



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
FACULTAD TÉCNICA PARA EL DESARROLLO  
INGENIERIA ELECTRONICA EN CONTROL Y AUTOMATISMO**

**TESIS DE GRADO**

**Previo a la obtención del título de:  
INGENIERO ELECTRÓNICO EN CONTROL Y AUTOMATISMO**

**TÍTULO:**

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO PARA EL CONTROL Y  
REGISTRO DE TEMPERATURA A TRAVÉS DE LA COMUNICACIÓN ENTRE  
LOS SENSORES INTELIGENTES DS18B20 Y SU PROTOCOLO DE  
COMUNICACIÓN ONE-WIRE, ENTRE UNA TARJETA MAESTRA Y ESCLAVO,  
ENLAZADOS POR MÓDULOS INALÁMBRICOS.**

**AUTOR:**

**Tello Paute Omar Ivanoe  
Villares Velasco Alfredo Modesto**

**TUTOR:**

**Ing. Eduardo Mendoza Merchán**

**Guayaquil, Ecuador**

**2015**



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
FACULTAD TÉCNICA PARA EL DESARROLLO  
INGENIERIA ELECTRONICA EN CONTROL Y AUTOMATISMO**

**CERTIFICACIÓN**

Certifico que el presente trabajo fue realizado en su totalidad por los Srs. Tello Paute Omar Ivanoe y Villares Velasco Alfredo Modesto como requerimiento parcial para la obtención del título de INGENIERO ELECTRÓNICO EN CONTROL Y AUTOMATISMO

**TUTOR**

---

**Ing. Eduardo Mendoza Merchán**

**DIRECTOR DE LA CARRERA**

---

**Ing. Armando Heras**

**Guayaquil, Marzo del año 2015**



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
FACULTAD TÉCNICA PARA EL DESARROLLO  
INGENIERIA ELECTRONICA EN CONTROL Y AUTOMATISMO**

**DECLARACIÓN DE RESPONSABILIDAD**

**Tello Paute Omar Ivanoe  
Villares Velasco Alfredo Modesto**

**DECLARAMOS QUE:**

El proyecto de grado denominado “Diseño e implementación de un prototipo para el control y registro de temperatura a través de la comunicación entre los sensores inteligentes DS18B20 y su protocolo de comunicación ONE-WIRE, entre una tarjeta maestra y esclavo, enlazados por módulos inalámbricos”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía. Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

**Guayaquil, Marzo del año 2015**

LOS AUTORES

---

Tello Paute Omar Ivanoe

---

Villares Velasco Alfredo Modesto



**UNIVERSIDAD CATÓLICA  
DE SANTIAGO DE GUAYAQUIL  
FACULTAD TECNICA PARA EL DESARROLLO  
INGENIERIA ELECTRONICA EN CONTROL Y AUTOMATISMO**

**AUTORIZACIÓN**

**Nosotros, Tello Paute Omar Ivanoé y Villares Velasco Alfredo Modesto**

Autorizo a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del proyecto titulado: “Diseño e implementación de un prototipo para el control y registro de temperatura a través de la comunicación entre los sensores inteligentes DS18B20 y su protocolo de comunicación ONE-WIRE, entre una tarjeta maestra y esclavo, enlazados por módulos inalámbricos”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

**Guayaquil, Marzo del año 2015**

LOS AUTORES

---

Omar Ivanoé Tello Paute

---

Villares Velasco Alfredo Modesto

## **AGRADECIMIENTO I**

Agradezco a Dios y a la Virgen por ser la más fiel  
compañía de mi caminar.

Agradezco a mi madre Blanca quien siempre se  
esforzó por mi educación y formación profesional  
inculcándome valores y metas grandes.

Agradezco a mi padre quien ha sido ejemplo de  
éxito profesional y a quien debo mi desenvolvimiento  
profesional actual.

Agradezco a mi enamorada Melissa Suasnavas quien  
me ha apoyado durante mi vida profesional y universitaria.

También a mi espiritualidad Sodalicio de Vida Cristiana  
y a la comunidad que he conocido, en especial  
Manuel Morejón, Miguel Almeida, Daniel García,  
Francisco Molina, Cristian Montoya y Emilio Guillen.

Un agradecimiento especial al Eco. Carlos Miranda y  
al Ing. José Firmino por ser guía y apoyo  
profesional y espiritual. Porque con esfuerzo y dedicación  
me convertiré en el fruto de sus buenos deseos.

### **AUTOR:**

Tello Omar I.

## **AGRADECIMIENTO II**

El presente trabajo de tesis primeramente quiero agradecerte a ti Dios por bendecirme, por ser nuestra guía y por ser el ejemplo más grande de amor en este mundo.

A Mis padres por darme ejemplo de vida a seguir y mi familia fuente de apoyo constante e incondicional en toda mi vida y quiero expresar mis más grandes agradecimiento.

También quiero agradecer a mis profesores a quienes les debo gran parte de mis conocimientos porque todos han aportado con un granito de arena a mi formación, gracias a su paciencia y enseñanza.

A mi compañero de tesis un agradecimiento de todo corazón, por esta armonía grupal lo hemos logrado y a nuestro director de tesis quién por sus conocimiento, sus orientaciones, su manera de trabajar, su persistencia, su paciencia y motivación han sido fundamentales para mi formación como investigador.

Muchas gracias y que Dios los bendiga.

**AUTOR:**

Alfredo Villares V.

## **DEDICATORIA I**

Dedico esta tesis a Dios, a la Virgen  
y a mi familia Blanca Paute, Lalo Tello,  
Daniel Tello y María Fernanda Tello.  
A mi enamorada Melissa Suasnavas

**AUTOR:**

Tello Omar I.

## **DEDICATORIA II**

A mis padres con mucho amor y cariño les dedico todo mi esfuerzo y trabajo puesto para la realización de esta tesis.

Quienes Se han preocupado de mi desde el momento en que llegué a este mundo, me han formado para saber cómo luchar y no rendirse, mucho años después, sus enseñanzas no cesan, y aquí estoy, con un nuevo logro exitosamente conseguido, mi proyecto de tesis. Quiero de dictarles y expresar mis agradecimientos por tener una familia tan asombrosa.

**AUTOR:**

Alfredo Villares V.



## ÍNDICE GENERAL

CAPITULO I (ASPECTO GENERAL) .....	1
1.1 INTRODUCCIÓN .....	1
1.2 PLANTEAMIENTO DEL PROBLEMA .....	3
1.3 JUSTIFICACIÓN DE LA INVESTIGACIÓN .....	3
1.4 OBJETIVOS .....	4
1.4.1. Objetivo General .....	4
1.4.2. Objetivos Específicos .....	4
1.5 METODOLOGIA DE LA INVESTIGACION .....	5
1.6 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN .....	5
CAPÍTULO II (ANTECEDENTES Y MARCO TEÓRICO) .....	6
2.1 ANTECEDENTES .....	6
2.2 MARCO TEÓRICO .....	6
2.2.1. Tecnología 1-wire .....	6
2.2.1.1. Topologías de conexión entre dispositivos 1-wire .....	7
2.2.1.2. Protocolos y comandos .....	9
2.2.2. Sensor de temperatura digital DS18B20 .....	11
2.2.2.1. Modo de conexión .....	12
2.2.2.2. Características del sensor DS18B20 .....	14
2.2.2.3. Aplicaciones recomendadas para el Sensor Ds18B20 .....	14

2.2.2.4. Ventajas del DS18b20.....	15
2.2.2.5. Desventajas esta tecnología.....	15
2.2.3. Microcontrolador.....	16
2.2.3.1. Introducción a los microcontroladores.....	16
2.2.3.2. Arquitectura .....	16
2.2.3.3. Microcontrolador PIC16F877.....	17
2.2.3.4. Características del microcontrolador de la familia 16F87XA.....	19
2.2.3.5. Ventajas.....	20
2.2.3.6. Comunicación serie.....	21
2.2.3.7. La norma RS232.....	21
2.2.3.8. Características eléctricas de la señal.....	21
2.2.3.9. Característica mecánica de los conectores.....	22
2.2.4. Modulo Inalámbrico XBEE.....	23
2.2.4.1. ZigBee .....	24
2.2.4.2. Direcciones de los Módulos Xbee.....	24
2.2.4.3. Tipos de nodos ZigBee .....	25
2.2.4.4. Configuración del módulo Xbee.....	27
2.2.4.5. Comparativa de los módulos inalámbricos Xbee.....	28
2.2.4.6 Comunicaciones serie.....	30

2.2.4.6.1. Flujo de datos UART.....	30
2.2.4.6.2. Datos seriales.....	30
2.2.5. LCD.....	31
2.2.5.1. Descripción de los pines de un LCD alfanumérico.....	32
2.2.6. Diodo emisor de luz.....	33
2.2.6. 1. Clasificaciones y aplicaciones .....	33
2.2.7. Regulador LM7805.....	35
2.2.8. Integrado 74ls04 (NOT).....	36
2.2.9. Almacenamientos de Granos.....	37
2.2.9.1. Historia de almacenamiento.....	37
2.2.9.2. Importancia del almacenaje de los granos.....	37
2.2.9.3. Sistemas de almacenamiento.....	37
2.2.9.4. Capacidades de almacenamiento de silo.....	39
2.2.9.5. Clasificación por tamaño y geometría.....	39
2.2.9.6. Los aspectos importantes del almacenamiento .....	40
2.2.9.7. Periodos de almacenajes seguros y condiciones.....	41
CAPITULO III (DISEÑO DE HARDWARE).....	44
3.1 INTRODUCCIÓN.....	44
3.1.1. Diseño de la tarjeta en PROTEUS ISIS Professional 7.2V .....	45

3.1.1.1 Alimentación.....	45
3.1.1.2 Ruteado.....	46
3.1.2. Implementación de la tarjeta.....	48
3.1.3. Programación para comunicación usb/serial.....	51
3.1.4. Diseño del cable.....	53
3.1.5. Topología Bus.....	53
CAPITULO IV (DISEÑO DE SOFTWARE).....	61
4.1. INTRODUCCIÓN.....	61
4.1. 1. Programación del microcontrolador.....	61
4.1.1.1. La secuencia de programación.....	62
4.1.1.2. El programa del microcontrolador.....	64
4.1.2. Diseño del software de PC.....	66
CAPÍTULO V (PRESUPUESTO).....	73
5.1. INTRODUCCIÓN.....	73
CAPITULO VI (PRESENTACION DE RESULTADOS).....	75
6.1. INTRODUCCIÓN.....	75
6.1.1. Pruebas físicas.....	76
CONCLUSIONES.....	80
RECOMENDACIONES.....	81
BIBLIOGRAFÍA.....	83

GLOSARIO.....	89
ANEXOS.....	91

## ÍNDICE DE TABLAS

Tabla 2.1 Comandos de ROM utilizados por los dispositivos 1-wire.....	10
Tabla 2.2 Descripción de pines del sensor DS18b20.....	12
Tabla 2.3 Característica del sensor DS18B20.....	14
Tabla 2.4 Resolución del sensor DS18B20.....	14
Tabla 2.5 Arquitectura Von Neumann vs. Arquitectura Harvard.....	16
Tabla 2.6 Subfamilia de Microcontroladores Microchip.....	18
Tabla 2.7 Características del microcontrolador de la familia 16F87XA.....	20
Tabla 2.8 Comparativa de los módulos inalámbricos Xbee.....	29
Tabla 2.9 Descripción de los pines de un LCD alfanumérico.....	32
Tabla 2.10 Clasificación de los LED según su aplicación.....	34
Tabla 2.11 Aplicaciones de los LED.....	35
Tabla 3.1 Conexión de sensores en cable.....	54
Tabla 5.1 Presupuesto de prototipo de lectura de temperaturas.....	73
Tabla 5.2 Presupuesto de sistema con PLC.....	74
Tabla 6.1 Recolección de datos a temperatura ambiente.....	77
Tabla 6.2 Recolección de datos a temperatura ambiente con modificaciones.....	77

## ÍNDICE DE GRÁFICOS

Figura 2.1 Bus de comunicación one wire.....	7
Figura 2.2 Topologías de interconexión entre dispositivos de una red 1-wire.....	7
Figura 2.3 Time-slots de comunicación maestro esclavo.....	9
Figura 2.4 Las tres fases de transición del protocolo 1-wire.....	9
Figura 2.5 Estructura del identificador de los dispositivos de la tecnología One Wire.....	10
Figura 2.6 Sensor DS18B20.....	11
Figura 2.7 Encapsulamientos del sensor DS18B20 y distribución de pines.....	11
Figura 2.8 Diagrama de bloques del sensor DS18B20.....	12
Figura 2.9 Forma de conectar el ds18B20 con fuente externa y en modo parasito, sin fuente externa.....	13
Figura 2.10 Arquitectura Harvard.....	17
Figura 2.11 Los Empaquetado PDIP microcontrolador PIC 16F877A.....	17
Figura 2.12 Organización interna de los microcontroladores.....	19
Figura 2.13 Formato de comunicación serie.....	21
Figura 2.14 Diferencias entre los niveles de voltaje TTL y el formato RS232.....	22
Figura 2.15 conector macho DB9 y el conector del PC.....	23
Figura 2.16 Modulo Inalámbrico XBEE.....	23
Figura 2.17 Dirección del módulo inalámbrico XBEE.....	25
Figura 2.18 Red de malla ZigBee.....	26
Figura 2.19 herramienta de Digi, X-CTU.....	28
Figura 2.20 Diagrama de flujo de datos del sistema en un entorno de	

interfaz UART.....	30
Figura 2.21 Paquetes de datos UART 0x1F, transmitido a través del módulo de RF.....	31
Figura 2.22 LCD de 16 x 2 caracteres.....	32
Figura 2.23 LEDES.....	33
Figura 2.24 a) Conexión típica para la serie LM78XX, b) Distribución de pines del circuito integrado Lm7885.....	35
Figura 2.25 a) simbología y tabla de verdad de la compuerta lógica b) diagrama de conexión interna del integrado 74ls08.....	36
Figura 2.26 Esquema de los silos planos y cónicos a la derecha y la bodega plana a la izquierda, y en la mitad el sistema de volteo.....	38
Figura 2.27 Típicas geometrías de contenedores.....	40
Figura 2.28 Periodo de almacenamiento seguro maíz.....	42
Figura 2.29 Condiciones favorables para hongos e insectos.....	42
Figura. 3.1 Circuito de polarización de prototipo de lectura de temperaturas.....	45
Figura 3.2 Circuito de prototipo de lectura de temperaturas.....	46
Figura 3.3 Ruteado frontal de prototipo de lectura de temperaturas.....	47
Figura 3.4 Ruteado trasero de prototipo de lectura de temperaturas.....	47
Figura 3.5 Ruteado impreso frontal de prototipo de lectura de temperaturas.....	48
Figura 3.6 Ruteado impreso trasero de prototipo de lectura de temperaturas.....	49
Figura 3.7 Ruteado impreso frontal armado de prototipo de lectura de temperaturas.....	50
Figura 3.8 Ruteado impreso trasero armado de prototipo de lectura de temperaturas.....	50

Figura 3.9 Tarjeta XBee Xplorer Serial .....	51
Figura 3.10 Cable de conversión USB a Serial .....	52
Figura 3.11 Modulo de conexión inalámbrica XBee .....	52
Figura 3.12 Topología bus 1-wire .....	53
Figura 3.13 Topología bus 1-wire con diagrama de conexión para sensores DS18B20 .....	54
Figura 3.14 Implementación artesanal de sensor en cable de medición .....	55
Figura 3.15 Cable con 10 sensores DS18B20 .....	56
Figura 3.16 Numero de cables recomendado en silo de 9.1m diámetro .....	57
Figura 3.17 Numero de cables recomendado en silo de 18.2m diámetro .....	57
Figura 3.18 Área de sensibilidad de sistema de lectura de temperatura .....	58
Figura 3.19 Ubicación de cables y caja de lectura .....	59
Figura 3.20 Diseño de anclaje de cables .....	60
Figura 4.1 Diagrama de bloques .....	61
Figura 4.2 Interfaz de usuario programa lectura de temperaturas .....	67
Figura 4.3 Interfaz de usuario programa lectura de temperaturas con opción de comunicación .....	68
Figura 4.4 Interfaz de usuario programa lectura de temperaturas con opción de colección de datos automático .....	69
Figura 4.5 Interfaz de usuario programa lectura de temperaturas con opción de selección de alarmas por exceso de temperatura .....	69
Figura 6.1 Conexión del sistema y pruebas .....	76
Figura 6.2 Visualización de sistema con falla .....	78
Figura 6.3 Sistema superior con tarjetas de lectura y comunicación .....	79



## RESUMEN

El siguiente trabajo de graduación tiene como objeto el diseño e implementación de un prototipo para el control y registro de temperatura a través de la comunicación entre los sensores inteligentes DS18B20 y su protocolo de comunicación ONE-WIRE, entre una tarjeta maestra y esclavo, enlazados por módulos inalámbricos.

El control de temperatura está enfocado en adquirir los datos de temperatura de granos o semillas (maíz, soya, arroz, etc.) almacenados en silos o bodegas horizontales, este sistema permite al usuario obtener información casi en tiempo real desde un computador que esta enlazado al sistema de control, inalámbricamente, con lo cual permitirá detectar a tiempo los focos de calentamiento en los mismos, se evitara el deterioro de los granos o semillas. Facilitará tomar acciones inmediatas como dar ventilación y la movilización del grano.

Se realizaron búsquedas y pruebas para tener un conocimiento claro del funcionamiento de termometría en silos y posteriormente un estudio de los componentes y módulos electrónicos que incluyeron a lo largo de la investigación realizada.

Finalmente se logró diseñar un sistema de indicadores de temperatura con una interfaz amigable que facilita al usuario, en hardware logramos un gran desempeño en las transmisiones de datos ya que era transmitidos por módulos XBee.

Este prototipo está compuesto de componentes electrónicos muy fáciles de obtener en el mercado local así como económicos.

**Palabras claves:** One Wire, Temperatura, Sensor digital DS18B20.

## **ABSTRACT**

The following graduate work aims to design and implement a prototype for the monitoring and recording of temperature through communication between intelligent sensors DS18B20 and its communication protocol ONE-WIRE, between a master and slave card, linked by wireless modules.

Temperature control is focused on acquiring temperature data from grains or seeds (corn, soybeans, rice, etc.) stored in silos or horizontal wineries, this system allows the user to obtain information in near real time from a computer that is linked control system, wirelessly, which will allow early detection of outbreaks heating therein, the deterioration of the grains or seeds are avoided. It will facilitate immediate action as giving vent and mobilization of grain.

Searches and tests to have a clear understanding of how thermometry in silos and then a study of the components and devices included along conducted the initial research.

Finally it was possible to design a system of indicators of temperature with a friendly interface that facilitates user, hardware we achieved a great performance in data transmissions as it was borne XBee modules.

This prototype is composed of very easily available in the local market and economic electronic components.

**Keywords:** One Wire, Temperature Sensor DS18B20 digital.

# **CAPITULO 1**

## **ASPECTOS GENERALES**

### **1.1 INTRODUCCIÓN**

En Ecuador con la producción existente actual de los granos agrícolas como maíz, soya, no son suficiente para satisfacer la demanda de las fábricas de alimentos balanceados (según informe de Agencia de Noticias Andes, Abril, 2014.), por lo cual para mantener satisfecho al margen las demandas de granos que existe, las empresas optan por la importación de estos productos, lo cual conlleva adquirir grandes cantidades de estos productos y mantener por largos periodos almacenados en silos o galpones que van de 1.000 a 7.000 toneladas de granos, que en términos monetarios representa \$404.800.00 a \$2'833.600.00 (precio de maíz obtenido en MAGAP, Enero, 2014), es allí donde toma importancia el estudio para la creación de un prototipo que cumpla con las característica de poder mantener el producto en buen estado dentro de los silos o galpones, la investigación sea enfocado en la adquisición de temperatura mediante un microcontrolador enlazado por un solo cables diez sensores inteligentes DS18B20 que son sensores de temperatura que usan la tecnología 1-wire, estos sensores estarán en contacto con los granos.

El control de temperaturas en silo no es una aplicación muy común en la agroindustria, ya que las empresas o fábricas de producción requieren almacenar granos y mantener dichos granos en buen estado por periodos extendidos, ya sea para la fabricación de un producto en especial como en el caso de la BIOALIMENTAR CIA. LTDA. (BIOGRAIN “Centro de Almacenamiento y acopio de gráneles sólidos”) también se dedican hacer productos de nutrición en todas las cadenas agroalimentarias.

En el mundo en que vivimos exige continuamente evolucionar y desarrollarnos conjuntamente con él para poder mantener un equilibrio, a si también sucede en el área industrial del país en donde buscan procesos automatizado para obtener rentabilidad y ser más eficientes, por lo cual es importante el estudio de este tipo de tecnología y sistemas que será como un refuerzo de conocimientos teóricos para los futuros profesionales de las universidades y asimilen la realidad e innovaciones que se viven en el mercado.

Los problemas antes mencionados, fueron los factores fundamentales que dieron base para la realización de la presente investigación, en la cual se determinó la elaboración de un prototipo para el control y registro de temperatura a través de la comunicación entre los sensores inteligentes DS18B20 y su protocolo de comunicación ONE-WIRE, entre una tarjeta maestra y esclavo, enlazados por módulos inalámbricos.

La investigación está dividido en seis capítulos de una forma organizada y detallada:

El primer capítulo especifica la investigación utilizada, el enfoque y la metodología.

Se presenta la fundamentación teórica en el capítulo dos, que se estudió, como sensor de temperatura DS18B20, Microcontroladores “PIC”, etc.

El tercer capítulo y cuarto, se presenta el Desarrollo Experimental del Hardware y software del trabajo de Titulación.

En el Capítulo quinto y sexto, se presentan los presupuestos y resultados.

## **1.2 PLANTEAMIENTO DEL PROBLEMA**

Para las empresas o compañías agroindustriales que se dedican al almacenaje de granos y semillas, es de suma importancia realizar el monitoreo de temperatura de los granos, con el fin tomar las medidas necesarias para mantener un adecuado almacenaje.

La necesidad de elaborar un equipo electrónico que permiten medir la temperatura de un silo y el desarrollo de una interfaz para monitoreo mediante PC.

## **1.3 JUSTIFICACIÓN DE LA INVESTIGACIÓN**

Se realiza la investigación para resolver la problemática de los sistemas de monitoreo de temperatura en silos, ya que la producción de granos en especial del maíz ha mejorado el rendimiento de producción en el país y se debe cautelar la seguridad de la misma. Los silos o galpones que almacenan granos en Ecuador tienen una capacidades que van desde 1.000 a 7.000 toneladas, que en términos monetarios representa \$400.000.00 a \$3'080.00.00 en producto, los sistemas de termometría en silos existen falencia por lo cual cuando hay daños o fallos de estos sistemas en muchos casos no hay soporte porque estos sistemas de monitoreo de temperatura son traídos del exterior (Brasil, EEUU, China).

## **1.4. OBJETIVOS**

### **1.4.1. Objetivo General**

- Estudiar un sistema de termometría, basado en sensores inteligentes y una red one-wire, que permita la comunicación entre una tarjeta maestra y esclavo, enlazados por módulos inalámbricos

### **1.4.2. Objetivos Específicos**

- Realizar el diseño e implementar un prototipo para el control y registro de temperatura a través de la comunicación entre los sensores inteligentes DS18B20 y su protocolo de comunicación ONE-WIRE, entre una tarjeta maestra y esclavo, enlazados por módulos inalámbricos
- Desarrollar una comunicación inalámbrica basada en módulos de comunicación Xbee, para la comunicación entre los sensores y el PC.
- Diseñar software de control y una interfaz amigable con el usuario en una PC, que permita visualizar en tiempo real y los registro de temperatura almacenados.

## **1.5. METODOLOGÍA DE LA INVESTIGACIÓN**

En el proyecto de tesis se utilizaron métodos de acuerdo a las etapas en la investigación:

La metodología a utilizar es el empírico experimental dado de que se elaborará un diseño e implementación de un prototipo para el control y registro de temperatura a través de la comunicación entre los sensores inteligentes ds18b20 y su protocolo de comunicación one-wire, entre una tarjeta maestra y esclavo, enlazados por módulos inalámbricos.

## **1.6. TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN**

La fuente principal de la investigación es Internet, tesis relacionado a la investigación y tutoriales, esto ha proporciona un medio eficiente para la elaboración de éste proyecto. Se procura rebuscar metodologías y opciones para recolectar información, como medio de guía para constituir un diseño e implementar

Se utilizó la investigación de campo, el método deductivo y el método experimental para que de acuerdo a las instalaciones o infraestructuras del lugar determinar qué tipo de tecnología inalámbrica es la adecuada a utilizar en el proyecto.

Con la creación de este diseño de un sistema que permita medir temperaturas, supervisar, y generar un reporte, se logrará llevar el monitoreo y la activación de alarmas por medio de un software elaborado en la plataforma JAVA, recopilando la información y guardando en una base de datos, con una interfaz que permita la exportación a Excel.

## **CAPÍTULO II**

### **ANTECEDENTES Y MARCO TEÓRICO**

#### **2.1 ANTECEDENTES**

Pablo Avilés y Paul Salto (2010) diseñaron, construyeron e implementaron un triedro electrónico según la norma INEN para monitoreo de temperatura de la Empresa Fibra Acero. La tesis se estructura en tres capítulos: el primero engloba la parte teórica del proyecto. El capítulo dos abarca el diseño. Por último el capítulo tres, pruebas realizadas, contrastación y ventajas de este sistema. (Avilés, Pablo. Salto, Paúl.)

Esta tesis realizada con anterioridad se relaciona con el tema de estudio referente al uso del sensor de temperatura digital DS18B20.

#### **2.2 MARCO TEÓRICO**

A continuación, se procederá a realizar una descripción de los temas más importantes en el estudio de nuestro proyecto.

##### **2.2.1. Tecnología 1-wire**

Desarrollada a finales de los 90 por *Dallas Semiconductor*, que desde 2001 forma parte de Maxim Integrated (Elin. 1991), La tecnología 1-wire o el protocolo 1-Wire es un bus de comunicación, que permite la comunicación serial asincrónica entre un maestro y uno o varios dispositivos esclavos, por medio de un terminal de E/S de datos. La red 1-Wire a veces es llamado MicroLAN.



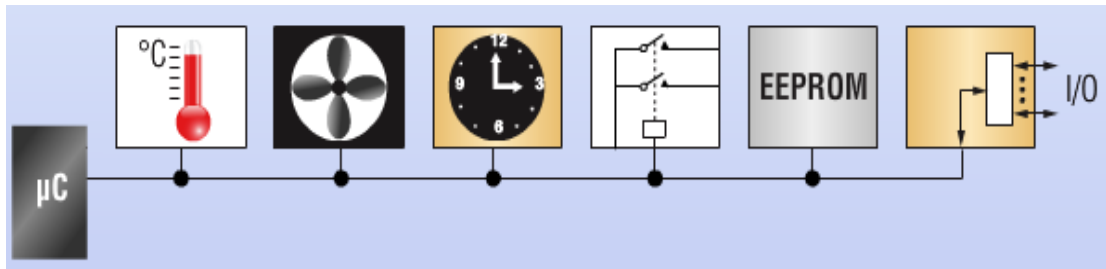


Figura 2.1 Bus de comunicación one wire  
 Fuente: tutorial en línea: <http://www.maximintegrated.com/en/products/1-wire/flash/overview/index.cfm>

Mayormente el maestro puede ser un microcontrolador. Los esclavos tienen una dirección única que los identifica grabada en el ROM por su fabricante, por lo cual garantiza un direccionamiento para que no exista confusión en el bus, cada dispositivo 1-wire posee un oscilador interno que se sincroniza con el del maestro cada vez que en la línea de datos aparece un flanco de bajada. (Avilés, Pablo. Salto, Paúl, 2010).

