

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN



**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
ADQUISICIÓN, VISUALIZACIÓN, REGISTRO Y TRANSMISIÓN
DE DATOS PARA UN HOLTER CARDÍACO**

Trabajo de Especial de Grado presentado
por el

Br. Abel José Arce Castañeda
ante la Facultad de Ciencias de la
Ilustre Universidad Central de Venezuela
como requisito parcial para optar al título
de: **Licenciado en Computación**

Con la tutoría de: Prof. Robinson Rivas
Prof. Esteban Álvarez

Julio-2017
Caracas-Venezuela

Agradecimientos

Quiero expresar mi agradecimiento por este gran logro, en primer lugar a mi familia quien siempre puedo contar con su apoyo incondicional, en especial a mi padre Abel, a mi madre Martha, quienes me han acompañado durante todas mis experiencias con aciertos y fallos, agradezco a dios por inmenso regalo, agradezco a mi hermana que siempre tiene palabras para hacer reír, por ser una de las personas mas cercanas a mi, que siempre puedo contar con su apoyo cuando mas lo necesito, agradezco a mis otros hermanos por todo el apoyo brindado, siempre presentes ante cualquier necesidad que he tenido.

Y a todos mis amigos y personas que conscientes o inconscientemente me han brindado su ayuda, para poder cumplir con éxito, todos los requisitos exigidos para este logro, y especialmente agradezco a mis tutores Robinson Rivas y Esteban Álvarez, por darme la oportunidad de trabajar en este proyecto, que dios bendiga sus conciencias.

Gracias.

Abel Jose Arce Castañeda

UNIVERSIDAD CENTRAL DE VENEZUELA

FACULTAD DE CIENCIAS

ESCUELA DE COMPUTACIÓN

CENTRO DE COMPUTACIÓN

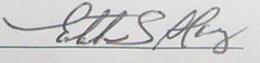
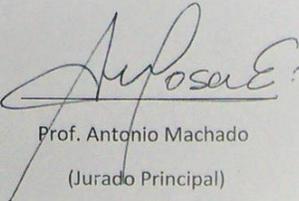


ACTA DEL VEREDICTO

Quienes suscriben, miembros del Jurado designados por el consejo de escuela de Computación, para examinar el Trabajo Especial de Grado, presentado por el Bachiller Abel Arce, portador de la cedula de identidad 23.499.394, con el título “**Diseño e implementación de un sistema de adquisición, visualización, registro y transmisión de datos para un Holter Cardíaco**”, a los fines de cumplir con el requisito legal para optar por el título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído como fue dicho trabajo por cada uno de los Miembros del Jurado, se fijó el día 11 de octubre de 2017, a las 3:00 PM, para que su autor lo defienda en forma pública, en el Centro de Computación, mediante la exposición oral de su contenido, y luego de la cual respondieron satisfactoriamente a las preguntas que le fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente Acta, en Caracas a los 11 días del mes de octubre del año dos mil diecisiete, dejándose también constancia de que actuó como Coordinador del Jurado el Profesor Tutor Robinson Rivas.

 _____ Prof. Robinson Rivas (Tutor)	 _____ Prof. Esteban Álvarez (Tutor)
 _____ Prof. Mercy Ospina (Jurado Principal)	 _____ Prof. Antonio Machado (Jurado Principal)

REPUBLICA DE VENEZUELA
Facultad de Ciencias

Escuela de Computación
Universidad Central - Caracas

RESUMEN

En este trabajo se desarrollaron, configuraron y probaron, los módulos que componen el Holter ECG, basado en hardware y software libre, para la utilización académica de la institución, el Holter permite obtener registros cardíacos que puedan ser vistos, analizados y estudiados, por parte de investigadores y/o especialistas en el área, apoyando así, la investigación.

El dispositivo fue desarrollado e implementado mediante la plataforma de software y hardware libre Arduino, además de elementos adicionales que fueron necesarios para la implementación como un software para PC, fue necesario la utilización de hardware diseñado e implementado por un tercero, tales como tarjeta lectora de memoria SD, tarjeta amplificadora de señal.

Dando como resultado un dispositivo propio de la institución, que cuenta con todas las bondades del software y hardware libre, el cual permitirá en un futuro realizar mejoras en el dispositivo en cuestión.

Palabras clave: Holter, ECG, Arduino, QRS, Bluetooth, registros, derivaciones, LCD, electrocardiografía.

Índice general

Lista de figuras	10
Lista de tablas	13
1. Propuesta	17
1.1. Planteamiento del problema	17
1.2. Justificación	18
1.3. Objetivos	18
1.3.1. Objetivos General	18
1.3.2. Objetivos Específicos	18
1.3.3. Alcances	19
2. Marco Conceptual	21
2.1. Microcontrolador	21
2.1.1. Microchip Tech. Inc.	21
2.1.2. Atmel Corporation	23
2.2. Arduino	24
2.2.1. Dispositivos Arduino	24
2.2.2. Procesador Digital de Señal (DSP)	25
2.3. ECG Holter	26
2.3.1. Análisis de la señal Electrocardiográfica	26
2.3.2. Diseño digital Holter ECG	28
2.3.3. Características necesarias para el desarrollo del Holter ECG	30
2.4. Especificaciones y características Arduino Mega 2560	32
2.4.1. Alimentación eléctrica	33
2.4.2. Memoria	34
2.4.3. Puertos de E/S	35
2.5. Software Libre	35
2.5.1. Hardware Libre	35
2.5.2. Copyright	36
2.5.3. Copyleft	36
2.6. Modulación por Codificación de Pulsos (PCM)	37

2.6.1.	Funciones de un codificador decodificador combinado	37
2.6.2.	Expansión - Compresión	38
2.7.	El Corazón Humano	39
2.7.1.	Anatomía del Corazón	39
2.7.2.	Configuración exterior del corazón	40
2.8.	El ciclo cardiaco	41
2.8.1.	Sístole auricular	41
2.8.2.	Fase de eyección	41
2.8.3.	Periodo diastólico. Fase de relajación isovolumétrica	43
2.8.4.	Fase de llenado rápido ventricular	43
2.8.5.	Fase de llenado lento; diástasis o estasis	44
2.9.	Nódulo Sinoauricular	44
2.10.	Electrocardiografía	45
2.10.1.	Activación Cardiaca	46
2.10.2.	Transmisión del impulso de las aurículas a los ventrículos	46
2.10.3.	Despolarización ventricular (Asa de QRS)	46
2.10.4.	Repolarización ventricular. (Asa de T)	47
2.11.	Registro del potencial eléctrico	47
2.11.1.	Eje eléctrico	48
2.11.2.	Onda P	49
2.11.3.	Complejo QRS	50
2.11.4.	Onda T	52
2.11.5.	Intervalo QT	53
2.11.6.	Segmento PR	53
2.11.7.	Segmento ST	54
3.	Implementación de la plataforma	57
3.1.	Tecnologías utilizadas	57
3.1.1.	Arduino IDE	57
3.1.2.	Writing Sketches	57
3.1.3.	Tarjeta de acondicionamiento	58
3.1.4.	Tarjeta de registro, interfaz SD, pulsadores y LEDs	59
3.1.5.	Pantalla de visualización	62
3.1.6.	Placa adaptadora pantalla LCD	62
3.1.7.	Módulo para conexión bluetooth	64
3.2.	Diseño	65
3.2.1.	Etapa de adquisición	65
3.2.2.	Etapa de procesamiento	66
3.2.3.	Etapa de almacenamiento	66
3.2.4.	Etapa de transmisión	66
3.2.5.	Esquema de pruebas	67

3.2.6. Indicadores esperados	68
3.3. Implementación de la plataforma	69
3.3.1. Etapa de adquisición	69
3.3.2. Implementación de la etapa de adquisición	71
3.3.3. Etapa de procesamiento	72
3.3.4. Etapa de almacenamiento	75
3.3.5. Etapa de transmisión	78
3.3.6. ECG Receiver	83
4. Pruebas de la plataforma	89
4.1. Resultados	94
4.2. Conclusiones	97
4.3. Recomendaciones	98

Índice de figuras

2.1. Monitor ECG Holter con electrodos	27
2.2. Onda PQRS	28
2.3. Diseño lógico Electrocardiograma	29
2.4. Codificación PCM de 8 bits	38
2.5. Vista posterior Corazón Humano	40
2.6. Vértices y Cavidades del Corazón	42
2.7. Nódulo Sinusal	45
2.8. Registro Electrocardiográfico	47
2.9. Intervalos de un electrocardiógrafo	48
2.10. Onda P	49
2.11. Origen onda P	50
2.12. Complejo QRS	51
2.13. Origen onda Q	51
2.14. Origen onda R	52
2.15. Onda T	52
2.16. Intervalo QT	53
2.17. Segmento PR	54
2.18. Segmento ST	54
2.19. Complejo PQRS e intervalo RR	55
3.1. Tarjeta Amplificadora	58
3.2. Tarjeta Amplificadora	60
3.3. Pines de conexión memoria SD	61
3.4. Pantalla TFT de 3.2" utilizada en el prototipo	62
3.5. Adaptador Pantalla TFT 3.2"	63
3.6. Conexión de pantalla LCD mediante placa adaptadora	63
3.7. Modificación por hardware en placa adaptadora (recuadro verde)	64
3.8. Módulo bluetooth (HC-06)	64
3.9. Arquitectura Holter	65
3.10. Interfaz de comunicación serial e inalámbrica	67
3.11. Diseño lógico Holter	68
3.12. Módulos Holter ECG y objetivos	69

3.13. Atmega 2560 Pescaler formula	71
3.14. Representación del mallado ECG	72
3.15. Conexión Bluetooth	79
3.16. Protocolo de comunicacion ECG	80
3.17. Dispositivo Bluetooth mediante puerto COM5	83
3.18. Emparejamiento del dispositivo	84
3.19. Puertos utilizados por Bluetooth	85
3.20. Primera vista ECG Receiver	85
3.21. Panel Principal ECG Receiver	86
3.22. Mensaje de estado de la aplicación	87
3.23. Mensaje de confirmación	87
4.1. Esquemática de conexión de pruebas	89
4.2. Graficación de señal generada	90
4.3. Representación de la señal sinusoidal en osciloscopio	91
4.4. Generador de señal GoldStar FG-8002	91
4.5. Registro voluntario número 1	92
4.6. Registro voluntario número 8	93
4.7. Configuración de 3 canales y un cable de 5 terminales	93
4.8. Gráfica sobre pantalla LCD Holter ECG voluntario 1	94
4.9. Gráfica sobre pantalla LCD Holter ECG voluntario 2	95
4.10. Gráfica en PC de registro de voluntario número 1	96
4.11. Gráfica en PC de registro de voluntario número 2	96

Índice de cuadros

2.1. Comparativa en frecuencia y amplitud (ECG,EEG,EMG)	28
2.2. Especificaciones técnicas Arduino Mega 2560	33
3.1. Conexiones de pines SD Arduino	62
3.2. Configuración de preescaler	71
4.1. Lista de voluntarios	92

INTRODUCCIÓN

El uso de la tecnología siempre ha sido una herramienta esencial para el mejoramiento de la calidad de vida del ser humano, hasta el punto de convertirse en una necesidad, una de las aplicaciones en las que actualmente la tecnología de información juega un papel esencial es en la medicina, los médicos se apoyan sobre una gran cantidad de herramientas que le permiten entre muchas otras cosas, diagnosticar la causa de cualquier irregularidad en el paciente, siendo así un gran aporte para la práctica de su labor, las herramientas empleadas en la medicina abarcan un gran abanico de posibilidades incluyendo dispositivos más elaborados como un electrocardiógrafo.

Uno de los órganos esenciales para la vida del ser humano es el corazón, este órgano como casi cualquier otro órgano puede sufrir de patologías, anormalidades e irregularidades, que afectan de forma negativa su funcionamiento y por ende pone en riesgo la vida del ser humano, en este caso los médicos hacen uso de distintos dispositivos para el monitoreo de la actividad cardiaca, uno de esos dispositivos es llamado Holter. Un Holter es un instrumento que sirve para registrar la actividad eléctrica del corazón durante un periodo de tiempo prolongado, también se llama monitor Holter cardiaco o Holter-ecg.

Un Holter ECG a diferencia de un electrocardiógrafo, registra las pulsaciones cardíacas durante un largo periodo de tiempo, puede ser de 24 horas o 48 horas dependiendo del dispositivo, mientras que el electrocardiógrafo solo muestra la actividad cardíaca en el momento, para realizar la captura de dicha actividad el Holter ECG tiene conectado una serie de electrodos que se adhieren al pecho del paciente, este puede contar con las llamadas derivaciones que son canales independientes donde se registra la actividad cardiaca y estan situados cerca al corazón, la idea es que el paciente después un lapso de tiempo se dirija al médico para que este retire el Holter y pueda leer los registros capturados, de esa manera el médico puede detectar cualquier patología que ponga en riesgo la vida del paciente.

Actualmente no existe un dispositivo Holter ECG en el que se puedan obtener los registros cardíacos sin que estén sujetos a licenciamiento privativo, donde es necesario pagar una cantidad de dinero antes de poder visualizar los registros, por lo que cualquier otro fin que se quiera dar a los registros capturados solo esta limitado a lo que el fabricante permita, que por lo general se limitan solo a la visualización.

El presente trabajo de investigación plantea como objetivo el registro, visualización, almacenamiento y transferencia de la señal electrocardiográfica (ECG) , para su posterior análisis, además de desarrollar una aplicación que sea capaz de obtener los registros cardiacos para el estudio y evaluación por parte de un especialista. Este trabajo se estructura de la siguiente manera:

- Primer capítulo donde se plantea el problema de investigación, objetivos y justificación.
- Segundo capítulo donde se encuentra toda la base conceptual, tales como microcontroladores, fabricantes y familias de microcontroladores, ondas ECG, electrocardiografía , origen de las ondas ECG.

- Tercer capítulo, en este capítulo hace referencia a las tecnologías utilizadas para el desarrollo del dispositivo, se plantea el diseño lógico implementado y por último todo el proceso de implementación.
- Cuarto capítulo se define el esquema de pruebas a las que fue sometido el dispositivo estructurados en dos fases, lista de voluntarios para esta fase, los resultados obtenidos y conclusiones.

Capítulo 1

Propuesta

1.1. Planteamiento del problema

Los dispositivos ECG-Holter que se encuentran en el mercado son bastante eficaces en su función, pero están desarrollados bajo software privativo, es decir tienen todas las limitantes que pueden tener cualquier software protegido bajo “copyright”, esto implica que no es posible ver, analizar, o estudiar los datos que son generados por el dispositivo, esta labor únicamente es posible a través del software que el mismo fabricante ofrece, limitando así cualquier otro uso que se le sea pensado dar.

Los fabricantes no parecen interesados en apoyar cualquier otro uso que le pueda dar a sus dispositivos, limitando así la capacidad por parte de los investigadores de obtener registros de pacientes para el análisis y procesamiento de dichos datos, con el fin de estudiar patrones de comportamiento en gran cantidad de patologías concernientes al corazón.

Esto genera la necesidad de desarrollar un dispositivo ECG-Holter, que apoye la investigación, dado que está basado en una plataforma libre, esta puede ser tomada, modificada y redistribuida a cualquier interesado, adaptado a sus distintas necesidades. El desarrollo de un dispositivo de estas características plantea una gran cantidad de consideraciones a nivel técnico, debido a la función de gran importancia que ejerce sobre el paciente.

Para el desarrollo del dispositivo es necesario la supervisión y evaluación de un especialista con el fin de cumplir con todos los requerimientos que sean necesarios para la realización de tan relevante función. En primera instancia es necesario llevar a cabo varios procesos para el desarrollo del ECG-Holter.

- Evaluación de los distintos dispositivos ofrecidos en el mercado para la implementación del ECG-Holter.
 - Desarrollo a nivel de software del módulo de transmisión de datos desde el dispositivo al PC.
 - Desarrollo del software del dispositivo, encargado de la captura y almacenamiento de todos los registros cardiacos.
-

- Desarrollo del software encargado de visualizar los registros obtenidos por el Holter.

1.2. Justificación

Es necesario determinar si con las herramientas, teorías, tecnologías existentes es posible plantear una solución viable para la comunidad al problema presente, así como la revisión de documentación que previamente puedan servir como contribución al trabajo actual.

Actualmente en el mercado existen fabricantes que ofrecen gran variedad de dispositivos Holter ECG, sin embargo el costo resulta poco accesible para la institución, es por ello que se evalúan factores clave como tecnologías o herramientas que dan lugar a la elaboración de un dispositivo que lleve a cabo la función de Holter ECG.

En un primer plano debido a que el Holter ECG corresponde a un elemento portable (el paciente debe cargar con el), la interacción de este dispositivo se hace cotidiano, por lo que es recomendable en menor medida la manipulación del dispositivo, es por ello que resulta una opción viable la transmisión de los registros vía inalámbrica, y así evitar daños al dispositivo.

1.3. Objetivos

1.3.1. Objetivos General

Diseño, desarrollo e implementación de un sistema, capaz de realizar una monitorización y registro Holter de la señal ECG, por medio de una plataforma electrónica, basada en software y hardware libre, para el almacenamiento, visualización y transmisión de la señal, para su posterior análisis.

1.3.2. Objetivos Específicos

- Revisar bibliografía de los siguientes aspectos: Electrocardiografía dinámica de Holter (24 horas). Microcontrolador: Arquitectura de un microcontrolador, microcontroladores Opensource.
 - Implementar un sistema de adquisición, capaz de monitorizar múltiples canales (derivaciones) correspondientes a la señal electrocardiográfica, minimizando la pérdida de información en el proceso de adquisición.
 - Mostrar por medio de una pantalla LCD, en tiempo real, la(s) derivación(es) que se van adquiriendo.
 - Grabar en un dispositivo digital, la(s) derivación(es) del registro electrocardiográfico, minimizando la pérdida de información en este proceso.
-

- Implementar la transferencia de los registros a un PC a través de una conexión tipo usb y/o inalámbrico.
- Implementar un formato para todos los registros, que incluya además de información del paciente, horas de inicio y fin, indicadores de posibles eventualidades señaladas por el paciente, entre otras.
- Evaluar y entonar el prototipo de adquisición implementado.

1.3.3. Alcances

Durante el desarrollo de esta investigación se logró la visualización, almacenamiento, transferencia de los registros capturados, a través de los 3 canales obtenidos. El ruido resulta un factor de suma importancia al momento de la fase de obtención de registros, el uso de electrodos en la medida de lo posible nuevos, resulta una mejora significativa en la calidad de los registros. El dispositivo logrando los tres procesos nombrado anteriormente (visualización, almacenamiento, transferencia) ya plantea un elemento que da pie a futuras investigaciones basadas en los datos generados por este.

Capítulo 2

Marco Conceptual

La elaboración de un dispositivo embebido para realizar una aplicación específica conlleva a utilizar herramientas específicas que se adapten a la necesidad, en el mercado existen gran variedad de herramientas y tecnologías que apoyan situaciones en las que se requiere un dispositivo potente, pequeño, durable, estable y de bajo costo, para este caso los microcontroladores es uno de ellos.

2.1. Microcontrolador

Un microcontrolador es un circuito integrado digital monolítico que contiene todos los elementos de un procesador digital secuencial síncrono programable de arquitectura Harvard o Princeton (Von Neumann) se le puede denominar también microcomputador integrado o empotrado y está especialmente orientado a tarea de control y comunicaciones. [1]

Actualmente existen varios fabricantes de microcontroladores, entre los más populares encontramos los microcontroladores PIC, de la empresa Microchip Tech. Inc. Atmel Corporation, Texas Instruments, Motorola e Intel, cada uno ofrece distintas categorías para los dispositivos, según sea la necesidad del consumidor, sin embargo existen dos fabricantes dominantes en el mercado de microcontroladores de bajo costo y de propósito general, el cual corresponde a los dos primeros nombrados.

2.1.1. Microchip Tech. Inc.

8-bit MCUs (Microcontrollers Unit)

Los Microcontroladores PIC de Microchip, están siendo utilizados en nuevas aplicaciones como teléfonos inteligentes, accesorios de audio, periféricos para juegos de vídeo y dispositivos médicos avanzados. Microchip ofrece soluciones para todo el rango de funcionamiento de microcontroladores de 8 bits, con herramientas de desarrollos fáciles de usar,

completa documentación técnica, y apoyo a través de una red distribución de dispositivos a nivel mundial, las familias ofrecidas por Microchip Tech son.

- **Microcontroladores PIC 10:** El PIC10FXXX es un microcontrolador de bajo costo, alto rendimiento, 8 bits, basados en memoria flash. Emplea una arquitectura RISC con solo 33 palabras (mnemotécnicos) de un solo ciclo de reloj por instrucción. Todas las instrucciones son de un solo ciclo (1 microsegundo) excepto para las instrucciones de salto (branch), que tienen dos ciclos. Esta familia ofrece un alto rendimiento en un MCU de bajo costo. El conjunto reducido de instrucciones RISC fáciles de usar y fácil de recordar reduce significativamente el tiempo de desarrollo.
 - **Microcontroladores PIC 12:** La Familia PIC12FXXX a diferencia de la anterior familia, esta ofrece instrucciones de función especial, las cuales son instrucciones de 12 bits, lo que resulta en una compresión 2:1 en código por ejemplo el PIC12F509 (ya que se utiliza 1 instrucción en lugar de dos) en comparación de otros microcontroladores de 8 bits de su clase.
 - **Microcontroladores PIC 16:** Además de los beneficios de las familias más básicas, la familia PIC16FXXX combina la flexibilidad de módulos analógicos integrados para crear un MCU de bajo coste adecuado para una amplia gama media de aplicaciones. Con un ADC incorporado, de doble Amplificadores Operacionales, y dos comparadores, el PIC16FXXX ofrece una excelente solución para el acondicionamiento de señales y aplicaciones de detección de señales.
 - **Microcontroladores PIC 18:** Esta nueva familia de dispositivos ofrece todas las ventajas de la PIC18F de alto rendimiento, con CPU de 8 bits, compilador de C optimizado y manejo de protocolo USB. Esta familia de 20 pines incluyen un oscilador interno 16 MHz que puede ser utilizado con el estándar 4X PLL para producir hasta 12 MIPS de rendimiento. Los periféricos analógicos estándar incluyen 12 canales de 10 bits para el convertidor A / D, módulos comparadores de tensión y una tensión de referencia para los comparadores son totalmente seleccionables por software. Los periféricos digitales incluyen características como la modulación por ancho de pulso, para la conversión D/A mediante modulación por ancho de pulsos (PWM), módulos de comunicación serial como (I2C, SPI). Estos dispositivos también cuentan con el modo “Latch SR” que soporta dispositivos de sensores capacitivos. Además de estos periféricos, los dispositivos cuentan con varios monitores de microcontroladores que proporcionan un funcionamiento fiable incluyendo BOR (Brown RESTABLEC) el cual cuando el PIC se encuentra inestable, este se reinicia automáticamente, POR (Power On Reset) y un WDT extendido. La incorporación de la tecnología nano-Watt XLP hacen estos dispositivos ideales para aplicaciones o diseños con pilas, donde se ve limitado el consumo de energía.
-

16-bit MCUs (Microcontrollers Unit)

La familia de microcontroladores PIC de 16 bits de Microchip (PIC24 MCU), incorporan controladores de señales digitales dsPIC que ofrecen a los diseñadores una forma rápida de reemplazo para microcontroladores de 8 bits PIC y una opción rentable para las MCU de 32 bits. Esta familia incluye desde microcontroladores de muy de baja potencia en comparación a los controladores de señal digital de alto rendimiento. Con la ejecución por instrucción de un solo ciclo de reloj, y un DMA (Direct Access Memory) mejorado, también incluyen acumuladores de 40 bits, ideal para aplicaciones de uso intensivo de cálculos matemáticos, como el control motor y la potencia digital.

En combinación con el hardware y el software libre, estos dispositivos de 16 bits son ideales para diseños que incluyen una alta eficiencia de control de motores, fuentes de alimentación digital (regulación) y de bajo consumo para una mayor duración de la batería en aplicaciones portátiles. Características táctiles y pantalla integrada ayudan a reducir los costes y simplificar los diseños de interfaces de usuario incluyendo mTouch® Sensing, y diversos controladores de pantalla.

También periféricos y software para conectividad USB especializados para estos dispositivos, protocolos inalámbricos que son fáciles en la comunicación con otros sistemas, además de manejo de protocolos ampliamente implementados como el TCP/IP.

