ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"Electrocardiógrafo Portátil usando Plataforma NIOS II "

TESINA DE SEMINARIO

Previa la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

Cindy Carolina Medina Collahuazo

Paúl Eduardo Quezada Oquendo

GUAYAQUIL – ECUADOR

Año 2014

AGRADECIMIENTO

A Dios por bendecirme con la familia que tengo.

A mis padres por la educación y por inculcarme buenos valores, además de todo el cariño, apoyo incondicional y confianza brindada durante toda mi carrera.

A mi hermano por ayudarme en mis estudios. Mi enamorado por demostrarme su amor y contagiarme de sus ganas de salir adelante y crecer profesionalmente.

Al ingeniero Ponguillo por compartir sus conocimientos y ayuda para culminar este seminario con éxito.

Cindy Carolina Medina Collahuazo

Quiero agradecer a Dios por darme una familia excelente llena de virtudes e ideales.

Agradezco infinitamente a mis padres por la educación y valores que me brindaron, lo que me ha permitido desenvolverme a lo largo de mi vida personal y estudiantil. Ellos son mi ejemplo de vida y superación.

A mis hermanas, porque gran parte de lo que soy ahora se debe a su constancia y a su apoyo en lo educativo y en lo cotidiano.

A mi enamorada por apoyarme incondicionalmente y contagiarme de ganas para mi desarrollo profesional.

Paúl Eduardo Quezada Oquendo

DEDICATORIAS

A Dios y a mi familia por todo el apoyo y confianza, que me han dado fuerzas para culminar con éxito mi objetivo.

Cindy Carolina Medina Collahuazo

A Dios, y a mi familia por todos los consejos brindados y el empuje necesario para cumplir el objetivo propuesto.

Paúl Eduardo Quezada Oquendo

TRIBUNAL DE SUSTENTACIÓN

_____

Ing. Ronald Ponguillo

PROFESOR DEL SEMINARIO DE GRADUACIÓN

_____

Ing. Víctor Asanza

PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este trabajo, nos corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL". (Reglamento de exámenes y títulos profesionales de la ESPOL).

_____

Cindy Carolina Medina Collahuazo

_____

Paúl Eduardo Quezada Oquendo

RESUMEN

En el presente trabajo, “Electrocardiógrafo usando Plataforma Nios II”, se muestra el diseño e implementación de un electrocardiógrafo que permite monitorear el ritmo y la frecuencia cardíaca. Este equipo es un prototipo con similares características al que posee un electrocardiógrafo real, donde se obtiene la señal cardíaca mientras el paciente este en reposo las cuales se almacenan en una tarjeta SD.

El proyecto fue desarrollado utilizando la tarjeta de desarrollo Nios II Embedded Evaluation Kit basado en un dispositivo FPGA Cyclone III de Altera, en el que se ha diseñado la arquitectura de un mini-computador basado en el microprocesador NIOS II.

El sistema se ha divido en etapas, en la cuales se encuentra la adquisión, acoplamiento de la señal cardíaca y la parte de digitalización para el almacenamiento y procesamiento de la señal para luego ser visualizada en

una LCD Touch Screen. El proyecto se lo ha estructurado en 4 capítulos que se detallan a continuación:

En el capítulo 1, damos a conocer los objetivos generales y específicos del proyecto, planteando los alcances y limitaciones.

En el capítulo 2, se describe la parte teórica, dando a conocer el funcionamiento, anatomía y fisiología del corazón, además de detallar cada elemento de la tecnología utilizada para el desarrollo del proyecto.

En el capítulo 3, se muestra el diseño e implementación del proyecto, donde describe detalladamente cada etapa del sistema que se implementó, además de la programación que se realizó para el procesamiento de la señal.

En el capítulo 4, se presenta los resultados obtenidos de las diferentes pruebas realizadas al sistema.

Finalmente, las conclusiones y recomendaciones del proyecto.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ABREVIATURAS

ADC	Convertidor Analógico - Digital
AV	Auriculoventricular
BGR	24bits de colores (8 bits Azúl, 8 bits Verde, 8 bits Rojo) dirigidos hacia el LCD panel.
CMRR	Razón de rechazo al modo común
CPLD	Complex Programmable Logic Device
ECG	Electrocardiograma
E/S	Entrada y Salida
FPGA	Field Programmable Gate Array
GPIO	General Purpose Input/Output
HSMC	High Speed Mezzanine Card
LCD	Liquid Crystal Display
LPM	Latidos por minuto
MTDB	Panel Táctil de la Tarjeta Multimedia

NEEK	NIOS II Embedded Evaluation Kit
PCB	Printed Circuit Board
PIC	Peripheral Interface Controller
RGB	Estándar de colores (Rojo, Verde, Azúl)
SA	Sinoauricular
SD	Secure Digital
TDM	Multiplexación por división de tiempo

INTRODUCCIÓN

En nuestra sociedad el mal hábito alimenticio y la falta de conciencia en el cuidado de la salud, han llevado a las personas a tener problemas cardíacos.

Los equipos tales como el electrocardiógrafo pueden ayudar a los médicos al diagnóstico y detección de problemas cardíacos tales como taquicardia o bradicardia por medio de electrocardiogramas (EGK) con el fin de evitar enfermedades a futuro.

Actualmente muchas tecnologías están basadas en los dispositivos FPGA (Field Programmable Gate Array), por su versatilidad y por permiten a los usuarios programar de acuerdo a sus requerimientos y necesidades.

De la amplia familia de FPGAs este proyecto se desarrolló con la tarjeta NEEK Cyclone III de Altera, mediante su puerto de expansión captura los datos digitalizados de la señal cardíaca, y a través de su LCD Touch Screen se puede visualizar la señal.

CAPÍTULO 1

1. GENERALIDADES

En este capítulo se plantea los objetivos, dando a conocer los alcances y las limitaciones del proyecto.

1.1 OBJETIVOS

1.1.1 OBJETIVOS GENERALES

El principal objetivo es implementar un electrocardiógrafo basado en el microprocesador NIOS II, capaz de almacenar los datos en una tarjeta SD y que pueda ser visualizada en una LCD Touch screen.

1.1.2 OBJETIVOS ESPECÍFICOS

- Diseñar la etapa de adquisión y acoplamiento de la señal cardíaca.
- Aprender el funcionamiento de la tarjeta NEEK para el manejo de su puerto de expansión y la pantalla Touch Screen.
- Diseñar la arquitectura de la mini-computadora basado en el microprocesador NIOS II.
- Crear un código que capture la señal digitalizada para que sean almacenadas en una tarjeta SD y pueda ser visualizada en la LCD Touch screen, además de ser mostrada en archivo .xls.

1.2 ALCANCES

Entre los alcances de proyecto se tiene:

- Lectura de la onda cardíaca por medio de un circuito electrónico.
- Digitalización de la señal cardíaca.

- Los datos digitalizados son convertidos a un voltaje adecuado para el puerto de expansión de la tarjeta NEEK.
- Captura de la señal cardíaca a través del puerto de expansión.
- Visualización en la LCD Touch screen de la señal cardiaca por medio de un código en lenguaje C en NIOS II IDE.
- Cálculo de la frecuencia cardíaca a través de un código en lenguaje C.
- Los datos obtenidos del puerto de expansión, la frecuencia y el posible diagnóstico de alguna anomalía cardíaca serán almacenados en la tarjeta SD.

1.3 LIMITACIONES

Entre las limitaciones del proyecto tenemos:

- Debido a que solo tiene 8 bits de entrada el puerto de expansión no se tiene una alta calidad de resolución de la señal.

- Al no haber en el mercado local el integrado AD620, se implementó un circuito equivalente al del amplificador de instrumentación con iguales características, pero sin embargo el integrado AD620 vienen con componentes encapsulados, obteniendo una mejor amplificación de la señal y más resistente al calor que es requerido por el uso continuo del electrocardiógrafo.

- Debido que el proyecto fue desarrollado con fines didácticos no se optó por componentes de elevado costo, como el uso de un convertidor más preciso.

CAPÍTULO 2

2. MARCO TEÓRICO

En este capítulo damos a conocer los principales conceptos que se han utilizado en este proyecto. Además de conocer el funcionamiento del corazón se da información detallada sobre la Tarjeta NIOS II Embedded Evaluation Kit de Altera, FPGA Cyclone III y Microprocesador NIOS II.

2.1 PRINCIPIOS Y FUNDAMENTOS DEL ELECTROCARDIÓGRAFO

2.1.1 ANATOMÍA Y FISIOLOGÍA DEL CORAZÓN

El corazón, es el principal órgano del sistema cardíaco responsable de recibir y bombear sangre para que circule por todo el cuerpo. Las dimensiones aproximadas son de 12 centímetros de largo, 9 centímetro de ancho, 6 centímetros de espesor y pesa entre 200 y 245 gramos. Su tamaño se aproximada al de un puño cerrado, pero no tiene la misma forma. [3]

El corazón se encuentra ubicado en el tórax en el mediastino medio, detrás del esternón y delante de la columna vertebral. Tiene una membrana de dos capas, denominada pericardio que envuelve el corazón como una bolsa. La capa externa del pericardio rodea el nacimiento de los principales vasos sanguíneos del corazón, está unida a la espina dorsal, al diafragma y a otras partes del cuerpo por medio de ligamentos. La capa interna del pericardio está unida al músculo cardíaco. Una capa líquida separa las dos capas de la membrana, permitiendo que el corazón se mueva al latir a la vez que permanece unido al cuerpo. [2]

El corazón tiene cuatro cavidades, las cavidades superiores se denominan aurícula izquierda, aurícula derecha y las cavidades inferiores se denominan ventrículo izquierdo y ventrículo derecho. Una pared muscular denominada tabique separa las aurículas izquierda, derecha y los ventrículos izquierdo y derecho, véase Figura 2-1. El ventrículo izquierdo es la cavidad más grande y fuerte del corazón. Las paredes del ventrículo izquierdo tienen un grosor de sólo media pulgada, pero tienen la fuerza suficiente para impulsar la sangre a través de la válvula aórtica hacia el resto del cuerpo. [2]

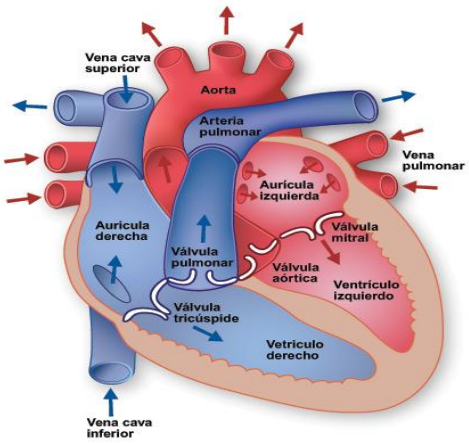


Figura 2-1 Anatomía del corazón [3]

2.1.2 ACTIVIDAD ELÉCTRICA DEL CORAZÓN

La actividad eléctrica del corazón genera diferencias de potencial en la superficie del cuerpo, que se inicia en el nódulo sinoauricular (SA), también llamado como marcapasos natural del corazón, localizado en la parte superior de la aurícula derecha. Los impulsos eléctricos se propagan por las aurículas permitiendo el flujo de sangre hacia los ventrículos, el impulso viaja hacia el nodo auriculoventricular (AV) en el cual se detiene alrededor de 0.1 segundos, y posteriormente pasa por la fibras musculares de los ventrículos permitiendo el paso de la sangre hacia los pulmones y el resto del cuerpo, véase Figura 2-2. [3]

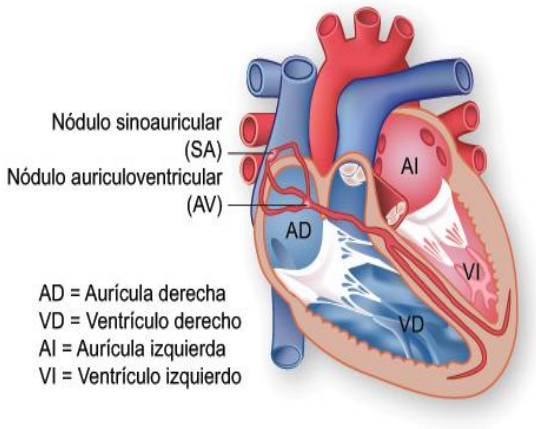


Figura 2-2 Sistema de conducción del corazón [3]

2.1.3 ELECTROCARDIÓGRAFO

Dispositivo que adquiere señales bioeléctricas del corazón, el registro y análisis de estos eventos bioeléctricos son importantes desde el punto de vista clínico. Esta actividad sincronizada, en la que intervienen muchas células, la cual puede registrarse mediante métodos no invasivos, es decir, con el empleo de electrodos de metal colocados en la superficie del cuerpo, de la que se obtiene el electrocardiograma (ECG) que es un registro gráfico en función del tiempo de la actividad eléctrica del corazón.

Nos suministra información sobre el corazón a través de un ECG que se registran durante cada ciclo cardíaco (latido) una serie de curvas u ondas por arriba o por debajo de una línea basal o nivel isoeléctrico. Con la ayuda de un médico especialista se puede diagnosticar si el paciente tiene algún problema o arritmia cardíaca.

2.1.3.1 CARACTERÍSTICAS DEL ECG

Como ya se explicó anteriormente, el inicio de la generación de impulsos eléctricos provienen del nodo

sinoauricular y se desplaza hacia el nodo aurícula ventricular para distribuirse por toda la superficie y realizar la contracción. Cuando el potencial eléctrico se desplaza a través del corazón, se excitan las células que se encuentran a lo largo de su trayectoria.

Si se realiza la correlación de todos los potenciales generados por las diferentes células excitadas durante la transmisión del potencial de acción desde la aurícula hacia los ventrículos aparece el electrocardiograma (ECG). Las ondas características de un ECG se tratan de los voltajes eléctricos generados por el corazón durante un ciclo cardíaco. [7]

- Onda P: Debido a la despolarización de las aurículas.
- Complejo QRS: Debido a la repolarización de las aurículas y la despolarización de los ventrículos.
- Onda T: Corresponde a la repolarización ventricular.

- Onda U: El origen aún es desconocido aunque podría ser debida a la repolarización del sistema de conducción interventricular.

Con el registro de la señal del ECG, podemos realizar un diagnóstico electrocardiógrafo ya que nos da información sobre la coordinación entre los diferentes eventos que suceden durante un ciclo cardíaco.

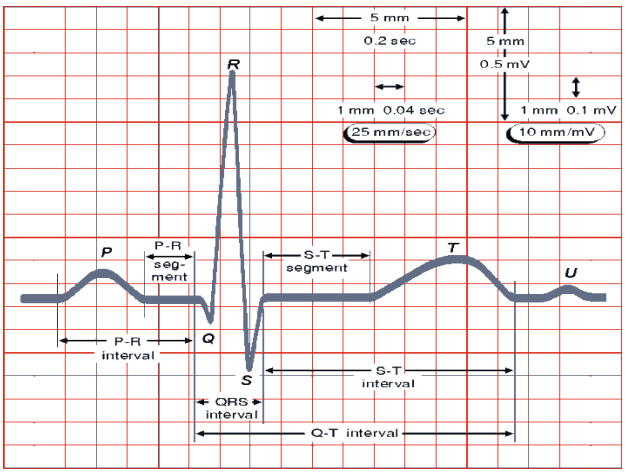


Figura 2-3 Onda cardíaca normal de un ECG [5]

El valor de la frecuencia normal corresponde entre 60 a 100 latidos por minuto el cual puede variar en función del estado en que se encuentre o la edad que tenga el

paciente. Los valores normales que corresponde al ECG normal de un adulto son: [4]

- Onda P: <120 ms
- Intervalo PR: 120-200 ms
- Complejo QRS: <120 ms
- Intervalo QT: <440-460 ms

En la Tabla I muestra los tiempos típicos de las ondas de un ECG.

Parámetros	Rango normal(s)
Intervalo PR	0.12 – 0.20
Intervalo QRS	0.06 – 0.10
Segmentos ST	0.05 – 0.15
Intervalo QT	0.35 – 0.44
Intervalo RR	0.6 – 1.0

Tabla II – Tiempos típicos de un ECG

2.1.4 DIAGNÓSTICO DEL PULSO CARDÍACO Y PRINCIPALES AFECCIONES

La amplitud del complejo QRS en un corazón normal es de 1mV y la duración del complejo QRS es de 0.08s – 0.09s.

Si el corazón no presenta hipertrofia auricular, la onda P tiene una amplitud de 0.1mV con una duración de 0.1s.

Para la onda T estos valores son aproximadamente el doble y puede ser diferenciada de la onda P, la onda T le sigue al complejo QRS después de 0.2s. [4]

2.1.4.1 RITMOS SUPRAVENTRICULARES

Los ritmos cardíacos se dividen en dos clases: supraventriculares (sobre los ventrículos) y ritmos ventriculares.

Los ritmos supraventriculares se dan en el cruce AV, y la activación viene de los ventrículos a lo largo del sistema de conducción en un estado normal como se aprecia en la Figura 2-4.

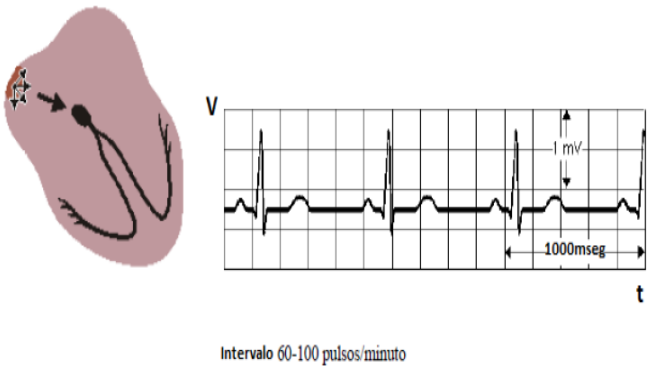


Figura 2-4 Ritmo sinusal normal

2.1.4.2 RITMO SINUSAL NORMAL

El ritmo sinusal normal es el ritmo de un corazón saludable, se lo puede diagnosticar observando las tres deflexiones P-QRS-T y se considera normal si la frecuencia está entre 60-100 pulsos/minuto. [4]

2.1.4.3 BRADICARDIA

La bradicardia se produce cuando el ritmo sinusal es inferior a 60 pulsos/minutos en la Figura 2-5 se observa los complejos normales y espaciado uniforme.

