

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
INFORMÁTICA

Grado en Ingeniería de la Salud
Mención en Ingeniería Biomédica

Holter basado en placa de desarrollo hardware Arduino
Holter Based on Arduino shields

Realizado por
Antonio Rodríguez Arjona
Tutorizado por
Rafael Jesús Navas González
Departamento
Departamento de Electrónica

UNIVERSIDAD DE MÁLAGA
MÁLAGA, junio 2019

Fecha defensa:

Fdo.: El Secretario del Tribunal

Agradecimientos:

A D. Rafael, por su apoyo y por haber sabido valorar el esfuerzo y la dedicación.

A D. Julián, y a todo el personal del departamento de Electrónica.

Al equipo técnico de Electromedicina del Hosp. Infanta Margarita de Cabra (Córdoba)

A mi familia, especialmente a mi padre, por su paciencia y por el esfuerzo realizado.

A Sara por esperar siempre lo mejor de mí.

A Dios, por ser el aliento de esperanza en los momentos más difíciles.

Resumen:

En este Trabajo Fin de Grado (TFG) se ha estudiado, diseñado e implementado un sistema básico para electrocardiografía ambulatoria (sistema Holter de monitorización) esto es, un sistema que permite el registro, almacenamiento, y análisis de la actividad eléctrica del corazón.

El sistema que se ha diseñado se basa en la placa comercial E-Health para Arduino (www.cooking-hacks.com). En él, junto al acondicionamiento, adquisición y almacenamiento de la señal, se ha dado especial importancia a dotar al sistema de conectividad con dispositivos móviles (BlueTooth), que permiten su configuración a distancia, y la transmisión de la información almacenada en el dispositivo, para su visualización en una aplicación (APP Android) diseñada para tal fin. Finalmente, como apoyo al análisis, se ha diseñado una aplicación en MatLab para facilitar el análisis de la señal ECG registrada, siguiendo la práctica médica.

Se trata pues de un trabajo en el que se han afrontado problemas muy diversos, desde los derivados de la propia implementación, tanto hardware como software: optimización de código y gestión de la memoria, interfaz manual, diseño de PCB; hasta cuestiones propias de la comunicación y conectividad entre dispositivos, programación APP, hasta la programación en lenguajes tipo MatLab.

Palabras Claves: Holter, Arduino, eHealth, electrocardiograma, bluetooth, Android, Matlab, tarjeta de memoria SD, electromedicina, monitorización

Abstract:

In this Work, a basic ambulatory system for electrocardiography (Holter monitor) is studied, designed and implemented. This device records and saves the electrical activity of heart, in order to be analyzed.

The designed system is based on E-Health shield which is a supporting feature for Arduino. Besides, bluetooth connectivity is added for the purpose of share information with smartphones, allowing the configuration from distance and to get files from a mobile app in an Android device. Finally, backing the analysis tools, a function on Matlab makes easier the interpretation of results.

This Work covers different issues, since hardware and software implementation and optimization, to subject related to communication and connectivity between devices, app programming and Matlab representations.

Keywords: Holter, Arduino, eHealth, electrocardiogram, bluetooth, Android, Matlab, SD Memory Card, electromedicine, monitoring

Índice

1. Introducción.....	11
1.1. Motivación.....	12
1.2. Objetivos.....	13
1.3. Estructura del documento.....	14
1.4. Elementos necesarios para la implementación del proyecto.....	15
2. Fundamentos de la Electrocardiografía.....	17
2.1. Generalidades sobre el corazón y el músculo cardíaco.....	17
2.2. Vectores de polarización.....	19
2.3. Electrocardiograma.....	21
2.4. Particularidades de la señal de ECG.....	23
3. ECG ambulatoria.....	25
3.1. Origen y evolución histórica.....	25
3.2. Características del Holter convencional.....	27
4. Especificación y metodología del proyecto.....	31
4.1. Requerimientos propios del dispositivo.....	31
4.2. Metodología.....	32
4.3. Propuesta de prototipo y concreción del proyecto.....	33
5. Implementación.....	37
5.1. Hardware.....	37
5.1.1. Arduino UNO.....	38
5.1.2. Placa multisensorial eHealth.....	40
5.1.3. Electroodos.....	43
5.1.4. Componentes de interfaz de usuario.....	44
5.1.5. Memoria SD.....	47
5.1.6. Conectividad Bluetooth.....	49
5.2. Software.....	52
5.2.1. Codificación de Arduino e interfaz manual.....	53
5.2.2. Interfaz Bluetooth y aplicación Android.....	57

5.2.3. Matlab.....	61
6. Conclusiones.....	67
7. Líneas Futuras.....	71
7.1. Diseño e implementación del circuito completo de ECG.....	71
7.2. Combinación de Holter de tensión y Holter convencional.....	71
7.3. Aplicación mejorada.....	72
7.4. Dispositivo Holter basado en Raspberry Pi.....	72
7.5. Análisis algorítmico a través de Matlab.....	73
ANEXOS.....	75
I. Código de Arduino	
II. Código de Matlab	
III. Archivo APK de la aplicación móvil HolterArduino	
IV. Desarrollo de la Placa de Circuito Impreso (PCB)	
V. Negativos de la Placa de Circuito Impreso (PCB)	
VI. Comentarios sobre los resultados obtenidos	
VII. Imágenes del sistema en funcionamiento	
REFERENCIAS.....	93
BIBLIOGRAFÍA GENERAL.....	95

1. Introducción

En este Trabajo de Fin de Grado (TFG) se ha implementado un sistema básico de electrocardiografía ambulatorio (sistema Holter de monitorización) para el registro de la actividad eléctrica del corazón. Así, se hace necesario el estudio del potencial eléctrico detectable producido en el corazón, los elementos de transducción para obtener la señal, el acondicionamiento de señal con el fin de obtener los datos útiles para la graficación del electrocardiograma, así como la gestión de dichos datos. En este caso, la gestión de la información consiste en el almacenamiento en una tarjeta de memoria SD (y gestión de la misma) y posterior transmisión de dicha información a través de bluetooth.

Todo el desarrollo, se hace con la pretensión de mantener intactas las cualidades más básicas de un dispositivo Holter, incluyendo aquí, por supuesto, la cualidad de portabilidad y la inclusión de una herramienta para interpretación y diagnóstico. Se trata de un proyecto completo en el que se afrontan problemas de muy diverso tipo: dificultades derivadas directamente de la implementación, problemas de optimización de memoria y código, y también, cuestiones propias de la comunicación y gestión de la información. De este modo, el trabajo de diseñar un dispositivo de estas características se enmarca en el campo de la electromedicina y la e-salud, dos de las ramas clásicas de lo que se conoce como Ingeniería Biomédica.

Por otro lado, y de una manera más concreta, en este primer capítulo se realiza una presentación general sobre qué se va a abordar a lo largo de este documento. Del mismo modo, se reflejarán las distintas motivaciones que nos empujan y animan a adentrarnos en el desarrollo de este proyecto y realizar una definición clara de los objetivos que se desean alcanzar. Por último, se explicará la estructura del documento y se determinarán los distintos elementos necesarios para la implementación completa del dispositivo y sus funcionalidades.

1.1. Motivación

Las enfermedades cardiovasculares son algo que, por desgracia, están muy presentes en la sociedad actual. En numerosos casos no es suficiente el empleo de la electrocardiografía para su diagnóstico, debido a que los eventos extraordinarios se pueden (y de hecho se suelen) dar en momentos muy puntuales del día a día del paciente. Esto dificulta su registro, ya que para ello debe coincidir la ocurrencia de dicho evento con el corto periodo de tiempo en el que transcurre el desarrollo de la prueba de diagnóstico.

El problema de registrar los eventos extraordinarios se resuelve fácilmente utilizando un sistema de Electrocardiografía Ambulatoria, es decir, un dispositivo Holter. Estos dispositivos suponen cada vez en mayor medida instrumentos básicos en las consultas de cardiología, por lo que están avanzando a pasos agigantados, reinventándose en cuanto a funcionalidad y comodidad.

Desde este proyecto se plantea la hipótesis de que Arduino, en su versión de Arduino UNO (uno de los más básicos) es capaz de soportar todos los requisitos que debe cumplir un equipo básico de electrocardiografía ambulatoria. Así, se diseña un dispositivo Holter basado en esta placa de desarrollo hardware, la cual se ha elegido por su sencillez en cuanto a diseño y especificaciones, por el deseo e inquietud de explotar al máximo este sistema tan usado en las prácticas de varias materias del Grado, así como por estar más que reconocido como un sistema clave en la docencia, dentro de los campos de la ingeniería y las tecnologías. Además, Arduino UNO es un sistema económico y que a la vez proporciona diversas ventajas en cuanto a versatilidad se refiere, debido al amplio abanico de posibilidades que ofrece, incluyendo la compatibilidad tan extensa con otros módulos, como es la placa a eHealth, otra pieza clave en el proyecto.

Por otro lado, se nos presentan diversos retos. El primero de ellos deriva directamente de la opcionalidad de recursos, ya que lo ideal es usar simplemente los elementos disponibles en el departamento de Electrónica de la Universidad de Málaga. Por otro lado, encontramos limitaciones implícitas en el uso del propio

Arduino UNO, pues la sencillez de su procesador o el escaso número de pines disponibles nos obliga a realizar diversas tareas de optimización.

1.2. Objetivos

Así, en base a esos motivos, y en consecuencia a los retos presentados, se proponen los objetivos a cumplir durante el desarrollo de este proyecto.

El más importante de estos objetivos es conseguir un dispositivo que se atenga a la definición estricta de Holter: siendo portátil, capaz de recabar la señal de ECG y almacenarla con el fin de representarla posteriormente para emitir un diagnóstico en base a ello. Todo esto se debe conseguir basándonos en la placa de Arduino UNO y el circuito de acondicionamiento del módulo de sensores eHealth. La simplicidad de ambos sistemas frente a las características que debe ofrecer un dispositivo de electrocardiografía ambulatoria nos obliga a plantear la hipótesis de si realmente esto es posible.

Además, aprovechando las facilidades en cuanto a compatibilidad con módulos de conectividad de distinta naturaleza propias de Arduino, se incluye, a modo de funcionalidad extra, la conexión bluetooth. No es frecuente encontrar dispositivos de ECG ambulatoria con conectividad bluetooth. Sin embargo, puede resultar interesante de cara a realizar una comprobación de correcto funcionamiento una vez colocado sobre el cuerpo del paciente y antes de que éste abandone la consulta.

Por otro lado, estaremos constantemente trabajando con uno de los biopotenciales más importantes en cuanto a pruebas diagnósticas se refiere, el ECG. Por ello, otro objetivo fundamental a alcanzar será la comprensión de sus características y su naturaleza.

Finalmente, y de un modo más general, este proyecto, comprendido como Trabajo de Fin de Grado, tiene como objetivo fundamental el culminar la capacidad de desarrollo e implicación de habilidades, combinando la creatividad personal y los conocimientos adquiridos durante el Grado. Así, se recorren campos de muy diverso tipo con el fin de solucionar los distintos problemas que surgen a lo largo del desarrollo del presente trabajo.

1.3. Estructura del documento

El presente documento se estructura en 7 capítulos distintos. Sin embargo, podemos encontrar tres partes diferenciadas. La primera parte, de naturaleza introductoria y de contextualización, comprende los dos primeros capítulos. A continuación, la segunda parte, donde encontraremos los capítulos 3, 4 y 5, donde comenzaremos definiendo el dispositivo Holter, recorreremos las características de un Holter común y acabaremos analizando cada detalle de nuestro dispositivo. Y por último, la parte final donde encontramos los capítulos 6 y 7, donde se realizará un balance final del proyecto y se propondrán distintas líneas de trabajo que cumplimenten o completen al presente.

En este primer capítulo, hemos realizado la presentación del proyecto, dejando claros la motivación y los objetivos que se persiguen. El segundo capítulo nos adentrará en la naturaleza biológica del biopotencial que se pretende registrar, así como en el estudio de los fundamentos gracias a los que funciona el equipo de electrocardiografía.

A partir del tercer punto nos centraremos en el dispositivo Holter como tal. Así, en este mismo capítulo se realizará una definición precisa del dispositivo que nos ocupa, contextualizándolo en el tiempo y analizando las características de un dispositivo de electrocardiografía ambulatoria convencional. Así, en el capítulo 4 se llevará a cabo un primer análisis de requisitos y se concretarán las líneas a seguir durante el desarrollo. Finalmente, en el capítulo 5 se abordan las características reales del proyecto final en todos sus aspectos, tanto a nivel de implementación hardware como software.

Finalmente, en el capítulo 6 se expondrán las conclusiones obtenidas por para terminar, con el capítulo 7, proponiendo líneas futuras de trabajo.

1.4. Elementos necesarios para la implementación del proyecto

Como se marca en los objetivos, trataremos de realizar la implementación completa tan sólo con elementos disponibles en el Departamento de Electrónica de la Universidad de Málaga, y usando herramientas software disponibles en el Servicio Central de Informática.

Así, necesitaremos una placa Arduino UNO y un módulo compatible eHealth, como la placa eHealth v2.0, los cuales conformarán el bloque funcional que registrará y coordinará todas las funcionalidades del sistema, y se encargará del acondicionamiento y procesamiento de la señal. Junto con la placa eHealth, se usarán los electrodos del Kit que la complementa y que ya se usaron en las prácticas de asignaturas como Instrumentación Biomédica. Además, serán necesarios para la implementación de la interfaz de usuario manual dos pulsadores, dos resistencias de $10k\Omega$ y dos condensadores de $22nF$, un display digital LCD 16x2 junto con un potenciómetro. Usaremos también las herramientas disponibles en el departamento para fabricar una placa de circuito impreso (PCB) para conseguir una implementación limpia y ordenada.

Para complementar el dispositivo con la interfaz bluetooth, usaremos también un módulo bluetooth HC-06 y un smartphone, preferentemente con el Sistema Operativo Android. Y por su parte, para dotar al dispositivo de memoria de almacenamiento, usaremos un módulo de memoria SD y una tarjeta de memoria microSD con adaptador al tamaño SD convencional (y compatible con el módulo). En cuanto a las herramientas software se refiere, se utilizarán las siguientes herramientas:

- el Matlab, para el diseño de la herramienta de graficación y diagnóstico;
- el IDE de Arduino, para el desarrollo y programación del sistema;
- el Software Eagle, para el diseño de la PCB
- El Software Fritzing, para el diseño del esquemático

En el caso de la aplicación móvil, se comenzará el desarrollo con Android Studio. Sin embargo, debido a que el fin real de dicha aplicación no es otro que el demostrar que nuestro dispositivo consta de una conectividad bluetooth fiable (y sin pretender ser uno de los principales elementos de nuestro proyecto), se decide finalmente utilizar una herramienta web llamada AppInventor, la cual

permite el desarrollo en un entorno online, y un progreso mucho más rápido. Dado el carácter sencillo de la aplicación que se pretende diseñar, esta herramienta es suficiente para suplir nuestras necesidades.

2. Fundamentos de la Electrocardiografía

Este capítulo está dedicado íntegramente al biopotencial cuya señal pretendemos captar usando el electrocardiógrafo. Así, quedarán reflejadas ciertas nociones biológicas básicas entorno al impulso nervioso, así como las características más importantes del corazón. Finalizará el capítulo realizando un análisis sobre las principales características de la señal del ECG desde el punto de vista técnico.

2.1. Generalidades sobre el corazón y el músculo cardíaco

El corazón se encuentra en el mediastino (“espacio medio” haciendo referencia al lugar situado entre los pulmones), concretamente en el mediastino anterior. Está recubierto por el epicardio a modo de lámina visceral y más internamente por el pericardio, a modo de lámina parietal. Se puede decir que es una estructura de forma conoidal, cuyo vértice apunta en dirección delante-izquierda-abajo.



Figura 2.1. Vista anterior del interior del órgano cardíaco. Se observan claramente las 4 cámaras, así como los tabiques musculares que las separan.
Fuente: *Practice Anatomy Lab 3.0*

Consta de cuatro cavidades, las aurículas izquierda y derecha y los ventrículos izquierdo y derecho (véase *Figura 2.1*). Las aurículas muestran paredes finas, lo que está directamente relacionado con una sístole mucho más débil que la ventricular. Entre sí, las aurículas se separan por el septo interauricular, y ambas están separadas de los ventrículos por las válvulas tricúspide (separación auriculoventricular derecha) y mitral (separación auriculoventricular izquierda). Continuando el septo interauricular y separando ambos ventrículos, encontramos el septo interventricular. Los ventrículos, estructuralmente, son similares a las aurículas, aunque tienen la particularidad de tener un músculo mucho mayor y más desarrollado. Sin embargo, es el músculo del ventrículo izquierdo el mayor y más poderoso de los dos, ya que es

el responsable de la eyección a alta presión de sangre hacia la circulación sistémica, mientras que la circulación pulmonar no requiere tanta presión [1].

Este órgano se divide en dos partes principales, tanto estructurales como funcionales. Así, se define al corazón derecho como la mitad del corazón conformada por la aurícula y ventrículo derechos. La aurícula izquierda recibe la sangre desoxigenada que desemboca desde la vena cava superior e inferior, y que, a su vez, procede de la circulación sistémica. Tras la diástole auricular, la sangre atraviesa la válvula tricúspide y llega al ventrículo derecho, cuya sístole bombeará la sangre hacia la circulación pulmonar a través de las arterias pulmonares. Por su parte, el corazón izquierdo es el conformado por la aurícula y ventrículo izquierdos. La aurícula recibe la sangre recién oxigenada desde la circulación pulmonar que desemboca a través de las venas pulmonares. La sístole auricular empuja la sangre a través de la válvula mitral hacia el ventrículo izquierdo, y éste, al contraerse, la bombeará de nuevo, a través de la aorta, hacia la red sistémica.

Por su parte, el músculo cardíaco tiene la peculiaridad de ser de contracción involuntaria a pesar de pertenecer a la clasificación de músculo estriado (al igual que el esquelético), es decir, su contracción se sustenta sobre el mecanismo de los sarcómeros de actina y miosina. Sin embargo, a diferencia del tejido muscular esquelético, el músculo cardíaco carece de epimisio y perimisio, por lo que no existe la aglomeración en fascículos. Sin embargo, el

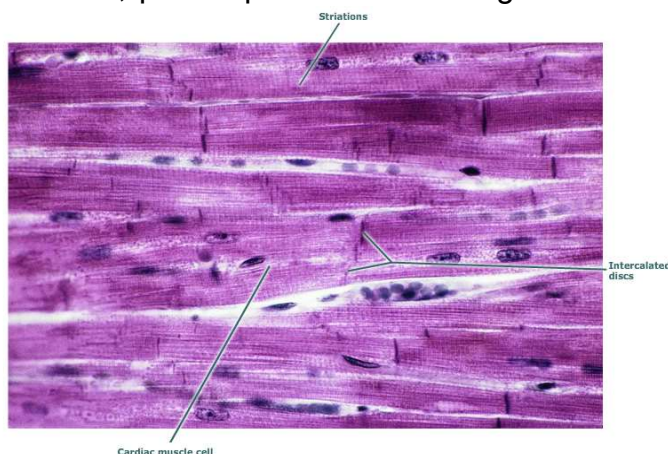


Figura 2.2. Vista a microscopía óptica del tejido cardíaco. Las estructuras observables son las fibras bien diferenciadas entre sí, en cuyo interior encontramos núcleo y estrías. Las estructuras que delimitan a la célula transversalmente son los discos intercalares. Fuente: *Practice Anatomy Lab 3.0*

tejido conectivo sí está presente, en forma de endomisio, y su función es recubrir y dar cohesión a cada una de las fibras musculares.

Otra peculiaridad a nivel celular es la existencia de los discos intercalares (véase *Figura 2.2*), los cuales son estructuras que únicamente están presentes en el

corazón, y cuya función es la de permeabilizar la unión entre las células, pero asegurando a la vez que esa permeabilización no debilita las uniones, si no que las fortalece y afianza. Esto se debe a los tres componentes de estos discos intercalares: las uniones GAP, las uniones estrechas y los desmosomas. Sí es cierto que todos estos elementos están presentes en otros tipos celulares, e incluso también en células vegetales, pero la particularidad reside en que todos ellos juntos solamente los encontramos en el músculo cardíaco, y además, conformando la estructura llamada disco intercalar [2].

2.2. Vectores de polarización

El corazón consta de un sistema específico de conducción y coordinación, el cual es posible gracias a las células musculares específicas que se distribuyen acorde a la anatomía de este órgano.

Así, cuando hablamos del tejido cardíaco específico, nos referimos a los nodos sinusal y auriculoventricular, el fascículo de His y sus ramificaciones, que son las ramas de Tawara, que a su vez se ramifican dando lugar a las células o fibras de Purkinje. “Independientemente de ello, el resto de las células musculares cardíacas son capaces de generar un estímulo y conducirlo” [3]. Además, estas células son tan específicas que han perdido la capacidad de contraerse, siendo su función primera y última la de generar y transmitir el impulso que provocará la actividad cardíaca.

Así, el nodo sinoauricular es el encargado de la actividad marcapasos. Está formado por un grupo de células del miocardio especializadas, situadas en la pared de la aurícula derecha, cerca de la apertura de la vena cava superior. El nodo auriculoventricular se sitúa en la parte inferior del septo interauricular, cercana al septo interventricular. Sus células también son capaces de llevar a cabo la actividad marcapasos, pero en mucha menor medida que como ocurre en el nodo sinoauricular, dominando este último conjunto de células dicha función. El nodo auriculo ventricular es crucial para la transmisión del impulso generado en la aurícula derecha hacia ambas aurículas.

