

SIBAIC

Sistema Básico de Interpretación Cardíaca

Máster en Sistemas Inteligentes
(Robótica de Servicios)

Trabajo final de máster

Autor: Fernando Javier Dueñas Carrascal

Tutor: Raúl Marín Prades

Fecha: abril de 2019



**UNIVERSITAT
JAUME•I**

Índice

1. Introducción	4
2. Justificación y Objetivos	6
3. Estado del arte	6
3.1 Artículos relacionados	6
3.1.1 Prototype of a Portable Platform for ECG Monitoring... ..	6
3.1.2 Development of Portable ECG Measurement Device.....	7
3.1.3 A Portable WiFi ECG. School of Biomedical Engineering	7
3.1.4 Low Cost Solar ECG with Bluetooth transmitter.....	8
3.2 Productos similares	9
3.2.1 Quirumed ECG300G	9
3.2.2 Cardioline ECG100L.....	9
3.2.3 Comen CM600	10
3.2.4 Dimed EYD20633	10
4. Componentes hardware utilizados	11
4.1 Listado de componentes	11
4.2 Descripción de los componentes.....	13
5. Interconexión del hardware	14
6. Descripción del software inicial	17
7. Proyecto SIBAIC	18
7.1 Funcionamiento general de SIBAIC	18
7.2 ¿Qué es un electrocardiograma?	18
7.3 Información inicial	19
7.4 Correlación Voltaje Arduino-Corazón.....	20
7.5 Primer test	22
7.6 Segundo test.....	24

7.7 Cortado de segmento PQRST	24
7.8 Análisis del segmento PQRST	25
8. Informe de resultados	30
9. Código Arduino.....	30
10. Aplicación para Android	39
10.1 Código App Inventor	39
10.2 Capturas App	41
11. Conclusiones.....	43
12. Agradecimientos	43
13. Citas	43
14. Bibliografía y Webgrafía	43

1. Introducción

En este proyecto trataremos de explicar el montaje, la programación y el funcionamiento de lo que hemos bautizado como **SIBAIC**. SIBAIC es el acrónimo de **Sistema Básico de Interpretación Cardíaca** y se trata de un dispositivo para la obtención, análisis e interpretación básica de señales cardíacas y su posterior envío por email a través de un teléfono móvil y una aplicación. Es un dispositivo completamente autónomo, lleva alimentación interna y no es necesario conectarlo a un ordenador.

Su funcionamiento es sencillo y guiado, el dispositivo hace dos test al paciente, uno para la frecuencia y ritmo cardíaco, y otro para el complejo PQRST. Los analiza y muestra un diagnóstico de posibles anomalías en pantalla. Una vez terminado el análisis nos permite enviar por Bluetooth los resultados a la aplicación para móviles Android, una vez en la aplicación, nos da la opción de enviarlos a un correo electrónico y de esta manera ser analizados en un ordenador por un profesional.

El proyecto en ningún caso pretende sustituir la figura del cardiólogo, esa no es su finalidad, la idea es hacer un pequeño prototipo que pueda llegar a evolucionar y llegar a lugares con pocos recursos y de esta forma poder hacer un análisis rápido y genérico para ayudar al cribado.

Empezaremos con una imagen general del sistema, seguidamente, mostraremos cómo está la situación actual con un estado del arte, continuaremos mostrando las piezas que componen el prototipo, su montaje y el código desde el que se parte, para finalizar enseñaremos cómo se ha programado el dispositivo y cómo se ha hecho la aplicación para móviles Android.

A continuación, podemos ver una imagen general del sistema y sus conexiones (Ilustración 1), con el flujo de información indicado con flechas:

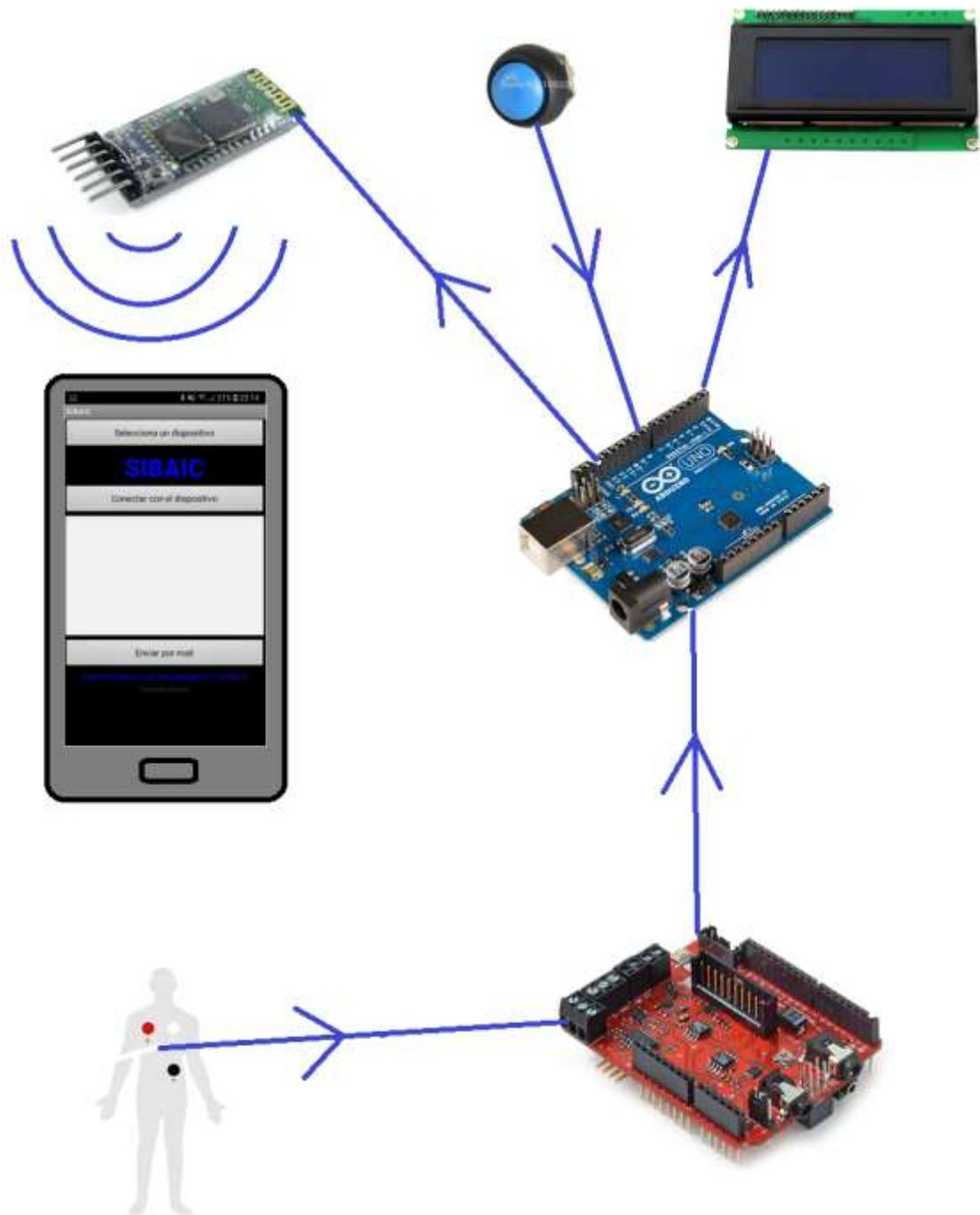


Ilustración 1: interconexión de los componentes.

La placa Arduino Uno (de color azul) se controlará desde el botón azul y recibirá información desde la placa e-Health (placa roja), ésta a su vez desde los electrodos conectados al cuerpo del paciente. La placa Arduino Uno enviará en tiempo real la información a la pantalla LCD y al finalizar el test, a través del dispositivo Bluetooth, se enviará a la aplicación para móviles Android.

2. Justificación y Objetivos

El proyecto parte del día 18 de abril de 2016, día en el que a mi hijo Jorge se le diagnostica una enfermedad rara del corazón, tras varios días de tratamiento, reacciona a la medicación de forma favorable y finalmente es dado de alta. A partir de ese momento tendría que tomar medicación diaria y revisiones cada 6 meses. Jorge tenía en ese momento 3 meses de edad. Siendo tan pequeño y al tratarse de una enfermedad rara, decidimos empezar este proyecto para poder monitorizar el estado de su corazón con mucha mayor frecuencia, ya que cada 6 meses nos parecía una eternidad. Ahí es donde empieza todo.

3. Estado del arte

3.1 Artículos relacionados

Lo primero que se hizo fue buscar artículos relacionados, los artículos más destacables de entre todos los encontrados serían los siguientes:

3.1.1 Prototype of a Portable Platform for ECG Monitoring...

B. Tovar Corona, J. E. Gonzalez Villarruel, H. Becerra Esquivel, A. Juárez Carrasco, A. Espíritu Santo Rincón, (2008), Prototype of a Portable Platform for ECG Monitoring and Diagnostic Applications, 2008 5th International Conference on Electrical Engineering, Computing Science and Automatic Control.

Este artículo describe el prototipo de un sistema portátil que registra y analiza el electrocardiograma y las señales en tiempo real. El propósito del sistema es monitorizar pacientes durante su actividad normal para ayudar a los médicos a diagnosticar el síndrome de QT largo y alertar para prevenir su repentina muerte. El sistema consiste en un sistema de grabación de señales cardíacas y un algoritmo implementado en un microcontrolador de baja potencia. El sistema ha sido probado en sujetos no patológicos que muestran valores precisos. Para la frecuencia cardíaca y el factor QTc el algoritmo también ha sido probado satisfactoriamente con señales creadas simulando patológicas y casos no patológicos. También fue probado con 10 pacientes de una base de datos estándar que muestra resultados satisfactorios.

3.1.2 Development of Portable ECG Measurement Device...

Gu-Young Jeong¹, Myoung-Jong Yoon, Kee-Ho Yu and Tae-Kyu Kwon⁴, Development of Portable ECG Measurement Device and PC Software for Automatic ST Analysis, International Conference on Control, Automation and Systems 2010 Oct. 27-30, 2010 in KINTEX, Gyeonggi-do, Korea.

El ECG se utiliza con el propósito de mantener una buena salud y controlar la función cardíaca de las personas de edad avanzada, así como para diagnosticar enfermedades cardíacas. El sistema de monitorización tiene una seguridad y precisión muy alta para prevenir el progreso de las enfermedades del corazón y la muerte súbita. Estos sistemas pueden detectar un cambio temporal del ECG, que es muy importante para diagnosticar enfermedades cardíacas como la isquemia miocárdica, la arritmia y el infarto cardíaco. En este artículo, describen el algoritmo de análisis de señales de ECG y el sistema de medición para ECG. El dispositivo ECG portátil de pequeño tamaño diseñado consistía en un amplificador de instrumentación, un microcontrolador, filtro y un módulo RF. El dispositivo desarrollado mide el ECG con cuatro electrodos en el cuerpo y detecta el complejo QRS y cambio de nivel del ST en tiempo real. También transmite la señal medida a un PC. El desarrollo del software para el análisis de ECG en un PC tiene la función de detectar los puntos característicos y el cambio de nivel ST.

3.1.3 A Portable WiFi ECG. School of Biomedical Engineering.

N. A. Abdul-Kadir, ¹N. S. Sahar, ²W. H. Chan, ^{1,3}F. K. C. Harun. A Portable WiFi ECG. School of Biomedical Engineering, School of Computing, School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, UTM Johor Bahru, Johor, Malaysia.

