



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA

UNIDAD CULHUACAN

INGENIERÍA EN COMUNICACIONES Y ELECTRÓNICA

**Monitor de ritmo cardiaco para dispositivos Android
mediante un enlace Bluetooth**

TESIS

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMUNICACIONES Y ELECTRÓNICA**

PRESENTAN

**C. Pedro Enrique Velázquez Elizondo
C. Luis Javier Villagrán Valencia**

DIRECTORES

Ing. Fermín Valencia Figueroa

M. en C. Mario Ponce Flores



Ciudad de México 2016



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
UNIDAD CULHUACÁN
SUBDIRECCIÓN ACADÉMICA
Oficina de Titulación Profesional

Ciudad de México a 30 de septiembre del 2016

Trabajo Escrito para la Opción de Titulación

1.- Proyecto de Investigación	<input type="checkbox"/>	2.- Tesis	<input checked="" type="checkbox"/>	3.- Curricular	<input type="checkbox"/>
4.- Memoria de Experiencia Profesional	<input type="checkbox"/>	5.- Práctica Profesional	<input type="checkbox"/>		

Que como prueba escrita de su Examen Profesional para obtener el Título de Ingeniero:

1.- En Comunicaciones y Electrónica	<input checked="" type="checkbox"/>	2.- Mecánico	<input type="checkbox"/>	3.- En Computación	<input type="checkbox"/>
-------------------------------------	-------------------------------------	--------------	--------------------------	--------------------	--------------------------

Deberá (n) desarrollar El (C), LA (C), LOS (C.C.)

C. PEDRO ENRIQUE VELÁZQUEZ ELIZONDO

C. LUIS JAVIER VILLAGRÁN VALENCIA

Título del Trabajo:

“MONITOR DE RITMO CARDIACO PARA DISPOSITIVOS ANDROID MEDIANTE UN ENLACE
 BLUETOOTH”

Justificación del Trabajo:

La tecnología es una buena herramienta para enfocarla hacia la medicina, para resolver problemas que a simple vista son muy pequeños pero que tienen una consecuencia mayor a la que es notorio, la implementación de aparatos electrónicos inalámbricos brinda escalabilidad y compatibilidad.

En la actualidad en nuestro país no se tiene un conocimiento a fondo de todas las ventajas, beneficios y aplicaciones de las comunicaciones inalámbricas para la implementación en la medicina. Aunado a esto la falta de capacitación hacia la tecnología Bluetooth hacen que se sigan utilizando aparatos alámbricos que complican e incomodan el área de trabajo los cuales son de gran importancia dentro de un hospital. Por esta razón, el presente trabajo pretende establecer los requerimientos de hardware para poder realizar el monitoreo de ritmo cardiaco y software para realizar una comunicación inalámbrica mediante Bluetooth y poderlo implementar en una plataforma de Android de bajo consumo.

CAPITULADO

- I. ESTADO DEL ARTE
- II. MARCO TEÓRICO
- III. DISEÑO E IMPLEMENTACIÓN
- IV. PRUEBAS Y RESULTADOS

M. EN C. MARIO PONCE FLORES
 Primer Asesor

Vo. Bo.

ING. FERMÍN VALENCIA FIGUEROA
 Segundo Asesor

APROBADO:

ING. FELICIANO PRIMO ISIDRO CRUZ
 JEFE DE LA CARRERA DE I.C.E

ING. CARLOS AQUINO RUIZ
 SUBDIRECTOR ACADÉMICO (INTERINO)

Agradecimientos

Luis Javier Villagrán Valencia

A mis padres Javier y Silvia porque siempre me han brindado su total apoyo, por cuidarme siempre y demostrarme amor, por enseñarme los valores de la honestidad, respeto y trabajo. Todo esto y más ha sido gracias a ellos.

A mis tíos José Luis, Raúl, Rosa María, Sergio y Gonzalo que han sido un gran pilar en mi educación y en mi vida para poder lograr lo que soy ahora; y a todos mis demás familiares y amigos que me brindan alegría en todo momento.

Por ultimo a mis maestros y asesores que me formaron académicamente, por ese tiempo que dedicaron para que todo esto fuese un éxito.

¡Gracias por todo!

Pedro Enrique Velázquez Elizondo

Agradezco a mis padres, hermanos, familia, profesores y amigos por ser parte de mi vida y por ser parte fundamental para lograr este objetivo, ya que en ellos siempre encontré apoyo y confianza en todo momento, siempre estaré con ustedes así como ustedes lo han estado conmigo.

Quiero agradecer sinceramente a mis asesores, los profesores Fermín Valencia Figueroa y Mario Ponce Flores por su asesoría y apoyo para realizar este trabajo.

¡Gracias!

Contenido

<i>Índice de figuras</i>	VI
<i>Índice de tablas</i>	VIII
<i>Planteamiento del problema</i>	IX
<i>Justificación</i>	X
<i>Objetivo general</i>	XI
<i>Objetivos específicos</i>	XI
Capítulo 1 Estado del Arte	12
1.1 <i>Introducción</i>	13
1.2 <i>Sistemas Comerciales</i>	14
1.2.1 <i>Finger Clip Heart Rate Monitor</i>	14
1.2.2 <i>Polar M400 White + Sensor de Frecuencia Cardiaca H7</i>	14
1.2.3 <i>Oxímetro de Pulso Dactilar SportGuard OLED SpO2</i>	15
1.2.4 <i>Fitbit Charge HR</i>	16
1.3 <i>Proyectos académicos con interés en el Ritmo Cardiaco</i>	16
1.3.1 <i>Diseño y construcción de un sistema de monitoreo de signos vitales</i>	16
1.3.2 <i>Sistema de adquisición y registro de señales electrocardiográficas</i>	17
Capítulo 2 Marco teórico	19
2.1 <i>Ritmo Cardiaco</i>	20
2.1.1 <i>Taquicardia</i>	21
2.1.2 <i>Bradycardia</i>	21
2.1.3 <i>Presión Arterial (PA)</i>	22
2.1.4 <i>Presión Sistólica y Diastólica</i>	22
2.2 <i>Pletismografía y Fotopletismografía</i>	23
2.2.1 <i>Pletismografía</i>	23
2.2.2 <i>Fotopletismografía</i>	23
2.3 <i>Sensores</i>	24
2.3.1 <i>Sensor SEN-11574</i>	26
2.4 <i>Microcontroladores</i>	27
2.4.1 <i>Microcontrolador PIC18F4550</i>	28
2.4.1.1 <i>Timer del PIC18F4550</i>	32

2.4.1.2	USART del PIC18F4550.....	32
2.4.1.3	ADC del PIC18F4550.....	33
2.4.1.4	Interrupciones del PIC18F4550.....	35
2.5	Tecnologías y protocolos inalámbricos.....	35
2.5.1	Bluetooth.....	35
2.5.1.1	Módulo HC-06.....	36
2.6	Sistema operativo Android.....	37
2.6.1	Aplicaciones para Android.....	40
2.6.2	MIT App Inventor.....	42
2.6.3	Editor de bloques y diseñador en App Inventor.....	42
Capítulo 3	Diseño e Implementación.....	45
3.1	Características generales del sistema.....	46
3.2	Diagrama a bloques.....	46
3.3	Caracterización del sensor de ritmo cardiaco.....	47
3.3.1	Acondicionador de señal.....	49
3.3.2	Diseño de circuitos de conmutación S1 y S2.....	52
3.4	Consideraciones técnicas y cálculos finales para los valores de resistores asociados al microcontrolador.....	58
3.4.1	Diseño para el suministro y sensado de la batería.....	58
3.4.2	Cálculo de resistores para Leds de los puertos del microcontrolador.....	60
3.5	Implementación del prototipo final.....	61
3.6	Diagrama de flujo.....	63
3.7	Aplicación final para el medidor de ritmo cardiaco.....	64
3.8	Desarrollo del programa para el microcontrolador.....	67
3.8.1	Manejo de Timer 0 con el microcontrolador.....	67
3.8.2	Manejo de Interrupciones del puerto B con el microcontrolador.....	71
3.9	Programa del microcontrolador para el medidor de ritmo cardiaco.....	74
Capítulo 4	Pruebas y Resultados.....	79
4.1	Comunicación Bluetooth y microcontrolador.....	80
4.1.1	Prueba 1 “Encendido de led’s por medio de un teléfono celular”.....	80
4.1.2	Prueba 2 “Envío de datos del microcontrolador al celular por medio de una aplicación”.....	86

4.1.3 Prueba 3 “Envío de datos al microcontrolador desde el celular mediante una aplicación”	91
4.2 Simulación del programa del microcontrolador para el medidor de ritmo cardiaco	95
4.2.1 Prueba 1	96
4.2.2 Prueba 2	97
4.2.3 Prueba 3	98
4.3 Pruebas físicas del prototipo final con la aplicación en Android	98
4.3.1 Enlace con el dispositivo Android y el prototipo final	100
4.3.2 Pruebas finales con personas de la comunidad	104
Conclusiones	107
Anexos	108
Condiciones de operación del PIC18F4550	109
Condiciones de operación regulador de voltaje 7805	110
Condiciones de operación transistor 2N3904	111
Estudio de viabilidad	113
Normatividad	116
Glosario	117
Referencias	119

Índice de figuras

Fig. 1.1 Monitor de ritmo cardiaco en hospitales	13
Fig. 1.2 Sistema de monitor con clip para el dedo	14
Fig. 1.3 Pulsímetro digital	15
Fig. 1.4 Medidor de pulso cardiaco	15
Fig. 1.5 Pulsera Fitbit enlazada a un Smartphone	16
Fig. 1.6 Sistema de monitoreo de signos vitales	17
Fig. 1.7 Sistema de monitoreo grafico de signos vitales	18
Fig. 2.1 Comportamiento del corazón	22
Fig. 2.2 Gráfica del ritmo cardiaco	23
Fig. 2.3 Técnicas de captura para fotopleismografía	24
Fig. 2.4 Sensor de ritmo cardiaco	26
Fig. 2.5 Señal de salida esperada del sensor de ritmo cardiaco	26
Fig. 2.6 Circuito del sensor de ritmo cardiaco	27
Fig. 2.7 Componentes internos del microcontrolador	28
Fig. 2.8 Microcontrolador PIC18F4550 encapsulado DIP	29
Fig. 2.9 Microcontrolador PIC18F4550 encapsulado TQF	30
Fig. 2.10 Diagrama a bloques PIC18F4550	31
Fig. 2.11 Cálculo Baud Rate	33
Fig. 2.12 Diagrama a bloques del convertidor A/D	34
Fig. 2.13 Logo Bluetooth	36
Fig. 2.14 Módulo Bluetooth HC-06	37
Fig. 2.15 Logo de Android	38
Fig. 2.16 Pantalla principal con S.O. Android	41
Fig. 2.17 Logo de MIT App Inventor	42
Fig. 2.18 Entorno de App Inventor	43
Fig. 2.19 Editor de bloques de App Inventor	43
Fig. 2.20 Aplicación QR Droid	44
Fig. 3.1 Diagrama a bloques del sistema	47
Fig. 3.2 Señal proporcionada por el sensor de ritmo cardiaco en una persona	47
Fig. 3.3 Señal proporcionada por el sensor de ritmo cardiaco en una segunda persona	48

<i>Fig. 3.4 Circuito del comparador</i>	49
<i>Fig. 3.5 Señal del sensor y señal obtenida al compararla</i>	50
<i>Fig. 3.6 Diagrama generalizado del acondicionador de señales</i>	51
<i>Fig. 3.7 Formas de onda del acondicionador de señales</i>	52
<i>Fig. 3.8 Circuito de conmutación</i>	53
<i>Fig. 3.9 Comparador con primer circuito de conmutación</i>	55
<i>Fig. 3.10 Señal de salida de comparador con primer circuito de conmutación</i>	56
<i>Fig. 3.11 Acondicionador de señal</i>	57
<i>Fig. 3.12 Señal acondicionada para el microcontrolador</i>	57
<i>Fig. 3.13 Bateria Lipo para el suministro eléctrico del circuito</i>	58
<i>Fig. 3.14 Arreglo de resistores para protección del microcontrolador</i>	60
<i>Fig. 3.15 Circuito final para el medidor de ritmo cardiaco</i>	61
<i>Fig. 3.16 PCB del circuito</i>	62
<i>Fig. 3.17 Placa 3D simulada por la parte superior</i>	62
<i>Fig. 3.18 Placa 3D simulada por la parte inferior</i>	62
<i>Fig. 3.19 Diagrama de flujo</i>	63
<i>Fig. 3.20 Diseño final de la aplicación en APP Inventor</i>	65
<i>Fig. 3.21 Bloques correspondientes a la vinculación del bluetooth</i>	66
<i>Fig. 3.22 Bloques correspondientes a la recepción de datos por bluetooth</i>	66
<i>Fig. 3.23 Bloques correspondiente para salir de la aplicación</i>	67
<i>Fig. 3.24 Primera parte del diagrama de flujo del manejo de Timer 0</i>	70
<i>Fig. 3.25 Segunda parte del diagrama de flujo del manejo de Timer 0</i>	70
<i>Fig. 3.26 Primera parte del diagrama de flujo del manejo de interrupciones</i>	73
<i>Fig. 3.27 Segunda parte del diagrama de flujo del manejo de interrupciones</i>	73
<i>Fig. 3.28 Diagrama de flujo del microcontrolador para el medidor de ritmo cardiaco</i>	78
<i>Fig. 4.1 Circuito para la prueba 1</i>	81
<i>Fig. 4.2 Diagrama de flujo del programa de la prueba 1</i>	81
<i>Fig. 4.3 Segunda parte del diagrama de flujo del programa de la prueba 1</i>	82
<i>Fig. 4.4 Diseño de la aplicación para Android correspondiente a la prueba 1</i>	85
<i>Fig. 4.5 Bloques para el diseño de la aplicación correspondiente a la prueba 1</i>	86
<i>Fig. 4.6 Circuito para la prueba 2</i>	87
<i>Fig. 4.7 Diagrama de flujo del programa para la prueba 2</i>	87

<i>Fig. 4.8 Aplicación para Android correspondiente a la prueba 2</i>	90
<i>Fig. 4.9 Bloques de la aplicación para Android para la prueba 2</i>	90
<i>Fig. 4.10 Circuito usado para la prueba 3</i>	91
<i>Fig. 4.11 Diagrama de flujo de la prueba 3</i>	92
<i>Fig. 4.12 Aplicación con barra deslizante virtual</i>	94
<i>Fig. 4.13 Configuración de bloques de la aplicación con barra deslizante virtual</i>	95
<i>Fig. 4.14 Simulación del funcionamiento del microcontrolador</i>	96
<i>Fig. 4.15 Primer resultado de Simulación</i>	97
<i>Fig. 4.16 Segundo resultado de Simulación</i>	97
<i>Fig. 4.17 Tercer resultado de Simulación</i>	98
<i>Fig. 4.18 Parte superior del medidor de ritmo cardiaco</i>	99
<i>Fig. 4.19 Parte inferior del circuito (Bluetooth)</i>	99
<i>Fig. 4.20 Sensor de ritmo cardiaco</i>	100
<i>Fig. 4.21 Consumo de corriente 59.6 mAh</i>	100
<i>Fig. 4.22 Aplicación en espera de la vinculación bluetooth</i>	101
<i>Fig. 4.23 Aplicación asociada al bluetooth</i>	102
<i>Fig. 4.24 Comparación del resultado del prototipo con el comercial</i>	102
<i>Fig. 4.25 Lectura de ritmo cardiaco a una persona</i>	103
<i>Fig. 4.26 Lectura de ritmo cardiaco a una persona después de correr</i>	104
<i>Fig. 4.27 Gráfica de comparativa de resultados</i>	105

Índice de tablas

<i>Tabla 2.1 Frecuencia cardiaca en hombres</i>	20
<i>Tabla 2.2 Frecuencia cardiaca en mujeres</i>	20
<i>Tabla 2.3 Características de sensores</i>	25
<i>Tabla 2.4 Microcontroladores familia 18F</i>	30
<i>Tabla 2.5 Actualizaciones de Android</i>	39
<i>Tabla 2.6 Características del sistema operativo Android</i>	40
<i>Tabla 3.1 Datos proporcionados por la hoja de datos del transistor 2N3904</i>	53
<i>Tabla 4.1 Pruebas en alumnos</i>	105

Planteamiento del problema

Hoy en día las enfermedades del corazón son bastante comunes, en México, al ser uno de los países con el mayor índice de obesidad en el mundo, según el INEGI, se tienen como consecuencia muchos problemas cardiacos, los cuales son una de las principales causas de muerte en nuestro país.

Es por ello que algunas de las causas más comunes de hospitalización son debido a problemas cardiacos, como ataques del corazón, infartos o ritmo cardiaco elevado, este tipo de padecimientos deben ser tratados lo antes posible ya que el paciente corre el riesgo de perder la vida, esto se debe a que mientras más crece una persona, se vuelve más propensa a alguna enfermedad relacionada con el corazón.

Debido a las enfermedades mencionadas es común que las personas asistan a revisiones de sus principales signos vitales, como presión y ritmo cardiaco e incluso niveles de azúcar, etc., esto implica que tengan que trasladarse hasta algún hospital o centro de salud, convirtiéndolo en una tarea complicada si se tiene que hacer con regularidad.

A pesar de que los problemas cardiacos son comunes en gente de la tercera edad y personas con obesidad en ocasiones no se cuenta con un medio accesible y económico con el cual puedan monitorear de forma sencilla su ritmo cardiaco en momentos determinados.

En lugares como hospitales, los pacientes en salas de recuperación y rehabilitación utilizan dispositivos muy grandes para la medición del ritmo cardiaco, esto provoca un problema al estar transportando dichos aparatos, los cuales ocupan mucho espacio en lugares donde debe estar despejada el área para cualquier emergencia médica.

Existen diversos aparatos en el mercado los cuales tienen la función de monitorear el ritmo cardiaco sin embargo tienen muchos inconvenientes como es el elevado costo, la mayoría son sistemas alámbricos, entre otros.

Justificación

El propósito principal de este proyecto es generar una herramienta con la cual se pueda brindar un mejor cuidado y tratamiento al corazón de cualquier persona, con solo un sensor, un teléfono celular y una interfaz que lo hagan interactuar de una forma sencilla y económica.

Cabe mencionar que monitorear el corazón desde casa ha ido en aumento debido a la alta demanda de estos servicios en los últimos años, además de que se eliminarían costos de transportación hacia algún hospital o centro médico.

Además de tratamiento a personas con problemas cardiacos, se puede utilizar en centros deportivos para atletas de distintas categorías, ya que durante los entrenamientos es necesario conocer el ritmo cardiaco de la persona para no someterlo a alguna rutina más intensa.

Tanto los atletas profesionales como los amateurs, han confiado durante décadas en la información provista por su monitor de ritmo cardíaco por las siguientes razones:

- 1. Un monitor de ritmo cardíaco es como un cuenta vueltas, dando una medición de la intensidad del ejercicio.*
- 2. Entrenar a su propio ritmo adecuado es posible con un monitor de ritmo cardíaco.*
- 3. La medición directa del ritmo cardíaco durante el ejercicio es la manera más precisa de graduar la intensidad.*
- 4. Maximiza los beneficios del ejercicio en un limitado período de tiempo.*
- 5. Es una herramienta para regular la frecuencia y la intensidad de la actividad deportiva.*

Si se realizan ejercicios como caminatas o rutinas en bicicleta, debe hacerse de una manera que no ponga en riesgo la salud, por eso, es importante que tener en cuenta cuánto aumenta el ritmo cardíaco, para evitar problemas con el corazón.

El ritmo cardíaco indica la actividad del corazón, cuyo principal trabajo es bombear la sangre oxigenada que se distribuye por todo el cuerpo. En reposo, habitualmente el corazón late entre 60 y 100 veces (bombeos) por minuto, y al entrar en movimiento se acelera porque la sangre demanda más oxígeno al corazón.

Objetivo general:

Diseñar un sistema para visualizar el valor del ritmo cardiaco de una persona mediante una aplicación Android para dispositivos móviles.

Objetivos específicos:

- *Estudiar las características de las señales eléctricas proporcionadas por el sensor de ritmo cardiaco.*
- *Diseñar un circuito acondicionador para el sensor de ritmo cardiaco que permita al microcontrolador reconocer la señal proporcionada.*
- *Programar un microcontrolador para transmitir el valor del ritmo cardiaco mediante un enlace Bluetooth.*
- *Diseñar una aplicación para dispositivos móviles Android con la cual se pueda monitorear el valor del ritmo cardiaco.*

Capítulo 1
Estado del Arte

1.1 Introducción

El avance de las tecnologías en el campo de la electrónica y su cobertura en prácticamente todos los ámbitos de la vida, han hecho que en los últimos años se encuentren disponibles dispositivos capaces de monitorear los signos vitales de las personas mediante sistemas inalámbricos y en una forma muy sencilla por medio de pantallas táctiles y sistemas móviles.

Actualmente, es común encontrar dispositivos con los que podemos conocer más sobre nuestro cuerpo, por ejemplo; las básculas, donde podemos conocer nuestro peso de una forma sencilla. Otro dispositivo es el Glucómetro, con el cual se puede conocer el nivel de glucosa en la sangre de algún paciente diabético.

El monitor de ritmo cardiaco es un instrumento con el cual se puede conocer el número de veces que se contrae el corazón durante un minuto, este valor puede variar debido a algunos factores como lo son la edad y la condición física.

En hospitales se usan algunos monitores de ritmo cardiaco, los cuales no son portátiles y se usan en los quirófanos durante cirugías, salas de recuperación entre otros, ver figura 1.1. Cabe mencionar que son costosos y además cuentan con un monitor para observar el comportamiento del corazón del paciente de una forma gráfica.



Fig. 1.1 Monitor de ritmo cardiaco en hospitales

1.2 Sistemas Comerciales

En el mercado existen diferentes tipos de monitores de ritmo cardiaco, todos con diferentes características y costos, algunos vienen integrados en relojes o en simples pulseras, a continuación se describen algunos de ellos:

1.2.1 Finger Clip Heart Rate Monitor

Este sistema está integrado por una aplicación para celulares iPhone y Android junto con un sensor que se conecta de forma alámbrica al celular, dicho sensor se coloca en el dedo o en el oído de la persona. Los datos que muestra en la pantalla son los valores del ritmo cardiaco, la frecuencia cardiaca máxima y mínima. En la figura 1.2 se muestra el sensor junto con el celular y la aplicación instalada.



Fig. 1.2 Sistema de monitor con clip para el dedo

1.2.2 Polar M400 White + Sensor de Frecuencia Cardiaca H7

Este sistema tiene un GPS integrado con seguimiento de actividad las 24 horas, contiene un pulsímetro utilizado normalmente para corredores debido a su diseño deportivo como se observa en la figura 1.3, contiene funciones avanzadas como el control de actividad con un medidor de ritmo cardiaco y un GPS el cual es capaz de medir la distancia recorrida. El resumen de la información puede ser enviado a un dispositivo móvil o incluso a una PC, el tipo de conectividad es vía Bluetooth.

Una característica muy importante es la vida útil de su batería recargable, ya que tiene aproximadamente un uso continuo de 8 horas y una resistencia al agua con una profundidad de 30 m.



Fig. 1.3 Pulsímetro digital

1.2.3 Oxímetro de Pulso Dactilar SportGuard OLED SpO2

Es un medidor de oxígeno en la sangre para el dedo con alta precisión y durabilidad, tiene una pantalla OLED el cual despliega el valor de la presión en tiempo real y su consumo de batería es bajo lo cual hace que tenga un rendimiento de 50 horas continuas. Se utiliza comúnmente en hospitales y cuidado médico casero por su tamaño óptimo como se observa en la figura 1.4. Tiene un apagado automático cuando no hay señal en el sensor después de 8 segundos.



Fig. 1.4 Medidor de oxígeno

1.2.4 Fitbit Charge HR

Este es un dispositivo portátil en forma de pulsera la cual con un solo toque puede medir al instante el ritmo cardiaco y mostrarlo en una pequeña pantalla OLED, su batería llega a durar hasta 5 días y es compatible con más de 120 Smartphones, su conectividad es vía Bluetooth, tiene grandes características como la resistencia al agua y temperaturas bajas de hasta -20°C. Su diseño es muy pequeño lo cual hace de esta pulsera muy portátil y es utilizada con frecuencia para deportistas de gran actividad física. Se puede comparar el tamaño con un Smartphone en la figura 1.5.



Fig. 1.5 Pulsera Fitbit enlazada a un Smartphone

1.3 Proyectos académicos con interés en el Ritmo Cardiaco

A lo largo de los últimos años, la electrónica ha ido abriéndose paso en el campo de la medicina debido a que los componentes necesarios cada vez son más sofisticados y confiables, así como de un costo bajo.

Además se ha buscado que las técnicas quirúrgicas, así como de medición de signos vitales, sean menos invasivas, es por ello que a continuación se presentan algunos proyectos académicos que están relacionados con el ritmo cardiaco.

1.3.1 Diseño y construcción de un sistema de monitoreo de signos vitales

Esta tesis tradicional fue presentada por los alumnos Diana Janet Olvera Téllez y José Uriel González Escalona de la ESIME Zacatenco en noviembre del 2013.

Este trabajo consistía en implementar un circuito capaz de medir temperatura y el ritmo cardiaco de una persona por medio de diferentes sensores.

La lectura de los sensores es mostrada en un display LCD en el circuito, lo cual lo hace un sistema alámbrico, todo el sistema se conecta a una batería de 9V no recargable.

Lo que se buscaba era realizar un prototipo de bajo costo con elementos del mercado nacional y que fuera fácil de usar. En la figura 1.6 se muestra una ilustración del sistema.

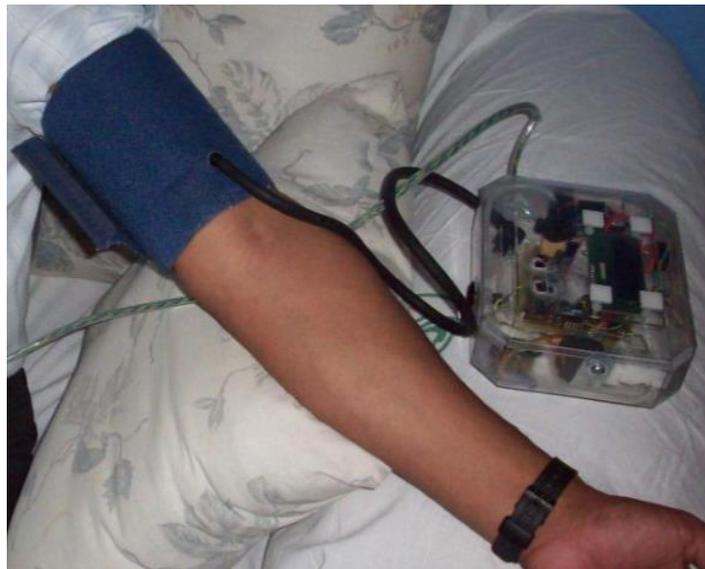


Fig. 1.6 Sistema de monitoreo de signos vitales

1.3.2 Sistema de adquisición y registro de señales electrocardiográficas

En el 2007 en la ESIME Zacatenco los alumnos Eloísa Díaz Cruz, Pablo Hernández del Valle y Oswaldo Weber Nicasio presentan una tesis en la cual llevan a cabo un sistema de medición de forma gráfica del ritmo cardiaco por medio de sensores implementados con diferentes filtros para acondicionar la señal.

Este sistema tiene un interfaz a la PC en la cual se puede observar dicha señal, en la figura 1.7 se observa una captura de la PC del sistema en funcionamiento.

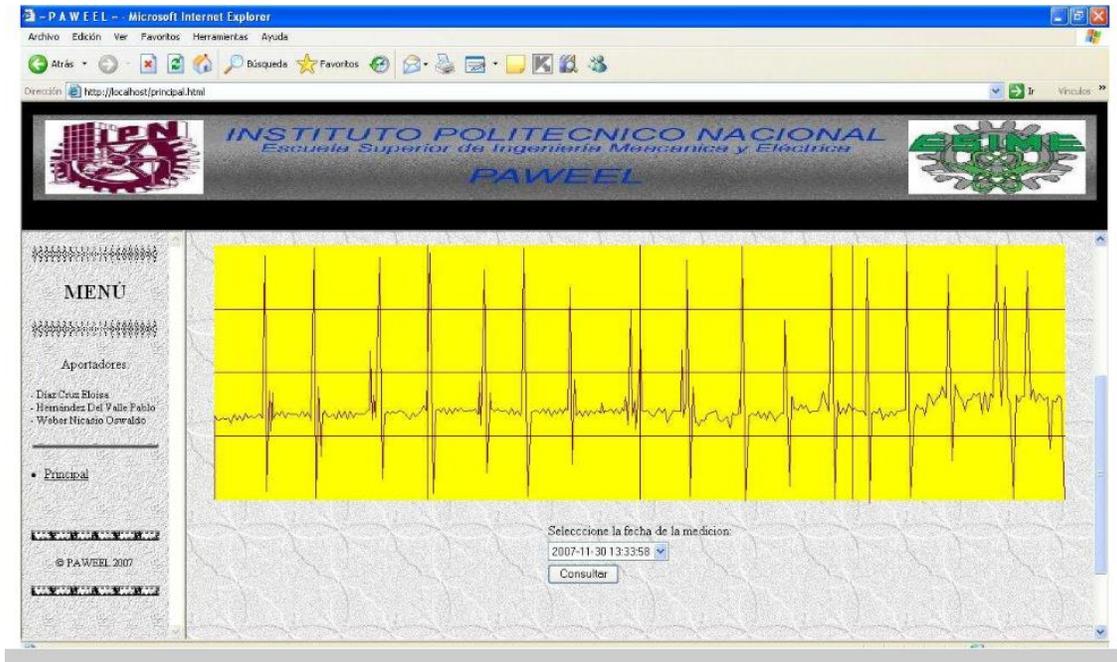


Fig. 1.7 Sistema de monitoreo grafico de signos vitales

Los proyectos anteriores están basados en sistema alámbricos, académicamente no se tienen tantas referencias de proyectos en los cuales involucre sistemas inalámbricos con tecnologías nuevas, sin embargo los anteriores son proyectos que sirven como una referencia o bibliografía para futuros proyectos.

Capítulo 2
Marco teórico

2.1 Ritmo Cardíaco

La frecuencia cardíaca o ritmo cardíaco es el número de latidos por unidad de tiempo, estos generalmente se miden en cantidad de pulsaciones por minuto, en seres humanos, la medida del ritmo cardíaco normal se encuentra entre 70 y 90 latidos o pulsaciones por minuto. Además, esta medición se utiliza para ayudar a diagnosticar enfermedades cardíacas y para el seguimiento de distintas condiciones médicas.

Para calcularla es necesario estar en reposo, ya que el ejercicio, sueño y otras situaciones pueden afectar la frecuencia, una forma de hacer la lectura es utilizando los dedos, no el pulgar, sobre la arteria radial que se encuentra en la muñeca o la arteria carótida que se ubica en el cuello.

En las siguientes tablas se observa el índice de frecuencia cardíaca de la persona según sea su género y/o edad.