Ya se ha aclarado de dónde surge el impulso eléctrico inicial, y ahora veremos su transmisión. De esta función se encargan el fascículo de His y sus ramificaciones, las ramas de Tawara, ambas de naturaleza muscular y conformadas por células especializadas, conocidas como fibras de Purkinje. Su tarea esencial es transmitir el impulso que se ha producido en la zona auricular hacia la zona ventricular y, con ello, a todo el músculo cardíaco.

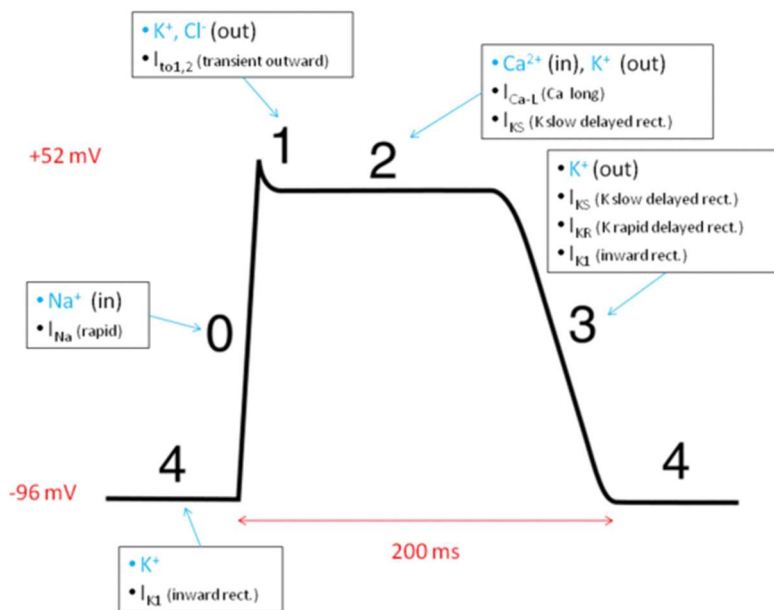


Figura 2.3. Potencial de Acción En las células cardíacas. Observamos el comportamiento del flujo de los distintos iones. Fuente: Quasar 18:07, 7 August 2009

Lo que nosotros pretendemos registrar mediante nuestro dispositivo, son esas variaciones eléctricas detectables desde la superficie corporal. Para ello, primero debemos comprender cómo es la actividad eléctrica a nivel celular. Existen cuatro fases diferentes (véase la *Figura 2.3*):

- **Fase 4:** Potencial de membrana en reposo. Las concentraciones iónicas se recuperan tras las alteraciones sufridas durante el potencial de acción.
- **Fase 0:** Pendiente ascendente del potencial de acción. Se trata de la mayor despolarización que se da en la célula. Se abren los canales rápidos de Na^+ , lo que da lugar a la dispersión de la despolarización por todo el sistema conductor del corazón.
- **Fase 1:** Repolarización rápida hasta la meseta. Se cierran los canales de Na^+ y se abren los canales de K^+ , lo que produce un estado de repolarización transitoria.

- *Fase 2:* Se llega a la meseta. La membrana se mantiene despolarizada debido a la apertura de los canales lentos de Ca^+ , y a su vez, a la salida de iones K^+ a través de los canales dependientes del voltaje.
- *Fase 3:* Repolarización completa hasta reposo (fase 4). La inactivación gradual de los canales de Ca^+ , supone la activación de los canales de K^+ , provocando la repolarización rápida.

Siguiendo ese patrón a nivel celular, es necesario recordar que la vía de conducción normal en el corazón comienza con la actividad marcapasos en el nodo sinoauricular, continua en el nodo auriculoventricular, sigue por el Haz de His, atravesando las ramas de Tawara hasta las últimas fibras de Purkinje, y finalmente, el músculo ventricular. Los intercambios iónicos que ocurren durante las cuatro fases del potencial de acción tienen lugar siguiendo ese recorrido, y son los responsables últimos de las corrientes eléctricas de las que, finalmente, se vale el electrocardiógrafo para obtener y representar la señal del ECG.

2.3. Electrocardiograma

Hay diversas formas de observar la actividad cardíaca. Entre otras, cabe destacar la medida de tensión arterial, la fonendoscopia (clásica y electrónica), monitorización del pulso mediante un pulsómetro (la mayoría de las veces acompañado por un oxímetro), e incluso utilizando el sistema de ecografía es común realizar ecocardiografías. Sin embargo, una de las pruebas diagnósticas más completas, y la prueba cardiológica por excelencia es sin duda el electrocardiograma.

Se puede definir el electrocardiograma (ECG) como “un registro de los cambios del potencial de la superficie del cuerpo producidos por la despolarización y repolarización del corazón” [4].

Un período normal del ECG está compuesto por la sucesión de ondas que componen los tres complejos principales: P-QRS-T. Así, en la definición de ECG referida anteriormente, se usan los términos despolarización y repolarización del corazón. Ambos conceptos se refieren, en su conjunto, al sumatorio de todo el proceso que desencadena el potencial de acción de cada una de las partes

implicadas (nodos sinoauricular y auriculoventricular, haz de His, ramas de Tawara y fibras de Purkinje), proceso que, aun siendo similar en cada una de las zonas, tiene lugar de forma correlativa en el tiempo, y no simultánea, lo cual queda reflejado en la forma del ECG.

Por tanto, y en consonancia a lo afirmado anteriormente, cada una de las ondas que dan pie al electrocardiograma completo se deben a la superposición de los distintos procesos de despolarización y repolarización individuales que se van dando en las distintas zonas del corazón (véase *Figura 2.4*).

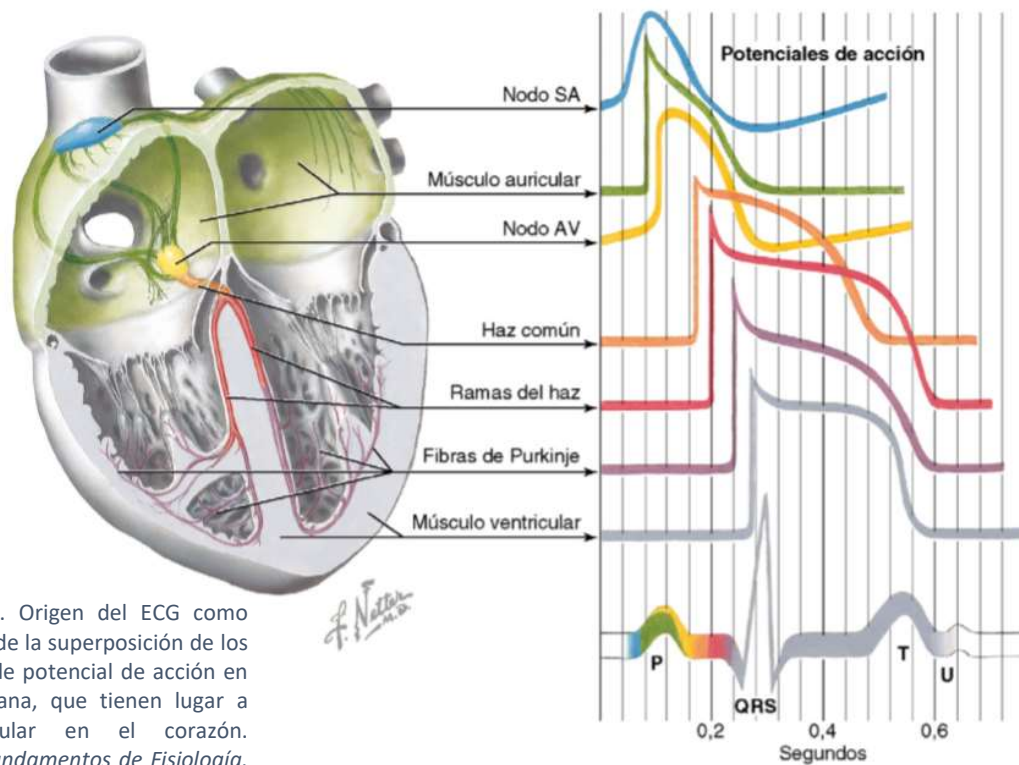


Figura 2.4. Origen del ECG como resultado de la superposición de los procesos de potencial de acción en la membrana, que tienen lugar a nivel celular en el corazón.
Fuente: *Fundamentos de Fisiología*.
Netter

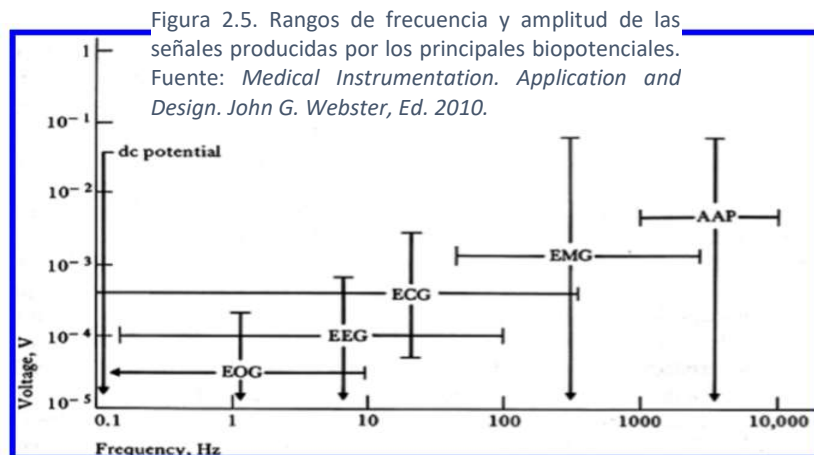
- **Onda P.** Causada por la despolarización auricular.
- **Complejo QRS.** Originado con la despolarización ventricular.
- **Onda T.** Se corresponde a la repolarización ventricular.

Para conseguir captar esta señal, el electrocardiógrafo convencional consta de un total de 10 electrodos, 4 conocidos como bipolares, los cuales se encargan de las derivaciones periféricas, y 6 precordiales, con los cuales es posible registrar las derivaciones del mismo nombre. Dicho esto, sólo queda añadir que en este proyecto trabajaremos sobre todo con la segunda derivación

periférica, la cual es la más conocida y la que se usa esencialmente para detectar la frecuencia cardíaca.

2.4. Particularidades de la señal de ECG

La señal registrable mediante el electrocardiógrafo es un biopotencial, es decir, el resultado de la actividad eléctrica de las células excitables, en este caso, del corazón. El medio interno de estas células cuando están en reposo se encuentra entre 40 y 90 mV, respecto al medio externo. Cuando se desata el potencial de acción, la despolarización produce un aumento del voltaje hasta los 120 mV. Tras esto, tiene lugar el período refractario, lo que limita la frecuencia de estimulación máxima de la célula a, aproximadamente, los 1000 impulsos /segundo. Estos datos se refieren a una vista general y aproximada, ya que cada biopotencial encierra sus propias características (véase *Figura 2.5*).



La medida y cuantificación de todos los biopotenciales tienen en común dos aspectos fundamentales: un reducido ancho de banda, que no supera los 10KHz, y una amplitud limitada entorno a los 90 mV. Sin embargo, estas medidas pueden variar de un individuo a otro, incluso si ambos son individuos sanos. De una forma más concreta, y centrándonos en el biopotencial que nos ocupa, encontramos que la frecuencia del ECG se sitúa en el rango comprendido entre los 0.1 y los 500 Hz, siendo el biopotencial con el rango de frecuencias más amplio, y cuya frecuencia mínima posible es menor. Por otro lado, la amplitud de la señal variará entre los 60 μ V y los 5mV.

Por último, es necesario destacar la importancia del carácter vectorial de los potenciales eléctricos del corazón. Gracias a ello, así como a la situación de los distintos electrodos, es posible detectar la situación del origen y dirección de propagación del impulso. Este es el fundamento en el que se basa el instrumento médico para la recomposición de la señal y la representación final del ECG [5].

3. ECG ambulatoria

Este capítulo nos abre la puerta que nos adentra de lleno en lo que realmente representa un dispositivo Holter. Se establecerá una primera definición formal partiendo de los inicios, de la necesidad que llevó a crearlo y de la evolución que ha sufrido con el paso del tiempo.

Tras este primer acercamiento histórico, abordaremos los aspectos fundamentales de los dispositivos de electrocardiografía ambulatoria que se usan en la actualidad en cualquier consulta cardiológica. Dichas características nos servirán como referencia a la hora de tomar decisiones de cara al posterior desarrollo de nuestro propio dispositivo.

3.1. Origen y evolución histórica

La primera vez que se consiguió monitorizar la actividad eléctrica del corazón de manera no invasiva fue gracias a Augustus Waller y su galvanómetro capilar de mercurio, en mayo de 1887. Sin embargo, la señal recabada era muy pobre, y tan solo se consiguieron reflejar dos deflexiones distorsionadas.

Posteriormente, Willem Einthoven comenzó a estudiar el comportamiento del ECG con una herramienta similar. Aunque esta vez, Einthoven fue capaz de mejorar estas distorsiones mediante cálculos matemáticos, llegando a obtener una buena representación clara de la señal de ECG.

Tras esta primera mejora mediante métodos matemáticos, trató de mejorar el electrocardiógrafo sustituyendo su antiguo instrumento por un galvanómetro de cuerda, el cual resultó ser mucho más preciso. Así, en 1901 publica su primer artículo sobre el nuevo galvanómetro, en 1902 aplica por primera vez el ECG en una consulta médica y en 1903 publica un artículo detallado sobre el galvanómetro, su aplicación, y una recopilación de señales capturadas con el mismo.

El científico siguió trabajando y perfeccionando tanto el método como el sistema de recopilación, desarrollando también el conocido triángulo de Einthoven, usado aún en la actualidad. De este modo, fue en 1924 cuando se le

otorgó el Premio Nobel de Medicina a Willem Einthoven , considerado el padre de la electrocardiografía actual [6].

Sin embargo, el desarrollo de los primeros equipos de electrografía ambulatoria está estrechamente ligado a la carrera profesional del físico e inventor Norman J. Holter. Holter fue un hombre versátil, estudioso de diversas disciplinas entorno a la física y con numerosos inventos y aportaciones dentro de diversos campos de la ciencia. Además de ello, merece una mención especial dentro del campo de la biofísica y medicina.

Pero no fue hasta la década de 1940 cuando empezó a interesarse por la actividad eléctrica en el cuerpo del ser humano, comenzando a trabajar con el doctor Joseph Gengerelli, profesor de Psicología en la Universidad de California, conocido por sus importantes aportaciones a la neurociencia.

Su proyecto conjunto iba dirigido a simular el comportamiento nervioso cuando se inducían alteraciones en el campo eléctrico del nervio. Todo ello llevó a Holter a la idea de monitorizar la actividad eléctrica del corazón. Si bien es cierto que el ECG existe desde principios de la década de 1920, Holter pensó en complementar esta prueba de diagnóstico incluyendo un sistema de monitorización capaz de adaptarse a la vida del paciente, pudiendo así ampliar el tiempo de registro.



Figura 3.2. Primer Dispositivo Holter de 38 kg de peso. Fuente: *Cardiology Journal*.

No obstante, el primer dispositivo desarrollado necesitaba dos grandes baterías que suministraran energía, lo cual incrementaba su peso, alcanzando los 38 kg (ver *Figura 3.1*). El hecho de que un dispositivo portátil de 38 kg de peso debiera ser colocado en personas cuya actividad cardíaca presentase anomalías suponía, obviamente, un grave problema. Aun así, este aspecto no se corrigió hasta 1961, cuando Norman J. Holter junto a su compañero William Glasscock volvieron a reevaluar el sistema.

Sí es cierto que el dispositivo original fue llevado a la práctica en los años 50, y además muy exitosamente, en un estudio realizado a 200 pacientes y dirigido por el propio Norman, diagnosticando correctamente diversos casos de angina de pecho en personas que se creían sanas. Esto tuvo un gran impacto entre médicos e investigadores, por lo que fue aumentando su popularidad incluso antes de salir a la venta, y siendo aceptado por la comunidad médica de forma definitiva en 1955.

Finalmente, el primer dispositivo se comercializa en 1962, desde los laboratorios *Del Mar Engineering Laboratories*, con una capacidad de grabación superior a las 24 horas. Los holters que vendrían más tarde mejorarían al primero en cuanto a optimización de velocidad de muestreo, minimizando su tamaño y peso, e incrementando su comodidad.

Norman J. Holter, pasaría a considerarse el padre de la electrocardiografía ambulatoria, aunque debemos añadir que renunció a los derechos de su gran invento donándolo a la medicina. Hoy en día su dispositivo es usado por todo el mundo para hacer un estudio cardiológico exhaustivo en pacientes cuya actividad cardíaca tiene un comportamiento anómalo [7].

3.2. Características del Holter convencional

En la actualidad, los dispositivos de ECG ambulatoria han evolucionado en una amplia variedad de aspectos y, además, en todos esos aspectos lo hacen en gran medida. Para realizar un análisis verídico acerca de lo que actualmente se está utilizando, hemos tomado como referencia principal las especificaciones técnicas del modelo de Holter *BI9800 TL*, de la compañía *Biomedical Instrument Co., Ltd*. Se ha elegido este sistema como modelo debido a que es uno de los más usados actualmente según el personal de electromedicina del Hospital Infanta Margarita, de Cabra, el cual está gestionado íntegramente por el Servicio Andaluz de Salud (SAS). Además, este sistema es un fiel representante de los dispositivos con características estándar, y por las que se rige el mercado de estos dispositivos en la actualidad.

Centrándonos ya en las características externas generales de un dispositivo Holter estándar de la actualidad, apreciaremos los siguientes elementos:

- Pantalla de interfaz de usuario, normalmente un display LCD que facilita la interacción entre el usuario y el dispositivo.
- Teclado de botonera simple, con tres funciones principales para navegar a través de las posibles opciones de programación (avanzar, retroceder y seleccionar).
- Enchufe multifuncional de 26 pines en su parte superior, los cuales sirven como entrada mientras el dispositivo está recogiendo y grabando datos en la memoria. También es el puerto que da conexión al cable USB cuando dicho registro ha finalizado y se pretenden recuperar los datos desde un ordenador.
- Oquedad de fijación del cordón para colgar en el cuello del paciente, así como posibilidad de incorporar una pinza de enganche y sujeción a un cinturón.
- Cobertura del compartimento donde se incluirá la fuente de alimentación, generalmente una pila alcalina o equivalente de 1,5V.

Los datos recabados desde el Holter serán almacenados en una tarjeta de memoria SD. El dispositivo es compatible con varios modelos y marcas, sin embargo, el fabricante asegura que pueden existir errores de compatibilidad en determinados casos. Además, el rango de capacidad de la tarjeta variará entre los 128 MB y 1 GB (dependiendo de la duración de la grabación, así como de la frecuencia de muestreo).

Por otro lado, esta información grabada en la memoria podrá ser analizada con un software específico y compatible con el modelo en cuestión y, por tanto, con el formato de los datos almacenados. Normalmente este software es distribuido por la misma empresa encargada de la venta del dispositivo.

La alimentación de estos sistemas comúnmente se sustenta en la inclusión de una pila seca de tipo *triple A* (AAA), de 1.5V, aunque se recomiendan más concretamente las alcalinas de alta duración. El dispositivo podrá

permanecer encendido ininterrumpidamente por más de 48 horas (suficiente para realizar la prueba).

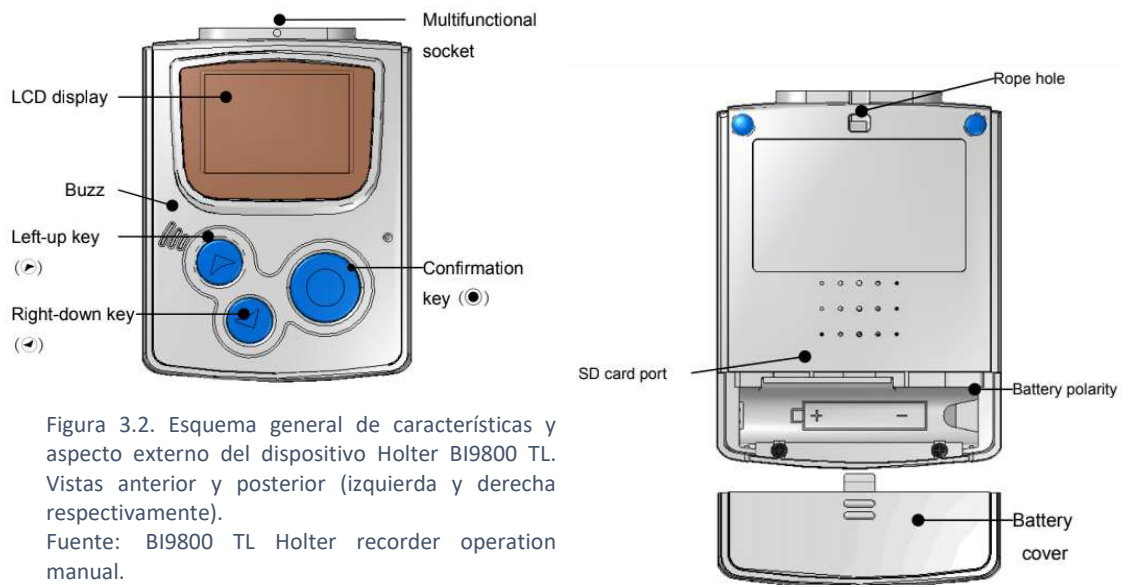


Figura 3.2. Esquema general de características y aspecto externo del dispositivo Holter BI9800 TL. Vistas anterior y posterior (izquierda y derecha respectivamente). Fuente: BI9800 TL Holter recorder operation manual.

En cuanto a características físicas se refiere (véase *Figura 3.2*), las dimensiones del dispositivo usado por el SAS son de 68x53x16 mm, mientras que su peso es de 42 gramos. El resto de modelos suelen estar entorno a estas medidas, salvo por los dispositivos especialmente pequeños, cuyas dimensiones decrecientan hasta aproximadamente 2/3 o la mitad del tamaño común.

