

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN



**“DESARROLLO DE UN SISTEMA DE TELEMEDICINA QUE UTILIZANDO
LA TECNOLOGÍA DE LAS TABLETAS CAPTE LA ACTIVIDAD
ELÉCTRICA DEL CORAZÓN DE UN PACIENTE Y JUNTO CON SU
INFORMACIÓN PERSONAL LA TRANSMITA POR INTERNET”**

TESINA DE SEMINARIO

Previo a la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

**INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRÓNICA Y
AUTOMATIZACIÓN INDUSTRIAL**

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

Gionella Yvett Rodríguez Alarcón

Blanche Jaqueline Verdezoto Mendoza

Francisco David Icaza Paredes

Guayaquil – Ecuador

2013

AGRADECIMIENTO

Agradecemos primeramente a Dios, por darnos la sabiduría y fortaleza para culminar este proyecto; a nuestros padres por su amor incondicional, a nuestro director de tesis, el M.Sc. Miguel Yapur Auad por guiarnos en la elaboración de esta tesis.

DEDICATORIA

Dedico esta tesina sobre todo a Dios, quien es la fuente de mi vida y Sabiduría, a mis padres Milton y Patricia, a mi hermana Evelyn y a mi abuelita Carmen, por su amor, por su apoyo incondicional por ser esos pilares en mi vida, por siempre tener una palabra de ánimos cuando más lo necesitaba. A ustedes que lo han dado todo por mi les dedico mis éxitos. Los amo.

Gionella Yvett Rodríguez Alarcón

Este trabajo va dedicado a mi mamá, que desde el cielo guía cada uno de mis pasos, a mi hermano y a la Sra. Idalia Zambrano, por su amor, consejos, paciencia y apoyo incondicional en toda mi carrera universitaria.

**Blanche Jaqueline Verdezoto
Mendoza.**

Esta tesina va dedicado a la esencia de mi vida Dios, a mis padres Venecia y Francisco por sus consejos, fortaleza, por estar conmigo en todo momento, por creer en mí cuando mi respirar se había debilitado, por sus oraciones en cada mañana, por su amor incondicional, por sus abrazos de seguridad, por su tiempo dedicado, a mi gran amigo Héctor por su apoyo y oraciones que me han ayudado en este largo camino, a mi novia por su paciencia, dedicación y su gran amor, por la felicidad que me ah dado en estos largos años de vida. A los chicos de Adolescentes Gv por su preocupación y oraciones.

Francisco David Icaza Paredes

TRIBUNAL DE SUSTENTACIÓN

Msc. Miguel Eduardo Yapur Auad

PROFESOR DEL SEMINARIO DE GRADUACIÓN

Msc. Luis Fernando Vásquez Vera

PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesina de Seminario, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral”.

(Reglamento de exámenes y títulos profesionales de la ESPOL)

Gionella Yvett Rodríguez Alarcón

Blanche Jaqueline Verdezoto Mendoza

Francisco David Icaza Paredes

RESUMEN

En el cuerpo humano se genera una gran variedad de señales eléctricas provocadas por la actividad que tiene lugar en los nervios y músculos que lo conforman.

El electrocardiograma es el requisito gráfico en función del tiempo de las variaciones de potencial eléctrico que se obtienen con el electrocardiógrafo.

Estas variaciones se las capta a través de electrodos a nivel de la piel.

El electrocardiógrafo se ha convertido, por su sencillez y bajo costo, en una de las técnicas de diagnóstico médico más utilizadas.

En este proyecto se diseñó e implementó un circuito electrocardiógrafo, el cual consiste en recoger la actividad eléctrica del corazón de un paciente; mediante comunicación Bluetooth se envían los datos procesados a una Tablet, que junto con la información personal del paciente, se transmite por Internet.

ÍNDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	III
TRIBUNAL DE SUSTENTACIÓN.....	V
DECLARACIÓN EXPRESA.....	VII
RESUMEN	VII
ABREVIATURAS	XII
SIMBOLOGÍA	XIII
ÍNDICE DE FIGURAS.....	XIVV
ÍNDICE DE TABLAS.....	XVII
INTRODUCCIÓN.....	XVII
CAPÍTULO 1	1
1 Conceptos Básicos	1
1.1 Anatomía del Corazón.....	1
1.2 El corazón y sus partes	2
1.3 Ciclo Cardíaco	3
1.4 Sistema de conducción eléctrica cardíaca.....	4
1.5 Electrocardiograma y su interpretación.....	4
1.6 Frecuencia Cardíaca	6
1.7 Derivaciones.....	7
CAPÍTULO 2.....	9
2 Transmisión de Datos	9

2.1	Descripción del Electrocardiógrafo	9
2.2	Diagrama de Bloques	10
2.3	Electrodos de Piel.....	11
2.4	Acoplamiento de Impedancia	11
2.5	Pre-amplificador	12
2.6	Aislamiento Óptico.....	13
2.7	Filtros Analógicos	14
2.7.1	Filtro Pasa-Alto	15
2.7.2	Filtro Pasa-Bajo	16
2.7.3	Filtro Rechaza-Banda	18
2.8	Amplificador.....	18
2.9	Compensación de Offset.....	19
2.10	Tarjeta de Adquisición de Datos	20
2.11	Módulo Bluetooth HC-06	21
CAPÍTULO 3.....		22
3	Diseño e implementación del CARDIOFGJ para la Tablet.....	22
3.1	Introducción	22
3.2	Lenguaje de Programación JAVA	23
3.2.1	Características	24
3.2.2	Tutorial de Java	25
3.3	Sistema Operativo Android	28
3.3.1	Historia.....	28
3.3.2	Versiones de Android	29
3.3.3	Estructura de un Proyecto de Android.....	30

3.4	Kit de desarrollo de Software SDK Android	31
3.5	Instalación del SDK.....	31
3.6	Requerimiento de la Aplicación.....	33
3.7	Desarrollo de la Aplicación	34
3.8	Menú Enfermero.....	35
3.8.1	Diagrama de Bloques	35
3.8.2	Explicación detallada del Procedimiento SDK.	36
3.9	Menú Doctor	41
3.9.1	Diagrama de Bloques	41
3.9.2	Explicación detallada del Procedimiento SDK.	42
CAPÍTULO 4.....		43
4.	Análisis y Funcionamiento del Equipo.....	43
4.1	Descripción del Proyecto.....	43
4.2	Materiales	45
4.3	Manejo del Equipo.....	47
4.3.1	Equipo a utilizar	47
4.3.2	Procedimiento.....	47
CONCLUSIONES		
RECOMENDACIONES.....		
APÉNDICE A		
APÉNDICE B		
APÉNDICE C		
APÉNDICE D		
APÉNDICE E.....		

BIBLIOGRAFÍA.....

ABREVIATURAS

SIGLAS	SIGNIFICADO
AV	Aurícula-Ventricular
RA	Brazo Derecho
LA	Brazo Izquierdo
RL	Pierna Derecha
BPM	Latidos Por Minuto
DC	Corriente Directa
EKG	Electrocardiograma
Avd	Ganancia en Modo Diferencial
Avc	Ganancia en Modo Común
CMRR	Razón de Rechazo de Modo Común
Rg	Resistencia Externa
SA	Sino-Auricular

SIMBOLOGÍA

SIMBOLO	SIGNIFICADO
Ω	Ohmio
μF	Micro-Faradio
nF	Nano-Faradio
$\text{K}\Omega$	Kilo-Ohmio
$\text{M}\Omega$	Mega-Ohmio
V	Voltios
mV	mili-voltio
A_v	Ganancia de Voltaje
ms	mili-segundos
f_c	Frecuencia de Corte
f_l	Frecuencia de Corte inferior
f_h	Frecuencia de Corte superior
Hz	Hertz

ÍNDICE DE FIGURAS

Fig. 1.1 El corazón	2
Fig. 1.2 El corazón y sus partes	3
Fig. 1.3 Onda Cardíaca e Intervalos	6
Fig. 1.4 Triángulo de Einthoven	8
Fig. 2.1 Diagrama de Bloques del Electrocardiógrafo	10
Fig. 2.2 Electrodo Tipo Parche	11
Fig. 2.3 Acoplamiento de Impedancia	12
Fig. 2.4 Amplificador de Instrumentación	13
Fig. 2.5 Conexión Etapa Opto-Acoplador	13
Fig. 2.6 Filtro Pasa-Alto	15
Fig. 2.7 Filtro Pasa-Bajo	17
Fig. 2.8 Filtro Rechaza-Banda	18
Fig. 2.9 Divisor de Voltaje	19
Fig. 2.10 Acondicionamiento de la Señal	20
Fig. 2.11 Tarjeta MEI&T04(PIC16F886)	21
Fig. 2.12 Módulo Bluetooth HC-06	21
Fig. 3.1 Logo de Android	28
Fig. 3.2 Estructura de un proyecto Android	30
Fig. 3.3 Tablet Smart PC	33
Fig. 3.4 Diagrama de Bloque Aplicación Enfermero	35

Fig. 3.5 Aplicación Enfermero	36
Fig. 3.6 Información Autores	36
Fig. 3.7 Pantalla Enfermero	37
Fig. 3.8 Menú Enfermero	37
Fig. 3.9 Ingreso de Datos	38
Fig. 3.10 Inicio obtención de datos	39
Fig. 3.11 Ver Gráfico	39
Fig. 3.12 Gráfico D1 (Derivación Bipolar)	40
Fig. 3.13 Diagrama de Bloque Aplicación Doctor	41
Fig. 3.14 Gráfico Señal Cardíaca	42
Fig. 4.1 a) Electrocardiograma en Protoboard	44
Fig. 4.1 b) Electrocardiograma en Osciloscopio Virtual	44
Fig. 4.2 Señal Cardíaca en la Tablet	48

ÍNDICE DE TABLAS

Tabla I Lista de Materiales	45
-----------------------------------	----

INTRODUCCIÓN

El presente trabajo está formado por cuatro capítulos. En el primero se explican los conceptos básicos del corazón, así como sus partes, derivaciones y el electrocardiograma.

En el segundo capítulo se describe cada una de las etapas del Electrocardiógrafo y la transmisión de datos vía Bluetooth.

En el tercer capítulo se detalla el funcionamiento de la aplicación que consta de dos partes: Menú Enfermero y Doctor.

En el Menú Enfermero se ingresan los datos del paciente, se obtiene por vía Bluetooth los datos respectivos y se envía la información al servidor.

En el Menú Doctor se descarga la información enviada por el Enfermero; después de su análisis se envía el diagnóstico respectivo al correo del paciente.

En el cuarto capítulo se detalla el análisis, funcionamiento del equipo, el material utilizado y el costo del proyecto.

CAPÍTULO 1

1. Conceptos Básicos.

1.1 Anatomía del corazón.

El corazón se encuentra en el tórax, entre los pulmones, detrás y levemente a la izquierda del esternón, recubierto por una membrana llamada pericardio; es el órgano principal del aparato circulatorio que bombea la sangre por todo el cuerpo, y proporciona oxígeno a los órganos y tejidos, con movimientos coordinados de contracción y dilatación, los cuales se producen en las aurículas y ventrículos respectivamente; una vez que se

produce la contracción de las aurículas da lugar a la dilatación de los ventrículos. [1]

En la Figura 1.1 se muestra la representación esquemática del corazón.

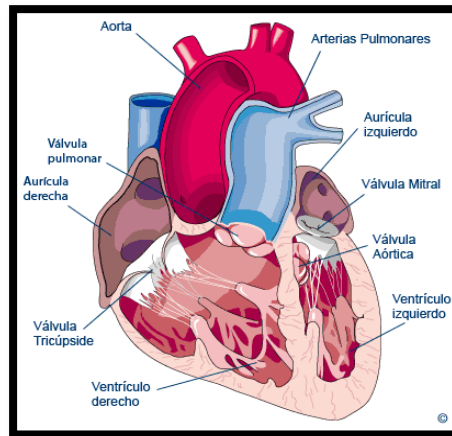


Fig. 1.1 El Corazón.

(Tomada de:

http://www.onmeda.es/enciclopedia/anatomia/anatomia_corazon.html)

1.2 El corazón y sus partes.

El corazón está dividido en cuatro partes: dos ubicadas en la parte superior que son la aurícula derecha y la aurícula izquierda; y dos situadas en la parte inferior, el ventrículo derecho y el ventrículo izquierdo. Las aurículas están separadas de los ventrículos a través de dos válvulas: Mitral y Tricúspide.

La aurícula derecha, o atrio derecho, recibe sangre pobre en oxígeno de la vena cava; la aurícula izquierda o atrio izquierdo recibe sangre rica en oxígeno de las cuatro venas pulmonares, el

ventrículo derecho recibe sangre pobre en oxígeno del atrio derecho y le envía a los pulmones a través de la arteria pulmonar, el ventrículo izquierdo recibe sangre rica en oxígeno del atrio izquierdo para enviarle al resto del cuerpo a través de la arteria aorta.

La válvula tricúspide separa y comunica el atrio derecho con el ventrículo derecho y la válvula bicúspide (válvula mitral) separa y comunica el atrio izquierdo con el ventrículo izquierdo. [2]

En la Figura 1.2 se muestra las partes del corazón.

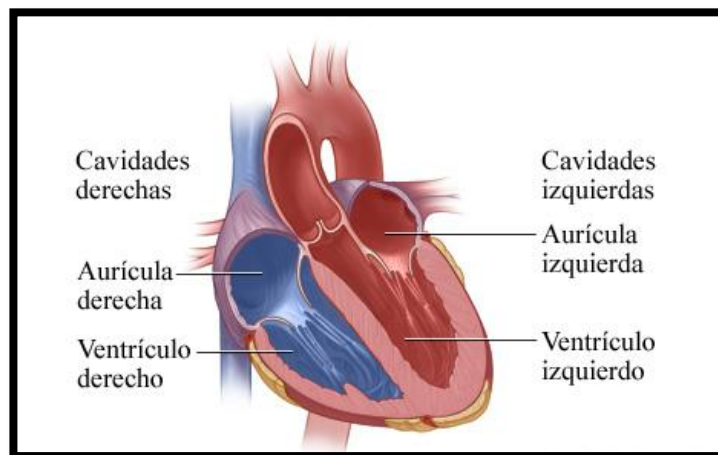


Fig. 1.2. El corazón y sus partes.

