



REPÚBLICA DEL ECUADOR

Escuela Politécnica Nacional

" E S C I E N T I A H O M I N I S S A L U S "

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.

Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.

No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

*Respeto hacia sí mismo y hacia los demás.*

# ESCUELA POLITÉCNICA NACIONAL

## FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

### DISEÑO Y CONSTRUCCIÓN DE UN EQUIPO PARA MONITOREO DE PARÁMETROS DE INTERÉS EN EL ENTRENAMIENTO DE UN ATLETA

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA Y TELECOMUNICACIONES

VINICIO RODRIGO AGUAGUIÑA CADENA

[viniaguica@gmail.com](mailto:viniaguica@gmail.com)

RUBÉN MAURICIO ENRÍQUEZ AGUILAR

[mauricioe80@gmail.com](mailto:mauricioe80@gmail.com)

DIRECTOR: DR. LUIS ANÍBAL CORRALES PAUCAR

[luis.corrales@epn.edu.ec](mailto:luis.corrales@epn.edu.ec)

CODIRECTOR: ING. OSCAR FERNANDO VÁSQUEZ YÉPEZ

[fvasquez@watching.com.ec](mailto:fvasquez@watching.com.ec)

Quito, Junio 2015

## DECLARACIÓN

Nosotros, Vinicio Rodrigo Aguaguiña Cadena y Rubén Mauricio Enríquez Aguilar, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

Vinicio Aguaguiña Cadena

---

Rubén Enríquez Aguilar

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Vinicio Rodrigo Aguaguña Cadena y Rubén Mauricio Enríquez Aguilar, bajo mi supervisión.

---

Dr. Luis Corrales  
DIRECTOR DE PROYECTO

---

Ing. Fernando Vásquez  
CODIRECTOR DE PROYECTO

## AGRADECIMIENTO

Primero a Dios y a la Virgen María quienes me acompañaron y me dieron la fuerza en el transcurso de todo el proceso para culminar esta meta, así como al Dr. Luis Corrales y al Ingeniero Fernando Vásquez por su gran paciencia, guía y sobre todo por creer en mi para realizar este proyecto.

A mis padres que me apoyaron en todo momento, a mis hermanos. A quienes se convirtieron en unos hermanos incondicionales como son los Ingenieros Katherine Guerra, Ximena Cordero, Gladys Cabascango y Ramiro Peñafiel.

Y un especial agradecimiento a mi compañero de tesis Mauricio Enríquez y a la Arquitecta Daniela Ortiz con quienes logramos convertir en realidad este sueño.

*Vinicio*

## DEDICATORIA

Dedico este trabajo a mis padres María y Juan, a mis hermanos Omar, Adriana y David a mis sobrinos Soraya, Ami, Jhoan y Evan, quienes, con su apoyo, esfuerzo, dedicación y alegría me dieron la energía necesaria para seguir con la culminación de este trabajo.

A las personas que en el transcurso de mi aprendizaje contribuyeron con sus enseñanzas, así como a esos amigos que se convirtieron en unos hermanos más como son Kate, Xime, Gladys y Ramiro. Y sobre todo a Dany quien me dio la fuerza para seguir con esta meta.

*Vinicio*

## AGRADECIMIENTO

Agradezco a Dios y a la Virgen María por sus infinitas y generosas bendiciones en mi vida, por darme la fuerza necesaria para continuar adelante y por permitirme disfrutar de momentos inolvidables en el tiempo con las personas que más amo mis padres y mí hermana.

A mis padres Washington Enríquez y María Elisa Aguilar, ya que gracias a sus enseñanzas y buenos valores nos inculcaron a mi hermana y a mí a ser siempre personas de bien y a mantenernos juntos como familia a pesar de la distancia. Papá, mamá, por su admirable perseverancia en todos los sentidos, por desvivirse por sus hijos, voluntad y amor inconmensurables les agradezco infinitamente.

A mi hermana Genny Enríquez por ser un ángel en mi vida, por ser una persona incondicional en cada momento, gran amiga y ser parte de la alegría de mi vida le doy mi más grande agradecimiento con mucho cariño y amor.

A Berthy Loor por brindarme su ayuda en la realización de esta meta anhelada, muchísimas gracias.

A mi amigo Vinicio Aguaguña le agradezco el haberme permitido realizar este proyecto y juntos como compañeros de tesis formar un gran equipo de trabajo. Amigo Vini Dios os bendiga a ti y a tu familia siempre.

Mi más sincero agradecimiento al Dr. Luis Corrales y al Ing. Fernando Vásquez, por su paciencia, tiempo y guías necesarias en la realización de este proyecto.

*Mauricio*

## DEDICATORIA

Dedico este sueño tan anhelado a mi familia, Padrecito, Madrecita y Ñañita ustedes son el tesoro más grande que Dios me ha dado, sentir su calidez y amor es como vivir con el abrazo y bendición de Dios. Los amo mucho.

*Mauricio*

## CONTENIDO

<b>DECLARACIÓN</b> .....	<b>I</b>
<b>CERTIFICACIÓN</b> .....	<b>II</b>
<b>AGRADECIMIENTO</b> .....	<b>III</b>
<b>DEDICATORIA</b> .....	<b>IV</b>
<b>AGRADECIMIENTO</b> .....	<b>V</b>
<b>DEDICATORIA</b> .....	<b>VI</b>
<b>CONTENIDO</b> .....	<b>VII</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>XI</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>XIII</b>
<b>RESUMEN</b> .....	<b>XIV</b>
<b>PRESENTACIÓN</b> .....	<b>XV</b>
<b>CAPÍTULO 1. PARÁMETROS Y DISPOSITIVOS A UTILIZAR EN EL SISTEMA PARA ENTRENAMIENTO</b> .....	<b>1</b>
<b>1.1 INTRODUCCIÓN</b> .....	<b>1</b>
<b>1.2 EL ENTRENAMIENTO [1]</b> .....	<b>1</b>
1.2.1 DEFINICIÓN.....	1
1.2.2 ESTRUCTURA DEL PLAN DE ENTRENAMIENTO [2] .....	3
1.2.3 PREPARACIÓN PARA EL ENTRENAMIENTO [2] .....	3
1.2.3.1 La entrada en calor [2] .....	4
1.2.3.2 Entradas en calor orientativas [2] .....	4
1.2.3.3 Sesión o trabajo principal [2] .....	5
1.2.3.4 Vuelta a la calma o enfriamiento [2].....	6
<b>1.3 PARÁMETROS DEL ENTRENAMIENTO</b> .....	<b>7</b>
1.3.1 TIEMPO [3] .....	7
1.3.1.1 El tiempo y como se relaciona en los entrenamientos y competencias [26].....	8
1.3.1.2 Consideraciones sobre el reglamento de la toma de tiempo [4].....	8
1.3.2 FRECUENCIA CARDIACA.....	9
1.3.2.1 Definición [5] .....	9
1.3.2.2 Relación de la frecuencia cardiaca en un entrenamiento [27] .....	10
1.3.2.3 Frecuencia cardiaca en reposo o en descanso [27].....	11
1.3.2.4 Frecuencia cardiaca máxima [6].....	12
1.3.2.4.1 Electrocardiograma ECG, ondas y segmentos .....	13
1.3.2.4.2 Contador de Frecuencia Cardiaca .....	14
1.3.3 TEMPERATURA CORPORAL .....	15

1.3.3.1	Valores de medición .....	15
1.3.3.2	Selección del lugar de medición .....	15
<b>1.4</b>	<b>DESCRIPCIÓN DEL SISTEMA .....</b>	<b>16</b>
1.4.1	MONITOR DE FRECUENCIA CARDIACA .....	17
1.4.1.1	Descripción general de dispositivos .....	17
1.4.1.2	Selección del dispositivo .....	17
1.4.2	TRANSDUCTORES DE TEMPERATURA [28] .....	18
1.4.2.1	Conceptos básicos .....	18
1.4.2.2	Tipos y características .....	19
1.4.3	MEDICIÓN DEL TIEMPO .....	20
1.4.3.1	Reloj en tiempo real [20] .....	20
1.4.4	DISPOSITIVOS DE TRANSMISIÓN Y RECEPCIÓN .....	21
1.4.4.1	Descripción del módulo transceptor seleccionado de RF. ....	21
1.4.5	MICROCONTROLADOR .....	23
1.4.5.1	AVR ATMEL risc en 8 bit [21] .....	23
1.4.5.2	<i>Software</i> para programación para el Microcontrolador ATMEGA 8. ....	25
	<b>CAPÍTULO 2. DISEÑO DEL SISTEMA. ....</b>	<b>26</b>
<b>2.1</b>	<b>CONSIDERACIONES DE DISEÑO. ....</b>	<b>26</b>
<b>2.2</b>	<b>DISEÑO DEL MÓDULO DEL ATLETA. ....</b>	<b>26</b>
2.2.1	REQUERIMIENTOS DEL MÓDULO DEL ATLETA. ....	27
2.2.2	DISEÑO DEL MONITOR DE FRECUENCIA CARDIACA .....	28
2.2.2.1	Manejo del <i>PulseSensor</i> .....	28
2.2.2.2	Descripción del proceso de la contabilización de pulsos .....	31
2.2.2.3	Descripción del muestreo de pulsos. ....	32
2.2.3	DISEÑO DEL MONITOR DE TEMPERATURA CORPORAL .....	35
2.2.3.1	Comandos de Operación .....	36
2.2.4	DISEÑO DEL TRANSMISOR .....	37
<b>2.3</b>	<b>DISEÑO DEL MÓDULO DEL ENTRENADOR .....</b>	<b>39</b>
2.3.1	REQUERIMIENTOS DEL MÓDULO DEL ENTRENADOR .....	40
2.3.2	DISEÑO CON EL RELOJ DE TIEMPO REAL .....	41
2.3.2.1	Comandos de operación .....	43
2.3.3	DISEÑO DE LOS TRANSMISORES. ....	46
2.3.4	DISEÑO DE LA INTERFAZ DE INGRESO Y SELECCIÓN DE DATOS .....	46
2.3.4.1	Comandos de Operación .....	47
2.3.5	INTERFAZ VISUAL PARA MODULO DEL ENTRENADOR .....	48

2.3.5.1	Descripción del dispositivo a utilizar LCD 16*2 .....	48
2.3.6	INTERFAZ MÓDULO DEL ENTRENADOR - PC. ....	49
2.3.6.1	Circuito Conversor de señal TTL a señal RS-232. [21] .....	49
2.3.6.2	Adaptador RS-232 a USB .....	50
2.3.7	DESARROLLO DE LA INTERFAZ DE DESCARGA Y VISUALIZACIÓN EN EL COMPUTADOR. ....	51
2.3.7.1	Requerimientos para el Software. ....	51
<b>2.4</b>	<b>ESQUEMA GENERAL DEL SISTEMA. ....</b>	<b>51</b>
<b>CAPÍTULO 3. CONSTRUCCIÓN DEL SISTEMA PARA ENTRENAMIENTOS ..</b>		<b>54</b>
<b>3.1</b>	<b>CONSTRUCCIÓN DEL MÓDULO DEL ATLETA. ....</b>	<b>54</b>
3.1.1	PRUEBAS INDIVIDUALES DE LOS BLOQUES DEL MÓDULO DEL ATLETA. ....	54
3.1.1.1	Bloque del Microcontrolador. ....	54
3.1.1.2	Bloque de Frecuencia Cardiaca .....	56
3.1.1.3	Bloque de Temperatura. ....	58
3.1.1.4	Bloque de Transmisión y Recepción. ....	59
3.1.1.5	Bloque de Alimentación. ....	61
3.1.2	CONSTRUCCIÓN DEL CIRCUITO FINAL DEL ATLETA. ....	61
3.1.3	DIAGRAMA DE FLUJO DE FUNCIONAMIENTO DEL MÓDULO DEL ATLETA. ....	64
<b>3.2</b>	<b>CONSTRUCCIÓN DEL MÓDULO DEL ENTRENADOR. ....</b>	<b>67</b>
3.2.1	PRUEBAS INDIVIDUALES DE LOS BLOQUES DEL MÓDULO DEL ENTRENADOR. ...	69
3.2.1.1	Bloque del Microcontrolador .....	69
3.2.1.2	Bloques del Reloj, Ingreso y Selección de Datos. ....	69
3.2.1.3	Bloque de Visualización en LCD. ....	70
3.2.1.4	Bloque de Transmisión y Recepción. ....	70
3.2.1.5	Bloque de Comunicación con el PC. ....	70
3.2.1.6	Bloque de Alimentación. ....	72
3.2.2	CONSTRUCCIÓN DEL CIRCUITO FINAL DEL ENTRENADOR. ....	73
3.2.3	DIAGRAMA DE FLUJO DE FUNCIONAMIENTO DEL MÓDULO DEL ENTRENADOR. ....	75
3.2.3.1	Diagrama de Flujo de Funcionamiento de la Opción de Menú 1Clock .....	77
3.2.3.2	Diagrama de Flujo de Funcionamiento de la opción de menú 2FCR (Frecuencia Cardiaca en Reposo) .....	79
3.2.3.3	Diagrama de Flujo de Funcionamiento de la opción de menú 3Sesion (Sesión de Entrenamiento) .....	83
3.2.3.4	Diagrama de Flujo de Funcionamiento de la Opción de Menú 4Load PC. ....	86
<b>3.3</b>	<b>DESARROLLO DEL SOFTWARE PARA LA INTERFAZ GRÁFICA DEL COMPUTADOR. ....</b>	<b>88</b>

3.3.1	DIAGRAMA DE FLUJO DE FUNCIONAMIENTO DE LA INTERFAZ EN EL PC.....	88
<b>3.4</b>	<b>FIJACIÓN DEL MÓDULO DEL ATLETA EN EL DEPORTISTA.....</b>	<b>93</b>
<b>3.5</b>	<b>COSTO DEL PROYECTO. ....</b>	<b>94</b>
3.5.1	COSTO DEL MÓDULO DEL ENTRENADOR.....	94
3.5.2	COSTO DEL MÓDULO DEL ATLETA. ....	96
3.5.3	COSTO TOTAL DEL PROYECTO.....	97
<b>CAPÍTULO 4. PRUEBAS DE OPERACIÓN Y ANÁLISIS DE RESULTADOS.....</b>		<b>98</b>
<b>4.1</b>	<b>PRUEBAS PARA IGUALAR RELOJ. ....</b>	<b>98</b>
4.1.1	ANÁLISIS DE RESULTADOS.....	99
<b>4.2</b>	<b>PRUEBAS DE FRECUENCIA CARDIACA EN REPOSO.....</b>	<b>99</b>
4.2.1	ANÁLISIS DE RESULTADOS.....	100
<b>4.3</b>	<b>PRUEBAS EN SESIÓN DE ENTRENAMIENTO. ....</b>	<b>102</b>
4.3.1	ANÁLISIS DE RESULTADOS.....	107
<b>4.4</b>	<b>PRUEBAS DE DESCARGA Y GRÁFICA DE DATOS.....</b>	<b>108</b>
4.4.1	DESCARGA DE DATOS:.....	108
4.4.2	GRÁFICA DE DATOS: .....	110
4.4.3	ANÁLISIS DE RESULTADOS.....	111
<b>CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES .....</b>		<b>112</b>
<b>5.1</b>	<b>CONCLUSIONES:.....</b>	<b>112</b>
<b>5.2</b>	<b>RECOMENDACIONES:.....</b>	<b>113</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>		<b>114</b>
<b>ANEXO A. DATASHEET MICROCONTROLADOR ATMEGA 8.....</b>		<b>118</b>
<b>ANEXO B. DATASHEET DS18B20.....</b>		<b>125</b>
<b>ANEXO C. DATASHEET DS1307.....</b>		<b>143</b>
<b>ANEXO D. DATASHEET LCD 16x2.....</b>		<b>154</b>
<b>ANEXO E. DATASHEET MAX232.....</b>		<b>165</b>
<b>ANEXO F. DATASHEET PULSESENSOR.....</b>		<b>171</b>
<b>ANEXO G. DATASHEET YS-1020UB.....</b>		<b>178</b>

## ÍNDICE DE FIGURAS

Figura 1. 1 Clasificación de la Intensidad, según la Edad y la Frecuencia Cardíaca Máxima.....	11
Figura 1. 2 Señal ECG. ....	14
Figura 1. 3 Diagrama de Módulos. ....	16
Figura 1. 4 Sensor de Pulsos. ....	18
Figura 1. 5 Sensor de Temperatura DS1820.....	20
Figura 1. 6 YS 1020UB Data Transceiver. ....	22
Figura 2. 1 Módulo del Atleta.....	27
Figura 2. 2 Forma de Onda de la Salida del PulseSensor. [11].....	28
Figura 2. 3 Región de Corte y Saturación de un Transistor .....	29
Figura 2. 4 Transistores NPN y PNP en Modo Switch en Base a la Región de Corte y Saturación. [16].....	29
Figura 2. 5 PulseSensor Acoplado al Transistor PNP Switch.....	31
Figura 2. 6 Diagrama de Conexiones del Sensor de Temperatura DS 18B20. [18].....	35
Figura 2. 7 Dimensiones y Distribución Pines del YS-1020UB. [19].....	38
Figura 2. 8 Tipos de Conexiones del Transceiver YS-1020UB. [23].....	39
Figura 2. 9 Diagrama de Conexiones del DS 1307. [20].....	41
Figura 2. 10 Bytes para la Configuración de Hora y Fecha del DS 1307. [20].....	43
Figura 2. 11 Diagrama de Conexión del Microcontrolador y Pulsadores. ....	46
Figura 2. 12 LCD 16*2. ....	48
Figura 2. 13 Diagrama de Conexiones de la Interfaz Microcontrolador – PC. ....	49
Figura 2. 14 Conexiones de Circuito entre un Microcontrolador ATMEGA 8 y el MAX232. .....	50
Figura 2. 15 Fotografía de Adaptador RS-232 a USB.....	51
Figura 2. 16 Esquema General del Sistema. ....	52
Figura 3. 1 Circuito del Módulo del Atleta.....	55
Figura 3. 2 Prueba en Protoboard del PulseSensor y Switch. ....	56
Figura 3. 3 Prueba de Ingreso de Pulsos al Microcontrolador con un Pulsador. ....	56
Figura 3. 4 Resultados mostrados en PC en sesión de Hyper Terminal de Ingreso de Pulsos al Microcontrolador.....	57
Figura 3. 5 Pinza de Sujeción.....	57
Figura 3. 6 PulseSensor en Lóbulo de la Oreja.....	58
Figura 3. 7 Entrenador de Microcontrolador con el Sensor de Temperatura DS18B20. ...	58
Figura 3. 8 Pruebas de los Tranceivers para Transmisión y Recepción de Datos. ....	60
Figura 3. 9 Resultado de los Datos Transmitidos y Recibidos como Caracteres ASCII....	60
Figura 3. 10 Batería SONY tipo N. ....	61
Figura 3. 11 Circuito Fabricado del Módulo del Atleta.....	62
Figura 3. 12 Piezas que conforman la Caja del Módulo del Atleta.....	62
Figura 3. 13 Módulo del Atleta terminado.....	63
Figura 3. 14 Diagrama de funcionamiento del Módulo del Atleta. ....	64
Figura 3. 15 Circuito del Módulo del Entrenador.....	68
Figura 3. 16 Prueba de Funcionamiento del Reloj DS1307. ....	69
Figura 3. 17 Resultados del Reloj en LCD. ....	69

Figura 3. 18 Puerto COM detectando con el adaptador USB a RS-232. ....	70
Figura 3. 19 Configuración de Puerto COM para Sesión de Hyper terminal .....	71
Figura 3. 20 Resultados de Hyper terminal del Cable Adaptador USB a RS-232. ....	71
Figura 3. 21 Conexión Cable Adaptador USB a RS-232 a PC. ....	72
Figura 3. 22 Pruebas de MAX232 – Adaptador USB a RS232 – PC.....	72
Figura 3. 23 Batería GP 200.....	73
Figura 3. 24 Circuito Fabricado del Módulo del Entrenador. ....	73
Figura 3. 25 Piezas que conforman la Caja del Módulo del Entrenador. ....	74
Figura 3. 26 Módulo del Entrenador terminado. ....	74
Figura 3. 27 Diagrama para la Selección de Procesos del Módulo del Entrenador .....	75
Figura 3. 28 Menú a mostrarse en el LCD. ....	76
Figura 3. 29 Diagrama de Funcionamiento para igualar el Reloj. ....	77
Figura 3. 30 Diagrama de Funcionamiento para Medir la FCR. ....	80
Figura 3. 31 Diagrama de Funcionamiento para toma de Datos en Sesión de Entrenamiento.....	83
Figura 3. 32 Diagrama de Funcionamiento para la Descarga de Datos hacia el PC.....	86
Figura 3. 33 Diagrama de Funcionamiento de la Interfaz Gráfica en Computador. ....	89
Figura 3. 34 Primera pantalla de Presentación del Proyecto .....	90
Figura 3. 35 Segunda Pantalla de Menú Principal.....	90
Figura 3. 36 Tercera Pantalla de Descarga de Datos. ....	91
Figura 3. 37 Cuarta Pantalla de Graficar Datos. ....	92
Figura 3. 38 Ubicación del Módulo del Atleta en el Deportista. ....	94
Figura 4. 1 LCD muestra Reloj Igualado. ....	99
Figura 4. 2 Datos obtenidos de la Frecuencia Cardíaca en Reposo.....	100
Figura 4. 3 PulseSensor conectado en PC0/ADC del Microcontrolador. ....	103
Figura 4. 4 Señal ECG. ....	104
Figura 4. 5 Señal Digitalizada ECG del PulseSensor.....	105
Figura 4. 6 Datos en Sesión de Entrenamiento.....	106
Figura 4. 7 Instructivo de pasos de la ventana de Descarga de Datos .....	109
Figura 4. 8 Ventana de Descarga de Datos e Información.....	109
Figura 4. 9 Gráfica de Datos obtenidos del Módulo.....	111

## ÍNDICE DE TABLAS

Tabla 1. 1 Ejemplo de entradas en calor orientativas. ....	5
Tabla 1. 2 Ejemplo de la vuelta a la calma.....	7
Tabla 1. 3 Valores medios de la Frecuencia Cardiaca en Reposo según la edad (Hombres).....	11
Tabla 1. 4 Valores medios de la Frecuencia Cardiaca en Reposo según la edad (Mujeres) .....	12
Tabla 1. 5 Tabla de Transductores de Temperatura. ....	19
Tabla 2. 1 Comandos de Configuración del DS 18B20. [18].....	36
Tabla 2. 2 Descripción y Funcionalidad de Pines del YS-1020UB. [19].....	38
Tabla 3. 1 Costo del Módulo del Entrenador. ....	96
Tabla 3. 2 Costo del Módulo del Atleta. ....	97
Tabla 3. 3 Costo Total del Proyecto .....	97
Tabla 4. 1 Información comparativa de datos medidos utilizando el sistema de monitoreo y métodos tradicionales.....	101
Tabla 4. 2: Cálculo de Errores.....	101
Tabla 4. 3 Ritmo Cardiaco y su duración del intervalo QT. ....	104
Tabla 4. 4 Información comparativa de datos medidos utilizando el sistema de monitoreo y métodos tradicionales en sesión de entrenamimientto. ....	107
Tabla 4. 5 Cálculo de errores para la Sesión de Entrenamiento .....	108
Tabla 4. 6 Archivo Excel Almacenados de la Sesión de Entrenamiento.....	110

## RESUMEN

El objetivo de este proyecto es el diseño y construcción de un sistema para entrenamientos que ayude a los entrenadores y atletas a registrar parámetros que les permita evaluar el sistema de preparación adoptado para satisfacer las expectativas de una sesión de entrenamiento deportiva para la mejora del nivel del atleta.

De acuerdo a este objetivo se diseñó y construyó una solución que consta de 2 módulos, uno para el atleta y otro para el entrenador. El módulo del atleta permite tomar parámetros de frecuencia cardíaca y temperatura los cuales son transmitidos inalámbricamente al módulo del entrenador. El entrenador puede observar en vivo durante el tiempo de la sesión de entrenamiento la evolución de los valores de estos signos vitales del atleta a través de un LCD, permitiendo tomar decisiones en ese instante. Además, una vez terminada la sesión de entrenamiento, el sistema permite descargar los datos obtenidos hacia un PC para un análisis posterior, permitiendo al entrenador tomar decisiones importantes sobre las series de ejercicios que realiza el deportista, con el objeto de orientar al atleta en un incremento constante del rendimiento físico.

Pruebas en reposo permitieron obtener valores de frecuencia cardíaca y temperatura con un porcentaje de error de medición del 0.35 % y 0.46 % respectivamente. Para pruebas en sesión de entrenamiento en una pista atlética el error de medición de la frecuencia cardíaca es de 0.39% y en cambio para valores de temperatura el error de medición está en un porcentaje de 1.61%; esto se debe a que las condiciones de medición del sensor de temperatura cambian.

## PRESENTACIÓN

La implementación del presente proyecto es una contribución tecnológica al deporte en el Ecuador, destinado como un sistema de ayuda y una herramienta para el entrenador de atletas, facilitando la toma de parámetros como la Temperatura y Frecuencia Cardíaca de un deportista, lo cual permitirá evaluar el rendimiento físico de la persona de acuerdo a una sesión de entrenamiento.

En base a lo antes mencionado, el presente documento consta de cinco partes principales, cuyo contenido se describe brevemente a continuación:

En la primera parte, se realiza una breve descripción de los parámetros, normas y reglamentos que se toman en cuenta para el entrenamiento y competencias atléticas que se deben analizar para el sistema de entrenamiento a realizar. También se describen los dispositivos que se utiliza en el sistema para entrenamientos.

En la segunda se detalla el diseño del sistema para entrenamiento en función de las especificaciones propuestas para el sistema a ser implementado.

La tercera parte presenta el desarrollo del hardware y software para la implementación y construcción del dispositivo.

En la cuarta parte se mostrarán los datos obtenidos en las pruebas de operación del proyecto y un análisis de resultados.

En la última parte se plantean las conclusiones derivadas del proyecto y las recomendaciones para futuros usuarios.

# **CAPÍTULO 1. PARÁMETROS Y DISPOSITIVOS A UTILIZAR EN EL SISTEMA PARA ENTRENAMIENTO.**

Este proyecto tiene como objetivo el diseñar y construir un sistema de monitoreo para entrenamientos en pista que permita llevar un control de la actividad física a desarrollarse por el atleta. El mismo sistema puede ser utilizado, para llevar un control del proceso de adaptación en cada sesión de entrenamiento entre la interacción de un atleta y de un entrenador, para de esta manera contribuir a la evaluación del plan de entrenamiento que se realiza.

## **1.1 INTRODUCCIÓN**

En este capítulo se hará una breve descripción de las normas y reglamentos de medición que intervienen en los parámetros de Tiempo, Frecuencia Cardíaca y Temperatura; que son los considerados para el diseño del equipo de monitoreo en el entrenamiento de un atleta. Cada uno de los parámetros que se toman en consideración depende mucho del tipo de entrenamiento y de lo que el entrenador desee conseguir.

Tomando en cuenta estos parámetros, el esfuerzo al que están ligados y la exactitud que se necesite, se hace una descripción del esquema del equipo para su monitoreo, así como una breve descripción de los dispositivos tomados en cuenta en el diseño.

## **1.2 EL ENTRENAMIENTO [1]**

### **1.2.1 DEFINICIÓN**

La práctica de un esfuerzo físico conlleva una serie de respuestas por parte del organismo. Si éste solo se realiza en un determinado momento, de manera aislada, como por ejemplo jugar un día un partido de tenis, nadar, jugar un partido de fútbol, realizar una caminata, y esto no se repite en dos o tres semanas, estamos frente a una "respuesta fisiológica aguda".

En este tipo de respuesta el organismo intenta suplir las necesidades temporales de este esfuerzo físico, pero éste no deja ninguna huella y solamente ha servido para pasar un rato aislado de deporte.

En resumen, se realiza una demanda aguda y se obtiene una respuesta también aguda. En cambio, cuando este esfuerzo físico se repite con una determinada intensidad, duración y periodicidad en el tiempo, se está frente a lo que se llama un "entrenamiento". Sobre las posibilidades de que estas tres variables difieran en mayor o menor grado va a influir, la edad del individuo y la condición física de la que se parte al principio de todo el proceso.

Además, sobre la peculiaridad de cada uno en responder de una manera u otra a este entrenamiento van a influir factores genéticos que van a condicionar que la respuesta sea diferente en unas personas u otras. La respuesta que se obtiene frente a este estímulo del ejercicio físico continuado es la de una adaptación crónica del organismo en general.

La adaptación es poder realizar un mismo esfuerzo con menor gasto energético cada vez, o por transición, poder realizar una prestación física mayor, con el menor gasto energético, esto, en consecuencia, es poder realizar cada vez esfuerzos de mayor calidad y cantidad.

Los efectos biológicos que van a tener lugar como consecuencia de esta adaptación crónica se van a manifestar básicamente sobre el sistema músculo-esquelético (hipertrofia del músculo, aumento de la fuerza, velocidad, elasticidad mayor de los ligamentos, etc. ), sobre la sangre (aumento del hematocrito, hematíes etc. ), sobre la composición corporal (reducción de la grasa corporal, aumento del músculo o masa magra etc..) pero sobre todo, los cambios más importantes son los que tienen lugar sobre el sistema de aporte de oxígeno que está constituido principalmente por el aparato respiratorio y por el sistema circulatorio (corazón, arterias, arteriolas, capilares, venas).

### **1.2.2 ESTRUCTURA DEL PLAN DE ENTRENAMIENTO [2]**

Un plan de entrenamiento es una consecución de trabajos que tienen un orden determinado y están puestos de una manera para poder permitirle a las células musculares, hormonas, nutrientes, minerales, electrolitos, etc. una reposición adecuada para volver a realizar otro esfuerzo, así también como a todos los tejidos que se ven implicados en la realización de dicho esfuerzo.

La rapidez y el grado de reposición con que esto se lleve a cabo es lo que más arriba se denomina adaptación. Por eso no se debe alterar el orden de los entrenamientos, ya que esto puede perjudicar a toda la estructura del plan de entrenamiento y, lo que es peor, la posibilidad de lesionarse.

Esta estructura está considerada en cinco partes las cuales tienen diferentes composiciones, según el objetivo que se persiga, y que conforman el entrenamiento como tal; estas partes son:

- a. Preparación para el entrenamiento
- b. La entrada en calor
- c. Entradas en calor orientativas
- d. Sesión o trabajo principal
- e. Vuelta a la calma o enfriamiento.

A continuación se hace una breve descripción de cada una de estas partes citadas del plan de entrenamiento, con las cuales se permitirá conocer cómo se ven afectados los parámetros que se consideran en el equipo de monitoreo.

### **1.2.3 PREPARACIÓN PARA EL ENTRENAMIENTO [2]**

Para comenzar una sesión de entrenamiento debe haber una preparación previa en donde se debe considerar:

- a. No se debe cargar excesivamente con alimentos al aparato digestivo, por lo menos dos horas antes del entrenamiento.

- b. Es conveniente ingerir una o dos frutas de fácil digestión 30 minutos antes de entrenar. Esto ayudará a sobrellevar la sesión del entrenamiento.
- c. Debe haber una hidratación equilibrada previa al entrenamiento. Una hidratación equilibrada hace referencia a que no se deben ingerir grandes cantidades de líquido en momentos previos al entrenamiento.
- d. Es necesario recordar que se está realizando un deporte de rendimiento, si antes del entrenamiento existe un cansancio excesivo y hay falta de ganas para entrenar, es preferible hacer una sesión de carácter regenerativo, es decir, un trabajo de relajamiento muscular y no "parar".

#### 1.2.3.1 La entrada en calor [2]

Es la primera parte de la sesión de entrenamiento y está dividida en dos partes:

- a. **General:** Consiste en la realización de estiramientos y movilidad articular para elevar la temperatura de las articulaciones y preparar a los músculos para un esfuerzo superior al del reposo.
- b. **Específica:** Tiene la función de preparar al organismo en su conjunto para la realización del esfuerzo que supone el trabajo central de la sesión de entrenamiento. Como se conoce, la parte central del entrenamiento puede variar tanto en volumen como en intensidad, así que la entrada en calor específica debe estar acorde a la parte central del entrenamiento.

La parte específica de la entrada en calor tiene como principal consecuencia el aumento de la temperatura corporal. Esto hace que el corazón eleve sus latidos por encima de la FC basal, que haya más irrigación en los músculos, que aumente el intercambio gaseoso en los pulmones, etc.

#### 1.2.3.2 Entradas en calor orientativas [2]

Este tipo de entradas en calor se las realiza orientadas a la disciplina deportiva o a la sesión de ejercicios físicos a realizar, por este motivo toman el nombre de entradas en calor orientativas.

Sirven tanto para un deporte de una o de más disciplinas. Son solo orientativas y deben realizarse de acuerdo al trabajo central de la sesión, según el siguiente criterio:

- Elongación muscular más movilidad articular: siempre
- Trabajos continuos largos y de baja intensidad: entrada en calor corta.
- Trabajos en intervalos o continuos intensivos (más del 90% del umbral anaeróbico): entrada en calor larga (según la disciplina).

En el Tabla 1.1 se muestra la forma en que se orienta el trabajo según un objetivo.

DISCIPLINA	TIPO DE SESION (ejemplos)	OBJETIVO DE LA ENTRADA EN CALOR	DURACIÓN
NATACIÓN	Continuo Extensivo	Relajar	5 minutos (100 a 200 metros)
	Intensidad Media.	Relajar	10 minutos (200 a 300 metros)
	Intensidad Elevada	Regenerar (remoción lactato)	15 minutos (400 a 600 metros)
CICLISMO	Continuo Extensivo	Relajar	15 minutos a 20 minutos
	Intensidad Media	Relajar	20 minutos a 30 minutos
	Contrareloj	Regenerar (remoción lactato)	45 minutos a 1 hora
CARRERA	Continuo Extensivo	Relajar	5 minutos
	Intervalos	Regenerar (remoción lactato)	15 minutos a 20 minutos
	Continuo Intensivo	Relajar, remover lactato si hay presente	10 minutos a 15 minutos

**Tabla 1. 1** Ejemplo de entradas en calor orientativas. [2]

### 1.2.3.3 Sesión o trabajo principal [2]

Luego de la entrada en calor continúa la segunda parte de la sesión de entrenamiento o núcleo de la sesión en un periodo de tiempo de 15 a 20 minutos, que es donde se realiza el trabajo principal y en la cual los parámetros que se consideran son un indicativo para saber si se cumplen o no con los objetivos.

Un parámetro importante es la frecuencia cardiaca basal (por la mañana), ya que si es más de un 20% superior que la habitual, ese día el entrenamiento deberá

cambiar de objetivo, para ser reemplazado por una sesión de recuperación. Cuando la sesión principal del entrenamiento, tiene una orientación relacionada con la calidad (ritmo de carrera, tolerancia láctica, etc.), si los tiempos propuestos no pueden ser mantenidos, es conveniente finalizar la sesión y regenerar. También es aplicable en función del método de entrenamiento: a mayor intensidad, más cortos los intervalos y a menor intensidad, mayores los intervalos. No se pueden esperar resultados diferentes cuando siempre se hace lo mismo, por lo que tener un registro de los parámetros considerados durante una sesión de ejercicio ayuda a saber si el trabajo desarrollado contribuye a mejorar.

#### **1.2.3.4 Vuelta a la calma o enfriamiento [2]**

Luego del trabajo principal – o núcleo del entrenamiento - se debe realizar la tercera parte, vuelta a la calma o enfriamiento, el cual tiene diferentes funciones según haya sido el entrenamiento principal. La función y concepto principal, es que la vuelta a la calma es el comienzo del proceso de recuperación de la sesión.

No necesariamente una vuelta a la calma es regenerativa., salvo cuando la intensidad del entrenamiento principal haya sido elevada. Hay dos objetivos bien importantes luego de la sesión principal: relajar y recuperar. Los dos tienen métodos de trabajo diferente. Los más representativos son:

- Objetivo de relajar: entrenar la flexibilidad
- Objetivo de recuperar: entrenar la fuerza

En la Tabla 1.2 presenta los objetivos de la vuelta a la calma según la sesión realizada.

Considerando las partes que conforman el plan de entrenamiento y las actividades que se realizan según sea el objetivo, a continuación se describe los parámetros de Tiempo, Frecuencia Cardíaca y Temperatura Corporal que se toman en cuenta y la relación e influencia que tienen para saber si se están cumpliendo los objetivos que se desean alcanzar en el entrenamiento, así como una descripción de las normas y reglamentos con las que son tomados dichos parámetros.

DISCIPLINA	TIPO DE SESION (ejemplos)	OBJETIVO DE LA VUELTA A LA CALMA	DURACIÓN
NATACIÓN	Continuo Extensivo	Relajar	5 minutos (100 a 200 metros)
	Intensidad Media.	Relajar	10 minutos (200 a 300 metros)
	Intensidad Elevada	Regenerar (remoción lactato)	15 minutos (400 a 600 metros)
CICLISMO	Continuo Extensivo	Relajar	15 minutos a 20 minutos
	Intensidad Media	Relajar	20 minutos a 30 minutos
	Contrareloj	Regenerar (remoción lactato)	45 minutos a 1 hora
CARRERA	Continuo Extensivo	Relajar	5 minutos
	Intervalos	Regenerar (remoción lactato)	15 minutos a 20 minutos
	Continuo Intensivo	Relajar, remover lactato si hay presente	10 minutos a 15 minutos

**Tabla 1. 2** Ejemplo de la vuelta a la calma. [2]

## 1.3 PARÁMETROS DEL ENTRENAMIENTO

### 1.3.1 TIEMPO [3]

Es una de las magnitudes más utilizadas para valorar el volumen de entrenamiento, sobre todo cuando se trata de deportes de equipo y en deportes individuales cuando se realiza un trabajo de alta intensidad. La utilización de Cronómetros con multitud de memorias y funciones, posibilita llevar un control del tiempo de forma instantánea y también a posterior. La mayor parte de ciclocomputadores, pulsómetros incorporan también esta función, incluso de forma acumulada.

La relación entre el tiempo de entrenamiento y la obtención de rendimiento físico también es importante para algunos autores, como en el caso de Bompa quien establece una relación entre el volumen de entrenamiento anual (en horas) para obtener un rendimiento deportivo aceptable en competiciones de diferente nivel:

- Mejores del Mundo: 1000 horas de entrenamiento anual.
- Competiciones Internacionales: 800 horas de entrenamiento anual.
- Competiciones Nacionales: 600 horas de entrenamiento anual.
- Competiciones Regionales: 400 horas de entrenamiento anual.

### **1.3.1.1 El tiempo y como se relaciona en los entrenamientos y competencias [26]**

En las pruebas atléticas el cálculo del tiempo invertido es una operación que debe realizarse con gran ajuste y exactitud, ya que los registros de los corredores suelen ser muy similares y el más mínimo error puede variar los resultados finales. A partir de una buena medición del tiempo es muy sencillo poder analizar la velocidad media que cada corredor ha empleado durante el recorrido.

Tan sólo hemos de emplear una fórmula matemática, que consiste en dividir espacio y tiempo. En las carreras lo esencial es correr una distancia en el menor tiempo posible.

El atleta pone a prueba sus capacidades físicas y la compara con la de los demás. Estas capacidades son:

- ✓ Potencia para impulsarse y tener amplitud de calzada.
- ✓ Equilibrio para ir en línea recta.
- ✓ Resistencia aeróbica o anaeróbica, según sea la distancia.
- ✓ Sentido estratégico para conducir la carrera.

La velocidad de un atleta es el resultado de dividir la distancia recorrida entre el tiempo total. La velocidad está determinada por:

- ✓ La potencia de las piernas
- ✓ La coordinación y el equilibrio
- ✓ El estilo de correr y el aprovechamiento de la inercia en las curvas

Todas las pistas de atletismo, circuitos, y zonas dedicadas a la práctica del atletismo deben someterse a un severo control que garantice el cumplimiento de todas las normas dictadas por la IAAF (Federación Internacional de Atletismo Amateur).

### **1.3.1.2 Consideraciones sobre el reglamento de la toma de tiempo [4]**

En el reglamento de la IAAF CAPÍTULO 5 (REGLAMENTO TÉCNICO) (ARTÍCULOS 100 AL 263) se toman en consideración los siguientes reglamentos para el diseño.

- Los participantes serán calificados en el orden en que cualquier parte de su tronco (y no la cabeza, el cuello, los brazos, las piernas, las manos o los pies) alcance el plano vertical del borde más próximo de la línea de llegada.
- En todas las carreras en pista, cronometradas manualmente, los tiempos deben leerse a la 1/10 de segundo superior. Los tiempos de carreras desarrolladas parcial o totalmente fuera del Estadio deben convertirse al próximo segundo entero mayor. Por ejemplo: 2h 09.44.3 en Maratón se convertirán en 2h 09.45. Si la aguja del reloj se para entre dos líneas indicadoras de tiempo, se considerará tiempo oficial el inmediato superior. Si se utiliza un cronómetro de 1/100 de segundo, o uno electrónico operado manualmente, con lectura digital, todos los tiempos que no terminen en cero en el segundo decimal se convertirán a la próxima 1/10 de segundo mayor. Por ejemplo: 10.11 de segundo se convertirá en 10.2.
- El tiempo se medirá desde el fogonazo de una pistola o aparato de salidas aprobado hasta el momento en que cualquier parte del cuerpo del atleta (es decir el tronco, excluyendo la cabeza, el cuello, brazos, piernas, manos o pies) alcance el plano perpendicular del borde más próximo de la línea de llegada.

### **1.3.2 FRECUENCIA CARDIACA**

#### **1.3.2.1 Definición [5]**

La frecuencia cardiaca es la magnitud decisiva para la adaptación al rendimiento, puesto que las cavidades del corazón por sus peculiaridades anatómicas disponen de un estrecho margen para aumentar el volumen de sangre por pulsación. La elevación del volumen sistólico como reacción a la adaptación a un trabajo sólo tiene lugar al inicio de la carga de trabajo después permanece estable aunque se aumente la intensidad de la carga.

El mayor aumento del volumen sistólico se produce inmediatamente después de finalizar una carga como consecuencia de que la frecuencia cardiaca disminuye muy rápidamente, mientras que la corriente venosa de retorno es aún muy elevada. El aumento del volumen sistólico una vez pasado el valor de carga máximo puede mantenerse hasta tres minutos, dependiendo de la intensidad de la carga. En

contraposición, la frecuencia cardiaca aumenta de forma lineal y en paralelo con la intensidad de la carga.

La frecuencia cardiaca es de suma importancia ya que es uno de los factores que influyen en el gasto cardíaco, además de ser un parámetro de fácil obtención que proporciona información de lo que ocurre a nivel cardiovascular. Un punto importante a tener en cuenta es que la frecuencia cardiaca en reposo se ve afectada por los cambios posturales.

Unos valores bajos de la frecuencia cardiaca pueden ser interpretados de diferentes maneras: una cardiopatía (corazón débil), o de un alto valor Volumen Sistólico (VS) que podría ser consecuencia a características congénitas o de entrenamiento.

Las mujeres tienen una frecuencia cardiaca más alta que los hombres del orden de 5 a 10 pulsaciones más elevadas por minutos: el motivo se cree que se debe a que la capacidad para transportar oxígeno es menor ya que los niveles de hemoglobina son algo inferiores que en los hombres, lo que implica un mayor gasto cardiaco.

La relación de la frecuencia cardiaca con la intensidad del ejercicio se puede establecer como consecuencia de la relación que existe entre el consumo de oxígeno y el gasto cardíaco. Para no caer en errores importantes hay que tener en cuenta los siguientes puntos:

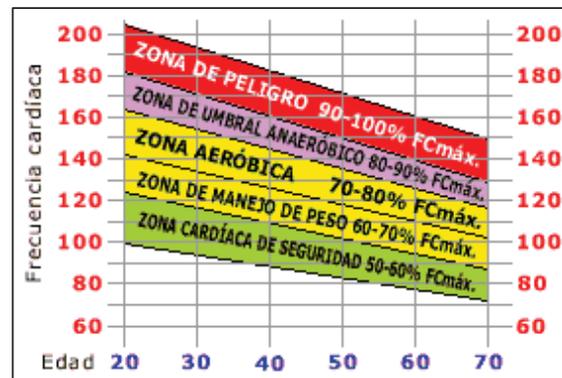
Si se compara atletas entrenados y no entrenados para una misma frecuencia cardiaca los entrenados tendrán mayor capacidad de trabajo, así como diferenciar los sexos, con el entrenamiento se produce una disminución de la frecuencia cardiaca para una misma intensidad. Para finalizar, con la edad va disminuyendo la frecuencia cardiaca para una misma intensidad.

### **1.3.2.2 Relación de la frecuencia cardiaca en un entrenamiento [27]**

Para realizar un plan de entrenamiento o para realizar esfuerzos de manera controlada se ve la necesidad de conocer al corazón. Con el conocimiento del estado físico a través del corazón se podrá realizar ejercicios y entrenamientos de manera óptima sin peligro para la integridad. Para ello se debe conocer la frecuencia cardiaca en reposo (FCR) y la frecuencia cardiaca máxima (FCmáx.).

A la hora de comenzar un plan de entrenamiento debe asegurarse de que se está trabajando al nivel de intensidad correcto con respecto al nivel de condición física y de la edad.

El Figura 1.1 muestra una clasificación de la intensidad aproximada con que se debe ejercitar para no correr riesgos de acuerdo a la Frecuencia Cardíaca Máxima.



**Figura 1. 1** Clasificación de la Intensidad, según la Edad y la Frecuencia Cardíaca Máxima. [27]

### 1.3.2.3 Frecuencia cardíaca en reposo o en descanso [27]

Es la frecuencia cardíaca que se posee en el momento de menos actividad física; es decir, en reposo. Por lo tanto, para calcular la FCR (frecuencia cardíaca en reposo o en descanso), se tendrá que tomarse el pulso, nada más despertar por la mañana cada día durante una semana y hacer la media. También se la puede tomar relajándose 5 minutos antes de la medición.

$$FCR = \frac{\sum FCR_{SEMANA}}{7}$$

Hombres				
Edad	Mal	Normal	Bien	Excelente
20-29	86+	70-84	62-68	60 o menos
30-39	86+	72-84	64-70	62 o menos
40-49	90+	74-88	66-72	64 o menos
50+	90+	76-88	68-74	66 o menos

**Tabla 1. 3** Valores medios de la Frecuencia Cardíaca en Reposo según la edad (Hombres). [27]

Mujeres				
Edad	Mal	Normal	Bien	Excelente
20-29	96+	78-94	72-76	70 o menos
30-39	98+	80-96	72-78	70 o menos
40-49	100+	80-98	74-78	72 o menos
50+	104+	84-102	76-82	74 o menos

**Tabla 1. 4** Valores medios de la Frecuencia Cardiaca en Reposo según la edad (Mujeres) [27]

Si los latidos son bajos, generalmente significa un corazón con una buena capacidad aeróbica. Si, por el contrario, los latidos son más altos de lo que normalmente se tiene, puede indicar que aún no se tiene una capacidad aeróbica adecuada, o bien cansancio, o alguna enfermedad que se esté presentando en el cuerpo indicando que algo no está funcionando bien.

La frecuencia cardiaca en reposo depende de los hábitos de vida y está influenciada por el entrenamiento, la recuperación de ejercicios del día anterior, el sueño, el nivel de estrés mental y los hábitos alimenticios.