Su armazón externo es de plástico moldeado que permite que el dispositivo sea resistente al agua. Además, el sistema Holter tiene dos modos de colocación durante el registro, según los dos sistemas descritos en las características externas: pinza de sujeción al cinturón o colgado mediante un cordón al cuello del paciente.

En cuanto a las características funcionales de estos sistemas se refiere, suele constar de 3 canales en su versión más básica, pudiendo llegar incluso hasta 8. Esto significa que añade una mayor cantidad de datos recabados, lo cual se puede traducir de dos maneras distintas, bien con un mayor número de derivaciones registradas, o bien, una mayor calidad en cuanto a la resolución y precisión de la señal se refiere.

Además, los dispositivos más usados almacenan la información recabada a través de estos canales; y lo hace sin formato de compresión. Los fabricantes se refieren a este aspecto usando el término “*full disclosure*” (en español se puede interpretar como revelado completo, sin alterar, modificar, comprimir o recortar, tal como el dispositivo interpreta la señal).

Para concluir el análisis del estado actual de la técnica, no está demás añadir que existe otra variante de Holter, el implantable (cuyo nombre completo es dispositivo implantable subcutáneo de electrocardiografía ambulatoria), el cual sirve para monitorizar eventos cuya frecuencia de aparición es tan poco común que puede escapar del estudio realizado en una prueba de 24 o 48 horas. Este dispositivo no está diseñado para estar grabando de forma continua, sino que se activa manualmente por el paciente o un familiar, o cuando se detectan anomalías en el comportamiento cardíaco. Está indicado en patologías difíciles de diagnosticar con las pruebas de diagnóstico comunes, como por ejemplo en síncope causados por arritmias [8].

4. Especificación y metodología del proyecto

Este capítulo comprende la descripción general del dispositivo ideal que se pretende alcanzar al finalizar el desarrollo del proyecto. Se comenzará con la relación de requerimientos generales y la metodología que se ha seguido para llevar a cabo el desarrollo de una manera ordenada. Finalmente, en base a lo redactado anteriormente, este capítulo concluirá con una propuesta concreta de lo que se va a realizar.

Este apartado se puede comprender como una introducción a la descripción de nuestro propio dispositivo Holter. Los requisitos que se desarrollarán y que serán determinantes para realizar la propuesta son requerimientos reales que se exigirían a cualquier Holter real.

4.1. Requerimientos propios del dispositivo

A continuación, se detallarán de manera clara y definida qué requisitos se exigen a nuestro dispositivo de manera concreta.

- Debe ser capaz de captar la señal de ECG, acondicionarla y procesarla con el fin de ser almacenada e interpretada posteriormente.
- El elemento central de control de todo el dispositivo será el hardware de Arduino UNO.
- El circuito de acondicionamiento de señal será el que ofrece la placa de sensores eHealth.
- El proyecto se debe culminar limitando el conjunto de elementos usados a las posibilidades y facilidades ofrecidas por el departamento de Electrónica de la Universidad de Málaga.
- Debe estar dotado de memoria externa. Dicha memoria será capaz de almacenar al menos un fichero de 48 horas de duración.
- Debe constar de interfaz física de usuario, incorporada en el cuerpo del propio dispositivo. Para facilitar el uso y manejo de la misma, se usarán al menos dos pulsadores.
- El dispositivo final debe ser portátil, lo cual implica un sistema de alimentación interna. Este requisito está implícito en su naturaleza de Dispositivo Holter.

- Se dotará de funcionalidad bluetooth con el fin de establecer conexión con dispositivos móviles como tablets o smartphones. El principal fin será la representación gráfica, pero no se exige como herramienta de diagnóstico.
- Se facilitará un sistema de graficación que permita una correcta interpretación de los datos registrados, así como la emisión de un posterior diagnóstico preciso.

4.2. Metodología

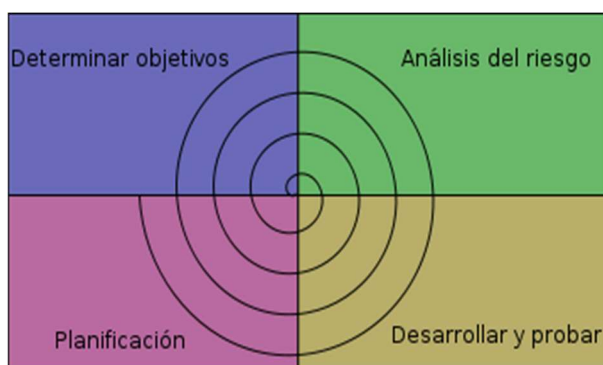


Figura 4.1. Esquema aclaratorio acerca del funcionamiento de la Metodología en Espiral. Fuente: Administración de Proyectos, Modelos de desarrollo (Google Site).

El método empleado para la organización del desarrollo de este proyecto sigue la filosofía de la espiral. La Metodología en Espiral consiste en minimizar los riesgos segmentando el proyecto en tareas más pequeñas y sencillas, fáciles de analizar, lo cual agiliza que el desarrollo evolucione en función de las diversas situaciones (adversas o no) que puedan surgir durante el avance del proyecto.

Así, cada vuelta que se le da a la espiral se corresponde con el desarrollo de una nueva funcionalidad del dispositivo global, teniendo todas las vueltas un mismo esquema común basado en las siguientes 4 tareas:

1. *Tarea Inicial.* En primer lugar, se determinan los objetivos a alcanzar en la presente etapa;
2. *Identificación de Riesgos.* Se deben evaluar las distintas posibilidades, así como identificar y resolver los riesgos;
3. *Fase de Desarrollo.* Se han de desarrollar y verificar los resultados de la iteración;
4. *Verificación y Prueba.* Por último, comprobar que estos resultados cumplen los objetivos fijados y que efectivamente son funcionales en el

proyecto final. Una vez realizado este paso, se comienza a plantear la siguiente fase.

Se ha elegido esta metodología debido a que el proyecto consta de varias funcionalidades bien diferenciadas tanto a nivel hardware como software, las cuales se van complementando unas a otras hasta llegar al proyecto final. En las distintas fases del proyecto se irán completando las vueltas de nuestra espiral, lo cual irá engrosando poco a poco este Holter en todos sus niveles, afectando el avance en cada fase, por tanto, a la funcionalidad global. Esta sistemática de trabajo permite añadir una función sin alterar necesariamente el resto, así como comprobar la funcionalidad total obtenida en la última fase, lo cual garantiza un avance del proceso sobre pasos más que afianzados.

4.3. Propuesta de prototipo y concreción del proyecto

A modo de solución a los requisitos planteados, y habiendo identificado cada uno de los posibles bloques funcionales, presentamos el siguiente diagrama como una primera vista general a lo que debe ser nuestro diseño (ver *Figura 4.2*). A continuación, se justificará detalladamente la presencia de cada uno de esos bloques.

El bloque principal entorno al cual se estructura todo el sistema es el compuesto por Arduino y el módulo eHealth. De este modo, Arduino aporta su procesador, así como los pines donde se conectarán el resto de elementos. Es en este conjunto donde tiene lugar la coordinación de absolutamente todas las funciones que el sistema llevará a cabo por sí mismo. Es también fundamental la presencia de eHealth, ya que, gracias a esta placa, es posible el acondicionamiento de la señal, así como su posterior lectura; funciones sin las que sería imposible definir nuestro sistema como dispositivo Holter.

Directamente conectado al circuito de acondicionamiento del bloque de procesado y control, encontramos el bloque funcional de captura de señal, el cual lo componen los electrodos a modo de interfaz entre el cuerpo del paciente y el dispositivo electromédico.



Otro conjunto esencial es el de interfaz manual de usuario, gracias al cual se pueden elegir y programar las funcionalidades del sistema. Más adelante veremos que, sólo parcialmente, este aspecto se puede sustituir por una interfaz de usuario a través de una aplicación móvil. Sin embargo, aún así es necesario el uso de la interfaz física para ordenar una posible conexión bluetooth. Gracias a este aspecto, el usuario tiene pleno control y consciencia sobre lo que está ocurriendo “en el interior” del sistema de manera oculta.

Un dispositivo de electrocardiografía ambulatoria comienza a tener sentido cuando se entiende como un sistema de grabación de datos. En caso de no ser así, los datos se retransmitirían a tiempo real, por lo que estaríamos hablando de algo parecido a un electrocardiograma convencional. Así, es necesaria la dotación de un módulo de memoria externa donde se almacenen todos los datos capturados, con el posterior fin de ser representados con la posibilidad de interpretación, y, en base a ello, la emisión de un diagnóstico.

Finalmente, hay que añadir que desde el primer momento en que se decide diseñar un sistema Holter, se decide incorporar conectividad bluetooth por diversos motivos. Desde el punto de vista de utilidad, supone una ventaja el hecho de poder comprobar desde un smartphone si el dispositivo funciona correctamente antes de que el paciente se lo lleve conectado durante 24 ó 48 horas. Por otro lado, el hecho de manejar una interfaz de usuario con pocos botones puede llegar a ser tedioso dentro de la sencillez de la programación, por lo que sería buena idea el poder codificar y programar el funcionamiento del sistema desde un dispositivo móvil. Por otro lado, desde el punto de vista académico, es interesante el ofrecer al alumno un primer acercamiento real a esta tecnología, ya que está hoy en día presente en la práctica totalidad de los dispositivos portátiles (no necesariamente electromédicos, de hecho, en los dispositivos Holter no es nada común encontrar algo así).

bloques funcionales. Hay que añadir que, en el caso de las piezas del bloque principal de procesamiento y control, el análisis de Arduino UNO y la placa eHealth se hará por separado, debido a la importancia de ambos.

5.1.1. Arduino UNO

La plataforma de Arduino consta de diversas placas Hardware distintas, cada una de ellas con sus propias particularidades y ventajas. Sin embargo, desde el desarrollo de este proyecto se ha decidido usar únicamente la placa Arduino UNO por presentarse como un elemento imprescindible en el Departamento de Electrónica, y por supuesto, debido a sus características técnicas (ver *Tabla 5.1*), así como por la compatibilidad que presenta respecto a la placa de sensores e-Health. Así, es necesario aclarar que la placa eHealth se encontrará superpuesta a Arduino. Esto significa que cuando digamos que un elemento se conecta a algún pin de Arduino, en realidad se conectará físicamente a eHealth, ya que esos pines de eHealth en realidad continúan hasta el nodo del pin equivalente en Arduino.

Especificaciones técnicas de la placa Arduino UNO	
Característica	Valor
Microcontrolador	<i>AT mega 328</i>
Voltaje Operativo	<i>5V</i>
Voltaje de entrada (recomendado)	<i>7-12 V</i>
Pines Entrada/Salida Digitales	<i>14 (6 de ellas son PWM)</i>
Pines de Entrada Analógicos	<i>6</i>
Memoria Flash	<i>32KB (AT mega 328) 0,5KB usados por Bootloader</i>
SRAM	<i>2KB (AT mega 328)</i>
EEPROM	<i>1KB (AT mega 328)</i>
Velocidad del Reloj	<i>16 MHz</i>

Tabla 5.3. Especificaciones técnicas de la placa Arduino UNO.
Fuente: *Blog del profesorado de la Red Social Educativa de Canarias, Gobierno de Canarias. (Visto en enero de 2019).*

Si nos detenemos a observar las características de Arduino, podemos deducir varios aspectos de especial interés. Por ejemplo, si nos fijamos en el microcontrolador y su memoria flash, vemos que no es demasiado potente ni tiene demasiada capacidad. Este es uno de los retos que se nos debían presentar según se predijo en la introducción. De hecho, una vez acabado el proyecto, el código cubre un total del 74% de la memoria del procesador. Este dato puede inducir a confusión, ya que aún queda más de un cuarto de memoria disponible. Sin embargo, si el programa llega a ocupar un 75% de la memoria, Arduino comienza a presentar problemas de estabilidad, lo que se traduce en la interrupción continua de su bucle principal, e incluso, reiniciándose continuamente. Debido a ello, este es el primer problema de optimización al que debemos hacer frente.

En cuanto a la velocidad de funcionamiento se refiere, 16MHz en principio es muy aceptable teniendo en cuenta que la frecuencia aproximada de un ECG convencional no supera los 500Hz.

Por otro lado, contamos con 14 pines digitales disponibles, los cuales, como veremos más adelante se agotarán fácilmente debido a los elementos usados. Así, estaremos obligados a prescindir de elementos que podrían facilitar la implementación, así como el manejo de la interfaz de usuario, mejorar las comunicaciones con un módulo bluetooth más potente, o incluso añadir alguna funcionalidad auxiliar más.

Sobre Arduino, cabe destacar el papel fundamental que juegan sus catorce pines digitales, ya que son la clave del amplio abanico funcional que ofrece Arduino UNO, y que nosotros agotaremos hasta el final en este proyecto. Así, los pines 0 y 1, o también RX y TX (abreviaturas de *recepción* y *transmisión*) representan la entrada y la salida (respectivamente) directas del puerto serie del sistema, por lo que serán esenciales en cuanto a la conexión bluetooth se refiere, como veremos más adelante. Por otro lado, los siguientes 6 pines están ocupados por la funcionalidad del display digital (LCD 16x2), siendo los pines del 2 al 5 (ambos inclusive) los encargados de enviar los datos a representar, y los pines 6 y 7 los que se conectan a las entradas E y RS respectivamente (control de inhabilitado y selector de registro respectivamente, los cuales permanecen estáticos).

El pin digital número 8 se conecta al circuito del botón seleccionar, con el cual se selecciona la opción elegida en la interfaz de usuario (visualizada en el LCD). De igual modo, desde el pin número 10 se lee la información del botón *pasar*, con el que se van cambiando las opciones a elegir en esta misma interfaz. Los pines que nos quedan, es decir, 9, 11, 12 y 13 están reservados a la transmisión de datos entre el microprocesador y el módulo de la tarjeta SD, ya que todos ellos, salvo el 9, están optimizados para la comunicación SPI. Desde el pin 9 en Arduino se controla el pin CS (*Chip Select* o *Selector de Tarjeta*), el cual debe estar a nivel bajo para que esté seleccionada la SD en cuestión.

Otra cuestión fundamental que se soluciona gracias a la estructura que aporta Arduino es la comunicación e intercambio de información entendido de manera más general. Es cierto que se precisa de un módulo bluetooth para dotar al sistema de este tipo de conectividad. Sin embargo, es su Puerto Serie la gran autopista a través de la cual circula la absoluta totalidad de buses de datos del sistema. En él se apoya tanto la comunicación entrante y saliente por los puertos Rx y TX (donde se conectará el módulo bluetooth), como el intercambio de datos entre la tarjeta SD y el procesador (usando la comunicación SPI, también mediante puerto serie).

Por último, debemos añadir que la alimentación del sistema será interna, usando una batería de litio del fabricante *Varta*, de 2600 mAh y 5V recargable. El cable se incluye en la batería mediante conexión USB. Dicho cable se ha manipulado con el fin de obtener la conexión directa a Arduino UNO gracias a la clavija Jack. El hecho de incluir un sistema de alimentación interno garantiza la portabilidad del dispositivo, así como la seguridad eléctrica del paciente, por lo que el dispositivo será eléctricamente seguro.

5.1.2. Placa multisensorial eHealth

La placa e-Health [9] es un módulo multisensorial compatible en superposición tanto con el hardware de Arduino UNO como con el de RaspBerry Pi. Dicha placa permite a los usuarios de ambas plataformas realizar medidas biométricas aplicables a experimentos y ejercicios relacionados con el campo médico. Así, las posibilidades que nos ofrece e-Health en su versión 2.0 (la que

usaremos en la implementación de nuestro diseño) nos abre un abanico de 10 sensores distintos, dejando disponibles los 14 pines digitales de Arduino, lo cual nos supone una ventaja enorme.

Las mediciones que se pueden realizar con la ayuda de e-Health de manera directa son un total de 10 (véase la *Figura 5.2*): el pulso, saturación de oxígeno en sangre, temperatura corporal, conductancia de la piel (también llamada actividad electrodérmica y actividad galvánica de la piel), glucometría, presión arterial, posición del paciente mediante un acelerómetro, electromiografía (EMG) no invasiva, espirometría, y electrocardiografía (ECG) [10].

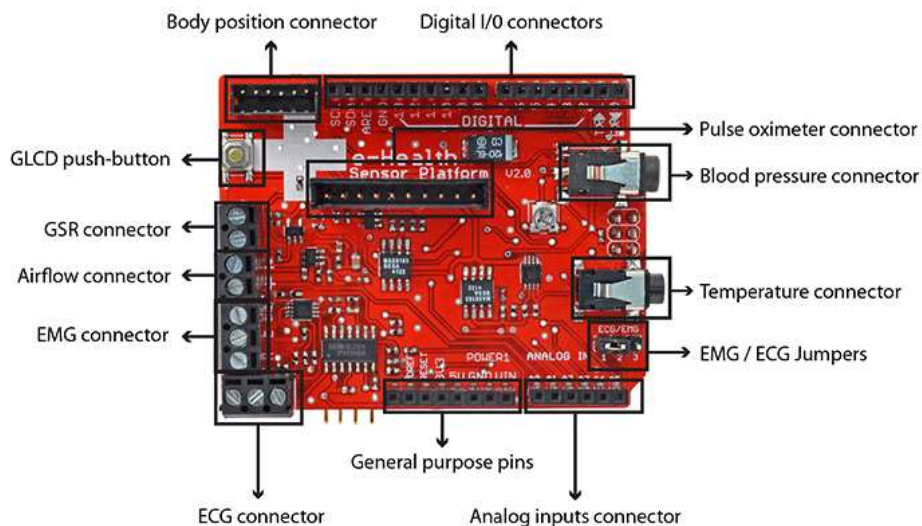


Figura 5.2. Fotografía superior de la placa e-Health, en la que están perfectamente identificados cada uno de los elementos de interés para cualquier usuario. Fuente: Web de TBEM Robótica y Electrónica. Subapartado dedicado a Arduino. <https://teslabem.com/arduino/e-health-sensor-platform-para-arduino-y-raspberry-pi/> (Visto última vez en mayo de 2019).

En la implementación de nuestro diseño, sólo nos interesa la funcionalidad del ECG. En el módulo e-Health, las opciones de ECG y de EMG comparten una parte del circuito de acondicionamiento debido a la similitud entre ambas bioseñales. Esto supone la necesidad de un *jumper* o puente, que sirve para seleccionar cuál es el circuito que debe ser usado; dicho de otro modo, selecciona la medida pretendida en cada ocasión. Por tanto, debemos asegurarnos de que se va a utilizar el hardware apropiado, por lo que es necesario comprobar que está correctamente posicionado, seleccionando el

circuito correspondiente al ECG. En la *Figura 5.2* se observa la posición correcta del *jumper* para realizar la prueba de ECG.

El elemento imprescindible de este módulo es el circuito de acondicionamiento de ECG, el cual hace posible la toma de muestras, su procesado y su posterior interpretación. Así, el primer paso para obtener la señal es colocar correctamente los electrodos (véase *Figura 5.3*).

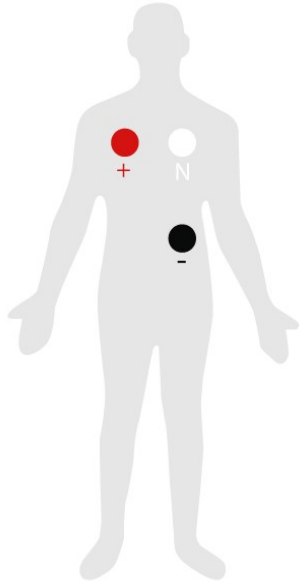


Figura 5.3. Correcta colocación de los tres electrodos de ECG de eHealth.

Fuente: <https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical> (visto por última vez en mayo de 2019).

El hecho de que haya tan sólo tres electrodos implica que la toma de muestras se base en la diferencia de potencial existente entre dos de los electrodos, mientras que un tercero se usa como referencia. Se trata de una medición diferencial, esto supone que, en sentido estricto, el ECG captado no se corresponde con ninguna de las derivaciones clásicas. Sin embargo, se puede tratar de manera similar a la derivación II periférica debido a las similitudes que presenta con la misma.

Como fruto de la amplificación y filtrado de la diferencia de potencial existente entre los electrodos ECG+ y ECG -, estando ambos medidos respecto al nodo de referencia.

A continuación de ser captada desde el cuerpo del paciente a través de los electrodos, la señal pasará al circuito de acondicionamiento (véase *Figura 5.4*). Así, en el nodo de salida del circuito encontramos una señal

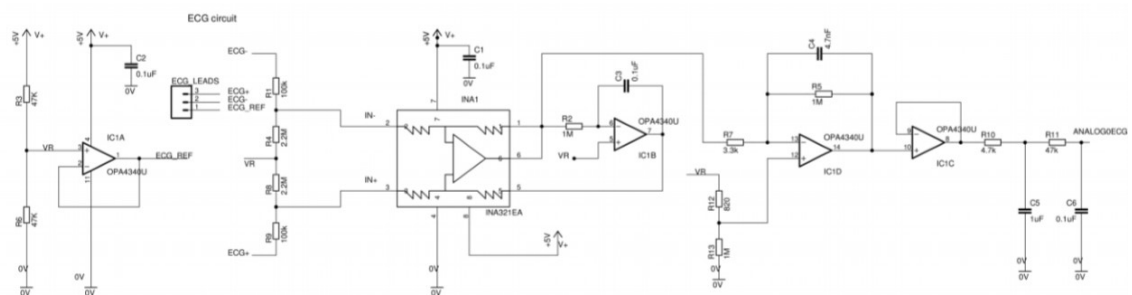


Figura 5.4. Circuito de Acondicionamiento de la placa eHealth para realizar el ECG.
Fuente: <https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical> (visto por última vez en mayo de 2019).