### 2.2.1.1. Topologías de conexión entre dispositivos 1-wire

Aunque las redes One-Wire son a menudo bastante libres en estructura, existen diferentes topologías de interconexión entre dispositivos de una red 1-wire como se presentan en el Figura 2.2

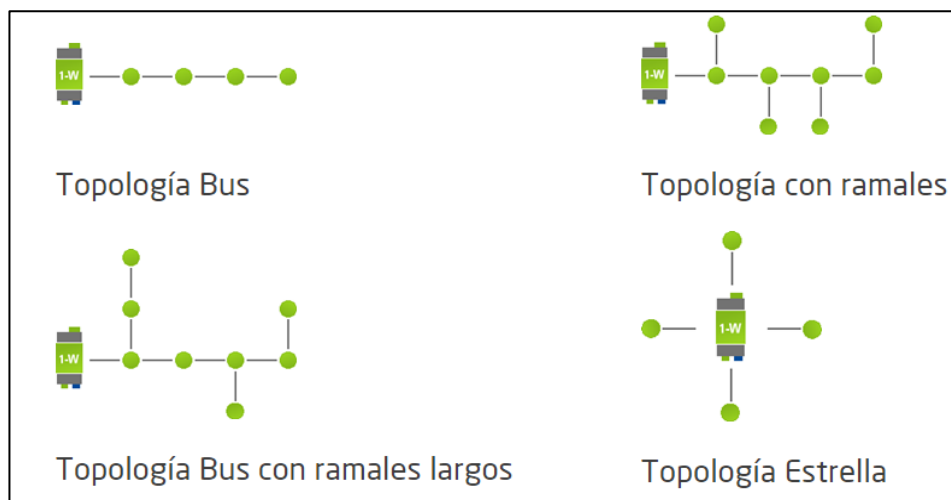


Figura 2.2 Topologías de interconexión entre dispositivos de una red 1-wire  
 Fuente: <http://www.loxone.com/eses/servicio/documentacion/extensiones/1-wire.html>

- Topología Bus

Sin ramales es la mejor solución de cableado. Permite un máximo de 350m con 20 sensores.

- Topología con ramales

Funciona igual que la de bus sin ramales (<3m). La longitud total del bus se ve reducida a 300m con 20 sensores.

- Topología Bus con ramales largos

Para el uso de esta tipo de topología (>3m) no es recomendada porque los datos del sensor no llegan a la tarjeta maestra por lo tanto es recomendado para el cableado a longitudes de 100m con 20 sensores. (Loxone. 2015)

- Topología Estrella

Está recomendada para instalaciones pequeñas porque dado a que incrementa la capacitancia equivalente en el punto central de conexión al estar las ramas conectadas en paralelos, ya que todos los sensores se unen en un punto. La longitud máxima del bus es de 100m con 20 sensores. . (Avilés, Pablo. Salto, Paúl, 2010).

- Topología en anillo

La topología en anillo no es soportada en las instalaciones 1-wire.

Los sensores conectados en anillo no se reciben datos. (Loxone. 2015)

### 2.2.1.2. Protocolos y comandos

La comunicación en el bus se hace por medio de Time-slots, el maestro inicia un time-slot generando un pulso durante un tiempo determinado, así un 1 lógico se verá como un time-slot corto, y un 0 lógico se verá como un time-slot largo. (Avilés, Pablo. Salto, Paúl, 2010). Pueden apreciarlo en el siguiente Figura 2.3

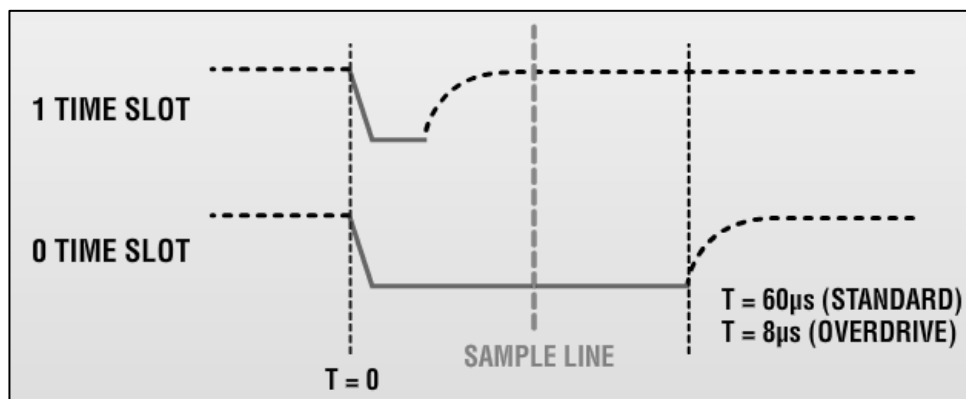


Figura 2.3 time-slots de comunicación maestro esclavo

Fuente: tutorial en línea: <http://www.maximintegrated.com/en/products/1-wire/flash/overview/index.cfm>

Todo acceso que se realice a los dispositivos 1-wire a través de la línea de datos debe seguir un estricto protocolo que comprende los siguientes pasos de la Figura 2.4 (Maxim integrate,2009).

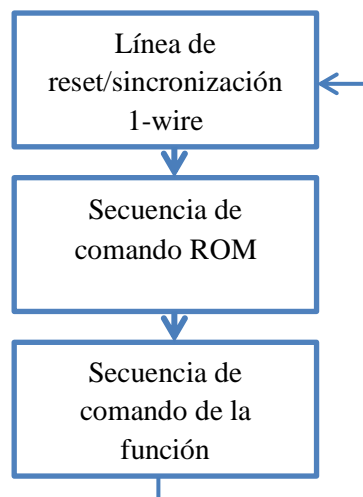


Figura 2.4 Las tres fases de transición del protocolo 1-wire

Fuente: tutorial en línea: <http://www.maximintegrated.com/en/products/1-wire/flash/overview/index.cfm>

El reset es usado para iniciar cualquier transferencia de datos entre maestro y cualquier otro elemento conectado a la red, Search Rom dicho comando permite conocer los identificadores de los dispositivos que estén conectados, entre otros. (Chávez, J. 2003).

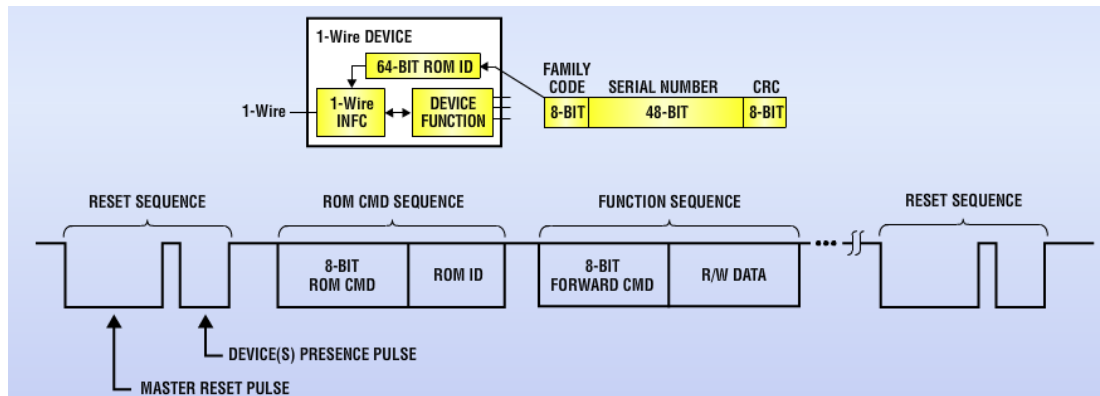


Figura 2.5 Estructura del identificador de los dispositivos de la tecnología One Wire

Fuente: tutorial en línea: <http://www.maximintegrated.com/en/products/1-wire/flash/overview/index.cfm>

Cada dispositivo posee un identificador de código 8 bytes de la familia, los 48 bit de número de serie y 8 bit CRC (código de detección de error en la transmisión) total 64 bit. (Avilés, Pablo. Salto, Paúl, 2010).

Los comandos ROM se relaciona con la búsqueda, lectura y utilización de la dirección de 64 bits "trama 1-wire" (Figura 2.4) que identifica a los esclavos.

Se detalla los comandos de Rom utilizados por los dispositivos 1-wire representado en la siguiente Tabla 2.1

INSTRUCCION	DESCRIPCION	PROTOCOLO
<b>Read ROM</b>	Lee el número de serie de un sensor cuando es el único conectado	<b>33h</b>
<b>Match ROM</b>	Hace que al siguiente comando solo responda un sensor determinado	<b>55h</b>
<b>Skip ROM</b>	Hace que todos los sensores conectados atiendan al siguiente comando	<b>CCh</b>
<b>Search ROM</b>	Comienza con el algoritmo de identificación de sensores conectados	<b>F0h</b>
<b>Alarm Search</b>	Comprueba si ha ocurrido una alarma de temperatura en algún sensor	<b>ECh</b>

Tabla 2.1 Comandos de Rom utilizados por los dispositivos 1-wire

Fuente: [http://biling.us.es/proyectos/buscar/antonio+ragel+morales/en/todo/and//en/todo/limitado\\_a/todos/entre/1970/y/2014//1](http://biling.us.es/proyectos/buscar/antonio+ragel+morales/en/todo/and//en/todo/limitado_a/todos/entre/1970/y/2014//1) . pág. 22

### 2.2.2. Sensor de temperatura digital DS18B20

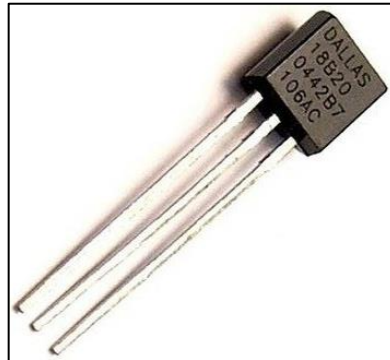


Figura 2.6 Sensor DS18B20

Fuente: <http://www.magdiblog.fr/wp-content/uploads/2013/12/DS18B20.jpg>

Es un termómetro digital de 1-Wire de Dallas Semiconductor, basada en las ventajas únicas de un solo bus, que permite a los usuarios configurar fácilmente una red de sensores, y permite múltiples puntos del circuito de medición de temperatura se vuelve simple, fiable con precisión de la medición, un amplio rango de temperaturas, tamaño pequeño y de fácil control.

En el grafico 1.7 muestra los diferentes encapsulamientos del sensor DS18B20 y distribución de pines.

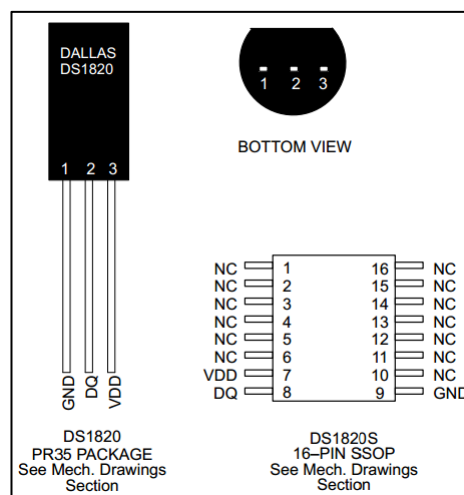


Figura 2.7 Encapsulamientos del sensor DS18B20 y distribución de pines  
Fuente: <http://www.magdiblog.fr/wp-content/uploads/2013/12/DS18B20.jpg>

El encapsulado del dispositivo a utilizarse es el TO92. Este posee tres pines. (Reinoso, Diego. Estrada, Juan. 2009)

PIN	NOMBRE	FUNCION
3	V <sub>DD</sub>	V <sub>DD</sub> Opcional, debe estar conectada a GND para operar en modo parasito
2	DQ	Entrada/salida de datos. (Open Drain) pin de interface 1-wire, también provee alimentación a si mismo cuando es usado en modo parasito.
1	GND	Tierra

Tabla 2.2 Descripción de pines del sensor DS18B20  
Fuente: Hoja técnica de Dallas Semiconductor Maxim, DS18B20

A continuación se muestra el diagrama de bloques del sensor DS18B20 en la Figura 2.8

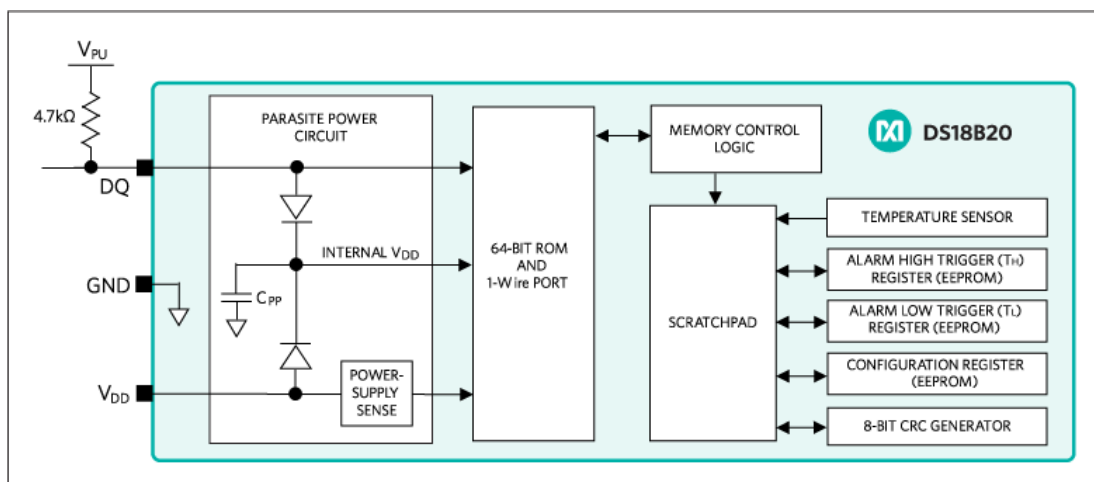


Figura 2.8 Diagrama de bloques del sensor DS18B20  
Fuente: <http://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/DS18B20.html#>

### 2.2.2.1. Modo de conexión

Existen dos modos de conexión:

Que puede ser alimentado por un suministro externo VDD, como muestra en la

Figura 2.9

Modo parásito permite funcionar sin un suministro local externo, no se recomienda el empleo por encima de +100°C dado que no puede mantener comunicaciones debido a las corrientes de escape que puede existir en esta temperatura. Así como se muestra en la Figura 2.9 (Avilés, Pablo. Salto, Paúl, 2010).

El bus de una línea requiere de una resistencia Pull-Up de 4,7KΩ, de esta forma el estado de reposo del bus será el nivel alto y nivel bajo durante más de 480 μs (microsegundo), se produce un reset. (Caicedo, Homero. Obando, Diego. 2011).

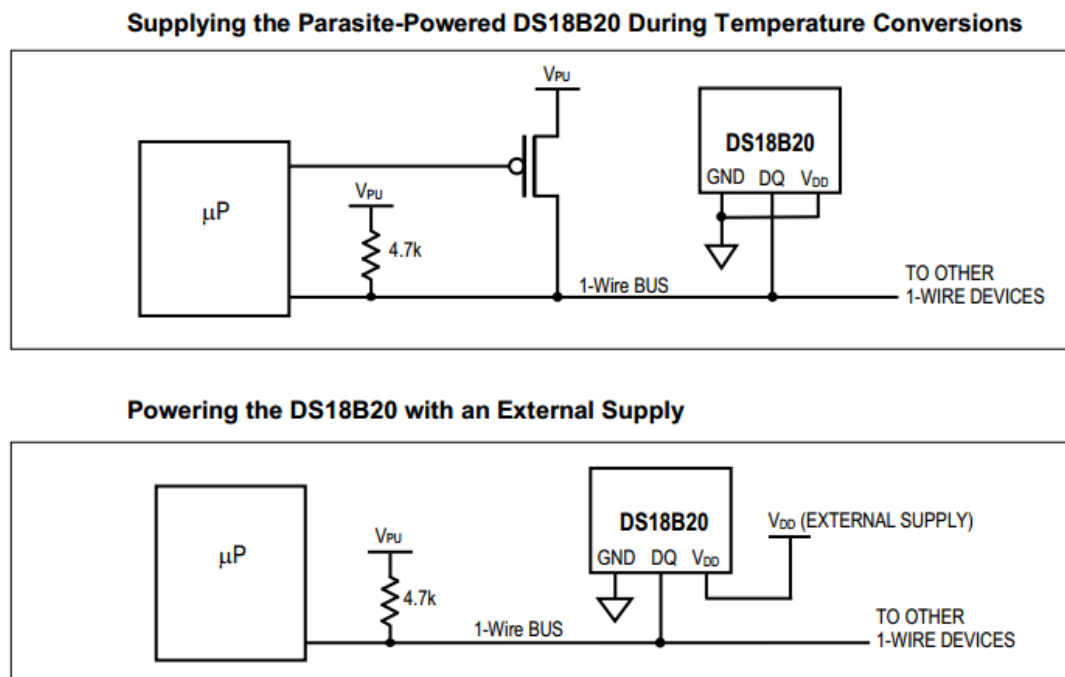


Figura 2.9 Forma de conectar el ds18B20 con fuente externa y en modo parásito, sin fuente externa

Fuente: <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

Como muestran las siguientes figuras, el termómetro puede ser alimentado por una fuente externa (con voltajes entre 3 y 5.5V) o través del propio bus 1–Wire en el llamado modo parásito.

### 2.2.2.2. Características del sensor DS18B20

En la Tabla 2.3 se muestran las características técnicas del sensor DS18B20

Voltaje de operación VDD	3 a 5V
Rango de temperatura de medición	-55°C a 125°C
Error de temperatura	± 0,5°C
Corriente máxima de operación en reposo IDDS	1nA
Corriente operación máxima IDDmax	1,5mA
Tiempo máximo de conversión del dato	750ms
Precisión	±0.5°C (de -10°C a +85°C)
Resolución:	9 a 12 bits

*Tabla 2.3 Característica del sensor DS18B20  
Fuente: Hoja técnica de Dallas Semiconductor Maxim, DS18B20*

Resolución del sensor de temperatura Ds18D20

<b>Resolución</b>	9 bits	10 bit	11 bit	12 bit
<b>Tiempo de conversión (ms)</b>	93.75	187.5	375	750
<b>LSB (° C)</b>	0.5	0.25	0,13	0.0625

*Tabla 2.4 Resolución del sensor DS18B20  
Fuente: Hoja técnica de Dallas Semiconductor Maxim, DS18B20*

### 2.2.2.3. Aplicaciones recomendadas para el Sensor Ds18B20

Se recomienda el DS18B20 para cualquier aplicación que requiere de 9 a 12 bits de resolución de temperatura. Este dispositivo ofrece mucha más flexibilidad y es más fácil de usar que el DS18S20. 1-Wire es una marca registrada de Cypress Semiconductor, Inc.

El Sensor de Temperatura DS1820 es básicamente controlar la temperatura del ambiente en el cual se encuentre instalado equipos tales como: servidores, equipos de



comunicaciones, Maquinaria agrícola, Climatización, Los dispositivos de GPS, Unidad de disco duro, Equipo Médico, Set-Top Boxes, etc.

#### **2.2.2.4. Ventajas del DS18B20**

Los sensores de temperatura DS18B20 presentan ventajas competitivas respecto a otro tipo de sensores, las cuales podemos definir en: ( Loxone. 2015)

- Protocolo especial 1-wire que permite enviar y recibir datos utilizando un solo cable, a diferencia de la mayoría de los protocolos que requiere dos vías.
- Pueden compartir el mismo bus. Cada dispositivo del bus tiene un número de serie único de 64 bits. (Tami,Camilo. Torres, Omar. Mejía, Andrés. 2014).
- Puede ser alimentado desde la línea de datos. O con fuente de alimentación externa.

#### **2.2.2.5. Desventajas esta tecnología**

- La potencia parásita a largas distancias son problemáticos con la transmisión de datos.( Laub. 2012)
- Debido a que utiliza un solo hilo para transmisión de datos reduce significativamente la velocidad de transmisión.

### 2.2.3. Microcontrolador

#### 2.2.3.1. Introducción a los microcontroladores

Un microcontrolador es un circuito integrado programable que contiene todos los componentes de un computador, el cual escapa de llevar a cabo los procesos lógicos entendiéndose como proceso lógico a las series de acciones las cuales se programan utilizando lenguaje ensamblador, Una vez programado y configurado el microcontrolador solamente lo introducimos a través de un dispositivo programador utilizando un software específico. Herrera, María. Chalco, Oscar. (2013).

Esta última característica es la que le confiere la denominación de controlador incrustado.

#### 2.2.3.2. Arquitectura

Inicialmente todos los microcontroladores adoptaron la arquitectura clásica de Von Neumann, pero actualmente se impone la arquitectura Harvard. En la Tabla 2.5 se muestran las diferencias entre estas dos arquitecturas. (González, Víctor. 2012).

	Arquitectura Von Neumann	Arquitectura Harvard
Fecha de creación	1945	1947
Primera Computadora	ENIAC	MARK I
Estructura	La UCP se comunica a través de un solo bus con un banco de memoria en donde se almacenan los códigos de instrucción del programa, como los datos que serán	Tiene un bus separado para el programa y otro para los datos. Así se evita el problema del cuello de botella de Von Newman y se obtiene un mejor
Memoria	Memoria de Datos Memoria de Instrucciones	Memoria Ram Memoria Rom Memoria Cache
Buses	Bus de Datos Bus de Direcciones Bus de Control	Bus de Datos Bus de Direcciones
Tipo de computadora	Supercomputadoras	Supercomputadoras
Semejanzas	Guardan información	Guardan información
Diferencias	Un solo bus de datos comunica Es la mas utilizada aunque es mas lenta.	Son dos buses independientes Mayor velocidad aunque no tan utilizada

Tabla 2.5 Arquitectura Von Neumann vs. Arquitectura Harvard

Fuente: <http://dianis-eliz.blogspot.com/>

La arquitectura Harvard, tiene un esquema en el que la CPU está conectado a dos memorias, por medios de dos buses separados. La memoria de instrucciones contiene el programa y es llamada memoria de programa; la otra solo almacena datos, llama memoria de datos. Ambos buses son independientes y pueden ser distintos. La Figura 2.10 muestra un esquema de esta arquitectura. (González, Víctor. 2012).

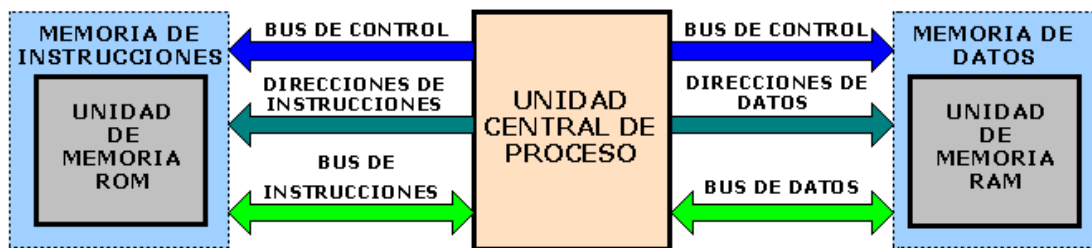


Figura 10 Arquitectura Harvard  
Fuente: <http://dianis-eliz.blogspot.com/>

### 2.2.3.3. Microcontrolador PIC16F877

#### PDIP

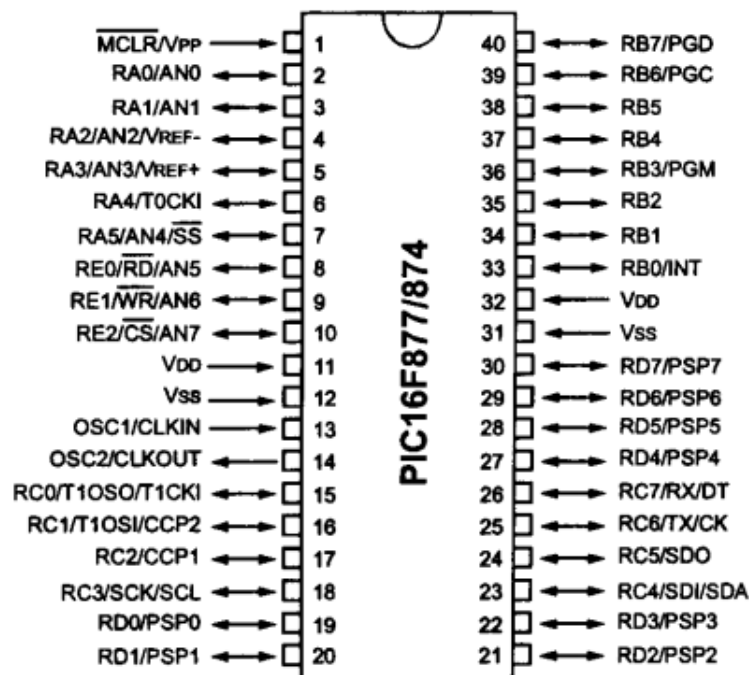


Figura 2.11 Los Empaquetado PDIP microcontrolador PIC 16F877A  
Fuente: <http://www.bairesrobotics.com.ar/data/pic16f877-guia%20detallada%20parte2.pdf>. Pág. 2

El microcontrolador PIC16F877 de Microchip pertenece a una gran familia de microcontroladores de 8 bits (bus de datos) que tienen las siguientes características generales que los distinguen de otras familias: (Bairesrobotics. 2007).

- Arquitectura Harvard

- Tecnología RISC

- Tecnología CMOS

Microchip ha dividido sus microcontroladores en tres grandes subfamilias de acuerdo al número de bits de su bus de instrucciones:

<b>Subfamilia</b>	<b>instrucciones</b>	<b>nomenclatura</b>
Base - Line	33 instrucciones de 12 bits	PIC12XXX y PIC14XXX
Mid - Range	35 instrucciones de 14 bits	PIC16XXX
High - End	58 instrucciones de 16 bits	PIC17XXX y PIC18XXX

*Tabla 2.6 Subfamilia de Microcontroladores Microchip*

*Fuente: <http://www.bairesrobotics.com.ar/data/pic16f877-guia%20detallada%20parte2.pdf>. Pág.*

*1*

Como se puede observar en la Figura 2.12 de la organización interna de los microcontroladores, estos poseen una variedad de periféricos que pueden ser utilizados según las necesidades. (Galarza, M. 2009)

Para este proyecto se utilizan algunos de estos módulos.

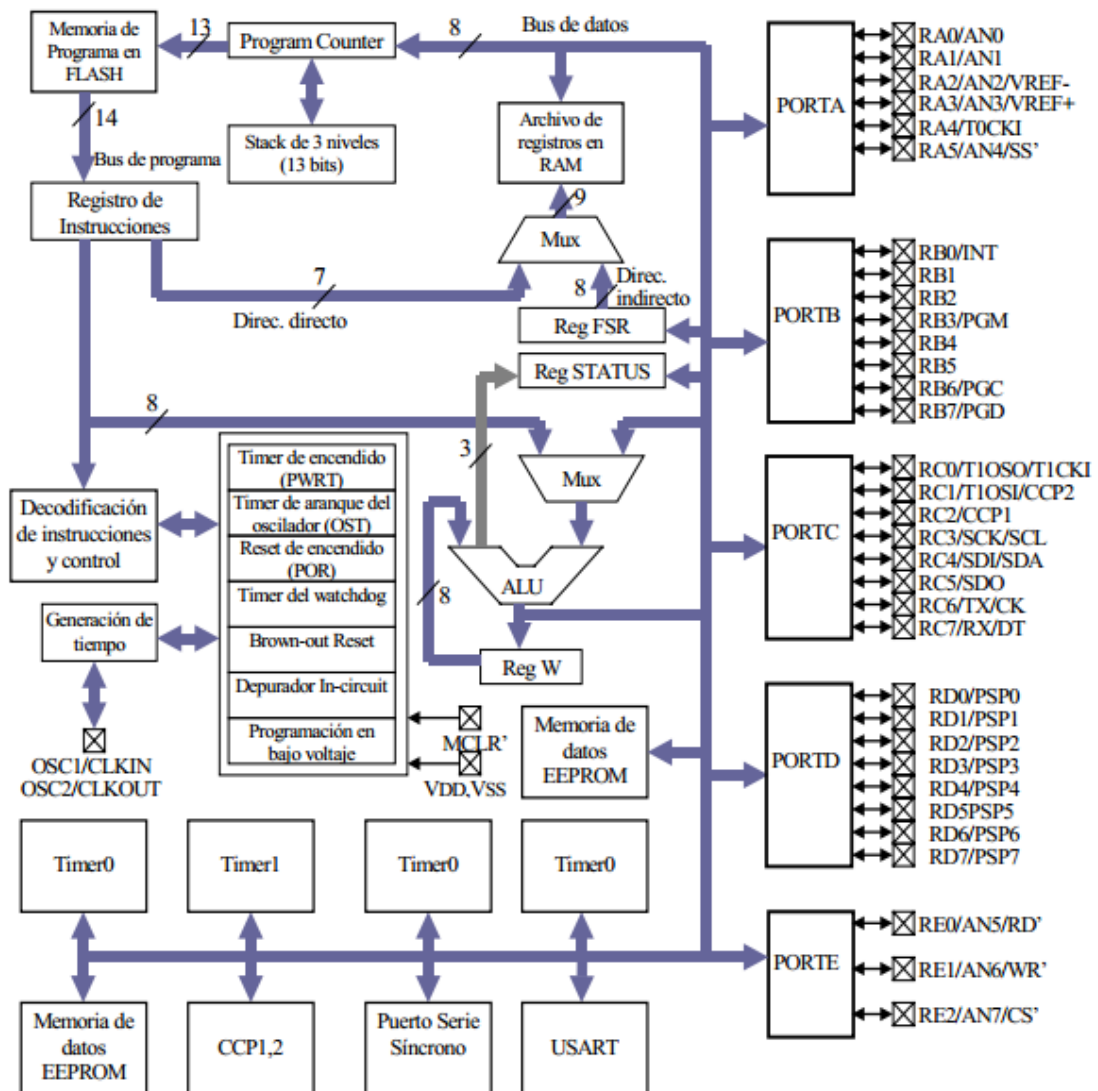


Figura 2.12 Organización interna de los microcontroladores

Fuente: <http://www.bairesrobotics.com.ar/data/pic16f877-guia%20detallada%20parte2.pdf>. Pág.