#### 1.3.2.4 Frecuencia cardiaca máxima [6]

Es la frecuencia máxima (teórica) que se puede alcanzar en un ejercicio de esfuerzo sin poner en riesgo la salud, siempre y cuando se encuentre en óptima condición física. Al alcanzar la frecuencia cardiaca máxima, teóricamente, se ha alcanzado la máxima capacidad de trabajo. La frecuencia cardiaca máxima es una herramienta para determinar la intensidad de los entrenamientos, lo cual se verá en el siguiente apartado.

Para calcular cuál es el límite máximo cardíaco, básicamente hay dos formas de realizarlo: la primera es a través de la realización de un test médico, (en bicicleta o cinta de correr) realizado por un cardiólogo o un médico del deporte. Y la segunda forma, más conocida pero no tan exacta, es a través de la llamada "fórmula por edad".

Existen otras fórmulas, algunas más detalladas que otras, para el cálculo ésta es bastante válida y segura.

$FC_{\text{máx.}} = 220 - \text{edad}$  (para hombres)

$FC_{\text{máx.}} = 226 - \text{edad}$  (para mujeres)

En principio este parámetro es el límite que no se debe pasar y se podrá referir a la Figura 1.1 para realizar un plan de acuerdo al objetivo; significa el 100% de esfuerzo al que el corazón deberá latir en un minuto.

Para el cálculo de la frecuencia cardiaca se hace referencia a dos métodos:

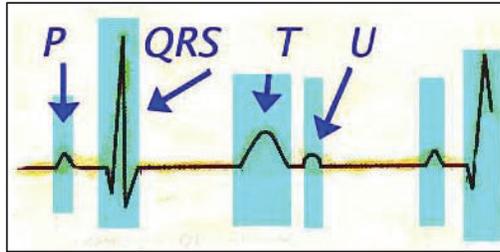
- ✓ Mediante las ondas que presenta un electrocardiograma, a través de las zonas en las cuales se las puede tomar, llamadas derivaciones electrocardiográficas.
- ✓ Mediante un contador cardiaco, el cual realiza la toma de medidas del paso del flujo sanguíneo en determinadas zonas del cuerpo humano.

A continuación se describen los métodos mencionados:

#### *1.3.2.4.1 Electrocardiograma ECG, ondas y segmentos*

El electrocardiograma (ECG), es una representación gráfica de la actividad eléctrica del corazón que ofrece información acerca del estado del músculo cardiaco. Esta representación consiste en una línea de base, varias deflexiones y ondas. El origen se encuentra en las células del músculo cardiaco, las cuales pueden ser excitadas eléctricamente, produciéndose un transporte de iones a través de su membrana, lo cual induce un potencial eléctrico variable en el interior y en el exterior [7].

El electrocardiograma (ECG) es llevado a cabo a partir de unos electrodos (conductores eléctricos) aplicados en la superficie de la piel desde distintos puntos (derivaciones), habitualmente en ambos brazos y piernas y sobre el tórax. Dentro del corazón minúsculos impulsos eléctricos controlan la función del corazón de expansión y contracción para bombear la sangre al cuerpo. Los electrodos colocados sobre la piel pueden detectar estos impulsos y transmitirlos al electrocardiógrafo por medio de cables. Las ondas del ECG se denominan P, Q, R, S, T, U y tienen voltaje positivo o negativo. Una forma de onda de señal cardiaca típica es como la mostrada en la Figura 1.2.



**Figura 1. 2** Señal ECG. [8]

La primera deflexión ascendente, P, es debido a la contracción de los atrios y se conoce como el complejo atrial. Las otras desviaciones, Q, R, S y T, son debidas a la acción de los ventrículos y se conoce como los complejos ventriculares.

La señal electrocardiográfica (ECG) tiene como parámetros relevantes:

- Rango de medida de amplitud de 500uV a 5mV, combinada con una componente DC de  $\pm 300\text{mV}$
- Frecuencia con componentes relevantes entre 0.5 Hz y 50 Hz para monitoreo
- Frecuencia con componentes relevantes entre 0.05 Hz y 150 Hz para diagnóstico.

Dado que la señal de la Figura 1.2 es relativamente periódica cada ciclo tiene la misma estructura. Se puede tomar cualquier punto de los señalados en la figura para calcular el ritmo cardiaco [8].

#### 1.3.2.4.2 Contador de Frecuencia Cardiaca

El contador de frecuencia cardiaca es un dispositivo capaz de cuantificar la cantidad de pulsos o latidos cardiacos que tienen una persona por minuto; para ello realiza la toma de datos mediante un sensor que contabiliza los pulsos cardiacos a través del paso del flujo sanguíneo en ciertas zonas del cuerpo humano ricas en capilares, esto lo hace mediante un dispositivo emisor – receptor de luz y esta medida convertida a señal eléctrica pasa por un sistema de amplificación y filtros, para obtener una señal digital deseada según el interés de uso del contador de frecuencia cardiaca.

### **1.3.3 TEMPERATURA CORPORAL**

La temperatura normal del organismo es la resultante de un equilibrio entre la producción de calor y su eliminación. El ser humano es homeotermo, es decir, un organismo de temperatura constante, cualquiera sea el ambiente, pero en mucho menos grado en la primera infancia y en la extrema vejez, por no haberse desarrollado en la primera y haberse perdido en la segunda, el complejo mecanismo regulador [9].

#### **1.3.3.1 Valores de medición**

La temperatura corporal de cada persona es diferente, estando la temperatura normal comprendida entre los 36,3 ° C y los 37,4 ° C. El incremento de temperatura muscular mediante el trabajo de entrenamiento, provoca un incremento de la velocidad corporal.

#### **1.3.3.2 Selección del lugar de medición**

Los puntos de medición ideales son cavidades corporales próximas a grandes vasos sanguíneos. Los puntos de medición más conocidos son la boca (oral), el recto (rectal), la axila (axilar) y el oído interno.

Debido a que las mediciones oral, rectal y oído interno son de carácter invasivo no presentan las facilidades para realizarlas en una persona que está en un entrenamiento, por lo que se realizara la toma de forma axilar.

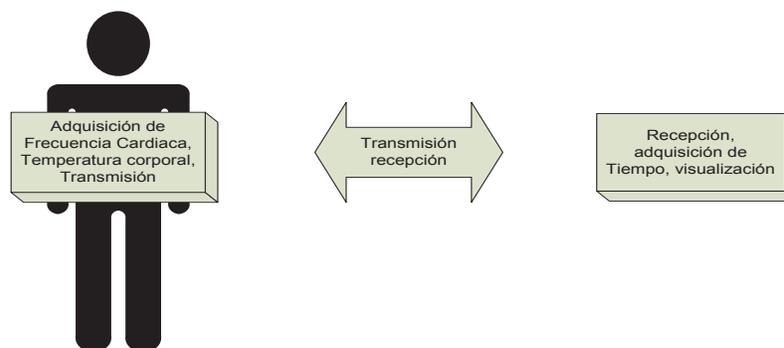
Se mide en la cavidad axilar. La cavidad axilar es seguramente el lugar, donde se empezó a medir la temperatura corporal por primera vez. Los primeros análisis de temperatura fueron realizados por el médico alemán Wunderlich, hace un siglo, y lo hizo exclusivamente en la cavidad axilar. Al publicarse sus investigaciones, que indujeron en todo el mundo la introducción de la medición diaria de la temperatura en los hospitales, la medición axilar se convirtió en el lugar de medición estándar para la temperatura corporal [10].

## 1.4 DESCRIPCIÓN DEL SISTEMA

El objetivo de este proyecto es el desarrollo e implementación de un equipo para el monitoreo de parámetros de importancia dentro de un entrenamiento, por lo que consta de dos módulos, uno que ira colocado en el atleta y otro que será manipulado por el entrenador evidenciándose la necesidad de una transmisión inalámbrica entre dichas partes.

El módulo colocado en el atleta se encargara de la adquisición de la Frecuencia Cardiaca y la Temperatura Corporal, para su posterior transmisión al módulo del entrenador.

El modulo manipulado por el entrenador será el encargado de la recepción de los parámetros transmitidos por el módulo del atleta y su almacenamiento. Además se encargara de la adquisición de los tiempos que guardarán una relación con los otros dos parámetros, pudiendo ser estos visualizados para poder saber si el plan de entrenamiento está cumpliendo con los objetivos trazados, dando una información en tiempo real de la respuesta del atleta a cada trabajo establecido.



**Figura 1. 3** Diagrama de Módulos.

A continuación se describen el funcionamiento básico de los sub-módulos y elementos que conformaran cada módulo.

## **1.4.1 MONITOR DE FRECUENCIA CARDIACA**

### **1.4.1.1 Descripción general de dispositivos**

Como se ha visto en el subcapítulo 1.3.2.4.1 la señal de Electrocardiograma es una señal muy débil (alrededor de 1 mV). Además de ser una señal de muy baja amplitud, hay algunos factores que pueden degradar esta señal antes de poder amplificarla y procesarla. Algunos de estos factores son por ejemplo el ruido y las interferencias. Estos factores adquieren especial importancia en la primera etapa de medida del sistema de ECG, por ejemplo los electrodos. Por eso, estos factores se deben eliminar en mayor medida que sea posible. Los cables son otro factor a tener en cuenta, ya que actúan como antenas y pueden introducir interferencias en el circuito. La interferencia más común es la de 50 Hz. Para solucionar este problema se debe utilizar cables apantallados.

En la elección de un amplificador operacional se debe tener en cuenta que debe tener un CMRR elevado (>70 dB) para poder eliminar la señal que se acople en modo común y solo amplificar la señal en modo diferencial.

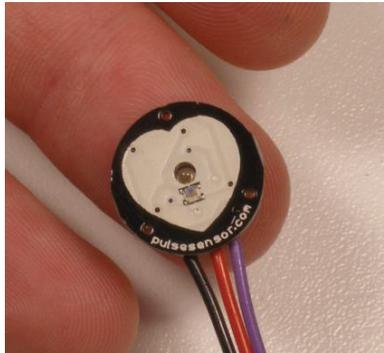
En relación al subcapítulo 1.3.2.4.2 el contador de pulsos no depende específicamente de una señal eléctrica del corazón que deba ser detectada, más bien se basa en detección de pulsos que se puede realizar a partir del paso de flujo sanguíneo en determinada zona del cuerpo humano por medio de un dispositivo fotoeléctrico fijado con una pinza. De acuerdo a esto, el contador de pulsos no presentaría los factores de ruido inducidos que se presentan en el ECG

### **1.4.1.2 Selección del dispositivo**

Como se ha detallado en la descripción de los dispositivos que permiten detectar la frecuencia cardiaca, el dispositivo que sirve para el objetivo del presente proyecto es el Contador de Pulsos, por motivos de que solo se necesita el conteo del pulsos del corazón sin necesidad de analizar a detalle la señal eléctrica emitida por el corazón y además se evita en mayor medida los factores de ruido e interferencias que podrían presentarse con el dispositivo de ECG.

En el mercado se ha encontrado un sensor fotoeléctrico para la frecuencia cardiaca, que lleva por nombre PulseSensor [11], el mismo que es un hardware abierto

desarrollado por Joel Murphy y Yuri Gitman. Este sensor es distribuido por sus desarrolladores y por la empresa Sparkfun con el número de parte SEN11574. [12]



**Figura 1. 4** Sensor de Pulsos. [11]

## 1.4.2 TRANSDUCTORES DE TEMPERATURA [28]

### 1.4.2.1 Conceptos básicos

Los transductores eléctricos de temperatura utilizan diversos fenómenos que son influidos por la temperatura y entre los cuales figuran:

- Variación de resistencia en un conductor (sondas de resistencia).
- Variación de resistencia de un semiconductor (termistores).
- Voltaje creado en la unión de dos metales distintos (termopares).
- Intensidad de la radiación total emitida por el cuerpo (pirómetros de radiación).

Otros fenómenos utilizados en laboratorio (velocidad del sonido en un gas, frecuencia de resonancia de un cristal, etc.). Los metales puros tienen un coeficiente de resistencia de temperatura positivo bastante constante. El coeficiente de resistencia de temperatura, generalmente llamado coeficiente de temperatura es la razón de cambio de resistencia al cambio de temperatura.

Cuando se usa un alambre de metal puro para la medición de temperatura, se le refiere como detector resistivo de temperatura, o RTD (por las siglas en ingles de *resistive temperature detector*). Cuando se usan óxidos metálicos para la medición de temperatura, el dispositivo formado así se llama Termistor. Los termistores tienen coeficientes de temperatura negativos grandes que no son constantes.

Como regla general, los termistores son preferibles cuando la banda de temperaturas esperada es angosta, mientras que los RTD son preferibles cuando la banda de temperatura esperada es amplia. En el siguiente informe se comparan los cuatro tipos más corrientes de transductores de temperatura que se usan en los sistemas de adquisición de datos: detectores de temperatura de resistencia (RTD), termistores, sensores de IC y termopares.

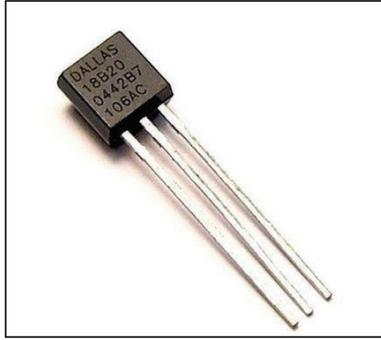
#### 1.4.2.2 Tipos y características

Ningún transductor es el mejor en todas las situaciones de medida, por lo que se tiene que saber cuándo debe utilizarse cada uno de ellos. Como se puede ver, en la Tabla 1.5 se están comparando los cuatro tipos de transductores de temperatura más utilizados, y refleja los factores que deben tenerse en cuenta: las prestaciones, el alcance efectivo, el precio y la comodidad.

	RTD	Termistor	Sensor de IC	Termopar
<b>Ventajas</b>	Más estable. Más preciso. Más lineal que los Termopares.	Alto rendimiento. Rápido. Medida de dos hilos.	El más lineal. El de más alto Rendimiento Económico.	Autoalimentado. Robusto. Económico. Amplia variedad de formas físicas. Amplia gama de temperaturas.
<b>Desventajas</b>	Caro. Lento. Precisa fuente de alimentación. Pequeño cambio de resistencia. Medida de 4 hilos. Autocalentable	No lineal. Rango de Temperaturas ilimitado. Frágil. Precisa fuente de alimentación. Autocalentable.	Limitado a < 250 °C. Precisa fuente de alimentación. Lento. Autocalentable. Configuraciones limitadas.	No lineal. Baja tensión. Precisa referencia. El menos estable. El menos sensible.

**Tabla 1. 5** Tabla de Transductores de Temperatura. [28]

En base a esta tabla se trabajara con un sensor IC ya que presta las mejores características para trabajar con el microcontrolador, y se ha seleccionado el perteneciente a la familia DALLAS DS1820 que tiene un margen de temperaturas de -55 a 125 grados con una resolución y una precisión de medio grado y funciona con el protocolo de comunicación 1wire de Dallas [13].



**Figura 1. 5** Sensor de Temperatura DS1820. [13]

### 1.4.3 MEDICIÓN DEL TIEMPO

Un reloj es un temporizador especial con la tarea de mantener el tiempo de la hora, día, mes, por lo que para medir el tiempo se utilizara un reloj en tiempo real el cual puede ser usado como alarma para diferentes eventos.

#### 1.4.3.1 Reloj en tiempo real [20]

Es un sistema que debe cumplir con alguna restricción temporal, es un completo reloj con alarma, calendario, interrupción periódica programable también posee SRAM permanente.

El *clock*/calendar proporciona segundos, minutos, horas, día, la fecha, el mes, y la información del año. El extremo de la fecha del mes se ajusta automáticamente por meses con menos de 31 días, incluyendo las correcciones por año bisiesto. El reloj funciona en el formato de 24 horas o 12-horas con el indicador de AM/PM.

La dirección y los datos se transfieren en serie vía un bus bidireccional de dos hilos I2C. El Bus I2C diseñado por *Philips*, este sistema de intercambio de información a través de tan solo dos cables permite a circuitos integrados y módulos OEM interactuar entre sí a velocidades relativamente lentas. Emplea comunicación serie, utilizando un conductor para manejar el *timming* (pulsos de reloj) y otro para intercambiar datos.

Este bus se basa en tres señales:

- SDA (*System Data*) por la cual viajan los datos entre los dispositivos.
- SCL (*System Clock*) por la cual transitan los pulsos de reloj que sincronizan el sistema.
- GND (Masa) Interconectada entre todos los dispositivos "enganchados" al bus.

Las líneas SDA y SCL son del tipo drenador abierto, similares a las de colector abierto pero asociadas a un transistor de efecto de campo (ó FET). Se deben poner en estado alto (conectar a la alimentación por medio de resistores *Pull-Up*) para construir una estructura de bus tal que se permita conectar en paralelo múltiples entradas y salidas.

El reloj en tiempo real seleccionado pertenece a la familia DALLAS DS1307 que presenta las características ya descritas.

#### **1.4.4 DISPOSITIVOS DE TRANSMISIÓN Y RECEPCIÓN**

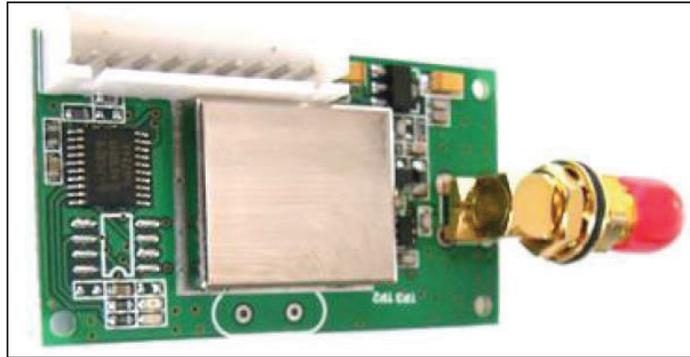
Con la utilización de las bandas ISM 433 y 868MHz que han estado disponibles durante algunos años, ahora los ingenieros están mirando hacia la banda 2,4GHz para las actuales y nuevas aplicaciones. La banda de 2,4GHz tiene el atractivo añadido de estar disponible en todo el mundo no como las bandas ISM a 433/868/915 MHz etc. Esto permite costes de desarrollo singular, opuesto a tener varias plataformas hardware para mantener y reducir los problemas logísticos que pueden aparecer cuando existen regulaciones geográficas diferentes para otras bandas ISM. La banda de 2,4GHz ha sido vista como un medio de comunicación local con un rango de distancias amplio.

##### **1.4.4.1 Descripción del módulo transceptor seleccionado de RF.**

El *transceiver* de RF seleccionado para el proyecto es YS-1020UB, el cual es mostrado en la Figura 1.6.

El *transceiver* YS-1020UB es un modulador de RF de baja potencia diseñado por profesionales UART de sistemas de transmisión de datos de corto alcance. Es un circuito integrado de *Texas Instruments chipcon* CC1020 RF, trabaja en frecuencias ISM. Integra transmisión y recepción en modo *half duplex*. El módulo puede ser conectado directamente a procesadores monolíticos, PC, dispositivos

RS-485 y otros componentes UART con RS-232, RS-485 y nivel UART / TTL de interferencia en puerto. Posee un amplio rango de temperatura de trabajo y puede ser utilizado en aplicaciones industriales, tanto en ambientes *indoor* como en *outdoor*.



**Figura 1. 6** YS 1020UB *Data Transceiver*. [14]

**Características:** [14]

- Frecuencia portadora: 433/450/868/915 MHz u otras opciones ISM, todas de libre licencia.
- Interface: RS-232 / RS-485 / TTL
- Multi-canales: 8 canales
- Velocidad de transmisión en aire: 1200 / 2400 / 4800 / 9600 / 19200 / 384000 bps.
- Transmisión transparente de la información, apto para cualquier estándar o protocolo.
- Formato Interface: 8N1 / 8E1 / 801, definida por el usuario.
- Modulación GFSK (*Gaussian Frequency Shift Keying*)
- Altas características anti - interferencia y bajo BER
- *Half Duplex*: Integración de transmisor y receptor, 10 ms, cambio automático para la recepción y envío.
- Bajo Consumo de Potencia y función *Sleep*.
- Rango de Temperatura: -35°C a + 75°C
- Trabajo en humedad: 10% al 90% de humedad relativa sin condensación.
- Impedancia: 50 Ohmios (SMA puerto de antena, teniendo muchas opciones de antenas)

- Cumple con la norma EN 300 220 y ARIB STD- T67

#### **1.4.5 MICROCONTROLADOR**

Un microcontrolador es un dispositivo electrónico capaz de llevar a cabo procesos lógicos. Estos procesos o acciones son programados en lenguaje ensamblador por el usuario, y son introducidos en este a través de un programador. Desde la invención del circuito integrado, el desarrollo constante de la electrónica digital ha dado lugar a dispositivos cada vez más complejos. Entre ellos los microprocesadores y los microcontroladores, los cuales son básicos en las carreras de ingeniería electrónica.

Un microcontrolador es un chip en cuyo interior se encuentra una CPU, Memoria, Reloj, Puertos de Comunicación y Módulos Periféricos de E/S. Cada uno de estos bloques internos, cumple una función específica y permite al diseñador un mejor control de los procesos del sistema.

##### **1.4.5.1 AVR ATMEL risc en 8 bit [21]**

Atmel ha llevado la filosofía de diseño RISC a los microprocesadores de 8 BIT. Adoptada AVR. Esta nueva arquitectura proporciona todos los beneficios habituales del RISC: Tasa de Reloj más rápida, mejor desempeño, y una optimización más eficiente en el compilador.

AVR compite con varias familias de microprocesadores bien establecidas en el mercado, tales como 8051, 6805 y 68HC11. A la competencia se suma la familia PIC de Microchip, que con un diseño más moderno se ha extendido rápidamente en los últimos años.

AVR es el primer diseño de CPU de ATMEL, una compañía más conocida por sus productos de memoria "flash" y EEPROM. La firma también produce y vende varios subproductos de la popular familia 8051 con la diferencia que estos están basados en "flash". La CUP AVR se asemeja a la mayoría de los microprocesadores RISC, pero con registros más pequeños

El núcleo exhibe 32 registros iguales de 8 BIT, y cualquiera de ellos pueden contener direcciones o datos. Como los punteros de dirección de 8 BIT resultan

medianamente inútiles aun para los dispositivos de 8 BIT, los últimos tres registros se pueden usar de a pares, como punteros de dirección. Apodados X, Y, y Z, estos tres meta-registros se pueden para cualquier operación de carga o almacenamiento.

A elección del programador, los punteros pueden ser post o preincrementados. Además, es posible agregar un desplazamiento de 6 BIT a los contenidos del puntero, esta opción resulta útil para direccional los elementos de un arreglo.

Este modo no es posible para el puntero X; ese código de operación (opcode) está reservado para la instrucción LDI (*Load immediate constant*).

A veces resulta difícil entusiasmarse acerca de los microprocesadores de 8 BIT; sin embargo el diseño ARV de ATMEL difiere de los demás de su clase tal como lo hizo la primera máquina RISC comparada con los grandes sistemas, hace más de una década. Con su archivo de registros más grande y con su conjunto ortogonal de instrucciones, ARV es mucho más moderna que su competencia.

Por ejemplo, los 8051, 6805 y PIC se las arreglan con un único acumulador; los 68HC11 y HC12 tienen simplemente dos. Esto hace que la arquitectura AVR sea más fácil de programar a nivel de lenguaje ensamblador, y que sea fácil de optimizar con un compilador. El gran conjunto de registros disminuye la dependencia respecto a la memoria, lo cual mejora la velocidad y achica las necesidades de almacenamiento de datos.

El conjunto de instrucciones de la AVR que se asemeja al RISC, también contribuye a mejorar su densidad de código respecto a otros microprocesadores de 8 BIT. Su instrucción CPI (*compare immediate*) evita la engorrosa construcción de cargar, substraer y verificar los *flags*, que usan el 6808 y el PIC. Usualmente, la suma de dos números en el 8051, 6805, o el PIC involucra barajar ambos operandos a través del acumulador y almacenar el resultado. La AVR simplemente suma dos registros con una instrucción, en un ciclo de máquina.

Hablar de alto rendimiento se justifica pues que, a 20 MHz, su velocidad máxima de reloj duplica con facilidad la de los demás chips de su clase. Y lo que es más importante, casi todas instrucciones se ejecutan en 1 o 2 ciclos de reloj, versus 5-10 ciclos para los chips 8051, 6805, 68HC11, y PIC.

Para ser más preciso, el HC12, 8051XA, y el 251 son diseños de 16 BIT, con trayectorias internas de datos de 16 BIT y operaciones aritméticas de 16 BIT, pero aun requieren de tres o más ciclos de reloj para efectuar los cálculos más simples y la mayoría de las operaciones básicas con los registros. Además las velocidades de reloj de estos dispositivos no son más rápidas que las de ATMEL, y no pueden compensar la ejecución inherentemente ineficiente.

Resumiendo, la AVR ofrece un desempeño de 16 BIT con un precio de 8 BIT para los usuarios que deseen memoria en el chip pero que no necesiten tipos de datos de 16 BIT o compatibilidad binaria con la generación anterior, ATMEL ofrece una mejor relación precio/desempeño a aquellos diseñadores que deseen independizarse de las familias más antiguas.

El Microcontrolador seleccionado es el AVR ATMEGA 8.

#### **1.4.5.2 *Software* para programación para el Microcontrolador ATMEGA 8.**

En el mercado se ha encontrado el paquete computacional BASCOM AVR, distribuido por MCS ELECTRONICS, el mismo que permite programar toda clase de microcontroladores AVR en un lenguaje Basic de alto nivel de una manera rápida y estructurada. En la página web oficial de MCS ELECTRONICS se puede descargar una versión DEMO que permite programar hasta 4 *Kbytes* de código fuente y una versión completa con un costo de 96 €. Además proporciona una guía de uso con los códigos de programación.

Una vez descrito los parámetros, normas y reglamentos a considerar en la práctica deportiva, así como también la descripción de los dispositivos a utilizar, a continuación en el siguiente capítulo se procede a detallar el diseño del sistema de entrenamiento.

## **CAPÍTULO 2. DISEÑO DEL SISTEMA.**

En este capítulo se detalla el diseño del sistema para entrenamiento en función de las especificaciones propuestas para el sistema a ser implementado, además se describe el software a ser utilizado para una correcta interacción del sistema.

### **2.1 CONSIDERACIONES DE DISEÑO.**

Para el diseño lo importante es partir de los requerimientos que debe cumplir un determinado proyecto. En el presente diseño se ha considerado la interacción entre los distintos sujetos que actúan en una sesión de entrenamiento y la necesidad de desplegar y analizar la información a obtener. Por lo antes mencionado, el sistema se divide en:

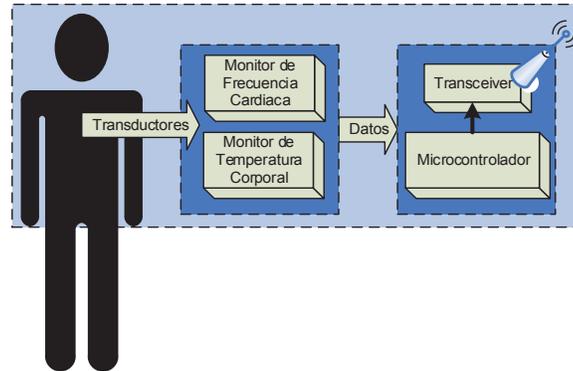
- a. Módulo del Atleta.
- b. Módulo del Entrenador.
- c. Interfaz de descarga y visualización en el computador.

Los dos primeros módulos son los encargados de recolectar los parámetros de Frecuencia Cardíaca, Temperatura Corporal y Tiempo; así que debe existir una interacción entre estos, pero al estar separados deben estar comunicados de forma inalámbrica. El módulo del entrenador debe poseer la capacidad de almacenar la información recolectada para que esta puede ser evaluada posteriormente en conjunto. De acuerdo a esto se determina la necesidad de una interfaz con el PC en el que se descargue, almacene y visualice la información de manera posterior.

A continuación se describe la función de cada parte del sistema a diseñar:

### **2.2 DISEÑO DEL MÓDULO DEL ATLETA.**

El módulo del atleta será el encargado de recolectar los parámetros de Frecuencia Cardíaca y Temperatura Corporal, y poseerá un sistema de control a través de un microcontrolador, el cual recolectará estos parámetros para su posterior transmisión hacia el módulo del entrenador, tal cual lo indica brevemente la Figura 2.1.



**Figura 2. 1** Módulo del Atleta.

### 2.2.1 REQUERIMIENTOS DEL MÓDULO DEL ATLETA.

Para el presente módulo los requerimientos que debe cumplir para la operación del mismo, son los siguientes parámetros:

- a) Debe esperar una orden de inicio por parte del módulo del entrenador y esta espera debe ser visualizada mediante un indicador en el dispositivo del atleta, que en este caso es un diodo LED, el mismo que permanecerá encendido mientras no reciba una orden de inicio.
- b) La orden enviada del módulo del entrenador será recibida en el puerto serial del microcontrolador del atleta, a través de los *transceivers* de RF. En el *software* del microcontrolador se tendrá una subrutina que tomará esta orden.
- c) La toma de datos del atleta se realizará mediante 2 subprocesos que son:
  - Muestreo de pulsos de Frecuencia Cardiaca.
  - Toma del dato medido del sensor de temperatura.
- d) Los datos obtenidos de frecuencia cardiaca y temperatura serán evaluados en el tiempo. Para tal efecto se ha escogido un periodo de evaluación de 10 segundos. Cada 10 segundos se tomará la muestra de frecuencia cardiaca y el dato medido de temperatura.
- e) Los datos evaluados serán transmitidos al módulo del entrenador cada 10 segundos.
- f) El tiempo de la sesión de entrenamiento considerado es 60 minutos.

De acuerdo a estos requerimientos, a continuación se procede con el diseño de los componentes que integran el módulo.

## 2.2.2 DISEÑO DEL MONITOR DE FRECUENCIA CARDIACA

Es un circuito desarrollado en modalidad abierta que consta de un sensor fotoeléctrico con un emisor de luz y un receptor, el mismo que al estar en contacto con ciertas partes del cuerpo humano que contengan vasos sanguíneos como el lóbulo del oído o el dedo de la mano, detectará la cantidad de luz reflejada en función del paso de la sangre.

Esta cantidad de luz es convertida a una señal eléctrica que es acondicionada y compensada teniendo a la salida del circuito una señal medible.

### 2.2.2.1 Manejo del *PulseSensor*

Este sensor consta de tres pines, de los cuales dos están dedicados a la alimentación de voltaje DC, el mismo que debe ser desde los 3 V hasta los 5 V para su correcto funcionamiento y además tiene un pin de salida de datos.

La salida de datos que proporciona el *PulseSensor* es una señal de voltaje analógica que oscila entre un nivel alto y un nivel bajo de acuerdo a lo mostrado en la Figura 2.2, la misma que puede ser utilizada según sea la necesidad.

Para el presente diseño esta señal deberá ser tomada solo para conteo de pulsos de frecuencia cardiaca, por lo que se realiza la conversión de esta salida a un tren de pulsos [11].

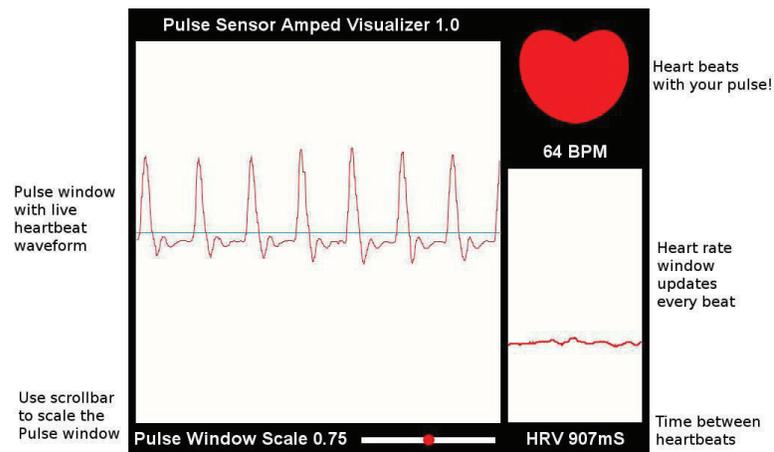
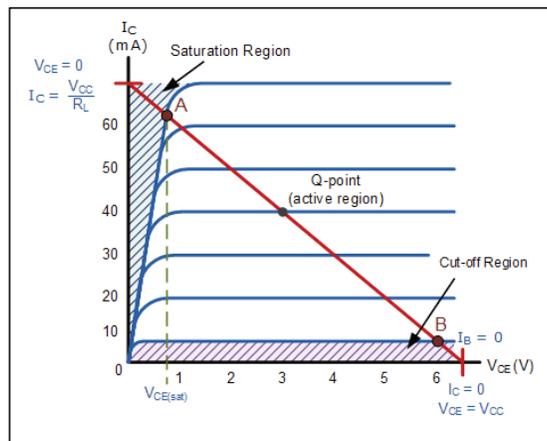


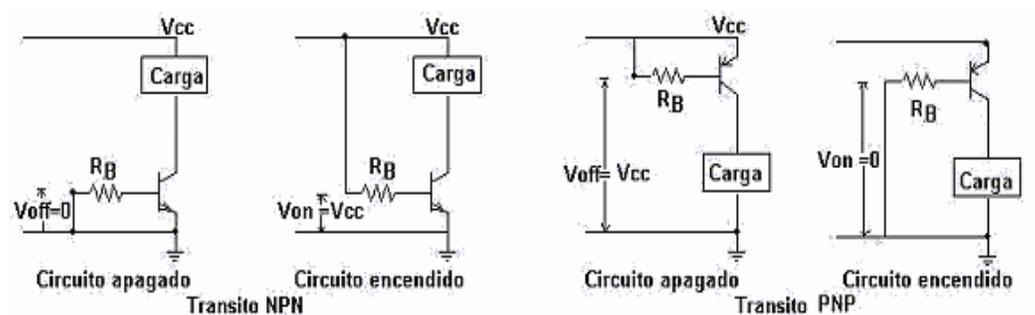
Figura 2. 2 Forma de Onda de la Salida del *PulseSensor*. [11]

Para la conversión de la señal de salida del *PulseSensor* a un tren de pulsos, se considera el diseño de un circuito sencillo utilizando un transistor como *switch*. Un transistor funciona como un *switch*, si este pasa rápidamente de su región de corte a saturación y viceversa, tal como se muestra en la Figura 2.3. En corte es un interruptor abierto y en saturación es un interruptor cerrado [15].



**Figura 2. 3** Región de Corte y Saturación de un Transistor. [15]

En la Figura 2.4 se muestran los circuitos con los transistores NPN y PNP en modo switch [16].



**Figura 2. 4** Transistores NPN y PNP en Modo *Switch* en Base a la Región de Corte y Saturación. [16]

Los datos para calcular un circuito de transistor para que funcione como *switch* son: El voltaje del circuito que se va a encender y la corriente que requiere con ese voltaje. El voltaje  $V_{cc}$  se hace igual al voltaje nominal del circuito y la corriente corresponde a la corriente de colector de saturación  $I_{C_{sat}}$ . Se calcula la corriente de

base de saturación mínima  $I_{Bsat}$  (Ec. 2.1) y luego la resistencia de base mínima  $R_B$  (Ec. 2.2).

$$I_{Bsat} = \frac{I_{Csat}}{\beta} \quad (\text{Ec. 2.1})$$

$$R_B = \frac{V_{in} - V_{BE}}{I_{Bsat}} \quad (\text{Ec. 2.2})$$

Para el diseño se considera el circuito con transistor PNP en modo *switch*, ya que no se comporta como inversor de señal a la salida, es decir, para un voltaje base emisor  $V_{BE}$  mayor a 0.7 voltios a la entrada se tendrá a la salida un voltaje positivo igual al voltaje Vcc (el voltaje Vcc es igual al voltaje de entrada  $V_{in}$ ) y para valores menores a 0.7 voltios se tendrá a la salida 0 voltios. De esta forma se consigue digitalizar la señal en la salida a un tren de pulsos referente a la frecuencia cardiaca del individuo a medir, los mismos que pueden ser contabilizados.

Para el proyecto el resultado de los cálculos se realizan teniendo en cuenta que el voltaje de Vcc es constante de 5 voltios, se trabaja con el transistor 2N3906 con un  $\beta$  igual a 100 y se requiere una corriente de colector  $I_{Csat}$  mínima de 100 mA. Con esta información se procede a calcular con las ecuaciones 2.1 y 2.2 para obtener los valores de  $I_{Bsat}$  y  $R_B$ .

$$I_{Bsat} = \frac{I_{Csat}}{\beta} = \frac{100 \text{ mA}}{100} = 1 \text{ mA}$$

$$R_B = \frac{V_{in} - V_{BE}}{I_{Bsat}} = \frac{5 \text{ V} - 0.7 \text{ V}}{1 \text{ mA}} = 4.3 \text{ k}\Omega$$

Con estos valores, el circuito con transistor PNP en modo *Switch* resultante será el mostrado en la Figura 2.5.

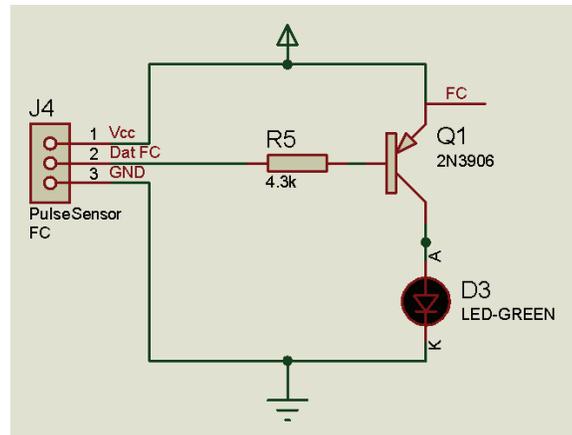


Figura 2. 5 *PulseSensor* Acoplado al Transistor PNP *Switch*.

#### 2.2.2.2 Descripción del proceso de la contabilización de pulsos

Como se describió en el Capítulo 1, el microcontrolador seleccionado es ATMEGA 8, el mismo que tiene dos puertos, PD.2/INT0, PD.3/INT1 que pueden ser activados por una interrupción externa.

Para el módulo del atleta se selecciona el puerto PD.2/INT0, al cual ingresará la señal digitalizada del sensor de pulsos. Mediante una interrupción externa programada en el microcontrolador y activada por el pulso del sensor permitirá contabilizar el pulso detectado.

La programación para activar la interrupción es la siguiente:

- Primero se configura el estado en la que se activará la interrupción, con la siguiente sintaxis:

```
CONFIG INTx = state [17]
```

Donde X puede ser 0 para el puerto PD.2/INT0 y 1 para el puerto PD.3/INT1

**state**, puede ser configurado como:

- LOW LEVEL: genera la interrupción cuando en el puerto se encuentra un nivel bajo de voltaje.
- FALLING: Genera la interrupción en un flanco descendente.
- RISING: Genera la interrupción en un flanco ascendente.

- CHANGE: Genera la interrupción en el cambio de borde o arista en el flanco ascendente y descendente.
- Segundo se configura la subrutina que se ejecutará con la interrupción, la misma que lleva la siguiente sintaxis:

```
On Int0 Count_int [17]
```

Donde On indica la activación, Int0 el puerto activado y Count\_int la subrutina a ejecutar.

- Tercero se configura una variable que contabilizará los pulsos, la misma que se presentará en su valor inicial de cero.

```
Dim Fc As Integer [17]
```

Donde Dim declara la variable Fc, e Integer es el tipo de variable.

Fc = 0 (se define la variable en valor inicial de cero)

- Cuarto, habilita las interrupciones, con la siguiente sintaxis:

```
Enable Int0 [17]
```

```
Enable Interrupts
```

- Quinto, se declara la subrutina y la acción o acciones que realizará al activarse la interrupción, para contabilizar el pulso. La sintaxis es:

Count\_int:

Se declara la subrutina

Fc = Fc + 1 'subrutina que cuenta las pulsaciones

Es la acción dentro de la subrutina.

Return

Es el comando que permite retornar al programa principal una vez terminada la acción de la interrupción.

En el proceso mostrado se está contabilizando los pulsos y a continuación se requiere obtener una muestra de los datos adquiridos para expresarlos en la unidad de la frecuencia cardiaca que es pulsos por minuto.

### 2.2.2.3 Descripción del muestreo de pulsos.

Se procederá a tomar una muestra de pulsos en un periodo de tiempo de 10 segundos y para obtener la FC de 1 minuto se realizará la multiplicación de la muestra por 6.

La toma de muestra mencionada será realizada mediante la programación de una subrutina que se activará mediante una interrupción interna proporcionada por un *Timer* configurado a los 10 segundos que se necesita. Esta subrutina tomará los datos almacenados en la variable *Fc* multiplicada por 6 y encerrará la misma para una nueva contabilización, así como iniciará nuevamente la interrupción interna del *Timer* para un nuevo muestreo dentro del siguiente periodo de 10 segundos.

El microcontrolador seleccionado posee dos *Timers/Counter* internos. Para el diseño se requiere configurar uno de ellos como *Timer* y se ha seleccionado el *Timer 1* que posee un contador de 16 bites.

La Programación para la interrupción interna es la siguiente:

- Primero, se procede a configurar el *Time/Counter* como *Timer* y la división del reloj del sistema con la siguiente sintaxis:

Config *Timer1* = *Timer* , Prescale = 1024 [17]

El tiempo que utiliza para contar los 16 bites del *Timer* se puede calcular mediante la siguiente expresión:

$$T = 65536 * \frac{Prescaler}{f_{osc}} \quad (\text{Ec. 2.3})$$

$$T = 65536 * \frac{1024}{11059200} = 6.068148148 \text{ segundos}$$

Donde 65536 es el número que está expresado en base decimal que se obtiene de los 16 bites que posee el *Timer*, la frecuencia de oscilación está dado por el cristal externo que utiliza el microcontrolador.

- Segundo, para los 10 segundos requeridos, se necesita conocer el número de veces que se utilizará el *timer* y se calcula con la siguiente fórmula:

$$N = \text{Número de veces que se utiliza el timer}$$

$$N = \frac{tr}{T} = \frac{\text{tiempo requerido}}{\text{Tiempo completo de Timer}} = \frac{10 \text{ segundos}}{6.068148148 \text{ segundos}} = 1.647949219$$

Esto indica que se utiliza el *Timer* más de una vez, teniéndose un primer conteo completo de los 16 bites del *Timer* en 6.068148148 *segundos* y como se requiere una muestra de 10 segundos, la diferencia del tiempo de muestra y el tiempo de conteo de timer, permitirá obtener el bite con el que se cargará el *Timer* en el porcentaje que se requiere en el segundo conteo.

$$t_{rest} = tmuestra - T = 10 - 6.068148148 = 3.93185185 \text{ segundos}$$

De acuerdo a esto el bite a cargarse en el segundo conteo, se calcula en base a:

$$Bite_{SEGUNDA CARGA} = 65536 - \frac{t_{rest} * 65536}{T} = 65536 - \frac{3.93185185 * 65536}{6.068148148}$$

$$Bite_{SEGUNDA CARGA} = 23072$$

**Timer1 = 23072** [17]

- Tercero, se configura la subrutina que se ejecutará con la interrupción

**On Timer1 Tiempo** [17]

Donde On indica la activación del *Timer1* y Tiempo es la subrutina a ejecutar.

- Cuarto, habilita las interrupciones e inicializamos la ejecución del *Timer1*, con la siguiente sintaxis:

**Enable Timer1** [17]

**Start Timer1**

- Quinto, se declara la subrutina y las acciones que realizará, al activarse el *Timer1*, de lo cual la sintaxis es:

Tiempo:

**Stop Timer1**

'Detiene al *Timer*. [17]

**If Ti < 1 Then**

'Ti se utiliza para cumplir 10 segundos ya que se requiere de 2 pasadas.

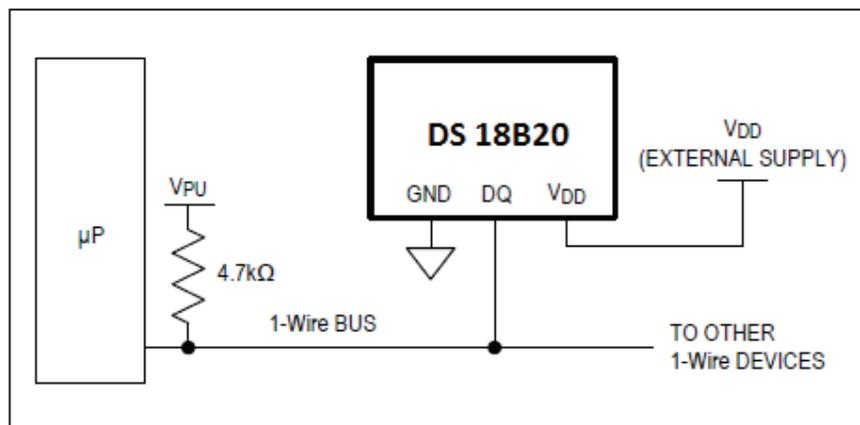
**Ti = Ti + 1**

'Aumenta el valor de Ti.

Timer1 = 0	'Se carga el contador (completo) del <i>Timer</i>
Start Timer1	'Se inicia nuevamente el contador.
Else	'Cuando a pasado 2 veces se ha cumplido con 10 segundos y se ejecuta.
Ti = 0	'Se inicializa contador de vueltas de <i>Timer</i>
Timer1 = 23072	'Se inicializa el <i>Timer</i> para cumplir con 10 Segundos.
Start Timer1	'Se inicializa el <i>Timer</i> nuevamente.
End If	
Return	'Retorno al programa principal
End	

### 2.2.3 DISEÑO DEL MONITOR DE TEMPERATURA CORPORAL

En referencia al Capítulo 1, donde se describe el sensor DS 18B20 con el que se va a trabajar en el presente proyecto, se procede a analizar la configuración y funcionamiento de este dispositivo para la obtención de información de temperatura. Este dispositivo dispone de 3 pines, donde dos de estos son destinados para alimentación de energía de DC que va desde 3.0 V a 5 V y una línea de datos que utiliza un protocolo propietario de Dallas 1wire (un hilo). El diagrama de conexiones para el uso del DS 18B20 se muestra en la Figura 2.6.



**Figura 2. 6** Diagrama de Conexiones del Sensor de Temperatura DS 18B20. [18]

Al sensor de temperatura DS 18B20 al módulo del atleta, se lo conecta al microcontrolador ATMEGA 8 a un puerto al que, mediante programación en BASCOM AVR, se lo definió para que trabaje con el protocolo 1wire.

### 2.2.3.1 Comandos de Operación

En la Tabla 2.1 se indican los comandos de operación del DS 18B20 basados en los cuales, se procedió a desarrollar el programa de lectura de datos del sensor de temperatura:

- Primero, se configura el puerto que va a trabajar con el protocolo 1wire.

Config 1wire = PORTB 0 [17]

COMMAND	DESCRIPTION	PROTOCOL	1-Wire BUS ACTIVITY AFTER COMMAND IS ISSUED	NOTES
<b>TEMPERATURE CONVERSION COMMANDS</b>				
Convert T	Initiates temperature conversion.	44h	DS18S20 transmits conversion status to master (not applicable for parasite-powered DS18S20s).	1
<b>MEMORY COMMANDS</b>				
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	BEh	DS18S20 transmits up to 9 data bytes to master.	2
Write Scratchpad	Writes data into scratchpad bytes 2 and 3 ( $T_H$ and $T_L$ ).	4Eh	Master transmits 2 data bytes to DS18S20.	3
Copy Scratchpad	Copies $T_H$ and $T_L$ data from the scratchpad to EEPROM.	48h	None	1
Recall E <sup>2</sup>	Recalls $T_H$ and $T_L$ data from EEPROM to the scratchpad.	B8h	DS18S20 transmits recall status to master.	
Read Power Supply	Signals DS18S20 power supply mode to the master.	B4h	DS18S20 transmits supply status to master.	