(Tomado de: <http://www.saberpractico.com/estudios/anatomia/partes-del-corazon-y-sus-funciones/>)

1.3 Ciclo Cardíaco.

El ciclo cardíaco es la secuencia de eventos eléctricos, mecánicos, sonoros y de presión, relacionados con el flujo de

su contracción y relajación de las cuatro cavidades cardíacas (aurículas y ventrículos), el cierre y apertura de las válvulas y la producción de ruidos a ellas asociados.

Este proceso transcurre en menos de un segundo. La recíproca de la duración de un ciclo es la frecuencia cardíaca (como se suele expresar en latidos por minuto, hay que multiplicar por 60 si la duración se mide en segundos). [3]

1.4 Sistema de conducción eléctrica cardíaca.

El sistema de conducción eléctrica del corazón permite que el impulso generado en el nodo sinusal (SA) sea propagado y estimule al miocardio (el músculo cardíaco), causando su contracción. Consiste en una estimulación coordinada del miocardio que permite la eficaz contracción del corazón, permitiendo de ese modo que la sangre sea bombeada por todo el cuerpo. [4]

1.5 Electrocardiograma y su interpretación.

El electrocardiograma (EKG) es una representación gráfica de la contracción cardíaca, en la cual se registra la actividad eléctrica del corazón, a través de los electrodos. [5]

Onda P: Representa la activación auricular, se caracteriza por ser la primera onda pequeña semicircular y positiva.

Onda Q: Es el comienzo de la activación ventricular, es pequeña, no es extensa ni profunda y representa la primera curva negativa seguida de la onda P.

Onda R: Representa la activación ventricular, es corta y alta. Es la primera onda positiva seguida de la onda Q.

Onda S: Representa la activación ventricular, es pequeña, al igual que la onda Q. Es la primera onda negativa seguida de la onda R.

Onda T: Pertenece a la formación del impulso, es decir, la repolarización de los ventrículos, es relativamente amplia, gruesa y semicircular.

Onda U: Es muy pequeña, positiva, semicircular que aparece seguida de la onda T y no siempre está presente. [6]

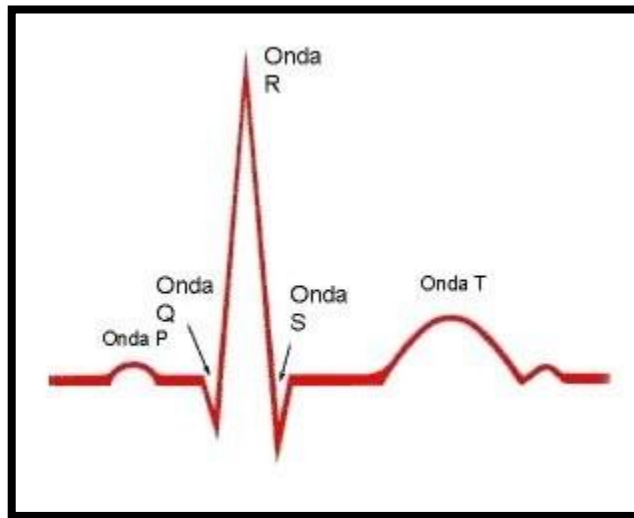


Fig. 1.3 Onda Cardíaca e Intervalos.

(Tomado de: <http://www.eccpn.aibarra.org/temario/seccion4/capitulo54/capitulo54.htm>)

1.6 Frecuencia Cardíaca.

Se define frecuencia cardíaca como el número de contracciones del corazón o pulsaciones por unidad de tiempo, ya que cuando se toma el pulso lo que se nota la contracción del corazón (sístole), es decir cuando expulsa la sangre hacia el resto del cuerpo.

En las niñas a partir de los 12 años y las mujeres en general suelen tener el pulso más rápido que los niños y los hombres.

En los adultos sanos oscila entre 60 y 100 latidos por minuto. El pulso podría fluctuar y aumentar con el ejercicio, las enfermedades y emociones. [7]

A continuación se detalla como varia el pulso de acuerdo a la edad:

Niños menores a un año: 130 a 140 Pulsaciones por minuto.

Niños: 80 a 100 Pulsaciones por minuto.

Adultos: 72 a 80 Pulsaciones por minuto.

Ancianos: 60 o menos pulsaciones por minuto.

1.7 Derivaciones.

Precordiales: Obtenidas con los electrodos situados sobre el tórax del paciente. Se los denomina V1, V2, V3, V4, V5 y V6.

Bipolares: Las que se obtienen con el llamado triángulo de Einthoven, registran la diferencia de potencial entre dos electrodos situados en extremidades diferentes. Se los denomina D1, D2 y D3.

Triángulo y Ley de Einthoven: Estas derivaciones tienen una proporción matemática, expresada en la Ley de Einthoven que nos dice: $D2=D1+D3$.

D1: Su vector está en dirección a 0° y la diferencia de potencial se registra entre brazo derecho y brazo izquierdo.

D2: Su vector está en dirección a 60° y la diferencia de potencial se registra entre brazo derecho y pierna izquierda.

D3: Su vector está en dirección a 120° y la diferencia de potencial se registra entre brazo izquierdo y pierna izquierda. [8]

En la Figura 1.3. se muestra la interpretación del Triángulo de Einthoven.

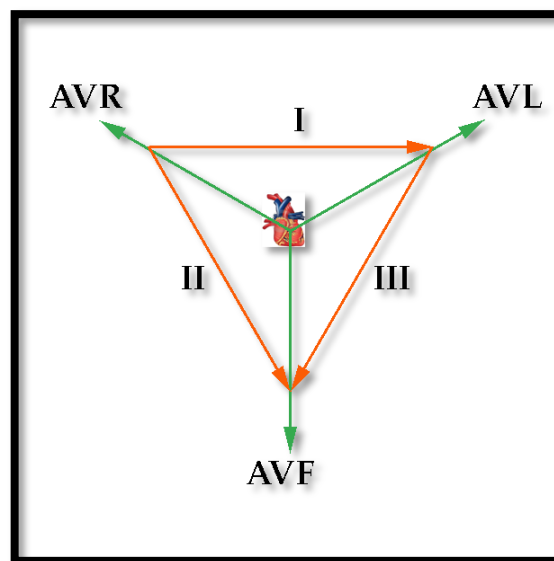


Fig. 1.4 Triángulo de Einthoven.

(Tomado de: <http://www.my-ekg.com/generalidades-ekg/derivaciones-cardiacas.html>)

CAPÍTULO 2

2.1 Descripción del Electrocardiógrafo.

El Electrocardiógrafo obtiene señales cardíacas mediante la utilización de los electrodos de piel. Las entradas del circuito provienen:

- Brazo Derecho.
- Brazo Izquierdo.
- Pierna Derecha (Referencia).

El hardware está formado por tres Tarjetas Electrónicas:

- 1.- Circuito Electrocardiógrafo.

2.- Tarjeta de Adquisición de Datos.

3.- Módulo Bluetooth HC-06.

2.2 Diagrama de Bloques.

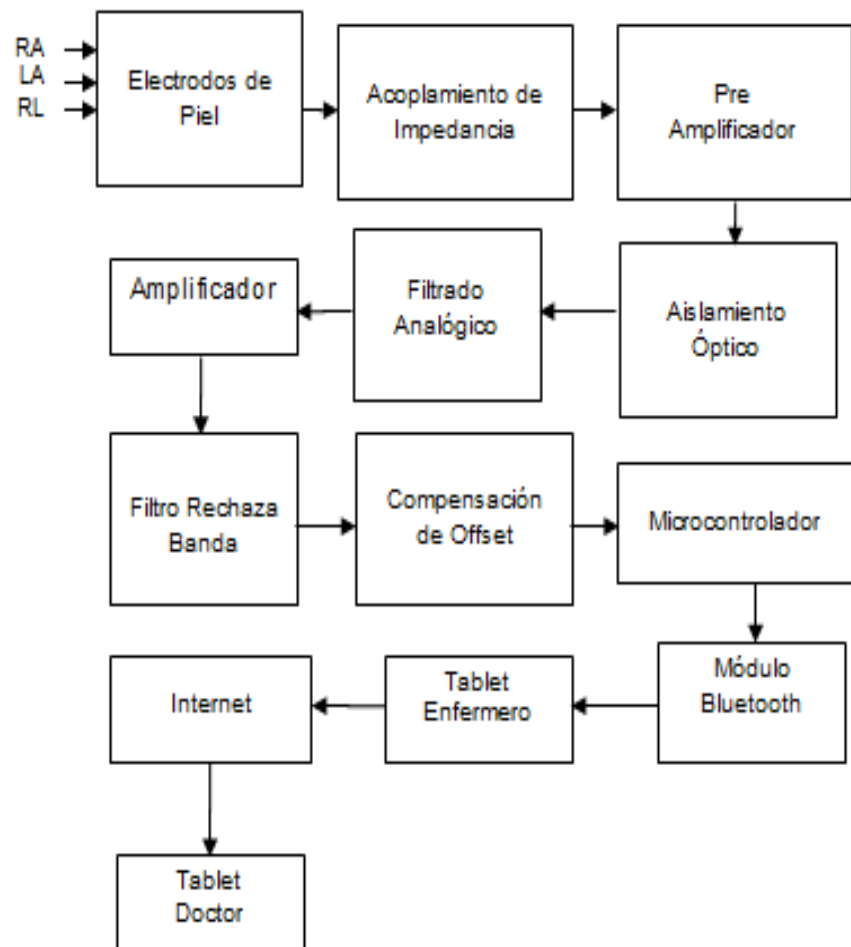


Fig. 2.1 Diagrama de Bloques del Electrocardiógrafo.

2.3 Electrodo de Piel.

Los electrodos son dispositivos que permiten convertir la corriente iónica del cuerpo en corriente eléctrica, para poder analizar el estado cardíaco del paciente.

Los electrodos usados en la adquisición de las señales cardíacas son Electrodo Tipo Parche. (Véase Figura 2.2)

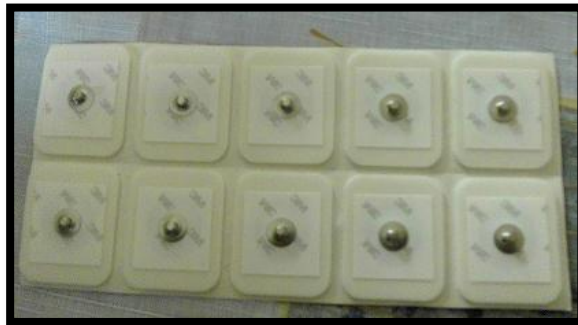


Fig. 2.2 Electrodo Tipo Parche.

2.4 Acoplamiento de Impedancia.

En esta sección lo que se realiza es acoplar la señal de entrada que proviene de los electrodos con el circuito Electrocardiógrafo, utilizando un amplificador operacional que tiene las siguientes características:

- Ancho de Banda muy alto.
- Impedancia de Entrada Alta.
- Impedancia de Salida Nula.

Se configuró el amplificador operacional para que trabaje como un seguidor de voltaje. Observar la Figura 2.3.

Donde se obtiene:

- $E1 = E1'$
- $E2 = E2'$ [9]

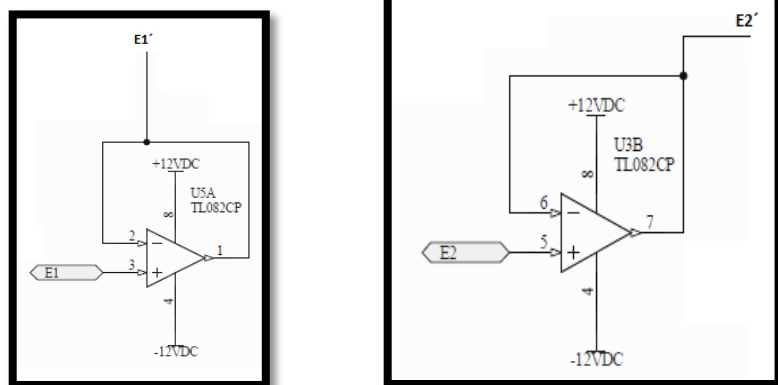


Fig. 2.3 Acoplamiento de Impedancia.

2.5 Pre-amplificador.

En esta etapa se utilizó un amplificador de instrumentación, el cual se puede construir por medio de opamps y resistencias, pero en la realización del proyecto se utilizó el encapsulado AD620AN.

El AD620AN realiza la resta de dos entradas ($E1'$ y $E2'$) y el resultado se multiplica por un factor de ganancia.

La fórmula que relaciona la Ganancia con la R_g es la siguiente:

$$R_g = \frac{49.4 \text{ K}\Omega}{G-1} \quad (2.1)$$

En la construcción de esta etapa se estableció una Ganancia de 100, despejando la fórmula anterior encontramos que R_g es 500Ω . [9]

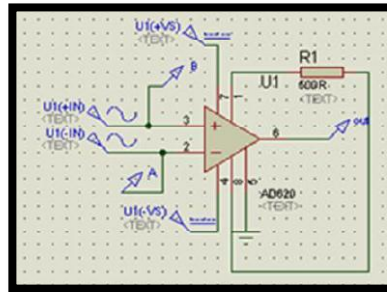


Fig. 2.4 Amplificador Instrumentación.

2.6 Aislamiento Óptico.

En esta etapa se divide en dos circuitos, acoplándose mediante el diodo emisor de luz y éste funcionará alimentándolo con el voltaje requerido, para después activar el circuito que se encuentra a continuación del transistor.

El circuito Opto-Acoplador que se utilizará es el Fototransistor 4N35 y consta de 6 pines. [9]

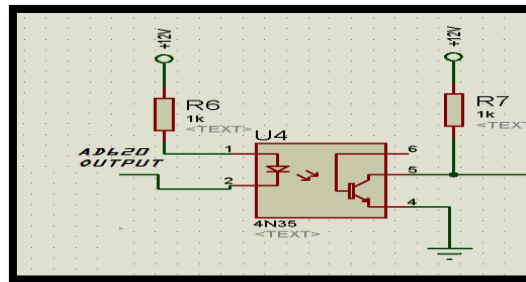


Fig. 2.5 Conexión Etapa Opto-Acoplador.