#### 2.2.3.4. Características del microcontrolador de la familia 16F87XA

Todos los dispositivos de esta familia comparten las características de arquitectura y tecnología, con las siguientes diferencias que se muestran en la Tabla 2.7

Características PICmicro Mid-Range	PIC16F873A	PIC16F874A	PIC16F876A	PIC16F877A
Frecuencia de Operación	DC-20 MHz	DC-20 MHz	DC-20 MHz	DC-20 MHz
Resets (y Retardos)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH memoria de programa (14-bit words)	4 K	4 K	4 K	4 K
Memoria de Datos (bytes)	192	192	368	368
EEPROM Memoria de Datos (bytes)	128	128	256	256
Interrupciones	14	15	14	15
I/O Puertos	A,B,C	A,B,C,D,E	A,B,C	A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM módulos	2	2	2	2
Comunicación Serial	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Comunicación Paralela	-	PSP	-	PSP
Módulo A/D 10 bits	5 entradas	8 entradas	5 entradas	8 entradas
Comparadores Análogos	2	2	2	2
Set de instrucciones	35 instrucciones	35 instrucciones	35 instrucciones	35 instrucciones

*Tabla 2.7 Características del microcontrolador de la familia 16F87XA  
Fuente: <http://bibdigital.epn.edu.ec/handle/15000/1276>. Pág. 20*

### 2.2.3.5. Ventajas

- El PIC 16f877 usa la tecnología flash (EEPROM) que permite el borrado rápido y reprogramación para acelerar la depuración de los programas.
- Posee de memoria de datos no volátil que lo usa para archivar variable y otros parámetros
- Tiene comparadores de magnitudes analógicas, puerto serie, acepta interrupciones, convertidores a/D y diversos temporizadores
- El TMR1 que lo tiene esta gama, utiliza un circuito oscilador que trabaja asincrónicamente dado que puede aumentar aunque el microcontrolador este en modo de reposo (sleep), posibilitando la implementación de un reloj en tiempo real. (González, Víctor. 2012).

### 2.2.3.6. Comunicación serie

Es un protocolo de comunicación entre dispositivos que toma este nombre debido a los bits que reciben uno detrás de otro en serie, el nivel lógico "1" denominado marca y el nivel lógico "0" denominado espacio. Ver Figura 2.13



Figura 2.13 Formato de comunicación serie  
Fuente: <http://bibdigital.epn.edu.ec/handle/15000/4869>. Pág. 13

Existen varios estándares que usan este formato y el más conocido es el estándar RS232. (González, Víctor. 2012).

### 2.2.3.7. La norma RS232

El protocolo RS-232 es una norma, que rige los parámetros de uno de los modos de comunicación serial. Se estandarizan las velocidades de transferencia de datos, la forma de control que utiliza dicha transferencia, los niveles de voltajes utilizados, el tipo de cable permitido, las distancias entre equipos, los conectores, etc. (González, Alejandro. 2012).

RS-232 es básicamente la selección de la velocidad en baudios (1200, 2400, 4800, etc.), la verificación de datos o paridad (parida par o paridad impar o sin paridad), los bits de parada luego de cada dato(1 ó 2), y la cantidad de bits por dato (7 ó 8), que se utiliza para cada símbolo o carácter enviado. (González, Alejandro. 2012).

### 2.2.3.8. Características eléctricas de la señal

Se establece que la longitud máxima entre el DTE (equipo Terminal de Datos) y el DCE (Equipo para la comunicación de datos) no debe ser mayor a los 15 metros, y la

velocidad máxima de transmisión es de 20.00 bps. Los niveles lógicos no son compatibles con los TTL. . (González, Víctor. 2012).

En la Figura 2.14 se aprecia las diferencias entre los niveles de voltaje TTL y el formato RS232.

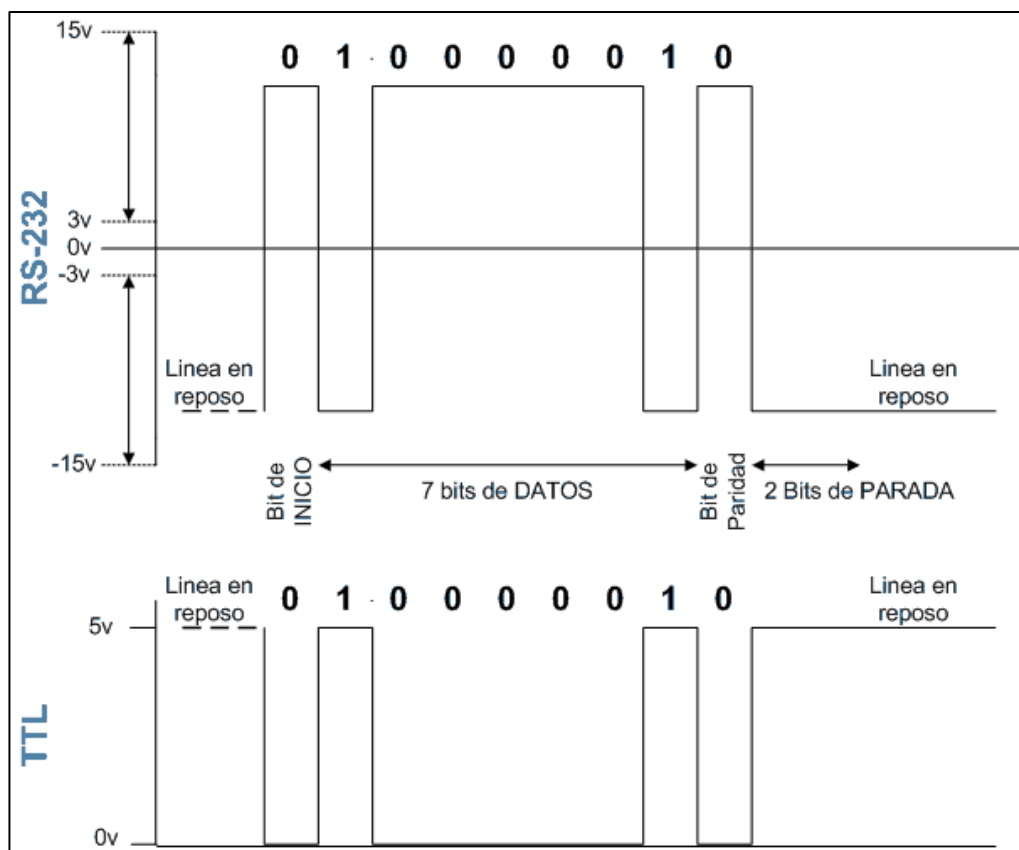


Figura 2.14 Diferencias entre los niveles de voltaje TTL y el formato RS232.  
Fuente: <http://linuxdroids.com/2010/02/08/adaptador-rs-232-a-ttl/>

### 2.2.3.9. Característica mecánica de los conectores

En la Figura 2.15 muestra los conectores macho DB9 y el conector del PC que indica la función de las nueve patillas de este conector muy popular.



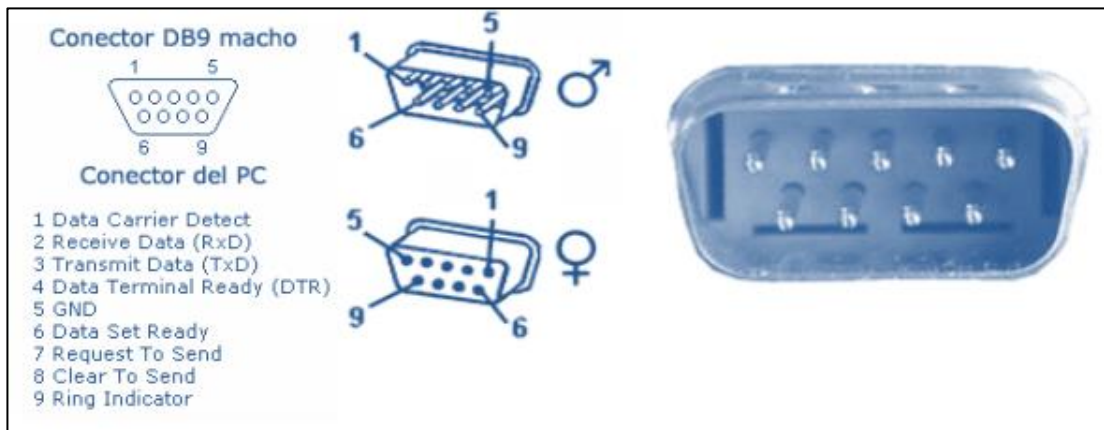


Figura 2.15 conector macho DB9 y el conector del PC  
 Fuente: <https://sites.google.com/site/comunicacionserie/>

#### 2.2.4. Modulo Inalámbrico XBEE

Un XBee es un componente autónomo modular, que utiliza la frecuencia de radio (RF) para intercambiar datos entre módulos XBee. Los módulos XBee transmiten a 2,4 GHz o de largo alcance a 900 MHz y tienen sus propios protocolos de red. Bell, Charles. (2013).

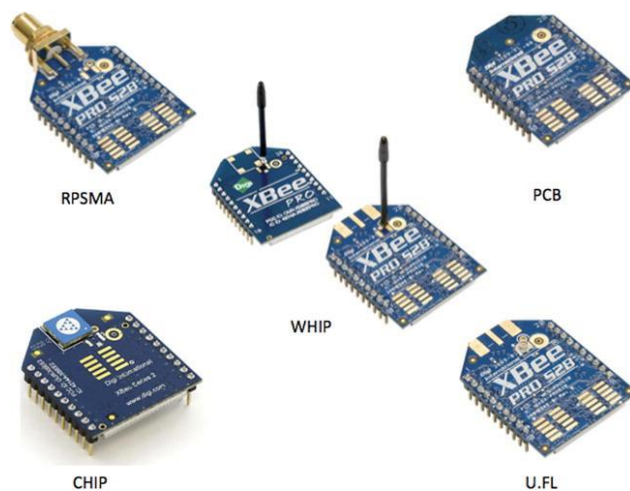


Figura 2.16 Modulo Inalámbrico XBEE  
 Fuente: [https://www.safaribooksonline.com/library/view/beginning-sensor-networks/9781430258247/9781430258247\\_Ch02.xhtml](https://www.safaribooksonline.com/library/view/beginning-sensor-networks/9781430258247/9781430258247_Ch02.xhtml)

El módulo XBee en sí es muy pequeña por lo que es fácil de incorporar en pequeños proyectos. Los módulos son también de baja potencia y pueden utilizar un modo especial de sueño para reducir aún más el consumo de energía.

Aunque el XBee no es un microcontrolador, pero si tiene una cantidad limitada de potencia de procesamiento que puede utilizar para controlar el módulo. (Bell, Charles. 2013).

Estos módulos utilizan el protocolo de red IEEE 802.15.4 de punto a multipunto o de redes peer-to-peer rápido. Están diseñados para aplicaciones de alto rendimiento que requieren baja latencia y sincronización de la comunicación predecibles. (Bell, Charles. 2013).

#### **2.2.4.1. ZigBee**

ZigBee un estándar abierto para la comunicación de red basado en el estándar IEEE 802.15.4. Los módulos RF XBee fueron diseñados para operar dentro del protocolo ZigBee y apoyar las necesidades únicas de bajo costo y bajo consumo de energía de redes inalámbricas de sensores. (Bell, Charles. 2013).

#### **2.2.4.2. Direcciones de los Módulos Xbee**

Los módulos XBee se identifican con un número de serie o dirección ubicada en la parte inferior del módulo. Sin embargo, puede encontrar la dirección utilizando la configuración de Digi aplicación o una aplicación de terminal serie simple. (Bell, Charles. 2013).

En la Figura 2.17 muestra la parte inferior de un módulo XBee.



Figura 2.17 Dirección del módulo inalámbrico XBEE

Fuente: [https://www.safaribooksonline.com/library/view/beginning-sensor-networks/9781430258247/9781430258247\\_Ch02.xhtml](https://www.safaribooksonline.com/library/view/beginning-sensor-networks/9781430258247/9781430258247_Ch02.xhtml)

Se puede observar los números impresos con el número de modelo. Utiliza estos números para formar una dirección de 64 bits única para cada módulo XBee. Esto también es conocido como la dirección de la radio. (Bell, Charles. 2013).

La dirección de la radio se utiliza para dirigir los mensajes para la entrega. En muchos aspectos, es similar a una dirección IP, pero en este caso es una dirección de radio específica. (Bell, Charles. 2013).

#### 2.2.4.3. Tipos de nodos ZigBee

Una red ZigBee la pueden formar, teóricamente, con hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad enorme de dispositivos.

Una red ZigBee la forman básicamente 3 tipos de elementos. Un único dispositivo Coordinador, dispositivos Routers y dispositivos finales (endpoints). (Daraviña, Giancarlo. Valencia, Richard. 2014).

Figura 2.18 muestra una red de malla típica de ZigBee

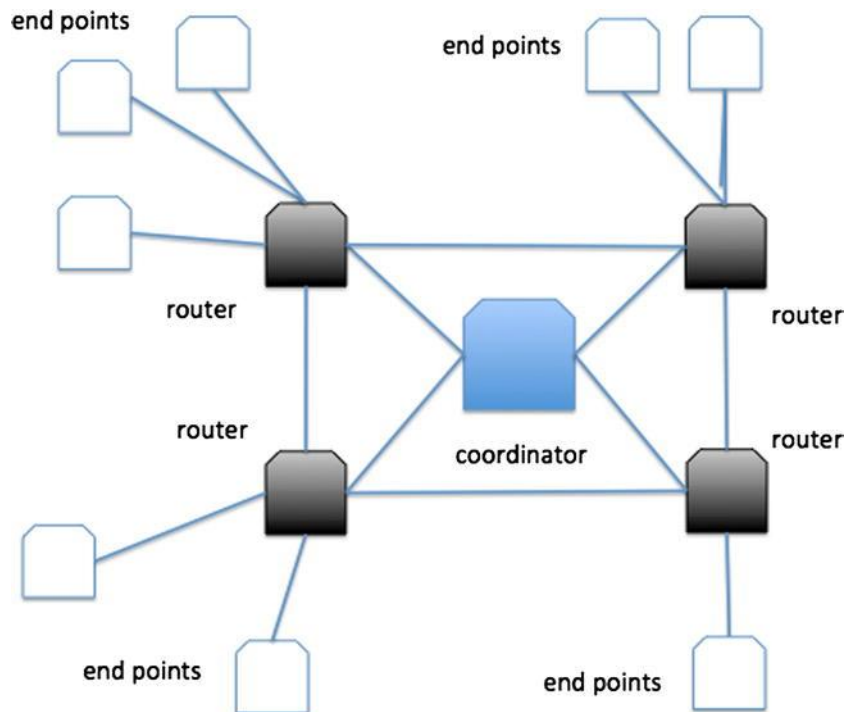


Figura 2.18 Red de malla ZigBee

Fuente: [https://www.safaribooksonline.com/library/view/beginning-sensor-networks/9781430258247/9781430258247\\_Ch02.xhtml](https://www.safaribooksonline.com/library/view/beginning-sensor-networks/9781430258247/9781430258247_Ch02.xhtml)

- **Coordinador:** El tipo de dispositivo más completo. debe de existir uno por red. sus funciones son de cargarse de controlar la res y los caminos que deben seguir los dispositivos para conectarse entre ellos. Carrera, Carlos. Guaiña, Luis. (2010).
- **Router:** Un nodo que está configurado como un router está diseñado para transmitir (ruta) información a otras radios. Los routers permiten la curación de las redes de malla uniendo las redes y el intercambio de mensajes de otros nodos. Los routers son generalmente alimentados con fuentes confiables, ya que debe ser confiable (Bell, Charles. 2013).

- Un Router puede participar en el enrutamiento de paquetes y está destinado a ser un nodo conectado a la corriente eléctrica. (Lara, María Belén. 2010)
  - Varios routers pueden funcionar en una PAN.
  - Puede enrutar paquetes de datos a/desde otros nodos.
  - Puede ser origen y destino de paquetes de datos.
- Dispositivo de fin: Un dispositivo final es un nodo que envía o recibe información a los nodos de router y coordinador. Tiene la ventaja de que menos procesamiento está pasando, por lo que el consumo de energía es menor. (Bell, Charles. 2013).
  - Varios dispositivos finales pueden operar en una PAN.
  - Puede ser origen y destino de paquetes de datos.
  - Todos los mensajes se transmiten a través de un Coordinador o Router.
  - Tienen bajo consumo.

#### **2.2.4.4. Configuración del módulo Xbee**

Para configuración módulo XBee, debe utilizar una máquina Windows.

La herramienta de Digi, X-CTU es una aplicación multiplataforma gratuito diseñado para permitir a los desarrolladores interactuar con módulos Digi RF a través de una interfaz gráfica fácil de usar. (Daraviña, Giancarlo. Valencia, Richard. 2014).

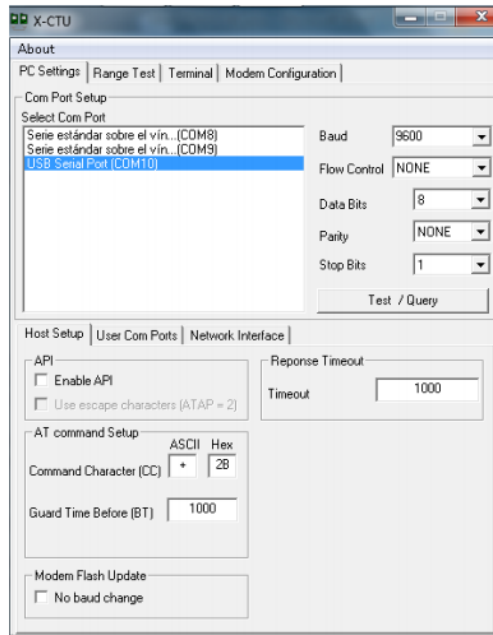


Figura 2.19 herramienta de Digi, X-CTU

Fuente: [https://www.safaribooksonline.com/library/view/beginning-sensor-networks/9781430258247/9781430258247\\_Ch02..xhtml](https://www.safaribooksonline.com/library/view/beginning-sensor-networks/9781430258247/9781430258247_Ch02..xhtml)

#### 2.2.4.5. Comparativa de los módulos inalámbricos Xbee

Existen 2 series de estos módulos: Serie 1 y la serie 2 sin embargo no son compatibles entre sí ya que utilizan distintos chipset y trabajan con protocolos diferentes. La serie 1 está basada en el chipset Freescale y son utilizados en redes punto a punto y punto a multipunto. Los módulos de la serie 2 están basados en el chipset de Ember y están diseñados para ser utilizados en aplicaciones que requieren repetidores o una red mesh. Ambos módulos pueden ser utilizados en los modos AT y API. (Duarte, Andres. 2014).

A continuación se observa una comparativa de 4 módulos inalámbricos serie 1 (Xbee 1 mW, Xbee Pro 60mW) y serie 2 (Xbee Series 2 1.25 mW ZigBee, Xbee Series 2 Pro 63mW ZigBee). Ver Tabla 2.8

Para el proyecto se usa el Xbee Pro 60mW serie 1

	Xbee 1mW 802.15.4 Chip Antenna Wire Antenna	Xbee PRO 60mW 802.15.4 Chip Antenna Wire Antenna	Xbee Series 2 1.25 mW ZigBee Wire Antenna	XBee Series 2 PRO 63 mW ZigBee Wire Antenna
<b>General</b>				
<b>Compatibility</b>	Series 1		Series 2	
<b>Digi part #</b>	XB24-ACI-001 XB24-AWI-001	XBP24-ACI-001 XBP24-AWI-001	XB24-Z7WIT - 004	XBP24-Z7WII - 004
<b>Parallax part #</b>	32404 32405	32406 32407	32408	32409
<b>Antenna</b>	Chip Wire	Chip Wire	Wire	Wire
<b>Performance</b>				
<b>Rd data Range</b>	250 kbps/up to 115 kbps	250 kbps/up to 115 kbps(1)	250 kbps/up to 1 115 kbps(1)	250 kbps/up to 1 115 kbps(1)
<b>Indoor Range</b>	100ft(30m)(2)	300ft(90m)(2)	133ft(40m)(2)	300ft(90m)(2)
<b>Outdoor Range</b>	300ft(100m)(2)	1 mile(1.6km)(2)	400ft(120m)(2)	2 mile(3.2km)(2)
<b>Transmit Power</b>	1mw(+0 dbm)	60 mw(+18 dbm)	1.25 mw(+1 dbm)	63mw(+18 dbm)
<b>Receiver sensitivity</b>	-92 dbm	-100 dbm	-98 dbm	-102 dbm
<b>Features</b>				
<b>Serial data Interface</b>	CMOS UART (no configuration needed)			
<b>Configuration method</b>	API or AT commands			
<b>Frequency band</b>	2.4 GHz			
<b>ADC input</b>	(7)10-bits ADC inputs			
<b>Digital I/O</b>	(8)Digital I/O		(10)Digital I/O	
<b>Networking And Security</b>				
<b>Topology</b>	Peer to peer, point to point, point to multipoint			
<b>encryption</b>	128 bits AES			
<b>Reliable Packet Delivery</b>	Retries/acknowledgments			
<b>Channels</b>	16 direct Sequence channels	12 direct Sequence channels	16 direct Sequence channels	12 direct Sequence channels
<b>ID</b>	PAN PID, 64 bits IEEE MAC			
<b>Power Requirements</b>				
<b>Supply voltage</b>	1.8 – 3.4 VDC (absolute)		2.1 - 3.6 VDC	2.7 - 3.6 VDC
<b>Transmit current</b>	45mA	215mA	35mA	205mA
<b>Receive Current</b>	50mA	55mA	38mA	48mA
<b>Power-down current</b>	<10 uA		<1uA	<3.5uA
<b>Regulatory Approvals</b>				
<b>FCC(USA)</b>	Our-Xbee	Our-XbeePro	Yes	
<b>IC(Canadá)</b>	4212A-Xbee	4212A-XbeePro	Yes	
<b>ETSI(Europe)</b>	Yes			
<b>C-TICK(Australia)</b>	Yes			
<b>Telec (Japan)</b>	Yes			

Tabla 2.8 Comparativa de los módulos inalámbricos Xbee

Fuente: <http://recursosbiblioteca.utp.edu.co/dspace/bitstream/11059/4326/1/6298D213.pdf> Pág. 23

Se puede observar en la Tabla 2.8 los módulos Pro se destacan por su gran alcance exterior.

#### 2.2.4.6 Comunicaciones serie

El módulo RF XBee® / XBee-PRO ® posee una interfaz física de comunicaciones serial asincrónica para conectarse a un dispositivo. A través de su puerto serial, el módulo puede comunicarse con cualquier lógica y voltaje compatible con la UART. (Lara, María Belén. 2010).

##### 2.2.4.6.1. Flujo de datos UART

Los dispositivos que tienen una interfaz UART se pueden conectar directamente a los terminales del módulo de RF como se muestra en la siguiente Figura 2.20. (Adafruit.2012).

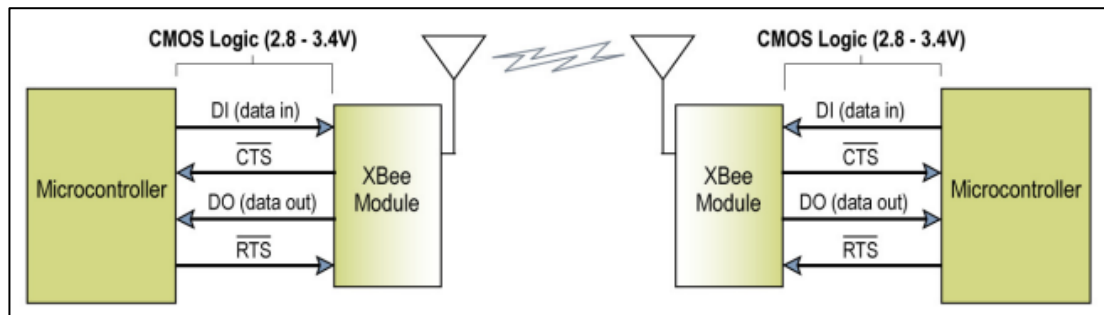


Figura 2.20 Diagrama de flujo de datos del sistema en un entorno de interfaz UART (Se distingue las señales Low con la línea horizontal sobre el nombre de la señal.)  
<http://www.adafruit.com/datasheets/XBee%20ZB%20User%20Manual.pdf> Pág. 26

##### 2.2.4.6.2. Datos seriales

Los datos ingresan al módulo UART a través del pin DI (pin 3) como una señal serial asíncrona. La señal debe estar en alto cuando no se están transmitiendo datos.



Cada byte de datos se compone de un bit de inicio (Low), 8 bits de datos (bit menos significativo primero) y un bit de parada (High).

La siguiente Figura 2.21 ilustra el patrón del bit serial de datos que pasan a través del módulo. (Adafruit.2012).

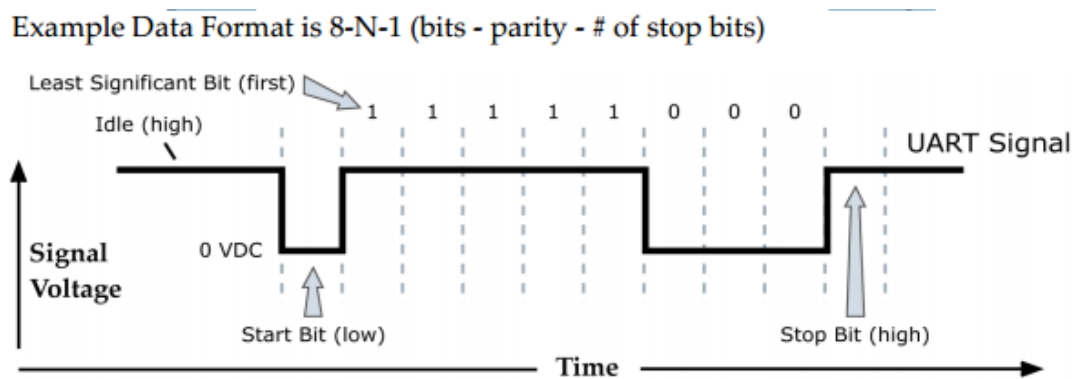


Figura 2.21 Paquetes de datos UART 0x1F (número decimal 31), transmitido a través del módulo de RF. Ejemplo de Formato de datos es 8-N-1 (bits - la paridad - # de bits de parada) <http://www.adafruit.com/datasheets/XBee%20ZB%20User%20Manual.pdf> Pág. 26

Consulte los Anexo para obtener más información.

### 2.2.5. LCD

LCD (Liquid Crystal Display) pantalla de cristal líquido, tienen la capacidad de mostrar cualquier carácter alfanumérico, permite representa la información que se genera de un microcontrolador de una forma fácil y económica. Son usados en la tecnología cotidiana en relojes, pantalla digicode, y así sucesivamente. Bayle, Julien. (2013).

