

TRABAJO FIN DE GRADO

Ingeniería Electrónica Industrial y Automática

Análisis, diseño y desarrollo de “shields” de Arduino para procesamiento de señales fisiológicas

Alumno: Arnau Pérez Talens

Tutor: Enrique Berjano Zanon

RESUMEN

Este proyecto surge a partir de la idea de Arduino, construir aparatos electrónicos hardware abierto, de tal forma que cualquier persona pueda aprender el funcionamiento. Este proyecto se centra en los sensores biomédicos, analizando algunos ejemplos para entender su funcionamiento y diseñar un sensor en particular, el electrocardiógrafo.

Para el diseño del electrocardiógrafo se habla del funcionamiento del corazón y qué significa la señal del electrocardiograma.

El diseño está implementado para utilizar Arduino Uno y Matlab, pero la estructura modular que sigue ayuda a que se pueda extender a otros programas y placas de Arduino con pequeñas modificaciones en el programa y diseño.

PALABRAS CLAVE:

ECG, Arduino, Shield, Electrocardiograma, Electrocardiógrafo, microcontrolador, sensor, señal biomédica.

RESUMEN

Este projecte sorgís a partir de la idea de Arduino, construir aparells electrònics hardware obert, de tal forma que qualsevol persona pot aprendre'n el funcionament. Este projecte es centra en els sensors biomèdics, analitzant alguns exemples per entendre el seu funcionament i dissenyar un sensor particular, el electrocardiògraf.

Per al disseny de l'electrocardiògraf es parla del funcionament del cor i què significa el senyal de l'electrocardiograma.

El disseny esta implementat per a utilitzar Arduino Uno i Matlab, però la estructura modular que segueix ajuda a que siga apte per a altres programes i plaques Arduino amb xicotetes modificacions en el programa i el disseny.

PARAULES CLAU:

ECG, Arduino, shield, Electrocardiograma, Electrocardiògraf, microcontrolador, sensor, senyal biomèdica.

ABSTRACT

This project came from the Arduino's idea, building open-hardware electronic devices, in order that any person can learn the device working. This project focuses on the biomedical sensors, analyzing some examples and understanding how they work and make the electrocardiograph design.

With the aim of design the electrocardiograph talks about the heart function and what the electrocardiogram signal means.

The design is implemented by Arduino Uno and Matlab but his modular structure allows an easy way to change the program and the Arduino's board with small variations in the program and design.

KEYWORDS:

ECG, Arduino, Shield, electrocardiograph, electrocardiogram, microcontroller, sensor, biomedical signal.

Documento I. Memoria

1.	Introducción	1
2.	Contextualización	1
2.1.	Señales biomédicas.....	1
2.2.	Funcionamiento del corazón	2
2.2.1.	ECG y sus derivadas.....	4
2.3.	Micro controladores	5
2.3.1.	Arduino y Shields.....	6
3.	Antecedentes	7
3.1.	Arduino para usos biomédicos	7
3.1.1.	E-health shield.....	7
4.	Análisis.....	25
4.1.	Alimentación eléctrica	25
4.2.	Seguridad eléctrica	26
4.3.	Captación de la señal	28
4.4.	Procesamiento de la señal	28
4.5.	Conversión analógico digital	33
4.6.	Comunicación	35
4.6.	Post-procesamiento digital y monitorización	36
5.	Diseño.....	37
5.1.	Front-end	37
5.2.	Multiplexado.....	38
5.3.	Amplificación de la señal	40
5.4.	Filtro analógico	41
5.4.1.	Alternativa.....	46
5.5.	Adaptación de la señal para Arduino.....	48
5.6.	Diseño PCB	50
5.6.	Procesamiento digital	51
5.6.1.	Conversión analógico digital	51
5.6.2.	Comunicación serie	51
5.6.3.	Filtro digital	52
5.6.4.	Monitorización de la señal	55
6.	Resultados	56
6.1.	Prueba del prototipo	56
6.2.	Montaje del circuito impreso.....	57
6.3.	Pruebas y resultados.....	59
6.4.	Manual de usuario	64
7.	Conclusiones	65
7.1.	Conclusiones	65
7.2.	Líneas de desarrollo futuro	65

Documento II. Presupuesto del proyecto

1.	Características del proyecto	67
2.	Costes	68
2.1.	Coste de los componentes y materiales	68
2.2.	Coste de horas de trabajo.....	69
2.3.	Coste del proceso de desarrollo	69
2.4.	Presupuesto total	70

Documento III. Pliego de condiciones

1.	Definición y alcance del pliego	71
2.	Condiciones generales.....	71
3.	Normativa de aplicación.....	71
4.	Condiciones particulares de los componentes del sistema	72
4.1.	Resistencias	72
4.2.	Condensadores	72
4.3.	Amplificadores operacionales.....	72
4.4.	Amplificador de instrumentación	73
4.5.	Conectores macho para PCB.....	73
4.6.	Bloque Terminal para PCB	73
4.7.	Multiplexor	73

Documento IV. Planos

1.	Plano cara superior de las pistas de cobre	74
2.	Plano cara inferior de las pistas de cobre.....	76
3.	Plano de taladrado	77
4.	Plano de soldadura de los path	78
5.	Plano de soldadura de componentes de montaje superficial	79
6.	Plano de soldadura de componentes de orificio pasante	80
7.	Plano componentes.....	81

ANEXO I: Código fuente

Bibliografía

Documento I. MEMORIA

1. Introducción

El electrocardiógrafo es un instrumento que capta las señales eléctricas del corazón mediante 4 electrodos colocados en las extremidades y 6 electrodos en las posiciones precordiales. El electrocardiógrafo capta y amplifica esta señal que pasa a llamarse electrocardiograma (ECG). La señal del electrocardiograma es utilizada para el diagnóstico de enfermedades del corazón, problemas en el funcionamiento de éste, antiguos infartos y cualquier otra patología. La principal ventaja del ECG es que es un método no invasivo que se puede utilizar de forma sencilla y eficaz.

Los microcontroladores son integrados programables muy útiles por su versatilidad, por el poco espacio que utilizan y su baja potencia, son un integrado sencillo de utilizar que ahorra tener que diseñar circuitos lógicos. Los microcontroladores son utilizados prácticamente en todos los aparatos electrónicos actuales, desde los más sencillos a los más sofisticados, dependiendo de la necesidad requerida. Los microcontroladores también son utilizados en el sector biomédico para el procesamiento de señales biomédicas, entre otros usos.

El objetivo de este proyecto es el estudio de distintos sensores biomédicos implementados para el microcontrolador Arduino y el diseño de un electrocardiograma para Arduino con un hardware y software abiertos para que toda persona interesada en aprender o mejorar el funcionamiento pueda acceder a él.

2. Contextualización

2.1. Señales biomédicas

Los organismos vivos están compuestos por distintos sistemas que están formados por órganos y tejidos, estos sistemas funcionan y trabajan juntos para hacer funcionar los organismos vivos. Cada sistema está formado por distintos subsistemas y cada uno tiene sus funciones fisiológicas. Estas funciones fisiológicas son complejas, incluyendo estimulaciones y controles hormonales o nerviosos, entradas y salidas en la forma física de la materia, neurotransmisores o información y acciones que pueden ser mecánicas, eléctricas o bioquímicas. Todos estos procesos fisiológicos se manifiestan a través de señales que reflejan su actividad y naturaleza. Estas señales pueden ser de muchos tipos incluyendo bioquímicas en forma de hormonas o neurotransmisores, eléctricos en forma de potencial o corriente y físicos en forma de presión o temperatura.

Las enfermedades o defectos en el sistema biológico pueden causar alteraciones en los procesos fisiológicos normales que afectan a la salud y al bienestar del sistema. Las patologías pueden modificar las señales fisiológicas y por tanto compararlas con las señales normales. Es una forma sencilla de detectar patologías, del mismo modo que, por ejemplo, una infección causa que la temperatura del cuerpo se eleve. Para detectar estas variaciones o valores se necesitan diferentes tipos de instrumentos con sensores distintos para detectar las funciones fisiológicas del cuerpo, termómetro para la temperatura, esfigmomanómetro para la presión arterial, electrocardiógrafo para la actividad del corazón, electromiografía para la actividad muscular y demás instrumentos para los distintas funciones fisiológicas y cada instrumento tiene sus métodos para procesar la señal.

2.2. Funcionamiento del corazón

El corazón es el órgano encargado de bombear la sangre que circula por el cuerpo, manteniendo el ritmo y presión adecuada en cada momento. Este órgano se compone de cuatro cámaras (dos superiores o aurículas y dos inferiores o ventrículos), comunicadas dos a dos por las válvulas aurículo-ventriculares (válvula tricúspide o derecha y válvula mitral o izquierda). Los ventrículos están separados de las arterias aorta y pulmonar por las válvulas semilunares (aórtica y pulmonar). La sangre fluye ininterrumpidamente hacia las aurículas a través de las venas cavas y pulmonares, y sale del corazón a través de la arteria aorta y pulmonar, estableciéndose así un doble circuito llamado sistema cardiovascular.

Existen dos fases principales en el ciclo del corazón: diástole y sístole. La diástole es la fase donde el corazón reposa o está inactivo y se llena. La sístole es la fase donde el corazón se contrae y bombea la sangre. Al principio de la diástole, las válvulas AV se relajan dejando paso a la sangre por la diferencia de presión que hay entre ellos llenando el ventrículo, éste no se llena del todo, se termina de llenar con la sangre que entra directamente de las venas (diástasis). Por último las aurículas se contraen para que el resto de sangre pase a los ventrículos. La siguiente fase es la sístole, la sístole empieza con una contracción ventricular isométrica, es decir sin variación del volumen, a la vez que las válvulas A-V se cierran. Con las válvulas cerradas y la contracción ventricular la presión aumenta hasta tener presión suficiente como para abrir las válvulas de la aórtica y la pulmonar, una vez abiertas estas válvulas el ventrículo empieza a vaciarse, la sangre se mantiene saliendo por la contracción ventricular (protodiástole), por último la fibras ventriculares se relajan, por eso la sangre vuelve al ventrículo ya que la presión en el ventrículo es inferior a la de la arteria en ese momento las válvulas semilunares se cierran impidiendo el retorno de la sangre finalizando el ciclo cardiaco.

Diástole	<ul style="list-style-type: none"> Diástole Diástasis Sístole auricular
Sístole	<ul style="list-style-type: none"> Contracción isométrica Vaciamiento Protodiástole Relajación isométrica

Para generar las distintas actividades del ciclo cardíaco es necesario que el corazón esté formado por fibras musculares y además que a estas fibras les llegue el potencial de acción que cause las contracciones de las fibras musculares. Las fibras musculares que componen el corazón están formadas por miocardiocitos, el ventrículo está formado por células miocárdicas y las aurículas están formadas por el tejido muscular llamado miocardio auricular.

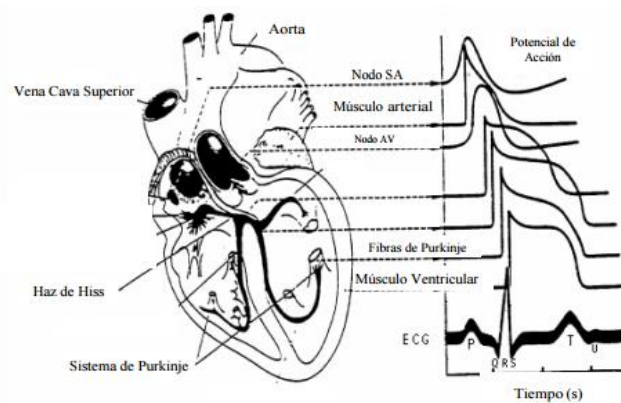


Figura 1. Actividad eléctrica de las distintas partes del corazón a lo largo del tiempo.

Para que llegue el potencial de acción a estos tejidos se necesitan otro tipo de células especializadas que transporten el potencial de acción. Este impulso se origina en un tejido distinto llamado nodo sinoauricular o nodo sinusal (nodo SA) (fig. 2). Este nodo se sitúa en la pared superior de la aurícula derecha. El nodo SA genera su propio potencial funcionando como marcapasos natural del corazón. El nodo SA se ajusta a la frecuencia que se necesite en cada momento pudiendo variar la frecuencia de latidos del corazón. Este potencial generado en el nodo SA se propaga a través de unas células especializadas que propagan este potencial a las aurículas. Este potencial también se propaga a los ventrículos a través de las Vías Internodales hasta llegar al nodo aurículo-ventricular (nodo A-V), donde hay un retraso en el potencial para provocar un desfase de tiempo entre las síntoles auricular y ventricular para que la contracción de los ventrículos no comience hasta que las aurículas hayan sido totalmente vaciadas. Este potencial que pasa por el nodo A-V se conecta con el Haz de Hiss, está formado por un tejido de células de conducción especializadas, el Haz de Hiss se separa en dos ramas que se distribuyen por las paredes del septum. De estas dos ramas aparecen las fibras de Purkinje, estas células especializadas su función es hacer llegar el potencial a todas las fibras musculares del miocardio simultáneamente que provocan la contracción del corazón.

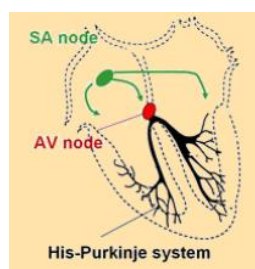


Figura 2. Tejidos del corazón encargados de generar el potencial y las células que transmiten este potencial.

2.2.1. ECG y sus derivadas

Estos potenciales de acción explicados en el apartado anterior se pueden captar mediante electrodos en la superficie. La señal de ECG es el único potencial extracelular en que no hay una gran diferencia entre individuos, tiene un carácter cuasi-periódico que proporciona una gran información acerca del corazón que puede servir para detectar posibles patologías del corazón o analizar el funcionamiento.

Dependiendo de la posición de los electrodos se puede captar distintas posiciones del vector electrográfico y por tanto poder observar distintas actividades del corazón explicadas en el apartado 2.1. Existen infinitas posiciones para colocar los electrodos a lo largo del cuerpo sin embargo sólo se utilizan un número establecido de estas.

Se pueden dividir en tres grupos:

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

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

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

De modo que cada derivación se puede definir como:

- Derivación I: Es el potencial diferencial entre el electrodo del brazo derecho (negativo) y el brazo izquierdo (positivo) (fig.3).
- Derivación II: Es el potencial diferencial entre el electrodo del brazo derecho (negativo) y el pie izquierdo (positivo) (fig. 3).
- Derivación III: Es el potencial diferencial entre el electrodo del brazo izquierdo (negativo) y el pie izquierdo (positivo) (fig. 3).

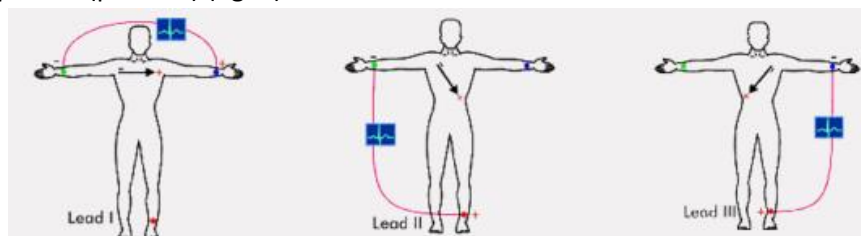


Figura 3. Esquema para la obtención de las tres derivadas principales.

- Derivación aVR: Es el potencial diferencial registrado entre el electrodo del brazo derecho (positivo) y un electrodo posicionado en el “infinito”, como no se puede situar en el infinito, se coge la tensión promediada del pie izquierdo y del brazo izquierdo (fig. 4).

- Derivación aVL: Es el potencial diferencial registrado entre el electrodo del brazo izquierdo (positivo) y un electrodo posicionado en el “infinito”, como no se puede situar en el infinito, se coge la tensión promediada del pie izquierdo y del brazo derecho (fig. 4).
- Derivación aVF: Es el potencial diferencial registrado entre el electrodo del pie izquierdo (positivo) y un electrodo posicionado en el “infinito”, como no se puede situar en el infinito, se coge la tensión promediada del brazo derecho y del brazo izquierdo (fig. 4).

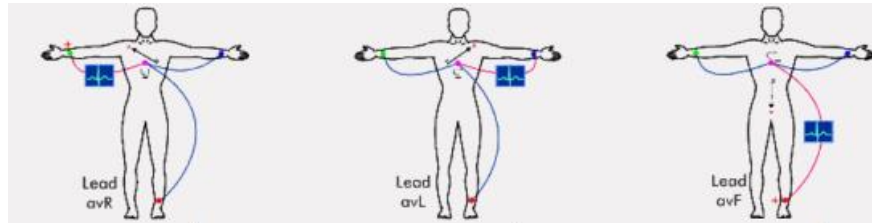


Figura 4. Esquema para la obtención de las tres derivadas aumentadas.

Las derivaciones precordiales son el potencial diferencial entre el electrodo situado en el tórax (positivo) también llamados V1, V2, V3, V4, V5 y V6 en sus posiciones normalizadas (fig.5) y el terminal negativo estaría en el infinito. En realidad se coge la tensión promediada del pie izquierdo, del brazo derecho y del brazo izquierdo, esté punto se conoce como CTW (Central terminal de Wilson). El CTW sirve de referencia independiente para las derivadas precordiales (V1, V2, V3, V4, V5 y V6).

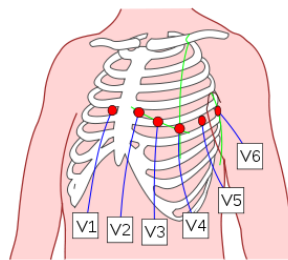


Figura 5. Colocación de los electrodos para las 6 derivaciones precordiales.

2.3. Microcontroladores

Los microcontroladores son circuitos integrados programables que contienen un sistema microprocesador completo es decir una unidad central procesamiento, memoria y periféricos de entradas/salidas. A diferencia del microprocesador de un ordenador que sólo incluye la función de unidad central de procesamiento y por tanto es más fácil convertirlo en una computadora con un menor número de periféricos.

El microcontrolador está diseñado para simplificar sistemas electrónicos complejos, reducir el coste, disminuir el coste económico y consumo energético. Los microcontroladores pueden ser de muchos tipos que se adaptan a cada situación. Sus principales características pueden venir definidas por el número de bits del procesador, el tipo de memoria que utiliza o la arquitectura. En la actualidad prácticamente todos los aparatos electrónicos utilizan microcontroladores pero dependiendo de las necesidades utilizan microcontroladores de distintas capacidades por ejemplo una lavadora utiliza un procesador de 4 u 8 bits mientras que un DVD utiliza un procesador de 32 ó 64 bits.

Un microcontrolador incluye un reloj interno y una memoria de acceso aleatorio (RAM) y/o memoria de sólo lectura (ROM, EPROM, EEPROM), memoria flash para almacenar los programas y datos. También incluyen otras funciones como distintos dispositivos de entrada/salida, temporizadores, UART y buses de interfaz especializados como I2C y CAN, convertidores analógico digital y convertidores digital analógico.

Existen distintas familias de microcontroladores dependiendo de la empresa que los fabrica y el número de bits. En general todos tienen especificaciones parecidas, pero las características como capacidad de cálculo, memoria, velocidad del reloj interno y otras pueden variar dependiendo del fabricante.

Un microcontrolador bastante utilizado por su sencillez y su open hardware es Arduino (fig. 6). Arduino es el nombre que recibe la placa, por ejemplo, el micro controlador de la placa Arduino Uno es el ATmega328p de 8 bits.

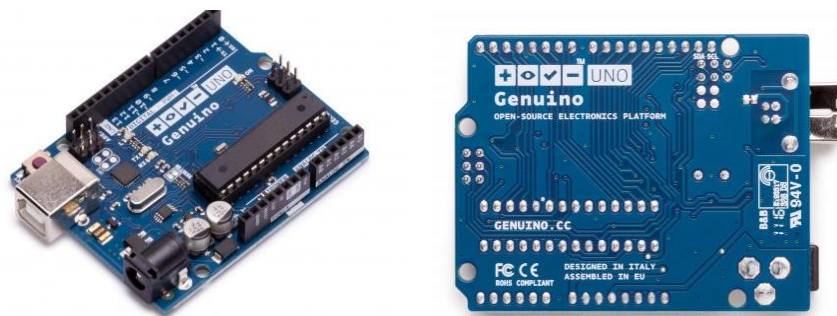


Figura 6. Placa Arduino Uno con su microcontrolador ATmega328p.

2.3.1. Arduino y Shields

Los Shields son placas de circuito impreso que se pueden conectar a Arduino directamente ampliando sus capacidades, sin tener la necesidad de montar circuitos externos, existen infinidad de clases de Shields (shields para control de motores, de comunicación inalámbrica...). Los Shields existen para otros modelos de microcontroladores pero su función depende del Shield, simplificar el diseño y facilitar el uso de los microcontroladores.

En este proyecto se aborda el diseño de ECG para Shield de Arduino, de esta forma sólo conectando el Shield se puede conseguir un ECG funcional. Además se puede acceder a los esquemas electrónicos y entender mejor el funcionamiento de un ECG.

3. Antecedentes

3.1.Arduino para usos biomédicos

Arduino, entre otras utilidades, puede servir para sensores biomédicos. Debido a su capacidad limitada los sensores biomédicos no son de uso clínico, sino de uso académico para entender el funcionamiento de las señales fisiológicas y el procesamiento de estas señales.

3.1.1. E-health shield

Un ejemplo el Shield E-Health (Cooking hacks Libelium, Zaragoza, España) es un Shield para Arduino y Raspberry Pi que adapta la señal que provienen de distintos sensores biomédicos como sensor de posición, glucómetro, sensor de temperatura corporal, esfigmomanómetro, pulsioxímetro, sensor de flujo de aire, sensor de respuesta galvánica de la piel (GSR), electrocardiograma (ECG) y electromiograma (EMG).

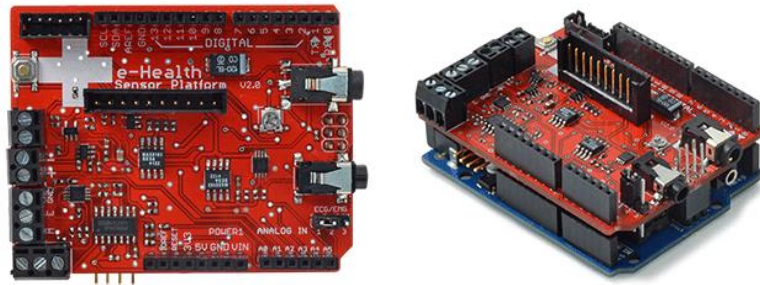


Figura 7. Shield E-Health de Cooking hacks.

3.1.1.1. Electrocardiógrafo

Uno de los sensores del Shiel E-health es el electrocardiógrafo. Este sensor se encarga de monitorizar el ritmo cardiaco del corazón, ya que solo contiene una de las 12 derivadas y por tanto su principal función es observar el ritmo cardiaco (pulsaciones por minuto).

Para entender el funcionamiento del canal ECG la primera parte debe ser un estudio teórico del esquema eléctrico del ECG.

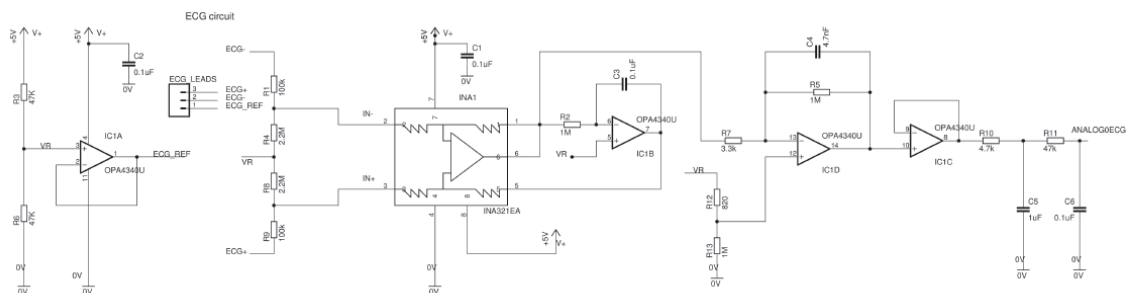


Figura 8. Esquema completo del ECG.

El circuito del ECG se puede dividir en etapas o bloques para analizarlo por partes.

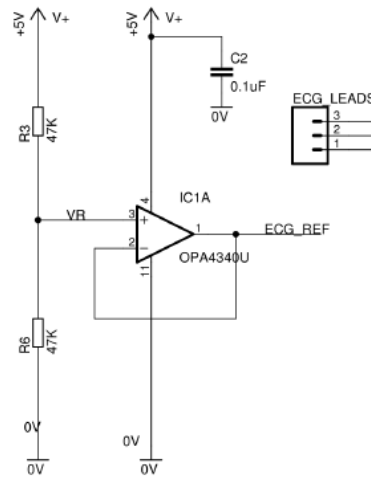


Figura 9. Esquema eléctrico del primer bloque del ECG.

La primera etapa es el front-end (fig. 9) que está compuesta por un divisor de tensión y un buffer o seguidor de tensión, la función principal de esta etapa es conseguir una tensión de referencia (V_r) para añadir un offset a la señal de ECG que contiene partes negativas y Arduino sólo trabaja con tensiones entre 0-5 V.

$$V_r = 5 * \frac{R6}{R6 + R3} = 2.5 V \quad (1)$$

La segunda etapa es un amplificador de instrumentación que amplifica la señal diferencial del ECG (fig. 10). La señal que proviene de los electrodos es una señal diferencial que capta el incremento de potencial entre dos puntos ($V+$ y $V-$) y una amplitud máxima de unos 2 mV. Por tanto esta señal debe ser amplificada y convertida en single-ended respecto una masa común al Shield y Arduino, para que Arduino pueda leerla correctamente.

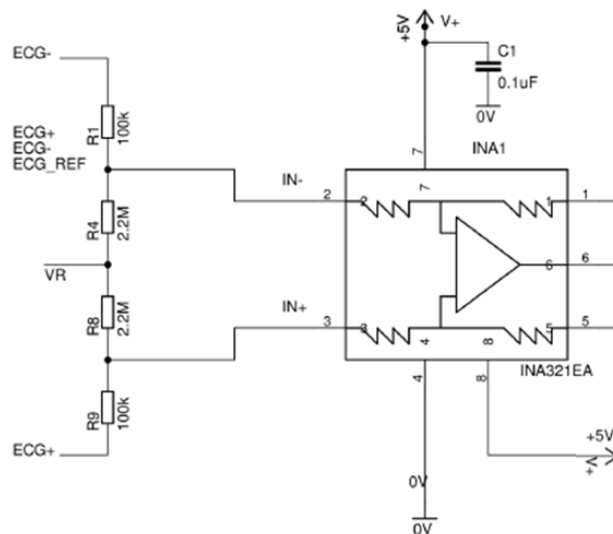


Figura 10. Bloque de amplificador de instrumentación.

Para entender el funcionamiento de esta etapa y cómo funciona el circuito hay que analizar la hoja de datos del amplificador de instrumentación INA321EA de Texas Instruments, Texas, EEUU (fig. 11).

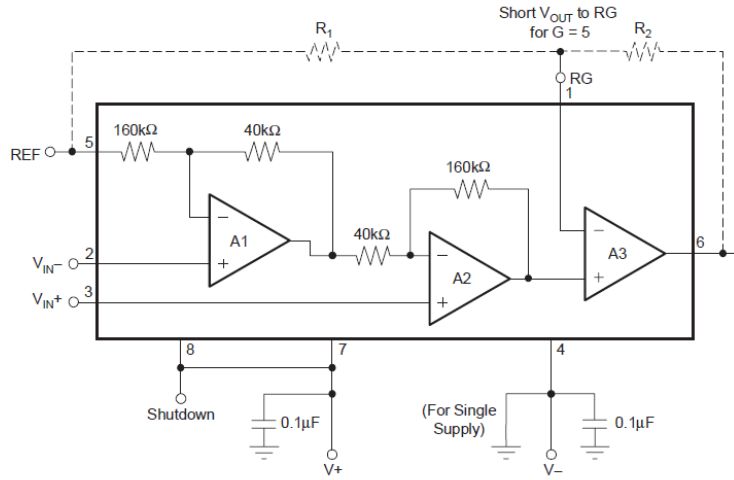


Figura 11. Esquema eléctrico del integrado INA321EA.

A partir de la hoja de datos del integrado se sabe que la ganancia del amplificador de instrumentación viene dada por la siguiente expresión:

$$G = 5 + 5 * \frac{R_2}{R_1} \quad (2)$$

$$V_o = V_{ref} + ((V_{IN+}) - (V_{IN-})) * G \quad (3)$$

Para conocer la ganancia observamos el siguiente bloque y vemos que los pines 1 y 6 del integrado están conectados, es decir $R_2=0$ (ecuación 2) por tanto la ganancia de este bloque es de 5 V/V. La tensión de salida del amplificador de instrumentación viene dada por la ecuación 3.

Se sustituyen los valores y se estudia el esquema completo y se obtiene la función de transferencia de todo el bloque:

$$\frac{V_R - ECG^-}{R_4 + R_1} = I_1 \quad (4)$$

$$\frac{ECG^+ - V_R}{R_8 + R_9} = I_2 \quad (5)$$

$$V_{IN-} = V_R - I_1 * R_4 \rightarrow V_{IN-} = V_R - \frac{V_R - ECG^-}{R_4 + R_1} * R_4 \quad (6)$$

$$V_{IN+} = ECG^+ - I_2 * R_8 \rightarrow V_{IN+} = ECG^+ - \frac{ECG^+ - V_R}{R_8 + R_9} * R_8 \quad (7)$$

$$V_{o1} = V_{ref} + \left(\left(ECG^+ - \frac{ECG^+ - V_R}{R_8 + R_9} * R_8 \right) - \left(V_R - \frac{V_R - ECG^-}{R_4 + R_1} * R_4 \right) \right) * 5 \quad (8)$$

Esta etapa es un integrador (fig. 12) que se conecta de la salida del amplificador de instrumentación a la referencia del mismo. Esto sirve para eliminar las componentes de baja frecuencia que puedan afectar a la línea base del ECG.