Tabla 2.1 Frecuencia cardíaca en hombres

Valores de frecuencia cardíaca en reposo para hombres

Edad	Mala	Normal	Buena	Muy Buena
20-29	86 o más	70-84	62-68	60 o menos
30-39	86 o más	72-84	64-70	62 o menos
40-49	90 o más	74-88	66-72	64 o menos
50-59	90 o más	74-88	68-74	66 o menos
60 o más	94 o más	76-90	70-76	68 o menos

Tabla 2.2 Frecuencia cardíaca en mujeres

Valores de frecuencia cardíaca en reposo para mujeres

Edad	Mala	Normal	Buena	Muy Buena
20-29	96 o más	78-94	72-76	70 o menos
30-39	98 o más	80-96	72-78	70 o menos
40-49	100 o más	80-98	74-78	72 o menos
50-59	104 o más	84-102	76-82	74 o menos
60 o más	108 o más	88-106	78-88	78 o menos

2.1.1 Taquicardia

La taquicardia es una condición donde la frecuencia cardíaca es más rápida de lo normal. Un corazón adulto sano normalmente late entre 60 y 100 veces por minuto cuando está en reposo. Una frecuencia cardíaca por encima de 100 latidos por minuto es generalmente aceptada como taquicardia.

Al tener taquicardia, la tasa en las cámaras superiores o cámaras inferiores del corazón, o ambos, se incrementan significativamente. La taquicardia puede causar mareo, dificultad para respirar, palpitaciones, dolor de pecho incluso un desmayo.

En algunos casos, las taquicardias pueden no causar síntomas o complicaciones. Sin embargo, las taquicardias pueden perturbar gravemente la función cardíaca normal, aumentar el riesgo de derrame cerebral, o causar un paro cardíaco repentino o la muerte.

La frecuencia cardíaca se controla por señales eléctricas enviadas a través de los tejidos del corazón. La taquicardia se produce cuando una anomalía en el corazón produce señales eléctricas rápidas.

2.1.2 Bradicardia

La bradicardia consiste en tener la frecuencia cardíaca más baja de lo normal. El corazón late normalmente entre 60 y 100 veces por minuto en un adulto en reposo. En caso de tener bradicardia, el corazón late menos de 60 veces minuto.

La bradicardia puede ser un problema grave si el corazón no bombea suficiente sangre rica en oxígeno al resto del cuerpo. Para algunas personas, sin embargo, la bradicardia no produce síntomas o complicaciones.

Al tener la bradicardia, el cerebro y otros órganos no pueden obtener el suministro de oxígeno que necesitan. Esta es causada por algo que interrumpe los impulsos eléctricos normales que controlan el ritmo de la acción de bombeo del corazón.

2.1.3 Presión Arterial (PA)

Es la presión que ejerce la sangre contra la pared de las arterias. Esta presión es imprescindible para que circule la sangre por los vasos sanguíneos y aporte el oxígeno y los nutrientes a todos los órganos del cuerpo para que puedan funcionar. Es un tipo de presión sanguínea que consta de dos componentes, la presión sistólica y diastólica. Ver figura 2.1 donde se explica el comportamiento del corazón.

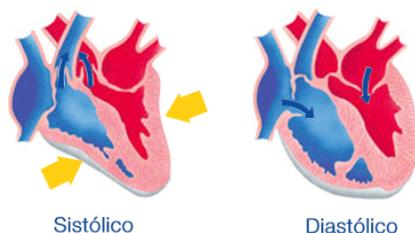


Fig. 2.1 Comportamiento del corazón

2.1.4 Presión Sistólica y Diastólica

Presión arterial sistólica: corresponde al valor máximo de la tensión arterial en sístole (cuando el corazón se contrae). Se refiere al efecto de presión que ejerce la sangre eyectada del corazón sobre la pared de los vasos. Explicación en figura 2.2.

Presión arterial diastólica: corresponde al valor mínimo de la tensión arterial cuando el corazón está en diástole o entre latidos cardíacos. Depende fundamentalmente de la resistencia vascular periférica. Se refiere al efecto de distensibilidad de la pared de las arterias, es decir, el efecto de presión que ejerce la sangre sobre la pared del vaso.

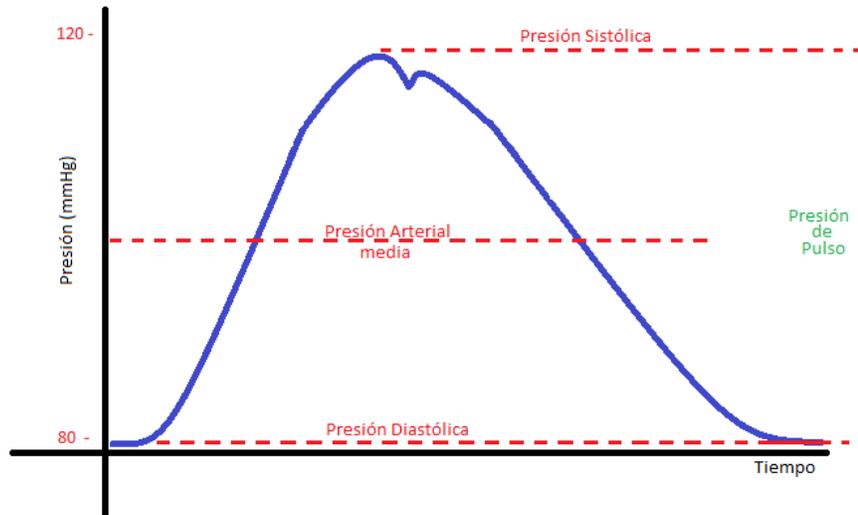


Fig. 2.2 Gráfica del ritmo cardiaco

2.2 Pletismografía y Fotopletismografía

2.2.1 Pletismografía

La pletismografía consiste en registrar de una manera no invasiva las variaciones de volumen sanguíneo en las diferentes partes del cuerpo de una persona, generalmente se hace en las extremidades, como lo pueden ser los dedos o los lóbulos de la oreja. Con esta técnica se obtiene el ritmo cardiaco de la persona.

2.2.2 Fotopletismografía

La fotopletismografía es una técnica asociada con la pletismografía pero con medio ópticos, un fotopletismógrafo se obtiene a menudo mediante el uso de un opto acoplador que ilumina la piel y mide los cambios en la absorción de la luz que se dan debido a que con cada ciclo cardiaco, el corazón bombea sangre a la periferia.

El fotopletismógrafo obtiene la señal de las variaciones en los dedos, como se observa en la figura 2.3, existen tres técnicas:

- Trans-iluminado
- Reflexión de luz
- Fibra óptica

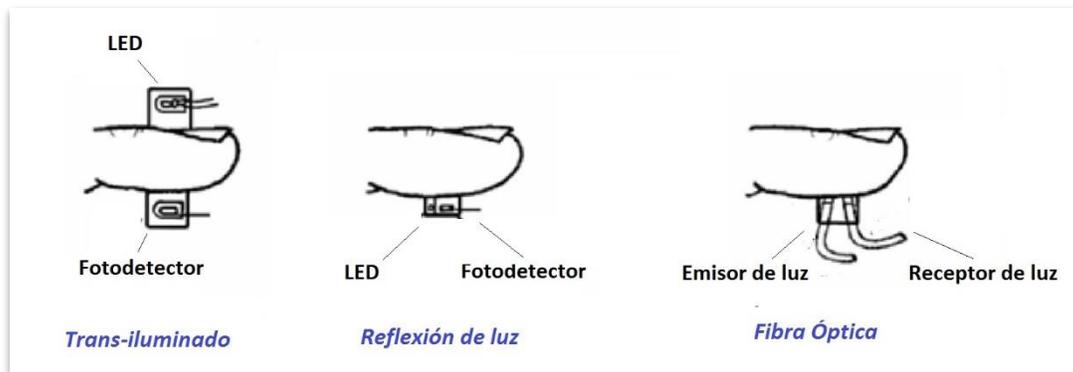


Fig. 2.3 Técnicas de captura para fotoplethysmografía

2.3 Sensores

Un sensor es un dispositivo capaz de transformar magnitudes físicas o químicas, llamadas variables de instrumentación, en magnitudes eléctricas, las cuales pueden acondicionarse para utilizarse en microcontroladores, CPLD, FPGA o cualquier otro sistema de control y monitoreo.

Los sensores a pesar de ser de diferentes tipos, comparten características similares, en la tabla 2.3 se muestran las características más comunes junto con una breve descripción.

En la actualidad existen diferentes tipos de sensores con los cuales se pueden conocer parámetros como temperatura, humedad, intensidad luminosa, distancia, etc. Dichos sensores tienen aún más clasificaciones y todos tienen características que los hacen indispensables dependiendo de la aplicación.

Tabla 2.3 Características de sensores

Característica	Descripción
Rango de medida	Dominio en la magnitud medida en el que puede aplicarse el sensor.
Precisión	Error de medida máximo esperado.
Offset	Valor de la variable de salida cuando la variable de entrada es nula. Si el rango de medida no llega a valores nulos de la variable de entrada, habitualmente se establece otro punto de referencia para definir el offset.
Resolución	Mínima variación de la magnitud de entrada que puede apreciarse a la salida.
Rapidez de respuesta	Puede ser un tiempo fijo o depender de cuánto varíe la magnitud a medir. Depende de la capacidad del sistema para seguir las variaciones de la magnitud de entrada.
Sensibilidad	Relación entre la variación de la magnitud de salida y la variación de la magnitud de entrada.

2.3.1 Sensor SEN-11574

El sensor SEN-11574 es en esencia un fotopleximógrafo, el cual es conocido como un dispositivo no invasivo para el monitoreo del ritmo cardiaco.



Fig. 2.4 Sensor de ritmo cardiaco

La señal de salida de este sensor es una señal analógica que representa las pulsaciones o ritmo cardiaco de la persona que se esté monitoreando, la forma de onda esperada según el fabricante es similar a la de la figura 2.5.

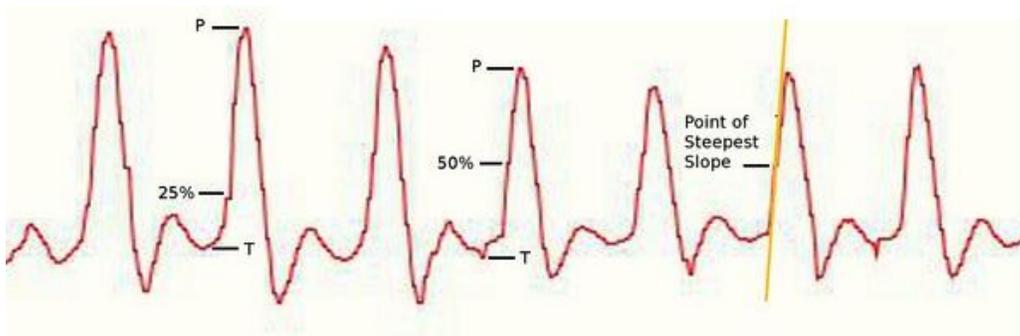


Fig. 2.5 Señal de salida esperada del sensor de ritmo cardiaco

Este sensor está compuesto por un sensor de luz ambiental y un LED color verde brillante de montaje superficial, contiene un diodo de protección en caso de una conexión de alimentación inversa. Además contiene un filtro activo para hacer más definida la señal de salida, y un amplificador para dar mayor ganancia de voltaje. En la figura 2.6 se muestra el esquemático del sensor a utilizar.

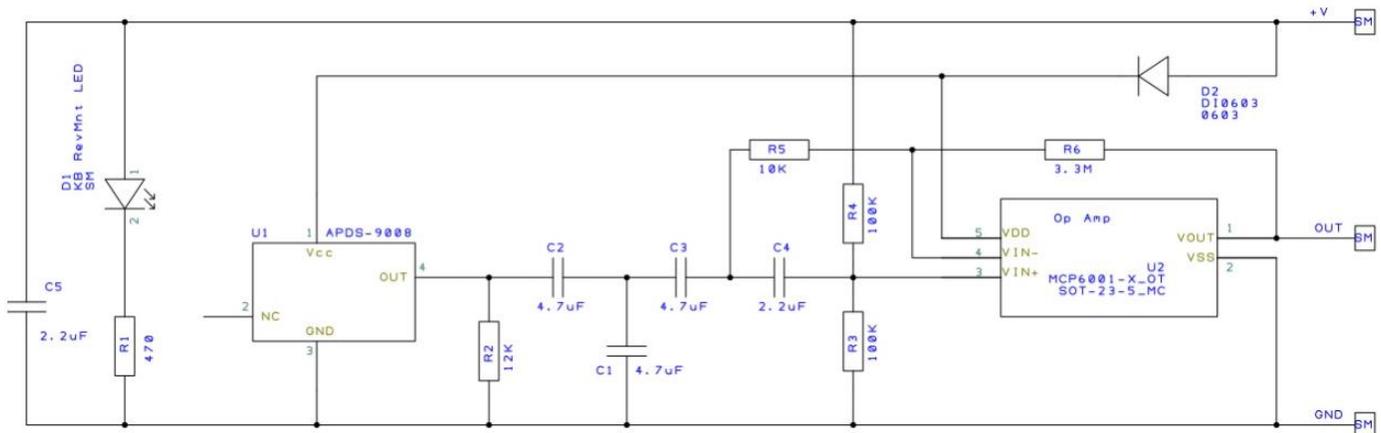


Fig. 2.6 Circuito del sensor de ritmo cardiaco

Características adicionales del sensor de ritmo cardiaco:

- Diámetro: 16 mm
- Espesor total: 3mm
- Voltaje de suministro: 3-5 V
- Consumo de corriente: 4mA a 5V

2.4 Microcontroladores

Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada, salida y periféricos, ver figura 2.7.

Todo microcontrolador requiere de un programa para que realice una función específica. Este se almacena normalmente en la memoria ROM. No está de más mencionar que sin un programa, los microcontroladores carecen de utilidad.

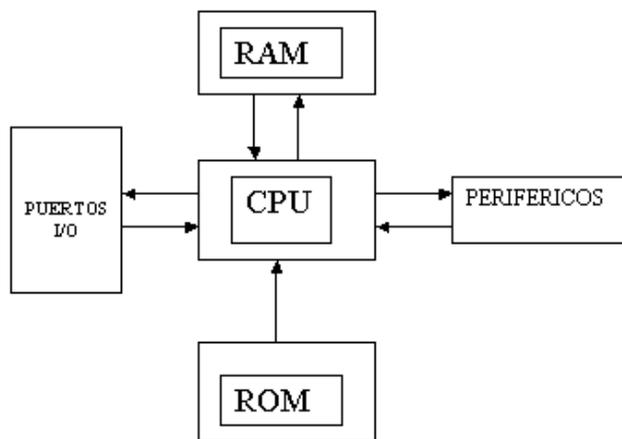


Fig. 2.7 Componentes internos del microcontrolador

El propósito fundamental de los microcontroladores es el de leer y ejecutar los programas que el usuario le escribe, es por esto que la programación es una actividad básica e indispensable cuando se diseñan circuitos y sistemas que los incluyan. El carácter programable de los microcontroladores simplifica el diseño de circuitos electrónicos.

Permiten flexibilidad, ya que un mismo circuito se puede utilizar para que realice diferentes funciones con solo cambiar el programa del microcontrolador.

Es común encontrar microcontroladores en campos como la robótica y la automatización, en la industria del entretenimiento, en las telecomunicaciones, en la instrumentación, en el hogar y en la industria automotriz.

Entre los microcontroladores más populares se encuentran los PIC's, los cuales son fabricados por Microchip, estos PIC's se dividen en varias familias como los PIC18F.

2.4.1 Microcontrolador PIC18F4550

El PIC18F4550 es un microcontrolador de la familia 18F de Microchip, este dispositivo cuenta con tecnología NanoWatt, cuenta con 40 y 44 pines dependiendo el encapsulado.

Algunas de las características relevantes del PIC18F4550 son las siguientes:

-Oscilador: Cuenta con un oscilador interno con frecuencias seleccionables desde 31kHz hasta 8MHz, puede ser configurado para usar cristal externo y PLL de alta precisión.

-Periféricos: Tres interrupciones externas, Cuatro Timer's; T0, T1, T2 y T3, Comparadores, Módulos de PWM, Convertidor A/D, USART y SPI.

Algunas características especiales son:

- Rango de operación de 2V a 5.5V
- Compatible con compiladores C

Este microcontrolador tiene diferentes encapsulados uno de ellos es el DIP, el cual es de fácil inserción en bases para circuitos integrados y protoboard, el encapsulado de la figura 2.8 cuenta con 40 pines y todos ellos con alguna función.

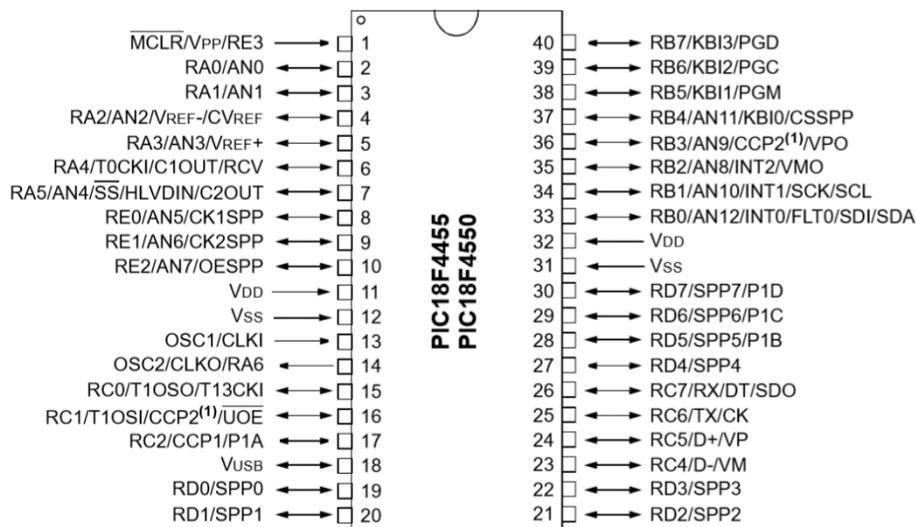


Fig. 2.8 Microcontrolador PIC18F4550 encapsulado DIP

El encapsulado TQF, es un encapsulado de montaje superficial de 44 pines, ver figura 2.9, una ventaja de este encapsulado es su tamaño, ya que es menor en comparación con el encapsulado DIP, sin embargo es forzoso soldar el microcontrolador, ya que no puede ser extraído múltiples veces, es por ello que se debe destinar algunas conexiones para poder ser reprogramado.

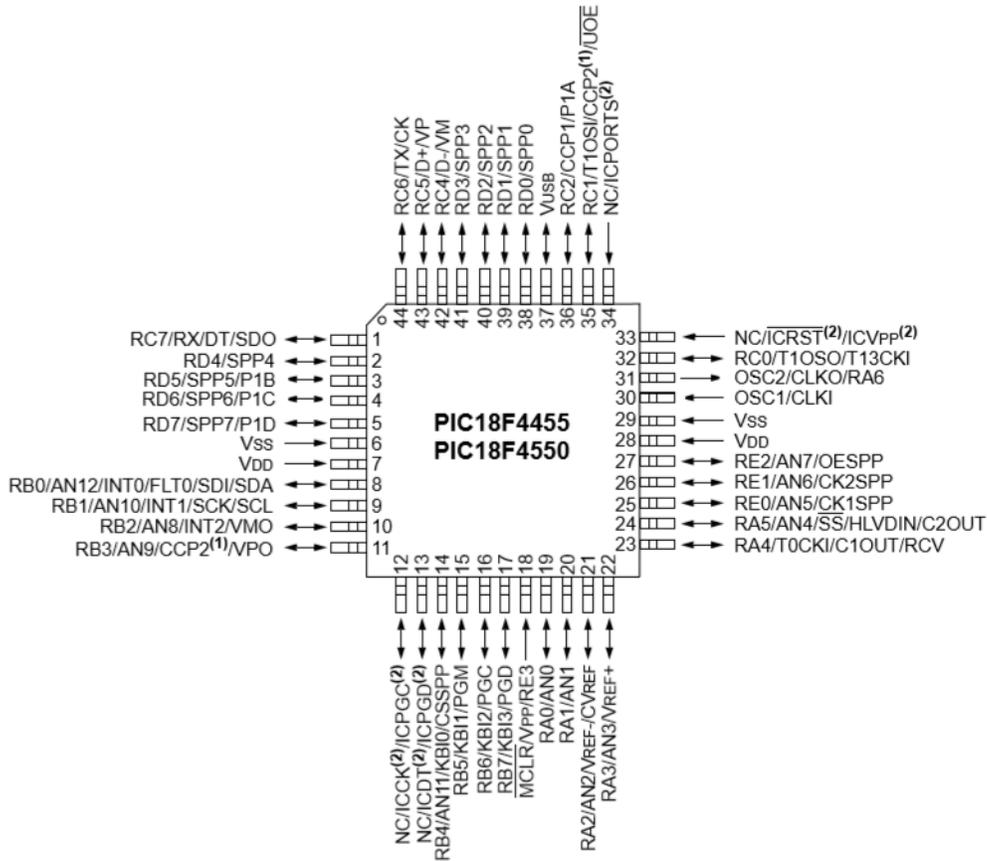


Fig. 2.9 Microcontrolador PIC18F4550 encapsulado TQF

En la siguiente tabla se enlistan algunas características del PIC18F4550 comparado con algunos otros microcontroladores similares.

Tabla 2.4 Microcontroladores familia 18F

Característica	PIC18F2455	PIC18F2550	PIC18F4550
Frecuencia de operación	DC-48 MHz	DC-48 MHz	DC-48 MHz
Fuentes de interrupción	19	19	20
Puertos de E/S	Puerto A,B,C (E)	Puerto A,B,C (E)	Puerto A, B, C, D, E
Timers	4	4	4
PWM	0	0	1
Comunicación Serial	USART	USART	USART, SPI
ADC	10 Canales	10 Canales	13 Canales
Comparador	2	2	2
Set de instrucciones	75 Instrucciones	75 Instrucciones	75 Instrucciones
Encapsulado	DIP	DIP	DIP, TQFP, QFN

En la siguiente figura se muestra un diagrama a bloques generalizado del microcontrolador PIC18F4550.

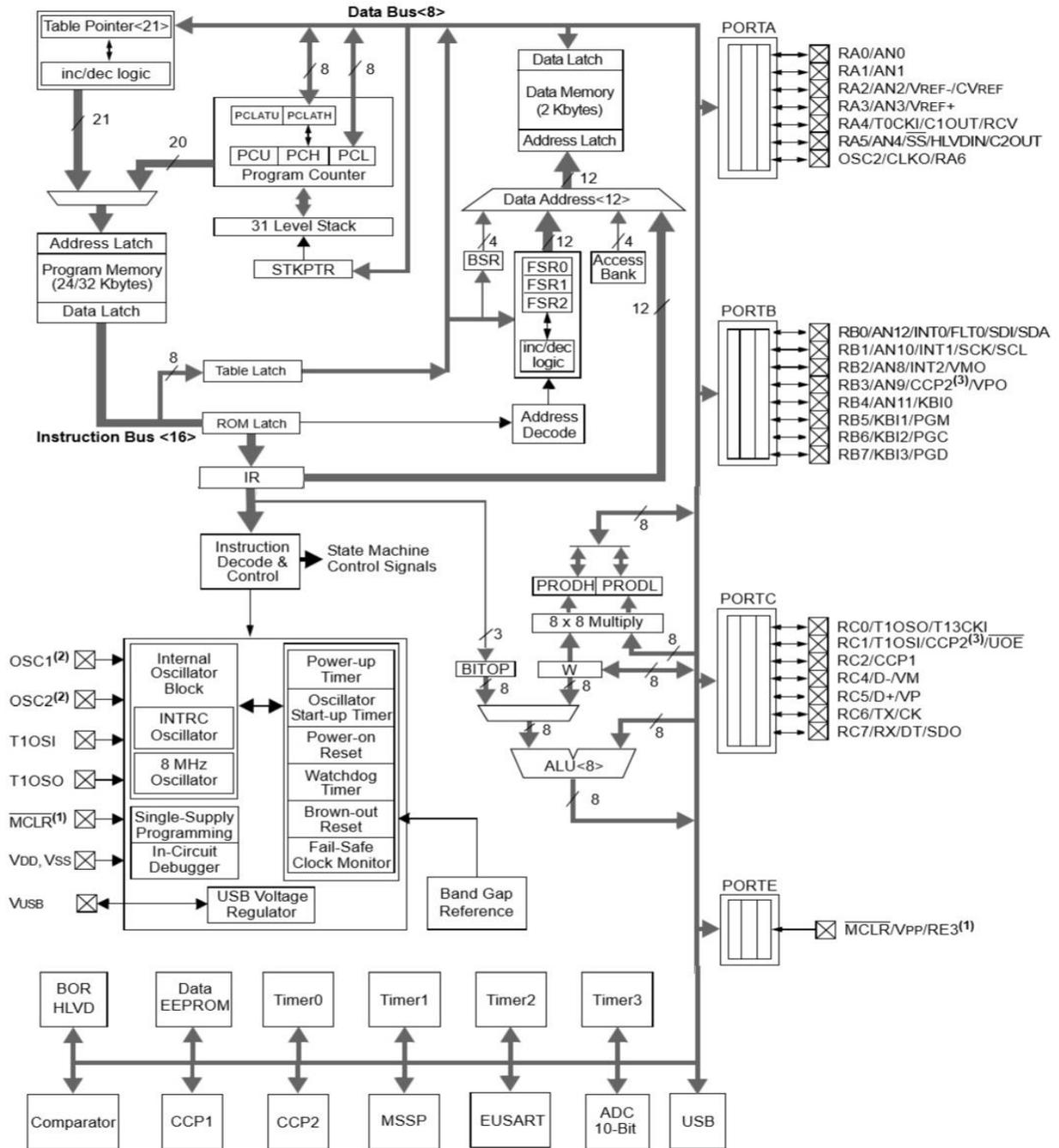


Fig. 2.10 Diagrama a bloques PIC18F4550

2.4.1.1 Timer del PIC18F4550

La mayoría de los microcontroladores tienen uno o varios timer's. Son muy útiles para medir el tiempo que ha pasado entre dos eventos, establecer tareas para ejecutarse a intervalos regulares, etc. Dependiendo del modelo los PIC's cuentan con un número variable de timer's.

El compilador XC8 tiene disponibles varias rutinas para configurar y operar con los timer's de una forma bastante cómoda. La configuración de los timer's está basada en ciclos del oscilador.

Un timer no es más que un contador cuya entrada está conectada al reloj del sistema. De hecho, la mayoría de los timer's pueden reconfigurarse como contadores.

El Timer0 del PIC18F4550 contiene las siguientes características:

- Funciona como temporizador o como contador, ambos de 8 o 16 bits.*
- Se puede leer y escribir en sus registros.*
- Fuente de reloj seleccionable entre interna o externa*
- Interrupción por desbordamiento.*

El registro T0CON controla todos los aspectos de la operación del módulo del Timer, incluyendo la selección del pre-escaler. La interrupción del Timer es generada cuando el Timer0 se desborda de FFh a 00h en su modo de 8 bits o de FFFFh a 00h en el modo de 16 bits, el desbordamiento activa la bandera de interrupción.

2.4.1.2 USART del PIC18F4550

El módulo transmisor-receptor asíncrono-síncrono universal, por sus siglas en Inglés USART (Universal Synchronous Asynchronous Receiver Transmitter) es uno de los dos módulos de entrada/salida serial. El USART también es conocido como una interfaz de comunicación serial.

El módulo USART puede ser configurado como un sistema Full-duplex asíncrono el cual puede comunicarse con periféricos de otros dispositivos, también puede ser configurado como un sistema asíncrono Half-duplex.

Una característica adicional implementada es la detección del Baud-Rate de forma automática, al igual que la calibración.

Los pines del módulo USART son multiplexados con el Puerto C, con los pines de RC6/TX/CK y RC7/RX/SDO.

Generador de Baud Rate (BRG)

El generador de Baud-Rate es un generador de 8 o 16 bits que trabaja en ambos modos, asíncrono y síncrono del módulo USART. El reloj del dispositivo, es decir, el oscilador, es usado para generar el Baud Rate deseado, este nuevo reloj podría ser operado a una frecuencia diferente.

El valor del Generador de Baud Rate se estima mediante el Baud Rate deseado y la frecuencia de oscilación. Existe una serie de fórmulas para calcular el valor del Baud Rate, en la figura 2.11 se muestra un ejemplo.

EXAMPLE 20-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:	
Desired Baud Rate	= Fosc/(64 ([SPBRGH:SPBRG] + 1))
Solving for SPBRGH:SPBRG:	
X	= ((Fosc/Desired Baud Rate)/64) - 1
	= ((16000000/9600)/64) - 1
	= [25.042] = 25
Calculated Baud Rate	= 16000000/(64 (25 + 1))
	= 9615
Error	= (Calculated Baud Rate - Desired Baud Rate)/Desired Baud Rate
	= (9615 - 9600)/9600 = 0.16%

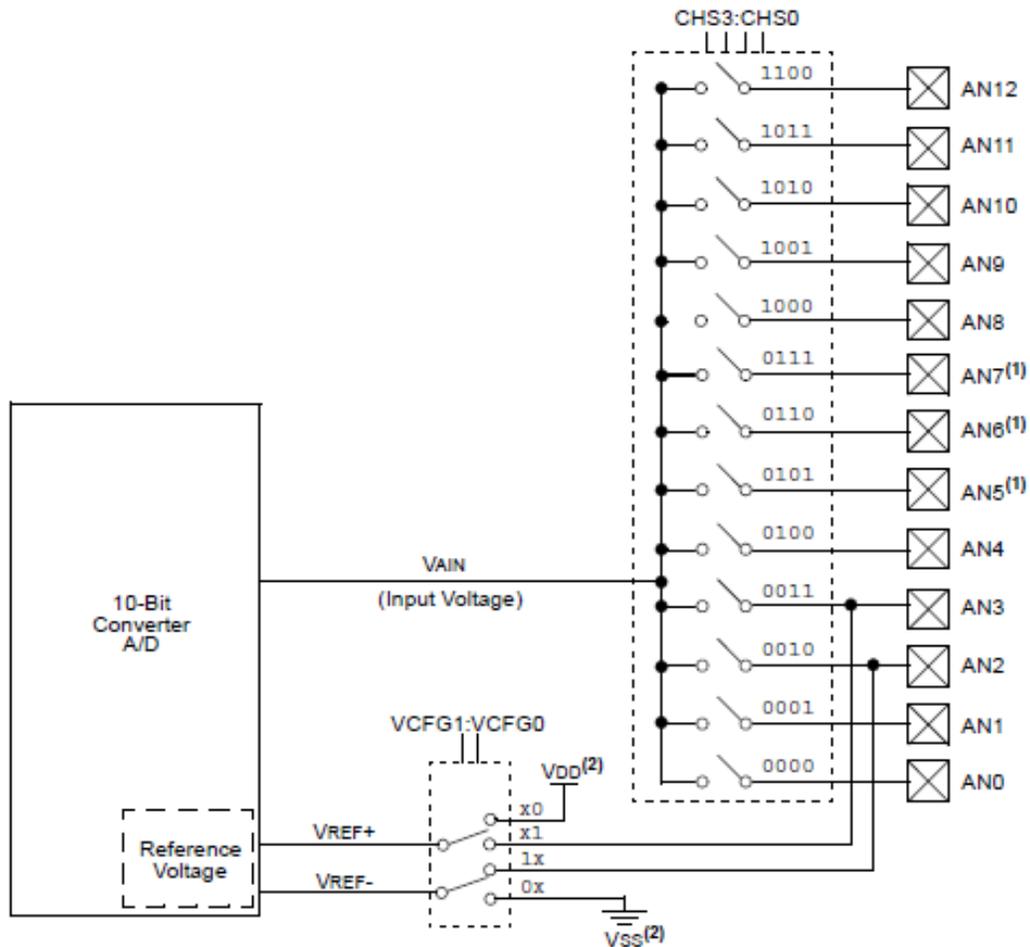
Fig. 2.11 Cálculo Baud Rate

2.4.1.3 ADC del PIC18F4550

El módulo convertidor analógico-digital tiene 13 canales. Este módulo permite la conversión de una señal analógica en la entrada a un número correspondiente de 10 bits.