En el circuito de acondicionamiento se pueden distinguir 4 estructuras bien diferenciadas. La primera no es más que el seguidor de tensión del electrodo de referencia, mientras que la segunda es la configuración típica del circuito en Modo Común, donde se incorporan ya las entradas diferenciales.

A continuación, encontramos el amplificador de instrumentación. Se trata del modelo INA321EA, del fabricante Texas Instrument, el cual tiene una ganancia igual a 5. Encontramos, a la salida del mismo, un amplificador operacional en configuración integradora al cual se conecta el nodo de referencial, lo cual ofrece al paciente protección de posibles sobretensiones.

Por último, lo que sigue es un filtro pasa-bajas, el cual está implementado usando un amplificador en su configuración de inversor, y otro seguidor de tensión precedido de nuevo por dos filtros pasa-bajas consecutivos. El hecho de incorporar dos filtros pasa-bajas hace que la frecuencia de corte total sea aún menor, por lo que se obtendrán resultados buenos teniendo en cuenta la baja frecuencia a la que trabaja la señal producida por el impulso cardíaco.

5.1.3. Electrodo

Como se ha dicho, nuestro sistema cuenta con sólo tres electrodos, ya que la captación de la señal se basa en el método diferencial. Esto quiere decir que la señal que pretendemos obtener es la diferencia entre el electrodo ECG+ y ECG-, ambos medidos respecto al de referencia.

Estos electrodos se deben entender como transductores que, al mismo tiempo hacen las veces de doble interfaz electrodo-electrolito y electrodo-piel. Esto se justifica debido a la forma de transmisión de la electricidad: en el tejido vivo es mediante iones mientras que en el sistema eléctrico son electrones. En el funcionamiento de los electrodos resulta fundamental el proceso de redox: en el electrodo se debe oxidar un elemento metálico, desprendiéndose un electrón en el electrodo y un catión hacia el electrolito; mientras que, en el electrolito, se oxida un catión, quedando un átomo neutro en el electrolito y pasando un electrón al electrodo. En la interfaz electrodo-piel el fundamento es el mismo, solo que, en este caso, es necesario incluir una especie de gel con alto contenido en aniones, sobre todo el anión cloro.

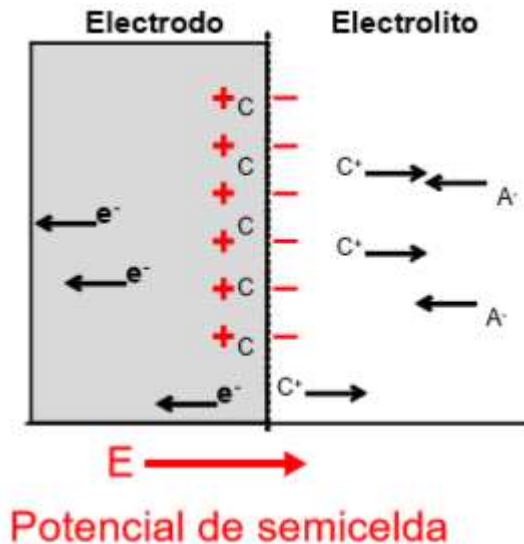


Figura 5.5. Origen del potencial de media celda
 Fuente: *Material Auxiliar de clase de Instrumentación Biomédica. Prof. Dr. R. Navas – Dpto. de Electrónica.*

Quando se coloca un electrodo sobre la piel, tiene lugar un proceso de distribución de cargas, apareciendo así lo que se conoce como potencial de media celda o semicelda (véase *Figura 5.5*). Esta interfaz se puede

modelar como tensión continua en serie con una impedancia. Aun así, tiene la particularidad de que, si existe algún tipo de movimiento entre ambas superficies, se generará cierta inestabilidad en este potencial, lo que se traducirá en artefactos en la señal final. En nuestro caso, como se puede ver en el *Anexo VI: Comentarios sobre los resultados obtenidos*, este efecto (o defecto, mejor dicho) será uno de los grandes condicionantes en el uso práctico de nuestro dispositivo.

5.1.4. Componentes de interfaz de usuario

En cuanto a la interfaz de usuario se refiere, debemos contar con tres elementos fundamentales: el display LCD y el conjunto conformado por los dos pulsadores y sus correspondientes implementaciones individuales.

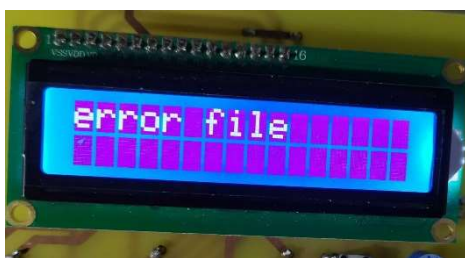


Figura 5.6. Mensaje del sistema en el display.
 Fuente: *Fotografía realizada por el alumno.*

La implementación del LCD (Liquid Crystal Display) es fundamental para facilitar el trabajo durante el desarrollo, ya que permite incluir todos los elementos de control necesarios (ver *Figura 5.6*), con el fin de comprobar el transcurso correcto de la ejecución del código. El LCD cuenta con un

total de 8 pines digitales para entrada y salida de datos. Sin embargo, nosotros sólo usaremos los últimos 4. Así, los pines digitales desde el D4 al D7 van conectados a los pines digitales entre el 2 y el 5 de Arduino. Esto es así porque

el bit más significativo es el D7, y como sólo necesitamos 4 pines, elegimos los 4 últimos.

Por otro lado, VDD (alimentación) va al nodo de alimentación a 5V de Arduino, VSS (tierra) y RW (alternar lectura o escritura; lo dejamos fijo en escritura) van al de tierra. El pin V0 (ajuste del contraste) se conecta a la salida del potenciómetro, ya que es la herramienta con la que controlaremos el contraste del display.

El pin RS (selección Dato/Control) se conecta al pin 7 y el pin E (habilitar o deshabilitar el display) al 6, ambos en Arduino. Por último, los pines A y K se conecta en alimentación y tierra respectivamente, ya que son el ánodo y cátodo del LED de luz de fondo del display.

Conexiones del Display LCD		
PIN	Función	Conexión en Arduino
VSS	GND (Tierra - 0V)	Tierra
VDD	Alimentación (5V)	Alimentación a 5V
V0	Ajuste de Contraste	Salida del Potenciómetro
RS	Selección Dato/Control	Pin digital 7
RW	Selección Lectura/Escritura	Tierra (fijado a escritura)
E	Habilitar/Deshabilitar	Pin digital 6
D 0-7	Pines digitales. El bit menos significativo es el D0, siendo el más significativo el D7.	Desde D4 hasta D7 se conectan a los digitales, respectivamente, del 2 hasta el 5. Desde D0 a D3 sin uso
A (LED+)	Ánodo del LED de luz de fondo	Alimentación a 5V
K (LED-)	Cátodo del LED de luz de fondo	Tierra

Tabla 5.2. Resumen de los pines del LCD.
Fuente: *Apuntes redactados por el profesor Daniel J. Bolaños.*

En la tabla 5.2 podemos comprobar un resumen donde se detallan las principales características de los pines de conexión de nuestro display digital.

El otro aspecto básico para completar la interfaz de usuario es el teclado. Éste incluye sólo dos botones, los cuales se corresponden con los dos únicos pines digitales libres de Arduino. Esto es debido a que el resto están ocupados por el LCD y el módulo de memoria SD, salvo los pines 0 y 1 que se corresponden con la comunicación directa a través del puerto serie, reservados para el módulo bluetooth. La interfaz manual podría ser más sencilla al uso, en caso de incorporar más de dos botones. Sin embargo, adecuándonos a las posibilidades de Arduino, y simplificando al máximo las posibles opciones, se puede simplificar el diseño de una interfaz que se maneje con solo dos botones, como explicaremos más adelante. Sí es necesario recalcar que un botón tendrá la función de avanzar en opción, mientras que el otro está destinado a seleccionar la opción que se ha elegido con el botón de cambio de opción.

En nuestro caso, ambos pulsadores están preestablecidos según la configuración a nivel bajo, es decir, se deberá leer un estado lógico *LOW* en el pin respectivo a cada botón para que se produzca algún cambio en el sistema, ya que su estado de reposo se corresponde con el nivel *HIGH*. La implementación de los pulsadores es simple. Sin embargo, el uso de estos elementos suele traer de manera implícita dos problemas típicos, el producido por el estado de alta impedancia y el efecto rebote o *debounce* (véase *Figura 5.7*).

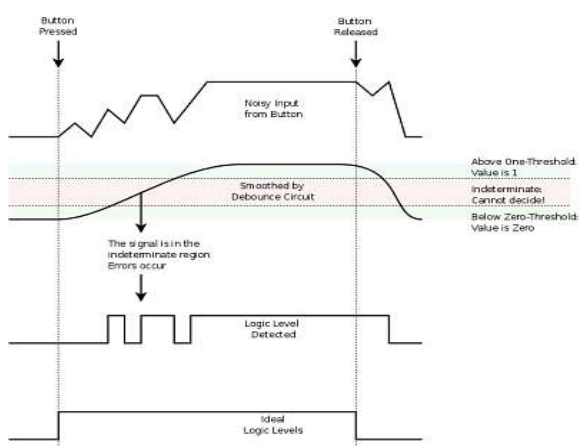


Figura 5.7. Distintas señales derivadas del efecto *debounce* y su corrección. Fuente: *Blog sobre programación "Harder, Better, Faster, Stronger"*. <https://hbfs.wordpress.com/2008/08/20/debouncing-using-binary-finite-impulse-reponse-filter/> (Visto por última vez en mayo de 2019).

Nota: Los niveles lógicos interpretados en la imagen no se corresponden con los que se dan realmente, ya que cuando se acciona el pulsador, el nivel lógico que se debe leer es el 0.

Para solucionar el estado de alta impedancia se han añadido dos resistencias de 10kΩ, mientras que, para paliar el rebote, se incluyen dos

condensadores de 22nF. Además, se debe recalcar que el efecto rebote está controlado también a nivel de software.

5.1.5. Memoria SD

En cualquier dispositivo Holter es necesario almacenar la información en una memoria externa de gran capacidad, pues el sistema deberá ser capaz de estar recabando información durante 48 horas consecutivas.

Conexiones del Módulo de la Memoria SD		
Pin	Función	Conexión en Arduino
Gnd	Conexión a masa (0V)	Tierra
3.3V	Alimentación a 3.3V	Alimentación a 3.3V
MISO (Master in, slave out)	Entrada de datos a la memoria SD	Pin digital 12
MOSI (Master out, slave in)	Salida de datos desde la memoria SD	Pin digital 11
CS (Chip Selector)	Habilita el <i>modo de lectura</i> del dispositivo receptor de los datos que se enviarán desde la memoria SD	Pin digital 9
SCK (Señal de reloj)	Sincroniza y marca la velocidad de transmisión de bits	Pin digital 13
Gnd	Conexión a masa	En desuso
5V	Alimentación a 5V	En desuso

Tabla 5.3. Resumen de los pines del módulo de inserción de la tarjeta de memoria SD.
Fuente: Prometec. *Bus SPI: Arduino y el Serial Peripheral Interface*.

Para poder utilizar la memoria, necesitamos un módulo que permita la conexión entre Arduino y la propia tarjeta SD. Dicho módulo contiene 8 pines dobles, sin embargo, nosotros sólo utilizaremos 6, ya que las conexiones

necesarias de la tarjeta SD se limitan a los dos pines de alimentación y tierra y los 4 propios de la comunicación SPI entre Arduino y la memoria.

Así, en nuestro caso encontramos que el módulo SD tiene los pines CS (selector de chip; envía la señal de habilitación para recibir datos al integrado conectado), MOSI (Salida de datos hacia otro integrado desde la SD), MISO (entrada de datos a la SD) y SCK (señal de reloj; marca la velocidad de transmisión de bits), los cuales se conectan en los pines 9, 11, 12 y 13 respectivamente.

Como ya se ha dicho, la comunicación entre Arduino y la tarjeta SD se basa en el protocolo SPI, el cual es un tipo de comunicación mediante puerto serie. En Arduino UNO solamente contamos con un único puerto serie, lo que significa que cuando está siendo usado no puede ser usado por otro dispositivo, ni si quiera por el propio Arduino. Sin embargo, este problema no toma tanta relevancia en el momento de usar la memoria como cuando se conecta el módulo bluetooth.

Este protocolo de comunicación fue desarrollado por la marca *Motorolla* en 1982. Es un tipo de comunicación a través del puerto serie, y se sustenta en la interconexión síncrona de 4 hilos de comunicación: uno de entrada de datos (pin MISO: Master in, Slave out -*Maestro dentro, Esclavo fuera*-), otro de salida (pin MOSI: Master out, Slave in -*Maestro fuera, Esclavo dentro*-), uno para la señal de reloj (pin CLK: Clock -*Reloj*-) y el habilitador del sistema receptor de datos enviados desde la propia SD (CS: Chip Select -*Selector de Chip*-) [11].

El protocolo SPI se basa en los dos perfiles comunes en este tipo de comunicación: el máster o maestro, que genera la señal de reloj, y el esclavo.

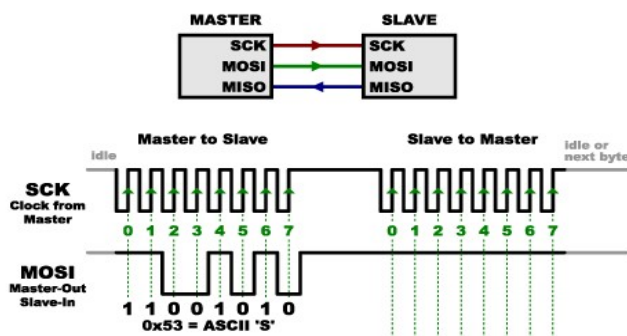


Figura 5.8. Intercambio de señales de manera síncrona y coordinada en la comunicación basada en el protocolo SPI.

Fuente: *Prometec. Bus SPI. Arduino y el Serial Peripheral Interface.*

Cuando el maestro envía información, lo hace a través del hilo MOSI, mientras que el esclavo responderá a través del MISO.

Al ser el dispositivo maestro el que genera la señal de reloj, debe tener alguna referencia

por parte del esclavo sobre si va a realizar algún tipo de respuesta, así como la cantidad de información que va a ser transmitida (véase *Figura 5.8*).

En cuanto a la memoria en sí misma se refiere, estamos usando la tarjeta microSD M203 de Toshiba, con una capacidad de 16 GB (la menor de su generación), la cual es más que suficiente para poder llevar a cabo el almacenamiento de varios ficheros de nuestro Holter de unas 48 horas de duración (el máximo). El propio fabricante se refiere a esta generación como *High Speed M203* debido a la máxima velocidad de lectura, llegando a los 100MB/s.

El uso de este modelo de tarjeta microSD está indicado sobre todo para dispositivos móviles, tablets y cámaras de fotos. Sin embargo, viene acompañada de serie por un adaptador a tamaño SD (convencional), el cual a su vez lleva incorporado un sistema de seguridad mediante bloqueo físico. Dicho adaptador hace posible que podamos usar el módulo SD común de Arduino, obviando que en realidad la memoria sea del tamaño microSD [12].

5.1.6. Conectividad Bluetooth

El hecho de añadir la función de conectividad bluetooth puede despertar cierta controversia, debido a las posibilidades que ofrece, es decir, cabe plantear la incógnita de para qué exactamente queremos añadir esta característica. Desde el punto de vista formativo es interesante manejar este tipo de comunicación, pero esto no es un motivo determinante para ser añadido al proyecto final. Por otro lado, sería llamativo poder conectarse desde el teléfono móvil y poder visualizar a tiempo real la actividad captada. Sin embargo, esto último no es factible, ya que habría que interrumpir el registro en la memoria debido a que tanto el módulo bluetooth como la SD funcionan a través del puerto serie, y el proceso se ralentizaría demasiado, hasta el punto de perder muestras registrables.

Por otro lado, al entrevistar a ciertos auxiliares sanitarios que han manejado este tipo de dispositivos, se advirtió la necesidad de corroborar el correcto funcionamiento del mismo, una vez instalado el Holter sobre el cuerpo del paciente. De hecho, afirmaban que tras haber colocado un sistema homólogo

sobre el paciente y al recabar los datos pasadas las 24 horas requeridas, se habían observado casos en los que no se había registrado correctamente ningún dato determinante por problemas relacionados con la propia colocación inicial. Este problema lo intentamos suplir mediante una programación especial de 30 segundos recomendada para que, desde la aplicación instalada en un sistema Android, se recupere la gráfica de la señal, y así corroborar que efectivamente, todo está en orden.

De este modo, se decide crear una aplicación con el fin de dar al usuario la opción de elegir una interfaz más cómoda que la manual basada en sólo dos botones, además de incluir esa otra opción de prueba gráfica para corroborar que el dispositivo funciona correctamente una vez instalado en el cuerpo del paciente, considerándose así el incluir la conectividad bluetooth como una ventaja de verdadero interés desde el punto de vista técnico y funcional.

En la implementación se está usando un módulo bluetooth del tipo HC-06, que consta de 4 pines: dos de ellos son alimentación y masa, mientras que los otros dos son los pines TX y RX, y se han de conectar en los puertos RX y TX de Arduino respectivamente (a modo de conexión cruzada: en efecto, el de transmisión del bluetooth se conecta en el de recepción de Arduino y viceversa).

Dicho módulo se entiende como una versión simplificada de su hermano, el HC-05. La diferencia principal radica en sus pines: mientras que el HC-05 tiene 6 pines, el HC-06 tan sólo cuenta con 4. Se trata de los pines básicos, los cuales son los dos correspondientes al intercambio de datos a través del puerto serie, tanto el RX (recepción de datos), como el TX (emisión o transmisión de datos), así como un pin de alimentación a 5V o a 3.3V dependiendo de la versión, y el pin de conexión a masa, o tierra (GND-0V).

En cuanto a hardware se refiere, ambos modelos son similares, de hecho, en cierta bibliografía [13] se afirma que un HC-06 se puede transformar en un HC-05 añadiéndole los pines que le falta y reprogramando su firmware. Las ventajas que plantea el HC-05 respecto al HC-06 de cara a nuestro proyecto, se limitan al control de habilitación y del estado de conexión o desconexión. Sí es cierto que el módulo HC-05 admite los perfiles tanto de maestro como de

esclavo, mientras que el HC-06 tan sólo puede comportarse como esclavo, y eso en realidad juega en favor de este último. Esto se debe a que

Además, el módulo HC-06 no espera un comando terminador de línea como el '\n', si no que salta por tiempo [14]. Esto puede suponer una desventaja a la hora de controlar los envíos de datos, sin embargo, es positivo de cara a simplificar el código.

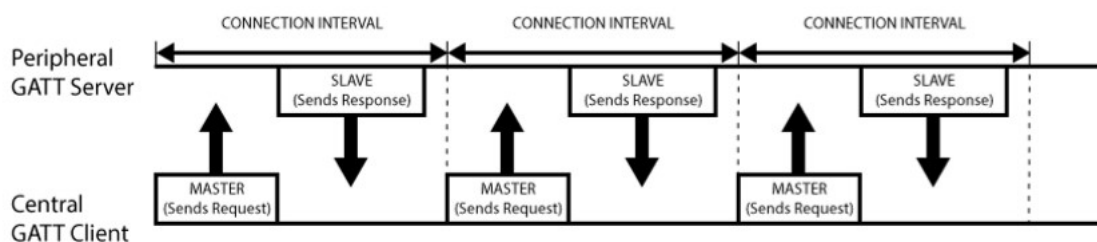


Figura 5.9. Intercambio de información entre un dispositivo maestro y otro esclavo.
Fuente: <http://learn.adafruit.com>

Por tanto, al elegir el módulo HC-06 hemos minimizado el número de pines necesarios, hemos simplificado el código necesario para establecer una conexión, ya que no necesitamos (o no podemos) saber ni ordenar que el dispositivo se conecte o finalice una línea; y, por último, sólo se comporta como esclavo, por lo que no es necesario elegir a qué dirección se debe conectar nuestro dispositivo, simplemente se deja elegir; además, envía información si y solo si recibe un bit desde el dispositivo maestro (véase *Figura 5.9*).

En su estructura física, básicamente se compone de dos piezas: un dispositivo de radio, encargado de modular y transmitir la señal, y un controlador digital, el cual está formado por una CPU, un procesador de señales digitales que hace la función de controlador de enlace.

Al incluir el módulo bluetooth conectado a los puertos específicos de comunicación Rx y Tx de Arduino, en realidad lo estamos conectando directamente al puerto serie. De hecho, estos pines están destinados expresamente a transmisión serie de señales TTL. Sin embargo, esto puede suponer un problema en el momento de cargar código nuevo o recuperar información desde el ordenador directamente. Esto se debe a que la conexión al

ordenador es a través de este mismo puerto, lo que produce colisión de datos sin poder, ni tan siquiera, cargar código en la placa con el módulo conectado. Por ello, desde su implementación, es necesario desmontarlo cada vez que se necesite actualizar la programación de Arduino.

5.2. Software

En este proyecto se han trabajado tres formas de desarrollo software. La codificación de Arduino, la cual supone el corazón del software desarrollado en este proyecto, comprende la interfaz manual del dispositivo, que está a su vez basada en el manejo de dos pulsadores. Por su parte, la interfaz bluetooth asentada sobre la aplicación desarrollada para Android. Finalmente, la función creada en Matlab y basada en la librería *Scrollplot*, y con la que se complementa al dispositivo con la herramienta propia e imprescindible para la correcta lectura e interpretación de los datos.