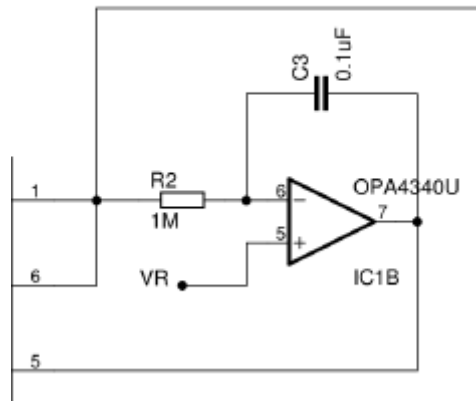


Figura 12. Integrador de la referencia del amplificador de instrumentación.

$$\frac{V_I - V_R}{R_2} = \frac{V_R - V_o}{Z_c} \rightarrow \frac{V_R - V_o}{V_I - V_R} = \frac{Z_c}{R_2} = \frac{1}{j\omega R_2 C_3} \quad (9)$$

$$\omega_0 = \frac{1}{R_2 C_3} = 10 \frac{rad}{s} \rightarrow f_0 = 1.59 Hz \quad (10)$$

Conectado con el anterior bloque se ve que $V_{ref} = V_o$

La siguiente etapa es un filtro paso bajo de primer orden activo formado por un amplificador operacional. Con una ganancia de 303.03 V/V (ecuación 11) y frecuencia de corte de 33.86 Hz (ecuación 12).

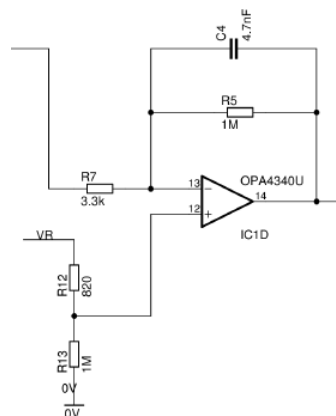


Figura 13. Filtro paso bajo de orden 1 activo.

Para frecuencias bajas se puede considerar el condensador en abierto y por tanto el cálculo de la ganancia es el siguiente:

$$G = -\frac{R_5}{R_7} = \frac{1 * 10^6}{3.3 * 10^3} = 303.03 \frac{V}{V} \quad (11)$$

$$f_c = \frac{1}{2\pi * R_5 C_4} = \frac{1}{2\pi * 1 * 10^6 * 4.7 * 10^{-9}} = 33.86 \text{ Hz} \quad (12)$$

El último bloque está formado por un buffer o seguidor de tensión para evitar problemas de desadaptación por impedancias seguido de dos filtros paso bajo de primer orden.

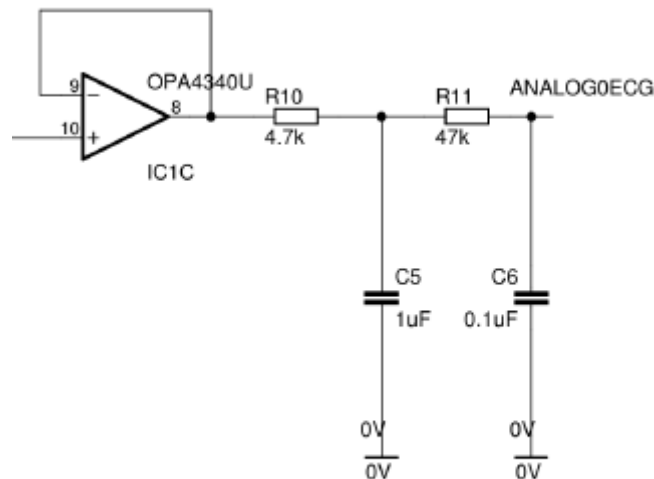


Figura 14. Dos filtros paso bajo de primer orden pasivos montados en serie.

Filtro paso bajo de primer orden pasivo sus frecuencias de corte son:

$$f_{01} = \frac{1}{2\pi R_{10} C_5} = \frac{1}{2\pi * 4.7k * 1\mu} = 33.86 \text{ Hz} \quad (13)$$

$$f_{02} = \frac{1}{2\pi R_{11} C_6} = \frac{1}{2\pi * 47k * 0.1\mu} = 33.86 \text{ Hz} \quad (14)$$



Figura 15. Diagrama de bloques del ECG.

El diagrama de bloque resultante del ECG a partir de los cálculos teóricos (fig. 15). Una vez obtenido los resultados teóricos se comprueba el funcionamiento en el laboratorio haciendo el diagrama de bode del ECG, para ello se alimenta el Shield a través del Arduino conectado al puerto USB del ordenador, conectado a la entrada del ECG el generador de funciones, el ECG tiene 3 entradas, positivo negativo y neutro, de los análisis teóricos se ha observado que el terminal neutro es una referencia que devuelve el ECG para añadir un offset a la señal y eliminar la parte negativa del ECG, para esta prueba no es necesario, por tanto se conecta el terminal negativo a la masa del generador de funciones y el positivo al positivo y a la salida se conecta el osciloscopio TBS 1052B-EDU de tektronix, para monitorizar el resultado (fig. 16).

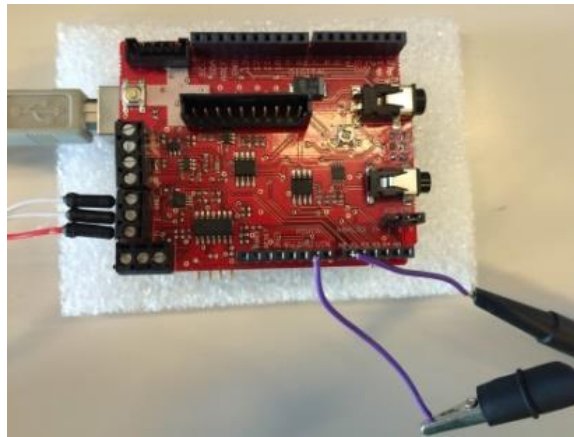


Figura 16. Montaje del Shield para hacer las pruebas en el laboratorio.

La Tabla 1 muestra los valores tomados de tensión de entrada y tensión de salida a distintas frecuencias. Se calculan las ganancias de cada frecuencia. La prueba se hizo dos veces para obtener un resultado más fiable.

Bode1					Bode 2				
Frecuencia	Vi(p)(mV)	Vo(pp)(mV)	G(V/V)	dB	frecuencia	Vi(p)(mV)	Vo(p)(mV)	G(V/V)	G(dB)
1,23	100	497	4,97	13,92713	1,2	100	504	5,04	14,04861
3	100	496	4,96	13,90963	2,8	100	504	5,04	14,04861
10	100	496	4,96	13,90963	10	100	504	5,04	14,04861
20	100	456	4,56	13,1793	20	100	464	4,64	13,33036
30	100	392	3,92	11,86572	30	100	376	3,76	11,50376
40	100	320	3,2	10,103	40	100	312	3,12	9,883092
50	100	272	2,72	8,691378	50	100	256	2,56	8,164799
60	100	224	2,24	7,00496	60	100	208	2,08	6,361267
70	100	184	1,84	5,296356	70	100	179	1,79	5,057061
80	100	160	1,6	4,0824	80	100	152	1,52	3,636872
90	100	136	1,36	2,670778	90	100	128	1,28	2,144199
100	100	108	1,08	0,668475	100	100	116	1,16	1,28916
200	100	36	0,36	-8,87395	200	100	34	0,34	-9,37042
300	100	24	0,24	-12,3958	300	100	20	0,2	-13,9794
400	100	20	0,2	-13,9794	400	100	12	0,12	-18,4164
500	100	14	0,14	-17,0774	500	100	8	0,08	-21,9382
600	100	10	0,1	-20	600	100	5,6	0,056	-25,0362
700	100	5,6	0,056	-25,0362	700	100	4,4	0,044	-27,1309
800	100	6	0,06	-24,437	800	100	4,4	0,044	-27,1309
900	100	4	0,04	-27,9588	900	100	4,2	0,042	-27,535
1000	100	6	0,06	-24,437	1000	100	3,6	0,036	-28,8739

Tabla 1. Datos obtenidos de hacer el diagrama de bode del ECG.

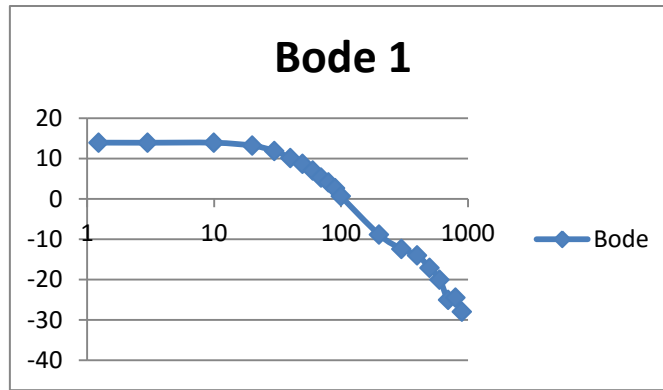


Figura 17. Bode 1 obtenido a partir de la tabla Excel.

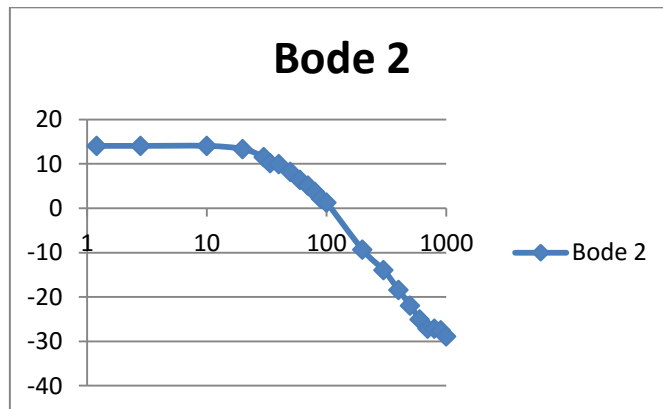


Figura 18. Bode 2 obtenido a partir de la tabla Excel.

A partir de los bodes (fig. 17 y fig. 18) se puede concluir que todo el sistema de procesado de la señal de ECG actúa como un filtro paso bajo con frecuencia de corte superior de unos 40 Hz.

Por último se comprueba el funcionamiento real del ECG utilizando un simulador de ECG (fig. 19). Ya que este aparato no añade ninguna interferencia, debido a que el cuerpo humano es 70% de agua y actúa como una antena, captando cualquier interferencia del ambiente, como la interferencia de la red eléctrica de 50 Hz.



Figura 19. Simulador de ECG para las tres derivadas principales.

Se conectan los electrodos al simulador siguiendo las instrucciones del Shield e-health.

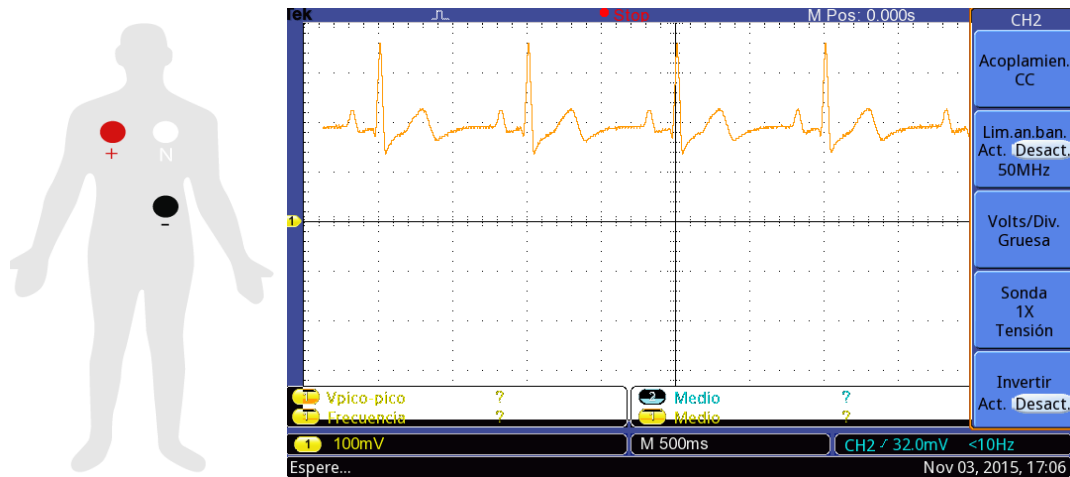


Figura 20. Resultado de conectar el electrocardiógrafo al simulado y monitorizado con el osciloscopio.

3.1.1.1. Electromiógrafo

Al igual que en el corazón, los músculos del resto del cuerpo tienen un comportamiento parecido, por tanto cuando un músculo es ejercitado se puede captar una señal eléctrica en la superficie de la piel que puede ser captada con unos electrodos

Una señal de EMG refleja la actividad eléctrica de los músculos. Tiene una amplitud entre $50 \mu\text{V}$ y 30mV , dependiendo del músculo, y una frecuencia de 7Hz hasta 20kHz dependiendo del tamaño del músculo. Arduino capta la integral de la señal por ello las frecuencias que Arduino captura son de entre 7Hz y 20Hz aproximadamente. Así pues el programa de Arduino y Matlab usado para el ECG puede servir, ya que el rango de frecuencias del EMG es menor que el del ECG y por tanto entra en el rango de frecuencias que puede procesar sin perder información.

Lo primero es realizar un análisis teórico del circuito electrónico del canal de EMG. El circuito electrónico se puede separar en distintos bloques (fig. 21).

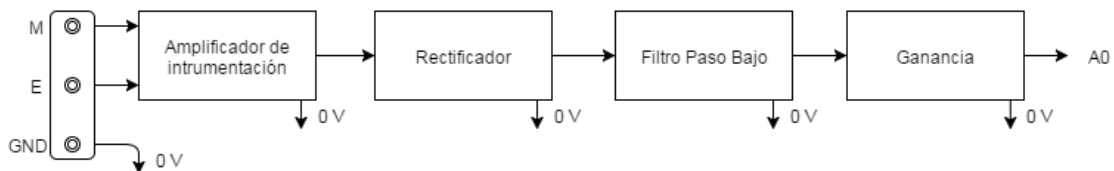


Figura 21. Diagrama de bloques del esquema electrónico.

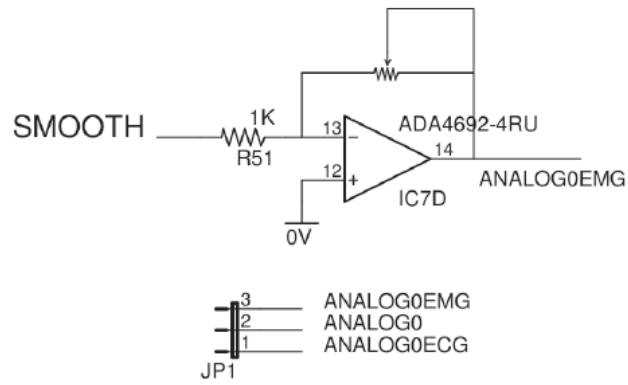


Figura 22. Esquema eléctrico de la etapa final.

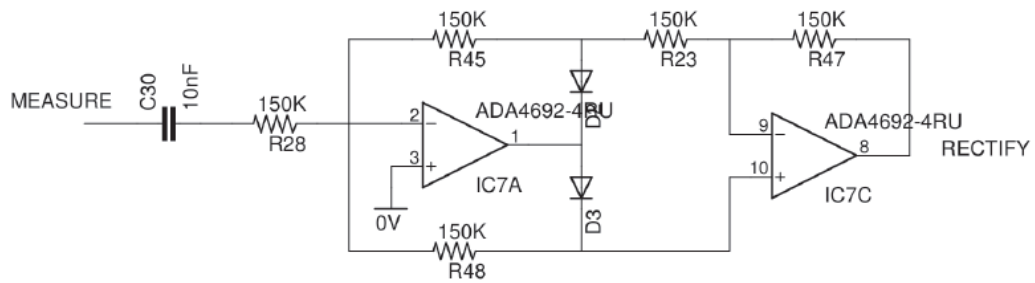


Figura 23. Esquema eléctrico del rectificador de media onda activo.

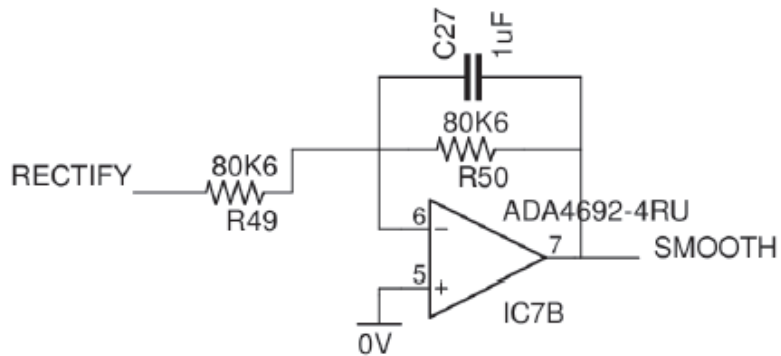


Figura 24. Filtro paso bajo activo de primer orden.

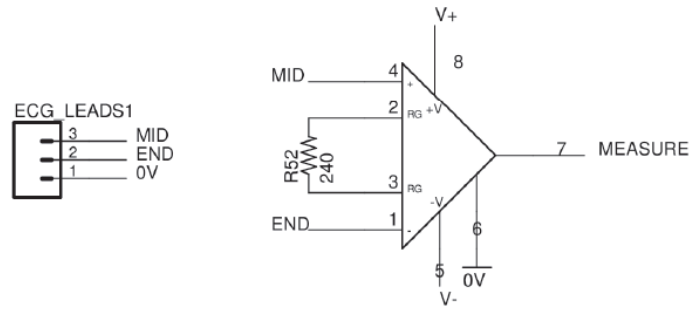


Figura 25. Esquema eléctrico de la etapa de amplificación con un amplificador de instrumentación.

La señal que proviene de los electrodos utilizados para captar el EMG es una señal diferencial, y por ello el primer bloque está basado en un amplificador de instrumentación (fig. 26). La función del bloque es convertir la señal diferencial en una señal single-ended, y amplificándola.

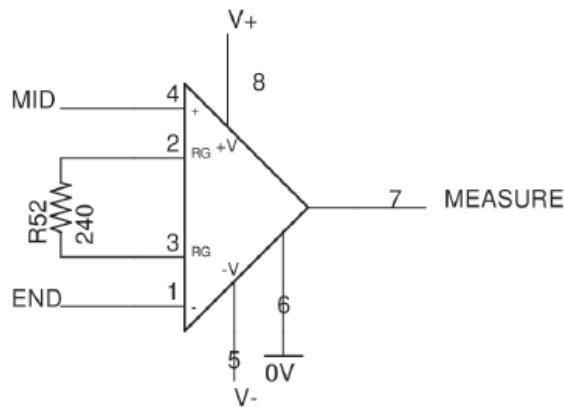


Figura 26. Primera etapa del canal EMG basado en un amplificador de instrumentación AD8211 de Analog Devices, Massachusetts, EEUU.

A partir de las hojas de características del AD8211 se conoce la expresión para calcular la ganancia (ecuación 15).

$$G = 1 + \frac{49.4 \cdot 10^3}{R52} = 1 + \frac{49.4 \cdot 10^3}{240} = 206.83 \frac{V}{V} \quad (15)$$

Para entender el funcionamiento de la etapa de rectificación se debe hacer un análisis teórico de que ocurre en este bloque:

$$\text{Si } V_i < 0 V \quad (16)$$

$$I_1 = I_2 + I_3 \rightarrow D_1 = \text{abierto } D_2 = \text{Conduce} \rightarrow V_2^+ = V_2^- = V_{o1} \quad (17)$$

$$I_1 = \frac{0 - V_i}{Z} \quad I_2 = I_4 = \frac{V_{o2} - V_{o1}}{R} \quad I_3 = \frac{V_{o1} - 0}{R} \quad (18)$$

$$\frac{0 - V_i}{z} = \frac{V_{o2} - V_{o1}}{R} + \frac{V_{o1}}{R} \rightarrow -\frac{V_i}{z} = \frac{V_{o2}}{R} \quad (19)$$

$$\frac{V_{o2}}{V_i} = -\frac{R}{Z} \quad \text{Cuando } V_i < 0 V \quad (20)$$

$$\text{Si } V_i > 0 V \quad (21)$$

$$I_1 = I_2 + I_3 \rightarrow D_1 = \text{Conduce } D_2 = \text{Abierto} \quad (22)$$

$$\frac{V_i - 0}{z} = \frac{0 - V_{o1}}{R} \rightarrow \frac{V_{o1}}{V_i} = -\frac{R}{Z} \quad (23)$$

$$\frac{V_{o1} - 0}{R} = \frac{0 - V_{o2}}{R} \rightarrow \frac{V_{o2}}{V_{o1}} = -1 \quad (24)$$

$$V_{o1} = -V_i * \frac{R}{Z} \quad (25)$$

$$\frac{V_{o2}}{V_i} = \frac{R}{Z} \quad \text{Cuando } V_i > 0 V \quad (26)$$

Por tanto la salida siempre es positiva, es un circuito rectificador de media onda.

La figura 27 muestra el filtro paso bajo de primer orden activo destinado a suavizar la señal proporcionada por el rectificador.

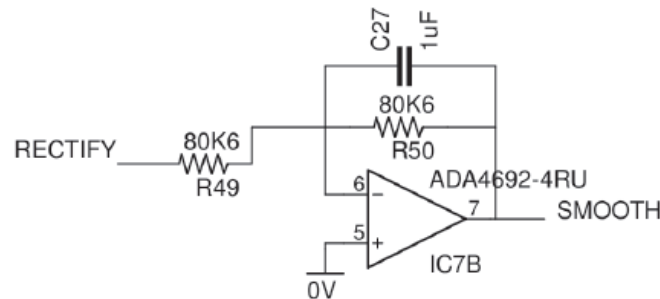


Figura 27. Filtro paso bajo de primer orden activo utilizado en el canal EMG.

Un filtro paso bajo de primer orden activo del que se conoce su función de transferencia y a partir de ella se puede calcular que frecuencia de corte y ganancia tiene esta etapa.

$$\frac{V_o}{V_i} = -\frac{R_{50}}{R_{49} R_{50} C_{27} \omega} \rightarrow f_c = \frac{1}{2\pi RC} = \frac{1}{80,6 * 10^3 * 10^{-6}} = 1.974 Hz \quad (27)$$

$$G = -\frac{80 k\Omega}{80 k\Omega} = -1 \frac{V}{V} \quad (28)$$

El integrado ADA4692-4RU de Analog Devices, Massachusetts, EEUU, está compuesto por 4 amplificadores operacionales ADA4692 que son operacionales de baja potencia, bajo ruido y salida rail-to-rail. La alimentación de los 4 operacionales se hace a través de los mismos pines conectados a 5 V de alimentación que proporciona el Shield.

La figura 28 muestra el circuito de la etapa final. Es un amplificador inversor no diferencial basado en un amplificador operacional ADA4692 de Analog Device, Massachusetts, EEUU.

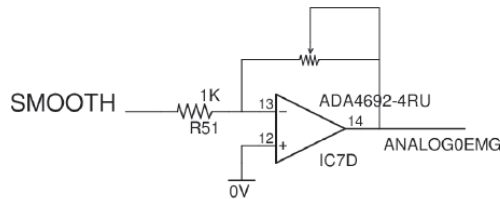


Figura 28. Amplificador inversor de tensión variable utilizado en la etapa final del canal EMG.

$$\frac{V_o}{V_i} = \frac{-POT}{R_{51}} \quad (29)$$

Por tanto la ganancia de este bloque es variable dependiendo del valor que se ajuste el potenciómetro que no se conoce sus valores máximos ni mínimos.

Una vez se tiene el funcionamiento se comprueba el funcionamiento real. Para ello se colocan los electrodos en el brazo, situando el positivo y el negativo en los extremos de un músculo y el neutro en un lugar opuesto a ambos electrodos para tenerlo de referencia (fig. 29).

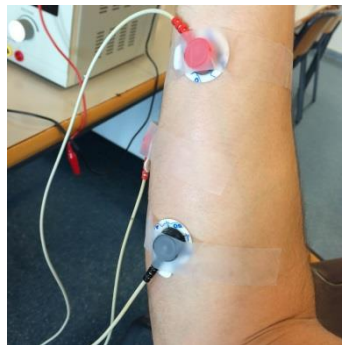


Figura 29. Electrodo situado en el musculo del antebrazo, positivo rojo y negativo negro, referencia electrodo situado en la parte de atrás del antebrazo.

Con los electrodos colocados en el brazo se conecta el shield al Arduino para alimentarlo, pero con la salida del shield conectada al osciloscopio para captar la señal.

Se han hecho distintas pruebas, haciendo fuerza permanente (fig. 31), moviendo la mano (fig. 31) y en estático o reposo (fig. 32).

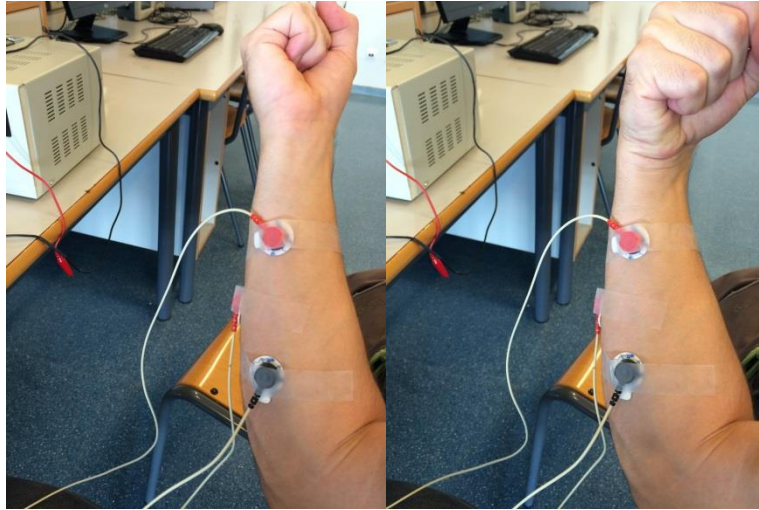


Figura 30. Movimiento del brazo para hacer la prueba.

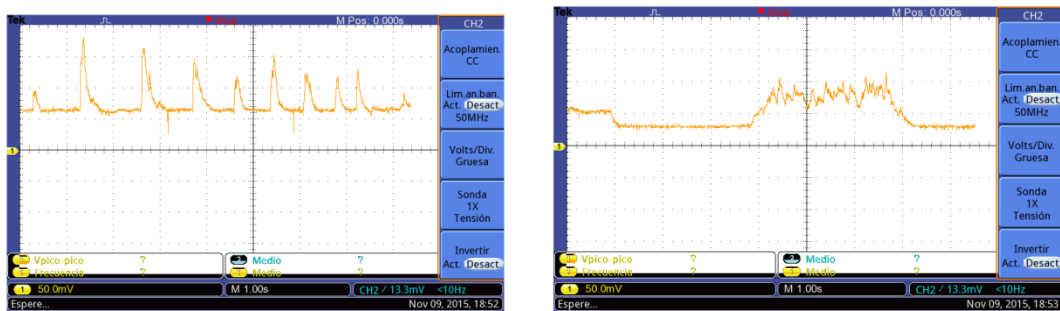


Figura 31. Izquierda movimiento intermitente de la mano, derecha fuerza de forma permanente.

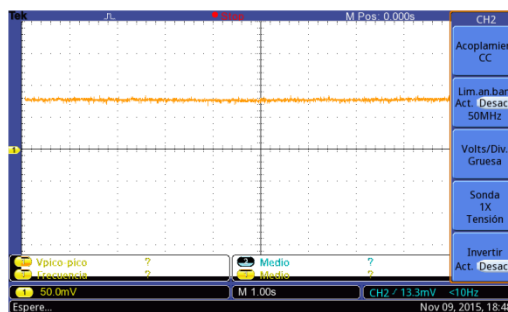


Figura 32. Brazo en reposo.

3.1.1.2. Pulsioxímetro

Este sensor consiste en una pinza que se sitúa en la punta del dedo corazón. Sirve para detectar el ritmo cardiaco y el porcentaje de saturación de oxígeno de la hemoglobina. El pulsioxímetro en la parte inferior de la pinza tiene un led que emite luz roja (longitud de onda de 660 nm) y luz infrarroja (longitud de onda de 920 nm), ya que lo que se intenta detectar es la oxihemoglobina y la desoxihemoglobina dos moléculas distintas que tienen una absorbanca de la luz distinta, la oxihemoglobina absorbe la luz del espectro infrarrojo y la desoxihemoglobina absorbe la luz de espectro rojo y al otro extremo de la pinza se sitúa un

foto receptor que capta la luz que ha pasado por el dedo. Con esta lectura el sensor tiene un algoritmo que calcula la saturación de oxígeno en sangre a partir de cuánta luz roja e infrarroja se ha absorbido, además con este método también se puede detectar el pulso del corazón ya que dependiendo de la presión el nivel de absorbancia se ve afectado.

En este Shield el pulsioxímetro es externo y el Shield sólo se encarga de procesar una señal digital que llega del pulsioxímetro (fig. 33).



Figura 33. Izquierda pinza donde se sitúa el sensor, derecha conexión paralelo con el Shield.

Para el procesamiento de los datos se utiliza el código y librerías que ofrecen los de la empresa Libelium ya que sin conocer el funcionamiento exacto del pulsioxímetro no se pueden sincronizar correctamente los dos dispositivos y daría lugar a una mala comunicación.