En los registros ADRESH y ADRESL se almacenan los primeros 8 bits y los últimos 2 bits de la palabra de 10 bits generada por la conversión respectivamente. El registro ADCON0 controla la operación del módulo A/D.

Cada puerto asociado con el convertidor A/D puede ser configurado como entrada digital o analógica, este convertidor tiene la característica única de poder ser operado mientras el dispositivo está en modo “sleep”. En la figura 2.12 se muestra el diagrama a bloques del convertidor A/D.



- Note 1:** Channels AN5 through AN7 are not available on 28-pin devices.
Note 2: I/O pins have diode protection to VDD and VSS.

Fig. 2.12 Diagrama a bloques del convertidor A/D

2.4.1.4 Interrupciones del PIC18F4550

El microcontrolador PIC18F4550 tiene múltiples fuentes de interrupción, una característica es la de interrupción de prioridad que permite asignar alta o baja prioridad a cada una de las interrupciones.

Existen 10 registros los cuales son usados para el control de las interrupciones, los registros son:

- RCON*
- INTCON*
- INTCON2*
- INTCON3*
- PIR1, PIR2*
- PIE1, PIE2*
- IPR1, IPR2*

El puerto B tiene interrupciones de cambio de estado, es decir, cuando el puerto B reconoce un cambio de estado lógico se activa una interrupción, estas interrupciones solo se activa en cambios de estado del puerto RB7, RB6, RB5 y RB4. La bandera de interrupción es RBIF.

2.5 Tecnologías y protocolos inalámbricos

La norma IEEE 802.15.4 se encarga de normalizar la tecnología inalámbrica de bajo alcance en redes inalámbricas de área personal ya sea con Bluetooth, ZigBee, etc.

2.5.1 Bluetooth

Bluetooth es un protocolo inalámbrico que permite la conexión entre dos dispositivos, se ha caracterizado por su bajo consumo de energía, además de un alcance considerable entre 5 y 20 metros dependiendo de los obstáculos que se encuentren en el medio. Otra característica sobresaliente es su baja complejidad en cuanto a hardware y en consecuencia su bajo costo. El logo del Bluetooth es una “B”.



Fig. 2.13 Logo Bluetooth

El Bluetooth opera a una frecuencia de 2.4 GHz y opera en diferentes configuraciones tales como:

- Transmisión omnidireccional*
- Comunicación punto a punto*
- Comunicación de datos y voz.*

El principio de funcionamiento está basado en una comunicación full-dúplex y una modulación GFSK (Gaussian Frequency Shift Keying). GFSK es una modulación donde un 1 lógico es representado mediante una desviación positiva (incremento) de la frecuencia de la onda portadora, y un 0 mediante una desviación negativa (decremento) de la misma.

2.5.1.1 Módulo HC-06

Los módulos de Bluetooth HC-05 y HC-06 son módulos muy populares para aplicaciones con microcontroladores PIC y Arduino. Se trata de dispositivos relativamente económicos y que habitualmente se venden en un formato que permite insertarlos en un protoboard y cablearlo directamente a cualquier microcontrolador, incluso sin realizar soldaduras.

Una de las principales ventajas del módulo HC-06, además de su pequeño tamaño y sus buenas características de transmisión y recepción que le brindan un alcance muy amplio (por tratarse de un sistema local Bluetooth), es el bajo consumo de corriente que posee tanto en funcionamiento, como en modo de espera, es decir, alimentado con energía, pero sin conexión o enlace a otro dispositivo.



Fig. 2.14 Módulo Bluetooth HC-06

Características generales del módulo HC-06

- Voltaje de alimentación 3.6V - 6 V*
- Antena PCB integrada*
- Led indicador de estado*

Cabe mencionar que el módulo HC-06 es un módulo esclavo, mientras que el HC-05 es maestro y requiere ser programado mediante una hiperterminal para darle un nombre y una contraseña.

2.6 Sistema operativo Android

Android es el sistema operativo para dispositivos móviles el cual no solo se encuentra en teléfonos celulares, tablets, televisores, relojes inteligentes y automóviles último modelo.



Fig. 2.15 Logo de Android

La plataforma de hardware principal de Android es la arquitectura ARM. Hay soporte para x86 en el proyecto Android-x86, y Google TV utiliza una versión especial de Android x86.

Android al contrario de otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se desarrolla de forma abierta y se puede acceder tanto al código fuente como a la lista de incidencias donde se pueden ver problemas aún no resueltos y reportar problemas nuevos.

El que se tenga acceso al código fuente no significa que se pueda tener siempre la última versión de Android en un determinado móvil, ya que el código para soportar el hardware de cada fabricante normalmente no es público, así que faltaría una parte básica del firmware para poder hacerlo funcionar en dicha terminal, y porque las nuevas versiones de Android suelen requerir más recursos, por lo que los modelos más antiguos quedan descartados por razones de memoria (RAM), velocidad de procesador, etc.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas Java en máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas están escritas en lenguaje C e incluyen:

- Un administrador de interfaz gráfica*
- Un framework OpenCore*
- Una base de datos relacional SQLite*
- Una interfaz de programación de API gráfica*
- Un motor de renderizado Webkit*
- Un motor gráfico SGL*

-SSL

-Una biblioteca estándar de C Bionic

Las actualizaciones del sistema operativo Android típicamente arreglan errores y generan nuevas funciones. Generalmente cada actualización del sistema operativo Android es desarrollada bajo un nombre en código de un elemento relacionado a postres. En la tabla 2.5 se muestran las actualizaciones de las versiones de Android.

Así como avanzan las actualizaciones de las versiones, las características del sistema operativo van mejorando, en la tabla 2.6 se mencionan algunas de las características más notables del sistema operativo Android.

Tabla 2.5 Actualizaciones de Android

Nombre Actualización	Numero de versión	Traducción
Apple pie	1.0	Tarta de manzana
Banana Bread	1.1	Pan de plátano
Cupcake	1.5	Panque
Donut	v1.6	Rosquilla
Eclair	v2.0/v2.1	Pepito
Froyo	v2.2	Yogurt Helado
Gingerbread	v2.3	Pan de jengibre
Honeycomb	v3.0/v3.1/v3.2	Panal de miel
Ice Cream Sandwich	v4.0	Sándwich helado
Jelly Bean	v4.1/v4.2/v4.3	Gominola
Kit kat	v4.4	Kit kat
Lollipop	v5.0/v5.1	Paleta
Marshmallow	v6.0	Malvavisco

Tabla 2.6 Características del sistema operativo Android

Característica	Descripción
Diseño de dispositivo	La plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D y 3D
Almacenamiento	SQLite, una base de datos liviana, que es usada para propósitos de almacenamiento de datos
Conectividad	Android soporta las siguientes tecnologías y protocolos de conectividad: GSM, Bluetooth, WiFi, entre otros.
Hardware adicional	Android soporta cámara fotográfica y video, pantallas táctiles, GPS, acelerómetros, sensores de proximidad, sensor de luz, etc.
Entorno de desarrollo	Incluyen emulador de dispositivos, herramientas de depuración de memoria, y análisis de rendimiento de software, algunas plataformas son: Eclipse, Android Studio y App Inventor.

2.6.1 Aplicaciones para Android

Las aplicaciones se desarrollan habitualmente en lenguaje Java con Android Software Development Kit (Android SDK), pero están disponibles otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C o C++, Google App Inventor, un entorno visual para programadores novatos y varios marcos de aplicaciones basadas en la web multiteléfono.

El desarrollo de aplicaciones para Android no requiere de lenguajes complejos de programación, con el lenguaje Java y estar en posesión del kit de desarrollo de software o «SDK» provisto por Google el cual se puede descargar gratuitamente, se puede iniciar el desarrollo de aplicaciones.

Todas las aplicaciones están comprimidas en formato APK, que se pueden instalar sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos.

En la figura 2.16 se muestra la pantalla principal de un teléfono celular con sistema operativo Android, también se puede apreciar algunas de las aplicaciones ya instaladas.



Fig. 2.16 Pantalla principal con S.O. Android

“Google Play Store” es la tienda en línea de software desarrollado por Google para dispositivos Android. Una aplicación llamada “Play Store” que se encuentra instalada en la mayoría de los dispositivos Android y permite a los usuarios navegar y descargar aplicaciones publicadas por los desarrolladores. Google retribuye a los desarrolladores el 70% del precio de las aplicaciones.

Por otra parte, los usuarios pueden instalar aplicaciones desde otras tiendas virtuales, tales como Appstore o SlideME o directamente en el dispositivo si se dispone del archivo APK de la aplicación.

2.6.2 MIT App Inventor

MIT App Inventor es una herramienta de programación patrocinada por el Instituto Tecnológico de Massachusetts (MIT). App Inventor está basado en bloques que permiten a los usuarios programar y construir aplicaciones completamente funcionales para dispositivos Android. App Inventor funciona como un servidor web administrado por personal del MIT's Center for Mobile Learning.



Fig. 2.17 Logo de MIT App Inventor

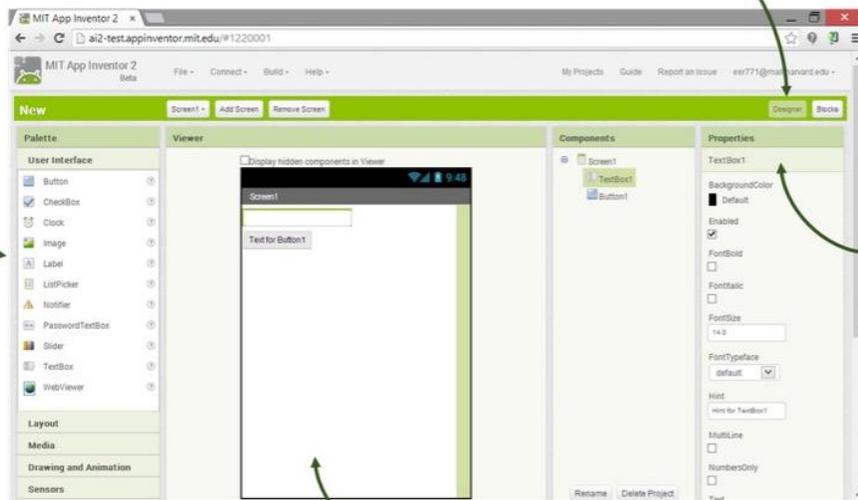
2.6.3 Editor de bloques y diseñador en App Inventor

En la plataforma de desarrollo de App Inventor se tienen dos entornos en los cuales se deberán agregar elementos gráficos, este es llamado Diseñador, mientras que en el otro entorno llamado Editor de bloques se deberán unir los bloques correspondientes al elemento seleccionando acorde a la función a realizar por dicho elemento.

En la figura 2.18 se muestra la plataforma de desarrollo junto con algunos comandos básicos para su manejo.

Paleta: Encuentra tus componentes y agrégalos a la pantalla de tu App.

Botón de diseñador: Da click para cambiar a modo Diseñador o Editor bloques



Propiedades: Selecciona un componente para cambiar sus propiedades, como color o tamaño.

Viewer: Dibuja componentes de la paleta para verlos en la App.

Fig. 2.18 Entorno de App Inventor

Cuando se han seleccionado los componentes que se desean en la aplicación, se puede cambiar a modo Editor de bloques donde se podrán enlazar los bloques para hacer que la aplicación responda según lo decida el programador.

Bloques generales: bloques generales para comportamientos y sentencias comunes

Botón de Bloques: Da click para cambiar a modo de Bloques o Diseñador.



Fig. 2.19 Editor de bloques de App Inventor

Cuando se ha compilado correctamente la aplicación, el usuario tiene dos opciones para instalarla en el dispositivo; se puede descargar directamente por medio de una conexión del celular a la computadora, la segunda opción es usando una aplicación que pueda leer códigos QR, como lo es QR Droid, la aplicación se muestra en la figura 2.20, con esta herramienta solo se requiere de un escaneo para descargar la aplicación e inmediatamente instalarla.

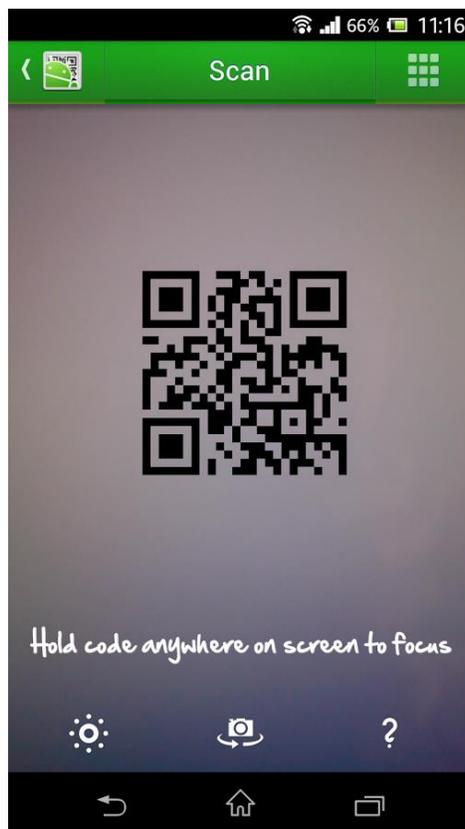


Fig. 2.20 Aplicación QR Droid

Capítulo 3
Diseño e Implementación

En este capítulo de definirán las características generales del sistema a desarrollar, así como los bloques que lo conforman; finalmente se diseñaran cada uno de ellos y se desarrollaran los programas (Software) correspondientes.

3.1 Características generales del sistema

El sistema de monitoreo de ritmo cardiaco que se pretende diseñar requiere de un sensor no invasivo el cual proporcione información en tiempo real de los latidos de corazón de algún usuario, dicha información debe ser procesada por un dispositivo digital, es por ello que se requiere de un microcontrolador o microprocesador el cual necesitará estar asociado a un módulo inalámbrico. Esta información será enviada a través de un módulo Bluetooth a una interfaz gráfica; en este caso será un aplicación para el sistema operativo Android en la cual podremos visualizar de manera numérica el ritmo cardiaco del usuario, para esta aplicación es necesario contar con un teléfono celular con sistema operativo Android, cabe mencionar que si el celular no cuenta con la instalación de la aplicación que se diseñará no funcionará el sistema.

3.2 Diagrama a bloques

Para realizar las funciones antes mencionadas se requiere de un sensado del ritmo cardiaco, se hará por medio de un sensor de fotoplethismografía. La señal de respuesta será una señal analógica, la cual será acondicionada a niveles lógicos.

Después mediante un microcontrolador se interpretará la señal digital para darle un valor al ritmo cardiaco, dicha información será enviada vía Bluetooth con un módulo asociado al microcontrolador. De igual forma se tendrá un sensado permanente del voltaje de suministro para garantizar que el nivel de voltaje es adecuado para una medición correcta.

El dispositivo Android, como lo puede ser un celular o una tablet tendrá instalada la aplicación diseñada en App Inventor, con esta interfaz gráfica el usuario tendrá el monitoreo de su ritmo cardiaco.

El diagrama a bloques de la figura 3.1 muestra el funcionamiento del sistema de monitoreo. A continuación se describirán todas las etapas que lo componen:

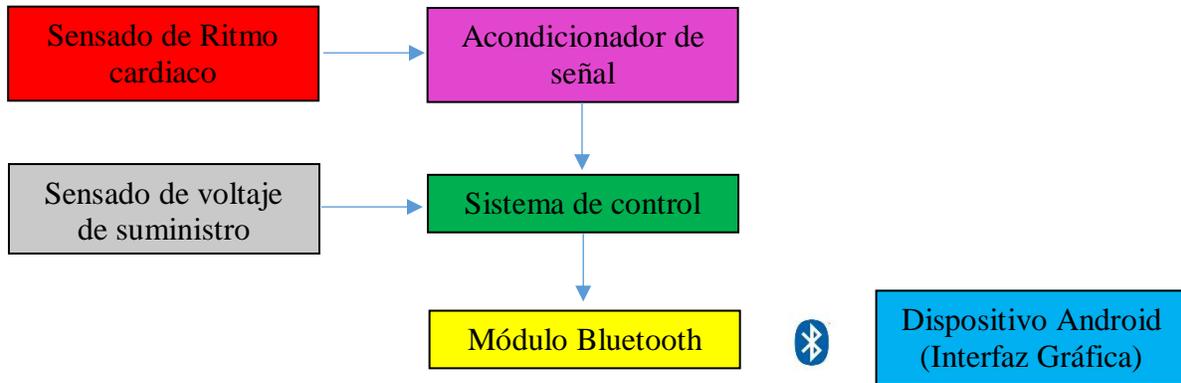


Fig. 3.1 Diagrama a bloques del sistema

3.3 Caracterización del sensor de ritmo cardiaco

Como primera etapa del sistema se implemento con diferentes personas el sensor de ritmo cardiaco, esto con el fin de conocer las señales eléctricas que proporciona y así encontrar patrones y diferencias. El sensor se alimentó y se conectó con un canal del osciloscopio para ver las señales en tiempo real. En la figura 3.2 se muestra una señal de una persona con el sensor en el lóbulo de la oreja.

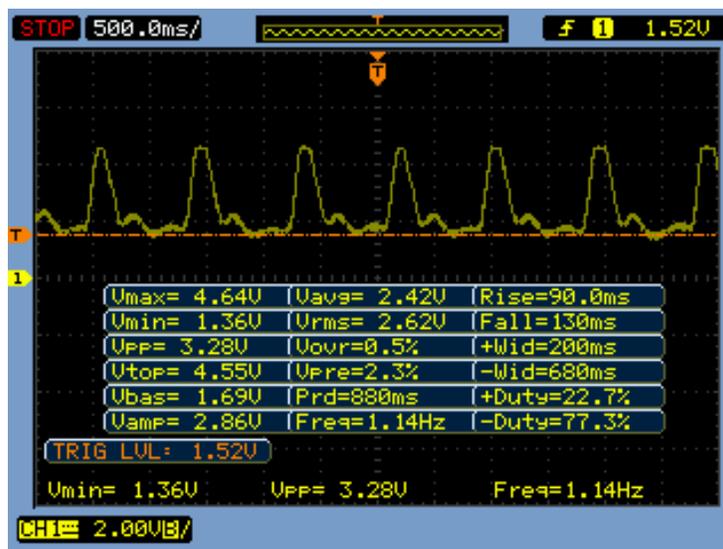


Fig. 3.2 Señal proporcionada por el sensor de ritmo cardiaco en una persona

La señal tiene un frecuencia de 1.14 Hz, lo que sería un pulso en promedio por segundo, es decir, el ritmo cardiaco de la persona sería un poco más de 60 pulsaciones por minuto. La señal tiene valores máximos y mínimos de voltaje; 4.64v y 1.36v respectivamente, pero no tiene una forma cuadrada, sino es una señal analógica.

En la figura 3.3 se muestra otra captura de la respuesta del sensor, está captura se generó con otra persona colocando el sensor en el lóbulo de la oreja, como frecuencia se obtuvo 1.47 Hz y los niveles de voltaje fueron 4.32v y 1.68v los cuales son muy parecidos, a los niveles de la persona anterior.

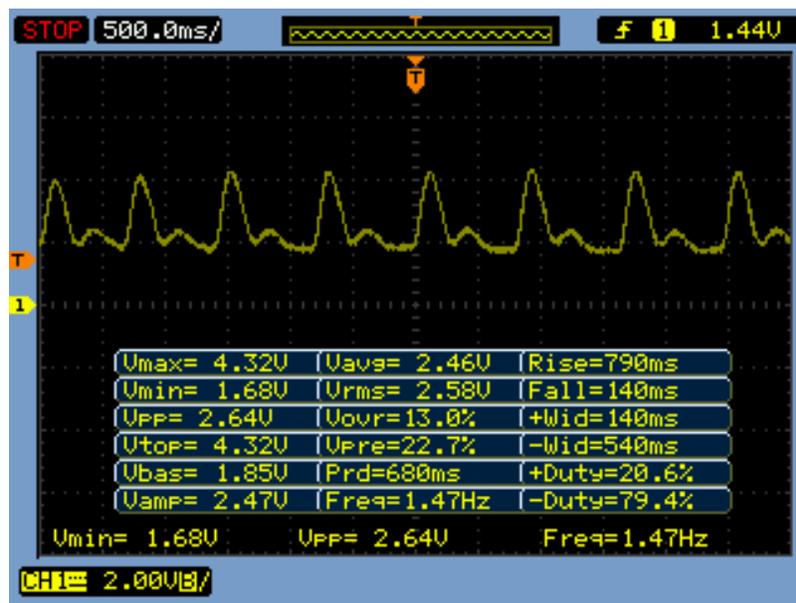


Fig. 3.3 Señal proporcionada por el sensor de ritmo cardiaco en una segunda persona

Esta misma implementación se realizó con otras personas de diferentes edades y condición física, sirvió para conocer las características de la señal y con ellas poder diseñar un acondicionar de señales adecuado y así generar una señal con niveles de cero y uno lógicos, ya que siempre que se coloque la señal en el lóbulo se obtendrá un nivel de voltaje máximo y mínimo de entre 4.5v y 1.5v en promedio directamente del sensor.

3.3.1 Acondicionador de señal

Los acondicionadores de señal son circuitos que convierten los parámetros eléctricos de salida de los sensores o transductores en una señal eléctrica con parámetros mejor definidos o que se pueden medir fácilmente. El conjunto sensor-acondicionador es la interfaz entre la variable física y la entrada del circuito electrónico que vaya a procesar la información contenida en dicha señal.

Existen diferentes motivos por los cuales las señales de los sensores deben ser acondicionadas, por ejemplo, niveles de amplitud mayores o menores a los que el circuito de procesamiento puede responder. En este caso para la señal que se requiere es una señal con niveles lógicos y no una señal analógica como se observó en las figuras 3.2 y 3.3.

En la primera etapa se tiene un amplificador operacional TL081 configurado como comparador, en la entrada negativa se tiene un resistor variable para generar un nivel de CD de 3.35v, este nivel de CD será la referencia para detectar solo las partes de mayor amplitud de la señal generada por el sensor de ritmo cardiaco, esto porque en algunas personas las señales de ritmo cardiaco contienen un doble pico, el cual está definido por el nivel de oxigenación de sangre de la persona.

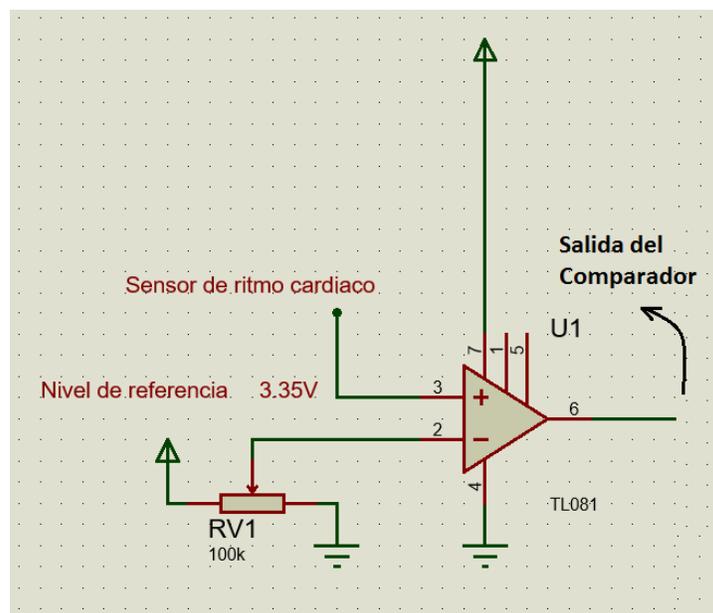


Fig. 3.4 Circuito del comparador

La señal producida a la salida del comparador se ilustra en la figura 3.5, como se observa, la señal tiene una forma de onda casi cuadrada, pero no tiene niveles cercanos a 5 y 0 volts, sino de 2.75 v y 0.8 v, lo cual puede producir que el microcontrolador no detecte de forma correcta los estados lógicos de 1 y 0.

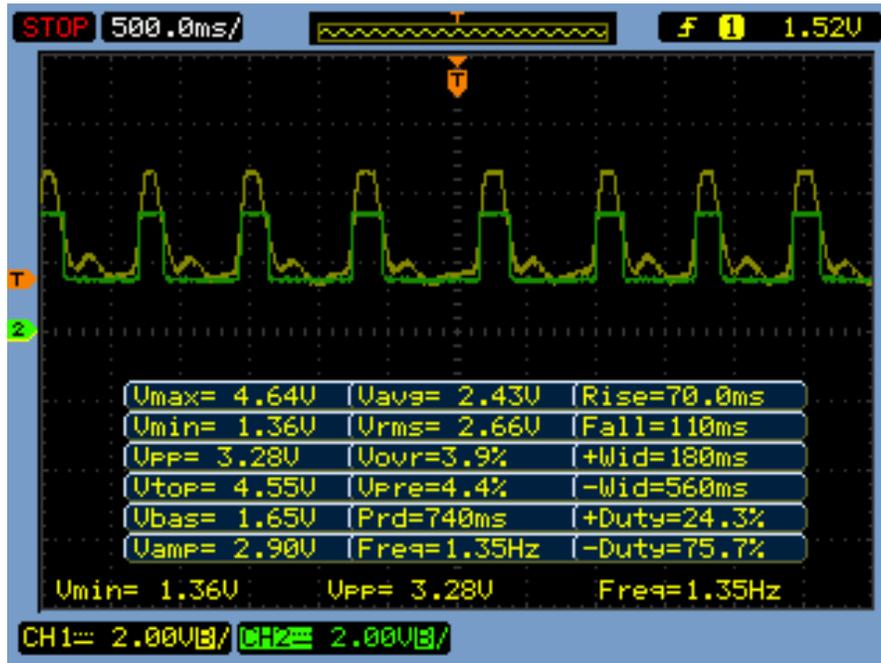


Fig. 3.5 Señal del sensor y señal obtenida al compararla

Debido a lo anterior se generó una estrategia para acondicionar la señal, se diseñó un circuito donde la salida del comparador fuera conectada a un transistor configurado como interruptor para conseguir una señal con niveles cercanos a los niveles aceptados por TTL, después un transistor adicional, de igual forma configurado como interruptor para lograr que los niveles de la señal más altos o picos del sensor coincidieran con los niveles positivos de la señal final.

En la figura 3.6 se muestra el diagrama generalizado del acondicionar de señal diseñado. En la primera parte se tiene el comparador antes mencionado, seguido de un primer interruptor y después el segundo y último interruptor, ambos implementados con transistores 2N3904.

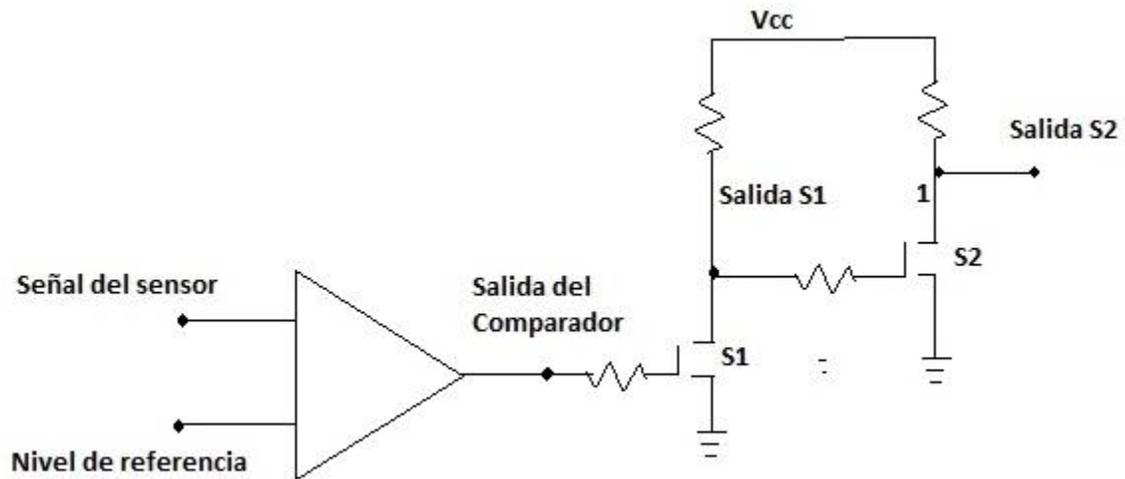


Fig. 3.6 Diagrama generalizado del acondicionador de señales

Las formas de onda esperadas del acondicionador anterior se ilustran en la figura 3.7. En la figura A se muestra la señal obtenida del sensor, la cual es la señal analógica ya mencionada previamente, en la figura B se ilustra la salida del comparador, una señal cuadrada montada sobre un nivel de CD, en la figura C se obtiene la forma de onda resultante del primer interruptor S1, por último en la figura D se obtiene la señal final con niveles positivos correspondientes a los picos de la señal generada por el sensor, la cual es proporcionada por el interruptor S2.

Los interruptores S1 y S2 se implementaron por medio de transistores configurados como interruptores, es decir, operan en la zona de corte y saturación, a estos circuitos se le puede llamar “circuitos de conmutación”.

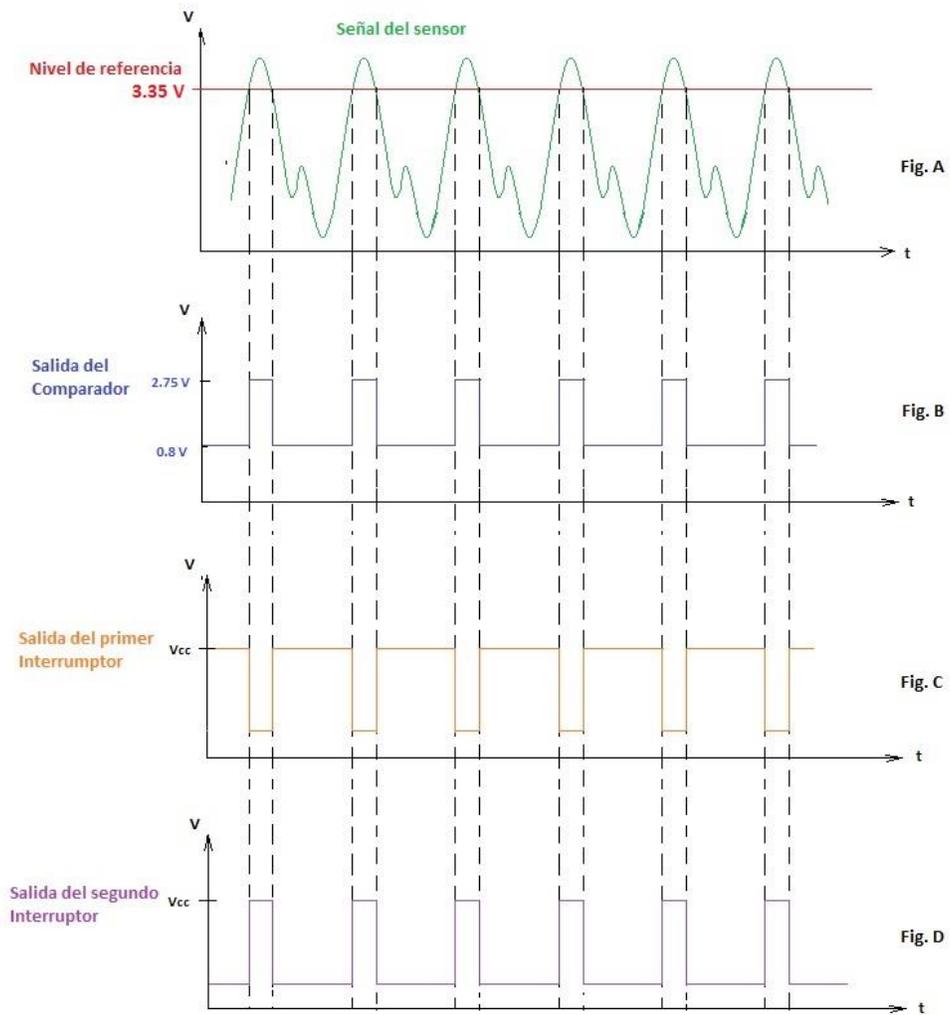


Fig. 3.7 Formas de onda del acondicionador de señales

3.3.2 Diseño de circuitos de conmutación S1 y S2

Para el diseño se utilizó el transistor 2N3904 cuyos voltajes de operación para corte y saturación se muestran en la tabla 3.1, estos valores han sido obtenidos de la hoja de especificaciones proporcionada por el fabricante.

