308.html> (consultado 12 de diciembre del 2017)
- [16] Microchip, RN42, Class 2 Bluetooth, modulo with EDR support.
- [17] Microchip, PIC24fj128gc10 family, 16 bit flash microcontrollers with 12 bits pipeline A/D sigma-Delta A/D, USB on-the-go and XLP technology.

[18] Microchip, MCP73833/4, Stand-Alone Linear Li-Ion / li-Polymer Charge

[19] Sedghamiz. H, "Matlab Implementation of Pan Tompkins ECG QRS detector.", March 2014.
https://www.researchgate.net/publication/313673153_Matlab_Implementation_of_Pan_Tompkins_ECG_QRS_detect (consultado 28 de abril del 2018)

[20] PAN.J, TOMPKINS. W.J,"A Real-Time QRS Detection Algorithm" IEEE %TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. BME-32, NO. 3, MARCH 1985.

Presupuestos

1. Características del proyecto

Este proyecto es de propósito académico, no se evaluará como si fuera un producto comercial y por tanto no se considerarán aspectos como el número de unidades que se deben fabricar para amortizar la inversión o punto de equilibrio de la inversión. Es decir solo se va a realizar tan solo una estimación del presupuesto de inversión para desarrollar un prototipo de uso académico.

En primer lugar, se calcula el coste de los materiales y recursos utilizados para el montaje y por último el coste de instrumentos y recursos informáticos proporcionados por el departamento y la universidad

2. Costes

2.1 Costes de los Componentes y materiales del dispositivo

Los siguientes componentes forman parte del sistema y se tienen en cuenta para el cálculo del coste del producto.

Componente	Cantidad	Refer.	Métrica	Cantidad Mínima	Precio	Total	Total unidad
Condensador 1 uF	21	2495145	402	21	0,128	2,688	0,128
Condensador 10 uF	4	2218854	402	10	0,21	2,1	0,21
Condensador 0.1 uF	12	9402047	402	12	0,12	1,44	0,12
Condensador 4,7 uF	7	2309027	402	10	0,271	2,71	0,271
Condensador 2,2 uF	4	2346879	402	10	0,321	3,21	0,321
Condensador 15 pF	2	1758945	402	10	0,0076	0,076	0,0076
Condensador 12 pF	2	2611891	402	10	0,0229	0,229	0,0229
Resistencias 1K3	1	2324261	402	10	0,0165	0,165	0,0165
Resistencias 1K	2	2447120	402	10	0,0029	0,029	0,0029
Resistencias 560	3	2059223	402	10	0,0164	0,164	0,0164
Resistencias 220	2	2078857	402	10	0,0111	0,111	0,0111
Resistencias 100 k	2	1469671	402	10	0,0573	0,573	0,0573
Resistencias 47	3	2078852	402	3	0,0054	0,0162	0,0054
Resistencias 10 K	3	2447096	402	10	0,0029	0,029	0,0029
Resistencias 4 M	1	2340632	2010	5	1,17	5,85	1,17
Resistencias 40 K	1	1652794	402	10	0,0396	0,396	0,0396
Resistencias 1 M	3	1469667	402	10	0,0573	0,573	0,0573
Resistencias 2.2 K	1	2447148	402	10	0,0033	0,033	0,0033
Resistencias 8.2 K	1	2140851	402	10	0,0573	0,573	0,0573
Resistencias 100	1	2302528	402	10	0,0299	0,299	0,0299

Resistencias 1	1	2078841	402	10	0,0208	0,208	0,0208
LED 1 y 5 (verde)	2	1525570		2	0,255	0,51	0,255
LED 2 (azul)	1	2217978		2	0,104	0,208	0,104
LED 3 y 6(rojo)	2	2507530		5	0,93	4,65	0,93
LED 4 (blanco)	1	2251493		5	0,27	1,35	0,27
Transistor 1	1	1653622		5	0,119	0,595	0,119
Cristal PIC	1			1		0	0
Conector bateria	1			1		0	0
KSSG33J12	1	2215087		5	2,01	10,05	2,01
Diodo (D1)	1	1056502		5	0,331	1,655	0,331
NCP551SN30	4	2318526		5	0,647	3,235	0,647
MCP73833	1	1332160		1	0,799	0,799	0,799
PIC24FJ128GC006	1	2355355		1	4,96	4,96	4,96
ADAS1000BSTZ	1	2213574		1	49,76	49,76	49,76
Mini usb b type	1	1125348		1	0,587	0,587	0,587
USD	1	2134442		1	4,13	4,13	4,13
Bobinas 4,7 uH	5	2215636		1	0,124	0,124	0,124
Switch 3 posiciones	1			1		0	0
Bluetooth RN42	1	2143310		1	15,58	15,58	15,58
Cristal 8,192 MHz	1	810-2514		1	1,338	1,338	1,338
Total:						121,0032	84,5152