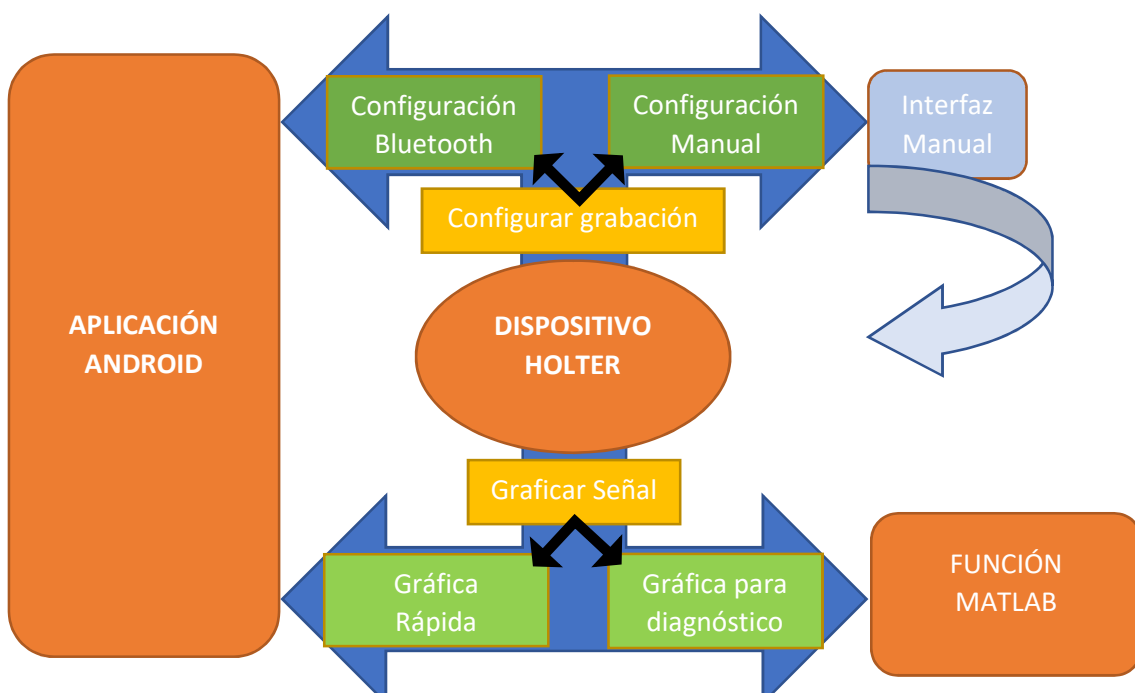


Figura 5.10. Esquema general de funcionamiento del conjunto de actividades.
Fuente: *reconstruido por el alumno*

Como veremos más adelante, los tres programas se diseñan e implementan de forma independiente, pero con el fin de trabajar conjuntamente, siendo el programa de Arduino el centro de control de todos los procesos. La

principal expresión de este aspecto lo encontramos en la combinación entre Arduino y la aplicación para Android (véase *Figura 5.10*).

5.2.1. Codificación de Arduino e interfaz manual

Arduino se programa siguiendo sus dos funciones principales: el *setup()*, la función de inicio del sistema, y el *loop()*, la función principal desde la que invocaremos todos los hilos de la interfaz. Se ejecuta igual que un bucle infinito, por lo que llegado el momento de finalizar el proceso de registro / recuperación de archivo, habrá que forzar la salida.

Así, al encenderse, lo primero que aparece en el display es el mensaje: “*Holter:TFG Iniciando...*”, mensaje que aparece en el *setup()*. Mientras aparezca este mensaje, lo que estará ocurriendo es la inicialización de pulsadores y memoria SD. Además, el sistema esperará dos segundos y medio para dar al usuario un tiempo cómodo para leer el mensaje.

A continuación, se comenzará a ejecutar el *loop()*, o bucle principal. Para comprender el funcionamiento del *loop()* de una forma sencilla y general, podemos observar el Diagrama de Flujo en la *Figura 5.11*. En el bucle principal, antes de cambiar el mensaje del display, se inicializa el puerto serie. A pesar de que lo común es inicializarlo en el *setup()*, en nuestro caso, esto se debe realizar en el bucle principal porque, antes de que acabe el bucle y vuelva a ejecutarse es recomendable cerrar las conexiones, por lo que al reiniciarse el bucle se deben de volver a abrir. Recordemos que cuando el bucle acaba se vuelve de nuevo a la primera parte del propio bucle y no al *setup()*.

Por este mismo motivo, es necesario poner todas las variables globales a su estado predeterminado, es decir, vaciar el nombre del fichero, y volver falsas las variables que indican si se usará bluetooth, si se desea recuperar archivo o si se desea volver al inicio del bucle principal. Se llama también a la función *iniciarVariables()*, en la que se inicializan otras variables globales que se usan en distintas funciones (un contador, una variable booleana que indica si se desea finalizar). A continuación, se muestran las instrucciones de utilización de los botones seleccionar y pasar.

Lo siguiente que ocurre es la llamada a la función *RecuperarRegistro()*. Aquí, el usuario determina mediante la variable *recup* si desea recuperar archivo o si desea saltarse este paso. En caso afirmativo, se debería conectar el bluetooth. En caso negativo, se le pregunta al usuario si desea usar el bluetooth para configurar el dispositivo. Si se responde que sí, la variable *bl* tomará el valor true y se deberá conectar el bluetooth.

En caso negativo, comenzará la fase de programación del registro de la forma manual. Así, se sucederá la invocación a las funciones de elegir fichero, elegir tiempo, elegir frecuencia de muestreo y finalmente *tomaDatos()*, desde donde se regula toda la toma de muestras y almacenamiento. Antes de cada una de estas funciones, exceptuando esta última, se debe llamar a la función de iniciación de variables.

Tras concluir el registro, se cerrará el archivo y el sistema preguntará al usuario si desea continuar o desea finalizar. En caso de querer continuar, se volverá al inicio del bucle principal, repitiéndose otra vez todo lo descrito. En caso de finalizar, se provocará la salida del bucle, mientras que el display indicará que se ha finalizado todo el proceso, pudiéndose desconectar el sistema de la fuente de alimentación.

Las distintas opciones de configuración del sistema se pueden ver en la *tabla 5.4*.

Programación de grabación del Holter	
Variable por determinar	Opciones posibles
Nombre del Fichero	4 caracteres alfanuméricos (en caso de usar más caracteres podrán surgir problemas para crear el registro)
Duración de la prueba o tiempo de muestreo	30s, 24 horas y 48 horas
Frecuencia de Muestreo	100Hz, 200Hz y 1000Hz

Tabla 5.4. Opciones de programación de un registro Holter con nuestro dispositivo.
Fuente: *Diseñado por el alumno.*

Todas las funciones en las que intervienen los botones funcionan del mismo modo. El botón *pasar* va incrementando el contador *cnt*, lo que hace que se vaya avanzando en la opción a elegir gracias al uso de bloques *switch*, con los que se van mostrando por el display las distintas opciones. Cuando se pulsa el botón de selección, la variable elegir que antes estaba a cero, ahora toma el valor del contador, invocando a la función *set* que sí determina el valor final elegido. En caso de no tomar ninguna decisión, la función se vuelve a llamar a sí misma con los parámetros tal cual se han dejado.

El caso de la elección del nombre del fichero es algo especial, ya que en este caso la variable elegir sirve para determinar cuántos caracteres, de los 4 posibles, han sido ya definidos. Cuando la variable elegir alcanza el valor 4, muestra por pantalla. Al estar esos 4 caracteres almacenados en la variable global *nomFich* simplemente se le debe incluir la cadena *.txt*, para, posteriormente, crear un fichero con ese nombre en la tarjeta SD desde la función *tomaDatos()*.

Desde el punto de vista del dispositivo Holter, la interfaz bluetooth comienza cuando se decide recibir un archivo o se desea realizar la configuración del sistema desde la aplicación, lo cual es mucho más rápido y cómodo.

Cuando se decide recuperar el archivo, en realidad se elige obtener el archivo en el dispositivo móvil para graficarlo posteriormente en la aplicación. Así, se invoca a la función *BLsetting()*, desde donde se obliga al sistema a quedar en estado de espera hasta que recibe un carácter con el que se inicia la conexión. Se rompe así el estado de espera, y se llama la función *BLreception()*. Esta función es la que hace posible toda la recepción de datos desde bluetooth, pues es la encargada de interpretarlos. Consta de dos bloques muy similares (véase *Figura 5.11*). El primero se ejecuta si la variable de recuperación está en nivel *True*, y simplemente trata de almacenar la variable recogida desde bluetooth.

Posteriormente, se llama a la función *SendRegistro*, desde donde se abre el archivo a enviar y se va enviando mediante buffers de 300 caracteres. Al finalizar, se cierra la conexión serie, y se vuelve al comienzo del bucle principal.

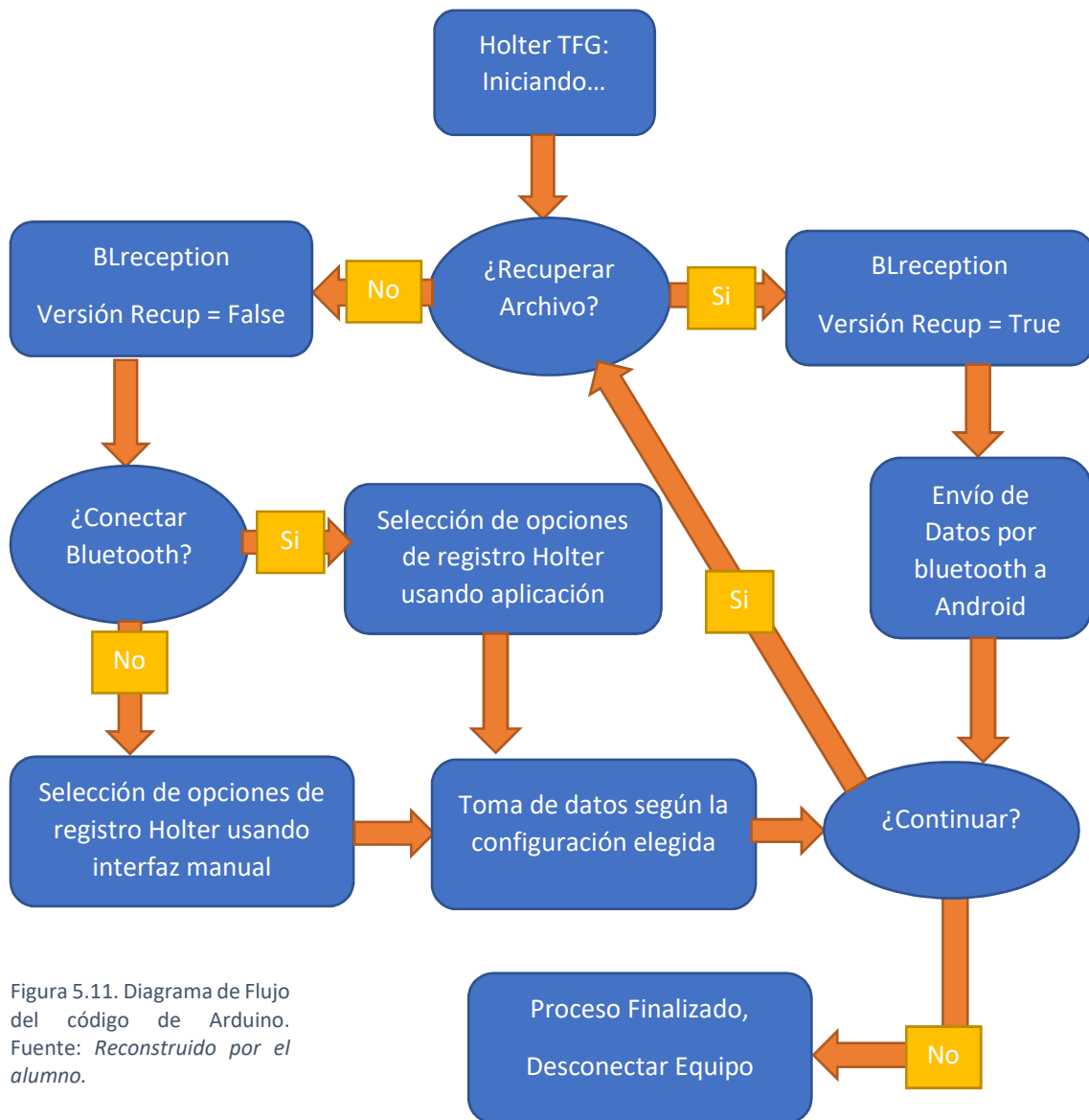


Figura 5.11. Diagrama de Flujo del código de Arduino.
Fuente: *Reconstruido por el alumno.*

Cuando avanzamos a la siguiente opción y se elige conectar el bluetooth, ocurre exactamente lo mismo que lo explicado anteriormente, con la particularidad de que no se activa el primero de los bloques de `BLreception()`, se activa el segundo.

En este bloque se deben separar los 4 caracteres del nombre del registro de los siguientes 2, en los que se adjuntan directamente los valores equivalentes a la variable elegir de las funciones de elección de los parámetros de frecuencia y duración. Estos caracteres se deben interpretar como tipo int, por lo que al provocar el cambio de char a int, se interpretan los valores propios en ASCII, por lo que habrá que restarles el valor del cero, 48 (véase *Figura 5.12*).

```
tReg=(int)c-48;  
char c = Serial.read();  
VelM=(int)c-48;
```

Figura 5.12. Conversión de tipo *char* a tipo *int*.
Fuente: Tomada por el alumno directamente del código.

Tras tener todas las opciones elegidas, se procede con la toma de datos, según el proceso descrito en el apartado anterior.

5.2.2. Interfaz Bluetooth y aplicación Android

En el caso de la aplicación diseñada para el proyecto, se trata de un programa muy sencillo, cuya única importancia radica en otorgar al dispositivo Holter de una herramienta que haga las veces de interfaz y de representación gráfica que difiera de lo convencional. Por ser el más frecuente entre los usuarios, la aplicación que se va a diseñar va dirigida a los dispositivos Android.

Android es un sistema operativo que en sus inicios fue desarrollado por la compañía de software Android Inc., y que desde 2005 es gestionado por Google. Está basado en el Kernel de Linux (y otros softwares de código abierto), y fue concebido especialmente como software para dispositivos con pantalla táctil [15].

Para desarrollar cualquier aplicación para este sistema operativo, habitualmente se utiliza el lenguaje Java acompañado del kit de desarrollo software de Android (SDK), el cual es distribuido gratuitamente por Google. El SDK es un conjunto de herramientas de desarrollo para este sistema operativo, el cual incluye, entre muchos otros elementos, dos indispensables: un emulador y Android Studio, el cual es el IDE oficial de Android.

Además de esto, es necesaria una interfaz de programación de aplicaciones (API), las cuales harán posible el desarrollo del diseño gráfico. De una manera muy estrecha están relacionadas la API y la versión de Android para la que se quiere programar. La justificación de esto es simple: no se puede exigir el uso de una funcionalidad a un dispositivo que no está dotado ni a nivel hardware ni software para la misma. Por esto mismo, lo común es elegir una versión relativamente antigua (sin que esté desfasada), como por ejemplo las versiones Android Ice Cream Sandwich, concretamente, la más recomendada

actualmente es la 4.0.3. Así, se puede asegurar el correcto funcionamiento en esta versión y en las superiores (véase *Figura 5.13*).

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99,6%
4.2 Jelly Bean	17	98,1%
4.3 Jelly Bean	18	95,9%
4.4 KitKat	19	95,3%
5.0 Lollipop	21	85,0%
5.1 Lollipop	22	80,2%
6.0 Marshmallow	23	62,6%
7.0 Nougat	24	37,1%
7.1 Nougat	25	14,2%
8.0 Oreo	26	6,0%
8.1 Oreo	27	1,1%

Figura 5.13. Conjunto de versiones de Android superiores a la versión 4.0. En cada versión se aclara el código de la API correspondiente, así como el porcentaje de dispositivos Android que soportarían en la actualidad una aplicación desarrollada para cada versión. Fuente: *Imagen rescatada del entorno Android Studio*.

Cuando la aplicación se ha terminado, se debe construir el archivo de extensión *.apk*, el cual es el archivo comprimido en formato APK que se instalará en el dispositivo móvil, pudiéndose ejecutar la aplicación diseñada.

Debido a que el verdadero carácter de la aplicación no es otro que demostrar de algún modo que la funcionalidad bluetooth del dispositivo Holter funciona correctamente, finalmente y tras haber comenzado los trabajos de programación en Android Studio, se decidió implementarla usando AppInventor debido a la rapidez con la que se podía acabar una aplicación tan sencilla como la nuestra.

Dicho todo esto, es el momento de plantear las distintas posibilidades derivadas de concebir la aplicación como una herramienta más del propio dispositivo de ECG ambulatoria. Las necesidades detectadas principalmente se reducen a conseguir una interfaz rápida, cómoda y sencilla con la que programar nuestro Holter, así como el hecho de poder graficar pequeños registros de prueba. Todo ello debe realizarse únicamente a través de bluetooth.

Por esto, nuestra aplicación está capacitada para enviar la configuración al holter de manera que éste la reconozca y la interprete correctamente, para recibir y almacenar un fichero desde el holter y, por último, graficarlo.

Desde la interfaz de la aplicación, encontramos 4 pantallas distintas. La primera de ellas, la de Inicio, es la que da paso a dos de las pantallas secundarias. Para ello, encontramos 3 botones (véase imagen 5.14) con los que

podremos acceder al Menú Principal, a la sección de Gráfica o simplemente consultar información general sobre la aplicación.



Figura 5.14. Pantalla Inicio de la aplicación HolterArduino. Fuente: Tomada por el alumno.



Figura 5.15. Pantalla del Menú Principal de la aplicación HolterArduino. Fuente: Tomada por el alumno.

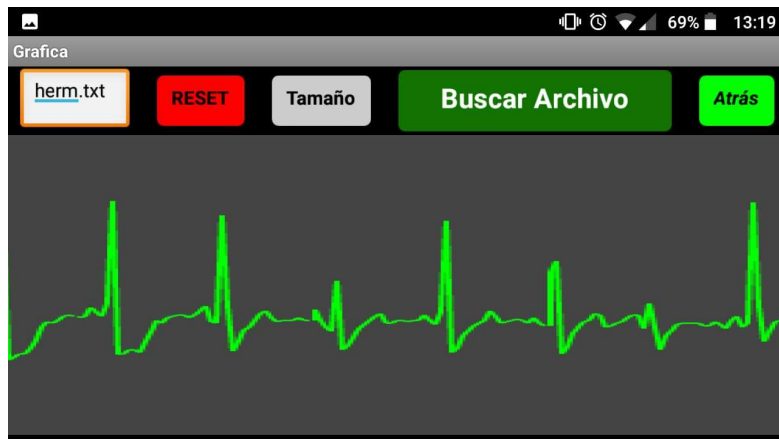


Figura 5.16. Pantalla de gráfica de la aplicación HolterArduino. Se está representando el registro herm.txt Fuente: Tomada por el alumno.

A continuación, se indicará la funcionalidad de los distintos botones y la importancia de las pantallas a las que dirigen. Se comprenderá aún mejor gracias al Diagrama de Flujo de la aplicación HolterArduino en la *Figura 5.17*.

- *Interfaz Holter.* Nos conduce a la pantalla del *Menú Principal* (véase *Figura 5.15*), a través de la cual podremos conectarnos a nuestro dispositivo Holter mediante el bluetooth (debe estar activado antes), para posteriormente configurar todo el proceso de registro. También da acceso a la subpantalla propia de obtención del archivo desde el Holter a través de bluetooth. Por todo ello, es la pantalla más importante de la aplicación a nivel funcional.

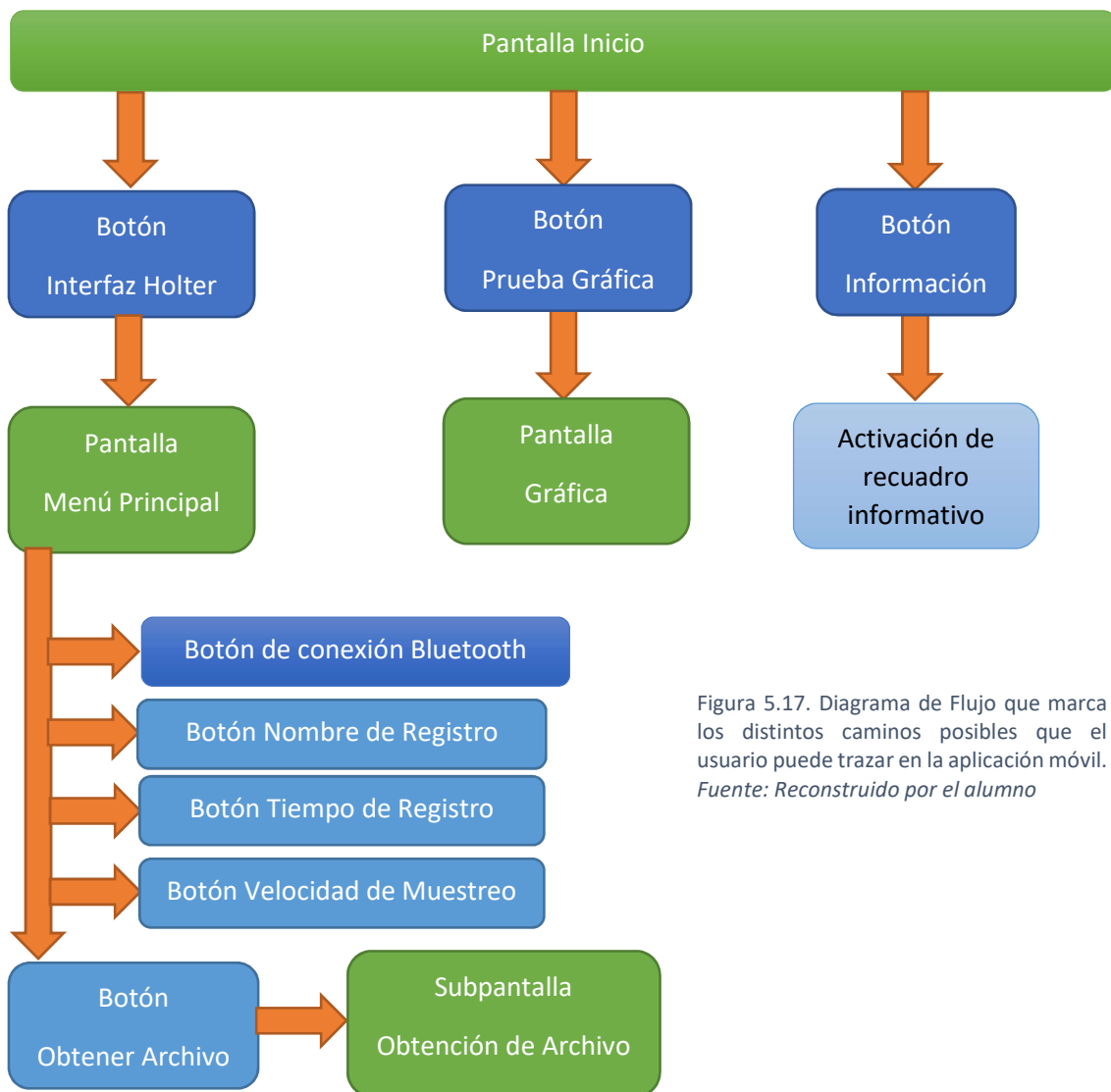


Figura 5.17. Diagrama de Flujo que marca los distintos caminos posibles que el usuario puede trazar en la aplicación móvil.
Fuente: Reconstruido por el alumno

- *Prueba Gráfica.* Nos derivará a la pantalla *Gráfica* (véase *Figura 5.16*), donde tendrá lugar todo el proceso de graficar el archivo recibido mediante bluetooth. Para ello, se buscará en la carpeta predeterminada el nombre introducido en la caja de texto cuando se pulse el botón *Buscar*

Archivo. Si el registro existe se graficará de inmediato; en caso contrario, saldrá un mensaje indicándolo en la pantalla. El botón *Reset* reinicia todo el proceso de graficar, y pulsando el botón *Tamaño*, podremos comprobar el número de muestras presentes en el archivo a graficar.