32-bit MCUs (Microcontrollers Unit)

La familia de microcontroladores Pic 32 bits de Microchip con núcleo MIPS microAptiv o M4K ofrecen microcontroladores de alto rendimiento, y todas las herramientas necesarias para desarrollar proyectos. Estas incluyen MPLAB® y software Harmony, herramientas de desarrollo de bajo coste, gran compatibilidad con periféricos y toda la familia de 16 bits PIC. PIC32 MCU permite crecer fácilmente debido a su amplia compatibilidad, también ofrece el poder de procesamiento y memoria de alta gama, para aplicaciones más exigentes.

2.1.2. Atmel Corporation

La facilidad de uso, bajo consumo de energía y un alto nivel de integración, son las características principales de los microcontroladores Atmel AVR de 8 bits y 32 bits (MCU). Optimizados para reducir el tiempo de desarrollo, se basan en la arquitectura de código más eficiente a nivel de programación C y montaje del circuito. Una vez que los dispositivos se han puesto en marcha, la familia AVR permite volver a utilizar las experiencias previas para mejorar viejas implementaciones. Los dispositivos ofrecidos por Atmel Corporation incluyen.

8-bit Tiny AVR

Los microcontroladores (MCUs Atmel tinyAVR) se han optimizado para aplicaciones que requieren rendimiento, eficiencia energética y facilidad de uso en un pequeño inte-

grado. Todos los dispositivos tinyAVR se basan en la misma arquitectura y compatible con otros dispositivos AVR. ADC integrado, memoria EEPROM y detector de brown-out le permite crear aplicaciones sin necesidad de añadir componentes externos. La familia tiny AVR permite el desarrollo de código en C o Ensamblador, proporciona una simulación precisa del ciclo, y se integra perfectamente con kits de iniciación, programadores, depuradores, kits de evaluación, además de diseños de referencia.

32-bit AVR MCUs

La Familia de AVR 32-bit UC3, más allá de alto rendimiento y bajo consumo de energía, posee soporte nativo de manejo de cálculo en punto fijo mediante DSP integrado, memoria SRAM de doble ancho de bus, bus de datos de múltiples capas, controlador DMA y un sistema de interrupciones en los periféricos, que lo hace un dispositivo de alta gama para las aplicaciones más exigentes. En algunos de los dispositivos de esta familia incluyen una unidad de punto flotante integrada (FPU), que mejora el rendimiento de la aritmética con números decimales, con una mayor precisión y un rango más amplio.

Además de la familia PIC por parte de Microchip Tech. también existen otros fabricantes de microcontroladores, en este caso, uno de los dispositivos de gran reputación y popularidad es Arduino, basado en microcontroladores Atmel.

2.2. Arduino

Arduino es una plataforma física de cómputo de código abierto, para la creación de objetos interactivos el cual pueden incluir elementos adicionales. Arduino fue diseñado para artistas, diseñadores y otras personas que quieren incorporar la computación física en sus diseños sin tener que convertirse primero en ingenieros eléctricos, gracias a esto Arduino puede adaptarse de manera rápida a los requerimientos establecidos, siendo un punto a favor para esta plataforma. El hardware y el software de Arduino es de código abierto. La filosofía de código abierto fomenta una comunidad que comparte su conocimiento. Gran variedad de proyectos de ejemplo se presentan no sólo como imágenes del proyecto, sino que incluyen instrucciones para hacer su propio proyecto. El hardware y el software de Arduino es de código abierto, la filosofía de código abierto fomenta que la comunidad comparta su conocimiento. [2]

2.2.1. Dispositivos Arduino

Se pueden clasificar en 5 categorías, cada una con características adaptadas según sus necesidades, estas pueden incluir “tarjetas o placas” que son básicamente el microcontrolador embebido en una placa junto a otros componentes complementarios que facilitan la manipulación e interacción con el microcontrolador, estas placas pueden a su vez ser

complementadas con Módulos, Shields (placas independientes que cumplen una función en específico) o Kits, las categorías son las siguientes.

- **Nivel básico (entrada):** Contempla los dispositivos de baja y media gama tales como, ARDUINO UNO, ARDUINO 101, ARDUINO PRO (Placas), ARDUINO PRO MINI, ARDUINO MICRO, ARDUINO NANO (Modulos), ARDUINO STARTER KIT, ARDUINO BASIC KIT (KITs), ARDUINO MOTOR SHIELD (SHIELD).
- **Características mejoradas:** Contempla dispositivos de alta gama tales como ARDUINO MEGA, ARDUINO ZERO, ARDUINO DUE (Placas), ARDUINO PROTO SHIELD (SHIELD).
- **Internet de las cosas (IoT):** Contempla dispositivos embebidos adaptados para comunicacion inalambrica, ARDUINO YUN, ARDUINO MKR1000 (KIT), ARDUINO ETHERNET SHIELD, ARDUINO GSM SHIELD, ARDUINO WIFI SHIELD 101.
- **Portátil:** Contempla dispositivos integrados el cual incluyen todos los modulos necesarios para adaptarse a gran variedad de aplicaciones, ARDUINO GEMMA, LILYPAD ARDUINO USB, LILYPAD ARDUINO MAIN BOARD, LILYPAD ARDUINO SIMPLE, LILYPAD ARDUINO SIMPLE SNAP (Placas).
- **Impresión 3D:** Dispositivo adaptado para la implementacion de una impresora 3D, MATERIA 101.

2.2.2. Procesador Digital de Señal (DSP)

Las funciones fundamentales para el desarrollo del ECG es necesario el procesamiento de señales analógicas, para esto existen dispositivos dedicados llamados DSP. Un DSP es un microprocesador que posee una CPU de gran potencia y que está preparado para la realización del mayor número de operaciones aritméticas en el menor tiempo posible. Los DSP son similares a otros microprocesadores y microcontroladores que se encuentran en el mercado con la salvedad de que su circuitería interna ha sido optimizada para la realización de funciones tales como el filtrado, la correlación, el análisis espectral, etc., de una señal digital o analógica de entrada al sistema. Esta característica es la que le confiere al DSP, en el entorno de las telecomunicaciones y de la tecnología de la información, la importancia que tiene. Inicialmente, los DSP disponían de un mercado reducido (sólo se empleaban en sistemas que requerían una elevada potencia de cálculo, como ocurre en instrumentación electrónica de precisión, osciloscopios digitales, etc.) aunque, con la revolución y expansión relacionada con las telecomunicaciones, esta tendencia ha cambiado claramente. Las aplicaciones basadas en DSPs son innumerables y cada día van surgiendo más posibilidades cubriendo, en la actualidad, prácticamente todos los campos de la industria (telecomunicaciones, control, instrumentación, análisis de imagen y voz,

automóvil, medicina, etc.), la aplicación de un DSP en el desarrollo de un dispositivo como ECG Holter, representaría una mejora significativa ya que este dispositivo realiza el trabajo de análisis y tratamiento de señales analógicas, quitándole este trabajo a la CPU, este tipo de dispositivos por lo general son de alto costo. [3]

La adquisición de un DSP resultaría de alto costo monetario para la implementación del proyecto, siendo el dispositivo ideal, pero inviable, sin contar que muchos de estos dispositivos están bajo patentes.

2.3. ECG Holter

La monitorización continua del ritmo cardíaco durante 24 horas (Holter) es un registro de 2 ó 3 derivaciones ECG, que permite el estudio de los pacientes con diversos síntomas, tales como palpitaciones, síncope, disnea, angina que sugieren la presencia de una patología arrítmica especialmente.

Su función consiste en la instalación de una máquina de registro continuo de 2 ó 3 derivaciones ECG, que posteriormente es revisada en un sistema lector. El paciente debe consignar los síntomas que presente durante las 24 horas que estará con la máquina. Esto permite relacionar el síntoma consignado con el ritmo en ese período. Además mediante programas computacionales, el sistema lector puede identificar automáticamente los períodos de frecuencia cardíaca mayor y menor, los períodos de ritmo irregular, los cambios en la configuración del complejo QRS, etc. [4]

2.3.1. Análisis de la señal Electrocardiográfica

Es necesario conocer la naturaleza de las señales que deben procesarse para el desarrollo del dispositivo. Una de las labores más importantes en un trabajo de este tipo es determinar las características de la señal a adquirir y a tratar. En este sentido hay dos enfoques equivalentes que permiten obtener información en dominios diferentes. Se denomina Dominio del Tiempo al análisis de la Señal en relación al Tiempo y Dominio de la Frecuencia al análisis de las componentes frecuenciales de la señal. Esto es relevante tanto desde el punto de vista electrónico como desde el punto de vista computacional. Respecto del Dominio del Tiempo, como se muestra en la Figura 2.2, en todo trazo electrocardiográfico se identifican algunas formas de onda bien conocidas. Para una descripción detallada del significado de cada una de las formas de onda relevantes. Solo se considera relevante el complejo QRS (formado por las ondas Q, R y S y cuya polaridad depende de la derivación que se esté registrando), cabe señalar que un parámetro importante a ser medido en electrocardiografía es la duración del ciclo cardíaco, con lo cual es posible determinar la frecuencia cardíaca, elemento clave en el diagnóstico electrocardiográfico. Un esquema para realizar esta medición consiste en obtener el inverso multiplicativo de la duración de dos complejos QRS sucesivos. [5]

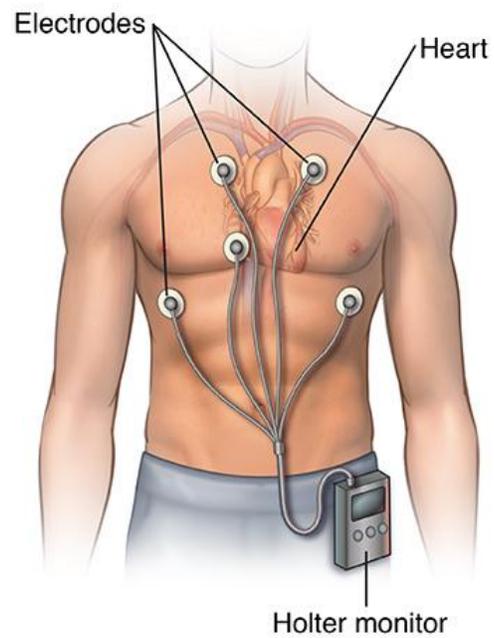


Figura 2.1: Monitor ECG Holter con electrodos
Fuente: premierheartcarett.com (2017)

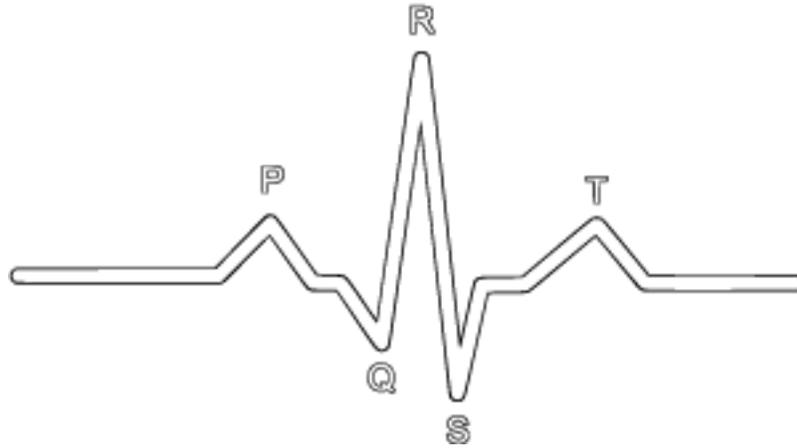


Figura 2.2: Onda PQRST
Fuente: Elaboracion propia

Basado en los estudios de los mismos autores, en términos del dominio frecuencial, que muestran que la señal electrocardiográfica tiene un espectro frecuencial que va desde los 0 Hz a los 100 Hz resumiendo en el siguiente cuadro.

Cuadro 2.1: Comparativa en frecuencia y amplitud (ECG,EEG,EMG)

Señal	Amplitud (mv)	Rango de Frecuencia (Hz)
ECG	0.02 - 5.0	0.05 - 100
EEG	0.0002 - 0.3	DC - 150
EMG	0.1 - 5.0	DC - 10000

Cabe aclarar que “DC” hace referencia a Componente Directo es decir que no presenta frecuencia (0 Hz), también es necesario saber que, EEG es un análisis que se utiliza para detectar anomalías relacionadas con la actividad eléctrica del cerebro, la electromiografía (EMG) es una técnica para la evaluación y registro de la actividad eléctrica producida por los músculos esqueléticos este análisis es llamado también como electromiograma.

2.3.2. Diseño digital Holter ECG

El mismo autor planteó un diseño lógico para el desarrollo del dispositivo, este dispositivo representa un trabajo previo el cual da un punto de apoyo referencial para el diseño e implementación del ECG. Este diseño del circuito del electrocardiógrafo digital fue analizado varios diseños de electrocardiógrafos presentes en el mercado. Resulta fundamental ver la construcción del diseño desde un punto de vista modular. De esta forma se

pueden aislar problemáticas de manera individual. En la Figura 2.3 se puede apreciar un diagrama de bloques del electrocardiógrafo que se construyó en este trabajo. Las señales producidas por fenómenos bioeléctricos son potenciales eléctricos relativamente pequeños y requieren sofisticados amplificadores para facilitar su cuantificación. El propósito principal del amplificador es incrementar el nivel de la señal que se obtiene con los electrodos excluyendo cualquier tipo de interferencia. Tal y como se señala, la cuantificación puede realizarse usando configuraciones Unipolares o Bipolares. En el modo de cuantificación unipolar se mide la diferencia entre la señal y una tierra común. En el modo bipolar se mide la diferencia de dos orígenes de voltaje (en este caso los electrodos), respecto a una tierra común. En este caso los potenciales de interferencia generados en el punto de cuantificación aparecen en la entrada del amplificador como señales de interferencia en modo común. Uno de los orígenes de ruido que más interfieren en la señal ECG es la interferencia capacitiva debido al cuerpo del paciente. Este voltaje de interferencia se acopla a la señal electrocardiográfica, llegando a tener valores de 2.4 V aproximadamente. Este valor está muy por encima de los valores en los que fluctúa la señal electrocardiográfica (1 mV y 10 mV). Además de la interferencia capacitiva debido al cuerpo del paciente, es preciso señalar la interferencia capacitiva debido al equipo de medida, que se produce por la fuente de alimentación del equipo al cual llegan las señales electrocardiográficas. Otra fuente de ruido es la denominada interferencia inductiva, que es causada por la red eléctrica, la cual produce campos magnéticos que varían con el tiempo, los que a su vez inducen voltajes en la red formada por los electrodos del paciente, en la siguiente figura podemos ver el diseño planteado.

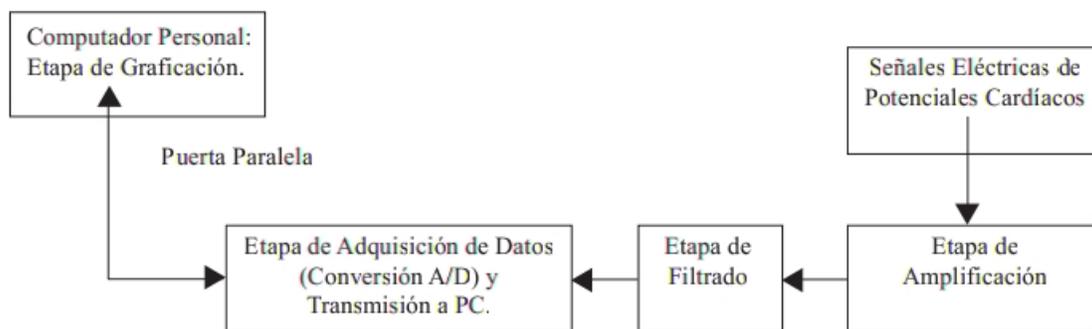


Figura 2.3: Diseño lógico Electrocardiograma
Fuente: Chritian Vidal, Leopoldo Pavesi (2004)

Este tipo de implementación contempla los módulos básicos para el desarrollo del dispositivo, cabe destacar que este representa una implementación alternativa mediante unos módulos de adquisición y tratamiento de las señales y luego la etapa de graficación, este último es directamente graficado en el computador, sin necesidad de memorias ni

microcontroladores.

2.3.3. Características necesarias para el desarrollo del Holter ECG

El Laboratorio del Postgrado de Instrumentación de la Facultad de Ciencias posee varios modelos de microprocesadores y microcontroladores, para uso académico. Los principales microcontroladores que se disponen son los PIC de la familia 18FXXXX y Arduino Mega basados en los microcontroladores ATmel Mega, la comparación de las necesidades planteadas en este trabajo se harán bajo estos dos dispositivos, y así poder determinar la mejor opción en términos de rendimiento, facilidad de uso, documentación y alcance.

El proceso básico de procesamiento de la señal, será en primera instancia la captura, procesamiento y por último guardado de los registros, comportándose de manera similar a un DSP (Procesador digital de señal), en más detalle.

Captura

La señal generada por el corazón son señales que entran en la categoría “analógicas”, debido a su naturaleza variante, el arduino es un dispositivo totalmente digital sin embargo para poder manejar este tipo de señales, cuenta con un subsistema de entrada salida que es encargado de representar la señal analógica en un valor numérico, conocido como “convertidor analógico digital” o (CAD), dicho módulo, en un determinado momento realiza una captura luego cuantifica la intensidad de la señal y da como resultado un valor numérico. Un fenómeno eléctrico cualquiera que sea, tiene una determinada duración, para poder representar de manera más adecuada el comportamiento de la señal, el dispositivo debe realizar un gran número de capturas, este proceso es lo que se conoce como modulación.

La captura no es posible realizarla directamente desde los electrodos al microcontrolador, las señales generadas por el corazón generalmente son del orden de los milivoltios (mv), por lo cual no son magnitudes manejables para los puertos de entrada y salida del microcontrolador, cuyas magnitudes manejables rondan desde 0.7V hasta los 5V. Para esto, ya se dispone de un elemento de hardware intermedio que funciona como amplificador, es decir es capaz de elevar los voltajes del orden de los mv hasta el rango manejable por el microcontrolador.

Existen varios métodos para la codificación de una señal analógica, siendo la modulación por impulsos codificados o mejor conocida como PCM la más popular, el módulo CAD del microcontrolador, debe ser capaz de realizar capturas con una resolución mínima de 9 bits (0 - 511) para evitar así la menor pérdida de información posible.

Procesamiento

Una vez obtenida la señal cuantificada, es necesario que exista un proceso que permita tomar dichos datos y guardarlos en la memoria, al mismo tiempo que se realiza una llamada de conversión, de esta manera el CPU del microcontrolador obtiene el valor de la conversión, guarda el resultado e invoca otra llamada de conversión, este proceso se debe hacer lo más rápido posible, a fin de perder la menor cantidad tiempo entre cada una de las llamadas, y tener la mejor representación posible del fenómeno en cuestión. En este punto es donde entra en juego el rendimiento de la CPU, los microcontroladores ATmega debido a su arquitectura de CPU, puede llegar a un reloj máximo de 16 Mhz, con un rendimiento de 16 MIPS (millones de instrucciones por segundo), es decir una instrucción es ejecutada un 1 ciclo de instrucción, esta ventaja es otorgada gracias a su arquitectura en comparación a su competidor PIC, también es necesario aprovechar al máximo la utilización de la CPU a nivel de software, por ejemplo mientras se invoca la llamada a la conversión analógico-digital se realiza el guardado en una memoria externa el valor de la captura anterior, solapando así las tareas para aprovechar de mejor manera la utilización de este recurso.

Almacenamiento

Es necesario que todo lo descrito anteriormente sea almacenado para su posterior uso, en este caso el arduino cuenta con una interfaz de interconexión para tarjetas de memoria Flash (SD). Cuando el dispositivo es activado y empieza el proceso de captura, dichas capturas continuamente son guardadas en la memoria SD, el tiempo de escritura en la memoria SD puede variar, dependiendo de la clase que esta pertenezca, la clasificación de las memorias pueden variar desde la clase 1 hasta 10, las clases de mayor clasificación tienen una velocidad de escritura/lectura mucho mayores con respecto a la anterior, por lo tanto es deseable una memoria con la mayor calificación posible.

Aunque la velocidad de escritura no solamente está limitada por la memoria está también influenciada por parte de la biblioteca que gestiona este recurso, la velocidad de escritura/lectura se encuentra determinada por muchos factores, el software debe hacer el mejor uso posible de la memoria, sin importar sus características.

Alimentación eléctrica

La familia de microcontroladores PIC 18FXXXX poseen un rango de voltaje de operación muy similar al otros dispositivos tales como son los ATmel Mega, es decir un voltaje de operación de 4.5 – 5V. Una de las grandes ventajas que ofrecen los dispositivos Arduino, es que poseen un circuito regulador de voltaje embebido en la placa del microcontrolador, de esa manera no es necesario regular el voltaje de alimentación con algún otro integrado externo, ahorrando así tiempo en diseño e implementación. Así mismo este circuito funciona como mecanismo de protección del microcontrolador ante un fallo de alimentación,

o corto circuito en muchos de los pines de conexión con la placa.

Puertos E/S

En número de puertos de E/S que disponen los microcontroladores de ambos fabricantes varía según el modelo que se utilice, sin embargo las características generales de los puertos de E/S, son muy similares, tales como manejo de señales analógicas, corriente máxima de carga por pin, aproximadamente unos 40 mA, y resistencias “pull-up”, este último es una tecnología que implementan ambos fabricantes el cual consiste en una resistencia integrada en el microcontrolador para proteger los puertos, y por lo general son manejados totalmente por software. En base a las bondades de la placa arduino, esta ofrece una amplia gama de módulos externos complementarios que se adaptan perfectamente a la placa, pudiendo así ampliar su funcionabilidad de manera rápida, gracias al diseño y disposición de los puertos de E/S.

En base a la comparación de las características deseables para el desarrollo de este trabajo, la gran documentación libre, popularidad, fácil aprendizaje y el rápido desarrollo, que la gran mayoría de los dispositivos Arduino ofrecen, se decidió utilizar este dispositivo para la elaboración del Holter, específicamente un Arduino Mega 2560, en más detalle.

2.4. Especificaciones y características Arduino Mega 2560

El arduino Mega 2560 es un microcontrolador basado en el ATmega2560, tiene 54 entradas/salidas digitales, 16 entradas analógicas, 4 módulos UART (interfaz de comunicación serial), con un reloj principal de 16 MHz, y conexión USB 2.0. Con mayor detalle se estructura siguiente tabla.

Cuadro 2.2: Especificaciones técnicas Arduino Mega 2560

Microcontrolador	ATmega2560
Voltaje de operación	5V
Voltaje de entrada (recomendado)	7 - 12V
Voltaje de entrada (máximo)	6 - 20V
E/S Digitales	54 (de los cuales 15 proveen salida por PWM)
Entradas analógicas	16
Corriente de E/S (5V)	20 mA
Corriente de E/S (3.3V)	50 mA
Memoria no volátil (Flash)	256 KB el cual 8KB es para el bootloader
SRAM	8 KB
EEPROM	4 KB
Reloj de CPU	16 Mhz
Largo	101.52 mm
Ancho	53.3 mm
Peso	37 g

2.4.1. Alimentación eléctrica

Es importante considerar el consumo eléctrico del microcontrolador ya que se desea implementar un dispositivo de bajo consumo eléctrico, a fin de ser lo más portable posible. El Mega 2560 puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

La alimentación (no USB) externa puede venir con un adaptador de CA a CC o batería. El adaptador se puede conectar al conectar un enchufe de 2,1 mm de “centro positivo” en el conector de alimentación de la placa. La alimentación por batería se puede conectar en los pines de GND y el pin Vin.

La tarjeta (arduino) puede funcionar con un suministro externo de 6 a 20 voltios. El pin de 5V puede ser alimentado con menos de cinco voltios pero la tarjeta puede volverse inestable. Si se utiliza más de 12V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios, los pines de alimentación son los siguientes.

- **Voltaje de entrada (Vin):** El voltaje de entrada a la junta cuando se trata de

utilizar una fuente de alimentación externa (5 voltios de la conexión USB u otra fuente de alimentación regulada). Es posible suministrar tensión a través de este pin, o, si el suministro de tensión a través de la toma de poder, acceder a él a través de este pin.

- **5V:** Este pin como salida una 5V regulada del regulador en el tablero. El tablero puede ser alimentado ya sea desde la toma de alimentación de Corriente Continua (CC) (7 - 12 V), el conector USB (5V), o el pin VIN del tablero (7-12V). El suministro de tensión a través de los pines de 5V o 3.3V no pasa por el regulador, y puede dañar su tablero. No aconsejamos ella.
- **3V3:** Un suministro de 3,3 voltios generada por el regulador integrado. Corriente máxima es de 50 mA. Pin IOREF. Este pin en el tablero proporciona la referencia de tensión con la que opera el microcontrolador. Un mecanismo configurado puede leer el voltaje pin instrucción IOREF y seleccionar la fuente de alimentación mas adecuada o habilitar reductores de voltaje en las salidas para trabajar con 5V o 3.3V.
- **GND:** Patillas de tierra.