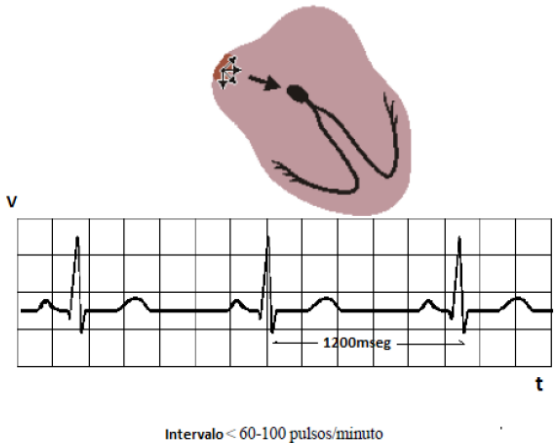


Figura 2-5 Bradicardia sinusal

2.1.4.4 TAQUICARDIA

El ritmo sinusal mayor a 100 pulsos/minutos se lo considera como taquicardia sinusal, esto se da por estrés físico o por el resultado de un corazón congestivo, véase Figura 2-6.

El espacio entre las ondas es corto pero uniforme.

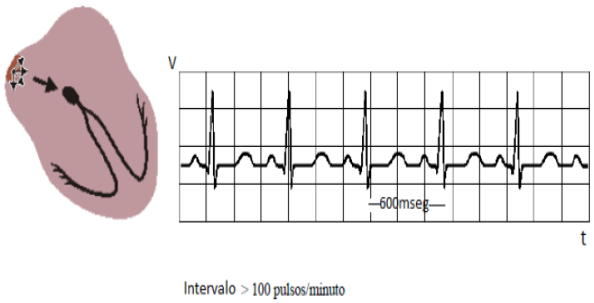


Figura 2-6 Taquicardia sinusal

2.2 PLATAFORMA TECNOLÓGICA

2.2.1 TARJETA DE DESARROLLO NEEK CYCLONE III

NIOS II Embedded Evaluation Kit Altera Cyclone III Edition es una tarjeta de desarrollo, que contiene una matriz de puertas programable (FPGA), una tarjeta LCD Multimedia y herramientas de desarrollo. Todo el sistema es un sistema-en-un-chip (SOPC) diseñado utilizando FPGAs.

La tarjeta de Desarrollo NEEK Cyclone III posee los siguientes accesorios [1]:

- Tarjeta SD 128 MB card
- Adaptador USB-to-SD
- Cable USB
- Fuente de alimentación 9 V
- Nios II Evaluation Kit CD-ROM
- Cable Ethernet (RJ-45) (7ft.)
- Adaptador cruzado Ethernet

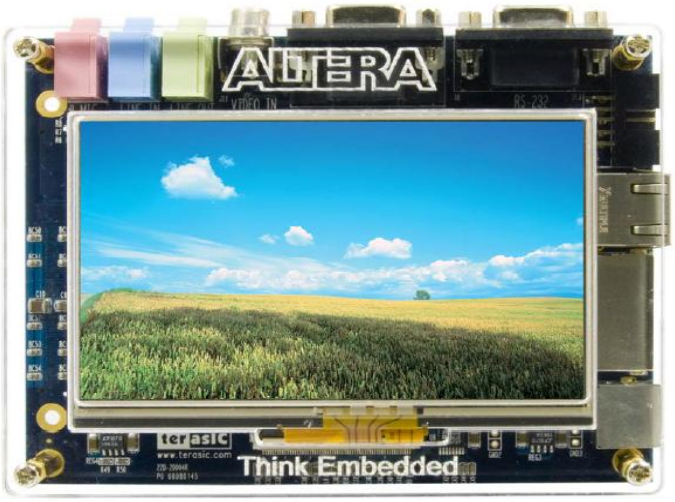


Figura 2-7 Tarjeta NEEK Cyclone III de Altera [2]

Los periféricos que incluyen en la tarjeta NEEK Cyclone III [2]:

- Dispositivo CPLD MAX II 2210
- Entrada tarjeta SD
- Oscilador de reloj 100-MHz
- 24-bit CD-quality audio CODEC with line-in, line-out and microphone-in jacks
- VGA DAC (10-bit high-speed triple DACs) con conector de salida VGA
- Video decoder (NTSC/PAL/SECAM) y conector TV-in

- 10/100M Ethernet Physical Layer Transceiver
- Transceptor RS-232 y conector de 9 pines
- Conector PS/2 mouse/teclado
- Módulo LCD táctil de matriz 800x480
- I2C Serial EEPROM
- I/O uso general

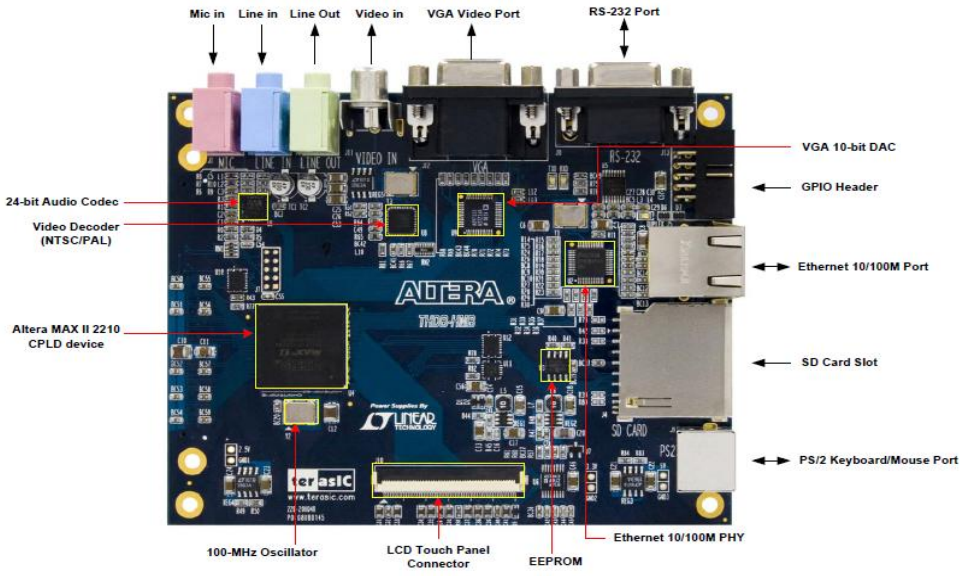


Figura 2-8 Periféricos de la NEEK Cyclone III de Altera [2]

2.2.2 FPGA CYCLONE III

Una FPGA es un dispositivo semiconductor diseñado para ser configurado por el programador o el diseñador después de su

fabricación. La configuración de una FPGA es generalmente usando un lenguaje de Descripción de Hardware.

Las FPGA internamente tienen componentes lógicos programables llamados Bloques Lógicos con interconexión reprogramable que permiten a los bloques ser cableados entre sí. Los bloques lógicos pueden ser configurados para realizar complejas funciones combinatoriales u otras básicas como compuertas lógicas. [3]

FPGA Cyclone III basa su capacidad computacional en elementos lógicos organizados en bloques lógicos basados en una LUT de cuatro entradas, para optimizar el gasto de recursos. Además de ofrecer una combinación única de gran funcionalidad, baja potencia y bajo costo.

Figura 2-9 FPGA Cyclone III [5]

2.2.3 MICROPROCESADOR NIOS II

NIOS II es un procesador de 32 bits diseñado exclusivamente para las FPGA de Altera, que proporciona una infraestructura completa para crear sistemas de microprocesador embebido, según las necesidades del diseñador, por medio de la combinación de una serie de componentes configurables sobre sus FPGAs.

Para desarrollar Sistemas embebidos, Altera proporciona un entorno de desarrollo al que denomina Qsys , que permite la configuración a medida del sistema microprocesador NIOS II y que gracias a la herramienta de síntesis Quartus II puede ser implementado directamente sobre una FPGA.

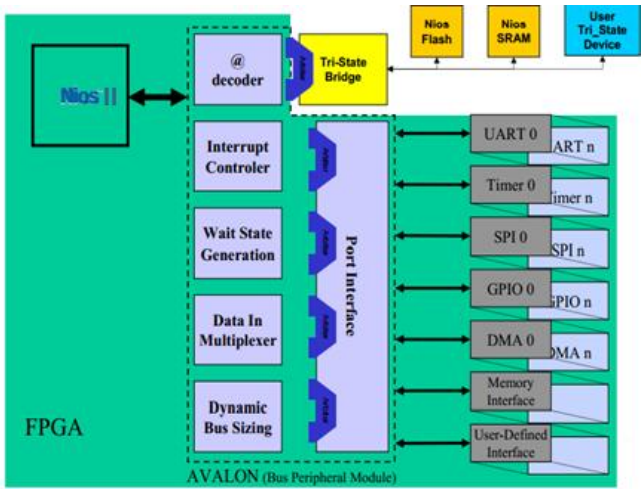


Figura 2-10 Microprocesador basado en NIOS II [8]

Este sistema microprocesador está compuesto por el núcleo procesador NIOS II, memoria interna de programa y de datos, periféricos integrados e interfaces para memoria externa y/o entrada/salida.

El microprocesador NIOS II se lo configura como el usuario lo requiera y este puede ser implementado en tres configuraciones diferentes: [1]

- El NIOS II/f (“fast”) es la versión diseñada para un rendimiento superior, y que proporciona opciones de configuración para aumentar su desempeño, como memorias caché de instrucciones y datos, o una unidad de manejo de memoria (MMU, Memory Management Unit).

- El NIOS II/s (“standard”) es la versión que contiene la unidad aritmético lógica (ALU, Arithmetic Logic Unit) y busca combinar rendimiento y consumo de recursos.

- El NIOS II/e (“economy”) es la versión que requiere menor cantidad de recursos de la FPGA, pero también tiene limitaciones de funciones configurables por el

usuario. Además carece de las operaciones de multiplicación y división.

Cada versión se complementa con diferentes componentes tales como memoria, periféricos que por medio de su interconexión a través del bus de comunicación al que denominan “Avalon Switch Fabric”, para obtener un sistema Nios II completo en un chip (SOC, System On Chip).

2.2.3.1 CARACTERÍSTICAS Y ARQUITECTURA

NIOS II es un procesador de 32 bits de propósito general, basado en una arquitectura tipo Harvard, dado que usa buses separados para instrucciones y datos cuyas principales características son:

- Tamaño de palabra asignado de 32 bits.
- Juego de instrucciones RISC de 32 bits.
- 32 registros de propósito general de 32 bits (r0 – r31)

- 6 registros de control de 32 bits (ctl0 - ctl5)32 fuentes de interrupción externa.
- Capacidad de direccionamiento de 32 bits.
- Operaciones de multiplicación y división de 32 bits.
- Instrucciones dedicadas para multiplicaciones de 64 y 128 bits.
- Instrucciones para operaciones de coma flotante en precisión simple.
- Acceso a variedad de periféricos integrados e interfaces para manejo de memorias y periféricos.

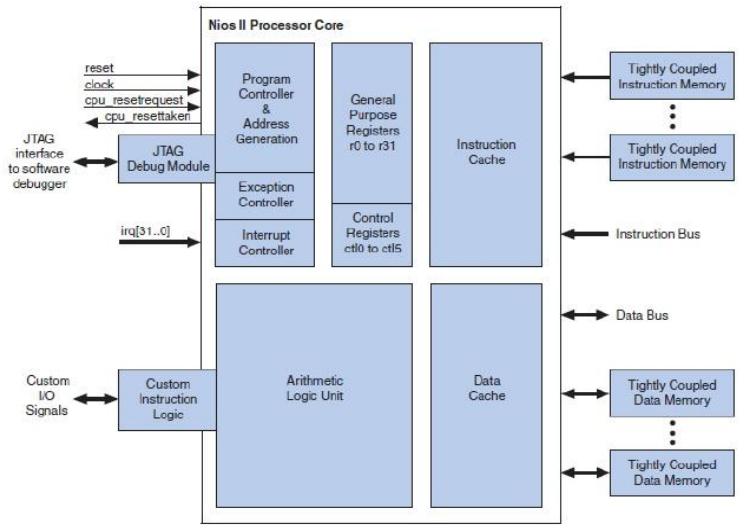


Figura 2-11 Diagrama de Bloques de la Arquitectura NIOS II [1]

2.2.4 CPLD MAX II

CPLD MAX II (EPM2210F324) desempeña un papel esencial en la multiplexación por división de tiempo de señales para LCD y del bus de datos de colores VGA, así como también el cambio de nivel de voltaje para la FPGA y las interfaces de los chips, permitiendo mayor funcionalidad dada la limitación de los pines del conector HSMC, el CPLD MAX II proporciona el controlador del bus.

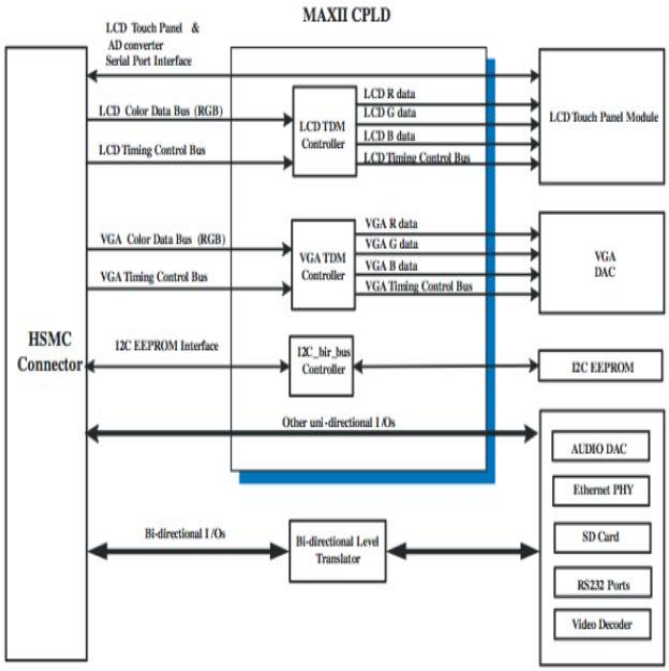


Figura 2-12 Diagrama de Bloque del controlador de bus MTDB [10]

2.2.5 LCD TOUCH PANEL

Tiene una LCD Touch Screen de 4.3“ con una resolución de 800x480 pixeles. A través del LCD Controller podemos manejar los colores del LCD Touch Panel. En el bloque TDM LCD, los datos de entrada de 8 bits (datos de color BGR) que esta dado por el bus de datos de salida de 24 bits (8 bits B + 8bit G + 8bit R) dirigidos hacia el LCD panel. [10]

El diagrama de tiempo del controlador TDM LCD como se muestra en la Figura 2-13 se observa la señal HC_LCD_DATA de 8 bits que contiene los datos de los pixeles de color, cada pixel se representa en tres secuencias de reloj cuyos datos se presentan como “BGR”, además del pulso HC_HD que determina la posición del color azul y se inicia en cada 3 señales de reloj en un periodo de pixel. Los valores de verde y rojo para ese mismo pixel se presentan en los siguientes 2 ciclos de reloj. [10]

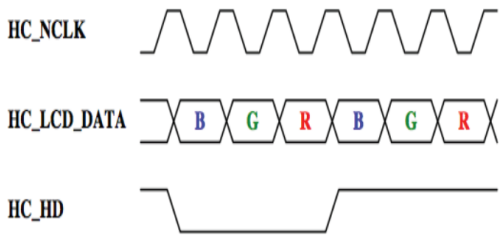


Figura 2-23 Diagrama de tiempo de las entradas del VGA TDM Controller [10]

En el diagrama de tiempo de la Figura 2-14 se muestra las salidas. El bloque TDM LCD genera un reloj NCLK y los datos RGB de 24 bits en el panel LCD. La señal de NCLK funciona 1/3 de la HC_NCLK del reloj entrante. [10]

Los colores de entrada del bus de datos HC_VGA_DATA cambia de 8 bits a 10 bits y el controlador VGA TDM utiliza el HC_VGA_HS para determinar la posición de la muestra de color azul. El diagrama de tiempo del controlador VGA TDM es similar al del controlador TDM LCD. [10]

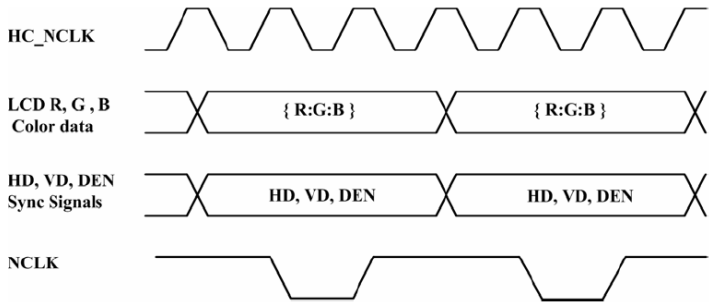


Figura 2-34 Diagrama de tiempo de las salidas del LCD TDM Controller [10]

CAPÍTULO 3

3. DISEÑO Y CONSTRUCCIÓN DEL ELECTROCARDIÓGRAFO

En este capítulo se presenta el diseño e implementación del proyecto, mostrando las diferentes etapas que conforman el electrocardiógrafo y como van interconectadas entre sí.