- Información. Aparecerá un cuadro de texto con información acerca de la aplicación y la intención con la que se diseñó.

5.2.3. Matlab

Comprender el formato de los datos es crucial para poder construir una herramienta de graficación. Así, cuando hablamos de un registro de 30 segundos, entendido como archivo creado por nuestro dispositivo, en realidad nos enfrentamos a un fichero *.txt* con cerca de $3 \cdot 10^3$, $6 \cdot 10^3$ o $3 \cdot 10^4$ muestras¹, dependiendo del valor de la frecuencia de muestreo (respectivamente 100, 200 ó 1000 Hz). Cada muestra se almacena solamente como un número con hasta tres cifras decimales, las cuales se corresponden con el valor del voltaje del ECG.

A esto debemos añadir que la primera línea del archivo se reserva a la duración de la grabación total en segundos. Esto es importante debido a que no se almacena el vector tiempo, por lo que habrá que reconstruirlo en el momento de realizar la gráfica. Así, el proceso de reconstrucción se reduce a lo mostrado en la *Figura 5.18*.

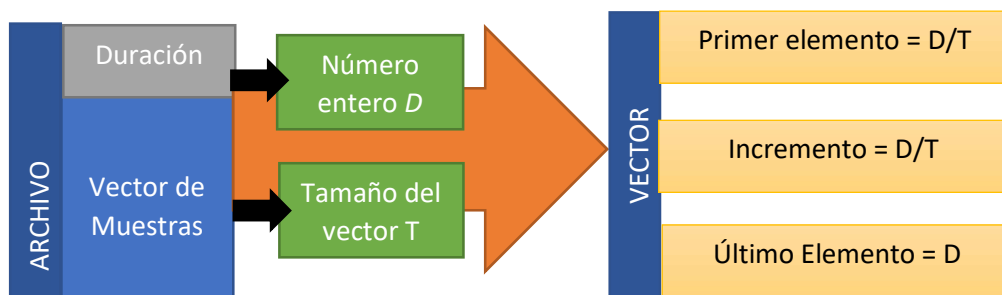


Figura 5.18. Esquema sobre los parámetros usados para la reconstrucción del vector. Se declara el vector usando la estrategia *ValorInicial, ValorIncremento, ValorFinal*, tan propio de Matlab.

Fuente: *Representación en Matlab mediante la función plot*.

¹ Nos referimos al número de muestras como una aproximación, debido a que es frecuente perder información a la hora de registrarla. Estas pérdidas no son significativas de cara a la interpretación de los resultados.

Sin embargo, el hecho de poder representar un registro de nuestro dispositivo en Matlab no tiene demasiado interés, partiendo de que cada registro es simplemente un vector con un gran número de elementos. Por ello, la importancia de la función que se ha diseñado en realidad recae expresamente en la inclusión de herramientas que permiten al médico interpretar el resultado y emitir un diagnóstico.

Para ello debemos tener en cuenta varios aspectos. El primero de ellos, y tal vez, el más intuitivo, consiste en resolver el problema de la longitud de la señal. Cuando se pretende representar una gráfica que ha abarcado unos cuantos segundos es sencillo, ya que se conseguiría invocando la función *plot()* o alguna similar. Sin embargo, la gráfica mínima que se debe poder representar abarca 30 segundos de grabación, grabando los datos con una frecuencia de muestreo aproximada de 100 Hz en la configuración que usa la menor frecuencia, lo cual supone representar 3000 puntos. La función *plot()* no es válida, ya que concentra la gráfica demasiado, dificultando en gran medida la tarea de interpretar la señal (véase en *Figura 5.19*).

En primera instancia, este problema se trató de resolver dividiendo la gráfica en 2 y 3 tramos, solución que puede ser factible para gráficas de 30 segundos, pero que para gráficas de 24 horas o 48 horas es completamente disfuncional (véase *Figura 5.20*).

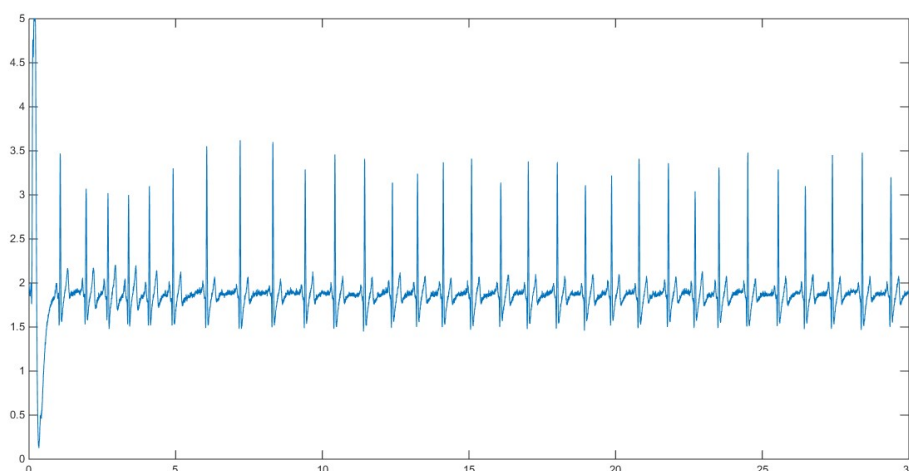


Figura 5.19. Gráfica en la que se representa el valor recabado desde el Holter respecto al tiempo. Grabación realizada durante 30 segundos.
Fuente: *Representación en Matlab mediante la función plot.*

La solución final por la que se opta requiere el uso de la librería llamada *scrollplot*, la cual está disponible en la web oficial de Mathworks [16]. Se trata de un complejo código de más de 1700 líneas, el cual permite la adición de dos pequeñas ventanas paralelas a los ejes X e Y, y que dotarán a nuestra figura de la capacidad de desplazamiento colineal con los mismos. Para nosotros esto es muy interesante, porque además es posible fijar un tamaño de ventana estándar, aunque se permita después ampliar la gráfica con respecto a un solo eje.

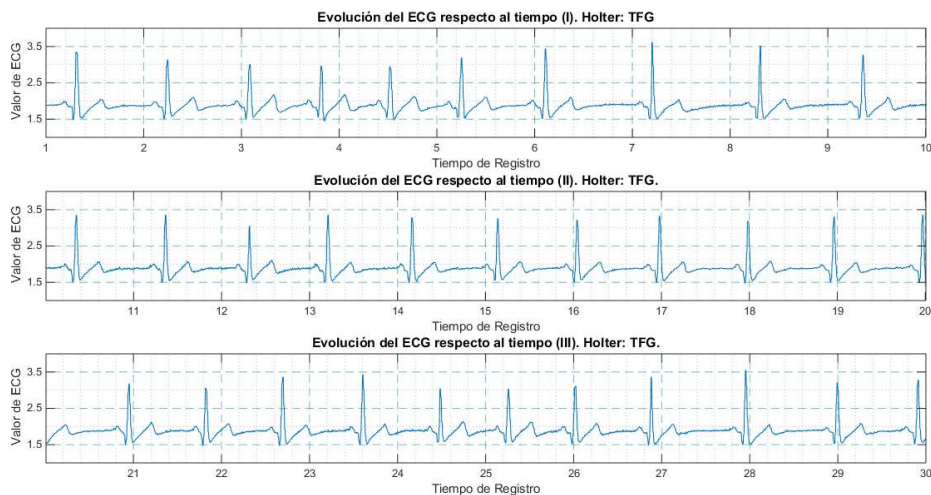


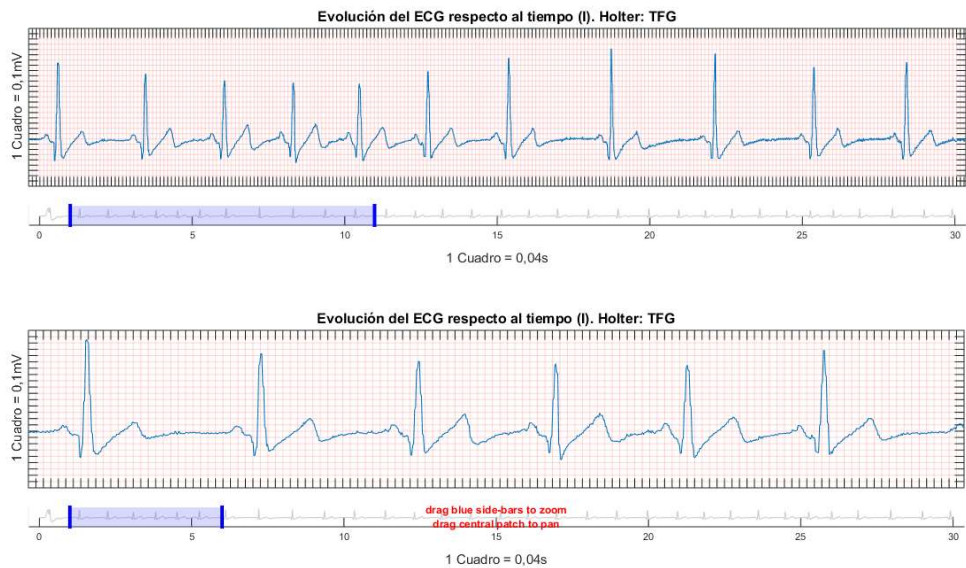
Figura 5.20. Gráfica en la que se representa el valor recabado desde el Holter respecto al tiempo. En esta ocasión se pone a prueba la división del registro en tres tramos distintos con el fin de representarlo en su totalidad. Grabación realizada durante 30 segundos. Fuente: Representación en Matlab mediante la función *plot*, combinada con *subplot* y realizando la división de los vectores.

En nuestro caso, solamente usaremos la función *scroll* en sentido horizontal, respecto al eje de abscisas, es decir, el eje temporal, ya que no tiene sentido realizar un desplazamiento vertical en la representación de un ECG. El resultado final consta de dos partes: una ventana superior donde se grafica un tramo del total la señal; y otra ventana inferior y más pequeña, la cual representa el total de la grabación, y a través de la cual vamos haciendo avanzar el par de cursores que determinan lo que debe de representarse en la zona superior.

Como se ha mencionado, efectivamente se trata de dos cursores que funcionan como una única unidad. El hecho de que haya dos cursores se justifica debido a que, con la distancia que existe entre ambos, se puede determinar el tamaño del fragmento representado en la zona superior, detalle que aparece explícitamente en las Figuras 5.21 y 5.22. Resulta también llamativo que el *grid* o malla de fondo no se autodimensiona, si no que permanece constante con los

valores que indican, y no con el zoom que se le impone. Esto es interesante, como veremos a continuación.

El otro aspecto fundamental para el diagnóstico, que también se pretendía solucionar con la representación mediante Matlab es la funcionalidad que tradicionalmente se le asignaba al papel milimetrado, entendido éste como herramienta comparativa y de medición de parámetros propios del ECG.



Figuras 5.21 (superior) y 5.22 (inferior). Ambas figuras hacen referencia a la representación de un registro de 30 segundos grabado con el dispositivo Holter. Se ha utilizado la librería *Scrollplot* para obtener ambas gráficas. En la figura 5.21. Observamos un tamaño de ventana seleccionado por el usuario, y que está marcado por los cursores. La imagen 5.22 se presenta como el tamaño estándar de representación. Fuente: *Representación en Matlab mediante la función Scrollable implementada por el alumno y basada en la librería Scrollplot.*

Además, como requisito siempre se exige no sólo la presencia de una cuadrícula, si no que la cuadrícula se mantenga constante en los valores que determinen desde el principio. En la lectura en papel milimetrado, este problema no existe debido a la incapacidad de realizar un aumento de la señal o la imagen. Sin embargo, es importante tener en cuenta esto desde el principio cuando la visualización se realiza en dispositivos que sí permiten dicha funcionalidad.

Es por esto por lo que no podemos conformarnos con el hecho de añadir un *grid* común con su variante *grid minor*. De hecho, para obtener el resultado necesario, hay que tratar los ejes de la cuadrícula de manera independiente según su orientación [17]. Así, en realidad lo que se hace es declarar los distintos

ejes que queremos que se representen, fijados cada cierta distancia hasta un límite preestablecido. Realizándolo de esta manera, se puede obtener el resultado que se observa, como se dijo anteriormente, en las Figuras 5.21 y 5.22.

Para el profesional médico que debe interpretar la gráfica del ECG es completamente necesario la presencia de una cuadrícula fija que se atenga a las medidas que, según los estándares, se suelen representar. Así, para fijar los ejes, fue necesario utilizar como modelo un papel milimetrado y una regla de medición especial para interpretación de ECG utilizada por profesionales del SAS.

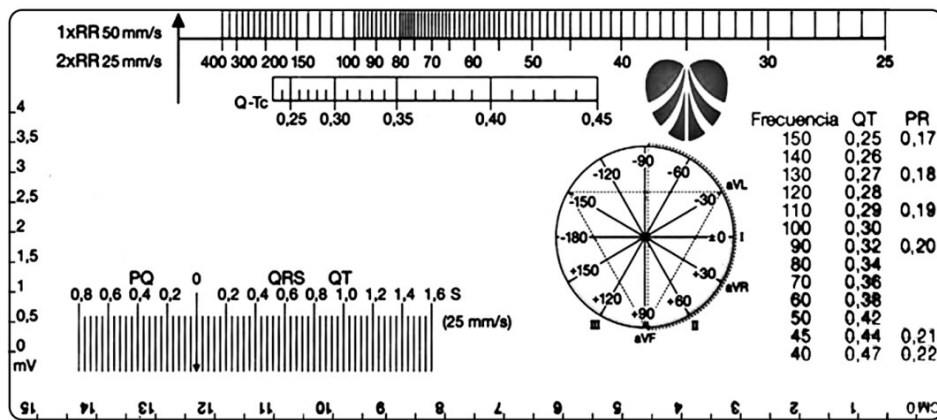


Figura 5.23. Regla cardiológica para medición de frecuencia cardíaca y otros parámetros en un electrocardiograma sobre papel milimetrado estándar. Fuente: *Pinterest*.

Observamos que en la regla cardiológica (véase Figura 5.23) todos los parámetros se miden en base al papel milimetrado, por lo que es necesario darle mucha importancia a uso de la cuadrícula fina, pues es determinante para realizar las medidas oportunas tanto en el eje de abscisas (tiempo) como en el eje de ordenadas (voltaje). El papel milimetrado va avanzando a una velocidad de 25 mm/s, por lo que cada cuadro de la cuadrícula menor equivale a 0.04s. Por otro lado, un milímetro equivale a 0.1 mV [18]. Estas mismas medidas nos servirán para comprobar si nuestro dispositivo está calibrado. Para ello, usaremos un electrocardiograma real, realizado al propio alumno, para compararlo con la gráfica de la representación en Matlab (véanse Figuras 5.24 y 5.25).

Como se puede observar, la diferencia pico-pico se mantiene en 15mm, o hablando en diferencia de potencial, se mantiene en 1.5 mV.

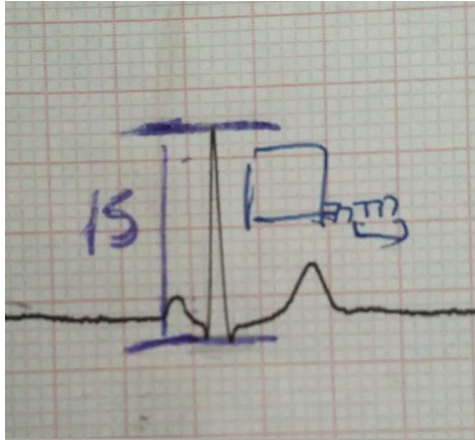


Figura 5.24. Papel milimetrado con señal de ECG real realizada en el ambulatorio de Lucena (Córdoba). El médico de cabecera del alumno señala las medidas de la diferencia de potencial (en cuadros) que debe marcar un ECG de un paciente sano. Fuente: Tomada por el alumno.



Figura 5.25. Imagen obtenida mediante Matlab. Representación de grabación de nuestro Holter. Se realizan las medidas y hay coherencia entre lo indicado por el doctor en el papel milimetrado real y lo obtenido por nuestro dispositivo. Fuente: Tomada por el alumno.

6. Conclusiones

En este capítulo se expondrán las principales conclusiones obtenidas directamente del trabajo y estudio realizados durante el desarrollo del presente proyecto. Serán plasmadas de forma clara y concisa, y servirán a modo de reflejo del valor académico y formativo propio de un Trabajo de Fin de Grado. Tras la enumeración de las conclusiones y para finalizar, se realizará una reflexión general de carácter personal entorno a este documento comprendido como trabajo final del grado.

- En respuesta a la hipótesis planteada acerca de si Arduino UNO es capaz de soportar las principales características y requerimientos propios de un sistema de electrocardiografía ambulatoria; se propone, a modo de solución, el desarrollo e implementación del dispositivo hacia el cual está dedicado el presente documento.
- Se presenta un dispositivo capaz de capturar la señal de ECG, almacenarla en una memoria externa con el fin de ser representada en el futuro, con la cualidad de ser portátil gracias a su fuente de alimentación interna, y con una interfaz de usuario propia y autosuficiente. Todo ello concuerda con la definición de dispositivo Holter, lo cual corrobora la afirmación anterior.
- Además de las funcionalidades propias del dispositivo Holter, se han explorado otras funcionalidades relacionadas con el mismo. Así, se incorpora al trabajo una herramienta de visualización que permite la correcta interpretación, y portando, la emisión del diagnóstico en base a la representación gráfica.
- En la misma línea del punto anterior, se han intentado buscar funcionalidades extraordinarias en cuanto a un Holter convencional se refiere. Así, la dotación de conectividad bluetooth no es algo común en este tipo de dispositivos, y sin embargo, el resultado puede ser no solo satisfactorio, presentando una nueva dotación de utilidad de cara a un caso práctico real.
- Arduino, desde el punto de vista de la docencia y la formación, es muy interesante, pues permite al alumno de ingeniería mantener un primer

contacto con el mundo de los microcontroladores y la programación de los mismos de una manera cercana y sencilla de comprender. A lo largo del desarrollo de este trabajo, el alumno se ha visto comprometido a redescubrir este sistema, así como a explorar todas las posibilidades que desde su software más sencillo se pueden llegar a alcanzar.

- Otro de los aspectos más constructivos de este tipo de sistemas es que obligan a trabajar pensando siempre en un nivel de implementación muy bajo, a nivel de elementos electrónicos. Esto supone una ventaja a la hora de conocer cómo funcionan las comunicaciones a estos niveles, los buses de datos en el puerto serie, la deconstrucción y posterior reconstrucción que se realiza en el envío de ficheros completos mediante bluetooth, y directamente como consecuencia de ello, los problemas derivados que surgen relacionados con el código ASCII y similares.
- En cuanto a la gestión de la memoria externa se refiere, hay que indicar que el hecho de incluir una tarjeta de memoria SD no es lo más habitual cuando se habla de Arduino. Así, se ha realizado un trabajo de inmersión en cuanto a gestión de la memoria se refiere, y esto conlleva investigar los fundamentos principales de la comunicación SPI.
- El hecho de haber vuelto a manejar Matlab pone de manifiesto la gran versatilidad de este software matemático y su utilidad en el campo de la ingeniería. En esta ocasión, la programación de la función de representación es sencilla. El aspecto más importante a destacar entorno a esto es el estudio de las herramientas necesarias para la interpretación de los resultados (herramientas diseñadas por ingenieros para el desarrollo profesional de médicos, fin último de la Ingeniería Biomédica en general). Por otro lado, encontramos la relevancia de la librería *Scrollplot*, cuyo estudio es muy enriquecedor para cualquier programador familiarizado con Matlab.
- El diseño y fabricación de la placa de circuito impreso ha supuesto también un reto, debido a que es la segunda vez que se le plantea algo similar al alumno. Pero a diferencia de la primera, en la que todo el proceso era guiado y supervisado paso a paso, ahora se le plantea al alumno el completo desarrollo de la PCB, desde la creación de las propias librerías. Por primera vez, el alumno se ve implicado en el proceso de

fabricación, manipulación del taladro y realización de soldaduras de los distintos componentes.