**Note 1:** For parasite-powered DS18S20s, the master must enable a strong pullup on the 1-Wire bus during temperature conversions and copies from the scratchpad to EEPROM. No other bus activity may take place during this time.

**Note 2:** The master can interrupt the transmission of data at any time by issuing a reset.

**Note 3:** Both bytes must be written before a reset is issued.

**Tabla 2. 1** Comandos de Configuración del DS 18B20. [18]

- Segundo, se define una subrutina que realice la lectura de los datos del sensor de temperatura

Declare Sub Temperature [17]

- Tercero, se declara la subrutina y las acciones que realizará, al ser llamada desde el programa principal, de la forma siguiente:

```

Sub Temperature [17]

    1wreset                                'Resetea el bus

    1wwrite &HCC : 1wwrite &H44           'CCh indica que el siguiente comando debe ser
                                          atendido.

                                          '44h da la orden de comenzar con la conversión.

    Waitms 300

    1wreset                                'Resetea el bus.

    1wwrite &HCC                           'CCh indica que el siguiente comando debe ser
                                          atendido.

    1wwrite &HBE                           'BEh indica que lee los 9 bytes de datos de su RAM

    For I = 1 To 8

        Ar(i) = 1wread()                  ' Se almacena los datos leídos de la RAM en una
                                          variable.

    Next

    T = Makeint(ar(1) , Ar(2))            'Convierte 2 bytes en uno individual
                                          =(256*MSB)+LSB

    Temp = T                               'Se almacena una variable para que exista la
                                          conversión a número decimal.

    Temp = Temp / 16                      'Valor transformado a decimal (entero más decimas
                                          de valor).

    1wreset                                'Avance de Línea.

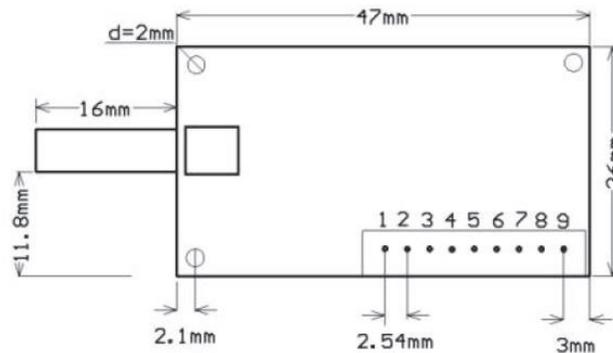
End Sub                                  'Retorna al programa principal.

```

#### 2.2.4 DISEÑO DEL TRANSMISOR.

El *transceiver* YS – 1020 UB tiene 9 pines, dos de ellos dedicados a la alimentación de DC desde 3.3 V hasta 5.5 V. Posee 2 líneas de datos seriales de transmisión y

recepción en niveles TTL, 2 líneas de datos seriales de transmisión y recepción que trabajan con los protocolos RS-232 o RS-485, un puerto de control del estado de reposo y un puerto de test para pruebas de fábrica. Lo descrito se muestra en la Figura 2.7.



**Figura 2. 7** Dimensiones y Distribución Pines del YS-1020UB. [19]

Pin	Pin name	Description	Level	Connection with terminal	Remands
1	GND	Grounding of power supply		Ground	
2	Vcc	Power supply DC	+3.3~5.5V		
3	RXD/TTL	Serial data receiving end	TTL	TxD	
4	TXD/TTL	Serial data transmitting end	TTL	RxD	
5	DGND	Digital grounding			
6	A(TXD)	A of RS-485 or TXD of RS-232		A(RxD)	
7	B(RXD)	B of RS-485 or RXD of RS-232		B(TxD)	
8	Sleep	Sleep control (input)	TTL	Sleep signal	Low level sleep
9	Test	Ex-factory testing			

NOTE: Generally the module is in receiving status, if the Sleep pin (No.8) continuously connects low level (>200millisecond), the module will be in sleep status, modules can not receive or transmit any data when sleep.

**Tabla 2. 2** Descripción y Funcionalidad de Pines del YS-1020UB. [19]

A continuación se describe las diferentes formas en que el *transceiver* se puede conectar (Ver Figura 2.8):

- TTL RF modem: TTL (Lógica Transistor Transistor), trabaja entre 0 V y 5 V como niveles lógicos. Permite trabajar directamente con las salidas seriales del microcontrolador sin la necesidad de conversión RS-232.

- RS-232 RF modem: El estándar RS-232, permite trabajar directamente con los puertos DB-9 y DB-25 presentes en computadores y otros dispositivos.
- RS-485 RF modem: Estándar conocido como EIA-485, permite trabajar directamente con dispositivos que soporten este estándar. [19]

Para el diseño del módulo del atleta se considera trabajar con las líneas del estándar TTL conectados directamente a los puertos que maneja la comunicación serial del microcontrolador, para que el hardware contenga una menor cantidad de componentes físicos.

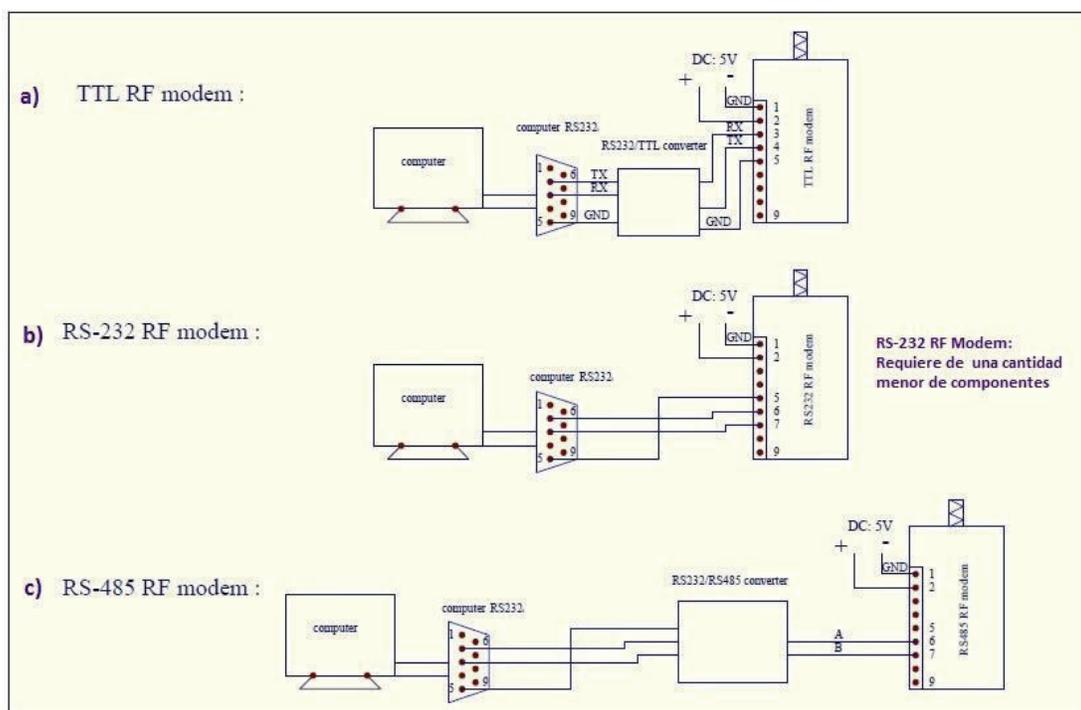


Figura 2. 8 Tipos de Conexiones del *Transceiver* YS-1020UB. [23]

## 2.3 DISEÑO DEL MÓDULO DEL ENTRENADOR

El módulo del entrenador es el encargado de iniciar la sesión de entrenamiento, dando la orden al módulo del atleta que inicie la toma de datos del deportista. La toma de datos que realiza el módulo del atleta será enviada hacia el módulo del entrenador, el mismo que la recibe en forma inalámbrica para su posterior almacenamiento interno y presentación en forma visual de la información tomada del deportista mediante un LCD. El módulo del entrenador posee un reloj en tiempo

real y de acuerdo al funcionamiento de este, se encarga de registrar los datos recibidos desde el deportista en función del tiempo, logrando así identificar las diferentes etapas de la sesión de entrenamiento.

### 2.3.1 REQUERIMIENTOS DEL MÓDULO DEL ENTRENADOR

Para el Módulo del Entrenador los requerimientos identificados son:

- a) En inicio el módulo del entrenador debe permitir igualar datos relacionados a la fecha y hora actualizados.
- b) Debe permitir seleccionar una orden de mando de tal forma que indique al módulo del atleta que debe realizar: toma de datos o detener la toma de datos.
- c) Mediante el uso de 2 pulsadores a nivel de hardware, se permitirá al entrenador realizar las acciones de: Igualar reloj, seleccionar opciones de menú y envío de órdenes al módulo del atleta. El uso de estos pulsantes será:
  - El primer pulsador, dependiendo del número de veces pulsado, permitirá el ingreso de un número, que puede ser el ingreso de un dato que necesita el microcontrolador o la selección de una opción.
  - El segundo pulsador realizará las veces de una tecla *Enter* o Aceptar la información ingresada por el primer pulsador.
- d) El envío y recepción de información debe realizarse en forma inalámbrica hacia el módulo del atleta.
- e) Debe permitir la visualización de la información a través de un LCD de 16\*2 como: Fecha, hora, menús de selección y en medición de datos del atleta la frecuencia cardiaca y temperatura.
- f) Debe permitir la descarga de datos tomados de una sesión de entrenamiento a través de una interfaz de comunicación hacia un computador.

De acuerdo a estos requerimientos, a continuación se procede con el diseño de los componentes que integran el módulo.

### 2.3.2 DISEÑO CON EL RELOJ DE TIEMPO REAL

El DS1307 es un dispositivo integrado que utiliza el protocolo de bus de comunicación en serie I<sup>2</sup>C (*Inter-Integrated-Circuit*). La principal característica de este protocolo es que utiliza dos líneas para transmitir la información. Una para los datos y otra para la señal de reloj. Adicionalmente, tiene una salida de onda cuadrada programable. A continuación se muestra en la Figura 2.9, las conexiones que tendrá el DS 1307 con el microcontrolador.

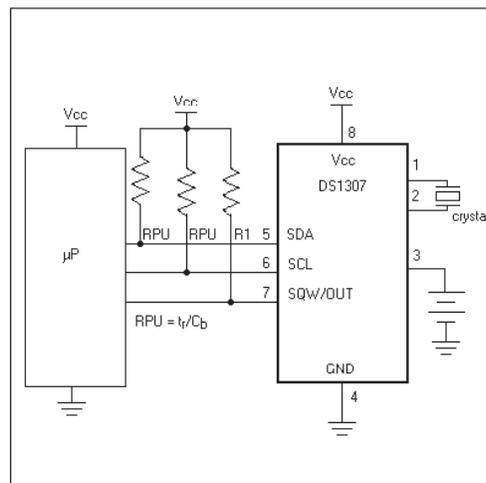


Figura 2. 9 Diagrama de Conexiones del DS 1307. [20]

- V<sub>CC</sub> es el voltaje de alimentación que se encuentra entre 4.5 V y 5 V.
- V<sub>bat</sub> es el voltaje provisto por una batería de 3 V.
- SCL (*Serial Clock Input*), es usado para sincronizar el movimiento de datos en la interfaz serial.
- SDA (*Serial Data Input/Output*), es el pin de entrada y salida de la interfaz a través de una interfaz seria de 2 hilos. Esta salida requiere una resistencia de *pull-up*.
- SQW/OUT (*Square wave/Output Driver*), este pin puede ser seteado para una salida de onda cuadrada, con las frecuencias de 1 Hz, 4 Hz, 8 Hz y 32 Hz. Esta salida requiere una resistencia de *pull-up*.
- X1, X2, se conectan a un cristal externo de 32.768 KHz.

De acuerdo a la Figura 2.9, para el cálculo de la resistencia de *pull-up* RPU se usa la siguiente ecuación:

$$RPU \geq \frac{V_{CC} - V_{OLm\acute{a}x}}{I_{OL}} [23]$$

Donde:

$V_{DD}$  es el voltaje de alimentación de 5 voltios.

En las características eléctricas en DC para el DS1307 se tiene que  $V_{OLm\acute{a}x}$  es igual 0.4 voltios y  $I_{OL}$  es 1.5 mA. [20]

Según estos valores, el cálculo de las resistencias de *Pull-up* es el siguiente:

$$RPU \geq \frac{V_{CC} - V_{OLm\acute{a}x}}{I_{OL}} = \frac{5\text{ v} - 0.4\text{ v}}{1.5\text{ mA}} = 3066.66\ \Omega$$

En el diseño las resistencias de *Pull-Up* tendrán un valor de 4.7 K $\Omega$ .

R1 corresponde a la línea SQW/OUT del DS1307, la cual en el presente proyecto no se utiliza.

La Figura 2.10 describe la información que permite configurar la hora y fecha del DS 1307, la misma que de 8 bytes.

Se define la dirección de byte de escritura HD0 (*bin* 11010000), donde los 7 primeros bites indican la condición esclavo del módulo y el ultimo bit es el que define escritura. Adicionalmente, se define la dirección del registro de escritura HD1 (*bin* 11010001), donde los 7 primeros bites indican la condición esclavo del módulo y el ultimo bit es el que define lectura.



`Const Ds1307r = &HD1` 'Se carga una constante con el byte de comando de lectura.

`Config Clock = User` 'Se declara la sentencia para configurar en forma manual por el usuario.

`Config Date = Dmy , Separator = /` 'Se configura la forma de presentar los datos en formato Día/Mes/Año.

- Tercero, se define la subrutina para la lectura del reloj DS 1307, con las siguientes líneas de comando:

`Getdatetime: [17]` 'Subrutina para lectura del reloj

`I2cstart` 'Genera byte comando de inicio del bus I2C

`I2cwrite Ds1307w` 'Envía el byte de escritura al bus I2C

`I2cwrite 0` 'Envía el byte de inicio en el bus I2C.

`I2cstart` 'Genera byte comando de inicio del bus I2C

`I2cwrite Ds1307r` 'Envía el byte de escritura al bus I2C

'En las siguientes líneas de comando, se realiza la lectura de los valores del reloj y los almacena en sus variables correspondientes

`I2cwrite _sec , Ack` 'Segundos

`I2cwrite _min , Ack` 'Minutos

`I2cwrite _hour , Ack` 'Horas

`I2cwrite Weekday , Ack` 'Día de la semana

`I2cwrite _day , Ack` 'Día del mes

`I2cwrite _month , Ack` 'Mes del año.

`I2cwrite _year , Nack` 'Año.

`I2cstop`

`_sec = Makedec(_sec) : _min = Makedec(_min) : _hour = Makedec(_hour)`

'Makedec convierte un código binario a decimal

```
_day = Makedec(_day) : _month = Makedec(_month) : _year = Makedec(_year)
```

#### Return

- Cuarto, se define las subrutinas para la escritura del reloj DS 1307 con las siguientes líneas de comando:

```
Settime: [17] 'Subrutina que permite ingresar segundos, Minutos,
              Hora
```

```
_sec = Makebcd(_sec) : _min = Makebcd(_min) : _hour = Makebcd(_hour)
```

'Makebcd, convierte en decimal a binario.

```
I2cstart 'Genera byte comando de inicio del bus I2C
```

```
I2cwbyte Ds1307w 'Envía el byte de escritura al bus I2C.
```

```
I2cwbyte 0 'Envía el byte de inicio en el bus I2C.
```

'En las siguientes líneas de comando, se realiza la escritura de los valores en el reloj

```
I2cwbyte _sec 'Segundos
```

```
I2cwbyte _min 'Minutos
```

```
I2cwbyte _hour 'Horas
```

```
I2cstop
```

```
Return
```

```
Setdate: 'Subrutina que permite ingresar Día, Mes, Año
```

```
_day = Makebcd(_day) : _month = Makebcd(_month) : _year = Makebcd(_year)
```

'Makebcd, convierte en decimal a binario.

```
I2cstart 'Genera byte comando de inicio del bus I2C
```

```
I2cwbyte Ds1307w 'Envía el byte de escritura al bus I2C.
```

```
I2cwbyte 4 'Envía el byte de inicio en el bus I2C.
```

'En las siguientes líneas de comando, se realiza la escritura de los valores en el reloj

```
I2cwbyte _day 'Día
```

```
I2cwbyte _month 'Mes
```

[I2cwrite\\_year](#)

'Año

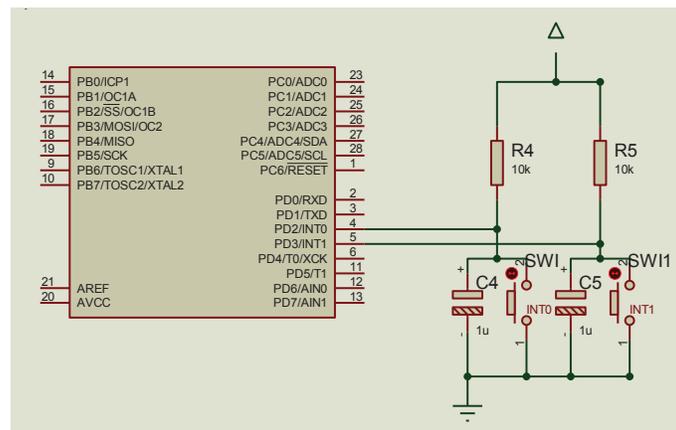
[I2cstop](#)

[Return](#)

### 2.3.3 DISEÑO DE LOS TRANSMISORES.

Tal cual se describió en el apartado 2.1.1.3, se utilizará el mismo Transceiver YS-1020UB, siendo la única diferencia el uso de las líneas de trabajo en el estándar RS-232.

### 2.3.4 DISEÑO DE LA INTERFAZ DE INGRESO Y SELECCIÓN DE DATOS.



**Figura 2. 11** Diagrama de Conexión del Microcontrolador y Pulsadores.

Conforme a la Figura 2.11, para los pulsadores se utilizarán los puertos del microcontrolador de la siguiente forma:

- Pulsador Ingreso de Datos al puerto PD.2/INT0
- Pulsador de Aceptar al puerto PD.3/INT1

Para la eliminación de los rebotes producidos por los pulsadores se utilizarán las resistencias  $R_4$  y  $R_5$  de un valor de 10 KΩ y los capacitores  $C_4$  y  $C_5$  de valor de 1 μF. Estos valores vienen dados por la ecuación:

$$T_e = R * C$$

Donde  $T_e$  es el periodo de estabilización que es mayor al tiempo de un rebote presente, siendo el tiempo de rebote aproximadamente 2 milisegundos. De acuerdo a esto, los valores de resistencia y capacitores se justifican de acuerdo al siguiente cálculo:

$$T_e = R * C = (10 * 10^3) * (1 * 10^{-6}) = 10 \text{ milisegundos}$$

#### 2.3.4.1 Comandos de Operación.

- Primero: Se configura la subrutina que se activará al producirse la interrupción ocasionada por el pulsador. Con la siguiente sentencia de comando:

**On** Int1 Confir

Donde On indica la activación, Int1 el puerto activado y Confir el nombre de la subrutina a ejecutar.

- Segundo: Se configura el estado en el que se activará la interrupción con el siguiente comando:

**Config** Int1 = Falling

Donde Config realiza la configuración de la Interrupción 1 que se activará con el estado de un flanco descendente a través del comando Falling.

- Tercero: Se habilita la interrupción Int1, con el comando:

**Enable** Int1 'Se habilita las interrupción de entrada INT1

- Cuarto: Se declara la subrutina y las acciones que realizará con la activación de la interrupción

<b>Confir:</b>	<span style="color: green;">'Subrutina para confirmar</span>
<b>Disable</b> Int1	<span style="color: green;">'Deshabilita la interrupción</span>
Acep = 2	<span style="color: green;">'Se cambia la variable a 2 que es el indicador de confirmación</span>
<b>Cls</b>	<span style="color: green;">'Borra El LCD</span>
<b>Lcd</b> "Aceptar"	<span style="color: green;">'Confirma selección</span>
<b>Enable</b> Int1	<span style="color: green;">'Habilita la interrupción</span>
<b>Return</b>	<span style="color: green;">'Sale de la subrutina</span>
<b>End</b>	

Para cada pulsador se seguirá el mismo proceso.

### 2.3.5 INTERFAZ VISUAL PARA MODULO DEL ENTRENADOR

#### 2.3.5.1 Descripción del dispositivo a utilizar LCD 16\*2

El presente proyecto tiene contemplado, dentro del módulo del entrenador la visualización de los datos tomados por el módulo del atleta. De acuerdo a esto, se decidió usar un dispositivo LCD de 16\*2 como el que se muestra en la Figura 2.12.



Figura 2. 12 LCD 16\*2.

Los comandos utilizados en el microcontrolador para el manejo del LCD son:

- Primero, se describen los comandos para la configuración del modo de trabajo, así como también los pines utilizados en el manejo del LCD.

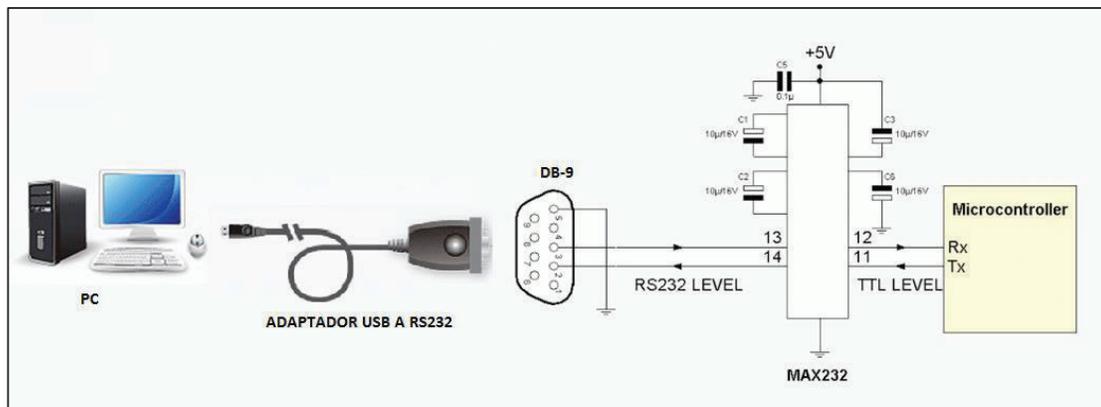
<code>Config Lcdmode = Port [17]</code>	'Indica que se utilizaran puertos al azar
<code>Config Lcdbus = 4</code>	'Indica que se utiliza bus de 4 para el LCD
<code>Config Lcdpin = Pin , Db4 = Portc.0 , Db5 = Portc.1 , Db6 = Portc.2 , Db7 = Portc.3 , E = Portb.1 , Rs = Portb.0</code>	'Se configura los puertos para el bus de datos y los puertos para el control del LCD
<code>Config Lcd = 16 * 2</code>	'Configuración los pines de un LCD de 16 caracteres en 2 líneas

- Segundo, se describen los comandos que se usan para el despliegue y manejo de la información.

Cls	'Se limpia las líneas del LCD
Locate 1 , 2	'Ubica el texto inicial en la primera fila y segunda columna.
Lcd "Politécnica"	'Despliegue de la información en el LCD
Locate 2 , 3	'Ubica el texto en la segunda fila y tercera columna
Lcd "NACIONAL"	'Despliegue de la información en el LCD

### 2.3.6 INTERFAZ MÓDULO DEL ENTRENADOR - PC.

Con el objetivo de descargar los datos almacenados en el módulo del entrenador a un computador, se contempló el uso del puerto serial del microcontrolador y el puerto USB presente en los computadores y laptops actuales, para lo cual se utilizó el siguiente diagrama de conexiones que se presenta en la Figura 2.13:



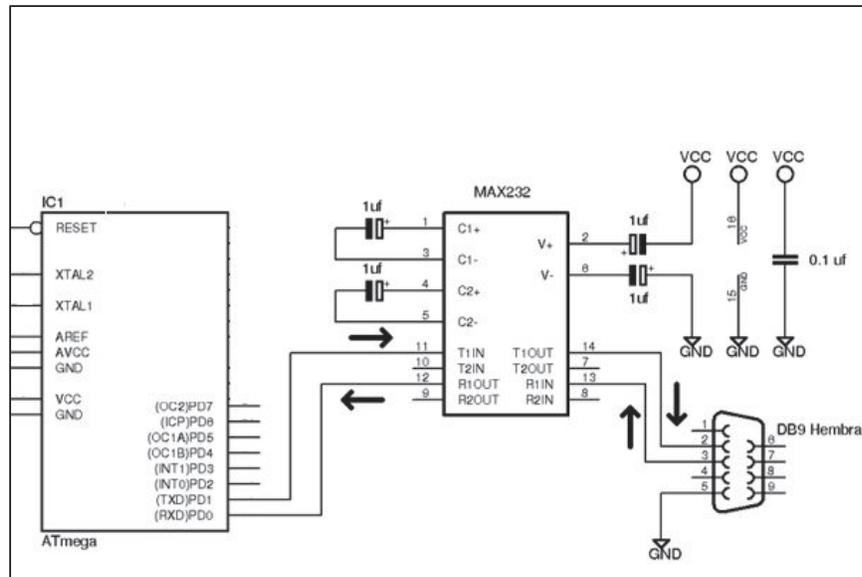
**Figura 2. 13** Diagrama de Conexiones de la Interfaz Microcontrolador – PC.

#### 2.3.6.1 Circuito Conversor de señal TTL a señal RS-232. [21]

Para realizar el conversor de señal TTL a señal del estándar RS-232 se utiliza el integrado MAX232. El MAX232 es un circuito integrado que convierte los niveles de las líneas de un puerto serie RS-232 a niveles TTL y viceversa. Para ello necesita

una alimentación de Vcc de 5V, ya que genera internamente algunas tensiones que son necesarias para el estándar RS-232.

Para el Módulo del entrenador se añade a la circuitería un MAX232 según el diagrama que se muestra en la Figura 2.14.



**Figura 2. 14** Conexiones de Circuito entre un Microcontrolador ATMEGA 8 y el MAX232.

Para el diseño del presente proyecto de acuerdo a la Figura 2.14 los valores de los capacitores de  $1\mu\text{F}$  son los recomendados por el *datasheet* del CI MAX232.

### 2.3.6.2 Adaptador RS-232 a USB

Para lograr acoplar el módulo del entrenador con el PC o Laptop se ha optado por el uso de un adaptador de RS-232 a USB, como el que se muestra en la Figura 2.15, ya que en la actualidad en los computadores modernos no se tiene la interfaz de entrada RS-232 con conector DB9 hembra, más bien se tiene el uso masificado del puerto USB.

Este tipo de adaptador al ser conectado al puerto USB del computador, crea automáticamente un puerto COM, el cual puede ser manejado a través de un

lenguaje de programación como por ejemplo Visual Basic y de esta forma poder desplegar en el computador la información almacenada en el microcontrolador del módulo del entrenador.



**Figura 2. 15** Fotografía de Adaptador RS-232 a USB.

### **2.3.7 DESARROLLO DE LA INTERFAZ DE DESCARGA Y VISUALIZACIÓN EN EL COMPUTADOR.**

Para el procesamiento y manejo de la información a descargar desde el módulo del entrenador y su posterior visualización en forma gráfica en el computador, se ha optado por el software de programación Visual Basic 2010, debido a su alto desarrollo y gran material de apoyo para el interacción con sistemas electrónicos externos al computador, así como también la información propia del computador.

#### **2.3.7.1 Requerimientos para el Software.**

El programa cumplirá los siguientes requerimientos:

- a) Interactuar con el módulo del entrenador para realizar la descarga de información hacia el computador.
- b) Guardar los datos descargados en un archivo bajo formato Excel.
- c) Graficar los datos almacenados en el computador.

### **2.4 ESQUEMA GENERAL DEL SISTEMA.**

En base al diseño de los dos módulos a continuación se presenta el diagrama del sistema general en la Figura 2.16.

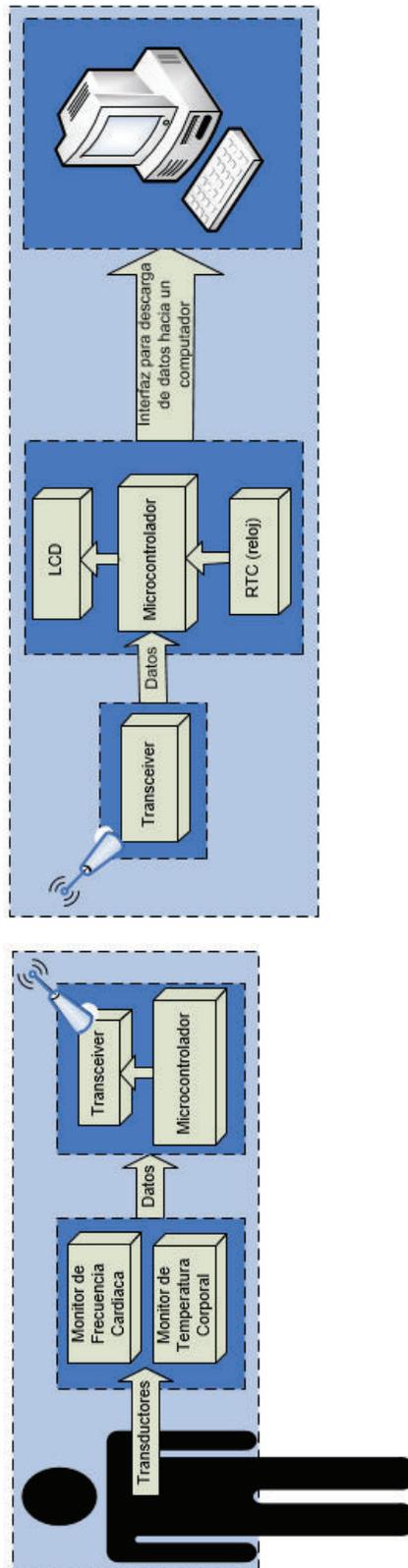


Figura 2. 16 Esquema General del Sistema.

Una vez realizado el diseño de los módulos del sistema de entrenamientos, a continuación en el siguiente capítulo se procede a la construcción del sistema a nivel de hardware y software.

## **CAPÍTULO 3. CONSTRUCCIÓN DEL SISTEMA PARA ENTRENAMIENTOS**

En este capítulo se detalla la construcción de los diferentes módulos del sistema por hardware y software, así como también la solución de problemas individuales con el objetivo de probar cada una de las partes del sistema. También se realizan consideraciones en la construcción del Módulo del Atleta debido a que estará sometido a movimientos bruscos propios de la sesión de entrenamiento.

### **3.1 CONSTRUCCIÓN DEL MÓDULO DEL ATLETA.**

Tomando como referencia los requerimientos descritos en el Capítulo 2 del presente documento y luego de haber determinado las partes que componen cada módulo, se procede a detallar la integración de los bloques que constituyen el módulo del atleta, siendo estos los siguientes:

- Bloque del Microcontrolador.
- Bloque de Frecuencia Cardiaca.
- Bloque de Temperatura.
- Bloque de Transmisión y Recepción.
- Bloque de Alimentación.

El diagrama del circuito que muestra las conexiones entre los diferentes bloques del módulo del atleta se muestra en la Figura 3.1.

El circuito se construyó en base a pruebas realizadas en forma individual a los bloques del módulo del atleta. Adicionalmente, por motivos de que el módulo estará sometido a los movimientos del deportista la placa de circuito impreso será construida a doble lado y cada componente electrónico será sujeto con doble punto de soldadura dándole mayor soporte. Estas pruebas se describen a continuación.

#### **3.1.1 PRUEBAS INDIVIDUALES DE LOS BLOQUES DEL MÓDULO DEL ATLETA.**

##### **3.1.1.1 Bloque del Microcontrolador.**

Teniendo en cuenta que el bloque del microcontrolador es el componente central que administra la información que le brindan los demás bloques, su funcionamiento

está ligado a cada uno de los demás. Por lo tanto el bloque del microcontrolador se describirá en conjunto.

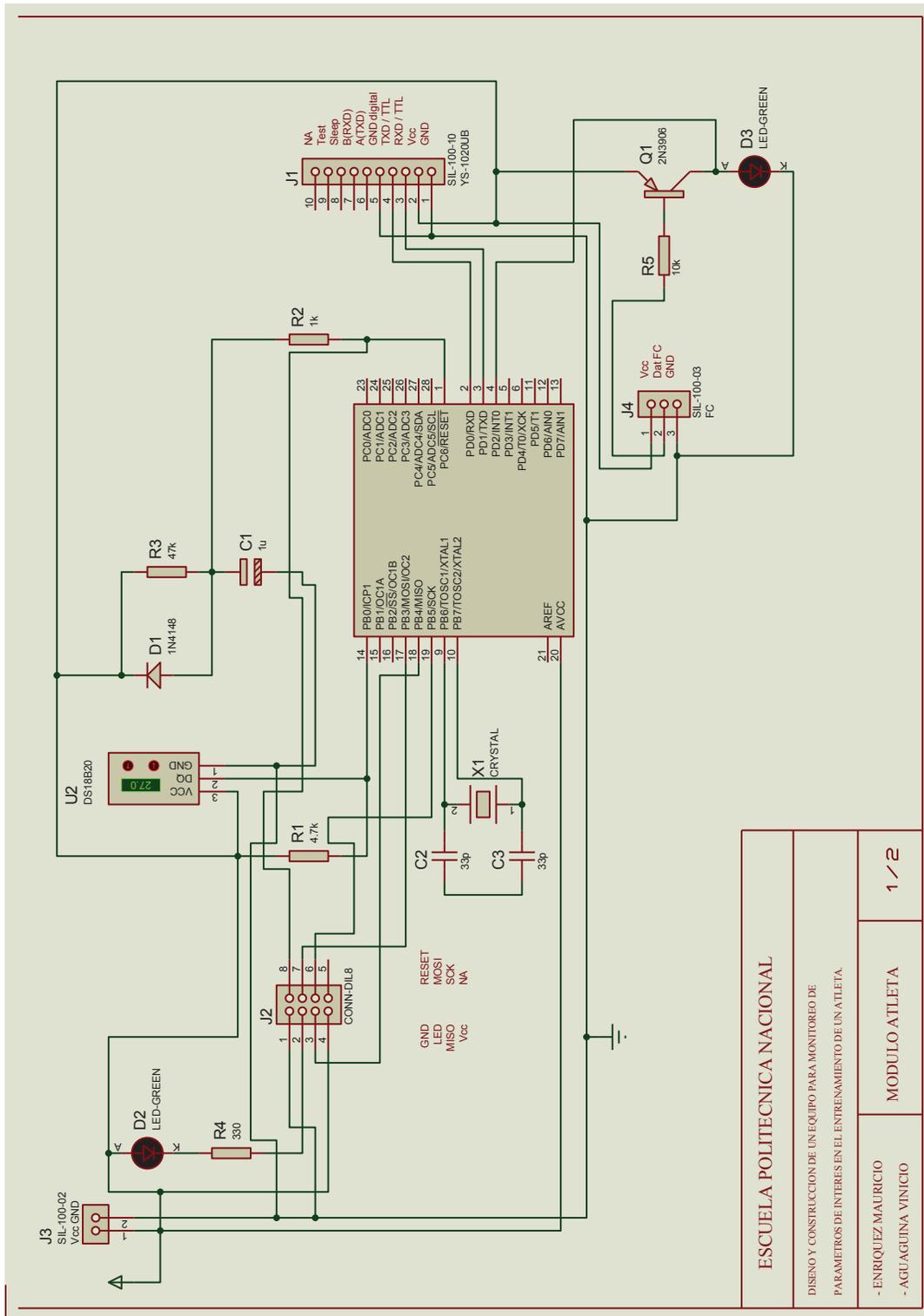


Figura 3. 1 Circuito del Módulo del Atleta.

### 3.1.1.2 Bloque de Frecuencia Cardiaca

El bloque de frecuencia cardiaca se armó según el circuito mostrado en la Figura 2.5.

- **Primero:** Se realizó un test de funcionamiento del *PulseSensor* empleando un *switch* en base a un transistor en un *protoboard*. Se observó un adecuado funcionamiento en cuanto a la obtención de los pulsos de frecuencia cardiaca corroborado por el encendido y apagado del led del circuito (Ver la Figura 3.2).

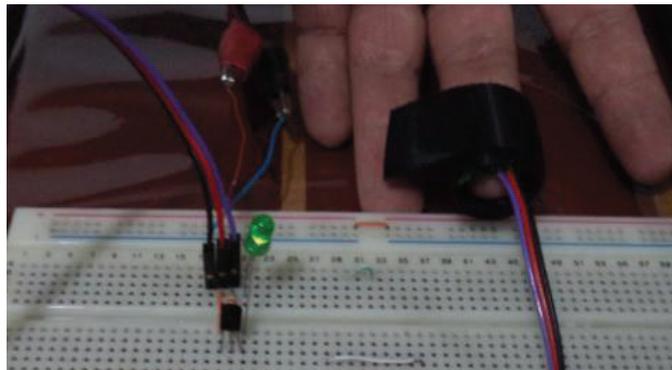


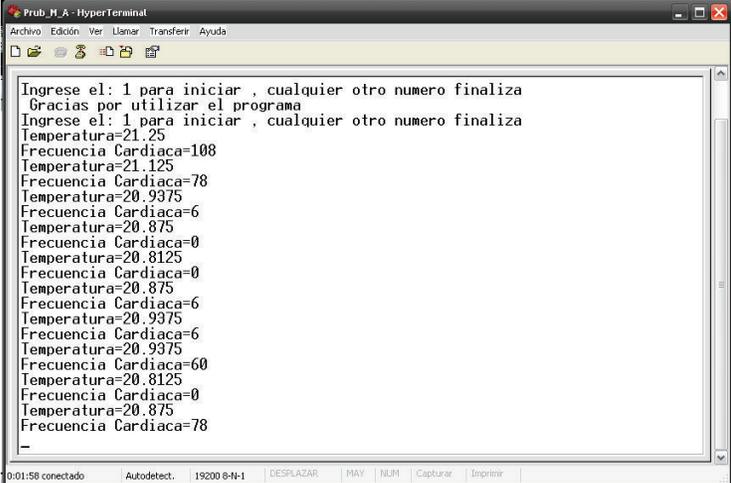
Figura 3. 2 Prueba en *Protoboard* del *PulseSensor* y *Switch*.

- **Segundo:** Utilizando el circuito prototipo del entrenador de microcontroladores se procedió a probar, mediante un pulsador conectado al puerto del microcontrolador PD.2/INT0, el ingreso manual de los pulsos de frecuencia cardiaca, como se muestra en la Figura 3.3.



Figura 3. 3 Prueba de Ingreso de Pulsos al Microcontrolador con un Pulsador.

- **Tercero:** En el microcontrolador se programó la subrutina que limita el tiempo a 10 segundos el tiempo de muestreo de pulsos ingresados manualmente. Los resultados fueron obtenidos correctamente y se enviaron mediante una comunicación serial hacia el computador, visualizando los datos a través de una sesión de *hyper terminal*. (Ver Figura 3.4)



```

Ingrese el: 1 para iniciar , cualquier otro numero finaliza
Gracias por utilizar el programa
Ingrese el: 1 para iniciar , cualquier otro numero finaliza
Temperatura=21.25
Frecuencia Cardiaca=108
Temperatura=21.125
Frecuencia Cardiaca=78
Temperatura=20.9375
Frecuencia Cardiaca=6
Temperatura=20.875
Frecuencia Cardiaca=0
Temperatura=20.8125
Frecuencia Cardiaca=0
Temperatura=20.875
Frecuencia Cardiaca=6
Temperatura=20.9375
Frecuencia Cardiaca=6
Temperatura=20.9375
Frecuencia Cardiaca=60
Temperatura=20.8125
Frecuencia Cardiaca=0
Temperatura=20.875
Frecuencia Cardiaca=78

```

**Figura 3. 4** Resultados mostrados en PC en sesión de *Hyper Terminal* de Ingreso de Pulsos al Microcontrolador.

- **Cuarto:** El *PulseSensor* viene con una pinza metálica para la sujeción al lóbulo de la oreja del atleta como punto de medición de los pulsos. La pinza al tener un color metalizado daba problemas cuando se trabajaba bajo la luz del día y se procedió a pintarla de color negro para evitar los rebotes de luz que afectan la medición de los pulsos (Ver Figuras 3.5 y 3.6).



**Figura 3. 5** Pinza de Sujeción.

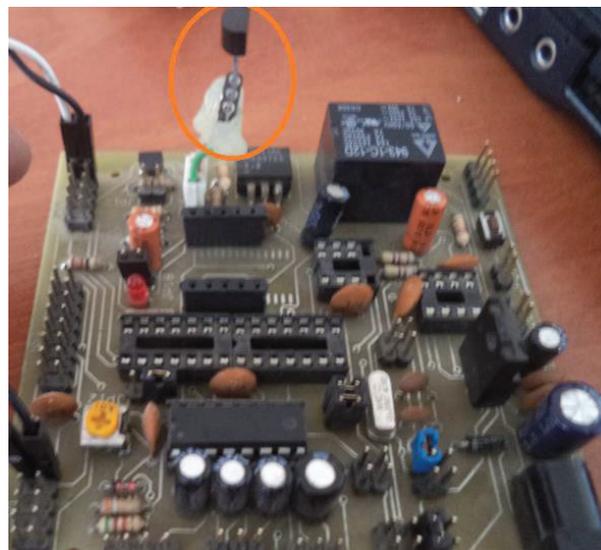


**Figura 3. 6** *PulseSensor* en Lóbulo de la Oreja.

- **Quinto:** Comprobado el funcionamiento descrito en los pasos anteriores, se procedió a realizar pruebas con el *PulseSensor*, *switch* y microcontrolador sobre el circuito prototipo, de lo cual se obtuvieron los resultados esperados.

### 3.1.1.3 Bloque de Temperatura.

En base al diseño del monitor de temperatura corporal, en el microcontrolador se realizó la programación para declarar el puerto PB.0 para que realice la lectura del sensor de temperatura DS18B20, con línea de comunicación de datos 1wire; de la misma manera se desarrolló e implementó la subrutina “temperatura”. Las pruebas de funcionamiento se las realizó en el circuito prototipo de entrenamiento de microcontroladores de acuerdo a lo mostrado en la Figura 3.7.



**Figura 3. 7** Entrenador de Microcontrolador con el Sensor de Temperatura DS18B20.

Los resultados se obtuvieron de la misma manera en un PC y visualizando las mediciones del sensor de temperatura en *hyper terminal*, de acuerdo a lo presentado en la Figura 3.4.

Cabe recalcar que para almacenar el dato entregado por el sensor, se pueden declarar dos tipos de variables, una como byte y la otra como single; la primera almacenará solo valores enteros de la lectura, mientras que la segunda presentará valores con decimales de la lectura. Para el proyecto se seleccionó la variable single.

#### 3.1.1.4 Bloque de Transmisión y Recepción.

Para la comunicación entre *transceivers* se utiliza al microcontrolador como el sistema de control de envío y recepción de información, para lo cual se utilizan los siguientes comandos:

➤ **Comandos de transmisión:**

➤ **Print var**

Enviar el dato con la terminación de retorno de carro CR (ASCII 13) y fin de línea LF (ASCII 10)

➤ **Print var;**

Enviar los datos sin los comandos de terminación de carro y sin fin de línea.

➤ **Print var; chr(13);**

Enviar los datos solo con el retorno de carro.

➤ **Print var; chr(10);**

Enviar los datos solo con el fin de línea.

➤ **Comandos de Recepción:**

➤ **Input**

Esperar un dato hasta encontrar el carácter de retorno de carro.

➤ **Input bin**

Esperar un dato sin necesidad del comando de retorno de carro.

➤ **Inkey**

Retornar el valor ASCII del primer carácter en el buffer de entrada.

➤ **Interrupciones:**

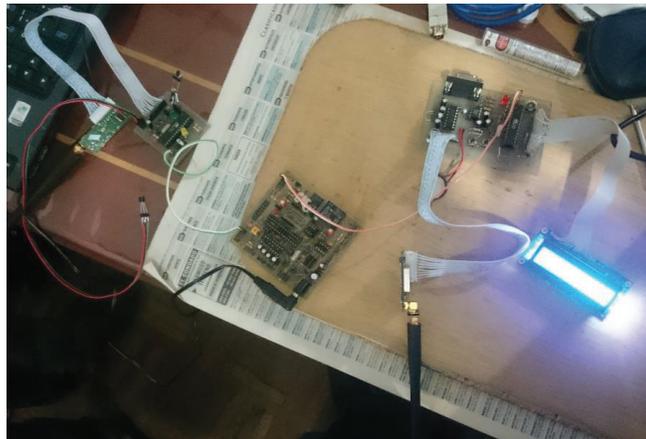
➤ **URXC**

Habilitar la interrupción de recepción en el puerto serial.

➤ **UTXC**

Habilita la interrupción de transmisión en el puerto serial.

Se realizó pruebas con la transmisión de un módulo con los comandos **Print var; chr(13);**, mientras que el otro módulo activaba la interrupción de recepción con el comando **URXC**, que llamaba a una subrutina en la cual se utilizó el comando **Input bin**. Al poner en funcionamiento los módulos de transmisión y recepción la comunicación entre ellos fue exitosa y se obtuvieron resultados en el LCD del lado del entrenador, los mismos que se presentaron como caracteres ASCII (Ver Figuras 3.8 y 3.9).



**Figura 3. 8** Pruebas de los *Transceivers* para Transmisión y Recepción de Datos.



**Figura 3. 9** Resultado de los Datos Transmitidos y Recibidos como Caracteres ASCII.

Como se puede observar en la Figura 3.9 se tiene un despliegue de los datos enviados desde el módulo del atleta con el siguiente formato: 1(cr13),FC(cr13),T°(cr13),2(cr13)

Se puede verificar que están llegando los datos de:

1 = Carácter ASCII 49

Retorno de carro = Carácter ASCII 13

FC = valor 0 = Carácter ASCII 48

Retorno de carro = Carácter ASCII 13

T° = 20.8 = Caracteres ASCII 50-48-46-56

Con los datos que se obtuvieron se concluyó que los datos transmitidos fueron los que se esperaba en la recepción.

#### 3.1.1.5 Bloque de Alimentación.

Se realizó las pruebas con una fuente de alimentación DC regulada a 5 voltios y con una batería marca SONY tipo N de 3.6 voltios y que entrega una corriente de 630 mAh.

La batería Sony tipo N proporcionó una independencia de funcionamiento continuo del módulo del atleta en un periodo de tiempo aproximado de 2 horas. En la Figura 3.10 se muestra la batería descrita.