El desarrollo de la información y la tecnología de la comunicación ha mejorado la atención médica a través de la salud, proporcionando un software de análisis más sofisticado para hacer, en tiempo real, el procesamiento de señales del ECG. Por otra parte, la investigación intensa se ha dedicado al desarrollo a un precio asequible y fiable. El ECG es usado para monitorización diaria y uso ambulatorio. Las aplicaciones pueden proporcionar ECGs asequibles en tiempo real con un procesamiento adecuado para el sistema de vigilancia y alerta. En este estudio, se desarrolla un canal de comunicación inalámbrica con el ECG usando un módulo WiFi ESP8266 y un BMD101, dispositivo de bio-señal Neurosky en un chip (SoC). El SoC está diseñado con una potente estructura DSP que tiene un filtro, amplificador,

convertidor analógico-digital de 16 bits y un integrado. La señal de referencia de reloj es de 22.1MHz. El circuito de ECG está equipado con un solo conductor de dos entradas de positivo y negativo. El tamaño del circuito de ECG mide 4.5 x 3 cm, largo x ancho y es un dispositivo portátil. El diseño del circuito de ECG también puede capturar tanto ritmo sinusal como auricular.

3.1.4 Low Cost Solar ECG with Bluetooth transmitter.

Parin Dedhia Harsh Doshi Mrunal Rane Govinda Ahuja Biomedical Engineering Biomedical Engineering Biomedical Engineering Electronics Engineering D.J.Sanghvi c.o.e. D.J.Sanghvi c.o.e. D.J.Sanghvi c.o.e. NMIMS University Mumbai, India Mumbai, India Mumbai, India Mumbai, India. *Low Cost Solar ECG with Bluetooth transmitter*. 2012 International Conference on Biomedical Engineering (ICoBE), Penang, Malaysia, 27-28 February 2012.

La enfermedad cardiovascular es la principal causa de muerte del mundo, se contabilizan 16,7 millones o el 29,2% del total de muertes en el mundo, todos los años. La prevalencia informada de enfermedad coronaria (CHD) en adultos se han multiplicado por 4 en los últimos 40 años e incluso en las zonas rurales la prevalencia se ha duplicado en los últimos 30 años. El electrocardiógrafo (ECG) es un instrumento de diagnóstico que mide la actividad eléctrica del corazón, aportando valiosa información de trastornos cardíacos como infarto o aterosclerosis. Los instrumentos de ECG utilizados en hospitales son voluminosos y funcionan con alto voltaje que hace que la forma de onda del ECG se distorsione y un amplificador de aislamiento es necesario para la seguridad del paciente por lo que es caro, por lo tanto, no se puede utilizar en áreas subdesarrolladas. Hemos desarrollado un ECG portátil de bajo costo con energía solar. El 63% de Todos los hogares rurales en la India no tienen electricidad y usan combustible para iluminación. Por lo tanto, ser alimentado por energía solar eliminará La dependencia de la electricidad y se puede utilizar fácilmente en áreas ya sea Sin electricidad o con mucha carga. El electrocardiograma estará libre de la línea de ruido de frecuencia. El ECG también se comunica con el móvil del paciente o una computadora con la ayuda de un transmisor Bluetooth.

3.2 Productos similares

Después de realizar una búsqueda en el mercado hemos podido encontrar varios dispositivos de similares características. Los más significativos son los siguientes:

3.2.1 Quirumed ECG300G

Electrocardiógrafo ECG300G (Ilustración 2) de tres canales con análisis simultáneo de 12 derivaciones y sistema de impresión térmica de formas de onda ECG.

Funciones: Registro y visualización de formas de onda ECG en modos automático y manual, medición automática y diagnóstico automático de parámetros de forma de onda ECG. Gestión de historiales en memoria.



Ilustración 2: Imagen de electrocardiógrafo Quirumed ECG300G.

3.2.2 Cardioline ECG100L

Electrocardiógrafo de 12 derivaciones PORTÁTIL (Ilustración 3), completamente diagnóstico que visualiza, obtiene, imprime y memoriza los trazados ECG, para adultos y niños, junto con sus mediciones.

Una interfaz de usuario intuitiva guía al usuario a través de los diferentes pasos necesarios para obtener el



Ilustración 3: Imagen del electrocardiógrafo Cardioline ECG100L.

electrocardiograma. En la pantalla aparecen diversos mensajes que informan visualmente al usuario sobre las operaciones en curso o le advierten en caso de errores (por ejemplo, en caso de separación o desprendimiento de los electrodos).

3.2.3 Comen CM600

El ECG CM600 (Ilustración 4) es un equipo de 6 canales con una batería integrada de iones de litio recargable, impresora térmica y pantalla TFT LCD de 5,7 pulgadas. Los exámenes de ECG pueden ser evaluados con especial rapidez gracias a la medición automática y la función de interpretación. La detección de marcapasos y la interpretación de resultados, así como su posterior impresión facilitan el diagnóstico clínico.



Ilustración 4: Imagen del electrocardiógrafo Comen CM600.

3.2.4 Dimed EYD20633

Electrocardiografo (Ilustración 5) de 12 canales con interpretación. Permite imprimir 12 canales. Pantalla de 5,7". Teclado alfanumérico. Memoria interna para 100 ecg. Batería recargable de litio de duración 6 horas recarga 5 horas. Filtro digital completo. 12 derivación



Ilustración 5: Imagen del electrocardiógrafo Dimed EYD20633.

A continuación, compararemos en una tabla resumida las características más significativas de los productos comerciales encontrados (Ilustración 6):

Marca y modelo	€	Batería	Impresora	Diagnóstico en el dispositivo	Envío de resultados por email	LINK
SIBAIC	85€	V	X	V	V	-
Quirumed ECG300G	530€	V	V	V	X	LINK
Cardioline ECG100L	999€	V	V	X	USB	LINK
Comen CM600	1.022€	V	V	V	USB	LINK
Dimed EYD20633	3.371€	V	V	V	X	LINK

Ilustración 6: Tabla comparativa.

4. Componentes hardware utilizados

4.1 Listado de componentes

A continuación se detalla un listado de componentes utilizados en el proyecto:

- Una placa Arduino Uno (Ilustración 7).
- Un Shield e-Health V2.0 de la empresa Cooking Hacks (Ilustración 8).
- Un kit de 3 electrodos (Ilustración 9).
- Una pantalla LCD de 20x4 píxeles con módulo I2C (Ilustración 10).
- Una caja universal para proyectos de electrónica (Ilustración 11).
- Un botón azul universal. (Ilustración 12)
- Módulo HC-05 Bluetooth. (Ilustración 13)



Ilustración 7: Imagen de placa Arduino UNO.



Ilustración 8: Imagen de placa E-Health.



Ilustración 9: kit de 3 electrodos.



Ilustración 10: pantalla LCD de 20x4 píxeles con módulo I2C



Ilustración 11: caja universal para proyectos de electrónica.



Ilustración 12: botones de universales.



Ilustración 13: módulo HC-05 Bluetooth.

4.2 Descripción de los componentes

La placa **Arduino UNO** (Ilustración 7) forma parte de una plataforma de hardware y software de código abierto, es una sencilla placa con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación Processing.

El **Shield E-health** (Ilustración 8) es una placa de adquisición de datos que es capaz de obtener, a través de sus sensores, los siguientes parámetros: temperatura corporal, electromiograma, *electrocardiograma* (que es el único que usaremos en este proyecto), caudal de aire, respuesta galvánica de la piel, presión sanguínea, pulso y glucómetro.

El **kit de 3 electrodos** (Ilustración 9) está compuesto por 3 cables que van desde el SIBAIC hasta el cuerpo del paciente.

La **pantalla LCD** (Ilustración 10) de 20x4 pixeles a la cual le hemos instalado un módulo I2C, el cual permite reducir el cableado de 10 a 2 cables (más otros 2 de alimentación).

La **caja universal** (Ilustración 11) es una caja genérica de plástico negro para proyectos de electrónica.

El **botón** (Ilustración 12) es un pulsador universal de color azul.

El módulo **Bluetooth HC-05** (Ilustración 13) nos permitirá enviar al móvil los datos obtenidos y desde éste a una dirección de correo electrónico mediante una sencilla aplicación creada en APP Inventor.

5. Interconexión del hardware

Las conexiones electrónicas se hicieron de la siguiente forma:

La placa de Arduino y el Shield e-health se unieron mediante el pin Analog 0 y alimentación (5V y GND).

El botón se conectó al pin Digital 4 con una resistencia de 10K Ohm (Ilustración 14).

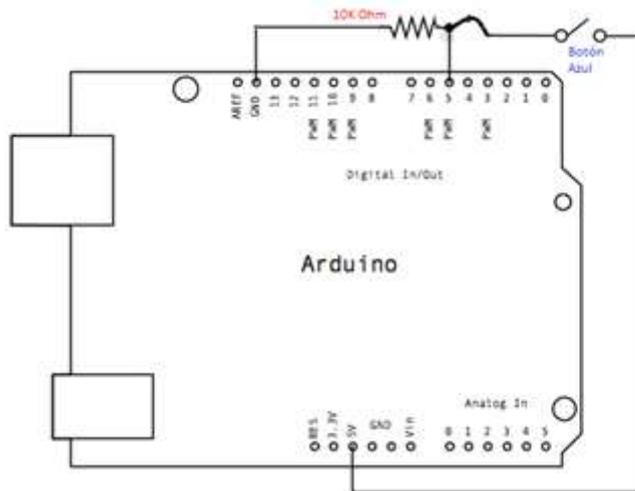


Ilustración 14: esquema de conexión del pulsador.

La pantalla LCD se conectó a las entradas SDA y SCL del Arduino (Pines 4 y 5 Analog respectivamente) mediante su placa de bus I2C (Ilustración 15).

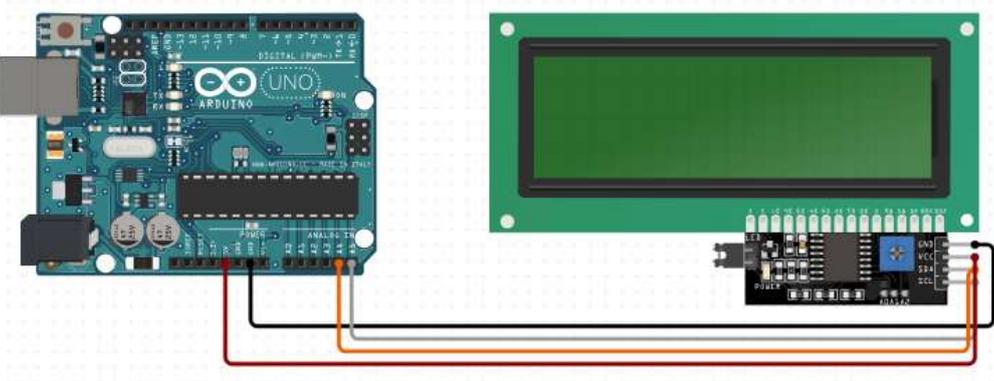


Ilustración 15: esquema de conexión del módulo I2C a la placa Arduino Uno.