Existen dos grandes familias de las pantallas LCD:

- LCD de caracteres se basa en una matriz de caracteres (columnas x filas)
- LCD gráfico, se basa en una matriz de píxeles

El proyecto de tesis incluye una LCD de caracteres

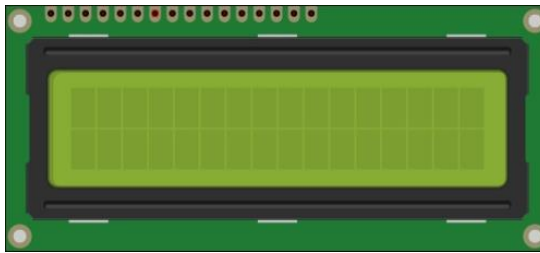


Figura 2.22 LCD de 16 x 2 caracteres

Fuente: [https://www.safaribooksonline.com/library/view/beginning-arduino/9781430232407/project\\_23\\_basic\\_lcd\\_control.html#project\\_23\\_basic\\_lcd\\_control](https://www.safaribooksonline.com/library/view/beginning-arduino/9781430232407/project_23_basic_lcd_control.html#project_23_basic_lcd_control)

Existen una gran cantidad de placas de circuito impreso que incluyen una pantalla LCD.

### 2.2.5.1. Descripción de los pines de un LCD alfanumérico

El número de pines de un display alfanumérico es normalmente de 14 o de 16 si el LCD es retro iluminado y son compatibles TTL. En la Tabla 2.9 se muestra el significado de las señales de cada pin. (Suarez, Jose. 2006).

Nº PIN	SÍMBOLO	DESCRIPCIÓN
1	$V_{SS}$	Alimentación negativa (masa)
2	$V_{DD}$	Alimentación positiva (+5Vdc)
3	$V_O$	Ajuste del contraste
4	RS	Selección de Registro (Register Select) RS = 1 $\Rightarrow$ Registro de Datos RS = 0 $\Rightarrow$ Registro de Instrucciones
5	$R/\bar{W}$	Lectura / Escritura (Read/Write) $R/\bar{W} = 1 \Rightarrow$ Lectura $R/\bar{W} = 0 \Rightarrow$ Escritura
6	E	Habilitación del display (Enable) E = 1 $\Rightarrow$ Habilitado E = 0 $\Rightarrow$ Deshabilitado
7-14	DB[0..7]	Pines del 0 al 7 del bus de datos bidireccional

Figura 2.9 Descripción de los pines de un LCD alfanumérico

Fuente: [http://eii.unex.es/profesores/jisuarez/descargas/ip/lcd\\_alfa.pdf](http://eii.unex.es/profesores/jisuarez/descargas/ip/lcd_alfa.pdf) Pág.4

Existen tres tipos de señales en el LCD: de alimentación correspondientes a los pines 1, 2, y 3. El de control que son los pines 4, 5 y 6. Las Señales de datos del pin 7 al 14 forman un bus de datos bidireccional de 8 bits 0 de 4 bits por donde son escritos los datos e instrucciones y leer el estado del display. (Suarez, José. 2006).

### 2.2.6. Diodo emisor de luz

La tecnología conocida como LED por sus siglas en inglés (**light emitting diode**), es un componente optoelectrónico pasivo y, más concretamente, un diodo que emite luz. En otra palabra es un diodo o dispositivo que permite que la electricidad pase a través de ella en una sola dirección, que aporta luz. La luz puede variar de infrarrojos (no visible) a través del espectro de color, dependiendo del material semiconductor que se utiliza.

Por ejemplo, un LED rojo podría utilizar aluminio fosfuro de indio galio (AlInGaP) y una caída de tensión en particular (de un lado del diodo a la otra) para emitir el color deseado. (Sal Cangeloso, 2012).

#### 2.2.6.1. Clasificaciones y aplicaciones

Existen muchas variedades de formas, tamaños y colores, como se muestra en la Figura 2.23. (Don Wilcher. 2014.)

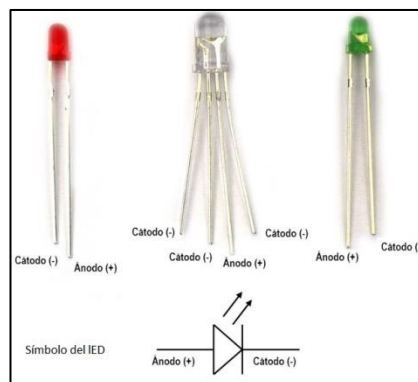
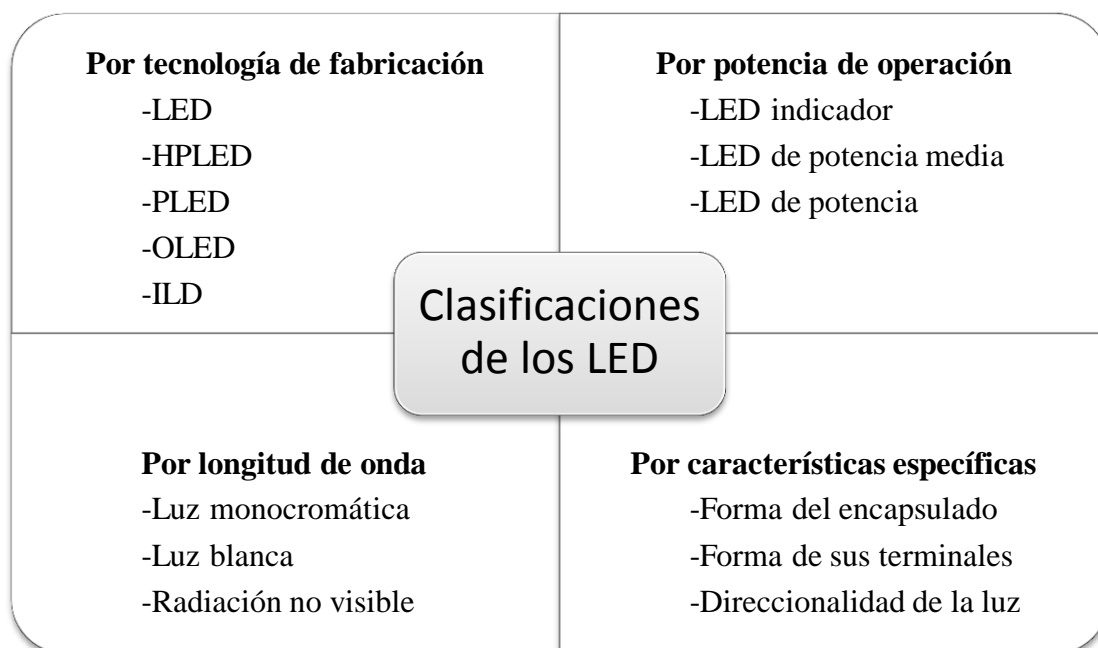


Figura 2.23 LEDs

Fuente: <https://www.safaribooksonline.com/library/view/make-basic-arduino/9781449360658/ch04.html>

Dependiendo de la aplicación, los LED se pueden clasificar de acuerdo a la Figura 2.10, la cual se describe a continuación:



*Tabla 2.10 Clasificación de los LED según su aplicación.  
Fuente: [http://jupiter.utm.mx/~tesis\\_dig/10877.pdf](http://jupiter.utm.mx/~tesis_dig/10877.pdf). Pág. 17*

La Tabla 2.4 muestra alguna de las principales aplicaciones de los LEDs

SECTOR	APLICACIÓN
Iluminación decorativa	<ul style="list-style-type: none"> <li>– Luces para escenarios y teatros.</li> <li>– Luces para bares, discotecas, etc.</li> <li>– Luces navideñas y para otras celebraciones.</li> <li>– Anuncios publicitarios luminosos.</li> </ul>
Iluminación arquitectónica	<ul style="list-style-type: none"> <li>– Iluminación de interiores y exteriores.</li> <li>– Iluminación de museos.</li> <li>– Fachadas de edificios.</li> </ul>
Dispositivos electrónicos	<ul style="list-style-type: none"> <li>– Fototerapia, fotoquimioterapia, esterilización (radiación UV).</li> <li>– Medición de glucosa y oxígeno en la sangre (rojo e IR).</li> <li>– Curación dental (azul).</li> <li>– Iluminación para endoscopia, cirugías y revisiones dentales (luz blanca).</li> </ul>
Medicina	<ul style="list-style-type: none"> <li>– Flash para cámaras digitales.</li> <li>– Luz de fondo para pantallas LCD y plasma.</li> </ul>

	<ul style="list-style-type: none"> <li>– Iluminación de teclados, para dispositivos móviles.</li> <li>– Proyectores.</li> </ul>
Otros aplicaciones	<ul style="list-style-type: none"> <li>– Vehículos: Luces de lectura, faros, iluminación de paneles indicadores.</li> <li>– Visión artificial (machine vision).</li> <li>– Alumbrado público: semáforos, calles, parques, jardines y carreteras.</li> </ul>

*Tabla 2.11 Aplicaciones de los LED.*  
Fuente: [http://jupiter.utm.mx/~tesis\\_dig/10877.pdf](http://jupiter.utm.mx/~tesis_dig/10877.pdf). Pág. 24

### 2.2.7. Regulador LM7805

Es un regulador de voltaje que entrega 5V constante y una corriente máxima de 1 A, pertenece a la serie LM78XX. Son utilizados para energizar a dispositivos o componentes electrónicos que requieran de esta característica del regulador. Para mayor detalle referirse al anexo LM78XX. El voltaje de entrada ( $V_{in}$ ) puede ser de hasta 12 volts. González, Elfrich. (2009).

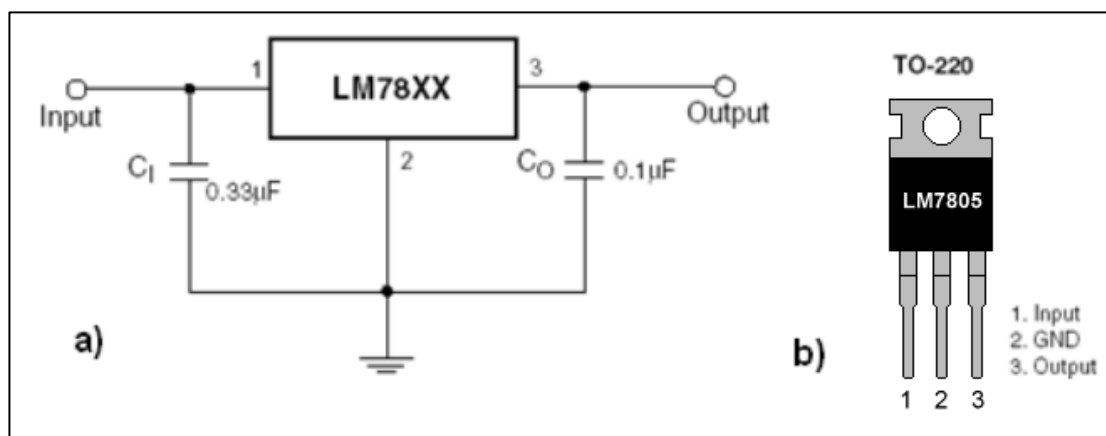


Figura 2.24 a) Conexión típica para la serie LM78XX, b) Distribución de pines del circuito integrado Lm7885

Fuente <http://bibdigital.epn.edu.ec/handle/15000/1588> Pág.89

### 2.2.8. Integrado 74ls04 (NOT)

La compuerta lógica NOT invierte la señal que entra, la simbología de esta compuerta y su tabla de verdad se muestra y el diagrama del integrado 74LS04 contiene un conjunto de 6 compuertas lógica NOT en la figura 1.25 se presenta.

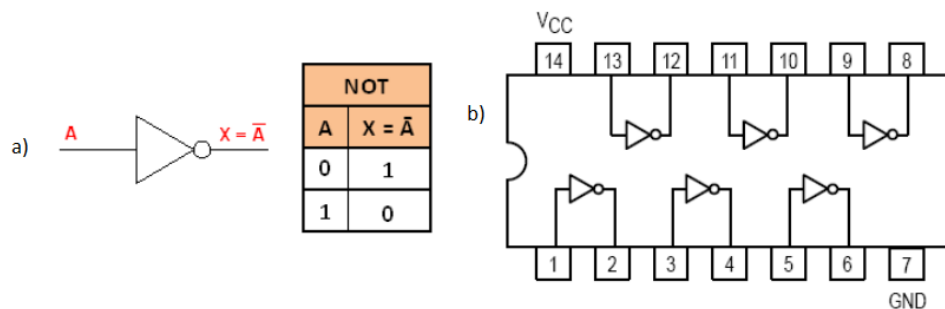


Figura 2.25 a) simbología y tabla de verdad de la compuerta lógica b) diagrama de conexión interna del integrado 74ls08

Fuente <http://bibdigital.epn.edu.ec/handle/15000/1588> Pág.88

El voltaje de alimentación típico para este integrado es de 5V. El voltaje de entrada para que se considere en alto debe ser mayor o igual a 2V y menor o igual al voltaje de alimentación, con esta consideración la compuerta entregara a la salida un valor de 0V. En cambio, si el voltaje de entrada es menor a 2V entonces la compuerta tomara el voltaje de entrada como en bajo y su salida entregara 5V. González, Elfrich. (2009).

La familia 74xx ha evolucionado a través de muchas generaciones, y la letra (s) que se inserta después de la "74" dirá que la generación que está tratando. Algunas generaciones han incluido:

- 74L ,74LS, 74C, 74HC,74AHC

Y más. En términos generales, las generaciones posteriores tienden a ser más rápido o más versátil que las generaciones anteriores. (Platt, Charles. 2009).

## **2.2.9. Almacenamientos de Granos**

### **2.2.9.1. Historia de almacenamiento**

Históricamente, el concepto de almacenamiento va tan lejos como los antiguos egipcios que construyeron silos para proteger granos fuera de la humedad y almacenarlos para su uso cuando hubo un golpe de hambruna. El conocido ejemplo de ello es la historia de José, cuando supervisó el racionamiento y el almacenamiento de granos, así como los registros guardados de las cantidades almacenadas. Esto ha evolucionado a lo largo de las generaciones y las naciones. Khalil, Mostafa. (2012).

### **2.2.9.2. Importancia del almacenaje de los granos**

- Como un amortiguador frente a la oferta irregular o estacional - los productos agrícolas han sido almacenados en silos durante miles de años.
- Como una reserva contra la demanda irregular, el caso de la mayoría de las materias primas energéticas.
- Como una cobertura contra cualquier otro suministro o interrupción logística, que de otro modo requeriría la pausa de un proceso industrial. Helyette, Geman (2015).

### **2.2.9.3. Sistemas de almacenamiento**

Los sistemas de almacenamiento se clasifica según la atmósfera del lugar donde se guardan los granos en:

- I) **Atmósfera normal:** Es un almacenamiento en el cual el aire que rodea a los granos prácticamente tiene la misma composición que el aire atmosférico. Es el tipo de almacenamiento más conocido y dentro de éste, los sistemas más comunes son: Silos malla de alambre, Celdas, Galpones, etc.
- II) **Atmósfera artificial:** Es un sistema de almacenamiento, en el cual se procura o se crea un microclima en el interior del lugar donde se almacenan los granos, con el fin de restringir la disponibilidad del oxígeno del aire y así poder disminuir los procesos de respiración de los hongos e insectos o si existe calentamiento lo enfría y lo uniformarlo la temperatura.

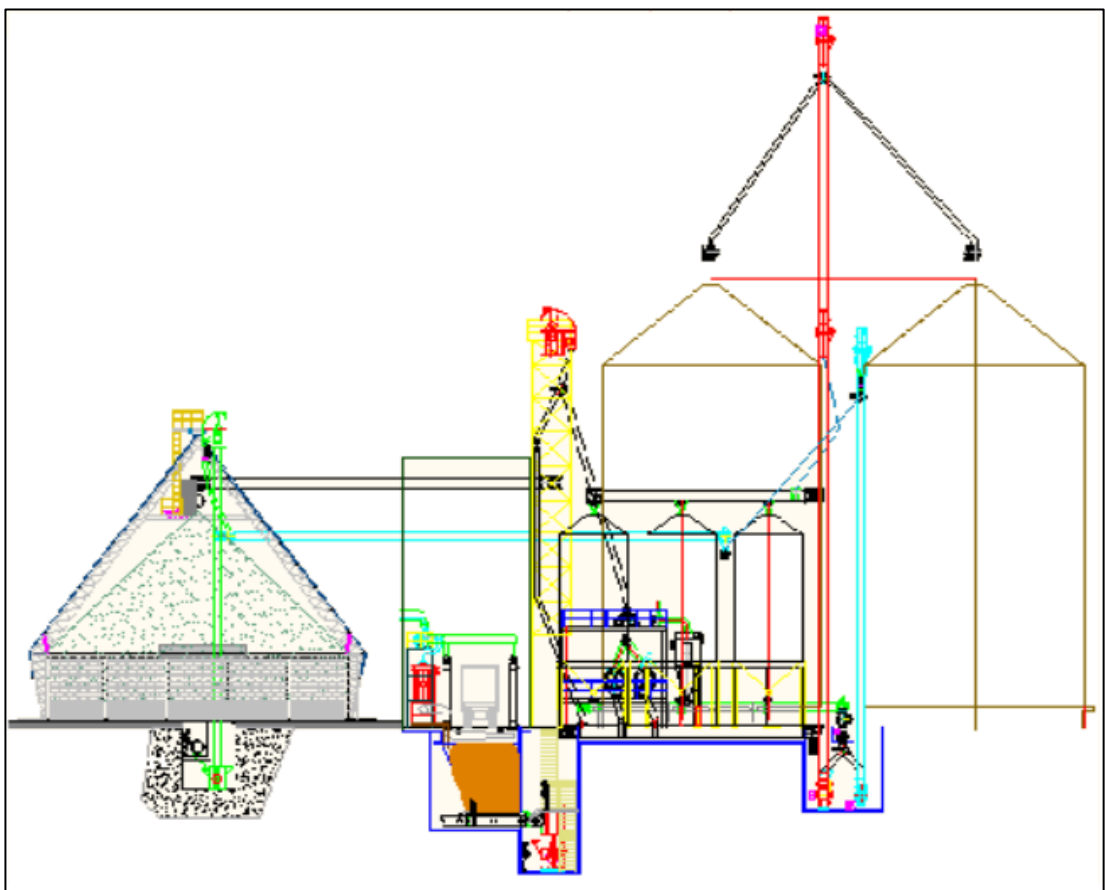


Figura 2.26 Esquema de los silos planos y cónicos a la derecha y la bodega plana a la izquierda, y en la mitad el sistema de volteo.

Fuente: <http://www.guayas.gob.ec/dmdocuments/medio-ambiente/EIA%20SILOS%20Y%20GALPON%20EXPALSA.pdf> Pág. RE-6



Los silos se utilizan, en una amplia gama de la industria, para almacenar sólidos en cantidades muy variables. Pueden ser de acero, de hormigón armado o de otros materiales y su descarga puede, así mismo, ser por gravedad o utilizando procedimientos mecánicos. (Ccarita, Fredy. 2011),

#### **2.2.9.4. Capacidades de almacenamiento de silo**

De acuerdo con el sistema B.M.H.B. (Código inglés para el diseño de silos y tolvas), en las clases o categorías siguientes:

Clase 1 Silos pequeños cuya capacidad es menor de 100 toneladas

Clase 2 Silos de capacidad intermedia (de 100 t a 1000 t)

Clase 3 Silos grandes (de capacidades superiores a 1000 t).

#### **2.2.9.5. Clasificación por tamaño y geometría**

Según el tamaño, la geometría, el patrón de flujo de descarga, del material almacenado. (Ccarita, Fredy. 2011),

El tamaño y geometría dependen de los requerimientos funcionales tales como el volumen de almacenamiento, el sistema y forma de descarga, las propiedades del material almacenado, el espacio disponible, consideraciones de tipo económico, etc.. (Ccarita, Fredy. 2011),

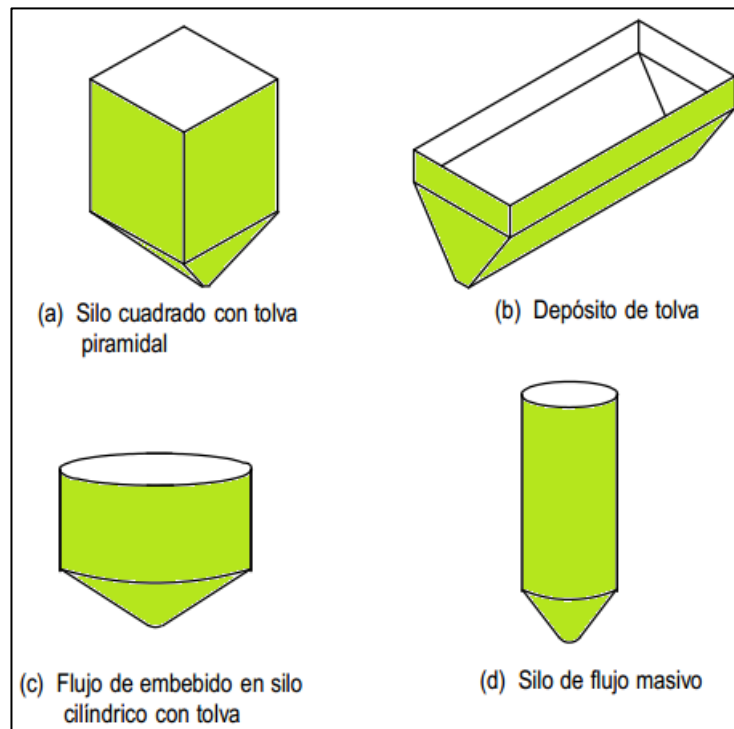


Figura 2.27 Típicas geometrías de contenedores

Fuente: <file:///C:/Users/veivalmo/Downloads/trabajodesilo-131105155234-phpapp02.pdf> Pág. 5

### 2.2.9.6. Los aspectos importantes del almacenamiento

El principal objetivo es controlar la temperatura del granel: enfriarlo y uniformarlo.

- A medida que aumenta la humedad de los granos por encima de la humedad que entraron durante el almacenamiento, aumenta el deterioro, principalmente causado por el desarrollo de hongos, levaduras y bacterias. Estos microorganismos necesitan de humedad para crecer y a medida que se van desarrollando, al aumentar el nivel de respiración y aumentan la temperatura de la masa de los granos. (Casini, Cristiano. Rodríguez, Juan. 2003)
- Hacer un control estricto de los insectos porque perjudican en gran proporción a los granos. En este caso, también hay aumento de calor por la

respiración de los insectos, que calienta la masa de los granos. (Casini, Cristiano. Rodríguez, Juan. 2003)

Utilizar la termometría para detectar posibles aumentos de temperatura en el granel, y controlar con aireación.

Durante el almacenamiento deben Ingresar granos limpios y secos para evitar la acumulación de material fino en el centro del granel (dificulta el pasaje de aire). En algunos casos conviene colocar desparramadores de granos (evita la acumulación de material fino en el centro del granel). (PRECOP. 2007).

Durante la aireación hay que tener en cuenta la humedad relativa (H.R.), debe ser inferior a 70%, o de lo contrario cuando se cuente con 5°C o más de diferencia de temperatura entre el aire y el grano (aire más frío que el grano), independientemente de la H.R. del aire, dado que el deterioro de los granos se manifiesta, en primer orden, por la pérdida del poder germinativo (P.G.), luego por la disminución del peso hectolitrito (P.H.) o viceversa. (PRECOP. 2007).

#### **2.2.9.7. Periodos de almacenajes seguros y condiciones**

En la figura nos muestra una relación de humedad del maíz con los días que puede estar el maíz en buenas condiciones y en la figura nos muestra las condiciones favorables para que pueda existir insectos durante el almacenamiento. Ediagro, (diciembre, 2006).

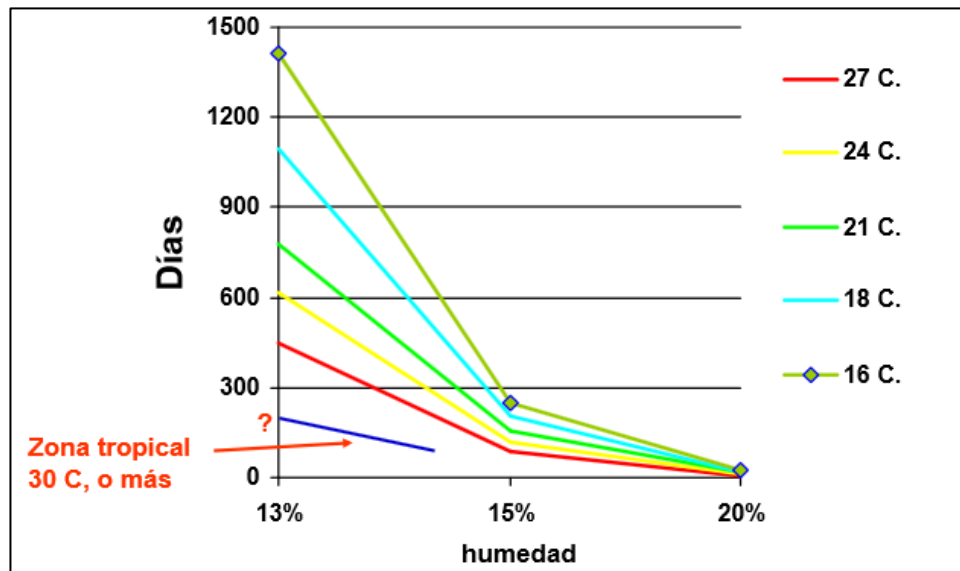


Figura 2.28 Periodo de almacenamiento seguro maíz  
 Fuente <http://www.conarroz.com/pdf/6Conservaciondegranos.pdf>. Pág.17

En la Figura 2.28 se interpreta de esta manera: el maíz de 13% de humedad, manteniéndose a una temperatura constante de 27 °C se puede conservar aproximadamente más de 400 días.

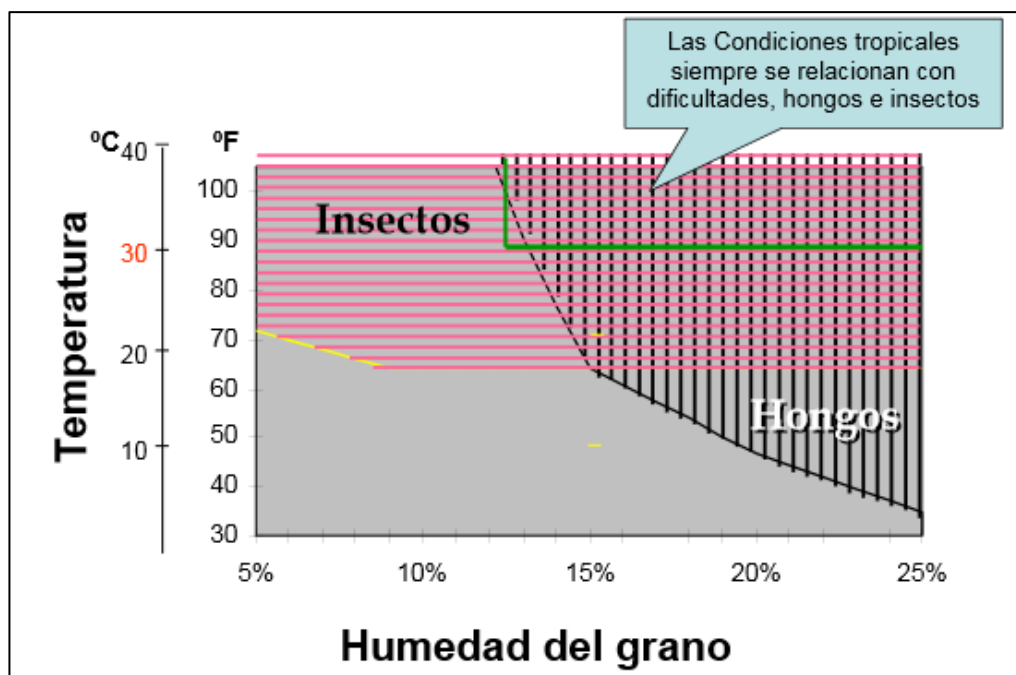


Figura 2.29 Condiciones favorables para hongos e insectos  
 Fuente: <http://www.conarroz.com/pdf/6Conservaciondegranos.pdf>. Pág.18

En la Figura 2.29 muestra los parámetros desde y hasta qué punto pueden producirse hongos e insectos

El maíz, la soja y el sorgo, que se conocen generalmente como "granos para alimentación animal", debido a su uso principal de los animales; el trigo y el arroz, que se conoce como "granos de alimentos", ya que están dirigidos principalmente al consumo humano. (Helyette Geman. 2015).