Desde el punto de vista de la rama biomédica, en este proyecto se han aplicado conocimientos de prácticamente la totalidad de los campos que se estudian durante el grado. En la parte de contextualización, se hace un trabajo de investigación tratando de explicar cómo se transmite el impulso que produce el latido, pasando por las características más específicas del músculo cardíaco que lo hacen posible, y todo con el fin de comprender mejor el trabajo de los electrodos de nuestro dispositivo.

Así, con los electrodos empezamos el trabajo más puramente técnico. El estudio de circuitos eléctricos y/o electrónicos se hace presente desde el principio. Por otro lado, los conocimientos adquiridos en asignaturas como instrumentación biomédica o electromedicina han sido fundamentales para poder comprender todo el tratamiento que sufre una señal desde que es detectada por los electrodos hasta que es emitida en una gráfica, en la pantalla de un dispositivo. Así mismo, es imposible conocer el funcionamiento de los buses de datos y todas las comunicaciones que se dan en nuestro sistema sin antes haber comprendido asignaturas como Arquitectura de Computadores o Redes y Sistemas Distribuidos. Y por último, del mismo modo, asignaturas como Estructuras de Datos y Algoritmos han sido piezas fundamentales para poder gestionar y optimizar los algoritmos y funciones que hacen que Arduino pueda funcionar como un Holter.

En definitiva, el proyecto en su práctica totalidad ha servido como un reto que se supera mediante la aplicación de los conocimientos adquiridos durante estos cuatro cursos, suponiendo así, en su conjunto, un acercamiento al mundo real de la ingeniería.

7. Líneas Futuras

En este capítulo se expondrán las propuestas a posibles proyectos futuros que puedan completar, complementar o relacionarse de algún modo con el presente trabajo

7.1. Diseño e implementación del circuito completo de ECG

En esta primera propuesta, se reta al alumno a diseñar todo el circuito de acondicionamiento de señal. Del mismo modo que en el presente se utiliza el circuito incorporado en eHealth, en la propuesta, se utilizaría e diseñado por el alumno. Además, se podría estudiar la posible inclusión de las derivaciones, al menos, las periféricas.

Dicho proyecto se enmarcaría en el ámbito de la electrónica y la instrumentación biomédica, y permitiría una inmersión completa en el campo de la electromedicina.

7.2. Combinación de Holter de tensión y Holter convencional

Además del Holter convencional, existe un tipo de dispositivo de monitorización ambulatoria de la presión arterial, el cual se conoce comúnmente como Holter de presión arterial o Holter de tensión. Al igual que el ECG ambulatorio, registra las variaciones de la presión arterial durante 24 ó 48 horas, y la diferencia funcional fundamental con respecto al Holter de ECG reside en la sustitución de los electrodos por un esfigmomanómetro [19].

Teniendo en cuenta que desde la plataforma de eHealth también es posible medir la tensión arterial, podría ser interesante desarrollar un Holter combinado de tensión y electrocardiografía, con el cual monitorizar ambos parámetros al mismo tiempo.

Desde el punto de vista técnico, supondría al alumno un esfuerzo de optimización de código ya que, como se justificó anteriormente, en el presente

proyecto se ha llevado a Arduino al borde de la estabilidad. También supondría rediseñar la aplicación para Android y completar el código de análisis en Matlab.

Desde el punto de vista médico, su utilidad radicaría en el estudio de anomalías cardíacas en las que se relacionen los niveles de tensión con diversos episodios detectables desde el ECG.

7.3. Aplicación mejorada

Como se justificó anteriormente, la aplicación de este proyecto es sencilla debido al fin al que está destinada: completar la funcionalidad bluetooth de nuestro dispositivo.

Sin embargo, sería interesante de cara a futuros trabajos el implementar una aplicación verdaderamente potente, la cual esté dirigida a establecer la comunicación paciente-doctor, de manera que el doctor pueda recuperar el registro desde su consulta, mientras el paciente aún sigue en su casa.

El planteamiento se asentaría sobre la propia conectividad bluetooth entre el dispositivo móvil del paciente y el holter; mientras que la comunicación entre los dispositivos móviles tanto del paciente como del doctor utilizaría la conectividad a través de internet.

7.4. Dispositivo Holter basado en Raspberry Pi

Otra propuesta interesante desde el punto de vista de las distintas plataformas de aprendizaje, podría ser la implementación de un sistema similar al que se presenta aquí, pero con la particularidad de desarrollarse expresamente para Raspberry Pi en lugar de Arduino.

El hardware que ofrece la plataforma de Raspberry ofrece aún más posibilidades, ya que supera a Arduino en complejidad. Al tener un procesador más potente, se podrían estudiar las posibilidades de realizar análisis algorítmicos de detección de patrones en tiempo real, activando alarmas en caso de detectar arritmias (por ejemplo), o encender un led con cada latido del corazón.

Además, está diseñado para plantear opciones de comunicación mediante wifi y bluetooth mucho más completas, y a la vez más complejas, que Arduino. Se plantearía así la pregunta de cuál de las dos plataformas cubriría mejor las exigencias de un dispositivo médico de estas características, y cuál podría llegar a incluir un mayor número de funcionalidades que las complementen.

7.5. Análisis algorítmico a través de Matlab

Una de las carencias de este dispositivo frente a los que actualmente se están comercializando es la presencia de algoritmos de reconocimiento de anomalías.

Hay muchos estudios matemáticos acerca de algoritmos de reconocimiento de patrones en la señal del ECG, llegando a ser algunos de ellos de gran complejidad e interés, matemáticamente hablando.

Una posibilidad es el uso del dispositivo holter que aquí nos ocupa, en conjunto con su formato de datos, con el fin de obtener registros de ECG para ser analizados. Se podrían implementar diversos algoritmos de este tipo utilizando Matlab con el fin de realizar comparativas evaluando calidad de análisis, y utilidad y tal vez intentar incluirlo al dispositivo desde el propio IDE de Arduino (aunque para ello sería necesario prescindir de ciertas funciones).

ANEXOS

I. Código de Arduino

Se encuentra en el CD Adjunto. Nombre del fichero comprimido:

HolterArduino.zip

II. Código de Matlab

Se encuentra en el CD Adjunto. Nombre del fichero comprimido:

MatlabCode.zip²

III. Archivo APK de la aplicación móvil HolterArduino

Se encuentra en el CD Adjunto. Nombre del fichero comprimido:

HolterArduino.apk

² Encontraremos, además, tres archivos de prueba, creados usando las distintas frecuencias de muestreo. También está el archivo de la librería *scrollplot*, en la que se basa nuestro programa *scrollable.m*.

IV. Desarrollo de la Placa de Circuito Impreso (PCB)

Con el fin de simplificar la implementación del prototipo, facilitar su utilización, mejorar la estética, incrementar su fiabilidad y resistencia a golpes o movimientos bruscos, así como para mejorar la comodidad y, también, para cumplimentar la labor docente y el aprendizaje del alumno, se decide trasladar las conexiones realizadas mediante cables (véase *Figura A.1*) a una placa de circuito impreso.

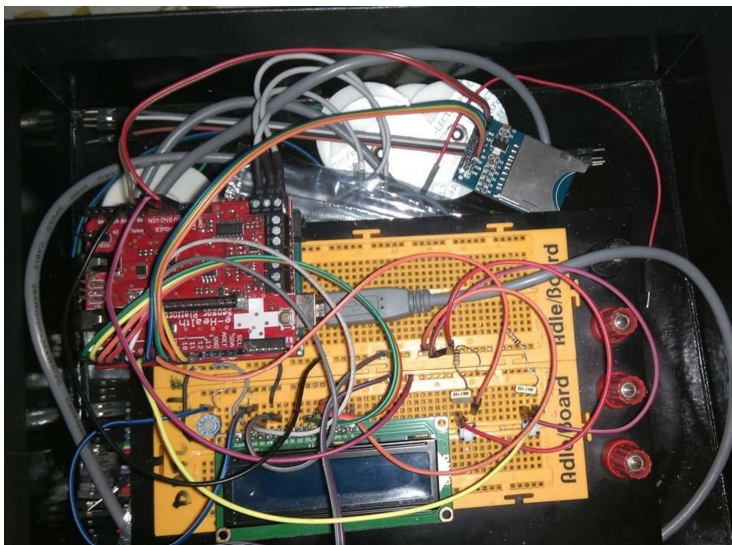


Figura A.1. Fotografía del primer prototipo de este proyecto. Se puede observar con claridad la necesidad de reorganización y unificación de todos los circuitos. Fuente: Fotografía realizada por el propio alumno

Para facilitar el inicio del proceso ha sido determinante el hecho de que, con cada avance en las distintas etapas, se ha ido completando el esquemático general (véase *Figura A.2*). Dicho esquemático se ha diseñado usando el software libre llamado *Fritzing*, y resume todas las conexiones realizadas entre los distintos dispositivos y elementos electrónicos y los pines de Arduino. Con este esquema general, es muy sencillo comenzar a trabajar con el siguiente software, el cual es específico para diseño de PCBs llamado Eagle.

Hay que recordar que aunque las conexiones realmente son como se refleja en el esquemático, superpuesta a la placa Arduino UNO siempre está la placa eHealth.

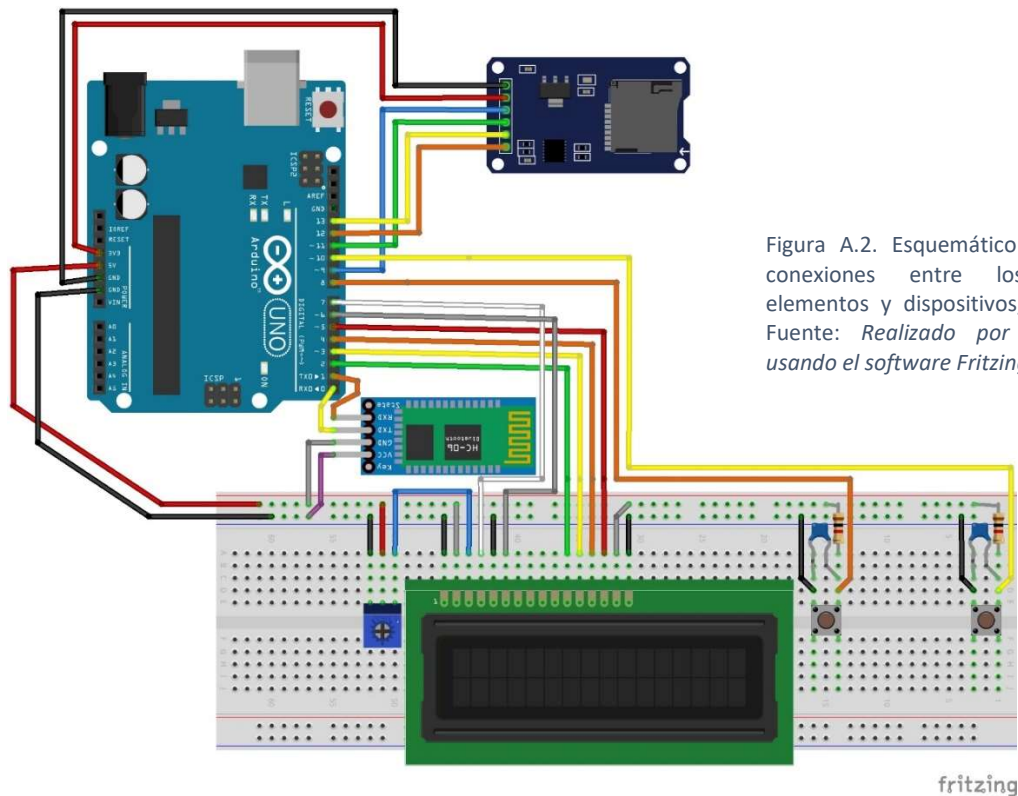


Figura A.2. Esquemático general de conexiones entre los distintos elementos y dispositivos, y Arduino. Fuente: Realizado por el alumno usando el software Fritzing.

En nuestro caso, se ha desarrollado el diseño de la placa con la versión de prueba de Eagle profesional 7.6.0, con la licencia más básica, pero a la vez suficiente, ya que permite trabajar con un máximo de 2 capas (muy adecuado para nuestro proyecto).

El primer paso es desarrollar el esquemático, con el fin de entablar las conexiones en el diseño final de la placa. Así, creamos las librerías de los elementos que no existen, los cuales son el módulo de la memoria SD, el módulo Bluetooth y los pulsadores. Para ello, podemos beneficiarnos de las librerías existentes, realizando una copia de estas, renombrando y editando sobre ellas.

Es importante tener en cuenta, en el momento de crear un elemento nuevo, que después hay que conectar los distintos pines entre el esquemático y el símbolo, es decir, la representación que se verá en el diseño de la placa propiamente dicho. El proceso de entablar dicha conexión se debe realizar desde el *package* de la librería. En el caso del módulo de memoria SD cobra una mayor importancia debido a la duplicidad de los pines.

En todos los casos, el procedimiento a seguir es el mismo. Se comienza dibujando el esquemático, ya que es el lugar en el que se establecerán el número

de conexiones que se pueden entablar con el dispositivo que se está implementando. Lo siguiente será el diseño del símbolo, en el cual se ha de tener especial cuidado con las medidas de los dispositivos, siendo lo más verosímiles posibles con el fin de hacer un diseño óptimo de nuestra PCB. El último paso consiste en completar el *package* uniendo las conexiones que se declararon en el esquemático y que deben estar también presentes en el símbolo. Sirva de ejemplo la librería *MySD* creada por el alumno para visualizar estos tres elementos (véase *Figura A.3*).

Una vez creados los elementos en cuestión, se incluyen en el esquemático con el fin de trazar las conexiones que realizarán las pistas posteriormente. Para ello, será fundamental respetar las conexiones que se especificaron anteriormente, las cuales están resumidas en el esquemático general trazado en *Fritzing* (*Figura A.2*). Así, el resultado final (véase *Figura A.4*) será mucho menos visual que en el anterior esquemático y, de hecho, no aportará nada nuevo, pues esto mismo ya se hizo en el primer esquemático, el cual será mucho más fácil de interpretar.

Para comprender mejor tanto el esquemático como el posterior diseño de la PCB, se puede consultar toda la simbología utilizada desde el programa Eagle para llevar a cabo ambas representaciones. Todo ello se resume en la *Tabla A.1*.

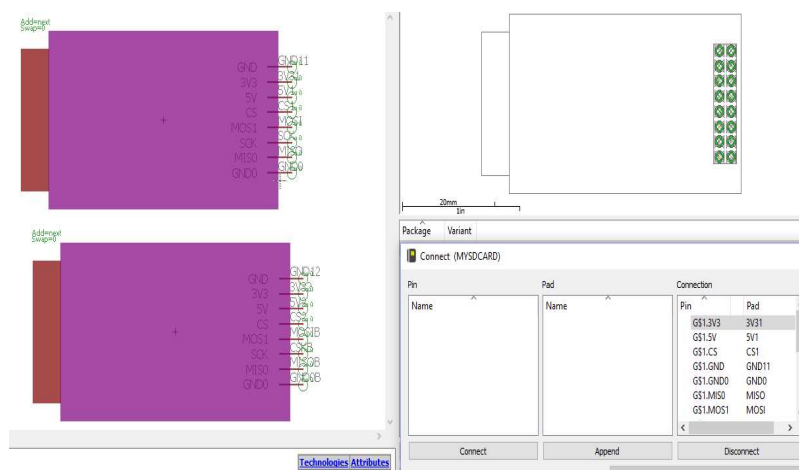
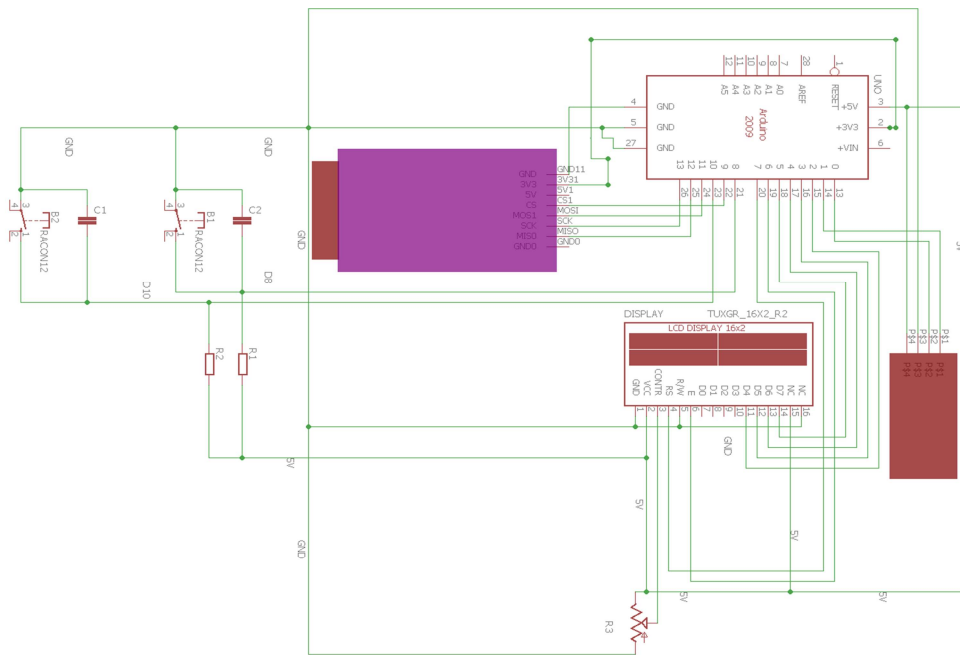


Figura A.3. Librería correspondiente al módulo de la tarjeta de memoria SD, la cual ha sido desarrollada por el alumno. Podemos observar a la izquierda la representación que tendrá en el esquemático, la cual es doble debido a la duplicidad de pines. A la derecha y arriba, encontramos el símbolo que será visualizado en la PCB. A la derecha y abajo veremos la ventana propia del *package*, donde se encuentra la corroboración del correcto establecimiento de las relaciones entre los pines del esquemático y el símbolo.
Fuente: Realizado por el alumno usando el software Eagle.

Relación de elementos necesarios para el diseño de la PCB			
Librería y Dispositivo	Esquemático	Símbolo en PCB	Origen de librería
MySDcard			Módulo de la memoria SD. Creada por el alumno.
HC-06			Módulo Bluetooth. Creada por el alumno.
switch-misc (RACON12)			Pulsador. Propia de Eagle. Se modifica el tamaño del símbolo.
Display-lcd (TUXGR_16X2_R2)			Display LCD. Propia de Eagle.
Pot (TRIM_US-S75P)			Potenciómetro. Propia de Eagle.
RCL R-EU_0204/5 R-EU_			Resistencia. Propia de Eagle.
RCL C-EU050-025X075 C-EU			Condensador. Propia de Eagle.
Arduino ARDUINO2009_2			Sitio web de Eagle en Autodesk.

Tabla A.1. Tabla resumen de la simbología de las librerías usada en Eagle, para llevar a cabo tanto el esquemático como el diseño de la placa de circuito impreso (PCB).



Tamaño recomendado de los elementos de la PCB	
Elemento	Medida recomendada
Anchura de pistas de datos	Superior a 1.7 mm
Anchura de pistas de alimentación y tierra	Superior a 1.9 mm
Diámetro de PADs	2 mm
Diámetro de Vias	2mm
Diámetro de broca	1 mm

Tabla A.2. Medidas recomendadas para los elementos de la PCB.
Fuente: Personal docente del Dpto. de Electrónica de la Universidad de Málaga

Una vez establecidos los parámetros marcados anteriormente, es el momento de distribuir los elementos sobre la PCB. Es necesario no sobrepasar el tamaño máximo posible de $100 \times 125 \text{ mm}$, ya que se corresponde con el tamaño máximo del material en el que se imprimirá nuestro circuito.

Otro aspecto a tener en cuenta es el lugar exacto en el que se insertará nuestra placa. El hecho de que se deba colocar sobre la placa e-Health, la cual a su vez se inserta sobre Arduino, se debe tener especialmente en cuenta que los pines encargados del anclaje de la misma se colocarán sobre la cara *Bottom*, es decir, las soldaduras, a pesar de colocarse normalmente sobre esta cara, en este caso concreto, se realizarán en la cara *Top*. Esto es determinante debido a que las pistas se deben trazar precisamente por la cara por donde se realizará la soldadura, con el fin de garantizar la conectividad.

Otra observación a tener en cuenta a la hora de distribuir las pistas es el hecho de que el módulo de la tarjeta de memoria SD se debe anclar con los pines orientados hacia la cara *Top*, por lo que antes de trazar estas pistas, se debe activar la función *mirror* (espejo) sólo para este elemento.

El resto de soldaduras de pines sí se realizarán por el envés de nuestra placa, como es convencional hacerlo. Todas salvo alguna de las patas de las resistencias y los condensadores, ya que usaremos sus PADs a modo de Vias, lo cual simplificará mucho el circuito.

Siguiendo todas estas indicaciones se obtendrán los archivos con los que se realizará la impresión sobre la placa fotosensible. En este tipo de impresión

no es necesaria la generación de ficheros *.gerber*, sino que lo único necesario son las copias en blanco y negro y en papel de cada una de las caras de la placa (véase *Anexo V*).