En la Figura 2.5 se observa que la entrada del Circuito Opto-Acoplador está conectada a la salida del Amplificador de Instrumentación, el cual entrega una ganancia de 100 veces, negativa; el ánodo del diodo va conectado a una resistencia de 1K que se alimenta del voltaje del circuito, lo cual permite limitar la corriente de la fuente.

La otra parte es conectar una resistencia Pull-Up de 1K para polarizar la salida mediante la fuente de voltaje.

2.7 Filtros Analógicos.

En este Electrocardiógrafo se implementará un filtro Pasa-Banda que consta de dos partes: la primera de un Filtro Pasa-Alto con frecuencias de un rango de 0.1 Hz a 1 Hz y la segunda parte por un Filtro Pasa-Bajo con $f_h = 106.103$ Hz. [9]

2.7.1 Filtro Pasa–Alto.

Se utilizará un Filtro Pasa-Alto Sallen-Key Chebyshev de segundo orden que estará localizado después de la ganancia del Pre-Amplificador de Instrumentación con una frecuencia de corte 0,723 Hz.

Se ha seleccionado este filtro debido a que existe una componente DC, que se forma por el offset de los amplificadores operacionales y por la polarización de los electrodos; esto origina que los amplificadores operacionales siguientes se saturen creando un mal funcionamiento del equipo.

La frecuencia de corte seleccionada está dentro del rango que se mencionó anteriormente, esto se realiza con el fin de eliminar el ruido.

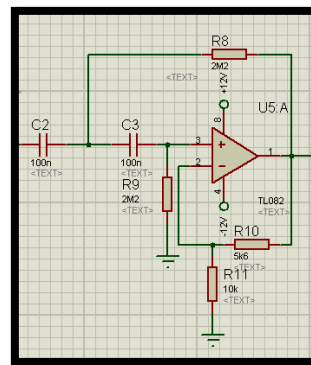


Fig. 2.6 Filtro Pasa–Alto.

La frecuencia de corte inferior se la calcula de la siguiente

forma:

$$C2=C3; R8=R9 \quad (2.2)$$

$$fl = \frac{1}{2\pi\sqrt{C2 C3 R8 R9}} \quad (2.3)$$

$$fl = \frac{1}{2\pi\sqrt{100n * 100n * 2.2M * 2.2M}} \quad (2.4)$$

$$fl = \frac{1}{2\pi(100n)(2.2M)} \quad (2.5)$$

$$fl = 0.723 \text{ Hz} \quad (2.6)$$

La ganancia del filtro Pasa-Alto se calcula de la siguiente

forma:

$$A_v = \frac{R10+R11}{R11} \quad (2.7)$$

$$A_v = \frac{5.6 \text{ K} + 10 \text{ K}}{10 \text{ K}} \quad (2.8)$$

$$A_v = 1.56 \quad (2.9)$$

2.7.2 Filtro Pasa-Bajo.

Se utilizará también un filtro Pasa-Bajo de Sallen Key, fijando una frecuencia de corte en 106,103Hz, esto se lo consigue variando el valor de las resistencias, que nos dará como resultado a que señales superiores a la frecuencia de corte ingresen al circuito Electrocardiógrafo, logrando así no captar señales de ruido.

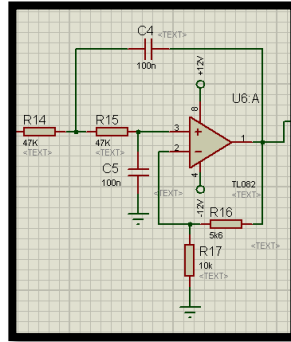


Fig. 2.7 Filtro Pasa-Bajo.

La frecuencia de corte superior se la calcula de la siguiente forma:

$$C4=C5; R14=R15 \quad (2.10)$$

$$f_h = \frac{1}{2\pi\sqrt{C4 C5 R14 R15}} \quad (2.11)$$

$$f_h = \frac{1}{2\pi\sqrt{100n * 100n * 15k * 15k}} \quad (2.12)$$

$$f_h = \frac{1}{2\pi(100n)(15k)} \quad (2.13)$$

$$f_h = 106.103 \text{ Hz}$$

$$(2.14)$$

La ganancia del filtro Pasa-Bajo se calcula de la siguiente forma:

$$A_v = \frac{R17+R16}{R17} \quad (2.15)$$

$$A_v = \frac{5.6 \text{ K} + 10 \text{ K}}{10 \text{ K}} \quad (2.16)$$

$$A_v = 1.56 \quad (2.17)$$

2.7.3 Filtro Rechaza-Banda.

La característica esencial del Filtro Notch es descartar las frecuencias que estén interfiriendo en el circuito; éstas proceden del ruido eléctrico del ambiente, de las lámparas fluorescentes y también de los electrodos.

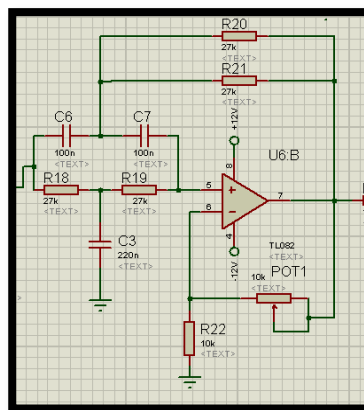


Fig. 2.8 Filtro Rechaza-Banda.

2.8 Amplificador.

Para la parte de amplificación se utilizará un opamp no inversor. Sabiendo que el voltaje del pin negativo es exactamente igual al voltaje del pin positivo, podemos

relacionar el voltaje de entrada con el voltaje de salida a través de un divisor de voltaje.

En este diseño se eligió un ganancia de 85, y a R1 se le dio un valor de 1.2k. Despejando la fórmula $V_o = V_{in}(1 + (R_2/R_1))$ se obtiene que $R_2 = 100k$. [9]

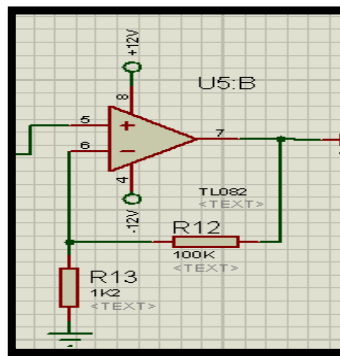


Fig. 2.9 Divisor de Voltaje.

2.9 Compensación de Offset.

La última etapa del circuito Electrocardiógrafo, se implementó la etapa de acoplamiento de la señal. Logrando remover momentáneamente la parte negativa de la señal sin alterarla, solo se necesita elevar el nivel DC de la señal cardíaca, para así utilizar de manera adecuada el microcontrolador, que trabaja solo con valores positivos entre 0 y 5V.

Se utiliza el opamp restador inversor, restando la señal menos un valor negativo, en este caso de $-1.2V$, aumentando así el nivel DC de la señal, y convirtiendo la señal totalmente positiva, para que el microcontrolador trabaje de manera óptima. [9]

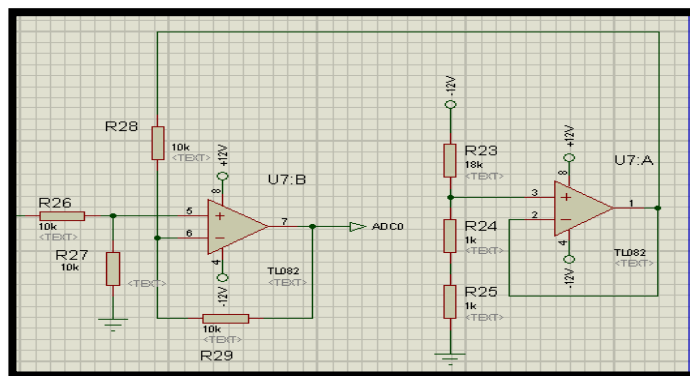


Fig. 2.10 Acondicionamiento de la Señal.

2.10 Tarjeta de Adquisición de Datos.

Es un módulo de entrenamiento y desarrollo que permite realizar múltiples funciones con el microcontrolador 16F886.

Ayudará a recibir la señal del circuito Electrocardiógrafo por intermedio del Microcontrolador (PIC16F886), para luego enviar la señal cardíaca a través del bluetooth.

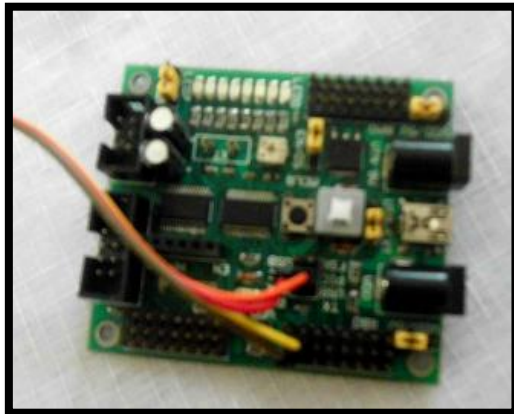


Fig. 2.11 Tarjeta MEI&T04(PIC16F886).

2.11 Módulo Bluetooth HC-06.

Características:

- Protocolo bluetooth: Bluetooth especificación V2.0+EDR
- Frecuencia: 2.4Ghz ISM Band
- Rango de baudios ajustable: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
- Distancia: 10 metros.

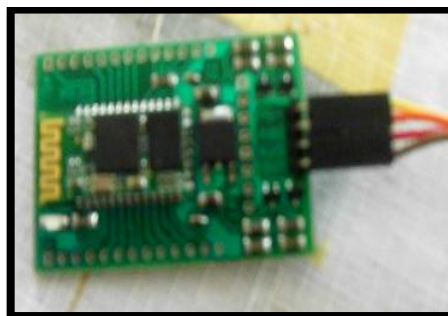


Fig. 2.12 Módulo Bluetooth HC-06.

CAPITULO 3

3. Diseño e implementación del CARDIOFGJ para la Tablet.

3.1 Introducción.

Durante los últimos años el crecimiento de la tecnología ha sido muy notable, cada día aparecen nuevas creaciones que hacen cosas fantásticas, donde lo imposible ahora es posible de solucionar.

Cuando no existía ni Internet ni los teléfonos celulares, era muy difícil comunicarse entre personas que vivían lejos, a no ser por una carta o por una llamada telefónica convencional de larga distancia, lo cual era un proceso complejo.

Ahora con el uso de internet desde casa o por teléfonos inteligentes, se facilita la comunicación con cualquier persona del mundo, a través de redes sociales, correos electrónicos o videollamadas.

Cabe recalcar que son muchas empresas las que están en este entorno creando nuevos programas, aplicaciones, para mejorar nuestras vidas, innovando continuamente. Un ejemplo es Google que adquirió la marca Android y junto con otras grandes compañías como Motorola, LG, Intel, Samsung Electronics, T-Mobile formó la Open Handser Alliance (OHA), con el único fin de crear estándares abiertos para dispositivos móviles y lanzando así el primer sistema operativo Android.

3.2 Lenguaje de Programación JAVA.

JAVA fue creado por la compañía Sun Microsystems Inc, con el fin de crear un lenguaje que pudiera funcionar en redes de computadoras y también que fuera independiente de la plataforma en la que se vaya a ejecutar.

El lenguaje se creó con 5 objetivos principales:

- Programación orientada a objetos.

- Que sea ejecutable en múltiples sistemas operativos.
- Incluir por defecto soporte para trabajo en red.
- Diseñarse para ejecutar código en sistemas remotos de forma segura.
- Fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como por ejemplo C++.

El objetivo principal es hacer que grandes proyectos sean fáciles de manejar y gestionar, dando así una mejor calidad y reduciendo el número de proyectos que no funcionan. [9.1]

3.2.1 Características.

Sun describe al lenguaje Java de la siguiente manera:

- Simple.
- Orientado a Objetos.
- Distribuido.
- Interpretado.
- Robusto.
- Seguro.
- Arquitectura Neutral.
- Alto Rendimiento.
- Dinámico.

Todas estas características sirven para describir a JAVA. Pero hay tres que son las más importantes [10]:

- Portabilidad.
- Arquitectura Neutral.
- Simplicidad.

3.2.2 Tutorial de Java.

Comentarios

En Java hay tres tipos de comentarios:

// comentarios para una sola línea

/* comentarios de una o

más líneas

*/

/** comentario de documentación, de una o más líneas

*/

Identificadores

Los identificadores sirven para nombrar variables, funciones, clases y objetos.

En Java un identificador comienza con una letra, un subrayado (_) o un símbolo de dólar (\$). Los siguientes caracteres

pueden ser letras o dígitos. Se distinguen las mayúsculas de las minúsculas y no hay longitud máxima.

Serían identificadores válidos:

```
nombre_usuario
```

```
Nombre_Usuario
```

```
_variable_del_sistema
```

```
$transaccion
```

y su uso sería, por ejemplo:

```
int contador_principal;
```

```
char _lista_de_ficheros;
```

```
float $cantidad_en_Ptas;
```

Arrays

En Java un array es realmente un objeto, porque tiene redefinido el operador [].

Para crear un array en Java hay dos métodos básicos. Crear un array vacío:

```
int lista[] = new int[30];
```

o se puede crear ya el array con sus valores iniciales:

```
String apellidos[] = {  
    "Icaza", "Verdezoto", "Rodriguez"  
};
```

Esto es equivalente a:

```
String nombres[];  
nombres = new String[3];  
nombres[0] = new String( "Icaza" );  
nombres[1] = new String( "Verdezoto" );  
nombres[2] = new String( "Rodriguez" );
```

Todos los arrays en Java son estáticos.

Operadores

Los operadores de Java son parecidos a los de C.

Los operadores numéricos se comportan como esperamos:

```
int + int = int
```

Para las cadenas, se pueden utilizar los operadores relacionales para comparaciones además de + y += para la concatenación:

```
String nombre = "nombre" + "Apellido";
```

El operador = siempre hace copias de objetos, marcando los antiguos para borrarlos. [11]

3.3 Sistema Operativo Android.

Android es un sistema operativo de Linux que se ha diseñado para dispositivos móviles con pantalla táctil, como son los teléfonos inteligentes o tabletas. Android fue mostrado en el año 2007 por intermedio de la fundación Open Handset Alliance. El primer móvil con el sistema operativo Android fue el HTC Dream, el cual se vendió en octubre de 2008. [12]

Su logo es un robot llamado "Andy" que fue diseñado por Irina Blok, una persona que ha trabajado para Google, HP y Yahoo. [13]



Fig. 3.1 Logo de Android.