2.4.2. Memoria

El Atmega2560 tiene 256 KB de memoria flash para almacenar el código (de los cuales 8 KB se utiliza para el cargador de arranque), 8 KB de SRAM y 4 KB de EEPROM (que puede ser leído y escrito con la biblioteca EEPROM), cabe destacar que cada tipo de memoria cumple funciones distintas.

- **Flash:** Es un tipo de memoria no volátil y es utilizada para almacenar el programa compilado, de los cuales ya de antemano se utilizan 8KB para el programa de arranque.
 - **SRAM:** Por su acrónimo en inglés (Static Random Access Memory), es una tipo de memoria de muy alta velocidad volátil, se denomina estática, ya que la memoria no necesita ejecutar “ciclos” de actualización del contenido de cada una de las celdas, que pueden contener un 1 o 0 lógico, su función es almacenar información que requiera la CPU para la ejecución del programa.
 - **EEPROM:** Es un tipo de memoria no volátil muy similar a la RAM, y permite almacenar datos que sea necesario su posterior uso, incluso después de haber apagado el dispositivo, se utiliza para guardar variables de estado que permitan al programa ejecutarse sin perder datos o cualquier otro tipo de dato que sea necesario que prevalezca en el tiempo.
-

2.4.3. Puertos de E/S

Cada uno de los 54 pines digitales en el arduino Mega se puede utilizar como una entrada o salida, utilizando la función `pinMode()`, `digitalWrite()`, o la función `digitalRead()`, que operan a 5 voltios. Cada pin puede proporcionar o recibir 20 mA, como condición de funcionamiento recomendada y tiene una resistencia de pull-up (desconectado por defecto) de 20 a 50 k ohmios. Un máximo de 40 mA es el valor que no debe superarse para evitar daños permanentes en el microcontrolador.

2.5. Software Libre

Software libre es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es decir, el software libre es una cuestión de libertad, no de precio. Para entender el concepto, piense en libre como en libre expresión, no como "gratis". En inglés a veces se dice "software free", en lugar de "free software", para mostrar que no queremos decir que es gratis [6]. Aunque el software libre nació como una respuesta ante el movimiento que conservan y protegen los derechos de autor, también conocidos como "Copyright", el software libre no renuncia a estos principios ya que solo plantea una alternativa que brinda a la comunidad la capacidad de desarrollo y mejoramiento de un mismo proyecto basado en libertades, el software libre también definió sus pautas legales conocidas como "Copyleft". Las libertades inherentes al software libre son. [6]

- **Libertad 0:** Libertad para ejecutar el programa sea cual sea el propósito.
- **Libertad 1:** Libertad para estudiar el funcionamiento del programa y adaptarlo a tus necesidades - el acceso al código fuente es condición indispensable para esto.
- **Libertad 2:** Libertad para redistribuir copias y ayudar así a tu prójimo.
- **Libertad 3:** Libertad para mejorar el programa y luego publicarlos para el bien de toda la comunidad - el acceso al código fuente es condición indispensable para esto.

2.5.1. Hardware Libre

El hardware abierto o libre toma las mismas ideas del Software libre para aplicarlas en su campo. Es una propuesta casi tan antigua como la del software libre, sin embargo su empleo no es tan directo. Compartir diseños hardware es más complicado. No hay una definición exacta. Incluso el propio Richard Stallman (Fundador del movimiento Libre) afirma que las ideas del software libre se pueden aplicar a los archivos necesarios para su diseño y especificación (esquemas, PCB, etc.), pero no al circuito físico en sí. Al no existir una definición clara de hardware abierto, cada autor lo interpreta a su manera.

Se han creado licencias, algunas de las cuales están todavía en desarrollo. Para abordar el estudio comenzaremos estableciendo una primera clasificación, según su naturaleza. Existen los siguientes tipos: Hardware estático. Conjunto de elementos materiales de los sistemas electrónicos. Tiene una existencia física (se puede “tocar”). Esta propiedad no la tiene el software, por lo que surgen una serie de problemas que se discuten en el apartado. Comenzado en 1997 por Microbótica S.L. Se propone una definición práctica de hardware estático libre y se establece una clasificación de los diseños, en función de las herramientas de desarrollo empleadas, que limitan en mayor o menor medida las libertades ofrecidas por el autor. Se considera que hardware abierto y hardware libre son sinónimos, y se utilizan indistintamente. Sin embargo, existe ambigüedad en la lengua Inglesa. El término free hardware puede significar tanto hardware libre como hardware gratuito. Esto último es incorrecto. El Hardware reconfigurable. Es el que se describe mediante un lenguaje HDL, (Hardware Description Language, lenguaje de descripción hardware) y que permite especificar con todo detalle su estructura y funcionalidad. A partir de este código se generan unos archivos de configuración (bitstreams) que son aquellos que se pueden proteger bajo licencia libre. [7]

2.5.2. Copyright

El copyright es usado para indicar que una obra está sujeta al derecho de autor. El derecho de autor (del francés *droit d’auteur*) es un conjunto de normas y principios que regulan los derechos morales y patrimoniales que la ley concede a los autores (los derechos de autor), por el solo hecho de la creación de una obra literaria, artística o científica, tanto publicada o que todavía no se haya publicado. En el Derecho anglosajón se utiliza la noción de copyright (traducido literalmente como derecho de copia) que, por lo general, comprende la parte patrimonial de los derechos de autor (derechos patrimoniales). [7] Una obra pasa al dominio público cuando los derechos patrimoniales han expirado. Esto sucede habitualmente trascurrido un plazo desde la muerte del autor (post mortem auctoris). En Venezuela, el plazo es de 60 años desde la muerte del autor. Dicha obra entonces puede ser utilizada en forma libre, respetando los derechos morales, el órgano encargado de regular la propiedad intelectual es el Servicio Autónomo de la Propiedad Intelectual (SAPI), que forma parte del Ministerio del Poder Popular para la Industria y Comercio. [8]

2.5.3. Copyleft

El copyleft es un método general para hacer un programa (u otro trabajo) libre, y requiere que todas las versiones modificadas y extendidas del programa sean libres también. La forma más sencilla de hacer un programa software libre es ponerlo en el dominio público, sin copyright. Esto permite a la gente compartir el programa y sus mejoras, si así lo desean. Pero también permite a las personas que no cooperan convertir el programa en software privativo. Ellos pueden hacer cambios, muchos o pocos, y distribuir

el resultado como un producto patentado. Las personas que reciben el programa con esas modificaciones no tienen la libertad que el autor original les dio, ya que el intermediario se la ha retirado. Copyleft también proporciona un incentivo para que otros programadores puedan unirse al software libre.

2.6. Modulación por Codificación de Pulsos (PCM)

Debido a la implementación de los módulos conversores analógicos digitales de los microcontroladores este tipo de digitalización de señal resulta ser la más popular. La Modulación por Codificación de Pulsos (PCM) es la forma más simple de codificación de forma de onda, este método se utiliza para codificar señales analógicas (por ejemplo la voz humana) en una señal digital. La señal digital se utiliza para reconstruir la señal analógica. La transmisión analógica no es particularmente eficiente, cuando la relación de señal a ruido de una señal analógica se deteriora debido a la atenuación, la amplificación de la señal también amplifica el ruido por otra parte las señales digitales se separan más fácilmente del ruido y se pueden regenerar en su estado original, la conversión de señales analógicas en señales digitales, elimina los problemas causados por la atenuación.

La señal digital se utiliza para reconstruir la señal analógica, la precisión con la que la señal analógica puede ser reproducida depende en parte del número de bits utilizados para codificar la señal original, la modulación de código de pulso es una extensión de la modulación de impulsos en amplitud (PAM), en el que una señal muestreada se compone de un tren de impulsos, donde cada impulso corresponde a la amplitud de la señal en el tiempo de muestreo correspondiente (la señal es modulada en amplitud). Cada valor de la muestra analógica se cuantifica en un valor discreto para la representación como una palabra de código digital. La modulación de código de pulso es la técnica de conversión de uso más frecuente de analógico a digital, y se define en la especificación UIT-T G.711. Las partes principales de un sistema de conversión son el codificador (el convertidor de analógico a digital) y el decodificador (el convertidor de digital a analógico). El codificador / decodificador combinado se conoce como un codec. [9]

2.6.1. Funciones de un codificador decodificador combinado

- **Captura:** La captura es el proceso de lectura de los valores de la señal analógica filtrada a intervalos de tiempo discretos (es decir, a una frecuencia de captura constante, llamada “frecuencia de muestreo”). Un científico llamado Harry Nyquist descubrió que la señal analógica original podría ser reconstruido si se tomaron muestras suficientes. Se demostró que si la frecuencia de muestreo es al menos dos veces la frecuencia más alta de la señal analógica de entrada, la señal podría ser reconstruida utilizando un filtro de paso bajo.
- **Cuantificación:** Cuantificación es el proceso de asignar un valor discreto de una

gama de valores posibles para cada muestra obtenida. El número de valores posibles dependerá del número de bits usados para representar cada muestra. Cuantificación se puede lograr por cualquiera de las dos formas, es decir, por redondeo la señal hacia arriba, hacia abajo, o truncar la señal al valor más cercano que es menor o mayor (según sea el caso) que la muestra real. El proceso da como resultado una forma de onda escalonada se asemeja a la señal fuente. La diferencia entre la muestra y el valor asignado al que se conoce como el ruido de cuantificación (o error de cuantificación).

- Codificación:** La codificación es el proceso de representar los valores de muestra como un número binario en el rango de 0 a n . El valor de n se elige como una potencia de 2, dependiendo de la precisión requerida. El aumento de n reduce el tamaño del paso entre los niveles de cuantificación adyacentes y por lo tanto reduce el ruido de cuantificación. El lado negativo de esto es que la cantidad de datos digitales requeridos para representar la señal analógica aumenta.

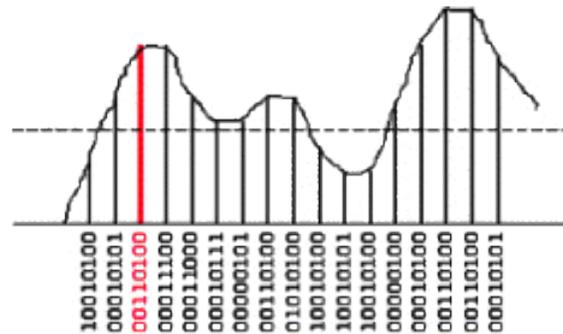


Figura 2.4: Codificación PCM de 8 bits

Fuente: duiops.net PCM (2017)

2.6.2. Expansión - Compresión

El trabajo con niveles de señal muy pequeñas (en comparación con el intervalo de cuantificación) puede producir más errores, la compresión se puede utilizar para aumentar la precisión de tales señales. Este es el proceso de distorsión de la señal analógica de un modo controlado antes de que la cuantificación se lleve a cabo, se comprimen los valores más grandes en la fuente y luego la expansión de ellos en el extremo receptor. Hay dos estándares utilizados: “A-law” en Europa, y “u-law” en los EE.UU. Las muestras de las señales analógicas de entrada se comprimen en segmentos logarítmicos, cada segmento se cuantifica, y se codifica usando cuantificación uniforme. El proceso de compresión es logarítmico, la compresión aumenta a medida que el aumento de muestra señales (las señales de muestra más grandes se comprimen más de las señales de muestra más pequeños, haciendo que el ruido de cuantificación aumente a medida que se incrementa la señal de

la muestra). Un incremento logarítmico en el ruido de cuantificación en todo el rango dinámico de una señal de muestreo de entrada da una relación de señal a ruido que es casi constante en una amplia gama de niveles de entrada. A razón de ocho bits por muestra (64 kbits por segundo) da una señal reconstruida que está muy cerca del original. Las ventajas de este sistema son la baja complejidad, poca demora, y reproducción de alta calidad de voz. Las desventajas son una relativamente alta velocidad de bits y una alta susceptibilidad a los errores de canal.

2.7. El Corazón Humano

El corazón es una víscera cónica, con la punta dirigida hacia abajo y a la izquierda, situada en la cavidad torácica, entre ambos pulmones. El corazón humano está constituido por el miocardio (pared de fibras musculares), unidas por tejido conjuntivo y revestido exterior e interiormente por una membrana serosa, llamada pericardio y encorado, respectivamente.

El corazón se contrae (sístole) 70-80 veces por minuto, mediante el acotamiento de las fibras musculares; entre las contracciones se establecen periodos de relajación (diástole) en los que la sangre penetra en las cavidades. Estas contracciones están provocadas por un sistema nervioso autónomo, regulado por el sistema vegetativo bajo el gobierno de un núcleo situado en el bulbo (centro cardíaco) y por influencias hormonales (adrenalina, etc.).

Funcionalmente, consta de dos partes: el corazón derecho (formado por la aurícula y Ventrículo derecho, separados por la válvula tricúspide) recibe la sangre de todo el organismo por las venas cavas que desembocan en la aurícula, pasando la sangre al ventrículo al abrirse la válvula. Desde allí, es impulsada a los pulmones donde es oxigenada, convirtiéndose en sangre arterial que, por las venas pulmonares, ingresa en el corazón izquierdo (aurícula y ventrículo izquierdos, separados por la válvula Mitral) donde sigue un recorrido análogo; el ventrículo izquierdo la impulsa luego a la arteria aorta, que la distribuye a todo el cuerpo.

El paso de la sangre de ambos ventrículos a las respectivas arterias está regulado por las válvulas sigmoideas. La contracción de ambos ventrículos se efectúa simultáneamente; las aurículas también se contraen, lo que hacen antes que los ventrículos, ayudando con ello a pasar la sangre hacia éstos. [10]

2.7.1. Anatomía del Corazón

El corazón está situado en la cavidad torácica y ocupa la región intermedia entre las dos regiones pleuro-pulmonares. Su forma es la de una pirámide triangular. El eje mayor de esta pirámide, es decir, la dirección del corazón varía según la forma del tórax. Cuando el tórax tiene unas dimensiones medias, tanto el corazón como su eje mayor se aproximan a la horizontal y se dirigen oblicuamente hacia adelante, hacia la izquierda y ligeramente

hacia abajo. A causa de esto, la base del corazón normal, que está en la parte posterior, mira a la vez hacia atrás y a la derecha; el vértice está inclinado hacia adelante y a la izquierda.

La consistencia del corazón es dura; su coloración rojiza. Su peso aumenta gradualmente con la edad. Es un poco mayor en el hombre que en la mujer; alcanza en la edad adulta, por término medio, 270 gramos en el hombre y 260 gramos en la mujer.

En la Figura 2.5 se puede apreciar la forma y disposición del corazón.

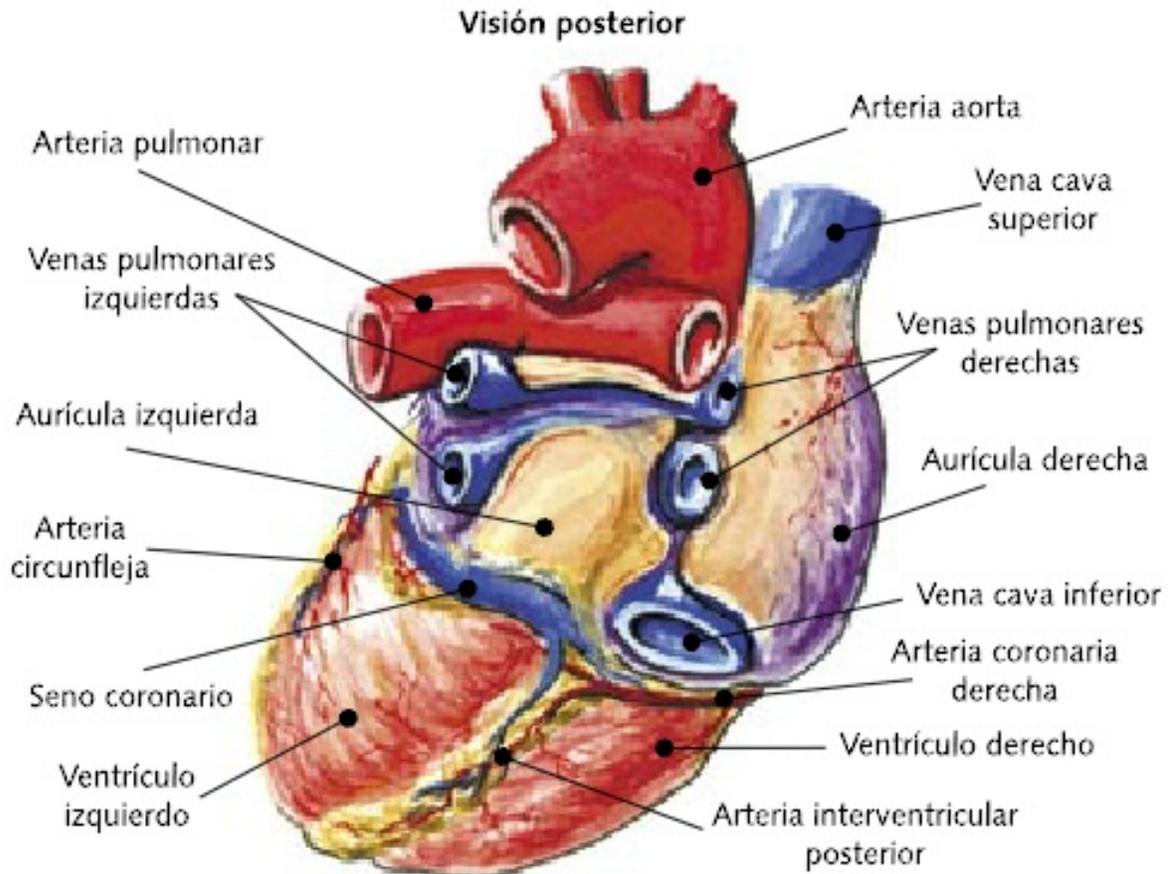


Figura 2.5: Vista posterior Corazón Humano

Fuente: Frank H. Netter, MD Atlas de anatomía humana. 4ta edición, 2007

2.7.2. Configuración exterior del corazón

El corazón presenta a causa de su forma, tres caras, tres bordes, una base y un vértice. El corazón está formado de cuatro partes: las aurículas derecha e izquierda y los ventrículos

derecho e izquierdo. La aurícula derecha está situada detrás del ventrículo derecho; la aurícula izquierda está detrás del ventrículo izquierdo.

Los límites de las aurículas y de los ventrículos están indicados en la superficie exterior del corazón por los surcos interventricular, interauricular y aurícula-ventricular.

- **Caras:** Las caras son, cara anterior o esternocostal, otra inferior o diafragmática y otra lateral izquierda.
- **Bordes:** Las tres caras del corazón están separadas por tres bordes, uno derecho y dos izquierdos. El borde derecho, situado entre la cara anterior y la cara inferior, es agudo. Los bordes izquierdos son redondeados y separan la cara lateral izquierda de las caras anterior e inferior.
- **Base:** La base del corazón está constituida únicamente por las aurículas. Está dividida en dos segmentos, derecho e izquierdo, por el surco interauricular.
- **Vertice:** El vértice o punta del corazón está dividido en dos partes por una ligera depresión que une el surco interventricular anterior al surco interventricular posterior; la derecha, pequeña, corresponde al ventrículo derecho, la izquierda, más voluminosa, pertenece al ventrículo izquierdo y ocupa el vértice del corazón.

2.8. El ciclo cardiaco

El ciclo cardíaco es la sucesión de acontecimientos auriculares y ventriculares que se repiten en cada contracción. Es un fenómeno complejo y de extraordinaria rapidez. Se puede dividir el ciclo cardíaco en sístole y diástole. A su vez estos períodos se dividen en diversas fases.

2.8.1. Sístole auricular

El primer acontecimiento del ciclo cardíaco es la descarga del nodo sinusal, que despolariza en primer lugar ambas aurículas, dando lugar a la onda P del electrocardiograma. Tras la activación eléctrica la aurícula se contrae, aumenta la presión intra-auricular, y como en este momento las aurículas y ventrículos están en comunicación porque las válvulas aurículo ventriculares están abiertas, se envía una cantidad adicional a la sangre en el ventrículo, induciendo una pequeña elevación del volumen ventricular y un aumento transitorio de la presión auricular y ventricular.

2.8.2. Fase de eyección

La función primordial de bomba del corazón se cumple en la fase de eyección, en la que ambos ventrículos vacían el 70 % de su contenido, expulsando el volumen de eyección en

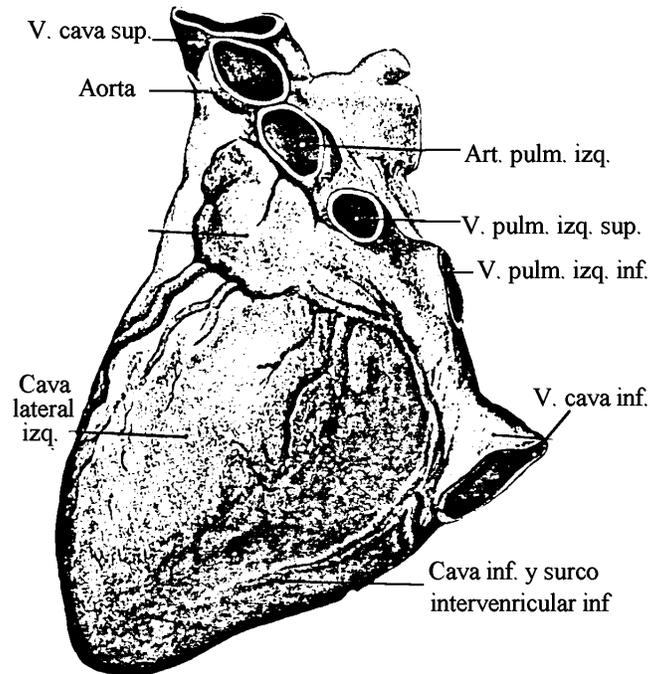


Figura 2.6: Vértices y Cavidades del Corazón

Fuente: Intrumentación Biomédica (2016)

contra de unas altas resistencias el ventrículo izquierdo, y en contra de bajas resistencias el derecho. El ventrículo no se vacía por completo durante la sístole. Siempre queda una cierta cantidad de sangre al final de la sístole, unos 60 ml que se denomina volumen sistólico final o volumen residual. El volumen sistólico de eyección es, pues, la diferencia entre el volumen diastólico final y el volumen sistólico final, que en nuestro ejemplo es de 70 ml. La relación entre el volumen de eyección y el volumen diastólico final se conoce como fracción de eyección, que es uno de los índices más precisos de la función ventricular. Determinada por medios angiográficos, la fracción de eyección varía entre el 60-75 %.

En condiciones de estrés, bajo la influencia de las catecolaminas, el corazón se puede contraer más poderosamente y vaciar su contenido casi por completo, siendo entonces la fracción de eyección de hasta el 80-85 %. Cuando la función ventricular está deprimida, aumenta el volumen sistólico final y disminuye la fracción de eyección, al recurrir al corazón a la ley de Starling. La función ventricular es anormal cuando la fracción de eyección es inferior al 50 % y está gravemente deprimida cuando es del 30 %.

2.8.3. Periodo diastólico. Fase de relajación isovolumétrica

Al cerrarse la válvula aórtica, el ventrículo vuelve a ser una cámara cerrada. Esta fase se denomina de relajación isovolumétrica, porque el volumen sistólico final o volumen residual no varía hasta que se abren las válvulas auriculoventriculares y comienza la fase siguiente de llenado. El comienzo de la diástole se suele referir a la producción del segundo tono. Que indica el final de la protodiástole y el principio de la fase de relajación isovolumétrica. El segundo tono se produce cuando la válvula aórtica se abomba y tensa hacia la cavidad ventricular, unos 30 ms después del cruce de presiones ventricular y aórtica. Inmediatamente después del cierre de la válvula aórtica y lo mismo ocurre con el ventrículo derecho- la presión ventricular excede todavía la presión auricular, ambas válvulas, mitral y tricúspide, permanecen cerradas y la sangre no entra ni sale de los ventrículos.