Tabla 3.1 Datos proporcionados por la hoja de datos del transistor 2N3904

VOLTAJES DE UNIONES DE TRANSISTOR 2N3904		
$V_{CE_{SAT}}$	$V_{BE_{SAT}}$	β
0.2	0.85	100

El circuito se muestra en la figura 3.8, como se puede observar involucra el cálculo de dos resistores R_c y R_b , los valor de V_{cc} y V_{in} son conocidos, siendo V_{cc} el voltaje regulador y V_{in} el voltaje mayor de la señal proporcionada por el sensor.

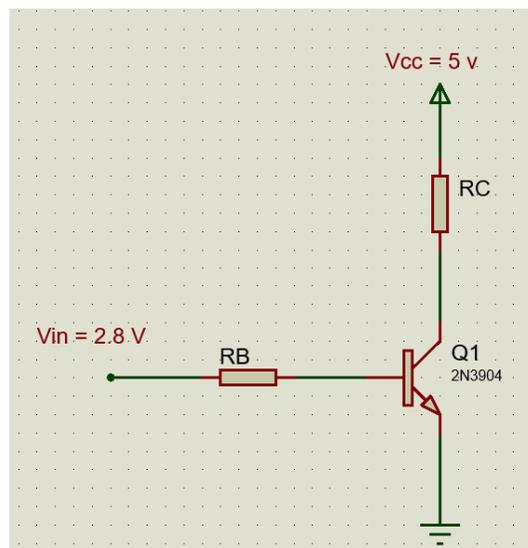


Fig. 3.8 Circuito de conmutación

Análisis por rama:

Saturación: $V_{cc} = R_c I_C + V_{CESAT}$ 1

De la ecuación 1 se despeja R_c y se obtiene la ecuación 2:

$$R_c = \frac{V_{cc} - V_{CESAT}}{I_C}$$
 2

Sustituyendo valores en la ecuación 2 y proponiendo 20 mA para I_C se tiene:

$$R_C = \frac{V_{CC} - V_{CESAT}}{I_C} = \frac{5 \text{ v} - .2 \text{ v}}{20 \text{ mA}} = 240 \Omega$$

Por lo tanto $R_C=240 \Omega$. Para el resistor R_B se ha usado el siguiente análisis;

Con la ecuación 3 se calculó la corriente de la base del transistor I_B ;

$$I_B = \frac{I_C}{\beta} \dots\dots\dots 3$$

Sustituyendo valores:

$$I_B = \frac{I_C}{\beta} = \frac{20 \text{ mA}}{100} = 200 \mu A$$

Con el valor de $I_B = 200 \mu A$ se puede hacer el análisis de la rama con la ecuación 4;

$$V_{IN} = R_B I_B + V_{BESAT} \dots\dots\dots 4$$

Despejando R_B se obtiene la ecuación 5;

$$R_B = \frac{V_{IN} - V_{BESAT}}{I_B} \dots\dots\dots 5$$

Sustituyendo valores en la ecuación 5 se obtiene;

$$R_B = \frac{V_{IN} - V_{BESAT}}{I_B} = \frac{2.8 \text{ v} - 0.85 \text{ v}}{200 \mu A} = 9750 \Omega$$

Con lo anterior obtenemos $R_B = 9.75k\Omega$

Con los cálculos anterior se logró definir valores a los resistores, seleccionando los valores cercanos a los comerciales; siendo $R_c=220\Omega$ y $R_b =10k\Omega$.

En la figura 3.9 se muestra el comparador con el primer circuito de conmutación.

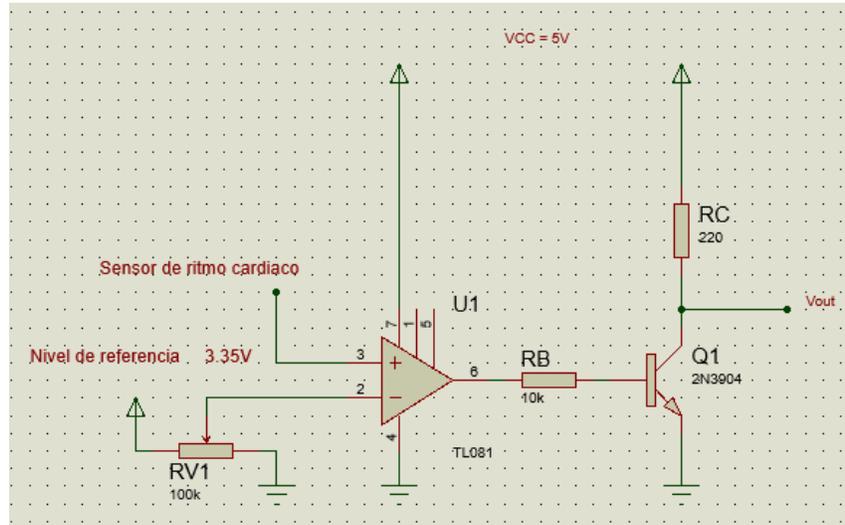


Fig. 3.9 Comparador con primer circuito de conmutación

La señal de salida de esta etapa se ilustra en la figura 3.10. En esta imagen se puede observar que se ha invertido los niveles de la señal como se esperaba, ya que los picos de la señal del sensor ahora corresponden a el nivel bajo de la señal entregada a la salida del circuito de conmutación; sin embargo los niveles que maneja la señal de salida son de 3.9 v y 1.4 v.

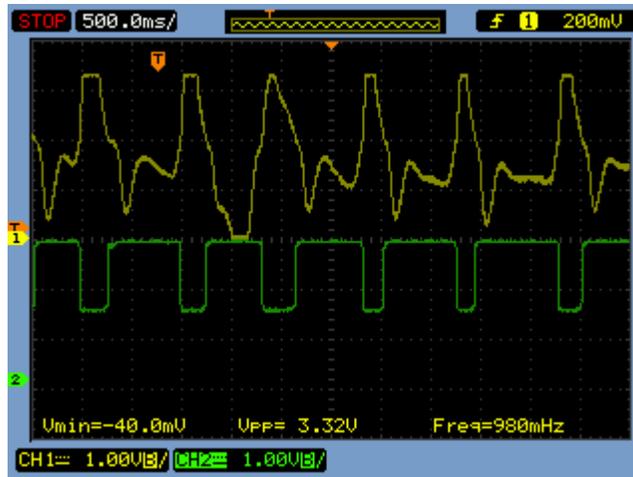


Fig. 3.10 Señal de salida de comparador con primer circuito de conmutación

Para el segundo circuito de conmutación se hizo uso de las ecuaciones 1, 2, 3, 4 y 5 para obtener los valores de los resistores R_b y R_c , tomando en cuenta que $V_{in} = 3.9$ v. Los valores de los resistores fueron $R_c = 220 \Omega$ y $R_b = 15.25$ k Ω .

En la figura 3.11 se muestra el circuito completo del acondicionador que se utilizó para el procesamiento de la señal analógica proporcionada por el sensor de ritmo cardiaco. Se ha añadido un resistor en serie con un led; si existen 0 volts entre el resistor y el led, dicho resistor estará aterrizado a tierra y se requiere que no demande una cantidad corriente considerable; si se selecciona una corriente pequeña de 300 μ A se tiene;

$$R_6 = \frac{V_s - V_d}{I} = \frac{5V - 1.8V}{300\mu A} = 10.66 \text{ k}\Omega \dots\dots\dots 6$$

La salida del segundo transistor estará unida mediante un resistor al nodo formado entre el diodo y el resistor de 10 k Ω , se ha seleccionado de la siguiente forma para que aporte la mitad de la corriente que requiere el led, es decir, 5mA, y se tiene un nivel de voltaje máximo de 4.96 v;

$$R_5 = \frac{V - V_d}{I} = \frac{4.96v - 1.8V}{5mA} = 633 \Omega \dots\dots\dots 7$$

Por lo tanto el valor comercial que se requiere es de 680 Ω .

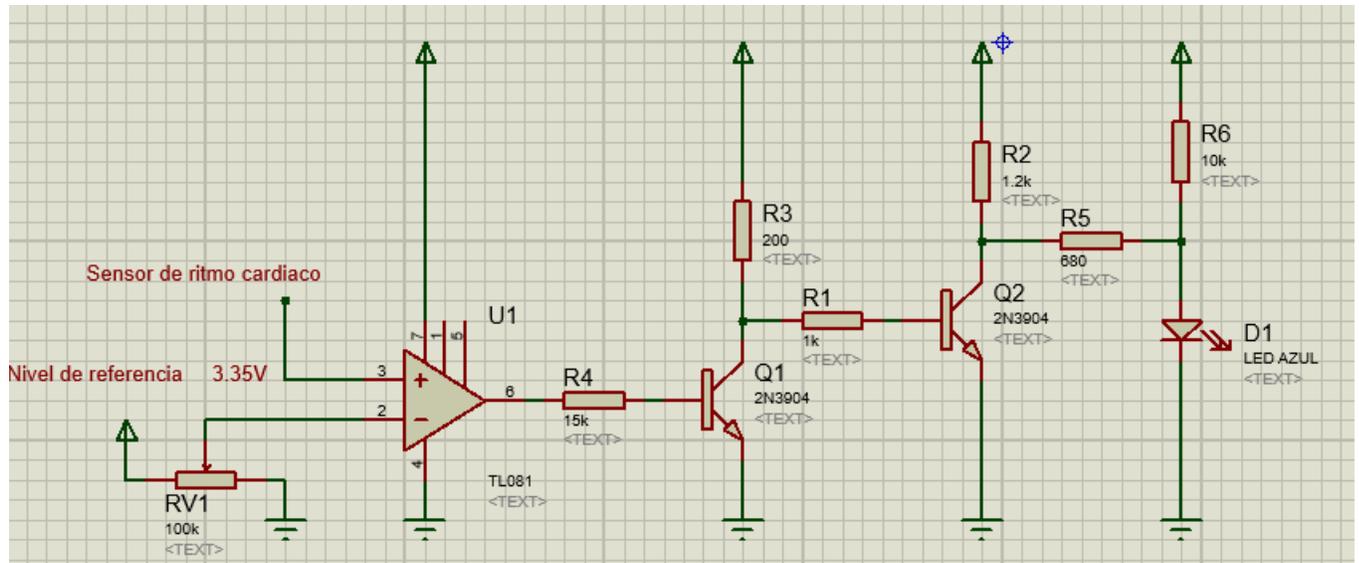


Fig. 3.11 Acondicionador de señal

Una vez diseñado el acondicionador de señal en su totalidad se estimuló con el sensor y la respuesta fue una señal cuadrada con valores más cercanos a los que trabaja TTL, en la figura 3.12 se muestran las formas de onda, y los niveles de la señal cuadrada, siendo de 4.96 v y 1.36 v.

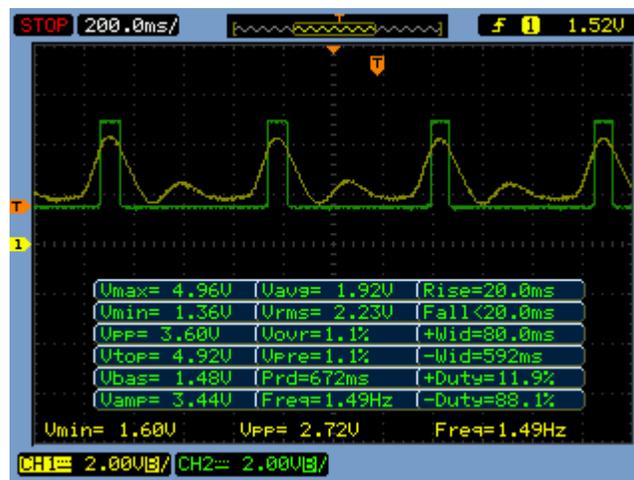


Fig. 3.12 Señal acondicionada para el microcontrolador

3.4 Consideraciones técnicas y cálculos finales para los valores de resistores asociados al microcontrolador

3.4.1 Diseño para el suministro y sensado de la batería

El voltaje de la batería Li-po es de 7.4 volts o menos dependiendo de la carga y 250 mAh según el fabricante, en la figura 3.13 se ilustra dicha batería, sin embargo en pruebas experimentales la batería ha llegado a cargar hasta 9 v. Es por ello que para alimentar todo el sistema se ha usado un regulador 7805, el cual proporciona un voltaje fijo regulado de 5v con el cual se puede energizar todo el circuito. Para hacer la lectura del voltaje real de la batería se ha diseñado un divisor de voltaje el cual asegurará que el voltaje que llega al convertidor analógico-digital del microcontrolador no supere los 5v para no dañarlo.



Fig. 3.13 Bateria Li-po para el suministro eléctrico del circuito

El circuito de la figura 3.14 se muestra el arreglo de resistores usados para asegurar la protección del microcontrolador, se ha propuesto una resistencia de $R_{11}=10\text{ k}\Omega$ en serie con otra resistencia (R_{10}) de valor a calcular. Tomando en cuenta que el valor máximo de voltaje que tendrá el resistor de $10\text{ k}\Omega$ será de 5v para el ADC, se hace el cálculo de la corriente por ley de Ohm;

$$I = \frac{V}{R_{11}} = \frac{5V}{10K\Omega} = 500 \mu A \dots\dots\dots \mathbf{8}$$

La corriente obtenida con la ecuación 8 es la misma que circulará por el resistor de valor desconocido. El voltaje de dicho resistor es la diferencia del voltaje de suministro mayor, el cual es cuando la batería está completamente cargada, y los 5 volts máximos deseados;

$$\text{Voltaje de resistor desconocido} = V_s - 5v = 9v - 5v = 4v \dots\dots \mathbf{9}$$

Teniendo el voltaje y la corriente que circulan por el resistor, despejando R de la ley de Ohm se puede conocer el valor;

$$R_{10} = \frac{V}{I} = \frac{4v}{500 \mu A} = 8 K\Omega \dots\dots\dots \mathbf{10}$$

El resistor desconocido ahora tiene un valor de 8 kΩ, sin embargo ese valor no existe comercialmente hablando, por lo tanto se ha tenido que usar el mas cercano que es de 8.2 kΩ.

Con lo anterior, aseguramos que entre los resistores R₁₁ y R₁₀ siempre exista un voltaje de 5v o menos, el cual será conectado al puerto del ADC.

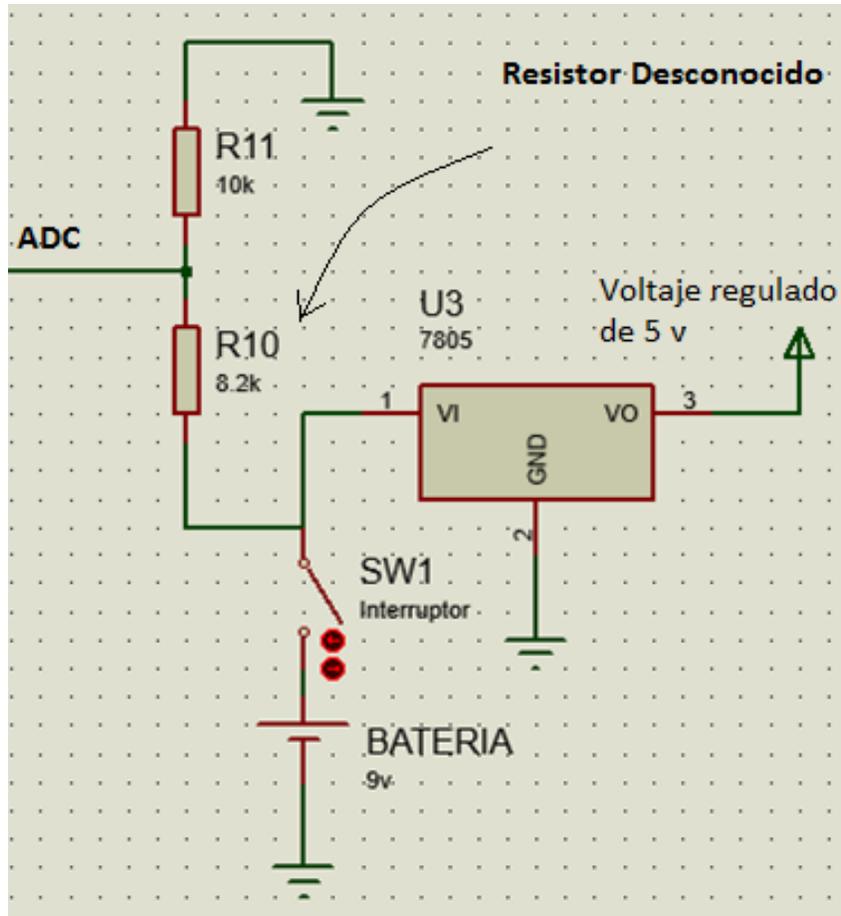


Fig. 3.14 Arreglo de resistores para protección del microcontrolador

3.4.2 Cálculo de resistores para Leds de los puertos del microcontrolador

Los diodos emisores de luz, conocidos como “led”, tienden a consumir 1.8v y 2.2v si están hechos de Silicio, por lo tanto para conocer la resistencia de protección que deben llevar se requiere de un cálculo con la ley de Ohm, donde al voltaje de alimentación se le resta el voltaje del diodo, es decir 1.8v. La corriente a la que se debe someter un diodo para hacerlo iluminar de una forma considerable es de entre 10mA y 15mA, cabe mencionar que los pines del microcontrolador solo son capaces de dar hasta 25mA, ya mencionado lo anterior se realizó el cálculo con un valor de 10mA;

$$R = \frac{V_s - V_d}{I} = \frac{5\text{ v} - 1.8\text{ v}}{10\text{ mA}} = 320\ \Omega \dots\dots\dots 11$$

Por lo tanto el valor que se ha usado es de 330 Ω , debido a las restricciones comerciales en los valores de resistores.

3.5 Implementación del prototipo final

Una vez escrito el código que deberá tener grabado el microcontrolador y la aplicación para el celular, se puede llevar a cabo el montaje en placa del prototipo con el siguiente diagrama esquemático el cual contiene todas las partes que componen el circuito.

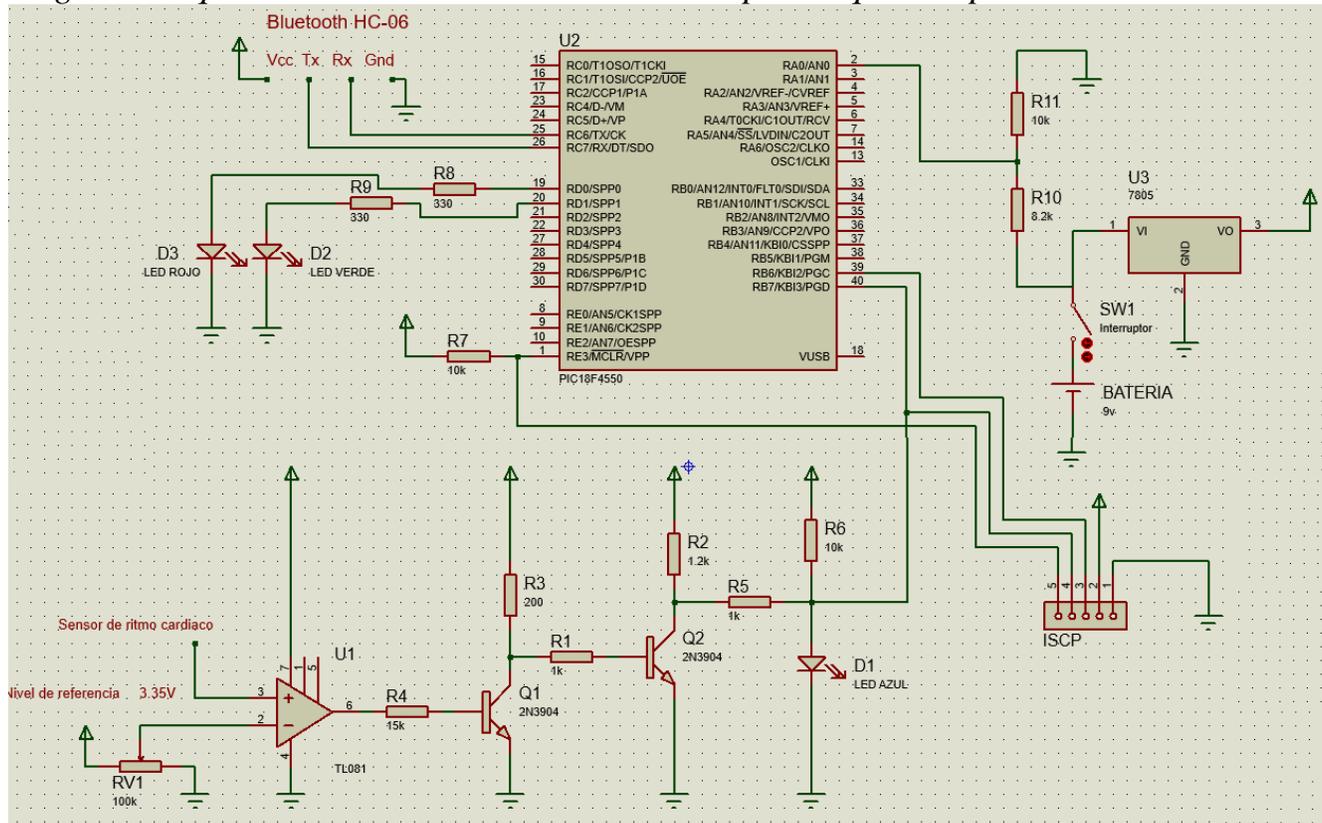


Fig. 3.15 Circuito final para el medidor de ritmo cardiaco

El diseño del PCB se hizo con ayuda del software Proteus 8 Professional, en la figura 3.16 se muestra la imagen de las pistas y los puntos de soldadura para el circuito, el diseño se hizo de tal forma que se distribuyan los componentes en ambas caras del circuito para poder reducir el tamaño y sea más cómodo de usar. En la figura 3.17 y 3.18 se muestran las simulaciones del circuito en tres dimensiones, esto es de gran ayuda para saber cómo quedará el prototipo final.

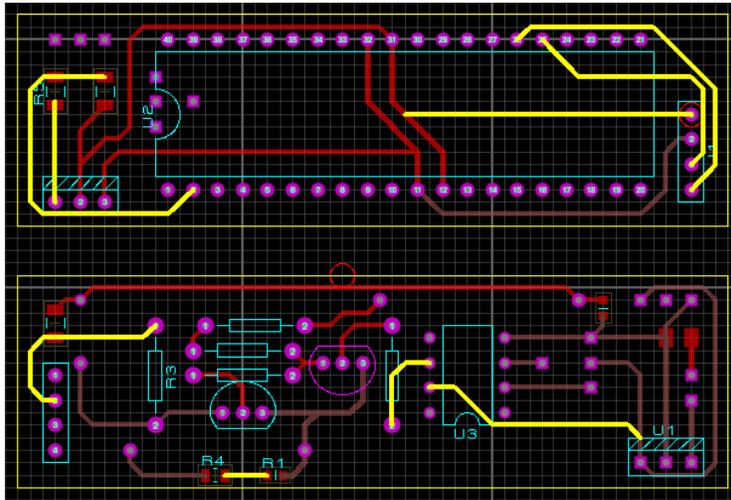


Fig. 3.16 PCB del circuito

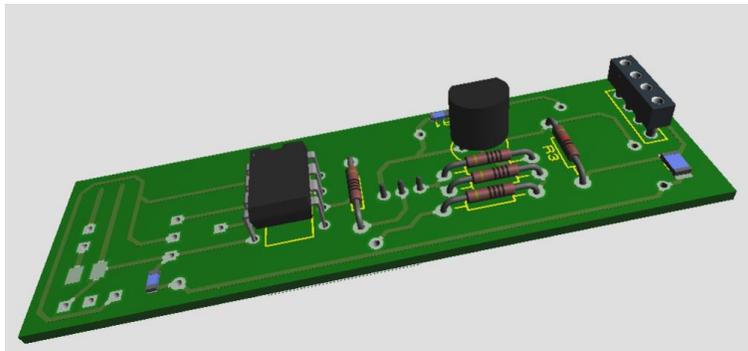


Fig. 3.17 Placa 3D simulada por la parte superior

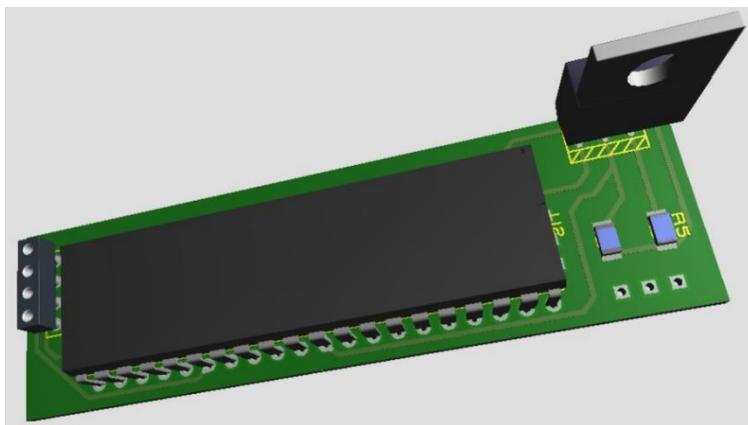


Fig. 3.18 Placa 3D simulada por la parte inferior

3.6 Diagrama de flujo

Para ilustrar el desarrollo y construcción del sistema se utilizará el diagrama de flujo de la figura 3.19, posteriormente se explicará cada una de las etapas.

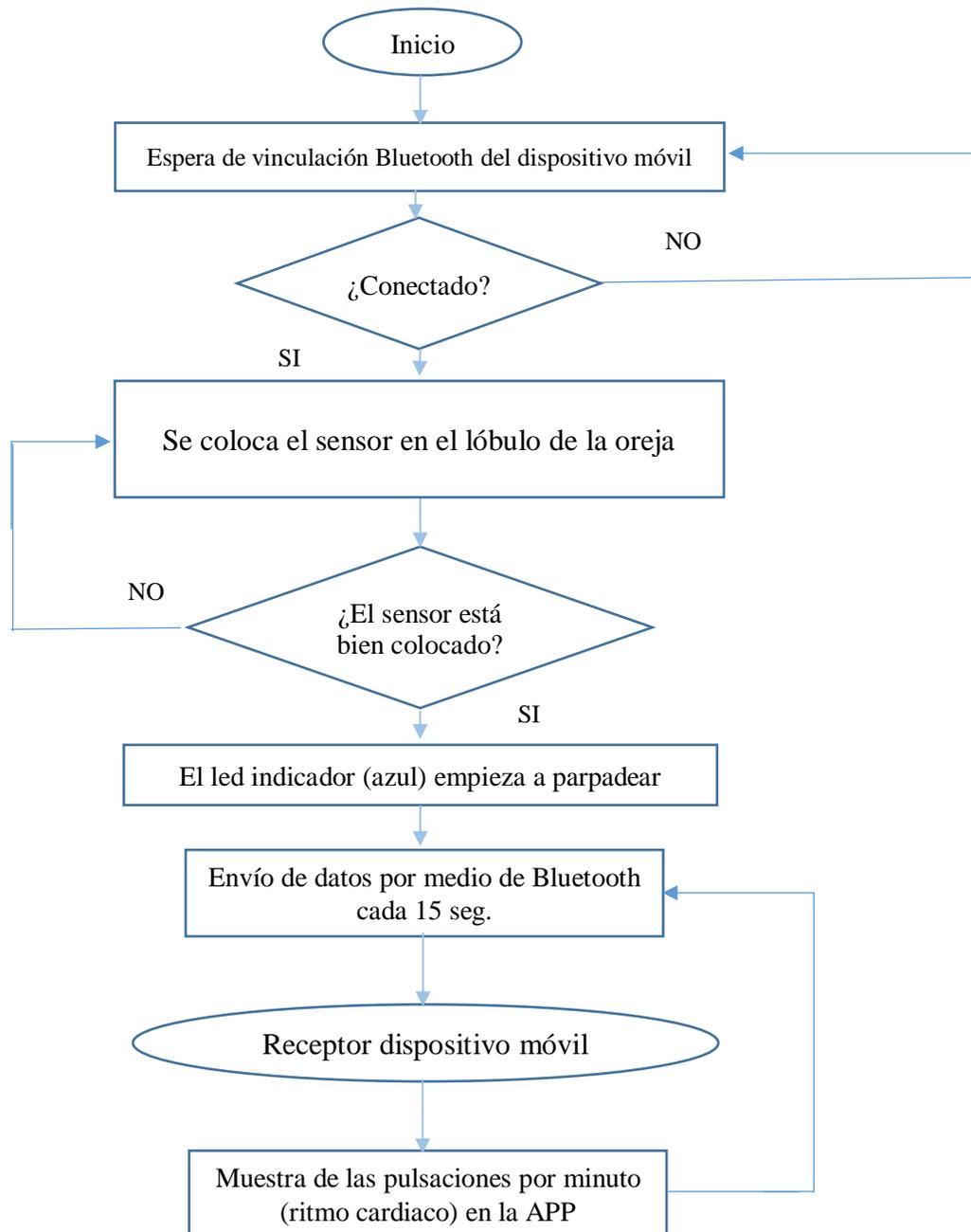


Fig. 3.19 Diagrama de flujo

En la primera etapa se inicia con la vinculación del Bluetooth con el dispositivo Android, una vez que se conecte exitosamente, se mostrará un mensaje de “CONECTADO” en el dispositivo móvil. En tiempo real se estará sensando el ritmo cardiaco, la señal proporcionada por el sensor se acondicionara a niveles de cero y uno lógicos, dicha señal se llevará a un puerto del microcontrolador para contar las veces que se interrumpe y así hacer el cálculo promedio de pulsos en un minuto, el cual será el valor del ritmo cardiaco de la persona.

El microcontrolador tendrá un canal del convertidor A/D el cual estará sensando el nivel de voltaje de la batería de suministro para garantizar que se llevará a cabo una medición correcta. El nivel de voltaje será acondicionado por medio de un divisor de voltaje para generar un nivel de 0v a 5v y así no dañar el microcontrolador.

Para lograr lo anterior primero se llevaron a cabo pruebas y experimentos para asociar el microcontrolador y el Bluetooth a una aplicación para celulares, de esta forma poder diseñar una aplicación final para hacer la lectura del ritmo cardiaco, después de ello se diseñará una estrategia para acondicionar el sensor y así escribir un programa para el microcontrolador, que enviará el valor del ritmo cardiaco por el módulo USART al Bluetooth.