3.3.1 Historia.

Fue desarrollado en sus inicios por Android Inc., que luego fue comprada por Google en el año 2005. El Sistema Android cuenta

con una gran cantidad de desarrolladores que han alcanzado 1.000.000 de aplicaciones (donde dos tercios son gratuitos). Los programas están escritos en el lenguaje de programación Java.

El sistema Android fue anunciado junto con la creación de la Open Handset Alliance, el 5 de noviembre de 2007, esta alianza consta de 78 compañías de hardware, software y telecomunicaciones. [14]

3.3.2 Versiones de Android.

- | | |
|------------------------------------|------------------------|
| - Android 1.0 (Apple Pie) | 23 septiembre del 2008 |
| - Android 1.1 (Banana Bread) | 9 febrero del 2009 |
| - Android 1.5 (Cupcake) | 30 abril del 2009 |
| - Android 1.6 (Donut) | 15 septiembre del 2009 |
| - Android 2.0 /2.1 (Eclair) | 26 octubre del 2009 |
| - Android 2.2 (Froyo) | 20 mayo del 2010 |
| - Android 2.3 (Gingerbread) | 06 diciembre del 2010 |
| - Android 3.0/3.1/3.2 (Honeycomb) | mediados del 2011 |
| - Android 4.0 (Ice Cream Sandwich) | noviembre del 2011 |
| - Android 4.1 (Jelly Bean) | junio del 2012 |
| - Android 4.2 (Jelly Bean) | noviembre del 2012 |
| - Android 4.3. (Jelly Bean) | 24 julio del 2013 [15] |

3.3.3 Estructura de un proyecto Android.

Cuando se construye un proyecto nuevo de Android en Eclipse se genera automáticamente la estructura de carpetas que son necesarias para generar la respectiva aplicación, esto se lo puedo observar en la figura 3.2 que se presenta a continuación. Este esquema o estructura será común en cualquier aplicación, independiente de su tamaño y complejidad.

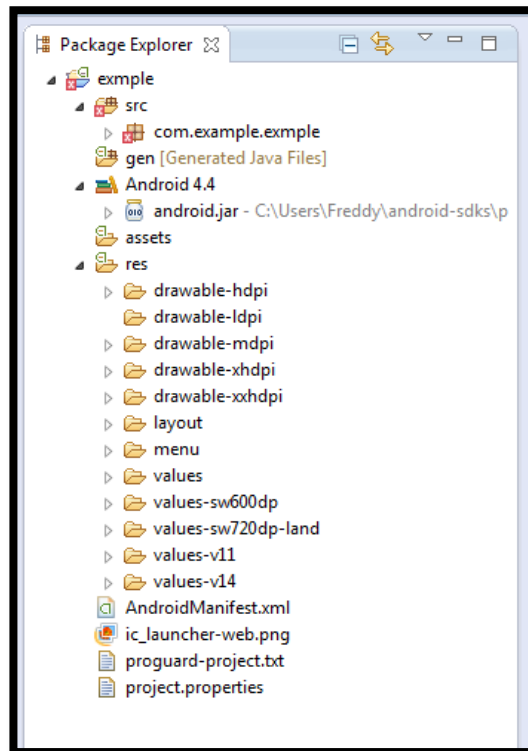


Fig. 3.2 Estructura de un Proyecto Android.

3.4 Kit de desarrollo de Software SDK Android.

El kit de desarrollo de software o SDK (Software Development Kit) es un conjunto de herramientas que permite al programador crear aplicaciones para un sistema específico. Los SDK incluyen códigos de ejemplos, notas técnicas u otras ayudas para aclarar ciertas dudas de programación.

Para desarrollar aplicaciones en Android no es necesario saber lenguajes complejos de programación. Lo único necesario es tener los conceptos básicos de JAVA y tener el kit de desarrollo de software para Android, el cual se lo puede descargar gratuitamente.

Cada nueva versión que aparece de Android, también se crea una nueva versión de SDK para android, en este proyecto se utiliza el SDK 4.1 de Android ya que el sistema operativo de las Tablets es Android 4.1.

3.5 Instalación del SDK.

Al comenzar a desarrollar una aplicación se necesita tener instalado el entorno de desarrollo integrado Eclipse y el SDK 4.1 de Android, para configurar el SDK 4.1 de Android se necesita instalar el paquete SDK de arranque, que lo encontramos en la

página de desarrolladores de Android, una vez instalado el paquete, procedemos a instalar el plug-in para Eclipse ADT.

Es importante comprobar que esté instalado el JDK, una vez que se compruebe esto, se procede a descargar el motor de arranque paquete SDK de la página de descarga del SDK, se descomprime el archivo.zip en un lugar específico del computador.

Se anota el nombre y la ubicación del directorio de SDK en el sistema, debido a que tenemos que hacer referencia al directorio SDK más tarde.

Android nos ofrece un plug-in para el IDE de Eclipse, ADT, así podemos configurar de manera rápida nuevos proyectos de Android, crear una aplicación de interfaz de usuario, depurar las aplicaciones utilizando las herramientas SDK de Android.

Para poder descargar el ADT, nos dirigimos a Star Eclipse, Help -> Install New Software. Damos click en Add y en el dialogo que aparece se pone ADT Plugin en el Nombre y en Location la URL.

3.6 Requerimiento de la Aplicación.

Como se ha indicado en este capítulo Android es un sistema operativo para dispositivos móviles de última tecnología como son el caso de los teléfonos inteligentes, tabletas, Google TV; con lo cual podemos decir que los requerimientos de una aplicación deben ser muy exigentes.

Las Tabletas para el cual se desarrolló la aplicación es SMART TABLET PC.



Fig. 3.3 Tablet Smart Pc

(Tomado de:

https://www.google.com.ec/search?q=SMART+TABLET+PC&source=lnms&tbn=isch&sa=X&ei=Z_h_UoTBOJCekAfB3oGQDQ&ved=0CAcQ_AUoAQ&biw=1366&bih=667#facrc=_&imgdii=_&imgrc=L_eDrnko49A3oM%3A%3BwOPmJRybrOOdPM%3Bhttp%253A%252F%252Fimage.made-in-china.com%252F2f0j00AKdTINmCyGqf%252F7-Inch-Multi-Touch-Android-4-0-Cortex-A9-Wi-Fi-Smart-Tablet-PC-LY-W701-.jpg%3Bhttp%253A%252F%252Flongnuo88.en.made-in-china.com%252Fproduct%252FFPKgJrJjOvncx%252FChina-7-Inch-Multi-Touch-Android-4-0-Cortex-A9-Wi-Fi-Smart-Tablet-PC-LY-W701-.html%3B701%3B413)

La Tablet presenta las siguientes características:

- Sistema Operativo Android 4.1.
- Memoria Ram 512 MB.
- Pantalla de 7 pulgadas.
- Bluetooth.
- Conexión Wifi.

Las características descritas son las más importantes para el desarrollo del proyecto, también es importante indicar que se necesitará crear un correo electrónico gmail, que se lo realiza desde la propia Tablet, conectándose a Internet, por la red Wifi.

3.7 Desarrollo de la Aplicación.

La aplicación CardioFGJ tiene el objetivo principal de visualizar las señales cardíacas del corazón, como también la información básica del paciente, consta de dos modos de funcionamiento, el modo enfermero y el modo doctor.

3.8 Menú Enfermero.

3.8.1 Diagrama de Bloques.

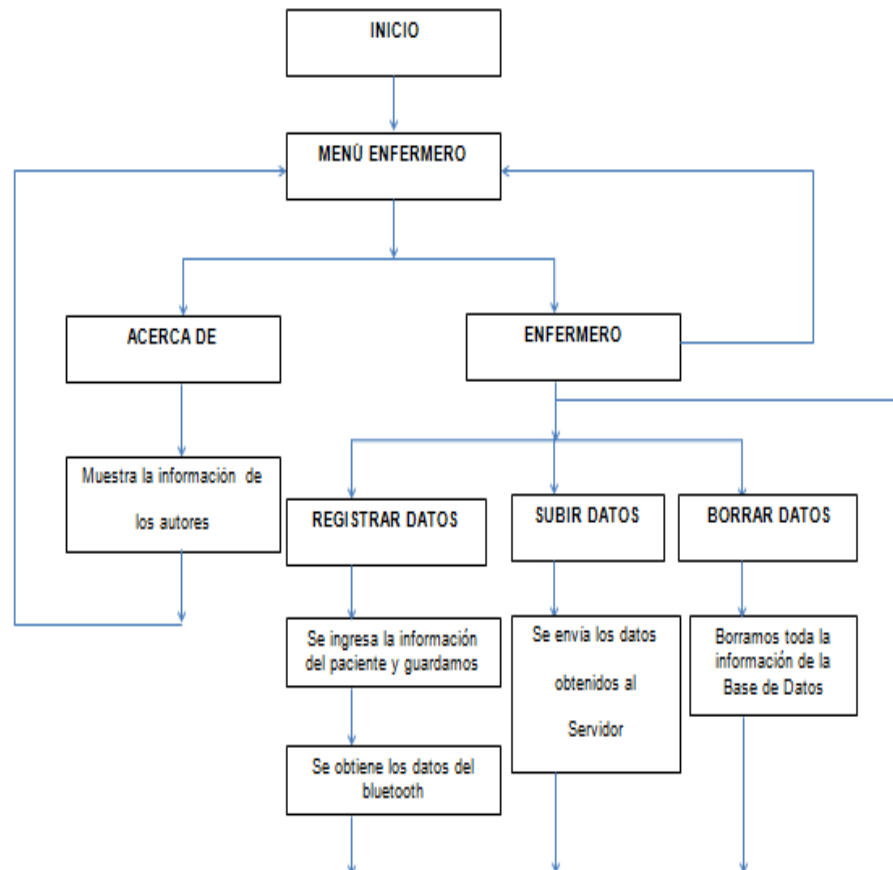


Fig. 3.4 Diagrama de Bloque Aplicación Enfermero.

3.8.2 Explicación detallada del procedimiento.

Cuando se abre la aplicación CARDIOFGJ Enfermero se muestra dos opciones: Botón ENFERMERO y Botón Acerca de.



Fig. 3.5 Aplicación Enfermero.

Al dar clic en el botón Acerca de aparece la información de los autores del proyecto y su respectivo número telefónico de contacto.



Fig. 3.6 Información Autores.

Al dar clic en Ok se regresa al Menú Principal del Enfermero.



Fig. 3.7 Pantalla Enfermero.

Al dar clic en el botón Enfermero aparece un nuevo menú con tres opciones:

- 1.- Registrar Datos.
- 2.- Subir Datos.
- 3.- Borrar Datos.



Fig. 3.8 Menú Enfermero.

Al hacer clic en el botón Ingresar Datos aparece la siguiente ventana, cada paciente tiene como opción a dos exámenes:

The image shows a mobile application interface for 'CardioFGJ'. It features a list of input fields for patient information: 'Nombre:', 'Apellidos:', 'Sexo:', 'Cédula:', 'Teléfono:', 'Dirección:', and 'Correo:'. Each field is followed by a horizontal line for text entry. Below these fields is a grey button labeled 'GUARDAR'. The background is a light blue gradient.

Fig. 3.9 Ingreso de Datos.

Se ingresa la información básica del paciente:

- Nombres.
- Apellidos.
- Sexo.
- Cédula..
- Teléfono.
- Dirección.
- Correo Electrónico.

Y posteriormente al dar clic en guardar, aparece el siguiente mensaje, esperando dar clic en Yes, para comenzar adquirir los datos del corazón a través del Módulo Bluetooth.

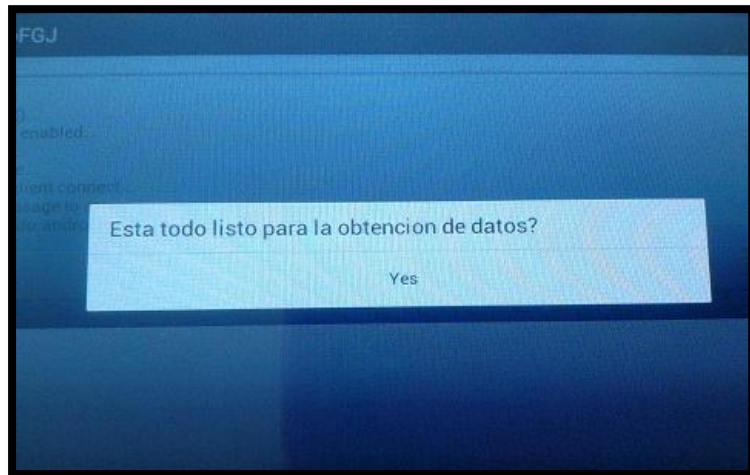


Fig. 3.10 Inicio obtención de Datos.

Una vez que se cumpla el período de adquisición de datos aparece la siguiente ventana donde se muestran los datos adquiridos, esperando que el Enfermero presione Ver Gráfico.



Fig. 3.11 Ver Gráfico.

Al dar clic en ver gráfico aparece una nueva ventana donde se muestra los Datos obtenidos de la Derivación D1.



Fig. 3.12 Gráfico D1 (Derivación Bipolar).

Para subir los datos obtenidos al Servidor, regresamos al Menú Enfermero y se da clic en el Botón Subir Datos.

3.9 Menú Doctor.

3.9.1 Diagrama de Bloques.

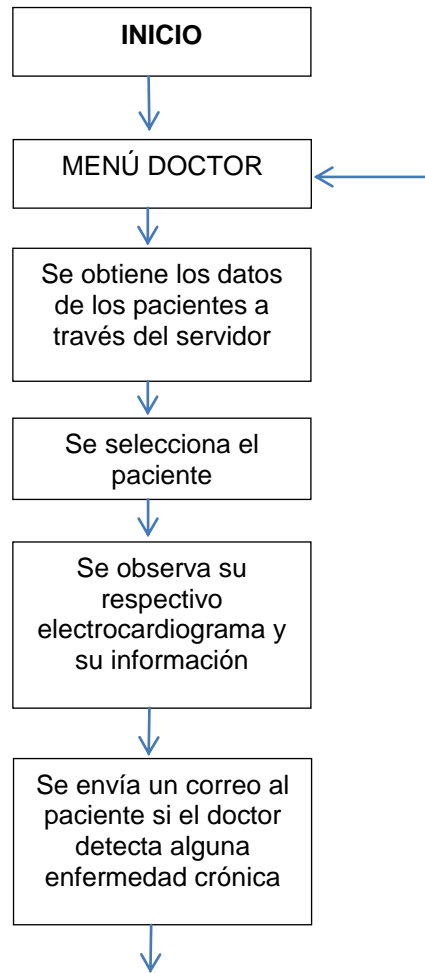


Fig. 3.13 Diagrama de Bloques Aplicación Doctor.