El programa que se instala en Arduino para hacer funcionar el pulsioxímetro y visualizar los datos es el siguiente

<code>#include <PinChangeInt.h></code>	Carga las bibliotecas de e-health
<code>#include <PinChangeIntConfig.h></code>	
<code>#include <eHealth.h></code>	
<code>#include <eHealthDisplay.h></code>	
<code>int cont = 0;</code>	Inicio de contador
<code>void setup() {</code>	Inicio de la configuración de Arduino
<code>Serial.begin(115200);</code>	Velocidad de la comunicación serie
<code>eHealth.initPulsioximeter();</code>	Función que inicia las variables necesarias para el pulsioxímetro
<code>PCIntPort::attachInterrupt(6, readPulsioximeter, RISING);}</code>	Configura un pin como interrupción para poder ejecutar la función de lectura
<code>void loop() {</code>	Inicio del bucle principal
<code>Serial.print(eHealth.getBPM());</code>	Lectura y envío por el puerto serie los datos de pulsaciones por minuto
<code>Serial.print(eHealth.getOxygenSaturation());</code>	Lectura y envío por el puerto serie los datos de saturación de oxígeno en sangre
<code>Serial.print("\n");</code>	Salto de línea
<code>delay(500);}</code>	Retraso que viene definido por el tiempo de muestreo en milisegundos
<code>void readPulsioximeter(){</code>	Función para leer el sensor
<code>cont ++;</code>	Contador
<code>if (cont == 50)</code>	Se toman sólo 50 muestras para reducir el retraso en la toma de datos
<code>eHealth.readPulsioximeter();</code>	Función que lee los datos que envía el pulsioxímetro
<code>cont = 0;}}</code>	

Para monitorizar aparte de poder observar los resultados en el dispositivo se puede utilizar Matlab como se hace en los otros sensores.

Para poder procesar la señal proveniente del pulsioxímetro mediante Matlab hay dos alternativas:

- 1) Simplemente tomar una muestra de la oxigenación en sangre y de las pulsaciones por minuto en un instante concreto, como se haría en algunos análisis para conocer el estado general del paciente y por tanto el resultado se mostraría en pantalla como dos valores.
- 2) Construir una gráfica en función del tiempo que muestre la evolución de la saturación de oxígeno en sangre y las pulsaciones para poder tener una observación del progreso de estas variables. En este caso se utilizará el GUI de Matlab que permite visualizar datos en tiempo real.

3.1.1.3. Sensor de flujo de aire

Este sensor mide el flujo de aire a través de un sensor situado directamente en los orificios nasales, sirve para monitorizar la frecuencia y amplitud de la respiración.

Para entender el funcionamiento de este sensor se hace un análisis teórico del esquema eléctrico (fig. 34).

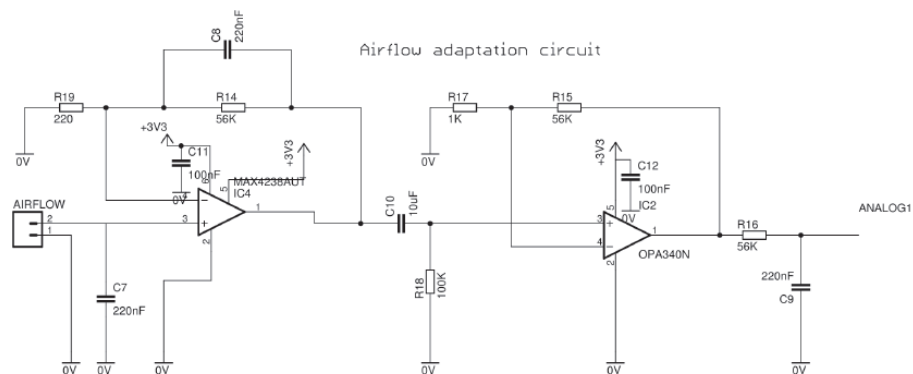


Figura 34. Circuito electrónico de la etapa de captación y amplificación de la señal.

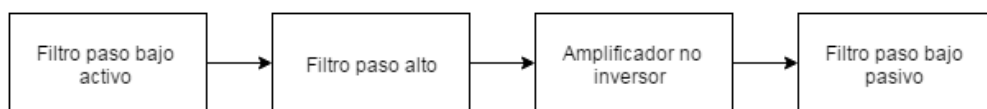


Figura 35. Diagrama de bloques del circuito electrónico de la etapa de captación y amplificación de la señal.

Antes de la primera etapa está conectado el sensor AirFlow que consiste en un termopar que proporciona tensión dependiendo de la temperatura.

En la etapa de filtro paso bajo activo además del propio filtro, la primera etapa del sensor está conectado entre la masa y el terminal V+ del operacional, tiene un condensador

que conecta a masa por tanto el sensor más el condensador compondrían un filtro paso bajo de primer orden pasivo antes de llegar al terminal V+ del operacional.

El filtro paso bajo activo su frecuencia de corte superior se calcularía de la siguiente forma:

$$f_o = \frac{1}{2\pi * R_{14} * C_8} = 12.91 \text{ Hz} \quad (30)$$

La ganancia máxima del sistema sería:

$$G = \frac{R_{14}}{R_{19}} = \frac{57 \text{ k}\Omega}{220 \Omega} = 259.09 \frac{\text{V}}{\text{V}} \quad (31)$$

La segunda etapa, el filtro paso alto pasivo tendría una frecuencia de corte inferior de:

$$f_o = \frac{1}{2\pi * C_{10} * R_{18}} = 0.15 \text{ Hz} \quad (32)$$

La tercera etapa, un amplificador no inversor su ganancia es de:

$$G = \frac{R_{15}}{R_{17}} + 1 = \frac{15 \text{ k}\Omega}{1 \text{ k}\Omega} + 1 = 57 \frac{\text{V}}{\text{V}} \quad (33)$$

Y por último la cuarta etapa, un filtro paso bajo pasivo de primer orden con frecuencia de corte superior de:

$$f_o = \frac{1}{2\pi * R_{16} * C_9} = 12.91 \text{ Hz} \quad (34)$$

Una vez se obtiene el análisis se hacen pruebas en el laboratorio, el sensor es un termopar es decir proporciona una tensión en función de la temperatura. Por tanto se mide a la salida del sensor la tensión que proporciona a distintas temperaturas sin conectarlo al Shield.

Para medir la temperatura se utilizó un termómetro electrónico del laboratorio y así conocer las temperaturas. La temperatura de 22 grados era la temperatura ambiental del laboratorio, 27 y 35 grados se obtuvieron calentando con las manos el sensor y el termómetro a la par y por último la temperatura de 45 grados calentando con un soldador el sensor.

La tabla 3 incluye los resultados obtenidos del experimento.

Temperatura (°C)	Tensión (mV)
22	0,1
27	0,2
35	0.6
45	0.9

Tabla 3.

Con los datos obtenidos se puede calcular la recta de tendencia, $V=0.0365*T-0.7271$ con un coeficiente de determinación del 0.98.

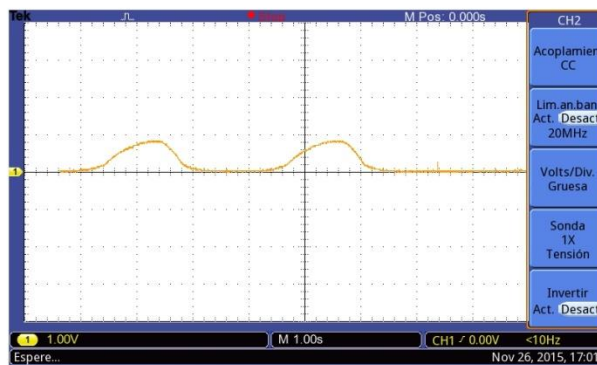


Figura 36. Señal captada por el osciloscopio a la salida del Shield.

Por último hacemos una prueba colocando el sensor en la nariz y colocando una sonda en el pin A1 donde sale la señal ya procesada y se ve (fig.36). Los picos corresponden a la expiración de aire.

Para analizar la señal de respiración se debe asignar un periodo de muestreo, ese periodo debe ser mayor que el de la respiración ya que se debe capturar toda la señal o el máximo posible. Para ello existen distintos criterios, en este caso sabiendo que en el procesamiento de señal tiene un ancho de banda de 0.15-13 Hz, con una frecuencia dos veces superior a la frecuencia máxima se cumpliría el teorema de Nyquist pero para asegurarse que se captura bien toda la señal se asigna una frecuencia de muestreo aproximadamente 10 veces mayor, 100 Hz de frecuencia de muestreo.

Por tanto se quiere muestrear y enviar datos cada 10 ms. Arduino se conoce que muestrea a 8.62 kHz. Por tanto el cálculo para conocer la velocidad de envío será:

$$T_{total} = T_{muestreo} + 2 * T_{envio} \quad (35)$$

$$10 \text{ ms} = 116 \mu\text{s} + 2 * T_{envio} \quad (36)$$

$$T_{envio} = 9884 \mu\text{s} \rightarrow f_{envio} = 101.17 \text{ Hz} \quad (37)$$

$$f_{envio} * 11 \text{ bits} = 1112.9 \text{ baudios} \quad (38)$$

La velocidad mínima que debe tener el programa de Arduino y Matlab para que cumpla las especificaciones debe ser de 9600 baudios.

3.1.1.4. GSR (Sensor de respuesta galvánico)

Este sensor consiste en medir la conductividad de la piel situando dos electrodos en los dedos. Para entender el funcionamiento se hace un análisis previo teórico.

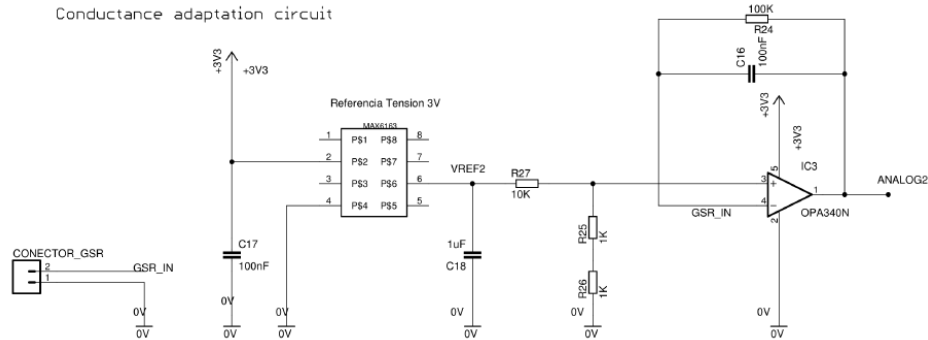


Figura 37. Esquema del circuito de adaptación de la señal GSR.

El circuito MAX6163 de Maxim Integrados, California, EEUU, es un integrado que proporciona señales de referencia de gran precisión y baja potencia. El pin 6 del integrado proporciona una tensión de 3 V. Las resistencias R25, R26 y R27 forman un divisor de tensión que convierten los 3 V en 0.5 V. Esta tensión es introducida en el terminal V+ del amplificador operacional OPA340N de Texas Instruments, Texas, EEUU.

Para calcular V_{out} del divisor de tensión los cálculos correspondientes son los siguientes:

$$V_{out} = \frac{V_{in} * (R26 + R25)}{R27 + R26 + R25} \quad (39)$$

$$V_{out} = \frac{3 * 2}{10 + 2} = 0.5 \text{ V} \quad (40)$$

El último bloque es un filtro paso bajo de orden 1 activo formado por operacional. Por tanto la frecuencia de corte se calcula:

$$f_0 = \frac{1}{2\pi * R_{24} * C_{16}} = \frac{1}{2\pi * 100 * 10^3 * 100 * 10^{-9}} = 15.91 \text{ Hz} \quad (41)$$

En el bloque de amplificación de la señal influye el usuario, ya que el usuario se conecta entre la masa y el terminal V-. Por tanto el usuario pasa a formar parte de la función de transferencia. Para simplificar el cálculo se puede suponer que el usuario es una impedancia de valor R y sustituir en el esquema eléctrico y cálculos posteriores.

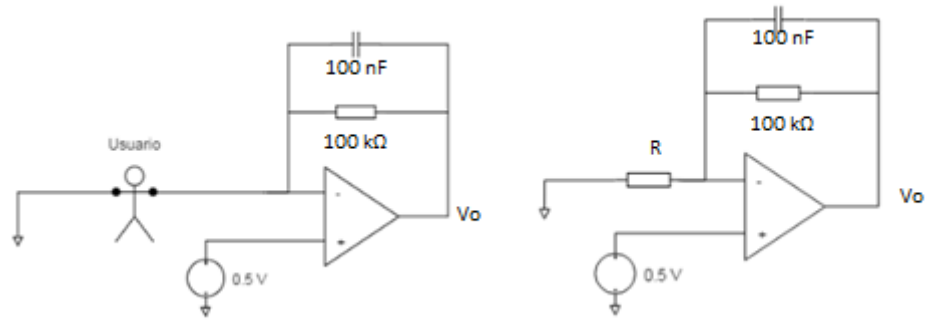


Figura 38. Donde el usuario está conectado en el esquema eléctrico (izquierda) y esquema simplificado (derecha).

Por tanto, suponiendo que no hay ruido y sólo hay componente continua el cálculo de la R, que es la impedancia del usuario conectado al GSR se puede calcular a partir de la tensión de salida:

$$\frac{V^- - 0}{R} = \frac{V_o - V^-}{100 \text{ k}\Omega} \rightarrow \frac{V^-}{V_o - V^-} * 100 \text{ k}\Omega = R \quad (42)$$

$$\frac{0.5 * 100 * 10^3}{V_o - 0.5} = R \quad (43)$$

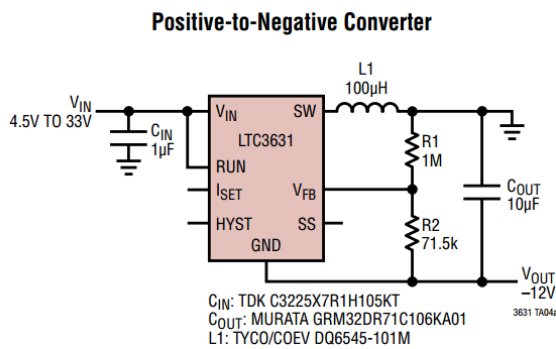
A partir de esta ecuación, y haciendo uso del programa de Arduino, se podrá calcular el valor de la resistencia del cuerpo. El valor de V_o es el dato que Arduino leerá por el canal Analógico 2 que está conectado el Shield.

4. Análisis

4.1. Alimentación eléctrica

Por lo general los Shield son alimentados a través de la tensión que proporciona Arduino (0 V a 5 V) y una intensidad máxima de unos 500 mA. Los potenciales del corazón tienen componentes negativas en algunas partes de la señal, por tanto si se desea procesar esa parte de la señal se necesita que la alimentación del Shield proporcione tensiones negativas.

Para obtener tensiones negativas existen convertidores DC-DC que pueden invertir la tensión. Un ejemplo es el integrado LTC3631 de Linear Technology, California, EEUU, que con la configuración propuesta por el fabricante en la hoja de características puede proporcionar una tensión simétrica y una intensidad máxima de 50 mA (fig 39).



Maximum Load Current vs Input Voltage

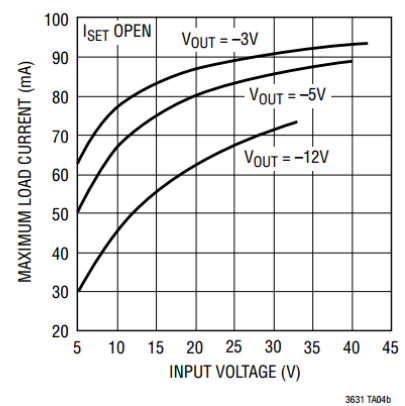


Figura 39. Izquierda, esquema propuesto por el fabricante, derecha, grafica de intensidad que puede proporcionar el integrado.

Esta intensidad es insuficiente para alimentar todo el circuito, de las pruebas hechas en la placa de prototipos se ha estimado que el Shield consume entorno 140 mA, por tanto se necesitarían varios de este integrado o una etapa previa de potencia para alimentar el circuito. Este proyecto no está centrado en el diseño de una etapa de potencia por eso simplemente se utiliza una fuente de alimentación externa que pueda proporcionar tensiones simétricas y corriente suficiente. Para conectar la alimentación se coloca un bloque terminal para PCB de 3 pines (tensión positiva, tensión negativa y masa). Es importante en este punto, conectar la masa del Shield con la masa del Arduino, ya que si las masas están aisladas la lectura del Arduino no sería correcta.

4.2.Seguridad eléctrica

La tecnología en el área de la salud ha mejorado la medicina en todas sus especialidades, mejorando la calidad y reduciendo la mortalidad. De todos modos estas nuevas tecnologías desarrolladas pueden llegar a exponer al individuo a más peligros que si no estuviera rodeado de estos instrumentos médicos, los peligros existen tanto para el paciente para como el médico. Por lo tanto es importante que éstas sean seguras y no tengan peligros para ambos.

Este apartado se centra en la seguridad eléctrica, que es uno de los peligros que ésta conlleva. A veces estos peligros se pueden deber al mal uso del aparato pero también hay que tener en cuenta cualquier posible fallo y proveer los medios para evitar que éstos afecten al usuario o que el efecto esté lo más atenuado posible.

La corriente eléctrica cuando recorre el cuerpo humano puede suponer un peligro para el individuo dependiendo de la magnitud, de la intensidad, de la frecuencia, del tiempo de duración y de la zona que afecte.

En relación a la magnitud se pueden diferenciar distintos grupos:

- Nivel de percepción (0.5 a 5 mA), es el nivel de intensidad que una individuo entorno a los 70 kg notaría el efecto de intensidad recorriendo el cuerpo.

- Contracciones involuntarias (10 mA), con este nivel de intensidad el individuo no podría controlar las contracciones del músculo, llegando a ser doloroso si se mantienen durante un periodo prolongado de tiempo.

- Parálisis respiratoria (20 mA), a este nivel de intensidad las contracciones musculares son tan fuertes que se pierde el control de los músculos respiratorios y causando por tanto parálisis respiratoria.

- Fibrilación ventricular (80 a 400 mA), el corazón como se comenta en el apartado 2.1, es susceptible a la actividad eléctrica por tanto si esta corriente parte llega a pasar en parte por el corazón ésta resulta ser suficiente como para excitar parte del musculo del corazón, la actividad del corazón normal es afectada y por tanto éste puede aumentar el ritmo cardiaco a 300 ppm debido a las ondas que llegan al ventrículo, esto puede causar que el corazón se pare causando la muerte en pocos minutos.

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

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

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

Existen una serie de normativas de los requisitos de los aparatos electrónicos:

- Marcado CE de productos sanitarios:
- Directiva 93/42 (RD 414/96)
- Normativa, marcado, documentación técnica, control fabricación)
- Norma EN 60601 Equipos Electromédicos, requisitos generales para la seguridad
- Normas adicionales con requisitos particulares:
- EN 60601-2-47 Requisitos particulares para la seguridad básica y el funcionamiento esencial de los sistemas de electrocardiografía ambulatoria. (ratificada por AENOR en julio de 2015)

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

-Equipo Clase I

Aislamiento básico

Partes metálicas accesibles conectadas a tierra

-Equipo Clase II

Aislamiento reforzado o doble aislamiento

No conectado a tierra

-Equipo Clase III

Alimentación a muy baja tensión (hasta 24 V AV o 50 V DC)

-Equipo con alimentación interna

No requiere conexión a la red eléctrica (equipos a baterías)

No puede funcionar si se conecta a la red (por ej., para cargar la batería)

4.3. Captación de la señal

Para captar la señal del ECG se colocan cuatro electrodos en las extremidades y seis electrodos en las posiciones precordiales. Como se explica en el apartado 2.1 la actividad eléctrica del corazón genera un potencial, que en los tejidos de la piel y músculos se presenta como cargas ionizadas, estas cargas ionizadas generan un potencial eléctrico dependiendo del elemento que se ionice (Sodio +, Calcio +, Cloro -, Potasio +). En reposo el músculo tiene una carga negativa, es decir, las células de éste se encuentran polarizadas y cuando llega un estímulo éstas se despolarizan y esto genera el potencial que se capta desde la superficie mediante unos electrodos colocados en la superficie de la piel. Los electrodos convierten la corriente iónica de la despolarización de las células en corriente eléctrica que circula por el cable del electrodo, esta tensión es la que se utiliza para procesar la señal.

4.4. Procesamiento de la señal

El sistema de procesamiento de la señal está compuesto por dos subsistemas principales. El subsistema analógico y subsistema digital, cada subsistema está compuesto por distintas etapas (fig. 40).

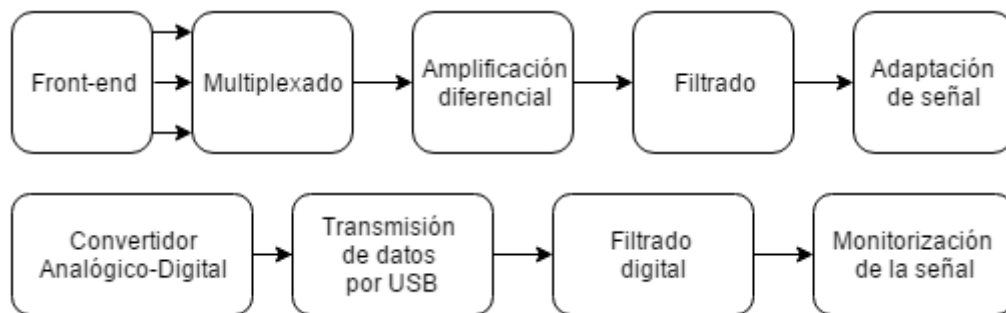


Figura 40. Diagrama de bloques de la etapa de procesamiento de señal. Arriba procesamiento analógico, abajo procesamiento digital.

El subsistema analógico está integrado en el Shield mientras que el subsistema digital pasa por el Arduino y el ordenador.

El subsistema analógico se encarga de captar la señal y procesarla para el subsistema digital. Las señales fisiológicas son continuas y pueden tomar cualquier valor de amplitud, es decir decimales infinitos. Por tanto la mejor forma es captarlos también de forma analógica aunque existen métodos más sofisticados que captan las señales de forma digital directamente.

El front-end es la primera etapa que se encuentra la señal que proviene directamente de los electrodos. Esta etapa debe garantizar que esta señal no se altere ni pierda información de interés. Para lograrlo se utiliza un seguidor de tensión que evita la desadaptación por impedancia y la segunda sub-etapa de este bloque es la red de Wilson, que sirve para generar la central terminal de Wilson (CTW), Wilson propuso ese potencial como referencia para las derivadas unipolares. Sirve como referencia cero de potenciales del cuerpo humano, como se explica en el apartado 2.2.1. Esta referencia cero se sitúa en el cuarto electrodo que está conectado a la pierna derecha. Por último para mejorar la calidad de la señal se conectan las pantallas de los cables del electrodo a un divisor de tensión que proviene del CTW y se conecta a masa, de esta forma se logra reducir drásticamente la interferencia de 50 Hz.

Arduino, como se ha dicho en apartados anteriores, es un micro controlador con algunas ventajas pero su principal inconveniente es la velocidad limitada y el número delimitado de pines analógicos. Un electrocardiógrafo clínico puede llegar a procesar las 12 derivaciones a la vez, la placa Arduino Uno solo tiene 6 entradas analógicas, además la velocidad de procesamiento también influye. Si se desea muestrear cada canal a 500 Hz, como Arduino solo puede hacer una tarea a la vez debe muestrear cada canal por separado, uno detrás de otro. Para cumplir esa frecuencia debe muestrear a 3000 Hz (500 Hz * 6 canales). Eso deja un tiempo de 333 μ s para muestrear, convertir el dato a digital y enviar el dato por el puerto serie.

Para saber si Arduino puede muestrear las 6 derivadas a la vez se calcula de forma teórica cuanto tiempo requiere convertir un dato de analógico a digital.

A partir de la información obtenida de la hoja de características del atmega328p la conversión analógico digital requiere 13 ciclos de reloj menos en la primera conversión que requiere 25 ciclos. El reloj de Arduino tiene una velocidad de 16 MHz y la preescala predeterminada por Arduino es de 128 por tanto el tiempo de muestreo es 200 μ s (ecuación 45) en la primera conversión y 104 μ s (ecuación 46). Como se está conmutando entre distintos pines analógicos la conversión se reinicia en cada muestreo, por tanto el valor que se toma es de 200 μ s.

$$preescala * ciclos * frecuencia\ reloj^{-1} = tiempo\ total \quad (44)$$

$$128 * 25 * \frac{1}{16 * 10^6} = 200 \mu s \quad (45)$$

$$128 * 13 * \frac{1}{16 * 10^6} = 104 \mu s \quad (46)$$

También se debe contar el tiempo que se necesita para enviar los datos. La máxima velocidad que puede proporcionar Arduino es 115200 baudios. La función "serial.print()" de Arduino envía los datos de forma ASCII, es decir convierte el dato en caracteres y envía estos valores en formato ASCII, por ejemplo si envía el valor 758, la función "serial.print()" está enviando 0x55 0x53 0x56 este dato ocupa dos bytes aunque en el código se esté enviando un uint8_t o int8_t que tienen un tamaño de 1 byte.

El tiempo que se requiere para enviar un dato a 115200 baudios es de 138.8 μ s.

$$\text{número de bits} * \text{frecuencia de envío}^{-1} = \text{tiempo total} \quad (47)$$

$$16 * \frac{1}{115200} = 138.8 \mu\text{s} \quad (48)$$

En total el tiempo que tarda en enviar un dato es de $138.8+200=338.8 \mu\text{s}$ superior al tiempo necesario, sin tener en cuenta que el programa también utiliza tiempo para compilar las líneas de código y llamar a las funciones correspondientes. Hay funciones y librerías para enviar los datos de forma más eficiente ocupando menos espacio esto complica el código y además también consumiría mucha capacidad computacional del Arduino. Por esa razón en este proyecto se descarta el diseño de las derivadas precordiales y además se procesan tan sólo 3 derivadas a la vez.

Por tanto la siguiente etapa es la de multiplexado, para multiplexar la señal existen distintas alternativas, utilizar jumpers que el usuario modificaría manualmente o multiplexores analógicos que se programan automáticamente desde el Arduino.

La ventaja de utilizar los jumpers es su bajo coste y sencillez a la hora de diseñar el esquema eléctrico pero su principal inconveniente es que es el usuario quien debe colocar los jumpers y si no conoce bien el esquema eléctrico puede dejar lugar a errores y la ventaja de los multiplexores analógicos es que desde el programa de Arduino se modifica y se elige el canal que se va a procesar y de esta forma no dejan lugar a error. La opción final elegida es utilizar multiplexores para facilitar el uso.

El siguiente bloque se encarga de amplificar la señal de ECG, ya que la señal de ECG original tiene un valor entre $0.4 - 2 \text{ mV}$, además es una señal diferencial. Este bloque se encarga de convertir la señal diferencial en single-ended y añadir una ganancia a la señal.

La ganancia total del sistema debe ser de unos 1000 V/V para obtener una señal en la salida de $0.4-2 \text{ V}$, una amplitud suficiente para que Arduino pueda procesar esta señal. Esta ganancia se reparte entre el bloque de amplificación y filtros analógicos, asignando una ganancia de 50 V/V al bloque de amplificación y 20 V/V a la etapa de filtrado.

El bloque de filtrado se encarga de eliminar las interferencias que puedan aparecer en la señal de ECG sin eliminar información de interés. Además también hace función de filtro antialiasing para una correcta conversión analógico digital.

Las interferencias de la señal del ECG se pueden dividir en cuatro:

-Oscilaciones de la línea base (fig. 41)

Las posibles causas de la oscilación de la línea base son: la respiración, transpiración de la piel, movimiento del cuerpo y contacto de los electrodos. La amplitud de la perturbación puede ser mayor que el conjunto QRS y de frecuencia menor de 1 Hz



Figura: Sömmo y Laguna, 2003

Figura 41. Interferencia causada por una oscilación en la línea base.

-Artefactos de movimiento (fig. 42)

Las causas de los artefactos de movimiento son los contactos de los electrodos si se mueven. La amplitud puede ser mayor que el conjunto QRS y el espectro es igual que el espectro PQRST.

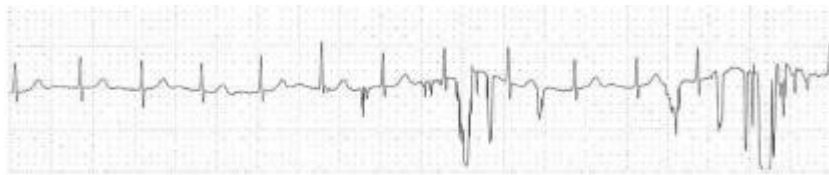


Figura: Sömmo y Laguna, 2003

Figura 42. Interferencia causada por el movimiento de artefactos.

-Interferencia de red (fig. 43)

Las interferencias de red provienen principalmente de la red eléctrica, la amplitud puede variar dependiendo de muchos factores y la frecuencia es de 50 Hz.

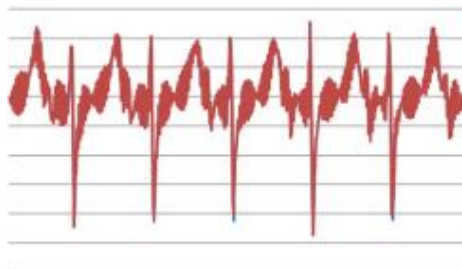


Figura 43. Interferencia causada por la red eléctrica de 50 Hz.

-Ruido electromiográfico (EMG) (fig. 44)

Ruidos electromiográficos provienen de la actividad muscular, la amplitud puede variar dependiendo del esfuerzo muscular el espectro es igual o mayor que el espectro PQRST y es una perturbación intermitente.

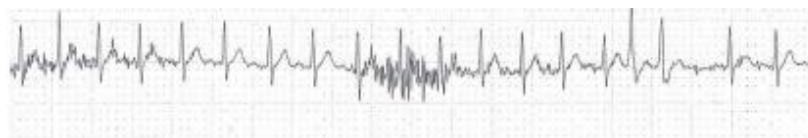


Figura: Sömmo y Laguna, 2003

Figura 44. Interferencia causada por la actividad muscular.

En la siguiente tabla se muestra las características principales que se tienen en cuenta a la hora de seleccionar un filtro:

Características de diseño	Opciones
Tecnología empleada	Filtros pasivos Filtros activos Filtros digitales
Tipo de aproximación empleada	Butterworth Chebyshev Cauer o elípticos
Orden del filtro	Primer orden Segundo orden Tercer orden Cuarto orden Quinto orden Mayores
Esquema a implementar	Rauch Realimentación positiva

Características del filtro a diseñar:

-Ancho de banda de 0.1 Hz a 100 Hz

-Ganancia de 20 V/V

Las características seleccionadas para el filtro paso bajo son:

Características de diseño	Opciones
Tecnología empleada	Filtros pasivos Filtros activos Filtros digitales
Tipo de aproximación empleada	Butterworth Chebyshev Cauer o elípticos
Orden del filtro	Primer orden Segundo orden Tercer orden Cuarto orden Quinto orden Mayores
Esquema a implementar	Rauch Realimentación positiva

El filtro paso bajo además de estas características tiene una frecuencia de corte de 100 Hz.