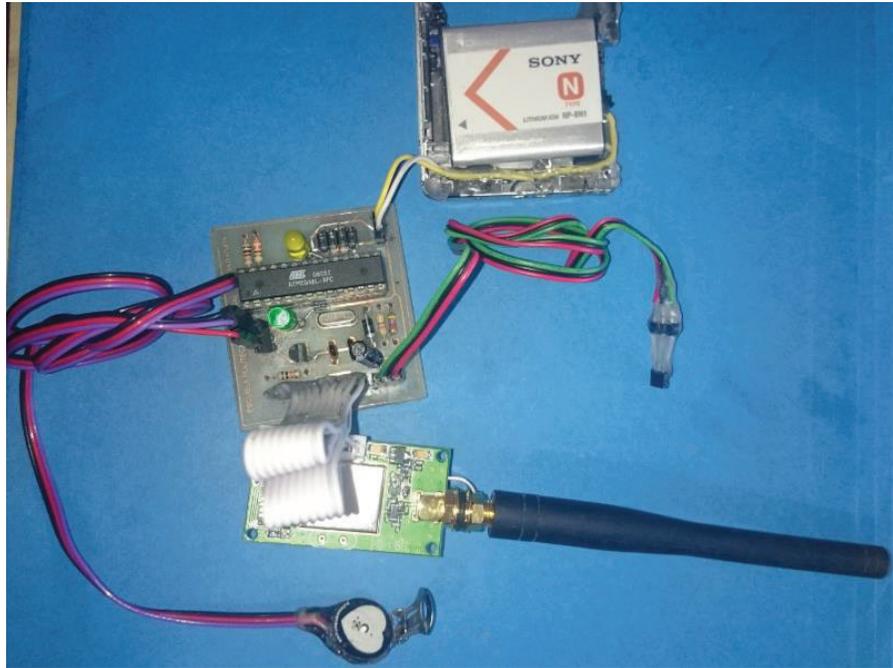


**Figura 3. 10** Batería SONY tipo N.

#### 3.1.2 CONSTRUCCIÓN DEL CIRCUITO FINAL DEL ATLETA.

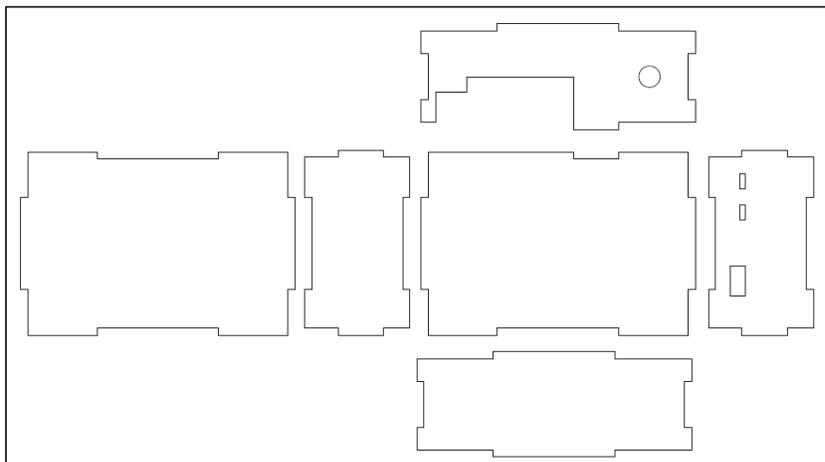
En base a las pruebas realizadas individualmente de los bloques, se procedió a fabricar el circuito final, el mismo que se muestra en la Figura 3.11.

En este circuito se realizaron las mismas pruebas individuales, teniendo resultados adecuados.



**Figura 3. 11** Circuito Fabricado del Módulo del Atleta.

Para la construcción de la caja que protege al circuito Impreso y sus elementos se consideró un material acrílico, para lo cual se tomaron las medidas de los elementos que estarían dentro de la caja y su disposición. Se realizó un plano de la caja en AUTOCAD y con la ayuda de un plotter de corte laser se obtuvo las piezas que conforman el armazón de la caja (Ver Figura 3.12)

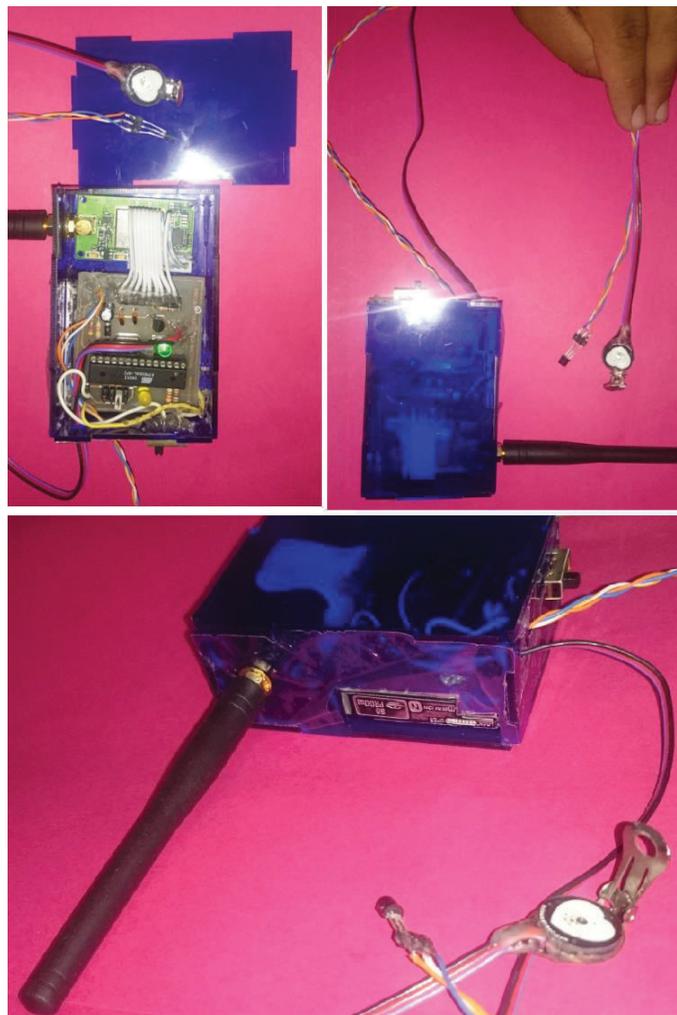


**Figura 3. 12** Piezas que conforman la Caja del Módulo del Atleta.

Para unir las piezas de la caja se utilizó pegamento de contacto UHU (extra fuerte), el mismo que por sus características permite firmeza de adhesión a las paredes de la caja, así como también es resistente a la humedad. La resistencia a la humedad del pegamento en las paredes de la caja evitará que la sudoración del atleta afecte al circuito interno.

Debido a que los cables de los sensores de frecuencia cardíaca y temperatura en pruebas de campo estarán expuestas a una manipulación y movilidad, se ha colocado pegamento de silicón en las uniones de los cables en ambos extremos, lo que ayuda en firmeza de las uniones e impermeabilización de los elementos.

La caja final y los componentes del Módulo del Atleta se presentan en la Figura 3.13



**Figura 3. 13** Módulo del Atleta terminado.

### 3.1.3 DIAGRAMA DE FLUJO DE FUNCIONAMIENTO DEL MÓDULO DEL ATLETA.

En base a las pruebas realizadas con los bloques del módulo del atleta en cuanto a hardware y software, se procede a presentar el diagrama de flujo del funcionamiento integral de los bloques del módulo en la Figura 3.14.

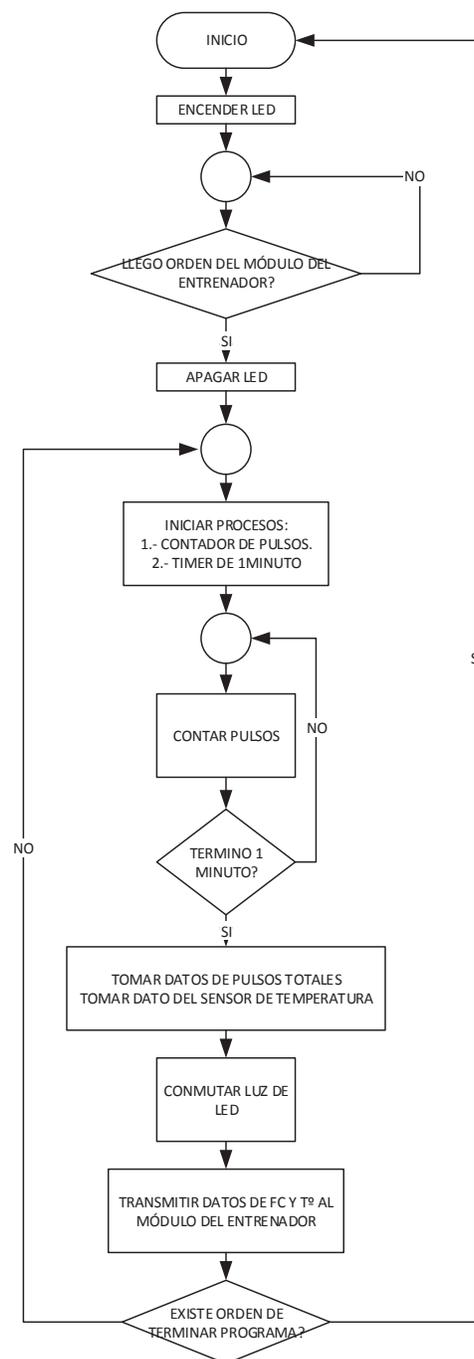


Figura 3. 14 Diagrama de funcionamiento del Módulo del Atleta.

## Descripción

El programa realiza los siguientes pasos:

➤ **Encender LED:**

Encender Módulo del Atleta.

Encender LED.

**Fin de tarea.**

➤ **Llego orden del Módulo del Entrenador:**

Iniciar interrupción de puerto serial a la espera de la orden de inicio que será dada por el módulo del entrenador.

- Si no recibe orden:

Mantener Módulo del Atleta a la espera de una orden del Módulo del Entrenador.

- Si recibe orden:

Reiniciar interrupción de puerto serial a la espera de nueva orden.

Continuar con la siguiente tarea.

**Fin de tarea.**

➤ **Apagar LED:**

Recibir orden de inicio desde el Módulo del Entrenador.

Apagar LED.

**Fin de tarea.**

➤ **Inicias Procesos:**

**Contar Pulsos:**

Iniciar conteo de pulsos del atleta.

**Iniciar *Timer*:**

Iniciar *timer* seteado a 1 minuto.

**Fin de tarea.**

➤ **Contar Pulsos:**

Detectar por interrupción externa pulso.

Incrementar valor de pulsos.

Esperar que *Timer* finalice su periodo de tiempo de 1 minuto.

**Fin de tarea.**

➤ **Terminó 1 minuto:**

Si no termina 1 minuto

Continuar esperando.

Si termina un minuto

Activar bandera.

**Fin de tarea.**

➤ **Tomar datos:**

Verificar bandera activa

Leer valor de pulsos totales

Leer valor de temperatura.

**Fin de tarea.**

➤ **Conmutar LED:**

Encender LED apagado. Ó

Apagar LED encendido.

**Fin de tarea.**

➤ **Transmisión de datos:**

Deshabilitar interrupción de puerto serial

Transmitir

Valor de pulsos contados

Valor de temperatura.

Reiniciar contador de pulsos.

Reiniciar *timer* 1 minuto.

Habilitar interrupción de puerto serial

**Fin de tarea.**

➤ **Existe orden terminar programa:**

Verificar orden del módulo del entrenador que indique fin de toma de datos.

- Si no existe orden

Continuar esperando orden.

- Si existe orden

Terminar toma de datos.

Volver a esperar orden de inicio.

**Fin de tarea.**

### **3.2 CONSTRUCCIÓN DEL MÓDULO DEL ENTRENADOR.**

De la misma manera, tomando en cuenta los diseños descritos en el Capítulo 2, se procede a construir e integrar los bloques que constituyen módulo del entrenador, siendo estos los siguientes:

- Bloque del Microcontrolador.
- Bloque del Reloj, Ingreso y Selección de Datos.
- Bloque de Visualización en LCD.
- Bloque de Transmisión y Recepción.
- Bloque de Comunicación con el PC.
- Bloque de Alimentación.

El diagrama del circuito que muestra las conexiones entre los diferentes bloques del módulo del entrenador se encuentra en la Figura 3.15.

El circuito se construyó realizando pruebas en forma individual de los bloques del módulo del entrenador hasta que cumplan los requerimientos descritos en el Capítulo 2 en un circuito prototipo correspondiente a un entrenador de microcontroladores. Estas pruebas se describen a continuación.

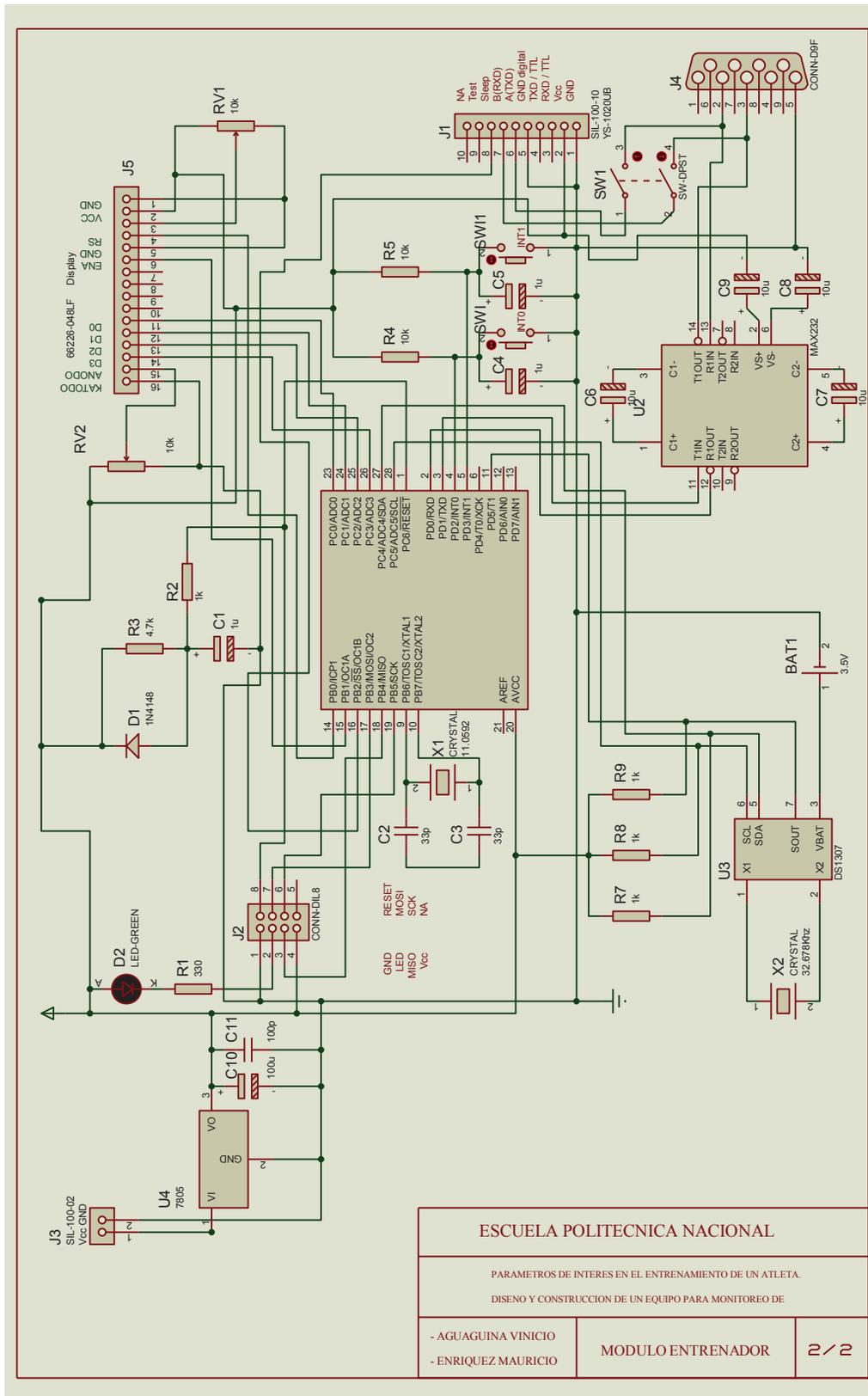


Figura 3. 15 Circuito del Módulo del Entrenador.

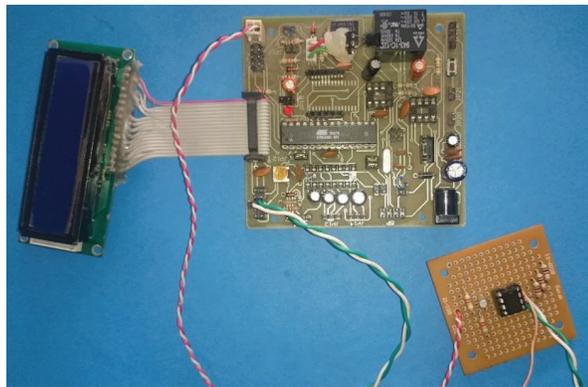
### 3.2.1 PRUEBAS INDIVIDUALES DE LOS BLOQUES DEL MÓDULO DEL ENTRENADOR.

#### 3.2.1.1 Bloque del Microcontrolador

Teniendo en cuenta que el bloque del microcontrolador es el componente central que administra la información que le brindan los demás bloques, esto implica que el funcionamiento del microcontrolador está ligado a cada uno de los demás bloques y se describirá en conjunto.

#### 3.2.1.2 Bloques del Reloj, Ingreso y Selección de Datos.

De acuerdo al circuito mostrado en la Figura 2.9 y en base al diseño del reloj RTC DS1307, se procedió a conectar los pines 5/SDA y 6/SCL a los puertos del microcontrolador PC.4/SDA y PC.5/SCL. Se realizó pruebas con el uso del entrenador de microcontroladores (Ver Figura 3.16).



**Figura 3. 16** Prueba de Funcionamiento del Reloj DS1307.

La configuración del reloj en cuanto a hora y fecha se la realizó conforme a lo indicado en el diseño de la interfaz de ingreso y selección de datos.

Los resultados indicados en el LCD fueron correctos, los cuales son mostrados en Figura 3.17



**Figura 3. 17** Resultados del Reloj en LCD.

### 3.2.1.3 Bloque de Visualización en LCD.

Se utilizó las líneas de comando ya descritas para probar el Bloque del LCD 16\*2. En las pruebas preliminares se observó que al conectar las líneas de datos desde el microcontrolador hacia el LCD (Db4, Db5, Db6 y Db7) en pines que pertenecen a puertos diferentes, no se mostraba la información, por lo tanto las salidas hacia el LCD deben estar conectadas en pines del mismo puerto del microcontrolador para resultados positivos. Las líneas de control del LCD pueden estar conectadas a un puerto diferente.

Los resultados obtenidos del manejo del LCD se muestran de igual forma en la Figura 3.17.

### 3.2.1.4 Bloque de Transmisión y Recepción.

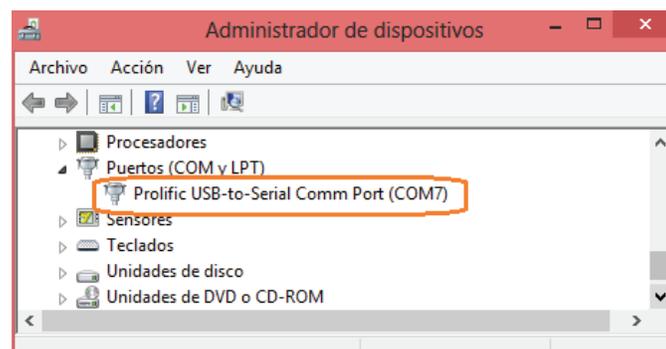
Este bloque ya fue descrito en la construcción del módulo del atleta, ya que los *transceivers* trabajan en conjunto como Transmisor y Receptor.

### 3.2.1.5 Bloque de Comunicación con el PC.

Para probar las partes que componen el circuito mostrado en la Figura 2.12 y luego de su construcción se realizaron las siguientes pruebas:

En primer lugar se probó el cable adaptador USB a RS-232, para lo cual primero se instaló el driver que para el caso del sistema operativo de Windows 8 fue el driver PL2303 de 64 bits.

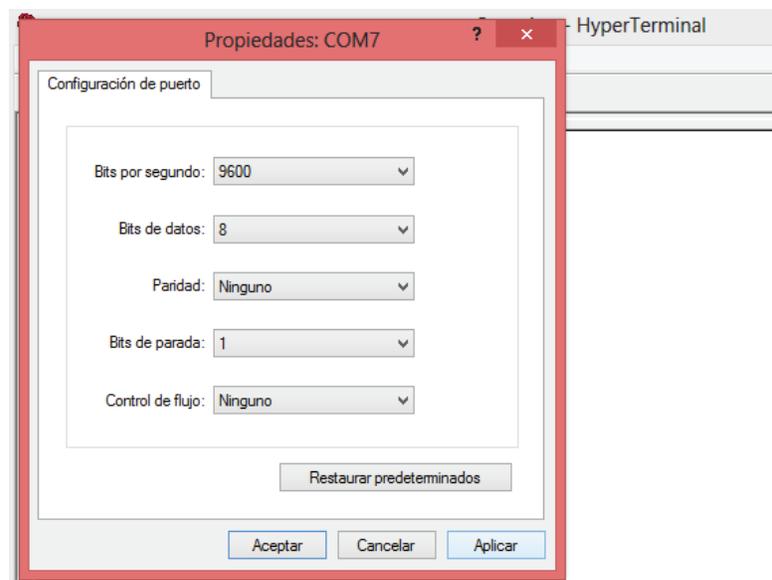
Una vez instalado el driver, al conectar el adaptador al computador reconoce que tiene conectado un dispositivo en el puerto COM, según muestra la Figura 3.18 en referencia al Administrador de Dispositivos del sistema operativo Windows 8.



**Figura 3. 18** Puerto COM detectando con el adaptador USB a RS-232.

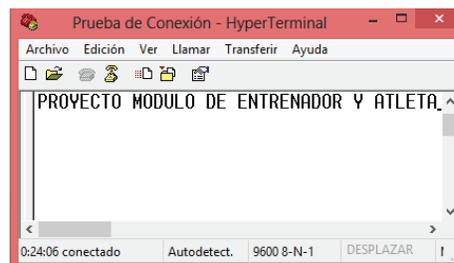
Es importante tener en cuenta que para versiones de sistemas operativos Windows 7 y versiones superiores a este, la empresa *Microsoft* ya no trae consigo la aplicación para desplegar una sesión de *hyper terminal*. Por lo tanto fue necesario descargar la aplicación vía internet.

Una vez conectado el adaptador se procede a iniciar una sesión de *hyper terminal* en el computador, configurando la conexión del puerto COM como se indica en la Figura 3.19. En el cable adaptador del lado del conector DB-9 es necesario realizar un puente entre los pines 2 y 3 (Tx y Rx).



**Figura 3. 19** Configuración de Puerto COM para Sesión de *Hyper terminal*.

Los resultados del funcionamiento del adaptador USB a RS-232 son mostrados en la figura 3.20.



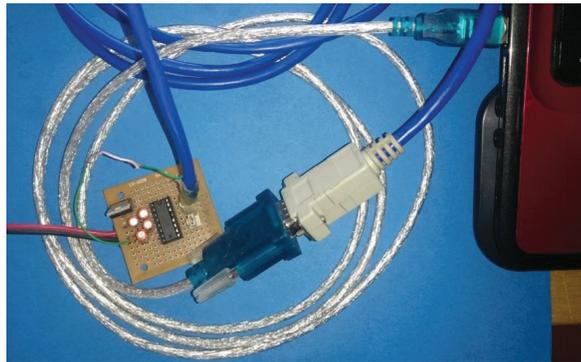
**Figura 3. 20** Resultados de *Hyper terminal* del Cable Adaptador USB a RS-232.

En la Figura 3.21 se muestra también la conexión física realiza del adaptador y computador.



**Figura 3. 21** Conexión Cable Adaptador USB a RS-232 a PC.

En segundo lugar se prueba con el circuito integrado MAX 232 con el cable adaptador USB a RS-232 y el PC. (Ver Figura 3.22)



**Figura 3. 22** Pruebas de MAX232 – Adaptador USB a RS232 – PC.

Cabe recalcar que para la comunicación serial entre diferentes dispositivos que tienen el estándar RS-232 para realizar la conexión física se requiere de un cable cruzado.

#### 3.2.1.6 Bloque de Alimentación.

Se realizó las pruebas en base a fuente de alimentación DC regulada a 5 voltios. Para independizar el circuito y obtener movilidad, se probó con una batería marca GP 200 de características:

- 8.4 voltios
- Corriente de 200 mAh

Esta batería se utilizó como entrada al CI regulador de Voltaje LM7805 y se obtuvo el voltaje de salida de 5 voltios.

La batería GP 200 proporcionó una independencia de funcionamiento continuo del módulo del entrenador en un periodo de tiempo aproximado de más de 2 horas. En la Figura 3.23 se muestra la batería descrita.

Se utilizó esta batería por razones de que el circuito del entrenador tiene elementos que trabajan sobre los 4 voltios de alimentación DC.

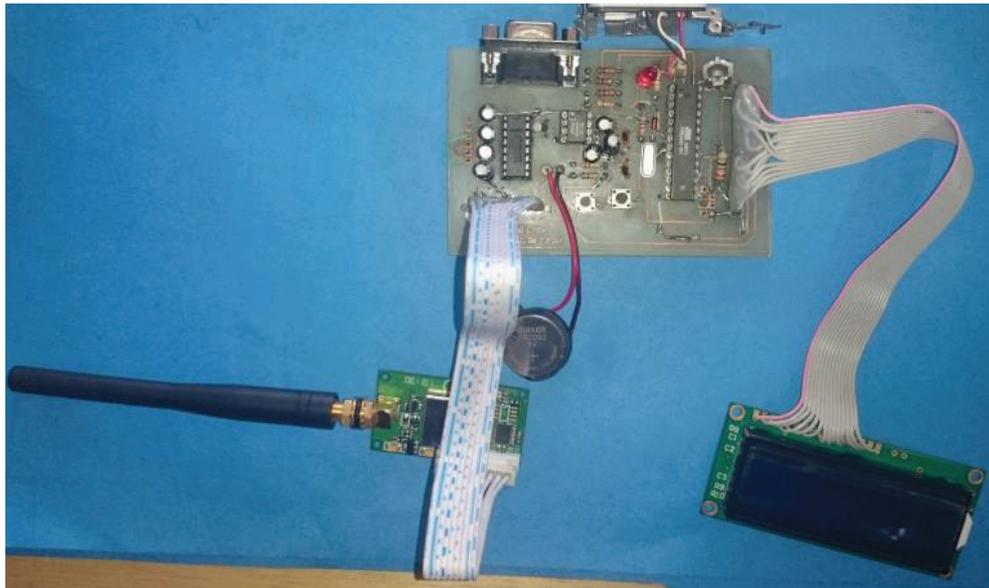


**Figura 3. 23** Batería GP 200.

### 3.2.2 CONSTRUCCIÓN DEL CIRCUITO FINAL DEL ENTRENADOR.

En base a las pruebas realizadas individualmente de los bloques, se procedió a fabricar el circuito final, el mismo que se muestra en la Figura 3.24.

En este circuito se realizaron las mismas pruebas individuales, teniendo resultados positivos.



**Figura 3. 24** Circuito Fabricado del Módulo del Entrenador.

Para la construcción de la caja que protege al circuito Impreso y sus elementos, se tomó las mismas consideraciones y proceso de fabricación utilizados en el Módulo del Atleta. Así, las piezas que conforman el armazón de la caja se muestran en la Figura 3.25



**Figura 3. 25** Piezas que conforman la Caja del Módulo del Entrenador.

La caja final y los componentes del Módulo del Entrenador se presentan en la Figura 3.26

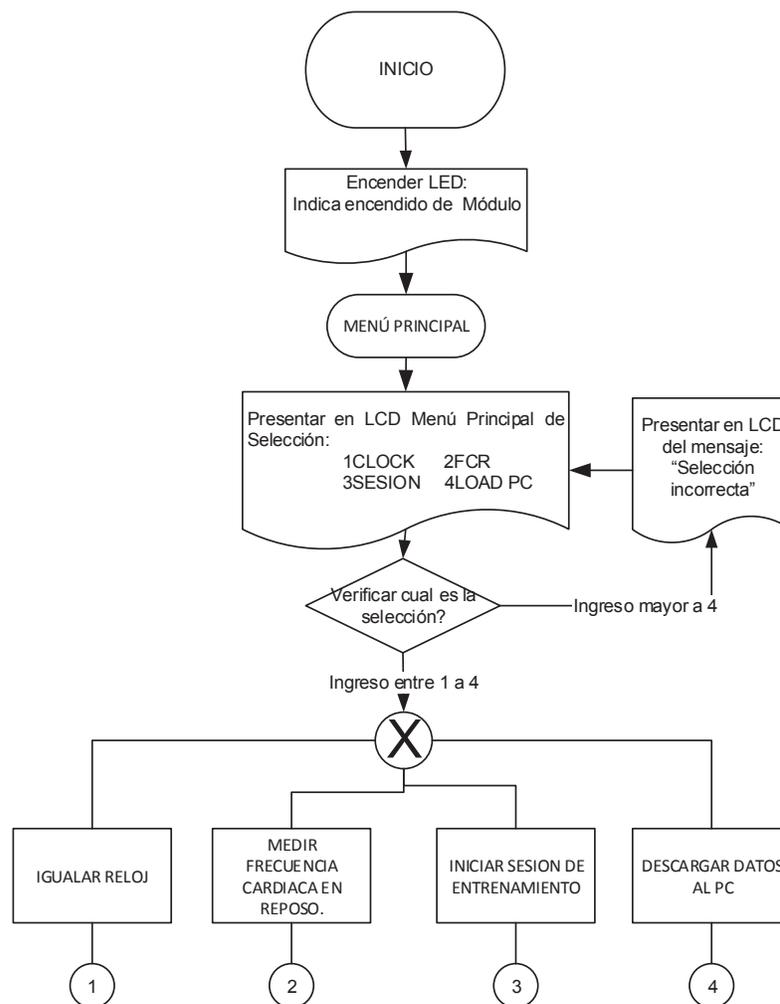


**Figura 3. 26** Módulo del Entrenador terminado.

### 3.2.3 DIAGRAMA DE FLUJO DE FUNCIONAMIENTO DEL MÓDULO DEL ENTRENADOR.

En base a las pruebas realizadas con los bloques del módulo del entrenador en cuanto a *hardware* y *software*, se ha visto la necesidad de integrar un menú de selección para los diferentes procesos que realiza este módulo, el mismo que se presenta en la Figura 3.27.

El programa realiza los siguientes pasos:



**Figura 3. 27** Diagrama para la Selección de Procesos del Módulo del Entrenador.

➤ **Encender de LED:**

Encender Módulo del Entrenador.

Encender LED.

**Fin de tarea.**

➤ **Presentar en LCD:**

Presentar Menú Principal:

<b>1Clock</b>	<b>2FCR</b>
<b>3Sesion</b>	<b>4Load PC</b>

**Fin de tarea.**

➤ **Verificar cual es la Selección:**

Identificar selección ingresada por usuario.

- Si la selección es 1  
Igualar reloj.
- Si la selección es 2  
Frecuencia cardiaca en reposo.
- Si la selección es 3  
Sesión de Entrenamiento.
- Si la selección es 4  
Descargar al computador.
- Si la selección es mayor a 4  
Presentar mensaje "Selección incorrecta".  
Presentar el Menú Principal.

**Fin de tarea.**

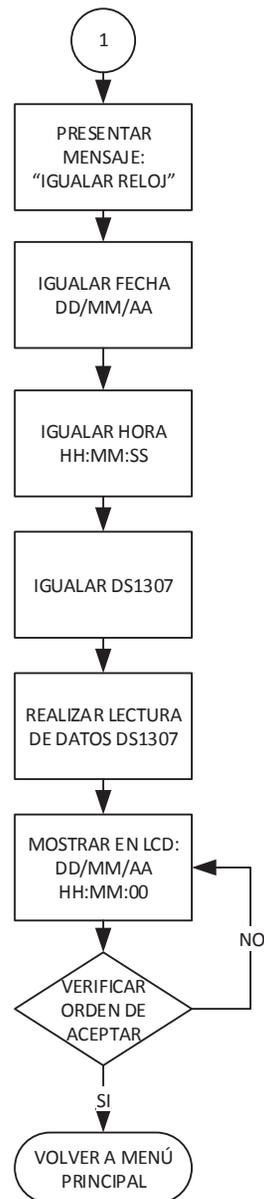
En la Figura 3.28 muestra al LCD con el menú principal de opciones a desplegar.



**Figura 3. 28** Menú a mostrarse en el LCD.

### 3.2.3.1 Diagrama de Flujo de Funcionamiento de la Opción de Menú 1Clock

En el momento en el que el usuario elige la opción 1Clock, el microcontrolador realizará el proceso relacionado a igualar la hora y la fecha del sistema. Es necesario indicar que todos los valores ingresados por el usuario del módulo a través de los pulsadores se presentarán en el LCD. El diagrama en cuestión se presenta en la Figura 3.29.



**Figura 3. 29** Diagrama de Funcionamiento para igualar el Reloj.

El programa realizará los siguientes pasos:

➤ **Presentar Mensaje:**

Mostrar mensaje "IGUALAR RELOJ" durante 3 segundos.

**Fin de tarea.**

➤ **Igualar Fecha:**

Mostrar mensaje en LCD "Ingrese Día"

Esperar ingreso del dato.

Confirmar ingreso de dato.

Almacenar variable.

Mostrar mensaje en LCD "Ingrese Mes"

Esperar ingreso del dato.

Confirmar ingreso de dato.

Almacenar variable.

Mostrar mensaje en LCD "Ingrese Año"

Esperar ingreso del dato.

Confirmar ingreso de dato.

Almacenar variable.

**Fin de tarea.**

➤ **Igualar Hora:**

Mostrar mensaje en LCD "Ingrese Hora"

Esperar ingreso del dato.

Confirmar ingreso de dato.

Almacenar variable.

Mostrar mensaje en LCD "Ingrese Minutos"

Esperar ingreso del dato.

Confirmar ingreso de dato.

Almacenar variable.

**Fin de tarea.**

➤ **Igualar DS1307:**

Configurar reloj con datos ingresados.

Mostrar en LCD mensaje “Reloj Igualado”

**Fin de tarea.**

➤ **Realizar lectura de datos DS1307:**

Leer datos de fecha y hora del reloj.

**Fin de tarea.**

➤ **Mostrar en LCD:**

Presentar en LCD.

DD/MM/AA SALIR

HH:MM:SS ACEPTAR

**Fin de tarea.**

➤ **Volver al menú Principal:**

Verificar orden aceptar:

- Si no existe aceptar

Leer datos de reloj.

Mostrar en LCD

DD/MM/AA SALIR

HH:MM:SS ACEPTAR

- Si existe aceptar

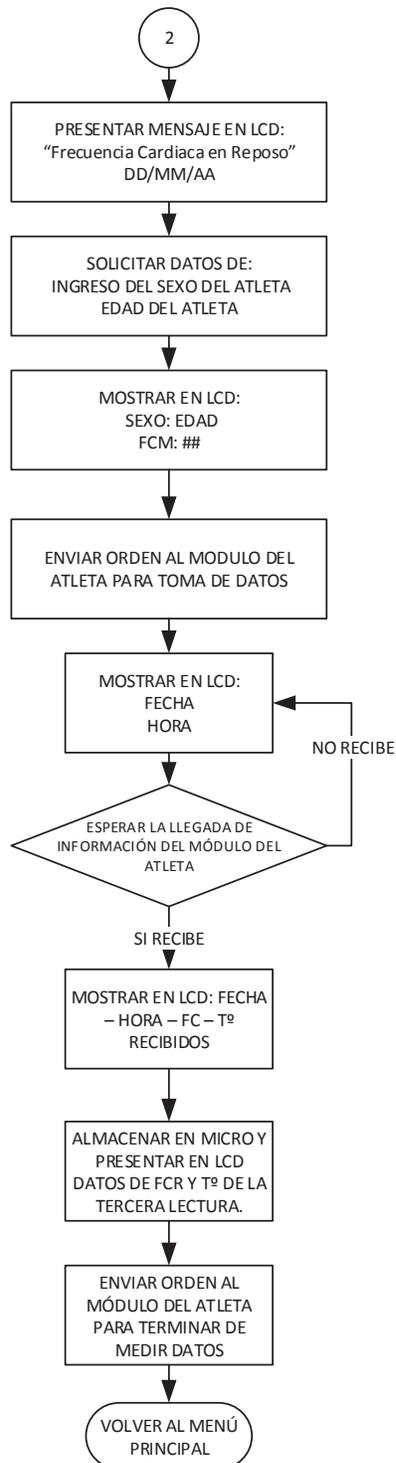
Volver a Menú Principal

**Fin de tarea.**

### 3.2.3.2 Diagrama de Flujo de Funcionamiento de la opción de menú 2FCR (Frecuencia Cardíaca en Reposo)

Este proceso hace referencia a la orden enviada desde el módulo del entrenador al módulo del atleta para que este realice la acción de medir la frecuencia cardíaca y temperatura del deportista en reposo. El entrenador determinará bajo observación

del atleta el instante en el que el sistema tome la medida y guardará esta información. El diagrama a describir se muestra en la Figura 3.30.



**Figura 3. 30** Diagrama de Funcionamiento para Medir la FCR.

El programa realiza los siguientes pasos:

➤ **Presentar mensaje en LCD:**

Leer fecha del reloj.

Mostrar mensaje:

“Frecuencia Cardiaca en Reposo”

DD/MM/AA.

**Fin de tarea.**

➤ **Solicitar datos:**

Ingresar sexo del atleta

- Si dato ingresado es 1 femenino.
- Si dato ingresado es 2 masculino.

Ingresar la Edad del Atleta.

Calcular de la frecuencia cardiaca máxima FCM, según las siguientes ecuaciones:

$$\textit{Hombres} = FCM = 220 - \textit{edad}$$

$$\textit{Mujeres} = FCM = 226 - \textit{edad}$$

**Fin de tarea.**

➤ **Presentar en LCD durante 3 segundos los datos:**

Sexo: edad a

FCM=

**Fin de tarea.**

➤ **Enviar orden:**

Transmitir orden de inicio de toma de datos al Módulo del Atleta.

**Fin de tarea.**

➤ **Mostrar en LCD:**

Realizar Lectura del DS1307.

Mostrar datos de:

DD/MM/AA

HH:MM:SS

**Fin de tarea.**

➤ **Esperar información:**

Esperar datos medidos desde el Módulo del Atleta de FC y T°

- Si no recibe información:

Leer reloj

Mostrar en LCD:

DD/MM/AA

HH:MM:SS

- Si recibe información, continuar con la siguiente acción.

**Fin de tarea.**

➤ **Mostrar en LCD:**

Mostrar mensaje de datos recibidos:

DD/MM/AA FC=

HH:MM:SS T=

**Fin de tarea.**

➤ **Almacenar en Micro y Presentar en LCD:**

Almacenar datos recibidos de FC y T°.

Mostrar en LCD durante 8 segundos:

FCR=

T[C]=

Guardar datos en EEPROM

**Fin de tarea.**

➤ **Enviar orden al Módulo del Atleta:**

Transmitir orden de parar la medición de datos al Módulo del Atleta.

**Fin de tarea.**

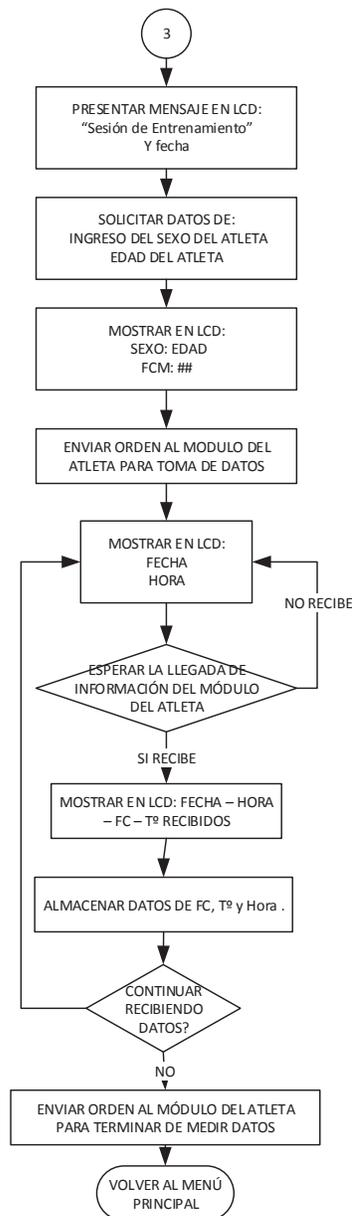
➤ **Volver al Menú Principal:**

Retornar al Menú Principal de opciones de programa.

**Fin de tarea.**

### 3.2.3.3 Diagrama de Flujo de Funcionamiento de la opción de menú 3Sesion (Sesión de Entrenamiento)

Este proceso hace referencia a la orden enviada desde el módulo del entrenador al módulo del atleta para que este realice la acción de medir la frecuencia cardiaca y temperatura durante la sesión principal de entrenamiento. El diagrama a describir se muestra en la Figura 3.31.



**Figura 3. 31** Diagrama de Funcionamiento para toma de Datos en Sesión de Entrenamiento.

El programa realiza los siguientes pasos:

➤ **Presentar mensaje en LCD:**

Leer fecha del reloj.

Mostrar mensaje:

“Sesión de Entrenamiento”

DD/MM/AA.

**Fin de tarea.**

➤ **Solicitar datos:**

Ingresar sexo del atleta

- Si dato ingresado es 1 femenino.
- Si dato ingresado es 2 masculino.

Ingresar la Edad del Atleta.

Calcular de la frecuencia cardiaca máxima FCM, según las siguientes ecuaciones:

$$\text{Hombres} = FCM = 220 - \text{edad}$$

$$\text{Mujeres} = FCM = 226 - \text{edad}$$

**Fin de tarea.**

➤ **Presentar en LCD durante 3 segundos los datos:**

Sexo: edad a

FCM=

**Fin de tarea.**

➤ **Enviar orden:**

Transmitir orden de inicio de toma de datos al Módulo del Atleta.

**Fin de tarea.**

➤ **Mostrar en LCD:**

Realizar Lectura del DS1307.

Mostrar datos de:

DD/MM/AA

HH:MM:SS

**Fin de tarea.**

➤ **Esperar información:**

Esperar datos medidos desde el Módulo del Atleta de FC y T°

- Si no recibe información:

Leer reloj

Mostrar en LCD:

DD/MM/AA

HH:MM:SS

- Si recibe información, continuar con la siguiente acción.

**Fin de tarea.**

➤ **Mostrar en LCD:**

Mostrar mensaje de datos recibidos:

DD/MM/AA FC=

HH:MM:SS T=

**Fin de tarea.**

➤ **Almacenar en Micro y Presentar en LCD:**

Almacenar datos recibidos de FC y T°.

Mostrar en LCD durante 8 segundos:

FCR=

T[C]=

Guardar datos en EEPROM

**Fin de tarea.**

➤ **Continuar recibiendo datos:**

Verificar orden aceptar:

- Si no existe aceptar

Leer datos de reloj.

Mostrar en LCD

DD/MM/AA FC=

HH:MM:SS T=

- Si existe aceptar, continuar con la siguiente tarea.

**Fin de tarea.**

➤ **Enviar orden al Módulo del Atleta:**

Transmitir orden de parar la medición de datos al Módulo del Atleta.

**Fin de tarea.**

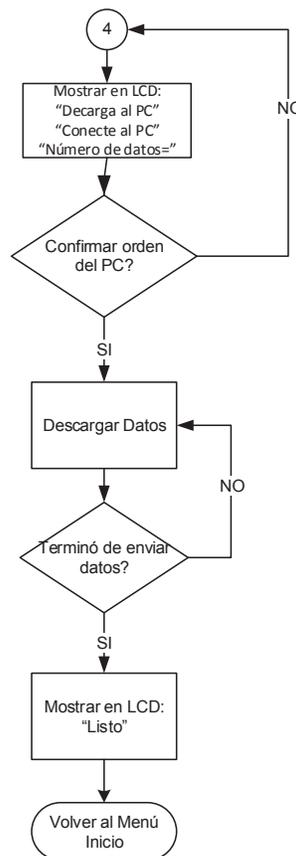
➤ **Volver al Menú Principal:**

Retornar al Menú Principal de opciones de programa.

**Fin de tarea.**

### 3.2.3.4 Diagrama de Flujo de Funcionamiento de la Opción de Menú 4Load PC.

Este proceso hace referencia a la opción 4 del menú principal, en la cual el entrenador al seleccionar esta opción se solicita al módulo iniciar la descarga hacia el PC de los datos almacenados, los cuales recibió anteriormente del módulo del atleta durante: la toma de datos en reposo y de la sesión de entrenamiento. El diagrama se muestra en la Figura 3.32.



**Figura 3. 32** Diagrama de Funcionamiento para la Descarga de Datos hacia el PC.

El programa realiza los siguientes pasos:

➤ **Mostrar en LCD:**

Presentar el mensaje:

“Descarga al PC”

“Conecte al PC”

“Número de datos=”

**Fin de tarea.**

➤ **Confirmar orden del PC:**

Esperar orden del PC

- Si no recibe orden:  
Continuar esperando
- Si recibe orden, continuar con la siguiente acción.

**Fin de tarea.**

➤ **Descargar datos:**

Leer datos de EEPROM.

Transmitir datos leídos al PC.

**Fin de tarea.**

➤ **Terminó de enviar datos:**

- Si no envió todos los datos  
Continuar con la descarga de datos
- Si terminó de enviar los datos  
Continuar con la siguiente acción.

**Fin de tarea.**

➤ **Mostrar en LCD:**

Presentar el mensaje “Listo”

**Fin de tarea.**

➤ **Volver al Menú Principal:**

Retornar al Menú Principal de opciones de programa.

**Fin de tarea.**

### **3.3 DESARROLLO DEL SOFTWARE PARA LA INTERFAZ GRÁFICA DEL COMPUTADOR.**

En esta sección se describe sobre la programación en el computador de la HMI (*Human Machine Interface*), de tal forma que exista la interacción con el módulo del entrenador para que descargue los datos almacenados hacia el computador, así como también permita realizar la gráfica de los datos obtenidos ya sea de una reciente descarga o de información guardada en el ordenador.