El módulo Bluetooth HC-05 se conectó de la siguiente manera, pines 8, 9, 10 y 11 del Arduino a Vcc, Key, Txd y Rxd del Bluetooth respectivamente (Ilustración 16):

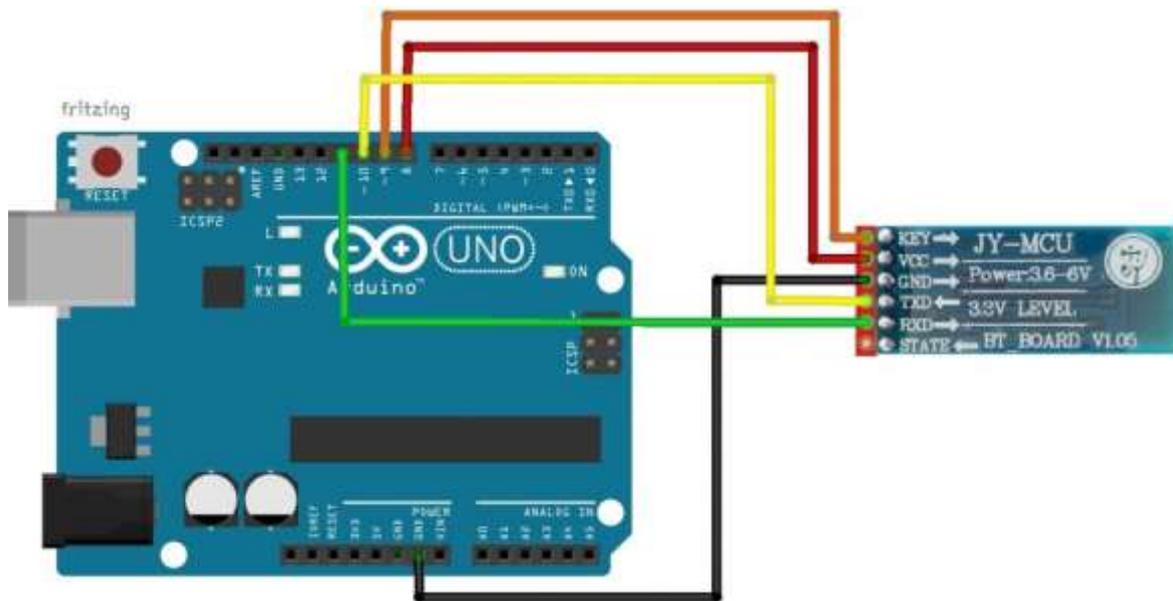


Ilustración 16: conexión del módulo Bluetooth HC-05 a la placa Arduino Uno.

Se ha hecho un cableado con interruptor para poder alimentarlo con una pila de 9V y no tener que depender de un USB ni de un PC, de esta forma es una herramienta completamente autónoma.

Los 3 electrodos van conectados a la toma del Shield e-health y en el otro extremo al cuerpo del paciente siguiendo el siguiente esquema (Ilustración 17), a ese tipo de conexión se conoce en medicina como derivación primera.

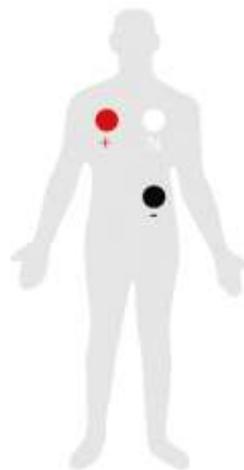


Ilustración 17: posición de los 3 electrodos

Se ha utilizado una caja para poder transportarlo, el resultado del montaje es el siguiente:
Caja cerrada con compartimento para electrodos nuevos (Ilustración 18):



Ilustración 18: caja cerrada

Caja abierta con SIBAIC, electrodos y pegatina de esquema de conexión (Ilustración 19):



Ilustración 19: caja abierta

SIBAIC solo con cables de electrodos conectados (Ilustración 20):



Ilustración 20: SIBAIC con cables conectados.

6. Descripción del software inicial

El shield e-Health viene con una librería llamada eHealth.h y el siguiente código de ejemplo, es desde el que se ha partido para hacer todo el proyecto:

```
#include <eHealth.h>

// The setup routine runs once when you press reset:
void setup() {
  Serial.begin(115200);
}

// The loop routine runs over and over again forever:
void loop() {

  float ECG = eHealth.getECG();

  Serial.print("ECG value : ");
  Serial.print(ECG, 2);
  Serial.print(" V");
  Serial.println("");

  delay(1);    // wait for a millisecond
}
```

Si conectamos los electrodos a un paciente y ejecutamos este código (cambiando Serial.begin a 9600) vemos que, de forma cíclica, se muestra por el monitor de puerto serie un valor en voltios cada cierto tiempo, el resultado es el siguiente (Ilustración 21):

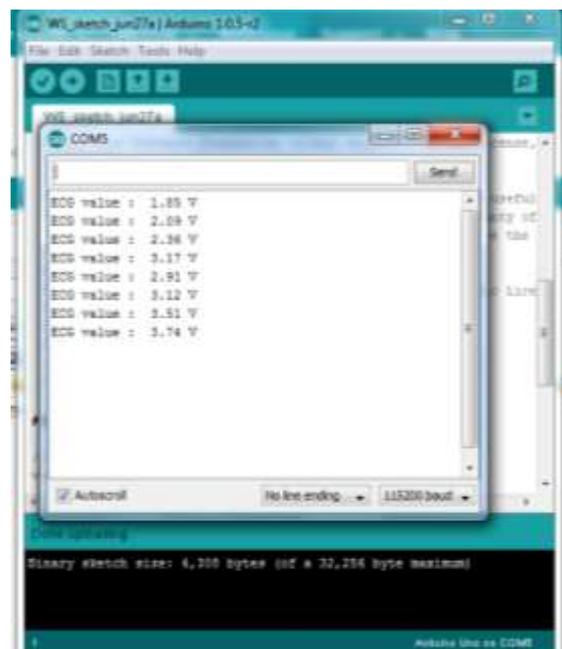


Ilustración 21: salida por puerto serie del código inicial.

Si esta señal la mostramos (sin el texto extra del código) por la salida *Serial Plotter* de Arduino, obtenemos éste gráfico (Ilustración 22), el cual es muy parecido al típico latido que conocemos:



Ilustración 22: salida convertida a modo gráfico.

Hasta este punto todo ha sido trivial, lo que se ha hecho ha sido interconectar los componentes, montarlos en una caja y cargar el código que viene con la placa.

7. Proyecto SIBAIC

7.1 Funcionamiento general de SIBAIC

SIBAIC funciona de la siguiente manera: una vez conectados los electrodos al paciente, lo encendemos y tras una presentación, nos indica que pulsemos el botón azul para empezar el test. El sistema hace dos capturas de datos y una vez concluido, nos lo hace saber y muestra los resultados en pantalla. Acto seguido nos ofrece mandarlo a la App de móvil.

7.2 ¿Qué es un electrocardiograma?

Para entender el funcionamiento lo primero es conocer cómo funciona un electrocardiograma. El electrocardiograma es una prueba que registra la actividad eléctrica del corazón que se produce en cada latido. Esta actividad eléctrica se registra desde la

superficie corporal del paciente y así se observan las diferentes ondas que representan los estímulos eléctricos de las aurículas y los ventrículos.

Existen sólo 2 magnitudes en el electrocardiograma:

- Anchura, para referirnos al tiempo (expresado en centésimas de segundo)
- Altura o profundidad (según la onda sea positiva o negativa respectivamente) para aludir al voltaje (que es la fuerza de los potenciales eléctricos).

La anchura se representa horizontalmente y la altura o profundidad verticalmente, el voltaje que reciben los músculos del corazón en cada latido se representa de la siguiente forma, cada onda tiene un nombre que se representa por una letra: P, Q, R, S y T (Ilustración 23):

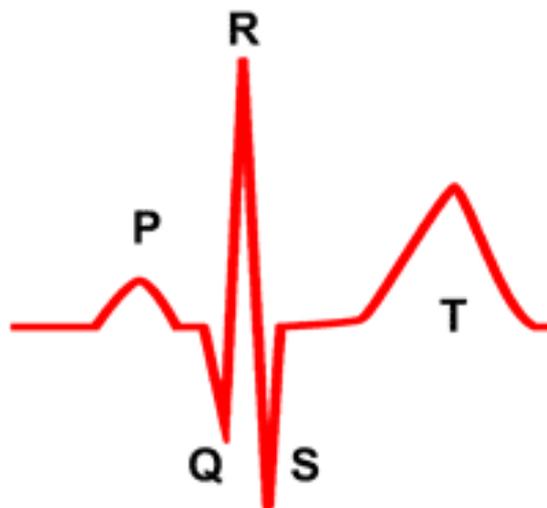


Ilustración 23: representación de las ondas PQRST.

7.3 Información inicial

La información con la que partimos es una señal de voltaje entre 0 y 5V (función `eHealth.getECG()`), con una frecuencia que desconocemos. El primer reto es conocer la frecuencia de esos datos, lo que hacemos es usar la función de Arduino `millis()`, ésta devuelve el número de milisegundos que han transcurrido desde que Arduino se ha encendido (o reseteado).

Con esto, si junto al voltaje mostramos el resultado de la función `millis()`, veremos la frecuencia con la que se muestran los datos.

7.4 Correlación Voltaje Arduino-Corazón

En el electrocardiograma, la señal del corazón se mide en voltios, pero esta señal en voltios que tenemos del Arduino no es un valor absoluto del corazón, es simplemente el valor de entrada de Arduino, el cual es siempre entre 0 y 5V, sea cual sea la información que le entra. El corazón trabaja con un voltaje mucho menor, de entre 0mV y 2mV aproximadamente.

Por este motivo, lo primero que necesitábamos era conocer la correlación entre la entrada que nos estaba dando Arduino y los valores reales del corazón. Para ello nos prestaron un electrocardiógrafo profesional, modelo Healthcare MAC 600 (Ilustración 24). Éste lo conectamos en paralelo para poder obtener los valores de la misma persona en el mismo momento y así poder hacer la correlación, además de paso podríamos ver la precisión de nuestro hardware.



Ilustración 24: electrocardiógrafo Healthcare MAC 600

Lo que hicimos fue hacer una captura de datos durante unos segundos con los dos dispositivos a la vez y comparar, en varios latidos, la altura o profundidad de la onda para conocer la correlación entre el voltaje de Arduino y el voltaje real del corazón. Los datos del electrocardiógrafo profesional los imprimimos en el papel milimetrado de la impresora que tiene y así podíamos medirlos (Ilustración 25).

Esto se podía conocer ya que existe un estándar de representación cardiaca siendo la relación siguiente:

- Cada 5 cuadrados, horizontalmente son 0,2 segundos
- Cada 5 cuadrados, verticalmente representan 0,5mV.

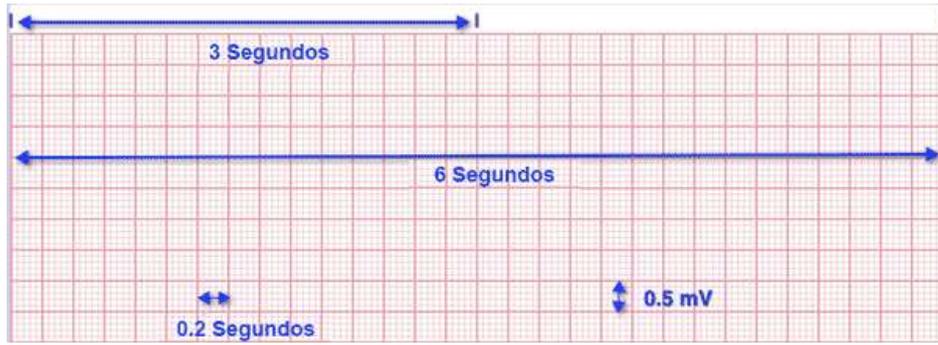


Ilustración 25: papel milimetrado genérico de electrocardiógrafo.