A la placa resultante aún se le deben taladrar todas las Vías y PADs, para posteriormente soldar todos y cada uno de los elementos precisos. Durante el proceso de soldadura, se utilizaría un multímetro con el fin de comprobar la continuidad de los distintos circuitos de la placa, así como corregir las soldaduras en caso de que se unan pistas que no pertenecen al mismo circuito.

Una vez concluido este proceso, se deben comprobar todas las funcionalidades de los circuitos. En nuestro caso, fue necesario corregir las soldaduras de los pines hembra del display LCD, y añadir una pista que no se llegó a trazar debido a un error que pasó desapercibido en el esquemático.

Una vez concluida la revisión, se observó que el circuito impreso funcionaba de manera idéntica al circuito primitivo, por lo que se declaró el resultado como satisfactorio, dando por concluido el desarrollo del dispositivo.

Si bien es cierto que nuestro sistema no es demasiado grande, su tamaño supera a los homólogos del mercado. El elemento que engrosa sus medidas es la placa de circuito impreso. Esto se debe principalmente a las tecnologías usadas para su diseño y fabricación, los cuales no permiten una terminación proporcionalmente menor. Los aspectos que han determinado el tamaño actual son básicamente los descritos en la *Tabla A.2*, donde se exigen unas medidas mínimas para los elementos de la propia PCB.

V. Negativos de la Placa de Circuito Impreso (PCB)

Se encuentran en el CD Adjunto. Nombre de fichero comprimido:

PCB.zip³

³ Encontraremos tres archivos: pcb, pcbTOP y pcbBOTTOM.

VI. Comentarios sobre los resultados obtenidos

En cuanto al formato de almacenamiento de los datos, hay que decir que, en una de las últimas revisiones del proyecto, concretamente la realizada para analizar el funcionamiento de todos los circuitos y componentes antes de comenzar a desarrollar la placa de circuito impreso, se advierte del excesivo uso de decimales que no aportan información crucial de cara a un posible diagnóstico. Efectivamente, los datos detectados desde los electrodos del Holter se almacenaban como números reales de hasta 5 decimales. Además, en la misma línea se guardaba el valor del tiempo en milisegundos, lo cual podría llegar a hacer de un fichero sencillo algo completamente desmesurado, para grabaciones de horas de duración. Hasta este momento, en cada línea del registro se estaban almacenando un total de 15 caracteres, en cerca de 3000 líneas en el archivo más pequeño.

Realizando la comprobación, se asume que con 3 decimales se obtiene una gráfica aceptable y apta para la interpretación de la señal. Por otro lado, se opta por crear el vector tiempo desde el dispositivo de graficación, eliminando así una gran cantidad de información superflua. Para ello, se almacena en la primera línea del registro la duración en segundos que tendrá la prueba, con el fin de hacer el cálculo de la frecuencia de muestreo, y así realizar la gráfica respecto a un vector de tiempo verosímil.

Todo ello conduce, finalmente, a que en cada línea del archivo se almacenen tan sólo 7 caracteres (número entero, carácter punto, dos números decimales, más los caracteres de salto de línea y retorno de carro). Nada tienen que ver los archivos del primer planteamiento con respecto al planteamiento actual. Esto produce una gran ventaja no sólo a nivel de almacenamiento en la memoria, sino también en cuanto a velocidad de transmisión a través de bluetooth.

Estas readaptaciones de formato repercutieron también en la función programada en Matlab. El mayor impacto recae sobre el hecho de que no existe un vector de tiempo, sino que hay que construirlo en base a dos datos: la duración en segundos de la grabación, que se corresponde con el primer elemento del registro; así como el número total menos uno (porque la primera

línea no es una muestra) de líneas almacenadas en el propio registro, es decir, el número de muestras.

En la aplicación, por su parte, se decide obviar la primera línea del registro, debido a que su principal función no es el diagnóstico, si no el realizar una comprobación de la correcta colocación y limpieza de la señal. Tampoco toma mayor importancia en cuanto al almacenamiento en la memoria interna se refiere.

En general, las grabaciones recabadas con el sistema son bastante aceptables en cuanto a calidad y resolución. Como se ha mencionado, existen tres posibilidades de frecuencia de muestreo, 100, 200 y 1000 Hz. En realidad, la diferencia es apreciable pero no es significativa debido a la frecuencia a la que se detecta la señal de salida del circuito de acondicionamiento.

El problema real de los resultados obtenidos lo observamos, tal y como se predijo en el apartado 5.1.3 *Electrodos*, debido al movimiento de la piel respecto al electrodo.

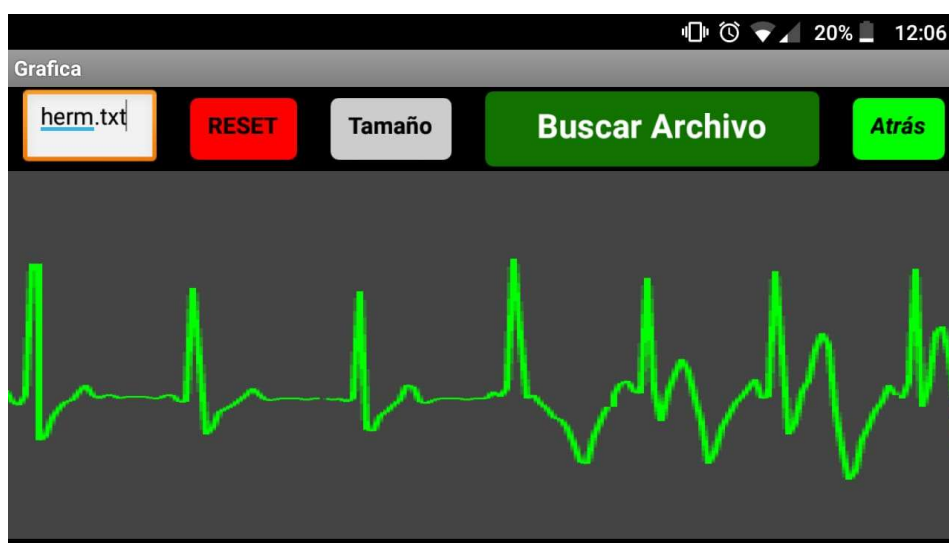


Figura A.5. Captura de pantalla de gráfica realizada desde la aplicación HolterArduino. Se ha representado un registro realizado a un paciente primero en reposo, y después cantando. Se observa el momento justo en el que empieza a cantar.

Fuente: Realizada por alumno con su propio dispositivo móvil.

Cuando el paciente mantiene reposo durante la grabación, o simplemente realiza movimientos suaves, la señal registrada no presenta alteraciones extraordinarias. Sin embargo, se pueden registrar alteraciones cuando se

realizan movimientos bruscos como un salto, el hecho de toser o incluso cantar (véase *Figura A.5*). Se ha observado que los comportamientos bruscos en la respiración quedan reflejados en la grabación, alterando así la señal registrada. Esto se explica, como ya se ha dicho, debido al movimiento brusco de la piel, que se traduce en movimiento de la misma respecto al electrodo.

Esto puede suponer un problema grave debido a que el Holter es un sistema diseñado para acompañar al paciente durante una jornada cotidiana. Es cierto que el sistema de ECG de eHealth está diseñado precisamente para realizar esta prueba diagnóstica, para la cual se requiere un comportamiento estático del paciente. Aun así, nuestro sistema podría servir como Holter convencional en casos concretos de pacientes con una vida sedentaria y tranquila.

VII. Imágenes del sistema en funcionamiento



Tabla A.3. Imágenes de los distintos estados del Sistema mostrados a través del display LCD.
Fuente: Fotografías tomadas por el alumno.

Índice de Figuras

- Figura 2.4. Vista anterior del interior del órgano cardíaco. Se observan claramente las 4 cámaras, así como los tabiques musculares que las separan. Fuente: Practice Anatomy Lab 3.0
- Figura 2.2. Vista a microscopía óptica del tejido cardíaco. Las estructuras observables son las fibras bien diferenciadas entre sí, en cuyo interior encontramos núcleo y estrías. Las estructuras que delimitan a la célula transversalmente son los discos intercalares. Fuente: Practice Anatomy Lab 3.0
- Figura 2.3. Potencial de Acción En las células cardíacas. Observamos el comportamiento del flujo de los distintos iones. Fuente: Quasar 18:07, 7 August 2009
- Figura 2.4. Origen del ECG como resultado de la superposición de los procesos de potencial de acción en la membrana, que tienen lugar a nivel celular en el corazón. Fuente: Fundamentos de Fisiología. Netter
- Figura 2.5. Rangos de frecuencia y amplitud de las señales producidas por los principales biopotenciales. Fuente: Material Auxiliar de clase. Instrumentación Biomédica. Prof. Dr. R. Navas-Dpto. de Electrónica.
- Figura 3.5. Primer Dispositivo Holter de 38 kg de peso. Fuente: Cardiology Journal.
- Figura 3.2. Esquema general de características y aspecto externo del dispositivo Holter BI9800 TL. Vistas anterior y posterior (izquierda y derecha respectivamente). Fuente: BI9800 TL Holter recorder operation manual.
- Figura 4.1. Esquema aclaratorio acerca del funcionamiento de la Metodología en Espiral. Fuente: Administración de Proyectos, Modelos de desarrollo (Google Site).
- Figura 4.2. Diagrama de Bloques Funcionales del dispositivo de ECG ambulatoria. Propuesta concreta del sistema a desarrollar. Fuente: Diseñado por el alumno. (algunas imágenes tomadas de las webs de Cooking Hacks y Toshiba).
- Figura 5.1. Esquemático general de conexiones entre los distintos elementos y dispositivos, y Arduino. Se destaca que estas conexiones no

se realizan directamente sobre Arduino, si no que la placa eHealth se encuentra superpuesta al mismo, tal y como aparece en la fotografía de la derecha. Fuente: Realizado por el alumno usando el software Fritzing.

- Figura 5.2. Fotografía superior de la placa e-Health, en la que están perfectamente identificados cada uno de los elementos de interés para cualquier usuario. Fuente: Web de TBEM Robótica y Electrónica. Subpartado dedicado a Arduino. <https://teslabem.com/arduino/e-health-sensor-platform-para-arduino-y-raspberry-pi/> (Visto última vez en mayo de 2019).
- Figura 5.3. Correcta colocación de los tres electrodos de ECG de eHealth. Fuente: <https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical> (visto por última vez en mayo de 2019).
- Figura 5.4. Circuito de Acondicionamiento de la placa eHealth para realizar el ECG. Fuente: <https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical> (visto por última vez en mayo de 2019).
- Figura 5.5. Origen del potencial de media celda Fuente: Material Auxiliar de clase de Instrumentación Biomédica. Prof. Dr. R. Navas – Dpto. de Electrónica.
- Figura 5.6. Mensaje del sistema en el display. Fuente: Fotografía realizada por el alumno.
- Figura 5.7. Distintas señales derivadas del efecto debounce y su corrección. Fuente: Blog sobre programación “Harder, Better, Faster, Stronger. <https://hbfs.wordpress.com/2008/08/20/debouncing-using-binary-finite-impulse-reponse-filter/> (Visto por última vez en mayo de 2019).
- Figura 5.8. Intercambio de señales de manera síncrona y coordinada en la comunicación basada en el protocolo SPI. Fuente: Prometec. Bus SPI. Arduino y el Serial Peripheral Interface.
- Figura 5.9. Intercambio de información entre un dispositivo maestro y otro esclavo. Fuente: <http://learn.adafruit.com>

- Figura 5.10. Esquema general de funcionamiento del conjunto de actividades. Fuente: reconstruido por el alumno
- Figura 5.11. Diagrama de Flujo del código de Arduino. Fuente: Reconstruido por el alumno.
- Figura 5.12. Conversión de tipo char a tipo int. Fuente: Tomada por el alumno directamente del código.
- Figura 5.13. Conjunto de versiones de Android superiores a la versión 4.0. En cada versión se aclara el código de la API correspondiente, así como el porcentaje de dispositivos Android que soportarían en la actualidad una aplicación desarrollada para cada versión. Fuente: Imagen rescatada del entorno Android Studio.
- Figura 5.14. Pantalla Inicio de la aplicación HolterArduino. Fuente: Tomada por el alumno.
- Figura 5.15. Pantalla del Menú Principal de la aplicación HolterArduino. Fuente: Tomada por el alumno.
- Figura 5.16. Pantalla de gráfica de la aplicación HolterArduino. Se está representando el registro herm.txt Fuente: Tomada por el alumno.
- Figura 5.17. Diagrama de Flujo que marca los distintos caminos posibles que el usuario puede trazar en la aplicación móvil. Fuente: Reconstruido por el alumno
- Figura 5.18.
- Figura 5.19. Gráfica en la que se representa el valor recabado desde el Holter respecto al tiempo. Grabación realizada durante 30 segundos. Fuente: Representación en Matlab mediante la función plot.
- Figura 5.20. Gráfica en la que se representa el valor recabado desde el Holter respecto al tiempo. En esta ocasión se pone a prueba la división del registro en tres tramos distintos con el fin de representarlo en su totalidad. Grabación realizada durante 30 segundos. Fuente: Representación en Matlab mediante la función plot, combinada con subplot y realizando la división de los vectores.
- Figuras 5.21. (superior) y 5.22 (inferior). Ambas figuras hacen referencia a la representación de un registro de 30 segundos grabado con el dispositivo Holter. Se ha utilizado la librería Scrollplot para obtener ambas

gráficas. En la figura 5.21 Observamos un tamaño de ventana seleccionado por el usuario, y que está marcado por los cursores. La imagen 5.22 se presenta como el tamaño estándar de representación. Fuente: Representación en Matlab mediante la función Scrollable implementada por el alumno y basada en la librería Scrollplot.

- Figura 5.23. Regla cardiológica para medición de frecuencia cardíaca y otros parámetros en un electrocardiograma sobre papel milimetrado estándar. Fuente: Pinterest.
- Figura 5.24. Papel milimetrado con señal de ECG real realizada en el ambulatorio de Lucena (Córdoba). El médico de cabecera del alumno señala las medidas de la diferencia de potencial (en cuadros) que debe marcar un ECG de un paciente sano. Fuente: Tomada por el alumno.
- Figura 5.25. Imagen obtenida mediante Matlab. Representación de grabación de nuestro Holter. Se realizan las medidas y hay coherencia entre lo indicado por el doctor en el papel milimetrado real y lo obtenido por nuestro dispositivo. Fuente: Tomada por el alumno.
- Figura A.1. Fotografía del primer prototipo de este proyecto. Se puede observar con claridad la necesidad de reorganización y unificación de todos los circuitos. Fuente: Fotografía realizada por el propio alumno.
- Figura A.2. Esquemático general de conexiones entre los distintos elementos y dispositivos, y Arduino. Fuente: Realizado por el alumno usando el software Fritzing.
- Figura A.3. Librería correspondiente al módulo de la tarjeta de memoria SD, la cual ha sido desarrollada por el alumno. Podemos observar a la izquierda la representación que tendrá en el esquemático, la cual es doble debido a la duplicidad de pines. A la derecha y arriba, encontramos el símbolo que será visualizado en la PCB. A la derecha y abajo veremos la ventana propia del package, donde se encuentra la corroboración del correcto establecimiento de las relaciones entre los pines del esquemático y el símbolo. Fuente: Realizado por el alumno usando el software Eagle.
- Figura A.4. Esquemático general de conexiones entre los distintos elementos y dispositivos, y Arduino. Se trata del paso previo al diseño de

la placa de circuito impreso. Fuente: Realizado por el alumno usando el software Eagle.

- Figura A.5. Captura de pantalla de gráfica realizada desde la aplicación HolterArduino. Se ha representado un registro realizado a un paciente primero en reposo, y después cantando. Se observa el momento justo en el que empieza a cantar. Fuente: Realizada por alumno con su propio dispositivo móvil.

Índice de Tablas

- Tabla 5.1. Especificaciones técnicas de la placa Arduino UNO.
Fuente: Blog del profesorado de la Red Social Educativa de Canarias, Gobierno de Canarias. (Visto en enero de 2019).
- Tabla 5.2. Resumen de los pines del LCD. Fuente: Apuntes redactados por el profesor Daniel J. Bolaños.
- Tabla 5.3. Resumen de los pines del módulo de inserción de la tarjeta de memoria SD. Fuente: Prometec. Bus SPI: Arduino y el Serial Peripheral Interface.
- Tabla 5.4 Opciones de programación de un registro Holter con nuestro dispositivo. Fuente: Diseñado por el alumno.
- Tabla A.1. Tabla resumen de la simbología de las librerías usada en Eagle, para llevar a cabo tanto el esquemático como el diseño de la placa de circuito impreso (PCB).
- Tabla A.2. Medidas recomendadas para los elementos de la PCB.
Fuente: Personal docente del Dpto. de Electrónica de la Universidad de Málaga
- Tabla A.3. Imágenes de los distintos estados del Sistema mostrados a través del display LCD. Fuente: Fotografías tomadas por el alumno.

REFERENCIAS

- [1] Netter. Fundamentos de Fisiología. Capítulo 8: Revisión del Corazón y la circulación. Susan E. Mulroney, Adam K. Myers. Edición 2011.
- [2] Anotaciones del profesor Dr. José A. Andrades. Catedrático del dpto. de Biología Celular, Genética y Fisiología de la Universidad de Málaga.
- [3] Netter. Medicina Interna. Cardiología. Fundamentos.
- [4] Netter. Fundamentos de Fisiología. Capítulo 9: Electrofisiología cardíaca. Susan E. Mulroney, Adam K. Myers. Edición 2011.
- [5] Tema 4: Biopotenciales: origen, características, adquisición y amplificación. Instrumentación Biomédica. Material Auxiliar de Clase de Instrumentación Biomédica Prof. Dr. R. Navas - Dpto. de Electrónica.
- [6] Willem Einthoven and the Birth of Clinical ECG a Hundred Years Ago. Cardiac Electrophysiology Review. January. 2003, Vol 7, Is 1, pp 99–104.
- [7] Normal J. “Jeff” Holter (1914-1983). Cardiology Journal. 2009, Vol. 16, No. 4, pp. 386–387.
- [8] Protocolo de Enfermería de Holter Implantable. Complejo Hospitalario Universitario de Albacete. Unidad de Arritmias. 2012.
- [9] Toda la información técnica acerca de eHealth se consulta en la web de su principal distribuidor, Cooking-Hacks <https://www.cooking-hacks.com>.
- [10] <https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical> (Última visita en mayo de 2019)
- [11] Prometec. Bus SPI: Arduino y el Serial Peripheral Interface. <https://www.prometec.net/bus-spi/> (visto por última vez en mayo de 2019)
- [12] Toshiba. Flash Memory Products. Datos Técnicos. <https://www.toshiba-memory.com/es/products/toshiba-microsd-cards-high-speed-m203/> (Visto por última vez en mayo de 2019)

[13] Leído en el blog específico de tecnología llamado Geek Factory; en la dirección <https://www.geekfactory.mx/tutoriales/bluetooth-hc-05-y-hc-06-tutorial-de-configuracion/#> (visto por última vez en mayo de 2019).

[14] Blog educativo sobre Arduino llamado aprendiendo Arduino. <https://aprendiendoarduino.wordpress.com/tag/hc-05/> (visto por última vez en mayo de 2019).

[15] source.android.com (visto por última vez en mayo de 2019)

[16] Página oficial de Mathworks. Desde aquí se puede obtener información acerca de las funcionalidades de Scrollplot, e incluso, descargar el propio código. https://es.mathworks.com/matlabcentral/fileexchange/14984-scrollplot-scrollable-x-y-axes?s_tid=FX_rc1_behav (visto por última vez en mayo de 2019).

[17] Foro oficial de Mathworks, respuesta del equipo técnico de la compañía. <https://es.mathworks.com/matlabcentral/answers/95511-in-matlab-is-there-a-way-to-set-the-grid-at-a-spacing-different-from-the-ticks-on-the-axes> (visitado por última vez en mayo de 2019)

[18] Información presente tras el papel milimetrado de ECG del SAS.

[19] PIC solution: <http://www.picsolution.com/es/consejos/para-ayudarte-a-elegir/tension-controlada-las-24-horas.html> (Visitada por última vez en mayo de 2019)

BIBLIOGRAFÍA GENERAL

Arduino y eHealth

- Arduino: curso práctico de formación. Torrente Artero, Óscar. Madrid: RC Libros, 2013.
- <https://aprendiendoarduino.wordpress.com> (Última visualización el 9 de mayo de 2019)
- <https://openwebinars.net/blog/tutorial-arduino-entradas-2-botones/> (Última visualización el 9 de mayo de 2019)
- Cooking Hacks Forum by Libelium. E-Health shield & Arduino 1.5.7 December 2014. <https://www.cooking-hacks.com/forum/viewtopic.php?f=20&t=6023> (Visitada por última vez en mayo de 2019).
- <https://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical> (Última visualización el 9 de mayo de 2019)

Comunicaciones

- <https://www.prometec.net/bt-hc06/> (Última visualización el 9 de mayo de 2019)
- Aplicaciones de Controladores Bluetooth en Robótica. Capítulo 4: bluetooth. Documento encontrado en la Biblioteca de Ingeniería de la Universidad de Sevilla.

Ingeniería y maquinaria médica

- BI9800TL+ Holter recorder operation manual. Biomedical Instrument Co., Ltd.
- Medical Instrumentation. Application and Design. John G. Webster, Ed. 2010

Medicina

- Dar sentido al ECG (I), Manual Práctico. Andrew R. Hounhton y David Gray. Tercera Edición, Ed. J&C Ediciones Médicas, 2009
- Netter, Medicina Interna. Thomas Bottcher, Stephanie Engelhardt y Martin Kortenhaus. Primera Edición, Ed Masson, 2004.
- Revista Española de Cardiología. Guías de práctica clínica de Cardiología en la monitorización ambulatoria del electrocardiograma y presión arterial. Vol. 53 Núm. 01, pp 91-109.