2.2 Horas de trabajo

Desglose de las horas dedicadas en este proyecto:

- Estudio previo del ADAS1000: 25 horas
- Diseño de esquemas: 15 horas
- Diseño PCB: 30 horas
- Montaje del dispositivo: 10 horas
- Programación firmware: 20 horas
- Diseño software monitorización: 50 horas
- Puesta en funcionamiento: 80 horas
- Resultados y pruebas: 5 horas
- Redacción de la documentación: 75 horas

Horas totales = 310 horas.

El precio de la hora de un ingeniero ronda los 35 €/hora.

Precio de las horas de trabajo invertidas = 10850 €

2.3 Coste de los materiales utilizados del diseño y montaje del dispositivo

Estos costes no derivan directamente de los materiales, si no que son los costes de amortización de instrumentos utilizados en el laboratorio y programas informáticos utilizados para desarrollar el prototipo, estos materiales han sido proporcionados por el Departamento de Ingeniería Electrónica de la Universidad Politécnica de Valencia, aun así se debe calcular el coste de amortización correspondiente.

El coste de amortización se calcula a partir del precio/hora de un producto y el tiempo que se ha usado. El precio/hora se calcula a partir del precio del producto dividido por el tiempo de vida útil. Para hacer el cálculo se ha hecho una aproximación de la vida útil de los productos.

Nombre	Precio	Vida útil / licencia	€/h	Horas utilizado	Total
Matlab	69 €	1 año -> 1785 h	0.038	50 h	1.9
Mplab	0 €	-	-	10 h	0
Siow CCS compiler	350 €	5 años -> 8925	0.039	20 h	0.78
Ordenador Hp pavilon	800 €	25.000 h	0.032	165 h	5.28
Pickit 3	40 €	25.000 h	0.0016	5 h	0.008
Electrodos	2 €	1 uso	-	-	2

La suma total del coste de amortización es de 13.04 €

2.4 Presupuesto total

Coste componentes	Coste proceso de desarrollo	Horas de trabajo	Coste total
84.5	13.04	10850	10947.54 €

Anexo I

Código Interface


```

function varargout = interfaz(varargin)
% INTERFAZ MATLAB code for interfaz.fig
%     INTERFAZ, by itself, creates a new INTERFAZ or
raises the existing
%     singleton*.
%
%     H = INTERFAZ returns the handle to a new
INTERFAZ or the handle to
%     the existing singleton*.
%
%
INTERFAZ('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in INTERFAZ.M with the
given input arguments.
%
%     INTERFAZ('Property','Value',...) creates a new
INTERFAZ or raises the
%     existing singleton*. Starting from the left,
property value pairs are
%     applied to the GUI before interfaz_OpeningFcn
gets called. An
%     unrecognized property name or invalid value
makes property application
%     stop. All inputs are passed to
interfaz_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose
"GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
interfaz

% Last Modified by GUIDE v2.5 01-May-2018 22:38:17

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton,
                  ...
                  'gui_OpeningFcn',
@interfaz_OpeningFcn, ...