#### **3.3.1 DIAGRAMA DE FLUJO DE FUNCIONAMIENTO DE LA INTERFAZ EN EL PC.**

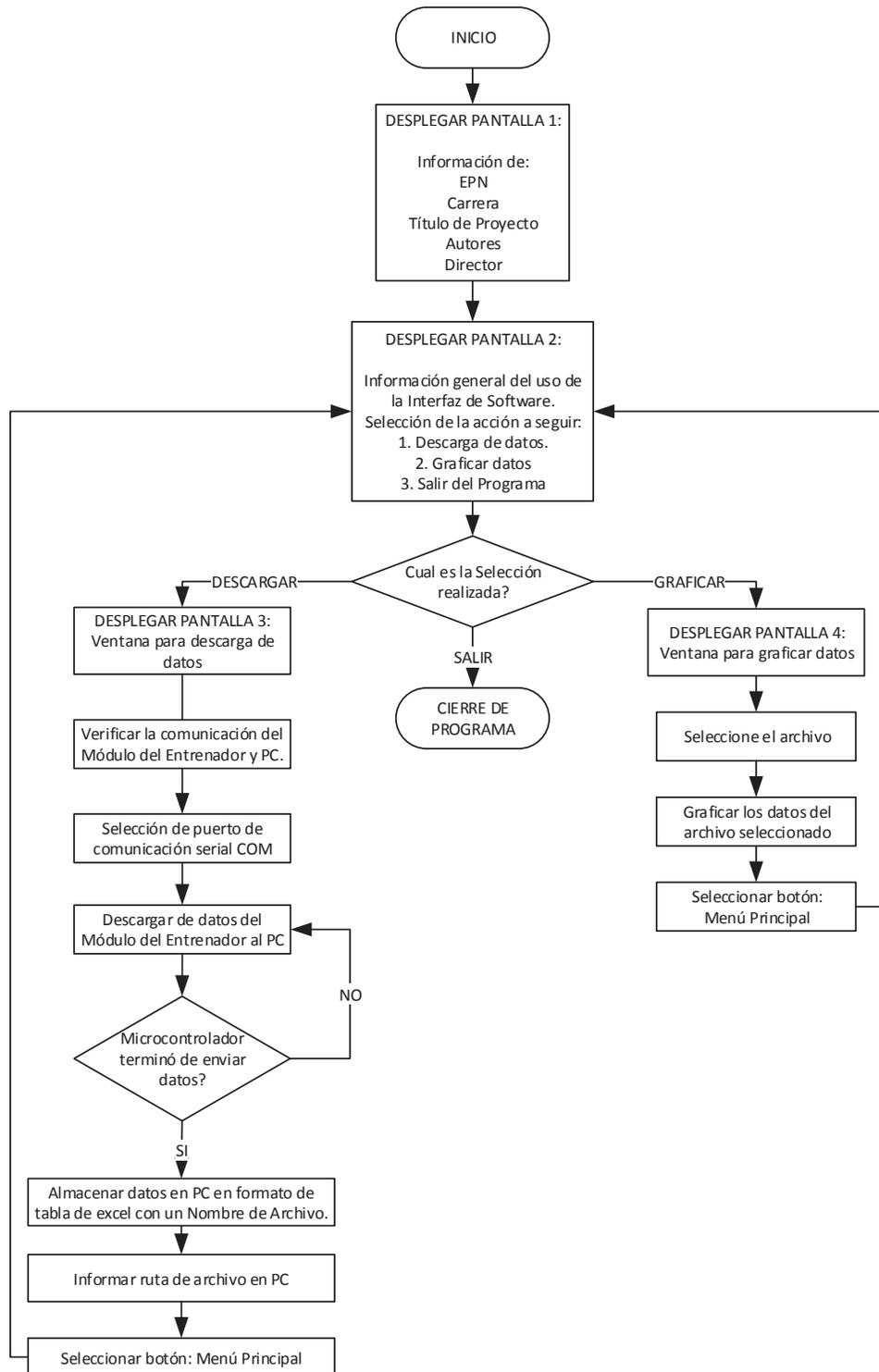
El software que se implementó está desarrollado en base al diagrama de funcionamiento de la interfaz que está indicado en la Figura 3.33, el cual tiene los siguientes pasos:

➤ **Desplegar Pantalla 1 (Ver Figura 3.34):**

Mostrar en la pantalla inicial durante 7 segundos:

- Universidad
- Carrera
- Nombre del Proyecto
- Autores del Proyecto
- Director del Proyecto
- Logotipo

**Fin de tarea**



**Figura 3. 33** Diagrama de Funcionamiento de la Interfaz Gráfica en Computador.



Figura 3. 34 Primera pantalla de Presentación del Proyecto.

➤ **Desplegar Pantalla 2 (Ver Figura 3.35):**

Presentar las opciones:

1. Descargar: Despliega la Pantalla 3.
2. Graficar: Despliega la Pantalla 4.
3. Salir: El programa cierra las ventanas desplegadas.

**Fin de tarea.**

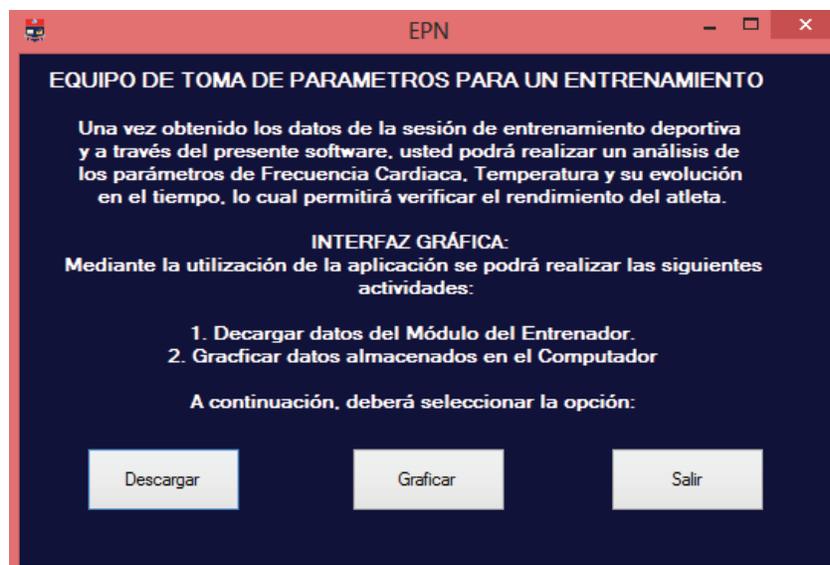
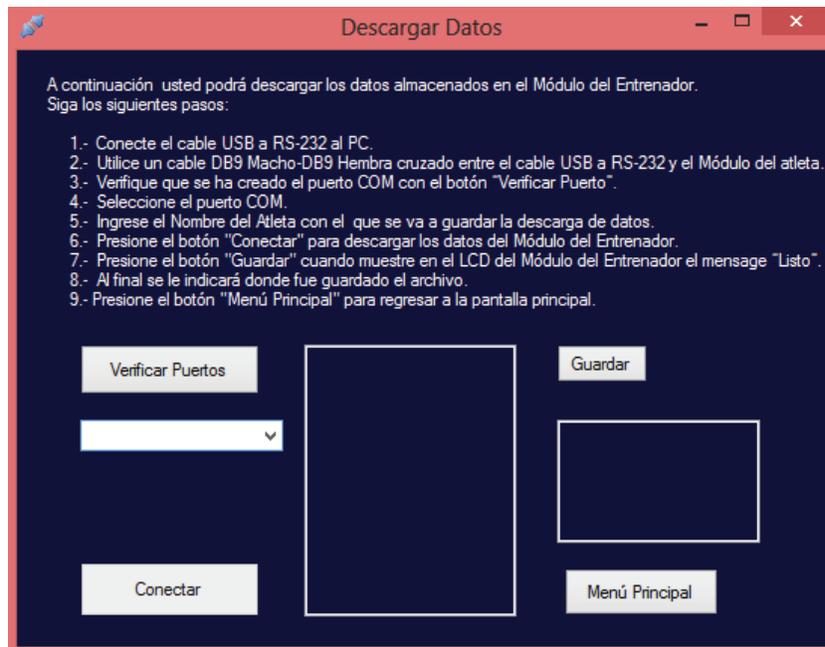


Figura 3. 35 Segunda Pantalla de Menú Principal.

➤ **Desplegar Pantalla 3 (Ver Figura 3.36):**

Seleccionar con un clic en botón “Descargar” en la ventana Menú Principal.

El programa realiza las siguientes tareas:



**Figura 3. 36** Tercera Pantalla de Descarga de Datos.

- **Verificar puertos:**

Realizar clic en botón verificar puertos.

Verificar puertos COM.

Seleccionar el puerto COM.

Habilitar campo para el ingreso del nombre del atleta.

**Fin de tarea.**

- **Descargar datos:**

Enviar orden al Módulo del Entrenador mediante un clic en botón “Conectar”.

Recibir datos del Módulo del Entrenador.

Mostrar la información recibida en un TextBox.

**Fin de tarea.**

- **Almacenar datos en PC:**  
Realizar clic en botón “Guardar”.  
Desplegar ventana para selección de ruta.  
Seleccionar ruta en PC para almacenar datos recibidos.  
Almacenar datos en hoja de Excel.  
Mostrar ruta de almacenamiento.  
**Fin de tarea.**
- **Seleccionar botón Menú Principal:**  
Dar clic en botón “Menú Principal”.  
Regresar a ventana Menú Principal.  
**Fin de tarea.**

➤ **Desplegar Pantalla 4 (Ver Figura 3.37):**

Seleccionar con un clic el botón “Graficar” en la Pantalla 2.

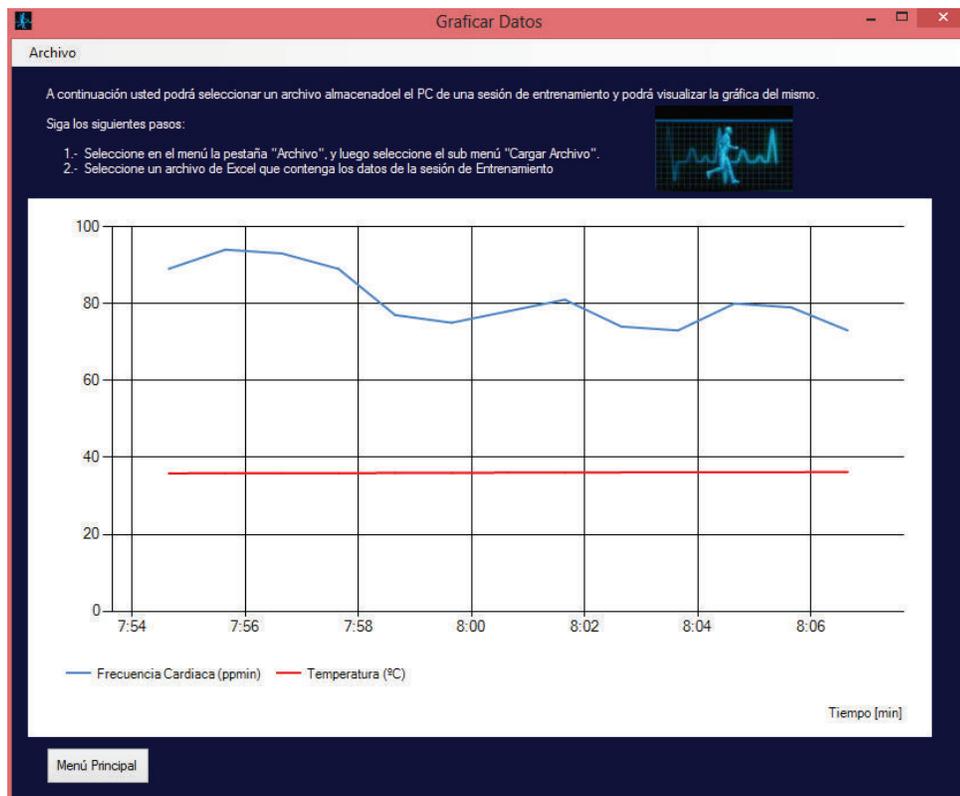


Figura 3. 37 Cuarta Pantalla de Graficar Datos.

El programa realiza las siguientes tareas:

- **Seleccionar Menú Archivo:**  
Acceder al menú de nombre “Archivo”  
    Ingresar al submenú “Cargar Archivo”.  
    Seleccionar documento Excel con datos descargados.  
**Fin de tarea.**
- **Graficar datos:**  
Leer datos almacenados de archivo de Excel.  
Graficar datos de Frecuencia Cardiaca versus Tiempo y Temperatura versus el Tiempo.  
**Fin de tarea.**
- **Seleccionar botón Menú Principal:**  
Dar clic en botón “Menú Principal”.  
Regresar a ventana Menú Principal.  
**Fin de tarea.**

### **3.4 FIJACIÓN DEL MÓDULO DEL ATLETA EN EL DEPORTISTA.**

Considerando que en la pista de entrenamiento un deportista requiere de la mayor movilidad posible, el Módulo del Atleta será ubicado en el punto de menor molestia y mayor estabilidad independiente de los movimientos propios de la actividad física a realizar. Se ha observado que la ubicación apropiada para el dispositivo es en la espalda alta sobre la columna vertebral.

El dispositivo estará sujeto a un soporte o corrector de postura de espalda de material de neopreno, el mismo que por sus condiciones no interfiere en los movimientos del atleta y es cómodo.

La Figura 3.38 muestra la ubicación del Módulo del Atleta de acuerdo a lo descrito.

El sensor de temperatura será ubicado y fijado en la axila del atleta mediante un adhesivo.

El *PulseSensor* se ubicará en el lóbulo de la oreja y sujeto con la pinza metálica.



**Figura 3. 38** Ubicación del Módulo del Atleta en el Deportista.

### **3.5 COSTO DEL PROYECTO.**

A continuación se detalla los costos por cada módulo del sistema de entrenamiento.

#### **3.5.1 COSTO DEL MÓDULO DEL ENTRENADOR.**

En la Tabla 3.1 se presentan los costos de cada uno de los elementos que lo componen.

Módulo del Entrenador				
Elemento	Capacidad	Cantidad	Costo unitario USD	Costo Total USD
R1	330 $\Omega$	1	0,02	0,02
R2	1 k $\Omega$	1	0,02	0,02
R3	4,7 k $\Omega$	1	0,02	0,02
R4	10 k $\Omega$	1	0,02	0,02
R5	10 k $\Omega$	1	0,02	0,02
R7	1 k $\Omega$	1	0,02	0,02
R8	1 k $\Omega$	1	0,02	0,02
R9	1 k $\Omega$	1	0,02	0,02
RV1 Potenciómetro	10 k $\Omega$	1	0,45	0,45
RV2 Potenciómetro	10 k $\Omega$	1	0,45	0,45
C1	1 Uf	1	0,06	0,06
C2 Cerámico	33 Pf	1	0,24	0,24
C3 Cerámico	33 Pf	1	0,24	0,24
C4	1 Uf	1	0,06	0,06
C5	1 Uf	1	0,06	0,06
C6	10 Uf	1	0,06	0,06
C7	10 Uf	1	0,06	0,06
C8	10 Uf	1	0,06	0,06
C9	10 Uf	1	0,06	0,06
C10	100 Uf	1	0,06	0,06
C11 Cerámico	100 Pf	1	0,06	0,06
U4	LM 7805	1	0,4	0,4
Disipador	Aluminio	1	0,45	0,45
D1	1n4148	1	0,06	0,06
D2 Led	Rojo	1	0,07	0,07
Microcontrolador	ATMEGA 8	1	4,11	4,11
Cristal	11059200 Hz	1	1,3	1,3
Cristal	32678000 Hz	1	1,3	1,3
<i>Transceiver</i>	YS-1020UB	1	30	30
U2	Max RS232	1	2,5	2,5
DB9	Hembra	1	0,35	0,35
U3	DS 1307	1	2,68	2,68
LCD	2x16	1	5,09	5,09
<i>switch</i>	4 pat 3 pos	1	0,14	0,14
Comutador SW1	2X2	1	1,5	1,5
Pulsador	4 pines	2	0,28	0,56
Pin Hembra	regleta	2	0,36	0,72
Pin Macho	regleta	2	0,45	0,9
Zócalo	2x14 pines	1	0,13	0,13
Zócalo	2x7 pines	1	0,13	0,13
Zócalo	2x4 pines	1	0,13	0,13

Batería 9V recargable	GP 299	1	12	12
Terminal batería 9 v		1	0,67	0,67
Cargador		1	6	6
Batería 3,2 v	CR2032	1	0,67	0,67
Porta Batería	GP2032	1	0,67	0,67
Baquelita 7x10 cm	Doble Lado	1	1	1
Cable Adaptador	USB- RS232	1	12	12
Cable Adaptador	DB9H – DB9M	1	2	2
Postes Metálicos		14	0,31	4,34
Barra Silicona		2	0,5	1
Pegamento UHU		1	1,5	1,5
			Subtotal	96,45
			IVA 12%	11,57
			<b>Total</b>	<b>108,02</b>

**Tabla 3. 1** Costo del Módulo del Entrenador.

### 3.5.2 COSTO DEL MÓDULO DEL ATLETA.

En la Tabla 3.2 se presentan los costos de cada uno de los elementos que lo componen.

Moduló del Atleta				
Elemento	Capacidad	Cantidad	Costo unitario USD	Costo Total USD
R1	4,7 k $\Omega$	1	0,02	0,02
R2	1 k $\Omega$	1	0,02	0,02
R3	47 k $\Omega$	1	0,02	0,02
R4	330 $\Omega$	1	0,02	0,02
R5	10 k $\Omega$	1	0,02	0,02
C1	1 Uf	1	0,06	0,06
C2 Cerámico	33 Pf	1	0,24	0,24
C3 Cerámico	33 Pf	1	0,24	0,24
D1	1n4148	1	0,06	0,06
D2 Led	Rojo	1	0,07	0,07
D3 Led	Verde	1	0,07	0,07
Q1	2N3906	1	0,07	0,07
Microcontrolador	ATMEGA 8	1	4,11	4,11
Cristal	11059200 Hz	1	1,3	1,3
Transceiver	YS-1020UB	1	30	30
Transductor	DS 18B20	1	3,13	3,13
Transductor	Pulsesensor	1	35	35

Switch	4 pines 3 posiciones	1	0,14	0,14
Pin Hembra	regleta	2	0,36	0,72
Pin Macho	regleta	2	0,45	0,9
Zócalo	2x14 pines	1	0,13	0,13
Batería 3,6V recargable	Sony N	1	20	20
Cargador	Sony N	1	20	20
Baquelita 5x5 cm	Doble lado	1	0,58	0,58
Arnés corrector espada		1	15,18	15,18
Plancha Acrílico	2mmx30cmx60cm	1	6,25	6,25
Corte Laser Acrílico		1	3	3
Cable UTP	1 m	1	0,76	0,76
Postes Metálicos		4	0,31	1,24
			Subtotal	143,35
			IVA 12%	17,20
			<b>Total</b>	<b>160,55</b>

**Tabla 3. 2** Costo del Módulo del Atleta.

### 3.5.3 COSTO TOTAL DEL PROYECTO.

En la Tabla 3.3 se presenta el costo total del sistema de entrenamientos.

Módulo	PRECIO USD
Módulo del Entrenador	108,02
Módulo del Atleta	160,55
<b>Total</b>	<b>268,57</b>

**Tabla 3. 3** Costo Total del Proyecto

Terminada la construcción de los Módulos del Atleta y del Entrenador, así como también obtenida la interfaz de descarga de datos para el PC, en el siguiente capítulo se procederá a probar y evaluar el funcionamiento del sistema en un entrenamiento deportivo.

## **CAPÍTULO 4. PRUEBAS DE OPERACIÓN Y ANÁLISIS DE RESULTADOS**

Las pruebas a realizarse se diseñaron para validar el equipo de monitoreo desarrollado y construido, acorde a las funcionalidades que tienen los dispositivos que lo conforman.

Se realizaron pruebas tanto para hardware y software para probar la integridad del sistema de monitoreo.

Las pruebas se dividieron en:

- Pruebas para Igualar Reloj.
- Pruebas de Frecuencia Cardíaca en Reposo.
- Pruebas en Sesión de Entrenamiento.
- Pruebas de Descarga de Datos.

### **4.1 PRUEBAS PARA IGUALAR RELOJ.**

Las pruebas para “Igualar reloj” se ejecutaron sobre el Módulo del Entrenador, para lo cual se realizó lo siguiente:

- a. Se seleccionó en el menú principal la opción 1Clock.
- b. Se procedió a ingresar los datos correspondientes a la fecha:
  - Día.
  - Mes.
  - Año.
- c. Se ingresó los datos de la Hora:
  - Hora
  - Minutos.

En el LCD se desplegaron los datos ingresados tal como se muestra en la Figura 4.1.

#### 4.1.1 ANÁLISIS DE RESULTADOS.

Se observó lo siguiente:

- Los datos ingresados con los pulsadores para la fecha y hora se desplegaron correctamente en el LCD.
- Se verificó que en comparación a un reloj de pulsera o el reloj interno del PC, el reloj del Módulo del Entrenador se mantiene estable sin atrasarse o adelantarse en el tiempo.
- Se verificó que en caso de apagarse el Módulo del Entrenador y volver a encenderlo, el DS1307 mantuvo la hora correcta gracias a la batería interna que alimenta a este circuito integrado.



**Figura 4. 1** LCD muestra Reloj Igualado.

Se concluye que esta parte del sistema tiene un correcto funcionamiento.

#### 4.2 PRUEBAS DE FRECUENCIA CARDIACA EN REPOSO.

Para probar la funcionalidad de la toma del dato de la frecuencia cardiaca en reposo del atleta, se hizo trabajar en conjunto el Módulo del Atleta y el Módulo del Entrenador. Se realizaron los siguientes pasos.

- Se solicitó a un voluntario que se ubique cómodamente, con el objetivo de que pueda relajarse como paso previo a la medición.
- Se ubicó el sensor de temperatura en la zona media de la axila, como si se tratara de un termómetro para aplicaciones médicas.
- Se ubicó el sensor de frecuencia cardiaca en el lóbulo de la oreja con la pinza.
- En el Módulo del Entrenador en el menú principal se seleccionó la opción 2FCR.

➤ El Módulo del Entrenador solicitó que se ingresen datos de:

- Sexo del individuo.
- Edad del individuo.

Una vez ingresado los datos en el LCD se mostró la información del sexo y la edad del individuo, así como también el resultado del cálculo de la frecuencia cardiaca máxima de la persona que, como se indico depende de los parámetros de sexo y edad.

- Se observa que el módulo del atleta empieza a tomar los datos al apagarse la luz de 1 primer LED que confirma la recepción de la orden del Módulo del Entrenador para empezar a medir datos. Adicionalmente, un segundo LED titila de acuerdo a cada pulso detectado de la frecuencia cardiaca de la persona.
- Se observa que en el LCD del Módulo del Entrenador se presenta en forma constante el correr del reloj.
- Una vez cumplido la medida de datos, el LCD mostró la información de Frecuencia Cardiaca en Reposo y Temperatura como se muestra en la Figura 4.2.



**Figura 4. 2** Datos obtenidos de la Frecuencia Cardiaca en Reposo.

#### 4.2.1 ANÁLISIS DE RESULTADOS.

De acuerdo a la información obtenida de la medición de los módulos, se comparó los resultados de la siguiente forma:

- Para la frecuencia cardiaca, se utilizó un estetoscopio que permitió contabilizar manualmente los latidos de la persona. El dato obtenido con el estetoscopio comparado al que se consiguió con el sistema de monitoreo de varias mediciones realizadas se observa que difiere aproximadamente en un pulso. Para la temperatura se comparó el dato del sistema de monitoreo con el obtenido de un termómetro bucal comercial utilizado para

finés médicos. En la Tabla 4.1 se muestra una serie de datos tomados con el sistema de monitoreo y comparados con los métodos tradicionales.

Nº de muestra	SISTEMA DE MONITOREO		METODOS TRADICIONALES	
	FC	Tº	Estetoscopio FC	Termómetro Comercial Tº
1	70	35,12	70	35,2
2	72	35,25	72	35,3
3	73	35,37	73	35,5
4	70	35,5	70	35,5
5	71	35,5	72	35,8
6	77	35,56	77	35,6
7	75	35,68	75	35,9
8	74	35,75	74	35,9
9	76	35,81	76	36,1
10	76	35,87	75	36,1
11	74	35,87	75	36,1
12	75	35,93	76	36,1
13	79	36	79	36,3
14	74	36	74	36,3
15	79	36,06	79	36,1

**Tabla 4. 1** Información comparativa de datos medidos utilizando el sistema de monitoreo y métodos tradicionales.

Nº de muestra	% Error de FC	% Error de Tº
1	0,00000	0,22727
2	0,00000	0,14164
3	0,00000	0,36620
4	0,00000	0,00000
5	1,38889	0,83799
6	0,00000	0,11236
7	0,00000	0,61281
8	0,00000	0,41783
9	0,00000	0,80332
10	1,33333	0,63712
11	1,33333	0,63712
12	1,31579	0,47091
13	0,00000	0,82645
14	0,00000	0,82645
15	0,00000	0,11080
<b>PROMEDIO</b>	<b>0,35809</b>	<b>0,46855</b>

**Tabla 4. 2: Cálculo de Errores.**

Para la Tabla 4.1, se realiza el cálculo de errores según la siguiente ecuación:

$$\%Error = \left| \frac{Valor\ met.\ tradicional - Valor\ Sist.\ de\ Monitoreo}{Valor\ met.\ tradicional} \right| * 100$$

El cálculo de errores se presenta en la Tabla 4.2, de lo cual se tiene un porcentaje de error de medición del 0.35% para la frecuencia cardiaca y un porcentaje de error del 0.46% para la temperatura

Los errores presentados por el sistema se deben al procesamiento interno de los datos propios de la cuantización del microcontrolador.

En base a los resultados obtenidos en mediciones en reposo se concluye que el sistema de monitoreo obtiene datos dentro de rangos aceptables.

### **4.3 PRUEBAS EN SESIÓN DE ENTRENAMIENTO.**

Para probar la toma de los datos en una sesión de entrenamiento del atleta, nuevamente se hace trabajar en conjunto el Módulo del Atleta y el Módulo del Entrenador. El funcionamiento durante esta fase, en cuanto al ingreso y toma de datos, es idéntico a lo que se hizo durante la frecuencia cardiaca en reposo, con la diferencia que son varias medidas continuas en cada minuto hasta el término de la sesión y también difiere el ambiente en que se toman las mediciones. Se realizaron los siguientes pasos.

- Se colocó en el deportista el soporte de neopreno con el Módulo del Atleta en su espalda.
- Se ubicó el sensor de temperatura en la zona media de la axila.
- Se ubicó el sensor de frecuencia cardiaca en el lóbulo de la oreja con la pinza.
- En el Módulo del Entrenador en el menú principal se seleccionó la opción 3Sesion.
- El Módulo del Entrenador solicita que se ingresen datos de.
  - Sexo del individuo.
  - Edad del individuo.

Una vez ingresado los datos en el LCD se mostró la información del sexo y la edad del individuo, así como también el resultado del cálculo de la frecuencia cardiaca máxima de la persona que, como se indicó depende de los parámetros de sexo y edad.

- Se observa que el módulo del atleta empieza a tomar los datos al apagarse la luz de un primer LED, que confirma la recepción de la orden del Módulo del Entrenador para empezar a medir datos.
- Durante la adquisición de datos de FC y T° se observó que mientras el deportista está en movimiento y expuesto a la luz del día, el sensor de pulsos cardiacos es muy susceptible a estos factores externos y entregaba datos erróneos en las mediciones. Para readecuar este resultado de la experiencia, fue necesario introducir en la programación del microcontrolador del Módulo del Atleta un tratamiento de análogo a digital a la señal del *PulseSensor* así como también un cambio a nivel de hardware. Esto se describe a continuación:

#### Cambio a nivel de Hardware:

A nivel de hardware en el Módulo del Atleta se cambió la entrada del *PulseSensor* en el microcontrolador del puerto PD2/INT0 al puerto PC0/ADC0, con el fin de utilizar el conversor análogo a digital (Ver Figura 4.3).

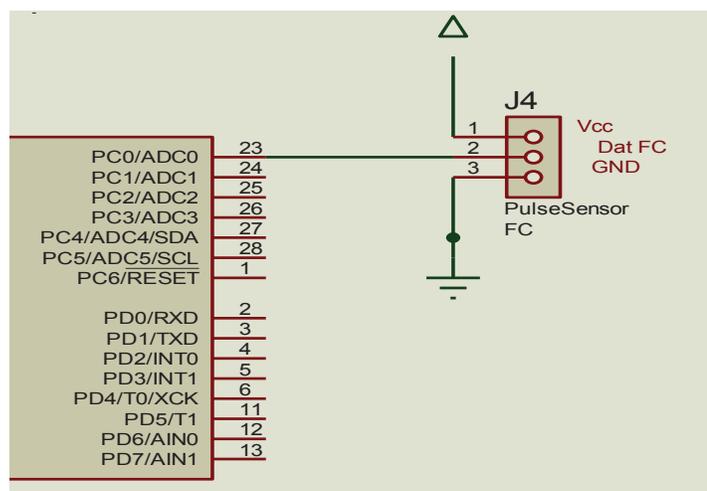
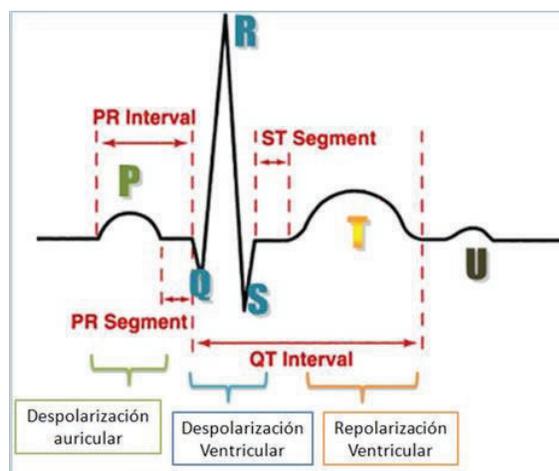


Figura 4. 3 *PulseSensor* conectado en PC0/ADC del Microcontrolador.

### Cambio a nivel de Software:

A nivel de software se utilizó un algoritmo de programación basado en el siguiente análisis:

Para la obtención del pulso a nivel digital, se toma como base la señal ECG como indica la Figura 4.4, la cual está constituida de una onda P, el complejo QRS, la onda T y la onda U.



**Figura 4. 4** Señal ECG.

El Intervalo QT comprende desde el inicio del complejo QRS hasta el final de la onda T, donde su duración estará entre 320 y 400 milisegundos. Estos valores de periodo de duración de la onda QT en relación al ritmo cardiaco varían de acuerdo a la Tabla 4.1

Ritmo cardiaco	Duración QT (s)
60	0.33 - 0.43
70	0.31 - 0.41
80	0.29 - 0.38
90	0.28 - 0.36
100	0.27 - 0.53
120	0.25 - 0.32

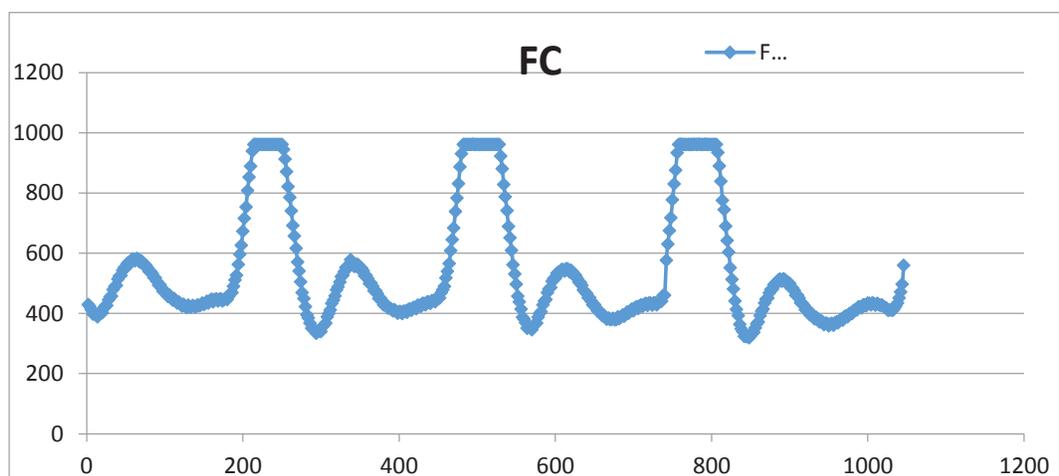
**Tabla 4. 3** Ritmo Cardiaco y su duración del intervalo QT.

El periodo de tiempo entre latidos es conocido como IBI (*Inter Beat Interval*). Por ejemplo para una frecuencia cardiaca de 60 pulsos por minuto se tiene

un IBI = 1 segundo y para una frecuencia cardiaca de 120 pulsos por minuto se tiene un IBI = 0,5 segundos.

Con estos datos teóricos de la señal ECG, se procede a digitalizar la señal análoga entregada por el *PulseSensor*. Para ello se utiliza un muestreo de conversión de señal analógica a digital cada 2 milisegundos, es decir, la frecuencia de muestreo es de 500 Hz.

Del muestreo realizado a la señal entregada por el *PulseSensor* se reconstruye la onda ECG, lo cual se muestra en la Figura 4.5.



**Figura 4. 5** Señal Digitalizada ECG del *PulseSensor*.

Para obtener el pulso a contabilizar, se analiza el segmento de onda desde Q hasta R de la señal digital, en base a las siguientes consideraciones:

- El valor IBI es de 0,6 segundos.
- Desde el primer intervalo QR para que se produzca el siguiente intervalo QR debe pasar el tiempo de por lo menos 3/5 de IBI.
- Se establece un umbral de media amplitud llamado *thresh* del intervalo QR.
- Un pulso a contar se estaría produciendo cuando este tenga un valor de amplitud superior al umbral *thresh*.
- El convertor AD del microcontrolador utiliza un registro de 10 bits que va desde 0 a 1023, por lo tanto, para obtener el valor del umbral *thresh* se considera el valor medio del registro que es 512.

- Se toma un periodo inicial de 250 milisegundos de no búsqueda del pulso para evitar falsos positivos entregados en la caída de tensión en el intervalo RS.
- Se restringe en amplitud cualquier valor superior a los 960, con el fin de evitar malas lecturas del sensor al estar expuesto totalmente a la luz del ambiente.
- En base a dos pulsos consecutivos encontrados se establece un nuevo valor de IBI y en base a los puntos QR se puede establecer un nuevo umbral *thresh*, lo cual ayudará a detectar los nuevos pulsos con el aumento de la frecuencia cardiaca del atleta y a encontrar pulsos que tiene una señal de baja amplitud, por lo tanto la digitalización de la señal se adapta a los cambios que la señal análoga pueda tener.

Así de esta manera, con el análisis descrito, el programa realiza la contabilización de pulsos en cada minuto. Cada pulso detectado se visualiza por el parpadeo de un LED.

Con los cambios a nivel de hardware y software descritos se procedió a continuar las pruebas de la sesión de entrenamiento, observando una adecuada obtención de datos a nivel de los pulsos cardiacos como se muestra en la Figura 4.6.



**Figura 4. 6** Datos en Sesión de Entrenamiento.

### 4.3.1 ANÁLISIS DE RESULTADOS.

De acuerdo a la información obtenida de la medición de los módulos, se comparó los resultados de la siguiente forma:

- Para la frecuencia cardiaca, se utilizó un estetoscopio que permitió contabilizar manualmente los latidos de la persona. El dato obtenido con el estetoscopio comparado al que se consiguió con el sistema de monitoreo de varias mediciones realizadas se observa que difiere aproximadamente en un pulso o dos. Para ello se pidió al atleta parar sus ejercicios para realizar la medición manual de los latidos del corazón.
- Para la temperatura se comparó el dato del sistema de monitoreo con el obtenido de un termómetro bucal. En la Tabla 4.4 se muestra una serie de datos tomados con el sistema de monitoreo y comparados con los métodos tradicionales.

Nº de muestra	SISTEMA DE MONITOREO		METODOS TRADICIONALES	
	FC	Tº	Estetoscopio FC	Termómetro Comercial Tº
1	106	34,93	107	35,3
2	99	34,5	99	35,4
3	97	34,68	99	35,7
4	103	34,87	104	35,3
5	106	34,87	106	35,6
6	99	34,68	99	35,8
7	102	35,81	102	35,1
8	111	35,81	111	35,5
9	104	35,68	104	35,8
10	107	35,81	107	35,8

**Tabla 4. 4** Información comparativa de datos medidos utilizando el sistema de monitoreo y métodos tradicionales en sesión de entrenamimientto.

Para la Tabla 4.4, se realiza el cálculo de errores según la siguiente ecuación:

$$\%Error = \left| \frac{Valor\ met.\ tradicional - Valor\ Sist.\ de\ Monitoreo}{Valor\ met.\ tradicional} \right| * 100$$

Nº de muestra	% Error de FC	% Error de Tº
1	0,93458	1,04816
2	0,00000	2,54237
3	1,02041	2,85714
4	0,96154	1,21813
5	0,00000	2,05056
6	0,00000	3,12849
7	0,99010	2,02279
8	0,00000	0,87324
9	0,00000	0,33520
10	0,00000	0,02793
<b>PROMEDIO</b>	<b>0,39066</b>	<b>1,61040</b>

**Tabla 4. 5** Cálculo de errores para la Sesión de Entrenamiento.

El cálculo de errores se presenta en la Tabla 4.5, de lo cual se tiene un porcentaje de error de medición del 0.39% para la frecuencia cardiaca y un porcentaje de error del 1.61% para la temperatura

Los errores presentados por el sistema se deben al procesamiento interno de los datos propios de la cuantización del microcontrolador.

En base a los resultados obtenidos en mediciones en sesión de entrenamiento se concluye que el sistema de monitoreo obtiene datos dentro de rangos aceptables.

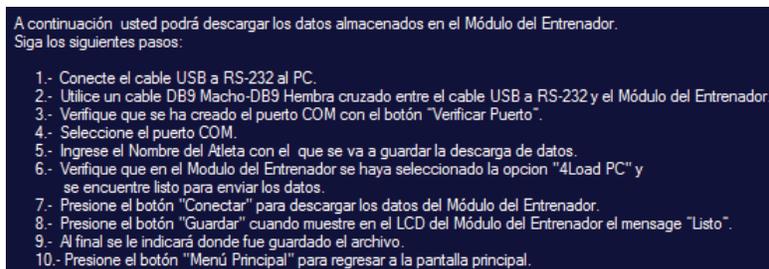
#### **4.4 PRUEBAS DE DESCARGA Y GRÁFICA DE DATOS.**

Para probar la funcionalidad de la descarga de datos desde el Módulo del Entrenador hacia el PC es necesario tomar en cuenta la interacción entre el módulo y el computador, de acuerdo a la descripción de los siguientes pasos.

##### **4.4.1 DESCARGA DE DATOS:**

En el computador se debe abrir el programa desarrollado en Visual Basic 2010, luego en la ventana del menú principal de opciones se selecciona el botón “Descargar”.

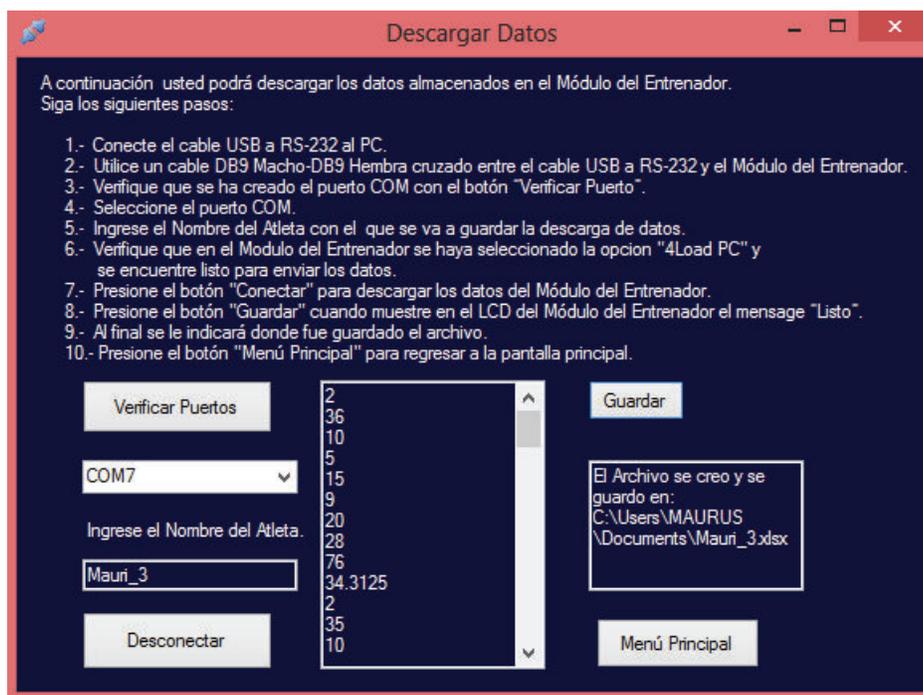
En la ventana desplegada para la función de Descarga, se presenta visualmente un instructivo referente a los pasos a seguir para la correcta descarga de datos del módulo. Estos pasos se muestran en la Figura 4.7.



**Figura 4. 7** Instructivo de pasos de la ventana de Descarga de Datos.

De acuerdo a lo descrito se realizó:

- Se verificó la existencia de un puerto COM para la conexión.
- Se ingresó el nombre del "Atleta".
- En el módulo del entrenador al seleccionar la opción "4LoadPC" en el LCD se muestra la cantidad de datos almacenados, lo cual será el indicativo de que el módulo está listo para el envío de datos.
- En la ventana de descarga se presiona el botón "Conectar", acción que realiza inmediatamente la descarga de datos desde el módulo.
- En la parte central de la ventana de trabajo se visualizan los datos descargados.



**Figura 4. 8** Ventana de Descarga de Datos e Información.

- Una vez descargados los datos, presionamos el botón guardar y al término del almacenamiento de los mismos en el PC la ventana muestra la ruta donde fue guardado la información (Ver Figura 4.8).
- Una muestra de los datos guardados en el PC del archivo Excel generado es el que se muestra en la Tabla 4.2.

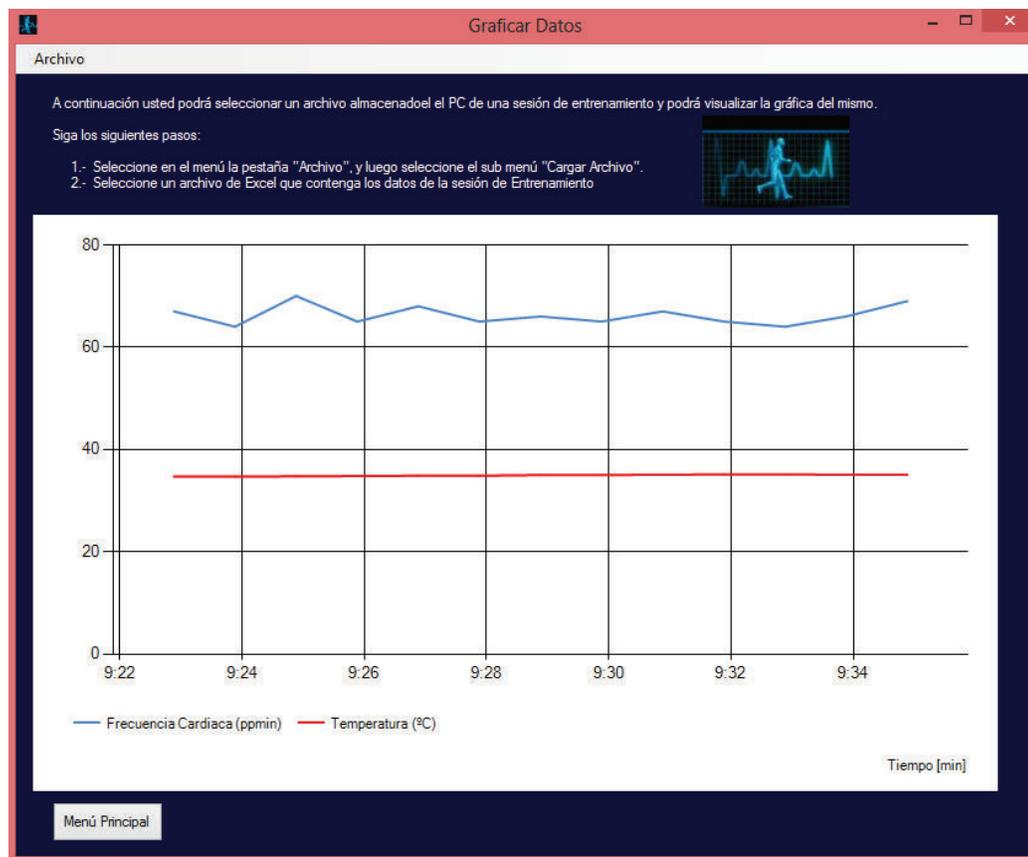
Nombre:	Vinicio3	Hora:	FC:	Temp °C:
Sexo:	Hombre	7:57:01	89	34,75
Edad:	35	7:58:01	99	34,9375
FCM:	191	7:59:01	111	34,875
Fecha:	21/3/15	8:00:01	114	34,6875
		8:01:01	117	34,6875
		8:02:01	121	34,625
		8:03:01	123	34,75
		8:04:01	125	34,8125
		8:05:01	121	34,8125
		8:06:01	115	34,8125
		8:07:01	110	34,9375
		8:08:01	107	34,9375

**Tabla 4. 6** Archivo Excel Almacenados de la Sesión de Entrenamiento.

#### 4.4.2 GRÁFICA DE DATOS:

Una vez realizada la descarga de datos en el PC y guardado en archivo Excel, en la ventana de menú principal se puede seleccionar el botón de graficar, el cual al ser presionado despliega una ventana que permite graficar los datos descargados. Para ello se realiza los siguientes pasos:

- En la ventana desplegada en la parte superior permite acceder a un menú con el nombre de “Archivo”.
- El menú Archivo contiene un submenú “Cargar Archivo”, el cual permitirá buscar y seleccionar el documento Excel con los datos que se descargaron del atleta.
- Una vez seleccionado el archivo, la ventana grafica inmediatamente los datos obtenidos de la sesión de entrenamiento, como se muestra en la Figura 4.9.



**Figura 4. 9** Gráfica de Datos obtenidos del Módulo.

#### 4.4.3 ANÁLISIS DE RESULTADOS.

De las pruebas realizadas y sus resultados se puede concluir que:

- La interfaz realizada está cumpliendo con los objetivos de descargar, almacenar y graficar los datos obtenidos del Módulo del Entrenador.
- El archivo Excel que se almacena en el PC es de fácil acceso y manejo.
- La gráfica permite evaluar el proceso de adaptación del deportista a la sesión de entrenamiento adoptada.

De acuerdo a las pruebas realizadas es posible concluir que el funcionamiento integral del sistema de monitoreo en un entrenamiento, cumple con los objetivos planteados de diseño y construcción. En el siguiente capítulo se detallan las conclusiones y recomendaciones del análisis y funcionamiento del sistema implementado.

## CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES

En base a las pruebas realizadas y los resultados obtenidos con el sistema de entrenamiento construido, se determinan las siguientes conclusiones.

### 5.1 CONCLUSIONES:

- a. En base a las pruebas realizadas en una pista atlética de medidas estándar de 400 metros, se comprueba que el sistema cumple con el objetivo de medir los datos de Frecuencia Cardíaca y Temperatura en forma inalámbrica.
- b. Dentro de las funcionalidades que tiene el Módulo del Entrenador está la interfaz visual de un LCD, que en las pruebas de los módulos permite al entrenador observar durante la sesión de entrenamiento los datos tomados del atleta en forma continua durante la sesión de entrenamiento, permitiendo tomar decisiones sobre el plan de ejercicios que está ejecutando durante la sesión deportiva. Adicionalmente, sin los dispositivos desarrollados el entrenador tendrá que obtener los parámetros manualmente o realizar constantes pausas en el entrenamiento para realizar la toma manual de datos del atleta.
- c. Mediante el uso de la interfaz HMI del PC es posible realizar la descarga de los datos medidos y también la gráfica de los mismos, lo cual permite comprobar que el microcontrolador del Módulo del Entrenador almacenó la información durante toda la sesión de entrenamiento, permitiendo al entrenador evaluar los parámetros obtenidos en función de su plan de ejercicios.
- d. En base al porcentaje de error de medición del 0.35% obtenido en las pruebas de reposo como en la sesión de entrenamiento del 0.39%, permite concluir que la digitalización de la señal análoga obtenida del *PulseSensor* mejoró los datos medidos de frecuencia cardíaca.
- e. En referencia al error de medición obtenido de los datos del sensor de temperatura tanto en pruebas de reposo del 0.46% como en sesión de entrenamiento de un 1.61%, permite concluir que en reposo el sensor tiene

condiciones óptimas de medición mientras que para la sesión de entrenamiento las condiciones difieren debido a la mayor transferencia de calor del ambiente al dispositivo a causa del movimiento del atleta.

## **5.2 RECOMENDACIONES:**

Gracias a las experiencias adquiridas en el desarrollo del proyecto se puede extraer las siguientes recomendaciones.