Características elegidas para el filtro paso alto.

Características de diseño	Opciones
Tecnología empleada	Filtros pasivos Filtros activos Filtros digitales
Tipo de aproximación empleada	Butterworth Chebyshev Cauer o elípticos
Orden del filtro	Primer orden Segundo orden Tercer orden Cuarto orden Quinto orden Mayores
Esquema a implementar	Rauch o MFB (Multi feed-back) Realimentación positiva

Este filtro paso alto se utiliza para eliminar el offset que se pueda generar por el modo común o por la diferencia de tensiones en los electrodos que pueden llegar a saturar la señal y perder toda la información del ECG. Esta frecuencia de corte es de 0.1 Hz.

Antes de la conversión analógico-digital se necesita adaptar la señal para que Arduino pueda procesarla, eso significa que la señal debe estar comprendida entre los valores 0 V y 5 V. La señal obtenida con el procesamiento del Shield tiene una amplitud de pico a pico menor de 5 V pero contiene parte negativa. Para eliminar las componentes negativas hay que añadir un offset a la señal de tal forma que toda la señal se desplace al semiplano positivo sin afectar a la ganancia total del sistema ni frecuencias de la señal.

Para eliminar este offset primero se debe saber cuánto vale la componente negativa de la señal, para ello se han hecho diversas pruebas con distintas derivadas en una placa de prototipo con un canal de ECG montado en ella. En las figuras se observa que este valor puede variar entre personas y derivadas. Este valor no puede ser muy elevado porque la señal de ECG no puede pasar los 5 V de tensión máxima, por tanto se debe implementar una solución donde este offset sea variable.

Para ello se ha implementado un divisor de tensión con un potenciómetro, de tal forma que el usuario pueda ajustar el offset hasta obtener el valor deseado. La desventaja del divisor de tensión es que consume corriente de forma constante, por tanto es importante lograr un consumo mínimo en esta etapa utilizando resistencias de valor elevado.

4.5. Conversión analógico digital

La etapa de conversión analógico-digital es la etapa intermedia entre el subsistema analógico y el subsistema digital. Esta etapa convierte los valores continuos en el tiempo y que puede tomar cualquier valor de amplitud (infinitos decimales) en valores discretos en el

tiempo y con decimales finitos. Esto es necesario para que las computadoras, en este caso Arduino y el ordenador, puedan trabajar con los datos.

Existen distintos tipos de muestreo:

-Muestreo por polling: Se comprueba cíclicamente que ha finalizado el muestreo y se utiliza un bit para informar, se puede utilizar con un generador de eventos externo o un pulso proveniente de un oscilador. La implementación es muy simple pero el periodo de muestreo viene dado por el hardware utilizado y monopoliza la CPU.

-Muestreo por retraso sw: Se comprueba el tiempo al inicio y al final del muestreo para conocer el tiempo que ha transcurrido en las operaciones para detener el programa al final del proceso y cumplir el tiempo definido de muestreo. Se necesita un sistema operativo multitarea (un algoritmo que pueda controlar la tarea que ha de ejecutar el procesador), que contenga una función que detenga el programa hasta que se finalice el tiempo indicado. La principal ventaja es que la implementación es muy sencilla y sus desventajas son que necesita un procesador que soporte multitarea y un contador a tiempo real que asegure la exactitud en el muestreo.

-Muestreo por interrupción: Se necesita un reloj para activar la interrupción y que el procesador pueda programar rutinas de manejo de interrupción, cuando se activa la interrupción se detiene el código y salta a la subrutina que es la interrupción, una vez finalizada la interrupción vuelve al código principal. Es una técnica bastante utilizada en microcontroladores, DSPs... Su principal ventaja es que libera el uso de la CPU y la principal desventaja es que es más complejo de implementar, más lento que polling (por necesidades de hacer el cambio de contexto) y es necesario deshabilitar el resto de interrupciones si se quiere mantener el muestreo regular.

-Muestreo por DMA: El hardware de adquisición envía o recibe datos directamente a/de la memoria del computador, se necesita que la computadora contenga un reloj, la diferencia es que este método transfiere directamente los datos a la memoria, mediante cada tic del reloj el dato se guarda en la memoria, el método es más rápido y se realiza en background por tanto deja la CPU libre para otros procesos.

Se utilizará Arduino, para ello se debe estudiar si cumple con las características necesarias para implementar uno de los anteriores métodos de muestreo. Arduino tiene un reloj interno que se puede activar en segundo plano, tiene un reloj para interrupciones que se puede activar a través de dos pines de la placa (dependiendo de qué modelo de microcontrolador este usando pueden tener más o menos pines de interrupción) y una función que detiene el código y espera un tiempo definido. Por tanto Arduino tiene la capacidad de implementar cualquier de los métodos anteriores.

La opción más sencilla de implementar es el muestreo por retraso SW, las principales limitaciones de este método es que monopoliza el uso de la CPU y además requiere de un reloj de precisión que actúe en segundo plano para cumplir el tiempo de muestreo definido. Para el diseño del ECG, Arduino solo tendrá una tarea que cumplir por tanto no es un problema que la CPU quede monopolizada por el muestreo de la señal, por otro lado Arduino contiene un reloj

que actúa en segundo plano y al que se puede acceder con facilidad garantizando una buena precisión. Por último existe una función que retrasa el sistema para cumplir el tiempo de muestreo necesario.

Al inicio del bucle se guarda en una variable el tiempo del reloj interno, se inicia la conversión analógica a digital y una vez terminada se guarda el tiempo del reloj interno en otra variable, estas dos variables definen el tiempo transcurrido en el código, con este valor se resta al tiempo de muestreo y se aplica a 'microdelay()', esta función retrasa el código tantos microsegundos como valor entero se le introduzca, de esta forma el código se detiene hasta que se haya cumplido el tiempo de muestreo. La siguiente tabla es un ejemplo del código que explica el funcionamiento del programa para hacer el muestreo por retraso.

time1=micros()	Guarda el tiempo del reloj
ECGbin=analogRead(0)	Muestrea y convierte el valor analógico
Serial.println(ECGbin)	Envía por el Puerto serie el valor
time2=micros()-time1	Comprueba la diferencia de tiempo que ha transcurrido
delayMicroseconds((1000000/FRECUENCIA)-time2)	Detiene el programa hasta cumplir el tiempo de muestreo definido

4.6. Comunicación

Para comunicar el Shield y Arduino con el ordenador donde se monitorizará la señal se comunicará por el puerto serie, el atmega328p directamente no puede comunicarse ya que la única comunicación serie que tiene es la UART distinta al tipo de comunicación que utiliza el puerto USB sin embargo para poder comunicarse por el puerto USB ya sea para cargar programas o compartir información la placa de Arduino UNO contiene otro integrado llamado ATmega8U2 IC, que sirve de interfaz para la comunicación entre Arduino y el puerto USB.

Arduino tiene una función llamada 'Serial.print()' que envía datos ya sea enteros, coma flotante o caracteres por el puerto USB, la velocidad de comunicación se configura con la función 'Serial.begin()' donde los valores que puede tomar son 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 o 115200 baudios. En el apartado 5.1.2 se hace el cálculo para la situación extrema donde se quieren procesar los 6 canales analógicos y se utiliza la máxima velocidad por tanto para 3 canales la velocidad de comunicación es menor, partiendo de la información del apartado 5.1.2 se calcula la velocidad de comunicación, en la ecuación 53 se multiplica por 32 ya que se envían dos valores, uno que corresponde al valor de la lectura y otro que sirve de protocolo para indicar que dato se está enviando:

$$T_{Muestreo} = T_{ADC} + T_{envio} = \frac{1}{f_{muestreo}} = \frac{1}{1500} = 666.66 \mu s \quad (49)$$

$$T_{ADC} = 204 \mu s \quad (50)$$

$$T_{envio} = n_{bits} * \frac{1}{Baudios} \quad (51)$$

$$T_{muestreo} - T_{ADC} = T_{envio} = 462.66 \mu s \quad (52)$$

$$462.66 \mu s = 32 * \frac{1}{Baud} \rightarrow Baud = 32 * \frac{1}{462.6 * 10^{-6}} = 75011.72 \text{ Baudios} \quad (53)$$

Arduino tiene unas velocidades predeterminadas como se dice en el párrafo anterior por tanto se elige el valor predeterminado superior más cercanos, es decir, la velocidad de envío es de 115200 baudios.

Los datos enviados por Arduino van llegando por el puerto USB y automáticamente son almacenados en un buffer que tiene el puerto serie, es necesario un programa que recoja estos datos para continuar con el procesamiento de señal.

Los datos una vez llegan al buffer del puerto serie están listos para ser utilizados. Existen muchas alternativas para el post-procesamiento de los datos: programas de ámbito científico de licencia, programas open-software o incluso crear un programa propio.

Un programa de ámbito científico bastante utilizado en las Universidades es el Matlab; está compuesto por distintas herramientas matemáticas útiles en muchos ámbitos de la ingeniería. Éste se puede utilizar para recopilar datos que llegan por el puerto serie, en Matlab se puede programar un “script” para ejecutar un código y captar los datos que llegan.

4.6. Post-procesamiento digital y monitorización

Una vez se tienen los datos, existen distintas alternativas para monitorizarlos; estas alternativas se pueden diferenciar en dos grupos, a tiempo real y “offline”.

A tiempo real los datos van llegando, se procesan si es necesario e inmediatamente son mostrados por pantalla uno a uno. “Offline” los datos van llegando y se van guardando en un vector o buffer para luego procesar los datos si es necesario y mostrarlos por pantalla todo el vector a la vez.

Alternativas tiempo real	Alternativas “off-line”
Matlab	Matlab
Programas open-software	Excel
Labview	Programas a bajo nivel (cualquier lenguaje programación)
Programas a bajo nivel (cualquier lenguaje programación)	

Tiempo real

Ventajas	Desventajas
Sencillez y comodidad para observar los datos como se desarrollan en el tiempo	Nivel de programación elevado, se necesitan conocimientos de programación altos
Se pueden tomar muestras de tamaño indefinido, es decir tan largas como el individuo quiera	El procesamiento de datos (filtros digitales) añade retrasos a la señal, además el propio procesador puede retrasar la toma de datos si su capacidad es muy limitada

Offline

Ventajas	Desventajas
Sencillo de implementar	No se observan los cambios al instante
El procesador no es una limitación ya que puede tomar más tiempo para procesar	El tamaño de la muestra es limitado, viene definido por el software o por el individuo

Este proyecto está centrado en el diseño del sensor biomédico y el procesamiento de la señal por tanto no es necesario el desarrollo de un software a tiempo real, además la necesidad de implementar filtros digitales complica el diseño del programa para procesamiento a tiempo real. Por tanto la opción implementada es “offline”, el programa elegido es el Matlab como se explica en el apartado anterior.

5. Diseño

5.1. Front-end

El front-end es la primera etapa de procesamiento de la señal. La función de este bloque es conectar los electrodos que captan la señal y las etapas de procesamiento de señal, está formado por unos seguidores de tensión para evitar desadaptación por impedancias. Como se puede observar en el apartado 4.3 el equivalente que forman los electrodos y piel, la impedancia de salida no es nula. Por tanto si se conecta a otra etapa donde la impedancia de entrada no es infinita se produce una desadaptación por impedancia y la tensión del sistema es distinta de la que sería en abierto. Por tanto, para evitar este problema se coloca un seguidor de tensión con un operacional que tiene impedancia de entrada muy elevada ($10^{13} \Omega$) que se puede considerar infinita y por tanto no ocurre la desadaptación por impedancia.

La red de Wilson está formada por varias resistencias de valor típico (10 k Ω) y tolerancia 0.1 %, es importante una tolerancia baja para evitar problemas de offset en los operacionales. Para conseguir el punto medio de las tres extremidades se conectan los puntos mediante unas resistencias de 10 k Ω (R_5 , R_9 y R_7), este punto medio es devuelto al cuarto electrodo situado en el pie derecho, pasando por un filtro paso bajo de frecuencia de corte de 77 kHz, este filtro paso bajo reduce los ruidos de alta frecuencia que se puedan introducir, el cuarto electrodo ayuda a reducir el modo común y reducir el ruido de la interferencia de red.

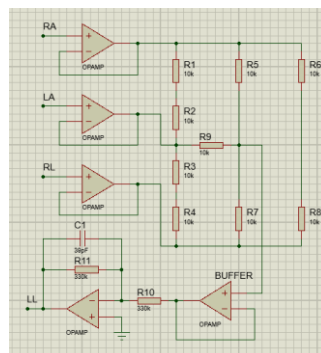


Figura 45. Esquema eléctrico del front-end, formado por los seguidores de tensión y el CTW.

La red de Wilson también sirve para obtener las derivadas aumentadas, estas derivadas se obtienen en el punto intermedio de dos extremidades. Para conseguir este punto intermedio, se conectan los dos terminales que provienen de las extremidades mediante dos resistencias, de esta forma se consigue una tensión intermedia (R_1 , R_2 , R_3 , R_4 , R_6 y R_8) entre los dos puntos, estas resistencias tienen un valor típico de 10 k Ω .

Por último para reducir drásticamente la interferencia de la red eléctrica se conecta un divisor de tensión a las pantallas de los cables de los electrodos y que provenga del electrodo de la pierna izquierda (LL) que sirve de referencia (fig. 46), con este montaje se devuelve parte de la señal de modo común reduciendo la interferencia de 50 Hz (fig. 47)

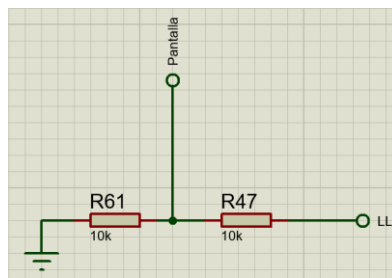


Figura 46. Divisor de tensión de la pantalla para reducir la interferencia de red.

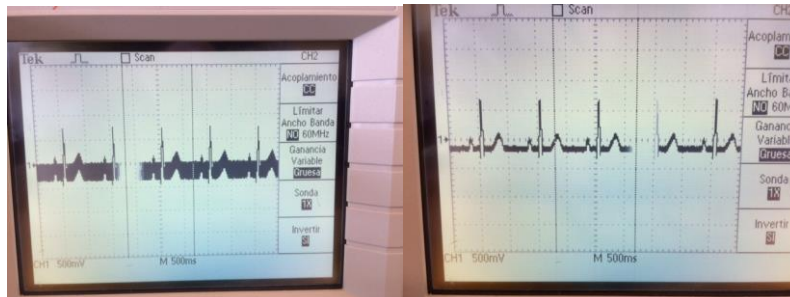


Figura 47. Pruebas con el prototipo, izquierda sin la pantalla conectada a masa, derecha con la pantalla conectada a masa

5.2. Multiplexado

Para el multiplexado de la señal se utiliza el integrado 74HCT4053, Texas Instruments, Texas EEUU, un multiplexador 2 a 1 de 3 canales, pudiendo elegir que canales conmutar a través de 3 bits selectores (tabla 4).

Enable	Estado de las entradas			Canales activos
	S_2	S_1	S_0	
L	L	L	L	C0,B0,A0
L	L	L	H	C0,B0,A1
L	L	H	L	C0,B1,A0
L	L	H	H	C0,B1,A1
L	H	L	L	C1,B0,A0
L	H	L	H	C1,B0,A1
L	H	H	L	C1,B1,A0
L	H	H	H	C1,B1,A1
H	X	X	X	Ninguna

Tabla 4. Tabla de verdad del integrado 74HCT4053 dada por el fabricante.

Con este multiplexor se puede conmutar entre las derivadas estándar y las aumentadas, conectando las tres salidas a las tres líneas de procesamiento de señal (A,B y C). Antes del multiplexado hay 12 líneas, dos por cada derivada, ya que la señal es diferencial, por tanto se necesitan 2 multiplexores.

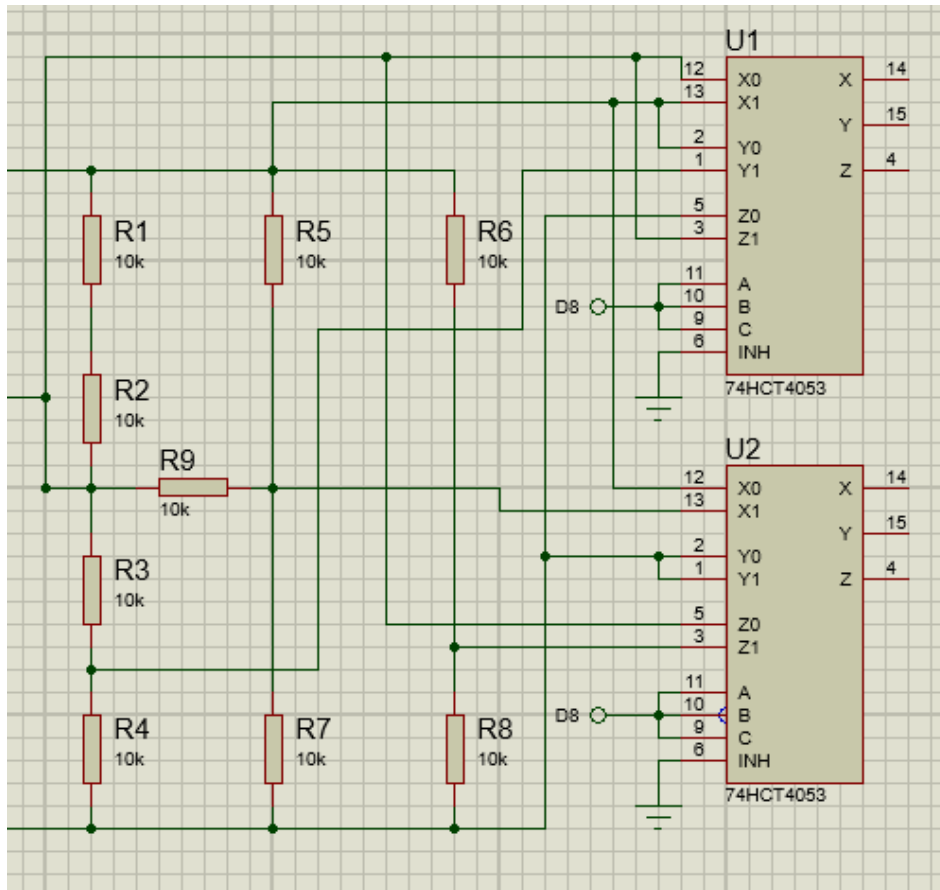


Figura 48. Esquema eléctrico de la conexión de los multiplexores al CTW.

El diseño queda hecho de tal forma que por los canales de subíndice 0 entran las líneas que corresponden a las derivadas estándar, y por los canales de subíndice 1 entran las líneas que corresponden a las derivadas aumentadas (fig. 48).

DI=LA-RA	LA	X0 (U1)	$aVR=RA-(LA+RL)/2$	RA	X1 (U1)
	RA	Y0 (U1)		$(LA+RL)/2$	Y1 (U1)
DII=LL-RA	RL	Z0 (U1)	$aVL=LA-(RA+RL)/2$	LA	Z1 (U1)
	RA	X0 (U2)		$(RA+RL)/2$	X1 (U2)
DIII=LL-LA	RL	Y0 (U2)	$aVF=RL-(RA+LA)/2$	RL	Y1 (U2)
	LA	Z0 (U2)		$(RA+LA)/2$	Z1 (U2)

Tabla 5. Resumen de las conexiones de la figura 48.

A través de los pines 11, 10 y 9 se selecciona que salida se quiere activar (tabla 5), como se quieren conmutar todas las entradas a la vez, se unen los pines.

5.3. Amplificación de la señal

Para la amplificación de la señal se utilizan los amplificadores de instrumentación AD620 de Analog Devices de Massachusetts, EEUU, un amplificador de instrumentación con una ganancia ajustable con una resistencia externa que el rango de la ganancia es de 1 a 10000 V/V, baja potencia (1.3 mA máxima corriente de alimentación), 50 μ V máxima tensión de entrada de offset, 1 nA máxima corriente de polarización, bajo ruido y alto rechazo a modo común. Este integrado es utilizado en distintos aparatos médicos que requieren buena precisión.

La ecuación 54 expresa la tensión de salida en función de la tensión de referencia, tensión de entrada y tensión de modo común. La tensión de modo común viene dada por las ecuaciones 55 y 56.

$$V_{out} = (V^+ - V^-) * G + V_{ref} + A_c * V_c \quad (54)$$

$$CMRR = 20 \log_{10} \left(\frac{G}{A_c} \right) \rightarrow A_c = \frac{G}{10^{\frac{CMRR}{20}}} \quad (55)$$

$$A_c = \frac{50}{10^5} = 5 * 10^{-4} \frac{V}{V} \quad (56)$$

El rechazo a modo común es bastante elevado y se puede despreciar ($A_c \cong 0$) en el cálculo de ganancia del sistema y tensión de referencia por tanto se considera $A_c * V_c = 0$.

Esta etapa debe tener una ganancia de 50 V/V, a partir de la hoja de datos proporcionada por el fabricante se calcula la resistencia ganancia necesaria:

$$G = 1 + \frac{49.7 k}{R_g} \rightarrow R_g = \frac{49.7 k}{49} \cong 1 k\Omega \quad (57)$$

Por último la V_{ref} debe ser 0 ya que esta etapa es previa a un filtro paso alto que elimina la línea base de la señal. Por tanto cualquier valor de referencia que se añada es eliminado a posteriori.

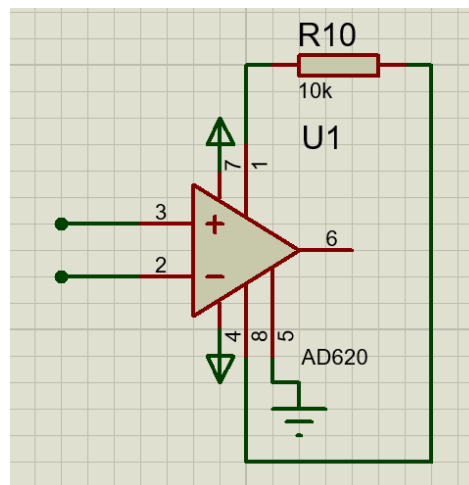


Figura 49. Esquema eléctrico de los amplificadores de instrumentación.

5.4. Filtro analógico

Para diseñar el filtro paso alto, se utilizan las características definidas en el apartado 4.3, por tanto para calcular la frecuencia de corte se utiliza la ecuación 58.

$$f_{corte} = \frac{1}{2\pi RC} \rightarrow f_{corte} = 0.1 \text{ Hz} \quad (58)$$

A partir de esta ecuación se puede calcular los valores de resistencia y condensador pero tiene dos incógnitas por tanto se define uno de los dos valores para calcular el otro y cumplir con la ecuación. Se define el valor del condensador ya que este valor viene más limitado, condensadores no electrolíticos de capacitancias muy elevadas son difíciles de encontrar, caros o de gran tamaño. Por tanto se fija el valor de $C=1 \mu\text{F}$. Una vez calculado se obtiene el valor de $R=1.59 \text{ M}\Omega$. Este valor es bastante elevado para resistencias por tanto se utilizan dos resistencias en serie de valores menores que sumen el valor equivalente o más aproximado. $R_1=953 \text{ k}\Omega$ y $R_2=510 \text{ k}\Omega$, estos son los valores estandarizados la suma de los dos es inferior a la del resultado calculado. Se rehace el cálculo y se obtiene que la frecuencia de corte es de 0.11 Hz un valor aceptable para el filtro que se está diseñando

$$0.1 * C * 2\pi = \frac{1}{R} \rightarrow R = (0.1 * 1 * 10^{-6} * 2\pi)^{-1} R = 1.59 \text{ M}\Omega \quad (59)$$

$$f_{corte} = \frac{1}{2\pi * 1.4 * 10^6 * 1 * 10^{-6}} = 0.11 \text{ Hz} \quad (60)$$

El filtro paso bajo que se tiene que diseñar también hace función de filtro anti-aliasing por eso se utiliza un filtro de tipo Butterworth por su respuesta plana y proporciona una buena precisión para la conversión de datos, el filtro de quinto orden ha sido elegido para una buena atenuación de las interferencias de alta frecuencia. Se ha elegido la estructura Rauch que utiliza menos componentes y por tanto requiere menos espacio.

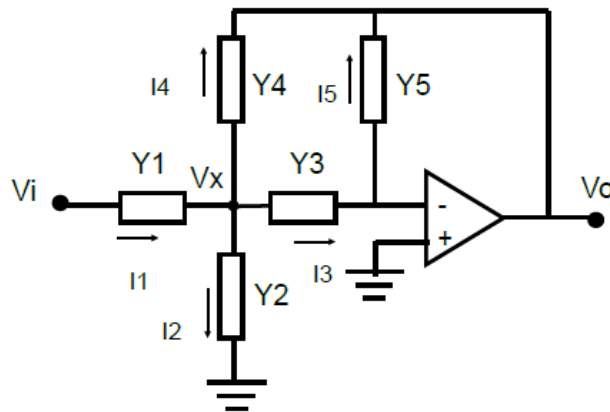


Figura 50. Esquema general estructura MFB orden 2.

En la figura 50 se puede ver el esquema general de la estructura MFB de orden 2 y a partir de la estructura se puede sacar la función de transferencia general. El esquema de un filtro paso bajo de orden 2 y estructura MFB se muestra en la figura 50 y su función de transferencia. Para definir la ganancia, frecuencia de corte y el coeficiente de amortiguamiento se utilizan la formulas siguientes:

$$\frac{V_o}{V_i} = \frac{Y_1 * Y_3}{Y_5 * (Y_1 + Y_2 + Y_3 + Y_4) + Y_3 * Y_4} \text{ donde } Y_i = \frac{1}{Z_i} \quad (61)$$

Para diseñar un filtro paso bajo las impedancias Y_5 y Y_2 se sustituyen por condensadores y el resto de impedancias por resistencias.

$$\frac{V_o}{V_i} = \frac{\frac{R_4}{R_1}}{R_3 * R_4 * C_2 * C_5 * (\omega j)^2 + R_3 * R_4 * \left(\frac{1}{R_1} + \frac{1}{R_3} + \frac{1}{R_4}\right) * C_5 * \omega j + 1} \quad (62)$$

$$K = -\frac{R_4}{R_1} \quad (63)$$

$$\omega_0 = \frac{1}{\sqrt{R_3 * R_4 * C_2 * C_5}} \quad (64)$$

$$a = \frac{1}{2} * R_3 * R_4 * C_5 * \frac{\frac{1}{R_1} + \frac{1}{R_3} + \frac{1}{R_4}}{\sqrt{R_3 * R_4 * C_2 * C_5}} \quad (65)$$

El filtro tiene orden 5 y esta estructura tiene orden dos por tanto se utilizan dos filtros de orden 2 y uno de orden uno en serie para obtener en total un filtro de orden 5. Siguiendo esta estructura, la ganancia total de los 3 filtros debe valer 20 V/V. En los filtros MFB la ganancia es negativa pero al utilizar dos la ganancia es positiva, otro detalle a tener en cuenta es que si definimos previamente $R_8=R_7=R_6=R_5=R_4=R_3=R$, las formulas se simplificando:

$$K = -\frac{R}{R} = -1 \quad (66)$$

$$\omega_0 = \frac{1}{\sqrt{R^2 * C_2 * C_5}} = \frac{1}{R\sqrt{C_2 * C_5}} \quad (67)$$

$$a = \frac{1}{2} * R^2 * C_5 * \frac{\frac{3}{R}}{R * \sqrt{C_2 * C_5}} = \frac{1}{2} * R^2 * C_5 * \frac{3}{R^2 * \sqrt{C_2 * C_5}} = \frac{1}{2} * C_5 * \frac{3\sqrt{C_2 * C_5}}{C_2 * C_5} = \frac{3}{2} * \frac{\sqrt{C_2 * C_5}}{\sqrt{C_2^2}} = \frac{3}{2} * \sqrt{\frac{C_5}{C_2}} \quad (68)$$

$$a = \frac{3}{2} * \sqrt{\frac{C_5}{C_2}} \quad (69)$$

Los valores de ganancia, frecuencia de corte y coeficiente de amortiguación vienen dados por las características del filtro $K=20$ V/V y $\omega_0=2\pi*100$ rad/s, en un filtro tipo Butterworth los coeficientes de amortiguación tienen valores predeterminados dependiendo del orden, para un filtro de orden 5 los valores son $a_1=0.618$ y $a_2=1.618$ a partir de estos datos se determinan los valores de resistencias y condensadores de los dos filtros. La ganancia debe repartirse entre los tres filtros, pero a partir del cálculo anterior se ha definido que la ganancia será unitaria en los filtros de segundo orden por tanto el filtro de primer orden es el que tiene toda la ganancia.

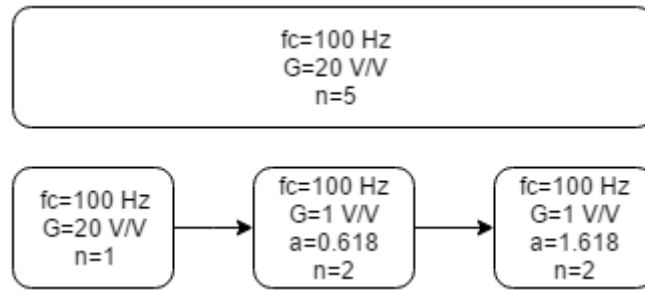


Figura 51. Diagrama de bloques de la etapa de filtrado analógico.