3.9.2 Explicación detallada del procedimiento.

El doctor descarga la información del paciente, analiza y emite el resultado respectivo por correo electrónico.



Fig. 3.14 Gráfico Señal Cardíaca.

CAPÍTULO 4

4. Análisis y funcionamiento del Equipo.

4.1 Descripción del Proyecto.

Este circuito Electrocardiógrafo es capaz de captar la señales cardíacas, por intermedio del módulo Bluetooth, que se comunicará con la aplicación del Enfermero para luego ser enviado dicho resultado a la Tablet del Doctor, para su debido diagnóstico.

El costo del proyecto es muy económico, su implementación se realizó inicialmente en protoboard realizando las respectivas pruebas, para posterior hacerlo en impreso.

En la figura 4.1 se muestra el equipo desarrollado.



Fig. 4.1 a) Electrocardiógrafo en protoboard.

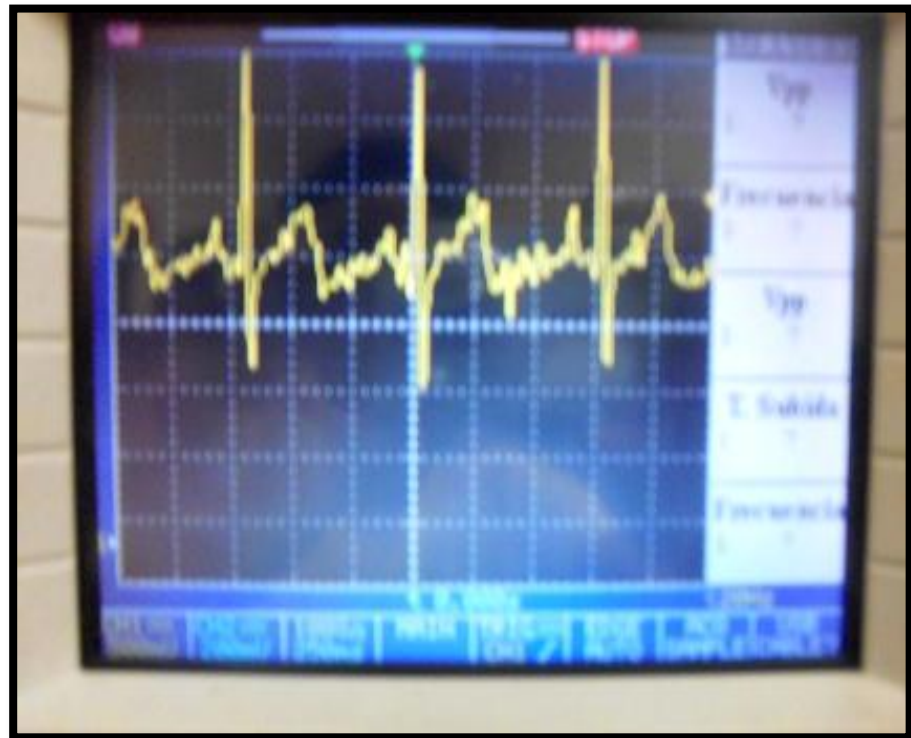


Fig. 4.1 b) Electrocardiograma en Osciloscopio Virtual.

4.2 Materiales.

Se utilizó la siguiente lista de materiales:

Tabla I Lista de Materiales

Cantidad	Descripción	Valor Unitario	Valor Total
2	Resistencia 2.2 K	0.05	0.10
1	Resistencia 1 M	0.05	0.05
8	Resistencia 10 K	0.05	0.40
5	Resistencia 1 K	0.05	0.25
2	Resistencia 2.2 M	0.05	0.10
1	Resistencia 510	0.05	0.05
1	Resistencia 100 K	0.05	0.05
2	Resistencia 47 K	0.05	0.10
4	Resistencia 27 K	0.05	0.20
1	Resistencia 18 K	0.05	0.05
1	Potenciómetro 10 K	0.30	0.30
9	Capacitor 100 nF	0.05	0.45
2	Capacitor 47 uF	0.05	0.10
1	Resistencia 1.2 K	0.03	0.03
2	Resistencia 5.6 K	0.03	0.06
1	AD620AN	12.00	12.00

1	Regulador de Voltaje +12V, -12V	30.00	30.00
1	Módulo Bluetooth HC-06	40.00	40.00
1	Tarjeta de Adquisición de Datos	70.00	70.00
5	CI TL082	0.20	1.00
1	CI 4N35	0.25	0.25
1	PIC 16F885	12.00	12.00
3	Electrodos	0.60	1.80
3	Cables	-	-
1	Tarjeta Impresa	30.00	30.00
2	Tablets Smart Pc	200.00	400.00

El valor final del proyecto es \$599,34, los cables tienen un costo de \$0, porque fueron prestados por el Laboratorio de Electrónica Medica.

4.3 Manejo del Equipo.

4.3.1 Equipo a utilizar.

El equipo a utilizar se detallan a continuación:

- Silla o Cama.
- Electrocardiógrafo.
- Electrodo y cables.
- Material conductor: gel o alcohol.
- Tarjeta de Adquisición de Datos.
- Módulo Bluetooth HC-06.
- 2 Tablets.

4.3.2 Procedimiento.

Para la realización de un electrocardiograma es necesario:

- El paciente debe estar relajado.
- El paciente se debe quitar todo objeto metálico (por ejemplo pulseras, anillos, etc.)
- Se debe sentar en una silla o acostar en una cama.
- Se debe aplicar gel conductor en la superficie del electrodo.
- Se coloca los tres electrodos en los brazos y en la pierna derecha que se utilizará como referencia.
- Se debe colocar los cables en los electrodos:

- ✓ Cable E1 en Brazo Izquierdo.
 - ✓ Cable E2 en Brazo Derecho.
 - ✓ Cable Ref. en Pierna Derecha.
- Conectar todo el equipo y encenderlo hasta obtener la señal cardíaca del paciente en la tablet del enfermero.

Una vez conectado todo el equipo se obtuvieron los siguientes resultados.

Señal Bipolar (D1)



Fig. 4.2 Señal Cardíaca en la Tablet

Con esto se demuestra el correcto funcionamiento del proyecto, los resultados obtenidos fueron los anhelados.

CONCLUSIONES

Con la realización de este proyecto, se ha obtenido un gran resultado, una señal electrocardiográfica muy buena, con lo que se comprueba que el desarrollo de la aplicación funciona correctamente.

El proyecto tiene un costo aceptable y su uso es muy sencillo, utilizando la tecnología de las Tablets.

Concluido el proyecto se pudo observar que la señal cardíaca del cuerpo es muy pequeña, por eso se utilizó un pre-amplificador y un amplificador, que nos ayuda a aumentar la amplitud.

Es importante tener un alto CMRR para lograr eliminar la mayor cantidad de ruido, y así visualizar una imagen óptima.

El Electrocardiógrafo trabaja a una frecuencia correcta que está dentro del rango 0.1 Hz hasta 150 Hz.

Para obtener una señal más pura se utilizaron filtros específicos para eliminar el ruido que se presenta en el circuito.

Antes de enviar la señal, se la pasa por una etapa que se conoce como compensación offset; esto es necesario para evitar que se dañe el circuito por sobrevoltaje.

Tenemos que recalcar que al microcontrolador no le pueden llegar voltajes negativos, por eso fue necesario aumentar un voltaje offset de 2V con una configuración sumador no inversor.

Con el microcontrolador se hace la adaptación de la señal analógica convirtiéndola en digital para enviarla a través del Bluetooth hasta la aplicación del Enfermero.

El enfermero se encargará de ingresar los datos del paciente y obtener la señal cardíaca, para luego subirla hasta el servidor.

El doctor se encargará de descargar los datos enviados por el enfermero, analizando la señal cardíaca y emitiendo el debido informe a través de correo electrónico.

RECOMENDACIONES

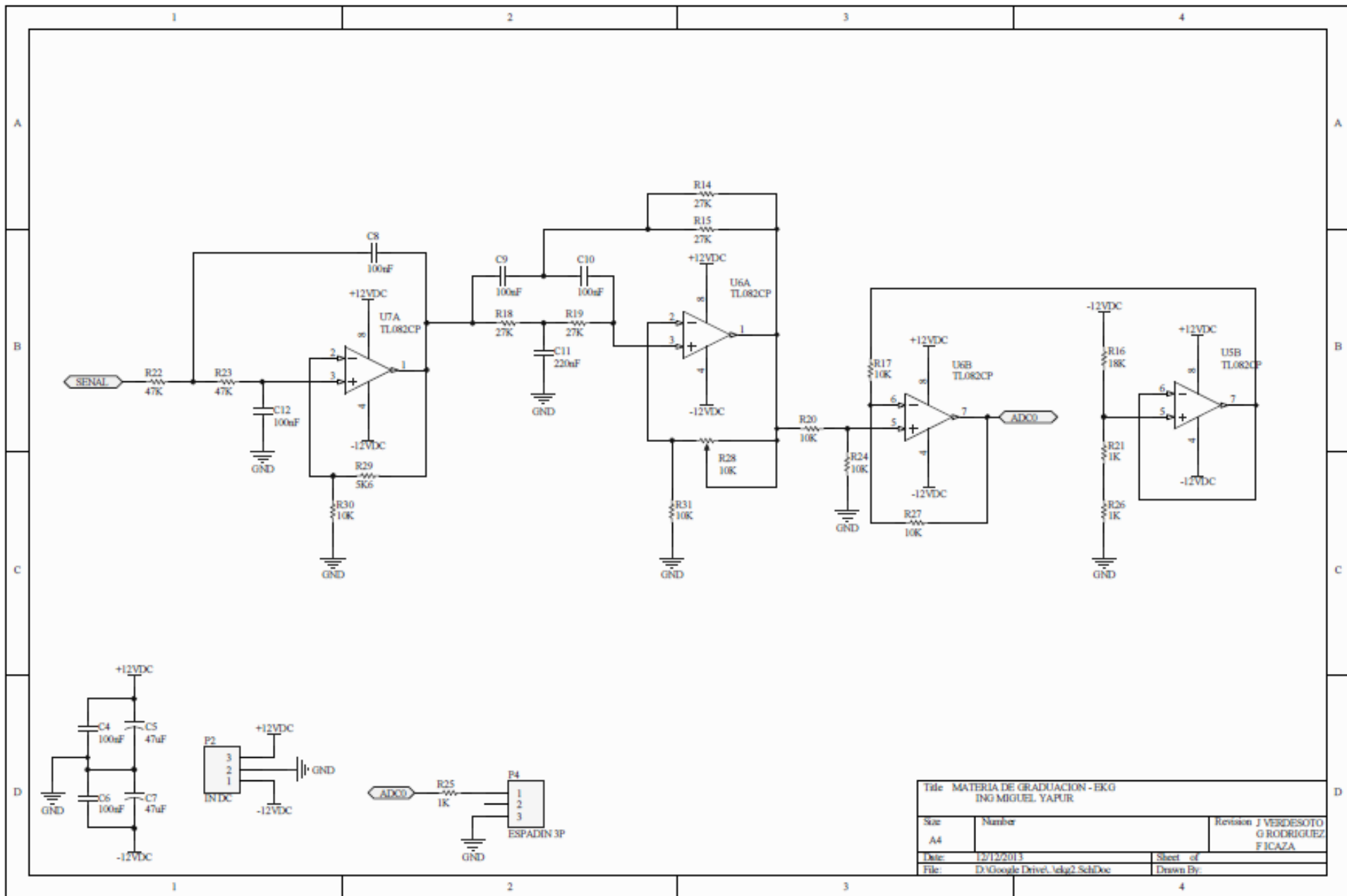
Hay que tener en cuenta las siguientes recomendaciones:

- 1.- El diseño de los filtros debe ser correcto, para evitar que señales no deseadas ingresan al circuito.
- 2.-El uso de los electrodos es muy importante, porque de ellos depende que la señal sea más confiable.
- 3.- El paciente durante todo el proceso debe estar relajado.
- 4.- Se debe quitar todo objeto metálico, ya que son los principales causantes del ruido.
- 5.- Se deben apagar los celulares que estén cerca, porque son causante de interferencia.
- 6.- El paciente debe estar con ropa cómoda, aunque es más aconsejable que se ponga una bata.