Como el ventrículo comienza a expandirse al relajarse la musculatura cardíaca, la presión ventricular desciende bruscamente, mientras que la presión aórtica, que disminuye gradualmente, se mantiene elevada durante toda la diástole. La fase de relajación isovolumétrica termina cuando se cruzan las presiones auriculares y ventriculares. En este momento, las válvulas auriculoventriculares se abren y comienza el llenado ventricular. Esta fase de diástole precoz es la imagen en espejo de la fase de contracción isovolumétrica. En este momento, el ventrículo izquierdo pasa de ser un sistema de alta presión a uno de baja presión. La onda T del ECG, que señala la repolarización ventricular, tiene lugar en esta fase de relajación isovolumétrica.

La fase de relajación isovolumétrica termina cuando se cruzan las presiones auriculares y ventriculares. En este momento, las válvulas auriculoventriculares se abren y comienza el llenado ventricular. Esta fase de diástole precoz es la imagen en espejo de la fase de contracción isovolumétrica. En este momento, el ventrículo izquierdo pasa de ser un sistema de alta presión a uno de baja presión.

La onda T del ECG, que señala la repolarización ventricular, tiene lugar en esta fase de relajación isovolumétrica.

2.8.4. Fase de llenado rápido ventricular

Cuando la presión ventricular desciende por debajo de la auricular, las válvulas auriculoventriculares se abren y los ventrículos comienzan a llenarse, muy rápidamente al principio y más lentamente después. El llenado ventricular se divide en tres fases: fase de llenado rápido o pasivo, fase de llenado lento o diástasis y fase de llenado auricular o activo, que ha sido considerado ya. La fase de llenado rápido, la más importante, es relativamente precoz y muy rápida, incluso más rápida que la fase de eyección.

El hecho de que la eyección ventricular sea prácticamente completa durante la sístole precoz y el llenado ventricular lo sea durante la diástole precoz, tiene un significado funcional importante: asegura que la eyección y el llenado no están seriamente compro-

metidos cuando aumenta la frecuencia ventricular tal como sucede en el ejercicio o en el estrés emocional. En la primera parte de la fase de llenado ventricular hay probablemente un mecanismo de succión activa.

2.8.5. Fase de llenado lento; diástasis o estasis

El ventrículo continúa llenándose lentamente en esta fase, por lo que la curva de presión intraventricular se horizontaliza con un ascenso muy lento. Esta fase, casi inmóvil desde el punto de vista hemodinámico, es larga en la bradicardia y desaparece en la taquicardia.

2.9. Nódulo Sinoauricular

El nódulo sinoauricular (abreviatura SA) o de Keith y Flack es una de las estructuras que componen el sistema de conducción del corazón; recibe el nombre común de «marcapasos del corazón». Normalmente, es donde se origina el impulso eléctrico que da origen a un latido cardíaco. Se encuentra en la pared posterolateral superior de la aurícula derecha, bajo la desembocadura de la vena cava superior. Es una estructura fusiforme desde el punto de vista morfológico, con un tamaño proporcional al tamaño del corazón, el cual varía entre 5mm a 30mm con una media de 15 mm de extensión por 5 mm de grosor variando de 1,5mm a 5mm. Histológicamente se encuentra formado por un conjunto de células (células P, células Transicionales y células de Purkinje) en íntima relación con fibras del sistema nervioso autónomo y fibras colágenas. El registro osciloscopio de los potenciales de acción de las células de este nodo muestran dos características importantes.

Ausencia de fase de reposo: Después de la repolarización, el potencial de membrana no se mantiene estable, sino que asciende lentamente, hasta que al llegar a los ± 40 milivoltios, comienza espontáneamente una nueva fase de excitación.

Baja velocidad en la fase de excitación: La entrada masiva de iones de sodio en el interior de la célula no es tan rápida como en las demás células cardíacas, sino que la fase de despolarización se instaura lentamente. El cambio de potencial tiene una velocidad de 1-2 voltios/segundo, frente a los 100-200 voltios/segundo en otras células.

Esta forma del potencial de membrana explica el automatismo de las células del marcapasos sinusal. No es necesaria la llegada de un estímulo para provocar el cambio de la permeabilidad de la membrana a los iones, sino que dicha permeabilidad al sodio primero y al potasio se instaura espontánea y cíclicamente a un ritmo de 60 a 100 veces por minuto.

Aunque el nódulo sinoauricular funciona automáticamente, su funcionamiento está regulado por el sistema nervioso vegetativo. [11]

2.10. Electrocardiografía

La electrocardiografía convencional estudia el registro de la actividad eléctrica cardíaca mediante unos aparatos denominados electrocardiógrafos. Esta actividad se registra en forma de una línea que presenta distintas inflexiones que se corresponden con el paso del estímulo eléctrico desde el lugar donde normalmente se origina (nodo sinusal), hasta los ventrículos a través del sistema específico de conducción (SEC). Se recuerda que dicho sistema está formado por el nodo sinusal, las vías preferenciales de conducción interauricular e internodal, el nodo auriculoventricular (AV), el haz de His, las dos ramas del haz de His y sus divisiones, con sus respectivas redes de Purkinje. Cuando el estímulo llega a la unión Purkinje-músculo, se produce el acoplamiento excitación-contracción. [12]

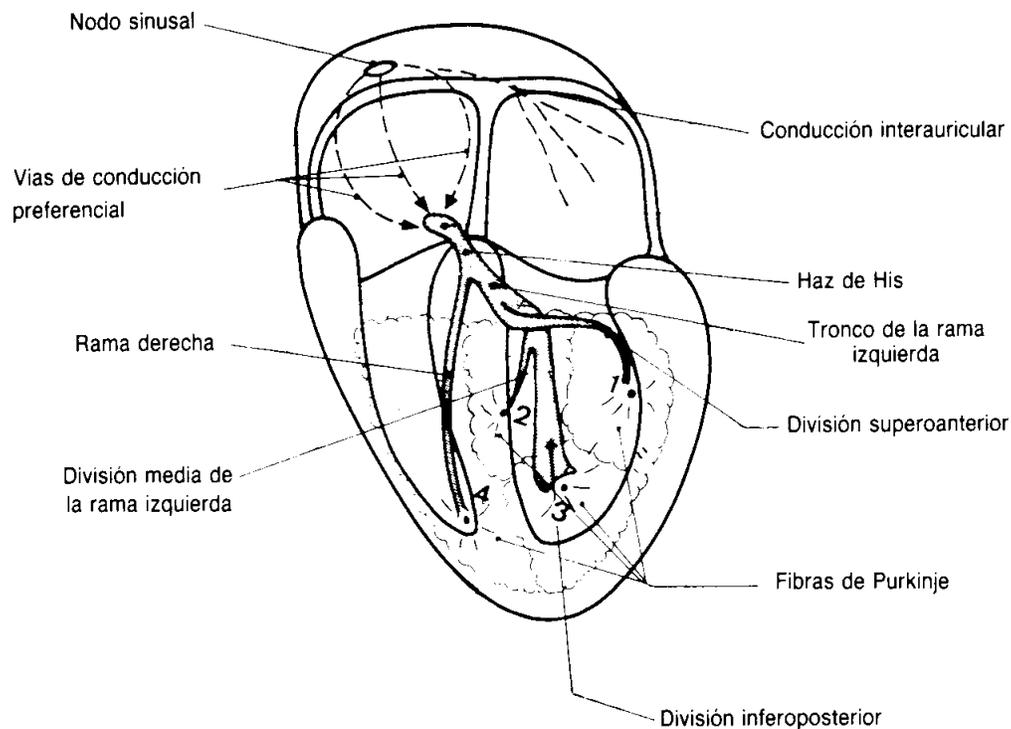


Figura 2.7: Nódulo Sinusal

Fuente: Instrumentación Biomédica (2016)

El ECG es fundamental para el estudio de los pacientes con arritmias cardíacas, trastornos de la Conducción, síndrome de preexcitación y cardiopatía isquémica. También es muy importante para el estudio evolutivo de las demás cardiopatías y para realizar trabajos epidemiológicos. Su valor, además, se ha visto incrementado con la introducción de técnicas que permiten conocer la actividad eléctrica cardíaca de forma dinámica, como son la electrocardiografía de esfuerzo y el registro continuo del ECG en cinta magnética

y con interpretación acelerada (electrocardiografía de Holter), y con la aplicación a la electrocardiografía convencional de los conocimientos adquiridos gracias a las investigaciones y los estudios realizados con las técnicas de electrocardiografía intracavitaria. Sin embargo, a pesar de su inestimable valor diagnóstico, la electrocardiografía puede ser un arma más peligrosa que eficaz si se confía en exceso ante un ECG normal o valoramos en demasía un ECG patológico. La electrocardiografía es una técnica diagnóstica que nos proporciona una ayuda a veces decisiva, pero que el enfoque global del paciente (diagnóstico, pronóstico y terapéutico) sólo debe realizarse mediante la correlación de la clínica con la electrocardiografía y otras técnicas auxiliares.

2.10.1. Activación Cardíaca

Con los aparatos convencionales de electrocardiografía sólo se registra la activación de la masa muscular auricular y ventricular, no siendo posible grabar la activación del nodo sinusal ni la del resto del sistema específico de conducción.

El sitio normal de formación de impulsos marcapasos en el corazón es el nodo sinusal, pequeña estructura en forma de elipse, de unos 15 mm de longitud, situada cerca de la unión de la vena cava superior con la aurícula derecha. A partir del impulso sinusal se produce la despolarización encadenada de todo el corazón, primero de las aurículas, para después propagarse el estímulo a través del sistema específico de conducción y despolarizar los ventrículos.

2.10.2. Transmisión del impulso de las aurículas a los ventrículos

La onda de despolarización auricular se propaga muy lentamente por la parte alta (AN) y media (N) del nodo AV, debido a la presencia de células de respuesta lenta, que determinan una conducción decremental. La conducción no se detiene porque, al llegar a la parte baja del nodo (zona NH), las células son predominantemente de respuesta rápida y la velocidad de conducción es mayor. Al llegar el estímulo al haz de His, la velocidad de conducción aumenta mucho. Las fibras del haz de His ya están predestinadas hacia la rama derecha o izquierda.

2.10.3. Despolarización ventricular (Asa de QRS)

Una vez que el estímulo llega a las dos ramas, se transmite por ellas a una velocidad de conducción parecida a la del haz de His (1,5-2 m/seg). La despolarización ventricular se puede dividir en tres fases cada una de las cuales se puede expresar por un vector. Durante la primera fase (10 mseg) la despolarización inicial de los ventrículos y de la parte media del tabique origina un vector pequeño que va hacia la derecha, hacia delante y hacia arriba o abajo (con mayor frecuencia hacia arriba); durante la segunda fase (30-40 mseg) se despolariza la mayor parte de la pared libre de ambos ventrículos y la parte baja del

tabique, originándose un vector importante dirigido hacia la izquierda, algo hacia atrás y en general hacia abajo, en la tercera fase (15-25 mseg) se origina un vector pequeño dirigido hacia arriba algo hacia atrás y a la derecha.

2.10.4. Repolarización ventricular. (Asa de T)

En el corazón humano el proceso de repolarización de la pared libre del ventrículo izquierdo es prácticamente responsable de toda la repolarización de los ventrículos.

En resumen, el estímulo eléctrico originado en el nodo sinusal se propaga sucesivamente al músculo auricular y, a través del nodo AV y del sistema His-Purkinje a los ventrículos. Cuando el proceso de activación ventricular (despolarización + repolarización) ha terminado, lo que coincide con el final de la sístole, se produce una fase de reposo celular (diástole eléctrica).

Existe, pues, una estructura (el nodo sinusal) con automatismo, que transmite los estímulos que en ella se genera a todo el corazón. El proceso es parecido al que ocurre con una fila de fichas de dominó en la que la primera (equivalente al nodo sinusal) tiene movimiento propio, al tocar a la segunda, le transmite dicho movimiento, que pasa luego a las sucesivas fichas, hasta que todas (todas las estructuras del corazón) se hayan caído (despolarizado). Después, todas las fichas se pondrían de nuevo en pie gracias a un movimiento intrínseco (repolarización), para iniciar, al cabo de cierto tiempo (fase de reposo), un nuevo proceso de caída (despolarización).

2.11. Registro del potencial eléctrico

El electrocardiograma entrega información para el médico especialista del corazón. Esta representación es una línea base y varias deflexiones, ver Figura 2.8.



Figura 2.8: Registro Electrocardiográfico
Fuente: Zuñiga Fonseca, Cortes Olmos (2004)

En el cuerpo humano se generan señales eléctricas, provocadas por la actividad química en los nervios y músculos. En el corazón, se produce un patrón de variaciones de voltaje. El registro y análisis de estos eventos bioeléctricos son importantes en la práctica clínica y la investigación. Los potenciales se generan a nivel celular, cada una de las células es un generador de voltaje. El trazado de un electrocardiograma registra un latido cardiaco, que consiste en una onda P, un complejo de ondas QRS y una onda T. La onda U que es muy pequeña y no es posible percibir. Los vectores de activación o despolarización llevan la carga eléctrica positiva por delante del vector y en su parte final llevarán la carga negativa. En el trazo electrocardiográfico, de acuerdo con la ley del dipolo dice que “todo vector de activación que se acerque al electrodo explorador generará una deflexión positiva, y al alejarse se generaría una deflexión negativa”. Durante la repolarización el vector lleva carga negativa al frente, al ir restableciendo el gradiente transmembrana a su situación original, y llevar cargas positivas al final, de esta forma la ley del dipolo para la repolarización nos dice que “a medida que un vector de repolarización se acerca al electrodo explorador nos generará una deflexión negativa y al alejarse de este una deflexión positiva”.

2.11.1. Eje eléctrico

El eje eléctrico es la dirección general del impulso a través del corazón. Con una lectura desde izquierda a derecha, la magnitud representada en el eje eléctrico esta en mili voltios, el estándar de los electrodiágrafos es de 0.1 milivoltios por cada milímetro cuadrado en el papel o monitor electrocardiográfico, y la velocidad de representación es de 25 milímetros por segundo, como se puede apreciar en la Figura 2.9

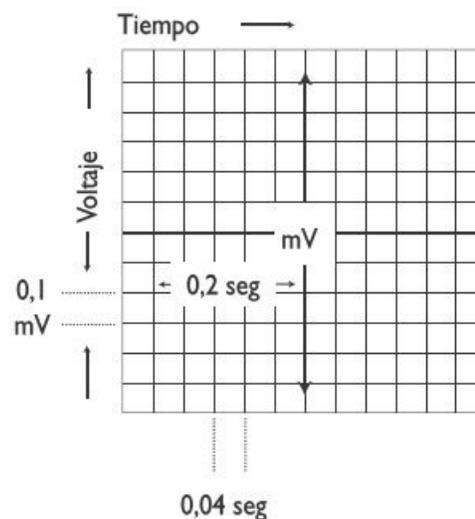


Figura 2.9: Intervalos de un electrocardiograma
Fuente: Zuñiga Fonseca, Cortes Olmos (2004)

2.11.2. Onda P

Es la primera marca identificable en el ECG que corresponde a la llegada de la señal de activación a las aurículas. Su duración es de 0.06 a 0.10 segundos y su voltaje es de 0.15 mV. Cuando las aurículas se llenan de sangre, el nodo sinoauricular se dispara, las señales eléctricas se dispersan por todas las aurículas y causa que se despolaricen. Está representado por la onda P del ECG. La contracción auricular, o sístole auricular ocurre más o menos 100 milisegundos después del inicio de la onda P.

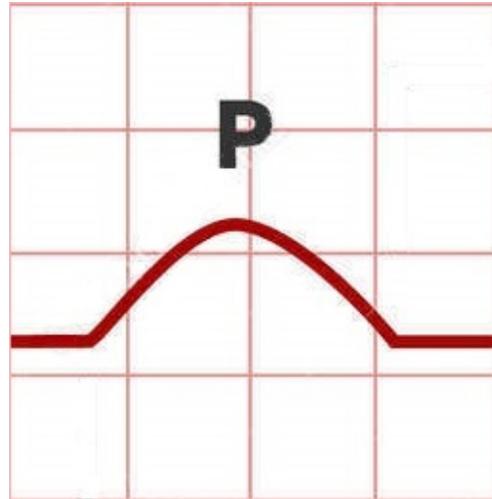


Figura 2.10: Onda P
Fuente: Elaboración propia

El origen de la señal eléctrica se produce cuando el Nódulo Sinoauricular (marcapasos del corazón) proporciona el impulso eléctrico que se distribuye por todas las aurículas, formando la onda P. Finalizado la onda P, el impulso eléctrico es conducido hacia el Nodo atrioventricular, es importante destacar que este impulso tiene un retraso de aproximadamente 100 ms antes de llegar al nodo atrioventricular, debido al proceso de sístole auricular.

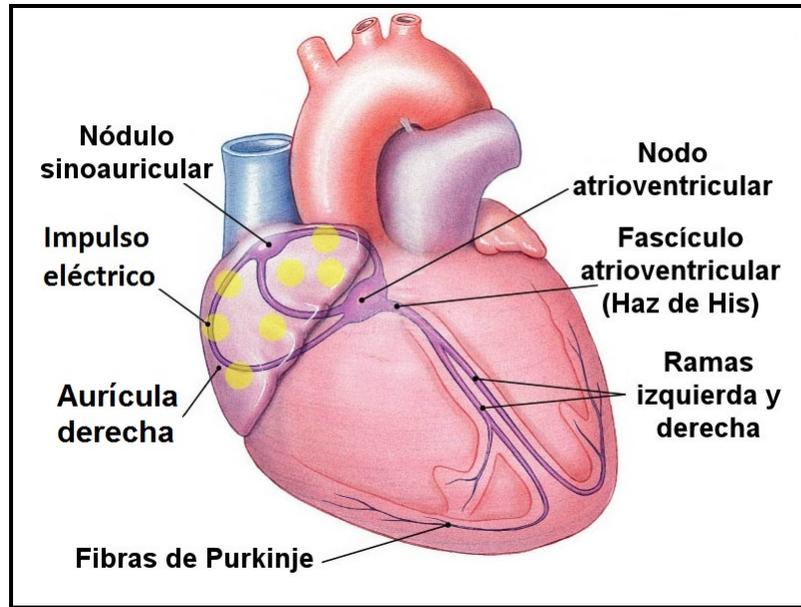


Figura 2.11: Origen onda P
Fuente: Elaboración propia

2.11.3. Complejo QRS

Es la marca más característica de la señal electrocardiográfica que representa la manifestación de la señal de activación a ambos ventrículos. Se define a Q como la primera deflexión negativa del complejo ventricular. A la deflexión positiva del complejo ventricular se denomina R. La onda S es la deflexión negativa que sigue a la onda R. La duración normal del complejo QRS es de 0.06 a 0.10 segundos. La duración del QRS normal se mide desde el inicio de la onda Q o de la R hasta el final de la onda R o S.

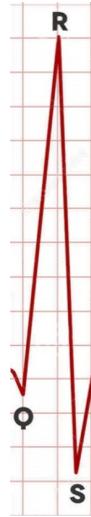


Figura 2.12: Complejo QRS

La onda Q se origina por la despolarización del tabique interventricular específicamente el Haz de His, llegando hasta las fibras de Purkinje.

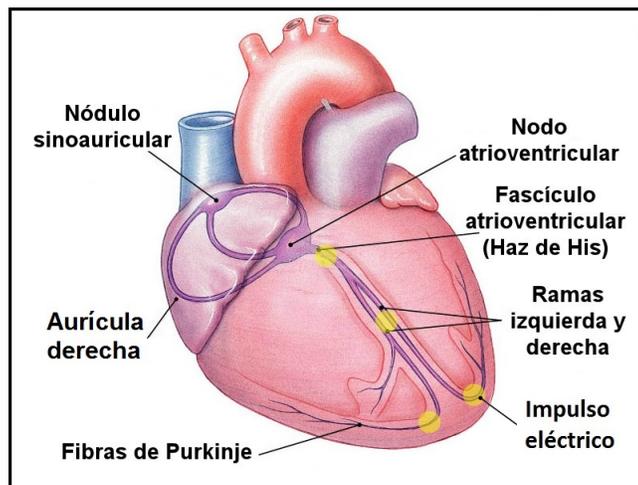


Figura 2.13: Origen onda Q

La onda R, se produce por la despolarización de la masa principal de los ventrículos, el impulso eléctrico llega hasta la unión de las fibras de Purkinje y el músculo, despolarizando gran parte de la masa ventricular.

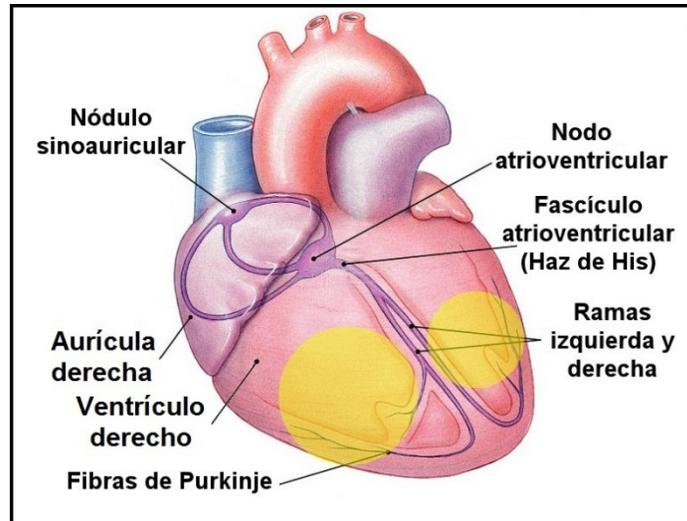


Figura 2.14: Origen onda R

La onda S representa la última etapa de despolarización ventricular, es decir cuando el impulso eléctrico ha alcanzado toda la masa ventricular, dando paso a la contracción ventricular (Fase de inyección).

2.11.4. Onda T

Luego de la fase de inyección, la onda T es generada por la repolarización ventricular, es decir cuando los ventrículos una vez contraídos vuelven a su estado inicial, en el registro electrocardiográfico esta es asimétrica, con la primera rama lenta, la segunda más rápida y un vértice redondeado, su duración aproximadamente es de 0,20 segundos o menos y su amplitud máxima es menor a 0,5 mV.

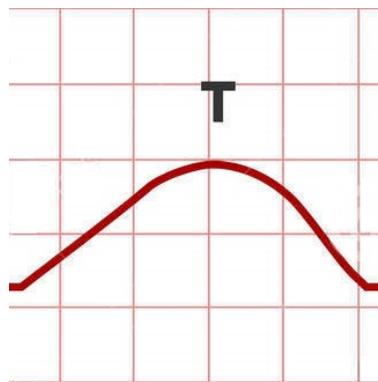


Figura 2.15: Onda T

2.11.5. Intervalo QT

El intervalo QT comprende desde el inicio del complejo QRS hasta el final de la onda T y representa la despolarización y repolarización ventricular. Su duración es de 0.04 segundos.

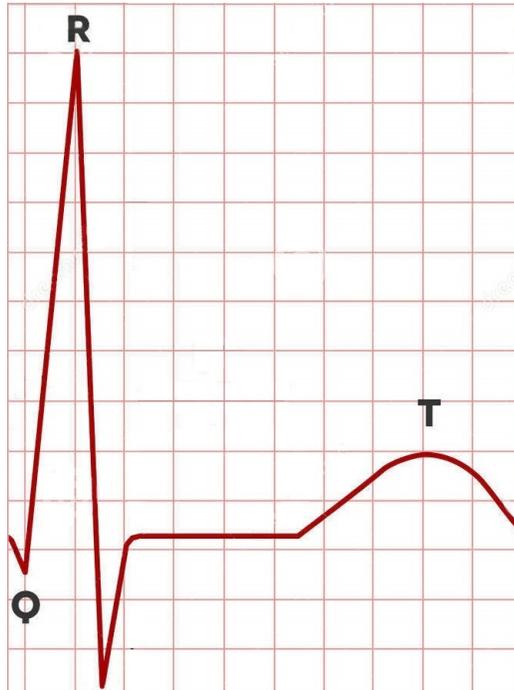


Figura 2.16: Intervalo QT

2.11.6. Segmento PR

El segmento PR muestra el tiempo que mide la primera activación ventricular y se calcula desde el comienzo de la onda P hasta el inicio del complejo QRS. La duración normal es de 0.12 a 0.20 segundos. De manera habitual, el segmento PR debe seguir una línea isoeletrica y se extiende desde el fin de la onda P hasta el comienzo del complejo ventricular (onda Q), la línea oscura representa el segmento PR en la Figura 2.17.