Para los componentes de los filtros de segundo orden los cálculos son los siguientes:

$$2 * \pi * 100 = \frac{1}{R\sqrt{C_2 * C_5}} \quad (70)$$

$$0.618 = \frac{3}{2} * \sqrt{\frac{C_5}{C_2}} \quad (71)$$

$$1.618 = \frac{3}{2} * \sqrt{\frac{C_7}{C_6}} \quad (72)$$

El sistema de ecuaciones se reduce a dos ecuaciones con tres incógnitas por tanto se obtiene un sistema compatible indeterminado con infinitas soluciones para obtener una única solución se define $R=10 \text{ k}\Omega$ un valor estándar. Por tanto para el primer filtro de segundo orden:

$$200\pi = \frac{1}{10^3\sqrt{C_2 * C_5}} \rightarrow 200\pi * 10^4 = \frac{1}{\sqrt{C_2 * C_5}} \quad (73)$$

$$0.618 = \frac{3}{2} * \sqrt{\frac{C_5}{C_2}} \rightarrow \frac{0.618 * 2}{3} = \sqrt{\frac{C_5}{C_2}} \quad (74)$$

$$\left(\frac{1.236}{3}\right)^2 = \frac{C_5}{C_2} \rightarrow C_2 * 0.169 = C_5 \quad (75)$$

Se sustituye C_5 en la ecuación 76.

$$200\pi * 10^4 = \frac{1}{\sqrt{C_2 * C_2 * 0.169}} \quad (76)$$

$$200\pi * 10^4 = \frac{1}{C_2\sqrt{0.169}} = \frac{1}{0.412 * C_2} \quad (77)$$

$$C_2 = \frac{1}{200 * \pi * 10^4 * 0.412} = 386 \text{ nF} \quad (78)$$

$$C_2 * 0.169 = C_5 \rightarrow C_5 = 65.23 \text{ nF} \quad (79)$$

Estos valores no son los valores estandarizados de condensadores, deben redondearse a valores estándar más cercanos. Con los valores normalizados $C_2=390 \text{ nF}$ y $C_5=68 \text{ nF}$.

Para el segundo filtro de segundo orden:

$$200\pi = \frac{1}{10^3 \sqrt{C_6 * C_7}} \rightarrow 200\pi * 10^4 = \frac{1}{\sqrt{C_6 * C_7}} \quad (80)$$

$$1.618 = \frac{3}{2} * \sqrt{\frac{C_7}{C_6}} \rightarrow \frac{1.618 * 2}{3} = \sqrt{\frac{C_7}{C_6}} \quad (81)$$

$$\left(\frac{3.236}{3}\right)^2 = \frac{C_7}{C_6} \rightarrow C_6 * 1.163 = C_7 \quad (82)$$

Se sustituye C_7 en la ecuación 83.

$$200\pi * 10^4 = \frac{1}{\sqrt{C_6 * C_6 * 1.163}} \quad (83)$$

$$200\pi * 10^4 = \frac{1}{C_6 \sqrt{1.163}} = \frac{1}{1.078 * C_6} \quad (84)$$

$$C_6 = \frac{1}{200 * \pi * 10^4 * 1.078} = 148 \text{ nF} \quad (85)$$

$$C_6 * 1.163 = C_7 \rightarrow C_7 = 172.18 \text{ nF} \quad (86)$$

Estos valores no son los valores estandarizados de condensadores, deben redondearse a los valores estándar más cercanos. Con los valores normalizados $C_6=150 \text{ nF}$ y $C_7=180 \text{ nF}$.

Por último queda calcular el valor de la ganancia de filtro paso bajo y sus componentes. La ganancia de un filtro paso bajo de primer orden se calcula con la ecuación 87 y su frecuencia de corte con la ecuación 88.

$$K = \frac{R_2}{R_1} + 1 \quad (87)$$

$$\omega_0 = \frac{1}{RC} \quad (88)$$

El valor de ganancia debe ser de 20 V/V y la frecuencia de corte de 100 Hz.

$$20 = \frac{R_2}{R_1} + 1 \quad (89)$$

$$200\pi = \frac{1}{RC} \quad (90)$$

Se obtiene un sistema de ecuaciones con dos ecuaciones y cuatro incógnitas por tanto se tiene dos grados de libertad. Para calcular el valor de ganancia se elige un valor estándar para la resistencia $R_2=19 \text{ k}\Omega$ y para la frecuencia de corte se define un valor de C que puede ser más limitado a la hora de encontrar condensadores de gran capacitancia, por tanto se elige un valor normal de $C=100 \text{ nF}$.

$$20 = \frac{20 * 10^3}{R_1} \rightarrow R_1 = 1 \text{ k}\Omega \quad (91)$$

$$R = \frac{1}{200\pi * 100 * 10^{-9}} = 15.9 \text{ k}\Omega \quad (92)$$

Los valores obtenidos se deben estandarizar según la serie E24 por tanto los valores son $R_1=1 \text{ k}\Omega$, $R_2=19 \text{ k}\Omega$, $R=15 \text{ k}\Omega$ y $C=100 \text{ nF}$

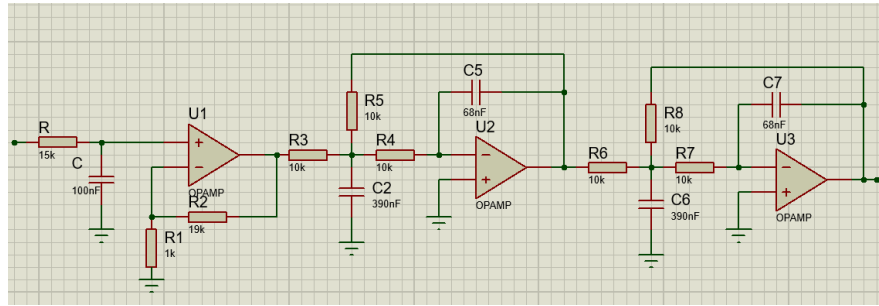


Figura 52. Esquema del filtro paso bajo de quinto orden diseñado.

Una vez diseñado se comprueba el funcionamiento implementando el filtro en una placa de prototipo y haciendo el diagrama de bode (tabla 6).

frecuencia	Vin (pp)(mV)	Vout (pp)(V)	G (V/V)	G (dB)
1,4	200	3,54	17,7	24,96
2,17	200	3,54	17,7	24,96
5,2	200	3,54	17,7	24,96
10	200	3,54	17,7	24,96
20	200	3,54	17,7	24,96
30	200	3,54	17,7	24,96
40	200	3,54	17,7	24,96
50	200	3,54	17,7	24,96
60	200	3,54	17,7	24,96
70	200	3,44	17,2	24,71
80	200	3,1	15,5	23,81
90	200	2,46	12,3	21,80
100	200	1,92	9,6	19,65
150	200	0,3	1,5	3,52
200	200	0,047	0,235	-12,58
300	200	0,012	0,06	-24,44

Tabla 6. Datos obtenidos para hacer el diagrama de bode del filtro.

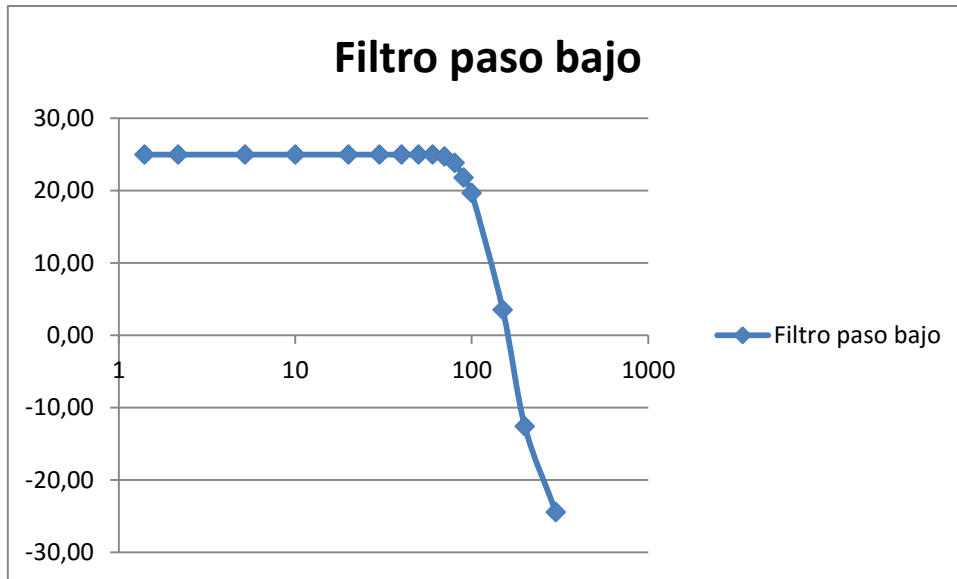


Figura 53. Diagrama de bode del filtro paso bajo

5.4.1. Alternativa

En la PCB diseñada hay tres filtros analógicos paso bajo, por esa razón 3 filtros paso bajo de orden 5 necesitan un espacio grande dificultando el diseño de la PCB por esta razón se opta por el diseño de un filtro paso bajo de primer orden.

Se mantiene el valor de la ganancia ya que si tiene toda la ganancia en un mismo bloque las interferencias del ECG se amplifican superponiéndose a la señal de interés.

Por tanto se diseña un circuito RC con un operacional que tenga una ganancia de 20 V/V y una frecuencia de corte de 100 Hz.

$$f_{corte} = \frac{1}{R_1 C * 2\pi} = 100 \text{ Hz} \quad (93)$$

Se elige un valor estándar del condensador para definir el valor de R_1 , el valor elegido es $C=100 \text{ nF}$.

$$100 * 10^{-9} * 2 * \pi = \frac{1}{R_1} \rightarrow R_1 = (100 * 10^{-9} * 2 * \pi)^{-1} = 100 \quad (94)$$

$$R_1 = 15.9 \text{ k}\Omega \quad (95)$$

El valor más cercano comercial es 15 k Ω .

Por último queda calcular la ganancia del filtro, para ello se utiliza un amplificador operacional no inversor.

$$G = 1 + \frac{R_2}{R_3} = 20 \quad (96)$$

Si se define R_3 con un valor estándar de 1 k Ω

$$1 + \frac{R_2}{1k\Omega} = 20 \rightarrow R_2 = 19 k\Omega \quad (97)$$

El valor más cercano comercial es el valor de 18 kΩ.

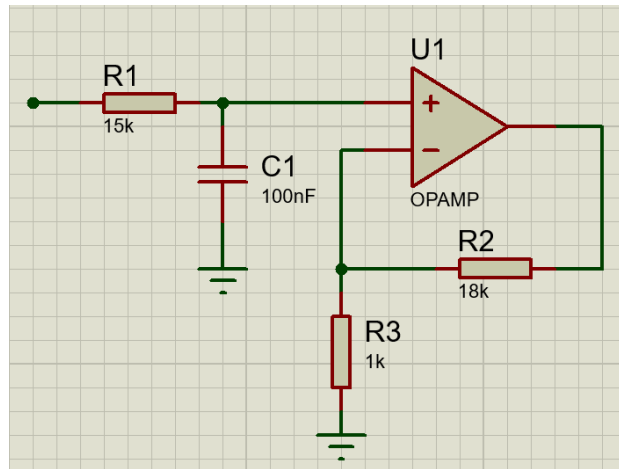


Figura 54. Filtro paso bajo de primer orden activo.

Esta reducción de orden en el filtrado puede implicar una pérdida de calidad en éste, por eso en la figura 54, a simple vista, el ruido de alta frecuencia parece atenuado. Para asegurar este resultado de forma más exacta se capta la misma señal de ECG con el Matlab y más adelante (apartado 5.6.3) se aplica un diagrama de Fourier para observar el espectro de frecuencias de la señal de ECG, en la figura 63 se ve como aún hay frecuencias mayores de 100 Hz que pasan del filtro por eso se debe implementar un filtro paso bajo digital para terminar de filtrar debidamente la señal.

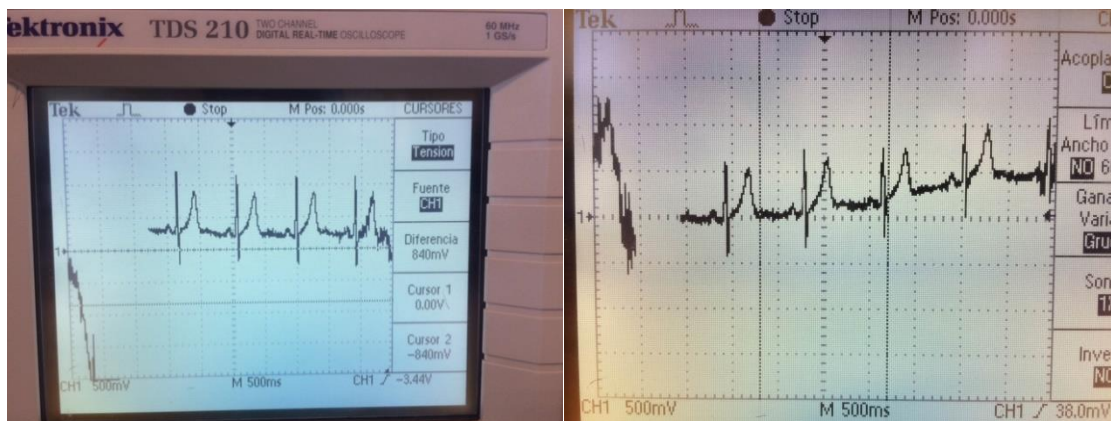


Figura 55. Izquierda ECG implementado con un filtro paso bajo de 5 orden, derecha ECG implementado con un filtro paso bajo de 1 orden.

5.5. Adaptación de la señal para Arduino

Como se comenta en el apartado 4.3 se debe añadir una etapa que sume una tensión de off-set para eliminar las componentes negativas de la señal de ECG.

Para ello se diseña un divisor de tensión con una resistencia variable para poder modificar el valor de off-set, se implementa un seguidor de tensión para que haya una desadaptación por impedancias, el consumo de corriente del divisor de tensión viene dado por la formula 98. El operacional utilizado tiene una intensidad máxima de polarización según la hoja de características de 10 pA. Luego para calcular la intensidad que consume el divisor de tensión se considera la intensidad de polarización 0. La intensidad viene dada por la siguiente formula

$$I_{divisor} = \frac{5 - 0}{R_1 + R_{V1}} \quad (98)$$

Por tanto para conseguir que la intensidad sea baja se necesita valores de resistencia elevados.

$$I_{divisor} = \frac{5}{100 * 10^3} = 50 \mu A \quad (99)$$

Con resistencias del orden de centenares de ohmios se consigue un consumo bajo. Como se comenta en los párrafos anteriores se coloca una resistencia variable para poder modificar el valor. Este valor de off-set no debe ser muy elevado por ello se coloca una resistencia de valor fijo y una resistencia variable en serie, para que el valor máximo no sature la señal y el valor mínimo llegue a valer 0.

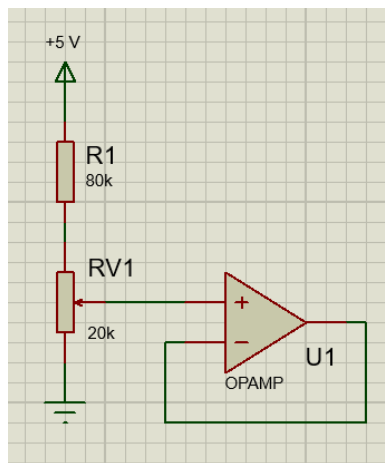


Figura 56. Esquema eléctrico del divisor de tensión.

Con los valores de la figura 56 se consigue que la tensión varíe entre 0 V y 1 V.

$$V_o = V_i * \frac{R_{V1}}{R_1 + R_{V1}} \quad (100)$$

$$V_{o\ max} = \frac{5 * 20}{100} = 1\ V \quad (101)$$

$$V_{0\ min} = \frac{5 * 0}{100} = 0\ V \quad (102)$$

Una vez se tiene el off-set se debe añadir a la señal de ECG, es importante no modificar las características principales como ganancia y frecuencia, por eso se implementa un sumador no inversor con operacionales, la ganancia es de 1 V/V. Para el valor de resistencia se elige un valor estándar de 10 kΩ.

$$G = V_{ECG} * \frac{R_5}{R_2} + V_{offset} * \frac{R_5}{R_3} \quad (103)$$

$$V_{offset} = 0 \quad (104)$$

$$G = V_{ECG} * \frac{R_5}{R_2} \rightarrow G = 1 \quad (105)$$

$$R_5 = R_2 \quad (106)$$

$$V_{ECG} = 0 \quad (107)$$

$$G = 1 = V_{offset} * \frac{R_5}{R_3} \quad (108)$$

$$R_5 = R_3 \quad (109)$$

$$R_5 = R_4 = R_3 = R_2 = R_1 = 10\ k\Omega \quad (110)$$

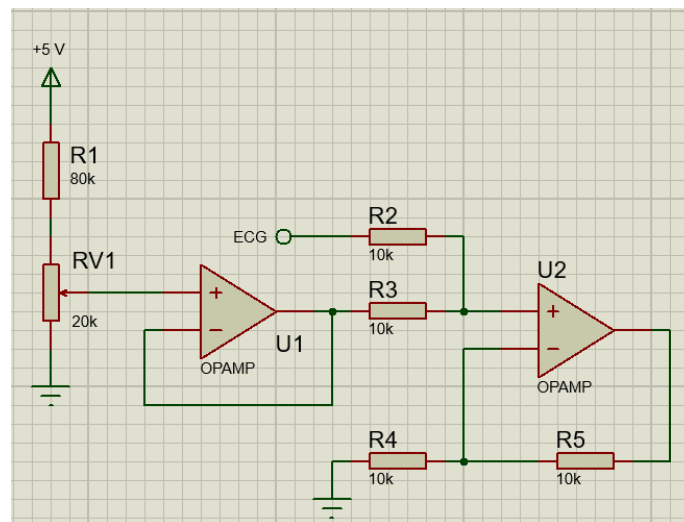


Figura 57. Esquema eléctrico del sumador de tensión para añadir un off-set a la señal del ECG.

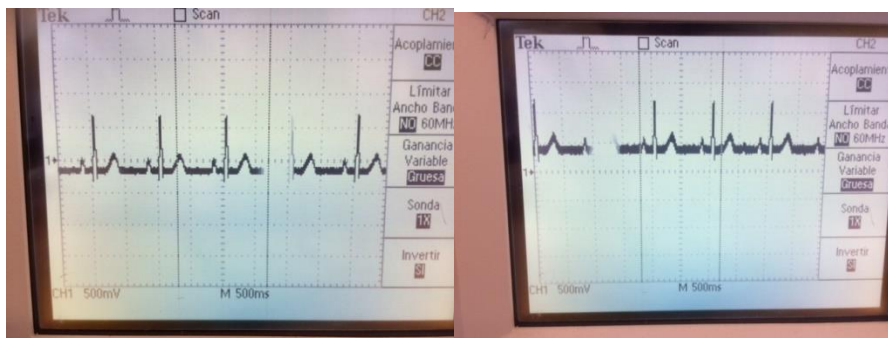


Figura 58. Señal generada por el simulador de ECG. Izquierda, señal sin off-set añadido. Derecha, señal con un off-set de 0.5 V.

5.6. Diseño PCB

Los bloques una vez diseñados se unen entre ellos para formar el sistema total. A partir de la etapa de multiplexado se separan tres líneas distintas con los mismos filtros y sumadores diseñados en los apartados anteriores.

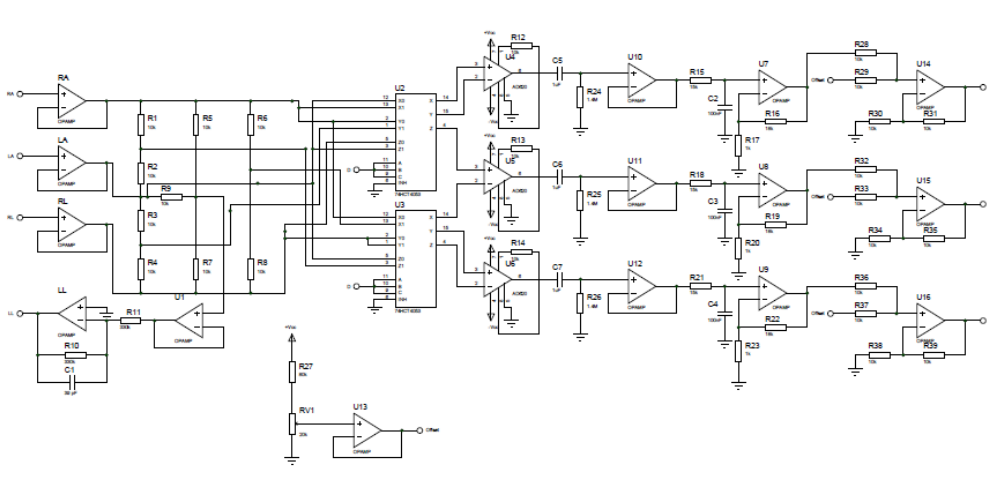


Figura 59. Esquema eléctrico total del sistema.

Con el diseño hecho en ISIS de Proteus, existe el programa Ares de Proteus, este programa sirve para diseñar los circuitos impresos, a partir de un esquema eléctrico. (fig. 59).

Para el diseño se siguen unas pautas necesarias para poder hacer el circuito impreso. Las pistas deben tener una anchura mínimo de 2 mm y una separación entre pistas de 2mm. Los path para hacer el cambio de cara, la circunferencia interior debe medir al menos 3 mm y la exterior 5 mm. El tamaño mínimo para los componentes SMD de es el 0810 y el encapsulado de los integrados SOIC 14.

Para simplificar el diseño y mejorar la calidad de la señal se ha creado un plano de masa, es decir toda la placa del circuito impreso está conectada a masa y aparte separada del resto de pistas.

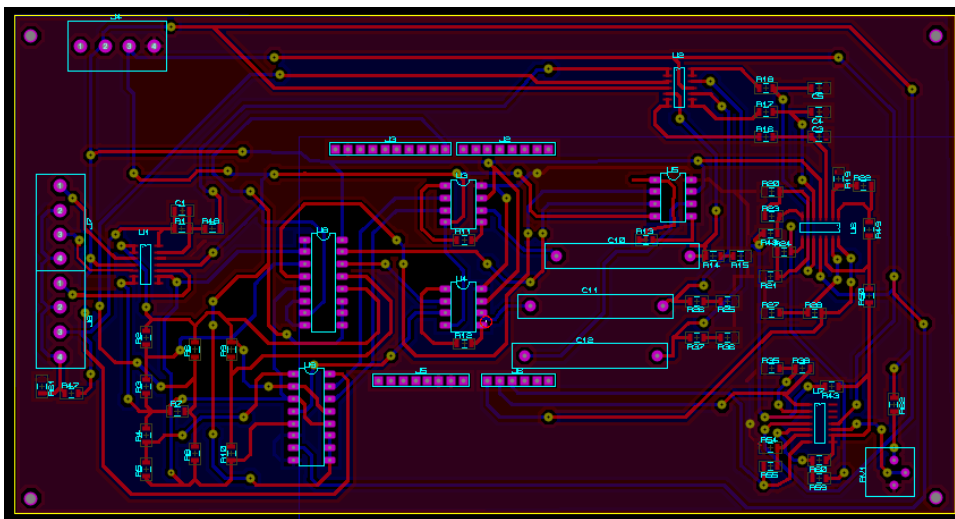


Figura 60. Esquema de ARES para diseñar la placa de circuito impreso.

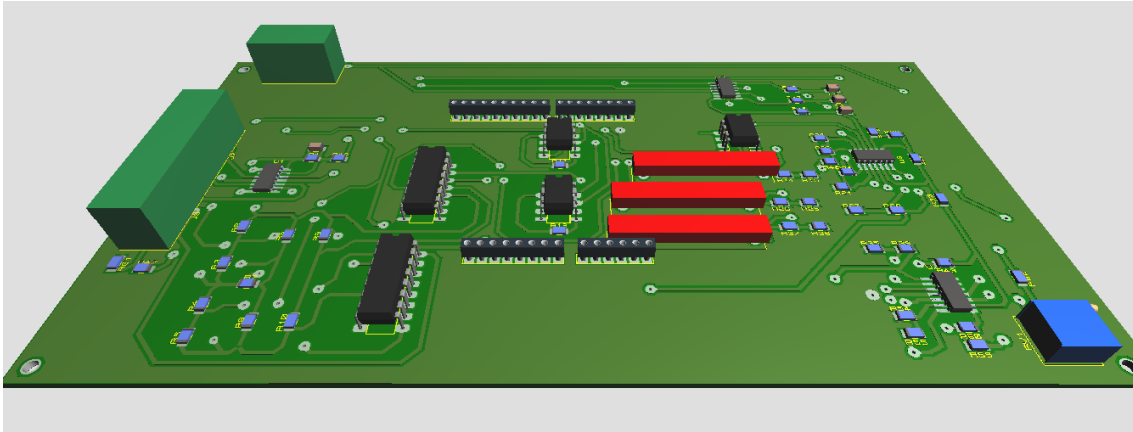


Figura 61. Simulación del resultado de la placa de circuito impreso en ARES.

5.6. Procesamiento digital

5.6.1. Conversión analógico digital

Arduino incorpora un convertidor analógico digital de 10 bits, que se puede utilizar con la función 'analogRead()', las tensiones de entrada entre 0 y 5 voltios las integra entre los valores 0 y 1023, eso significa una resolución de $5\text{ V} / 1024$ unidades, es decir, 4.9 mV por unidad. El rango de entrada y la resolución pueden ser modificados con la función 'analogReference()'. El valor de referencia no puede superar los 5 V y la resolución no puede ser mayor a 10 bits. El tiempo que se necesita para hacer una conversión está explicado en el apartado 4.3. Como se explica en el apartado 4.4 Arduino hace un muestreo por retraso SW, por tanto después de función 'analogRead()' se debe programar un retraso que detenga el programa hasta que se complete el tiempo de muestreo, Arduino tiene la función 'microdelay()', esta función detiene el programa tantos microsegundos como se necesiten.

Para conseguir que el tiempo de cada iteración corresponda con el periodo de muestreo. Se incluye una variable que accede al reloj interno de Arduino y toma el valor que corresponda, una vez ha hecho la conversión y enviado el dato, se vuelve a consultar el valor del reloj, se restan estos dos valores y se obtiene el tiempo que ha transcurrido, por último a la función 'microdelay()' se le pasa el valor de muestreo menos el valor de tiempo que ha transcurrido y de esta forma consigue que cada iteración tarde el mismo tiempo que el periodo de muestreo.

Por último se envía un número que sirve de protocolo para que Matlab reconozca cuando ha terminado de llegar el último dato y por tanto pueda identificar el siguiente valor como primer dato de los tres que deben llegar.

5.6.2. Comunicación serie

Como se explica en el apartado 5.1.2 los datos no llegan sólo con la información si no acompañados de unos protocolos que el programa que los recibe debe entender para poder leer el dato correctamente; además deben tener la misma velocidad de transmisión de datos para una correcta lectura. En el párrafo anterior se define la velocidad en la cual Arduino envía los datos (115200 baudios), por tanto Matlab se configura a esta velocidad, además Matlab tiene distintas herramientas para la lectura de los datos recibidos por el puerto serie, como la función "fscanf()", esta función lee los datos que llegan por el puerto USB, está

configurado para leer datos del puerto USB y el parámetro que devuelve puede ser un carácter, un entero o de coma flotante, esto dependerá del parámetro que se configure. En el script de Matlab se debe incluir el protocolo de comunicación creado para que pueda identificar la posición de los datos que llegan.

Matlab es un programa matemático con muchas herramientas que permiten usarlo de muchas formas, en este caso se utiliza Matlab para captar los datos que llegan por el puerto USB, procesarlos y monitorizarlos, en este apartado se explica la parte de cómo se reciben los datos.

<code>while constante==0</code>	Iniciar bucle
<code>variable=fscanf(PS, '%d');</code>	Leer número entero del puerto serial
<code>if variable==50000 variable==40000</code>	Comprobar si se ha recibido el número del protocolo que indica el fin de la cadena.
<code>constante=1;</code>	Activar variable que indica que se ha recibido el fin de la cadena
<code>else</code>	
<code>constante=0;</code>	El número de fin de la cadena no ha sido recibido
<code>end</code>	
<code>end</code>	Fin del bucle

Con este bucle Matlab no continua con el programa hasta que recibe el numero 50000 ó 40000, estos números son los que indican el final de la cadena de datos y además identifican si se está enviando los datos de las derivadas aumentadas (50000) o los datos de las derivadas principales (40000).

5.6.3. Filtro digital

En el apartado 5.4.1 se reduce el orden del filtro paso bajo por tanto es importante implementar el filtro paso bajo de orden 4 para obtener de forma global un filtro de orden 5. Antes de implementar el filtro se toma una muestra de ECG para hacer un diagrama de Fourier y observar el espectro de frecuencias.

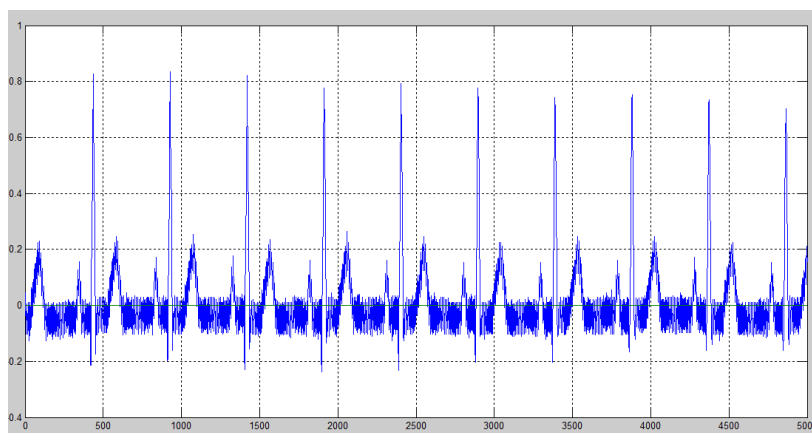


Figura 62. Señal captada con el simulador de ECG y el prototipo de ECG.