- a. En la medicina preventiva, en referencia a la frecuencia cardiaca máxima de una persona de acuerdo a su edad y sexo, se puede establecer límites de trabajo, teniendo en el dispositivo del entrenador alarmas visuales que indiquen si la frecuencia cardiaca se encuentra cumpliendo los límites en la zona de trabajo esperada o se encuentra por debajo de esta zona e incluso advertir que se está entrando en zonas de riesgo para la salud del deportista.
- b. En el ámbito deportivo, en el país se tiene los centros de alto rendimiento donde este proyecto podría tener una ampliación de funcionamiento, de tal forma que el módulo del entrenador pueda obtener datos de varios deportistas a la vez y al mismo tiempo, de acuerdo a esto el entrenador podría tener una comparación de rendimiento físico de sus atletas bajo la misma sesión de entrenamiento. Como resultado el entrenador podrá determinar que deportista se adapta mejor a su plan de ejercicios.
- c. Mediante la miniaturización de los circuitos impresos y sus componentes del módulo del atleta se podría obtener un dispositivo más cómodo para su transporte, pudiendo llegar incluso a la masificación de este producto.
- d. Adicionalmente a lo ya mencionado, se recuerda al usuario del módulo del entrenador y al deportista que al momento de realizar la medición de la frecuencia cardiaca en reposo, el atleta debe tener un proceso de calma de 5 a 10 minutos antes de poder tomar esta medición. Hay que tener en cuenta que este proceso de medición es una fase previa antes de que el atleta realice una fase de calentamiento o incluso de la misma fase de entrenamiento.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] J. Calabuig. Corazón de atleta. [En línea]. Available: [http://josegasparpf.blogspot.com/2012\\_09\\_01\\_archive.html](http://josegasparpf.blogspot.com/2012_09_01_archive.html). [Último acceso: 2012].
- [2] E. Bollana. Pautas y recomendaciones generales de entrenamiento. [En línea]. Available: <http://www.fcmax.com/nota.php?nota=778>. [Último acceso: 2010]
- [3] *Biolaster*. Control del Entrenamiento. [En línea]. Available: [http://www.biolaster.com/rendimiento\\_deportivo/control\\_entrenamiento](http://www.biolaster.com/rendimiento_deportivo/control_entrenamiento). [Último acceso: 2015].
- [4] IAAF *Athletics*. Reglas de Competición 2014-2015. Cap 5. Reglas técnicas. pp 129 – 280.
- [5] L. Jaime, Monitoreo Automático de Signos Vitales, Universidad Tecnológica de Querétaro, Querétaro, 2011.
- [6] M. Gallo, La preparación física básica, Universidad Tecnológica de Pereria Ciencias del Deporte y la Recreación Practica Básica II, 2009.
- [7] Fetecua, Juan Gabriel y Rodríguez, Andrea Liliana. Caracterización y clasificación de señales electrocardiográficas con métodos no lineales y wavelets adaptativas. Trabajo de grado, Ingeniería eléctrica. Pereira : s.n., 2006.
- [8] Ariza. A, Libro texto y material de laboratorio para la optativa introducción a la ingeniería biomédica, Universidad Pontificia Bolivariana Seccional Bucaramanga, 2009.
- [9] Chemes. C, La enfermería y la valoración de los signos vitales, Escuela de Enfermería UNT, Argentina, Tucumán, 2008.
- [10] Wunderlich CA. *The course of the temperature in diseases: a guide to clinical thermometry*. Am J Med Sci 1869; 57: 425-47.
- [11] *PULSESENSOR*, Blog de desarrolladores. [En línea] Available: <http://www.pulsesensor.com>

- [12] SPARKFUN ELECTRONICS, Distribuidor de hardware Electrónica. [En línea] Available: <https://www.sparkfun.com>
- [13] Atariguana. C, Mosquera. W, Diseño de un sistema de control automático para optimizar el rendimiento de un horno túnel, Universidad Politécnica Salesiana, 2002.
- [14] YS 1020ub, RF DATA TRANCEIVER Datasheet, ShenZhe Yishi Electronic.
- [15] ELECTRONICS TUTORIALS, Información y tutoriales destinado a estudiantes de electrónica. [En línea] Available: [http://www.electronics-tutorials.ws/transistor/tran\\_4.html](http://www.electronics-tutorials.ws/transistor/tran_4.html).
- [16] UNIVERSIDAD NACIONAL DE COLOMBIA, Dirección nacional de Innovación académica, electrónica Básica. [En línea] Available: [http://www.virtual.unal.edu.co/cursos/ingenieria/2001771/html/cap04/04\\_05\\_01.html](http://www.virtual.unal.edu.co/cursos/ingenieria/2001771/html/cap04/04_05_01.html)
- [17] BASCOM AVR HLP HELP, [En línea] Available: [http://mcselec.com/index.php?option=com\\_docman&task=doc\\_download&gid=188&Itemid=54](http://mcselec.com/index.php?option=com_docman&task=doc_download&gid=188&Itemid=54).
- [18] DS18S20 *High-Precision 1-Wire Digital Thermometer, Datasheet, Maxim Integrated.*
- [19] YS-1020UB *RF DATA TRANCEIVER, Datasheet,* [En línea] Available: <http://yinshi.net.cn>.
- [20] DS1307 *64 x 8 Serial Real-Time Clock, DALLAS SEMICONDUCTOR, Datasheet.*
- [21] CARLETTI, Eduardo J., ROBOTS, Comunicación - MAX232 - Conversor TTL-RS232, 2014, [En línea] Available: [http://robots-argentina.com.ar/Comunicacion\\_max232.htm](http://robots-argentina.com.ar/Comunicacion_max232.htm)
- [22] AGUAYS, Orlando. ITAZ Klever., Instrumentación Electrónica, “Sensor de Temperatura Digital DS18B20”, Escuela Politécnica Nacional. [En línea] Available: [http://www.academia.edu/4639177/Trabajo\\_exposicion\\_CI\\_DS18B20](http://www.academia.edu/4639177/Trabajo_exposicion_CI_DS18B20)

- [23] *BUDGE ELECTRONICS, Electronics for everyone, Wireless Transceiver 1020UB RS-232*, "Conexiones del Transceiver YS1020UB". [En línea] Available:  
<http://www.budgetronics.eu/index.php?action=article&aid=820&lang=EN>
- [24] *UM10204, I2C-bus specification and user manual, User Manual*, Abril 2014.  
[En línea] Available:  
[http://www.nxp.com/documents/user\\_manual/UM10204.pdf](http://www.nxp.com/documents/user_manual/UM10204.pdf)
- [25] Enrique Bollana. PAUTAS Y RECOMENDACIONES GENERALES DE ENTRENAMIENTO. [En línea]. Available:  
<http://www.fcmax.com/nota.php?nota=778>
- [26] Publicado por Mauricio Espinoza. PRUEBAS DE VELOCIDAD "ATLETISMO". [En línea]. Available: <http://pruebasdevelocidadcf.blogspot.com/>
- [27] OoCities.org. Frecuencia Cardiaca. [En línea]. Available:  
<http://www.oocities.org/ar/sistemacirculatorio00/frecuenciacardiaca.htm>
- [28] Ramon F Mateo G. Transductores de temperatura. [En línea]. Available:  
<http://www.monografias.com/trabajos3/transductores/transductores.shtml>

# ANEXOS

**ANEXO A. DATASHEET MICROCONTROLADOR ATMEGA 8.**

## Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 130 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16MIPS Throughput at 16MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 8Kbytes of In-System Self-programmable Flash program memory
  - 512Bytes EEPROM
  - 1Kbyte Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
  - In-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Three PWM Channels
  - 8-channel ADC in TQFP and QFN/MLF package
    - Eight Channels 10-bit Accuracy
  - 6-channel ADC in PDIP package
    - Six Channels 10-bit Accuracy
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF
- Operating Voltages
  - 2.7V - 5.5V (ATmega8L)
  - 4.5V - 5.5V (ATmega8)
- Speed Grades
  - 0 - 8MHz (ATmega8L)
  - 0 - 16MHz (ATmega8)
- Power Consumption at 4Mhz, 3V, 25°C
  - Active: 3.6mA
  - Idle Mode: 1.0mA
  - Power-down Mode: 0.5µA



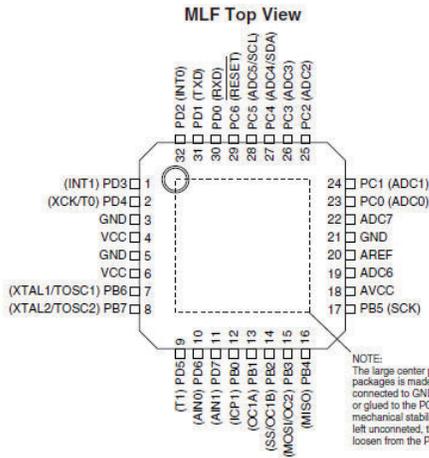
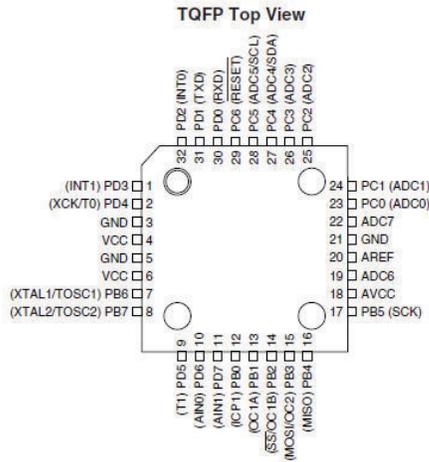
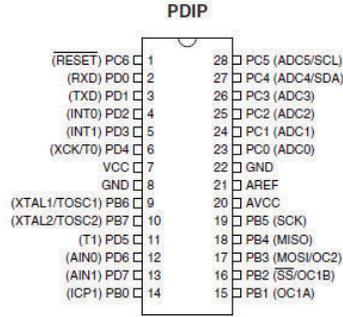
**8-bit Atmel with  
8KBytes In-  
System  
Programmable  
Flash**

**ATmega8  
ATmega8L**

Rev.2486AA-AVR-02/2013

# ATmega8(L)

## Pin Configurations



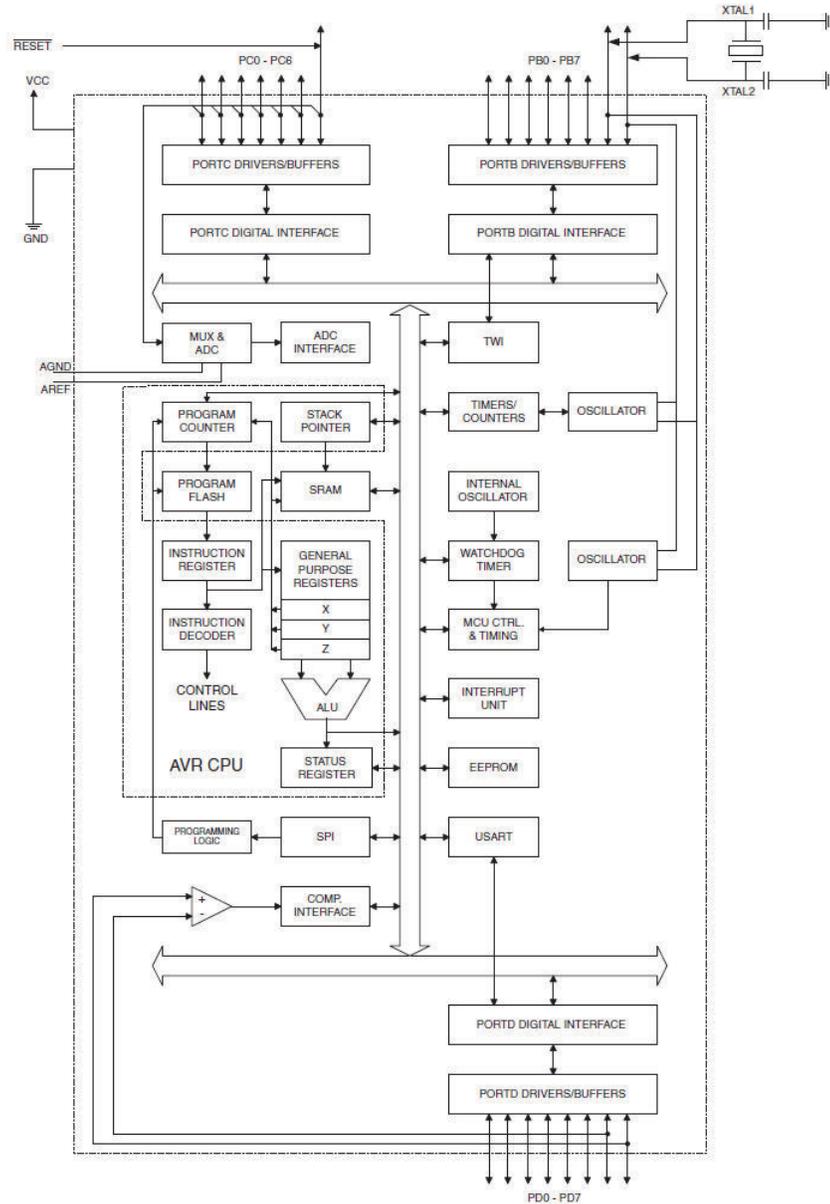
# ATmega8(L)

## Overview

The Atmel®AVR® ATmega8 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega8 achieves throughputs approaching 1MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed.

## Block Diagram

Figure 1. Block Diagram



---

## ATmega8(L)

The Atmel®AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega8 provides the following features: 8 Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes of EEPROM, 1 Kbyte of SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, a 6-channel ADC (eight channels in TQFP and QFN/MLF packages) with 10-bit accuracy, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next Interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The Flash Program memory can be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash Section will continue to run while the Application Flash Section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega8 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega8 is supported with a full suite of program and system development tools, including C compilers, macro assemblers, program simulators, and evaluation kits.

### Disclaimer

Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Minimum and Maximum values will be available after the device is characterized.

## ATmega8(L)

### Pin Descriptions

<b>VCC</b>	Digital supply voltage.
<b>GND</b>	Ground.
<b>Port B (PB7..PB0) XTAL1/XTAL2/TOSC1/ TOSC2</b>	<p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.</p> <p>Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.</p> <p>If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.</p> <p>The various special features of Port B are elaborated in “<a href="#">Alternate Functions of Port B</a>” on <a href="#">page 58</a> and “<a href="#">System Clock and Clock Options</a>” on <a href="#">page 25</a>.</p>
<b>Port C (PC5..PC0)</b>	<p>Port C is an 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>
<b>PC6/<math>\overline{\text{RESET}}</math></b>	<p>If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.</p> <p>If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in <a href="#">Table 15 on page 38</a>. Shorter pulses are not guaranteed to generate a Reset.</p> <p>The various special features of Port C are elaborated on <a href="#">page 61</a>.</p>
<b>Port D (PD7..PD0)</b>	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega8 as listed on <a href="#">page 63</a>.</p>
<b><math>\overline{\text{RESET}}</math></b>	<p>Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in <a href="#">Table 15 on page 38</a>. Shorter pulses are not guaranteed to generate a reset.</p>

---

**ATmega8(L)**

<b>AV<sub>CC</sub></b>	AV <sub>CC</sub> is the supply voltage pin for the A/D Converter, Port C (3..0), and ADC (7..6). It should be externally connected to V <sub>CC</sub> , even if the ADC is not used. If the ADC is used, it should be connected to V <sub>CC</sub> through a low-pass filter. Note that Port C (5..4) use digital supply voltage, V <sub>CC</sub> .
<b>AREF</b>	AREF is the analog reference pin for the A/D Converter.
<b>ADC7..6 (TQFP and QFN/MLF Package Only)</b>	In the TQFP and QFN/MLF package, ADC7..6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

**ANEXO B. DATASHEET DS18B20.**

## DS18S20

## High-Precision 1-Wire Digital Thermometer

### General Description

The DS18S20 digital thermometer provides 9-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18S20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of  $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  and is accurate to  $\pm 0.5^{\circ}\text{C}$  over the range of  $-10^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ . In addition, the DS18S20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18S20 has a unique 64-bit serial code, which allows multiple DS18S20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18S20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

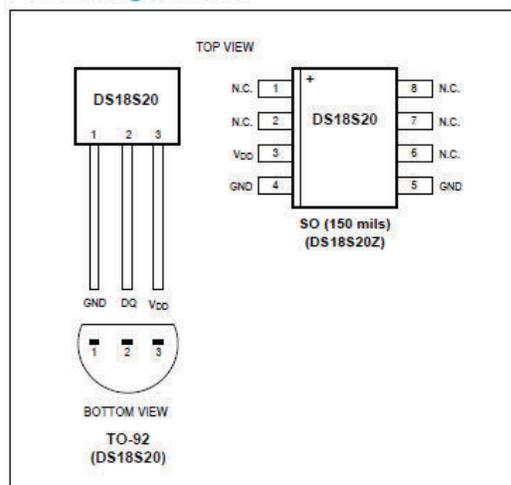
### Applications

- Thermostatic Controls
- Industrial Systems
- Consumer Products
- Thermometers
- Thermally Sensitive Systems

### Benefits and Features

- Unique 1-Wire<sup>®</sup> Interface Requires Only One Port Pin for Communication
- Maximize System Accuracy in Broad Range of Thermal Management Applications
  - Measures Temperatures from  $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  ( $-67^{\circ}\text{F}$  to  $+257^{\circ}\text{F}$ )
  - $\pm 0.5^{\circ}\text{C}$  Accuracy from  $-10^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$
  - 9-Bit Resolution
  - No External Components Required
- Parasite Power Mode Requires Only 2 Pins for Operation (DQ and GND)
- Simplifies Distributed Temperature-Sensing Applications with Multidrop Capability
  - Each Device Has a Unique 64-Bit Serial Code Stored in On-Board ROM
- Flexible User-Definable Nonvolatile (NV) Alarm Settings with Alarm Search Command Identifies Devices with Temperatures Outside Programmed Limits
- Available in 8-Pin SO (150 mils) and 3-Pin TO-92 Packages

### Pin Configurations



*Ordering Information appears at end of data sheet.*

*1-Wire is a registered trademark of Maxim Integrated Products, Inc.*

## DS18S20

## High-Precision 1-Wire Digital Thermometer

## Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground.....	-0.5V to +6.0V	Storage Temperature Range.....	-55°C to +125°C
Continuous Power Dissipation ( $T_A = +70^\circ\text{C}$ ).....		Lead Temperature (soldering, 10s).....	+260°C
8-Pin SO (derate 5.9mW/°C above +70°C).....	470.6mW	Soldering Temperature (reflow).....	
3-Pin TO-92 (derate 6.3mW/°C above +70°C).....	500mW	Lead(Pb)-free.....	+260°C
Operating Temperature Range.....	-55°C to +125°C	Containing lead(Pb).....	+240°C

These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

## DC Electrical Characteristics

( $V_{DD} = 3.0\text{V}$  to  $5.5\text{V}$ ,  $T_A = -55^\circ\text{C}$  to  $+125^\circ\text{C}$ , unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	$V_{DD}$	Local Power (Note 1)	+3.0		+5.5	V
Pullup Supply Voltage	$V_{PU}$	Parasite Power	+3.0		+5.5	V
		Local Power	+3.0		$V_{DD}$	
Thermometer Error	$t_{ERR}$	-10°C to +85°C			±0.5	°C
		-55°C to +125°C			±2	
Input Logic-Low	$V_{IL}$	(Note 1, 4, 5)	-0.3		+0.8	V
Input Logic-High	$V_{IH}$	Local Power	+2.2		The lower of 5.5 or $V_{DD}$ + 0.3	V
		Parasite Power	+3.0			
Sink Current	$I_L$	$V_{IO} = 0.4\text{V}$ (Note 1)	4.0			mA
Standby Current	$I_{DDS}$	(Note 7, 8)		750	1000	nA
Active Current	$I_{DD}$	$V_{DD} = 5\text{V}$ (Note 9)		1	1.5	mA
DQ Input Current	$I_{DQ}$	(Note 10)		5		μA
Drift		(Note 11)		±0.2		°C

Note 1: All voltages are referenced to ground.

Note 2: The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to  $V_{PU}$ . In order to meet the  $V_{IH}$  spec of the DS18S20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus:  $V_{PU\_ACTUAL} = V_{PU\_IDEAL} + V_{TRANSISTOR}$ .

Note 3: See typical performance curve in Figure 1.

Note 4: Logic-low voltages are specified at a sink current of 4mA.

Note 5: To guarantee a presence pulse under low voltage parasite power conditions,  $V_{ILMAX}$  may have to be reduced to as low as 0.5V.

Note 6: Logic-high voltages are specified at a source current of 1mA.

Note 7: Standby current specified up to +70°C. Standby current typically is 3μA at +125°C.

Note 8: To minimize  $I_{DDs}$ , DQ should be within the following ranges:  $GND \leq DQ \leq GND + 0.3\text{V}$  or  $V_{DD} - 0.3\text{V} \leq DQ \leq V_{DD}$ .

Note 9: Active current refers to supply current during active temperature conversions or EEPROM writes.

Note 10: DQ line is high ("high-Z" state).

Note 11: Drift data is based on a 1000-hour stress test at +125°C with  $V_{DD} = 5.5\text{V}$ .

DS18S20

High-Precision 1-Wire Digital Thermometer

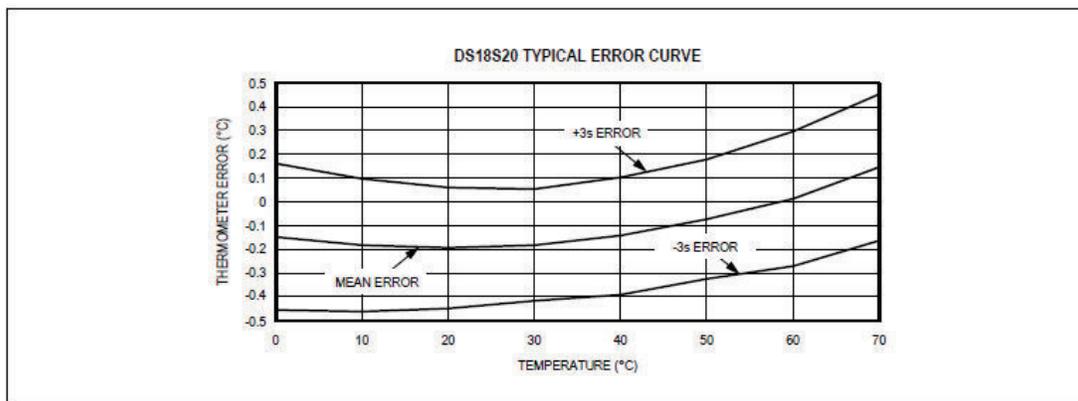


Figure 1. Typical Performance Curve

### AC Electrical Characteristics—NV Memory

( $V_{DD} = 3.0V$  to  $5.5V$ ;  $T_A = -55^{\circ}C$  to  $+100^{\circ}C$ , unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
NV Write Cycle Time	$t_{WR}$			2	10	ms
EEPROM Writes	$N_{EEWR}$	$-55^{\circ}C$ to $+55^{\circ}C$	50k			writes
EEPROM Data Retention	$t_{EEDR}$	$-55^{\circ}C$ to $+55^{\circ}C$	10			years

### AC Electrical Characteristics

( $V_{DD} = 3.0V$  to  $5.5V$ ;  $T_A = -55^{\circ}C$  to  $+125^{\circ}C$ , unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Temperature Conversion Time	$t_{CONV}$	(Note 12)			750	ms
Time to Strong Pullup On	$t_{SPON}$	Start Convert T Command Issued			10	$\mu s$
Time Slot	$t_{SLOT}$	(Note 12)	60		120	$\mu s$
Recovery Time	$t_{REC}$	(Note 12)	1			$\mu s$
Write 0 Low Time	$t_{LOW0}$	(Note 12)	60		120	$\mu s$
Write 1 Low Time	$t_{LOW1}$	(Note 12)	1		15	$\mu s$
Read Data Valid	$t_{RDV}$	(Note 12)			15	$\mu s$
Reset Time High	$t_{RSTH}$	(Note 12)	480			$\mu s$
Reset Time Low	$t_{RSTL}$	(Note 12, 13)	480			$\mu s$
Presence-Detect High	$t_{PDHIGH}$	(Note 12)	15		60	$\mu s$
Presence-Detect Low	$t_{PDLLOW}$	(Note 12)	60		240	$\mu s$
Capacitance	$C_{IN/OUT}$				25	pF

Note 12: See the timing diagrams in [Figure 2](#).

Note 13: Under parasite power, if  $t_{RSTL} > 960\mu s$ , a power-on reset may occur.

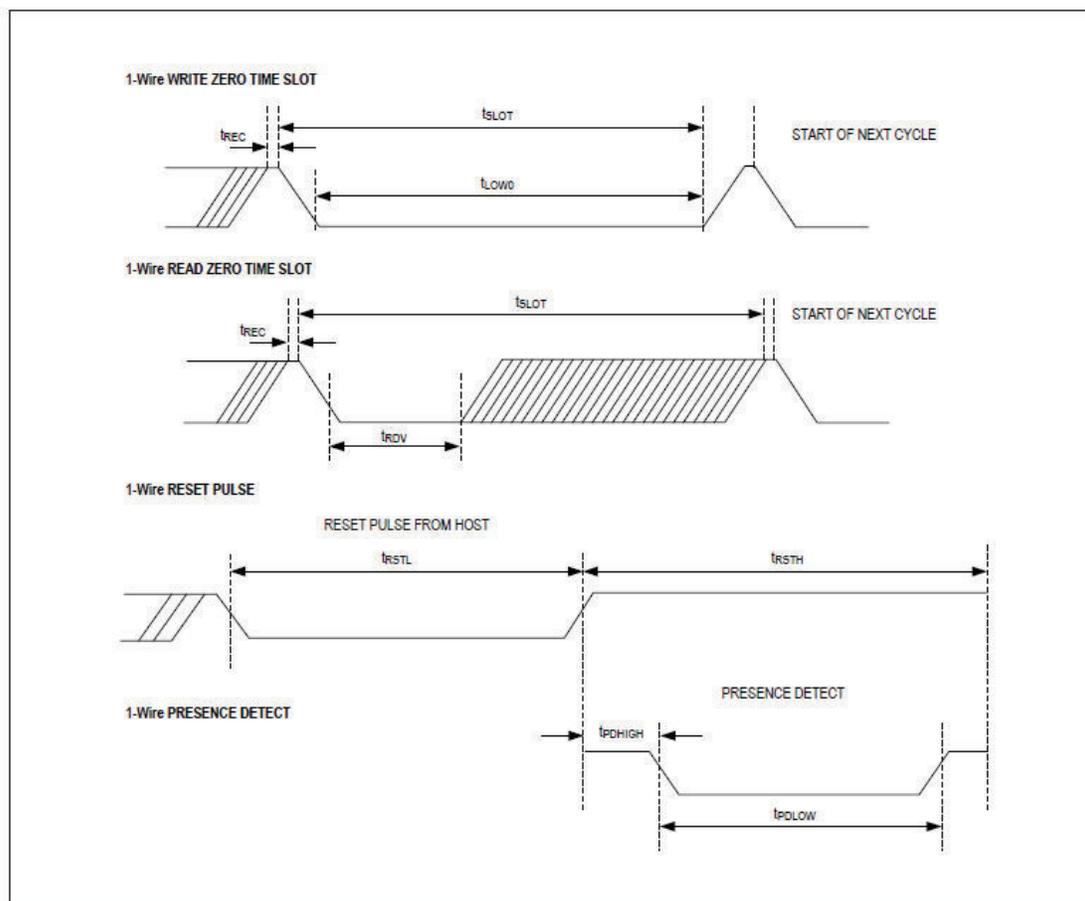


Figure 2. Timing Diagrams

## Pin Description

PIN		NAME	FUNCTION
TO-92	SO		
1	5	GND	Ground
2	4	DQ	Data Input/Output. Open-drain 1-Wire interface pin. Also provides power to the device when used in parasite power mode (see the <i>Powering the DS18S20</i> section.)
3	3	$V_{DD}$	Optional $V_{DD}$ . $V_{DD}$ must be grounded for operation in parasite power mode.
—	1, 2, 6, 7, 8	N.C.	No Connection

## DS18S20

## High-Precision 1-Wire Digital Thermometer

### Overview

Figure 3 shows a block diagram of the DS18S20, and pin descriptions are given in the *Pin Description* table. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers ( $T_H$  and  $T_L$ ). The  $T_H$  and  $T_L$  registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

The DS18S20 uses Maxim's exclusive 1-Wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18S20). In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited. The 1-Wire bus protocol, including detailed explanations of the commands and "time slots," is covered in the *1-Wire Bus System* section.

Another feature of the DS18S20 is the ability to operate without an external power supply. Power is instead supplied through the 1-Wire pullup resistor via the DQ pin when the bus is high. The high bus signal also charges an

internal capacitor ( $C_{PP}$ ), which then supplies power to the device when the bus is low. This method of deriving power from the 1-Wire bus is referred to as "parasite power." As an alternative, the DS18S20 may also be powered by an external supply on  $V_{DD}$ .

### Operation—Measuring Temperature

The core functionality of the DS18S20 is its direct-to-digital temperature sensor. The temperature sensor output has 9-bit resolution, which corresponds to  $0.5^\circ\text{C}$  steps. The DS18S20 powers-up in a low-power idle state; to initiate a temperature measurement and A-to-D conversion, the master must issue a Convert T [44h] command. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18S20 returns to its idle state. If the DS18S20 is powered by an external supply, the master can issue "read-time slots" (see the *1-Wire Bus System* section) after the Convert T command and the DS18S20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. If the DS18S20 is powered with parasite power, this notification technique cannot be used since the bus must be pulled high by a strong pullup during the entire temperature conversion. The bus requirements for parasite power are explained in detail in the *Powering The DS18S20* section.

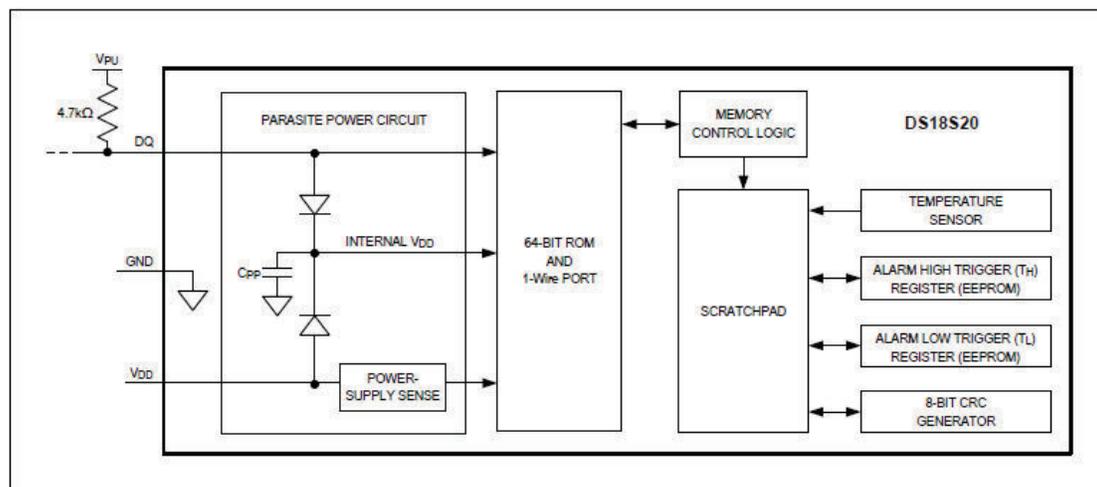


Figure 3. DS18S20 Block Diagram

## DS18S20

## High-Precision 1-Wire Digital Thermometer

The DS18S20 output data is calibrated in degrees centigrade; for Fahrenheit applications, a lookup table or conversion routine must be used. The temperature data is stored as a 16-bit sign-extended two's complement number in the temperature register (see Figure 4). The sign bits (S) indicate if the temperature is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. Table 1 gives examples of digital output data and the corresponding temperature reading.

Resolutions greater than 9 bits can be calculated using the data from the temperature, COUNT\_REMAIN and COUNT\_PER\_°C registers in the scratchpad. Note that the COUNT\_PER\_°C register is hard-wired to 16 (10h). After reading the scratchpad, the TEMP\_READ value is obtained by truncating the 0.5°C bit (bit 0) from the temperature data (see Figure 4). The extended resolution temperature can then be calculated using the following equation:

$$\text{TEMPERATURE} = \text{TEMP\_READ} - 0.25 \frac{\text{COUNT\_PER\_C} - \text{COUNT\_REMAIN}}{\text{COUNT\_PER\_C}}$$

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>-1</sup>
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
MS BYTE	S	S	S	S	S	S	S	S

S = SIGN

Figure 4. Temperature Register Format

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
S	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>5</sup>	2 <sup>5</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>

Figure 5. T<sub>H</sub> and T<sub>L</sub> Register Format

## Operation—Alarm Signaling

After the DS18S20 performs a temperature conversion, the temperature value is compared to the user-defined two's complement alarm trigger values stored in the 1-byte T<sub>H</sub> and T<sub>L</sub> registers (see Figure 5). The sign bit (S) indicates if the value is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. The T<sub>H</sub> and T<sub>L</sub> registers are nonvolatile (EEPROM) so they will retain data when the device is powered down. T<sub>H</sub> and T<sub>L</sub> can be accessed through bytes 2 and 3 of the scratchpad as explained in the Memory section.

Only bits 8 through 1 of the temperature register are used in the T<sub>H</sub> and T<sub>L</sub> comparison since T<sub>H</sub> and T<sub>L</sub> are 8-bit registers. If the measured temperature is lower than or equal to T<sub>L</sub> or higher than T<sub>H</sub>, an alarm condition exists and an alarm flag is set inside the DS18S20. This flag is updated after every temperature measurement; therefore, if the alarm condition goes away, the flag will be turned off after the next temperature conversion.

Table 1. Temperature/Data Relationship

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+85.0*	0000 0000 1010 1010	00AAh
+25.0	0000 0000 0011 0010	0032h
+0.5	0000 0000 0000 0001	0001h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1111	FFFFh
-25.0	1111 1111 1100 1110	FFCEh
-55.0	1111 1111 1001 0010	FF92h

\*The power-on reset value of the temperature register is +85°C.

The master device can check the alarm flag status of all DS18S20s on the bus by issuing an Alarm Search [ECh] command. Any DS18S20s with a set alarm flag will respond to the command, so the master can determine exactly which DS18S20s have experienced an alarm condition. If an alarm condition exists and the  $T_H$  or  $T_L$  settings have changed, another temperature conversion should be done to validate the alarm condition.

### Powering The DS18S20

The DS18S20 can be powered by an external supply on the  $V_{DD}$  pin, or it can operate in “parasite power” mode, which allows the DS18S20 to function without a local external supply. Parasite power is very useful for applications that require remote temperature sensing or those with space constraints. Figure 3 shows the DS18S20’s parasite-power control circuitry, which “steals” power from the 1-Wire bus via the DQ pin when the bus is high. The stolen charge powers the DS18S20 while the bus is high, and some of the charge is stored on the parasite power capacitor ( $C_{PP}$ ) to provide power when the bus is low. When the DS18S20 is used in parasite power mode, the  $V_{DD}$  pin must be connected to ground.

In parasite power mode, the 1-Wire bus and  $C_{PP}$  can provide sufficient current to the DS18S20 for most operations as long as the specified timing and voltage requirements are met (see the *DC Electrical Characteristics* and the *AC Electrical Characteristics*). However, when the DS18S20 is performing temperature conversions or copying data from the scratchpad memory to EEPROM, the operating current can be as high as 1.5mA. This current can cause an unacceptable voltage drop across the weak 1-Wire pullup resistor and is more current than can be supplied by  $C_{PP}$ . To assure that the DS18S20 has sufficient supply current, it is necessary to provide a strong pullup on the 1-Wire bus whenever temperature

conversions are taking place or data is being copied from the scratchpad to EEPROM. This can be accomplished by using a MOSFET to pull the bus directly to the rail as shown in Figure 6. The 1-Wire bus must be switched to the strong pullup within  $10\mu\text{s}$  (max) after a Convert T [44h] or Copy Scratchpad [48h] command is issued, and the bus must be held high by the pullup for the duration of the conversion ( $t_{CONV}$ ) or data transfer ( $t_{WR} = 10\text{ms}$ ). No other activity can take place on the 1-Wire bus while the pullup is enabled.

The DS18S20 can also be powered by the conventional method of connecting an external power supply to the  $V_{DD}$  pin, as shown in Figure 7. The advantage of this method is that the MOSFET pullup is not required, and the 1-Wire bus is free to carry other traffic during the temperature conversion time.

The use of parasite power is not recommended for temperatures above  $100^\circ\text{C}$  since the DS18S20 may not be able to sustain communications due to the higher leakage currents that can exist at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that the DS18S20 be powered by an external power supply.

In some situations the bus master may not know whether the DS18S20s on the bus are parasite powered or powered by external supplies. The master needs this information to determine if the strong bus pullup should be used during temperature conversions. To get this information, the master can issue a Skip ROM [CCh] command followed by a Read Power Supply [B4h] command followed by a “read-time slot”. During the read-time slot, parasite powered DS18S20s will pull the bus low, and externally powered DS18S20s will let the bus remain high. If the bus is pulled low, the master knows that it must supply the strong pullup on the 1-Wire bus during temperature conversions.

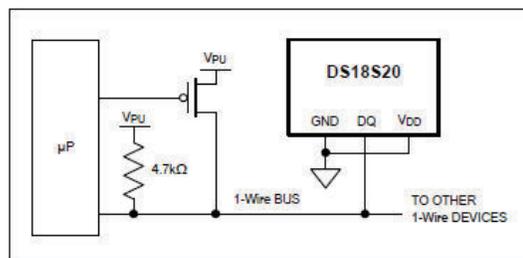


Figure 6. Supplying the Parasite-Powered DS18S20 During Temperature Conversions

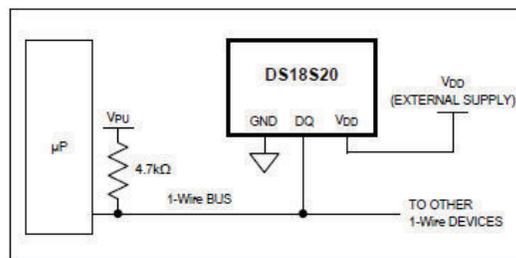


Figure 7. Powering the DS18S20 with an External Supply

### 64-Bit Lasered ROM Code

Each DS18S20 contains a unique 64-bit code (see [Figure 8](#)) stored in ROM. The least significant 8 bits of the ROM code contain the DS18S20's 1-Wire family code: 10h. The next 48 bits contain a unique serial number. The most significant 8 bits contain a cyclic redundancy check (CRC) byte that is calculated from the first 56 bits of the ROM code. A detailed explanation of the CRC bits is provided in the [CRC Generation](#) section. The 64-bit ROM code and associated ROM function control logic allow the DS18S20 to operate as a 1-Wire device using the protocol detailed in the [1-Wire Bus System](#) section.

### Memory

The DS18S20's memory is organized as shown in [Figure 9](#). The memory consists of an SRAM scratchpad with nonvolatile EEPROM storage for the high and low alarm trigger registers (T<sub>H</sub> and T<sub>L</sub>). Note that if the DS18S20 alarm function is not used, the T<sub>H</sub> and T<sub>L</sub> registers can serve as general-purpose memory. All memory commands are described in detail in the [DS18S20 Function Commands](#) section.

Byte 0 and byte 1 of the scratchpad contain the LSB and the MSB of the temperature register, respectively. These bytes are read-only. Bytes 2 and 3 provide access to T<sub>H</sub> and T<sub>L</sub> registers. Bytes 4 and 5 are reserved for internal use by the device and cannot be overwritten; these bytes will return all 1s when read. Bytes 6 and 7 contain the COUNT REMAIN and COUNT PER °C registers, which can be used to calculate extended resolution results as explained in the [Operation—Measuring Temperature](#) section.

Byte 8 of the scratchpad is read-only and contains the CRC code for bytes 0 through 7 of the scratchpad. The DS18S20 generates this CRC using the method described in the [CRC Generation](#) section.

Data is written to bytes 2 and 3 of the scratchpad using the Write Scratchpad [4Eh] command; the data must be transmitted to the DS18S20 starting with the least significant bit of byte 2. To verify data integrity, the scratchpad can be read (using the Read Scratchpad [BEh] command) after the data is written. When reading the scratchpad, data is transferred over the 1-Wire bus starting with the least significant bit of byte 0. To transfer the T<sub>H</sub> and T<sub>L</sub> data from the scratchpad to EEPROM, the master must issue the Copy Scratchpad [48h] command.

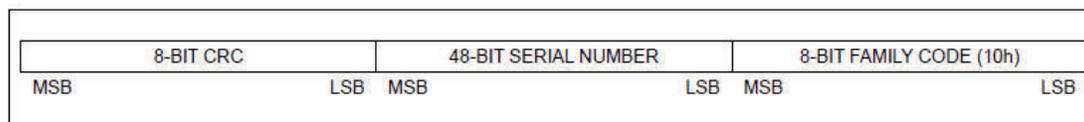


Figure 8. 64-Bit Lasered ROM Code

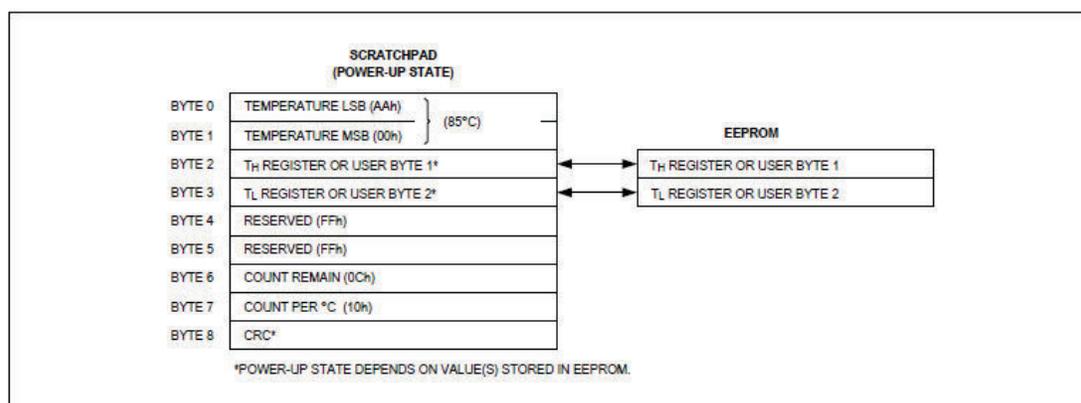


Figure 9. DS18S20 Memory Map

Data in the EEPROM registers is retained when the device is powered down; at power-up the EEPROM data is reloaded into the corresponding scratchpad locations. Data can also be reloaded from EEPROM to the scratchpad at any time using the Recall E<sup>2</sup> [B8h] command. The master can issue "read-time slots" (see the [1-Wire Bus System](#) section) following the Recall E<sup>2</sup> command and the DS18S20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done.

### CRC Generation

CRC bytes are provided as part of the DS18S20's 64-bit ROM code and in the 9th byte of the scratchpad memory. The ROM code CRC is calculated from the first 56 bits of the ROM code and is contained in the most significant byte of the ROM. The scratchpad CRC is calculated from the data stored in the scratchpad, and therefore it changes when the data in the scratchpad changes. The CRCs provide the bus master with a method of data validation when data is read from the DS18S20. To verify that data has been read correctly, the bus master must re-calculate the CRC from the received data and then compare this value to either the ROM code CRC (for ROM reads) or to the scratchpad CRC (for scratchpad reads). If the calculated CRC matches the read CRC, the data has been received error free. The comparison of CRC values and the decision to continue with an operation are determined entirely by the bus master. There is no circuitry inside the DS18S20 that prevents a command sequence from proceeding if the DS18S20 CRC (ROM or scratchpad) does not match the value generated by the bus master.

The equivalent polynomial function of the CRC (ROM or scratchpad) is:

$$\text{CRC} = X^8 + X^5 + X^4 + 1$$

The bus master can re-calculate the CRC and compare it to the CRC values from the DS18S20 using the polynomial generator shown in [Figure 10](#). This circuit consists of a shift register and XOR gates, and the shift register bits are initialized to 0. Starting with the least significant bit of the ROM code or the least significant bit of byte 0 in the scratchpad, one bit at a time should be shifted into the shift register. After shifting in the 56th bit from the ROM or the most significant bit of byte 7 from the scratchpad, the polynomial generator will contain the re-calculated CRC. Next, the 8-bit ROM code or scratchpad CRC from the DS18S20 must be shifted into the circuit. At this point, if the re-calculated CRC was correct, the shift register will contain all 0s. Additional information about the Maxim 1-Wire cyclic redundancy check is available in *Application Note 27: Understanding and Using Cyclic Redundancy Checks with Maxim iButton Products*.

### 1-Wire Bus System

The 1-Wire bus system uses a single bus master to control one or more slave devices. The DS18S20 is always a slave. When there is only one slave on the bus, the system is referred to as a "single-drop" system; the system is "multidrop" if there are multiple slaves on the bus.

All data and commands are transmitted least significant bit first over the 1-Wire bus.

The following discussion of the 1-Wire bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing).

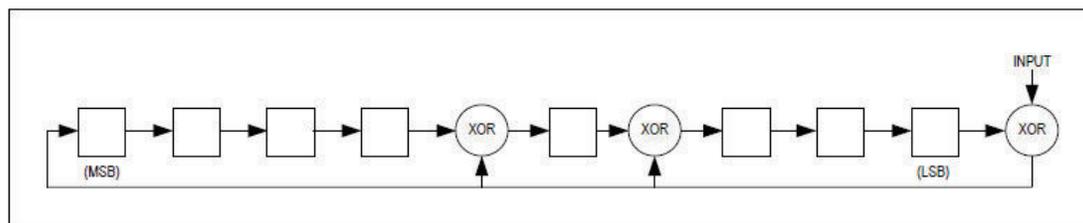


Figure 10. CRC Generator

### Hardware Configuration

The 1-Wire bus has by definition only a single data line. Each device (master or slave) interfaces to the data line via an open drain or 3-state port. This allows each device to “release” the data line when the device is not transmitting data so the bus is available for use by another device. The 1-Wire port of the DS18S20 (the DQ pin) is open drain with an internal circuit equivalent to that shown in [Figure 11](#).

The 1-Wire bus requires an external pullup resistor of approximately 5k $\Omega$ ; thus, the idle state for the 1-Wire bus is high. If for any reason a transaction needs to be suspended, the bus MUST be left in the idle state if the transaction is to resume. Infinite recovery time can occur between bits so long as the 1-Wire bus is in the inactive (high) state during the recovery period. If the bus is held low for more than 480 $\mu$ s, all components on the bus will be reset.

### Transaction Sequence

The transaction sequence for accessing the DS18S20 is as follows:

- Step 1. Initialization
- Step 2. ROM Command (followed by any required data exchange)
- Step 3. DS18S20 Function Command (followed by any required data exchange)

It is very important to follow this sequence every time the DS18S20 is accessed, as the DS18S20 will not respond if any steps in the sequence are missing or out of order. Exceptions to this rule are the Search ROM [F0h] and Alarm Search [ECh] commands. After issuing either of these ROM commands, the master must return to Step 1 in the sequence.

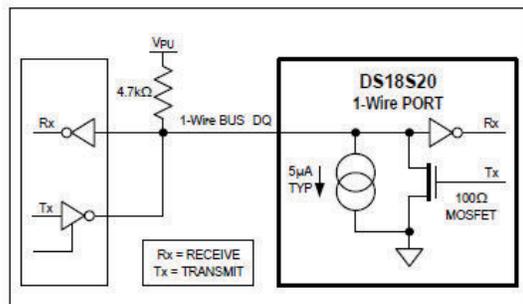


Figure 11. Hardware Configuration

*iButton* is a registered trademark of Maxim Integrated Products, Inc.