3.7 Aplicación final para el medidor de ritmo cardiaco

En base a lo experimentado con el entorno de desarrollo de aplicaciones para Android, App Inventor, se ha diseñado la siguiente aplicación final para este proyecto, la cual será instalada en un teléfono celular para el monitoreo del ritmo cardiaco.

En la figura 3.20 se muestra cómo será la interfaz gráfica para el usuario en el celular o tablet.



Fig. 3.20 Diseño final de la aplicación en APP Inventor

El arreglo de bloques se conforma de la siguiente manera:

En la figura 3.21 se muestra la parte en donde se selecciona el dispositivo Bluetooth a conectar, una vez seleccionado se completa la acción con el botón “CONECTAR”, al momento de que es conectado de manera exitosa arrojará un mensaje diciendo “CONECTADO” y estará lista para su uso.

```

when Seleccion_de_bluetooth .BeforePicking
do set Seleccion_de_bluetooth . Elements to BluetoothClient1 . AddressesAndNames

when Seleccion_de_bluetooth .AfterPicking
do set CONECT . Text to Seleccion_de_bluetooth . Selection

when CONECTAR .Click
do set Seleccion_de_bluetooth . Selection to call BluetoothClient1 .Connect
address Seleccion_de_bluetooth . Selection
set CONECT . Text to " CONECTADO "

```

Fig. 3.21 Bloques correspondientes a la vinculación del bluetooth

En la figura 3.22 se muestran los timer, con los cuales estará recibiendo el mensaje por el Bluetooth. El clock 1 se ejecuta cada segundo, lo que reciba el Bluetooth será mostrado en la salida de texto llamada mensaje. El clock 2 sirve para limpiar la salida de texto 0.5 segundos después y poder estar listos para la siguiente medición.

```

when Clock1 .Timer
do if BluetoothClient1 . IsConnected
then set Mensaje . Text to join Mensaje . Text
call BluetoothClient1 .ReceiveText
numberOfBytes call BluetoothClient1 .BytesAvailableToReceive

when Clock2 .Timer
do set Mensaje . Text to ""
set Clock2 . TimerInterval to Clock2 . TimerInterval

```

Fig. 3.22 Bloques correspondientes a la recepción de datos por bluetooth

Por último tenemos el bloque donde se dará la instrucción de salida de la aplicación, cuando el usuario ya no siga con el monitoreo, ver figura 3.23, al presionarlo, automáticamente se desvinculara del bluetooth del circuito.



Fig. 3.23 Bloques correspondiente para salir de la aplicación

Con esta aplicación es con la que se desarrolló el prototipo del medidor de ritmo cardiaco, el cual recibirá el valor, dicho valor proviene de la lectura del sensor que tendrá el usuario.

3.8 Desarrollo del programa para el microcontrolador

Para diseñar el programa que necesitará el microcontrolador se realizaron pruebas con las diferentes herramientas con las que cuenta, las cuales fueron mencionadas en el capítulo anterior.

El software requiere de un Timer el cual estará configurado a un cierto tiempo, al agotarse ese tiempo volverá a iniciar el conteo, durante ese lapso, los cambios de estado que reciba en un puerto activaran las interrupciones haciendo un conteo de los pulsos, los cuales serán el equivalente al ritmo cardiaco, es por ello que se han escrito algunos programas con manejo de Timer e interrupciones antes del desarrollo del programa principal.

3.8.1 Manejo de Timer 0 con el microcontrolador

Con el siguiente programa se hace uso de las interrupciones por medio del Timer 0, cada que se interrumpe el timer debido a un desbordamiento se incrementa una variable llamada contador, dicho valor es mostrado en el puerto D y cada que se genera ese cambio el puerto RC0 cambia de estado para avisar que se actualizó. En la figura 3.24 y 3.25 se muestran los diagramas de flujo de este programa.

```

/*****
/*  Uso de Timer 0
/*****
#include <pic18f4550.h>
#include <xc.h> //Librería principal XC8
#include <plib/delays.h> //Librería para retardos
#include <plib/timers.h>

```

```

#include <stdio.h>    //Librería para entrada y salida estándar
#include <stdlib.h>   //Librería estándar

// CONFIG1L
#pragma config PLLDIV = 1    // PLL Prescaler Selection bits (No prescale (4 MHz oscillator input drives PLL directly))
#pragma config CPUDIV = OSC1_PLL2// System Clock Postscaler Selection bits ([Primary Oscillator Src: /1][96 MHz PLL Src: /2])
#pragma config USBDIV = 1    // USB Clock Selection bit (used in Full-Speed USB mode only; UCFG:FSEN = 1) (USB clock source
comes directly from the primary oscillator block with no postscale)
// CONFIG1H
#pragma config FOSC = INTOSCIO_EC// Oscillator Selection bits (Internal oscillator, port function on RA6, EC used by USB (INTIO))
#pragma config FCMEN = OFF    // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor disabled)
#pragma config IESO = OFF    // Internal/External Oscillator Switchover bit (Oscillator Switchover mode disabled)

// CONFIG2L
#pragma config PWRT = OFF    // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOR = ON      // Brown-out Reset Enable bits (Brown-out Reset enabled in hardware only (SBOREN is disabled))
#pragma config BORV = 3     // Brown-out Reset Voltage bits (Minimum setting)
#pragma config VREGEN = OFF  // USB Voltage Regulator Enable bit (USB voltage regulator disabled)

// CONFIG2H
#pragma config WDT = OFF    // Watchdog Timer Enable bit (WDT enabled)
#pragma config WDTPS = 32768 // Watchdog Timer Postscale Select bits (1:32768)

// CONFIG3H
#pragma config CCP2MX = ON  // CCP2 MUX bit (CCP2 input/output is multiplexed with RC1)
#pragma config PBADEN = ON  // PORTB A/D Enable bit (PORTB<4:0> pins are configured as analog input channels on Reset)
#pragma config LPT1OSC = OFF // Low-Power Timer 1 Oscillator Enable bit (Timer1 configured for higher power operation)
#pragma config MCLRE = OFF  // MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disabled)

// CONFIG4L
#pragma config STVREN = ON  // Stack Full/Underflow Reset Enable bit (Stack full/underflow will cause Reset)
#pragma config LVP = ON    // Single-Supply ICSP Enable bit (Single-Supply ICSP enabled)
#pragma config ICPRT = OFF  // Dedicated In-Circuit Debug/Programming Port (ICPORT) Enable bit (ICPORT disabled)
#pragma config XINST = OFF  // Extended Instruction Set Enable bit (Instruction set extension and Indexed Addressing mode
disabled (Legacy mode))

// CONFIG5L
#pragma config CP0 = OFF    // Code Protection bit (Block 0 (000800-001FFFh) is not code-protected)
#pragma config CP1 = OFF    // Code Protection bit (Block 1 (002000-003FFFh) is not code-protected)
#pragma config CP2 = OFF    // Code Protection bit (Block 2 (004000-005FFFh) is not code-protected)
#pragma config CP3 = OFF    // Code Protection bit (Block 3 (006000-007FFFh) is not code-protected)

// CONFIG5H
#pragma config CPB = OFF    // Boot Block Code Protection bit (Boot block (000000-0007FFFh) is not code-protected)
#pragma config CPD = OFF    // Data EEPROM Code Protection bit (Data EEPROM is not code-protected)

// CONFIG6L
#pragma config WRT0 = OFF    // Write Protection bit (Block 0 (000800-001FFFh) is not write-protected)
#pragma config WRT1 = OFF    // Write Protection bit (Block 1 (002000-003FFFh) is not write-protected)
#pragma config WRT2 = OFF    // Write Protection bit (Block 2 (004000-005FFFh) is not write-protected)
#pragma config WRT3 = OFF    // Write Protection bit (Block 3 (006000-007FFFh) is not write-protected)

// CONFIG6H
#pragma config WRTC = OFF    // Configuration Register Write Protection bit (Configuration registers (300000-3000FFFh) are not
write-protected)
#pragma config WRTB = OFF    // Boot Block Write Protection bit (Boot block (000000-0007FFFh) is not write-protected)
#pragma config WRTD = OFF    // Data EEPROM Write Protection bit (Data EEPROM is not write-protected)

// CONFIG7L
#pragma config EBTR0 = OFF   // Table Read Protection bit (Block 0 (000800-001FFFh) is not protected from table reads executed in
other blocks)

```

```

#pragma config EBTR1 = OFF // Table Read Protection bit (Block 1 (002000-003FFFh) is not protected from table reads executed in
other blocks)
#pragma config EBTR2 = OFF // Table Read Protection bit (Block 2 (004000-005FFFh) is not protected from table reads executed in
other blocks)
#pragma config EBTR3 = OFF // Table Read Protection bit (Block 3 (006000-007FFFh) is not protected from table reads executed in
other blocks)

// CONFIG7H
#pragma config EBTRB = OFF // Boot Block Table Read Protection bit (Boot block (000000-0007FFFh) is not protected from table
reads executed in other blocks)
int contador; //Variables globales a utilizar en el programa
void main(void) // Función principal
{

    ADCON1=0x0F; //Declaración de los puertos digitales
    CMCON = 0x07; //Se apagan los comparadores

    TRISD = 0x00; //Puerto D como salida
    TRISCbits.TRISC0 = 0; //Puerto RC0 como salida

    OSCCON=0b01101100; //Frecuencia de oscilador interno a 4MHz

    TOCON = 0b10000110; //Declaracion del Timer 0

    INTCONbits.GIE = 1; //INTERRUPCIONES GLOBALES HABILITADAS
    INTCONbits.TMR0IE = 1; //INTERRUPCIONES DEL TMR1 HABILITADAS

    while(1) //Bucle infinito
    {}
}

void interrupt int_TMR0() //Función para la interrupción del Timer 0
{
    if (INTCONbits.TMR0IF == 1) //Si se activa la bandera de interrupción
    {
        PORTCbits.RC0 ^= 1; //Se cambia de estado el puerto RC0
        ++contador; //Se incrementa la variable contador
        PORTD = contador; //Se muestra la variable en el puerto D

        INTCONbits.TMR0IF = 0; //Se limpia la bandera de interrupción
    }
}

```

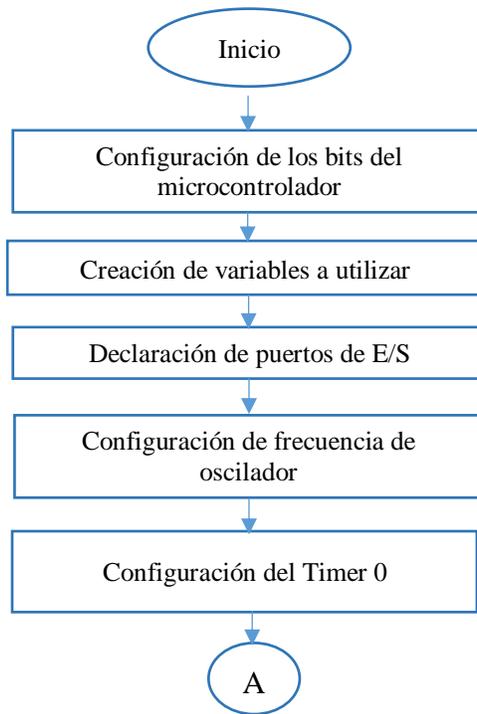


Fig. 3.24 Primera parte del diagrama de flujo del manejo de Timer 0

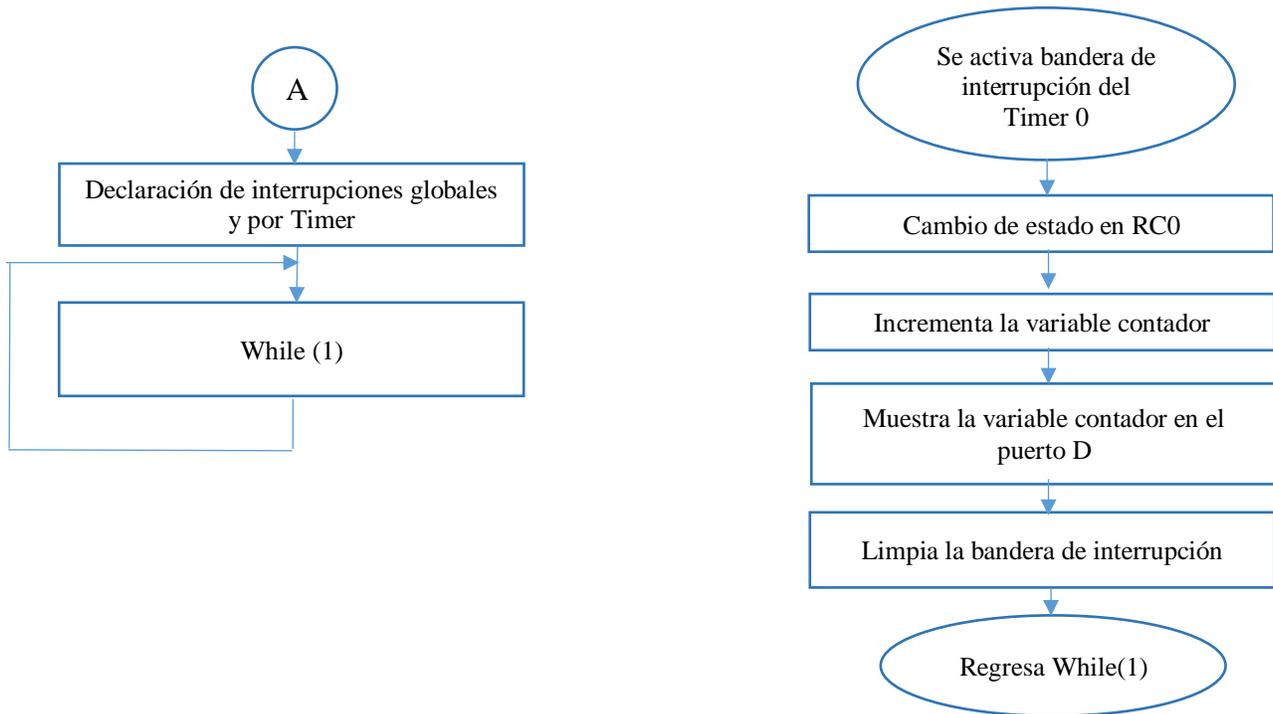


Fig. 3.25 Segunda parte del diagrama de flujo del manejo de Timer 0

3.8.2 Manejo de Interrupciones del puerto B con el microcontrolador

Para el manejo de interrupciones se escribió el siguiente programa con el cual se puede contar las veces en que el puerto B cambia de estado, ya sea 1 a 0 o 0 a 1. En las figuras 3.26 y 3.27 se muestran los diagramas de flujo correspondientes.

```

/*****
/* Manejo de interrupciones del Puerto B */
*****/
#include <pic18f4550.h>
#include <xc.h> //Librería principal XC8
#include <plib/delays.h> //Librería para retardos
#include <plib/timers.h>

#include <stdio.h> //Librería para entrada y salida estandar
#include <stdlib.h> //Librería estandar

// CONFIG1L
#pragma config PLLDIV = 1 // PLL Prescaler Selection bits (No prescale (4 MHz oscillator input drives PLL directly))
#pragma config CPUDIV = OSC1_PLL2 // System Clock Postscaler Selection bits ([Primary Oscillator Src: /1][96 MHz PLL Src: /2])
#pragma config USBDIV = 1 // USB Clock Selection bit (used in Full-Speed USB mode only; UCFG:FSEN = 1) (USB clock source comes directly from the primary oscillator block with no postscale)

// CONFIG1H
#pragma config FOSC = INTOSCIO_EC // Oscillator Selection bits (Internal oscillator, port function on RA6, EC used by USB (INTIO))
#pragma config FCMEN = OFF // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor disabled)
#pragma config IESO = OFF // Internal/External Oscillator Switchover bit (Oscillator Switchover mode disabled)

// CONFIG2L
#pragma config PWRT = OFF // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOR = ON // Brown-out Reset Enable bits (Brown-out Reset enabled in hardware only (SBOREN is disabled))
#pragma config BORV = 3 // Brown-out Reset Voltage bits (Minimum setting)
#pragma config VREGEN = OFF // USB Voltage Regulator Enable bit (USB voltage regulator disabled)

// CONFIG2H
#pragma config WDT = OFF // Watchdog Timer Enable bit (WDT enabled)
#pragma config WDTPS = 32768 // Watchdog Timer Postscale Select bits (1:32768)

// CONFIG3H
#pragma config CCP2MX = ON // CCP2 MUX bit (CCP2 input/output is multiplexed with RC1)
#pragma config PBADEN = ON // PORTB A/D Enable bit (PORTB<4:0> pins are configured as analog input channels on Reset)
#pragma config LPT1OSC = OFF // Low-Power Timer 1 Oscillator Enable bit (Timer1 configured for higher power operation)
#pragma config MCLRE = OFF // MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disabled)

// CONFIG4L
#pragma config STVREN = ON // Stack Full/Underflow Reset Enable bit (Stack full/underflow will cause Reset)
#pragma config LVP = ON // Single-Supply ICSP Enable bit (Single-Supply ICSP enabled)
#pragma config ICPRT = OFF // Dedicated In-Circuit Debug/Programming Port (ICPORT) Enable bit (ICPORT disabled)
#pragma config XINST = OFF // Extended Instruction Set Enable bit (Instruction set extension and Indexed Addressing mode disabled (Legacy mode))

// CONFIG5L
#pragma config CP0 = OFF // Code Protection bit (Block 0 (000800-001FFFh) is not code-protected)
#pragma config CP1 = OFF // Code Protection bit (Block 1 (002000-003FFFh) is not code-protected)
#pragma config CP2 = OFF // Code Protection bit (Block 2 (004000-005FFFh) is not code-protected)
#pragma config CP3 = OFF // Code Protection bit (Block 3 (006000-007FFFh) is not code-protected)

// CONFIG5H
#pragma config CPB = OFF // Boot Block Code Protection bit (Boot block (000000-0007FFFh) is not code-protected)

```

```

#pragma config CPD = OFF // Data EEPROM Code Protection bit (Data EEPROM is not code-protected)

// CONFIG6L
#pragma config WRT0 = OFF // Write Protection bit (Block 0 (000800-001FFFh) is not write-protected)
#pragma config WRT1 = OFF // Write Protection bit (Block 1 (002000-003FFFh) is not write-protected)
#pragma config WRT2 = OFF // Write Protection bit (Block 2 (004000-005FFFh) is not write-protected)
#pragma config WRT3 = OFF // Write Protection bit (Block 3 (006000-007FFFh) is not write-protected)

// CONFIG6H
#pragma config WRTC = OFF // Configuration Register Write Protection bit (Configuration registers (300000-3000FFh) are not
write-protected)
#pragma config WRTB = OFF // Boot Block Write Protection bit (Boot block (000000-0007FFh) is not write-protected)
#pragma config WRTD = OFF // Data EEPROM Write Protection bit (Data EEPROM is not write-protected)

// CONFIG7L
#pragma config EBTR0 = OFF // Table Read Protection bit (Block 0 (000800-001FFFh) is not protected from table reads executed in
other blocks)
#pragma config EBTR1 = OFF // Table Read Protection bit (Block 1 (002000-003FFFh) is not protected from table reads executed in
other blocks)
#pragma config EBTR2 = OFF // Table Read Protection bit (Block 2 (004000-005FFFh) is not protected from table reads executed in
other blocks)
#pragma config EBTR3 = OFF // Table Read Protection bit (Block 3 (006000-007FFFh) is not protected from table reads executed in
other blocks)

// CONFIG7H
#pragma config EBTRB = OFF // Boot Block Table Read Protection bit (Boot block (000000-0007FFh) is not protected from table
reads executed in other blocks)

/***** Variables globales a utilizar en el programa *****/
int contador;
/*****

void main(void) // Función principal
{
    ADCON1=0x0F; //Declaración de los puertos digitales
    CMCON = 0x07;

    TRISD = 0x00;

    TRISBbits.TRISB4 = 1;

    OSCCON=0b01101100; //Frecuencia de oscilador interno a 4MHz

    INTCONbits.GIE = 1; //INTERRUPCIONES GLOBALES HABILITADAS
    INTCONbits.RBIE = 1; //INTERRUPCIONES DEL RB HABILITADAS

    while(1)
    {
    }
}

void interrupt int_RB()
{
    if (INTCONbits.RBIF == 1)
    {
        ++contador;
        PORTD = contador;
        LATB=PORTB;
        INTCONbits.RBIF = 0;
    }
}

```

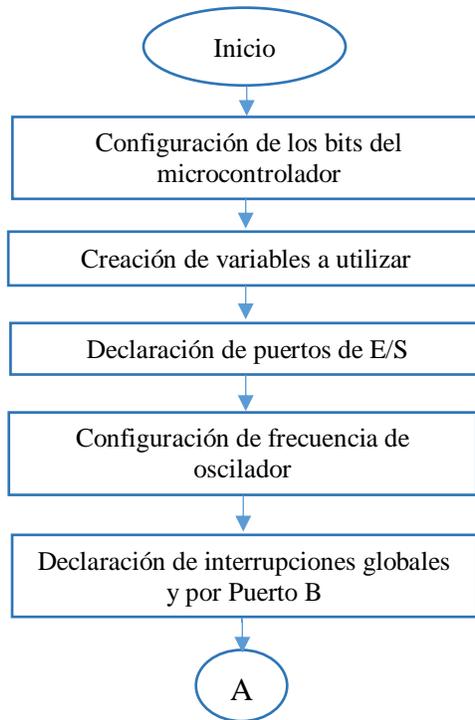


Fig. 3.26 Primera parte del diagrama de flujo del manejo de interrupciones

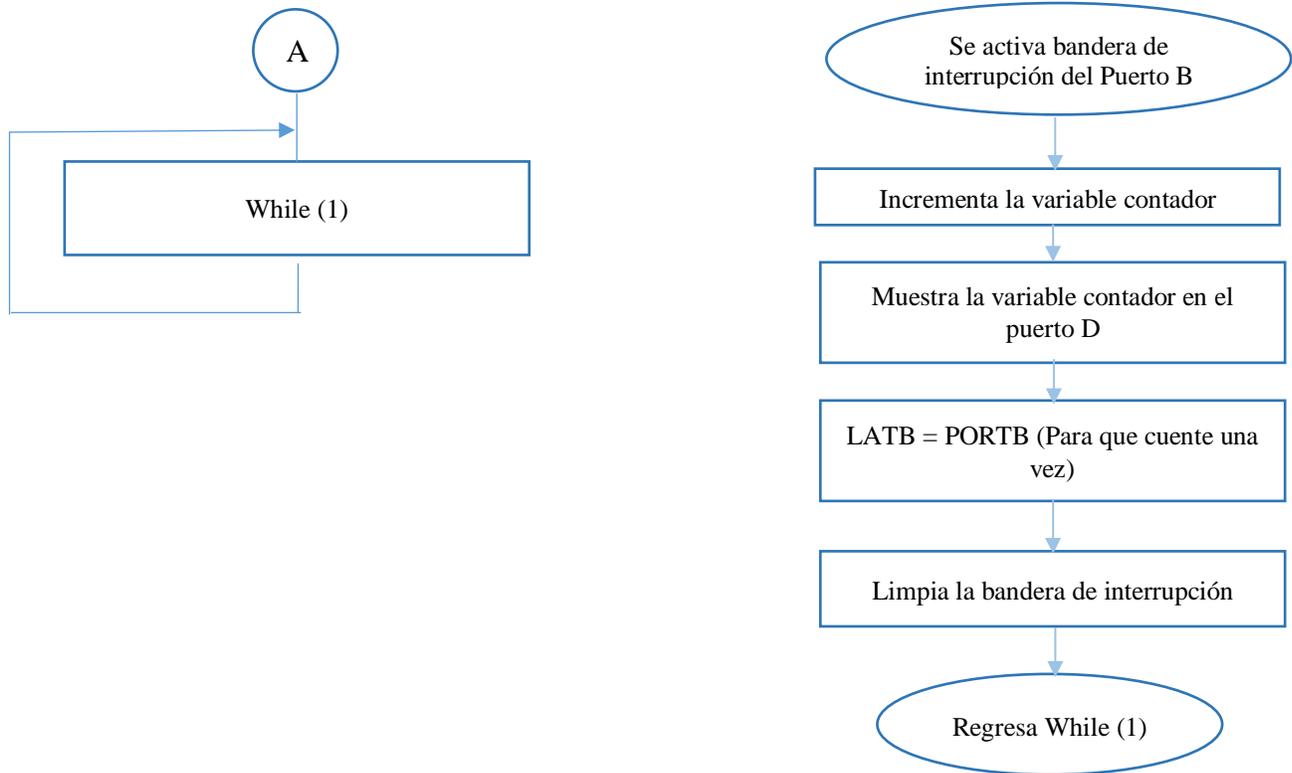


Fig. 3.27 Segunda parte del diagrama de flujo del manejo de interrupciones

3.9 Programa del microcontrolador para el medidor de ritmo cardiaco

Con los códigos anteriores se ha escrito el siguiente programa para el microcontrolador. Este programa contará los pulsos que lleguen al puerto RB7, dichos pulsos son generados por el acondicionador de señal, empezará a contar y cuando el Timer se agote después de 16 segundos hará un cálculo promedio de los pulsos que se tendría en un minuto, dando así, el valor promedio del ritmo cardiaco de la persona, este valor será enviado por el módulo USART al módulo Bluetooth HC-06, el cual se encargará de enviarlo a la aplicación. El programa además tendrá un canal de ADC que se activará cada que el timer 0 se desborde, si la lectura del voltaje es correcta, permitirá el envío del ritmo cardiaco, de lo contrario, enviará un mensaje de “Batería baja”. El diagrama de flujo del programa se muestra en la figura 3.28.

```
/* **** */
/* Programa final para el medidor de ritmo cardiaco */
/* **** */

#include <xc.h> //Librería principal XC8
#include <plib/usart.h> //Librería para USART
#include <plib/adc.h> //Librería del ADC
#include <plib/timers.h> //Librería del Timer
#include <stdio.h> //Librería para entrada y salida estándar
#include <stdlib.h> //Librería estándar

// CONFIG1L
#pragma config PLLDIV = 1 // PLL Prescaler Selection bits (No prescale (4 MHz oscillator input drives PLL directly))
#pragma config CPUDIV = OSC1_PLL2 // System Clock Postscaler Selection bits ([Primary Oscillator Src: /1][96 MHz PLL Src: /2])
#pragma config USBDIV = 1 // USB Clock Selection bit (used in Full-Speed USB mode only; UCFG:FSEN = 1) (USB clock source comes directly from the primary oscillator block with no postscale)

// CONFIG1H
#pragma config FOSC = INTOSCIO_EC // Oscillator Selection bits (Internal oscillator, port function on RA6, EC used by USB (INTIO))
#pragma config FCMEN = OFF // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor disabled)
#pragma config IESO = OFF // Internal/External Oscillator Switchover bit (Oscillator Switchover mode disabled)

// CONFIG2L
#pragma config PWRT = OFF // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOR = ON // Brown-out Reset Enable bits (Brown-out Reset enabled in hardware only (SBOREN is disabled))
#pragma config BORV = 3 // Brown-out Reset Voltage bits (Minimum setting)
#pragma config VREGEN = OFF // USB Voltage Regulator Enable bit (USB voltage regulator disabled)

// CONFIG2H
#pragma config WDT = OFF // Watchdog Timer Enable bit (WDT enabled)
#pragma config WDTPS = 32768 // Watchdog Timer Postscale Select bits (1:32768)

// CONFIG3H
#pragma config CCP2MX = OFF // CCP2 MUX bit (CCP2 input/output is multiplexed with RC1) on
#pragma config PBADEN = ON // PORTB A/D Enable bit (PORTB<4:0> pins are configured as analog input channels on Reset)
#pragma config LPT1OSC = OFF // Low-Power Timer 1 Oscillator Enable bit (Timer1 configured for higher power operation)
#pragma config MCLRE = OFF // MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disabled)

// CONFIG4L
#pragma config STVREN = OFF // Stack Full/Underflow Reset Enable bit (Stack full/underflow will cause Reset)
```

```

#pragma config LVP = OFF // Single-Supply ICSP Enable bit (Single-Supply ICSP enabled)
#pragma config ICPRT = OFF // Dedicated In-Circuit Debug/Programming Port (ICPORT) Enable bit (ICPORT disabled)
#pragma config XINST = OFF // Extended Instruction Set Enable bit (Instruction set extension and Indexed Addressing mode
disabled (Legacy mode))

// CONFIG5L
#pragma config CP0 = OFF // Code Protection bit (Block 0 (000800-001FFFh) is not code-protected)
#pragma config CP1 = OFF // Code Protection bit (Block 1 (002000-003FFFh) is not code-protected)
#pragma config CP2 = OFF // Code Protection bit (Block 2 (004000-005FFFh) is not code-protected)
#pragma config CP3 = OFF // Code Protection bit (Block 3 (006000-007FFFh) is not code-protected)

// CONFIG5H
#pragma config CPB = OFF // Boot Block Code Protection bit (Boot block (000000-0007FFh) is not code-protected)
#pragma config CPD = OFF // Data EEPROM Code Protection bit (Data EEPROM is not code-protected)

// CONFIG6L
#pragma config WRT0 = OFF // Write Protection bit (Block 0 (000800-001FFFh) is not write-protected)
#pragma config WRT1 = OFF // Write Protection bit (Block 1 (002000-003FFFh) is not write-protected)
#pragma config WRT2 = OFF // Write Protection bit (Block 2 (004000-005FFFh) is not write-protected)
#pragma config WRT3 = OFF // Write Protection bit (Block 3 (006000-007FFFh) is not write-protected)

// CONFIG6H
#pragma config WRTC = OFF // Configuration Register Write Protection bit (Configuration registers (300000-3000FFh) are not
write-protected)
#pragma config WRTB = OFF // Boot Block Write Protection bit (Boot block (000000-0007FFh) is not write-protected)
#pragma config WRTD = OFF // Data EEPROM Write Protection bit (Data EEPROM is not write-protected)

// CONFIG7L
#pragma config EBTR0 = OFF // Table Read Protection bit (Block 0 (000800-001FFFh) is not protected from table reads executed in
other blocks)
#pragma config EBTR1 = OFF // Table Read Protection bit (Block 1 (002000-003FFFh) is not protected from table reads executed in
other blocks)
#pragma config EBTR2 = OFF // Table Read Protection bit (Block 2 (004000-005FFFh) is not protected from table reads executed in
other blocks)
#pragma config EBTR3 = OFF // Table Read Protection bit (Block 3 (006000-007FFFh) is not protected from table reads executed in
other blocks)

// CONFIG7H
#pragma config EBTRB = OFF // Boot Block Table Read Protection bit (Boot block (000000-0007FFh) is not protected from table
reads executed in other blocks)

//***** Variables globales a utilizar en el programa *****

unsigned char Mensaje[] = "Bateria baja";
float pulso, ritmo;
unsigned int contador = 0;
unsigned char ResultString[10];
int ADCresult;
float voltaje;
//*****

void main(void) // Función principal
{
    ADCON1=0x0F; //Declaración de los puertos digitales
    CMCON = 0x07;

    TRISD = 0x00;

    TRISBbits.TRISB7 = 1;