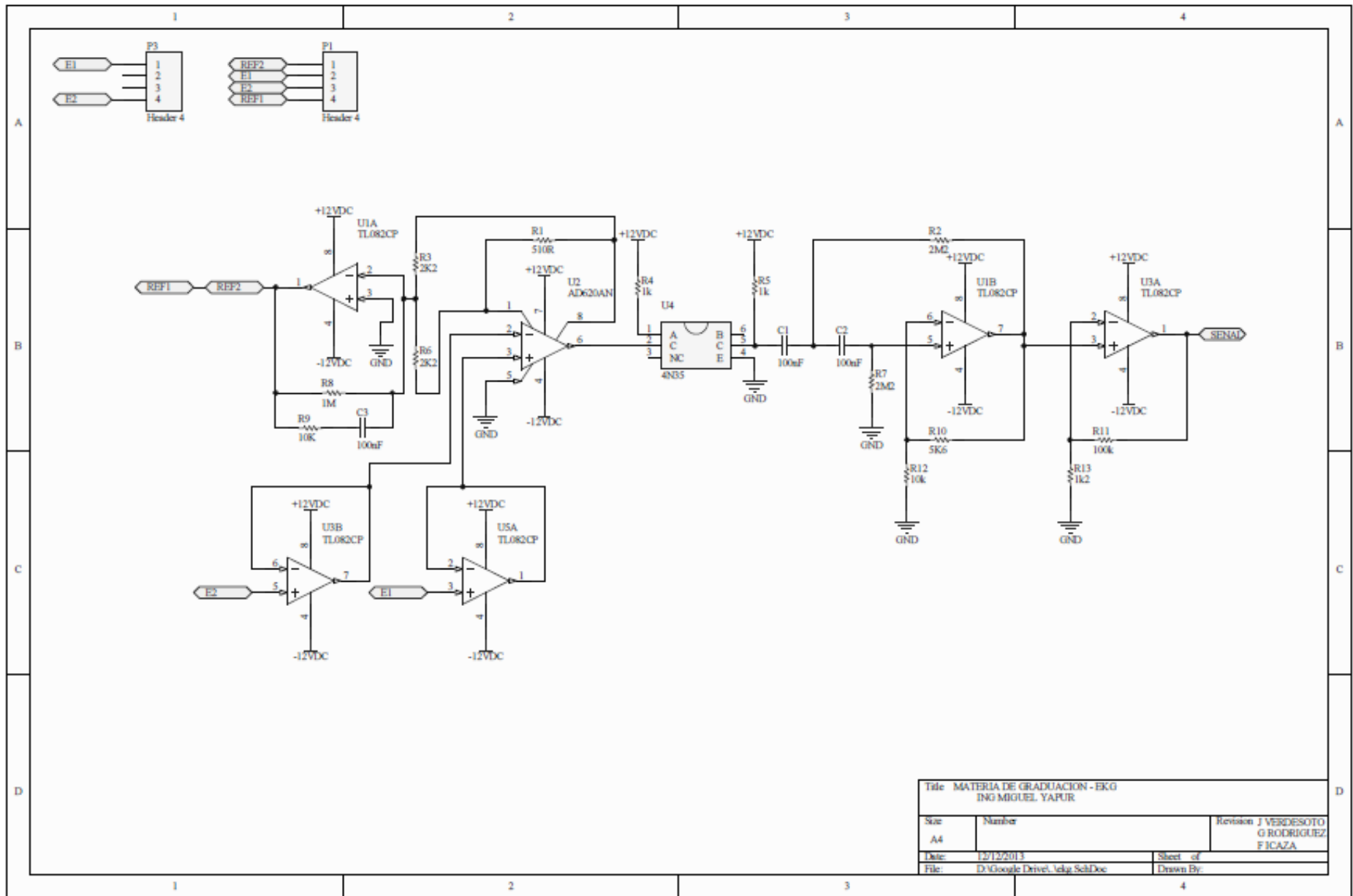
7.- Internet debe estar en una señal óptima para enviar los datos al servidor.

APÉNDICE

APÉNDICE A
ESQUEMÁTICO



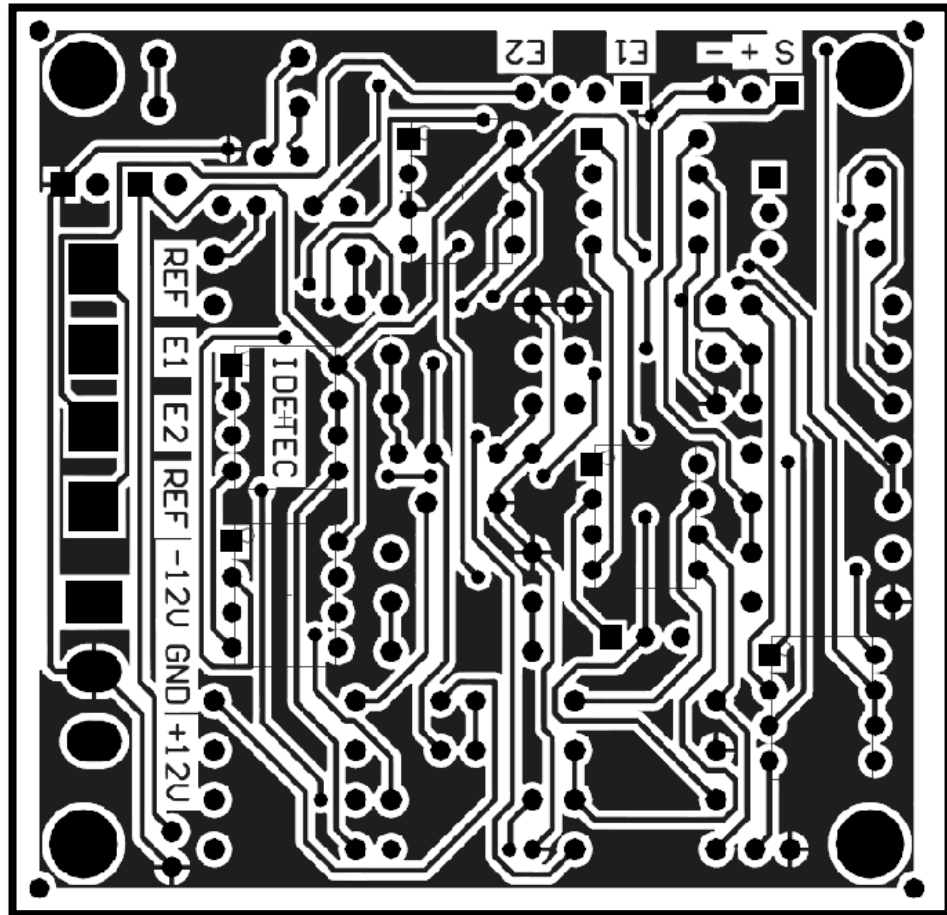
Title: MATERIA DE GRADUACION - EKG ING MIGUEL YAPUR		
Size: A4	Number:	Revision: J VERDESOTO G RODRIGUEZ FICAZA
Date: 12/12/2013	Sheet: of	Drawn By:
File: D:\Google Drive\1elg2 SchDoc		

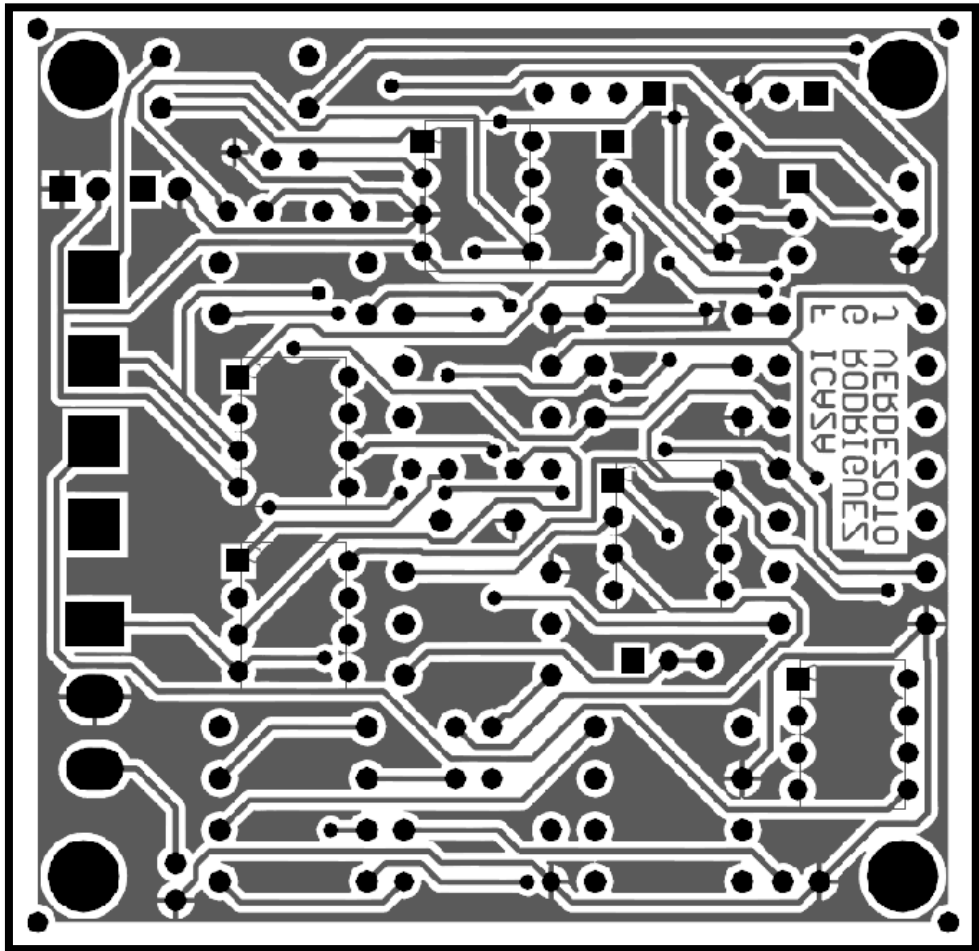


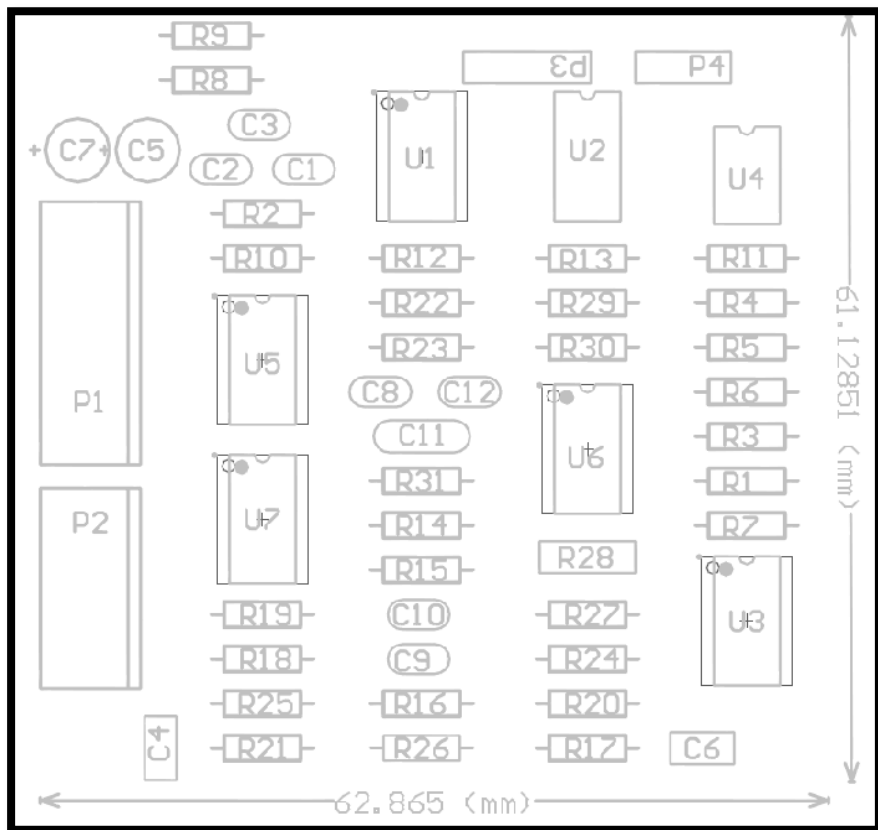
Title: MATERIA DE GRADUACION - ECG ING MIGUEL YAPUR		
Size: A4	Number:	Revision: J VERDESOTO G RODRIGUEZ F ICAZA
Date: 12/12/2013	Sheet of:	
File: D:\Google Drive\... .schDoc	Drawn By:	

APÉNDICE B

DISEÑO DE PISTAS







APÉNDICE C

CÓDIGO FUENTE

```
program maineqg
```

```
dim adcin as word
```

```
dim txt as string[5]
```

```
main:
```

```
    OSCCON = 0x75
```

```
.....
```

```
    TRISA = 0xFF
```

```
.....
```

```
    TRISB = 0xFF
```

```
.....
```

```
    TRISC = 0xBF
```

```
    PORTC = 0X00
```

```
.....
```

```
    ANSEL = 0xFF
```

```
    ANSELH = 0xFF
```

```
.....
```

```
    UART1_Init(9600)
```

```
.....
```

```
    Delay_ms(100)
```

```
.....
```

WHILE 1

 adcin = Adc_Read(1)

.....

 WordToStrWithZeros(adcin, txt)

.....

' UART1_Write("<")

 UART1_Write(txt[0])

 UART1_Write(txt[1])

 UART1_Write(txt[2])

 UART1_Write(txt[3])

 UART1_Write(txt[4])

' UART1_Write(">")

 UART1_Write(0x0D)

 UART1_Write(0x0A)

.....

 Delay_ms(30)

WEND

end.

APÉNDICE D

HOJA DE DATOS

AD620AN



Low Cost Low Power Instrumentation Amplifier

AD620

FEATURES

Easy to use

- Gain set with one external resistor (Gain range 1 to 10,000)
- Wide power supply range (± 2.3 V to ± 18 V)
- Higher performance than 3 op amp IA designs
- Available in 8-lead DIP and SOIC packaging
- Low power, 1.3 mA max supply current

Excellent dc performance (B grade)

- 50 μ V max, input offset voltage
- 0.6 μ V/ $^{\circ}$ C max, input offset drift
- 1.0 nA max, input bias current
- 100 dB min common-mode rejection ratio (G = 10)

Low noise

- 9 nV/ $\sqrt{\text{Hz}}$ @ 1 kHz, input voltage noise
- 0.28 μ V p-p noise (0.1 Hz to 10 Hz)

Excellent ac specifications

- 120 kHz bandwidth (G = 100)
- 15 μ s settling time to 0.01%

APPLICATIONS

- Weigh scales
- ECG and medical instrumentation
- Transducer interface
- Data acquisition systems
- Industrial process controls
- Battery-powered and portable equipment

CONNECTION DIAGRAM

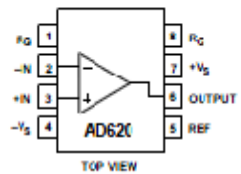


Figure 1. 8-Lead PDIP (N), CERDIP (Q), and SOIC (R) Packages

PRODUCT DESCRIPTION

The AD620 is a low cost, high accuracy instrumentation amplifier that requires only one external resistor to set gains of 1 to 10,000. Furthermore, the AD620 features 8-lead SOIC and DIP packaging that is smaller than discrete designs and offers lower power (only 1.3 mA max supply current), making it a good fit for battery-powered, portable (or remote) applications.

The AD620 with its high accuracy of 40 ppm maximum nonlinearity, low offset voltage of 50 μ V max, and offset drift of 0.5 μ V/ $^{\circ}$ C max, is ideal for use in precision data acquisition systems, such as weigh scales and transducer interfaces. Furthermore, the low noise, low input bias current, and low power of the AD620 make it well suited for medical applications, such as ECG and noninvasive blood pressure monitors.