Con esta información, comparando las 2 adquisiciones de datos podíamos obtener la siguiente correlación:

Cogiendo los puntos máximo y mínimo, la salida Arduino da valores entre 2.79V y 1.47V, los cuales se ajustan a 1.1mV y -0.1mV respectivamente del electrocardiógrafo profesional.

Con esta información ya podíamos sacar la correlación (Ilustración 26):

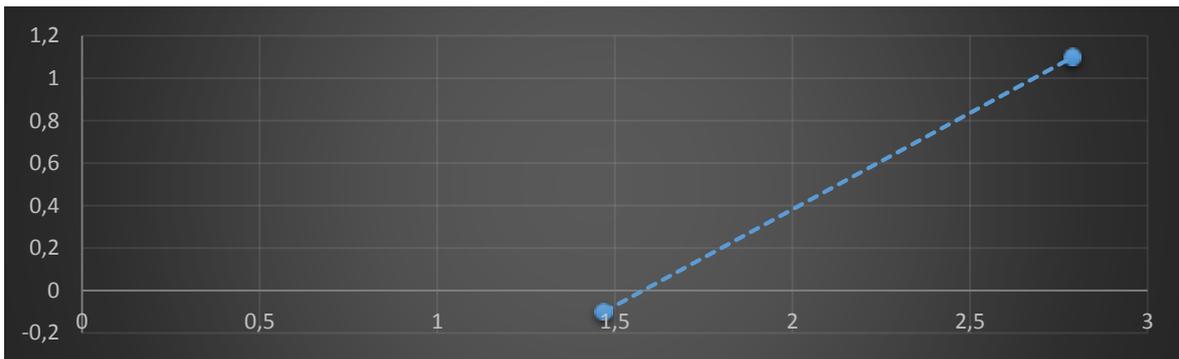


Ilustración 26: gráfico de correlación.

Lo siguiente que nos dimos cuenta era que, dependiendo de los decimales que queríamos, la cantidad de muestras por segundo bajaba o subía. Consideramos que 2 decimales eran suficientes, la correlación fue la siguiente (Ilustración 27):

Decimales	8	6	4	3	2	1	0
Muestras por segundo	83	99	124	142	166	200	331

Ilustración 27: correlación decimales/muestras por segundo.

7.5 Primer test

El primer test que hace SIBAIC es para mostrar la frecuencia cardíaca y analizar las posibles arritmias.

La forma que se nos ocurrió para obtener la **frecuencia** fue calcular la distancia entre las ondas R, las más altas, cada conjunto de ondas es un latido. Si le restábamos al instante donde empieza una onda R, el instante de su anterior onda R, obteníamos el tiempo en milisegundos entre una y otra. De esta forma, para obtener la frecuencia sólo teníamos que hacer una división sabiendo que la frecuencia es la inversa del tiempo.

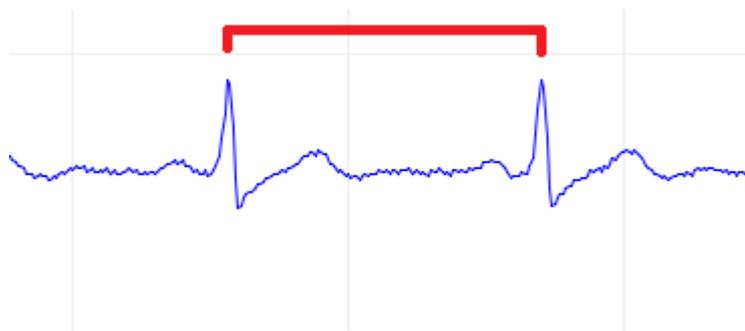


Ilustración 28: gráfico que muestra la distancia entre crestas.

Para conocer el momento de una onda R lo que hicimos fue crear un filtro sobre los voltajes de entrada. Para conocer esto hicimos un sencillo programa para obtener los valores mínimo y máximos, los resultados fueron entre 1.5 y 3V aproximadamente, con esto veríamos en qué intervalo se movían los datos.

Nos dimos cuenta de que las ondas R eran las únicas superiores a 2.5V así que hicimos un filtro en ese voltaje, para ello guardábamos con `millis()` el momento en el que el voltaje era mayor a 2.5V, esperábamos 250 milisegundos para que terminara la onda R y volvíamos a capturar el siguiente momento superior a 2.5V (Ilustración 29).



Ilustración 29: distancia entre cortes de subida en ondas R contiguas.

Restando a ese instante el anterior y con la fórmula que hemos comentado, teníamos la frecuencia cardíaca tras cada latido y en tiempo real. La frecuencia cardíaca se mide en latidos por minuto, con lo que nuestra fórmula quedó así:

$$\text{Frecuencia} = 60000 / (\text{"Momento onda R"} - \text{"Momento onda R anterior"})$$

Decidimos comparar nuestra primera muestra con el electrocardiógrafo profesional y el resultado fue exactamente el mismo, en cada latido obtenían el mismo resultado, 77 latidos por minuto en esta captura (Ilustración 30).

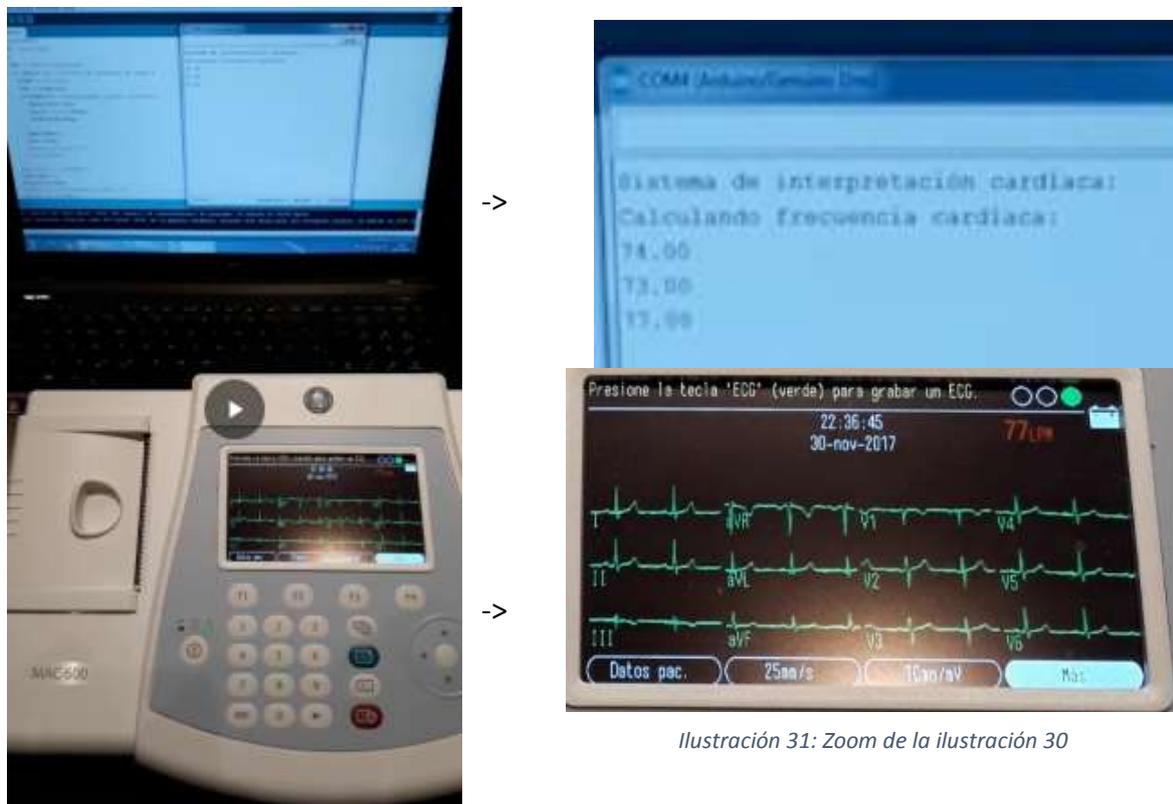


Ilustración 31: Zoom de la ilustración 30

Ilustración 30: comparación entre dispositivos.

Una vez conocemos la frecuencia, podíamos deducir la frecuencia cardíaca media y posibles taquicardias, bradicardias o arritmias.

La frecuencia cardíaca media la obteníamos sumando 10 muestras y dividiendo entre 10. Una frecuencia cardíaca normal de un adulto en reposo está entre 60 y 100 latidos por minuto, si está por debajo se podría considerar lo que se conoce como bradicardia y si está por encima

sería lo que llamamos taquicardia. Esta información se analiza y se muestra en la pantalla LCD una vez finaliza el primer test.

Para la **arritmia** (anomalía de la periodicidad del latido cardiaco) no nos vale la frecuencia media, ya que la media puede detectarse como normal si los tiempos entre latidos se van compensando unos con otros. Lo que hacemos en este caso es analizar varias frecuencias y obtener la desviación típica y si es mayor a 10 consideramos que no es regular y lo indicamos como tal.

7.6 Segundo test

Lo que hace el segundo test es analizar un conjunto de ondas PQRST. Dependiendo de la duración de cada una de ellas, significa que se tiene una cardiopatía u otra, cabe comentar que (salvo el intervalo QT que veremos más adelante) éstas son invariables con respecto a la frecuencia, es decir, a una mayor frecuencia no son menos duraderas, simplemente hay menor tiempo entre un grupo PQRST y el siguiente.

7.7 Cortado de segmento PQRST

Lo primero que hacemos es coger un solo segmento PQRST entero para analizarlo. Esto lo hacemos de la siguiente manera (Ilustración 32):

- Calculamos primero el tiempo que transcurre en ese momento entre latido y latido (con la función anterior de frecuencia sobre la onda R).
- Dejamos que transcurra la mitad del tiempo que hemos obtenido. (Por ajustes de error ese tiempo es el 0.75 en lugar de 0.5).
- Capturamos durante esa cantidad de tiempo que hemos calculado, todo lo que ocurre a continuación.

De esta forma tendremos todo lo que ocurre durante un latido entero. Usamos una precisión de dos decimales y almacenamos 250 muestras que guardamos en un *array* que son las que aproximadamente dura el segmento para, a continuación, analizarlo con nuestro algoritmo, cada muestra es un voltaje y lo guardamos multiplicándolo por 100 para trabajar sin decimales.



Ilustración 32: explicación gráfica del corte PQRST en rojo.

7.8 Análisis del segmento PQRST

Para poder analizar las posibles anomalías lo que necesitamos es saber la duración de las ondas PQRST, para eso tenemos que conocer los puntos de inicio y fin de las ondas, lo que hemos llamado *cortes*. Para ayudarnos a encontrar esos cortes usaremos lo que hemos denominado Estados.

De todas las ondas posibles, sólo analizaremos las siguientes duraciones (rodeadas en rojo): Intervalo PR, Complejo QRS, Intervalo QT, Onda P y la altura de la Onda R (Ilustración 33).