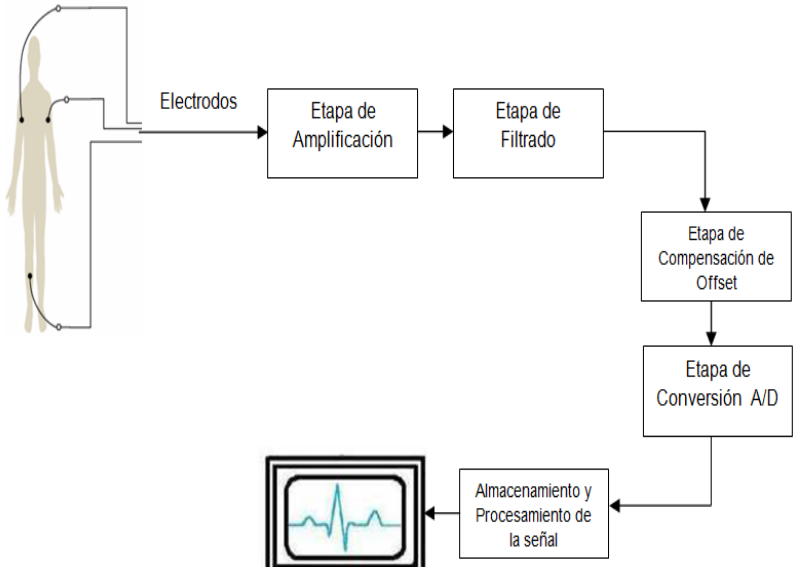


Figura 3-1 Diagrama de Bloques del sistema

En la Figura3-1 podemos apreciar el diagrama de bloque general, es decir, los elementos que conforman el sistema.

3.1 ETAPA DE AMPLIFICACIÓN

La señal cardíaca obtenida es de una diferencia de potencial en el orden de los mV, por lo que debe ser amplificada al nivel de los voltios. En esta etapa de amplificación se ha utilizado un amplificador de instrumentación debido a su alta impedancia de entrada y por su alta relación de rechazo en modo común (CMRR).

Para el amplificador de instrumentación se ha usado el amplificador operacional TL084 como se muestra en la Figura 3-2, donde la ganancia de voltaje está representada por:

$$Av = \frac{R6}{R2}\left(1 + \frac{2R3}{R1}\right)$$

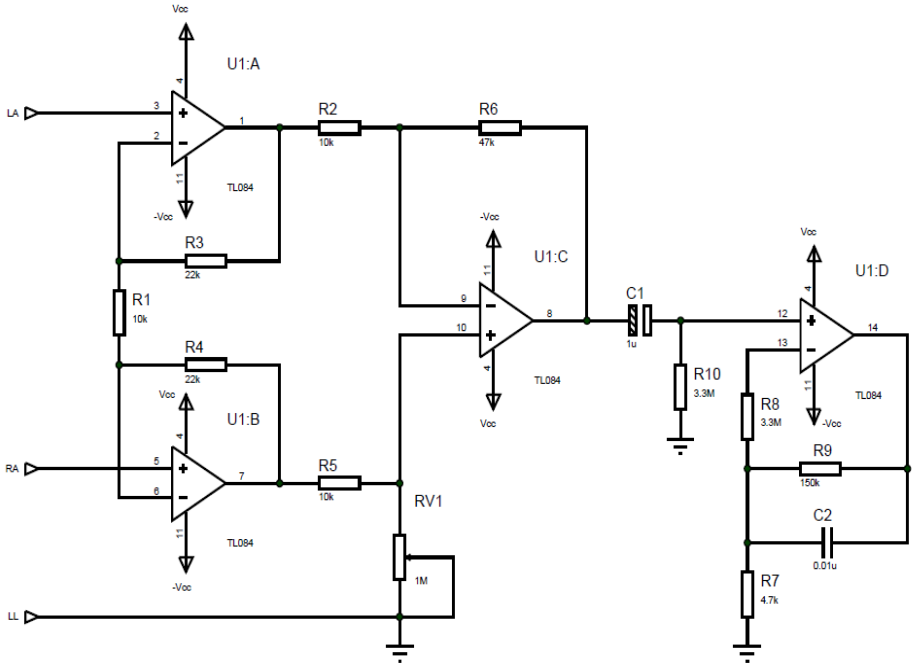


Figura 3-2 Amplificador de Instrumentación

3.2 ETAPA DE FILTRADO

3.2.1 FILTRO PASA-BANDA

Se utilizó un filtro pasa-banda con el objetivo de eliminar frecuencias menores a 0.1Hz y mayores a 150Hz, debido a que el rango de frecuencias de la señal cardíaca no sobrepasa dichos límites. En la Figura 3-3 se muestra el filtro pasa-banda diseñado para el rango de frecuencias de 0.1Hz a 150Hz.

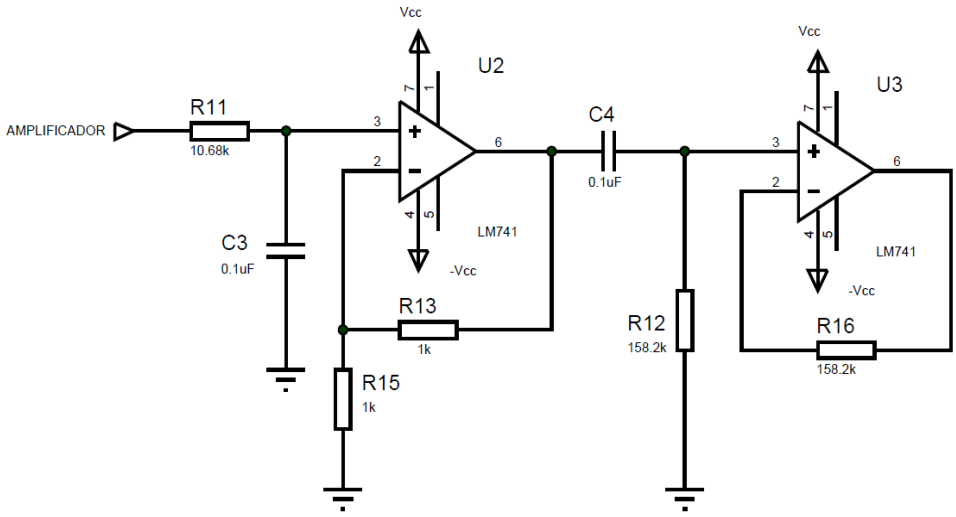


Figura 3-3 Filtro Pasa-Banda

En la Figura 3-3 el circuito pasa-banda consta de un filtro pasa-bajo y un filtro pasa-alto. Donde la frecuencia de corte inferior f_L del filtro pasa-alto esta dado por:

$$f_L = \frac{1}{2\pi.R.C} = 0.1\ Hz$$

La frecuencia de corte superior f_H está determinada por:

$$f_H = \frac{1}{2\pi.R.C} = 150\ Hz$$

3.2.2 FILTRO NOTCH

El filtro Notch o también llamado rechaza-banda es utilizado para atenuar el ruido inducido por la línea de transmisión eléctrica, el cual es de 60Hz.

Donde la frecuencia de rechazo es:

$$f_o = \frac{1}{2\pi.R.C} = 60Hz$$

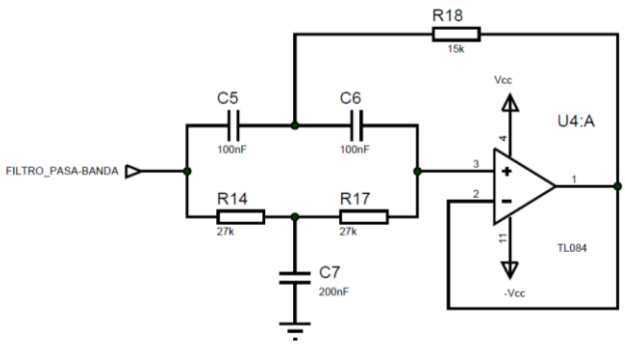


Figura 3-4 Filtro Notch

3.3 ETAPA DE COMPENSACIÓN DE OFFSET

La señal cardíaca obtenida en la parte analógica debe ser acondicionada a niveles de 0 a 5V que permiten los microcontroladores, para esto se ha implementado un amplificador operacional como sumador inversor adhiriendo un nivel DC a la señal que será regulada por un potenciómetro, para que la señal no esté por debajo de cero voltios.

A continuación se muestra el circuito del sumador inversor utilizado:

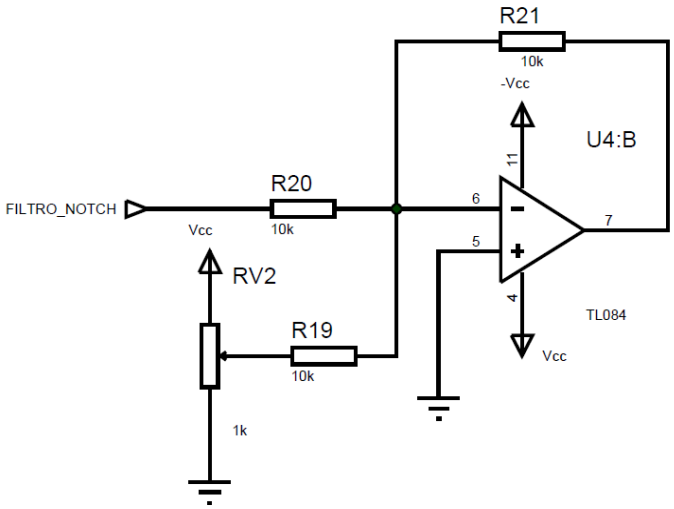


Figura 3-5 Sumador Inversor

Luego pasa por un amplificador inversor como se muestra en la Figura 3-6 para obtener la señal original.

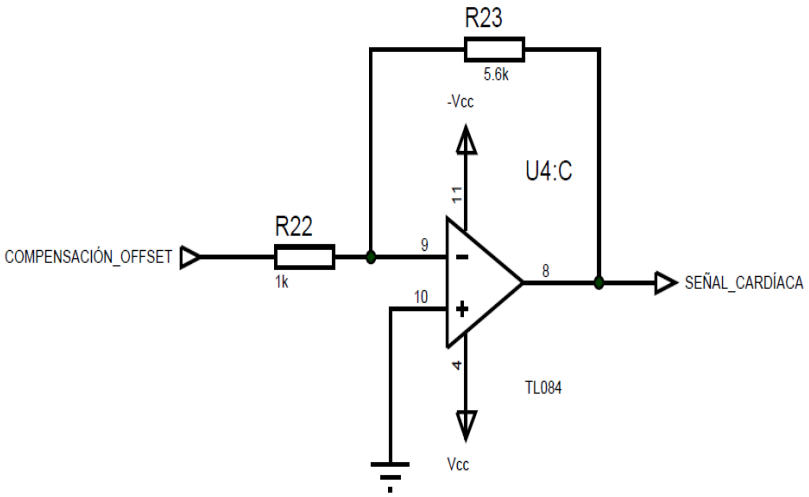


Figura 3-6 Amplificador Inversor

3.4 ETAPA DE CONVERSIÓN ANALÓGICA DIGITAL

En esta etapa de digitalización de la señal cardíaca, se utilizó el PIC 16F887 como convertidor Analógico-Digital, que tiene 14 canales y una resolución de 10 bits.

La resolución es el valor digital que toma cada bit en la salida en función al valor de tensión de referencia que corresponde entre 0 a 5V. A continuación la expresión para el cálculo de la resolución: [9]

$$\text{Resolución} = (\text{Vref}^+ - \text{Vref}^-)/\,1024$$

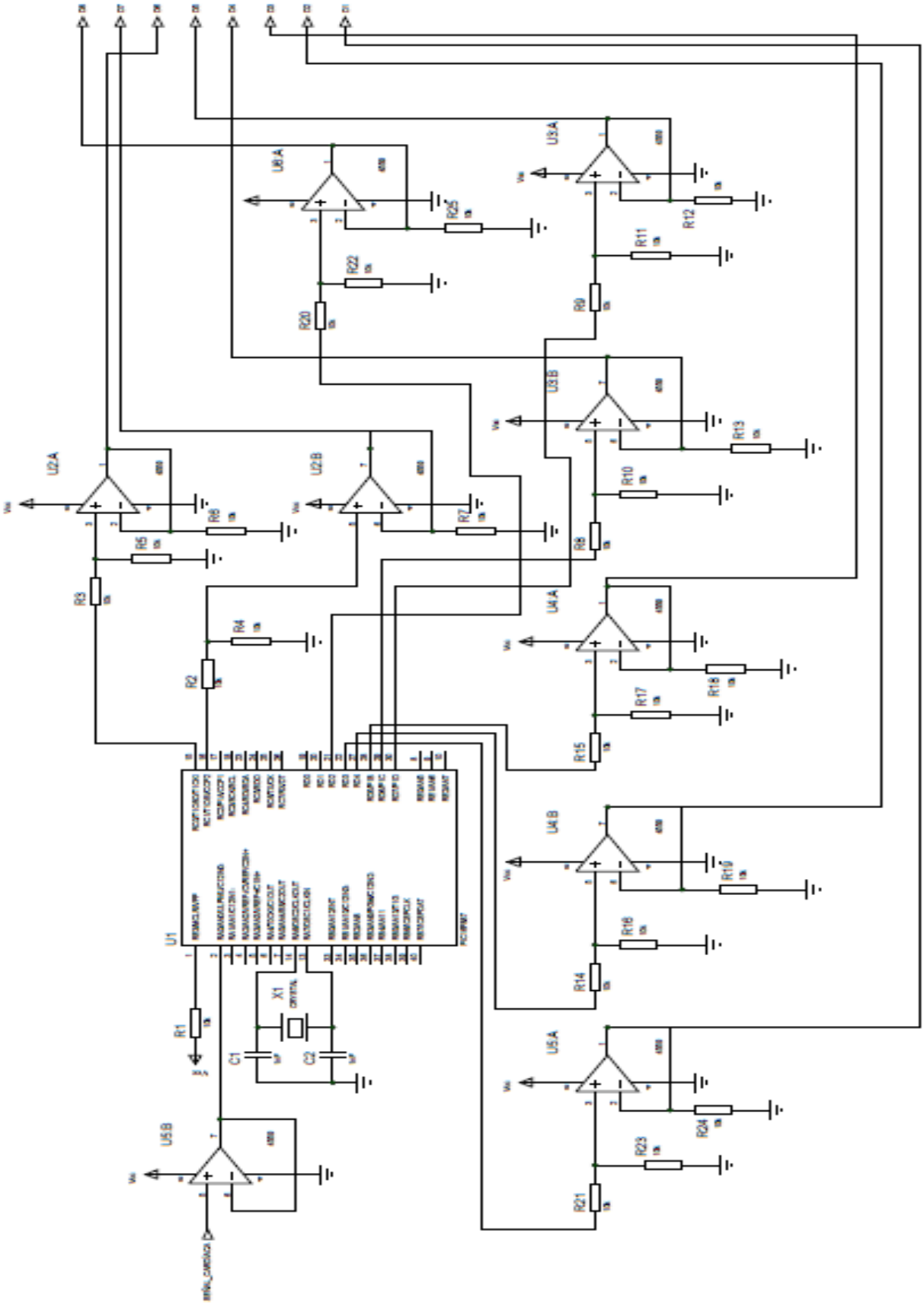


Figura 3-7 Convertidor ADC

En la Figura 3-7 se muestra el convertidor analógico-digital. En la salida del convertidor ADC se ha implementado un seguidor de voltaje que de acuerdo a la resolución de cada bit éste tomará valores entre 0 – 2.5V que son los requeridos para el puerto de expansión de la tarjeta NEEK.

3.4.1 CÓDIGO CONVERTIDOR A/D

El convertidor genera una salida binaria de 10 bits utilizando el método de aproximaciones sucesiones que son almacenadas en registros ADC (ADRESL y ADRESH).

El funcionamiento del convertidor A/D está bajo el control de los bits de cuatro registros:

- ADRESH Registro alto del resultado de la conversión A/D
- ADRESL Registro bajo del resultado de la conversión A/D
- ADCON0 Registro de control 0
- ADCON1 Registro de control 1

Donde el Port A es configurado como entrada, el Port C y Port D son las salidas del convertidor donde el Port C tendrá los 8 bits más significativos.

A continuación el código del convertidor Analógico-Digital:

```
void main()
{
    ANSELH = 0;
    C1ON_bit = 0;
    C2ON_bit = 0;
    ANSEL = 0x04;
    TRISA = 0xFF;
    TRISC = 0;
    TRISB = 0;
    do
     {
        temp_res = ADC_Read(2);
        PORTB = temp_res;
        PORTC = temp_res >> 8;
     } while(1);
 }
```

3.5 DISEÑO HARDWARE EN QSYS

El software QSYS de altera, permite diseñar un sistema basado en el Procesador Embebido NIOS II y configurar todos los componentes necesarios, generando automáticamente la lógica de interconexión, implementándolo en un solo chip y optimizando así el sistema.

El Procesador Embebido NIOS II configurado mediante la herramienta QSys es implementado en el dispositivo FPGA a través de Quartus II.