## **CAPITULO III.**

### **DISEÑO DE HARDWARE**

#### **3.1 INTRODUCCIÓN**

El diseño del hardware es tan indispensable como el software en esta implementación, debido a que el software de simulación de diseño electrónico ayudara a saber con certeza si se puede fabricar dicho proyecto. Se menciona proyecto y no solo la tarjeta, porque el sistema es completo y complejo, donde se necesitara enlazar la electrónica, los microcontroladores y la programación en lenguaje de computadora es en Java.

Es de importancia saber los componentes a conseguir localmente y los que son de importación para generar un desarrollo fácil y que el producto pueda tener reposición en el futuro. Con esto se garantizara la vida útil del proyecto.

La estructura que se utiliza es una tarjeta principal receptora que estará alado de un PC y una tarjeta secundaria que recogerá la información de los sensores. La PC maneja un software desarrollado en Java y desde este equipo se solicitara la recolección de datos. La comunicación entre la pc y la tarjeta principal se hará mediante el Xbee Xplorer Serial.

La tarjeta principal maneja un Xbee pro de 60 mW y la tarjeta secundaria usara un modelo similar para poder comunicarse hasta en 1 kilómetro.

La tarjeta secundaria se compone de un microcontrolador marca PIC y este recopila los datos del cable conectado a una de sus entradas. Mostrará las temperaturas de los

diferentes sensores DS18B20 que se ubican en un solo cable mediante protocolo de comunicación y comunicación 1-wire.

### 3.1.1 Diseño de la tarjeta en PROTEUS ISIS Professional 7.2V

El diseño se hará con el software Proteus Isis Profesional para probar el circuito antes de realizar la placa.

#### 3.1.1.1 Alimentación

Primero se diseña el circuito de polarización con la alimentación de 110VAC y un transformador que lo convertirá a 12 VDC para usar en el circuito. También se implementa un circuito rectificador para que la corriente pase de alterna a directa.

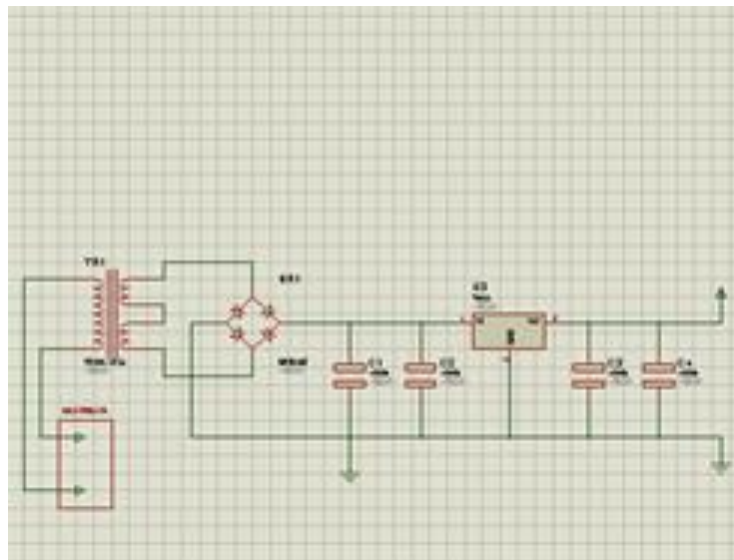
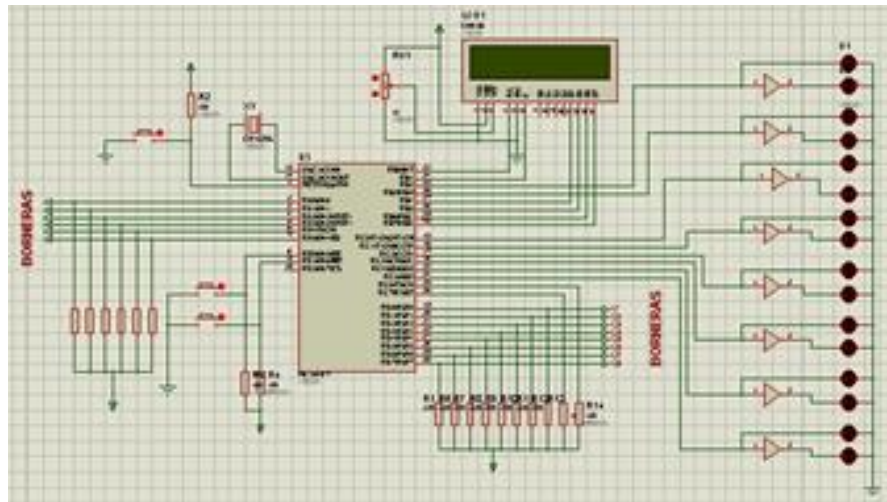


Figura 3.1 Circuito de polarización de prototipo de lectura de temperaturas  
Autor: Omar Tello y Alfredo Villares

Se ha utilizado un microcontrolador modelo 16F877A de la marca PIC en donde se alimentara por el circuito antes mencionado



*Figura 3.2 Circuito de prototipo de lectura de temperaturas  
 Autor: Omar Tello y Alfredo Villares*

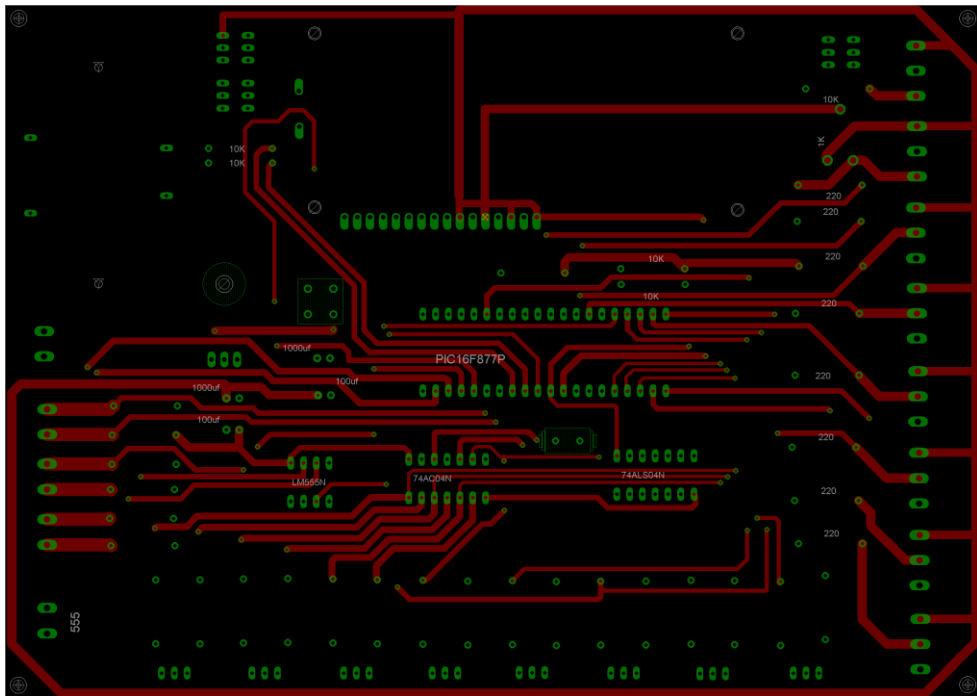
Se ha seleccionado un circuito basado en un microcontrolador marca PIC y se añadirá según el software. Esta herramienta permitirá simular el proceso, dando la posibilidad de cambiar piezas. Luego de varias pruebas y simulaciones se podrá determinar que el circuito es óptimo para su desarrollo físico.

### **3.1.1.2 Ruteado**

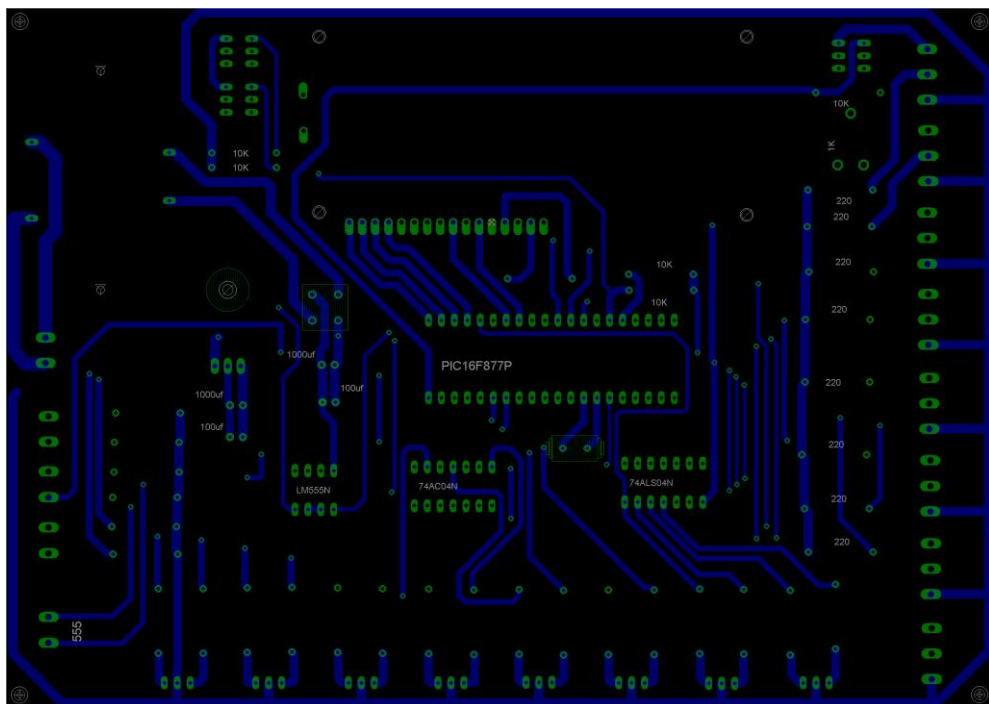
Cuando se tiene el diseño electrónico, se define el tamaño de la placa a ser fabricada. El software automáticamente ubica los elementos del circuito para mostrar un preliminar del diseño a ser impreso. El diseñador debe tener criterio cuidado bordes, orificios para usar tornillos, funcionalidad, ergonomía de trabajo y estética.

Se obtiene como resultado dos imágenes que serán para la placa en dos niveles o capas. Esto hará que la placa sea más eficiente y pequeña. Se usaran dos caras y se ahorrara espacio en la impresión.





*Figura 3.3 Ruteado frontal de prototipo de lectura de temperaturas  
Autor: Omar Tello y Alfredo Villares*



*Figura 3.4 Ruteado trasero de prototipo de lectura de temperaturas  
Autor: Omar Tello y Alfredo Villares*

### 3.1.2. Implementación de la tarjeta

Las placas han sido desarrolladas a lo largo de la carrera de Ingeniería Electrónica como parte fundamental de la implementación de proyectos. Durante este proceso, los profesores indican cómo realizar placas artesanalmente mediante el uso de placas, marcadores y ácidos. En la placa se raya la pista con marcador y luego se limpia el excedente con ácido.

Para elaborar placas artesanales de una manera más estética se imprime en papel el circuito y luego de proceder a plancharlo en la placa. Se procede a realizar pasos similares a los detallados anteriormente donde se usa marcador y ácido para limpiar la placa.

El prototipo que se elaborara tendrá como base la expansión a mayor número de cables, pero en esta ocasión se usara un solo cable para demostrar el principio de la transmisión inalámbrica de datos de temperatura en el interior del silo.

La fabricación de la placa se realizó en una empresa con maquinaria especializada en impresión de placas electrónicas para obtener el mejor resultado.

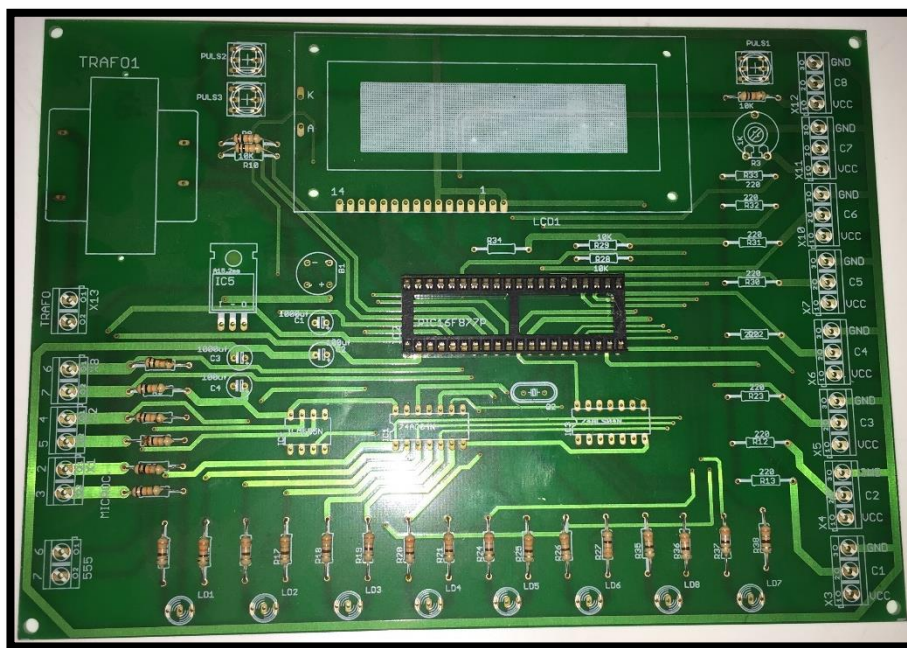
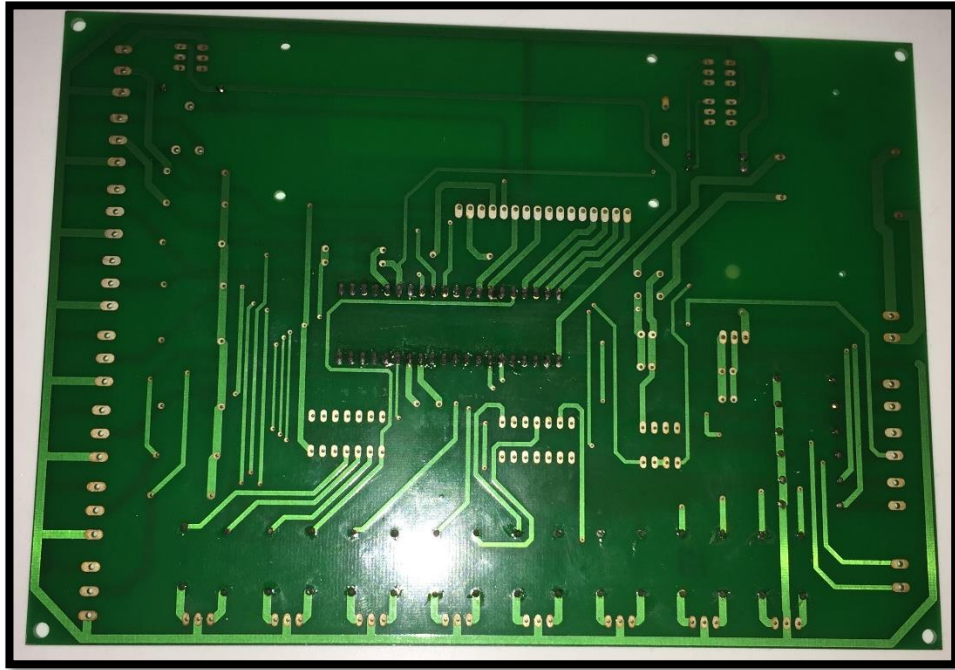


Figura 3.5 Ruteado impreso frontal de prototipo de lectura de temperaturas  
Autor: Omar Tello y Alfredo Villares



*Figura 3.6 Ruteado impreso trasero de prototipo de lectura de temperaturas  
Autor: Omar Tello y Alfredo Villares*

Luego de imprimir la tarjeta se procede a ensamblar los elementos electrónicos que se detalla en el esquemático.

Luego se sueldan las piezas como los integrados, condensadores, LCD, PIC, bornes, entre otros.

Se usa la técnica con soldadura de estaño, cautín, pasta de soldadura electrónica.

Se usan herramientas varias para ayudar a fijar la posición correcta y poder cortar los excedentes.

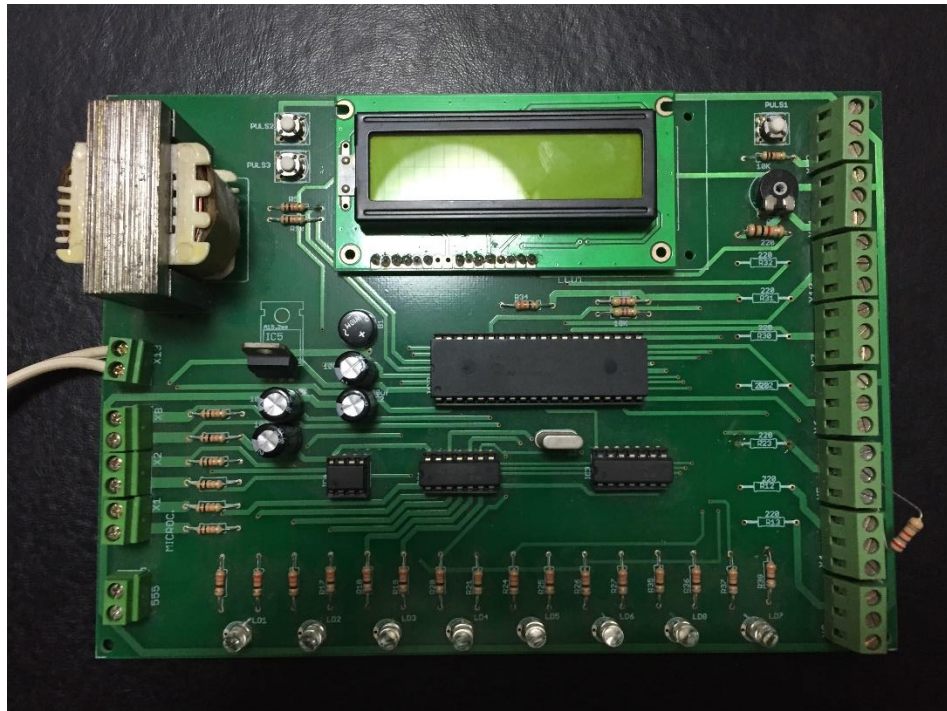


Figura 3.7 Ruteado impreso frontal armado de prototipo de lectura de temperaturas  
Autor: Omar Tello y Alfredo Villares

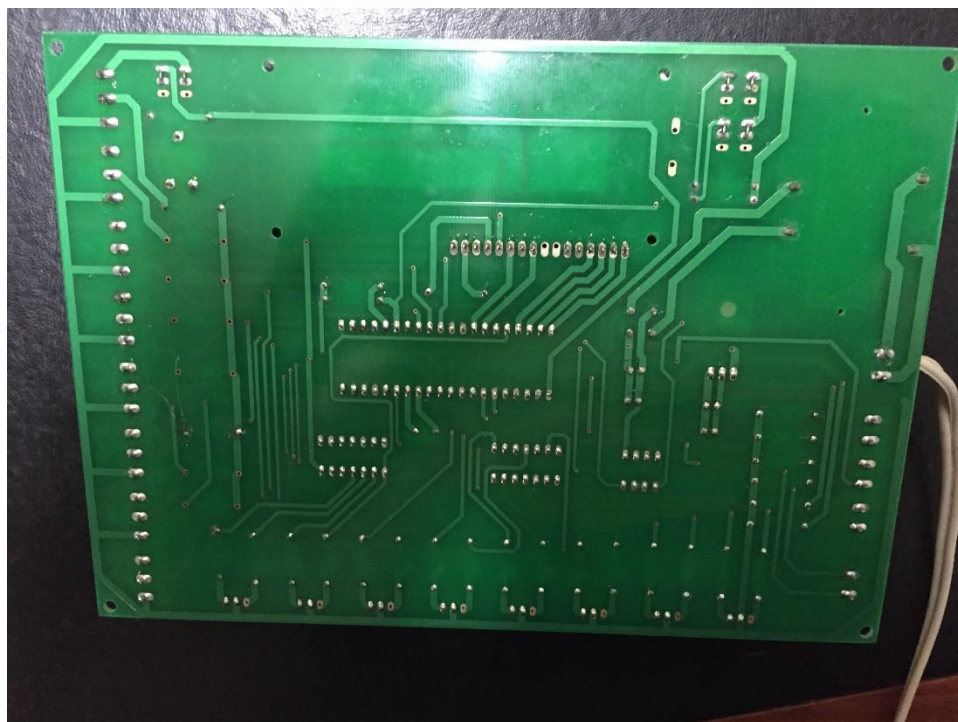
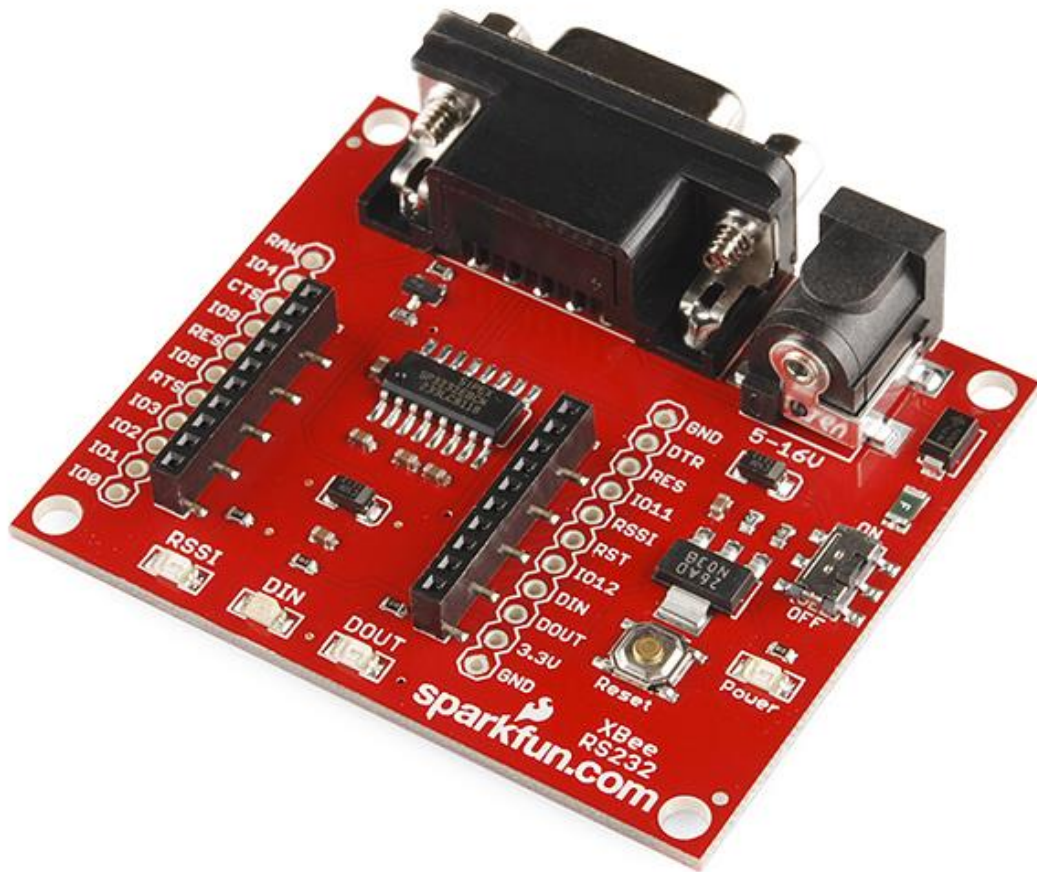


Figura 3.8 Ruteado impreso trasero armado de prototipo de lectura de temperaturas  
Autor: Omar Tello y Alfredo Villares

### 3.1.3. Programación para comunicación usb/serial

La comunicación USB se hará por el XBEE XPLOERER. Este se maneja por plug and play y será comunicado por un convertidor usb serial.



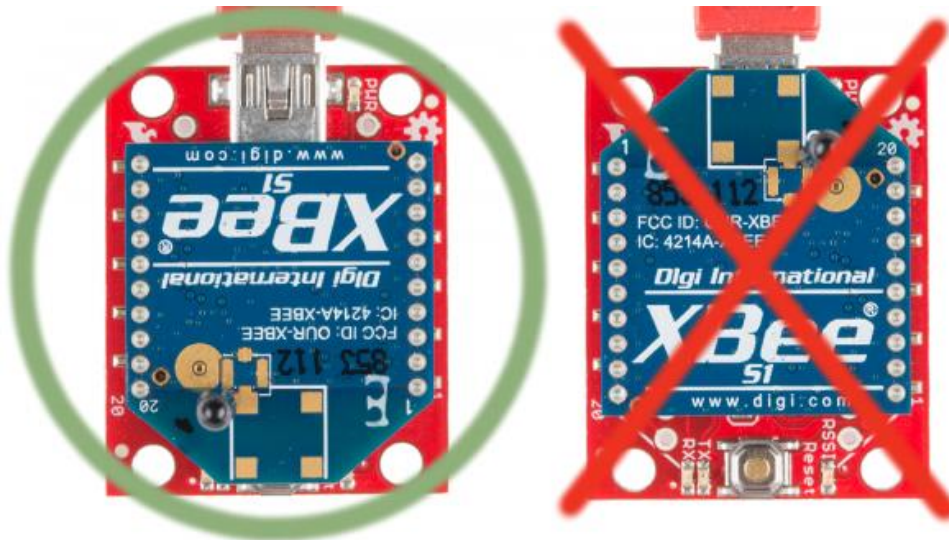
*Figura 3.9 Tarjeta XBee Xplorer Serial*  
*Fuente: <https://learn.sparkfun.com/tutorials/exploring-xbees-and-xctu>*

El convertidor de USB a serial puede ser genérico o de alguna marca recomendada. En este caso se citara la marca Manhattan, cuyo distribuidor es Juan Marcett con locales en la ciudad de Guayaquil



*Figura 3.10 Cable de conversión USB a Serial*  
*Autor: <https://learn.sparkfun.com/tutorials/exploring-xbees-and-xctu>*

La correcta posición del hardware es la siguiente:



*Figura 3.11 Modulo de conexión inalámbrica XBee*  
*Autor: <https://learn.sparkfun.com/tutorials/exploring-xbees-and-xctu>*

Hay que insertar según las huellas coincidiendo la parte angular con las líneas punteadas. Se debe tener cuidado al insertar los pines del Xbee en la tarjeta debido a que se puede doblar alguno.

### 3.1.4. Diseño del cable

Para el prototipo es necesario calcular altura del elemento donde se va a utilizar. Como ejemplo de la tesis se usara un silo de 20 metros de altura.

Es importante recordar que:

Según los análisis económicos del prototipo se determinó que una de las razones del sistema es que era el más benéfico en relación número de lecturas y costo.

En comparativa, 10 sensores del sistema de este proyecto cuestan \$40 y 10 PT100 cuestan \$600 según valores locales en Ecuador.

La topología a ser usada es la tipo bus 1-wire



Figura 3.12 Topología bus 1-wire

Autor: <http://www.loxone.com/es/servicio/documentacion/extensiones/1-wire.html>

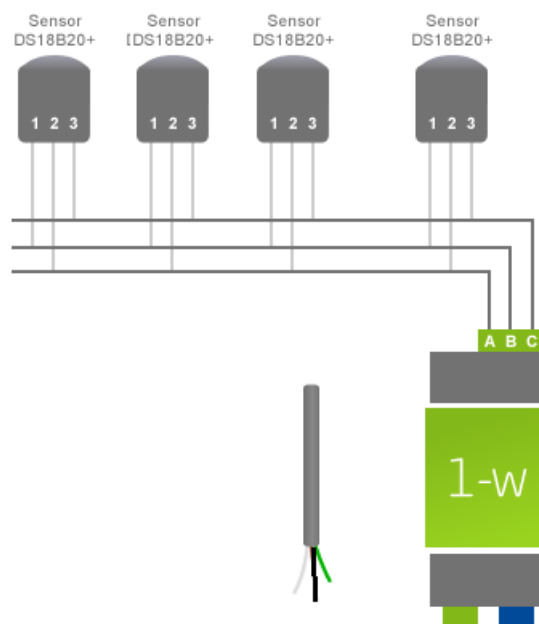
### 3.1.5. Topología Bus

La topología bus sin ramales es la mejor solución de cableado para sensores. Permite un máximo de 350m con 20 sensores a lo largo del cable.

El tipo de conexión que se usara se detallara a continuación para realizar según datasheet y polarización.