```

```

        'gui_OutputFcn',
@interfaz_OutputFcn, ...
        'gui_LayoutFcn', [] , ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if narginout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before interfaz is made visible.
function interfaz_OpeningFcn(hObject, eventdata,
handles, varargin)
global pan;
pan=0;
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% varargin   command line arguments to interfaz (see
VARARGIN)

% Choose default command line output for interfaz
handles.output = hObject;

%Hacemos que no se vean los números y rayas del eje:

%Bloquear parar y guardar
set(handles.conectar, 'Enable', 'On');
set(handles.parar, 'Enable', 'Off');
set(handles.guardar, 'Enable', 'Off');
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes interfaz wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

```

```

% --- Outputs from this function are returned to the
command line.
function varargout = interfaz_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning output args (see
VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future
version of MATLAB
% handles     structure with handles and user data (see
GUIDATA)

% Get default command line output from handles
structure
varargout{1} = handles.output;

function bluetooth_name_Callback(hObject, eventdata,
handles)
% hObject     handle to bluetooth_name (see GCBO)
% eventdata   reserved - to be defined in a future
version of MATLAB
% handles     structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
bluetooth_name as text
%           str2double(get(hObject,'String')) returns
contents of bluetooth_name as a double

% --- Executes during object creation, after setting
all properties.
function bluetooth_name_CreateFcn(hObject, eventdata,
handles)
% hObject     handle to bluetooth_name (see GCBO)
% eventdata   reserved - to be defined in a future
version of MATLAB
% handles     empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.

```

```

%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in conectar.
function conectar_Callback(hObject, eventdata,
handles)
global true;
global value_derivada;
global pan;

set(handles.ritmo_cardiaco, 'Enable', 'Off');
set(handles.conectar, 'Enable', 'Off');
set(handles.parar, 'Enable', 'On');
LSB=2.8*10^-4;
true=1;
cla
value_derivada=get(handles.derivadas,'Value')
hh = figure('units','pixels','position',[500 500 200
100],'windowstyle','modal');
uicontrol('style','text','string','Conectando al
Bluetooth','units','pixels','position',[75 20 60 50]);

nombre_bt=get(handles.bluetooth_name,'String')
bt=Bluetooth(nombre_bt, 1);
fopen(bt);

close(hh)

tiempo=str2double(get(handles.tiempo_ventana,'String')
) %Segundos
tiempo1=0;
tiempo2=tiempo;
xlim([0 tiempo]);
fm=166; %Frecuencia de muestreo en herzios

%Init display
h1 = animatedline;
h2 = animatedline;
h3 = animatedline;
h4 = animatedline;
h5 = animatedline;

```

```

x = linspace(0,tiempo,tiempo*fm);

k=1;
contador=0;

l1='';
l2='';
l3='';
l4='';
l5='';

dato1=zeros(4,1);
dato2=zeros(4,1);
dato3=zeros(4,1);
dato4=zeros(4,1);
dato5=zeros(4,1);
dato6=zeros(4,1);
ritmo=zeros(1,500);
if pan==1
    for i=1:500
        dato1(:,1)=fread(bt,4);
        dato2(:,1)=fread(bt,4);
        dato3(:,1)=fread(bt,4);
        dato4(:,1)=fread(bt,4);
        dato5(:,1)=fread(bt,4);
        dato6(:,1)=fread(bt,4);
        ritmo(1,1)=hex2dec([dec2hex(dato1(3,1),2)
dec2hex(dato1(2,1),2) dec2hex(dato1(1,1),2)]);
        ritmo=circshift(ritmo,1);
    end
end

tic
while (true==1)
    dato1(:,1)=fread(bt,4);
    dato2(:,1)=fread(bt,4);
    dato3(:,1)=fread(bt,4);
    dato4(:,1)=fread(bt,4);
    dato5(:,1)=fread(bt,4);
    dato6(:,1)=fread(bt,4);

    plead1=hex2dec([dec2hex(dato1(3,1),2)
dec2hex(dato1(2,1),2) dec2hex(dato1(1,1),2)]);