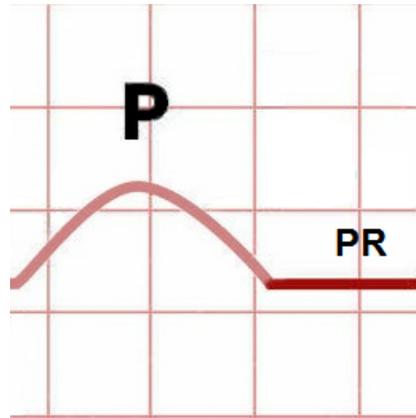


Figura 2.17: Segmento PR

2.11.7. Segmento ST

El segmento ST corresponde a la fase de repolarización ventricular o contracción muscular produciendo el bombeo de la sangre, este segmento comprende desde el final de la onda S hasta el inicio de la onda T, este segmento se registra como una línea isoelectrónica representada por la línea oscura en la Figura 2.18.

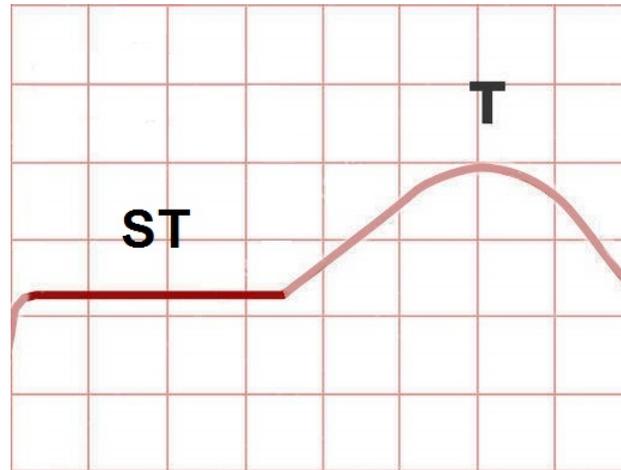


Figura 2.18: Segmento ST

Un registro general del complejo PQRST corresponde a la unión de todas las ondas antes expuestas. El intervalo RR, es la distancia entre dos ondas R sucesivas, en el ritmo sinusal, este intervalo debe ser constante. El intervalo RR se mide desde el inicio de una onda R hasta el inicio de la siguiente y su longitud depende de la frecuencia cardíaca, en estado de reposo la frecuencia cardíaca oscila entre 50 y 100 latidos por minuto en adultos, cuando se realiza algún ejercicio físico este puede pasar de 100 latidos por minuto.

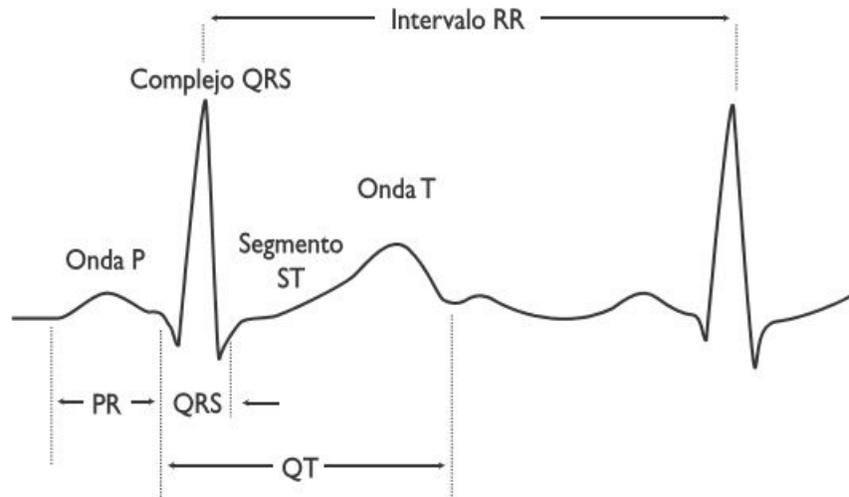


Figura 2.19: Complejo PQRST e intervalo RR
Fuente: Quora.com , Ray Schilling (Abril 2017)

En los capítulos sucesivos se plantea el diseño del software y hardware para el dispositivo ECG, teniendo como base los puntos ya tocados, a fin de cumplir con los objetivos ya propuestos.

Capítulo 3

Implementación de la plataforma

En este capítulo se describen las características de los elementos a nivel tanto de software como de hardware utilizados para la implementación de la plataforma, así mismo se describen los procesos llevados a cabo para el diseño e implementación.

3.1. Tecnologías utilizadas

3.1.1. Arduino IDE

El Arduino Integrated Development Environment (IDE), es un software que contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comunes y una serie de menús. Se conecta al hardware Arduino y comunicarse con ellos.

3.1.2. Writing Sketches

Los programas escritos con Arduino Software (IDE) se denominan sketches. Estos bocetos se escriben en el editor de texto y se guardan con la extensión de archivo .ino. El editor tiene funciones para cortar / pegar y para buscar / reemplazar texto. El área de mensajes da un “feedback” mensajes o errores. La consola muestra el texto generado por el software Arduino (IDE), incluyendo mensajes de error completos y advertencias o recomendaciones. La esquina inferior derecha de la ventana muestra la tarjeta configurada y el puerto serie. Los botones de la barra de herramientas le permiten verificar y cargar programas, crear, abrir y guardar bocetos y abrir el monitor en serie.

- **Verify:** Verifica si el código es capaz de compilar.
 - **Upload:** Compila el código y lo carga en la tarjeta configurada.
 - **New:** Crea un nuevo sketch.
-

- **Open:** Presenta un menú de todos los sketch en su repertorio de sketches. Al hacer clic en uno, se abrirá dentro de la ventana actual sobrescribiendo su contenido.
- **Save:** Guarda los cambios realizados sobre el sketch.
- **Serial Monitor:** Abre un terminal serie, para la comunicación entre el dispositivo y el PC.

3.1.3. Tarjeta de acondicionamiento

Basado en la placa amplificadora desarrollado por la Ing. Aida Perez. incluye al amplificador, filtros, amplificación posterior, voltaje de offset y obtención de voltaje negativo para los amplificadores. Se construyeron tres tarjetas iguales, una para cada canal. La figura 5.11 presenta la tarjeta de acondicionamiento final. En la tarjeta de cada canal se requieren tres entradas, pero si se usa el cable de 5 terminales, existen terminales que son comunes a más de una tarjeta. Para ello se dispone de tres jumpers, cada uno de los cuales debe cortocircuitarse en la tarjeta de un canal cuando un mismo electrodo con su terminal se utilice en dos canales. Se cortocircuitará con un pequeño conector sólo en el canal adicional donde se requiera el mismo terminal del cable. En el caso del Jumper J-IN3 el mismo puede dejarse libre ya que la entrada IN3 es la referencia del paciente, la cual es común y ya está conectada en las tres tarjetas de acondicionamiento.

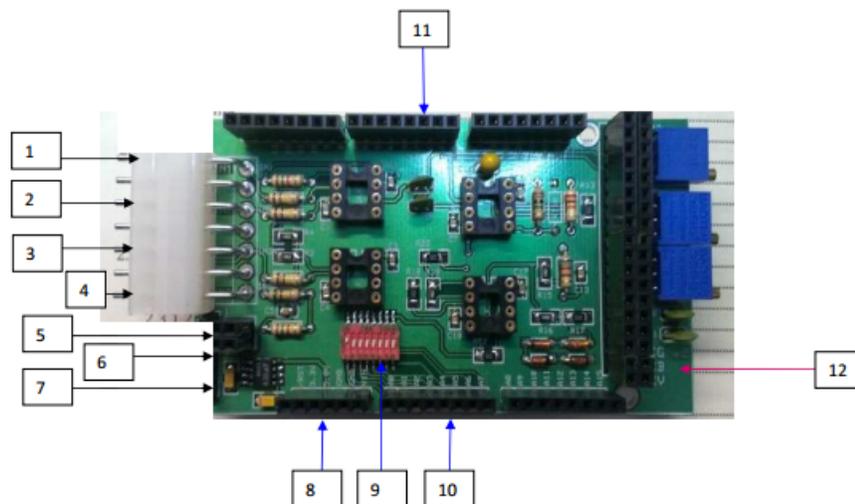


Figura 3.1: Tarjeta Amplificadora

Fuente: Aida Perez, 2017

- **1:** Entrada IN1 de la señal electrocardiográfica (entrada para el electrodo negativo).
- **2:** Entrada IN2 de la señal electrocardiográfica (entrada para el electrodo positivo).

- **3:** Entrada de referencia IN3 (electrodo de referencia).
- **4:** Referencia (0 V).
- **5:** Jumper de entrada IN1 (J-IN1) para canales que compartan entrada con IN1.
- **6:** Jumper de entrada IN2 (J-IN1) para canales que compartan entrada con IN2.
- **7:** Jumper de entrada IN3 (J-IN1) para canales que compartan entrada con IN3. Debeconectarse el jumper JIN3 en las tres tarjetas para que compartan el electrodo de referencia.
- **8:** Pines de alimentación (5 V, 3.3 V y 0 V).
- **9:** Dip-switch (interruptores) para seleccionar la entrada analógica a la cual se conecta el canal acondicionado. Tiene disponibilidad para seleccionar entre las primeras 8 entradas analógicas.

3.1.4. Tarjeta de registro, interfaz SD, pulsadores y LEDs

Se construyó la tarjeta diseñada para la etapa de registro, la cual está conformada por la interfaz para la memoria SD o micro SD y los pulsadores y leds para interacción con el programa.

En la figura3.2 se muestra dicha tarjeta. La tarjeta tiene los pines para alimentación, pines de entrada-salida digitales y pines para la pantalla (LCD) a fin de hacerla compatible con la tarjeta Arduino ATmega 2560. También cuenta con pines TX y RX en un conector para uso posterior de Bluetooth (no implementado su uso en este prototipo, se deja abierto para su inclusión).

- **1:** Conectores de las entradas – salidas digitales (pulsadores y leds).
 - **2:** Conector para la memoria SD o micro SD con su adaptador.
 - **3:** Conectores para la alimentación (5 V y 0 V).
 - **4:** Conectores para la pantalla LCD y para la interfaz SD (pines 50, 51, 52 y 53 para la SD).
 - **5:** Conector disponible para módulo Bluetooth.
 - **6:** Led 1 (azul).
 - **7:** Led 2 (verde).
 - **8:** Led 3 (Rojo).
 - **9:** Botón 1.
-

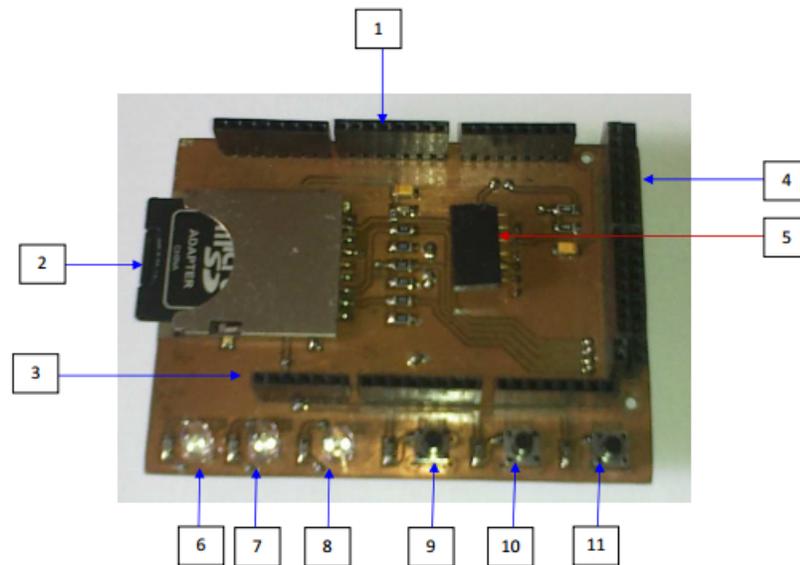


Figura 3.2: Tarjeta Amplificadora
Fuente: Aida Perez, 2017

- **10:** Botón 2.
- **11:** Botón 3.

Los leds permiten indicar el estado de ciertas operaciones del microcontrolador: el led 1 (azul) señala cuando se inicia el modo registro de datos, en ese punto el programa ha creado un archivo nuevo en memoria SD y se encuentra escribiendo las muestras digitalizadas en el. El led 3 se enciende cuando el dispositivo se encuentra en modo transmisión, en este el Holter no está obteniendo ningún tipo de registro únicamente verificando si se ha recibido algún dato mediante bluetooth. Los pulsadores o botones permiten la interacción con el programa. Botón 1: En el modo graficación, permite al operador cambiar de canal luego del canal 3, el holter cambia a modo registro de datos. Botón 2: Registrar un evento, únicamente este se efectúa cuando el dispositivo se encuentra en modo registro de datos. Botón 3: inicia el modo transmisión, si se desea detener el modo transmisión, se debe presionar el botón 1.

Conexión de tarjeta SD mediante interfaz SPI (Serial Peripheral Interface)

La interfaz periférica (SPI) es un protocolo de datos en serie síncrono utilizado por microcontroladores para comunicarse con uno o más dispositivos periféricos rápidamente en distancias cortas. También se puede utilizar para la comunicación entre dos microcontroladores.

Con una conexión SPI siempre hay un dispositivo maestro (generalmente un microcontrolador) que controla los dispositivos periféricos. Normalmente hay tres líneas comunes

a todos los dispositivos:

- **MISO (Master In Slave Out):** La línea Slave para enviar datos al maestro.
- **MOSI (Master Out Slave In):** La línea maestra para enviar datos a los periféricos.
- **SCK:** Los pulsos de reloj que sincronizan la transmisión de datos generada por el maestro.
- **SS:** El pin de cada dispositivo que el maestro puede usar para activar y desactivar dispositivos específicos.

Cuando el pin de Selección de Esclavo de un dispositivo está bajo, se comunica con el maestro. Cuando es alto, ignora al maestro. Esto le permite tener varios dispositivos SPI compartiendo las mismas líneas MISO, MOSI y CLK.

En general, hay cuatro modos de transmisión. Estos modos controlan si los datos se desplazan hacia adentro y hacia afuera en el flanco ascendente o descendente de la señal de reloj de datos (llamada la fase de reloj) y si el reloj está inactivo cuando es alto o bajo (llamada polaridad del reloj). En la definición de los pines de conexión de conexión del estándar SD es la siguiente:

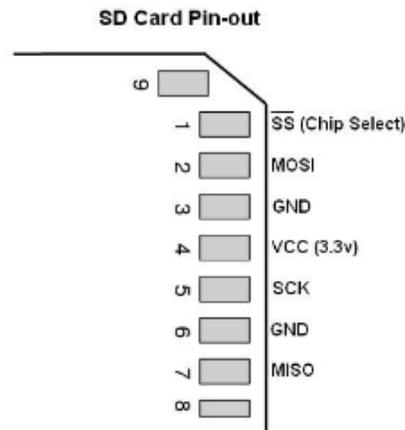


Figura 3.3: Pines de conexión memoria SD
Fuente: dharmanitech.com, Dharmani (2016)

Esta placa se encuentra conectada al arduino de tal manera que corresponde a la tabla 3.1:

Cuadro 3.1: Conexiones de pines SD Arduino

SPI	Pines Digitales Arduino MEGA 2560
SS	53
MOSI	51
MISO	50
SCK	52

3.1.5. Pantalla de visualización

Para la visualización se utilizó una pantalla de cristal líquido (LCD) gráfica de 3.2" modelo TFT 320QVT del fabricante LC Studio. La figura 3.4 muestra la pantalla utilizada.



Figura 3.4: Pantalla TFT de 3.2" utilizada en el prototipo

Fuente: Aida Perez, 2017

3.1.6. Placa adaptadora pantalla LCD

Para la utilización de la pantalla LCD 3.2", es necesario la utilización de una placa adaptadora, además de proveer una interfaz de conexión para memoria SD, en este modelo de placa adaptadora existe una limitante ya que como es común el bus de comunicación entre la pantalla LCD y la interfaz SD, no es posible realizar ambas tareas a la vez (escritura en memoria y graficación).

Una característica de la pantalla LCD, es que no es posible a nivel de software apagar la pantalla LCD, ya que para la implementación del prototipo este debe ser portable, es decir funcionar bajo batería, para el ahorro de energía es necesario la inactivación o apagado de la pantalla.

Para el control de backlight, fue necesario una modificación a nivel de hardware a

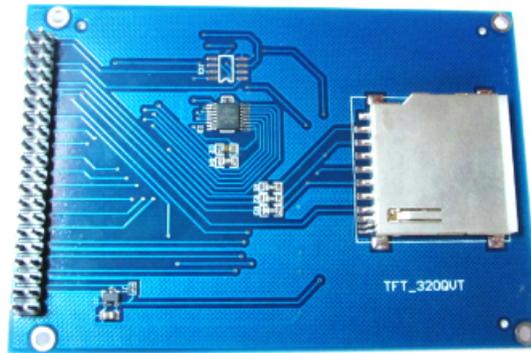


Figura 3.5: Adaptador Pantalla TFT 3.2”

Fuente: elecfreaks.com , TFT LCD Module (2016)

la placa adaptadora. La placa inicialmente para controlar el brillo de fondo de la pantalla comúnmente llamado “backlight” implementa un potenciómetro, el cual aumenta o disminuye el voltaje aplicado al backlight, regulando así el brillo, la limitante en este sentido es que el potenciómetro solo regulable de manera mecánica, a través de un pequeño destornillador y haciendo girar la perilla.

Para poder lograr la misma función pero controlado totalmente por el arduino, era necesario el reemplazo del potenciómetro por un transistor NPN y la base del mismo conectado a un pin del arduino.

Para la regulación del brillo de la pantalla, es por medio de la conversión de digital a analógico llamado Modulación por ancho de pulso (PWM).



Figura 3.6: Conexión de pantalla LCD mediante placa adaptadora

Fuente: sainsmart.com (2016)

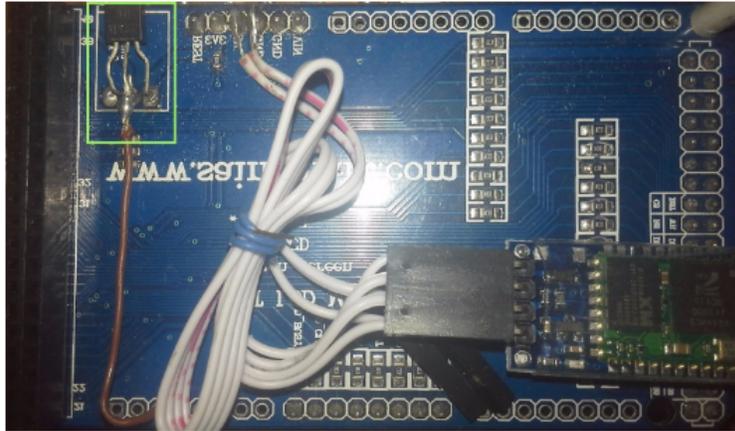


Figura 3.7: Modificación por hardware en placa adaptadora (recuadro verde)

3.1.7. Módulo para conexión bluetooth

En cuanto al establecimiento de conexión bluetooth se utilizó el módulo HC-06, el cual es relativamente popular en el mercado, económico, y cuenta amplia documentación por parte de la comunidad arduino, el módulo HC-06 funciona como Maestro-esclavo, es decir, es capaz de generar y recibir conexiones.

El HC-05 tiene un modo de comandos AT que debe activarse mediante un estado alto en el PIN34 mientras se enciende (o se resetea) el módulo. En el modo de comandos AT, es posible configurar el módulo bluetooth y cambiar parámetros como el nombre del dispositivo, password, modo maestro/esclavo, etc.

Para la comunicación con el módulo bluetooth y configurarlo, es necesario tener acceso al módulo mediante una interfaz serial. Es posible usar un arduino con un par de cables (aprovechando el puente USB-Serial del Arduino), un kit para XBee o un simple MAX3232 en el puerto serie de la PC.



Figura 3.8: Módulo bluetooth (HC-06)

Fuente: Amazon.com (2016)

3.2. Diseño

Basado en los problemas planteados en el capítulo 1, se propone una solución basada en software y hardware libre que permita la adquisición, procesamiento, graficación, almacenamiento y transmisión de los registros cardiacos, a fin de proporcionar una manera accesible a los datos, además de otorgar una plataforma abierta ayudando a continuar y promover la investigación en el área.

Para la implementación de la solución se plantea una arquitectura modular, en la que básicamente se contará con cuatro módulos, Adquisición, procesamiento, almacenamiento y transmisión. Para la implementación de cada uno de los módulos son necesarios el desarrollo de procesos internos, como se plantea en la siguiente Figura.

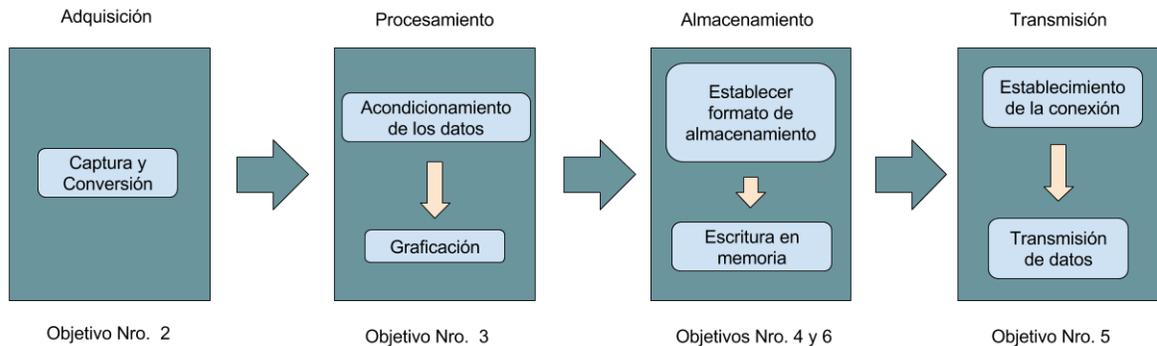


Figura 3.9: Arquitectura Holter

3.2.1. Etapa de adquisición

Captura y conversión: Una vez obtenida la señal amplificada (partiendo de la tarjeta de acondicionamiento de señales para Holter desarrollada por la Ing. Aida Pérez, Universidad de Carabobo), el microcontrolador debe capturar la señal y cuantificar, para esto el microcontrolador Atmega 2560 consta de un módulo CAD (Convertidor Analógico Digital), con un máximo de resolución de 10 bits (deseable), esto implica que los valores numéricos retornados por este módulo serán comprendidos entre 0 y 1023, donde 0 corresponde a 0V, y 5V a 1023.

Previo a la conversión, es necesario configurar el módulo, entre las más importantes, puertos de E/S a utilizar, puertos de referencia y resolución de conversión, una vez realizadas todas las configuraciones, ya está preparado para iniciar la conversión cada vez que se invoque el módulo CAD.

3.2.2. Etapa de procesamiento

- **Acondicionamiento de los datos:** A medida que los registros del módulo CAD se obtienen estos deben almacenarse temporalmente, para esto es necesario la implementación al menos un buffer en el dispositivo, este buffer será el soporte para el procesamiento, por ejemplo; para graficar los datos obtenidos es necesario convertir un bloque de datos en un arreglo de bytes, o para escribir en la memoria es necesario convertirlos en un bloque más grande de datos, sea para graficar o guardar los registros ambas funciones deben ser excluyentes, es decir el microcontrolador puede en un momento dado dedicarse al guardando de los datos a la memoria, o a la graficación, de esa manera no se disminuye la tasa de capturas del módulo CAD.
- **Graficación:** En esta etapa se toman solo los datos de la derivación por un solo canal, o en tal caso es posible elegir qué canal utilizar, la biblioteca de graficación requiere que los datos sean dispuestos en un arreglo de bytes, para luego ser graficados.

3.2.3. Etapa de almacenamiento

- **Establecer formato de almacenamiento:** Esta corresponde una de las funciones principales para un Holter, la capacidad de almacenar los datos en el tiempo determinado, sea este 24 horas 48 horas o más, sin embargo en vista que los datos provenientes de las 3 canales (derivaciones) se encuentran todos juntos almacenados en buffer es necesario implementar un formato de almacenamiento de tal manera que el software pueda interpretar los archivos almacenados en memoria y estos se encuentren bajo un formato, así mismo es importante resaltar que es necesario el manejo de fecha y hora, de esa manera se podrá saber con exactitud, a qué hora y fecha fue realizado un registro determinado. Una vez establecido el formato de almacenamiento, se procede a la escritura en memoria.
- **Escritura en memoria:** Antes de proceder a escribir es necesario establecer la comunicación serial con la memoria SD, para ello es utilizada una biblioteca que es proporcionada por la comunidad de Arduino, luego se crea el archivo donde se escribirán los datos, y se procede a escribir, esto debido a que es una operación de E/S es recomendable realizarlo por bloques de bytes, y así aprovechar al máximo la operación de E/S.