```

```

TRISCBits.TRISCO = 0;

OSCCON=0b01101100; //Frecuencia de oscilador interno a 4MHz

//***** Configuración del módulo USART para el Bluetooth *****

OpenUSART( USART_TX_INT_OFF & //Interrupción por Transmisión apagada
           USART_RX_INT_OFF & //Interrupción por Recepción apagada
           USART_ASYNC_MODE & //Modo Asincrono
           USART_EIGHT_BIT & //Datos de 8 bits
           USART_CONT_RX &
           USART_BRGH_HIGH, 25 //Velocidad ajustada para 9600 bauds
);

//***** Configuración del módulo ADC *****

OpenADC( ADC_FOSC_2 &
         ADC_RIGHT_JUST &
         ADC_2_TAD,
         ADC_CH0 &
         ADC_INT_ON &
         ADC_REF_VDD_VSS,
         ADC_IANA
);

TOCON = 0b10000111;

INTCONbits.GIE = 1; //INTERRUPCIONES GLOBALES HABILITADAS
INTCONbits.TMR0IE = 1; //INTERRUPCIONES DEL TMR0 HABILITADAS
INTCONbits.RBIE = 1;

while(1) //Bucle infinito
{
}

void interrupt int_PORTB()
{

if(INTCONbits.RBIF == 1) // Si la bandera de interrupción del puerto B se activa
{
++contador; //Se incrementa contador
LATB = PORTB; //Se acumula la Latch
INTCONbits.RBIF = 0; //Apagamos bandera del puerto b
}

if(INTCONbits.TMR0IF == 1) //Si se activa la bandera por desbordamiento de Timer
{

ConvertADC(); //Se inicia la conversión de ADC
while(BusyADC());
ADCresult = ReadADC(); //Se almacena el valor de la lectura en ADCresult
voltaje = (ADCresult*0.879) / 100; //Se hace ajuste para medir el valor correcto

if(voltaje >= 6.5) //Si el voltaje es correcto realiza la medición de ritmo cardiaco
{
ritmo = (contador/2)*3.5; //Cuenta el numero de pulsos promedio en un minuto
sprintf(ResultString, "%0.3g", ritmo); //Cambie la variable a String
}
}
}

```

```

putsUSART((char * )ResultString); //La envía por el módulo USART

PORTDbits.RD0 ^= 1; //Cambia el estado de un led indicador
Contador = 0; //Se limpia la variable para una nueva muestra

INTCONbits.TMR0IF = 0; //Se limpia la bandera de interrupción del Timer
}

if(voltaje < 6.5) //SI el volaje es muy bajo
{ putsUSART((char * )Mensaje);} //Envia mensaje de batería baja
}
}

```

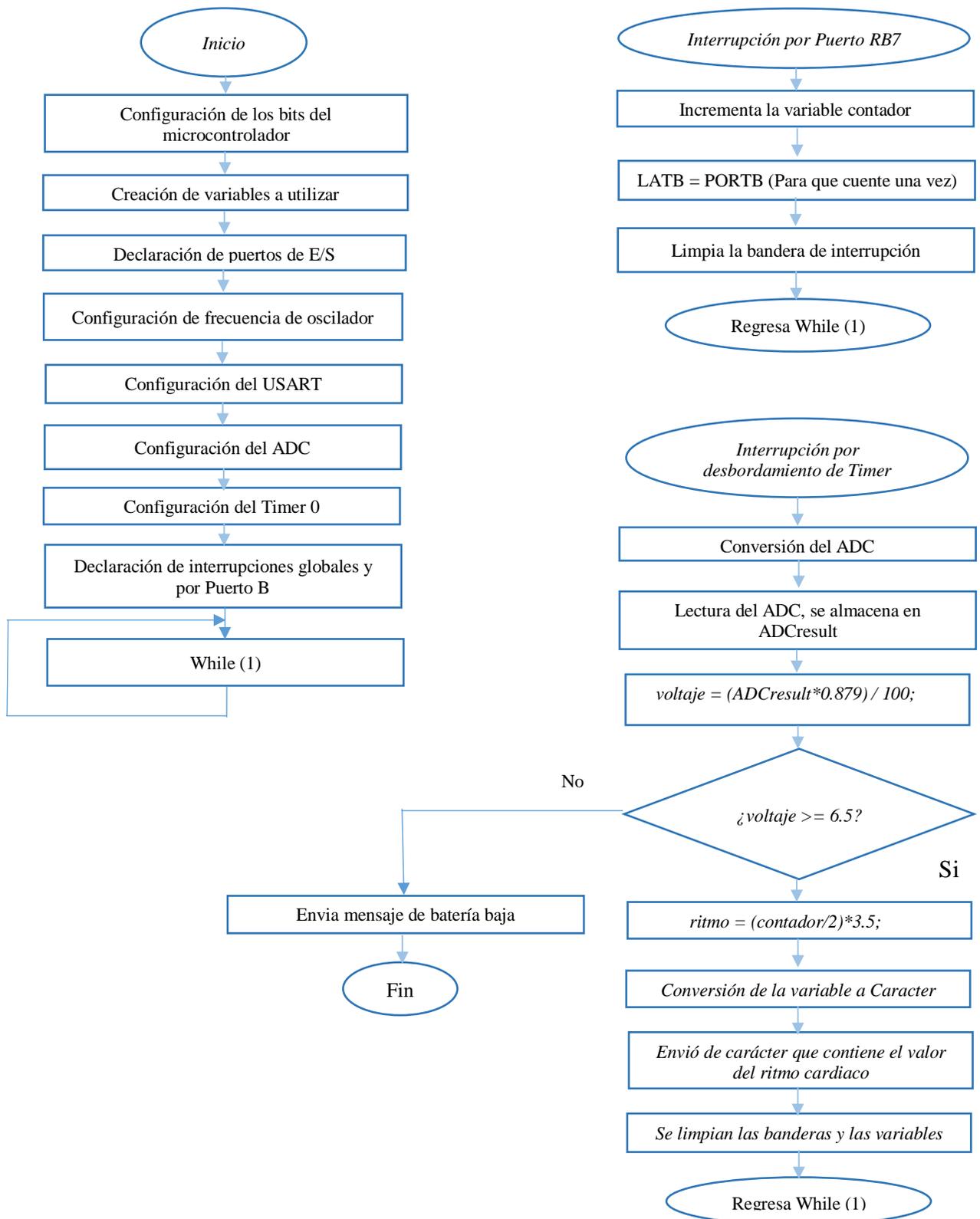


Fig. 3.28 Diagrama de flujo del microcontrolador para el medidor de ritmo cardiaco

Capítulo 4
Pruebas y Resultados

En este capítulo se hablara de los resultados obtenidos de las pruebas que se han realizado junto con las modificaciones del sistema. Las simulaciones son una herramienta poderosa para conocer de una manera anticipada los posibles resultados, es por ello que primero se generó trabajo de simulación antes de realizar pruebas físicas en personas y así estar seguros que funcionara correctamente.

4.1 Comunicación Bluetooth y microcontrolador

Para lograr asociar el bluetooth con el microcontrolador y establecer una interfaz con la aplicación del teléfono celular se realizaron diversas pruebas, cabe mencionar que dichas pruebas se realizaron con éxito y que por medio de ellas se logró tener un mejor entendimiento de cómo funciona la plataforma de App Inventor.

Cada uno de los programas que a continuación se muestran fueron escritos en MPLAB X IDE v3.00 y el compilador XC8 de Microchip.

4.1.1 Prueba 1 “Encendido de led’s por medio de un teléfono celular”

Esta primera prueba consiste básicamente en el encendido de dos led’s conectados a dos puertos del microcontrolador por medio de una aplicación para Android. El circuito propuesto se muestra en la figura 4.1. En dicho circuito se conecta el Bluetooth junto con dos led’s, todo el circuito debe ser alimentador por 5v.

En las figuras 4.2 y 4.3 se muestran el diagrama de flujo del programa del microcontrolador.

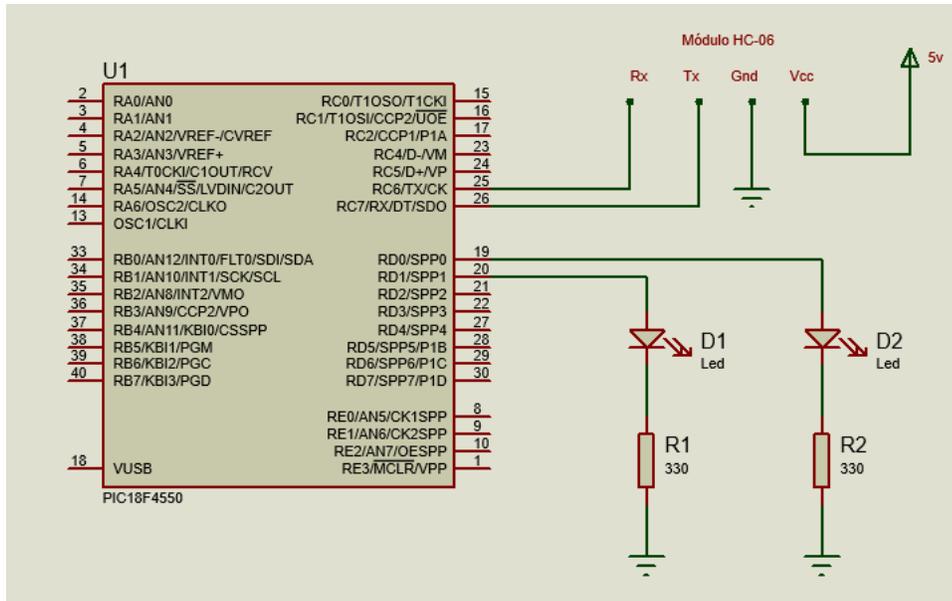


Fig. 4.1 Circuito para la prueba 1

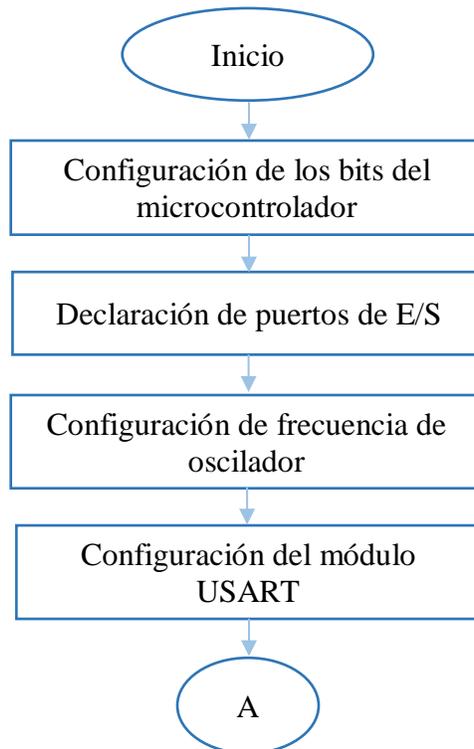


Fig. 4.2 Diagrama de flujo del programa de la prueba 1

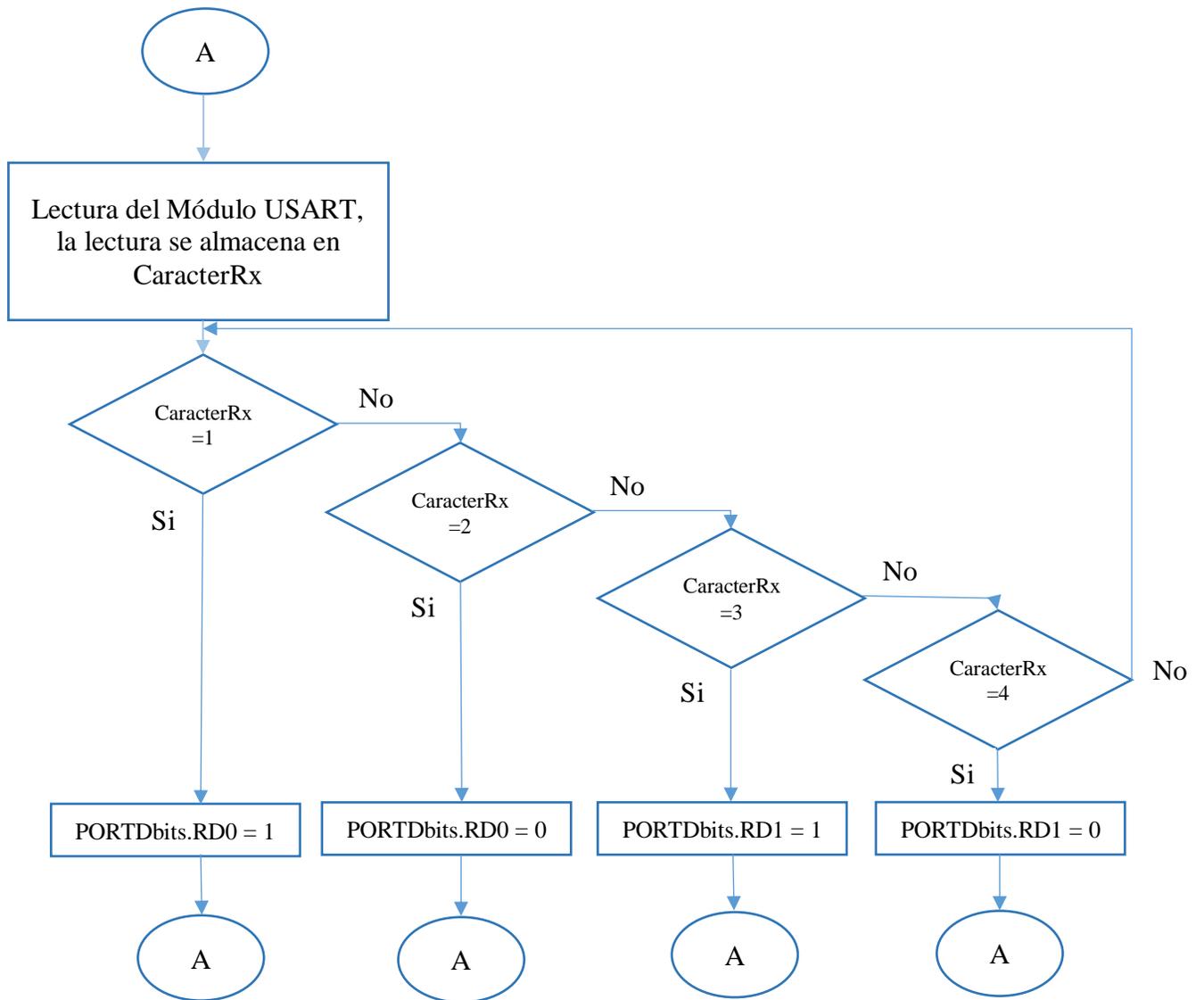


Fig. 4.3 Segunda parte del diagrama de flujo del programa de la prueba 1

A continuación se muestra el código del programa para el microcontrolador, al cual corresponde a la prueba 1.

```

/*****
/* Comunicación Bluetooth y PIC18F4550 */
/* Enciende y apaga dos Led's desde celular */
*****/

#include <xc.h> //Biblioteca de XC8
#include <plib/usart.h> //Biblioteca para el uso de funciones del módulo USART

//Bits de configuración
// CONFIG1L
#pragma config PLLDIV = 1 // PLL Prescaler Selection bits (No prescale (4 MHz oscillator input drives PLL directly))
#pragma config CPUDIV = OSC1_PLL2 // System Clock Postscaler Selection bits ([Primary Oscillator Src: /1][96 MHz PLL Src: /2])
#pragma config USBDIV = 1 // USB Clock Selection bit (used in Full-Speed USB mode only; UCFG:FSEN = 1) (USB clock source comes directly from the primary oscillator block with no postscale)

// CONFIG1H
#pragma config FOSC = INTOSCIO_EC // Oscilador interno
#pragma config FCMEN = OFF // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor disabled)
#pragma config IESO = OFF // Internal/External Oscillator Switchover bit (Oscillator Switchover mode disabled)

// CONFIG2L
#pragma config PWRT = OFF // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOR = ON // Brown-out Reset Enable bits (Brown-out Reset enabled in hardware only (SBOREN is disabled))
#pragma config BORV = 3 // Brown-out Reset Voltage bits (Minimum setting)
#pragma config VREGEN = OFF // USB Voltage Regulator Enable bit (USB voltage regulator disabled)

// CONFIG2H
#pragma config WDT = OFF // Watchdog Timer Enable bit (WDT disabled (control is placed on the SWDTEN bit))
#pragma config WDTPS = 32768 // Watchdog Timer Postscale Select bits (1:32768)

// CONFIG3H
#pragma config CCP2MX = ON // CCP2 MUX bit (CCP2 input/output is multiplexed with RC1)
#pragma config PBADEN = ON // PORTB A/D Enable bit (PORTB<4:0> pins are configured as analog input channels on Reset)
#pragma config LPT1OSC = OFF // Low-Power Timer 1 Oscillator Enable bit (Timer1 configured for higher power operation)
#pragma config MCLRE = OFF // MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disabled)

// CONFIG4L
#pragma config STVREN = ON // Stack Full/Underflow Reset Enable bit (Stack full/underflow will cause Reset)
#pragma config LVP = ON // Single-Supply ICSP Enable bit (Single-Supply ICSP enabled)
#pragma config ICSPRT = OFF // Dedicated In-Circuit Debug/Programming Port (ICPORT) Enable bit (ICPORT disabled)
#pragma config XINST = OFF // Extended Instruction Set Enable bit (Instruction set extension and Indexed Addressing mode disabled (Legacy mode))

// CONFIG5L
#pragma config CP0 = OFF // Code Protection bit (Block 0 (000800-001FFFh) is not code-protected)
#pragma config CP1 = OFF // Code Protection bit (Block 1 (002000-003FFFh) is not code-protected)
#pragma config CP2 = OFF // Code Protection bit (Block 2 (004000-005FFFh) is not code-protected)
#pragma config CP3 = OFF // Code Protection bit (Block 3 (006000-007FFFh) is not code-protected)

// CONFIG5H
#pragma config CPB = OFF // Boot Block Code Protection bit (Boot block (000000-0007FFFh) is not code-protected)
#pragma config CPD = OFF // Data EEPROM Code Protection bit (Data EEPROM is not code-protected)

// CONFIG6L
#pragma config WRT0 = OFF // Write Protection bit (Block 0 (000800-001FFFh) is not write-protected)
#pragma config WRT1 = OFF // Write Protection bit (Block 1 (002000-003FFFh) is not write-protected)
#pragma config WRT2 = OFF // Write Protection bit (Block 2 (004000-005FFFh) is not write-protected)
#pragma config WRT3 = OFF // Write Protection bit (Block 3 (006000-007FFFh) is not write-protected)

```

```

// CONFIG6H
#pragma config WRTC = OFF // Configuration Register Write Protection bit (Configuration registers (300000-3000FFh) are not
write-protected)
#pragma config WRTB = OFF // Boot Block Write Protection bit (Boot block (000000-0007FFh) is not write-protected)
#pragma config WRTD = OFF // Data EEPROM Write Protection bit (Data EEPROM is not write-protected)

// CONFIG7L
#pragma config EBTR0 = OFF // Table Read Protection bit (Block 0 (000800-001FFFh) is not protected from table reads executed in
other blocks)
#pragma config EBTR1 = OFF // Table Read Protection bit (Block 1 (002000-003FFFh) is not protected from table reads executed in
other blocks)
#pragma config EBTR2 = OFF // Table Read Protection bit (Block 2 (004000-005FFFh) is not protected from table reads executed in
other blocks)
#pragma config EBTR3 = OFF // Table Read Protection bit (Block 3 (006000-007FFFh) is not protected from table reads executed in
other blocks)

// CONFIG7H
#pragma config EBTRB = OFF // Boot Block Table Read Protection bit (Boot block (000000-0007FFh) is not protected from table
reads executed in other blocks)

void main(void) //Función principal
{
    char CaracterRx; //Variable para la transmisión

    ADCON1=0x0F; //Declaración de los puertos digitales
    CMCON = 0x07;

    TRISD = 0x00; //Puerto D como salida
    OSCCON=0b01101100; //FRECUENCIA DEL OSCILADOR INTERNO 4MHz

    OpenUSART( USART_TX_INT_OFF & //Interrupción por Tx deshabilitada
               USART_RX_INT_OFF & //Interrupción por Rx deshabilitada
               USART_ASYNC_MODE & //Modo Asincrono
               USART_EIGHT_BIT & //8 bits
               USART_CONT_RX &
               USART_BRGH_HIGH, 25 //9600 Bauds es la velocidad del módulo
    ); //HC-06, segun tabla de hoja de datos
    // "25" es la velocidad de transmisión del
    // PIC a 4MHz para transmitir a 9600 bauds

    while(1)
    {
        CaracterRx = ReadUSART(); //Leemos el dato recibido del bluetooth

        //Condiciones
        if( CaracterRx == 1 ) //Si se recibe un 1 de la transmisión
        {PORTDbits.RD0 = 1; } //Se enciende el Led del puerto D0

        if( CaracterRx == 2 ) //Si se recibe un 2 de la transmisión
        {PORTDbits.RD0 = 0; } //Se apaga el Led del puerto D0

        if( CaracterRx == 3 ) //Si se recibe un 3 de la transmisión
        {PORTDbits.RD1 = 1; } //Se enciende el Led del puerto D1

        if( CaracterRx == 4 ) //Si se recibe un 4 de la transmisión
        {PORTDbits.RD1 = 0; } //Se apaga el Led del puerto D1

    }
}

```

En la figura 4.4 se muestra la interfaz gráfica que se diseñó para la aplicación en Android que posteriormente fue instalada en un celular para hacer el experimento, en la figura 4.5 se muestran los bloques que se ensamblaron para hacer dicha aplicación funcional.

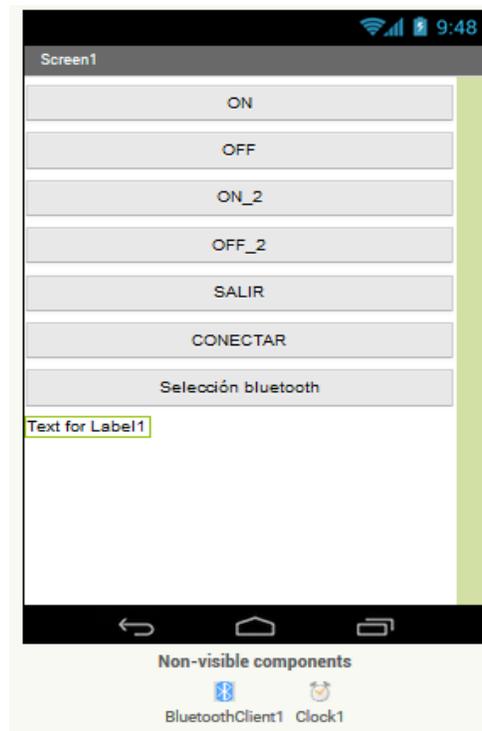


Fig. 4.4 Diseño de la aplicación para Android correspondiente a la prueba 1

Como se puede observar, en la aplicación se muestran cuatro botones para el encendido y apagado independiente de cada uno de los led's conectados al microcontrolador, además de un botón para salir de la aplicación, un botón para conectar el celular con el bluetooth y un último botón para seleccionar dicho bluetooth a conectar.

A dichos botones se les debe dar alguna función por medio de bloques correspondientes unidos entre sí, en la figura 4.5 se muestran los correspondientes a la prueba 1.

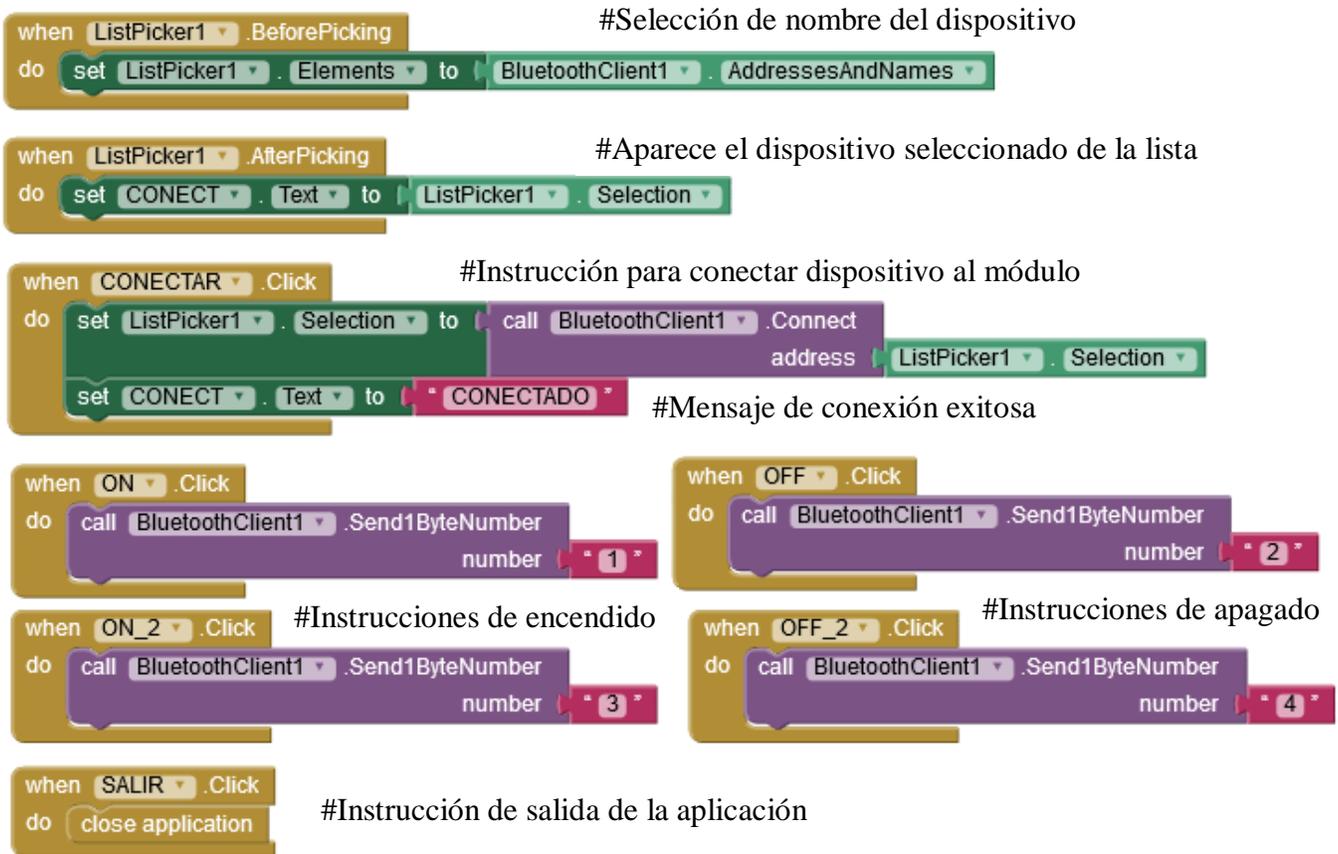


Fig. 4.5 Bloques para el diseño de la aplicación correspondiente a la prueba 1

Como se puede observar, en la figura 4.5 cada uno de los bloques amarillos representa a cada botón y le da una función a cada uno para saber que hacer al ser presionados. En este caso lo más importante es el envío de datos, cada que se presiona un botón, se envía un número, ese número es el que recibe el USART para las condiciones de encendido y apagado de los led's.

4.1.2 Prueba 2 “Envío de datos del microcontrolador al celular por medio de una aplicación”

Con esta prueba se ha diseñado un programa para el microcontrolador, donde se active un canal del convertidor analógico-digital, hacer una lectura de voltaje y enviar ese dato para visualizarlo en el celular. En la figura 4.6 se muestra el circuito para la prueba 2.