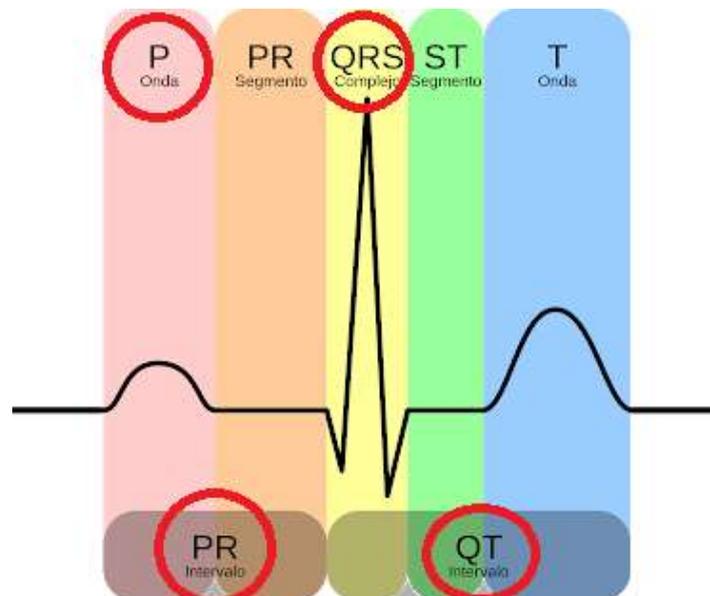


Ilustración 33: Ondas analizadas de entre todas las posibles.

Intervalo PR: se mide desde el inicio de la onda P hasta el inicio del complejo QRS, abarcando a la onda P y al segmento PR. Tiene que dar un valor comprendido entre 120 y 200ms. Si su duración fuera mayor a 200ms podría ser señal de un trastorno llamado: “*Bloqueo Auriculoventricular de Primer Grado*”.

Complejo QRS: Se mide desde el final del segmento PR hasta el inicio del segmento ST. Tiene que dar un valor comprendido entre 60 y 100ms. Si fuera mayor a 120ms podría ser señal de hipertrofias, hiperpotasemia, un marcapasos o hipotermia.

Intervalo QT: Se mide desde el inicio del complejo QRS hasta el final de la onda T. Tiene que dar un valor comprendido entre 350-450ms. Si fuera mayor de 580ms podría ser señal de hipercalcemia y si fueran inferiores a 320ms podría ser hipocalcemia.

Onda P: Tiene que dar un valor menor a 100ms, si su duración fuera superior a 120ms podría considerarse que hay un Bloqueo Interauricular.

Altura de la Onda R: Tiene que dar una altura menor a 2mV, si es mayor es un posible infarto.

Para obtener estos datos lo que haremos será lo siguiente:

El primer tramo del *array* coincide con la línea isoeletrica, la línea isoeletrica es una línea en la que se considera que no hay actividad eléctrica en el corazón, este valor es aproximadamente 178, lo que equivale a unos 0 mV reales.

En nuestro algoritmo empezamos a recorrer el *array* en el estado A (en azul) y no pasará al estado B hasta que no encuentre un valor superior a 185, en 185 consideramos que ya no está en la línea isoeletrica y que ya está en la onda P (aunque no sea en su inicio) (Ilustración 34).

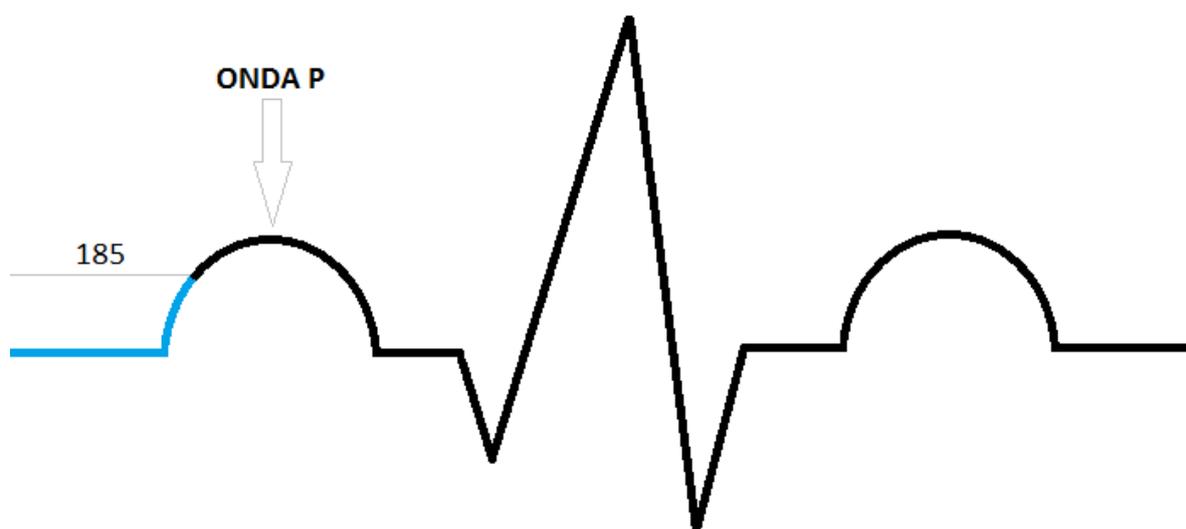


Ilustración 34: Primer corte, onda P.

A continuación, dentro de la onda P, sacamos el momento del valor máximo (lo llamamos *P_{MAXB}*) hasta que el valor descienda de 178, una vez descienda de 178 guardamos el instante como C2. En ese punto cambiamos a estado C (Ilustración 35).

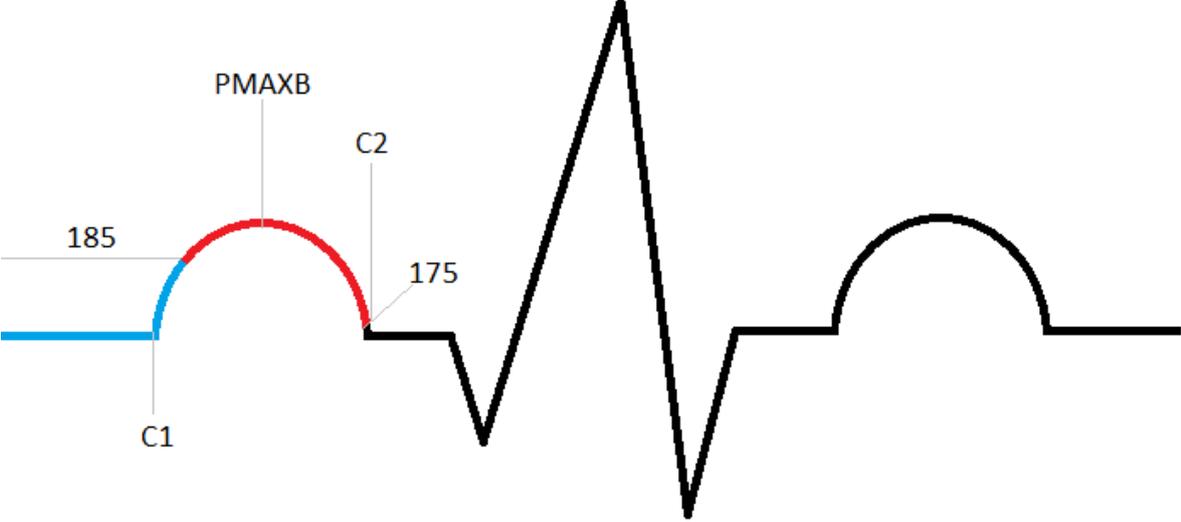


Ilustración 35: Instante C2 una vez baja de 178.

Estando en C (en verde), cuando el valor pasa de 225 cambiamos a Estado D (amarillo), estamos en la onda R (Ilustración 36).

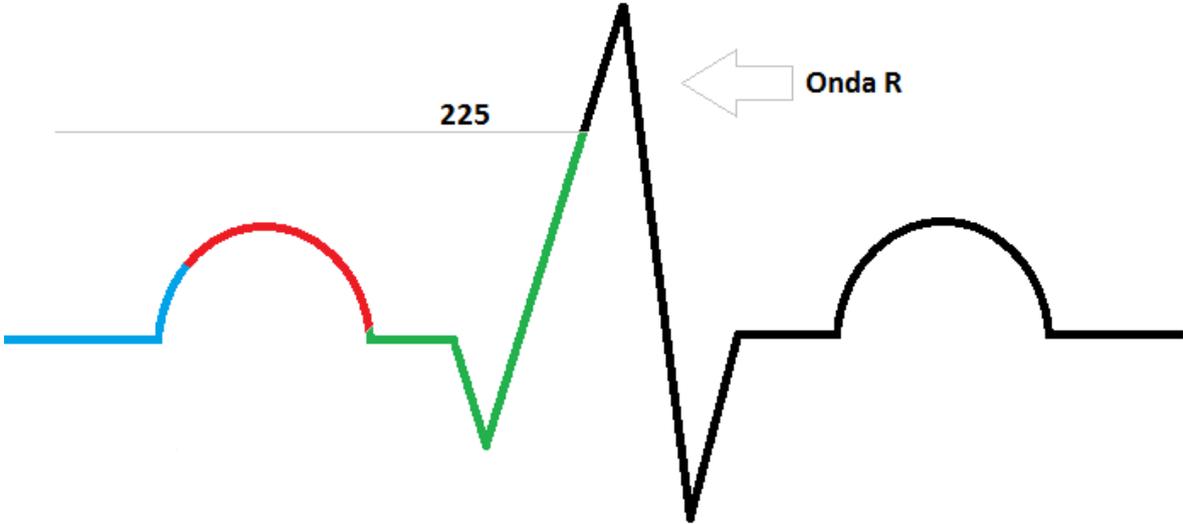


Ilustración 36: cambio a estado D al pasar de 225.

Estando en Estado D, es donde encontraremos el momento máximo y lo guardaremos en PMAX. En cuanto baje el valor de 178 (CRS) pasaremos a estado E (Ilustración 37).

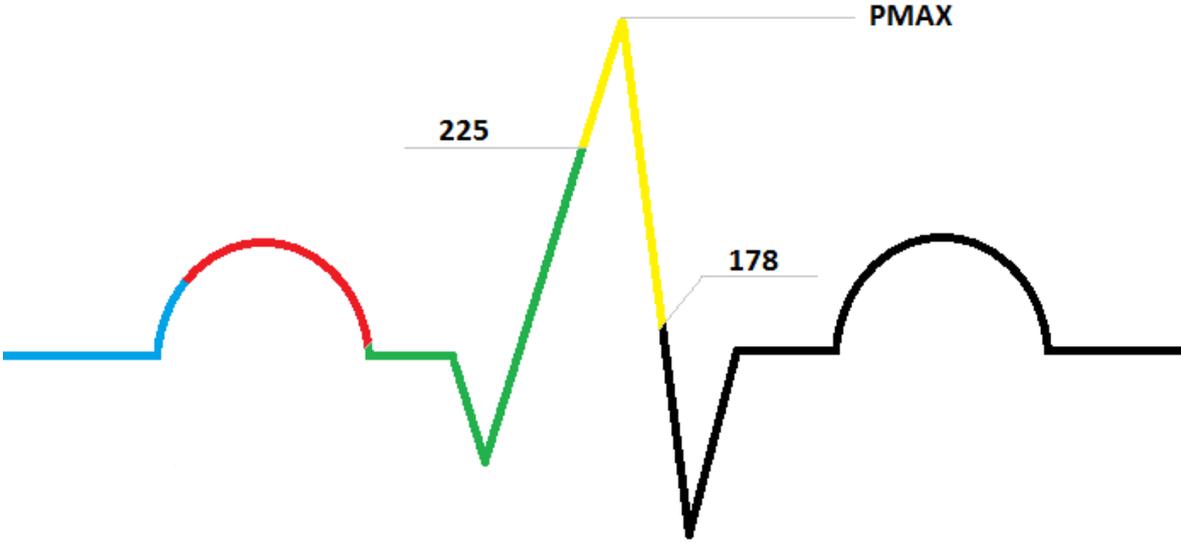


Ilustración 32: cambio a estado E al bajar de 178.