3.2.4. Etapa de transmisión

- **Establecimiento de la conexión:** Debido al protocolo Bluetooth, es necesario “emparejar” los dispositivos que actuarán en la comunicación, de esa manera cada vez que se desee transmitir datos, estos establecen la comunicación automáticamente, es deseable que esta comunicación sea únicamente punto a punto a fin de que
-

los datos no sean recibidos por terceros, siempre que exista cobertura de señal en ambos puntos de la comunicación, se transmiten los datos.

- **Transmisión de datos:** La transmisión de los datos viene influenciada por dos factores, la comunicación entre el microcontrolador y el módulo Bluetooth, y de este hasta el destino final, por defecto en la mayoría de los módulos Bluetooth comerciales poseen un estándar de velocidad de conexión de 9600 bps, y de transmisión inalámbrica de 2,1 Mbit/s, siendo la comunicación como lo muestra la Figura 3.15.

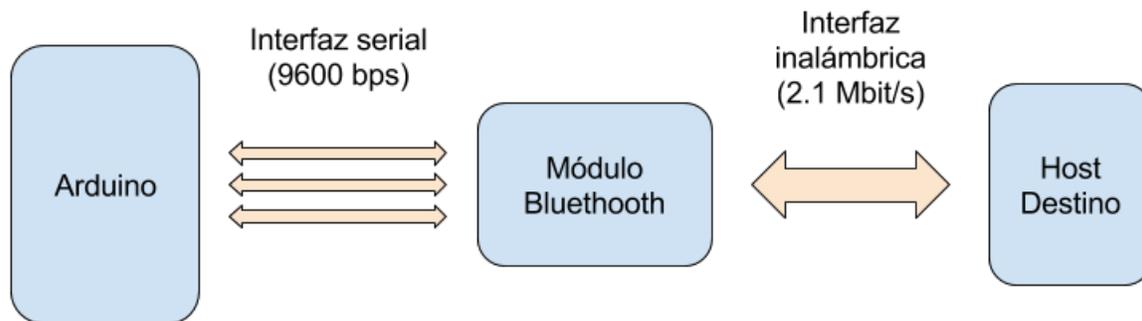


Figura 3.10: Interfaz de comunicación serial e inalámbrica

El diseño de los módulos involucrados en el desarrollo del dispositivo no solo implica un diseño lógico de software, también es necesario ensamblar elementos de hardware tales como las placas de acondicionamiento, que interconectadas pueden manejar hasta 3 canales de adquisición, visto de manera lógica todos estos componentes del Holter estarían conectados como en la figura 3.11.

La señal que es directamente recibida por los elementos enteramente de hardware tales como las placas amplificadoras, se encargan de acondicionar la señal para el arduino, este último a través de sus pines de entrada y salida recibe la señal acondicionada y la digitaliza, una vez realizado esto, el software se encargaría de procesar los datos, para luego entregar el resultado por medio del hardware correspondiente, puede ser pantalla LCD, escritura en memoria SD o comunicación Bluetooth.

3.2.5. Esquema de pruebas

El esquema de pruebas consta de 2 etapas. En la primera etapa de entonación se utilizará un generador de señales, esta señal puede tener forma sinusoidal o forma de sierra con amplitudes que van en el rango de (0.1mv – 5v). Inicialmente la frecuencia de la señal generada será de 1 Hz, gradualmente durante la fase de pruebas se irá aumentando la frecuencia de la señal generada hasta tener una frecuencia similar al de la actividad

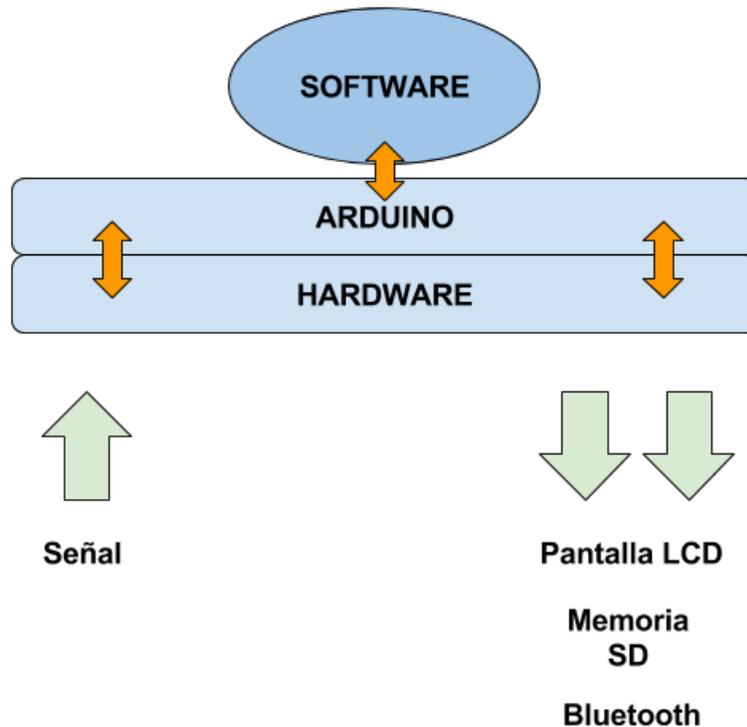


Figura 3.11: Diseño lógico Holter

cardíaca (Ver Tabla 2.1), por último la frecuencia se aumentara hasta un máximo de 250 Hz, a fin de determinar si el dispositivo cumple con el estándar de frecuencia de muestreo para ECG, es decir 500 muestras por segundo (Teorema de Nyquist) y así evaluar que los registros que se grafican o se almacenan correspondan en frecuencia y amplitud con el de las señales generadas.

Una vez superada la primera etapa, en esta segunda etapa el dispositivo será evaluado con registros de diferentes pacientes adultos de contextura variada, dichos registros serán evaluados por un médico especialista que determine si el Holter cumple con todos los estándares electrocardiográficos.

3.2.6. Indicadores esperados

- El mallado generado por el monitor (LCD) del dispositivo debe corresponder en longitud y altura con el de un registro electrocardiográfico.
- Las señales graficadas por el monitor (LCD) del dispositivo correspondan en frecuencia y amplitud al de las señales generadas.
- El grabado de los registros en la memoria correspondan al de las señales generadas.

- Durante el registro de actividad cardiaca, las señales graficadas por el monitor (LCD) del dispositivo sean las esperadas en el paciente, en este caso queda a criterio del médico especialista.
- Transmitir la información a un dispositivo externo (PC).

3.3. Implementación de la plataforma

Basado en los objetivos propuestos y en la metodología elegida, la plataforma se dividió en módulos, donde cada uno de ellos cumplen con uno o más objetivos, como se muestra en la figura 3.12



Figura 3.12: Módulos Holter ECG y objetivos

3.3.1. Etapa de adquisición

En esta primera etapa luego de pasar por la etapa de acondicionamiento de la señal, por parte de las placas amplificadoras, el arduino por medio de las patillas de entrada analógica (A0,A1,A2) obtiene la señal acondicionada, siendo este un valor apropiado en amplitud (Volts) manejable por el microcontrolador.

El microcontrolador por medio de modulo convertidor Analógico digital o (A/D) obtiene el potencial eléctrico de las patillas analógicas (A0,A1,A2), y digitaliza la magnitud de esta mediante modulación por ancho de pulso (PCM), la resolución del módulo CAD es de 10 bits (0-1023).

Cuando se activa el módulo CAD del microcontrolador es posible activar las interrupciones. Las interrupciones permiten al microcontrolador, ejecutar subrutinas en el programa inmediatamente y una vez termine esta, continuar ejecutando el programa desde el punto antes de la interrupción.

En el caso de la conversión analógico/digital el modulo realiza la conversión totalmente independiente de la CPU, una vez esta finalice, la interrupción es invocada y comienza

la ejecución de la subrutina, para ello los valores obtenidos de cada conversión son almacenados temporalmente en una variable, para luego ser procesados, almacenados en la memoria SD o graficados en pantalla.

Como se hizo referencia en la tabla 1, las ondas ECG puede variar entre 0.05 - 100 Hz, teniendo en cuenta ello, y basados en la ley de Nyquist el cual establece que la tasa de muestreo de una señal, debe ser como mínimo 2 veces la mayor frecuencia de la señal obtenida, en este caso la frecuencia cardiaca podría llegar a 100 Hz, es este sentido la frecuencia de muestreo debe ser como minimo 200 m./s (muestras por segundo), basándose en algunas referencias de Holter comerciales, la frecuencia de muestreo establecida fue de 500 m./s.

La tasa de muestreo debe ser igual por cada uno de los canales, el microcontrolador estará haciendo en total 1500 m./s. Para el cumplimiento de la tasa de muestreo, cada interrupción debe efectuarse 500 veces por segundo, es decir cada interrupción debe efectuarse cada 0.002 segundos o 2 milisegundos.

En la configuración de las interrupciones del microcontrolador, para que la interrupción pueda efectuarse cada 2 ms, debe utilizarse un timer o temporizador. Un timer es un módulo independiente del microcontrolador que se encarga de temporizar o llevar un conteo de ciclos del reloj, este es totalmente configurable, el tamaño de los registros de conteo es de 16 bits, es decir puede contar desde 0 hasta 65535, cuando el contador es desbordado (llega a su valor máximo) es cuando se produce una interrupción.

Sin embargo el desbordamiento de timer suele solo tomar unos pocos milisegundos, en el caso de que se desee que el desbordamiento tome más tiempo se utiliza un elemento adicional llamado “prescaler”, el prescaler se encarga de dividir la frecuencia de reloj de la CPU y en ese intervalo incrementar el contador del timer, el prescaler puede ser configurado con los siguientes valores: 1,8,64,256,1024.

La frecuencia del microcontrolador ATmega 2560, por defecto es de 16 Mhz, cada valor del prescaler determina por cuanto será dividido esta frecuencia, en la tabla 3.2 se muestra los valores que puede tomar el prescaler.

Donde C12, CS11, CS10 corresponde a los bits de configuración del prescaler, en el registro TCCR0B de los registros de funcion especial del arduino.

En la configuración del timer se determina como valor de conteo 3999 pudiendo tomar un valor entero de 16 bits. es decir, el timer contará desde 0 hasta 3999 produciendo la interrupción por timer, el cálculo para que la interrupción se produzca cada 2 ms o 500 interrupciones por segundo, se realizó basado en la fórmula de la documentación del fabricante, el cual plantea que la cantidad de interrupciones generadas será igual a la figura 3.13

Los valores de configuración serian los siguientes:

- **microcontroller clock frequency** = 16000000 Hz
 - **prescaler** = 8
-

Cuadro 3.2: Configuración de preescaler

CS12	CS11	CS10	Descripción
0	0	0	Sin fuente de reloj (Timer detenido)
0	0	1	Clk/1 (Sin preescaler)
0	1	0	Clk/8 (desde prescaler)
0	1	1	Clk/64 (desde prescaler)
1	0	0	Clk/256 (desde prescaler)
1	0	1	Clk/1024 (desde prescaler)
1	1	0	Conteo por reloj externo (flanco de bajada)
1	1	1	Conteo por reloj externo (flanco de subida)

$$\frac{\text{microcontroller clock frequency}}{(\text{prescaler}) \times (\text{compare match register value} + 1)}$$

Figura 3.13: Atmega 2560 Prescaler formula

- **compare match register value = 3999**

Teniendo como resultado: $16000000 / (8 * (3999 + 1)) = 500$

Es decir con esta configuración del preescaler obtenemos 500 interrupciones por parte del timer por segundo, siendo así el valor buscado para la obtención de las capturas logrando las 500 m/s (muestras por segundo).

3.3.2. Implementación de la etapa de adquisición

Es necesario en primer lugar definir las configuraciones del timer y asociar la función encargada de capturar los registros en este caso es `ISR_Samples`, se utilizara como timer para la temporización `timer1` ya por ser un timer de 16 bits (0 - 65535).

```
Timer1.attachInterrupt (ISR_Samples);
Timer1.initialize (3999);
interrupts ();
```

Se configura la temporización del timer donde en primer lugar se especifica qué función será ejecutada cuando se produzca la interrupción, se asigna un valor de conteo al registro, y por último se activan las interrupciones con la función “`interrupts()`”. Por último, se define la función que es ejecutada durante la interrupción, en este caso es importante notar que antes de salir de la rutina de interrupción, el contador del `timer1` es necesario asignarle el valor de conteo.

3.3.3. Etapa de procesamiento

Luego de obtenida la señal por medio del módulo CAD, se almacenan los registros en variables temporales para el acondicionamiento. El hardware utilizado para la graficación, es una pantalla TFT 3.2" y cuenta con una resolución de 320 x 240 píxeles, el primer elemento en consideración fue la correspondencia entre los valores obtenidos y la resolución de la pantalla, los valores obtenidos por parte del módulo CAD deben poder ser representados en la pantalla LCD, para ello, los valores obtenidos son divididos entre 4, teniendo como resultado $1024/4 = 256$, es decir 256 posibles valores para ser representados, en este sentido se tomó la relación entre estos registros como un valor entre 0 - 255 es correspondiente a un píxel en el eje "Y" de la gráfica.

Es importante denotar que la escala en la representación gráfica debe ser lo más fiel a los estándares electrocardiográficos, para esta implementación, se denoto un mallado de representación de 16x16 píxeles, el eje vertical corresponde a la amplitud en milivoltios y el eje horizontal el vector del tiempo, cada cuadrado del mallado corresponde a 3.2 milivoltios en el eje Y, y 40 ms en el eje X.

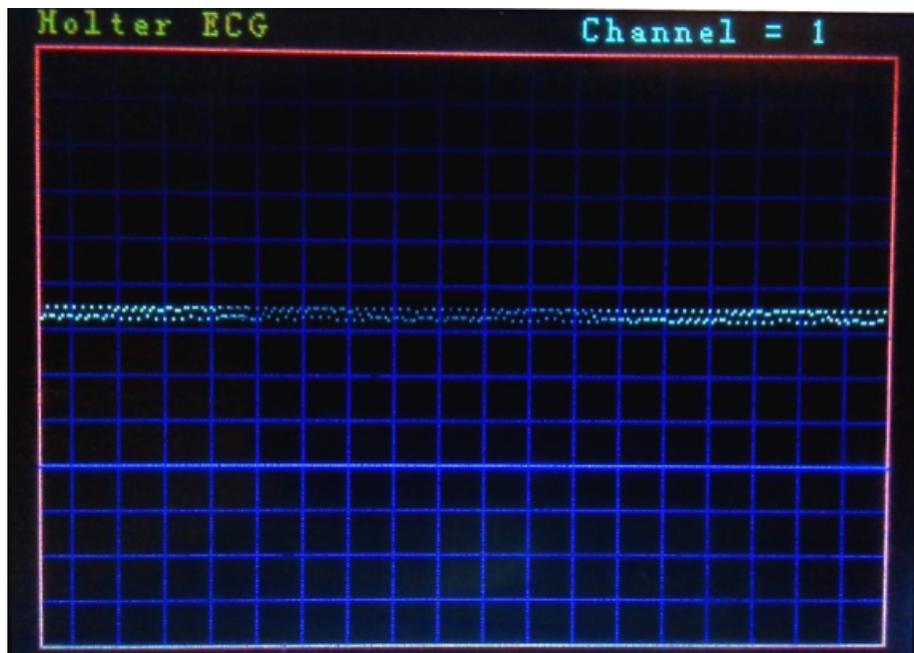


Figura 3.14: Representación del mallado ECG

Continuamente se realiza un barrido por toda la pantalla de izquierda a derecha dibujando los nuevos píxeles, los píxeles próximos a ser sustituidos se dibujan en un color distinto para dar mayor claridad visual, a la vez, constantemente se verifica el estado de los pulsadores por si no existe algún cambio en estos. Este proceso corresponde al código siguiente:

```
while(1){
  //Graficar
  //Obtener puntos a graficar
  for (i=297;i>0;i--){
    NXTFR[i]=analogRead(channel)/4;
    delayMicroseconds(12000);
  }
  //Calculo el promedio de los puntos
  max = 0;
  min = 224;
  for (i=297;i>0;i--){
    prom += NXTFR[i];
  }
  prom = prom/298;
  // Correccion de desplazamiento
  x = 120 - prom;
  for (i=297;i>0;i--){
    if((NXTFR[i]+x) > 225){ // Correccion de limites inicial
      NXTFR[i] = 120;
    }else if((NXTFR[i]+x) < 15){
      NXTFR[i] = 120;
    }
    if((NXTFR[i-1]+x) > 225){ // Correccion de limites final
      NXTFR[i-1] = 120;
    }else if((NXTFR[i-1]+x) < 15){
      NXTFR[i-1] = 120;
    }
  }

  if(NXTFR[i]>max){ // Minimos y maximos
    max = NXTFR[i];
  }else if(NXTFR[i] < min){
    min = NXTFR[i];
  }
}
//Amplificacion
int d = 224 - max;
int d2 = min - 16;
if(d2 > d){
  famp = (float) 224/max;
}else{
  famp = (float) 16/min;
}
```

```

}

// Amplificacion de la senial

for (i=297;i>0;i--){
  if (d2 > d){
    if (NXT_FR[i] > 120){
      NXT_FR[i] = (NXT_FR[i]*famp);
    }else{
      NXT_FR[i] -= (NXT_FR[i]*famp);
    }
  }else{
    if (NXT_FR[i] > 120){
      NXT_FR[i] += (NXT_FR[i]*famp);
    }else{
      NXT_FR[i] = (NXT_FR[i]*famp*2);
    }
  }
}
if (NXT_FR[i]+x >= 224){
  NXT_FR[i] = 224 - x;
} else if (NXT_FR[i]-x <= 15){
  NXT_FR[i] = 16;
}
if (NXT_FR[i] < 30 && i < 295 && i > 0){ //Suavizado
  NXT_FR[i] = (NXT_FR[i-1]+NXT_FR[i+1]+NXT_FR[i+2])/3;
}
}
//Graficacion de las lineas
for (i=297;i>0;i--){
  if (CUR_FR[i]!=0){ // Borrado de los valores
    myGLCD.setColor(0,0,0);
    myGLCD.drawLine(i+shift_x, CUR_FR[i]+x2, \
      (i-1)+shift_x, CUR_FR[i-1]+x2);
  }
  myGLCD.setColor(200,255,255);
  myGLCD.drawLine(i+shift_x, NXT_FR[i]+x, \
    (i-1)+shift_x, NXT_FR[i-1]+x);
  CUR_FR[i] = NXT_FR[i];
}
CUR_FR[0] = NXT_FR[0];
x2 = x;

```

```

// Verificacion del Boton de cambio de canal
buttonState = digitalRead(buttonPin);
if(buttonState ==LOW){
  if(channel==A0){
    channel = A1;
    myGLCD. print (" Channel =_2" ,120 ,239 ,179);
  }else if(channel==A1){
    channel = A2;
    myGLCD. print (" Channel =_3" ,120 ,239 ,179);
  }else if(channel==A2){
    channel=A0;
    myGLCD. clrScr ();
    //Turn OFF LCD Screen
    digitalWrite (13,LOW);
    break;
  }
}else if((digitalRead(buttonBluetooth) == LOW)){
  myGLCD. clrScr ();
  analogWrite(5, 128); // Device is reading/sending file
  channel = A0;
  BT_status = 1; //Bluetooth activated
  break;
}
}
}

```

Cada uno de los canales representa la señal obtenida por cada uno de los electrodos conectados en el cuerpo del paciente. El botón 1 permite al usuario cambiar de canal (3 canales) al mismo tiempo que se grafican las señales en tiempo real, esto permite al usuario verificar el canal donde se pueden obtener de manera más clara (mayor ganancia) las ondas ECG, una vez allí, es posible pasar a la etapa de almacenamiento mediante la pulsación del mismo botón 1.

3.3.4. Etapa de almacenamiento

El primer proceso para poder realizar operaciones con la memoria SD, es el establecer la sincronización o conexión con ella, para ello la comunidad de arduino nos ofrece algunas soluciones, entre las más populares se encuentra “SD.h” y “SDFat.h”, siendo la última la utilizada ya que implementa una serie de mejoras con respecto a la primera, como es la estabilidad en escritura/lectura, velocidad de operaciones, mayor compatibilidad con arduino mega 2560.

Esta biblioteca como primer elemento es necesario declarar los tipos de variables que

permiten la comunicación con la tarjeta SD, el tipo de dato de la variable es definida como “SdFat”, e inicializar dicha variable con el método que habilita la memoria SD para realizar operaciones “x.begin(SPEED, chipSelect)”, donde “x” representa el nombre de la variable del tipo SdFat, SPEED indica la velocidad de comunicación en el protocolo SPI, puede tomar valores HALF ó FULL, chipSelect define el número de pin que utiliza el arduino para habilitar la memoria sea lectura o escritura.

Cada vez que la escritura en memoria es activada el dispositivo generará un nuevo archivo de registros, este archivo sigue un patrón de nombre, definido como “Holter_capture_xxx.txt” donde xxx es un número incremental cada vez que se genera uno nuevo, empezando desde el “Holter_capture_000.txt” hasta el “Holter_capture_999.txt” una vez superado ese límite volverá a la cuenta inicial sustituyendo el archivo con el mismo nombre; el holter almacena en una variable en memoria EEPROM de tal manera el número de secuencia de archivos no se vea reiniciado, cuando este sea apagado.

Es importante tener en cuenta para la escritura en memoria implementa un formato para su posterior lectura y procesamiento, este formato fue establecido de la siguiente manera, la primera línea indica el encabezado de los datos en este caso “CH0,CH1,CH2,token”, donde las líneas subsecuentes corresponden a valores entre 0 - 1023, el primer valor corresponde al valor obtenido por el canal 0, seguido del canal 1 y por último el canal 2, cada uno de los valores separados el carácter ‘,’ y por último corresponde un valor que solo está presente cuando el usuario pulsa el botón de marca de evento, este valor está representado con el valor ‘1’ al final de la línea, por ejemplo:

```
CH0,CH1,CH2,token ;
54,102,402;
58,128,308;
102,180,853,1;
234,166,674,1;
192,141,520;
```

A su vez el proceso de almacenado en memoria puede ser detenido en cualquier momento mediante los pulsadores, constantemente el Holter además de guardar los valores capturados estará verificando el estado de los pulsadores, para entrar en modo graficación y cerrar el archivo utilizado. El código correspondiente al secuenciamiento de los archivos es el siguiente:

```
//Determinar el ultimo Archivo Creado
if(BT_status != 1){
    f0 = EEPROM.read(0);
    f1 = EEPROM.read(1);
    f2 = EEPROM.read(2);
    i = f0 + (f1*10)+ (f2*100);
```

```

i++;
if (i==1000)
    i=0;
filename [15] = (i/100) + '0';
filename [16] = (i/10)%10 + '0';
filename [17] = (i%10) + '0';
//****
f2 = i/100;
f1 = (i/10)%10;
f0 = i%10;
EEPROM.write(0, f0);
EEPROM.write(1, f1);
EEPROM.write(2, f2);

//****

```

Luego del secuenciamiento de los archivos, se procede a abrir un nuevo archivo para la escritura, al mismo tiempo de que se verifica el estado de los botones para activar/desactivar la captura o entrar en modo transmisión de datos.

```

if (!myFile.open(filename , ORDWR \ | O_CREAT \ | O_AT_END)) {
    myGLCD.clrScr ();
    myGLCD.setColor (255,255,255);
    myGLCD.print ("ERROR_OPENING_FILE" ,280,120,200);
    sd.errorHalt ("opening_file_for_write_failed");
} else {
    analogWrite(7, 64); //Device is writing file LED ON!
}
myFile.println ('CH0,CH1,CH2,token;'); //capture format
Timer1.attachInterrupt (ISR_Samples); //capture activated.
Timer1.initialize (3999);
interrupts ();
while (1) {
    delay (300);
    buttonState = digitalRead (buttonPin);
    if (buttonState == LOW) {
        analogWrite(7, 0); // File LED OFF
        channel = A0;
        break;
    } else if ((digitalRead (buttonBluetooth) == LOW)) {

```

```

        analogWrite(7, 0);           // File LED OFF
        analogWrite(5, 128);        //Device ON
        channel = A0;
        BT_status = 1;              //Bluetooth activated
        break;
    }
}
noInterrupts ();
myFile.close ();
//Turn ON LCD Screen
analogWrite(13,128);
delay(50);

//****

```

Por ultimo la funcion que es llamada cuando se produce la interrupcion de timer0 ISR_Samples() es la siguiente:

```

void ISR_Samples(){ // rutina captura senales
    myFile.print(analogRead(A0));
    myFile.print(",");
    myFile.print(analogRead(A1));
    myFile.print(",");
    myFile.print(analogRead(A2));
    if(digitalRead(buttonSignal) == LOW){
        myFile.print(",1;");
    }else{
        myFile.print(";");
    }
    Timer1.initialize(3999);
}

```

3.3.5. Etapa de transmisión

Para el establecimiento de la conexión por parte de otro host, se implementó una interfaz inalámbrica bluetooth con el modulo HC-06, este dispositivo entre sus características cuenta con una interfaz serial (RS-232) de comunicación, el arduino hace uso de esta interfaz para la configuración y transmisión de datos específicamente el puerto "Serial3" del arduino, correspondiendo al modelo propuesto en la figura 3.15.