(A) DQ	(2) DQ	(verde) DQ
(B) GND	(1) GND	(blanco) GND
(C) VDD	(3) VDD	(negro) VDD

*Tabla 3.1 Conexión de sensores en cable  
Autor: Omar Tello y Alfredo Villares*



*Figura 3.13 Topología bus 1-wire con diagrama de conexión para sensores DS18B20  
Autor: Omar Tello y Alfredo Villares*

En la fabricación del cable se abre un cable concéntrico de 3 hilos y se conecta las 3 patas del sensor en los respectivos colores. Todos los sensores serán conectados de la misma manera y serán marcados con un número para reconocimiento visual.



Luego de hacer la conexión, se procede a usar cinta aislante para evitar contactos entre las patas.

Por último, se sella usando silicón para lograr un mejor acabado y evitar contacto con el exterior.



*Figura 3.14 Implementación artesanal de sensor en cable de medición  
Autor: Omar Tello y Alfredo Villares*

Se puede apreciar el cable terminado en la figura 3.15



*Figura 3.15 Cable con 10 sensores DS18B20  
Autor: Omar Tello y Alfredo Villares*

Según datos de la empresa EDIAGROS de Brasil lo más recomendable para un silo de 9 metros de diámetro de usar 3 juegos de cables para cubrir el área de lectura de temperaturas.

En la figura 3.16 se ilustra de como iría los cables con los sensores DS18B20 para obtener una adecuada medición en los resultados que presenta el sistema

## 30' (9.1m) Silo

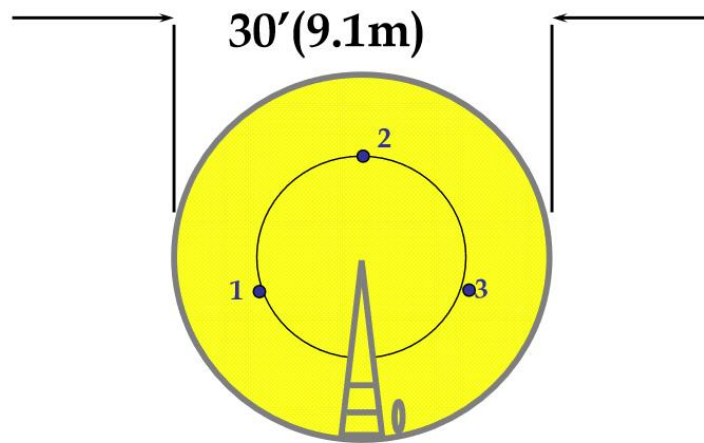


Figura 3.16 Numero de cables recomendado en silo de 9.1m diámetro  
Autor: <http://www.conarroz.com/pdf/6Conservaciondegranos.pdf>

Para un silo de 18 metros es recomendable el uso de 11 cables según la distribución de la siguiente figura.

## 60' (18.2m) Silo

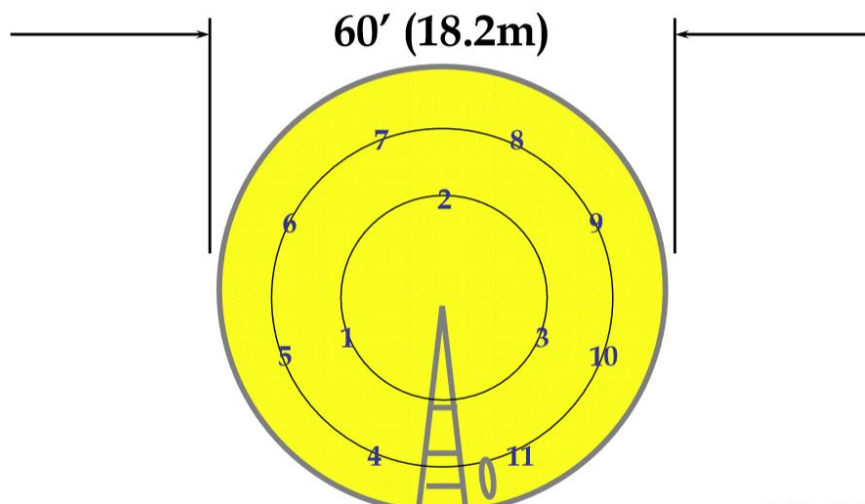


Figura 3.17 Numero de cables recomendado en silo de 18.2m diámetro  
Autor: <http://www.conarroz.com/pdf/6Conservaciondegranos.pdf>

Según los cálculos, el sensor tiene una distancia de cobertura de 3.65 metros a su alrededor.

12' (3.65m) Radio/2-3 Días

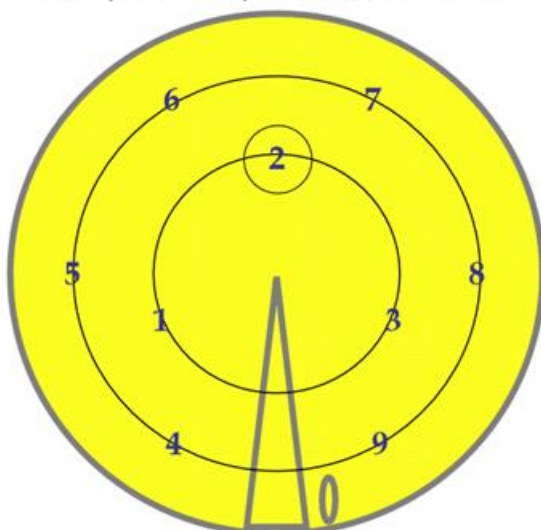


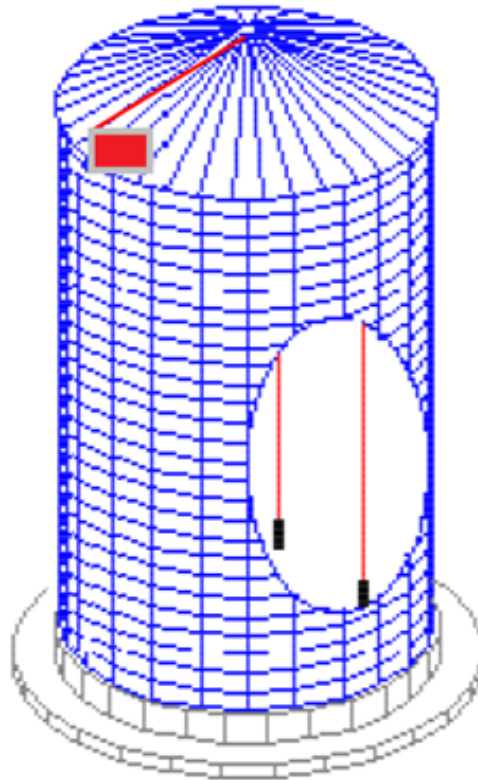
Figura 3.18 Área de sensibilidad de sistema de lectura de temperatura  
Autor: <http://www.conarroz.com/pdf/6Conservaciondegranos.pdf>

La ubicación de la caja de lectura será en la parte superior del silo, desde donde se transmitirá la información de la termometría.

Se usara esta posición para que el diseño de los cables de lectura no tenga mucho excedente de cables, que repercute en ahorro y eficiencia de uso de materiales.

El uso de la comunicación inalámbrica es exactamente por la misma razón de eficiencia en uso de recursos. Debido a la altura del silo, se evitara el uso de cables apantallados de larga distancia para la comunicación.

La distancia de sensores en el cable se ha decidido ser separado en 2 metros.



*Figura 3.19 Ubicación de cables y caja de lectura  
Autor: Omar Tello y Alfredo Villares*

En el momento de la instalación de los cables, estos deberán tener argollas superiores para anclarse al techo y argollas inferiores para anclaje inferior.

Luego de las argollas superiores los cables deben unirse en la salida superior del silo por un ducto de ventilación o un ducto diseñado para la salida de cables de lectura de temperatura.

La instalación de cables luego de pasar por el ducto, deberá tener tuberías rígidas debidamente instalada por profesionales para evitar problemas meteorológicos o ataques de plagas.

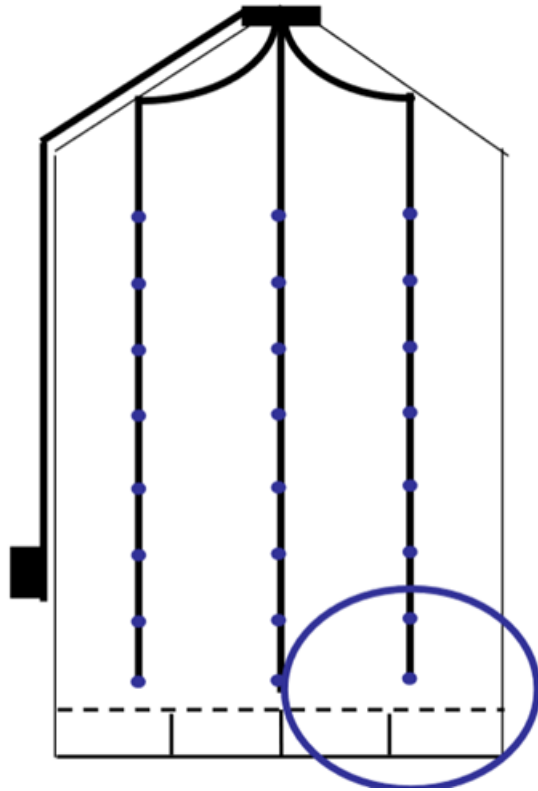


Figura 3.20 Diseño de anclaje de cables  
Autor: <http://www.conarroz.com/pdf/6Conservaciondegranos.pdf>

## CAPITULO IV.

### DISEÑO DE SOFTWARE

#### 4.1. INTRODUCCIÓN

El diseño del software abarca el diseño del programa a ser usado en el microcontrolador para que ejecute sus funciones de manera automática. También se elaborara el software de interfaz máquina-usuario para que se pueda interactuar con el sistema.

##### 4.1. 1. Programación del microcontrolador

El microcontrolador que se va a usar será marca PIC modelo 16F877A para esto se usara el software PIC C Compiler para que la función principal del elemento sea el de recopilar los datos de temperatura y lo derive hacia dos sectores. El primer dato será publicado en la pantalla LCD en la tarjeta y el segundo deberá ser transmitido vía inalámbrica para que lo reciba el computador.

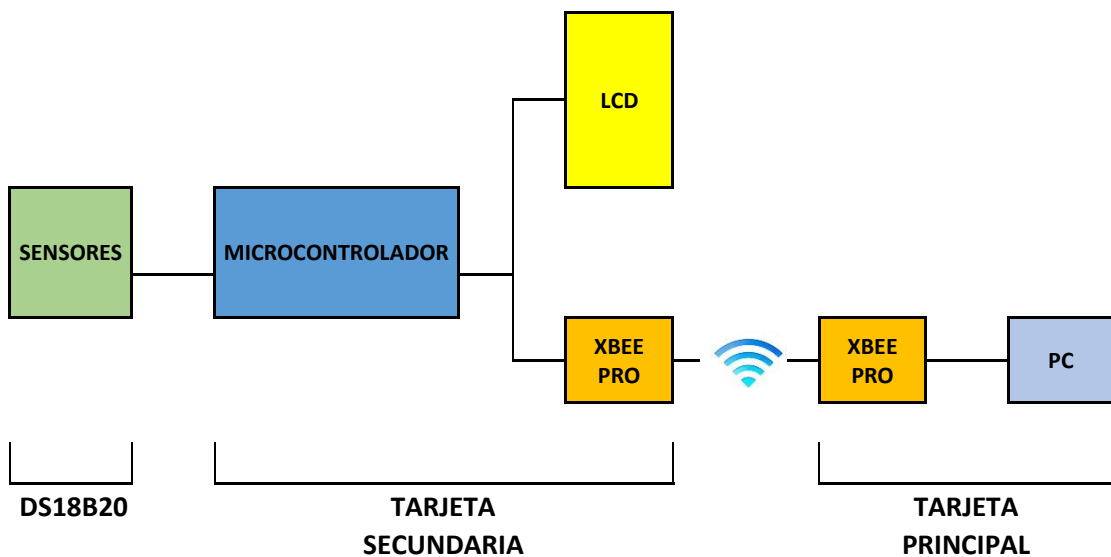


Figura 4.1 Diagrama de bloques  
Autor: Omar Tello y Alfredo Villares

Hay que tener en cuenta que al existir una comunicación 1-wire, la manera de establecer la comunicación con cada sensor es determinando su código único serial de 64-bits grabado en la memoria ROM del sensor.

Con esto se puede establecer una comunicación y requerir un paquete de datos, donde el sensor enviara la información de manera de paquete en una trama al microcontrolador.

#### **4.1.1.1. La secuencia de programación**

1. Se llama la librería del microcontrolador a ser utilizado para usar todas las bondades. Con esto también se limitara a funciones incorrectas al momento de compilar.
2. Se usa fusibles de programación que serán cargados al inicio para deshabilitar funciones que puedan causar fallas o habilitar funciones de ayuda en caso de colapso.
3. Se configura la velocidad del reloj para la velocidad del CPU, que en esta práctica será 4000000 o su equivalente en 4 MHz.
4. Se llama a librerías que contenga paquete de información de diferentes elementos. En este caso “1wire\_D2.c” es la librería perteneciente al protocolo de comunicación 1-wire. La librería “ds1820\_D2.c” es la que contiene la información del sensor de temperatura. Y “flex\_lcd.c” es la librería que maneja el LCD.
5. Se seleccionan los puertos del controlador. Se ha considerado el puerto “b” y “c”



6. Selección de los sensores según el código ROM. Se graban los códigos de los sensores en la memoria para que posteriormente poder enlazarlos con los sensores físicamente uno a uno de pendiente de su código único. La comunicación se realizara a través de un puerto del pic.

A continuación se listaran los códigos de los sensores a ser utilizados

```
40,106,87,102,4,0,0,209
40,81,74,102,4,0,0,84,
40,188,116,102,4,0,0,17,
40,144,69,102,4,0,0,147,
40,202,102,102,4,0,0,196,
40,60,99,102,4,0,0,214,
40,172,98,102,4,0,0,170,
40,121,98,102,4,0,0,133,
40,75,94,102,4,0,0,163,
40,6,43,102,4,0,0,35};
```

7. Función leer temperatura y mostrar en LCD. El controlador debe mostrar en secuencia y en orden las temperaturas que estén en el cable.
8. Función leer y enviar temperaturas como bucle infinito para que de esta manera la PC siempre que solicite información pueda recibir datos.

A continuación detallaremos una de las librerías que se van a utilizar:

```
/** read1wire() */
```

```
/*This function reads the 8 -bit data via the 1-wire sensor. */
```

```
/* */
```

```
/*Parameters: */
```

```
/*Returns: 8-bit (1-byte) data from sensor */
```

Esta librería describe su función como lectura de 8 bits en una vía de datos de un sensor 1-wire.

El sensor DS18B20 envía en un paquete de 8 bits los datos

#### 4.1.1.2. El programa del microcontrolador

```
#include <16F877.h> //tipo de microcontrolador
#FUSES NOWDT, HS, PUT, NOPROTECT, NODEBUG, BROWNOUT,
NOLVP, NOCPD, NOWRT //fusibles de programacion.
#use delay(clock=4000000) //velocidad del reloj
#use rs232(baud=9600, xmit=PIN_C6,rcv=PIN_C7)

#include <1wire_D2.c> //libreria para comunicacion onewire
#include <ds1820_D2.c> //libreria para el manejo del sensor de temperatura
#include <flex_lcd.c> ///libreria para el manejo de la LCD
#BYTE port_b=0x06 //asignacion del puerto C
#BYTE port_c=0x07 //asignacion del puerto C

const int datos1[80]={40,106,87,102,4,0,0,209, //Codigo ds1820 #01
                     40,81,74,102,4,0,0,84,
                     40,188,116,102,4,0,0,17,
                     40,144,69,102,4,0,0,147,
                     40,202,102,102,4,0,0,196,
                     40,60,99,102,4,0,0,214,
                     40,172,98,102,4,0,0,170,
```

```
40,121,98,102,4,0,0,133,  
40,75,94,102,4,0,0,163,  
40,6,43,102,4,0,0,35};
```

```
void medicion(void)  
{  
float temperature;//variable float para almacenar el valor de la temperatura  
int k=0;  
int i=0;  
lcd_gotoxy(1,1);  
lcd_putc("MIDIENDO ");  
  
for (i=0; i<78; i=i+8)  
{  
k=k+1;  
temperature =  
ds1820_read_slaveD2(datos1[i],datos1[i+1],datos1[i+2],datos1[i+3],datos1[i  
+4],datos1[i+5],datos1[i+6],datos1[i+7]); //lee la temperatura  
lcd_gotoxy(1,2); //posiciona el cursor  
printf(lcd_putc,"TEMP %u: %2.1f ",k, temperature/8); //presenta la  
temperatura en LCD  
lcd_putc(178); //simbolo de grados  
lcd_putc("C "); //  
delay_ms(1000);  
transmision(k,temperature);  
}  
  
}  
  
void transmision(int k,float temperature)  
{  
printf("T%d %4f\n\r",k,temperature);
```

```

}

void main()    //funcion principal
{
    int i=0;

    set_tris_c(0b10000000); //configura el puerto c como salida
    set_tris_b(0b00000000); //configura el puerto c como salida
    // set_tris_d(0b00000000); //configura el puerto c como salida
    // port_c=0; //pone el puerto C en cero
    lcd_init(); //inicializa la LCD

    lcd_putc("PROYECTO TITULACION"); //mensaje de inicio
    lcd_gotoxy(1,2);
    lcd_putc("MEDICION TEMP SILOS");
    delay_ms(1500);
    lcd_putc("\f");

    while (true) //bucle infinito
    {
        medicion();
    }

} //fin del programa

```

#### **4.1.2. Diseño del software de PC**

La primera pantalla tendrá opciones de conexión por medio de USB o serial. En el caso de este prototipo se usara comunicación por USB.

Esta es la interfaz hecha en la plataforma JAVA en donde se visualizara el silo con niveles y la cantidad de cables, para un fácil entendimiento del usuario.

Java es un lenguaje originalmente desarrollado por un grupo de ingenieros de Sun, utilizado por Netscape posteriormente como base para Javascript. Su uso se destaca en el Web, sirve para crear todo tipo de aplicaciones (locales, intranet o internet). Java es un lenguaje: de objetos independiente de la plataforma (Cabezas, Luis. 2010).

Los niveles corresponderán a nivel 1 como el nivel piso e incrementara con el tamaño del silo. Se pondrá como nivel final el 10.

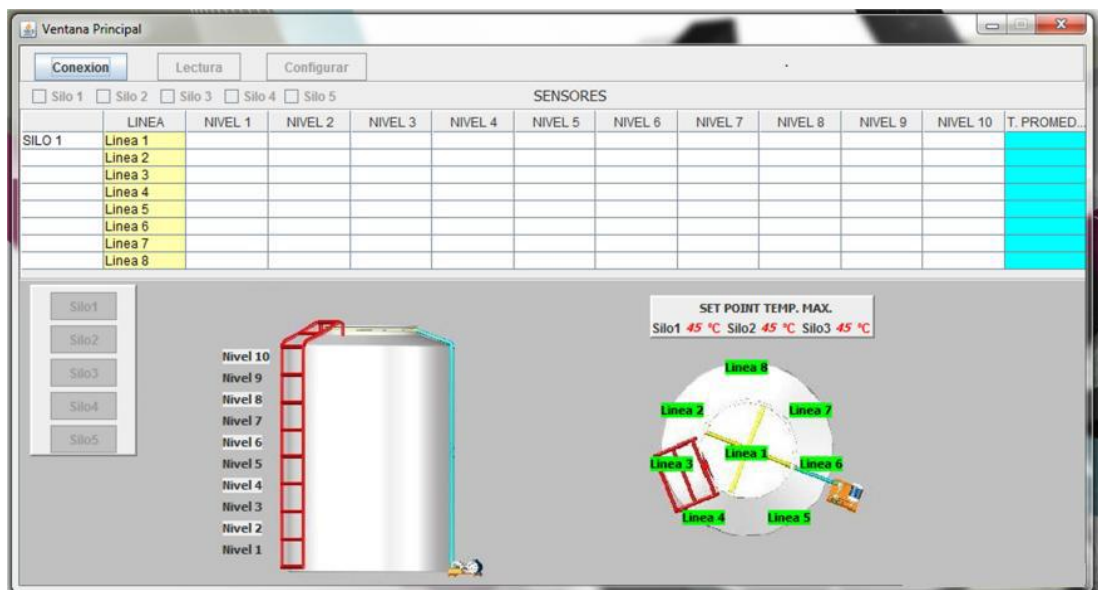
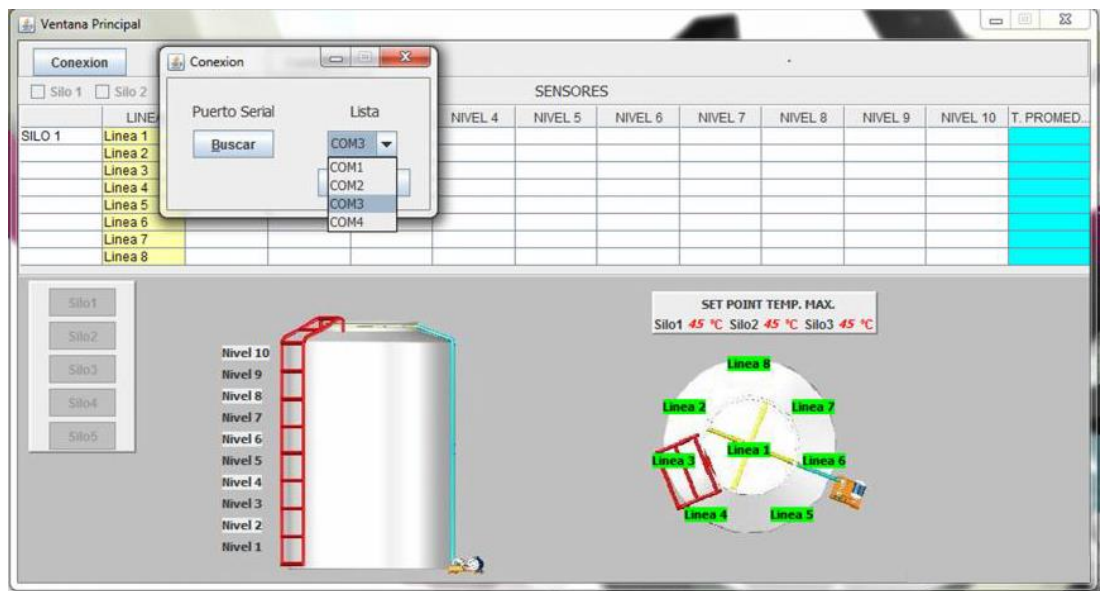


Figura 4.2 Interfaz de usuario programa lectura de temperaturas  
 Autor: Omar Tello y Alfredo Villares



*Figura 4.3 Interfaz de usuario programa lectura de temperaturas con opción de comunicación  
 Autor: Omar Tello y Alfredo Villares*

El sistema se debe configurar inicialmente para tener comunicación COM por el puerto COM3 que se ha configurado a una velocidad de 9600 baudios

El software tiene la opción de recolectar datos automáticamente o manual según lo requiera el usuario.

Para la recolección manual, el sistema recopilara los datos solo una vez mientras sea solicitado.

Para la recolección automática, el usuario seleccionara el ciclo en que se recolectara los datos y serán guardados.

La información se guarda automáticamente en la carpeta Mis Documentos y será en formato Excel para facilitar al usuario en el análisis de los datos.

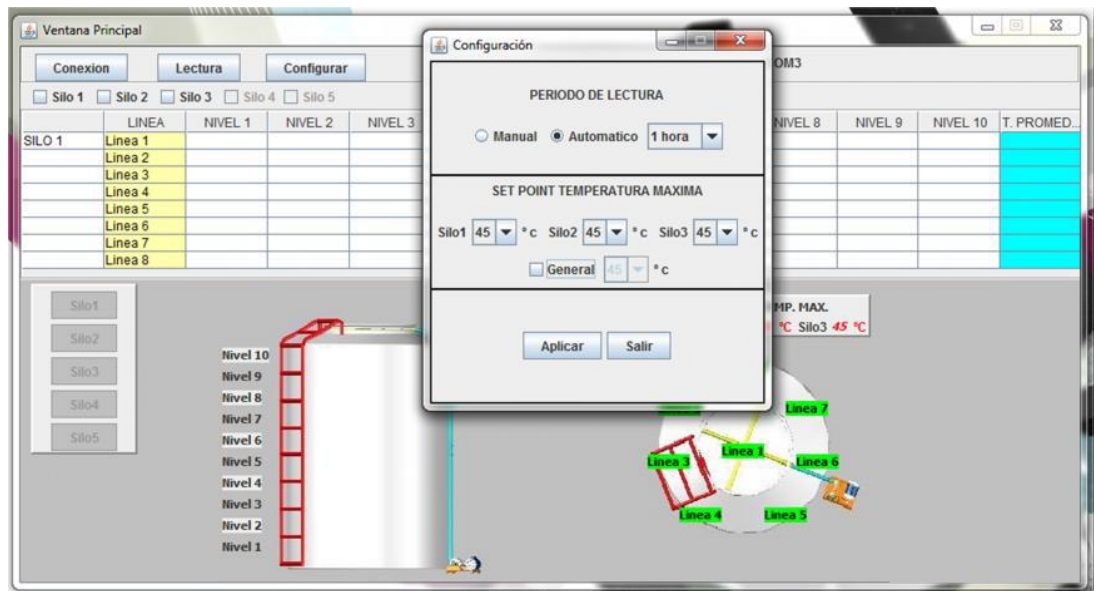


Figura 4.4 Interfaz de usuario programa lectura de temperaturas con opción de colección de datos automático  
 Autor: Omar Tello y Alfredo Villares

El software permite seleccionar puntos de alarma para el sistema, de tal manera que cuando se realice la recolección de datos pueda generar una alarma según el silo.

Esto ayudará a alertar al usuario de cuando algún parámetro este fuera de rango y poder tomar una decisión con criterio para solucionar antes de que suceda algo grave o se dañe los granos dentro del silo.

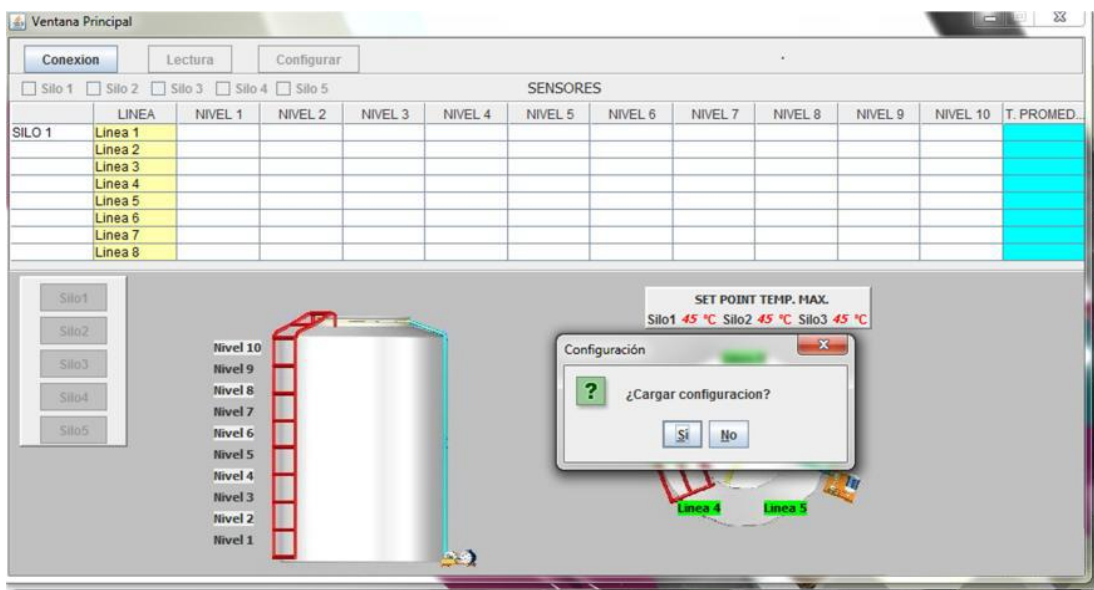


Figura 4.5 Interfaz de usuario programa lectura de temperaturas con opción de selección de alarmas por exceso de temperatura  
 Autor: Omar Tello y Alfredo Villares

Una vez ya definidas los parámetros de temperatura máxima en el programa, debemos cargar la configuración dando clic, en sí, para que el software analice los datos recibidos de la tarjeta implementada en el silo , esta comunicación de PC a la tarjeta maestra se transmite por RF radio frecuencia con la ayuda de dos módulos Xbee Pro S1 60mW.