```

```

    plead2=(hex2dec([dec2hex(dato2(3,1),2)
dec2hex(dato2(2,1),2) dec2hex(dato2(1,1),2)]))*LSB;
    plead3=(hex2dec([dec2hex(dato3(3,1),2)
dec2hex(dato3(2,1),2) dec2hex(dato3(1,1),2)]))*LSB;
    pleadV1=(hex2dec([dec2hex(dato4(3,1),2)
dec2hex(dato4(2,1),2) dec2hex(dato4(1,1),2)]))*LSB;
    pleadV2=(hex2dec([dec2hex(dato5(3,1),2)
dec2hex(dato5(2,1),2) dec2hex(dato5(1,1),2)]))*LSB;

    lead(1,k)=plead1*LSB;
    lead(2,k)=plead2;
    lead(3,k)=plead3;
    lead(4,k)=pleadV1;
    lead(5,k)=pleadV2;

    if pan==1
    ritmo(1,1)=plead1;
    ritmo=circshift(ritmo,1);
    [~,b,~]=pan_tompin(ritmo, fm, 0);
    rate=(fm/(b(2)-b(1)))*60;
        if contador>=166

set(handles.ritmo_display, 'String', num2str(rate));
        contador=0;
        end
        contador=contador+1;
    end
    plead=plead1*LSB;
    if (dato6(3,1)~=0)
        binario=de2bi(dato6(3,1),8);
        if binario(1)
            l1='LA off-';
        end
        if binario(2)
            l2='LL off-';
        end
        if binario(3)
            l3='V2 off-';
        end
        if binario(4)
            l4='V1 off-';
        end
        if binario(5)
            l5='RA off-';
        end
    end
end

```

```

set(handles.lead_off, 'string', strcat(l1,l2,l3,l4,l5));
    else
        set(handles.lead_off, 'string', 'Electrodos
OK');
        l1='';
        l2='';
        l3='';
        l4='';
        l5='';
    end
    switch value_derivada
        case 1
            addpoints(h1,x(k),plead1);
        case 2
            addpoints(h2,x(k),plead3);
        case 3
            addpoints(h3,x(k),plead2);
        case 4
            addpoints(h4,x(k),pleadV1);
        case 5
            addpoints(h5,x(k),pleadV2);
    end
    drawnow limitrate
    if(k==length(x))
        %k=0;
        toc
        tic
        tiempo1=tiempo1+tiempo;
        tiempo2=tiempo2+tiempo;
        x=linspace(0,tiempo2,tiempo2*fm);
        xlim([tiempo1 tiempo2]);
    end
    k=k+1;
end
drawnow

set(handles.grafical1, 'UserData', lead);

figure
subplot(3,2,1);
plot(lead(1,:));
title('Derivada 1');
xlabel('t (ms)');
ylabel('mV');

```

```

subplot(3,2,2);
plot(lead(2,:));
title('Derivada 2');
xlabel('t(ms)');
ylabel('mV');

subplot(3,2,3);
plot(lead(3,:));
title('Derivada 3');
xlabel('t(ms)');
ylabel('mV');

subplot(3,2,4);
plot(lead(4,:));
title('Derivada V1');
xlabel('t(ms)');
ylabel('mV');

subplot(3,2,5);
plot(lead(5,:));
title('Derivada V2');
xlabel('t(ms)');
ylabel('mV');

fclose(bt);
delete(bt);
clear bt;
% hObject    handle to conectar (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% --- Executes on button press in parar.
function parar_Callback(hObject, eventdata, handles)
global true;
set(handles.conectar, 'Enable', 'On');
set(handles.parar, 'Enable', 'Off');
set(handles.guardar, 'Enable', 'On');
set(handles.ritmo_cardiaco, 'Enable', 'On');
true=0;
% hObject    handle to parar (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB

```

```
% handles      structure with handles and user data (see
GUIDATA)
```

```
% --- If Enable == 'on', executes on mouse press in 5
pixel border.
```

```
% --- Otherwise, executes on mouse press in 5 pixel
border or over parar.
```

```
function parar_ButtonDownFcn(hObject, eventdata,
handles)
```

```
% hObject      handle to parar (see GCBO)
```

```
% eventdata    reserved - to be defined in a future
version of MATLAB
```

```
% handles      structure with handles and user data (see
GUIDATA)
```

```
% --- Executes on button press in checkbox1.
```

```
function checkbox1_Callback(hObject, eventdata,
handles)
```

```
% hObject      handle to checkbox1 (see GCBO)
```

```
% eventdata    reserved - to be defined in a future
version of MATLAB
```

```
% handles      structure with handles and user data (see
GUIDATA)
```

```
% Hint: get(hObject,'Value') returns toggle state of
checkbox1
```

```
function tiempo_ventana_Callback(hObject, eventdata,
handles)
```

```
% hObject      handle to tiempo_ventana (see GCBO)
```

```
% eventdata    reserved - to be defined in a future
version of MATLAB
```

```
% handles      structure with handles and user data (see
GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of
tiempo_ventana as text
```

```
%          str2double(get(hObject,'String')) returns
contents of tiempo_ventana as a double
```

```

% --- Executes during object creation, after setting
all properties.
function tiempo_ventana_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to tiempo_ventana (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in derivadas.
function derivadas_Callback(hObject, eventdata,
handles)
global value_derivada;
value_derivada=get(handles.derivadas,'Value')
% hObject    handle to derivadas (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% Hints: contents = cellstr(get(hObject,'String'))
returns derivadas contents as cell array
%         contents{get(hObject,'Value')} returns
selected item from derivadas

% --- Executes during object creation, after setting
all properties.
function derivadas_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to derivadas (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