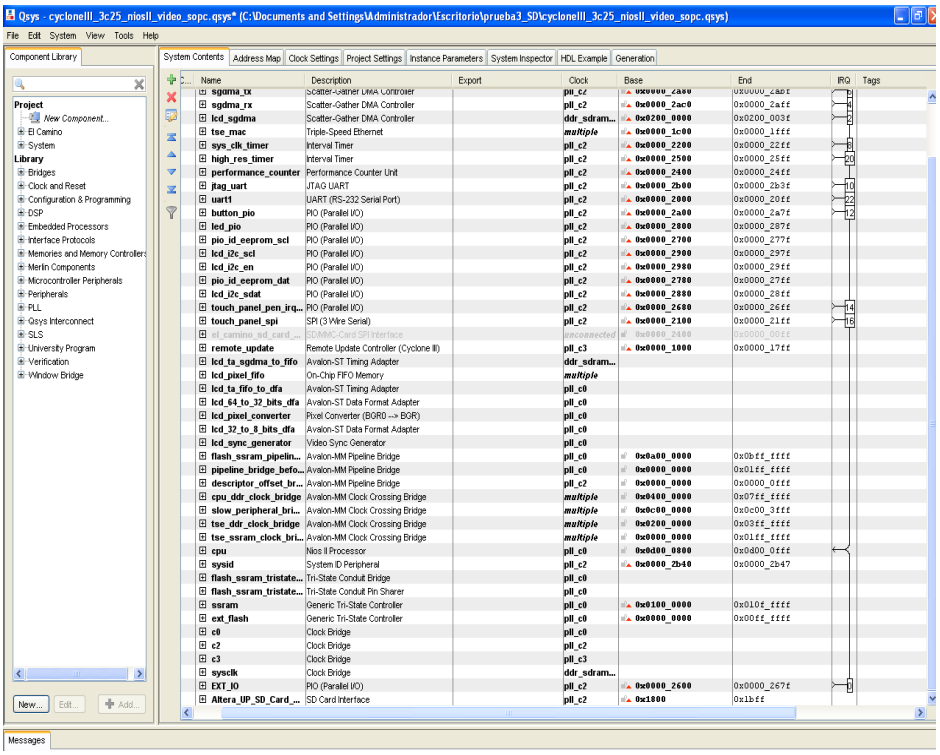


Figura 3-8 Diseño del sistema en QSys

En la Figura 3-8 se muestra el diseño del sistema en QSys, a continuación algunos de los elementos más relevantes:

- El Procesador NIOS II que es el Procesador Central (CPU).
- Controlador LCD

- Puerto paralelo de E/S llamado EXT_IO
- Interfaz para la tarjeta SD

3.5.1 PUERTO PARALELO DE E/S

La tarjeta NEEK consta de un puerto de expansión JP3. Para habilitar los 8 bit es necesario cambiar a modo 1 el registro de control del dispositivo MAX II. La utilidad para conmutar entre modo y EEPROM está localizada en la dirección "*Tool/HMB2_Configuration_Utility/*" que se encuentra en el CD MTDB System. Véase la Figura 3-9 que muestra el registro de control en modo 1.

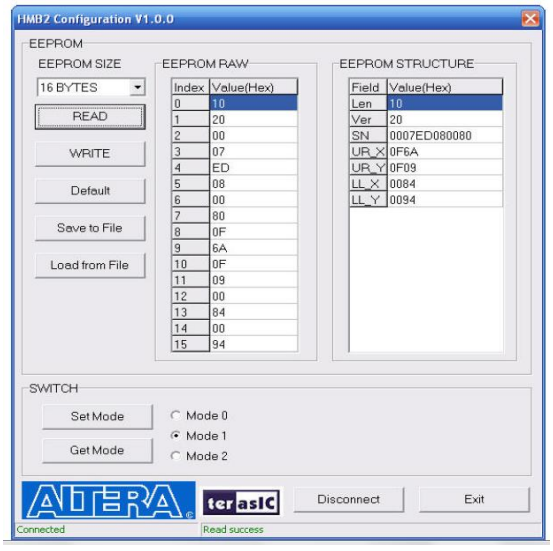


Figura 3-9 Modo de registro de control del MAXII

Una vez habilitado el modo 1, se configura como entrada de 8 bit el puerto de expansión como se puede observar en la Figura 3-10.

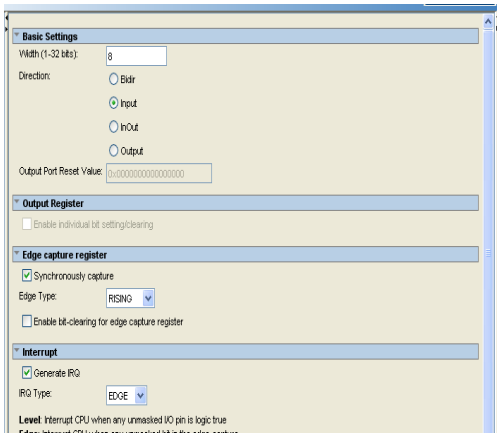


Figura 3-10 Configuración del GPIO en QSys

3.5.1.1 ASIGNACIÓN DE PINES DEL JP3

En la siguiente tabla se describe la asignación de pines de los 8 bit del puerto de expansión JP3.

Nombre de la Señal	Dirección	Locación PINES	Estándar I/O	Descripción
EXT_IO[7]	Entrada	PIN_H2	2.5 V (default)	Estos pines son las entradas de la señal cardíaca digitalizada.
EXT_IO[6]	Entrada	PIN_C1	2.5 V (default)	
EXT_IO[5]	Entrada	PIN_C2	2.5 V (default)	
EXT_IO[4]	Entrada	PIN_T16	2.5 V (default)	
EXT_IO[3]	Entrada	PIN_R16	2.5 V (default)	
EXT_IO[2]	Entrada	PIN_N15	2.5 V (default)	
EXT_IO[1]	Entrada	PIN_K5	2.5 V (default)	
EXT_IO[0]	Entrada	PIN_L5	2.5 V (default)	

Tabla II – Asignación de pines del puerto de expansión JP3

3.6 PROGRAMACIÓN EN LENGUAJE C USANDO SOFTWARE NIOS II

El programa se lo realizó en lenguaje C fue compilado usando el IDE NIOS II Eclipse. En la Figura 3-11 se muestra el algoritmo del sistema.

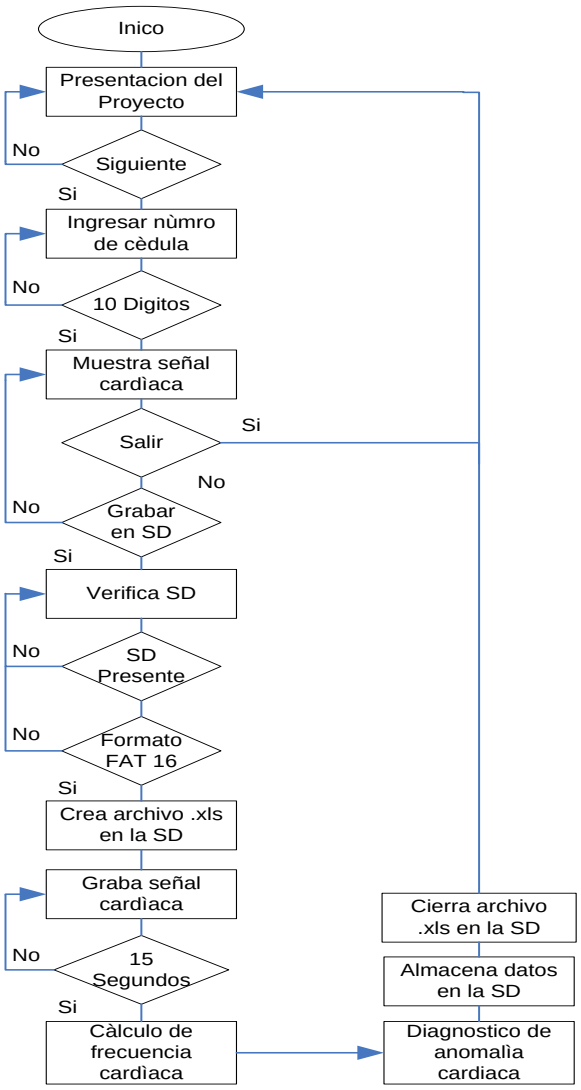


Figura 3-11 Diagrama de Flujo del programa

En el programa se han usado varias librerías tanto del lenguaje C como del IDE de NIOS II de Eclipse que se presentan a continuación:

```
/* librerias */
#include "system.h"
#include "alt_video_display/alt_video_display.h"
#include "alt_touchscreen/alt_touchscreen.h"
```

```
#include "alt_tpo_lcd/alt_tpo_lcd.h"
#include "fonts.h"
#include "stdlib.h"
#include <string.h>
#include <unistd.h>
#include "sys/alt_irq.h"
#include "altera_avalon_spi_regs.h"
#include "altera_avalon_pio_regs.h"
#include <io.h>
#include "sys/alt_cache.h"
#include "graphics_lib/simple_graphics.h"
#include <malloc.h>
#include "altera_avalon_spi.h"
#include <stdio.h>
#include "sys/alt_alarm.h"
#include "alt_types.h"
#include <altera_avalon_sgdma.h>
#include <altera_avalon_sgdma_descriptor.h>
#include <altera_avalon_sgdma_regs.h>
#include <altera_up_sd_card_avalon_interface.h>
```

Las librerías alt_tpo_lcd.h y alt_video_display.h permiten la visualización en la pantalla táctil, mientras que alt_touchscreen.h traduce la posición táctil en la pantalla a la entrada del programa como una interrupción del sistema, y simple_graphics.h contiene las funciones de los gráficos básicos que podrán mostrarse en pantalla.

```
// Verifica si existe el HW para leer/escribir memoria SD
device_reference = alt_up_sd_card_open_dev("/dev/Altera_UP_SD_Card_Avalon_Interface_0");
        if (device_reference != NULL)
        {
                while (1) //Lazo infinito esperando que inserte un SD
                {
                        //Verifica si existe una memoria SD insertada en el slot
                        if ((connected == 0) && (alt_up_sd_card_is_Present()))
                        {
                                //presenta en la LCD Mensaje
                                printf ("\ntajeta OK\n");
                                usleep(50000);
                                //Verifica formato de la SD
                                if (alt_up_sd_card_is_FAT16())
                                {
                                        printf("\nfat16 OK\n");
                                        usleep(50000);
                                        for (i = 0; i<8; i++)

                                            {
                                                    printf ("\n %c ", cedula[i]);
                                                sprintf (cedula2,"%s%c",cedula2, cedula[i]);
                                            }
// Crea el archivo donde se almacena la Información

sprintf(archivo,"%s%c%c%c%c",cedula2,extension,extension1,extension2,extension3);

base = alt_up_sd_card_fopen (archivo, true);

                                        printf("\nbase = %d \n",base);
                                                //a=alt_up_sd_card_fclose (base);
                                                break;
                                        }
                                        else
                                        {
                                                printf("\nfat16 ERROR\n");
                                                usleep(500000);
                                        }
                                        connected = 1;
                                }
                                else if ((connected == 0) && (alt_up_sd_card_is_Present() == false))
                                {
                                        printf ("\nSIN TARJETA\n");
                                        vid_draw_box (290,350,460,390, 0x9C0036, 1, display);
                                vid_print_string(355, 360, 0xFFFFF0, cour10_font, display, "POR FAVOR");
                        vid_print_string(310, 375, 0xFFFFF0, cour10_font, display, "INGRESE LA TARJETA");
                                        usleep(500000);
                                        connected = 0;
                                } } } return ;}
```

Figura 3-12 Verificación de tarjeta SD

En la Figura 3-12 se describe el código de verificación de la tarjeta SD en caso de que la tarjeta SD no se halla ingresado muestra el mensaje

que se debe ingresar la tarjeta, una vez insertada se crea el archivo .xls.

```
int presentacion (alt_video_display * display)
{
        alt_video_display_clear_screen(display,0);
vid_print_string (240, 39, 0XFFFFFF,cour10_font, display, "ESCUELA SUPERIOR POLITECNICA DEL
LITORAL");
        delay (1000);
        vid_print_string (248, 74, 0XFFFFFF, cour10_font ,display, "FACULTAD DE ELECTRICIDAD
Y COMPUTACION");
        delay(1000);
        vid_print_string (312, 109,0XFFFFFF, cour10_font, display, "MATERIA DE
GRADUACION:");
        delay(1000);
        vid_print_string (240, 144, 0XFFFFFF,cour10_font, display, "MICROPROCESADORES
PROGRAMABLES EMBEBIDOS");
        delay(1000);
        vid_print_string (220, 179, 0XFFFFFF, cour10_font, display, "ELECTROCARDIOGRAFO
PORTATIL BASADO EN NIOS II");
        delay(1000);
        vid_print_string (356, 214, 0XFFFFFF, cour10_font, display, "INTEGRANTES");
        delay (1000);
        vid_print_string (308, 249, 0XFFFFFF, cour10_font, display, "CINDY MEDINA
COLLAHUAZO");
        delay (1000);
        vid_print_string (320, 284, 0XFFFFFF, cour10_font, display, "PAUL QUEZADA OQUENDO");
        delay(1000);
        vid_print_string (324, 319, 0XFFFFFF, cour10_font,display, "DIRECTOR DE MATERIA");
        delay(1000);
        vid_print_string(280, 354, 0XFFFFFF, cour10_font, display, "MSC. RONALD PONGUILLO
INTRIAGO");
        delay(1000);
        vid_print_string (340, 389, 0XFFFFFF, cour10_font, display, "PERIODO LECTIVO");
        delay (1000);
        vid_print_string (356, 424, 0XFFFFFF, cour10_font, display, "2013 - 2014");
        delay (1000);
        return (0);
}
```

Figura 3-13 Código de presentación inicial

En el segmento de código de la Figura 3-13 permite visualizar en pantalla el tema de proyecto y autores.

```
int boton (alt_video_display * display, char string [],
                                    int h_start,
                                    int v_start,
                                    int h_end,
                                    int v_end,
                                    int h_char,
                                    int v_char)
int num_cedula (alt_video_display * display, alt_touchscreen *
screen,
                alt_touchscreen_scaled_pen_data pen_data)
int button_press (alt_video_display * display, alt_touchscreen *
screen,
                alt_touchscreen_scaled_pen_data pen_data, char
string[],
                int h_start,
                int v_start,
                int h_end,
                int v_end,
                int h_char,
                int v_char)
```

Figura 3-14 Funciones propias

En la Figura 3-14 se observa las funciones propias, creadas con el fin de facilitar la programación y la reducción del código del programa principal. La función ***int boton*** permite mostrar en pantalla una variable char en la posición deseada, mientras que la función ***button_press*** me indica que se ha presionado un botón.

```
alt_touchscreen_get_pen(screen,
    (&pen_data.pen_down),
    (&pen_data.x),
    (&pen_data.y));

 if (pen_data.x>=250&&pen_data.x<=300&&pen_data.y>=300&&pen_data.y<=320)
              {
                      press=1;
                      coordenada =8;
              vid_draw_box (250,300, 300,320, 0xCCCCCC, 1, display);
           vid_print_string (255, 305, 0, cour10_font, display, "SALIR");
              delay (1000);
              vid_draw_box (250,300 ,300,320, 0x9C0036, 1, display);
      vid_print_string (255,305, 0x9C0036, cour10_font, display, "SALIR");
              }
```

Figura 3-15 Función que obtiene las coordenadas del dato presionado

En la Figura 3-15 se muestra la función ***alt_touchscreen_get_pen*** que está en la librería ***alt_touchscreen.h*** del IDE NIOS II, que permite obtener las coordenadas X y Y de la posición del dato llamado “pen” y devuelve un valor booleano indicando si el “pen” ha sido presionado.

```
if (cont==798)
{
vid_print_string (220, 15,WHITE_16,cour10_font,display,"ELECTROCARDIOGRAFO PORTATIL
BASADO EN NIOS II");
                                                    alt_video_display_clear_screen (display,0);
                                                    cont =0;

                                                    horiz_start=0;
                                                    horiz_end=3;
                                                    }
                                                   usleep (500);
                                                   data = *(dat_io);
                                                   printf ("%d",data);
                                                   vert_end = 256 - (3*data)+100;
vid_draw_line (horiz_start, vert_start, horiz_end,vert_end, 1,0xFFFFFF, display);
                                                    vert_start=vert_end;
                                                    horiz_start=horiz_end;
                                                   horiz_end=horiz_start+3;
                                                   x=0;
 }
```

Figura 3-16 Captura de datos y visualización de la señal cardíaca

En la Figura 3-17 se muestra el código que se realizó para el cálculo de la frecuencia cardíaca.

```
if (data>=60) subida =1;
if (data<60) bajada=1;
if ((subida==1)&& (bajada==1))
{
cont_complejo++;
subida =0;
bajada =0;
      }
```

Figura 3-17 Código que permite calcular la frecuencia cardíaca

El siguiente código, nos permite visualizar en pantalla que anomalía presenta el paciente de acuerdo a su frecuencia cardíaca.

```
if (frec_cardiaca >100)
 {

 vid_print_string (430, 260, 0, cour10_font, display, "TAQUICARDIA");
 }

if (frec_cardiaca <60)
 {
  vid_print_string (320, 260, 0xFFFFFF, cour10_font,
display,bradicardia);
 }

if ((frec_cardiaca >=60) && (frec_cardiaca<=100))
  {
   vid_print_string (320, 260, 0xFFFFFF, cour10_font,
display,"NINGUNA");
  }
```

CAPÍTULO 4

4. PRUEBAS Y RESULTADOS

En este capítulo se muestra las pruebas realizadas, así como la visualización de la señal cardíaca tomadas de diferentes pacientes, el cálculo de la frecuencia cardíaca y el posible diagnóstico de alguna anomalía. Se ha planteado 4 escenarios para mostrar el funcionamiento del proyecto.

4.1 ESCENARIOS

- Escenario A: Colocación normal de los electrodos y toma de datos del paciente.

- Escenario B: Señal cardíaca tomada cuando el paciente este en estado de reposo.

- Escenario C: Señal cardíaca tomada después que el paciente haya realizado alguna actividad física

- Escenario D: Cálculo de la frecuencia cardíaca de forma analógica por medio del complejo QRS.

4.1.1 ESCENARIO A: COLOCACIÓN NORMAL DE LOS ELECTRODOS Y TOMA DE DATOS DEL PACIENTE.

Primero se debe limpiar la piel del paciente para eliminar sudor y grasa de la piel. Una vez que el paciente tenga la piel limpia y este en reposo, sentado o acostado se procede a colocar los electrodos en el pecho como se muestra en la Figura 4-1.