### Initialization

All transactions on the 1-Wire bus begin with an initialization sequence. The initialization sequence consists of a reset pulse transmitted by the bus master followed by presence pulse(s) transmitted by the slave(s). The presence pulse lets the bus master know that slave devices (such as the DS18S20) are on the bus and are ready to operate. Timing for the reset and presence pulses is detailed in the [1-Wire Signaling](#) section.

### ROM Commands

After the bus master has detected a presence pulse, it can issue a ROM command. These commands operate on the unique 64-bit ROM codes of each slave device and allow the master to single out a specific device if many are present on the 1-Wire bus. These commands also allow the master to determine how many and what types of devices are present on the bus or if any device has experienced an alarm condition. There are five ROM commands, and each command is 8 bits long. The master device must issue an appropriate ROM command before issuing a DS18S20 function command. A flowchart for operation of the ROM commands is shown in [Figure 16](#).

#### Search Rom [F0h]

When a system is initially powered up, the master must identify the ROM codes of all slave devices on the bus, which allows the master to determine the number of slaves and their device types. The master learns the ROM codes through a process of elimination that requires the master to perform a Search ROM cycle (i.e., Search ROM command followed by data exchange) as many times as necessary to identify all of the slave devices. If there is only one slave on the bus, the simpler Read ROM command (see below) can be used in place of the Search ROM process. For a detailed explanation of the Search ROM procedure, refer to the *iButton*<sup>®</sup> *Book of Standards* at [www.maximintegrated.com/ibuttonbook](http://www.maximintegrated.com/ibuttonbook). After every Search ROM cycle, the bus master must return to Step 1 (Initialization) in the transaction sequence.

#### Read ROM [33h]

This command can only be used when there is one slave on the bus. It allows the bus master to read the slave's 64-bit ROM code without using the Search ROM procedure. If this command is used when there is more than one slave present on the bus, a data collision will occur when all the slaves attempt to respond at the same time.

**Match ROM [55h]**

The match ROM command followed by a 64-bit ROM code sequence allows the bus master to address a specific slave device on a multidrop or single-drop bus. Only the slave that exactly matches the 64-bit ROM code sequence will respond to the function command issued by the master; all other slaves on the bus will wait for a reset pulse.

**Skip ROM [CCh]**

The master can use this command to address all devices on the bus simultaneously without sending out any ROM code information. For example, the master can make all DS18S20s on the bus perform simultaneous temperature conversions by issuing a Skip ROM command followed by a Convert T [44h] command.

Note that the Read Scratchpad [BEh] command can follow the Skip ROM command only if there is a single slave device on the bus. In this case, time is saved by allowing the master to read from the slave without sending the device's 64-bit ROM code. A Skip ROM command followed by a Read Scratchpad command will cause a data collision on the bus if there is more than one slave since multiple devices will attempt to transmit data simultaneously.

**Alarm Search [ECh]**

The operation of this command is identical to the operation of the Search ROM command except that only slaves with a set alarm flag will respond. This command allows the master device to determine if any DS18S20s experienced an alarm condition during the most recent temperature conversion. After every Alarm Search cycle (i.e., Alarm Search command followed by data exchange), the bus master must return to Step 1 (Initialization) in the transaction sequence. See the [Operation—Alarm Signaling](#) section for an explanation of alarm flag operation.

**DS18S20 Function Commands**

After the bus master has used a ROM command to address the DS18S20 with which it wishes to communicate, the master can issue one of the DS18S20 function commands. These commands allow the master to write to and read from the DS18S20's scratchpad memory, initiate temperature conversions and determine the power supply mode. The DS18S20 function commands, which are described below, are summarized in [Table 2](#) and illustrated by the flowchart in [Figure 17](#).

**Table 2. DS18S20 Function Command Set**

COMMAND	DESCRIPTION	PROTOCOL	1-Wire BUS ACTIVITY AFTER COMMAND IS ISSUED	NOTES
<b>TEMPERATURE CONVERSION COMMANDS</b>				
Convert T	Initiates temperature conversion.	44h	DS18S20 transmits conversion status to master (not applicable for parasite-powered DS18S20s).	1
<b>MEMORY COMMANDS</b>				
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	BEh	DS18S20 transmits up to 9 data bytes to master.	2
Write Scratchpad	Writes data into scratchpad bytes 2 and 3 ( $T_H$ and $T_L$ ).	4Eh	Master transmits 2 data bytes to DS18S20.	3
Copy Scratchpad	Copies $T_H$ and $T_L$ data from the scratchpad to EEPROM.	48h	None	1
Recall E <sup>2</sup>	Recalls $T_H$ and $T_L$ data from EEPROM to the scratchpad.	B8h	DS18S20 transmits recall status to master.	
Read Power Supply	Signals DS18S20 power supply mode to the master.	B4h	DS18S20 transmits supply status to master.	

**Note 1:** For parasite-powered DS18S20s, the master must enable a strong pullup on the 1-Wire bus during temperature conversions and copies from the scratchpad to EEPROM. No other bus activity may take place during this time.

**Note 2:** The master can interrupt the transmission of data at any time by issuing a reset.

**Note 3:** Both bytes must be written before a reset is issued.

**Convert T [44h]**

This command initiates a single temperature conversion. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18S20 returns to its low-power idle state. If the device is being used in parasite power mode, within 10 $\mu$ s (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for the duration of the conversion ( $t_{CONV}$ ) as described in the [Powering The DS18S20](#) section. If the DS18S20 is powered by an external supply, the master can issue read-time slots after the Convert T command and the DS18S20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. In parasite power mode this notification technique cannot be used since the bus is pulled high by the strong pullup during the conversion.

**Write Scratchpad [4Eh]**

This command allows the master to write 2 bytes of data to the DS18S20's scratchpad. The first byte is written into the  $T_H$  register (byte 2 of the scratchpad), and the second byte is written into the  $T_L$  register (byte 3 of the scratchpad). Data must be transmitted least significant bit first. Both bytes MUST be written before the master issues a reset, or the data may be corrupted.

**Read Scratchpad [BEh]**

This command allows the master to read the contents of the scratchpad. The data transfer starts with the least significant bit of byte 0 and continues through the scratchpad until the 9th byte (byte 8 – CRC) is read. The master may issue a reset to terminate reading at any time if only part of the scratchpad data is needed.

**Copy Scratchpad [48h]**

This command copies the contents of the scratchpad  $T_H$  and  $T_L$  registers (bytes 2 and 3) to EEPROM. If the device is being used in parasite power mode, within 10 $\mu$ s (max) after this command is issued the master must enable a strong pullup on the 1-Wire bus for at least 10ms as described in the [Powering The DS18S20](#) section.

**Recall E<sup>2</sup> [B8h]**

This command recalls the alarm trigger values ( $T_H$  and  $T_L$ ) from EEPROM and places the data in bytes 2 and 3, respectively, in the scratchpad memory. The master device can issue read-time slots following the Recall E<sup>2</sup> command and the DS18S20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done. The recall operation happens automatically at power-up, so valid data is available in the scratchpad as soon as power is applied to the device.

**Read Power Supply [B4h]**

The master device issues this command followed by a read-time slot to determine if any DS18S20s on the bus are using parasite power. During the read-time slot, parasite powered DS18S20s will pull the bus low, and externally powered DS18S20s will let the bus remain high. See the [Powering The DS18S20](#) section for usage information for this command.

**1-Wire Signaling**

The DS18S20 uses a strict 1-Wire communication protocol to ensure data integrity. Several signal types are defined by this protocol: reset pulse, presence pulse, write 0, write 1, read 0, and read 1. All these signals, with the exception of the presence pulse, are initiated by the bus master.

### Initialization Procedure—Reset And Presence Pulses

All communication with the DS18S20 begins with an initialization sequence that consists of a reset pulse from the master followed by a presence pulse from the DS18S20. This is illustrated in Figure 12. When the DS18S20 sends the presence pulse in response to the reset, it is indicating to the master that it is on the bus and ready to operate.

During the initialization sequence the bus master transmits (Tx) the reset pulse by pulling the 1-Wire bus low for a minimum of 480 $\mu$ s. The bus master then releases the bus and goes into receive mode (Rx). When the bus is released, the 5k $\Omega$  pullup resistor pulls the 1-Wire bus high. When the DS18S20 detects this rising edge, it waits 15 $\mu$ s to 60 $\mu$ s and then transmits a presence pulse by pulling the 1-Wire bus low for 60 $\mu$ s to 240 $\mu$ s.

### Read/Write Time Slots

The bus master writes data to the DS18S20 during write time slots and reads data from the DS18S20 during read-time slots. One bit of data is transmitted over the 1-Wire bus per time slot.

### Write Time Slots

There are two types of write time slots: "Write 1" time slots and "Write 0" time slots. The bus master uses a Write 1 time slot to write a logic 1 to the DS18S20 and a Write 0 time slot to write a logic 0 to the DS18S20. All write time slots must be a minimum of 60 $\mu$ s in duration with a minimum of a 1 $\mu$ s recovery time between individual write slots. Both types of write time slots are initiated by the master pulling the 1-Wire bus low (see Figure 13).

To generate a Write 1 time slot, after pulling the 1-Wire bus low, the bus master must release the 1-Wire bus within 15 $\mu$ s. When the bus is released, the 5k $\Omega$  pullup resistor will pull the bus high. To generate a Write 0 time slot, after pulling the 1-Wire bus low, the bus master must continue to hold the bus low for the duration of the time slot (at least 60 $\mu$ s). The DS18S20 samples the 1-Wire bus during a window that lasts from 15 $\mu$ s to 60 $\mu$ s after the master initiates the write time slot. If the bus is high during the sampling window, a 1 is written to the DS18S20. If the line is low, a 0 is written to the DS18S20.

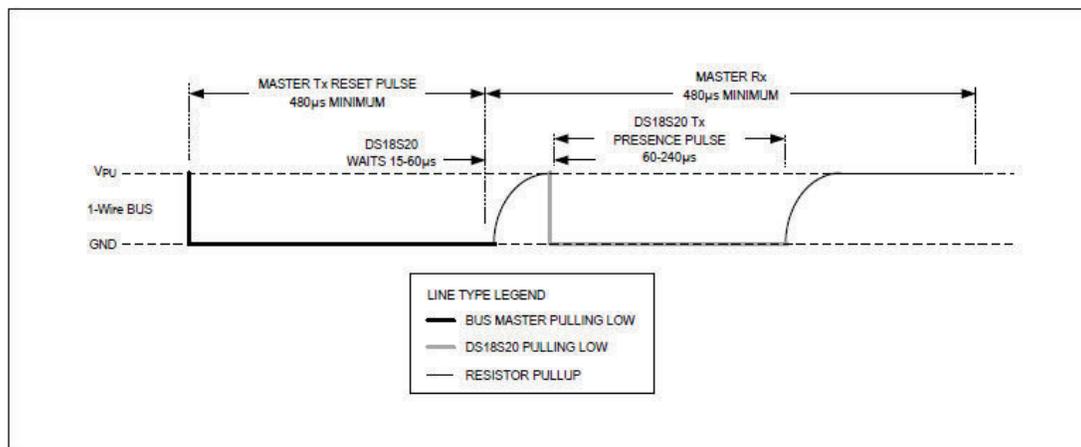


Figure 12. Initialization Timing

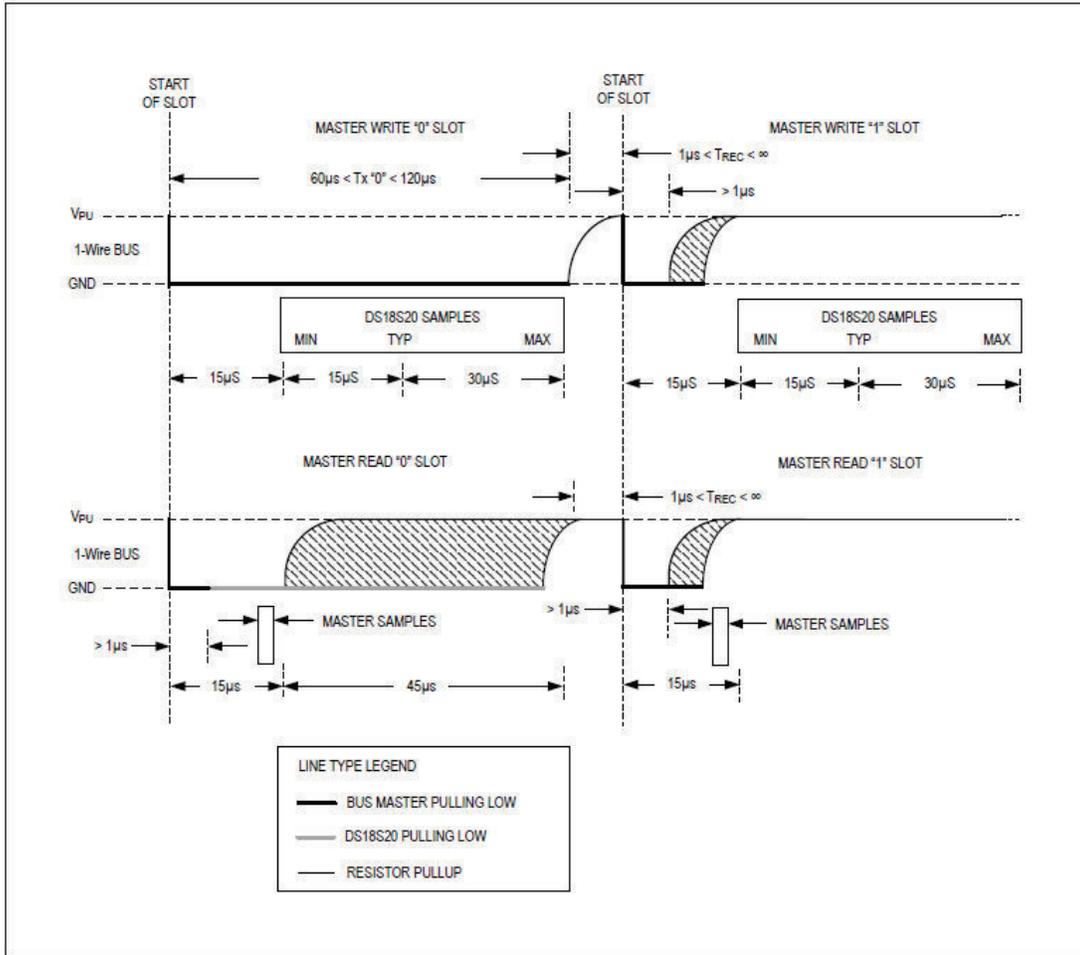


Figure 13. Read/Write Time Slot Timing Diagram

### Read-Time Slots

The DS18S20 can only transmit data to the master when the master issues read-time slots. Therefore, the master must generate read-time slots immediately after issuing a Read Scratchpad [BEh] or Read Power Supply [B4h] command, so that the DS18S20 can provide the requested data. In addition, the master can generate read-time slots after issuing Convert T [44h] or Recall E<sup>2</sup> [B8h] commands to find out the status of the operation as explained in the [DS18S20 Function Commands](#) section.

All read-time slots must be a minimum of 60µs in duration with a minimum of a 1µs recovery time between slots. A read-time slot is initiated by the master device pulling the 1-Wire bus low for a minimum of 1µs and then releasing the bus (see [Figure 13](#)). After the master initiates the

read-time slot, the DS18S20 will begin transmitting a 1 or 0 on bus. The DS18S20 transmits a 1 by leaving the bus high and transmits a 0 by pulling the bus low. When transmitting a 0, the DS18S20 will release the bus by the end of the time slot, and the bus will be pulled back to its high idle state by the pullup resistor. Output data from the DS18S20 is valid for 15µs after the falling edge that initiated the read-time slot. Therefore, the master must release the bus and then sample the bus state within 15µs from the start of the slot.

[Figure 14](#) illustrates that the sum of  $T_{INIT}$ ,  $T_{RC}$ , and  $T_{SAMPLE}$  must be less than 15µs for a read-time slot. [Figure 15](#) shows that system timing margin is maximized by keeping  $T_{INIT}$  and  $T_{RC}$  as short as possible and by locating the master sample time during read-time slots towards the end of the 15µs period.

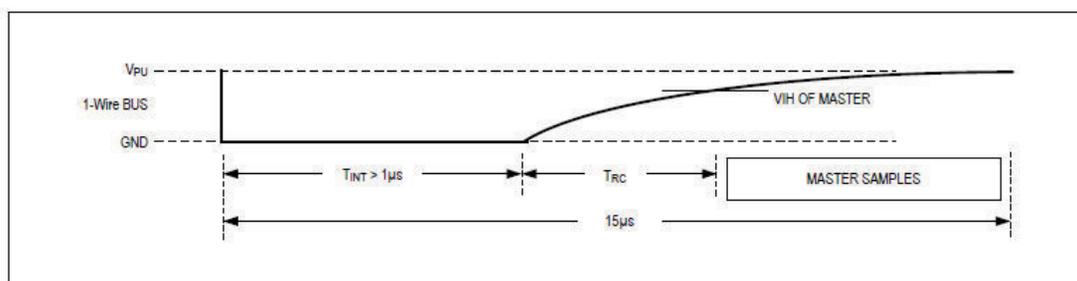


Figure 14. Detailed Master Read 1 Timing

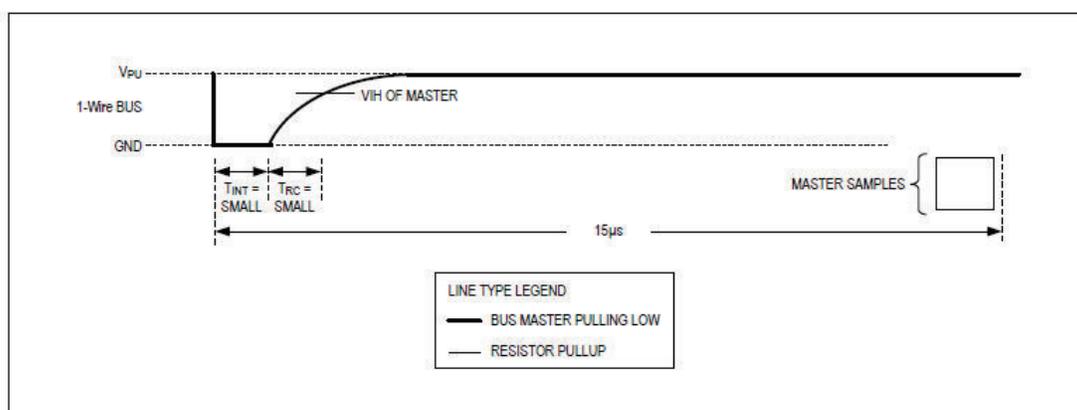


Figure 15. Recommended Master Read 1 Timing

### DS18S20 Operation Example 1

In this example there are multiple DS18S20s on the bus and they are using parasite power. The bus master initiates a temperature conversion in a specific DS18S20 and then reads its scratchpad and recalculates the CRC to verify the data.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18S20s respond with presence pulse.
Tx	55h	Master issues Match ROM command.
Tx	64-bit ROM code	Master sends DS18S20 ROM code.
Tx	44h	Master issues Convert T command.
Tx	DQ line held high by strong pullup	Master applies strong pullup to DQ for the duration of the conversion ( $t_{CONV}$ ).
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18S20s respond with presence pulse.
Tx	55h	Master issues Match ROM command.
Tx	64-bit ROM code	Master sends DS18S20 ROM code.
Tx	BEh	Master issues Read Scratchpad command.
Rx	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.

### DS18S20 Operation Example 2

In this example there is only one DS18S20 on the bus and it is using parasite power. The master writes to the  $T_H$  and  $T_L$  registers in the DS18S20 scratchpad and then reads the scratchpad and recalculates the CRC to verify the data. The master then copies the scratchpad contents to EEPROM.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18S20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	4Eh	Master issues Write Scratchpad command.
Tx	2 data bytes	Master sends two data bytes to scratchpad ( $T_H$ and $T_L$ )
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18S20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	BEh	Master issues Read Scratchpad command.
Rx	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated.
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18S20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	48h	Master issues Copy Scratchpad command.
Tx	DQ line held high by strong pullup	Master applies strong pullup to DQ for at least 10ms while copy operation is in progress.

### DS18S20 Operation Example 3

In this example there is only one DS18S20 on the bus and it is using parasite power. The bus master initiates a temperature conversion then reads the DS18S20 scratchpad and calculates a higher resolution result using the data from the temperature, COUNT REMAIN and COUNT PER °C registers.

MASTER MODE	DATA (LSB FIRST)	COMMENTS
Tx	Reset	Master issues reset pulse.
Tx	Presence	DS18S20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	44h	Master issues Convert T command.
Tx	DQ line held high by strong pullup	Master applies strong pullup to DQ for the duration of the conversion ( $t_{CONV}$ ).
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18S20 responds with presence pulse.
Tx	CCh	Master issues Skip ROM command.
Tx	BEh	Master issues Read Scratchpad command.
Rx	9 data bytes	Master reads entire scratchpad including CRC. The master then recalculates the CRC of the first eight data bytes from the scratchpad and compares the calculated CRC with the read CRC (byte 9). If they match, the master continues; if not, the read operation is repeated. The master also calculates the TEMP_READ value and stores the contents of the COUNT REMAIN and Count Per °C registers.
Tx	Reset	Master issues reset pulse.
Rx	Presence	DS18S20 responds with presence pulse.
—	—	CPU calculates extended resolution temperature using the equation in the <i>Operation—Measuring Temperature</i> section.

**ANEXO C. DATASHEET DS1307.**



## DS1307 64 x 8 Serial Real-Time Clock

[www.maxim-ic.com](http://www.maxim-ic.com)

### FEATURES

- Real-time clock (RTC) counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap-year compensation valid up to 2100
- 56-byte, battery-backed, nonvolatile (NV) RAM for data storage
- Two-wire serial interface
- Programmable squarewave output signal
- Automatic power-fail detect and switch circuitry
- Consumes less than 500nA in battery backup mode with oscillator running
- Optional industrial temperature range: -40°C to +85°C
- Available in 8-pin DIP or SOIC
- Underwriters Laboratory (UL) recognized

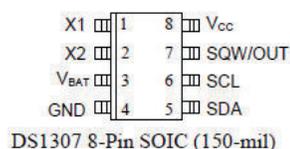
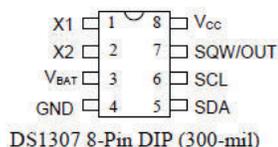
### ORDERING INFORMATION

DS1307	8-Pin DIP (300-mil)
DS1307Z	8-Pin SOIC (150-mil)
DS1307N	8-Pin DIP (Industrial)
DS1307ZN	8-Pin SOIC (Industrial)

### DESCRIPTION

The DS1307 Serial Real-Time Clock is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially via a 2-wire, bi-directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power sense circuit that detects power failures and automatically switches to the battery supply.

### PIN ASSIGNMENT



### PIN DESCRIPTION

V <sub>CC</sub>	- Primary Power Supply
X1, X2	- 32.768kHz Crystal Connection
V <sub>BAT</sub>	- +3V Battery Input
GND	- Ground
SDA	- Serial Data
SCL	- Serial Clock
SQW/OUT	- Square Wave/Output Driver



## SIGNAL DESCRIPTIONS

**V<sub>CC</sub>, GND** – DC power is provided to the device on these pins. V<sub>CC</sub> is the +5V input. When 5V is applied within normal limits, the device is fully accessible and data can be written and read. When a 3V battery is connected to the device and V<sub>CC</sub> is below 1.25 x V<sub>BAT</sub>, reads and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage. As V<sub>CC</sub> falls below V<sub>BAT</sub> the RAM and timekeeper are switched over to the external power supply (nominal 3.0V DC) at V<sub>BAT</sub>.

**V<sub>BAT</sub>** – Battery input for any standard 3V lithium cell or other energy source. Battery voltage must be held between 2.0V and 3.5V for proper operation. The nominal write protect trip point voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V<sub>BAT</sub> nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at 25°C. UL recognized to ensure against reverse charging current when used in conjunction with a lithium battery.

See “Conditions of Acceptability” at <http://www.maxim-ic.com/TechSupport/QA/ntrl.htm>.

**SCL (Serial Clock Input)** – SCL is used to synchronize data movement on the serial interface.

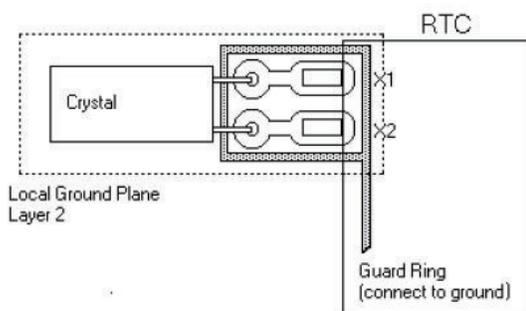
**SDA (Serial Data Input/Output)** – SDA is the input/output pin for the 2-wire serial interface. The SDA pin is open drain which requires an external pullup resistor.

**SQW/OUT (Square Wave/Output Driver)** – When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pull-up resistor. SQW/OUT will operate with either V<sub>cc</sub> or V<sub>bat</sub> applied.

**X1, X2** – Connections for a standard 32.768kHz quartz crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (CL) of 12.5pF.

For more information on crystal selection and crystal layout considerations, please consult Application Note 58, “Crystal Considerations with Dallas Real-Time Clocks.” The DS1307 can also be driven by an external 32.768kHz oscillator. In this configuration, the X1 pin is connected to the external oscillator signal and the X2 pin is floated.

## RECOMMENDED LAYOUT FOR CRYSTAL



## CLOCK ACCURACY

The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast. See Application Note 58, "Crystal Considerations with Dallas Real-Time Clocks" for detailed information.

Please review Application Note 95, "Interfacing the DS1307 with a 8051-Compatible Microcontroller" for additional information.

## RTC AND RAM ADDRESS MAP

The address map for the RTC and RAM registers of the DS1307 is shown in Figure 2. The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multi-byte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

**DS1307 ADDRESS MAP** Figure 2

00H	SECONDS
	MINUTES
	HOURS
	DAY
	DATE
	MONTH
	YEAR
07H	CONTROL
08H	RAM
3FH	56 x 8

## CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. The RTC registers are illustrated in Figure 3. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the BCD format. Bit 7 of register 0 is the clock halt (CH) bit. When this bit is set to a 1, the oscillator is disabled. When cleared to a 0, the oscillator is enabled.

**Please note that the initial power-on state of all registers is not defined. Therefore, it is important to enable the oscillator (CH bit = 0) during initial configuration.**

The DS1307 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10 hour bit (20-23 hours).

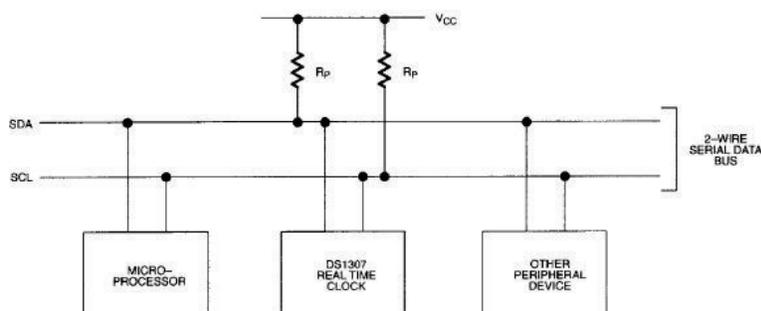
On a 2-wire START, the current time is transferred to a second set of registers. The time information is read from these secondary registers, while the clock may continue to run. This eliminates the need to re-read the registers in case of an update of the main registers during a read.



## 2-WIRE SERIAL DATA BUS

The DS1307 supports a bi-directional, 2-wire bus and data transmission protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are referred to as slaves. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1307 operates as a slave on the 2-wire bus. A typical bus configuration using this 2-wire protocol is shown in Figure 4.

### TYPICAL 2-WIRE BUS CONFIGURATION Figure 4



Figures 5, 6, and 7 detail how data is transferred on the 2-wire bus.

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is high will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

**Bus not busy:** Both data and clock lines remain HIGH.

**Start data transfer:** A change in the state of the data line, from HIGH to LOW, while the clock is HIGH, defines a START condition.

**Stop data transfer:** A change in the state of the data line, from LOW to HIGH, while the clock line is HIGH, defines the STOP condition.

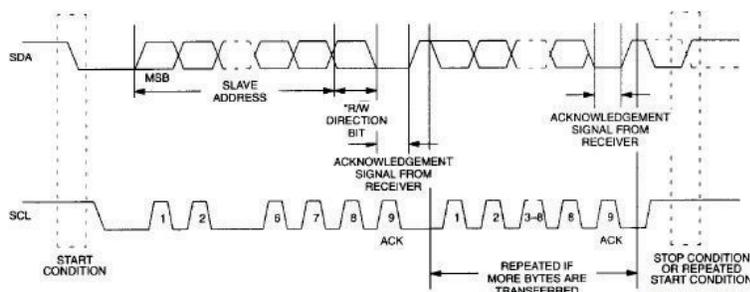
**Data valid:** The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal. The data on the line must be changed during the LOW period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between START and STOP conditions is not limited, and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth bit. Within the 2-wire bus specifications a regular mode (100kHz clock rate) and a fast mode (400kHz clock rate) are defined. The DS1307 operates in the regular mode (100kHz) only.

**Acknowledge:** Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line HIGH to enable the master to generate the STOP condition.

## DATA TRANSFER ON 2-WIRE SERIAL BUS Figure 5



Depending upon the state of the  $R/\overline{W}$  bit, two types of data transfer are possible:

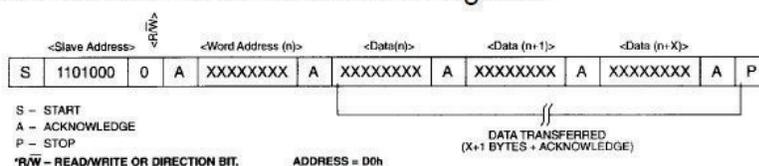
1. **Data transfer from a master transmitter to a slave receiver.** The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.
2. **Data transfer from a slave transmitter to a master receiver.** The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. This is followed by the slave transmitting a number of data bytes. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a “not acknowledge” is returned.

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred with the most significant bit (MSB) first.

The DS1307 may operate in the following two modes:

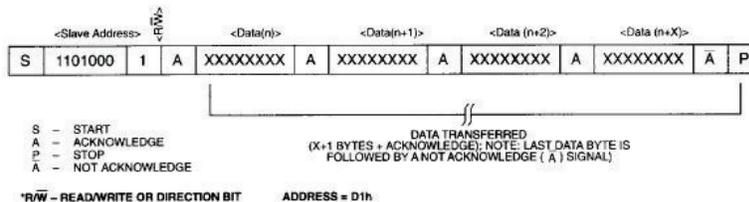
1. **Slave receiver mode (DS1307 write mode):** Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and \*direction bit (See Figure 6). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7 bit DS1307 address, which is 1101000, followed by the \*direction bit ( $R/\bar{W}$ ) which, for a write, is a 0. After receiving and decoding the address byte the device outputs an acknowledge on the SDA line. After the DS1307 acknowledges the slave address + write bit, the master transmits a register address to the DS1307. This will set the register pointer on the DS1307. The master will then begin transmitting each byte of data with the DS1307 acknowledging each byte received. The master will generate a stop condition to terminate the data write.

### DATA WRITE – SLAVE RECEIVER MODE Figure 6



2. **Slave transmitter mode (DS1307 read mode):** The first byte is received and handled as in the slave receiver mode. However, in this mode, the \*direction bit will indicate that the transfer direction is reversed. Serial data is transmitted on SDA by the DS1307 while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (See Figure 7). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7-bit DS1307 address, which is 1101000, followed by the \*direction bit ( $R/\bar{W}$ ) which, for a read, is a 1. After receiving and decoding the address byte the device inputs an acknowledge on the SDA line. The DS1307 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode the first address that is read is the last one stored in the register pointer. The DS1307 must receive a “not acknowledge” to end a read.

### DATA READ – SLAVE TRANSMITTER MODE Figure 7



**ABSOLUTE MAXIMUM RATINGS\***

Voltage on Any Pin Relative to Ground	-0.5V to +7.0V
Storage Temperature	-55°C to +125°C
Soldering Temperature	260°C for 10 seconds DIP See JPC/JEDEC Standard J-STD-020A for Surface Mount Devices

\* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

Range	Temperature	V <sub>CC</sub>
Commercial	0°C to +70°C	4.5V to 5.5V V <sub>CC1</sub>
Industrial	-40°C to +85°C	4.5V to 5.5V V <sub>CC1</sub>

**RECOMMENDED DC OPERATING CONDITIONS**

(Over the operating range\*)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V <sub>CC</sub>	4.5	5.0	5.5	V	
Logic 1	V <sub>IH</sub>	2.2		V <sub>CC</sub> + 0.3	V	
Logic 0	V <sub>IL</sub>	-0.5		+0.8	V	
V <sub>BAT</sub> Battery Voltage	V <sub>BAT</sub>	2.0		3.5	V	

\*Unless otherwise specified.

**DC ELECTRICAL CHARACTERISTICS**

(Over the operating range\*)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Leakage (SCL)	I <sub>LI</sub>			1	μA	
I/O Leakage (SDA & SQW/OUT)	I <sub>LO</sub>			1	μA	
Logic 0 Output (I <sub>OL</sub> = 5mA)	V <sub>OL</sub>			0.4	V	
Active Supply Current	I <sub>CCA</sub>			1.5	mA	7
Standby Current	I <sub>CCS</sub>			200	μA	1
Battery Current (OSC ON); SQW/OUT OFF	I <sub>BAT1</sub>		300	500	nA	2
Battery Current (OSC ON); SQW/OUT ON (32kHz)	I <sub>BAT2</sub>		480	800	nA	
Power-Fail Voltage	V <sub>PF</sub>	1.216 x V <sub>BAT</sub>	1.25 x V <sub>BAT</sub>	1.284 x V <sub>BAT</sub>	V	8

\*Unless otherwise specified.

**AC ELECTRICAL CHARACTERISTICS**

(Over the operating range\*)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
SCL Clock Frequency	$f_{SCL}$	0		100	kHz	
Bus Free Time Between a STOP and START Condition	$t_{BUF}$	4.7			$\mu s$	
Hold Time (Repeated) START Condition	$t_{HD:STA}$	4.0			$\mu s$	3
LOW Period of SCL Clock	$t_{LOW}$	4.7			$\mu s$	
HIGH Period of SCL Clock	$t_{HIGH}$	4.0			$\mu s$	
Set-up Time for a Repeated START Condition	$t_{SU:STA}$	4.7			$\mu s$	
Data Hold Time	$t_{HD:DAT}$	0			$\mu s$	4,5
Data Set-up Time	$t_{SU:DAT}$	250			ns	
Rise Time of Both SDA and SCL Signals	$t_R$			1000	ns	
Fall Time of Both SDA and SCL Signals	$t_F$			300	ns	
Set-up Time for STOP Condition	$t_{SU:STO}$	4.7			$\mu s$	
Capacitive Load for each Bus Line	$C_B$			400	pF	6
I/O Capacitance ( $T_A = 25^\circ C$ )	$C_{I/O}$		10		pF	
Crystal Specified Load Capacitance ( $T_A = 25^\circ C$ )			12.5		pF	

\*Unless otherwise specified.

**NOTES:**

1.  $I_{CCS}$  specified with  $V_{CC} = 5.0V$  and SDA, SCL = 5.0V.
2.  $V_{CC} = 0V$ ,  $V_{BAT} = 3V$ .
3. After this period, the first clock pulse is generated.
4. A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the  $V_{IHMIN}$  of the SCL signal) in order to bridge the undefined region of the falling edge of SCL.
5. The maximum  $t_{HD:DAT}$  has only to be met if the device does not stretch the LOW period ( $t_{LOW}$ ) of the SCL signal.
6.  $C_B$  – Total capacitance of one bus line in pF.
7.  $I_{CCA}$  – SCL clocking at max frequency = 100kHz.
8.  $V_{PF}$  measured at  $V_{BAT} = 3.0V$ .

**ANEXO D. DATASHEET LCD 16x2.**

## 1. Features

1. 5x8 dots with cursor
2. 16characters \*2lines display
3. 4-bit or 8-bit MPU interfaces
4. Built-in controller (ST7066 or equivalent)
5. Display Mode & Backlight Variations
6. ROHS Compliant

LCD type	<input type="checkbox"/> TN			
	<input type="checkbox"/> FSTN	<input checked="" type="checkbox"/> FSTN Negative		
	<input type="checkbox"/> STN Yellow Green	<input type="checkbox"/> STN Gray	<input type="checkbox"/> STN Blue Negative	
View direction	<input checked="" type="checkbox"/> 6 O'clock		<input type="checkbox"/> 12 O'clock	
Rear Polarizer	<input type="checkbox"/> Reflective	<input type="checkbox"/> Transflective	<input checked="" type="checkbox"/> Transmissive	
Backlight Type	<input checked="" type="checkbox"/> LED	<input type="checkbox"/> EL	<input type="checkbox"/> Internal Power	<input checked="" type="checkbox"/> 3.3V Input
		<input type="checkbox"/> CCFL	<input checked="" type="checkbox"/> External Power	<input type="checkbox"/> 5.0V Input
Backlight Color	<input checked="" type="checkbox"/> White	<input type="checkbox"/> Blue	<input type="checkbox"/> Amber	<input type="checkbox"/> Yellow-Green
Temperature Range	<input checked="" type="checkbox"/> Normal		<input type="checkbox"/> Wide	<input type="checkbox"/> Super Wide
DC to DC circuit	<input type="checkbox"/> Build-in		<input checked="" type="checkbox"/> Not Build-in	
Touch screen	<input type="checkbox"/> With		<input checked="" type="checkbox"/> Without	
Font type	<input checked="" type="checkbox"/> English-Japanese	<input type="checkbox"/> English-Europen	<input type="checkbox"/> English-Russian	<input type="checkbox"/> Other

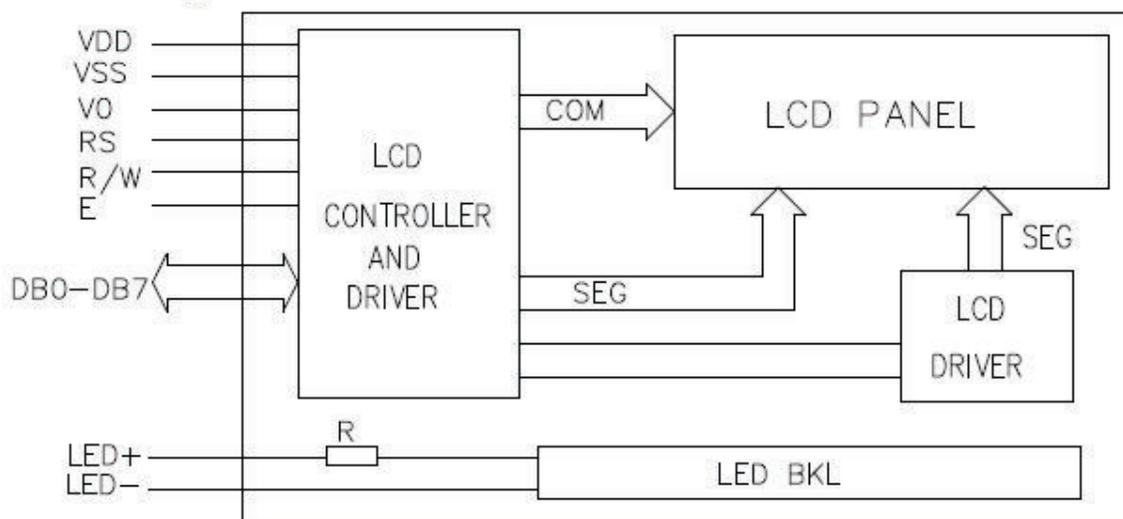
## 2. MECHANICAL SPECIFICATIONS

Module size	80.0mm(L)*36.0mm(W)* Max13.5(H)mm
Viewing area	64.5mm(L)*16.4mm(W)
Character size	3.00mm(L)*5.23mm(W)
Character pitch	3.51mm(L)*5.75mm(W)
Weight	Approx.

#### 4. Absolute maximum ratings

Item	Symbol	Standard			Unit
Power voltage	$V_{DD-VSS}$	0	-	7.0	V
Input voltage	$V_{IN}$	VSS	-	VDD	
Operating temperature range	$V_{OP}$	0	-	+50	°C
Storage temperature range	$V_{ST}$	-10	-	+60	

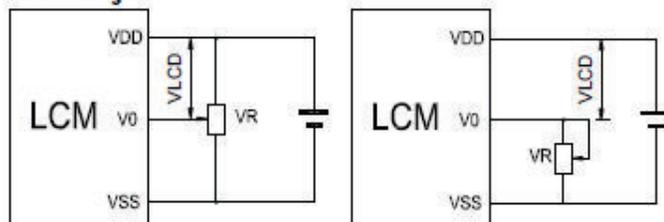
#### 5. Block diagram



#### 6. Interface pin description

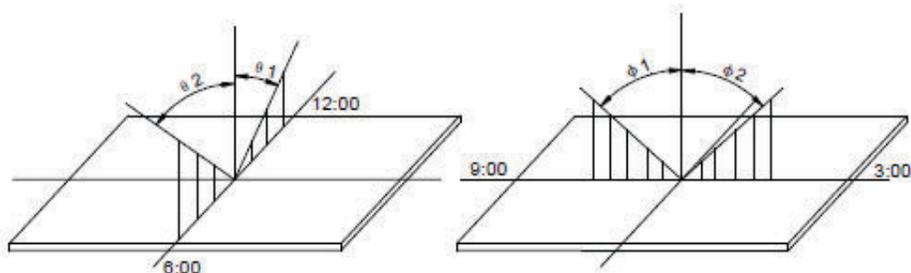
Pin no.	Symbol	External connection	Function
1	VSS	Power supply	Signal ground for LCM
2	VDD		Power supply for logic for LCM
3	V <sub>0</sub>		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7~10	DB0~DB3	MPU	Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.
11~14	DB4~DB7	MPU	Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU
15	LED+	LED BKL power supply	Power supply for BKL
16	LED-		Power supply for BKL

## 7. Contrast adjust



$V_{DD}$ - $V_0$ : LCD Driving voltage VR: 10k~20k

## 8. Optical characteristics



STN type display module ( $T_a=25^\circ\text{C}$ ,  $V_{DD}=3.3\text{V}$ )

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Viewing angle	$\theta 1$	$C_r \geq 3$		20		deg
	$\theta 2$			40		
	$\Phi 1$			35		
	$\Phi 2$			35		
Contrast ratio	$C_r$		-	10	-	-
Response time (rise)	$T_r$	-	-	200	250	ms
Response time (fall)	$T_f$	-	-	300	350	

## 9. Electrical characteristics

DC characteristics

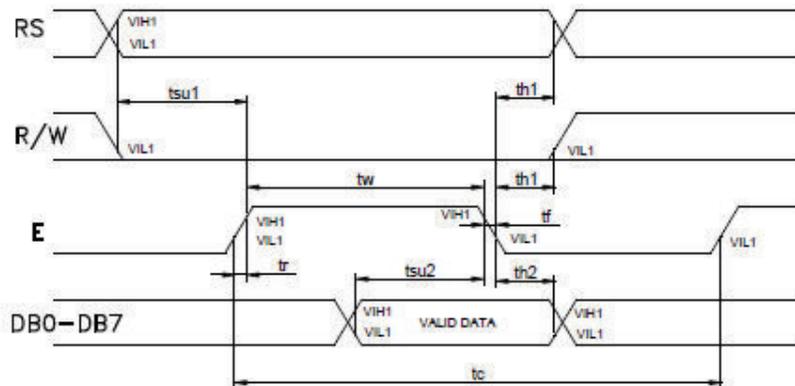
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Supply voltage for LCD	$V_{DD}-V_0$	$T_a=25^\circ\text{C}$	-	3.0	-	V
Input voltage	$V_{DD}$		3.1	3.3	3.5	
Supply current	$I_{DD}$	$T_a=25^\circ\text{C}$ , $V_{DD}=3.3\text{V}$	-	1.5	2.5	mA
Input leakage current	$I_{LKG}$		-	-	1.0	$\mu\text{A}$
"H" level input voltage	$V_{IH}$		2.2	-	$V_{DD}$	V
"L" level input voltage	$V_{IL}$	Twice initial value or less	0	-	0.6	
"H" level output voltage	$V_{OH}$	$LOH=-0.25\text{mA}$	2.4	-	-	
"L" level output voltage	$V_{OL}$	$LOH=1.6\text{mA}$	-	-	0.4	
Backlight supply voltage	$V_F$		-	3.0		
Backlight supply current	$I_{LED}$	$V_{LED}=3.3\text{V}$ $R=25\Omega$			16	mA

## 10. Timing Characteristics

Write cycle ( $T_a=25^\circ\text{C}$ ,  $V_{DD}=3.3\text{V}$ )

Parameter	Symbol	Test pin	Min.	Typ.	Max.	Unit
Enable cycle time	$t_c$	E	500	-	-	ns
Enable pulse width	$t_w$		300	-	-	
Enable rise/fall time	$t_r, t_f$		-	-	25	
RS; R/W setup time	$t_{su1}$	RS; R/W RS; R/W	100	-	-	
RS; R/W address hold time	$t_{h1}$		10	-	-	
Read data output delay	$t_{su2}$	DB0~DB7	60	-	-	
Read data hold time	$t_{h2}$		10	-	-	

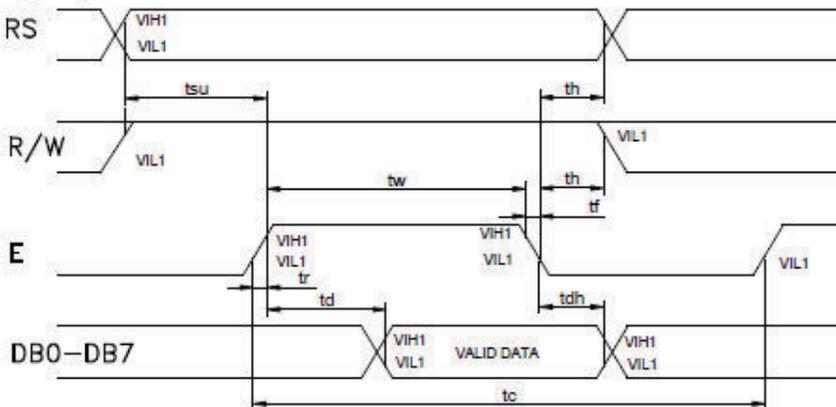
### Write mode timing diagram



Read cycle ( $T_a=25^\circ\text{C}$ ,  $V_{DD}=3.3\text{V}$ )

Parameter	Symbol	Test pin	Min.	Typ.	Max.	Unit
Enable cycle time	$t_c$	E	500	-	-	ns
Enable pulse width	$t_w$		300	-	-	
Enable rise/fall time	$t_r, t_f$		-	-	25	
RS; R/W setup time	$t_{su}$	RS; R/W RS; R/W	100	-	-	
RS; R/W address hold time	$t_h$		10	-	-	
Read data output delay	$t_d$	DB0~DB7	60	-	90	
Read data hold time	$t_{dh}$		20	-	-	

### Read mode timing diagram



## 11. FUNCTION DESCRIPTION

### 11.1 System Interface

This chip has all two kinds of interface type with MPU : 4-bit bus and 8-bit bus. 4-bit bus and 8-bit bus is selected by DL bit in the instruction register.

### 11.2 Busy Flag (BF)

When BF = "High", it indicates that the internal operation is being processed. So during this time the next instruction cannot be accepted. BF can be read, when RS = Low and R/W = High (Read Instruction Operation), through DB7 port. Before executing the next instruction, be sure that BF is not high.