The low input bias current of 1.0 nA max is made possible with the use of Superbeta processing in the input stage. The AD620 works well as a preamplifier due to its low input voltage noise of 9 nV/ $\sqrt{\text{Hz}}$ at 1 kHz, 0.28 μ V p-p in the 0.1 Hz to 10 Hz band, and 0.1 pA/ $\sqrt{\text{Hz}}$ input current noise. Also, the AD620 is well suited for multiplexed applications with its settling time of 15 μ s to 0.01%, and its cost is low enough to enable designs with one in-amp per channel.

Table 1. Next Generation Upgrades for AD620

Part	Comment
AD8221	Better specs at lower price
AD8222	Dual channel or differential out
AD8226	Low power, wide input range
AD8220	JFET input
AD8228	Best gain accuracy
AD8265	+2 precision op amps or differential out
AD8429	Ultra low noise

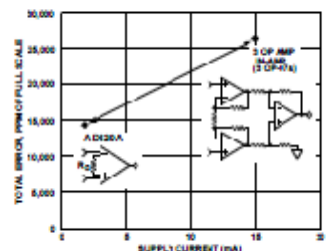


Figure 2. Three Op Amp IA Designs vs. AD620

SPECIFICATIONS

Typical @ 25°C; $V_S = \pm 15$ V and $R_L = 2$ k Ω , unless otherwise noted.

Table 2.

Parameter	Conditions	AD620A			AD620B			AD620S ¹			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
GAIN											
	$G = 1 + 49.4 (k\Omega/R_L)$	1			1			1			
Gain Range		10,000			10,000			10,000			
Gain Error ²	$V_{OUT} = \pm 10$ V										
G = 1		0.03	0.10		0.01	0.02		0.03	0.10		%
G = 10		0.15	0.30		0.10	0.15		0.15	0.30		%
G = 100		0.15	0.30		0.10	0.15		0.15	0.30		%
G = 1000		0.40	0.70		0.25	0.50		0.40	0.70		%
Nonlinearity	$V_{OUT} = -10$ V to $+10$ V										
G = 1-1000	$R_L = 10$ k Ω	10	40		10	40		10	40		ppm
G = 1-100	$R_L = 2$ k Ω	10	95		10	95		10	95		ppm
Gain vs. Temperature	$G = 1$	10			10			10			ppm/°C
	Gain > 1 ¹	-50			-50			-50			ppm/°C
VOLTAGE OFFSET											
(Total RTI Error = $V_{OS} + V_{OS}/G$)											
Input Offset, V_{OS}	$V_S = \pm 5$ V 10 ± 15 V	30	125		15	50		30	125		μ V
Overtemperature	$V_S = \pm 5$ V 10 ± 15 V	185			85			225			μ V
Average TC	$V_S = +5$ V $I_{IN} = 15$ V	0.3	1.0		0.1	0.6		0.3	1.0		μ V/°C
Output Offset, V_{OS}	$V_S = \pm 15$ V	400	1000		200	500		400	1000		μ V
Overtemperature	$V_S = \pm 5$ V 10 ± 15 V	1500			750			1500			μ V
Average TC	$V_S = +5$ V $I_{IN} = 15$ V	5.0	15		2.5	7.0		5.0	15		μ V/°C
Offset Referred to the Input vs. Supply (PSR)	$V_S = \pm 2.5$ V 10 ± 15 V										
G = 1		80	100		80	100		80	100		dB
G = 10		95	120		110	120		95	120		dB
G = 100		110	140		120	140		110	140		dB
G = 1000		110	140		120	140		110	140		dB
INPUT CURRENT											
Input Bias Current		0.5	2.0		0.5	1.0		0.5	2		nA
Overtemperature		2.5			1.5			4			nA
Average TC		5.0			5.0			8.0			pA/°C
Input Offset Current		0.3	1.0		0.3	0.5		0.3	1.0		nA
Overtemperature		1.5			0.75			2.0			nA
Average TC		1.5			1.5			6.0			pA/°C
INPUT											
Input Impedance											
Differential		10 Ω			10 Ω			10 Ω			Ω , pF
Common-Mode		10 Ω			10 Ω			10 Ω			Ω , pF
Input Voltage Range ³	$V_S = \pm 2.5$ V 10 ± 15 V	$-V_S + 1.9$	$+V_S - 1.2$		$-V_S + 1.9$	$+V_S - 1.2$		$-V_S + 1.9$	$+V_S - 1.2$		V
Overtemperature		$-V_S + 2.1$	$+V_S - 1.3$		$-V_S + 2.1$	$+V_S - 1.3$		$-V_S + 2.1$	$+V_S - 1.3$		V
	$V_S = +5$ V 10 ± 15 V	$-V_S + 1.9$	$+V_S - 1.4$		$-V_S + 1.9$	$+V_S - 1.4$		$-V_S + 1.9$	$+V_S - 1.4$		V
Overtemperature		$-V_S + 2.1$	$+V_S - 1.4$		$-V_S + 2.1$	$+V_S - 1.4$		$-V_S + 2.1$	$+V_S - 1.4$		V

AD620

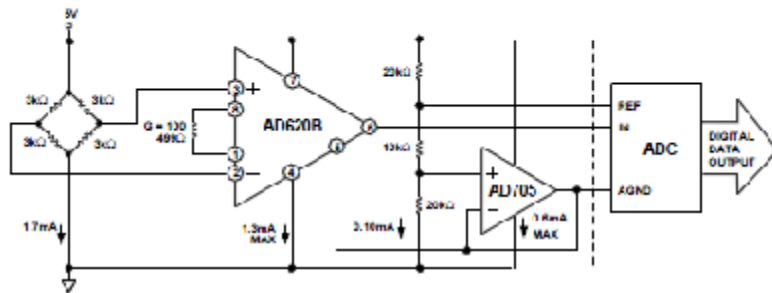


Figure 38. A Pressure Monitor Circuit that Operates on a 5 V Single Supply

Pressure Measurement

Although useful in many bridge applications such as weigh scales, the AD620 is especially suitable for higher resistance pressure sensors powered at lower voltages where small size and low power become more significant.

Figure 38 shows a 5 kΩ pressure transducer bridge powered from 5 V. In such a circuit, the bridge consumes only 1.7 mA. Adding the AD620 and a buffered voltage divider allows the signal to be conditioned for only 3.8 mA of total supply current.

Small size and low cost make the AD620 especially attractive for voltage output pressure transducers. Since it delivers low noise and drift, it also serves applications such as diagnostic noninvasive blood pressure measurement.

Medical ECG

The low current noise of the AD620 allows its use in ECG monitors (Figure 39) where high source resistances of 1 MΩ or higher are not uncommon. The AD620's low power, low supply voltage requirements, and space-saving 8-lead mini-DIP and SOIC package offerings make it an excellent choice for battery-powered data recorders.

Furthermore, the low bias currents and low current noise, coupled with the low voltage noise of the AD620, improve the dynamic range for better performance.

The value of capacitor C1 is chosen to maintain stability of the right leg drive loop. Proper safeguards, such as isolation, must be added to this circuit to protect the patient from possible harm.

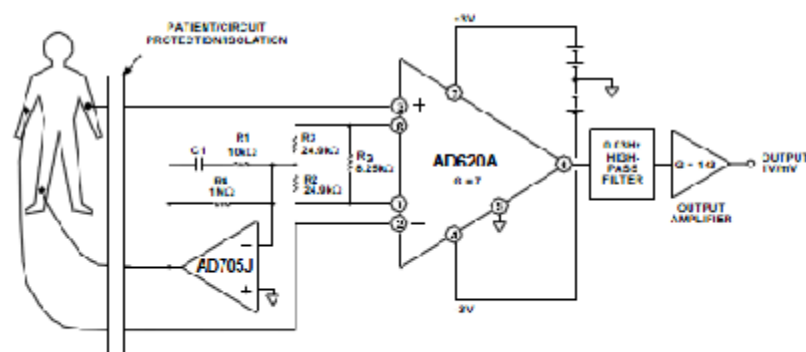


Figure 39. A Medical ECG Monitor Circuit

APÉNDICE E

PROGRAMACIÓN EN JAVA

Menú Doctor

```
package com.fgj.cardiofgj;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.OutputStream;

import java.util.ArrayList;

import java.util.List;

import org.apache.http.NameValuePair;

import org.apache.http.message.BasicNameValuePair;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import com.fgj.cardiofgj.R;

import com.fgj.cardiofgj.lib.Controlador;

import android.annotation.SuppressLint;

import android.annotation.TargetApi;

import android.app.Activity;
```

```
import android.app.AlertDialog;

import android.app.ProgressDialog;

import android.content.ContentValues;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.graphics.Bitmap;

import android.graphics.Bitmap.CompressFormat;

import android.graphics.BitmapFactory;

import android.net.Uri;

import android.os.Build;

import android.os.Bundle;

import android.os.Environment;

import android.os.StrictMode;

import android.provider.MediaStore;

import android.support.v4.widget.SimpleCursorAdapter;

import android.util.Log;

import android.view.KeyEvent;

import android.view.View;

import android.view.View.OnClickListener;
```

```
import android.widget.AdapterView.OnItemClickListener;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.ImageButton;

import android.widget.ImageView;

import android.widget.ListView;

import android.widget.Spinner;

import android.widget.TextView;

import android.widget.Toast;

/*

 * Clase principal para el usuario doctor que tiene como objetivo listar los

pacientes y escoger uno para ver su informacion

 * Autores: Francisco Icaza

 *           Gionella Rodriguez

 *           Jaqueline Verdezoto

 * */

@TargetApi(Build.VERSION_CODES.GINGERBREAD)

@SuppressLint("NewApi")

public class Menu_doctor extends Activity {

    public ProgressDialog dialog1;
```

```

        AdminSQLiteOpenHelper admin = new
AdminSQLiteOpenHelper(this,"repositorio.db", null, 1);

        @SuppressWarnings("NewApi")

Controlador con=new Controlador();

        @Override

        public void onCreate(Bundle savedInstanceState) {

                super.onCreate(savedInstanceState);

                StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();

                StrictMode.setThreadPolicy(policy);

                setContentView(R.layout.activity_menu_doctor);

                final ListView prueba1 = (ListView) findViewById(R.id.list1);

                prueba1.setScrollbarFadingEnabled(true);

                final ArrayList<String> list = new ArrayList<String>();

                SQLiteDatabase bd =admin.getReadableDatabase();

                bd.delete("paciente", null, null);

                bd.delete("muestras", null, null);

                bd.close();

```

```
JSONArray jArray = con.obtener_pacientes();

String pac_id;
String pa_id;
String pa_nombre;
String pa_apellidos;
String pa_sexo;
String pa_cedula;
String pa_telefono;
String pa_direccion;
String pa_correo;

//parse json data
try{
    SQLiteDatabase bd1 = admin.getWritableDatabase();
    for(int i=0;i<jArray.length();i++){
        JSONObject buf = jArray.getJSONObject(i);
        pac_id =buf.getString("pac_id");
        pa_id =buf.getString("pa_id");
        pa_nombre =buf.getString("pa_nombre");
        pa_apellidos =buf.getString("pa_apellidos");
        pa_sexo =buf.getString("pa_sexo");
```

```

pa_cedula      =buf.getString("pa_cedula");
pa_telefono    =buf.getString("pa_telefono");
pa_direccion   =buf.getString("pa_direccion");
pa_correo      =buf.getString("pa_correo");
ContentValues nuevoRegistro = new ContentValues();
    nuevoRegistro.put("pac_id", pac_id);
    nuevoRegistro.put("pa_id", pa_id);
    nuevoRegistro.put("pa_nombre", pa_nombre);
    nuevoRegistro.put("pa_apellidos", pa_apellidos);
    nuevoRegistro.put("pa_sexo", pa_sexo);
    nuevoRegistro.put("pa_cedula", pa_cedula);
    nuevoRegistro.put("pa_telefono", pa_telefono);
    nuevoRegistro.put("pa_direccion", pa_direccion);
    nuevoRegistro.put("pa_correo", pa_correo);
    bd1.insert("paciente", null, nuevoRegistro);
    list.add(buf.getString("pa_nombre") + " " +
buf.getString("pa_apellidos"));
        String mensaje = "PACIENTE
#" +buf.getString("pa_nombre")+ " "+buf.getString("pa_apellidos")+
RECIBIDO";
        Log.i("MENSAJE DEL SERVER:", mensaje);
    }

```

```
        bd1.close();
    }
    catch(JSONException e){
        Log.e("log_tag", "Error parsing data "+e.toString());
    }
}
```

```
String mu_id;
```

```
String fk_paciente;
```

```
String mu_tipo;
```

```
String mu_y_voltaje;
```

```
String mu_x_tiempo;
```

```
String mu_estado;
```

```
JSONArray jArray2 = con.obtener_muestras();
```

```
//parse json data
```

```
try{
```

```
    SQLiteDatabase bd2 = admin.getWritableDatabase();
```

```
    for(int i=0;i<jArray2.length();i++){
```

```
        JSONObject buf2 = jArray2.getJSONObject(i);
```

```
        mu_id =buf2.getString("mu_id");
```

```
        fk_paciente =buf2.getString("fk_paciente");
```

```
        mu_tipo =buf2.getString("mu_tipo");
```

```
        mu_y_voltaje =buf2.getString("mu_y_voltaje");
```



```

mu_x_tiempo          =buf2.getString("mu_x_tiempo");
mu_estado            =buf2.getString("mu_estado");

ContentValues nuevoRegistro2 = new ContentValues();
    nuevoRegistro2.put("mu_id", mu_id);
    nuevoRegistro2.put("fk_paciente", fk_paciente);
    nuevoRegistro2.put("mu_tipo", mu_tipo);
    nuevoRegistro2.put("mu_y_voltaje", mu_y_voltaje);
    nuevoRegistro2.put("mu_x_tiempo", mu_x_tiempo);
    nuevoRegistro2.put("mu_estado", mu_estado);
    bd2.insert("muestras", null, nuevoRegistro2);

    String mensaje = "PACIENTE
#" +buf2.getString("fk_paciente")+ " "+buf2.getString("mu_tipo")+ " RECIBIDO";

    Log.i("MENSAJE DEL SERVER:", mensaje);
}

bd2.close();
}

catch(JSONException e){
    Log.e("log_tag", "Error parsing data "+e.toString());
}

////////////////////////////////////