Dentro de la programación en Java se declaran primero las variables principales

```
public class ventana_principal extends javax.swing.JFrame {
    ventana_conexion vc = new ventana_conexion();
    ventana_configuracion vcon = new ventana_configuracion();
    Resumen resu = new Resumen();
    leer_excel leer = new leer_excel();
    //JTable tabla = new JTable();
    boolean recarga = false;
    boolean lee = true;
    boolean bs1 = false;
    boolean bs2 = false;
    boolean bs3 = false;
    int time = 0;
    int temp = 25;
    int temp2 = 25;
    int temp3 = 25;
    int tempg = 25;
    boolean tem = false; //check button general
    boolean aut = false;
    boolean aut2 = false;
    boolean aut3 = false;
    boolean salir = false;
    Timer timer;
    Timer timer2;
    Timer timer3;
    String character;
    String dato[] = new String[5];
    Character silo[] = new Character[5];
    int z=0;
    boolean cont = false;
    int u=0;
```

Las funciones principales del software se detallan a continuación



```

public void autoterminar2() {...6 lines }
public void auto3(int valor) {...4 lines }
public void autoterminar3() {...9 lines }
public void pause(int time) {...7 lines }
public static String getFechaActual() {...5 lines }
public static String getHoraActual() {...5 lines }
public HSSFCellStyle setestilos(short color,short borde,int centrar, int formato) {...46 lines }
public void llenar(char a, char b, String txt,int w) {...99 lines }
public void crearformato(int w) {...85 lines }
public void limpiar_jtable() {...19 lines }
public void sumatoria() {...28 lines }
public void verif_estados(int a) {...31 lines }
public void verif_estados_set(int si) {...32 lines }
public void leer_puerto() throws PortInUseException, UnsupportedCommOperationException, IOException {...108 lines }
public void salida(String s) {...3 lines }

public ventana_principal() throws IOException {...29 lines }

public void crear_libro() {...6 lines }
public void creardirectorio() {...15 lines }
public void resetear() {...4 lines }
public void editor() {...5 lines }
public class Reminder {...23 lines }
public class Reminder2 {...14 lines }
public class Reminder3 {...16 lines }
public void auto(int valor) {...4 lines }
public void autoterminar() {...5 lines }
public void reset_etiquetas() {...10 lines }
public void reset_obsilo() {...10 lines }
public void set_etiquetas(int f) {...10 lines }
public void settemptabla() {...5 lines }
public void auto2(int valor) {...4 lines }

```

```

public void bloquear(boolean conexion,boolean leer,boolean grabar,
    boolean exportar, boolean grafico, boolean boton1 , boolean boton2,
    boolean boton3){...28 lines }

public void borrar(){...14 lines }

public String getdir()throws IOException{...4 lines }

public void guardar()throws IOException{...32 lines }

public void conexion(){...7 lines }

public void conf_btn_s1(){...4 lines }

public void conf_btn_s2(){...4 lines }

public void conf_btn_s3(){...4 lines }

public void conf_btn_s4(){...4 lines }

public void conf_btn_s5(){...4 lines }

@SuppressWarnings("unchecked")
Generated Code

private void btn_conexionActionPerformed(java.awt.event.ActionEvent evt) {...4 lines }

private void btn_exportarActionPerformed(java.awt.event.ActionEvent evt) {...11 lines }

private void btn_exportarActionPerformed(java.awt.event.ActionEvent evt) {...11 lines }

private void btn_leerActionPerformed(java.awt.event.ActionEvent evt) {...12 lines }

private void btn_configurarActionPerformed(java.awt.event.ActionEvent evt) {...22 lines }

private void btn_graficoActionPerformed(java.awt.event.ActionEvent evt) {...3 lines }

private void btn_silo1ActionPerformed(java.awt.event.ActionEvent evt) {...7 lines }

private void btn_silo2ActionPerformed(java.awt.event.ActionEvent evt) {...7 lines }

private void btn_silo3ActionPerformed(java.awt.event.ActionEvent evt) {...7 lines }

private void cb_silo2ActionPerformed(java.awt.event.ActionEvent evt) {...4 lines }

private void cb_silo3ActionPerformed(java.awt.event.ActionEvent evt) {...4 lines }

private void cb_silo4ActionPerformed(java.awt.event.ActionEvent evt) {...4 lines }

private void cb_silo5ActionPerformed(java.awt.event.ActionEvent evt) {...4 lines }

private void btn_silo4ActionPerformed(java.awt.event.ActionEvent evt) {...3 lines }

private void formWindowClosing(java.awt.event.WindowEvent evt) {...12 lines }

private void cb_silo1ActionPerformed(java.awt.event.ActionEvent evt) {...4 lines }

/**...3 lines */

```

## Capítulo V

### PRESUPUESTO

#### 5.1. INTRODUCCIÓN

Los materiales usados durante el desarrollo del proyecto se detallan a continuación con sus respectivos precios para dimensionar la inversión económica del mismo.

Cant.	Item	Valor Unit.	Valor Total
1	Tarjeta de control	\$ 150	\$ 150
1	transformador 110-12V para tarjeta	\$ 40	\$ 40
1	LCD 2x16	\$ 20	\$ 20
1	PIC 16F877	\$ 16	\$ 16
8	Borneras de 3 polos	\$ 0,50	\$ 4
5	Borneras de 2 polos	\$ 0,40	\$ 2
1	Regulador LM7805	\$ 2	\$ 2
2	74LS04	\$ 0,80	\$ 2
4	Capacitores 20 uF	\$ 0,30	\$ 1
1	Integrado 555	\$ 0,50	\$ 1
1	elementos varios	\$ 15	\$ 15
10	DS18B20	\$ 4	\$ 40
2	Xbee Pro 60 mW	\$ 90	\$ 180
1	XbBee Xplorer serial	\$ 50	\$ 50
1	Breaker 2 polos 1 Amp	\$ 58	\$ 58
1	convertidor serial USB	\$ 15	\$ 15
25	cable concéntrico 3x#16	\$ 2	\$ 38
			<b>\$ 633</b>

*Tabla 5.1 Presupuesto de prototipo de lectura de temperaturas  
Autor: Omar Tello y Alfredo Villares*

La mano de obra para un solo elemento es alta. Se puede considerar que esta alrededor de los \$1500 debido a armar las tarjetas, cables e instalación del sistema.

El resto de cables puede estar en un estimado de \$500. Teniendo en cuenta esto, el total del proyecto está en \$2633.

Un sistema importado cuesta \$5000, donde es un sistema cerrado y muchas veces no existen repuestos locales.

Con esto se determina que la solución ofrecida en la tesis es factible en precio y en beneficios.

Se ha tomado en consideración la elaboración del sistema a través de PLC Siemens S7-300. Se detalla a continuación el presupuesto

<b>Cant.</b>	<b>Item</b>	<b>Valor unit.</b>	<b>Valor Total</b>
1	PLC 315 2 PN DP	4000	4000
4	Tarjeta AIx8	950	3800
32	PT100	60	1920
			<b>9720</b>

*Tabla 5.2 Presupuesto de sistema con PLC  
Autor: Omar Tello y Alfredo Villares*

Según la tabla 5.1 y la tabla 5.2 se considera un costo muy elevado y sin incluir mano de obra y cables. Siendo la opción de PLC y PT-100 no ejecutable en cuanto a inversión económica.

Para la propuesta del prototipo realizado en la tesis, se considerara su producción en un mayor número de unidades donde se estiman costos aproximados a \$2000, lo que haría más competitivas a esta propuesta.

## **CAPITULO VI**

### **PRESENTACIÓN DE RESULTADOS**

#### **6.1. INTRODUCCIÓN**

La impresión de la tarjeta a partir del ruteado se envió a Colombia. La calidad recibida es muy buena y nos facilitó el ensamblado. Los acabados son limpios y se cortaron los excedentes.

Los datos obtenidos mediante las pruebas, demuestran que el sistema es fiable y puede determinar varios puntos de temperatura como puntos fríos o focos de calor.

Según las tablas 6.1 y 6.2 donde se consiguieron las lecturas de las temperaturas se visualiza las diferentes etapas del cable. Con esto el propósito de realizar un prototipo da resultados favorables y cumple las funciones esperadas.

En la figura 6.4 muestra una avería del sistema, en donde se detecta que un sensor no transmite datos. Es fácil ver cuál es mediante la pantalla en la tarjeta, pero anticipadamente se detecta con mayor rapidez a través de un led ubicado en la parte inferior de la tarjeta.

Es un buen resultado del prototipo de obtener los datos de un cable de 25 metros de manera inalámbrica con una distancia de separación de 250 m.

El precio es un punto favorable en cuanto a su fabricación, ya que comprar en exterior o fabricar con PLC puede ser muy costoso.

### 6.1.1. Pruebas físicas

Se conecta el cable concéntrico de 3 hilos a la tarjeta y se energiza con 110 VAC del puerto eléctrico doméstico.

Primero se toman los datos de temperatura ambiente donde se está realizando la prueba.



*Figura 6.1 Conexión del sistema y pruebas  
Autor: Omar Tello y Alfredo Villares*

La tabla de datos obtenida se detalla a continuación tomando con referencia la temperatura ambiente

Nivel 10	28 C
Nivel 9	28 C
Nivel 8	28 C

Nivel 7	28 C
Nivel 6	28 C
Nivel 5	28 C
Nivel 4	28 C
Nivel 3	28 C
Nivel 2	28 C
Nivel 1	28 C

*Tabla 6.1 Recolección de datos a temperatura ambiente  
Autor: Omar Tello y Alfredo Villares*

Luego se simula con temperatura ambiente y en un sensor se usa una pistola de calor y en otro punto se usa un hielo.

Los detalles se listan en la siguiente tabla.

Nivel 10	28 C
Nivel 9	28 C
Nivel 8	28 C
Nivel 7	28 C
Nivel 6	28 C
Nivel 5	55 C
Nivel 4	28 C
Nivel 3	28 C
Nivel 2	5 C
Nivel 1	28 C

*Tabla 6.2 Recolección de datos a temperatura ambiente con modificaciones  
Autor: Omar Tello y Alfredo Villares*

Se puede notar que el funcionamiento del prototipo es correcto y que muestra temperaturas en diferentes niveles y grados.

La capacidad de dar la lectura de capas es exactamente lo necesario en la aplicación en real con un silo. De esta manera se puede localizar focos de calentamiento, o en su defecto líneas de aireación muy fuertes.

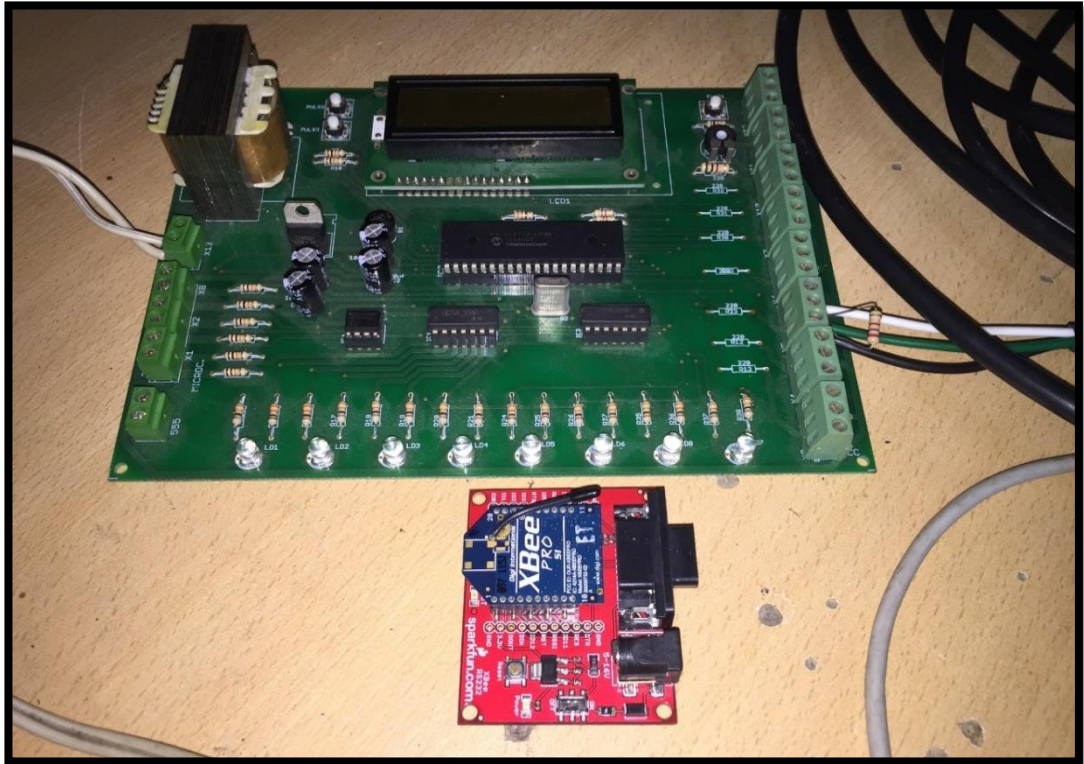
En la pantalla de la tarjeta se puede ver si existe alguna falla en un sensor.



*Figura 6.2 Visualización de sistema con falla  
Autor: Omar Tello y Alfredo Villares*

A continuación se muestra la tarjeta de recolección de datos y la tarjeta XBee Xplorer que será la que enviara los datos de manera inalámbrica.





*Figura 6.3 Sistema superior con tarjetas de lectura y comunicación  
Autor: Omar Tello y Alfredo Villares*

Las pruebas de distancia las realizamos hasta una distancia de 400 metros, existiendo comunicación del sistema.

Se realizó pruebas con inferencias de paredes en una cuadra y se obtuvo resultados positivos de comunicación.

## CONCLUSIONES

- Los sensores digitales DS18B20 por su valor económico, exactitud, fiabilidad, facilidad para uso de monitorización de temperatura y en particular el tiempo de respuesta es el elemento más indicado para el prototipo de lectura de temperaturas en silo
- El uso de nuevas tecnologías permite al sistema de control ser más eficiente, por ejemplo para la adquisición de señales de un sensor analógico de temperatura tomaba tiempo en la conversión analógica/Digital para entonces poder obtener datos y tratar la información. Con la tecnología 1-wire se realiza el paso directamente al existir la lectura de bits.
- El costo del proyecto debido a sus componentes electrónicos de alta precisión hacen que el proyecto sea rentable y sea posible su implementación.
- El diseño de un sistema de control de temperaturas es importante puesto que indica al usuario de manera inmediata por medio de la PC, de que existe alguna alerta de temperatura y pueda tomar acciones sobre el mismo.
- La interfaz del programa de computadora es amigable y de fácil entendimiento para el usuario.
- El sistema es muy fiable y tiene un diseño que permite corrección rápida. Esto se debe a que la tarjeta que lee las temperaturas va en la parte superior del silo y se interactúa directamente con los sensores por medio del LCD.

## RECOMENDACIONES

- Con el trabajo realizado hasta ahora, se puede determinar de manera teórica que el sistema trabajaría bien en silos de mayores dimensiones. La iniciativa de futuros ingenieros de continuar con la investigación y la implementación del sistema sería de avance tecnológico. También analizar otras variables como humedades externas e internas al silo.
- Las computadoras modernas no tienen puerto SERIAL y por esto se recomienda comprar el convertidor SERIAL-USB
- La tarjeta secundaria será montada en la parte superior del silo y eso provocara que parámetros ambientales por lo que se recomienda usar un tablero que este bajo techo.
- La tarjeta se ubicara dentro de un tablero y debido a la humedad, polvo y los insectos se recomienda sellar uniones y orificios con silicón o caucho para hermetizar.
- El cable estará en contacto con los granos en movimiento y en caída libre, por lo que se recomienda usar refuerzos sobre el cable como abrazadera metálica o recubrimiento plástico para prevenir la abrasión.
- En el montaje es recomendado tratar de usar el camino con menor cantidad de obstáculos en la línea de transmisión inalámbrica. Tratar de evitar árboles, postes o antenas grandes.
- Para realizar la programación del sistema hay que tener en cuenta que cada sensor tiene un código único para poder identificarlo, por lo que se recomienda etiquetarlos y ordenarlos.

- Se recomienda ser limpios en el sitio de trabajo al soldar elementos en la laca para evitar cortocircuitos luego al energizar.
- Se recomienda que al usar el software se introduzcan las alarmas para estar prevenidos.

## BIBLIOGRAFÍA

1. Adafruit. (2012). Datasheets Xbee. Retrieved Diciembre 16, 2014, from <http://www.adafruit.com/datasheets/XBee%20ZB%20User%20Manual.pdf>
2. Agencia de Noticias Andes (Abril, 2014). Ecuador estará en condiciones para exportar maíz en el 2015 Retrieved Diciembre 16, 2014, from <http://www.andes.info.ec/es/noticias/ecuador-estara-condiciones-exportar-maiz-2015.html>
3. Avilés, Pablo. Salto, Paúl. (2010). Diseño, construcción e implementación de un triedro electrónico según la norma INEN para monitoreo de temperatura de la Empresa Fibro Acero.( Cuenca). Retrieved Diciembre 18, 2014, from <http://www.dspace.ups.edu.ec/handle/123456789/422>
4. Bairesrobotics. (2007). Microcontroladores Retrieved Diciembre 18, 2014 from <http://www.bairesrobotics.com.ar/>
5. Bayle, Julien. (2013) C Programming for Arduino Retrieved Diciembre 16, 2014, from <https://www.safaribooksonline.com/library/view/c-programming-for/9781849517584/ch08s06.html>
6. Bell, Charles. (2013). Beginning Sensor Networks with Arduino and Raspberry Pi Retirved Diciembre 18, 2014 from <https://www.safaribooksonline.com/library/view/beginning-sensor-networks/9781430258247/>

7. Cabezas, Luis. (2010) Elaboración de un software libre estadístico como apoyo académico al cuarto nivel de la carrera de ingeniería en estadística informática e. Retrieved Diciembre 16, 2014, from <http://dspace.esPOCH.edu.ec/bitstream/123456789/753/1/226T0015.pdf>
8. Caicedo, Homero. Obando, Diego. (2011). Diseño e implementación de un sistema automatizado para el control del acuario del Museo de Ciencias Biológicas de la Escuela Politécnica Nacional. Quito. Retrieved Diciembre 18, 2014, from <http://bibdigital.epn.edu.ec/handle/15000/4602>
9. Carrera, Carlos. Guaiña, Luis. (2010). Control remoto de robots usando módulos de radiofrecuencia xbee a 2.4ghz con capacidad de comunicación serial a datalogger e interfaz gráfica. Retrieved Diciembre 18, 2014 from <http://www.dspace.espol.edu.ec/handle/123456789/11180>
10. Casini, Cristiano. Rodríguez, Juan. (2003). Conservación de Granos en Chacra con Sistemas Tradicionales. Retrieved Febrero 16, 2015, from [http://inta.gob.ar/documentos/conservacion-de-granos-en-chacra-con-sistemas-tradicionales/at\\_multi\\_download/file/INTA-%20Conservacion-Granos-En-Chacra-Con-Sistemas-Tradicionales.pdf](http://inta.gob.ar/documentos/conservacion-de-granos-en-chacra-con-sistemas-tradicionales/at_multi_download/file/INTA-%20Conservacion-Granos-En-Chacra-Con-Sistemas-Tradicionales.pdf)
11. Ccarita, Fredy. (2011). DISEÑO DE CONSTRUCCIONES SOLDADAS. Retrieved Febrero 16, 2015, from <http://es.slideshare.net/alanmecanica/diseo-de-un-silo-estructura-metalica>

12. Chávez, J. (2003). Red de comunicación One Wire para el monitoreo de variables atmosféricas. Retrieved Diciembre 18, 2014, from <http://bibliodigital.itcr.ac.cr/handle/2238/154>
13. Daraviña, Giancarlo. Valencia, Richard. (2014). Diseño de sistema de control de optimización ambiental para cultivo hidropónico. Retrieved Diciembre 16, 2015, from <http://repositorio.utp.edu.co/dspace/handle/11059/4326>
14. Don Wilcher (2014) Make: Basic Arduino Projects Retrieved Diciembre 16, 2014, from <https://www.safaribooksonline.com/library/view/make-basic-arduino/9781449360658/>
15. Duarte, Andres. (2014). Arduino y Xbee Retrieved Diciembre 16, 2014, from <http://www.andresduarte.com/arduino-y-xbee>
16. Ediagro, (diciembre, 2006). CONSERVACION DE GRANOS ALMACENADOS. Retrieved Diciembre 18, 2014, from <http://www.conarroz.com/pdf/6Conservaciondegranos.pdf>
17. Elin. (1991). Что такое 1-Wire?. Retrieved Diciembre 16, 2014, from <http://www.elin.ru/1-Wire/>
18. Galarza, M. (2009). Diseño y construcción del sistema de frenado dinámico para un vehículo eléctrico. Retrieved Diciembre 18, 2014, from <http://bibdigital.epn.edu.ec/handle/15000/1276>

19. González, Alejandro (2012). Documento sobre tipos de periféricos. Periféricos de entrada, salida y entrada/salida.. Retrieved Diciembre 18, 2014 from <http://sedici.unlp.edu.ar/handle/10915/30086>
20. González, Elfrich. (2009). DISEÑO Y CONSTRUCCION DE UN ROBOT SUMO. Retrieved Diciembre 16, 2014, from [http://webs.ono.com/lmoliver/usr\\_1995168598.pdf](http://webs.ono.com/lmoliver/usr_1995168598.pdf)
21. González, Víctor. (2012). Construcción de un sistema de control de iluminación y sistema de seguridad de puertas y ventanas de una casa prototipo mediante mensajes de texto SMS utilizando el microcontrolador 16F877. Diciembre 18, 2014 from <http://bibdigital.epn.edu.ec/handle/15000/4869>
22. Helyette, Geman (2015). Agricultural Finance: From Crops to Land, Water and Infrastructure. Retrieved Febrero 16, 2015, from <https://www.safaribooksonline.com/library/view/agricultural-finance-from/9781118827376/>
23. Herrera, María. Chalco, Oscar. (2013). Construcción de un módulo didáctico para el aprendizaje de microcontroladores controlado por Labview. Retrieved Diciembre 18, 2014 from <http://bibdigital.epn.edu.ec/handle/15000/6889>
24. Khalil, Mostafa. (2012). Storage Implementation in vSphere® 5.0. Retrieved Diciembre 16, 2014, from <https://www.safaribooksonline.com/library/view/storage-implementation-in/9780132842228/>



25. Lara, María Belén (2010). Implementación de un prototipo de comunicaciones mediante la utilización del estándar IEEE 802.15.4. Retrieved Diciembre 18, 2014 from <http://repositorio.espe.edu.ec/handle/21000/134>
26. Laub. 2012Sensorino - Auslesen verschiedener Sensoren mit Arduino Retrieved Diciembre 16, 2014, from [http://www.laub-home.de/wiki/Sensorino\\_-\\_Auslesen\\_verschiedener\\_Sensoren\\_mit\\_Arduino](http://www.laub-home.de/wiki/Sensorino_-_Auslesen_verschiedener_Sensoren_mit_Arduino)
27. Loxone. (2015). Instrucciones para la instalación de la 1-Wire Extension. Retrieved Diciembre 18, 2014, from <http://www.loxone.com/eses/servicio/documentacion/extensiones/1-wire.html>
28. MAGAP. (2014). BOLETÍN DE PRECIOS AL PRODUCTOR ANÁLISIS DE VARIACIONES. Retrieved Diciembre 16, 2014, from <http://sinagap.agricultura.gob.ec/phocadownloadpap/pproductor/2014/aBolet%C3%ADn%20de%20precios%20al%20productor%20Enero%202014.pdf>
29. Maxim integrate,(2009, Marzo). Comparación de la DS18B20 y DS18S20 1-Wire® Termómetros digitales. Retrieved Diciembre 18, 2014, from <http://www.maximintegrated.com/en/app-notes/index.mvp/id/4377>
30. Platt, Charles. (2009). Make: Electronics. Retrieved Diciembre 16, 2014, from <https://www.safaribooksonline.com/library/view/make-electronics/9781449377267/>
31. PRECOP. (2007). PROYECTO EFICIENCIA DE COSECHA Y POSTCOSECHA DE GRANOS. Retrieved Febrero 16, 2015, from <http://www.cosechaypostcosecha.org/data/folletos/conservacionDeGranos.pdf>

32. Reinoso, Diego. Estrada, Juan. (2009). Diseño e implementación de un prototipo de monitoreo remoto de contaminación ambiental utilizando tecnología GSM. Retrieved Diciembre 18, 2014, from <http://bibdigital.epn.edu.ec/handle/15000/1137>
33. Sal Cangeloso, (2012) Iluminación LED Retrieved Diciembre 16, 2014, from <https://www.safaribooksonline.com/library/view/led-lighting/9781449334758/>
34. Suarez, Jose. (2006). Como gobernar un display LCD alfanumérico basado en el controlador HD44780 Retrieved Diciembre 16, 2014, from <http://eii.unex.es/profesores/jisuarez/descargas/ip/>
35. Tami,Camilo. Torres, Omar. Mejía, Andrés. (2014). Prototipo sistema de monitoreo para el encalado en el proceso panelero. Retrieved Diciembre 16, 2014, from <http://repository.ucatolica.edu.co:8080/jspui/handle/10983/1388>
36. VIP, VIP (2014) Conexión de hardware DS18B20 y programación de software. Retrieved Diciembre 16, 2014, from <http://www.cqvip.com/qk/98452x/200112/6057374.html>

## GLOSARIO

<b>B.M.H.B.</b>	Código inglés para el diseño de silos y tolvas
<b>DCE</b>	Equipo para la comunicación de datos
<b>DTE</b>	Equipo Terminal de datos
<b>LCD</b>	Liquid Crystal Display (Pantalla de Cristal Líquido)
<b>LED</b>	(light emitting diode) Diodo emisor de luz.- fabricados con compuestos de galio, capaces de emitir luz cuando son atravesados por una corriente. Se denomina de este modo al diodo capaz de emitir energía luminosa a partir de la recombinación de electrones y huecos en su unión.
<b>mA,A</b>	Mili amperios. Unidad de medición de corriente eléctrica
<b>Materia prima</b>	La comida para animales, o la materia prima utilizada en una unidad de procesamiento (caña de azúcar es una materia prima para la producción de etanol en Brasil)
<b>mV,V</b>	Mili voltios, voltios. Unidad de medición de voltaje

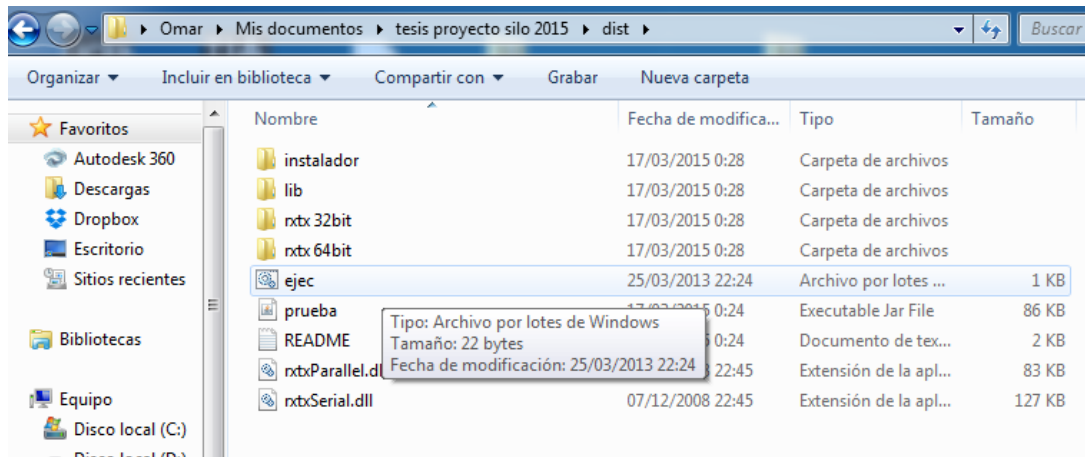
<b>One Wire</b>	Comunicación serial asíncrona mediante un solo conductor desarrollado por Dallas Semiconductor Inc.
<b>PAN</b>	Red de Área Personal. Una red de comunicación de datos que incluye un Coordinador y uno o más routers/dispositivos finales. La formación de la Red se rige por el ancho máximo de la red, máximo número de Routers y máximo número de dispositivos finales.
<b>PIC</b>	Programable Integrated Circuit (Circuito integrado Programable)
<b>RF</b>	Abreviatura de Radio Frecuencia. Este término se refiere a una corriente alterna (AC) con la característica especial de que, si ésta sirve de entrada a una antena, se genera un campo EM adecuado para las comunicaciones inalámbricas.
<b>Silo</b>	Estructura de contención usada para almacenar sólidos disgregados
<b>UART</b>	(Universal Asynchronous Receiver-Transmitter) es un componente de la computadora que controla las comunicaciones en los puertos en serie
<b>VAC</b>	Voltaje con corriente alterna. Las instalaciones eléctricas convencionales utilizan corriente AC.
<b>VDD</b>	Alimentación positiva (+5Vdc)
<b>VSS</b>	Alimentación negativa (masa)

# Anexos

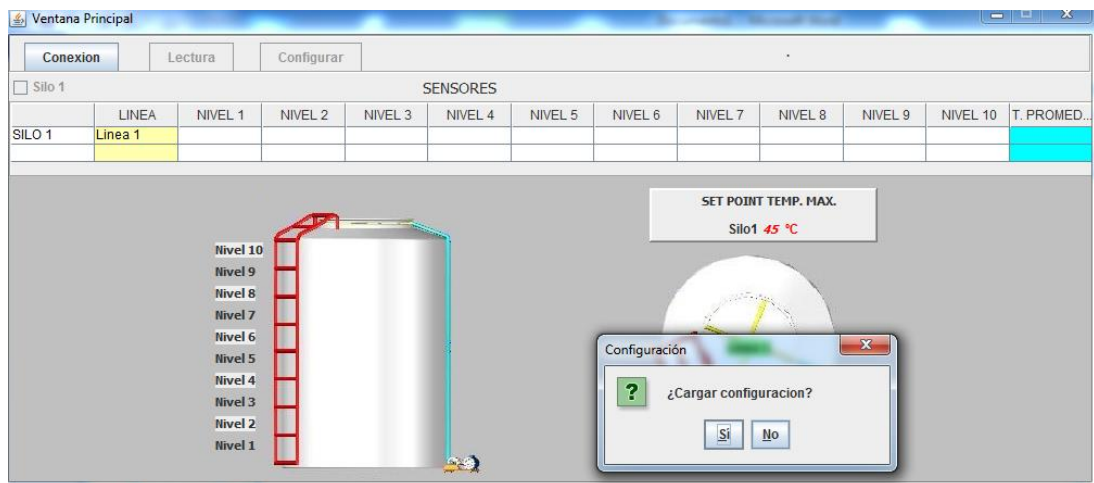
## ANEXO 1.