En estado E (en morado), en cuanto aumente de 190 pasará a estado F (Ilustración 38).



Ilustración 33: cambio a estado F en cuanto suba de 190.

En estado F (en negro), en cuanto baje de 178 pasará a estado G y guardaremos ese momento como C6 y con esto finaliza (Ilustración 39).

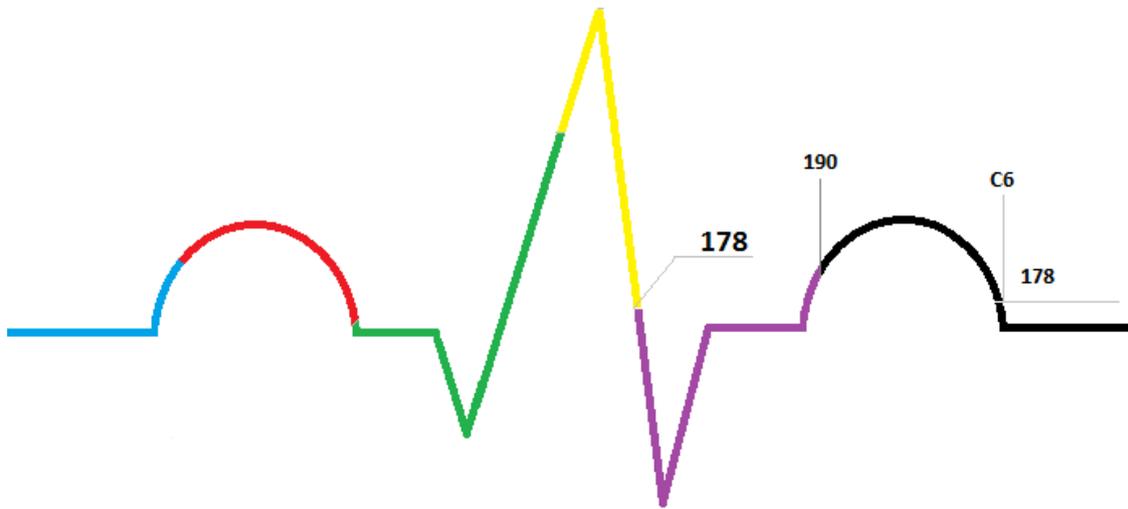


Ilustración 34: cambio a estado G al bajar de 178.

Los momentos (dentro del array) que hemos obtenido son: PMAXB, C2, PMIN, CRS, C5 y C6. Los valores que podemos deducir de éstos son: C1, C3 y C4.

El valor C1 lo deducimos sabiendo que las ondas P son simétricas, como tenemos el instante del valor máximo (PMAXB) y el instante C2, podemos sacar la duración de la onda P multiplicando por dos el tiempo entre éstos. La fórmula es la siguiente: $C1 = PMAXB * 2 - C2$;

El valor C3 lo deducimos sabiendo que está justo en el centro entre C2 y PMAX. La fórmula es la siguiente: $C3 = C2 + (PMAX - C2) / 2$;

El valor C4 lo deducimos sabiendo que PMIN está justo en mitad entre C4 y CRS. La fórmula es la siguiente: $C4 = PMIN * 2 - CRS$;

Las duraciones las calculamos de la siguiente manera:

Para el **Intervalo PR** restamos el momento C1 al C3.

Para el **Complejo QRS** restamos el momento C3 al C4.

Para el **Intervalo QT** restamos el momento C3 al C6.

Para la **Onda P** restamos el momento C1 al C2.

Para la altura de la **Onda R** miramos **PMAX** y hacemos la conversión entre V y mV.

8. Informe de resultados

Una vez hecho el test, analizamos los resultados y mostramos las posibles anomalías si están fuera de los rangos normales, una vez mostrados finaliza por completo el test.

En la siguiente URL podemos ver el SIBAIC en funcionamiento:

<https://youtu.be/06XPBhNfDwQ>

9. Código Arduino

Para entender mejor el código, mostramos un gráfico donde se pueden ver las partes que lo componen (Ilustración 40):

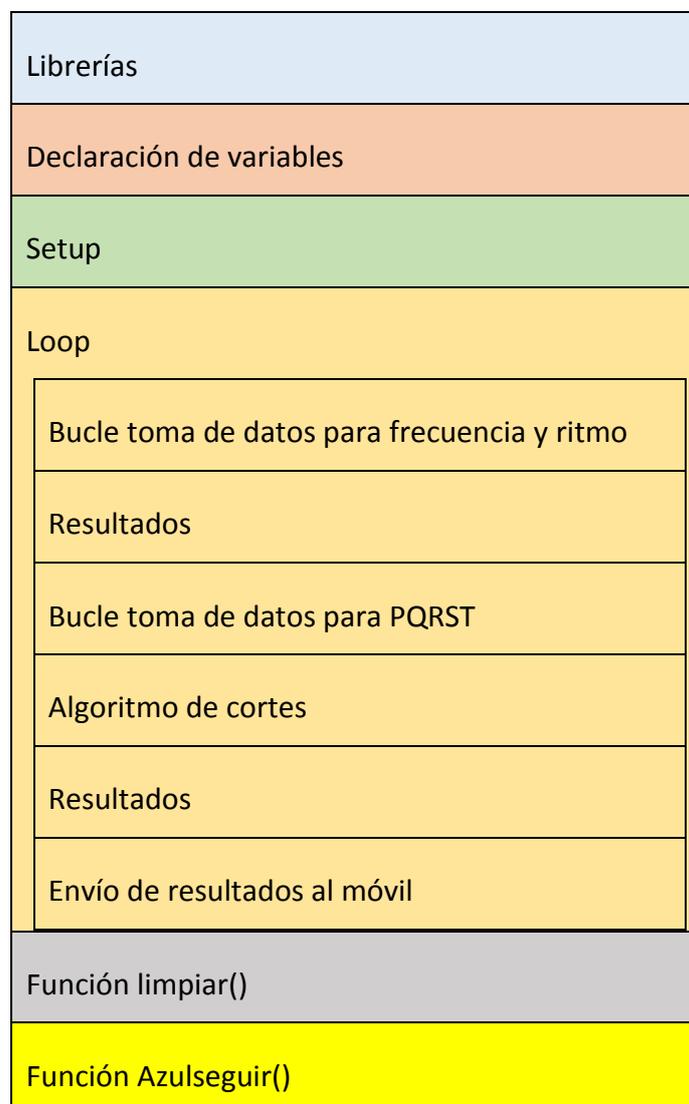


Ilustración 40: esquema general de código.

El código cargado en el Arduino es el siguiente:

```
#include <eHealth.h> // Librería del shield EHealth
#include <Wire.h> // Librería para comunicar con el ic2
#include <LiquidCrystal_I2C.h> // Librería para comunicar con lcd
#include <SoftwareSerial.h>
SoftwareSerial BT(10, 11); // 10 RX, 11 TX.
LiquidCrystal_I2C lcd(0x27, 20, 4); // LCD con 20 columnas y 4 filas

const int buttonAzul = 5; // Pin Arduino
int buttonStateA = 0; // Estado del botón
int CANT = 0; // Variable para acumular el número de muestras
int FREQUENCY[11]; // Array donde almacenará las frecuencias
int SAMPLES[250]; // Array que almacenará los voltajes
int MIN = 999; // Voltaje mínimo
int MAX = 0; // Voltaje máximo
int MAXB = 0; // Voltaje máximo de la onda P
int PMAX, PMAXB, PMIN, CRS, C1, C2, C3, C4, C6; // Momentos de corte
int VCC = 8; // Pin digital para alimentar el módulo Bluetooth HC-05
int EN = 9; // Pin digital para la habilitación del módulo Bluetooth
HC-05
int ENV_FR, ENV_ARR, ENV_PR, ENV_QRS, ENV_QTC, ENV_P, ENV_R;
//Variables de envío a APP

float ACUM = 0; // Variable que acumulará el tiempo total
float VOLT; // Variable que almacena el voltaje de salida del shield
(0-5V)
float FREQ; // Variable de salida de frecuencia en latidos por minuto
float DURATION = 0; // Variable que guarda el tiempo entre la entrada y
la última salida
float DISTANCIA = 0; // Duración del test
float DESTIP = 0; // Desviación típica

unsigned long GOIN = 0; // Variable para guardar el instante de entrada
en bucle
unsigned long GOOUT = 0; // Variable para guardar el instante de salida
del bucle

char ESTADO = 'A'; // Iniciamos el estado en A

void setup() { // Setup, sólo se ejecuta una vez
  pinMode(buttonAzul, INPUT); // El pin del botón será una entrada
  lcd.init(); // Inicializamos el lcd
  lcd.backlight(); // Encendemos la luz del lcd
  limpiar(); // Funcion limpiar para limpiar el lcd
  lcd.setCursor(0, 1); // Situamos el cursor en la fila primera (0)
columna segunda (1)
  lcd.print(F(" Sistema Basico de"));
  lcd.setCursor(0, 2);
  lcd.print(F(" Interpret. Cardiaca"));
  lcd.setCursor(0, 3);
  lcd.print(F(" Por Fernando Duenas"));
  pinMode(VCC, OUTPUT); // Pin digital 8 como salida
  pinMode(EN, OUTPUT); // Pin digital 9 como salida
  digitalWrite(EN, HIGH); // Habilitamos el módulo Bluetooth HC-05
  delay(500); // Esperamos 500 ms
  digitalWrite(VCC, HIGH); // Encendemos el módulo Bluetooth HC-05
  Serial.begin(9600); // Comienzo la comunicación con el monitor serie
  BT.begin(9600); // Comienzo de la comunicación con el módulo
Bluetooth HC-05 (por defecto)
```

```

    delay(2000); // Dejamos 2 segundos
}

void loop() { // Bucle que se repite
    Azulseguir(); // Función que espera la pulsación del botón azul para
continuar
    limpiar();
    lcd.print(F("  Analizando..."));
    do { // Bucle de adquisición de datos para frecuencia
        VOLT = eHealth.getECG(); // Guardamos el voltaje de ese momento (0-
5V)
        if (VOLT > 2.5) { //Filtro de voltajes de onda R
            GOIN = millis(); // Guardamos momento de entrada
            DURATION = GOIN - GOOUT; // Duracion es momento de entrada menos
salida
            FREQ = 60000 / DURATION; // Frecuencia =
(1/Tiempo)*60[seg/min]*1000[ms]
            if (CANT > 0) { // Me salto primer cálculo ya que es erroneo
                limpiar();
                lcd.print(F("  Analizando..."));
                lcd.setCursor(0, 2);
                lcd.print(F("    Muestra"));
                lcd.setCursor(14, 2);
                lcd.print(CANT);
                lcd.setCursor(5, 3);
                lcd.print(FREQ, 0);
                lcd.setCursor(8, 3);
                lcd.print(F("lat/min")); // Muestro frecuencia instantanea
(latido actual/anterior)
                ACUM = ACUM + FREQ; // Acumulo frecuencias
            }
            FREQUENCY[CANT] = FREQ; // Guardo frecuencia
            CANT = CANT + 1; // Añado una muestra más
            delay(250); // Retraso para no captar la misma onda R
            GOOUT = GOIN; // Actualizo tiempo de salida
        }
    } while (CANT < 11); // Tomo 10 latidos

    delay(2000);
    limpiar();
    lcd.print(F(" Frecuencia media:"));
    lcd.setCursor(5, 2);
    DURATION = 60000 / FREQ; // Calculo en milisegundos la duración entre
los últimos latidos
    FREQ = ACUM / (CANT - 1); //Frecuencia media
    lcd.print(FREQ, 0);
    lcd.setCursor(8, 2);
    lcd.print(F("lat/min"));
    Azulseguir();

    if (FREQ < 60) { //Análisis del resultado Bradicardia < 60 Normal <
100 Taquicardia
        limpiar();
        lcd.print(F("  ATENCION"));
        lcd.setCursor(0, 2);
        lcd.print(F("Posible bradicardia"));
        ENV_FR = 1;
    } else if (FREQ > 100) {
        limpiar();
        lcd.setCursor(0, 1);
        lcd.print(F("    ATENCION"));
    }
}