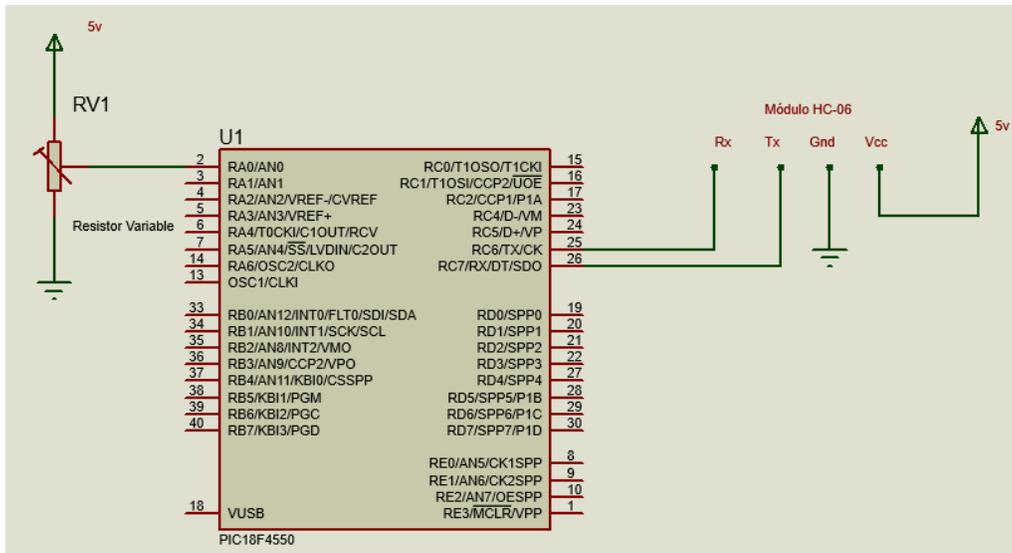


Fig. 4.6 Circuito para la prueba 2

El diagrama de flujo correspondiente al programa para esta segunda prueba es el siguiente;

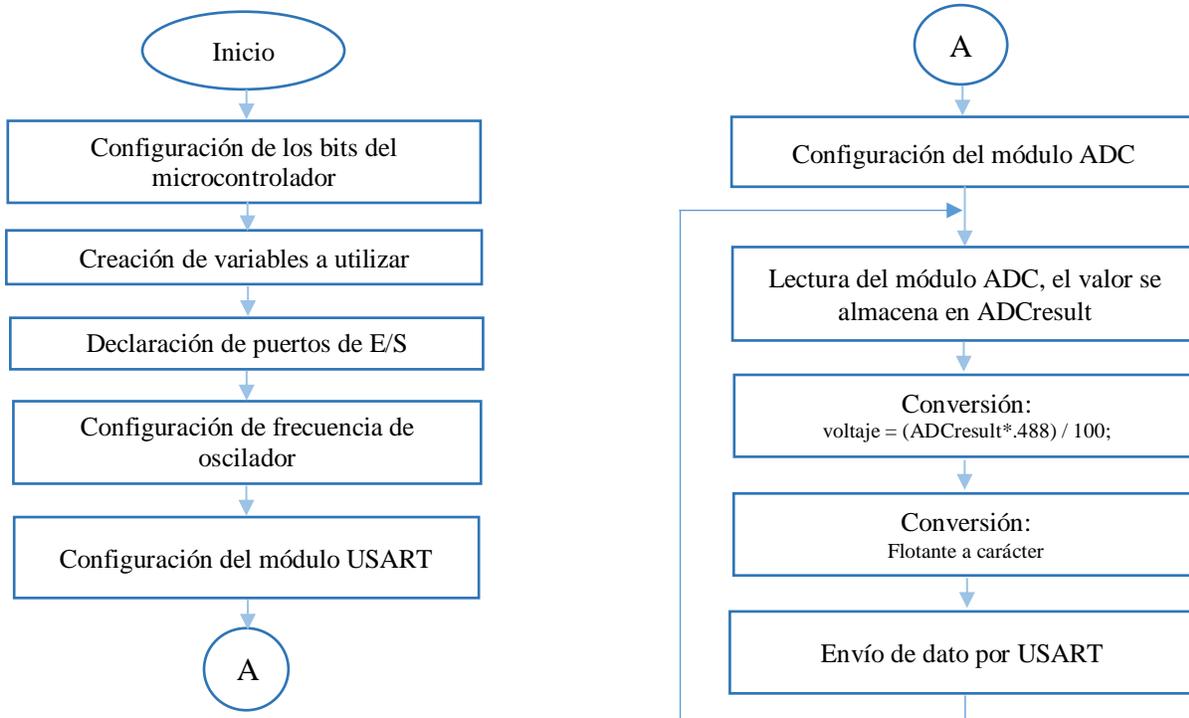


Fig. 4.7 Diagrama de flujo del programa para la prueba 2

El programa para el microcontrolador es el siguiente:

```

/*****
/* Comunicación Bluetooth y PIC18F4550 */
/* Activa un canal de ADC para medir un voltaje y mostrar la */
/* la información en el celular */
*****/

#include <xc.h> //Biblioteca de XC8
#include <plib/usart.h> //Biblioteca para las funciones del módulo USART
#include <plib/adc.h> //Biblioteca para las funciones del módulo ADC
#include <stdio.h> //Biblioteca para entrada y salida estandar
#include <stdlib.h>

// CONFIG1L
#pragma config PLLDIV = 1 // PLL Prescaler Selection bits (No prescale (4 MHz oscillator input drives PLL directly))
#pragma config CPUDIV = OSC1_PLL2 // System Clock Postscaler Selection bits ([Primary Oscillator Src: /1][96 MHz PLL Src: /2])
#pragma config USBDIV = 1 // USB Clock Selection bit (used in Full-Speed USB mode only; UCFG:FSEN = 1) (USB clock source
comes directly from the primary oscillator block with no postscale)

// CONFIG1H
#pragma config FOSC = INTOSCIO_EC // Oscillator Selection bits (Internal oscillator, port function on RA6, EC used by USB (INTIO))
#pragma config FCMEN = OFF // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor disabled)
#pragma config IESO = OFF // Internal/External Oscillator Switchover bit (Oscillator Switchover mode disabled)

// CONFIG2L
#pragma config PWRT = OFF // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOR = ON // Brown-out Reset Enable bits (Brown-out Reset enabled in hardware only (SBOREN is disabled))
#pragma config BORV = 3 // Brown-out Reset Voltage bits (Minimum setting)
#pragma config VREGEN = OFF // USB Voltage Regulator Enable bit (USB voltage regulator disabled)

// CONFIG2H
#pragma config WDT = ON // Watchdog Timer Enable bit (WDT enabled)
#pragma config WDTPS = 32768 // Watchdog Timer Postscale Select bits (1:32768)

// CONFIG3H
#pragma config CCP2MX = ON // CCP2 MUX bit (CCP2 input/output is multiplexed with RC1)
#pragma config PBADEN = ON // PORTB A/D Enable bit (PORTB<4:0> pins are configured as analog input channels on Reset)
#pragma config LPT1OSC = OFF // Low-Power Timer 1 Oscillator Enable bit (Timer1 configured for higher power operation)
#pragma config MCLRE = OFF // MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disabled)

// CONFIG4L
#pragma config STVREN = ON // Stack Full/Underflow Reset Enable bit (Stack full/underflow will cause Reset)
#pragma config LVP = ON // Single-Supply ICSP Enable bit (Single-Supply ICSP enabled)
#pragma config ICPRT = OFF // Dedicated In-Circuit Debug/Programming Port (ICPORT) Enable bit (ICPORT disabled)
#pragma config XINST = OFF // Extended Instruction Set Enable bit (Instruction set extension and Indexed Addressing mode
disabled (Legacy mode))

// CONFIG5L
#pragma config CP0 = OFF // Code Protection bit (Block 0 (000800-001FFFh) is not code-protected)
#pragma config CP1 = OFF // Code Protection bit (Block 1 (002000-003FFFh) is not code-protected)
#pragma config CP2 = OFF // Code Protection bit (Block 2 (004000-005FFFh) is not code-protected)
#pragma config CP3 = OFF // Code Protection bit (Block 3 (006000-007FFFh) is not code-protected)

// CONFIG5H
#pragma config CPB = OFF // Boot Block Code Protection bit (Boot block (000000-0007FFFh) is not code-protected)
#pragma config CPD = OFF // Data EEPROM Code Protection bit (Data EEPROM is not code-protected)

```

```

// CONFIG6L
#pragma config WRT0 = OFF // Write Protection bit (Block 0 (000800-001FFFh) is not write-protected)
#pragma config WRT1 = OFF // Write Protection bit (Block 1 (002000-003FFFh) is not write-protected)
#pragma config WRT2 = OFF // Write Protection bit (Block 2 (004000-005FFFh) is not write-protected)
#pragma config WRT3 = OFF // Write Protection bit (Block 3 (006000-007FFFh) is not write-protected)

// CONFIG6H
#pragma config WRTC = OFF // Configuration Register Write Protection bit (Configuration registers (300000-3000FFh) are not
write-protected)
#pragma config WRTB = OFF // Boot Block Write Protection bit (Boot block (000000-0007FFh) is not write-protected)
#pragma config WRWD = OFF // Data EEPROM Write Protection bit (Data EEPROM is not write-protected)

// CONFIG7L
#pragma config EBTR0 = OFF // Table Read Protection bit (Block 0 (000800-001FFFh) is not protected from table reads executed in
other blocks)
#pragma config EBTR1 = OFF // Table Read Protection bit (Block 1 (002000-003FFFh) is not protected from table reads executed in
other blocks)
#pragma config EBTR2 = OFF // Table Read Protection bit (Block 2 (004000-005FFFh) is not protected from table reads executed in
other blocks)
#pragma config EBTR3 = OFF // Table Read Protection bit (Block 3 (006000-007FFFh) is not protected from table reads executed in
other blocks)

// CONFIG7H
#pragma config EBTRB = OFF // Boot Block Table Read Protection bit (Boot block (000000-0007FFh) is not protected from table
reads executed in other blocks)

unsigned int ADCresult=0; //Variable para almacenar el valor del ADC
float voltaje; //Variable para el ajuste en volts
unsigned char ResultString[10]; //Variable para la conversión de float a string

void main(void)
{
    char CaracterRx; //Variable para la transmisión

    OSCCON=0b01101100; //Oscilador interno a 4Mhz

    OpenUSART( USART_TX_INT_OFF & //Interrupción por Tx deshabilitada
               USART_RX_INT_OFF & //Interrupción por Rx deshabilitada
               USART_ASYNC_MODE & //Modo Asincrono
               USART_EIGHT_BIT & //8 bits
               USART_CONT_RX &
               USART_BRGH_HIGH, 25 //9600 Bauds es la velocidad del módulo
               ); //HC-06, segun tabla de hoja de datos
               // "25" es la velocidad de transmisión del
               //PIC a 4MHz para transmitir a 9600 bauds

    OpenADC( ADC_FOSC_2 & //Frecuencia de ADC
             ADC_RIGHT_JUST & //Justificación a la derecha del dato
             ADC_2_TAD, //Dos tiempos de adquisición
             ADC_CH0 & //Canal 0
             ADC_INT_ON & //Interrupción activada
             ADC_REF_VDD_VSS, //Voltajes de suministro como referencia
             ADC_IANA //Solo el canal 0 activado
             );

    while(1)
    {

        ConvertADC(); //Se inicia la conversión del ADC
        while(BusyADC()); //Espera a que termine la conversión
        ADCresult = ReadADC(); //Almacena el valor de la lectura en la variable ADCresult
    }
}

```

```

voltaje = (ADCresult*.488) / 100; //Se convierte para ajustarlo al valor del voltaje
sprintf(ResultString, "%.3g", voltaje); //Se convierte de float a string

putsUSART((char *)ResultString); //Se envía por USART

}
CloseUSART();
CloseADC(); }

```

En la figura 4.8 se muestra el diseño de la aplicación para esta prueba. Esta aplicación cuenta con menos botones ya que en esencia solo se diseñó para recibir los datos que adquiere el microcontrolador. En la figura 4.9 se muestra la configuración de bloques para hacer funcionar la aplicación.



Fig. 4.8 Aplicación para Android correspondiente a la prueba 2

#Selección de nombre del dispositivo

```

when Selección_de_bluetooth .BeforePicking
do set Selección_de_bluetooth .Elements to BluetoothClient1 .AddressesAndNames

```

#Aparece el dispositivo seleccionado de la lista

```

when Selección_de_bluetooth .AfterPicking
do set CONECT .Text to Selección_de_bluetooth .Selection

```

#Instrucción para conectar dispositivo al módulo

```

when CONECTAR .Click
do set Selección_de_bluetooth .Selection to call BluetoothClient1 .Connect
address Selección_de_bluetooth .Selection
set CONECT .Text to "CONECTADO"

```

```

when Clock2 .Timer
do set Clock2 .TimerInterval to Clock2 .TimerInterval
set Mensaje .Text to " "

```

```

when SALIR .Click
do close application

```

#Instrucción para mostrar la información recibida

```

when Clock1 .Timer
do if BluetoothClient1 .IsConnected
then set Mensaje .Text to join Mensaje .Text
call BluetoothClient1 .ReceiveText
numberOfBytes call BluetoothClient1 .BytesAvailableToReceive

```

Fig. 4.9 Bloques de la aplicación para Android para la prueba 2

4.1.3 Prueba 3 “Envío de datos al microcontrolador desde el celular por medio de una aplicación”

En esta última prueba se logró enviar una dato de 0 a 255 en decimal desde el celular al microcontrolador, esto se hace por medio de una barra deslizable, también conocido como slider. En la figura 4.10 se muestra el circuito usado para la prueba, básicamente solo consiste en un conjunto de leds en el puerto D para visualizar el dato en forma binaria.

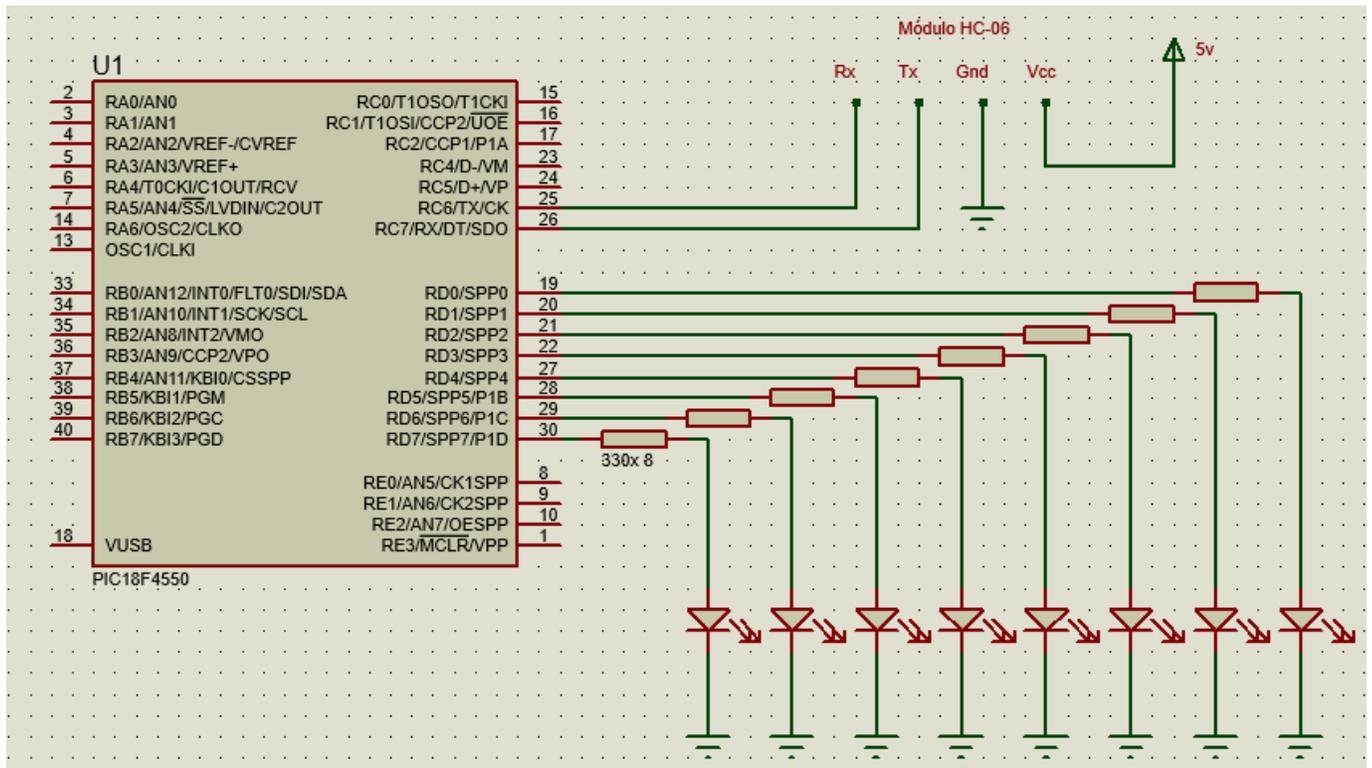


Fig. 4.10 Circuito usado para la prueba 3

En la figura 4.11 se ilustra el diagrama de flujo del programa correspondiente a la prueba 3, posteriormente se tiene el código correspondiente el cual se grabó en el microcontrolador.

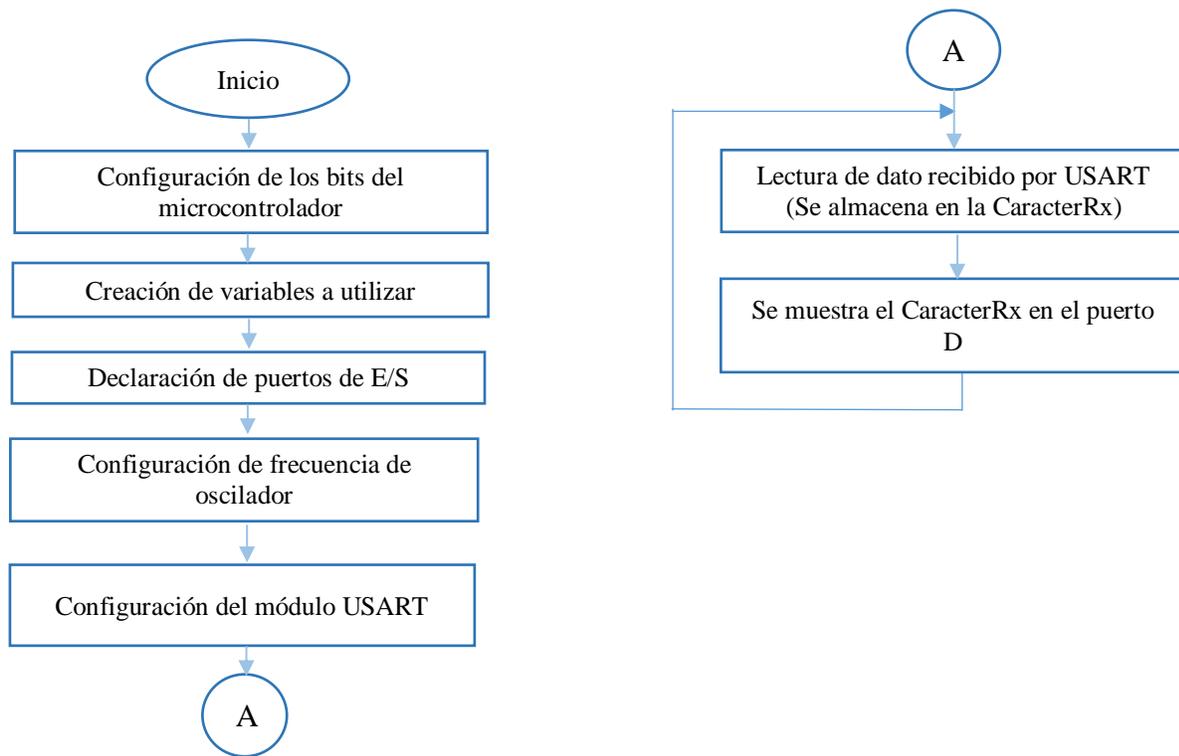


Fig. 4.11 Diagrama de flujo de la prueba 3

```

/*****
/* Comunicación Bluetooth y PIC18F4550
/* Prueba 3
/* Barra deslizante
*****/

#include <xc.h>
#include <plib/usart.h>

// CONFIG1L
#pragma config PLLDIV = 1 // PLL Prescaler Selection bits (No prescale (4 MHz oscillator input drives PLL directly))
#pragma config CPUDIV = OSC1_PLL2 // System Clock Postscaler Selection bits ([Primary Oscillator Src: /1][96 MHz PLL Src: /2])
#pragma config USBDIV = 1 // USB Clock Selection bit (used in Full-Speed USB mode only; UCFG:FSEN = 1) (USB clock source comes directly from the primary oscillator block with no postscale)

// CONFIG1H
#pragma config FOSC = INTOSCIO_EC // Oscilador interno
#pragma config FCMEN = OFF // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock Monitor disabled)
#pragma config IESO = OFF // Internal/External Oscillator Switchover bit (Oscillator Switchover mode disabled)

// CONFIG2L
#pragma config PWRT = OFF // Power-up Timer Enable bit (PWRT disabled)
#pragma config BOR = ON // Brown-out Reset Enable bits (Brown-out Reset enabled in hardware only (SBOREN is disabled))

```

```

#pragma config BORV = 3 // Brown-out Reset Voltage bits (Minimum setting)
#pragma config VREGEN = OFF // USB Voltage Regulator Enable bit (USB voltage regulator disabled)

// CONFIG2H
#pragma config WDT = OFF // Watchdog Timer Enable bit (WDT disabled (control is placed on the SWDTEN bit))
#pragma config WDTPS = 32768 // Watchdog Timer Postscale Select bits (1:32768)

// CONFIG3H
#pragma config CCP2MX = ON // CCP2 MUX bit (CCP2 input/output is multiplexed with RC1)
#pragma config PBADEN = ON // PORTB A/D Enable bit (PORTB<4:0> pins are configured as analog input channels on Reset)
#pragma config LPT1OSC = OFF // Low-Power Timer 1 Oscillator Enable bit (Timer1 configured for higher power operation)
#pragma config MCLRE = ON // MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin disabled)

// CONFIG4L
#pragma config STVREN = ON // Stack Full/Underflow Reset Enable bit (Stack full/underflow will cause Reset)
#pragma config LVP = ON // Single-Supply ICSP Enable bit (Single-Supply ICSP enabled)
#pragma config ICPRT = OFF // Dedicated In-Circuit Debug/Programming Port (ICPORT) Enable bit (ICPORT disabled)
#pragma config XINST = OFF // Extended Instruction Set Enable bit (Instruction set extension and Indexed Addressing mode disabled (Legacy mode))

// CONFIG5L
#pragma config CP0 = OFF // Code Protection bit (Block 0 (000800-001FFFh) is not code-protected)
#pragma config CP1 = OFF // Code Protection bit (Block 1 (002000-003FFFh) is not code-protected)
#pragma config CP2 = OFF // Code Protection bit (Block 2 (004000-005FFFh) is not code-protected)
#pragma config CP3 = OFF // Code Protection bit (Block 3 (006000-007FFFh) is not code-protected)

// CONFIG5H
#pragma config CPB = OFF // Boot Block Code Protection bit (Boot block (000000-0007FFh) is not code-protected)
#pragma config CPD = OFF // Data EEPROM Code Protection bit (Data EEPROM is not code-protected)

// CONFIG6L
#pragma config WRT0 = OFF // Write Protection bit (Block 0 (000800-001FFFh) is not write-protected)
#pragma config WRT1 = OFF // Write Protection bit (Block 1 (002000-003FFFh) is not write-protected)
#pragma config WRT2 = OFF // Write Protection bit (Block 2 (004000-005FFFh) is not write-protected)
#pragma config WRT3 = OFF // Write Protection bit (Block 3 (006000-007FFFh) is not write-protected)

// CONFIG6H
#pragma config WRTC = OFF // Configuration Register Write Protection bit (Configuration registers (300000-3000FFh) are not write-protected)
#pragma config WRTB = OFF // Boot Block Write Protection bit (Boot block (000000-0007FFh) is not write-protected)
#pragma config WRTD = OFF // Data EEPROM Write Protection bit (Data EEPROM is not write-protected)

// CONFIG7L
#pragma config EBTR0 = OFF // Table Read Protection bit (Block 0 (000800-001FFFh) is not protected from table reads executed in other blocks)
#pragma config EBTR1 = OFF // Table Read Protection bit (Block 1 (002000-003FFFh) is not protected from table reads executed in other blocks)
#pragma config EBTR2 = OFF // Table Read Protection bit (Block 2 (004000-005FFFh) is not protected from table reads executed in other blocks)
#pragma config EBTR3 = OFF // Table Read Protection bit (Block 3 (006000-007FFFh) is not protected from table reads executed in other blocks)

// CONFIG7H
#pragma config EBTRB = OFF // Boot Block Table Read Protection bit (Boot block (000000-0007FFh) is not protected from table reads executed in other blocks)

void main(void) //Función principal
{
    char CaracterRx; //Variable para la transmisión

    ADCON1=0x0F; //Declaración de los puertos digitales
    CMCON = 0x07;

```

```

TRISD = 0x00;           //Puerto D como salida
OSCCON=0b01101100;    //FRECUENCIA DEL OSCILADOR INTERNO 4MHz

OpenUSART( USART_TX_INT_OFF & //Interrupción por Tx deshabilitada
           USART_RX_INT_OFF & //Interrupción por Rx deshabilitada
           USART_ASYNC_MODE & //Modo Asincrono
           USART_EIGHT_BIT & //8 bits
           USART_CONT_RX &
           USART_BRGH_HIGH, 25 //9600 bauds es la velocidad del modulo
           );                 //HC-06, segun tabla de hoja de datos
                               //"25" es la velocidad de transmisión del
                               //PIC a 4MHz

while(1)
{
  CaracterRx = ReadUSART(); //Leemos el dato recibido del bluetooth

  PORTD = CaracterRx;

}
//CloseUSART();
}

```

Esta última prueba consiste en enviar datos al microcontrolador por medio de una barra deslizante virtual en el celular, en la figura 4.12 se muestra la interfaz de usuario de la aplicación en Android.



Fig. 4.12 Aplicación con barra deslizante virtual

En la figura 4.13 se muestra la configuración de bloques en App Inventor para poder enviar el dato actual de la barra deslizable de la aplicación.

```

when ListPicker1 .BeforePicking      #Selección de nombre del dispositivo
do set ListPicker1 .Elements to BluetoothClient1 .AddressesAndNames

when ListPicker1 .AfterPicking      #Aparece el dispositivo seleccionado de la lista
do set Label4 .Text to ListPicker1 .Selection

when CONECTAR .Click                # Instrucción para conectar dispositivo al módulo
do set ListPicker1 .Selection to call BluetoothClient1 .Connect
                                     address ListPicker1 .Selection
   set Label4 .Text to "CONECTADO"

when Slider1 .PositionChanged        # Instrucción para enviar dato del slider
thumbPosition
do set Label1 .Text to round Slider1 .ThumbPosition
   call BluetoothClient1 .Send1ByteNumber
                                     number round Slider1 .ThumbPosition

```

Fig. 4.13 Configuración de bloques de la aplicación con barra deslizante virtual

4.2 Simulación del programa del microcontrolador para el medidor de ritmo cardiaco

Para probar el código antes de llevarlo a la práctica se generó una simulación en Proteus 8 Professional, con la cual se representó el microcontrolador y un tren de pulsos similares a los del acondicionador de señal, además de un terminal serial para ver el dato enviado por el USART del microcontrolador. En la figura 4.14 se muestran las partes esenciales de la simulación.

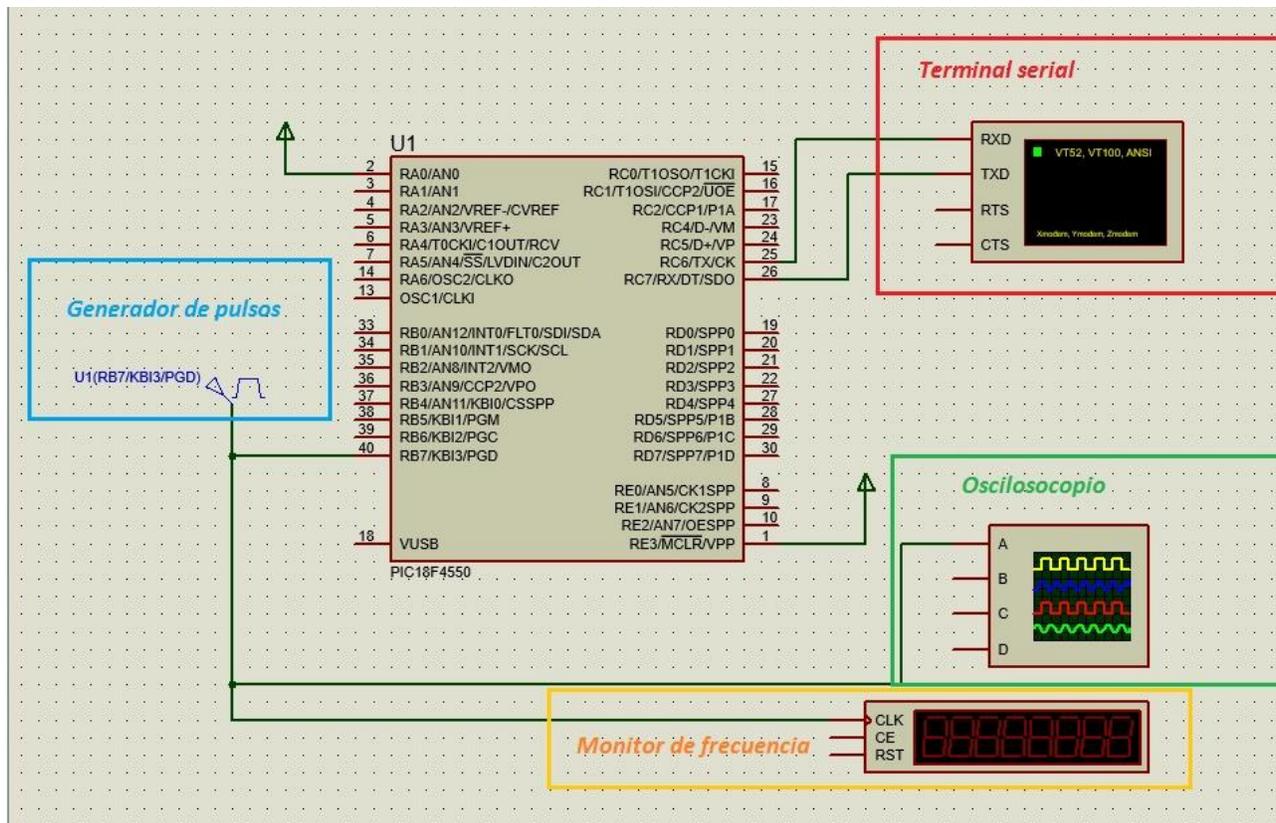


Fig. 4.14 Simulación del funcionamiento del microcontrolador

4.2.1 Prueba 1:

Para la primera prueba se generó un tren de pulsos con frecuencia de 1 Hz, para simular un ritmo cardiaco de 60 latidos por segundo, el cual fue conectado al puerto RB7 del microcontrolador. En la figura 4.15 se muestra el valor arrojado por el microcontrolador.

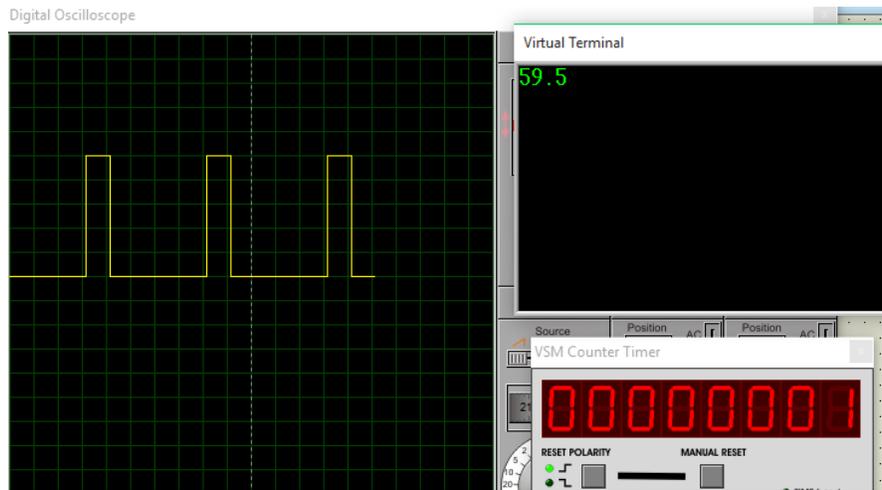


Fig. 4.15 Primer resultado de Simulación

Se puede observar en la figura 4.15 que el valor enviado es de 59.5 bpm, lo cual es bastante aceptable si se considera que el valor real es 60 bpm.

4.2.2 Prueba 2:

Para esta prueba se generó un tren de pulsos con frecuencia de 2 Hz, para simular un ritmo cardiaco de 120 latidos por segundo, el cual fue conectado al puerto RB7 del microcontrolador. En la figura 4.16 se muestra el valor arrojado por el microcontrolador.

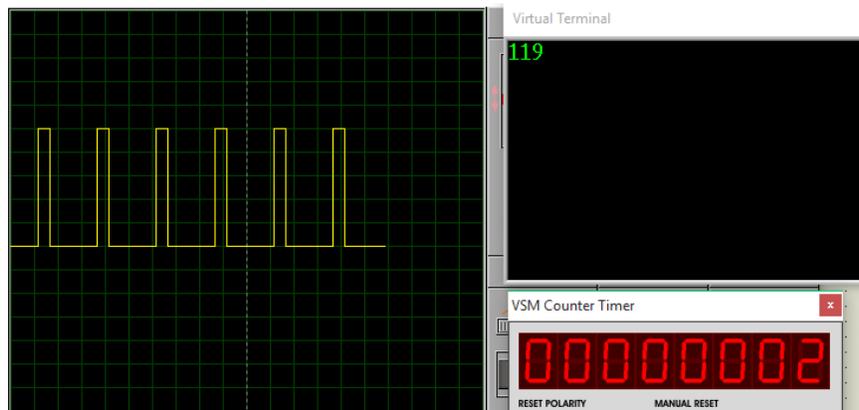


Fig. 4.16 Segundo resultado de Simulación

Como se observó en la figura anterior, el valor que envió el microcontrolador fue de 119 bpm, siendo 120 bpm el valor real.

4.2.3 Prueba 3:

Para esta última prueba se llevó a cabo la simulación con un voltaje bajo de 6.5 volts de suministro para verificar que la protección funciona correctamente. En la figura 4.17 se muestra el valor arrojado por el microcontrolador.