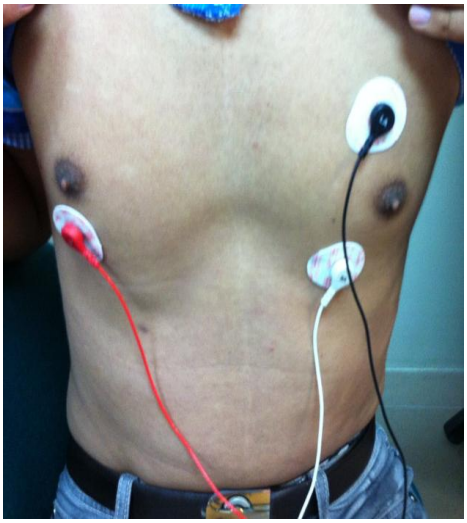


Figura 4-1 Colocación de los electrodos

Al presionar el botón iniciar electrocardiógrafo, permite al paciente ingresar el número de cédula como se muestra en la Figura 4-2.

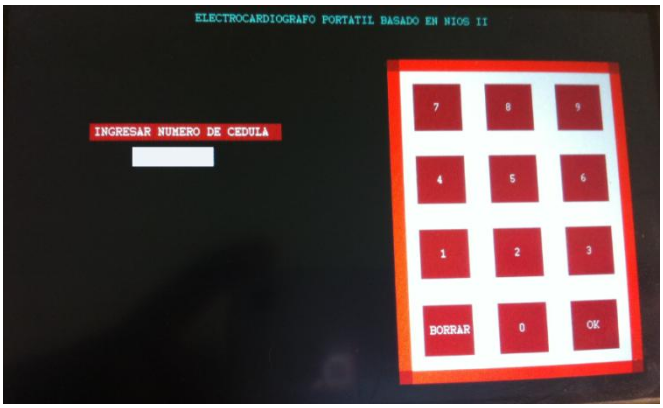


Figura 4-2 Ingreso número de cédula del paciente

4.1.2 ESCENARIO B: SEÑAL CARDÍACA TOMADA CUANDO EL PACIENTE ESTÁ EN ESTADO DE REPOSO.

Ya ingresado el número de cédula del paciente y presionado el botón “OK”, se procede a registrar y visualizar la señal cardíaca como se observa en la Figura 4-3.

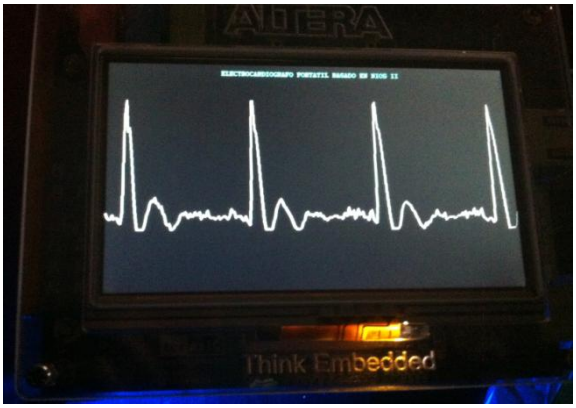


Figura 4-3 Señal Cardíaca del paciente en reposo

El programa también nos permite calcular los latidos por minutos del paciente, mostrando el posible diagnóstico en caso de padecer de alguna anomalía cardíaca. Como ya se expuso anteriormente el rango normal de la frecuencia cardíaca debe de estar entre 60-100 LPM.

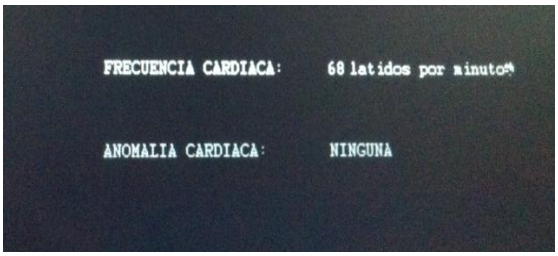


Figura 4-4 Cálculo de la frecuencia cardiaca del paciente en reposo

En la Figura 4-4 se muestra la frecuencia cardíaca y el diagnóstico de acuerdo a los latidos por minuto, como se observa se encuentra dentro del rango normal, en este caso no presenta ninguna anomalía cardíaca.

Habiendo presionado la opción “GRABAR”, los datos de la señal cardíaca se almacenan en la tarjeta SD en un archivo .xls. En la Figura 4-5 se observa la misma señal cardíaca, la frecuencia ahora en archivo .xls.

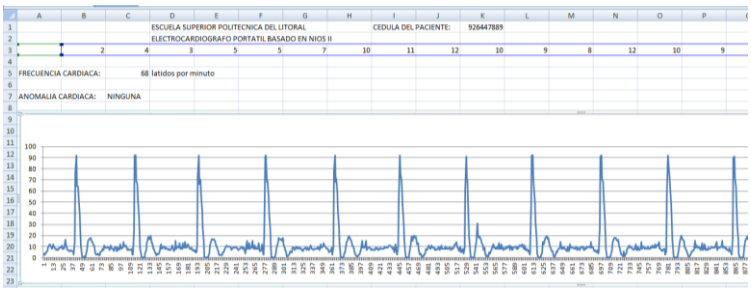


Figura 4-5 Gráfico de la señal cardíaca del archivo .xls

4.1.3 ESCENARIO C: SEÑAL CARDÍACA TOMADA DESPUÉS QUE EL PACIENTE HAYA REALIZADO ALGUNA ACTIVIDAD FÍSICA.

Se diagnosticó al paciente después de realizar un ejercicio físico y podemos ver en la Figura 4-6 que la señal cardíaca presenta las pulsaciones más rápidas en comparación con las pulsaciones en estado de reposo.

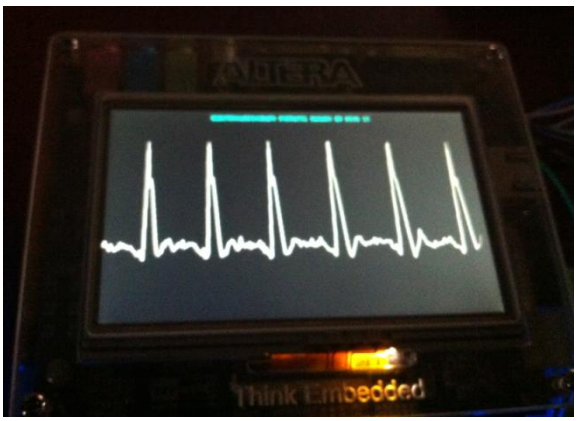


Figura 4-6 Señal Cardíaca después del ejercicio físico

La frecuencia cardíaca en este caso saldrá mayor a 100 LPM, pero esto no resulta un trastorno sino un proceso de adaptación y se denomina taquicardia sinusal. [11]

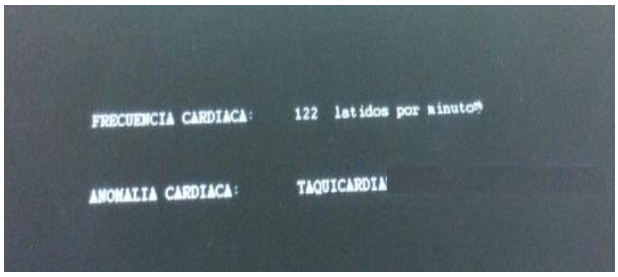


Figura 4-7 Taquicardia sinusal debido al esfuerzo físico

En la Figura 4-8 se visualiza los datos tomados del archivo .xls. Se almacenó la frecuencia y posible anomalía cardíaca.

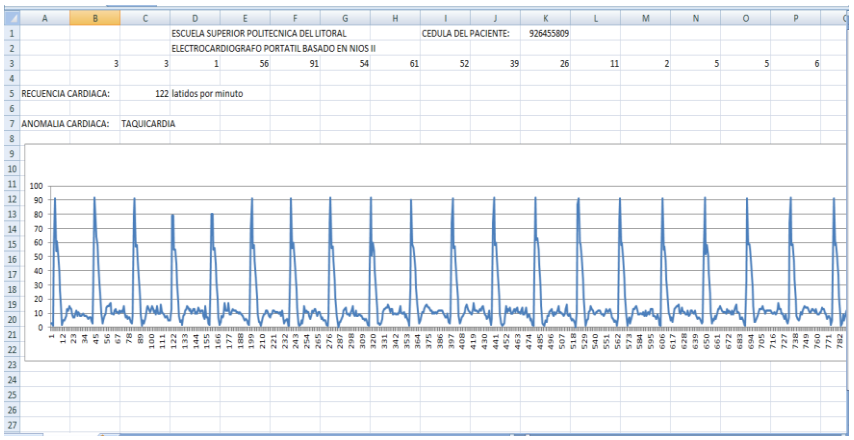


Figura 4-8 Gráfico de la señal cardíaca en archivo .xls

4.1.4 ESCENARIO D: CÁLCULO DE LA FRECUENCIA CARDÍACA DE FORMA ANALÓGICA POR MEDIO DEL COMPLEJO QRS.

La frecuencia cardíaca que se mostró en los escenarios anteriores se la obtuvo directamente de la señal cardíaca sin necesidad de filtrar el complejo QRS. En este escenario se pretende demostrar que también podemos calcular dicha frecuencia obteniendo pulsos correspondientes al complejo QRS. A continuación se describe la parte analógica de la obtención de pulsos.

Una vez filtrada la señal se utilizó un detector de complejo QRS para obtener los pulsos cardíacos. Se implementó un filtro pasa-banda cuya frecuencia central es de 17Hz (propia del complejo QRS) con un ancho de banda de banda BW de 8 Hz, como se muestra en la Figura 4-7.

$$f_c = \frac{1}{2\pi\sqrt{\frac{R_{25}+R_{26}}{R_{25}.C_8.R_{26}.C_9.R_{27}}}} = 17\ Hz$$

$$BW = \frac{1}{\pi.R_{27}.C_8} = 8Hz$$

Con un factor de calidad de:

$$Q = \frac{f_c}{BW} = 2.125$$

Siendo un factor aceptable.

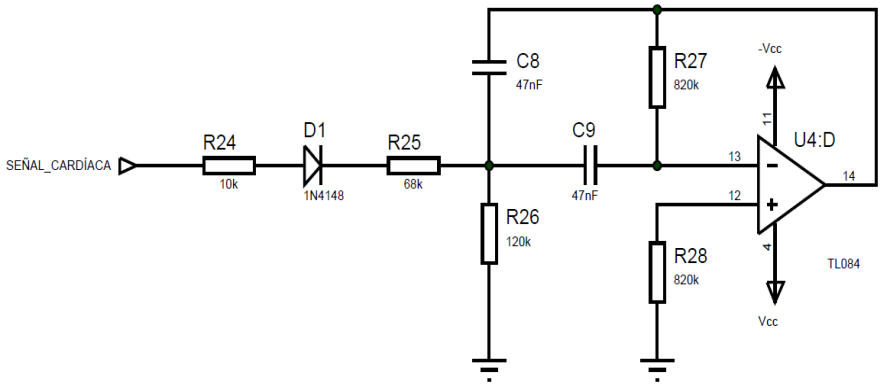


Figura 4-9 Circuito Pasa-Banda del complejo QRS

Luego de filtrar la onda del complejo QRS, éste es rectificado por el circuito mostrado en la Figura 4-10.

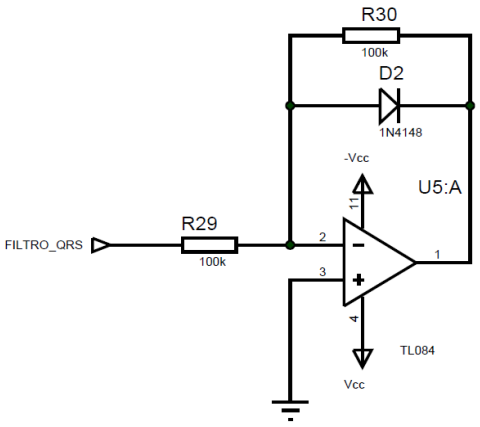


Figura 4-10 Rectificador de media onda

Ya obtenida la onda rectificada es comparada con un voltaje de referencia de 1.2V. Cuando la amplitud de la señal sobrepasa el voltaje de referencia, en la salida del comparador se tiene un nivel de 9V por lo que se debe acondicionar mediante un potenciómetro al nivel de voltaje deseado. En este caso la amplitud del pulso será de 2.5 V.

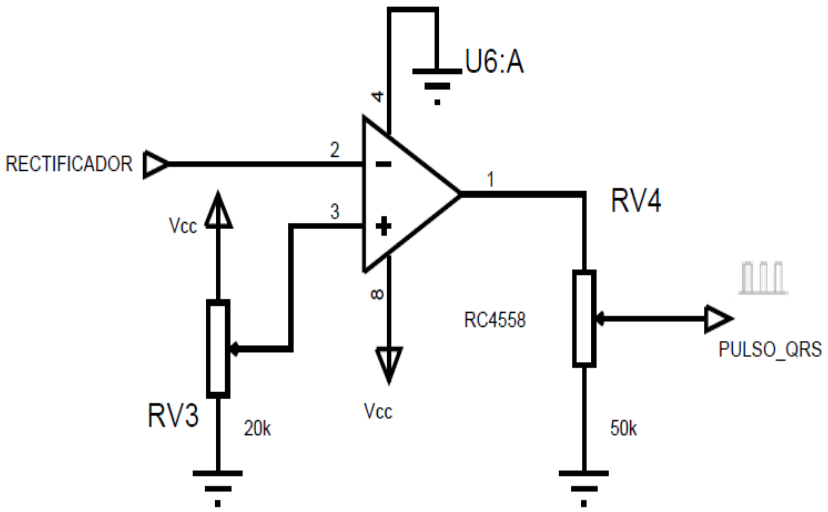


Figura 4-11 Comparador de Voltaje

En la Figura 4-12 con lo que se muestra los pulsos correspondientes al complejo QRS, queda demostrado que por medio del detector del complejo también podemos calcular la frecuencia cardíaca.

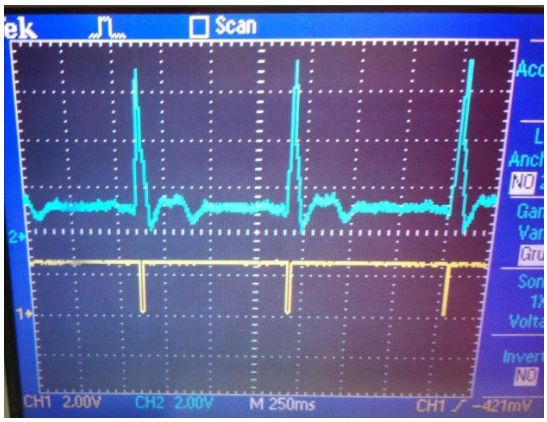


Figura 4-12 Pulsos correspondiente al complejo QRS

CONCLUSIONES

1. Con las pruebas realizadas podemos concluir que es posible calcular la frecuencia cardíaca por medio de la amplitud de la onda y también por el complejo QRS cuya frecuencia se encuentra en 17Hz, que después de rectificarla y compararla genera pulsos que se envía a la FPGA y procederá contarla por medio del código que se realizó en lenguaje C.

2. Por medio de las pruebas realizadas, se observa que el diseño del circuito de acoplamiento de la señal cardíaca funciona correctamente.

Para que el ECG no muestre pequeñas alteraciones, el paciente debe de estar en estado de reposo debido a la sensibilidad del circuito.

3. Para poder realizar la adquisición de datos por medio del puerto externo es necesario diseñar un convertidor de voltaje de 5V a 2.5V después del convertidor Analógico – Digital (ADC), debido a que el voltaje máximo en el puerto de la tarjeta NEEK es de 2.5V.

4. Para contrarrestar el problema de ruido de la señal cardíaca hay que colocar retardos de tiempo en el código de la adquisición de datos para evitar que el puerto externo tome como valor de entrada los niveles de voltajes provenientes del ruido, lo que nos da una mejor visualización de la señal en la LCD de la tarjeta NEEK.

RECOMENDACIONES

1. Debido a la amplitud de las señales biométricas que oscilan entre 0.5mV a 4mV, al ruido proveniente de diferentes fuentes (músculos, electrodos, cables), es necesario utilizar un amplificador de instrumentación cuya característica son alta impedancia de entrada y un alto rechazo al modo común (CMRR).

2. Se recomienda realizar el PCB bajo las normas de diseño para disminuir los efectos del ruido electromagnético.

3. Es necesario que se desinfecte la zona donde se colocan los electrodos, para remover impurezas de la piel debido a que disminuyen la calidad de la señal por la salinidad del sudor. Es importante que el paciente este en reposo cuando se esté realizando la prueba.

BIBLIOGRAFÍA

[1] Jorge Rodríguez Araújo, “Estudio del Procesador Nios II”,http://es.scribd.com/doc/28358833/Estudio-del-microprocesador-Nios-II, Octubre 2013

[2] Fecha de Consulta: Junio 2011. Sistema de conducción del corazón, http://www.bluejaygallery.com/download/Corazon.pdf, Octubre 2013

[3] Texas Heart Institute, Centro de Información Cardiovascular, Anatomía delcorazón,http://www.texasheartinstitute.org/HIC/anatomy_Esp/anato_sp.cfm, Octubre 2013

[4] La web del Electrocardiograma, Ritmo Cardíaco, http://www.my-ekg.com/como-leer-ekg/ritmo-cardiaco.html, Octubre 2013

[5] Electrocardiografía.es, Teoría del Electrocardiograma, Ondas componentes del ECG, http://www.electrocardiografia.es/ondas_ecg.html , Octubre 2013.

[6] Departamento de Ciencias Fisiológicas, Guías de Laboratorio, Electrocardiograma,http://fisiopuj.tripod.com/Guias/1_Electrocardiograma.pdf, Octubre 2013.