```

```

        final ArrayAdapter adapter = new
ArrayAdapter(this, android.R.layout.simple_list_item_1, list);

        prueba1.setAdapter(adapter);

        //Al escoger un item de la lista abra el visor de los resultados
        prueba1.setOnItemClickListener(new
AdapterView.OnItemClickListener() {

                @Override

                public void onItemClick(AdapterView<?> parent, final
View view,

                int position, long id) {

                        final String item = (String)
parent.getItemAtPosition(position);

                        int newPos0 = parent.getSelectedItemPosition() + 1;

                                Bundle bundle = new Bundle();

                                bundle.putString("nombres", "" + item);

                                        Intent ir_ob = new
Intent(Menu_doctor.this, TouchZoomExampleActivity2.class);

                                                startActivity(ir_ob.putExtras(bundle));

                }

```

```
        });  
    }  
    @Override  
    public boolean onKeyDown(int keyCode, KeyEvent event) {  
        // TODO Auto-generated method stub  
  
        switch(keyCode){  
            case KeyEvent.KEYCODE_BACK:  
                finish();  
                return true;  
        }  
        return super.onKeyDown(keyCode, event);  
    }  
}
```

Menú Enfermero

```
import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.FileWriter;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.util.ArrayList;

import org.apache.poi.hssf.record.SSTRecord;

import org.json.JSONException;

import org.json.JSONObject;

import com.fgj.cardiofgj.R;

import com.fgj.cardiofgj.lib.Controlador;

import android.net.Uri;

import android.os.AsyncTask;

import android.os.Build;

import android.os.Bundle;

import android.os.Environment;

import android.os.StrictMode;

import android.annotation.SuppressLint;
```

```
import android.annotation.TargetApi;

import android.app.Activity;

import android.app.AlertDialog;

import android.content.DialogInterface;

import android.app.ProgressDialog;

import android.content.Intent;

import android.database.Cursor;

import android.database.SQLException;

import android.database.sqlite.SQLiteDatabase;

import android.util.Log;

import android.view.Menu;

import android.view.View;

import android.widget.AdapterView;

import android.widget.Button;

import android.widget.Toast;

import au.com.bytecode.opencsv.CSVWriter;

/*

 * Clase principal para el usuario enfermero como es el registro, borrado, y

envio de datos recolectados

 * Autores: Francisco Icaza

 *           Gionella Rodriguez

 *           Jaqueline Verdezoto
```

```

* */

@TargetApi(Build.VERSION_CODES.GINGERBREAD)
@SuppressLint("NewApi")
public class Menu_enfermero extends Activity {

    public String paciente_elegido="";

    AdminSQLiteOpenHelper admin = new
AdminSQLiteOpenHelper(this,"repositorio.db", null, 1);

    @SuppressLint("NewApi")

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();

        StrictMode.setThreadPolicy(policy);

        setContentView(R.layout.activity_menu_enfermero);

        Button reg= (Button) findViewById(R.id.btn_registrar);

        Button bor= (Button) findViewById(R.id.btn_borrar);

        Button btn_lista_subir= (Button) findViewById(R.id.btn_subir);

        //al hacer click en el boton de registro

        reg.setOnClickListener(new View.OnClickListener() {

```

```

        @SuppressWarnings({ "deprecation", "static-access" })
        public void onClick(View view) {
            Intent ir_ver_enfermero = new
Intent(getApplicationContext(), Registrar_datos.class);
            startActivity(ir_ver_enfermero);
        }
    });

    //al hacer click en el boton de borrado
    bor.setOnClickListener(new View.OnClickListener() {
        @SuppressWarnings({ "deprecation", "static-access" })
        public void onClick(View view) {

            SQLiteDatabase bd =admin.getReadableDatabase();

            bd.delete("paciente", null, null);

            bd.delete("muestras", null, null);

            bd.close();

            Toast.makeText(Menu_enfermero.this, "Todos los datos de la base
borrados exitosamente! Puede volver a registrar nuevos datos.",
Toast.LENGTH_SHORT).show();

        }
    });

```

```

btn_lista_subir.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {

        String selectQuery = "select * from paciente";
        SQLiteDatabase db =
admin.getReadableDatabase();

        Cursor cursor = db.rawQuery(selectQuery, null);
        if (cursor.moveToFirst()) {
            do {

                Controlador con=new Controlador();

                String pac_id      =cursor.getString(0).trim();
                String pa_id      =cursor.getString(1);
                String pa_nombre  =cursor.getString(2);
                String pa_apellidos =cursor.getString(3);
                String pa_sexo    =cursor.getString(4);
                String pa_cedula  =cursor.getString(5);
                String pa_telefono =cursor.getString(6);
                String pa_direccion =cursor.getString(7);
                String pa_correo   =cursor.getString(8);

                JSONObject json =con.registrarPaciente(pac_id, pa_id, pa_nombre,
                pa_apellidos, pa_sexo, pa_cedula, pa_telefono, pa_direccion, pa_correo);

                try {

```



```

        if (json.getString("success") != null) {
            String res = json.getString("success");
            if(Integer.parseInt(res) == 1){
String mensaje = "PACIENTE #"+cursor.getString(0)+" ENVIADA";
                Log.i("MENSAJE DEL SERVER:", mensaje);
            }else{
String mensaje = "PACIENTE #"+cursor.getString(0)+" NO ENVIADA";
                Log.i("MENSAJE DEL SERVER:", mensaje);
            }
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
} while (cursor.moveToNext());

Toast.makeText(Menu_enfermero.this, "Datos de Pacientes subidos
satisfactoriamente!", Toast.LENGTH_SHORT).show();

}

cursor.close();

String selectQuery2 = "select * from muestras";

SQLiteDatabase db2 = dmin.getReadableDatabase();

Cursor cursor2 = db2.rawQuery(selectQuery2, null);

if (cursor2.moveToFirst()) {

```

```

do {
    Controlador con=new Controlador();
    String mu_id      =cursor2.getString(0).trim();
    String fk_paciente =cursor2.getString(1);
    String mu_tipo    =cursor2.getString(2);
    String mu_y_voltaje =cursor2.getString(3);
    String mu_x_tiempo =cursor2.getString(4);
    String mu_estado  =cursor2.getString(5);

JSONObject json =con.registrarmuestra(mu_id, fk_paciente, mu_tipo,
mu_y_voltaje, mu_x_tiempo, mu_estado);

    try {
        if (json.getString("success") != null) {
            String res = json.getString("success");
            if(Integer.parseInt(res) == 1){
String mensaje = "MUESTRA #"+cursor2.getString(0)+" ENVIADA";
                Log.i("MENSAJE DEL SERVER:", mensaje);
            }else{
String mensaje = "MUESTRA #"+cursor2.getString(0)+" NO ENVIADA";
                Log.i("MENSAJE DEL SERVER:", mensaje);
            }
        }
    }

```

```

        }
        } catch (JSONException e) {
e.printStackTrace();
        }
    } while (cursor2.moveToNext());

    Toast.makeText(Menu_enfermero.this, "Datos de Muestras subidos
satisfactoriamente!", Toast.LENGTH_SHORT).show();
    }

        cursor2.close();
        db2.close();
        db.close();
    }
});
}

```

//Clase principal para crear un progressdialog y verificar la exportacion y tiempo que se demora el visitante

```

public class ExportDatabaseCSVTask extends AsyncTask<String, Void,
Boolean> {
    private final ProgressDialog dialog = new
ProgressDialog(Menu_enfermero.this);

```

```

@Override

protected void onPreExecute() {

    this.dialog.setMessage("Exportando Base de datos...");

    this.dialog.show();

}

        protected Boolean doInBackground(final String... args) {

            File exportDir = new File(

                Environment.getExternalStorageDirectory(), "");

            if (!exportDir.exists()) {

                exportDir.mkdirs();

            }

//Creacion del archivo que sellenara de los datos

            File file = new File(exportDir,"datos_pacientes.csv");

            try {

                file.createNewFile();

                CSVWriter csvWrite = new CSVWriter(new FileWriter(file));

                SQLiteDatabase db = admin.getReadableDatabase();

                String[] cad=paciente_elegido.split(" ");

                Cursor curCSV = db.rawQuery("select

mu.mu_id,mu.fk_paciente,mu.mu_tipo,mu.mu_y_voltaje, mu.mu_x_tiempo,

mu.mu_estado,pa.pa_cedula,pa.pa_telefono,pa.pa_direccion,pa.pa_correo

```

```
from muestras mu,paciente pa where mu.fk_paciente=(select pac_id from
paciente where pa_nombre="+cad[0]+" and pa_apellidos="+cad[1]+""),
null);
```

```
csvWrite.writeNext(curCSV.getColumnNames());
```

```
while (curCSV.moveToNext()) {
```

```
    String[]
```

```
    nombres=admin.obtener_nombre_paciente(curCSV.getString(1));
```

```
        String arrStr[] = { curCSV.getString(0), nombres[0],
```

```
curCSV.getString(2),
```

```
curCSV.getString(3), curCSV.getString(4),
```

```
curCSV.getString(5) ,
```

```
nombres[1],nombres[2],nombres[3],nombres[4]
```

```
    };
```

```
        csvWrite.writeNext(arrStr);
```

```
    }
```

```
        csvWrite.close();
```

```
        curCSV.close();
```

```
        return true;
```

```
    } catch (SQLException sqlEx) {
```

```
        Log.e("MainActivity", sqlEx.getMessage(), sqlEx);
```

```
        return false;
```

```
    } catch (IOException e) {
```

```

        Log.e("MainActivity", e.getMessage(), e);

        return false;
    }
}

protected void onPostExecute(final Boolean success) {
    if (this.dialog.isShowing()) { this.dialog.dismiss(); }

    if (success) {
        Toast.makeText(Menu_enfermero.this, "Exportacion exitosa!",
Toast.LENGTH_SHORT).show();

        File exportDir = new File(
                                Environment.getExternalStorageDirectory(),
                                "");

        if (!exportDir.exists()) {
            exportDir.mkdirs();
        }

        File file = new File(exportDir, "datos_pacientes.csv");

        ///adju archivo q ya se puede obtener al mail

```

```

        Intent sendIntent = new Intent(Intent.ACTION_SEND);
        sendIntent.putExtra(android.content.Intent.EXTRA_EMAIL, new
String[]{""});
        sendIntent.putExtra(Intent.EXTRA_SUBJECT, "Envio de Base de
datos");
        sendIntent.putExtra(Intent.EXTRA_TEXT, "No responder a este
correo. Luego de descargar el adjunto ya puede ir a la Aplicacion CardioFGJ
y utilizar el modo DOCTOR para la revision de datos.");
        Uri uri = Uri.fromFile(file);
        sendIntent.putExtra(Intent.EXTRA_STREAM, uri);
        sendIntent.setType("application/csv");

        startActivity(Intent.createChooser(sendIntent, "Envia Base de
datos via GMAIL:"));
    } else {
        Toast.makeText(Menu_enfermero.this, "Export failed Archivo:
datos.csv", Toast.LENGTH_SHORT).show();
    }
}
}

@Override

```

```
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if it is  
present.  
    getMenuInflater().inflate(R.menu.menu_enferm, menu);  
    return true;  
}  
}
```


BIBLIOGRAFÍA

[1] Texas Heart Institute (Instituto del Corazón de Texas), Anatomía del corazón, http://www.texasheartinstitute.org/HIC/Anatomy_Esp/anato_sp.cfm, fecha de consulta junio 2013.

[2] Wikipedia, Corazón, <http://es.wikipedia.org/wiki/Coraz%C3%B3n>, fecha de consulta junio 2013.

[3] Wikipedia, Ciclo cardíaco

http://es.wikipedia.org/wiki/Ciclo_card%C3%ADaco, fecha de consulta junio 2013.

[4] Wikipedia, Sistema de conducción eléctrica del corazón, http://es.wikipedia.org/wiki/Sistema_de_conducci%C3%B3n_el%C3%A9ctrica_del_coraz%C3%B3n, fecha de consulta julio 2013.

[5] About.com, enfermedades del corazón.

<http://enfermedadescorazon.about.com/od/Diagnostico/a/Que-Es-Un-Electrocardiograma-o-ECG.htm>, fecha de consulta julio 2013.

[6] Onmeda.es, Interpretación del EKG.

http://www.onmeda.es/clinica/exploracion_tratamiento/ekg-interpretacion-del-ecg-2467-5.html, fecha de consulta julio 2013.

[7] Dalcame (grupo de investigación biomédica), Frecuencia Cardíaca.

<http://www.dalcame.com/fc.html#.UqnkjNJLOJk>, fecha de consulta Agosto 2013.

[8] My EKG La Web del electrocardiograma, Derivaciones Cardiacas del Electrocardiograma, <http://www.my-ekg.com/generalidades-ekg/derivaciones-cardiacas.html>, fecha de consulta Agosto 2013.

[9] Andrea Viviana Aguirre Fernández, Jorge Ignacio Enríquez Álvarez, Mediciones Electrocardiográficas, http://www.dspace.espol.edu.ec/bitstream/123456789/17100/1/PAPER3_final.pdf, fecha de consulta Agosto 2013.

[9.1] Wikipedia, Java (Lenguaje de Programación) http://es.wikipedia.org/wiki/Java_%28lenguaje_de_programaci%C3%B3n%29 fecha de consulta Agosto 2013.

[10] Oracle, Principales Características de Java, <http://personales.upv.es/rmartin/cursoJava/Java/Introduccion/PrincipalesCaracteristicas.htm>, fecha de consulta Septiembre 2013.

[11] Tutorial de Java, Programación en Java, <http://proton.ucting.udg.mx/tutorial/java/Cap1/progra.html>, fecha de consulta Septiembre 2013.

[12] Wikipedia, Android, <http://es.wikipedia.org/wiki/Android>, fecha de consulta Octubre 2013.

[13] El android libre, La historia del logo de Android, <http://www.elandroidelibre.com/2013/10/la-historia-del-logo-de-android.html>, fecha de consulta Octubre 2013.

[14] Wikipedia, Android, <http://es.wikipedia.org/wiki/Android>, fecha de consulta Octubre 2013.

[15] Wikipedia, Historial de versiones de Android, http://es.wikipedia.org/wiki/Anexo:Historial_de_versiones_de_Android, fecha de consulta Noviembre 2013.