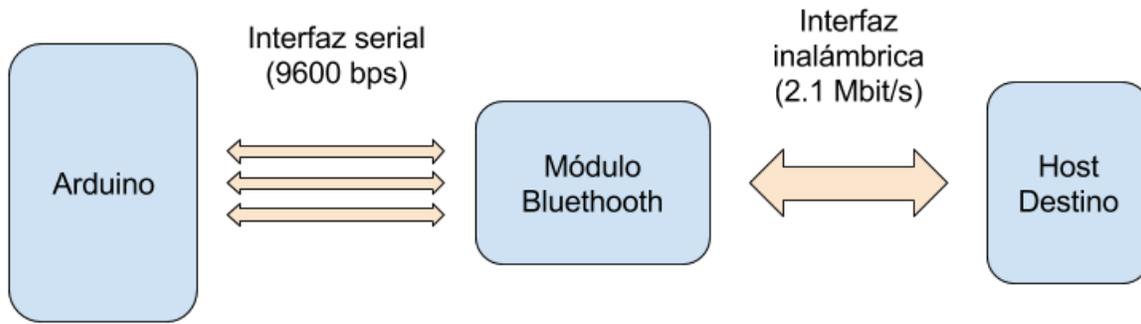


Figura 3.15: Conexión Bluetooth

Si bien en la comunicación bluetooth se encarga de establecer la conexión y emparejamiento de los dispositivos, fue necesario definir un “protocolo” para la comunicación entre el Holter ECG y el host destino, el cual se definió como se muestra en la figura 3.16.

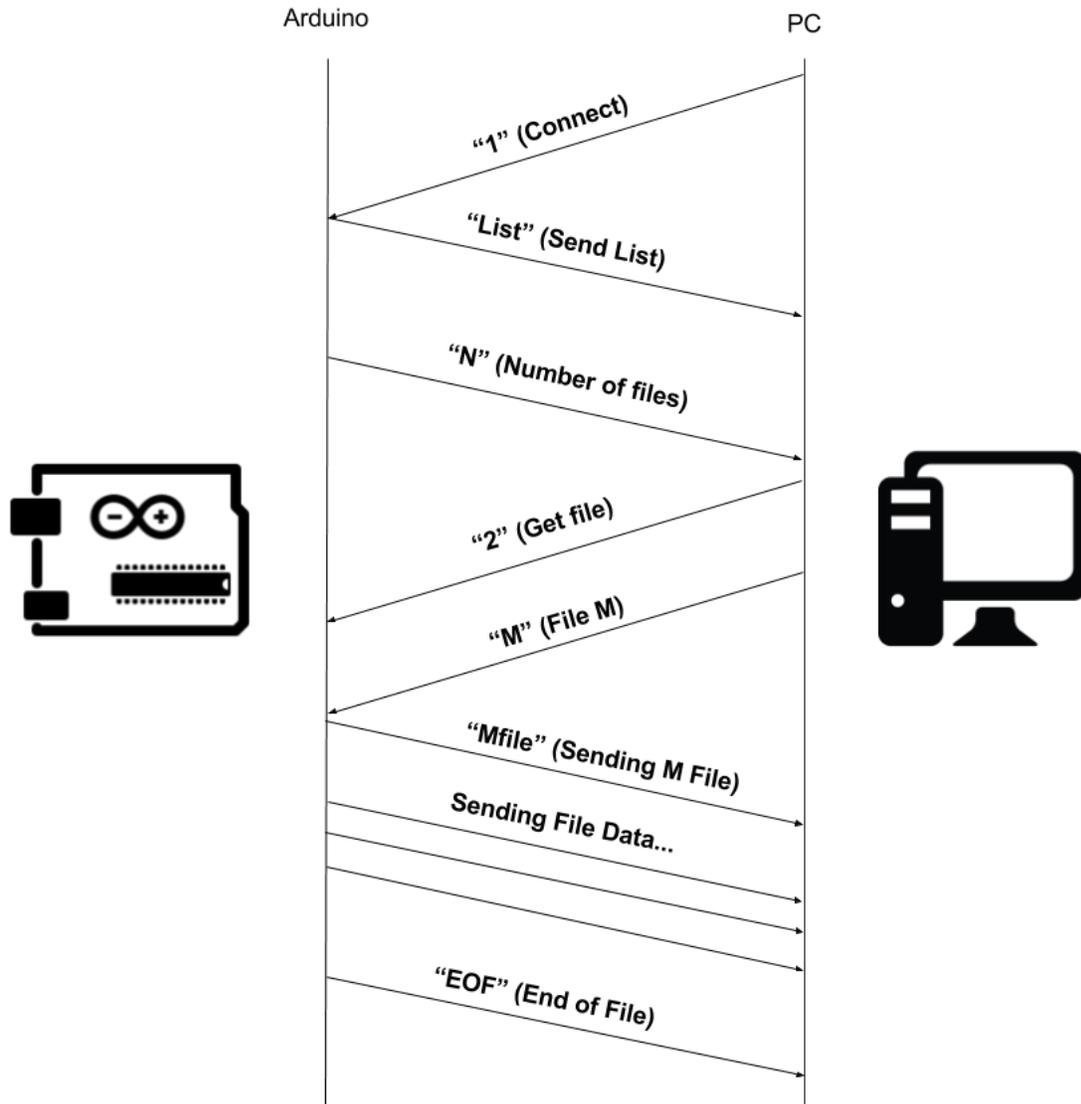


Figura 3.16: Protocolo de comunicacion ECG

En primer lugar el PC envía un caracter en este caso denotado por "1", con el fin de indicar al arduino que se desea establecer la comunicación para transmisión de archivos, el arduino responde con una cadena de caracteres "List" indicando que en la próxima trama se indica el número de archivos alojados en el arduino, luego de esto el arduino envía el número de archivos almacenados en la memoria, y de manera asíncrona el arduino solo espera que archivo desea obtener el Host remoto (PC), para ello, el PC envía el caracter "2" indicando que se desea obtener un archivo seguido de otra trama indicando el número de archivo a obtener, el arduino responde indicando que se procese a enviar el archivo

con el mensaje “Mfile” donde M representa el número de archivo solicitado, se procede a enviar todo el contenido del archivo, y por último cuando ya todo el archivo ha sido enviado, se envía el mensaje “EOF” indicando el fin del archivo y de la conexión.

Debido a que la conexión ya fue establecida previamente, el PC puede hacer solicitud de otro archivo en cualquier momento sin necesidad de tener que establecer la conexión nuevamente.

En la implementación desde el arduino para la transmisión de datos, este debe encontrar en modo “transmisión” para el envío o recepción de datos correspondiente al siguiente segmento de código:

```
while(BT_status){
  if(Serial3.available()) //Si llega un dato por el puerto BT
  {
    c = Serial3.read();

    if(c == '1'){
      Serial3.write("List\n");
      delay(25);
      f0 = EEPROM.read(0);
      f1 = EEPROM.read(1);
      f2 = EEPROM.read(2);
      i = f0 + (f1*10)+ (f2*100);
      Serial3.write(i);
      Serial.write(i);
    }else if(c=='2'){ // Envio de archivo
      int ln = 0;
      char fbuffer [] = "000file";
      while(c != '\n'){ // Se obtiene el archivo solicitado
        delay(75);
        c = Serial3.read();
        if(c != '\n'){
          fbuffer[ln] = c;
          ln++;
        }
      }
      if(ln==1){
        filename[17] = fbuffer[0];
      }else if(ln==2){
        filename[16] = fbuffer[0];
        filename[17] = fbuffer[ln-1];
      }
    }
  }
}
```

```

    }else if(ln==3){
        filename[15] = fbuffer[0];
        filename[16] = fbuffer[ln-1];
        filename[17] = fbuffer[ln-2];
    }
    fbuffer[0] = filename[15];
    fbuffer[1] = filename[16];
    fbuffer[2] = filename[17];

    if (!myFile2.open(filename, O_READ )){
        myGLCD.clrScr();
        myGLCD.setColor(255,255,255);
        myGLCD.print("ERROR_OPENING_FILE",280,120,200);
        sd.errorHalt("opening_file_for_write_failed");
    }else{
        size_t n;
        ln = 1;

        Serial3.write(fbuffer);
        while ((n = myFile2.fgets(line, sizeof(line))) > 0) {
            Serial3.write(line);
        }
        Serial3.write("EF");
        myFile2.close();
    }
}
}
}
if((digitalRead(buttonPin) == LOW)){
    analogWrite(5, 0); // Device is reading/sending file LED ON
    BT_status = 0;
    break;
}
}
}
}

```

Sin embargo para llevar a cabo la comunicación desde el PC fue necesario el desarrollo de una aplicación para la comunicación con el arduino, esta aplicación fue desarrollada en Java 1.8.0_131 y el IDE Netbeans 8.2, en un ambiente 64 bits, llamada “ECG Receiver”.

3.3.6. ECG Receiver

Esta aplicación fue desarrollada con mediante la librería “Java Simple Serial Connector” (JSSC) el cual permite el manejo de los puertos seriales o paralelos del PC, esta librería licenciada como código abierto, cuenta con gran soporte de la comunidad desarrolladora y es independiente del sistema operativo en el que se despliegue “crossplatform”, en el caso de que el PC no cuente con interfaz bluetooth, es posible utilizar un dispositivo Dongle bluetooth USB, los dispositivo bluetooth para el envío y recepción de datos, implementan una comunicación interna del tipo serial “COM” entre el dispositivo bluetooth (Dongle) y el PC, todos los datos que sean enviados hacia ese puerto, serán recibidos por el dispositivo remoto en este caso el Arduino.

Para el despliegue y utilización del ECG Receiver, es necesario contar con al menos JRE 1.8_101_x64 instalado en el sistema operativo; en primer lugar una vez conectado el dispositivo Dongle bluetooth este deberá abrir un puerto COM, es importante saber que puerto está utilizando para la comunicación, por lo general el dispositivo utilizara este puerto para la comunicación con el sistema operativo, y así poder ejercer configuraciones sobre el, bien sea Activar/Desactivar/Emparejar/Desemparejar, una manera de visualizar este puerto en un sistema operativo Windows es por medio del “administrador de dispositivos”, donde se debe visualizar el dispositivo con el puerto a utilizar, ver figura 3.17.

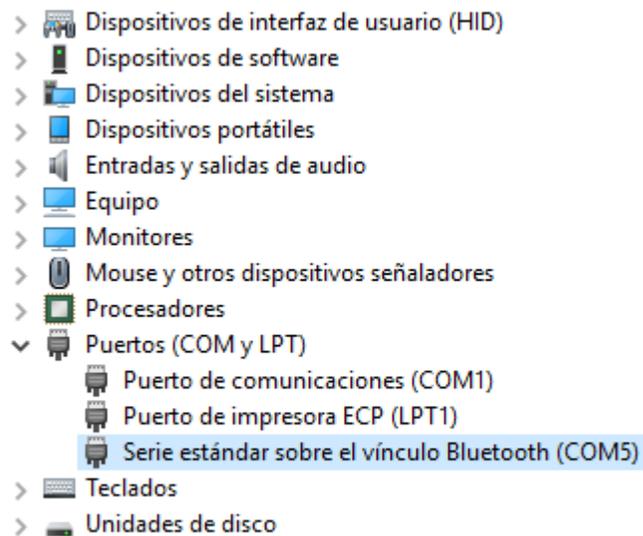
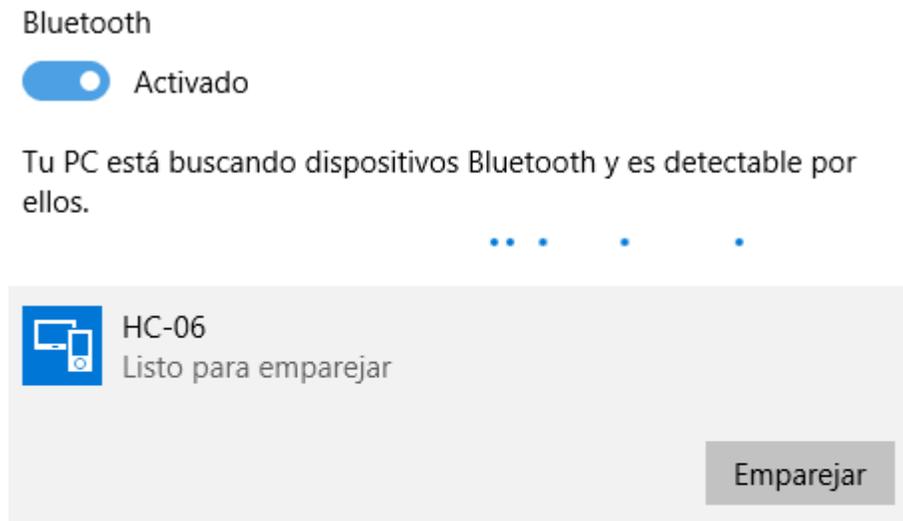


Figura 3.17: Dispositivo Bluetooth mediante puerto COM5

En este punto, es necesario emparejar los dispositivos previo a la comunicación, para este caso las tomas de ejemplo se realizaron bajo un ambiente Windows específicamente Windows 10, el dispositivo ECG debe estar encendido y preferiblemente en modo transmisión, como se muestra en la figura , donde se empareja el dispositivo muestra el nombre

asignado en este caso “HC-06” al momento de emparejar se debe introducir un pin de acceso, para el caso del HC-06 corresponde a “1234”.



Opciones de configuración relacionadas

[Más opciones de Bluetooth](#)

[Enviar o recibir archivos a través de Bluetooth](#)

Figura 3.18: Emparejamiento del dispositivo

Una vez emparejado el dispositivo, el sistema operativo procede a abrir otro puerto COM exclusivo para la comunicación con el dispositivo remoto, es importante tener en cuenta que este abre automáticamente 2 puertos COM, uno como salida y otro como entrada a efectos de la aplicación se utiliza el puerto definido como saliente, para ello en opciones de Bluetooth , en la pestaña “puertos COM” es posible ver los puertos utilizados, ver figura 3.19.

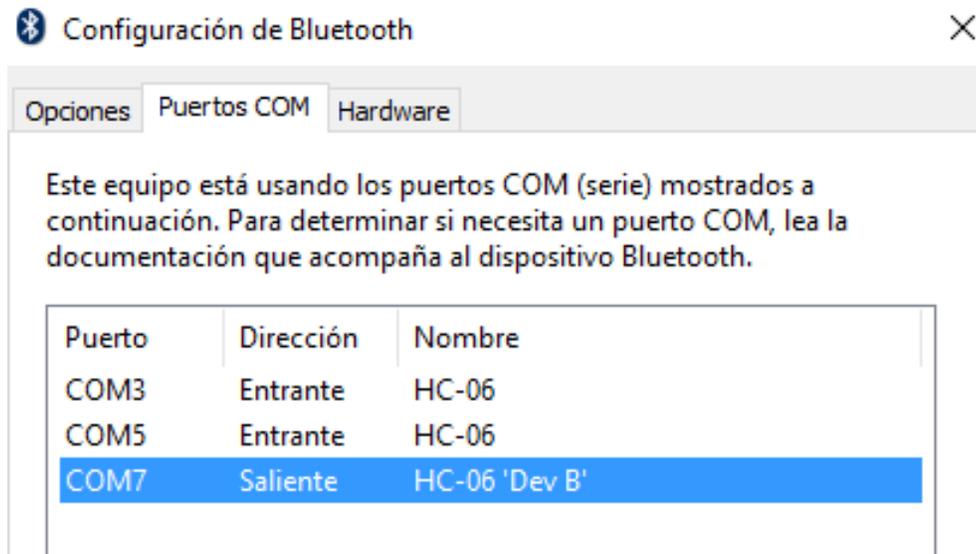


Figura 3.19: Puertos utilizados por Bluetooth

Teniendo en cuenta el puerto definido como saliente ya es posible iniciar la aplicación “ECG Receiver”, seleccionando el puerto en cuestión y estableciendo la conexión con el puerto COM.

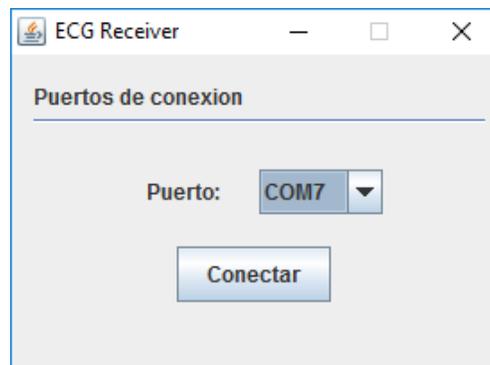


Figura 3.20: Primera vista ECG Receiver

Luego de establecer la conexión la aplicación muestra una lista de archivos disponibles para transferir, y un campo donde se indica donde se guardarán los archivos transferidos.

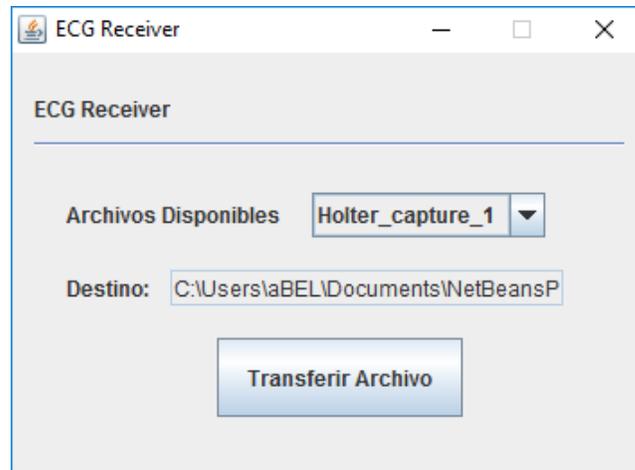


Figura 3.21: Panel Principal ECG Receiver

Al momento de que el proceso de transferencia de archivo esté ejecutándose un mensaje es mostrado ver figura 3.22, indicando que el proceso está en curso, una vez que se haya terminado la transferencia el mensaje de confirmación es mostrado ver figura 3.23.

Desde el panel frontal es posible transferir otros archivos de ser necesario pero con la limitante de que solo se puede transferir un archivo a la vez, una consideración en relación a la transmisión de archivos, es que en términos de peso, un archivo de registros de 24 horas puede rondar los 260MB - 281MB, la comunicación establecida en las interfaces seriales desde ambos puntos (PC y arduino) implementan una tasa de baudios de 9600 bps ya que otras tasas de baudios más altas se torna inestable, esta velocidad de comunicación indica que la tasa de transferencia es muy cercana a los 960 Bps (Bytes por segundo) por lo que una transferencia de un archivo de 260MB tomaría 75,23 horas.

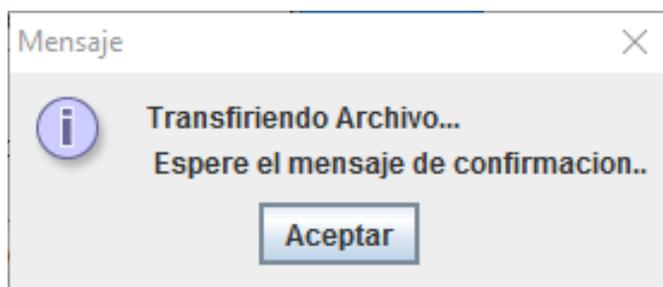


Figura 3.22: Mensaje de estado de la aplicación

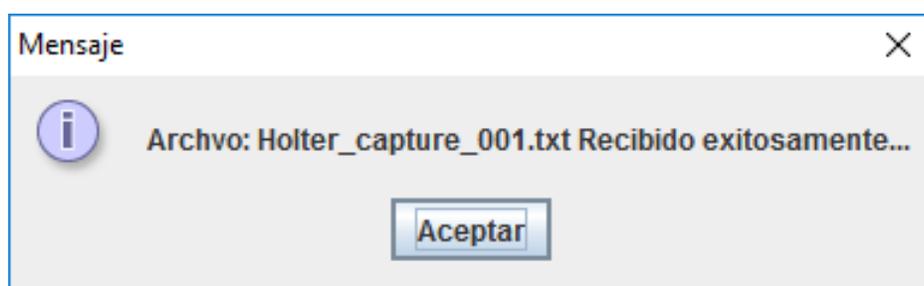


Figura 3.23: Mensaje de confirmación

Capítulo 4

Pruebas de la plataforma

La ejecución de las pruebas se realizaron en dos fases; la primera fase consta de la evaluación del dispositivo mediante equipos de medición en el laboratorio de postgrado, la segunda fase corresponde a la evaluar el desempeño del dispositivo sobre distintos pacientes.

- **Fase I**

Para la ejecución de esta fase se generaron varias señales de distintas características (amplitud, frecuencia, fase) que funcionara como señal de entrada para el Holter, para esto, se utilizó un equipo especializado específicamente un generador de señales, así mismo a manera de verificar si efectivamente las señales graficadas corresponden con las características de la señal recolectada. La conexión para la realización de las pruebas se muestra en la figura 4.1.

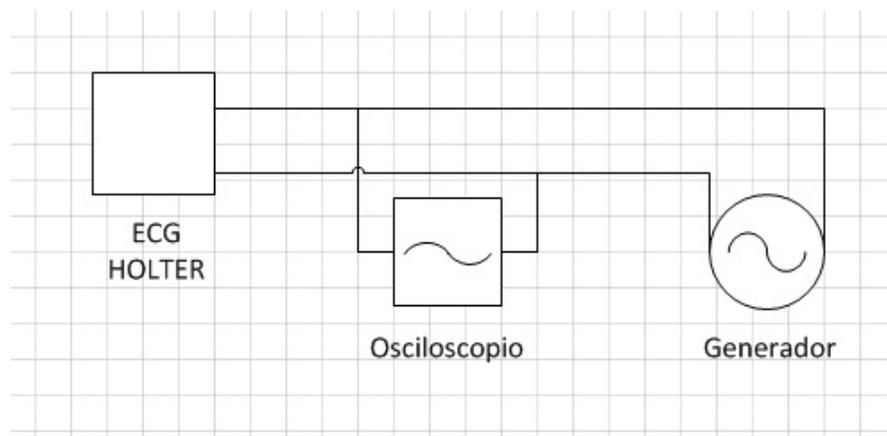


Figura 4.1: Esquemática de conexión de pruebas

Fue generada una señal de forma sinusoidal de 10 Hz con una amplitud de 2,3 Volts, y fase 0, la gráfica de la señal generada por el dispositivo corresponde con los resultados

esperados, para una señal 10 Hz, cada ciclo debe tener una duración de 100 ms, que en términos del mallado electrocardiográfico corresponde con 2 unidades y media (40 ms /unidad), y una amplitud por unidad de 247 mV, se puede apreciar en la figura 4.2

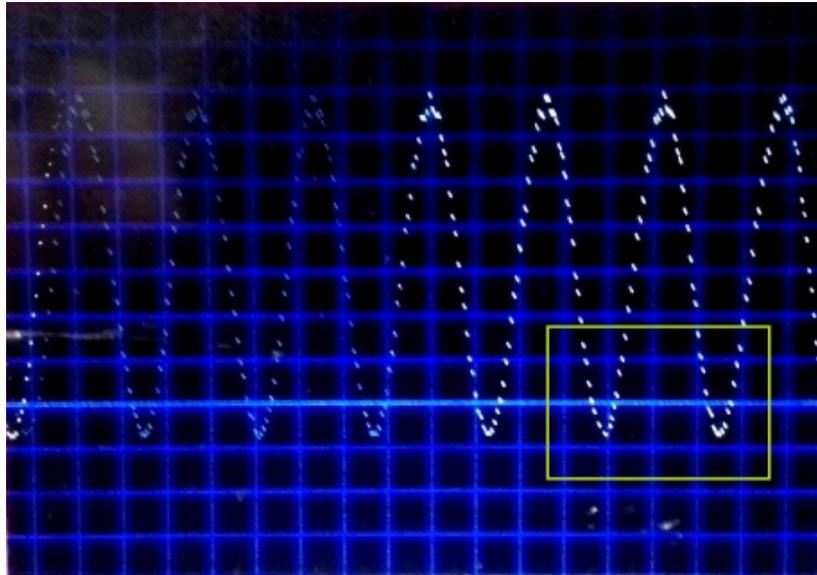


Figura 4.2: Graficación de señal generada

Es de notar en el recuadro verde, que la sucesión de puntos que van dibujando la forma sinusoidal de la señal, el primer punto mínimo de izquierda a derecha corresponde a una de las líneas verticales del mallado ECG, por lo que después de cumplirse un ciclo completo de la señal, en la gráfica debe corresponder a dos unidades y media del mallado ECG, efectivamente se puede apreciar que posee una buena aproximación al resultado esperado, de la misma manera pudo realizarse pruebas para frecuencias 20 Hz, 50 Hz, 120 Hz con resultados satisfactorios, en la figura 4.3 se puede ver la señal de 10 Hz representada en el osciloscopio, en la configuración del osciloscopio cada unidad corresponde a 22 ms en el tiempo y 0,25 mv en la amplitud.