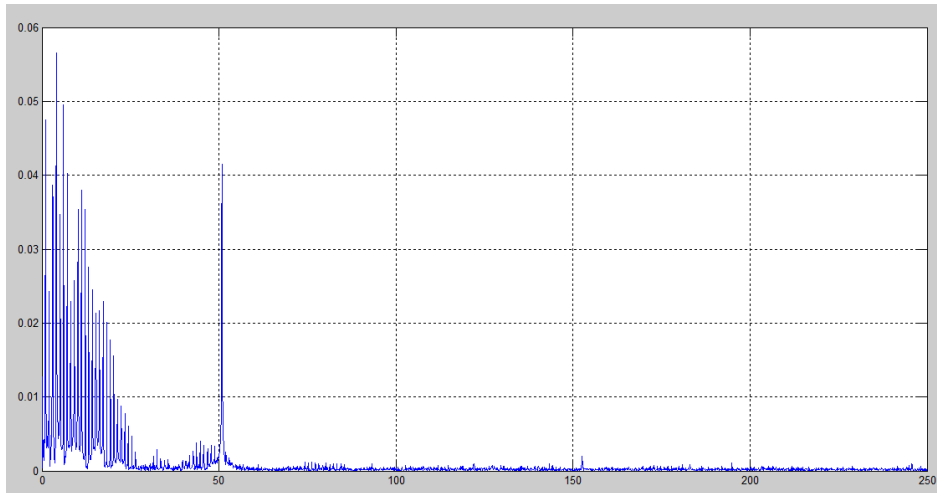


Figura 63. Diagrama de Fourier de la señal captada con el prototipo de ECG.

Comparando el diagrama de Fourier de la señal con el filtro de 5 orden y el de orden 1 se observa que aparecen frecuencias superior a 100 Hz por tanto se confirma la necesidad de implementar un filtro digital de orden 4. Además dentro del margen (0.1 a 100 Hz) de frecuencias naturales del ECG observamos un pico en la frecuencia de 50 Hz, este pico es la interferencia de la red eléctrica por tanto es importante eliminarlo implementando un filtro Notch.

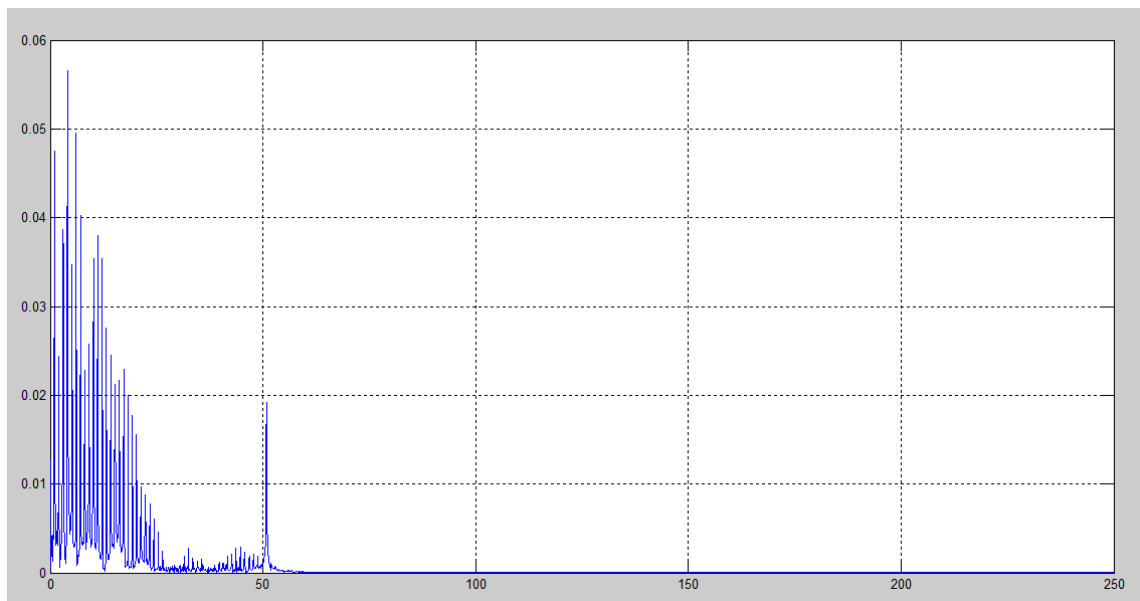


Figura 64. Diagrama de Fourier de la señal captada con el prototipo y filtrada digitalmente con un filtro paso bajo de 4 orden y frecuencia de corte 100 Hz.

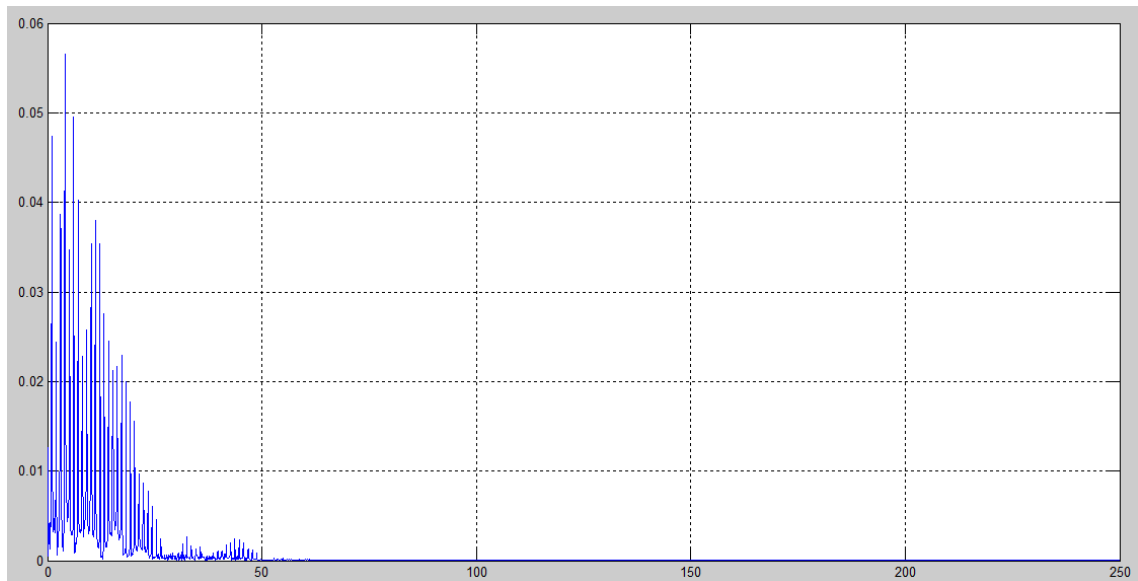


Figura 65. Señal captada con el prototipo de ECG y filtrada digitalmente con un filtro paso bajo de 4 orden y frecuencia de corte de 100 Hz y filtro Notch de frecuencia central de 50 Hz.

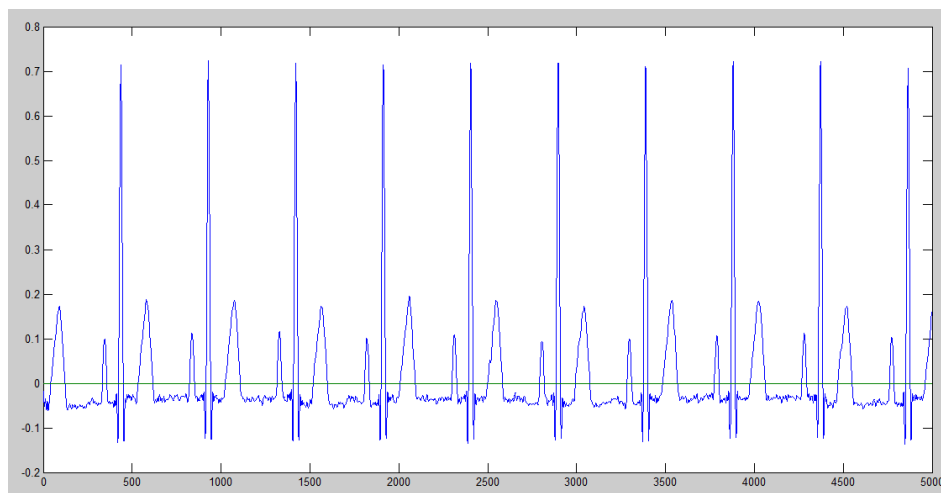


Figura 66. Señal obtenida con el simulador de ECG y el prototipo de ECG y filtrada digitalmente.

Como se está trabajando en Matlab sin ser a tiempo real, es decir almacenando los datos en un vector se utilizan unas funciones que proporciona Matlab sencillas de implementar. Para el filtro paso alto se utiliza la función 'butter' por tanto el filtro será de tipo Butterworth. A la función se le debe pasar el valor del orden del filtro, la frecuencia de corte entre 0-1 y el tipo de filtro (paso alto o paso bajo). Para definir la frecuencia de corte entre 0 y 1 se necesita la frecuencia de muestreo ya que, en digital, la variable tiempo desaparece y lo que se tiene son muestras que dependen de la frecuencia de muestreo, el parámetro que se le debe pasar es la frecuencia de corte partido la mitad de la frecuencia de muestreo y así obtener un valor entre 0 y 1, si el valor no estuviera en este intervalo no sería posible implementar y se deberían modificar los valores de frecuencia de muestreo o frecuencia de corte. La función devuelve dos parámetros que son el numerador y denominador de la función de transferencia del filtro diseñado que posteriormente se aplicaran para implementar el filtro

Luego para diseñar el filtro Notch se utiliza la función 'irrnotch' esta función también devuelve dos valores que corresponden a numerador y denominador de la función de transferencia, a esta función se le deben pasar los parámetros frecuencia central y el ancho de banda, la frecuencia central se calcula igual que en el filtro paso bajo y el ancho de banda viene definido por la frecuencia central (valores entre 0 y 1) partido el coeficiente de calidad del filtro. La principal ventaja de los filtros digitales es que se pueden implementar filtros muy grandes sin limitaciones debido a los componentes electrónicos (condensadores y resistencias) pero añaden un retraso en la función de transferencia si se trabaja a tiempo real, en este caso se trabaja "offline" por lo tanto no es una desventaja.

<code>Fs = 500;</code>	Frecuencia de muestreo
<code>T = 1/Fs;</code>	Periodo de muestreo
<code>fc=100;</code>	Frecuencia de corte del filtro paso bajo
<code>n=4;</code>	Orden del filtro paso bajo
<code>Wn=fc/(fs/2);</code>	Parámetro para la frecuencia de corte
<code>[b a] = butter(n,Wn,'low');</code>	Numerador y denominador del filtro paso bajo
<code>fcn=50;</code>	Frecuencia de corte del filtro Notch
<code>Wo=fcn/(fm/2);</code>	Parámetro para la frecuencia central
<code>BW=Wo/10;</code>	Ancho de banda del filtro
<code>[b2 a2]=irrnotch(Wo,BW);</code>	Numerador y denominador del filtro Notch
<code>x1=filtfilt(b,a,data1);</code>	Aplicar filtro paso bajo
<code>x2=filtfilt(b2,2,x1);</code>	Aplicar filtro Notch

Por último para aplicar los filtros se utiliza la función 'filtfilt'. A esta función se le pasan el numerador y denominador obtenido en las funciones de diseño del filtro y el vector donde está contenida la señal.

5.6.4. Monitorización de la señal

Para la monitorización de la señal se utiliza Matlab, en el apartado anterior se explica como se procesa la señal, una vez procesada tan solo hay que mostrarlo por pantalla.

Un paso previo a mostrar por pantalla es hacer que el vector esté en función del tiempo, el dato va llegando con cada muestra y el programa no sabe que tiempo tiene entre muestras, por eso se define el periodo de muestreo y se multiplica las coordenadas X del vector por el periodo de muestreo de esta forma cuando se dibuje la gráfica se pueda ver el tiempo de la función.

<code>while contador<=numero_muestras</code>	Bucle que depende del tamaño de la muestra
<code> b=T*1000+b;</code>	Variable que cuenta el valor del tiempo
<code> X(contador)=b;</code>	Vector tiempo
<code> contador=contador+1;</code>	Contador del bucle
<code>end</code>	Fin del bucle

Para muestrear por pantalla se abren 3 figuras que identifican que derivada se está observando:

<code>figure('Name','Serial communication: Matlab + Arduino DIII')</code>	Título de la figura
<code>title('SERIAL COMMUNICATION MATLAB+ARDUINO AVR');</code>	Título del gráfico
<code>xlabel('Número de muestra');</code>	Nombre del eje X
<code>ylabel('Voltaje (V)');</code>	Nombre del eje Y
<code>grid on;</code>	Activar rejilla
<code>ylim([-0.5 1.5]);</code>	Límites del eje Y
<code>xlim([0 T*numero_muestras]);</code>	Límites del eje X
<code>plot(tiempo,voltaje1);</code>	Dibujar gráfica, eje X tiempo eje Y datos voltaje1

El código de arriba es un ejemplo para la derivada aVR, esto se modifica de forma automática dependiendo de que derivada se esté procesando.

6. Resultados

6.1. Prueba del prototipo

Antes de montar el circuito impreso final es importante hacer un prototipo y comprobar el correcto funcionamiento. Para ello se ha utilizado una placa prototipo. Se ha montado la placa siguiendo el esquema de los apartados anteriores.

Para comprobar el funcionamiento se ha conectado los electrodos a un individuo y medido las derivadas D1, D2 y aVR.

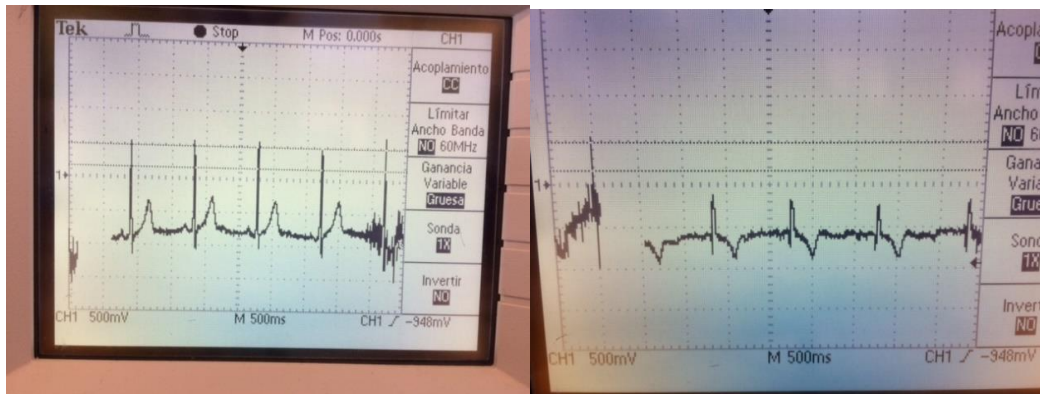


Figura 67. Señales capturadas con el prototipo de ECG, izquierda primera derivada, derecha segunda derivada.

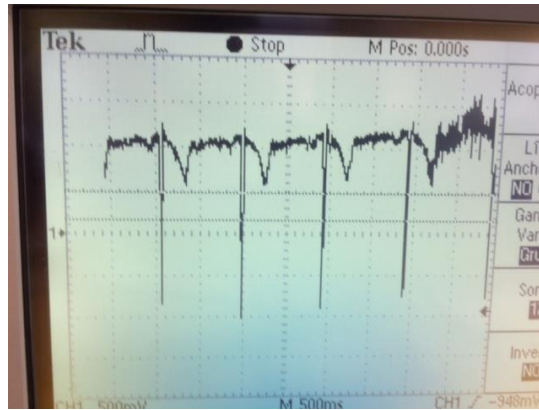


Figura 68. Derivada aVR capturada con el prototipo de ECG.

6.2.Montaje del circuito impreso

Una vez se tiene el diseño completo se monta un circuito impreso para comprobar el funcionamiento real del electrocardiograma. Para ello -utilizando el esquema diseñado en Ares- se imprime en un film transparente, pintando de negro lo que corresponde al cobre (fig.69).



Figura 69. Arriba, placa para circuito impreso cubierta de un film protector contra luz ultravioleta, abajo circuito impreso sobre film transparente.

Una vez se tiene impreso se superponen las dos caras y se dejan bien pegadas pero con un lado abierto para introducir la placa de cobre. Esta placa de cobre está recubierta de un material que reacciona a la luz ultravioleta. Por eso se cubre la placa dentro del film para tapar la superficie deseada de negro y protegerla de la luz ultravioleta y se introduce en la insoladora durante unos 200 segundos, esta máquina emite luz ultravioleta directamente sobre la placa de cobre, haciendo reaccionar la superficie no cubierta, una vez terminado el tiempo se extrae de la insoladora y se obtiene la placa con el circuito marcado. La superficie que no está marcada, es decir la que ha estado en contacto con la luz ultravioleta, se puede disolver con una disolución de sosa cáustica. De esta forma se elimina la película que cubría el cobre de la superficie no deseada.

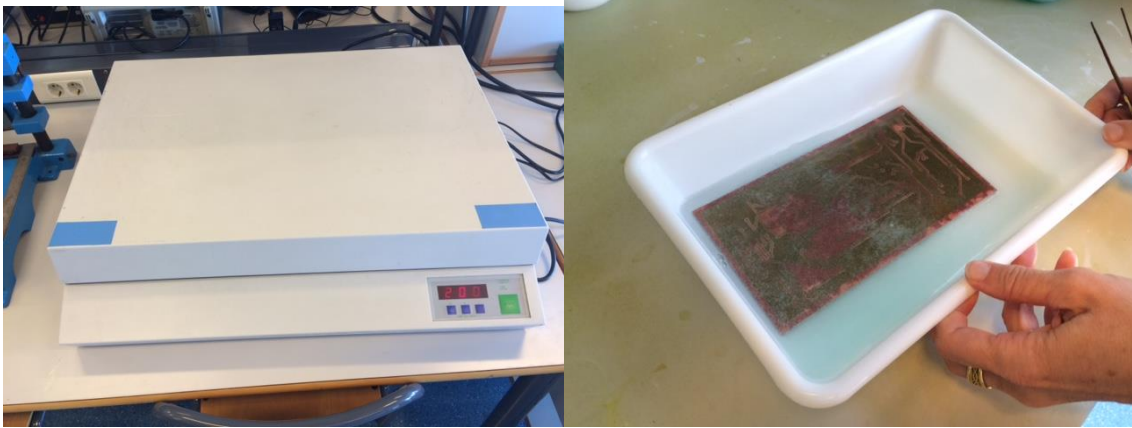


Figura 70. Izquierda insoladora para imprimir circuitos impresos, derecha circuito sumergido en la solución acida para disolver el cobre sobrante.

Ahora se tiene la placa impresa sobre el cobre, dejando las pistas deseadas cubiertas y el resto de superficie descubierta. La placa se sumerge en una mezcla de ácido clorhídrico y perborato sódico, esta mezcla disuelve el cobre que no está protegido, dejando el plástico de la placa. Tras un par de minutos queda disuelto todo el cobre sin cubrir. El último paso es eliminar con alcohol la película que protegía el cobre y limpiar bien la placa para una correcta soldadura.

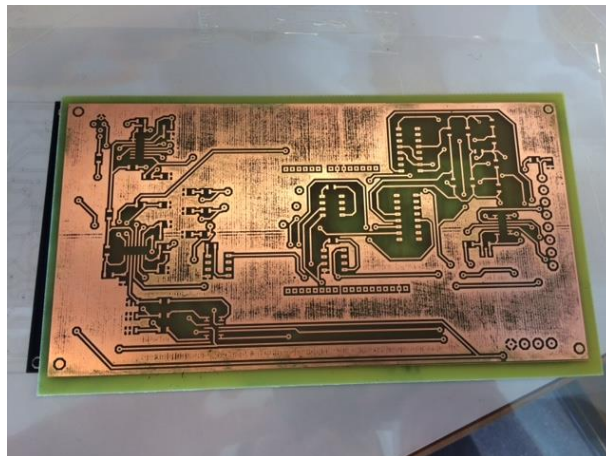


Figura 71. Circuito impreso resultante.

Una vez impresa la placa el siguiente paso es taladrar los agujeros pasantes de la placa para poder soldar los componentes de agujero pasante. Antes de empezar a soldar se sueldan los cambios de plano con un trozo de cable de aluminio soldado a ambas caras. Una vez soldado es importante comprobar que la placa está bien impresa y que las pistas estén bien conectadas usando un multímetro. Por último se terminan de soldar el resto de componentes.

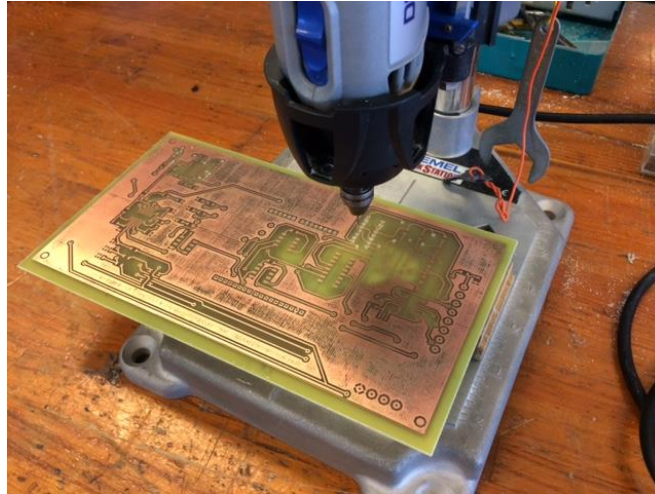


Figura 72. Proceso de taladrado del circuito impreso.

Una vez hechos los agujeros, se empieza la soldadura de los componentes SMD o de agujero pasante y cambios de plano. Por último con un multímetro se comprueba que todas las pistas y soldaduras están bien hechas.

6.3. Pruebas y resultados

Con la primera prueba del primer circuito impreso, aparecen fallos de diseño. Como los condensadores de 1 μF que son demasiados grandes para el diseño implementado, para solucionar el problema se ha taladrado dos agujeros nuevos fuera de las pistas de cobre y soldado con unos cables de aluminio (fig. 73, cables de color verde). Otro problema es un error en la conexión de los pines del AD620, para solucionarlo se han cortado las pistas de cobre con un bisturí y se han soldado de nuevo las pistas con un cable de aluminio.

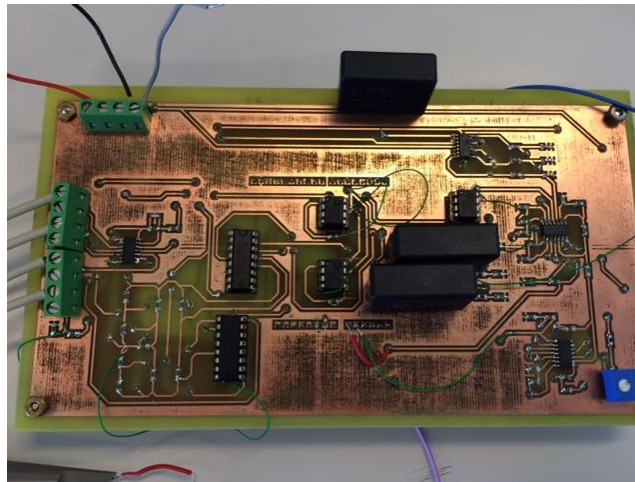


Figura 73. Circuito impreso soldado con los ajustes necesarios para el correcto funcionamiento.

Para la detección de errores se ha ido estudiando el circuito impreso por partes, conectando los componentes al prototipo implementado en la placa de prototipos, ya que la señal de ECG es del orden de mili voltios siendo imposible captarla con el osciloscopio. Para conectar el circuito impreso y el prototipo se han soldado cables de aluminio a las pistas de

cobre. Se ha puenteado los multiplexores para simplificar el circuito y reducir los posibles fallos (fig. 74).

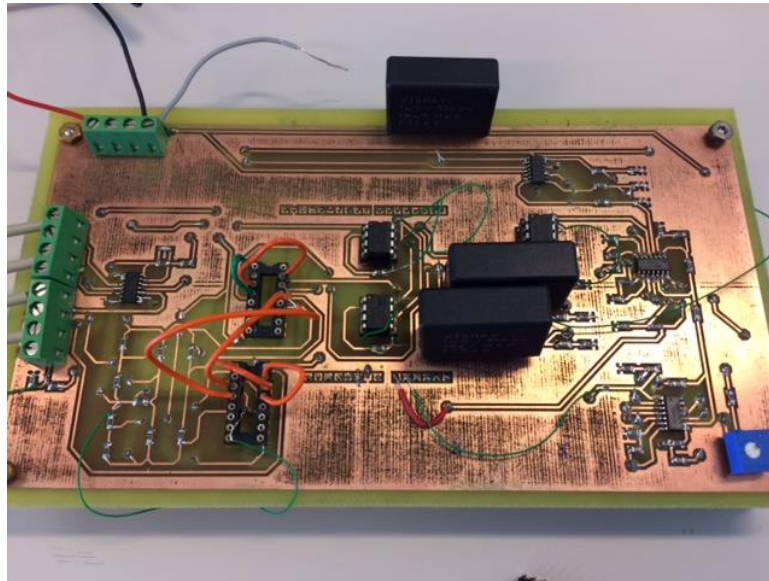


Figura 74. Montaje del circuito impreso con los multiplexores puenteados.

Con los errores solucionados, se ha modificado el esquema y actualizado el diseño (en el apartado 5, los diseños están actualizados y sin errores).

Una vez implementado y arreglado el circuito impreso se han hecho las primeras pruebas con Arduino y el ECG diseñado:

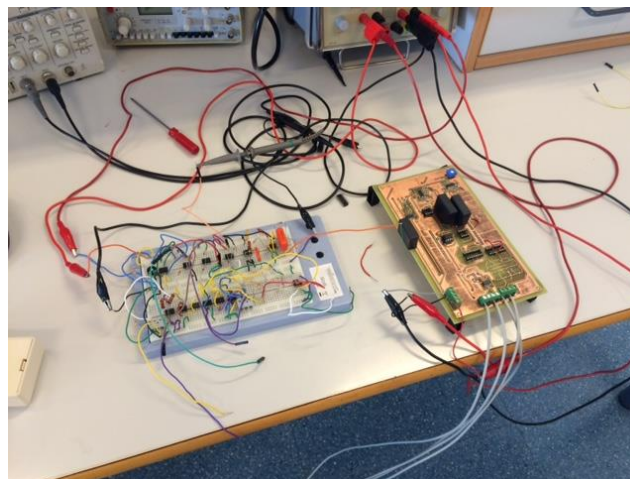


Figura 75. Montaje con el prototipo y el circuito impreso para comprobar el correcto funcionamiento.

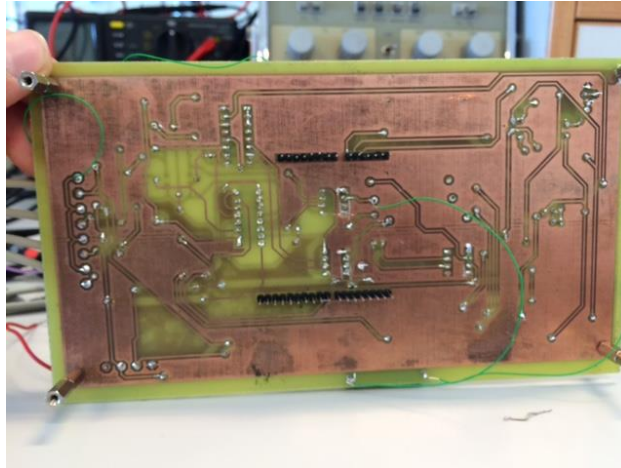


Figura 76. Cara inferior del circuito impreso con los ajustes necesarios.

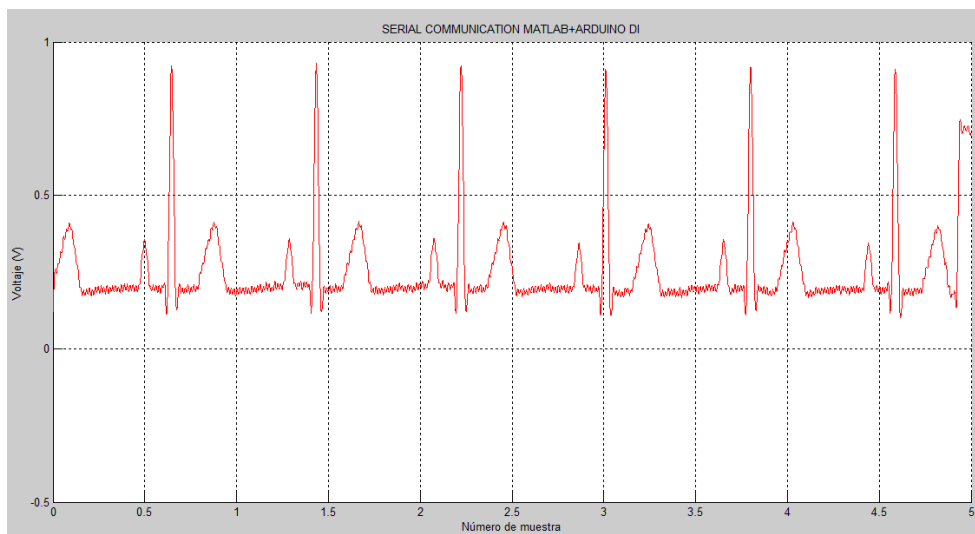


Figura 77. Primera derivada.

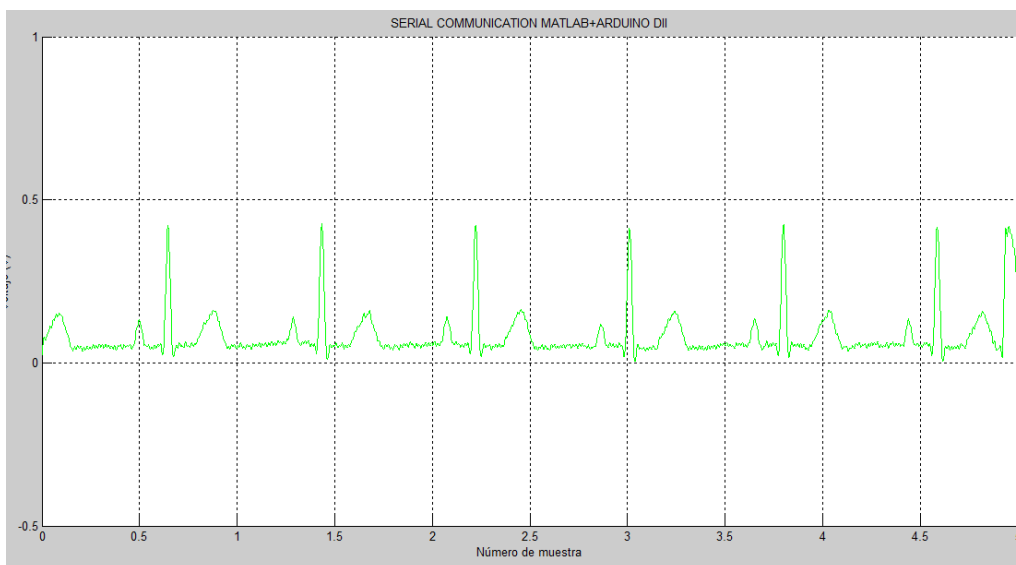


Figura 78. Segunda derivada.