```

```

    lcd.setCursor(0, 2);
    lcd.print(F("Posible taquicardia"));
    ENV_FR = 2;
} else {
    limpiar();
    lcd.print(F(" Diagnostico:"));
    lcd.setCursor(0, 2);
    lcd.print(F(" Frecuencia normal"));
    ENV_FR = 3;
}
Azulseguir();
limpiar();
lcd.print(F(" Calculando ritmo:"));
lcd.setCursor(0, 2);
lcd.print(F(" por favor espere..."));
delay(2000);

for (int i = 1; i < 11; i++) {
    DESTIP = DESTIP + ((FREQUENCY[i] - FREQ) * (FREQUENCY[i] - FREQ));
}
DESTIP = DESTIP / CANT;
DESTIP = sqrt(DESTIP); //Calculo la desviación típica
limpiar();
lcd.print(F("Desv. tipica:"));
lcd.setCursor(15, 1);
lcd.print(DESTIP);
delay(2000);
if (DESTIP > 10) { //Análisis de arritmia
    limpiar();
    lcd.setCursor(6, 1);
    lcd.print(F("ATENCIÓN:"));
    lcd.setCursor(0, 2);
    lcd.print(F(" Posible arritmia"));
    ENV_ARR = 1;
} else {
    limpiar();
    lcd.print(F("Resultado:"));
    lcd.setCursor(0, 2);
    lcd.print(F("Ritmo normal."));
    ENV_ARR = 2;
}
Azulseguir();
limpiar();
lcd.print(F(" Ejecutando test"));
lcd.setCursor(0, 2);
lcd.print(F(" por favor espere..."));
delay(2000);

do { // Bucle de adquisición de datos
    VOLT = eHealth.getECG(); // Guardamos el voltaje de ese instante
(0-5V)
    if ((VOLT > 2.5)) { // Si estamos ya en onda R
        delay(DURATION * 0.75); //Espero 0.75 latidos y empiezo a tomar
datos
        GOIN = millis(); // Guardamos momento de entrada
        for (int j = 0; j < 250; j++) { // Bucle de toma de datos
            VOLT = eHealth.getECG(); // Toma de datos
            SAMPLES[j] = VOLT * 100; // Guardamos el voltaje de ese
momento*100 (0-5V)
            delay(3);
        }
    }
}

```

```

        GOOUT = millis(); // Guardo salida
        break; // Salimos
    }
} while (0 < 1); // Hasta el break
DURATION = GOOUT - GOIN; // Calculo el tiempo que tarda la toma de
datos
DISTANCIA = DURATION / 250; // Calculo el tiempo entre muestra y
muestra

// Algoritmo de búsqueda de cortes
for (int i = 0; i < 250; i++) { // Recorro el vector con los datos
    if ((ESTADO == 'A') && (SAMPLES[i] > 185)) { // Si está en estado A
y supera los 185 pasa a ser estado B
        ESTADO = 'B'; // Cambio de estado
    }
    if ((ESTADO == 'B') && (SAMPLES[i] > MAXB)) { //Función para sacar
el momento del punto máximo de la onda P
        MAXB = SAMPLES[i]; // Actualizo valor máximo
        PMAXB = i; // Actualizo momento máximo
    }
    if ((ESTADO == 'B') && ((SAMPLES[i] < 178))) { // Si estamos en B y
baja de 178 guardamos y pasamos a estado C
        C2 = i; // Guardamos C2
        ESTADO = 'C'; // Cambio de estado
    }
    if ((ESTADO == 'C') && (SAMPLES[i] > 225)) { // Si pasa de 225
estamos en D y es la onda R
        ESTADO = 'D'; // Cambio de estado
    }
    if (SAMPLES[i] > MAX) {
        MAX = SAMPLES[i]; // Actualizamos máximo absoluto
        PMAX = i; // Guardamos momento
    } else if (SAMPLES[i] < MIN) { // Actualizamos mínimo abosluto
        MIN = SAMPLES[i]; // Actualizamos
        PMIN = i; // Guardamos momento
    }
    if ((ESTADO == 'D') && (SAMPLES[i] < 178)) { // Si estamos en D
(onda R), no hemos entrado y baja de 178, pasamos a estado E y
        CRS = i; // Guardamos momento de corte CRS
        ESTADO = 'E'; // Cambiamos a estado E
    }
    if ((ESTADO == 'E') && (SAMPLES[i] > 190)) {
        ESTADO = 'F'; // Cambiamos a estado F
    }
    if ((ESTADO == 'F') && (SAMPLES[i] < 178)) { // Si no hemos pasado
el C6
        C6 = i; // Guardamos momento C6
        ESTADO = 'G';
    }
}

C1 = PMAXB * 2 - C2; //Obtenemos momento C1
C3 = C2 + (PMAX - C2) / 2; //Obtenemos momento C3
C4 = PMIN * 2 - CRS; //Obtenemos momento C4

//Intervalo PR
lcd.clear();
lcd.setCursor(3, 0);
lcd.print(F("Intervalo PR:"));

```

```

lcd.setCursor(3, 1);
lcd.print(F("Rango 120-200ms"));
lcd.setCursor(3, 2);
lcd.print(F("Valor: "));
lcd.setCursor(10, 2);
lcd.print((C3 - C1)*DISTANCIA);
lcd.setCursor(16, 2);
lcd.print(F("ms"));
Azulseguir();
if ((C3 - C1)*DISTANCIA) > 200) { //Análisis
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("ATENCIÓN:"));
  lcd.setCursor(0, 1);
  lcd.print(F("Posible Bloqueo"));
  lcd.setCursor(0, 2);
  lcd.print(F("Auriculoventricular"));
  lcd.setCursor(0, 3);
  lcd.print(F("de Primer Grado"));
  ENV_PR = 1;
} else {
  limpiar();
  lcd.print(F("Resultado:"));
  lcd.setCursor(0, 2);
  lcd.print(F("Intervalo PR normal.));
  ENV_PR = 2;
}
Azulseguir();

//Complejo QRS
lcd.clear();
lcd.setCursor(3, 0);
lcd.print(F("Complejo QRS:"));
lcd.setCursor(3, 1);
lcd.print(F("Rango 60-100ms"));
lcd.setCursor(3, 2);
lcd.print(F("Valor: "));
lcd.setCursor(10, 2);
lcd.print((C4 - C3)*DISTANCIA);
lcd.setCursor(16, 2);
lcd.print(F("ms"));
Azulseguir();
if ((C4 - C3)*DISTANCIA) > 120) { //Análisis
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("ATENCIÓN:"));
  lcd.setCursor(0, 1);
  lcd.print(F("Posibles hipertrofias"));
  ENV_QRS = 1;
} else {
  limpiar();
  lcd.print(F("Resultado:"));
  lcd.setCursor(0, 2);
  lcd.print(F("Complejo QRS normal.));
  ENV_QRS = 2;
}
Azulseguir();

//Intervalo QTc
lcd.clear();
lcd.setCursor(3, 0);

```

```

lcd.print(F("Intervalo QTc:"));
lcd.setCursor(3, 1);
lcd.print(F("Rango 320-450ms"));
lcd.setCursor(3, 2);
lcd.print(F("Valor: "));
lcd.setCursor(10, 2);
lcd.print(((C6 - C3)*DISTANCIA) / (sqrt(DURATION / 1000)));
lcd.setCursor(17, 2);
lcd.print(F("ms  "));
Azulseguir();
if (((C6 - C3)*DISTANCIA) > 450) { //Análisis
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("ATENCIÓN:"));
  lcd.setCursor(0, 1);
  lcd.print(F("Posible hipercalcemia"));
  lcd.setCursor(0, 2);
  ENV_QTC = 1;
  Azulseguir();
} else if (((C6 - C3)*DISTANCIA) / (sqrt(DURATION / 1000)) < 320) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("ATENCIÓN:"));
  lcd.setCursor(0, 1);
  lcd.print(F("Posible hipocalcemia"));
  ENV_QTC = 2;
} else {
  limpiar();
  lcd.print(F("Resultado:"));
  lcd.setCursor(0, 2);
  lcd.print(F("Intervalo QT normal.));
  ENV_QTC = 3;
}
Azulseguir();

//Onda P
lcd.clear();
lcd.setCursor(3, 0);
lcd.print(F("Onda P :"));
lcd.setCursor(3, 1);
lcd.print(F("Menor a 100ms"));
lcd.setCursor(3, 2);
lcd.print(F("Valor: "));
lcd.setCursor(10, 2);
lcd.print((C2 - C1)*DISTANCIA);
lcd.setCursor(16, 2);
lcd.print(F("ms"));
Azulseguir();
if (((C2 - C1)*DISTANCIA) > 100) { //Análisis
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(F("ATENCIÓN:"));
  lcd.setCursor(0, 1);
  lcd.print(F("Posible Bloqueo"));
  lcd.setCursor(0, 2);
  lcd.print(F("Interauricular "));
  ENV_P = 1;
} else {
  limpiar();
  lcd.print(F("Resultado:"));
  lcd.setCursor(0, 2);