■ Fase II

En esta fase de pruebas se realizaron capturas de aproximadamente cinco minutos con voluntarios de las siguientes características:

Se observaron los tres canales de los diferentes voluntarios, logrando visualizarse las tres derivaciones, se tomaron capturas de aproximadamente 5 minutos de duración, con los distintos pacientes, un factor que es determinante en la calidad de las capturas es la reutilización de los electrodos, es importante reemplazarlos lo más frecuentemente por cada paciente registrado, aquí una captura del primer voluntario. Ver figura 4.5.

Todas las capturas a los voluntarios fueron realizadas siguiente la configuración de 5 terminales bajo 3 canales independientes.

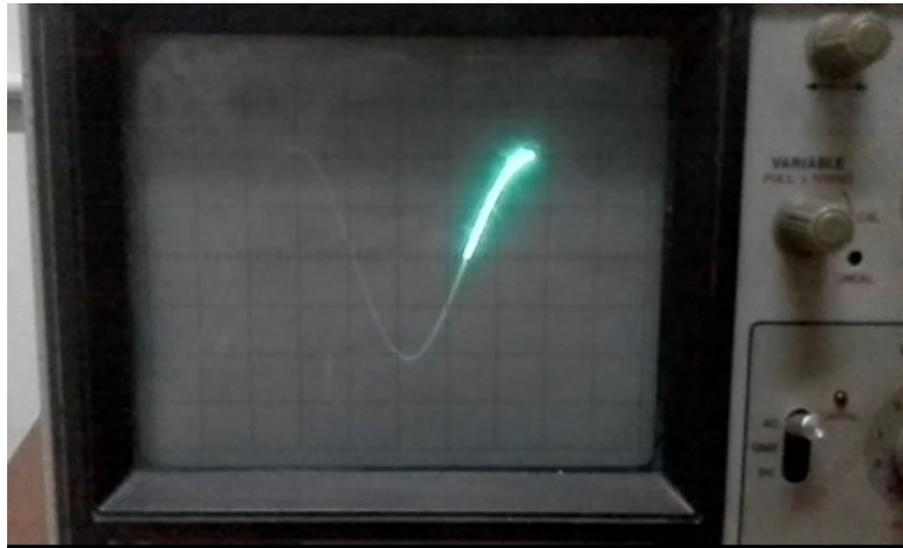


Figura 4.3: Representación de la señal sinusoidal en osciloscopio



Figura 4.4: Generador de señal GoldStar FG-8002

Cuadro 4.1: Lista de voluntarios

Número	Edad	Sexo	Peso (Kg)
1	25	Masculino	65
2	24	Femenino	57
3	28	Masculino	89
4	39	Masculino	70
5	26	Masculino	56
6	35	Femenino	55
7	24	Femenino	53
8	56	Femenino	68
9	22	Masculino	51
10	31	Masculino	57

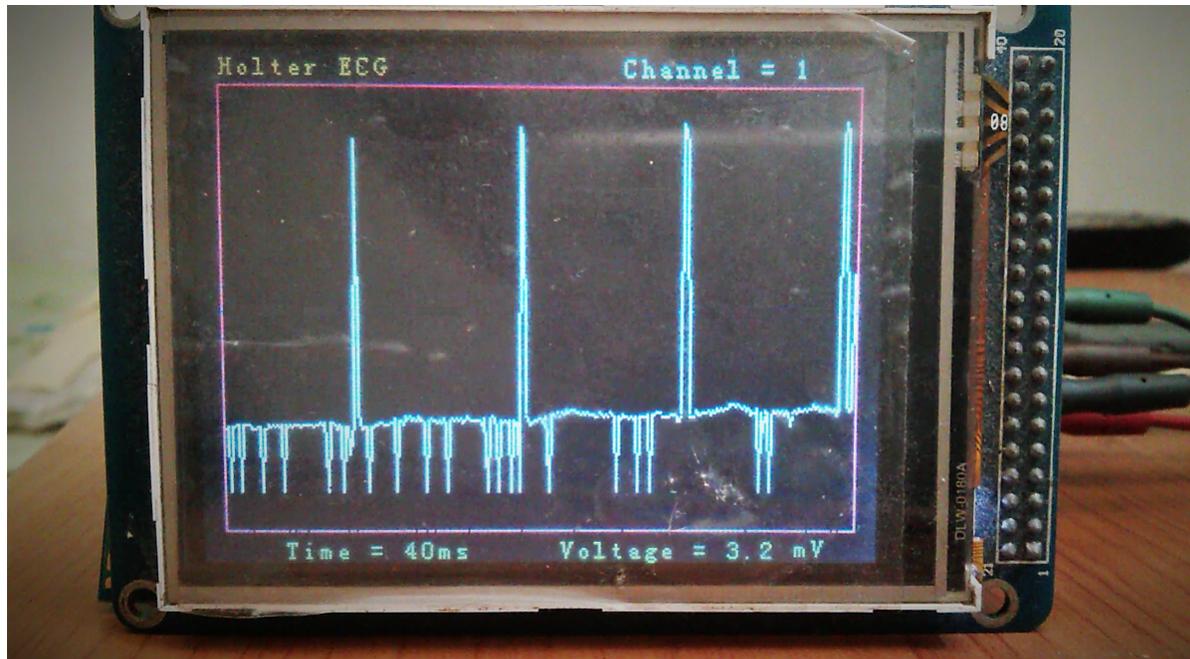


Figura 4.5: Registro voluntario número 1

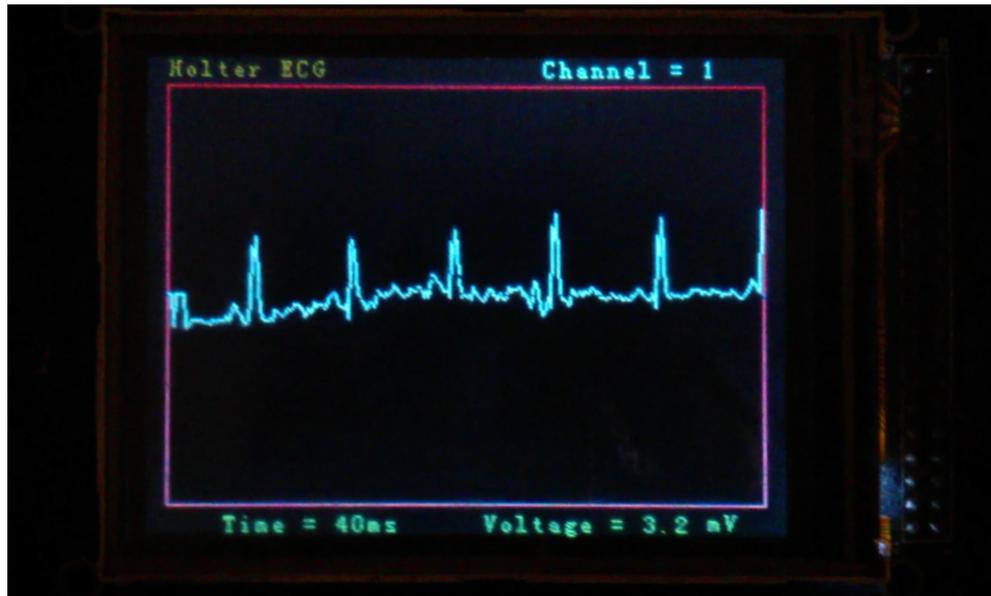


Figura 4.6: Registro voluntario número 8

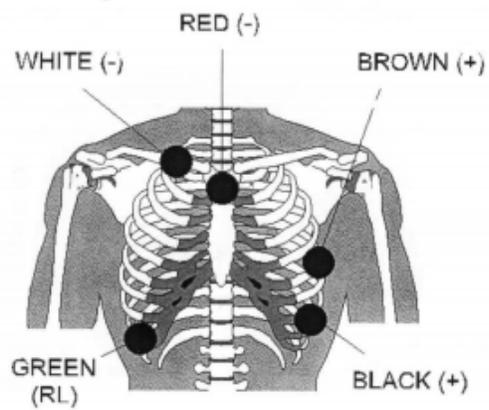


Figura 4.7: Configuración de 3 canales y un cable de 5 terminales
Fuente: Rozinn Electronics Operator's Manual, Model Number RZ151

4.1. Resultados

Con el trabajo realizado se logró diseñar, implementar y probar el dispositivo, capaz de obtener registros electrocardiográficos de pacientes, otorgando una herramienta que cumple con las necesidades de la comunidad universitaria para la investigación, fomentando y apoyando las bondades de la filosofía libre, el dispositivo consta de 7 tarjetas independientes que se acoplan e interconectan para llevar a cabo todas las funciones para el que fue diseñado, además de las tarjetas es necesario una serie de cables conectados en un extremo las entradas de las placas amplificadoras, y en el otro extremo los electrodos. La implementación de la fase de graficación debido a los recursos de hardware limitados del arduino se pudieron obtener hasta 83 muestras por segundo, pudiendo representar en pantalla hasta 3.5 segundos de registros ECG, en la figuras subsecuentes se pueden apreciar más de cuatro complejos, siendo este el resultado final de la graficación de las ondas en pantalla LCD.

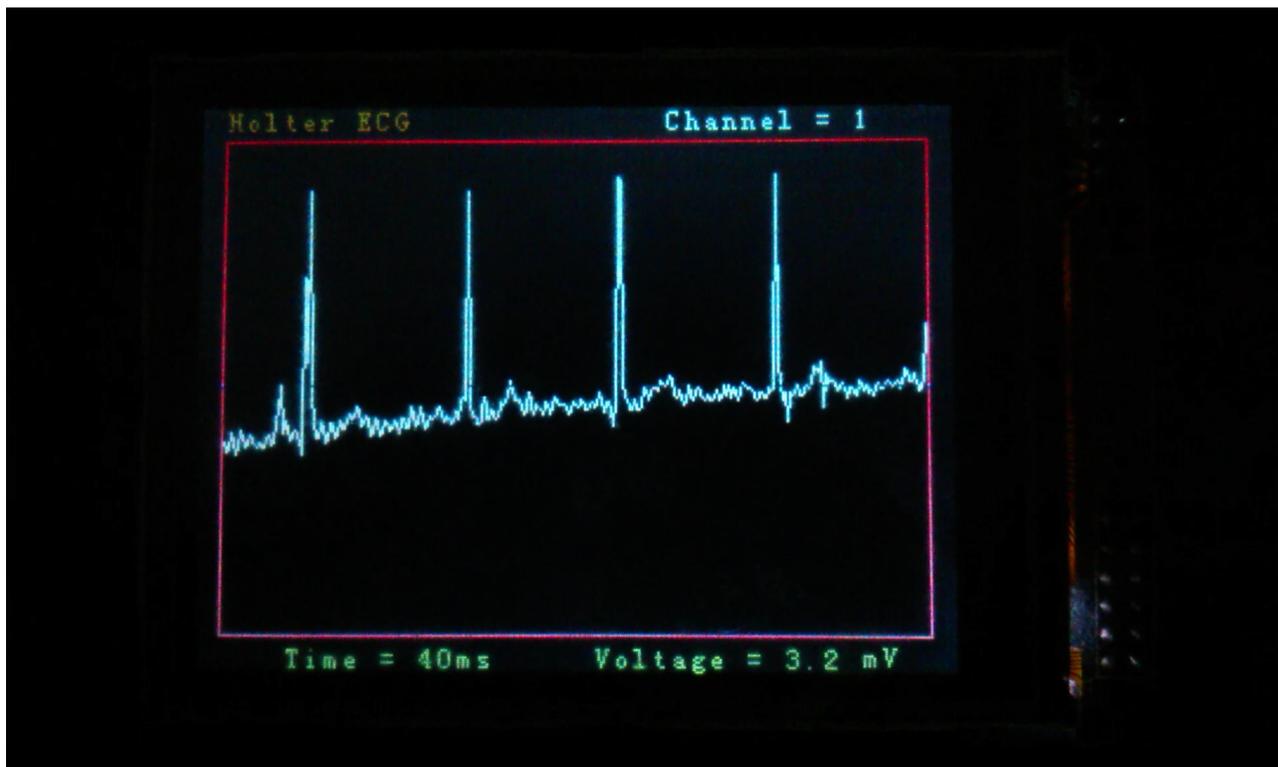


Figura 4.8: Gráfica sobre pantalla LCD Holter ECG voluntario 1

Este dispositivo almacena los registros en un archivo de texto plano en memoria SD, donde simultáneamente los 3 canales de entrada son almacenados a una tasa de 500 muestras por segundo, el almacenamiento de los registros pueden tomar un tamaño considerable dependiendo de la duración del registro, un registro de unos pocos minutos ronda los pocos KBs mientras que un registro de 24 horas puede alcanzar hasta 290 MB. La

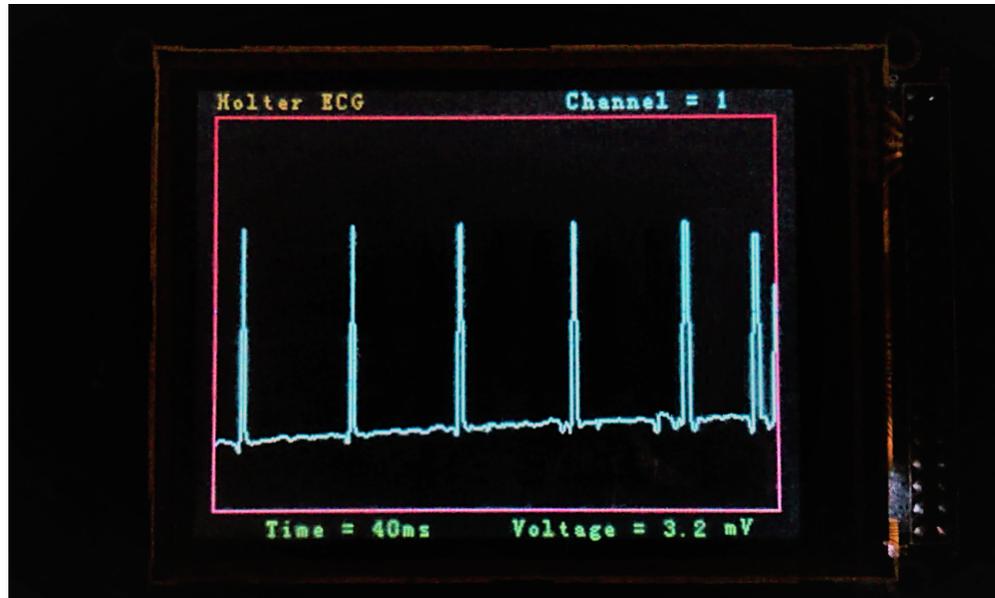


Figura 4.9: Gráfica sobre pantalla LCD Holter ECG voluntario 2

presencia de una memoria SD es indispensable para que el algoritmo pueda funcionar, esta memoria puede ser extraída del arduino en caso de requerirse, pudiendo acceder a todos los datos almacenados como una memoria común de esta tecnología, sin embargo es caso de insertarse de nuevo a la ranura en la placa del holter es necesario reiniciar el dispositivo arduino. Un valor agregado al arduino es la capacidad de transmitir de manera inalámbrica los datos almacenados en la memoria SD, un elemento fundamental es que el arduino debe estar en modo transmisión de datos, para así poder establecer la conexión y transmitir los datos. Como elemento adicional se desarrolló una aplicación capaz establecer la conexión vía bluetooth con el dispositivo y transmitir un archivo de registros al pc, para su posterior uso. La aplicación “ECG Receiver” es una aplicación específicamente diseñada y desarrollada para el software arduino implementado, el módulo HC-06 bluetooth implementado en el arduino, cuando establece la conexión con host, el led presente en este módulo dejara de parpadear y se mantendrá fijo, indicando que la conexión se ha establecido con éxito. Algunos de los registros obtenidos durante la fase de pruebas fueron transmitidos al PC y graficados de tal manera que se pueda tener una visión comparativa de las ondas ECG con respecto a las generadas por el arduino, pudiéndose detallar características de la señal como periodicidad, complejos QRS, calidad de los registros.

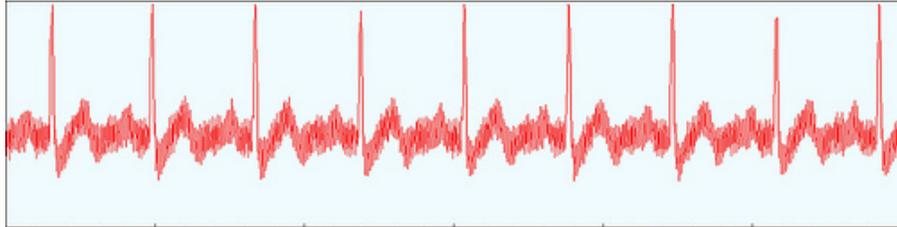


Figura 4.10: Gráfica en PC de registro de voluntario número 1

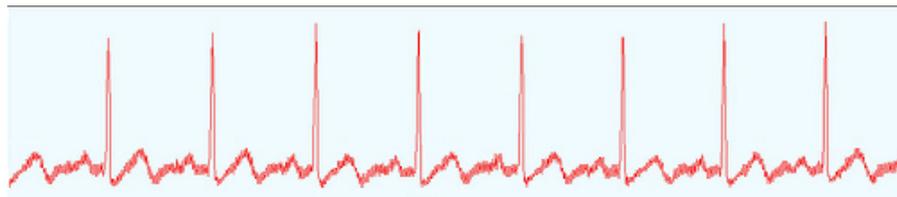


Figura 4.11: Gráfica en PC de registro de voluntario número 2

4.2. Conclusiones

Como elemento concluyente se pudo encontrar documentación que da lugar para la implementación de aplicaciones específicas, basado en dispositivos de propósito general como son los microcontroladores, donde se pudo estudiar las familias principales de los fabricantes tales como Microchip Technologies, Atmel Corporation y Arduino evaluando cada uno de ellos basado en las bondades ofrecidas, además de que debía someterse a la filosofía libre, siendo el ultimo nombrado el más indicado para llevar a cabo el trabajo realizado.

Se logró implementar un sistema de adquisición capaz de monitorizar múltiples canales (derivaciones), minimizando la pérdida de información en el proceso de adquisición, procesando la señal obtenida de tal manera de optimizar los recursos del hardware, adicionalmente se acoplo y utilizo las placas desarrolladas por la Ing. Aida Pérez para amplificar, filtrar y condicionar las señales generadas por el paciente.

Se implementó un algoritmo que permite acondicionar y graficar los registros obtenidos en tiempo real, representándolas en una escala visible y un intervalo de tiempo considerable de tal manera que puedan apreciarse al menos cuatro complejos completos en pantalla, a su vez estos cumplen con los estándares mínimos de una onda ECG, el algoritmo implementado hace uso del tiempo mínimo posible para esta tarea, pudiendo también verificar el estado de los pulsadores por si algunos de estos fue presionado. En la fase de pruebas pudo apreciarse la señal graficada y comparada con el osciloscopio, demostrando que el dispositivo es capaz de manejar ondas de baja frecuencia con una exactitud adecuada para esta tarea.

Fue implementado el grabado de los registros obtenidos en un dispositivo digital específicamente una memoria SD, esto haciendo uso de una placa desarrollada por la Ing. Aida Pérez el cual adapta una ranura SD, LEDs y pulsadores para la manipulación y configuración del dispositivo, la implementación realizada hace uso de todos estos recursos de hardware.

Se desarrolló e implemento la transmisión de archivos de registros vía bluetooth desde el dispositivo holter hasta un PC, para poder llevar a cabo este punto, se desarrolló una aplicación de escritorio que se encarga de establecer y recibir los archivos desde el dispositivo bluetooth del lado del Holter, para esto la PC debe contar una interfaz bluetooth.

Se logró implementar un formato en el almacenamiento de los registros, de tal manera que facilite y permita el procesamiento de los valores almacenados por parte de un software de un tercero, así mismo se desarrolló la capacidad por parte del Arduino de escribir eventualidades en el archivo de registros almacenado, donde el paciente en caso de sentir alguna anomalía pueda registrarlo como un punto de interés en el archivo de registros presionando uno de los botones, las fechas y horas de creación del archivo son las que implementa automáticamente la librería en el momento de creación del mismo.

Se evaluaron las gráficas y señales generadas por el Holter en pantalla, siendo este

una representación con las características principales de una señal ECG, esta señal puede verse más claro dadas mejores condiciones de relación señal/ruido o bien sea mejorando el algoritmo para obtener una señal más clara, el cual queda como un elemento de mejora en trabajos futuros.

4.3. Recomendaciones

- Utilizar electrodos nuevos y adicionar gel electroconductor al momento de realizar los registros.
 - Alimentar el dispositivo desde una fuente independiente de energia, por ejemplo una batería.
 - Llevar a cabo los registros en un ambiente con la menor cantidad de ruido electro-magnético posible, por ejemplo redes wifi, celulares, bluetooth.
 - No se debe tocar al paciente directamente cuando se realiza los registros ya que esto se ve reflejado como interferencia en los registros ECG.
 - El paciente debe estar aislado de cualquier elemento que este conectado directamente a tierra, por ejemplo estar sin calzado durante los registros.
-

Bibliografía

- [1] Manado E. Menendez L. Fernandez L. Lopez E. Microcontroladores PIC. Sistema integrado para el autoaprendizaje. 2007. https://books.google.co.ve/books?id=86uGLMp4vgAC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false (2016).
 - [2] Banzi M. Shiloh M. Getting Started with Arduino. 2014. https://media.digikey.com/pdf/Data%20Sheets/O'Reilly_PDFs/Make_Getting_Started_with_Arduino_3E_9781449363338.pdf (2016).
 - [3] Toral S. y Barrero Federico S. Gallardo, Lillo Javier. FUNDAMENTOS, ESTRUCTURA Y PROGRAMACIÓN DE LOS PROCESADORES DIGITALES DE SEÑAL. FAMILIA TMS320C3X DE TEXAS INSTRUMENTS. 2002. <https://idus.us.es/xmlui/handle/11441/49281> (2017).
 - [4] Dra Ines Banquin. Holter De 24 Horas. 2015.
 - [5] Leopoldo Pavesi Farriol Cristian Vidal Silva. DESARROLLO DE UN SISTEMA DE ADQUISICIÓN Y TRATAMIENTO DE SEÑALES ELECTROCARDIOGRÁFICAS. 2004. <http://www.scielo.cl/pdf/rfacing/v13n1/art05.pdf> (2016).
 - [6] Free Software Foundation. ¿Qué es el software libre? 2016. <http://www.gnu.org/philosophy/free-sw.es.html> (2017).
 - [7] Francisco Gómez-Arribas Iván González, Juan González. Hardware libre, clasificación y desarrollo de hardware reconfigurable en entornos GNU/Linux. 2003.
 - [8] Copyright Law of the United States. Circular 92, Library of Congress. 2011.
 - [9] Christopher J. Wells. Telecommunications - Pulse Code Modulation. 2016. <https://www.technologyuk.net/telecommunications/telecom-principles/pulse-code-modulation.shtml> (2016).
 - [10] Cortes Cristian Zuñiga Fonseca. Diseño y construcción de un electrocardiógrafo en una PC por puerto paralelo de bajo costo. 2010.
-

- [11] Dr. Gallegos Serrano Samuel P. Dr. Manuel Arteaga M. Proyecto docente desarrollo del corazón. 2010.
- [12] Rafael Barea Navarro. Instrumentación Biomédica. Departamento Electrónica. 2014.
-