```

```

% Hint: popupmenu controls usually have a white
background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in guardar.
function guardar_Callback(hObject, eventdata, handles)
carpeta_guardar = uigetdir('','Seleccionar carpeta
para guardar archivo');
filename =
strcat(carpeta_guardar,'\ECG_resultados.xlsx')
variableconlosdatos=get(handles.grafical,'UserData');
xlswrite(filename,variableconlosdatos)
% hObject    handle to guardar (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% --- Executes on button press in ritmo_cardiaco.
function ritmo_cardiaco_Callback(hObject, eventdata,
handles)
global pan
pan=1;
% hObject    handle to ritmo_cardiaco (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of
ritmo_cardiaco

```


Anexo II

Código Microcontrolador


```

#device PIC24FJ128GC006

#include <24FJ128GC006.h>

/*****
*/

#include "main.h"

/*****
*/

#use spi(MASTER, DI=PIN_D1, DO=PIN_D3, CLK=PIN_D2, ENABLE=PIN_D5, SAMPLE_RISE,
BITS=32, ENABLE_ACTIVE=0, IDLE=1, MSB_FIRST, STREAM=ADAS_SPI)

/*****
*/

void init_ADAS1000(void){

    //pin_select("INT2",ADAS_SPI_DRDY_PIN);
    delay_ms(300);
    output_low(ADAS_PWDN);//SET_PD;
    delay_ms(100);
    output_high(ADAS_PWDN);//CLEAR_PD;
    delay_ms(100);
    output_low(ADAS_RST);//SET_RESET;
    delay_us(1500);
    output_high(ADAS_RST); //CLEAR_RESET;
    delay_ms(100);

//setup test tone
/*
    spi_xfer(ADAS_SPI, 0x8500000B, 32); //CMREFCTL
    delay_us(1);
    spi_xfer(ADAS_SPI, 0x88F8001D, 32); //TESTTONE
    delay_us(1);
    spi_xfer(ADAS_SPI, 0x8B000008, 32); //FILTCTL
    delay_us(1);
    spi_xfer(ADAS_SPI, 0x8A07F61C, 32); //FRMCTL

```

```

delay_us(1);
spi_xfer(ADAS_SPI, 0x81F800AE, 32); //ECGCTL
delay_us(1);
spi_xfer(ADAS_SPI, 0x40000000, 32); //FRAMES
delay_us(1);
*/
    spi_xfer(ADAS_SPI, 0x00000000, 32); //NOP_
delay_us(1);
    spi_xfer(ADAS_SPI, 0x85E0000B, 32); //CMREFCTL
delay_us(1);
spi_xfer(ADAS_SPI, 0x8B000000, 32); //FILCTL LPF 450Hz //FILCTL LPF 150 Hz
delay_us(1);
spi_xfer(ADAS_SPI, 0x8A07F61C, 32); //FRMCTL: Heades; 5 ECG chanel, Lead off (7 chanel)
delay_us(1);
spi_xfer(ADAS_SPI, 0x81F804AE, 32); //ECGCTL
delay_us(1);
    spi_xfer(ADAS_SPI, 0x82000015, 32); //LOFFCTL
delay_us(1);
spi_xfer(ADAS_SPI, 0x40000000, 32); //FRAMES
delay_us(1);

}
void send32bitUart(int32 val){
    // With a pointer
    int8 *p;

    p = &val;
    putc(*p++);
    putc(*p++);
    putc(*p++);
    putc(*p++);
}