### 11.3 Address Counter (AC)

Address Counter (AC) stores DDRAM/CGRAM address, transferred from IR. After writing into (reading from) DDRAM/CGRAM, AC is automatically increased (decreased) by 1. When RS = "Low" and R/W = "High", AC can be read through DB0 – DB6 ports.

### 11.4 Display Data RAM (DDRAM)

DDRAM stores display data of maximum 80 x 8 bits (80 characters). DDRAM address is set in the address counter (AC) as a hexadecimal number.

Display position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
DDRAM address	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

### 11.5 CGROM (Character Generator ROM)

CGROM has a 5 x 8 dots 204 characters pattern and a 5 x 10 dots 32 characters pattern. CGROM has 204 character patterns of 5 x 8 dots.

### 11.6 CGRAM (Character Generator RAM)

CGRAM has up to 5 . 8 dot, 8 characters. By writing font data to CGRAM, user defined characters can be used.

Character Code (DDRAM Data)								CGRAM Address					Character Patterns (CGRAM Data)																				
b8	b7	b6	b5	b4	b3	b2	b1	b0	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0											
0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	-	-	-	1	1	1	1	1											
						0	0	0				0	0	0				0	0	0													
						0	0	0				0	0	0				0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
						0	0	0				0	0	0				0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
						0	0	0				0	0	0				0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
						0	0	0				0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
						0	0	0				0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
						0	0	0				0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	-	0	0	1	0	0	1	0	0	0	-	-	-	1	1	1	1	0											
						0	0	1				0	0	1				0	0	0	1												
						0	0	1				0	1	0				0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
						0	0	1				0	1	0				0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
						0	0	1				0	1	0				0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
						0	0	1				0	1	0				0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
						0	0	1				0	1	0				0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
						0	0	1				0	1	0				0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character patterns (CGRAM Data)

#### Notes:

- Character code bits 0 to 2 correspond to CGRAM address bits 3 to 5 (3 bits: 8 types).
- CGRAM address bits 0 to 2 designate the character pattern line position. The 8<sup>th</sup> line is the cursor position

and its display is formed by a logical OR with the cursor. Maintain the 8<sup>th</sup> line data, corresponding to the cursor display position, at 0 as the cursor display. If the 8<sup>th</sup> line data is 1, 1 bit will light up the 8<sup>th</sup> line regardless of the cursor presence.

3. Character pattern row positions correspond to CGRAM data bits 0 to 4 (bit 4 being at the left).

4. As shown Table, CGRAM character patterns are selected when character code bits 4 to 7 are all 0. However, since character code bit 3 has no effect, the R display example above can be selected by either character code 00H or 08H.

5. 1 for CGRAM data corresponds to display selection and 0 to non-selection.

"-": Indicates no effect.

### 11.7 Cursor/Blink Control Circuit

It controls cursor/blink ON/OFF at cursor position.

### 11.8 Outline

To overcome the speed difference between the internal clock of ST7066 and the MPU clock, ST7066 performs internal operations by storing control information to IR or DR. The internal operation is determined according to the signal from MPU, composed of read/write and data bus (Refer to Table7).

Instructions can be divided largely into four groups:

- 1) ST7066 function set instructions (set display methods, set data length, etc.)
- 2) Address set instructions to internal RAM
- 3) Data transfer instructions with internal RAM
- 4) Others

The address of the internal RAM is automatically increased or decreased by 1.

Note: during internal operation, busy flag (DB7) is read "High".  
 Busy flag check must be preceded by the next instruction.

11.9 Instruction Table

Instruction	Instruction code										Description	Execution time (fosc=270 KHZ)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM and set DDRAM address to "00H" from AC	1.53ms
Return Home	0	0	0	0	0	0	0	0	0	1	Set DDRAM address to "00H" From AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53ms
Entry mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction And blinking of entire display	39us
Display ON/OFF control	0	0	0	0	0	0	1	D	C	B	Set display (D), cursor (C), and Blinking of cursor (B) on/off Control bit.	
Cursor or Display shift	0	0	0	0	0	1	S/C	R/L	-	-	Set cursor moving and display Shift control bit, and the Direction, without changing of DDRAM data.	39us
Function set	0	0	0	0	1	DL	N	F	-	-	Set interface data length (DL: 8-bit/4-bit), numbers of display Line (N: =2-line/1-line) and, Display font type (F: 5x11/5x8)	39us
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address Counter.	39us
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address Counter.	39us
Read busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal Operation or not can be known By reading BF. The contents of Address counter can also be read.	0us
Write data to Address	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43us
Read data From RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43us

## NOTE:

When an MPU program with checking the busy flag (DB7) is made, it must be necessary  $1/2fosc$  is necessary for executing the next instruction by the falling edge of the "E" signal after the busy flag (DB7) goes to "Low".

## 11.3 Contents

## 1) Clear display

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

Clear all the display data by writing "20H" (space code) to all DDRAM address, and set DDRAM address to "00H" into AC (address counter).

Return cursor to the original status, namely, bring the cursor to the left edge on the first line of the display. Make the entry mode increment (I/D="High").

## 2) Return home

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	-

Return home is cursor return home instruction.

Set DDRAM address to "00H" into the address counter.  
Return cursor to its original site and return display to its original status, if shifted.  
Contents of DDRAM does not change.

## 3) Entry mode set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	SH

Set the moving direction of cursor and display.

**I/D: increment / decrement of DDRAM address (cursor or blink)**

When I/D="high", cursor/blink moves to right and DDRAM address is increased by 1.

When I/D="Low", cursor/blink moves to left and DDRAM address is increased by 1.

\*CGRAM operates the same way as DDRAM, when reading from or writing to CGRAM.

**SH: shift of entire display**

When DDRAM read (CGRAM read/write) operation or SH="Low", shifting of entire display is not performed. If SH="High" and DDRAM write operation, shift of entire display is performed according to I/D value. (I/D="high". shift left, I/D="Low". Shift right).

## 4) Display ON/OFF control

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

Control display/cursor/blink ON/OFF 1 bit register.

**D: Display ON/OFF control bit**

When D="High", entire display is turned on.

When D="Low", display is turned off, but display data remains in DDRAM.

**C: cursor ON/OFF control bit**

When D="High", cursor is turned on.

When D="Low", cursor is disappeared in current display, but I/D register preserves its data.

**B: Cursor blink ON/OFF control bit**

When B="High", cursor blink is on, which performs alternately between all the "High" data and display characters at the cursor position.

When B="Low", blink is off.

## 5) Cursor or display shift

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	-	-

Shifting of right/left cursor position or display without writing or reading of display data.

This instruction is used to correct or search display data.

During 2-line mode display, cursor moves to the 2nd line after the 40th digit of the 1st line.

Note that display shift is performed simultaneously in all the lines.

When display data is shifted repeatedly, each line is shifted individually.

When display shift is performed, the contents of the address counter are not changed.

**Shift patterns according to S/C and R/L bits**

S/C	R/L	Operation
0	0	Shift cursor to the left, AC is decreased by 1
0	1	Shift cursor to the right, AC is increased by 1
1	0	Shift all the display to the left, cursor moves according to the display
1	1	Shift all the display to the right, cursor moves according to the display

## 6) Function set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	-	-

**DL: Interface data length control bit**

When DL="High", it means 8-bit bus mode with MPU.

When DL="Low", it means 4-bit bus mode with MPU. Hence, DL is a signal to select 8-bit or 4-bit bus mode.

When 4-bit bus mode, it needs to transfer 4-bit data twice.

**N: Display line number control bit**

When N="Low", 1-line display mode is set.

When N="High", 2-line display mode is set.

**F: Display line number control bit**

When F="Low", 5x8 dots format display mode is set.

When F="High", 5x11 dots format display mode.

## 7) Set CGRAM address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

Set CGRAM address to AC.

The instruction makes CGRAM data available from MPU.

## 8) Set DDRAM address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0

Set DDRAM address to AC.

This instruction makes DDRAM data available from MPU.

When 1-line display mode (N=LOW), DDRAM address is form "00H" to "4FH". In 2-line display mode (N=High), DDRAM address in the 1st line form "00H" to "27H", and DDRAM address in the 2nd line is from "40H" to "67H".

## 9) Read busy flag &amp; address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0

This instruction shows whether SPLC780D is in internal operation or not.

If the resultant BF is "High", internal operation is in progress and should wait BF is to be LOW, which by then the next instruction can be performed. In this instruction you can also read the value of the address counter.

## 10) Write data to RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

Write binary 8-bit data to DDRAM/CGRAM.

The selection of RAM from DDRAM, and CGRAM, is set by the previous address set instruction (DDRAM address set, CGRAM address set).

RAM set instruction can also determine the AC direction to RAM.

After write operation. The address is automatically increased/decreased by 1, according to the entry mode.

## 11) Read data from RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	D7	D6	D5	D4	D3	D2	D1	D0

Read binary 8-bit data from DDRAM/CGRAM.

The selection of RAM is set by the previous address set instruction. If the address set instruction of RAM is not performed before this instruction, the data that has been read first is invalid, as the direction of AC is not yet determined. If RAM data is read several times without RAM address instructions set before, read operation, the correct RAM data can be obtained from the second. But the first data would be incorrect, as there is no time margin to transfer RAM data.

In case of DDRAM read operation, cursor shift instruction plays the same role as DDRAM address set

---

instruction, it also transfers RAM data to output data register.

After read operation, address counter is automatically increased/decreased by 1 according to the entry mode.

After CGRAM read operation, display shift may not be executed correctly.

NOTE: In case of RAM write operation, AC is increased/decreased by 1 as in read operation.

At this time, AC indicates next address position, but only the previous data can be read by the read instruction.

**ANEXO E. DATASHEET MAX232.**



MAX232, MAX232I

SLLS047M – FEBRUARY 1989 – REVISED NOVEMBER 2014

## MAX232x Dual EIA-232 Drivers/Receivers

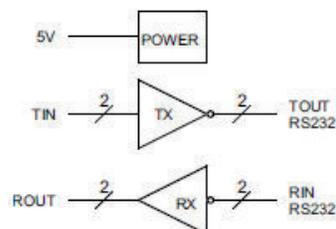
### 1 Features

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0- $\mu$ F Charge-Pump Capacitors
- Operates up to 120 kbit/s
- Two Drivers and Two Receivers
- $\pm$ 30-V Input Levels
- Low Supply Current: 8 mA Typical
- ESD Protection Exceeds JESD 22
  - 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1- $\mu$ F Charge-Pump Capacitors is Available With the MAX202 Device

### 2 Applications

- TIA/EIA-232-F
- Battery-Powered Systems
- Terminals
- Modems
- Computers

### 4 Simplified Schematic



### 3 Description

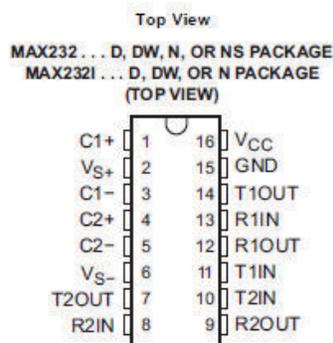
The MAX232 device is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept  $\pm$ 30-V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels.

Device Information<sup>(1)</sup>

ORDER NUMBER	PACKAGE (PIN)	BODY SIZE
MAX232x	SOIC (16)	9.90 mm x 3.91 mm
	SOIC (16)	10.30 mm x 7.50 mm
	PDIP (16)	19.30 mm x 6.35 mm
	SOP (16)	10.3 mm x 5.30 mm

(1) For all available packages, see the orderable addendum at the end of the datasheet.

## 6 Pin Configuration and Functions



### Pin Functions

PIN		TYPE	DESCRIPTION
NAME	NO.		
C1+	1	—	Positive lead of C1 capacitor
V <sub>S</sub> +	2	O	Positive charge pump output for storage capacitor only
C1-	3	—	Negative lead of C1 capacitor
C2+	4	—	Positive lead of C2 capacitor
C2-	5	—	Negative lead of C2 capacitor
V <sub>S</sub> -	6	O	Negative charge pump output for storage capacitor only
T2OUT, T1OUT	7, 14	O	RS232 line data output (to remote RS232 system)
R2IN, R1IN	8, 13	I	RS232 line data input (from remote RS232 system)
R2OUT, R1OUT	9, 12	O	Logic data output (to UART)
T2IN, T1IN	10, 11	I	Logic data input (from UART)
GND	15	—	Ground
V <sub>CC</sub>	16	—	Supply Voltage, Connect to external 5V power supply



## MAX232, MAX232I

SLLS047M – FEBRUARY 1989 – REVISED NOVEMBER 2014

www.ti.com

## 7 Specifications

### 7.1 Absolute Maximum Ratings<sup>(1)</sup>

over operating free-air temperature range (unless otherwise noted)

			MIN	MAX	UNIT
V <sub>CC</sub>	Input Supply voltage range <sup>(2)</sup>		-0.3	6	V
V <sub>S+</sub>	Positive output supply voltage range		V <sub>CC</sub> - 0.3	15	V
V <sub>S-</sub>	Negative output supply voltage range		-0.3	-15	V
V <sub>I</sub>	Input voltage range	T1IN, T2IN	-0.3	V <sub>CC</sub> + 0.3	V
		R1IN, R2IN		±30	
V <sub>O</sub>	Output voltage range	T1OUT, T2OUT	V <sub>S-</sub> - 0.3	V <sub>S+</sub> + 0.3	V
		R1OUT, R2OUT	-0.3	V <sub>CC</sub> + 0.3	
	Short-circuit duration	T1OUT, T2OUT	Unlimited		
T <sub>J</sub>	Operating virtual junction temperature			150	°C

(1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

(2) All voltages are with respect to network GND.

### 7.2 Handling Ratings

			MIN	MAX	UNIT
T <sub>stg</sub>	Storage temperature range		-65	150	°C
V <sub>(ESD)</sub>	Electrostatic discharge	Human body model (HBM), per ANSI/ESDA/JEDEC JS-001, all pins <sup>(1)</sup>	0	2000	V
		Charged device model (CDM), per JEDEC specification JESD22-C101, all pins <sup>(2)</sup>	0	1000	

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.

(2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

### 7.3 Recommended Operating Conditions

			MIN	NOM	MAX	UNIT
V <sub>CC</sub>	Supply voltage		4.5	5	5.5	V
V <sub>IH</sub>	High-level input voltage (T1IN, T2IN)		2			V
V <sub>IL</sub>	Low-level input voltage (T1IN, T2IN)				0.8	V
R1IN, R2IN	Receiver input voltage				±30	V
T <sub>A</sub>	Operating free-air temperature	MAX232	0		70	°C
		MAX232I	-40		85	

### 7.4 Thermal Information

THERMAL METRIC <sup>(1)</sup>	MAX232xD	MAX232xDW	MAX232xN	MAX232xNS	UNIT	
	SOIC 16 PINS	SOIC wide 16 PINS	PDIP 16 PINS	SOP 16 PINS		
R <sub>θJA</sub>	Junction-to-ambient thermal resistance	73	57	67	64	°C/W

(1) For more information about traditional and new thermal metrics, see the *IC Package Thermal Metrics* application report (SPRA953).

### 7.5 Electrical Characteristics — Device

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Figure 6)

PARAMETER	TEST CONDITIONS <sup>(1)</sup>	MIN	TYP <sup>(2)</sup>	MAX	UNIT	
I <sub>CC</sub>	Supply current	V <sub>CC</sub> = 5.5V, all outputs open, T <sub>A</sub> = 25°C		8	10	mA

(1) Test conditions are C1–C4 = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V

(2) All typical values are at V<sub>CC</sub> = 5 V, and T<sub>A</sub> = 25°C.



## 7.6 Electrical Characteristics — Driver

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS <sup>(1)</sup>		MIN	TYP <sup>(2)</sup>	MAX	UNIT
V <sub>OH</sub>	High-level output voltage	T1OUT, T2OUT	R <sub>L</sub> = 3 kΩ to GND	5	7		V
V <sub>OL</sub>	Low-level output voltage <sup>(3)</sup>	T1OUT, T2OUT	R <sub>L</sub> = 3 kΩ to GND		-7	-5	V
r <sub>O</sub>	Output resistance	T1OUT, T2OUT	V <sub>S+</sub> = V <sub>S-</sub> = 0, V <sub>O</sub> = ±2 V	300			Ω
I <sub>OS</sub> <sup>(4)</sup>	Short-circuit output current	T1OUT, T2OUT	V <sub>CC</sub> = 5.5 V, V <sub>O</sub> = 0 V		±10		mA
I <sub>IS</sub>	Short-circuit input current	T1IN, T2IN	V <sub>I</sub> = 0			200	μA

(1) Test conditions are C1–C4 = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V.

(2) All typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.

(3) The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

(4) Not more than one output should be shorted at a time.

## 7.7 Electrical Characteristics — Receiver

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS <sup>(1)</sup>		MIN	TYP <sup>(2)</sup>	MAX	UNIT
V <sub>OH</sub>	High-level output voltage	R1OUT, R2OUT	I <sub>OH</sub> = -1 mA	3.5			V
V <sub>OL</sub>	Low-level output voltage <sup>(3)</sup>	R1OUT, R2OUT	I <sub>OL</sub> = 3.2 mA			0.4	V
V <sub>IT+</sub>	Receiver positive-going input threshold voltage	R1IN, R2IN	V <sub>CC</sub> = 5 V, T <sub>A</sub> = 25°C		1.7	2.4	V
V <sub>IT-</sub>	Receiver negative-going input threshold voltage	R1IN, R2IN	V <sub>CC</sub> = 5 V, T <sub>A</sub> = 25°C	0.8	1.2		V
V <sub>hys</sub>	Input hysteresis voltage	R1IN, R2IN	V <sub>CC</sub> = 5 V	0.2	0.5	1	V
η	Receiver input resistance	R1IN, R2IN	V <sub>CC</sub> = 5 V, T <sub>A</sub> = 25°C	3	5	7	kΩ

(1) Test conditions are C1–C4 = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V.

(2) All typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.

(3) The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

## 7.8 Switching Characteristics

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS <sup>(1)</sup>		MIN	TYP <sup>(1)</sup>	MAX	UNIT
SR	Driver slew rate	RL = 3 kΩ to 7 kΩ, see Figure 4				30	V/μs
SR(t)	Driver transition region slew rate	see Figure 5			3		V/μs
	Data rate	One TOUT switching			120		kbit/s
t <sub>PLH</sub> Ⓢ	Receiver propagation delay time, low- to high-level output	TTL load, see Figure 3			500		ns
t <sub>PHL</sub> Ⓢ	Receiver propagation delay time, high- to low-level output	TTL load, see Figure 3			500		ns

(1) Test conditions are C1–C4 = 1 μF at V<sub>CC</sub> = 5 V ± 0.5 V.

## 10 Application and Implementation

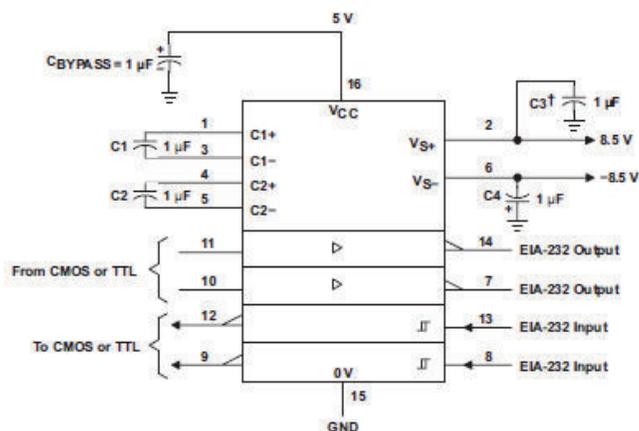
### NOTE

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

### 10.1 Application Information

For proper operation add capacitors as shown in Figure 6. Pins 9 through 12 connect to UART or general purpose logic lines. EIA-232 lines will connect to a connector or cable.

### 10.2 Typical Application



† C3 can be connected to V<sub>CC</sub> or GND.

NOTES: A. Resistor values shown are nominal.

B. Nonpolarized ceramic capacitors are acceptable. If polarized tantalum or electrolytic capacitors are used, they should be connected as shown. In addition to the 1- $\mu$ F capacitors shown, the MAX202 can operate with 0.1- $\mu$ F capacitors.

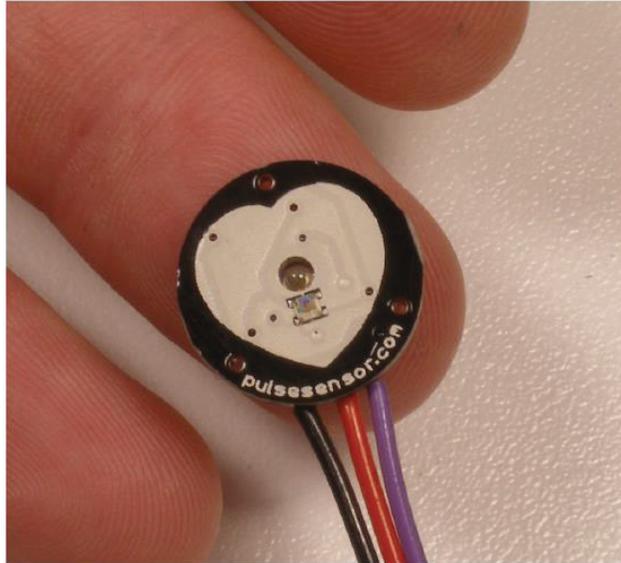
Figure 6. Typical Operating Circuit

#### 10.2.1 Design Requirements

- V<sub>CC</sub> minimum is 4.5 V and maximum is 5.5 V.
- Maximum recommended bit rate is 120 kbps.

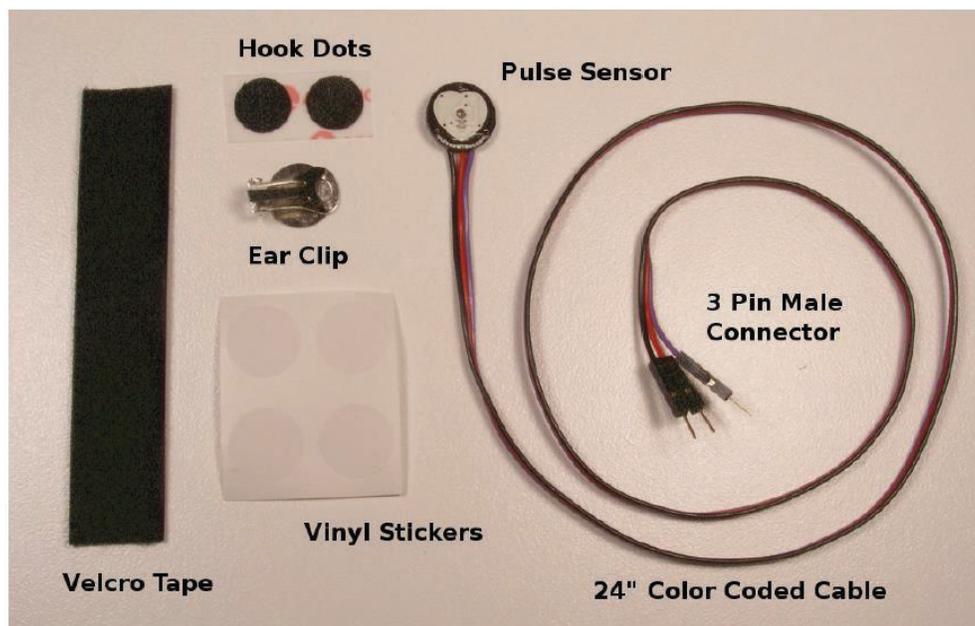
**ANEXO F. DATASHEET PULSESENSOR.**

## Pulse Sensor Getting Started Guide



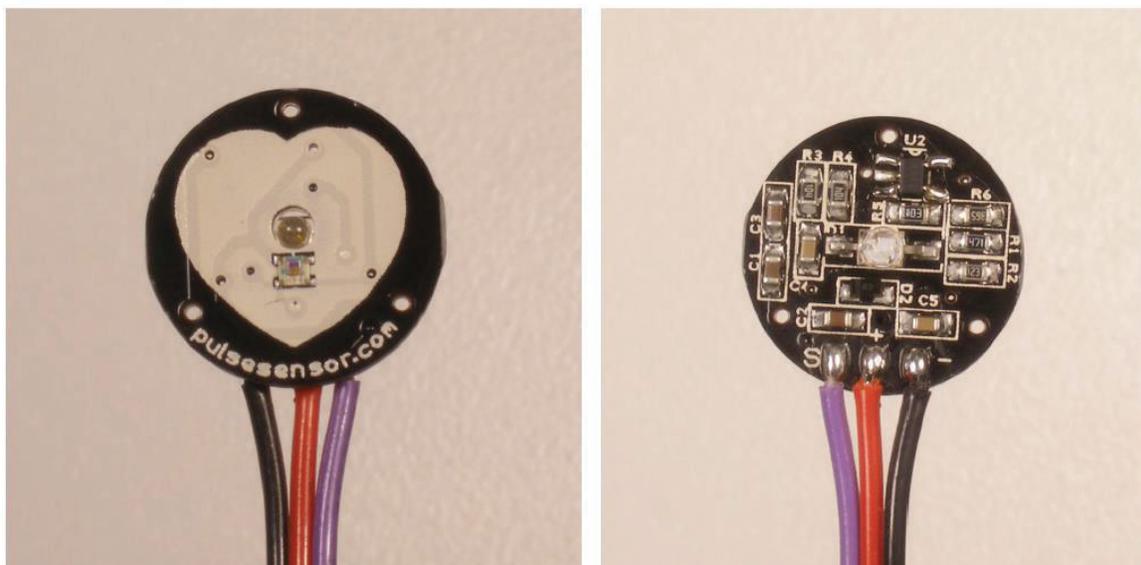
### Introduction:

Pulse Sensor is a well-designed plug-and-play heart-rate sensor for Arduino. It can be used by students, artists, athletes, makers, and game & mobile developers who want to easily incorporate live heart-rate data into their projects. The sensor clips onto a fingertip or earlobe and plugs right into Arduino with some jumper cables. It also includes an open-source monitoring app that graphs your pulse in real time.



The Pulse Sensor Kit includes:

- 1) A 24-inch Color-Coded Cable, with (male) header connectors. You'll find this makes it easy to embed the sensor into your project, and connect to an Arduino. No soldering is required.
- 2) An Ear Clip, perfectly sized to the sensor. We searched many places to find just the right clip. It can be hot-glued to the back of the sensor and easily worn on the earlobe.
- 3) 2 Velcro Dots. These are 'hook' side and are also perfectly sized to the sensor. You'll find these velcro dots very useful if you want to make a velcro (or fabric) strap to wrap around a finger tip.
- 4) Velcro strap to wrap the Pulse Sensor around your finger.
- 4) 3 Transparent Stickers. These are used on the front of the Pulse Sensor to protect it from oily fingers and sweaty earlobes.
- 5) The Pulse Sensor has 3 holes around the outside edge which make it easy to sew it into almost anything.

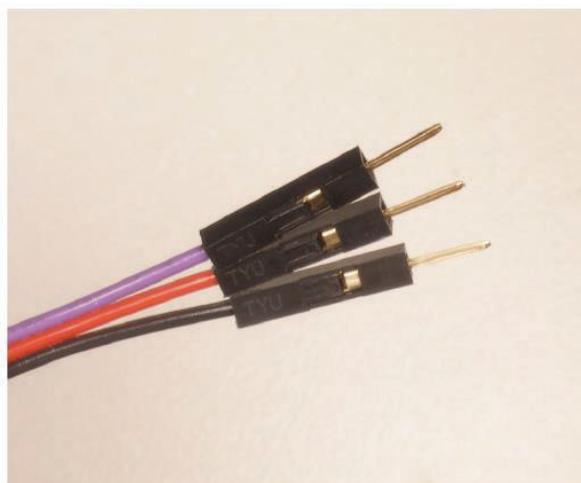


## Let's get started with Pulse Sensor Anatomy

The front of the sensor is the pretty side with the Heart logo. This is the side that makes contact with the skin. On the front you see a small round hole, which is where the LED shines through from the back, and there is also a little square just under the LED. The square is an ambient light sensor, exactly like the one used in cellphones, tablets, and laptops, to adjust the screen brightness in different light conditions. The LED shines light into the fingertip or earlobe, or other capillary tissue, and sensor reads the light that bounces back. The back of the sensor is where the rest of the parts are mounted. We put them there so they would not get in the way of the of the sensor on the front. Even the LED we are using is a reverse mount LED. For more about the circuit functionality, check out the Hardware page.[needs link]

The cable is a 24" flat color coded ribbon cable with 3 male header connectors.

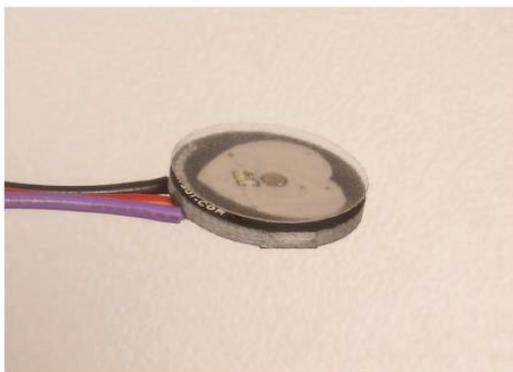
RED wire = +3V to +5V  
 BLACK wire = GND  
 PURPLE wire = Signal



The Pulse Sensor can be connected to arduino, or plugged into a breadboard. Before we get it up and running, we need to protect the exposed circuitry so you can get a reliable heart beat signal.

## Preparing the Pulse Sensor

Before you really start using the sensor you want to insulate the board from your (naturally) sweaty/oily fingers. The Pulse Sensor is an exposed circuit board, and if you touch the solder points, you could short the board, or introduce unwanted signal noise. We will use a thin film of vinyl to seal the sensor side. Find the small page of four clear round stickers in your kit, and peel one off. Then center it on the Pulse Sensor. It should fit perfectly.



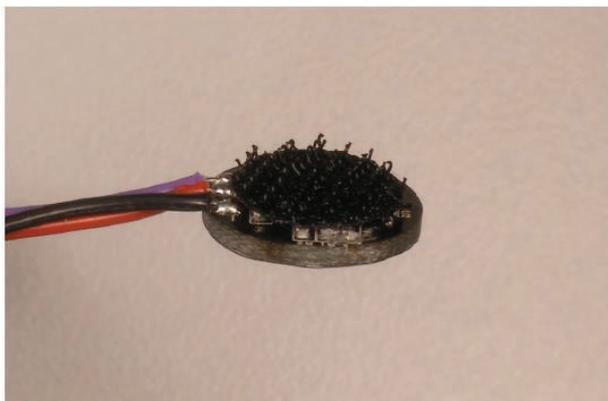
When you are happy with the way it's lined up, squeeze it onto the face all at once! The sticker (made of vinyl) will kind of stretch over the sensor and give it a nice close fit. If you get a wrinkle, don't worry, just press it down really hard and it should stick. We gave you 4, so you can replace it if necessary.

That takes care of the front side. The vinyl sticker offers very good protection for the underlying circuit, and we rate it 'water resistant'. meaning: it can stand to get splashed on, but don't throw it in the pool!

If this is your first time working with Pulse Sensor, you're probably eager to get started, and not sure if you want to use the ear-clip or finger-strap (or other thing). The back of the Pulse Sensor has even more exposed contacts than the front, so you need to make sure that you don't let it touch anything conductive or wet.

The easiest and quickest way to protect the back side from undesirable shorts or noise is to simply stick a velcro dot there for now. The dot will keep your parts away from the Pulse Sensor parts enough for you to get a good feel for the sensor and decide how you want to mount it. You'll find that the velcro dot comes off easily, and stores back on the little strip of plastic next to the other one.

Notice that the electrical connections are still exposed! We only recommend this as a temporary setup so you can get started. We show you how to better seal the Pulse Sensor later in this document.



## Running The Pulse Sensor Code

Get the latest Arduino and Processing Pulse Sensor software here <http://pulsesensor.com/downloads/>

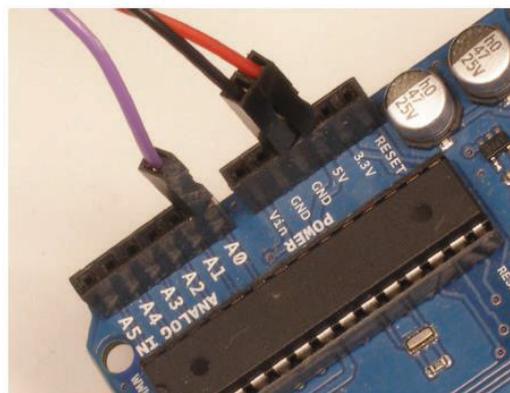
Arduino code is called "PulseSensorAmped\_Arduino-xx"

The Processing code is called "PulseSensorAmped\_Processing-xx"

We strongly advise that you DO NOT connect the Pulse Sensor to your body while your computer or arduino is being powered from the mains AC line. That goes for charging laptops and DC power supplies. Please be safe and isolate yourself from from the power grid, or work under battery power.

Connect the Pulse Sensor to: +V (red), Ground (black), and Analog Pin 0 (purple) on your favorite Arduino, or Arduino compatible device, and upload the 'PulseSensoAmped\_Arduino-xx' sketch.

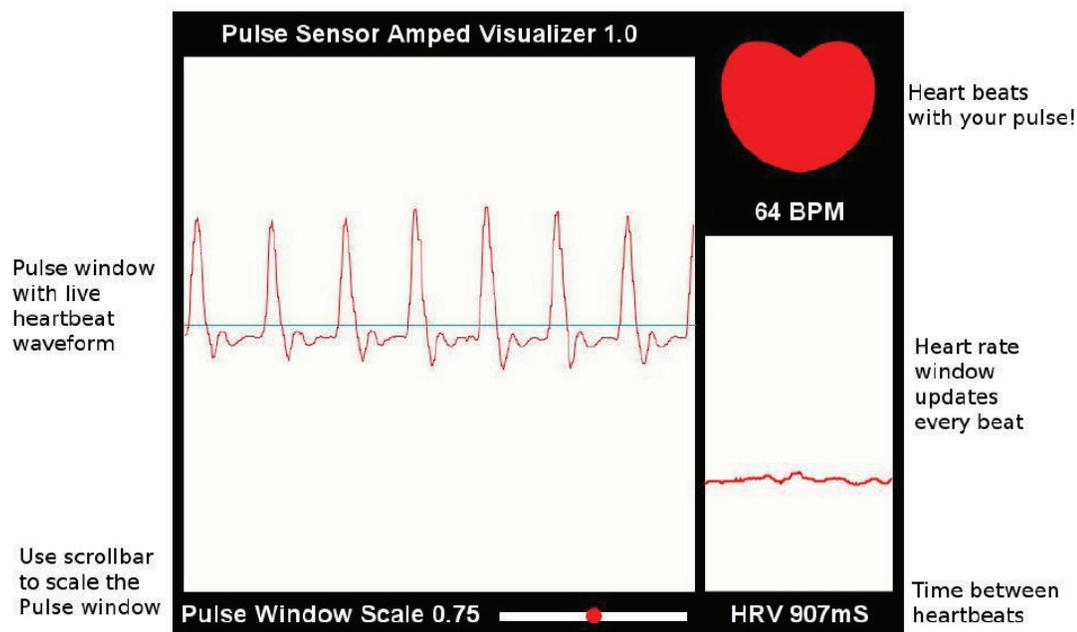
**note:** If you want to power Pulse Sensor Amped with low voltage (3.3V for example), make sure you have this line of code in the setup()  
`analogReference(EXTERNAL);`  
 Also, make sure that you apply the lower voltage to the Arduino Aref pin (next to pin 13).



After it's done uploading, you should see Arduino pin 13 blink in time with your heartbeat when you hold the sensor on your fingertip. If you grip the sensor too hard, you will squeeze all the blood out of your fingertip and there will be no signal! If you hold it too lightly, you will invite noise from movement and ambient light. Sweet

Spot pressure on the Pulse Sensor will give a nice clean signal. You may need to play around and try different parts of your body and pressures. If you see an intermittent blink, or no blink, you might be a zombie or a robot.

To view the heartbeat waveform and check your heart rate, you can use the Processing sketch that we made. Start up Processing on your computer and run the 'PulseSensorAmped\_Processing-xx' sketch. This is our data visualization software, and it looks like this.



**note:** If you get an error when starting this code, you may need to make sure you are selecting the right serial port. Check the Troubleshooting section below..

The large main window shows a graph of raw sensor data over time. The Pulse Sensor Data Window can be scaled using the scrollbar at the bottom if you have a very large or very small signal. At the right of the screen, a smaller data window graphs heart rate over time. This graph advances every pulse, and the Beats Per Minute is updated every pulse as a running average of the last ten pulses. The big red heart in the upper right also pulses to the time of your heartbeat. When you hold the Pulse Sensor to your fingertip or earlobe or (fill in body part here) you should see a nice heartbeat waveform like the one above. If you don't, and you're sure you're not a zombie, try the sensor on different parts of your body that have capillary tissue. We've had good results on the side of the nose, middle of the forehead, palm, and lower lip. We're all different, original organisms. Play around and find the best spot on you and your friends. As you are testing and getting used to the sensor. You may find that some fingers or parts of fingers are better than others. For example, I find that when I position the sensor so that the edge of the PCB is at the bottom edge of my earlobe I get an awesome signal. Also, people with cold hands or poor circulation may have a harder time reading the pulse. Run your hands under warm water, or do some jumping-jacks!

Arduino and Processing programming environments available for download here:

[www.arduino.cc](http://www.arduino.cc) [www.processing.org](http://www.processing.org)

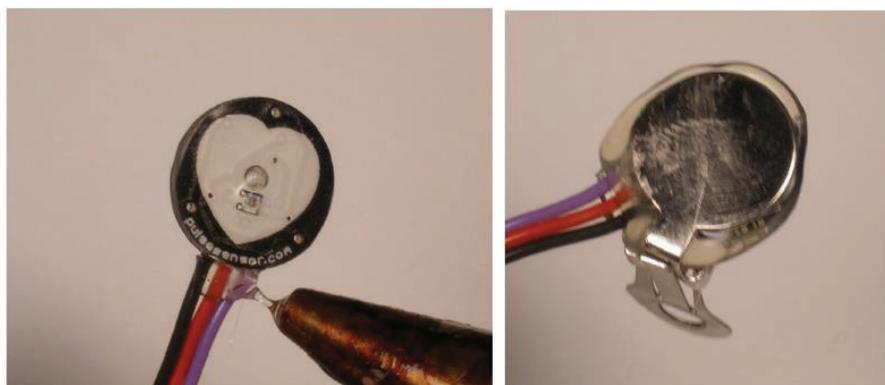
## Attaching the Ear Clip

We looked all over, and were lucky enough to find an ear clip that fits the Pulse Sensor perfectly. The earlobe is a great place to attach Pulse Sensor. Here's some instruction on how to do it.

It is important to apply some strain relief to the cable connection where it meets the Pulse Sensor PCB. The little wire connections can wear out and break (or short on something) over time. We can do this with hot glue, like we did in the previous example.

First, attach a clear vinyl sticker to the front of the Pulse Sensor if you have not already. Then, put a small dab of hot glue on the front of the cables right where they meet the PCB. Get some on the edge of the PCB too, that will help. Remember, if you don't like the blob you've made for any reason, it's easy to remove once it cools down.

Next place the Pulse Sensor face down, and put a dab of glue about the size of a kidney bean on the back as illustrated above. Center the round part of the ear clip on the sensor and press it into the hot glue. The tallest component on the back is the plastic body of the reverse mount LED, and if you press it evenly it will help keep the metal of the ear clip from contacting any of the component connections.



Allow the hot glue to ooze out around the ear clip. That will ensure good coverage. Take care not to let the hot glue cover around the ear clip hinge, as that could get in the way of it working. Trimming is easy with nail clippers (as above) or your trimming tool of choice. Don't trim the wires by mistake!!!



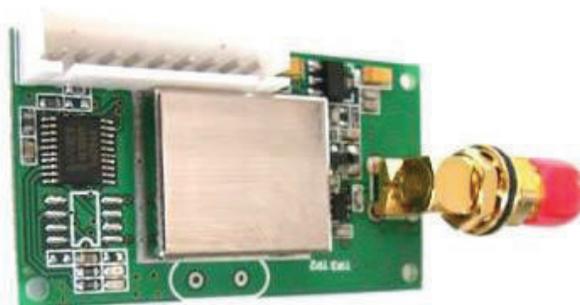
Take a good look at your work with a magnifying glass to be sure that you haven't made contact with any of the solder joints, then plug it in and test it. Hot glue is also great because it is easy to remove or re-work if you need to.

## Troubleshooting:

**ANEXO G. DATASHEET YS-1020UB.**



## YS-1020UB RF Data Transceiver



YS-1020 series Low power RF modules designed for the professional UART data transmission systems in short range. YS-1020 adapt Texas Instruments (Chipcon) CC1020 RF IC, works on ISM frequency band, half duplex integrated receiving and transmitting. Modules could directly connect with monolithic processors, PC, RS485 devices, and other UART components with RS-232, RS-485 and UART/TTL level interface port. Transparent data interface, nakedness, and wide temperature design handles most industrial application though indoor/outdoor environments.

### 1. Products Main Features

- \* Carrier frequency: 433/450/868/915MHz or ISM others optional, Free License;
- \* Interface: RS-232/ RS-485/ TTL optional;
- \* Multi-channels: 8 channels, expandable for 16/32 channels;
- \* Baud rate in air: 1200/2400/4800/9600/19200/38400bps, set before delivery;
- \* Transparent data transmission: What has been received is exactly what has been transmitted, suitable for any standard or non-standard user protocols;
- \* Interface format: 8N1/8E1/801 user-defined, or customization for other format interface;
- \* Modulation: GFSK. Based on the Gaussian Frequency Shift Keying (GFSK) modulation, High anti-interference and Low BER (Bit error Rate);
- \* Half duplex: Integration of receiver and transmitter, 10ms auto change for receiving and sending;
- \* Low power consumption and sleep function;
- \* Widen Temperature: -35℃~+75℃ (-31~167 F);
- \* Working humidity: 10%~90% relative humidity without condensation;
- \* Impedance: 50Ω (SMA antenna port, multiple antenna options available);
- \* Complying with EN 300220 and ARIB STD-T67.

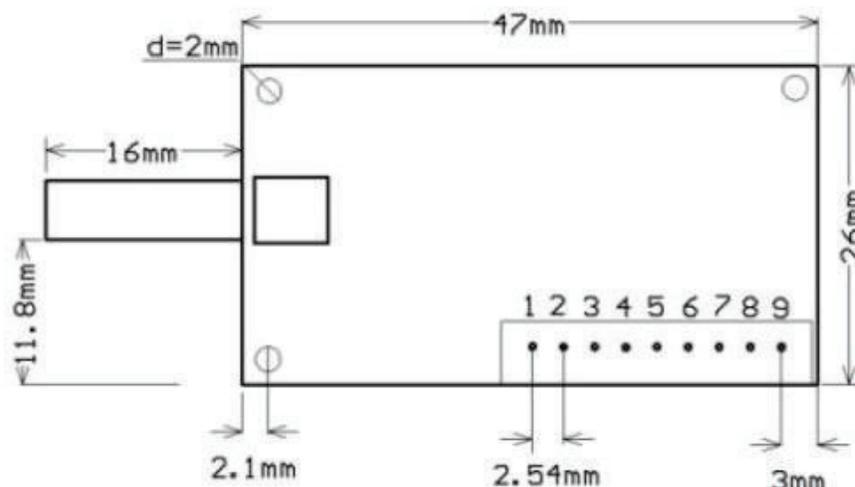
### 2. Application areas

- \* Automatic meter reading(AMR) and home automation ;
- \* Wireless smart terminal: POS, PDA,
- \* Wireless electronic display screen, LED display;
- \* Wireless remote control, Environment monitor, telemetry system;
- \* Check attendance system, Queue-management system and positioning in coal mine;
- \* RS-485 wire multi-drop system changeover wireless system;
- \* Industrial automatic data collection, Wireless Data Acquisition, Wireless sensor, SCADA;

### 3. Specifications

- \* RF power:  $\leq 50\text{mW}/17\text{dBm}$ ;
- \* Receiving current:  $<25\text{mA}$ ;
- \* Transmitting current:  $<55\text{mA}$ ;
- \* Sleep current:  $<20\mu\text{A}$ ;
- \* Power supply: DC 5v or 3.3V;
- \* Receiving sensitivity:  $-115\text{ dBm} (@9600\text{bps})$   
 $-120\text{ dBm} (@1200\text{bps})$ ;
- \* Size:  $47\text{mm}\times 26\text{mm}\times 10\text{mm}$  (without antenna port );
- \* Range:  $\leq 0.8\text{Km}$  ( $\text{BER}=10^{-3}@9600\text{bps}$ , when antenna is 2m above ground in open area),  
 $\leq 1\text{ Km}$  ( $\text{BER}=10^{-3}@1200\text{bps}$ , when antenna is 2m above ground in open area).

### 4. Installation dimension:



### 5. Interface definition:

Pin	Pin name	Description	Level	Connection with terminal	Remands
1	GND	Grounding of power supply		Ground	
2	Vcc	Power supply DC	+3,3~5,5V		
3	RXD/TTL	Serial data receiving end	TTL	TxD	
4	TXD/TTL	Serial data transmitting end	TTL	RxD	
5	DGND	Digital grounding			
6	A(TXD)	A of RS-485 or TXD of RS-232		A(RxD)	
7	B(RXD)	B of RS-485 or RXD of RS-232		B(TxD)	
8	Sleep	Sleep control (input)	TTL	Sleep signal	Low level sleep
9	Test	Ex-factory testing			

NOTE: Generally the module is in receiving status, if the Sleep pin (No.8) continuously connects low level ( $>200\text{millisecond}$ ), the module will be in sleep status, modules can not receive or transmit any data when sleep.



Only when the Sleep pin set in the state of high level ( $V_{IH} < 3.5V$ ) or hangs/empty, module can be in receiving status again. The delay time for conversion between sleeping and receiving is less than 150ms.

## 6. Setting of channel, interface, and data format:

User can change or view the module's parameter setting (interface baud rate and channel) by testing software "YSPRG.EXE" in the CD (Free). Channel 6 is default value.

### 1) Corresponding frequency points at 433MHz of 1~8 channels

Channel	Frequency	Channel	Frequency	Channel	Frequency	Channel	Frequency
1	429.0325MHZ	2	430.0325MHZ	3	431.0325MHZ	4	432.0325MHZ
5	433.0325MHZ	6	434.0325MHZ	7	435.0325MHZ	8	436.0325MHZ

### 2) Corresponding frequency points at 868MHz of 1~8 channels

Channel	Frequency	Channel	Frequency	Channel	Frequency	Channel	Frequency
1	867.0325MHZ	2	868.0325MHZ	3	869.0325MHZ	4	870.0325MHZ
5	871.0325MHZ	6	872.0325MHZ	7	873.0325MHZ	8	874.0325MHZ

## 7. Antenna configuration:

Many appropriate antennas for low power RF modules are selected for meeting different user antenna configurations. Please ask our Sales office for further information about the antenna's dimension and performance. The main options of antennas are exterior flagelliform rubber antenna with helical SMA joint, magnetic car antenna.



Standard: A0# Helical SMA antennas, L0# 9pin line

### Notes:

- ▲ Modules can share DC power supply with other equipment, Ensure the supply is stable (ideally <math><10\text{mVpk}</math> ripple).
- ▲ Keep the module away from other EMF generating components.
- ▲ Match 50Ω, 1/4wave antenna, high mount the antenna as close to the module as possible. Set antenna more than 2m above ground in open area to reach optimal range.