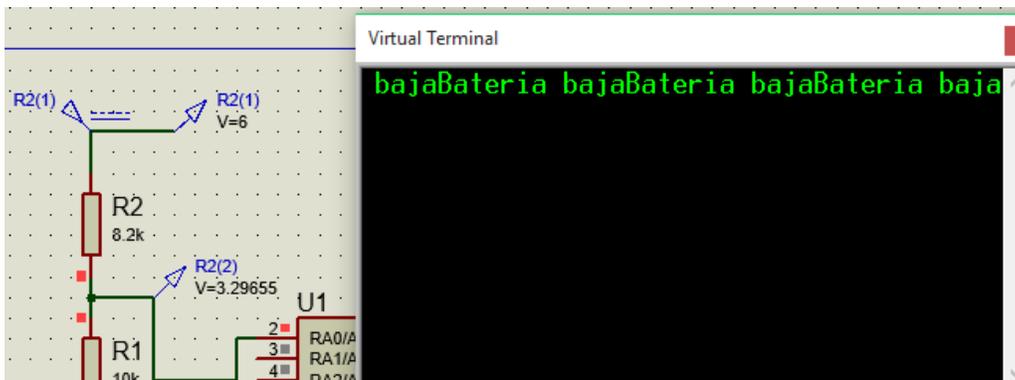


Fig. 4.17 Tercer resultado de Simulación

Como se aprecia en la figura anterior, en la terminal virtual se observa el mensaje de Bateria Baja, lo cual nos indica que funciona correctamente el programa del microcontrolador.

4.3 Pruebas físicas del prototipo final con la batería

Una vez terminado el prototipo y ser conectado en su totalidad, figuras 4.18 y 4.19, debe ser suministrado con la batería.



Fig. 4.18 Parte superior del medidor de ritmo cardiaco

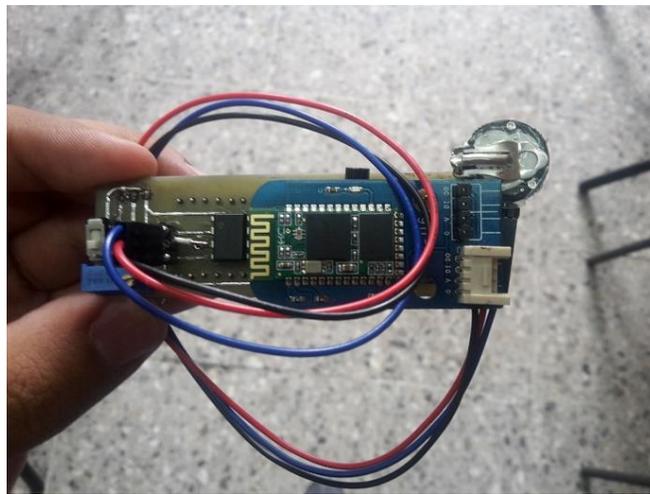


Fig. 4.19 Parte inferior del circuito (Bluetooth)

Cuando el sensor de ritmo cardiaco se energiza automáticamente encenderá el led emisor de color verde, lo cual sirve para indicarnos que está funcionando correctamente.



Fig. 4.20 Sensor de ritmo cardiaco

Es importante tener en cuenta el total de corriente que llega a consumir el prototipo ya trabajando de manera adecuada para saber el ahorro que tiene con respecto a otros dispositivos similares.



Fig. 4.21 Consumo de corriente 59.6 mAh

4.3.1 Enlace con el dispositivo Android y el prototipo final

Cuando se abra la aplicación en el celular aparecerá la siguiente pantalla, ver figura 22, la cual estará en espera de conectarse mediante Bluetooth al prototipo. Después de ser vinculada aparecerá un mensaje que dice CONECTADO, el cual nos ayudará a saber que el dispositivo se asoció correctamente, ver figura 4.23.

Una vez asociado de forma correcta, se deberá colocar el sensor en la oreja de la persona, una vez que el led azul parpadea quiere decir que está bien colocado el sensor y se enviará el valor al teléfono celular, en este caso se hizo una medición con una persona y el valor arrojado fue de 66.5 pulsaciones por minuto, después de esa prueba, se hizo una segunda lectura con un dispositivo comercial para verificar, el resultado fue de 66 pulsaciones por minuto, ver figura 4.24.



Fig. 4.22 Aplicación en espera de la vinculación bluetooth



Fig. 4.23 Aplicación asociada al bluetooth

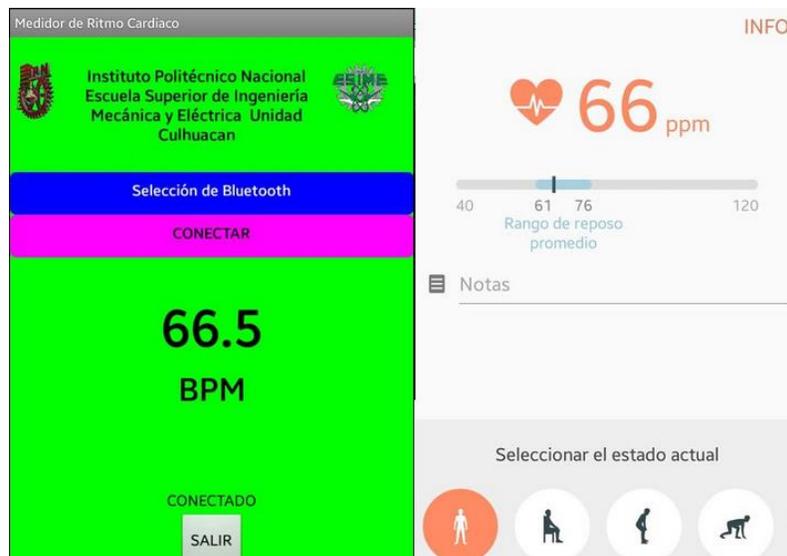


Fig. 4.24 Comparación del resultado del prototipo con el comercial

En la figura 4.25 se muestra una imagen en el momento de la prueba anterior, la prueba debe hacerse mientras la persona este calmada y no haga movimientos.

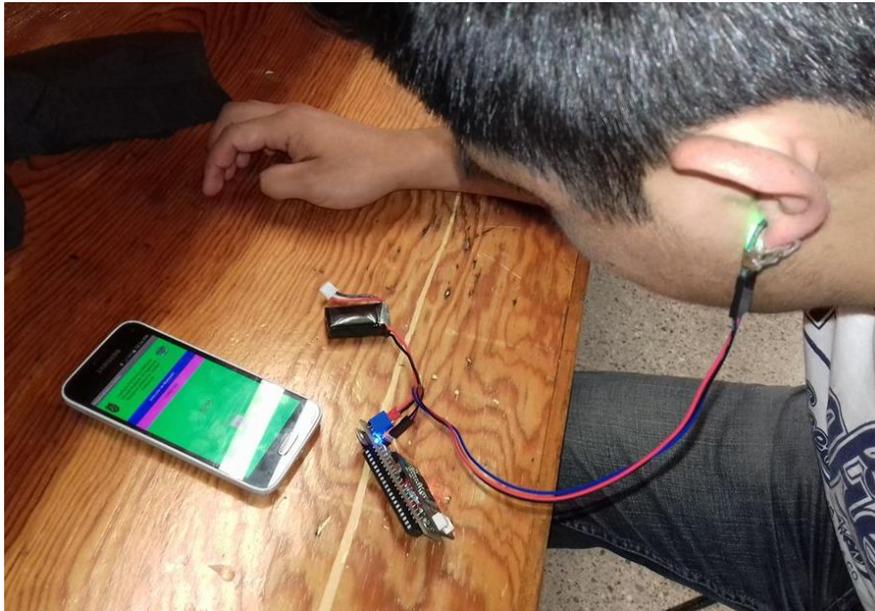


Fig. 4.25 Lectura de ritmo cardiaco a una persona

Una vez hecho la pruebas con personas en estado de reposo, se procedió a hacer una prueba cuando una persona eleva su ritmo cardiaco, lo cual se puede lograr haciendo que corra o que suba escaleras, en este caso se hizo la lectura a una persona después de correr, los resultados fueron bastante precisos entre el prototipo aquí presentado a un sistema comercial, en la figura 4.26 se muestra las capturas de pantalla con los resultados.

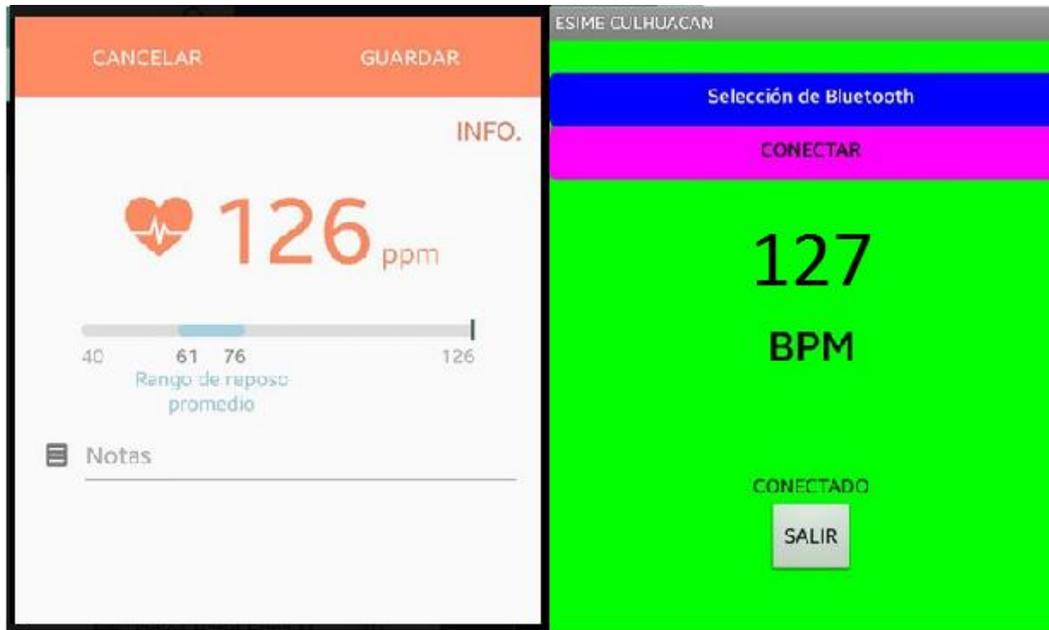


Fig. 4.26 Lectura de ritmo cardiaco a una persona después de correr

Como se puede observar, el aumento de ritmo cardiaco es considerable, y en ambas lectura se tienen valores similares en el mismo instante de tiempo, siendo 126 y 127 pulsaciones por minuto.

4.3.2 Pruebas finales con personas de la comunidad

En la siguiente tabla se muestran las pruebas que se realizaron a los alumnos de la ESIME Culhuacan, tiene por objetivo verificar y comparar las mediciones con respecto a un medidor de ritmo cardiaco comercial.

Tabla 4.1 Pruebas en alumnos

Nombre	Prototipo	Comercial
Luis	88	87
Ramón	66.5	67
Sofía	77	80
Yessica	70	70
Georgina	58	56
Gabriel	80.5	82
Daniel	73.5	72
Lorena	84	85
Yonathan	84	87
Pamela	73.5	76
Betsy	63	62
Cristopher	63	64
Raúl	70	71

Para entender mejor los resultados se ha generado una gráfica, ver figura 4.27, en la cual podemos observar que la tendencia de las curvas es casi idéntica, lo cual nos indica que la diferencia entre los resultados del prototipo que se presentan son similares a los de un sistema comercial.

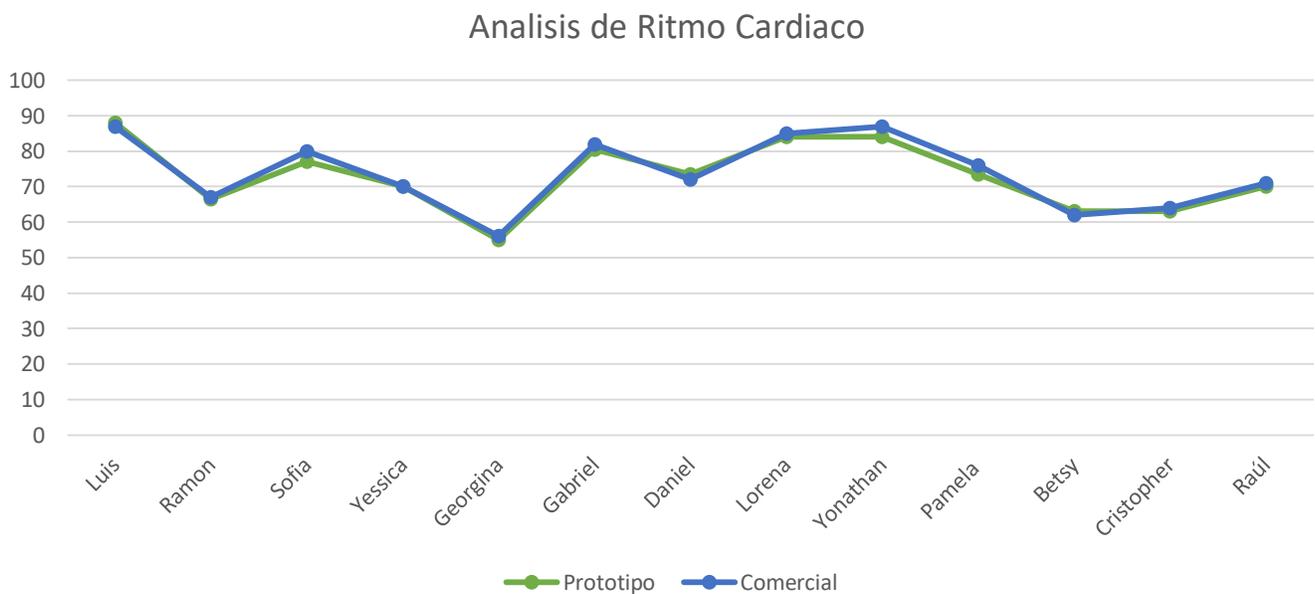


Fig. 4.27 Gráfica de comparativa de resultados

Los valores de ritmo cardiaco entre las personas varían, esto debido a que todos son de diferentes edades y condiciones físicas, sin embargo ambos sistemas hicieron lecturas muy parecidas entre sí, teniendo un rango de error de 3 unidades máximo.

Cabe mencionar que si la persona hace un movimiento o respirar profundamente, puede alterar la lectura real de su ritmo cardiaco, debido a eso existen en algunas ocasiones mayor diferencia entre lecturas.

Conclusiones

Con este trabajo se ha demostrado que gracias a las tecnologías actuales disponibles en México se pueden generar herramientas sofisticadas y fáciles de usar para cualquier usuario, las cuales ayudan a la sociedad a ser más consiente y responsable con su salud de una forma cómoda. Esto porque se genera un sistema el cual puede ser ampliamente usado; ya sea por doctores, deportistas o cualquier persona que requiera del monitoreo de su ritmo cardiaco.

Académicamente hablando, en el estado del arte no existen muchos trabajos donde se implemente este tipo de sistemas por medio de un enlace inalámbrico bluetooth a nivel licenciatura, lo cual hace que esta tesis se vuelva una propuesta nueva e interesante, por lo que se piensa que se convierte en un aporte considerable para distintas áreas; como la medicina, la electrónica y las comunicaciones.

Es necesario mencionar que se cubrieron todos los objetivos propuestos al principio del trabajo, sin embargo aún queda trabajo por hacer, algunas mejoras e ideas adicionales por implementar para hacerlo una herramienta mucho más completa, además se puede acceder a un campo de investigación muy amplio, ya sea desde el punto de vista médico con nuevas técnicas no invasivas o desde el punto de vista de ingeniería, haciendo mejoras al hardware o al algoritmo del software, así como estudiar mejor las señales del cuerpo y signos vitales para obtener más información con el mismo sistema electrónico.

ANEXOS

Condiciones de operación PIC18F4550



MICROCHIP PIC18F2455/2550/4455/4550

28/40/44-Pin, High-Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology

Universal Serial Bus Features:

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)
- 1-Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

Power-Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8 μ A typical
- Sleep mode currents down to 0.1 μ A typical
- Timer1 Oscillator: 1.1 μ A typical, 32 kHz, 2V
- Watchdog Timer: 2.1 μ A typical
- Two-Speed Oscillator Start-up

Flexible Oscillator Structure:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns ($T_{cy}/16$)
 - Compare is 16-bit, max. resolution 83.3 ns (T_{cy})
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I²C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

Special Microcontroller Features:

- C Compiler Optimized Architecture with optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
 - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Optional dedicated ICD/ICSP port (44-pin devices only)
- Wide Operating Voltage Range (2.0V to 5.5V)

Condiciones de operación regulador de voltaje 7805

μA7800 SERIES POSITIVE-VOLTAGE REGULATORS

SLVS056J – MAY 1978 – REVISED MAY 2003

recommended operating conditions

		MIN	MAX	UNIT
V_I Input voltage	μA7805C	7	25	V
	μA7808C	10.5	25	
	μA7810C	12.5	28	
	μA7812C	14.5	30	
	μA7815C	17.5	30	
	μA7824C	27	38	
I_O Output current		1.5		A
T_J Operating virtual junction temperature	μA7800C series	0	125	°C

electrical characteristics at specified virtual junction temperature, $V_I = 10$ V, $I_O = 500$ mA (unless otherwise noted)

PARAMETER	TEST CONDITIONS	T_J †	μA7805C			UNIT
			MIN	TYP	MAX	
Output voltage	$I_O = 5$ mA to 1 A, $V_I = 7$ V to 20 V, $P_D \leq 15$ W	25°C	4.8	5	5.2	V
		0°C to 125°C	4.75		5.25	
Input voltage regulation	$V_I = 7$ V to 25 V	25°C		3	100	mV
	$V_I = 8$ V to 12 V			1	50	
Ripple rejection	$V_I = 8$ V to 18 V, $f = 120$ Hz	0°C to 125°C	62	78		dB
Output voltage regulation	$I_O = 5$ mA to 1.5 A	25°C		15	100	mV
	$I_O = 250$ mA to 750 mA			5	50	
Output resistance	$f = 1$ kHz	0°C to 125°C		0.017		Ω
Temperature coefficient of output voltage	$I_O = 5$ mA	0°C to 125°C		-1.1		mV/°C
Output noise voltage	$f = 10$ Hz to 100 kHz	25°C		40		μV
Dropout voltage	$I_O = 1$ A	25°C		2		V
Bias current		25°C		4.2	8	mA
Bias current change	$V_I = 7$ V to 25 V	0°C to 125°C			1.3	mA
	$I_O = 5$ mA to 1 A				0.5	
Short-circuit output current		25°C		750		mA
Peak output current		25°C		2.2		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into

Condiciones de operación transistor 2N3904



2N3904

SMALL SIGNAL NPN TRANSISTOR

PRELIMINARY DATA

Ordering Code	Marking	Package / Shipment
2N3904	2N3904	TO-92 / Bulk
2N3904-AP	2N3904	TO-92 / Ammopack

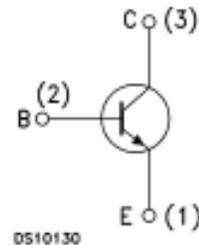
- SILICON EPITAXIAL PLANAR NPN TRANSISTOR
- TO-92 PACKAGE SUITABLE FOR THROUGH-HOLE PCB ASSEMBLY
- THE PNP COMPLEMENTARY TYPE IS 2N3906

APPLICATIONS

- WELL SUITABLE FOR TV AND HOME APPLIANCE EQUIPMENT
- SMALL LOAD SWITCH TRANSISTOR WITH HIGH GAIN AND LOW SATURATION VOLTAGE



INTERNAL SCHEMATIC DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CBO}	Collector-Base Voltage ($I_E = 0$)	60	V
V_{CEO}	Collector-Emitter Voltage ($I_B = 0$)	40	V
V_{EBO}	Emitter-Base Voltage ($I_C = 0$)	6	V
I_C	Collector Current	200	mA
P_{tot}	Total Dissipation at $T_C = 25^\circ\text{C}$	625	mW
T_{stg}	Storage Temperature	-65 to 150	$^\circ\text{C}$
T_J	Max. Operating Junction Temperature	150	$^\circ\text{C}$

2N3904

THERMAL DATA

$R_{thj-amb}$	Thermal Resistance Junction-Ambient	Max	200	$^{\circ}\text{C}/\text{W}$
$R_{thj-case}$	Thermal Resistance Junction-Case	Max	83.3	$^{\circ}\text{C}/\text{W}$

ELECTRICAL CHARACTERISTICS ($T_{case} = 25^{\circ}\text{C}$ unless otherwise specified)

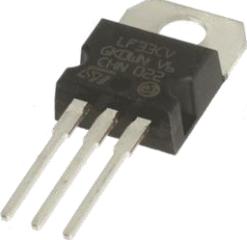
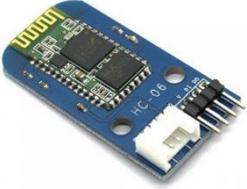
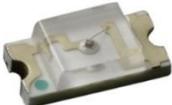
Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
I_{CEX}	Collector Cut-off Current ($V_{BE} = -3\text{ V}$)	$V_{CE} = 30\text{ V}$			50	nA
I_{BEX}	Base Cut-off Current ($V_{BE} = -3\text{ V}$)	$V_{CE} = 30\text{ V}$			50	nA
$V_{(BR)CEO}^*$	Collector-Emitter Breakdown Voltage ($I_B = 0$)	$I_C = 1\text{ mA}$	40			V
$V_{(BR)CBO}$	Collector-Base Breakdown Voltage ($I_E = 0$)	$I_C = 10\text{ }\mu\text{A}$	60			V
$V_{(BR)EBO}$	Emitter-Base Breakdown Voltage ($I_C = 0$)	$I_E = 10\text{ }\mu\text{A}$	6			V
$V_{CE(sat)}^*$	Collector-Emitter Saturation Voltage	$I_C = 10\text{ mA}$ $I_C = 50\text{ mA}$	$I_B = 1\text{ mA}$ $I_B = 5\text{ mA}$		0.2 0.2	V V
$V_{BE(sat)}^*$	Base-Emitter Saturation Voltage	$I_C = 10\text{ mA}$ $I_C = 50\text{ mA}$	$I_B = 1\text{ mA}$ $I_B = 5\text{ mA}$	0.85	0.85 0.95	V V
h_{FE}^*	DC Current Gain	$I_C = 0.1\text{ mA}$ $I_C = 1\text{ mA}$ $I_C = 10\text{ mA}$ $I_C = 50\text{ mA}$ $I_C = 100\text{ mA}$	$V_{CE} = 1\text{ V}$ $V_{CE} = 1\text{ V}$ $V_{CE} = 1\text{ V}$ $V_{CE} = 1\text{ V}$ $V_{CE} = 1\text{ V}$	60 80 100 60 30	300	
f_T	Transition Frequency	$I_C = 10\text{ mA}$ $V_{CE} = 20\text{ V}$ $f = 100\text{ MHz}$	250	270		MHz
C_{CBO}	Collector-Base Capacitance	$I_E = 0$ $V_{CB} = 10\text{ V}$ $f = 1\text{ MHz}$		4		pF
C_{EBO}	Emitter-Base Capacitance	$I_C = 0$ $V_{EB} = 0.5\text{ V}$ $f = 1\text{ MHz}$		18		pF
NF	Noise Figure	$V_{CE} = 5\text{ V}$ $I_C = 0.1\text{ mA}$ $f = 10\text{ Hz}$ to 15.7 KHz $R_G = 1\text{ K}\Omega$		5		dB
t_d	Delay Time	$I_C = 10\text{ mA}$ $I_B = 1\text{ mA}$			35	ns
t_r	Rise Time	$V_{CC} = 30\text{ V}$			35	ns
t_s	Storage Time	$I_C = 10\text{ mA}$ $I_{B1} = -I_{B2} = 1\text{ mA}$			200	ns
t_f	Fall Time	$V_{CC} = 30\text{ V}$			50	ns

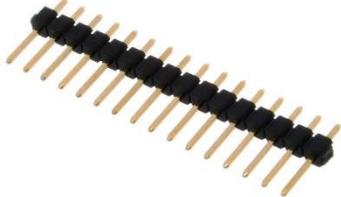
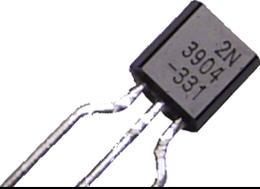
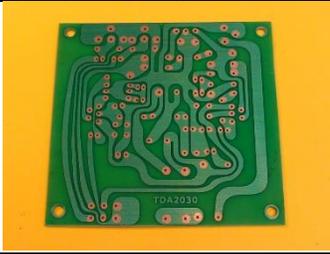
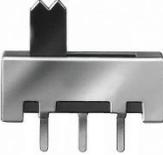
* Pulsed: Pulse duration = 300 μs , duty cycle $\leq 2\%$

Estudio de viabilidad

En este apartado se describen los costos de los materiales empleados, así como el costo final del proyecto, con el objetivo de mostrar que este trabajo es un sistema de bajo costo, considerando que los materiales pueden ser comprados por mayoreo para hacerlo aún más económicos.

- Tabla de costos directos

<i>Imagen</i>	<i>Elemento</i>	<i>Cantidad</i>	<i>Costo</i>
	<i>PIC18F4550</i>	<i>1</i>	<i>\$100.00</i>
	<i>Sensor SEN11574</i>	<i>1</i>	<i>\$350.00</i>
	<i>L7805CV</i>	<i>1</i>	<i>\$20.00</i>
	<i>HC-06</i>	<i>1</i>	<i>\$200.00</i>
	<i>Led superficial</i>	<i>2</i>	<i>\$10.00</i>

	<p><i>Header Hembra Torneado</i></p>	<p>2</p>	<p>\$20.00</p>
	<p><i>Header Macho</i></p>	<p>1</p>	<p>\$10.00</p>
	<p><i>Transistor 2N3904</i></p>	<p>2</p>	<p>\$18.00</p>
	<p><i>Fibra de vidrio</i></p>	<p>1</p>	<p>\$20.00</p>
	<p><i>Soldadura</i></p>	<p>1</p>	<p>\$50.00</p>
	<p><i>Batería Lipo 7.4v</i></p>	<p>1</p>	<p>\$290.00</p>
	<p><i>Interruptor 2 Posiciones</i></p>	<p>1</p>	<p>\$12.00</p>

	<i>Amplificador Operacional TL081</i>	<i>1</i>	<i>\$25.00</i>
	<i>Resistor</i>	<i>6</i>	<i>\$5.00</i>
	<i>Resistor Superficial</i>	<i>2</i>	<i>\$5.00</i>
	<i>Trimpot 100k</i>	<i>1</i>	<i>\$15.00</i>
<i>Total</i>			<i>\$1150.00</i>

Normatividad

El estándar IEEE 802.15.1 presenta una WPAN que utiliza tecnología inalámbrica Bluetooth. En este trabajo el término Bluetooth WPAN o simplemente el término IEEE 802.15.1 WPAN se refiere a una WPAN que utiliza tecnología Bluetooth inalámbrica.

La tecnología inalámbrica de Bluetooth utiliza un radio de corto alcance que ha sido optimizado para el ahorro de energía, operación adecuada de la batería, tamaño pequeño y para ser utilizada en aparatos personales de bajo peso. Una WPAN es capaz de soportar canales síncronos de comunicación para telefonía de voz y canales de comunicación asíncronas para comunicación de datos. Dichas facilidades permiten que una amplia gama de aplicaciones y de aparatos trabaje en una WPAN. Por ejemplo, un teléfono celular puede usar canales circuit-switched para transportar audio desde y para un receptor en el encabezado mientras se encuentra utilizando un canal packet-switched para intercambiar datos con una computadora portátil.

Una WPAN opera en la banda libre de los 2.4 GHz. Se utiliza un transceptor de fast frequency-hop (1600 hops/s) para evitar la interferencia y la caída de señales. Para reducir la complejidad del transceptor se utiliza la técnica binaria FSK (frequency shift keying) para transmitir símbolos con un rango de 1 Símbolos/s. Se utiliza un canal ranurado, cada ranura de tiempo tiene una duración de 625 μ s.

Glosario

Amateurs - *Que practica por placer una actividad, generalmente deportiva o artística, sin recibir habitualmente dinero a cambio.*

Arteria - *Vaso sanguíneo que conduce la sangre desde el corazón a las diversas partes del organismo.*

Bit - *En informática y otras disciplinas, unidad mínima de información, que puede tener solo dos valores (cero o uno).*

Byte - *Conjunto de 8 bits que recibe el tratamiento de una unidad y que constituye el mínimo elemento de memoria direccionable de una computadora.*

Corriente - *Magnitud física que nos indica la cantidad de electricidad que recorre un conductor, durante una unidad de tiempo.*

Conmutación - *Se considera como la acción de establecer una vía, un camino, de extremo a extremo entre dos puntos.*

CPLD - *Chip que puede emular el comportamiento de miles de puertas lógicas interconectadas entre sí y junto con algunos bits de estado, registros o flip-flops.*

Frecuencia - *Es una magnitud que mide el número de repeticiones por unidad de tiempo de cualquier fenómeno o suceso periódico.*

FPGA - *Dispositivo programable que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada mediante un lenguaje de descripción especializado.*

Glucosa - *La glucosa es una forma simple de azúcar que cumple una importante función en nuestro organismo.*

GPS - *Sistema americano de navegación y localización mediante satélites.*

Microcontrolador - *Es un circuito integrado que en su interior contiene una unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM), puertos de entrada y salida y periféricos.*

Hardware - *Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.*

Interfaz - *Dispositivo capaz de transformar las señales generadas por un aparato en señales comprensibles por otro.*

Monitorear - *Controlar el desarrollo de una acción o un suceso a través de uno o varios monitores.*

Omnidireccional - *Que se puede utilizar en todas las direcciones o sentidos.*

Oscilador - Sistema capaz de crear perturbaciones o cambios periódicos.

PCB - Printed Circuit Board, es la superficie constituida por caminos, pistas o buses de material conductor laminadas sobre una base no conductora.

RAM - Memoria de acceso aleatorio, memoria principal de la computadora, donde residen programas y datos.

ROM - Memoria de solo lectura, circuito integrado de memoria de solo lectura que almacena instrucciones y datos de forma permanente.

Smartphone - Teléfono celular con pantalla táctil, que permite al usuario conectarse a internet, gestionar cuentas de correo electrónico e instalar otras aplicaciones y recursos a modo de pequeño computador.

Software - Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

Temporizador - Dispositivo eléctrico que regula de forma automática el encendido y el apagado de una máquina, un instrumento, etc.

Transistor - Dispositivo semiconductor que cierra o abre un circuito o amplifica una señal.

TTL - Lógica transistor a transistor. Es una tecnología de construcción de circuitos electrónicos digitales.

USART - Dispositivo que sirve para transmitir o recibir datos secuenciales de manera síncrona o asíncrona.

Vascular - Relativo a los vasos o los conductos por los que circula la sangre u otros líquidos.

Voltaje - Potencial eléctrico, expresado en voltios.

Referencias bibliográficas y Cibergrafía

- *Wilmore, Jack H. & Costill, David L. - Fisiología del esfuerzo y del deporte*
- *Mercedes Granda miguel elena Mediavilla bolado “textos universitarios”
Instrumentación electrónica: Transductores y acondicionadores de señal
Editorial universidad Cantabria*
- *Adel S. Sedra & Kenneth C. Smith Circuitos Microelectrónicos Oxford University
Press*
- <http://www.inegi.org.mx/est/contenidos/proyectos/registros/vitales/mortalidad/tabulados/ConsultaMortalidad.asp>
- <https://es.wikipedia.org/wiki/Gluc%C3%B3metro>
- https://es.wikipedia.org/wiki/Ritmo_card%C3%ADaco
- <http://www.fundaciondelcorazon.com/prevencion/riesgo-cardiovascular/frecuencia-cardiaca.html>
- <http://www.imujer.com/salud/5165/como-medir-la-frecuencia-cardiaca>
- <http://www.cochesrc.com/conceptos-basicos-baterias-lipo-tension-capacidad-descarga-y-conexion-de-elementos-a2627.html>