[7] Enric Pastor y Juan López, Universidad Politécnica de Cataluña, Introducción al sistema NIOS II, http://docencia.ac.upc.edu/EPSC/SED/Apuntes/SED_Intro_NIOS.pdf, Octubre 2013

[8] MikroElektronika, Microcontrolador PIC 16F887 – Microcontroladores PIC – Programación en C con ejemplos, http://www.mikroe.com/chapters/view/81/, Octubre 2013

[9] Juan Carlos Manosalvas, Módulo 3: Convertidor Análogo Digital, http://www.alos.5u.com/pic16f877/Modulo3.html, Octubre 2013

[10] Terasic Technologies Inc, Multimedia Touch Panel Daughter Board (MTDB) V2.0 User Manual.

[11] Guía médica familiar, http://www.explored.com.ec/guia/fas865.htm, Noviembre 2013

ANEXOS

ANEXO A

CÓDIGO FUENTE

```
#include "system.h"
#include "alt_video_display/alt_video_display.h"
#include "alt_touchscreen/alt_touchscreen.h"
#include "alt_tpo_lcd/alt_tpo_lcd.h"
#include "fonts.h"
#include "stdlib.h"
#include <string.h>
#include <unistd.h>
#include "sys/alt_irq.h"
#include "altera_avalon_spi_regs.h"
#include "altera_avalon_pio_regs.h"
#include <io.h>
#include "sys/alt_cache.h"
#include "graphics_lib/simple_graphics.h"
#include <malloc.h>
#include "altera_avalon_spi.h"
#include <stdio.h>
#include "sys/alt_alarm.h"
#include "alt_types.h"
#include <altera_avalon_sgdma.h>
#include <altera_avalon_sgdma_descriptor.h>
#include <altera_avalon_sgdma_regs.h>
#include <altera_up_sd_card_avalon_interface.h>
#define WHITE_16                0xFFDF
#define EXT_IO_BASE 0xc002680
#define COMPLEX_BASE 0xc002600
#define LCD_I2C_SDAT_BASE 0xc002900
#define                  LCD_I2C_SCL_BASE                  0xc002980
```

```
#define LCD_I2C_EN_BASE 0xc002a00
#define ALT_VIDEO_DISPLAY_USE_HEAP -1
char esc_cedula[22]="CEDULA DEL PACIENTE: ";
#define TOUCH_PANEL_SPI_BASE 0xc002100
#define TOUCH_PANEL_SPI_IRQ_INTERRUPT_CONTROLLER_ID 0
#define TOUCH_PANEL_PEN_IRQ_N_BASE 0xc002700
#define TOUCHSCREEN_SAMPLE_RATE 50
#define TOUCH_PANEL_PEN_IRQ_N_FREQ 60000000u
#define TOUCH_PANEL_SPI_IRQ 16
#define ALTERA_UP_SD_CARD_AVALON_INTERFACE_0_BASE 0xc001800
#define                      ALTERA_UP_SD_CARD_AVALON_INTERFACE_0_NAME
"/dev/Altera_UP_SD_Card_Avalon_Interface_0"
// DECLARACION DE FUNCIONES
void VerificarSD(alt_video_display *);     //Función que verifica el
estado de la tarjeta SD y crea el archivo en Excel
int pru=0;
short int base;
char cedula[10] = "";
alt_up_sd_card_dev *device_reference = NULL;
bool a;
void VerificarSD(alt_video_display * display )
{
int connected = 0;
int i=0;
char cedula2[8]="";
char archivo[13]="";
char extension='.';
char extension1 ='x';
char extension2 ='l';
char extension3 ='s';

        // verifica si existe el HW para leer/escribir memoria SD
```

```
 device_reference =
alt_up_sd_card_open_dev("/dev/Altera_UP_SD_Card_Avalon_Interface_0")
;
        if (device_reference != NULL)
        {
                while(1) //Lazo infinito esperando que inserte un SD
                {
                        //Verifica si existe una memoria SD
insertada en el slot
                        if ((connected == 0) &&
(alt_up_sd_card_is_Present()))
                        {
                                //presenta en la LCD Mensaje
                                printf("\ntajeta OK\n");
                                usleep(50000);
                                //Verifica formato de la SD
                                if (alt_up_sd_card_is_FAT16())
                                {
                                        printf("\nfat16 OK\n");
                                        usleep(50000);
                                        for (i = 0; i<8; i++)

{
                                                printf ("\n %c
",cedula[i]);
                                                sprintf
(cedula2,"%s%c",cedula2,cedula[i]);

}
// Crea el archivo donde se almacena la Informacion
sprintf(archivo,"%s%c%c%c%c",cedula2,extension,extension1,extension2
,extension3);
     base = alt_up_sd_card_fopen(archivo, true);
```

```
  printf("\nbase = %d \n",base);

//a=alt_up_sd_card_fclose(base);
                                        break;
                                }
                                else
                                {
                                        printf("\nfat16 ERROR\n");
                                        usleep(500000);
                                }
                                connected = 1;
                        }
                        else    if    ((connected    ==    0)    &&
(alt_up_sd_card_is_Present() == false))
                        {
                                printf("\nSIN TARJETA\n");
                                vid_draw_box        (290,350,460,390,
0x9C0036, 1, display);
                                vid_print_string(355,  360,  0xFFFFF0,
cour10_font, display, "POR FAVOR");
                                vid_print_string(310,  375,  0xFFFFF0,
cour10_font, display, "INGRESE LA TARJETA");
                                usleep(500000);
                                connected = 0;
                        }
                }
        }
return;
}
void delay(int time)
{
 int i, j, k;
 for(i=0; i<=time; i++)
 for(j=0; j<=2000; j++) k = 1;
}
```

```
int boton(alt_video_display * display, char string [],
                                int h_start,
                                int v_start,
                                int h_end,
                                int v_end,
                                int h_char,
                                int v_char)
{
        vid_draw_box   (h_start,v_start,h_end,v_end,   0x9C0036,   1,
display);
        vid_print_string(h_char,   v_char,   0xFFFFF0,   cour10_font,
display, string);
        return (0);
}
int  num_cedula  (alt_video_display  *  display,  alt_touchscreen  *
screen,
                alt_touchscreen_scaled_pen_data pen_data)
{
int numero;
int cont_cedula=0;
int press_1=0;
int press_2=0;
int borrar =0;
int i=0;
int ok =0;
//char vacio="";
//sprintf (cedula,"%c","");
do
          {
           alt_touchscreen_get_pen(screen,
            (&pen_data.pen_down),
```

```
            (&pen_data.x),
            (&pen_data.y));
          press_1=0;
          press_2=0;
          borrar =0;
          i=0;
          if (pen_data.pen_down ==1){

if(pen_data.x>=570&&pen_data.x<=630&&pen_data.y>=370&&pen_data.y<=43
0 && cont_cedula <10) // 0
   {
                  press_1 = 1;
                  numero =0;
                  cont_cedula ++;
vid_draw_box (570,370 ,630,430, 0xCCCCCC, 1, display);
vid_print_string(595, 395, 0, cour10_font, display, "0");
delay (2000);
vid_draw_box (570,370 ,630,430, 0x9C0036, 1, display);
vid_print_string(595, 395, 0xFFFFF0, cour10_font, display, "0");
   }
if(pen_data.x>=480&&pen_data.x<=540&&pen_data.y>=280&&pen_data.y<=34
0 && cont_cedula <10) // 1
                                                                    {
                                    press_1 = 1;
                                    numero =1;
                                    cont_cedula ++;
vid_draw_box (480,280 ,540,340, 0xCCCCCC, 1, display);
vid_print_string(505, 305, 0, cour10_font, display, "1");
delay (2000);
```

```
                                                vid_draw_box    (480,280
,540,340, 0x9C0036, 1, display);
                                               vid_print_string(505,
305, 0xFFFFF0, cour10_font, display, "1");
                                                                        }
if(pen_data.x>=570&&pen_data.x<=630&&pen_data.y>=280&&pen_data.y<=34
0 && cont_cedula <10) // 2

      {
                                     press_1 = 1;
                                    numero =2;
                                    cont_cedula ++;
vid_draw_box (570,280 ,630,340, 0xCCCCCC, 1, display);
vid_print_string(595, 305, 0, cour10_font, display, "2");
delay (2000);
vid_draw_box (570,280 ,630,340, 0x9C0036, 1, display);

vid_print_string(595,  305,  0xFFFFF0,  cour10_font,  display,  "2");
      }
if(pen_data.x>=660&&pen_data.x<=720&&pen_data.y>=280&&pen_data.y<=34
0 && cont_cedula <10) // 3
{
                                   press_1 = 1;
                                   numero =3;
                                   cont_cedula ++;
vid_draw_box (660,280 ,720,340, 0xCCCCCC, 1, display);
vid_print_string(685, 305, 0, cour10_font, display, "3");
delay (2000);
vid_draw_box    (660,280    ,720,340,    0x9C0036,    1,    display);
```

```
vid_print_string(685, 305, 0xFFFFF0, cour10_font, display, "3");
}
if(pen_data.x>=480&&pen_data.x<=540&&pen_data.y>=190&&pen_data.y<=250          &&          cont_cedula          <10)          //          4
{
                                    press_1 = 1;
                                   numero =4;
                                   cont_cedula ++;
                                   vid_draw_box   (480,190   ,540,250,
0xCCCCCC, 1, display);
                                           vid_print_string(505,  215,
0, cour10_font, display, "4");
                                  delay (2000);
                                    vid_draw_box     (480,190,540,250,
0x9C0036, 1, display);
                         vid_print_string(505,215,          0xFFFFF0,
cour10_font,                    display,                    "4");
}
if(pen_data.x>=570&&pen_data.x<=630&&pen_data.y>=190&&pen_data.y<=250          &&          cont_cedula          <10)          //          5
{
                 press_1 = 1;
                 numero =5;
                 cont_cedula ++;
                 vid_draw_box   (570,190   ,630,250,   0xCCCCCC,   1,
display);
                vid_print_string(595,  215,  0,  cour10_font,  display,
"5");
                delay (2000);
                vid_draw_box    (570,190,630,250,    0x9C0036,    1,
display);
                 vid_print_string(595,215,   0xFFFFF0,   cour10_font,
display,                                                    "5");
```

```
}
if(pen_data.x>=660&&pen_data.x<=720&&pen_data.y>=190&&pen_data.y<=25
0 && cont_cedula <10) // 6
{
                         press_1 = 1;
                        numero =6;
                        cont_cedula ++;
                        vid_draw_box  (660,190  ,720,250,  0xCCCCCC,
1, display);
                        vid_print_string(685,  215,  0,  cour10_font,
display, "6");
                        delay (2000);
                vid_draw_box    (660,190,720,250,    0x9C0036,    1,
display);
                        vid_print_string(685,215,          0xFFFFF0,
cour10_font,                    display,                    "6");
}
if(pen_data.x>=480&&pen_data.x<=540&&pen_data.y>=100&&pen_data.y<=16
0          &&          cont_cedula          <10)          //          7
{
                         press_1 = 1;
                numero =7;
                cont_cedula ++;
vid_draw_box (480,100 ,540,160, 0xCCCCCC, 1, display);
vid_print_string(505, 125, 0, cour10_font, display, "7");
delay (2000);
vid_draw_box (480,100,540,160, 0x9C0036, 1, display);
vid_print_string(505,125, 0xFFFFF0, cour10_font, display, "7");
```

```
}
if(pen_data.x>=570&&pen_data.x<=630&&pen_data.y>=100&&pen_data.y<=16
0          &&          cont_cedula          <10)          //          8
{
                press_1 = 1;
                numero =8;
                cont_cedula ++;
vid_draw_box (570,100 ,630,160, 0xCCCCCC, 1, display);
vid_print_string(595, 125, 0, cour10_font, display, "8");
delay (2000);
vid_draw_box (570,100,630,160, 0x9C0036, 1, display);
vid_print_string(595,125, 0xFFFFF0, cour10_font, display, "8");
}
if(pen_data.x>=660&&pen_data.x<=720&&pen_data.y>=100&&pen_data.y<=16
0          &&          cont_cedula          <10)          //          9
{
                press_1 = 1;
                numero =9;
                cont_cedula ++;
vid_draw_box (660,100 ,720,160, 0xCCCCCC, 1, display);
vid_print_string(685, 125, 0, cour10_font, display, "9");
delay (2000);
vid_draw_box (660,100,720,160, 0x9C0036, 1, display);
 vid_print_string(685,125, 0xFFFFF0, cour10_font, display, "9");

}
```

```
if(pen_data.x>=660&&pen_data.x<=7200&&pen_data.y>=370&&pen_data.y<=4
30          &&          cont_cedula          ==10)          //          ok
{
                                    ok = 1;
                                    cont_cedula ++;
vid_draw_box (660,370 ,720,430, 0xCCCCCC, 1, display);
vid_print_string(488, 395, 0, cour10_font, display, "OK");
delay (2000);
vid_draw_box (660,370,720,430, 0x9C0036, 1, display);
vid_print_string(488,395,  0xFFFFF0,  cour10_font,  display,  "OK");
}
if(pen_data.x>=480&&pen_data.x<=540&&pen_data.y>=370&&pen_data.y<=43
0)                                  //                                  BORRAR
{
                            char cedula2[10]= "";
                            press_1 = 1;
                            borrar = 1;
              vid_draw_box (480,370 ,540,430, 0xCCCCCC, 1, display);
                    vid_print_string(488, 395, 0, cour10_font, display,
"BORRAR");
                    if (cont_cedula >= 0)
                                        {
                   for (i = 0; i<cont_cedula-1; i++)
                                {
                           printf ("\n %c ",cedula[i]);
                           sprintf (cedula2,"%s%c",cedula2,cedula[i]);
                                                    }
  sprintf (cedula,"%s",cedula2);
                            printf(" \n cedula = %s ",cedula);
                                                }
                            delay (2000);
vid_draw_box (480,370,540,430, 0x9C0036, 1, display);
vid_print_string(488,395, 0xFFFFF0, cour10_font, display, "BORRAR");
```

```
   if(cont_cedula>0)
                          cont_cedula --;

}

                  if (press_1==1){
                          if (borrar ==0&&cont_cedula <=10){
 sprintf (cedula,"%s%d",cedula,numero);
 vid_draw_box (150,180 ,240,200, 0xCCCCCC, 1, display);
 vid_print_string(155, 185, 0, cour10_font, display, cedula);
                                        }
                          if (borrar ==1)
                                        {
 vid_draw_box (150,180 ,240,200, 0xCCCCCC, 1, display);
 vid_print_string(155, 185, 0, cour10_font, display, cedula);
                                        }
                          }
                                                          }
         }
         while (cont_cedula<=10);
return cedula;
}
int  button_press(alt_video_display  *  display,  alt_touchscreen  *
screen,               alt_touchscreen_scaled_pen_data pen_data, char
string[],
                int h_start,
                int v_start,
                int h_end,
                int v_end,
                int h_char,
                int v_char)
{
        int press=0;
        printf("paso1");
vid_draw_box (h_start, v_start, h_end,v_end,0x9C0036, 1, display);
        printf("paso2");
```

```
vid_print_string(h_char,  v_char,  0xFFFFF0,  cour10_font,  display,
string);
         printf("paso3");
         do
         {
                 //printf("paso4");
          alt_touchscreen_get_pen(screen,
           (&pen_data.pen_down),
           (&pen_data.x),
           (&pen_data.y));
//printf("\nx = %d", pen_data.x);
//printf("\ny = %d", pen_data.y);
//printf("\npen_down = %d", pen_data.pen_down);

if(pen_data.x>=h_start&&pen_data.x<=h_end&&pen_data.y>=v_start&&pen_
data.y<=v_end)
           {
                    press=1;
vid_draw_box (h_start,v_start ,h_end,v_end, 0xCCCCCC, 1, display);
vid_print_string(h_char, v_char, 0, cour10_font, display, string);
delay(2000);           }
         }
         while (press==0);
vid_draw_box (h_start,v_start,h_end,v_end, 0x9C0036, 1, display);
vid_print_string(h_char,  v_char,  0xFFFFF0,  cour10_font,  display,
string);
          return press;
        }
int presentacion(alt_video_display * display)
{
        alt_video_display_clear_screen(display,0);
        vid_print_string(240,  39,  0XFFFFFF,cour10_font  ,display,
"ESCUELA SUPERIOR POLITECNICA DEL LITORAL");
        delay(1000);
        vid_print_string(248,  74,  0XFFFFFF,  cour10_font  ,display,
"FACULTAD DE ELECTRICIDAD Y COMPUTACION");
        delay(1000);
```

```
        vid_print_string(312,  109,0XFFFFFF,  cour10_font  ,display,
"MATERIA DE GRADUACION:");

        delay(1000);

        vid_print_string(240,  144,  0XFFFFFF,cour10_font  ,display,
"MICROPROCESADORES PROGRAMABLES EMBEBIDOS");

        delay(1000);

        vid_print_string(220,  179,  0XFFFFFF,  cour10_font  ,display,
"ELECTROCARDIOGRAFO PORTATIL BASADO EN NIOS II");

        delay(1000);

        vid_print_string(356,  214,  0XFFFFFF,  cour10_font  ,display,
"INTEGRANTES");

        delay(1000);

        vid_print_string(308,  249,  0XFFFFFF,  cour10_font  ,display,
"CINDY MEDINA COLLAHUAZO");

        delay(1000);

        vid_print_string(320,  284,  0XFFFFFF,  cour10_font  ,display,
"PAUL QUEZADA OQUENDO");

        delay(1000);

        vid_print_string(324,  319,  0XFFFFFF,  cour10_font,display,
"DIRECTOR DE MATERIA");

        delay(1000);

        vid_print_string(280,  354,  0XFFFFFF,  cour10_font  ,display,
"MSC. RONALD PONGUILLO INTRIAGO");

        delay(1000);

        vid_print_string(340,  389,  0XFFFFFF,  cour10_font  ,display,
"PERIODO LECTIVO");

        delay(1000);

        vid_print_string(356,  424,  0XFFFFFF,  cour10_font  ,display,
"2013 - 2014");

        delay(1000);

        return(0);

}
```

```
int coordenada(alt_video_display * display,alt_touchscreen * screen,
alt_touchscreen_scaled_pen_data pen_data)
{
int coordenada;
int press;
int ccmc;
 ccmc=50;
press =0;
do
         {
          alt_touchscreen_get_pen(screen,
           (&pen_data.pen_down),
           (&pen_data.x),
           (&pen_data.y));

if(pen_data.x>=250&&pen_data.x<=420&&pen_data.y>=150&&pen_data.y<=17
0)
           {
                   press=1;
                   coordenada =1;
           vid_draw_box  (250,150  ,420,170,  0xCCCCCC,  1,  display);
vid_print_string(255,   155,   0,   cour10_font,   display,   "INICIAR
ELECTROGRAFO");
           delay(1000);
           vid_draw_box (250,150 ,420,170, 0x9C0036, 1, display);
           vid_print_string(255,    155,    0x9C0036,    cour10_font,
display, "INICIAR ELECTROGRAFO");
           }
if(pen_data.x>=250&&pen_data.x<=300&&pen_data.y>=300&&pen_data.y<=32
0)
                       {
                               press=1;
                               coordenada =8;
                       vid_draw_box  (250,300  ,300,320,  0xCCCCCC,  1,
display);
                       vid_print_string(255,   305,   0,   cour10_font,
display, "SALIR");
                       delay(1000);
```

```
                        vid_draw_box  (250,300  ,300,320,  0x9C0036,  1,
display);
                        vid_print_string(255,305,             0x9C0036,
cour10_font, display, "SALIR");
                        }
          }
          while (press ==0);
return coordenada;
}
int main()
{
  alt_tpo_lcd s_lcd;          // lcd configurator struct
  alt_tpo_lcd *lcd = &s_lcd; // struct pointer
  volatile int * dat_io = (int*)(EXT_IO_BASE);
  volatile int * complejo = (int*)(COMPLEX_BASE);
  int cont_complejo=0;
  int datacomplejo=0;
  int complejopasado=0;
  int data=0;
  int result;
  int x;
  int y=0;
  int horiz_start=0;
  int vert_start=292;
  char dato[3] = "";
  int vert_end=0;
  int cont=0;
  int horiz_end=5;
  char vmedio[3]=" 40";
  //int result;
  int pen=0;
  char saludo1[40]="ESCUELA SUPERIOR POLITECNICA DEL LITORAL";
  char saludo2[45]="ELECTROCARDIOGRAFO PORTATIL BASADO EN NIOS II";
  char ScoreString[10];
  int coord;
```

```
  //volatile int * ext_io = (int*)(EXT_IO_BASE);
  // volatile int * leds = (int*)(LED_PIO_BASE);
  int grabar=0;
  alt_touchscreen_scaled_pen_data pen_data;
  alt_video_display * display;  // LCD display structure
  //alt_touchscreen_axis_swap_choice swap_xy;
  alt_touchscreen * screen;
   int subida=0;
   int bajada=0;
//Pointer to our font table
char* cour10_font = &cour10_font_array[0][0];

  // Specify base addresses of the each communication bus PIO,
  // as defined in system.h

  lcd->scen_pio = LCD_I2C_EN_BASE;
  lcd->scl_pio  = LCD_I2C_SCL_BASE;

  lcd->sda_pio  = LCD_I2C_SDAT_BASE;

  result = alt_tpo_lcd_init(lcd, 800, 480);
  if(result) {
    printf("LCD control register initialization failed\n");
  } else {
    printf("LCD control registers initilized\n ");
  }
  display = alt_video_display_init( "/dev/lcd_sgdma",
                                    800,
                                    480,
                                    32,
            ALT_VIDEO_DISPLAY_USE_HEAP,
            ALT_VIDEO_DISPLAY_USE_HEAP,
                                    1);
int frec_cardiaca=0;
char frecuencia_sd[3]="";
```

```
char frec_card[22]="FRECUENCIA CARDIACA: ";
char anomalia_cardiaca[19]="ANOMALIA CARDIACA: ";
char bradicardia[11]="BRADICARDIA";
char taquicardia[11]="TAQUICARDIA";
char ninguna[7]= "NINGUNA";
char xminuto[18]="latidos por minuto";
  //colour_swap();
int salir=0;
  //ALT_TOUCHSCREEN_SWAP_XY
  alt_video_display_clear_screen(display,0);
 alt_touchscreen_init
(screen,TOUCH_PANEL_SPI_BASE,TOUCH_PANEL_SPI_IRQ,TOUCH_PANEL_PEN_IRQ
_N_BASE,TOUCHSCREEN_SAMPLE_RATE,ALT_TOUCHSCREEN_SWAP_XY);
 alt_touchscreen_calibrate_upper_right (screen,
                      3946,    3849,     // ADC readings
                       799,       0   ); // pixel coords
   alt_touchscreen_calibrate_lower_left  (screen,
                      132,     148,      // ADC readings
                        0,     479   );  // pixel coords
alt_touchscreen_event_loop_update (screen);
pen=0;
 do{
 //inicio
presentacion(display);
button_press(display,screen,
pen_data,"SIGUIENTE",680,440,760,460,685,445);
delay (1000);
 //menu
 alt_video_display_clear_screen(display,0);
vid_print_string(220,15,WHITE_16,cour10_font,display,"ELECTROCARDIOG
RAFO PORTATIL BASADO EN NIOS II");
 boton(display,"INICIAR ELECTROGRAFO",250,150,420,170,255,155);
 boton(display,"SALIR",250,300,300,320,255,305);
 coord=coordenada(display,screen,pen_data);
```

```
switch (coord)
        {
                case 1: //Iniciar EKG
                {
                    alt_video_display_clear_screen(display,0);
vid_print_string(220,15,WHITE_16,cour10_font,display,"ELECTROCARDIOG
RAFO PORTATIL BASADO EN NIOS II");
vid_draw_box  (450,70 ,750,460, 0xFFFFFF, 1, display);
vid_draw_box  (450,70 ,750,85, 0xFF0000, 1, display);
vid_draw_box  (450,70 ,465,460, 0xFF0000, 1, display);
vid_draw_box  (735,70 ,750,460, 0xFF0000, 1, display);
vid_draw_box  (450,445 ,750,460, 0xFF0000, 1, display);
vid_draw_box  (450,70 ,465,85,0xA00000, 1, display);
vid_draw_box  (735,70 ,750,85,0xA00000, 1, display);
vid_draw_box  (735,445 ,750,460,0xA00000, 1, display)
vid_draw_box  (450,445 ,465,460,0xA00000, 1, display);
boton(display,"7",480,100,540,160,505,125);
boton(display,"8",570,100,630,160,595,125);
boton(display,"9",660,100,720,160,685,125);
boton(display,"4",480,190,540,250,505,215);
boton(display,"5",570,190,630,250,595,215);
boton(display,"6",660,190,720,250,685,215);
boton(display,"3",660,280,720,340,685,305);
boton(display,"1",480,280,540,340,505,305);
boton(display,"2",570,280,630,340,595,305);
boton(display,"0",570,370,630,430,595,395);
boton(display,"BORRAR",480,370,540,430,488,395);
boton(display,"OK",660,370,720,430,680,395);
boton(display,"INGRESAR NUMERO DE CEDULA",100,150,320,170,105,155);
                                                    vid_draw_box
(150,180 ,240,200,0xCCCCCC, 1, display);
num_cedula (display,screen,pen_data);
alt_video_display_clear_screen(display,0);
vid_print_string(220,15,WHITE_16,cour10_font,display,"ELECTROCARDIOG
RAFO        PORTATIL        BASADO        EN        NIOS        II");
//vid_draw_horiz_line( 3,795,250,WHITE_16,display );
```

```
vid_print_string(220,80,WHITE_16,cour10_font,display,ScoreString);
                  salir=0;
grabar =0;
do
{
        boton(display,"SALIR",120,370,280,440,170,395);
alt_touchscreen_get_pen(screen,
(&pen_data.pen_down),
(&pen_data.x),
 (&pen_data.y));
if (pen_data.pen_down ==1){
if(pen_data.x>=100&&pen_data.x<=300&&pen_data.y>=350&&pen_data.y<=46
0 ) {
salir =1;
           vid_draw_box (100,350,300,460, 0xCCCCCC, 1, display);
vid_print_string(570, 395, 0, cour10_font, display, "SALIR");
delay (2000);
vid_draw_box (100,350,300,460, 0x9C0036, 1, display);
vid_print_string(570, 395, 0xFFFFF0, cour10_font, display, "SALIR");
}
}//
cont =cont+3;
pru=pru+3;
boton(display,"GRABAR",520,370,680,440,570,395);
alt_touchscreen_get_pen(screen,
(&pen_data.pen_down),
(&pen_data.x),
(&pen_data.y));

if                      (pen_data.pen_down                      ==1){
if(pen_data.x>=500&&pen_data.x<=700&&pen_data.y>=350&&pen_data.y<=46
0                                )                                   {
grabar =1;
vid_draw_box     (500,350,700,460,     0xCCCCCC,     1,     display);
vid_print_string(570, 395, 0, cour10_font, display, "GRABAR");
delay (2000);
vid_draw_box (500,350,700,460, 0x9C0036, 1, display);
vid_print_string(570, 395, 0xFFFFF0, cour10_font, display,
"GRABAR");
}
}
vid_print_string(220,15,WHITE_16,cour10_font,display,"ELECTROCARDIOG
RAFO PORTATIL BASADO EN NIOS II");
```

```
if (cont==798)
{
vid_print_string(220,15,WHITE_16,cour10_font,display,"ELECTROCARDIOG
RAFO PORTATIL BASADO EN NIOS II");
// vid_draw_horiz_line( 3,795,250,WHITE_16,display );
alt_video_display_clear_screen(display,0);
cont =0;
horiz_start=0;
horiz_end=3;
}
usleep(4000);
data = *(dat_io);
printf(" %d",data);
sprintf(dato, "%d",data);
vert_end = 256 - (3*data)+100;
vid_draw_line(horiz_start,      vert_start,      horiz_end,vert_end,
1,0xFFFFFF, display);
vert_start=vert_end;
horiz_start=horiz_end;
horiz_end=horiz_start+3;
x=0;
}
while((grabar==0)&&(salir==0));
pru=0;
while((grabar==1)&&(salir==0)&&(pru<3990)){
cont =0;
pru=0;
horiz_start=0;
horiz_end=3;
VerificarSD (display); //VERIFICA SD
alt_video_display_clear_screen(display,0);
alt_up_sd_card_write(base,0x09);//salto de campo
alt_up_sd_card_write(base,0x09);//salto de campo
alt_up_sd_card_write(base,0x09);//salto de campo
do
{
alt_up_sd_card_write(base, saludo1[y]);
usleep(2);
```

```
y=y+1;
}while(y<40);
y=0;
usleep(20);
alt_up_sd_card_write(base,0x09);//salto de campo
alt_up_sd_card_write(base,0x09);//salto de campo
alt_up_sd_card_write(base,0x09);//salto de campo
alt_up_sd_card_write(base,0x09);//salto de campo
alt_up_sd_card_write(base,0x09);//salto de campo
do
{
alt_up_sd_card_write(base, esc_cedula[y]);
usleep(2);
y=y+1;
}while(y<21);
y=0;
usleep(20);
alt_up_sd_card_write(base,0x09);//salto de campo
alt_up_sd_card_write(base,0x09);//salto de campo
do
{
alt_up_sd_card_write(base, cedula[y]);
usleep(2);
y=y+1;
}while(y<11);
y=0;
usleep(20);
alt_up_sd_card_write(base,0x0D);//enter
alt_up_sd_card_write(base,0x09);//salto de campo
alt_up_sd_card_write(base,0x09);//salto de campo
alt_up_sd_card_write(base,0x09);//salto de campo
do
{
alt_up_sd_card_write(base, saludo2[y]);
usleep(2);
```

```
y=y+1;
}while(y<45);
y=0;
usleep(20);
alt_up_sd_card_write(base,0x0D);//enter
cont_complejo=0;
datacomplejo=0;
complejopasado =0;
do
{
//
vid_print_string(220,15,WHITE_16,cour10_font,display,"ELECTROCARDIOG
RAFO PORTATIL BASADO EN NIOS II");
cont =cont+3;
pru=pru+3;
alt_up_sd_card_write(base,0x09);//enter
//alt_up_sd_card_write(base,'1');
//alt_up_sd_card_write(base,0x09);//salto de campo
if (cont==798){
vid_print_string(220,15,WHITE_16,cour10_font,display,"ELECTROCARDIOG
RAFO PORTATIL BASADO EN NIOS II");
//vid_draw_horiz_line( 3,795,250,WHITE_16,display );
alt_video_display_clear_screen(display,0);
cont =0;
//data = *(ext_io);
//BUTTON_PIO_BIT_CLEARING_EDGE_REGISTER;
horiz_start=0;
horiz_end=3;
}
usleep(4000);
data = *(dat_io);

datacomplejo= *(complejo);
printf(" %d",data);
usleep(4000);
data = *(dat_io);
```

```
datacomplejo= *(complejo);
sprintf(dato, "%d",data);
usleep(20);
vert_end = 256 - (3*data)+100;
vid_draw_line(horiz_start,      vert_start,      horiz_end,vert_end,
1,0xFFFFFF, display);
vert_start=vert_end;
horiz_start=horiz_end;
horiz_end=horiz_start+3;
if(data>=60) subida =1;
if (data<60) bajada=1;
if ((subida==1)&&(bajada==1)){
        cont_complejo++;
        subida =0;
        bajada =0;

x=0;
do
{
alt_up_sd_card_write(base, dato[x]);
usleep(2);
x=x+1;
}while(x<3);
usleep(20);

/*  if ((datacomplejo==1)&&(complejopasado==0))
{
cont_complejo++;
}
complejopasado=datacomplejo;*/
}
while(pru<3990);
alt_up_sd_card_write(base,0x0D);//enter
alt_up_sd_card_write(base,0x0D);//enter
printf(" %d ",cont_complejo);
frec_cardiaca = cont_complejo * 2;
```