```

```

    //putc('\r');
    delay_us(1);
}
void main(void)
{
    int32 header;
    int32 lead_1;
    int32 lead_2;
    int32 lead_3;
    int32 lead_V1;
    int32 lead_V2;
    int32 lead_off;
    int32 respiration;
    int32 pace;
    int8 dato;
    int diezmar=0;

    init();

    /******* Conmutador ¿modo Holter o modo Real Time? *****/
        OFF( GN );
    delay_ms(100);
    if(input(RT)) Holter=0;
    else Holter=1;
    output_float(GN);
modo:
    beep(100,2);
    blink(125);
    if(Holter==0)
    {
        ON( led_AZUL);
    }
}

```

```

        ON( pwr_BT );
        delay_ms(500);
        ON( BT_RST );
        delay_ms(200);
        OFF(led_AZUL);
    }
else
    {
        ON( led_VERDE);
        OFF( pwr_BT );
        delay_ms(700);
        OFF(led_VERDE);
    }

set_adc_channel(Bat); delay_ms(100);
battery = read_adc();
while(LED_nivel_BATERIA()==1);
if(Holter == 1) if(battery < BAT_UMBRAL_5 )    goto dormir_profundamente;
if(Holter == 0) if(battery < (BAT_UMBRAL_5 - RmH )) goto dormir_profundamente;

if(Holter == 1)// No esta implementado
{
    all_leds_on();
    beep(2000,1);
    delay_ms(10000);
    goto dormir_profundamente;
}

if(Holter == 0){ // Real-time: adquisicion y comunicacion BT

initBT();

```

```

init_ADAS1000();
com:
    output_low(led_BLANCO);
    while(!input(BT_Connect))
    {
        delay_ms(100);
        output_toggle(led_AZUL);
        if(input(PIN_D4)) output_toggle(led_ROJO);
        else output_toggle(led_VERDE);
    }
    diezmar=0;
    output_low(led_AZUL);
    output_low(led_VERDE);
    output_high(led_BLANCO);
    for(;;){
        if(input(PIN_D4))
        {
            delay_ms(1);
        }
        else
        {
            delay_us(1);
            header = spi_xfer(ADAS_SPI,0x00000000,32);
            delay_us(1);
            lead_1= spi_xfer(ADAS_SPI,0x00000000,32);
            delay_us(1);
            lead_2= spi_xfer(ADAS_SPI,0x00000000,32);
            delay_us(1);
            lead_3= spi_xfer(ADAS_SPI,0x00000000,32);
            delay_us(1);
            lead_V1= spi_xfer(ADAS_SPI,0x00000000,32);

```

```

delay_us(1);
lead_V2= spi_xfer(ADAS_SPI,0x00000000,32);
delay_us(1);
//pace= spi_xfer(ADAS_SPI,0x00000000,32);
//delay_us(1);
//respiration= spi_xfer(ADAS_SPI,0x00000000,32);
//delay_us(1);
lead_off= spi_xfer(ADAS_SPI,0x00000000,32);
delay_us(1);

    if(diezmar==3)
    {
        //send32bitUart(header);
        send32bitUart(lead_1);
        send32bitUart(lead_2);
        send32bitUart(lead_3);
        send32bitUart(lead_V1);
        send32bitUart(lead_V2);
        //send32bitUart(pace);
        //send32bitUart(respiration);
        send32bitUart(lead_off);
        diezmar=0;
    }
    diezmar=diezmar+1;
}

        if(!input(BT_Connect)) goto com;
}

dormir_profundamente: // De aqui ya no sale NUNCA
    all_leds_on();

```

```
beep(800,3);  
OFF( PIN_BUZZER);
```

```
OFF( pwr_ANLG );  
OFF( pwr_uSD );  
OFF( pwr_BT );
```

```
all_leds_OFF();
```

```
// Deep Sleep: solo funciona el RTC
```

```
#asm
```

```
bset 0x0758,15
```

```
bset 0x0758,15
```

```
pwrsav 0
```

```
#endasm
```

```
}
```

```
/*  
*/
```