```

```

    lcd.print(F("Onda P normal."));
    ENV_P = 2;
}
Azulseguir();

//Altura Onda R
lcd.clear();
lcd.setCursor(3, 0);
lcd.print(F("Altura Onda R:"));
lcd.setCursor(3, 1);
lcd.print(F("Menor a 2mV"));
lcd.setCursor(3, 2);
lcd.print(F("Valor: "));
lcd.setCursor(10, 2);
lcd.print(SAMPLES[PMAX] * 0.01 - 1.55);
lcd.setCursor(14, 2);
lcd.print(F("mV"));
Azulseguir();
if ((SAMPLES[PMAX] * 0.01 - 1.55) > 2) { //Análisis
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(F("ATENCIÓN:"));
    lcd.setCursor(0, 1);
    lcd.print(F("Posible infarto. "));
    ENV_R = 1;
} else {
    limpiar();
    lcd.print(F("Resultado:"));
    lcd.setCursor(0, 2);
    lcd.print(F("Onda R normal."));
    ENV_R = 2;
}
Azulseguir();
limpiar();

lcd.setCursor(0, 1);
lcd.print(F("    Desea enviar?"));
lcd.setCursor(0, 2);
lcd.print(F("    Conecte la APP "));

buttonStateA = digitalRead(buttonAzul);
while (buttonStateA != HIGH) {
    buttonStateA = digitalRead(buttonAzul);
}
limpiar();
lcd.setCursor(0, 2);
lcd.print(" Enviando: ");
BT.print("Inicio de envío");
BT.print("\r\n");
for (int i = 0; i < 250; i++) {
    BT.print(SAMPLES[i]);
    BT.print("\r\n");
    lcd.setCursor(14, 2);
    lcd.print(i / 2.5);
    delay(50);
}
BT.print("Fin");
limpiar();
BT.print("\r\n");
if (ENV_FR == 1) {
    BT.print("ATENCIÓN: Posible bradicardia.");
}

```

```

} else if (ENV_FR == 2) {
    BT.print("ATENCIÓN: Posible taquicardia.");
} else {
    BT.print("Frecuencia normal.");
}
BT.print("\r\n");

if (ENV_ARR == 1) {
    BT.print("ATENCIÓN: Posible arritmia.");
} else {
    BT.print("Ritmo normal.");
}
BT.print("\r\n");

if (ENV_PR == 1) {
    BT.print("ATENCIÓN: Posible Bloqueo Auriculoventricular de Primer
Grado.");
} else {
    BT.print("Intervalo PR normal.");
}
BT.print("\r\n");

if (ENV_QRS == 1) {
    BT.print("ATENCIÓN: Posibles hipertrofias");
} else {
    BT.print("Complejo QRS normal.");
}
BT.print("\r\n");

if (ENV_QTC == 1) {
    BT.print("ATENCIÓN: Posible hipercalcemia.");
} else if (ENV_FR == 2) {
    BT.print("ATENCIÓN: Posible hipocalcemia.");
} else {
    BT.print("Intervalo QT normal.");
}
BT.print("\r\n");

if (ENV_P == 1) {
    BT.print("ATENCIÓN: Posible Bloqueo Interauricular.");
} else {
    BT.print("Onda P normal.");
}
BT.print("\r\n");

if (ENV_R == 1) {
    BT.print("ATENCIÓN: Posible infarto.");
} else {
    BT.print("Onda R normal.");
}
BT.print("\r\n");

lcd.setCursor(0, 3);
lcd.print(F("  Enviado con exito"));
delay(9999999);
}

void limpiar() { // Función de borrado lcd, título SIBAIC y apunte a
línea 2
    lcd.clear();
    lcd.setCursor(0, 0);
}

```

```

    lcd.print(F("      SIBAIC"));
    lcd.setCursor(0, 1);
}

void Azulseguir() { // Función de muestra de botón Azul para continuar
    delay(20);
    lcd.setCursor(0, 3);
    lcd.print(F(" Azul para seguir "));
    // buttonStateA = digitalRead(buttonAzul);
    // while (buttonStateA != HIGH) {
    //     buttonStateA = digitalRead(buttonAzul);
    // }
}

```

10. Aplicación para Android

10.1 Código App Inventor

App Inventor es un entorno de desarrollo de software creado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android. El usuario puede, de forma visual y a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web.

Lo primero que hace nuestro código es inicializar las variables MAC del emisor y la que contendrá el contenido del mensaje.

A continuación, si el Bluetooth no está activado, lo activa.

Antes de hacer clic en el botón "Selecciona un dispositivo" rellena la lista con los dispositivos.

Una vez seleccionas el dispositivo SIBAIC, cambia el texto del botón a "Conectar con el dispositivo" e introduce la MAC en la variable.

Una vez pulsas sobre el botón "Conectar con el dispositivo" se conecta y espera a que le lleguen los datos desde el SIBAIC.

Una vez los recibe, los introduce como cuerpo de mensaje de un email y al pulsar "Enviar", abre la aplicación predeterminada en el móvil y los envía.

A continuación, veremos gráficamente el código que hemos creado para la APP para Android con APP Inventor.

```
initialize global MAC to AA-AA-AA-AA-AA-AA
initialize global eltexto to ""

when Screen1 Initialize
do
  if not BTClient Enabled
  then
    call as_habilitarBT StartActivity

when LSTDispoBT BeforePicking
do
  set LSTDispoBT Elements to BTClient AddressesAndNames

when LSTDispoBT AfterPicking
do
  set LSTDispoBT Text to join "Conectar a" LSTDispoBT Selection
  set global MAC to segment text LSTDispoBT Selection
  start 1
  length 17

when BtnConectarBT Click
do
  if not BTClient IsConnected
  then
    if call BTClient Connect address get global MAC
    then
      set BtnConectarBT Text to "Dispositivo conectado"
      set cl_BT TimerEnabled to true
    else
      call BTClient Disconnect
      set BtnConectarBT Text to "Conectar"
      set cl_BT TimerEnabled to false

when cl_BT Timer
do
  set global eltexto to join get global eltexto
  call BTClient ReceiveText numberOfBytes call BTClient BytesAvailableToReceive
  set TxtMensajes Text to join get global eltexto
  call BTClient ReceiveText numberOfBytes call BTClient BytesAvailableToReceive

when BtnEnviar Click
do
  set ASMail Action to "android.intent.action.VIEW"
  set ASMail DataUri to join "mailto:al073651@uji.es"
  "?subject='Test'&body="
  get global eltexto
  call ASMail StartActivity
```

Ilustración 41: imagen del código de la App.

10.2 Capturas App

Al abrir la App nos aparece arriba un botón para seleccionar el dispositivo (Ilustración 42), al pulsarlo veremos el listado con los dispositivos enlazados y seleccionamos el Bluetooth SIBAIC, el cual ya hemos configurado con anterioridad (Ilustración 43).



Ilustración 42: pantalla inicial



Ilustración 43: listado dispositivos Bluetooth

Una vez seleccionado nos muestra la MAC (Ilustración 44) y nos da la opción de conectar con el dispositivo, una vez conectado nos lo indicará en el texto del botón (Ilustración 45).



Ilustración 44: dispositivo ya seleccionado



Ilustración 45: dispositivo conectado

En ese momento desde el SIBAIC ya podremos enviar los datos pulsando el botón azul, el móvil lo recibirá (Ilustración 46) y una vez llegue podremos pulsar en el botón “Enviar por email” que nos abrirá el gestor de correo con el contenido y la dirección configurada por defecto (en este caso la mía).



Ilustración 46: datos ya recibidos

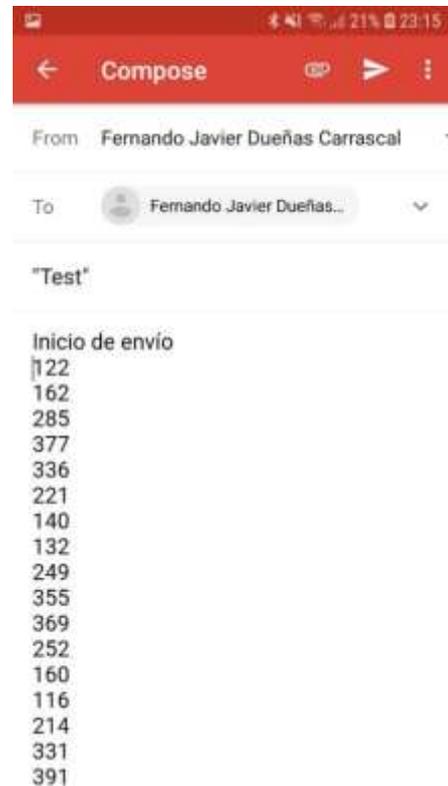


Ilustración 47: gestor de correo con datos

En el correo se recibirán los voltajes que se han analizado y a continuación el mismo texto de diagnóstico que ha aparecido por pantalla, si se abren con Excel, por ejemplo, podemos trabajar con los datos y también obtener la gráfica.



Ilustración 48: datos pasados a gráfico en Excel.

11. Conclusiones

Una vez se ha avanzado en el proyecto, nos damos cuenta de que realmente no es útil para poder monitorizar la enfermedad de mi hijo, sólo se puede monitorizar su progreso mediante un ecocardiógrafo profesional, a pesar de eso, decidimos terminarlo y que sirva como primer granito de arena para un proyecto que pudiera llegar a evolucionar y llegar a lugares con pocos recursos y de esta forma poder hacer un análisis rápido y genérico para ayudar al cribado.

12. Agradecimientos

Este proyecto se lo dedico a mis padres por apoyarme siempre, económicamente y sin límites en mis estudios, al Dr. Salvador por todo el material y el ofrecimiento, y a mi mujer Fátima por su dedicación, su apoyo y, sobre todo, por su paciencia.

13. Citas

“Cuando quieras emprender algo, habrá mucha gente que te dirá que no lo hagas; cuando vean que no pueden detenerte, te dirán cómo tienes que hacerlo; y cuando finalmente vean que lo has logrado, dirán que siempre creyeron en ti”. (John C. Maxwell)

“Imposible es sólo una palabra que usan los hombres débiles para vivir fácilmente en el mundo que se les dio, sin atreverse a explorar el poder que tienen para cambiarlo. Imposible no es un hecho, es una opinión. Imposible no es una declaración, es un reto. Imposible es potencial. Imposible es Temporal, Imposible no es nada” (Muhammad Ali)

“Sapientia sola libertas est”. (Lema de la UJI)

14. Bibliografía y Webgrafía

B. Tovar Corona, J. E. Gonzalez Villarruel, H. Becerra Esquivel, A. Juárez Carrasco, A. Espíritu Santo Rincón, (2008), Prototype of a Portable Platform for ECG Monitoring and Diagnostic Applications, 2008 5th International Conference on Electrical Engineering, Computing Science and Automatic Control.

Gu-Young Jeong¹, Myoung-Jong Yoon, Kee-Ho Yu and Tae-Kyu Kwon⁴, Development of Portable ECG Measurement Device and PC Software for Automatic ST Analysis, International Conference on Control, Automation and Systems 2010 Oct. 27-30, 2010 in KINTEX, Gyeonggi-do, Korea.

N. A. Abdul-Kadir, ¹N. S. Sahar, ²W. H. Chan, ^{1,3}F. K. C. Harun. A Portable WiFi ECG. School of Biomedical Engineering, School of Computing, School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, UTM Johor Bahru, Johor, Malaysia.

Parin Dedhia Harsh Doshi Mrunal Rane Govinda Ahuja Biomedical Engineering Biomedical Engineering Biomedical Engineering Electronics Engineering D.J.Sanghvi c.o.e. D.J.Sanghvi c.o.e. D.J.Sanghvi c.o.e. NMIMS University Mumbai, India Mumbai, India Mumbai, India Mumbai, India. *Low Cost Solar ECG with Bluetooth transmitter.* 2012 International Conference on Biomedical Engineering (ICoBE), Penang, Malaysia, 27-28 February 2012.

<http://www.fundaciondelcorazon.com/informacion-para-pacientes/metodos-diagnosticos/electrocardiograma.html>

<https://es.wikipedia.org>

<http://www.my-ekg.com>

http://amf-semfyc.com/web/article_ver.php?id=1083

<http://medicinageneraluniversal.blogspot.com/2017/01/que-es-un-electrocardiograma.html>