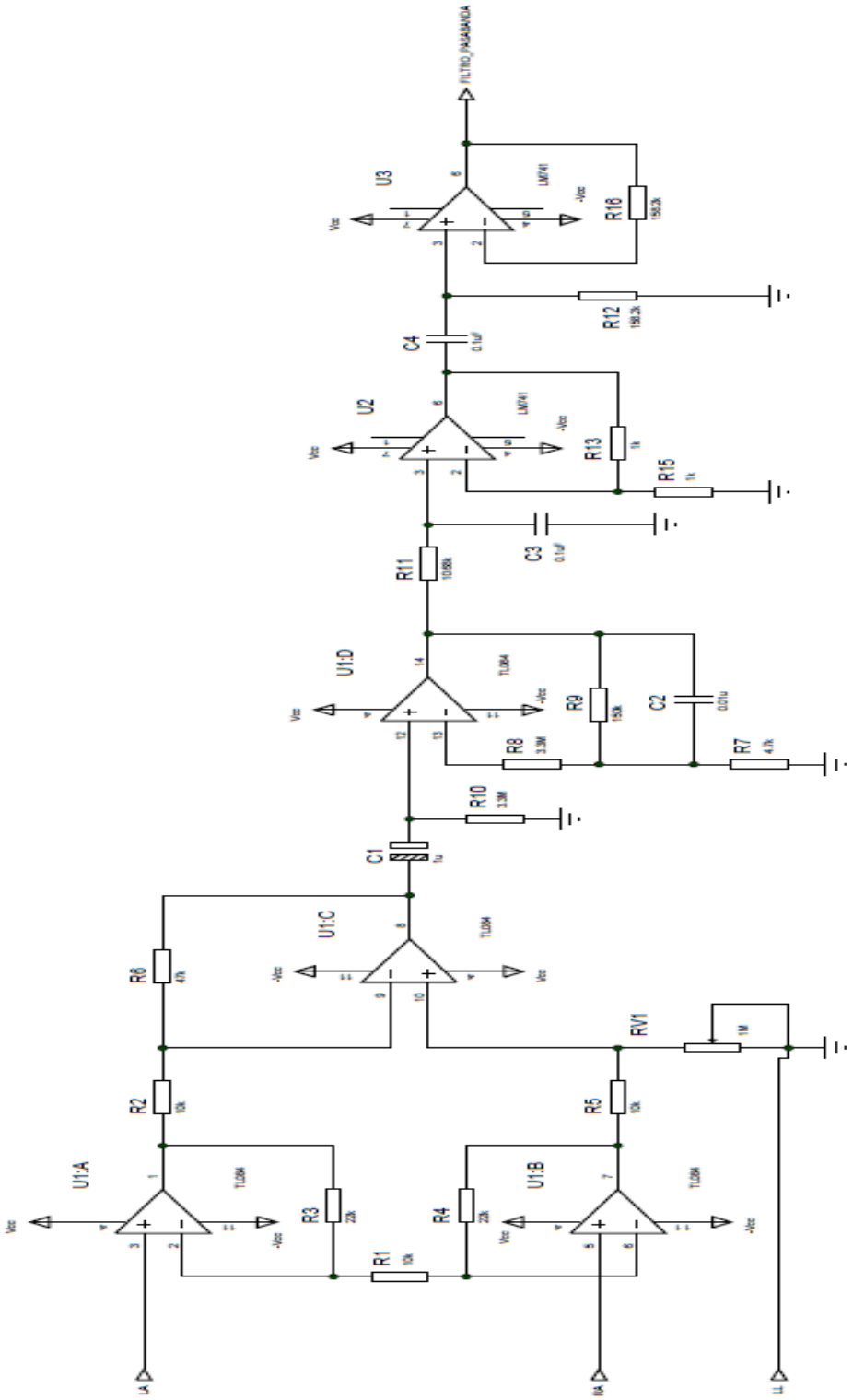
```
sprintf(frecuencia_sd, "%d",frec_cardiaca);

x=0;
do
{
alt_up_sd_card_write(base, dato[x]);
usleep(2);
x=x+1;
}while(x<3);
usleep(20);

/*  if ((datacomplejo==1)&&(complejopasado==0))
{
cont_complejo++;
}
complejopasado=datacomplejo;*/
}
while(pru<3990);
alt_up_sd_card_write(base,0x0D);//enter
alt_up_sd_card_write(base,0x0D);//enter
printf(" %d ",cont_complejo);
frec_cardiaca = cont_complejo * 2;
sprintf(frecuencia_sd, "%d",frec_cardiaca);


x=0;
do
{
        alt_up_sd_card_write(base, frec_card[x]);
usleep(2);
x=x+1;
}while(x<22);
usleep(20);
alt_up_sd_card_write(base,0x09);//salto de campo
alt_up_sd_card_write(base,0x09);//salto de campo
x=0;
```