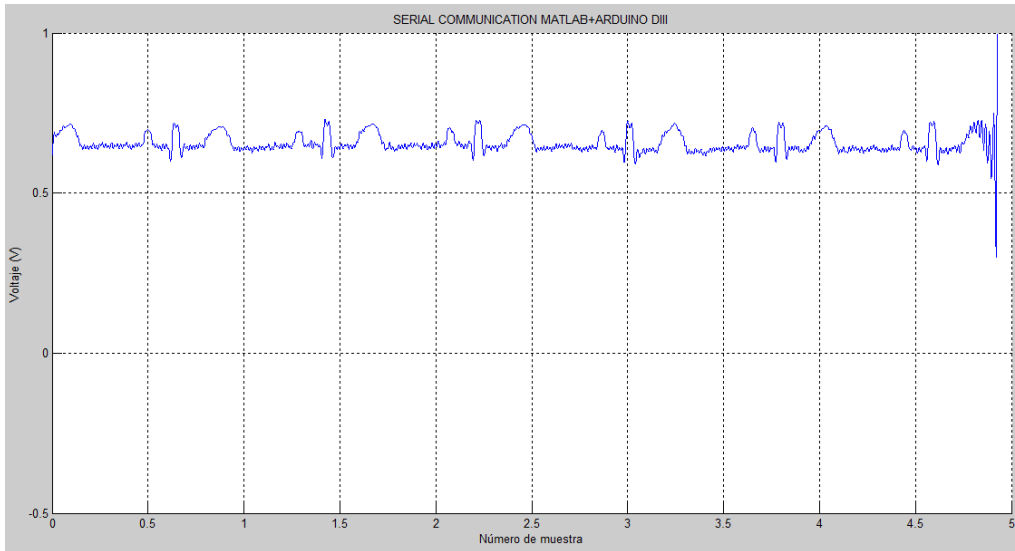


Figura 79. Tercera derivada.

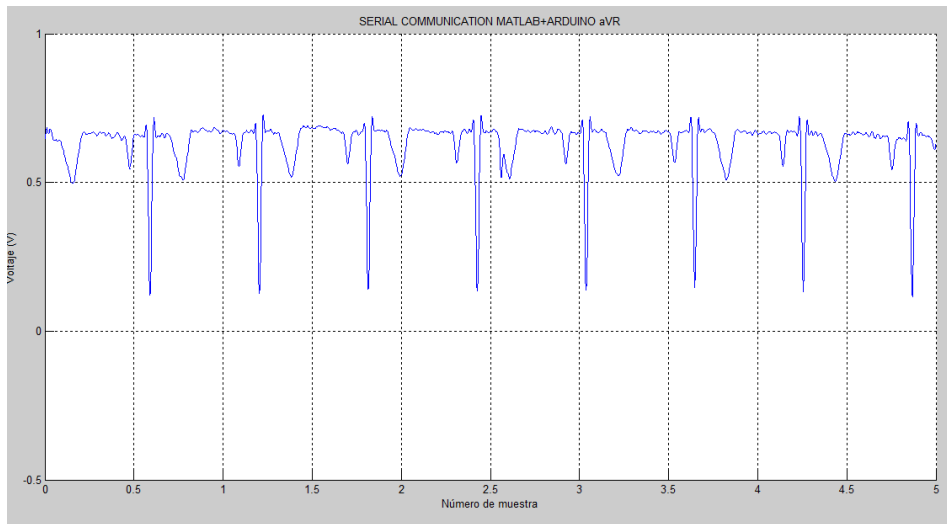


Figura 80. Derivada aVR

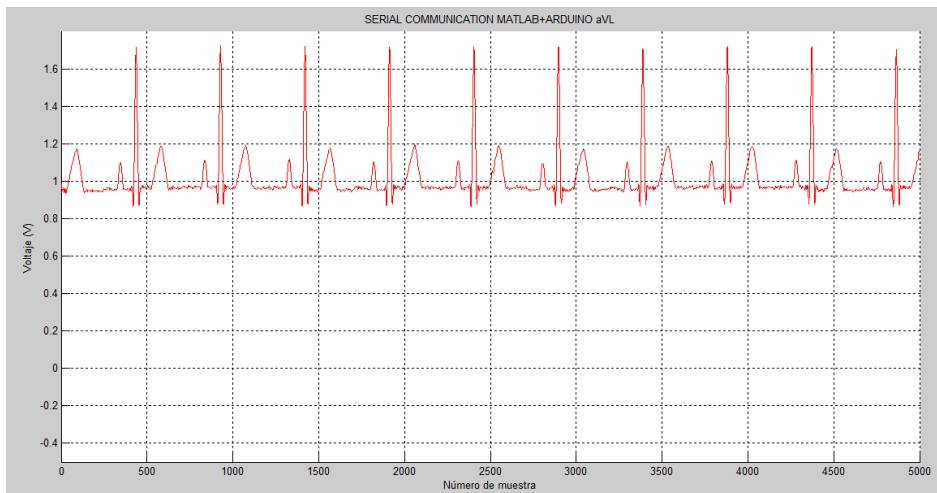


Figura 81. Derivada aVL.

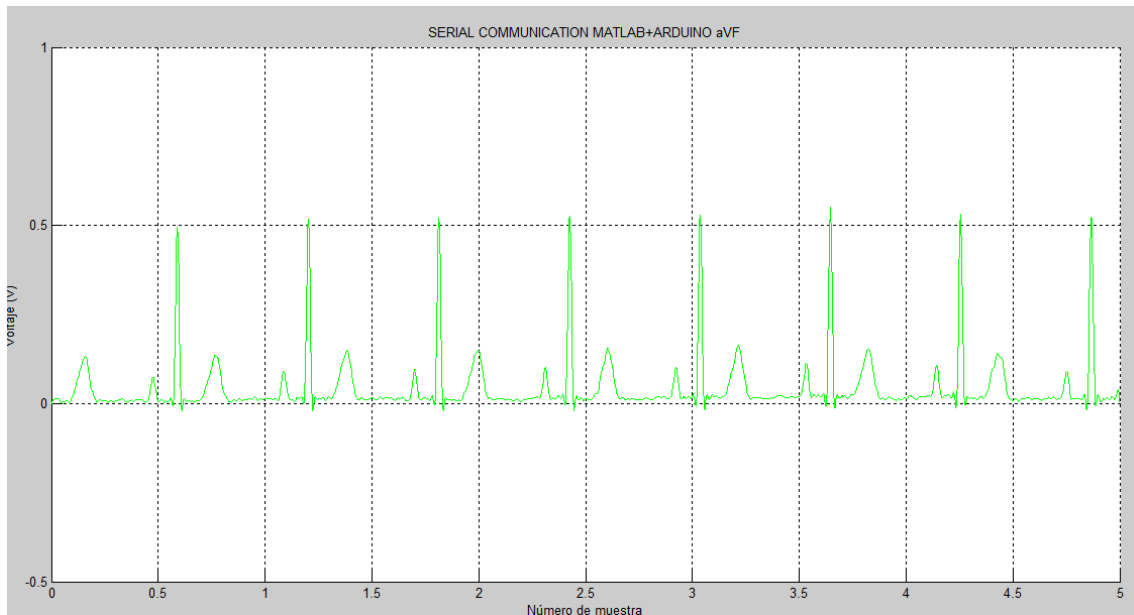


Figura 82. Derivada aVF.

Las figuras desde la 77 a la 82 corresponden a las derivadas que proporciona el electrocardiógrafo diseñado, los resultados obtenidos se corresponden con los esperados.

Por último se ha implementado un jumper para conectar los pines 9 y 10 o 10 y 11, para que el usuario pueda conmutar entre las tres derivadas principales y las tres derivadas aumentadas. El pin 9 está a nivel alto de forma permanente y el pin 11 a nivel bajo de forma permanente, de esta forma con el jumper se pueden conectar los pines 9 y 10 o 10 y 11. Arduino lee el nivel digital del pin 10 y a partir pone a nivel alto o bajo el pin 8 que hace que el multiplexor conmute entre las dos posiciones programadas.

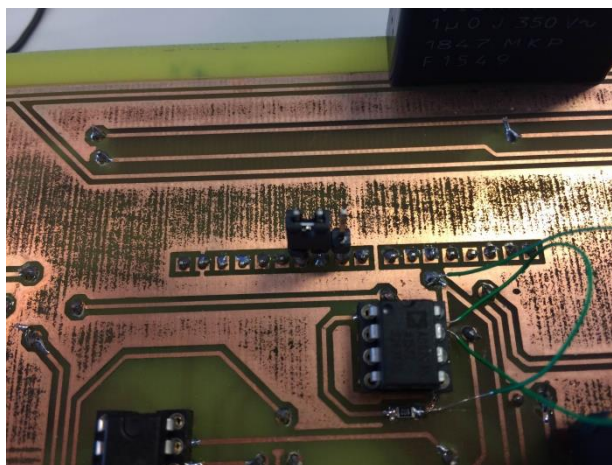


Figura 83. Jumpers selectores, contando desde la izquierda, primer y segundo conecta las derivadas aumentadas, segundo y tercero conecta las derivadas principales.

6.4. Manual de usuario

El proyecto está dividido en dos bloques, software y hardware. El software está formado por el programa que se carga en la placa Arduino y el script de Matlab.

El primer paso es cargar el programa ECG3leads a la placa Arduino, una vez cargado el programa, se pasa a la segunda parte del software, que es el programa de Matlab. En el script sólo que hay que modificar el puerto donde está conectado el Arduino (línea 6 del código de Matlab) y también se puede modificar el tiempo que se desea muestrear la señal (línea 16 del código de Matlab).

Una vez modificado y cargado el programa en la placa Arduino, se conecta la placa Arduino al Shield de ECG y en el Shield se conectan los cables de los electrodos, desde arriba se conecta en orden (fig. 85): Electrodo brazo derecho, pantalla, electrodo brazo izquierdo, pantalla, electrodo pierna derecha, pantalla, electrodo pierna izquierda y pantalla.

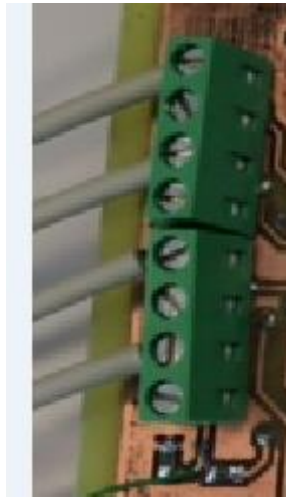


Figura 85. Colocación de los electrodos.

Para alimentar la placa se deben conectar 3 cables a la placa, tensión positiva, tensión negativa y masa. Estos se colocan en los conectores de la placa que hay en la parte superior izquierda (fig. 85) el orden empezando desde la derecha es: masa, tensión negativa, tensión positiva y por último queda una ranura sin usar.



Figura 85. Colocación de los cables de alimentación de la placa.

Con el jumper de la placa (fig. 83) se puede seleccionar las derivadas que se desean observar, si se conecta el pin 1 y 2 del jumper se activan las derivadas aumentadas y conectando los pines 2 y 3 se activan las derivadas principales. Con el jumper seleccionado, se conecta el shield a la fuente de alimentación, el usb a la placa y al ordenador y ya se puede ejecutar el script. Segundos después aparecerá por pantalla las 3 graficas correspondientes a las derivadas seleccionadas y su correspondiente nombre.

Del sistema completo se conocen distintos errores informáticos:

-La lectura del puerto se detiene porque "valorADC(1)=0". Este problema se soluciona volviendo a ejecutar el script.

-No se puede leer el puerto USB. Esto se puede deber a varios motivos, el primero y más común es que el puerto USB está siendo utilizado, por tanto si se desconecta y se vuelve a conectar el problema desaparece, en el caso de que el error permaneciera habría que revisar en que puerto está conectado Arduino (COM1, COM2, COM3...) y comprobar que en el script esté el puerto correcto.

-Debido al filtro paso alto con un condensador de alto valor, se debe esperar entorno a unos 10 segundos para que la señal sea visible correctamente.

7. Conclusiones

7.1. Conclusiones

En este proyecto se han conseguido los objetivos planteados, se ha diseñado un electrocardiógrafo que puede captar 6 derivadas de las 12 derivadas. La señal logra tener las características necesarias, ganancia de 1000 V/V y un ancho de banda de 0,1 a 100 Hz. La calidad es buena aunque se puede mejorar con el uso de filtros de mayor precisión y calidad.

El segundo objetivo del proyecto era lograr que éste usara un hardware y un software abiertos para que cualquier individuo pueda entender y mejorar el sistema, existen muchas líneas de desarrollo futuro que gracias a su hardware y software abierto pueden seguirse con facilidad.

7.2 Líneas de desarrollo futuro

Este proyecto está pensado para poder ser mejorado y continuado, su objetivo es académico pero aún tiene etapas que pueden ser desarrolladas. El diseño del proyecto tan solo consta de 6 derivadas de las 12 que se utilizan, con un pequeño ajuste en el diseño permite que se pueda añadir otro Shield y completar las 6 derivadas precordiales restantes.

La etapa de seguridad es comentada debido a su gran importancia en los instrumentos clínicos, como ya se ha dicho, este proyecto es para uso académico, sin embargo el diseño permite ser adaptado para posibles extensiones que permitan desarrollar una etapa de seguridad eléctrica que cumpla las normativas adecuadas.

Como se comenta en el apartado 4.6 para simplificar el diseño del programa se trabaja off-line, una línea de desarrollo futuro es actualizar el programa o utilizar uno nuevo que

permita la monitorización a tiempo real de las derivadas y el método de monitorización también puede ser modificado utilizando distintos métodos comentados en el apartado 4.6 o nuevas formas que puedan ser implementadas.

En este proyecto no se centra en el diseño de una fuente de alimentación adaptada para el ECG, por tanto esta parte queda abierta a posibles desarrollos, siguiendo unas pautas definidas por los componentes electrónicos del sistema se puede diseñar una fuente de alimentación externa, interna o autónoma para la alimentación del sistema.

Por último a partir de los fallos conocidos se puede desarrollar soluciones a estos problemas ya que este trabajo académico está abierto a estudios y mejorar por cualquier usuario.

Documento II. Presupuesto del proyecto

1. Características del proyecto

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

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

2. Costes

2.1. Coste de los componentes y materiales

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

Nombre componente	Cantidad	Precio unidad	Precio total (€)
Resistencia variable bourns serie 2286p 20k	1	1,63	1,63
Resistencia SMD 10 k 0,1% 0,125W 0805	25	0,334	8,35
Resistencia SMD 330k 0,1% 0,125W 0805	5	0,058	0,29
Resistencia SMD 1k 0,1% 0,125W 0805	10	0,308	3,08
Resistencia SMD 953k 0,1% 0,125W 0805	5	0,044	0,22
Resistencia SMD 510k 0,1% 0,125W 0805	5	0,22	1,1
Resistencias SMD 15k 0,1% 0,1W 0805	5	0,282	1,41
Resistencias SMD 18k 0,1% 0,125W 0805	5	0,274	1,37
Resistencias SMD 80k 0,1% 0,125W 0805	5	0,242	1,21
Condensador cerámico multicapa 39pF 0805	5	0,288	1,44
Condensador de película de polipropileno 1uF 350 Vac	3	1,94	5,82
Condensadores cerámicos multicapa 0,1uF	25	0,019	0,475
OPA4340UA (operacional cuádruple)	4	6,57	26,8
OPA4340EA (operacional cuádruple)	4	3,78	15,12
AD620	3	7,88	23,64
Multiplexor CD74HCT4053EE4	10	0,327	3,27
Conectores macho para PCB 2 pines	60	0,074	4,44
Enlace corto RS pro	10	0,324	3,24
Bloque Terminal para PCB	5	1,316	6,58
Electrodos	4	0,85	3,4
Conductor apantallado	4	9,6	38,4
Estaño	0.06	8,46	0,507
Pasta de soldadura	0.02	3,70	0,074
Placa Arduino Uno	1	22,34	22,34
Pulsoxímetro e-health	1	55	55
Shield Arduino e-health	1	75	75
Sensor de flujo de aire e-health	1	25	25
Sensor de repuesta galvánica de la piel e-health	1	30	30
Placa de circuito impreso	1	2,5	2,5
Cable USB	1	2,75	2,75
Gastos de envío	1	9	9
Total			372,94

2.2. Coste de horas de trabajo

El precio de la hora se supone que se paga como si fuera un técnico superior o el profesional correspondiente (coste 12 €/hora)

- Diseño y análisis
 - 300 horas.
- Prototipo
 - 20 horas
- Soldadura
 - 6 horas
- Pruebas
 - 5 horas

Total: 331 (h) x12 (€/h) = 3972 €

2.3. Coste del proceso de desarrollo

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

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

Nombre	Precio	Vida útil/licencia	€/h	Horas utilizado	Total
Matlab	69 €	1 año -> 1785 h	0.038	100 h	3.8 €
Arduino Software	0 €	-	-	-	0 €
Proteus 8	200 €	5 año -> 8925 h	0.0224	20 h	0.45 €
Osciloscopio Tektronis TDS 210	573 €	10.000 h	0.0573	50 h	2.86 €
Soldador JBC rework station	1900 €	10.000 h	0.19	6 h	1.14 €
Fuente de alimentación FAC662B, Promax	388 €	20.000 h	0.019	50 h	0.97 €
Generador de funciones GF-232, Promax	580 €	10.000 h	0.058	2 h	0.116 €

La suma total del coste de amortización es de 9.3 €.

2.4. Presupuesto total

El presupuesto total es la suma del coste del proceso de desarrollo y el coste de componentes y materiales utilizados:

Coste componentes y materiales	Coste horas de trabajo	Coste de proceso de desarrollo	Coste total
372.94 €	3972 €	9.3 €	4354.24 €

Documento III. Pliego de condiciones

1. Definición y alcance del pliego

El objeto de este documento es fijar las condiciones técnicas mínimas que debe cumplir la fabricación del electrocardiograma que no se detalle en los planos, especificando los requisitos de durabilidad, fiabilidad y seguridad. El ámbito de aplicación de este documento se extiende a todos los sistemas eléctricos y electrónicos que forman parte de la instalación. En determinados supuestos se podrán adoptar, por la propia naturaleza del mismo o del desarrollo tecnológico, soluciones diferentes a las exigidas en este documento, siempre que quede suficientemente justificada su necesidad y que no impliquen una disminución de las exigencias mínimas de calidad especificadas en el mismo.

2. Condiciones generales

1. Las condiciones generales del Pliego de Condiciones forman parte del proyecto y por tanto son las normas principales que rigen este proyecto.
2. El diseño del dispositivo ha sido aprobado por el Departamento de Ingeniería Electrónica de la Universidad Politécnica de Valencia.
3. Para la fabricación se deberá seguir las normativas correspondientes y métodos explicados en el apartado de normativa de aplicación.
4. Este proyecto es una guía de diseño de un dispositivo de ECG, existe la posibilidad de mejora y extensión, es un proyecto abierto a mejoras por parte de otros usuarios y se aceptan modificaciones del diseño siempre que cumplan las normativas correspondientes.

3. Normativa de aplicación

Normativa	Asunto
UNE EN 61010-1 UNE EN 61011/A	Requisitos de seguridad de equipos eléctricos de medida, control y su uso en laboratorio.
UNE 20621-2/2C	Circuitos impresos, métodos de ensayo
UNE 20621-7	Especificación par placas impresas de simple y doble cara sin agujeros metalizados
UNE 60902	Técnica de los circuitos impresos. Terminología
ANSI/AAMI ESI-1993	“Safe current limits for electromedical aparatos”. Establece los límites y las técnicas de medida para la medida de las corrientes de riesgo de los aparatos electromédicos en función de la frecuencia, las características de los aparatos y la naturaleza del contacto con el paciente

4. Condiciones particulares de los componentes del sistema

Características que deben tener los componentes para un correcto funcionamiento del sistema

4.1. Resistencias

Ver valor en el esquema eléctrico.

Características principales:

- Resistencias SMD
- Tamaño 0805
- Potencia 0,125W
- Tolerancia 0.1%

4.2. Condensadores

Condensadores 100 nF y 39 pF, características principales:

- Condensador cerámico multicapa
- Condensadores SMD
- Tensión 350 V ac
- Tamaño 0805

Condensadores 1 μ F, características principales:

- Condensador de película de polipropileno
- Condensador de orificio pasante
- Tensión 350 V ac
- Longitud 32 mm
- Profundidad 11 mm
- Altura 21 mm
- Paso del pin 27.5 mm
- Diámetro del pin 0.8 mm

4.3. Amplificadores operacionales

Amplificador operacional cuádruple, características principales

- Montaje superficial
- Encapsulado SOIC
- Tensión de alimentación de 3 a 5 V
- Salida tipo CMOS y Rail to Rail
- Longitud 8.65 mm
- Ancho 3.91 mm
- Ganancia de tensión típica de 124 dB

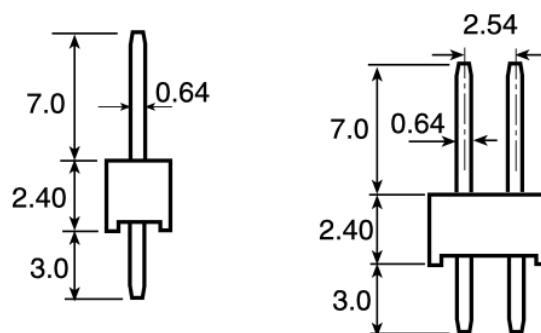
4.4. Amplificador de instrumentación

Amplificador de instrumentación de precisión

- Montaje de orificio pasante
- Encapsulado PDIP
- Alimentación de 3 a 5 V
- Tensión offset de entrada máximo de 0.125 mV
- CMRR mínimo de 73 dB
- Altura 3.3 mm
- Longitud 9.27 mm
- Ancho 6.35 mm

4.5. Conectores macho para PCB

Conector macho para PCB, dimensiones necesarias:



4.6. Bloque Terminal para PCB

Bloque terminal para PCB, características principales:

- Montaje de orificio pasante
- Profundidad 8.2 mm
- Anchura 10 mm
- Longitud 20 mm
- Admite cable macizo hasta 1.5 mm

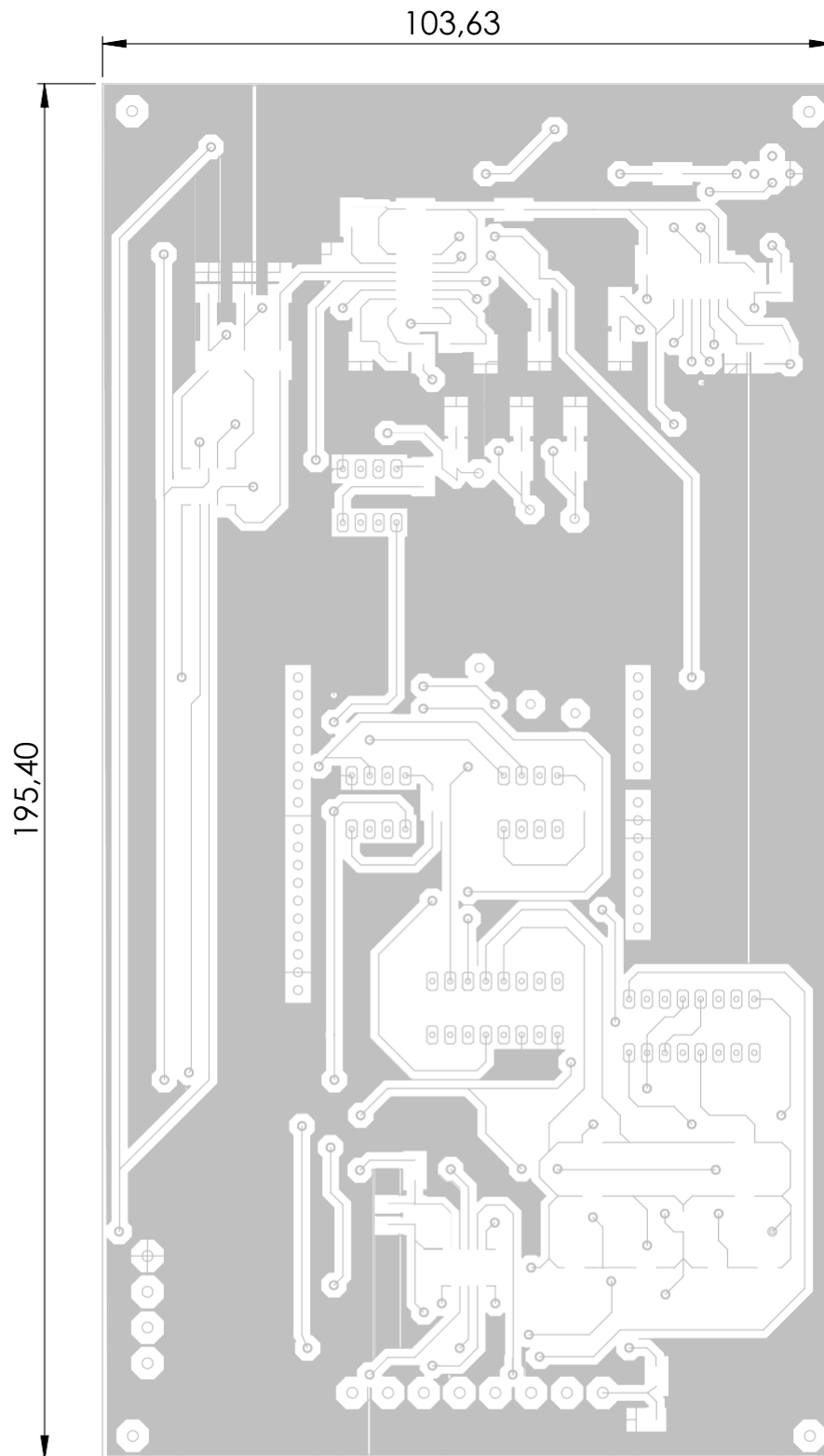
4.7. Multiplexor

Multiplexor analógico, características principales:

- Multiplexor 2 a 1
- Montaje superficial
- Multiplexor analógico
- Alimentación de 3 a 5 V

Documento IV. Planos

- 1. Plano cara superior de las pistas de cobre**
- 2. Plano cara inferior de las pistas de cobre**
- 3. Plano de taladrado**
- 4. Plano de soldadura de los path**
- 5. Plano de soldadura de componentes de montaje superficial**
- 6. Plano de soldadura de componentes de orificio pasante**
- 7. Plano componentes**

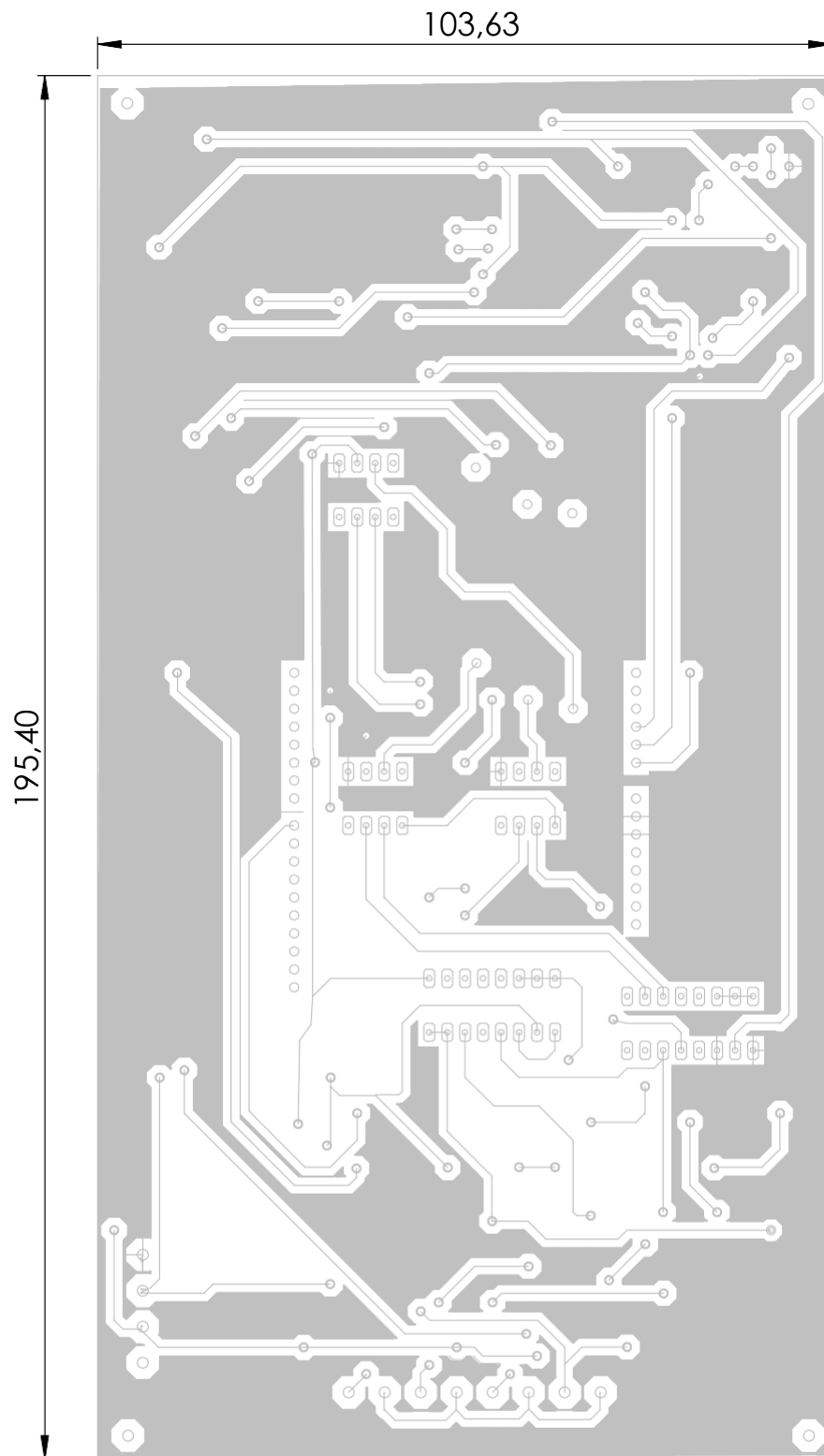


UNIVERSITAT POLITÉCNICA DE VALENCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO

ESTUDIO DE SENSORES BIOMÉDICOS Y DISEÑO DE UN ELECTROCARDIOGRAMA PARA ARDUINO

Autor: Arnau Pérez Talens	Nombre: Plano cara superior de las pistas de cobre	Valencia, Junio 2016	
		Escala 1:1	A4
		Plano N° 1	Página 1 de 1

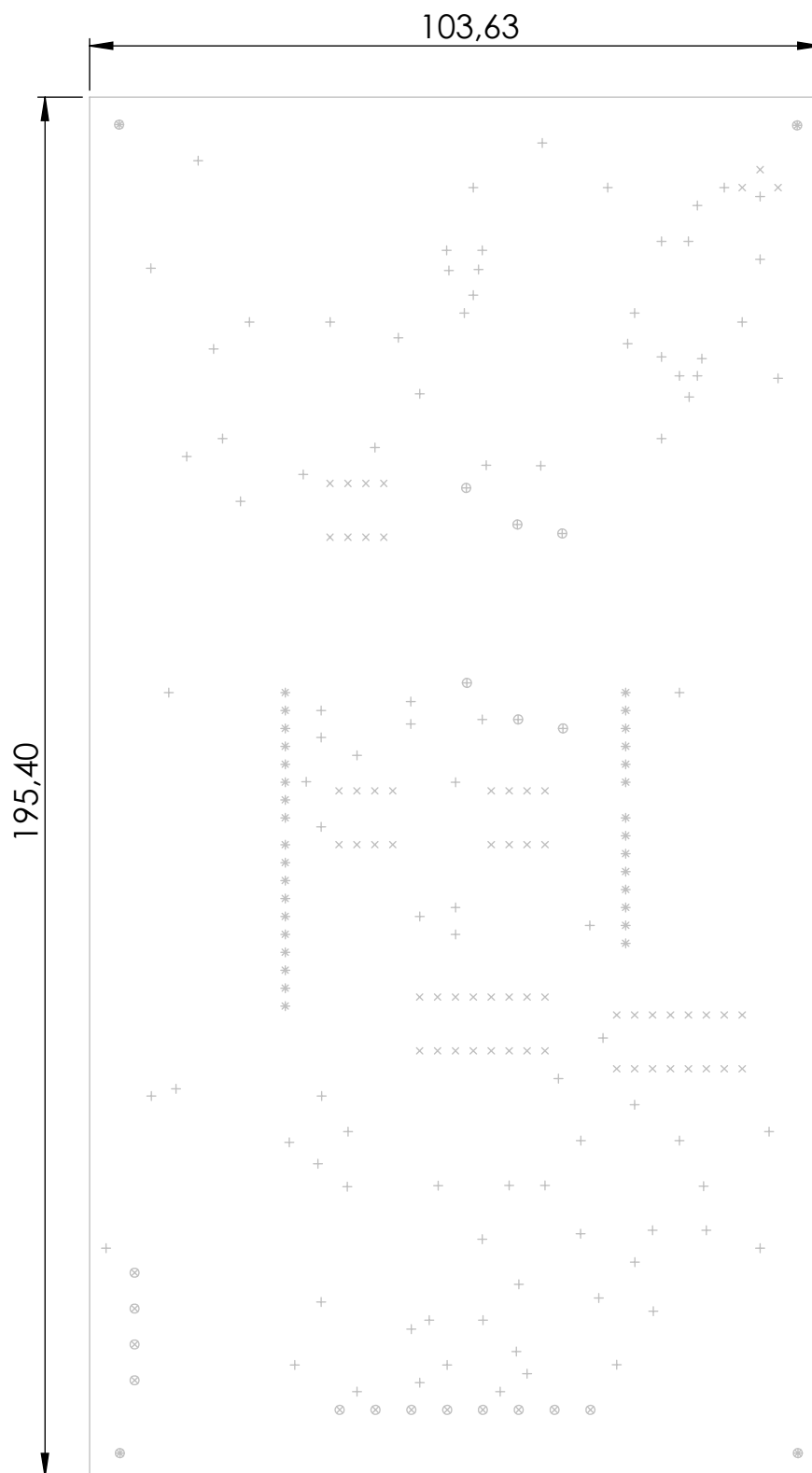


UNIVERSITAT POLITÈCNICA DE VALENCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO

ESTUDIO DE SENSORES BIOMÉDICOS Y DISEÑO DE UN ELECTROCARDIOGRAMA PARA ARDUINO

Autor: Arnau Pérez Talens	Nombre: Plano cara inferior de las pistas de cobre	Valencia, Junio 2016	
		Escala 1:1	A4
		Plano N°2	Página 1 de 1



UNIVERSITAT POLITÈCNICA DE VALENCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO

ESTUDIO DE SENSORES BIOMÉDICOS Y DISEÑO DE UN ELECTROCARDIOGRAMA PARA ARDUINO

Autor:

Arnau Pérez Talens

Nombre:

Plano de taladrado

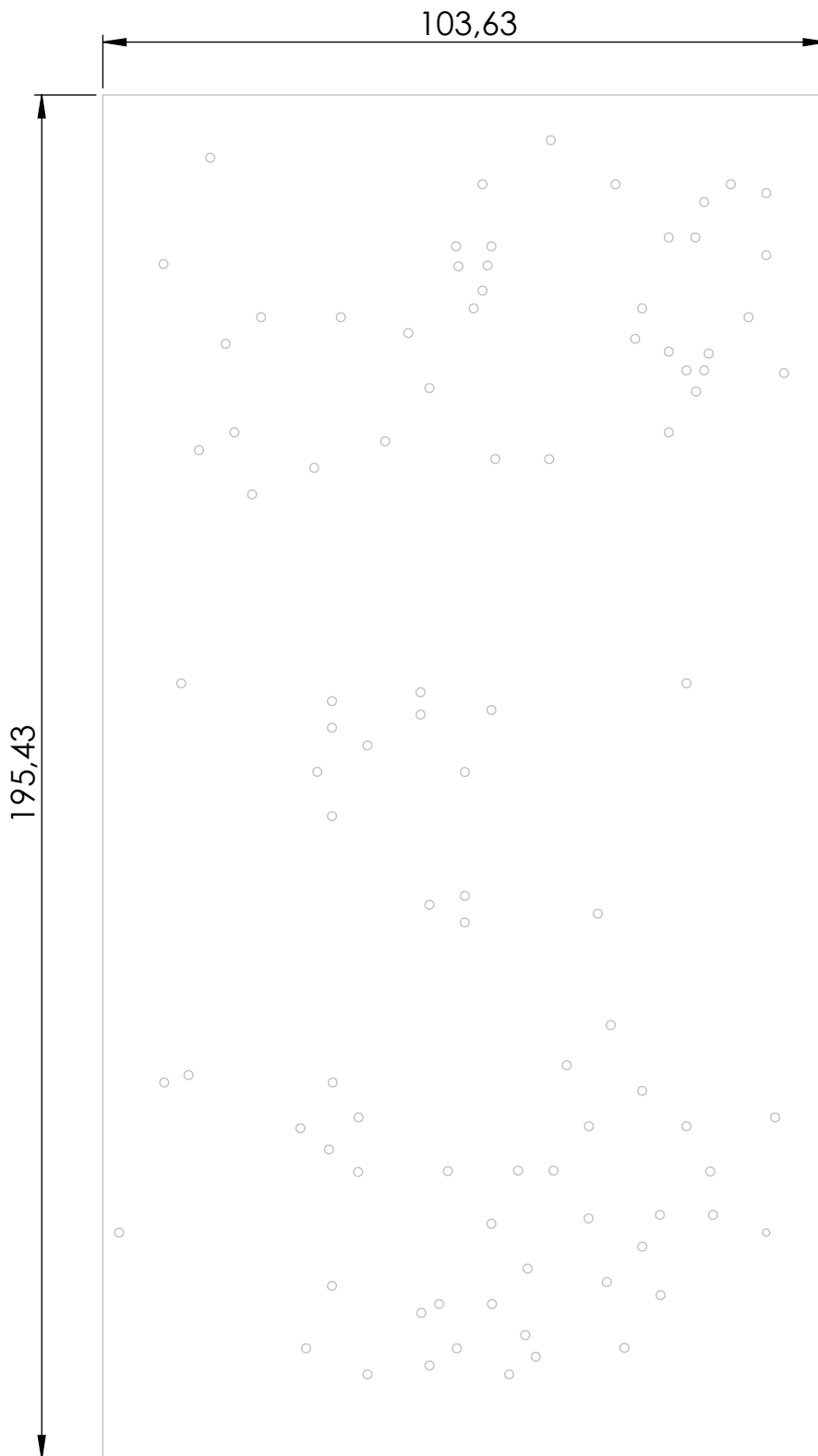
Valencia, Junio 2016

Escala 1:1

A4

Plano Nº 3

Página
1 de 1

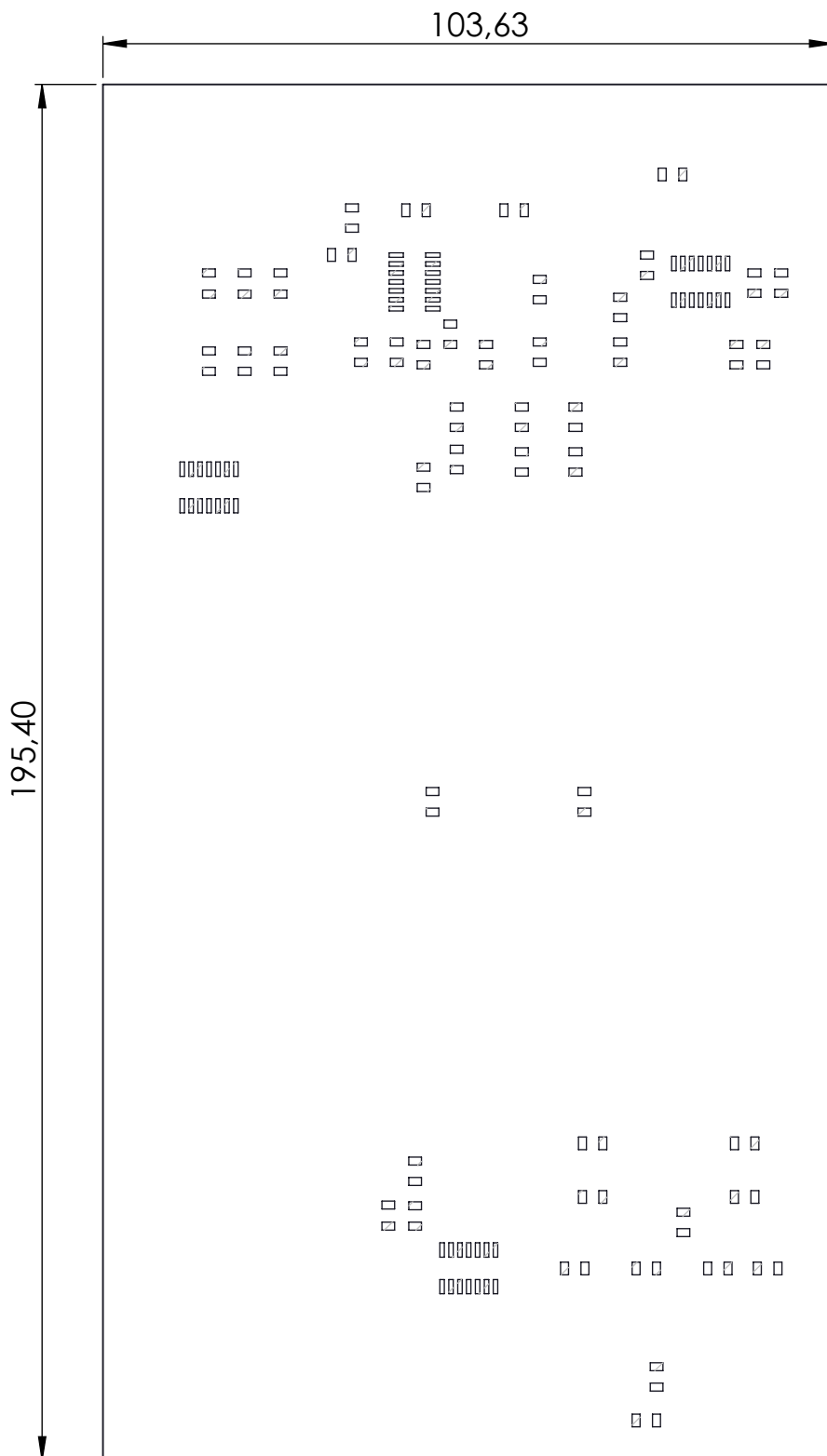


UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO

ESTUDIO DE SENSORES BIOMÉDICOS Y DISEÑO DE UN ELECTROCARDIOGRAMA PARA ARDUINO

Autor: Arnau Pérez Talens	Nombre: Plano de soldadura de los path	Valencia, Junio 2016	
		Escala 1:1	A4
		Plano Nº 4	Página 1 de 1



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO

ESTUDIO DE SENSORES BIOMÉDICOS Y DISEÑO DE UN ELECTROCARDIOGRAMA PARA ARDUINO

Autor:

Arnau Pérez Talens

Nombre:

Plano de soldadura de
componentes de montaje
superficial

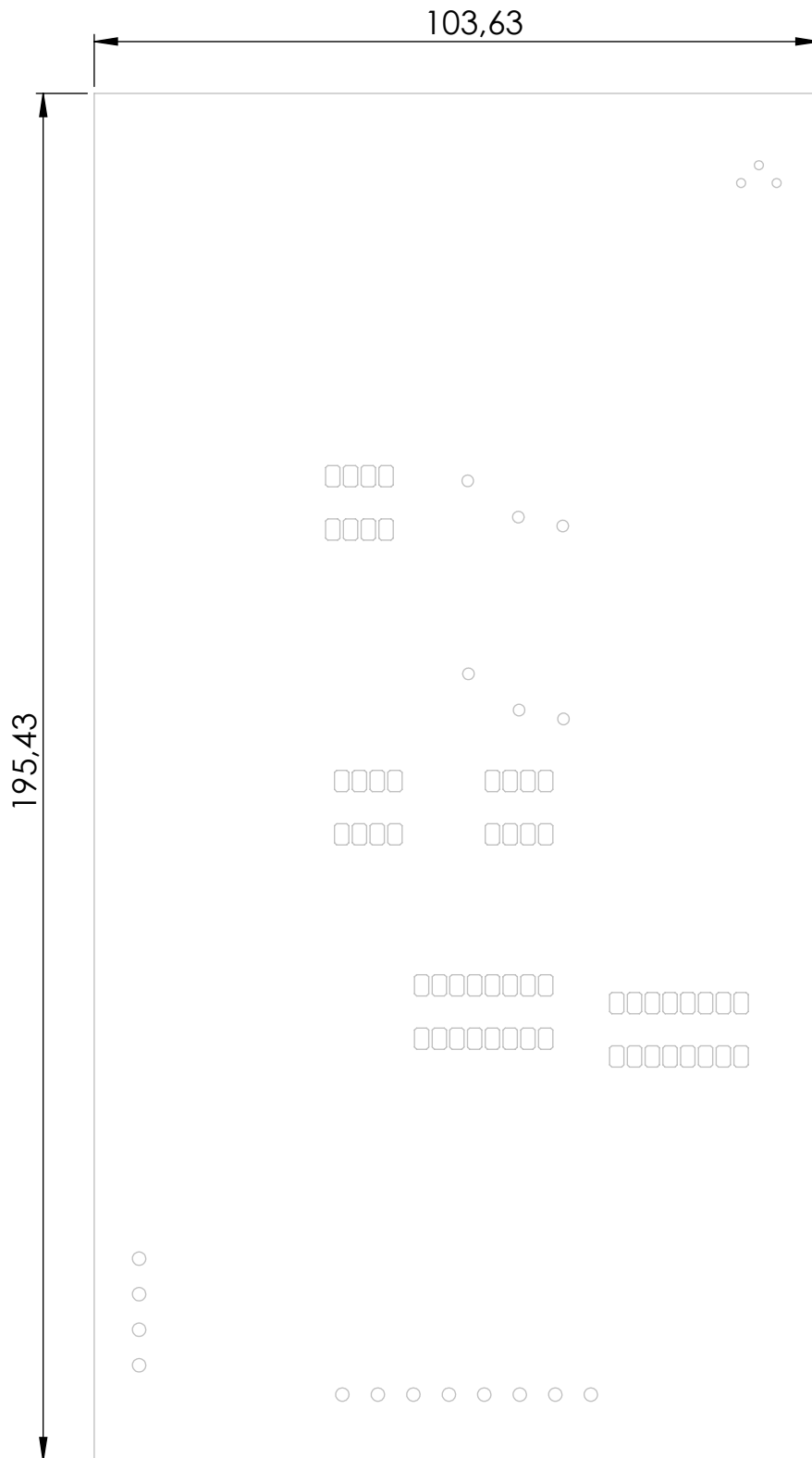
Valencia, Junio 2016

Escala 1:1

A4

Plano Nº 5

Página
1 de 1

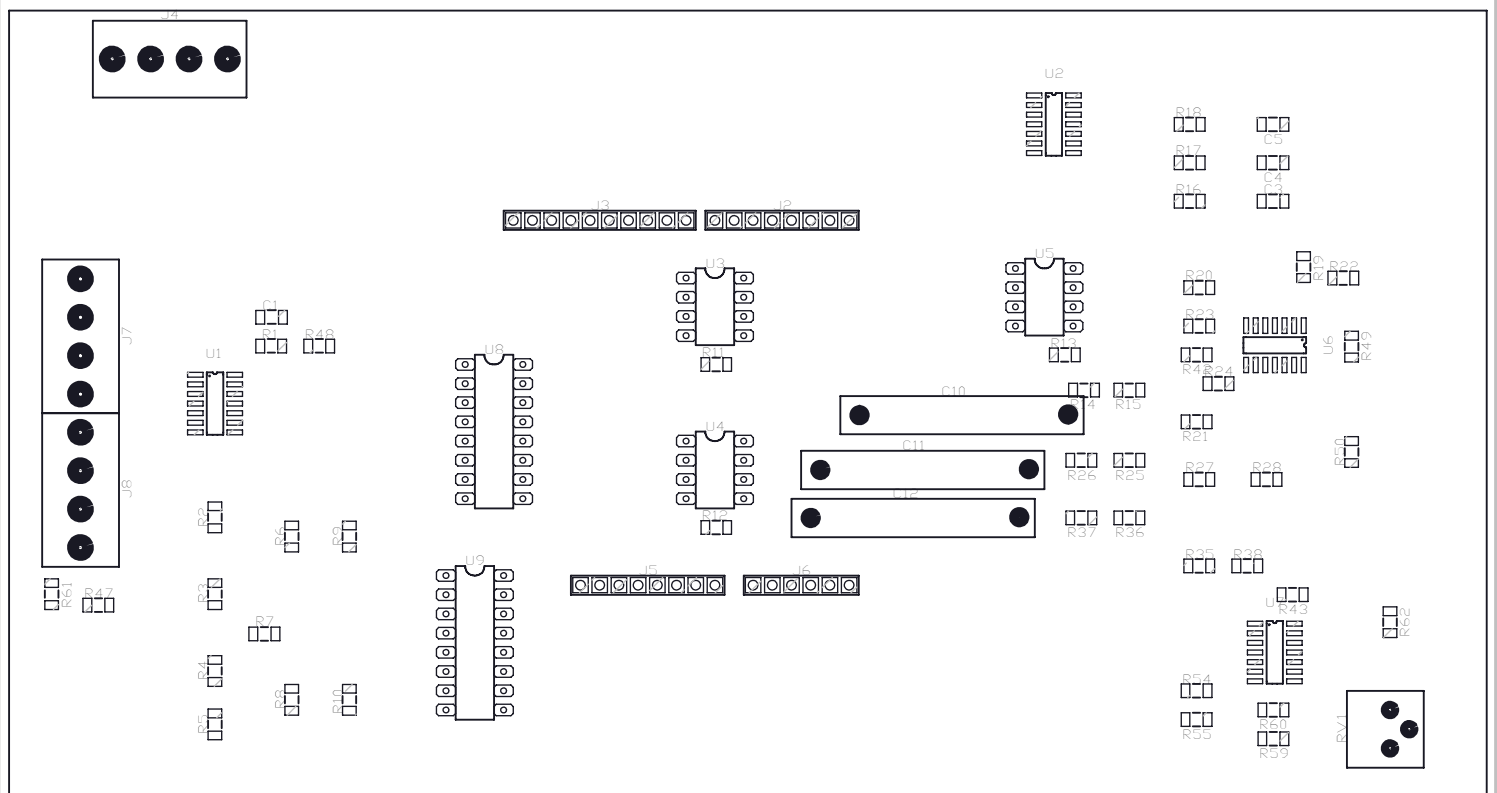


UNIVERSITAT POLITÈCNICA DE VALENCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO

ESTUDIO DE SENSORES BIOMÉDICOS Y DISEÑO DE UN ELECTROCARDIOGRAMA PARA ARDUINO

Autor: Arnau Pérez Talens	Nombre: Plano de soldadura de componentes de orificio pasante	Valencia, Junio 2016	
		Escala 1:1	A4
		Plano Nº 6	Página 1 de 1



J2, J3, J5, J6	Conector macho para PCB
J4, J7, J8	Bloque teriman para PCB
U3, U4, U5	AD620
U1, U2, U6, U7	OPA4340EA
U8, U9	74HTC4053
C10, C11, C12	Condensador 1 uF
C1	Condensador 39 pF
C3, C4, C5	Condensador 100 nF
RV1	Resistencia variable 20 k
R22, R23, R24:	Resistencia 18 k
R16, R17, R18	Resistencia 15 k
R14, R25, R36	Resistencia 510 k
R11, R12, R13, R19, R20, R21	Resistencia 1 k
R1, R48	Resistencia 330 k
R62	Resistencia 80 k
R2, R3, R4, R5, R6, R7, R8, R9, R10, R27, R28, R38, R42, R47, R49, R50, R59, R60, R61	Restistencia 10 k

UNIVERSITAT POLITÈCNICA DE VALENCIA

ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO

ESTUDIO DE SENSORES BIOMÉDICOS Y DISEÑO DE UN ELECTROCARDIOGRAMA PARA ARDUINO

Autor: Arnau Pérez Talens	Nombre: Plano de componentes	Valencia, Junio 2016	
		Escala 1:1	A4
		Plano N° 7	Página 1 de 1

ANEXO I CODIGO FUENTE

Código Matlab

```
clear all;
close all;
clc;

delete(instrfind({'Port'},{'COM3'})); % definir el puerto conectado el
Arduino
PS=serial('COM3');
PS.BaudRate=57600; %Velocidad de comunicacion
warning('off','MATLAB:serial:fscanf:unsuccessfulRead');

fopen(PS);
contador=1;

duracion_muestreo=5 %segundos
fs=500; %sampling rate
T=1/fs; %tiempo de muestreo de arduino
numero_muestras=duracion_muestreo*fs;%numero de muestras

voltaje1=zeros(1,numero_muestras);
voltaje2=zeros(1,numero_muestras);
voltaje3=zeros(1,numero_muestras);
x=0;
constante=0;

while constante==0
variable=fscanf(PS,'%d');
variable
if variable==50000 | variable==40000
constante=1
else
constante=0
end
end

for contador=1:numero_muestras
valorADC=fscanf(PS,'%d');
voltaje1(contador)=valorADC(1)*5/1024;
valorADC=fscanf(PS,'%d');
voltaje2(contador)=valorADC(1)*5/1024;
valorADC=fscanf(PS,'%d');
voltaje3(contador)=valorADC(1)*5/1024;
variable=fscanf(PS,'%d')
x=x+1
end
a=0;
tiempo=zeros(1,numero_muestras);
y=0;
for contador=1:numero_muestras
a=T+a;
tiempo(contador)=a;
y=1;
end
%invertir la señal
```



```

medial=median(voltaje1);
media2=median(voltaje2);
media3=median(voltaje3);
voltaje1inv=-voltaje1+abs(media1);
voltaje2inv=-voltaje2+abs(media2);
voltaje3inv=-voltaje3+abs(media3);
%etapa de filtro
    %fitro paso bajo
fcb=100;
n=4;
Wn=fcb/fs;
[b a] = butter(n,Wn,'low');
[z p k] = butter (n,Wn, 'low');
sys=tf(b,a);
    %filtro notch
fcn=50;
Wo=fcn/(fs/2);
BW=Wo/10;
[b1 a1]=iirnotch(Wo,BW);
%aplicar filtro
voltaje1=filtfilt(b,a,voltaje1inv);
voltaje1=filtfilt(b1,a1,voltaje1inv);
voltaje2=filtfilt(b,a,voltaje2inv);
voltaje2=filtfilt(b1,a1,voltaje2inv);
voltaje3=filtfilt(b,a,voltaje3inv);
voltaje3=filtfilt(b1,a1,voltaje3inv);

%Mostrar por pantalla
if(variable==50000)
figure('Name','Serial communication: Matlab + Arduino DIII')
title('SERIAL COMMUNICATION MATLAB+ARDUINO aVR');
xlabel('Número de muestra');
ylabel('Voltaje (V)');
grid on;
hold on;
ylim([-0.5 1.5]);
xlim([0 T*numero_muestras]);
plot(tiempo,voltaje1);

figure('Name','Serial communication: Matlab + Arduino DI ')
title('SERIAL COMMUNICATION MATLAB+ARDUINO aVL');
xlabel('Número de muestra');
ylabel('Voltaje (V)');
grid on;
hold on;
ylim([-0.5 1.5]);
xlim([0 T*numero_muestras]);
plot(tiempo,voltaje2,'red');

figure('Name','Serial communication: Matlab + Arduino DII')
title('SERIAL COMMUNICATION MATLAB+ARDUINO aVF');
xlabel('Número de muestra');
ylabel('Voltaje (V)');
grid on;
hold on;
ylim([-0.5 1.5]);
xlim([0 T*numero_muestras]);
plot(tiempo,voltaje3,'green');
end
if(variable==40000)
figure('Name','Serial communication: Matlab + Arduino DIII')

```

```

title('SERIAL COMMUNICATION MATLAB+ARDUINO DI');
xlabel('Número de muestra');
ylabel('Voltaje (V)');
grid on;
hold on;
ylim([-0.5 1.5]);
xlim([0 T*numero_muestras]);
plot(tiempo,voltaje1);

figure('Name','Serial communication: Matlab + Arduino DII')
title('SERIAL COMMUNICATION MATLAB+ARDUINO DII');
xlabel('Número de muestra');
ylabel('Voltaje (V)');
grid on;
hold on;
ylim([-0.5 1.5]);
xlim([0 T*numero_muestras]);
plot(tiempo,voltaje2,'red');

figure('Name','Serial communication: Matlab + Arduino DIII')
title('SERIAL COMMUNICATION MATLAB+ARDUINO DIII');
xlabel('Número de muestra');
ylabel('Voltaje (V)');
grid on;
hold on;
ylim([-0.5 1.5]);
xlim([0 T*numero_muestras]);
plot(tiempo,voltaje3,'green');
end
fclose(PS);
delete(PS);
%clear all;

```

Código de Arduino

```
# define FRECUENCIA 500

int ECGbin;

int state=0;

unsigned long time1=0;

unsigned long time2=0;

void setup() {

  Serial.begin(57600);

  pinMode(9, OUTPUT);

  pinMode(11, OUTPUT);

  pinMode(10, INPUT);

  digitalWrite(9,LOW);

  digitalWrite(11,HIGH);

}

void loop()

{

  state=digitalRead(10);

  if(state==HIGH){

    digitalWrite(8,HIGH); // derivadas aumentadas

    Serial.println(50000);

  }else{

    digitalWrite(8,LOW); //derivadas principales

    Serial.println(40000);

  }

  time1=micros();

  ECGbin=analogRead(0);
```

```
Serial.println(ECGbin);  
ECGbin=analogRead(1);  
Serial.println(ECGbin);  
ECGbin=analogRead(2);  
Serial.println(ECGbin);  
time2=micros()-time1;  
delayMicroseconds((1000000/FRECUENCIA)-time2);  
}
```

Bibliografía

Libros consultados a lo largo del proyecto:

- Vicent Torres. "Microcontroladores"
- José Adolfo Gonzalez Vázquez "Introducción a los microcontroladores"
- José María Ferrero Corral, Jose Maria Ferro y de Loma-Sosorio, Antonio Arnau Vives "Bioelectrónica: señales bioeléctricas"
- Rangaraj M.Rangayyan "Biomedical Signal Analysis. A case-study Approach"
- John G. Webster "Medical instrumentation, application and design"

Apuntes consultados a lo largo del proyecto:

Enrique Berjano (2015). Apuntes de Instrumentación electrónica: Diseño de filtros. *Universidad Politécnica de Valencia*.

Chema Ferrero (2015). Apuntes de bioelectricidad: Fundamentos bioeléctricos. *Universidad Politécnica de Valencia*.

Chema Ferrero (2015). Apuntes de bioelectricidad: Sistema cardiovascular (ECG). *Universidad Politécnica de Valencia*.

Páginas Web consultadas a lo largo del proyecto:

Marzo del 2016. Componentes electrónicos y hojas de características. Disponible en: www.ti.com

Mayo del 2016. Componentes electrónicos y hojas de características. Disponible en: <http://www.analog.com>

Junio del 2016. Normativas sobre circuitos impresos. Disponible en: <http://www.aenor.es>

Enero del 2016. Herramientas y toolbox para Matlab. Disponibles en: <http://es.mathworks.com>

Noviembre del 2015. Información sobre Arduino. Disponible en: www.arduino.cc

Febrero del 2016. Trabajo sobre el diseño de un ECG. <http://www.dalcame.com/ecg.html#.V1c-YPmLTIV>