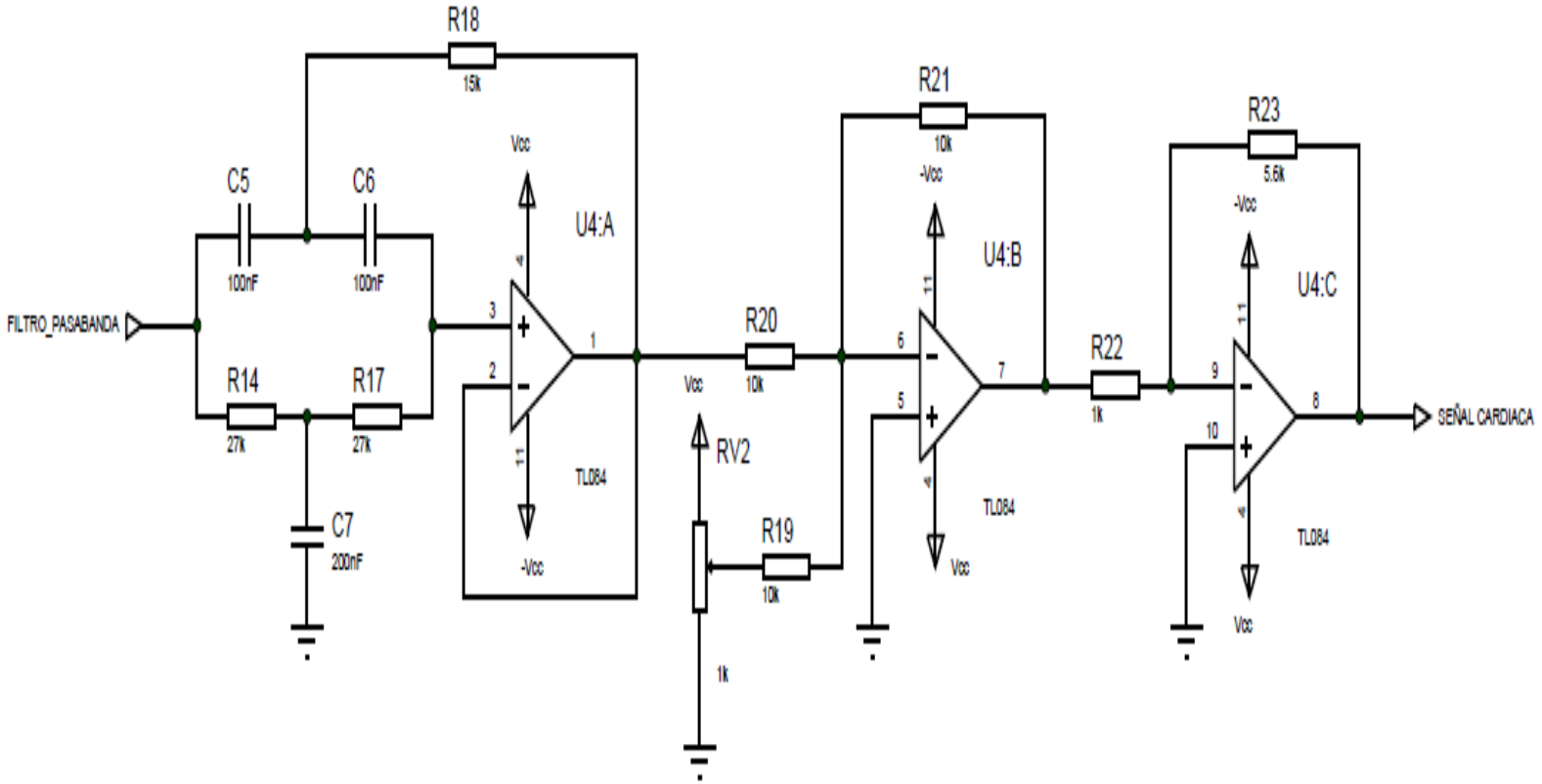
```
do
{
        alt_up_sd_card_write(base, frecuencia_sd[x]);
usleep(2);
x=x+1;
}while(x<3);
usleep(20);
alt_up_sd_card_write(base,0x09);//salto de campo
x=0;
do
{
alt_up_sd_card_write(base, xminuto[x]);
usleep(2);
x=x+1;
}while(x<18);
alt_up_sd_card_write(base,0x0D);//enter
usleep(20);
alt_up_sd_card_write(base,0x0D);//enter
x=0;
do
{
        alt_up_sd_card_write(base, anomalia_cardiaca[x]);
        usleep(2);
        x=x+1;
}while(x<19);
alt_up_sd_card_write(base,0x09);//salto de campo
alt_up_sd_card_write(base,0x09);//salto de campo
x=0;
if (frec_cardiaca >100){
        do
         {

alt_up_sd_card_write(base, taquicardia[x]);
usleep(2);
        x=x+1;
```

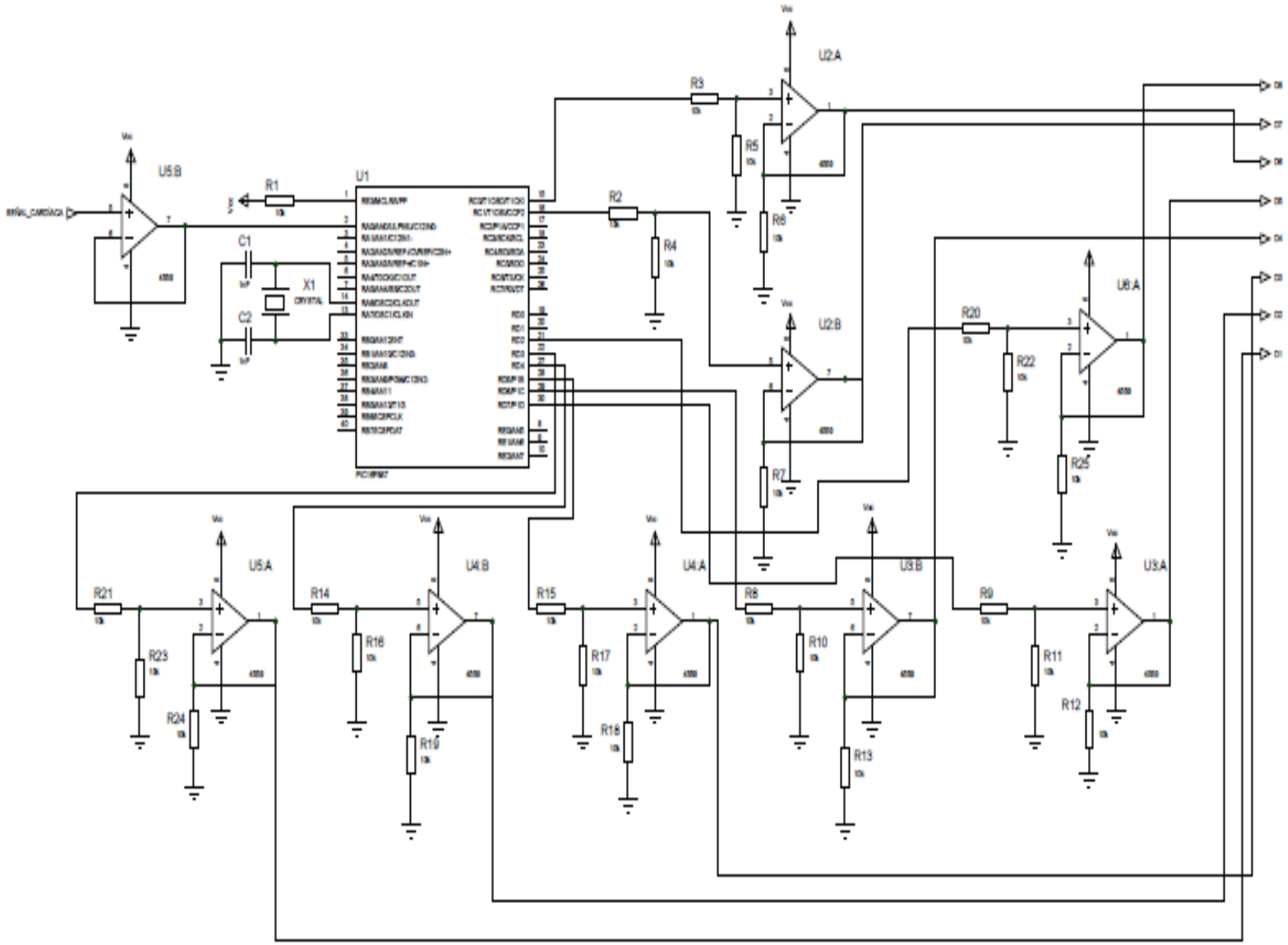
```
}while(x<11);
                                        }
if (frec_cardiaca <60){
do
{
        alt_up_sd_card_write(base, bradicardia[x]);
usleep(2);
x=x+1;
}while(x<11);
        }
if ((frec_cardiaca >=60) && (frec_cardiaca<=100)){
do
{
        alt_up_sd_card_write(base, ninguna[x]);
usleep(2);
x=x+1;
}while(x<7);
}
a=alt_up_sd_card_fclose(base);
alt_video_display_clear_screen(display,0);
vid_print_string(120,    200,    0xFFFFFF,    cour10_font,    display,
"FRECUENCIA CARDIACA: ");
vid_print_string(320,    200,    0xFFFFFF,    cour10_font,    display,
frecuencia_sd);
vid_print_string(360, 200, 0xFFFFFF, cour10_font, display, xminuto);
vid_print_string(120, 260, 0xFFFFFF, cour10_font, display, "ANOMALIA
CARDIACA: ");
if (frec_cardiaca >100){
vid_print_string(320,    260,    0xFFFFFF,    cour10_font,    display,
taquicardia);
vid_print_string(430, 260, 0, cour10_font, display, "TAQUICARDIA");
}
if (frec_cardiaca <60){
vid_print_string(320,    260,    0xFFFFFF,    cour10_font,    display,
bradicardia);
}
```

```
if ((frec_cardiaca >=60) && (frec_cardiaca<=100)){
vid_print_string(320,        260,        0xFFFFFF,        cour10_font,
display,"NINGUNA");

}
}
        usleep(5000000);

}
break;
case 2:
{
coord = 1;
}
break;
case 8: // PORTADA
{
break;
}
}
}
while (1);
return 1;
}
```

ANEXO B


FILTRO_PASABANDA
U3
LM741
Vcc
-Vcc
R18
150.2k
R12
150.2k
C4
0.1uF
U2
LM741
R13
1k
R15
1k
C3
0.1uF
R11
10.05k
U1:D
TL084
R9
150k
C2
0.01u
R8
3.3M
R7
4.7k
R10
3.3M
C1
1u
U1:C
TL084
R6
47k
RV1
1M
R2
10k
R5
10k
U1:A
TL084
R3
22k
R4
22k
U1:B
TL084
R1
10k
LA
HA
LL


R18
15k
C5
100nF
C6
100nF
FILTRO_PASABANDA
R14
27k
R17
27k
C7
200nF
Vcc
U4:A
TL084
-Vcc
R20
10k
Vcc
RV2
1k
R19
10k
R21
10k
-Vcc
U4:B
TL084
Vcc
R22
1k
R23
5.6k
-Vcc
U4:C
TL084
Vcc
SEÑAL CARDIACA


U1
U2:A
U2:B
U3:A
U3:B
U4:A
U4:B
U5:A
U5:B
U6:A
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15
R16
R17
R18
R19
R20
R21
R22
R23
R24
R25
C1
C2
X1
CRYSTAL
SEÑAL_CARDÍACA

Anexo C

4.12 General Purpose I/O Connector (Mode 1 and 2 only)

The MTDB v2.0 includes a standard GPIO header (J13). When the Mode Control Register in the MAX II device is set to 1, there are 8 pins available. When it is set to 2, there are two pins available (see "Level Translators and MAX II Mode Control Register" or Appendix). When the Mode Control Register is set to 0, the GPIO Header pins are disabled. Table 4.14 shows the pinout of General Purpose I/O Connector (Mode 1 and 2 Only) with HSMC connector.

Table 4.14. General Purpose I/O Connector (Mode 1 and 2 only) Pinouts.

HSMC Connector		Switch Pin No.		GPIO Header		
Signal Name	Pin No.	HSMC Connector Side Pin	Device Side Pin	Signal Name	Pin No.	Description
HC_TD_D2	62	U12.4	U12.17	DBG_D2	J13.5	GPIO data pin 2 (Mode Control Register = 1)
HC_TD_D3	66	U12.5	U12.16	DBG_D3	J13.6	GPIO data pin 3 (Mode Control Register = 1)
HC_TD_D4	68	U12.6	U12.15	DBG_D4	J13.7	GPIO data pin 4 (Mode Control Register = 1)
HC_TD_D5	72	U12.7	U12.14	DBG_D5	J13.8	GPIO data pin 5 (Mode Control Register = 1)
HC_TD_D6	74	U12.8	U12.13	DBG_D6	J13.9	GPIO data pin 6 (Mode Control Register = 1)
HC_TD_D7	78	U12.9	U12.12	DBG_D7	J13.10	GPIO data pin 7 (Mode Control Register = 1)
HC_TD_VS	84	U11.3	U11.12	DBG_D1	J13.3	GPIO data pin 1 (Mode Control Register = 1 or 2)
HC_TD_HS	86	U11.2	U11.13	DBG_D0	J13.1	GPIO data pin 0 (Mode Control Register = 1 or 2)

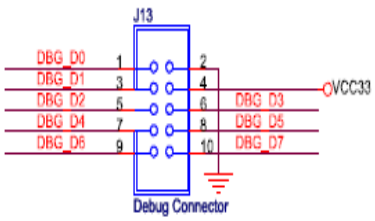


Figure 4.14. General Purpose I/O Connector Schematic.

Anexo D

MICROCHIP

PIC16F882/883/884/886/887

28/40/44-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

High-Performance RISC CPU:

- Only 35 instructions to learn:
  - All single-cycle instructions except branches
- Operating speed:
  - DC – 20 MHz oscillator/clock input
  - DC – 200 ns instruction cycle
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

Special Microcontroller Features:

- Precision Internal Oscillator:
  - Factory calibrated to ±1%
  - Software selectable frequency range of 8 MHz to 31 kHz
  - Software tunable
  - Two-Speed Start-up mode
  - Crystal fail detect for critical applications
  - Clock mode switching during operation for power savings
- Power-Saving Sleep mode
- Wide operating voltage range (2.0V-5.5V)
- Industrial and Extended Temperature range
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR) with software control option
- Enhanced low-current Watchdog Timer (WDT) with on-chip oscillator (software selectable nominal 268 seconds with full prescaler) with software enable
- Multiplexed Master Clear with pull-up/input pin
- Programmable code protection
- High Endurance Flash/EEPROM cell:
  - 100,000 write Flash endurance
  - 1,000,000 write EEPROM endurance
  - Flash/Data EEPROM retention: > 40 years
- Program memory Read/Write during run time
- In-Circuit Debugger (on board)

Low-Power Features:

- Standby Current:
  - 50 nA @ 2.0V, typical
- Operating Current:
  - 11 µA @ 32 kHz, 2.0V, typical
  - 220 µA @ 4 MHz, 2.0V, typical
- Watchdog Timer Current:
  - 1 µA @ 2.0V, typical

Peripheral Features:

- 24/35 I/O pins with individual direction control:
  - High current source/sink for direct LED drive
  - Interrupt-on-Change pin
  - Individually programmable weak pull-ups
  - Ultra Low-Power Wake-up (ULPWU)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (CV_{REF}) module (% of V_{DD})
  - Fixed voltage reference (0.6V)
  - Comparator inputs and outputs externally accessible
  - SR Latch mode
  - External Timer1 Gate (count enable)
- A/D Converter:
  - 10-bit resolution and 11/14 channels
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Enhanced Timer1:
  - 16-bit timer/counter with prescaler
  - External Gate Input mode
  - Dedicated low-power 32 kHz oscillator
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Enhanced Capture, Compare, PWM+ module:
  - 16-bit Capture, max. resolution 12.5 ns
  - Compare, max. resolution 200 ns
  - 10-bit PWM with 1, 2 or 4 output channels, programmable "dead time", max. frequency 20 kHz
  - PWM output steering control
- Capture, Compare, PWM module:
  - 16-bit Capture, max. resolution 12.5 ns
  - 16-bit Compare, max. resolution 200 ns
  - 10-bit PWM, max. frequency 20 kHz
- Enhanced USART module:
  - Supports RS-485, RS-232, and LIN 2.0
  - Auto-Baud Detect
  - Auto-Wake-Up on Start bit
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I^2C™ Master and Slave Modes with I^2C address mask

PIC16F882/883/884/886/887

Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	ECCP/ CCP	EUSART	MSSP	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)							
PIC16F882	2048	128	128	28	11	1/1	1	1	2	2/1
PIC16F883	4096	256	256	24	11	1/1	1	1	2	2/1
PIC16F884	4096	256	256	35	14	1/1	1	1	2	2/1
PIC16F886	8192	368	256	24	11	1/1	1	1	2	2/1
PIC16F887	8192	368	256	35	14	1/1	1	1	2	2/1