### MANUAL DE USUARIO

#### 1. Abrir ejecutable: prueba

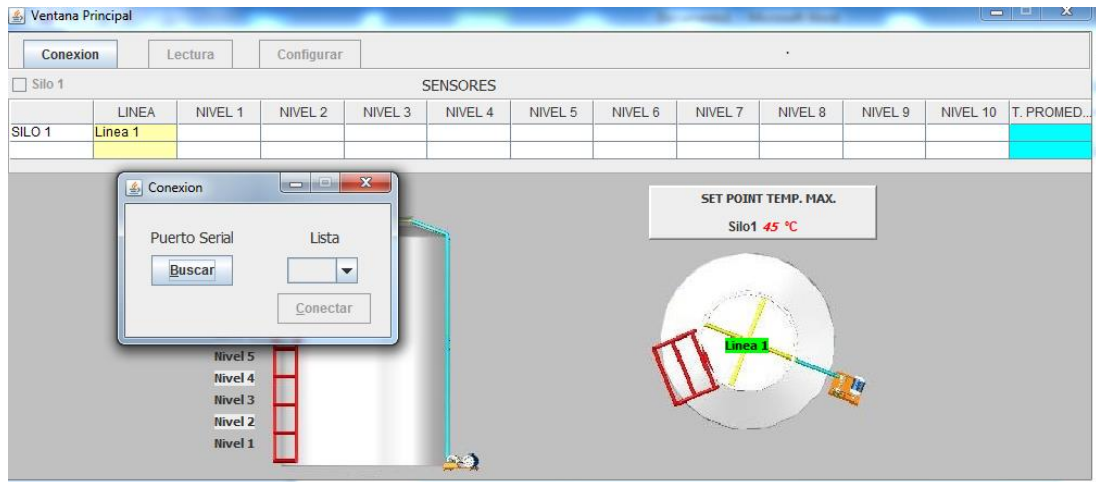


#### 2. El sistema pregunta cargar configuración. Se da clic en NO



#### 3. Se selecciona conexión. Se usa el puerto COM respectivo del equipo. Se conecta.

Luego se selecciona el silo y se habilita lectura manual. En caso de querer secuencia automática se selección el tiempo.



4. Para seleccionar alarmas se configura en la pestaña CONFIGURAR. Estas alarmas se notaran de color rojo en los cuadros para los valores mayores al seleccionado.

## ANEXO 2.

### Librería en JAVA

```
/******1Wire Class******/  
  
/*Description: This class handles all communication */  
  
/* between the processor and the 1wire */  
  
/* sensors.  
  
/*******/  
  
/*-----1-wire definitions-----*/  
  
#define ONE_WIRE_PIND2 PIN_B3  
  
/******1-wire communication functions******/  
  
/******onewire_reset******/  
*****/  
  
/*This function initiates the 1wire bus */  
  
/* */  
  
/*PARAMETERS: */  
  
/*RETURNS: */  
  
/******  
**/
```



```

void onewire_resetB3() // OK if just using a single permanently connected device

{

output_low(ONE_WIRE_PIND2);

delay_us( 500 ); // pull 1-wire low for reset pulse

output_float(ONE_WIRE_PIND2); // float 1-wire high

delay_us( 500 ); // wait-out remaining initialisation window.

output_float(ONE_WIRE_PIND2);

}

```

```

/***** onewire_write()
*****/

/*This function writes a byte to the sensor.*/

/* */

/*Parameters: byte - the byte to be written to the 1-wire */

/*Returns: */

/*****
**/

```

```

void onewire_writeB3(int data)

{

int count;

```

```

for (count=0; count<8; ++count)
{
    output_low(ONE_WIRE_PIND2);

    delay_us( 2 ); // pull 1-wire low to initiate write time-slot.

    output_bit(ONE_WIRE_PIND2, shift_right(&data,1,0)); // set output bit on 1-wire

    delay_us( 60 ); // wait until end of write slot.

    output_float(ONE_WIRE_PIND2); // set 1-wire high again,

    delay_us( 2 ); // for more than 1us minimum.

}
}

```

```

/***** read1wire() *****/

```

```

/*This function reads the 8 -bit data via the 1-wire sensor. */

```

```

/* */

```

```

/*Parameters: */

```

```

/*Returns: 8-bit (1-byte) data from sensor */

```

```

/*****

```

```

**/

```

```

int onewire_readB3()

```

```

{

```

```

    int count, data;

```

```
for (count=0; count<8; ++count)

{

output_low(ONE_WIRE_PIND2);

delay_us( 2 ); // pull 1-wire low to initiate read time-slot.

output_float(ONE_WIRE_PIND2); // now let 1-wire float high,

delay_us( 8 ); // let device state stabilise,

shift_right(&data,1,input(ONE_WIRE_PIND2)); // and load result.

delay_us( 120 ); // wait until end of read slot.

}

return( data );

}
```

### ANEXO 3.

#### Librería PIC C COMPILER

```
#include <16F877.h> //tipo de microcontrolador
#FUSES NOWDT, HS, PUT, NOPROTECT, NODEBUG, BROWNOUT,
NOLVP, NOCPD, NOWRT //fusibles de programacion.
#use delay(clock=4000000) //velocidad del reloj
#use rs232(baud=9600, xmit=PIN_C6,rcv=PIN_C7)

#include <1wire_D2.c> //libreria para comunicacion onewire
#include <ds1820_D2.c> //libreria para el manejo del sensor de temperatura
#include <flex_lcd.c> ///libreria para el manejo de la LCD
#BYTE port_b=0x06 //asignacion del puerto C
#BYTE port_c=0x07 //asignacion del puerto C

const int datos1[80]={40,106,87,102,4,0,0,209, //Codigo ds1820 #01
                     40,81,74,102,4,0,0,84,
                     40,188,116,102,4,0,0,17,
                     40,144,69,102,4,0,0,147,
                     40,202,102,102,4,0,0,196,
                     40,60,99,102,4,0,0,214,
                     40,172,98,102,4,0,0,170,
                     40,121,98,102,4,0,0,133,
                     40,75,94,102,4,0,0,163,
                     40,6,43,102,4,0,0,35};

void medicion(void)
{
float temperature;//variable float para almacenar el valor de la temperatura
int k=0;
int i=0;
lcd_gotoxy(1,1);
```

```

lcd_putc("MIDIENDO ");

for (i=0; i<78; i=i+8)
{
    k=k+1;
    temperature =
ds1820_read_slaveD2(datos1[i],datos1[i+1],datos1[i+2],datos1[i+3],datos1[i
+4],datos1[i+5],datos1[i+6],datos1[i+7]); //lee la temperatura
    lcd_gotoxy(1,2); //posiciona el cursor
    printf(lcd_putc,"TEMP %u: %2.1f ",k, temperature/8); //presenta la
temperatura en LCD
    lcd_putc(178); //simbolo de grados
    lcd_putc("C "); //
    delay_ms(1000);
    transmision(k,temperature);
}

}

void transmision(int k,float temperature)
{
    printf("T%d %4f\n\r",k,temperature);
}

void main() //funcion principal
{
    int i=0;

    set_tris_c(0b10000000); //configura el puerto c como salida
    set_tris_b(0b00000000); //configura el puerto c como salida
    // set_tris_d(0b00000000); //configura el puerto c como salida
    // port_c=0; //pone el puerto C en cero

```

```
lcd_init(); //inicializa la LCD

lcd_putc("PROYECTO TITULACION"); //mensaje de inicio
lcd_gotoxy(1,2);
lcd_putc("MEDICION TEMP SILOS");
delay_ms(1500);
lcd_putc("\f");

while (true) //bucle infinito
{
    medicion();
}

} //fin del programa
```

## ANEXO 4.



# DS18B20 Programmable Resolution 1-Wire Digital Thermometer

### DESCRIPTION

The DS18B20 digital thermometer provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 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 DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

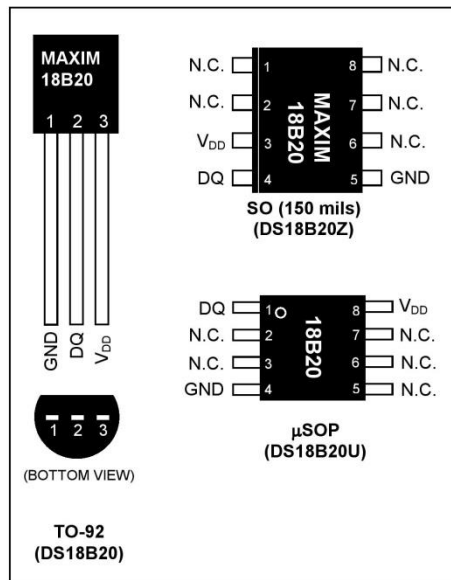
Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

### FEATURES

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Each Device has a Unique 64-Bit Serial Code Stored in an On-Board ROM
- Multidrop Capability Simplifies Distributed Temperature-Sensing Applications
- Requires No External Components
- Can Be Powered from Data Line; Power Supply Range is 3.0V to 5.5V
- 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  $+85^{\circ}\text{C}$
- Thermometer Resolution is User Selectable from 9 to 12 Bits
- Converts Temperature to 12-Bit Digital Word in 750ms (Max)

- User-Definable Nonvolatile (NV) Alarm Settings
- Alarm Search Command Identifies and Addresses Devices Whose Temperature is Outside Programmed Limits (Temperature Alarm Condition)
- Available in 8-Pin SO (150 mils), 8-Pin  $\mu\text{SOP}$ , and 3-Pin TO-92 Packages
- Software Compatible with the DS1822
- Applications Include Thermostatic Controls, Industrial Systems, Consumer Products, Thermometers, or Any Thermally Sensitive System

### PIN CONFIGURATIONS



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

For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim's website at [www.maximintegrated.com](http://www.maximintegrated.com).

REV: 042208

**ORDERING INFORMATION**

PART	TEMP RANGE	PIN-PACKAGE	TOP MARK
DS18B20	-55°C to +125°C	3 TO-92	18B20
DS18B20+	-55°C to +125°C	3 TO-92	18B20
DS18B20/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20-SL/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20-SL+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20U	-55°C to +125°C	8 $\mu$ SOP	18B20
DS18B20U+	-55°C to +125°C	8 $\mu$ SOP	18B20
DS18B20U/T&R	-55°C to +125°C	8 $\mu$ SOP (3000 Piece)	18B20
DS18B20U+T&R	-55°C to +125°C	8 $\mu$ SOP (3000 Piece)	18B20
DS18B20Z	-55°C to +125°C	8 SO	DS18B20
DS18B20Z+	-55°C to +125°C	8 SO	DS18B20
DS18B20Z/T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20
DS18B20Z+T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20

+Denotes a lead-free package. A "\*" will appear on the top mark of lead-free packages.

T&R = Tape and reel.

\*TO-92 packages in tape and reel can be ordered with straight or formed leads. Choose "SL" for straight leads. Bulk TO-92 orders are straight leads only.

**PIN DESCRIPTION**

PIN			NAME	FUNCTION
SO	$\mu$ SOP	TO-92		
1, 2, 6, 7, 8	2, 3, 5, 6, 7	—	N.C.	No Connection
3	8	3	V <sub>DD</sub>	Optional V <sub>DD</sub> . V <sub>DD</sub> must be grounded for operation in parasite power mode.
4	1	2	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 DS18B20</i> section.)
5	4	1	GND	Ground

**OVERVIEW**

Figure 1 shows a block diagram of the DS18B20, 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<sub>H</sub> and T<sub>L</sub>) and the 1-byte configuration register. The configuration register allows the user to set the resolution of the temperature-to-digital conversion to 9, 10, 11, or 12 bits. The T<sub>H</sub>, T<sub>L</sub>, and configuration registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

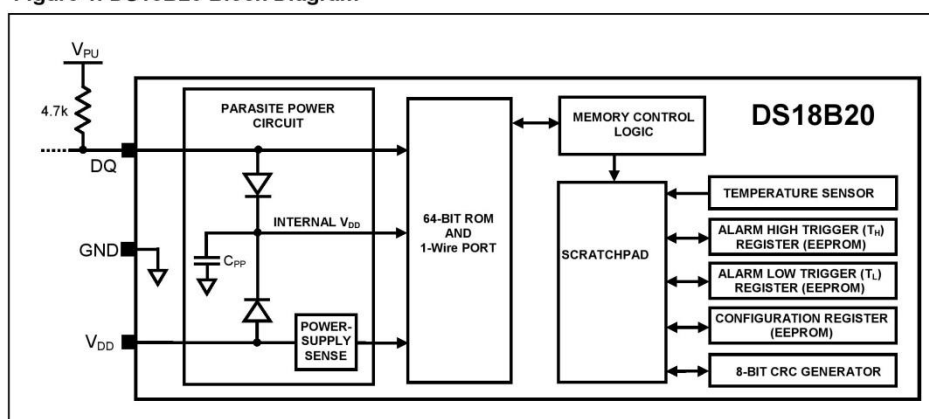
The DS18B20 uses Maxim's exclusive 1-Wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18B20). In this bus system, the 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 DS18B20 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 DS18B20 may also be powered by an external supply on  $V_{DD}$ .

**Figure 1. DS18B20 Block Diagram**



### OPERATION—MEASURING TEMPERATURE

The core functionality of the DS18B20 is its direct-to-digital temperature sensor. The resolution of the temperature sensor is user-configurable to 9, 10, 11, or 12 bits, corresponding to increments of 0.5°C, 0.25°C, 0.125°C, and 0.0625°C, respectively. The default resolution at power-up is 12-bit. The DS18B20 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 DS18B20 returns to its idle state. If the DS18B20 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 DS18B20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. If the DS18B20 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 DS18B20* section.

The DS18B20 output temperature data is calibrated in degrees Celsius; 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 2). The sign bits (S) indicate if the temperature is positive or negative: for positive numbers  $S = 0$  and for negative numbers  $S = 1$ . If the DS18B20 is configured for 12-bit resolution, all bits in the temperature register will contain valid data. For 11-bit resolution, bit 0 is undefined. For 10-bit resolution, bits 1 and 0 are undefined, and for 9-bit resolution, bits 2, 1, and 0 are undefined. Table 1 gives examples of digital output data and the corresponding temperature reading for 12-bit resolution conversions.

**Figure 2. Temperature Register Format**

LS BYTE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$
MS BYTE	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
	S	S	S	S	S	$2^6$	$2^5$	$2^4$

S = SIGN

**Table 1. Temperature/Data Relationship**

TEMPERATURE (°C)	DIGITAL OUTPUT (BINARY)	DIGITAL OUTPUT (HEX)
+125	0000 0111 1101 0000	07D0h
+85*	0000 0101 0101 0000	0550h
+25.0625	0000 0001 1001 0001	0191h
+10.125	0000 0000 1010 0010	00A2h
+0.5	0000 0000 0000 1000	0008h
0	0000 0000 0000 0000	0000h
-0.5	1111 1111 1111 1000	FFF8h
-10.125	1111 1111 0101 1110	FF5Eh
-25.0625	1111 1110 0110 1111	FE6Fh
-55	1111 1100 1001 0000	FC90h

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

### OPERATION—ALARM SIGNALING

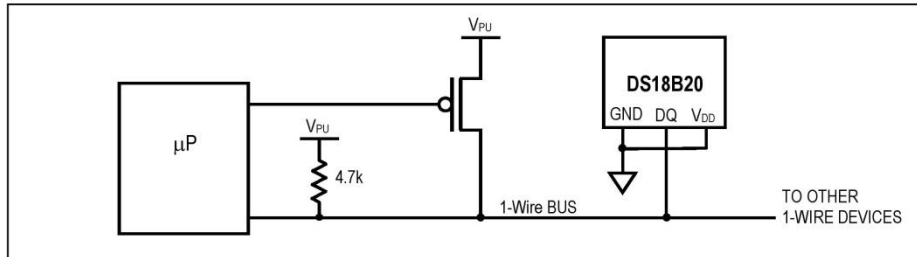
After the DS18B20 performs a temperature conversion, the temperature value is compared to the user-defined two's complement alarm trigger values stored in the 1-byte  $T_H$  and  $T_L$  registers (see Figure 3). The sign bit (S) indicates if the value is positive or negative: for positive numbers  $S = 0$  and for negative numbers  $S = 1$ . The  $T_H$  and  $T_L$  registers are nonvolatile (EEPROM) so they will retain data when the device is powered down.  $T_H$  and  $T_L$  can be accessed through bytes 2 and 3 of the scratchpad as explained in the *Memory* section.

**Figure 3.  $T_H$  and  $T_L$  Register Format**

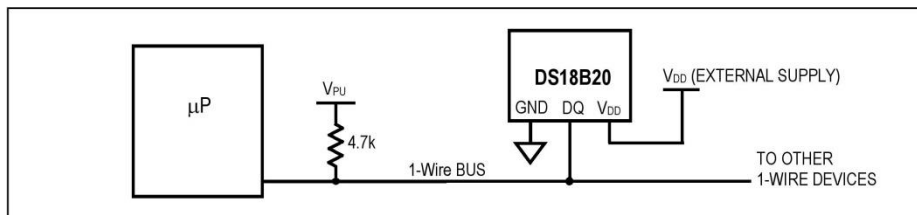
BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
S	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

Only bits 11 through 4 of the temperature register are used in the  $T_H$  and  $T_L$  comparison since  $T_H$  and  $T_L$  are 8-bit registers. If the measured temperature is lower than or equal to  $T_L$  or higher than or equal to  $T_H$ , an alarm condition exists and an alarm flag is set inside the DS18B20. 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.

**Figure 4. Supplying the Parasite-Powered DS18B20 During Temperature Conversions**



**Figure 5. Powering the DS18B20 with an External Supply**



### 64-BIT LASERED ROM CODE

Each DS18B20 contains a unique 64-bit code (see Figure 6) stored in ROM. The least significant 8 bits of the ROM code contain the DS18B20's 1-Wire family code: 28h. 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 DS18B20 to operate as a 1-Wire device using the protocol detailed in the *1-Wire Bus System* section.

**Figure 6. 64-Bit Lasered ROM Code**

8-BIT CRC		48-BIT SERIAL NUMBER				8-BIT FAMILY CODE (28h)	
MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB

## MEMORY

The DS18B20's memory is organized as shown in Figure 7. The memory consists of an SRAM scratchpad with nonvolatile EEPROM storage for the high and low alarm trigger registers ( $T_H$  and  $T_L$ ) and configuration register. Note that if the DS18B20 alarm function is not used, the  $T_H$  and  $T_L$  registers can serve as general-purpose memory. All memory commands are described in detail in the *DS18B20 Function Commands* section.

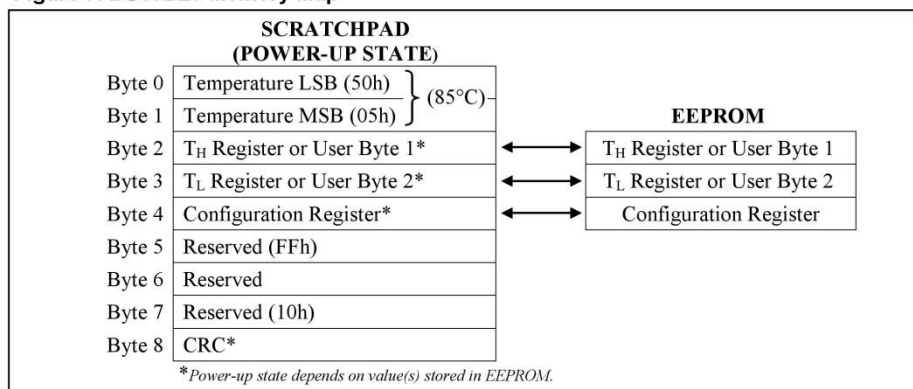
Byte 0 and byte 1 of the scratchpad contain the LSB and the MSB of the temperature register, respectively. These bytes are read-only. Bytes 2 and 3 provide access to  $T_H$  and  $T_L$  registers. Byte 4 contains the configuration register data, which is explained in detail in the *Configuration Register* section. Bytes 5, 6, and 7 are reserved for internal use by the device and cannot be overwritten.

Byte 8 of the scratchpad is read-only and contains the CRC code for bytes 0 through 7 of the scratchpad. The DS18B20 generates this CRC using the method described in the *CRC Generation* section.

Data is written to bytes 2, 3, and 4 of the scratchpad using the Write Scratchpad [4Eh] command; the data must be transmitted to the DS18B20 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  $T_H$ ,  $T_L$  and configuration data from the scratchpad to EEPROM, the master must issue the Copy Scratchpad [48h] command.

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  $E^2$  [B8h] command. The master can issue read time slots following the Recall  $E^2$  command and the DS18B20 will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done.

**Figure 7. DS18B20 Memory Map**



## CONFIGURATION REGISTER

Byte 4 of the scratchpad memory contains the configuration register, which is organized as illustrated in Figure 8. The user can set the conversion resolution of the DS18B20 using the R0 and R1 bits in this register as shown in Table 2. The power-up default of these bits is R0 = 1 and R1 = 1 (12-bit resolution). Note that there is a direct tradeoff between resolution and conversion time. Bit 7 and bits 0 to 4 in the configuration register are reserved for internal use by the device and cannot be overwritten.

**Figure 8. Configuration Register**

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0	R1	R0	1	1	1	1	1

**Table 2. Thermometer Resolution Configuration**

R1	R0	RESOLUTION (BITS)	MAX CONVERSION TIME	
0	0	9	93.75ms	( $t_{CONV}/8$ )
0	1	10	187.5ms	( $t_{CONV}/4$ )
1	0	11	375ms	( $t_{CONV}/2$ )
1	1	12	750ms	( $t_{CONV}$ )

## CRC GENERATION

CRC bytes are provided as part of the DS18B20's 64-bit ROM code and in the 9<sup>th</sup> 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 DS18B20. 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 DS18B20 that prevents a command sequence from proceeding if the DS18B20 CRC (ROM or scratchpad) does not match the value generated by the bus master.

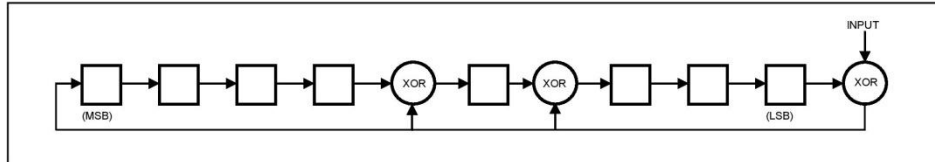
The equivalent polynomial function of the CRC (ROM or scratchpad) is:

$$CRC = X^8 + X^5 + X^4 + 1$$

The bus master can re-calculate the CRC and compare it to the CRC values from the DS18B20 using the polynomial generator shown in Figure 9. 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 DS18B20 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*.

**Figure 9. CRC Generator**



### 1-WIRE BUS SYSTEM

The 1-Wire bus system uses a single bus master to control one or more slave devices. The DS18B20 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.

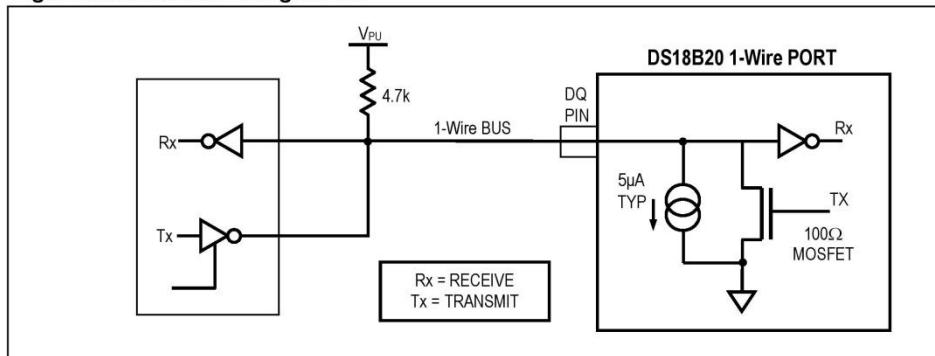
The following discussion of the 1-Wire bus system is broken down into three topics: hardware configuration, transaction sequence, and 1-Wire signaling (signal types and timing).

### 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 DS18B20 (the DQ pin) is open drain with an internal circuit equivalent to that shown in Figure 10.

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.

**Figure 10. Hardware Configuration**



**COPY SCRATCHPAD [48h]**

This command copies the contents of the scratchpad  $T_H$ ,  $T_L$  and configuration registers (bytes 2, 3 and 4) 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 DS18B20* section.

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

This command recalls the alarm trigger values ( $T_H$  and  $T_L$ ) and configuration data from EEPROM and places the data in bytes 2, 3, and 4, respectively, in the scratchpad memory. The master device can issue read time slots following the Recall E<sup>2</sup> command and the DS18B20 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 DS18B20s on the bus are using parasite power. During the read time slot, parasite powered DS18B20s will pull the bus low, and externally powered DS18B20s will let the bus remain high. See the *Powering the DS18B20* section for usage information for this command.

**Table 3. DS18B20 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	DS18B20 transmits conversion status to master (not applicable for parasite-powered DS18B20s).	1
<b>MEMORY COMMANDS</b>				
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	BEh	DS18B20 transmits up to 9 data bytes to master.	2
Write Scratchpad	Writes data into scratchpad bytes 2, 3, and 4 ( $T_H$ , $T_L$ , and configuration registers).	4Eh	Master transmits 3 data bytes to DS18B20.	3
Copy Scratchpad	Copies $T_H$ , $T_L$ , and configuration register data from the scratchpad to EEPROM.	48h	None	1
Recall E <sup>2</sup>	Recalls $T_H$ , $T_L$ , and configuration register data from EEPROM to the scratchpad.	B8h	DS18B20 transmits recall status to master.	
Read Power Supply	Signals DS18B20 power supply mode to the master.	B4h	DS18B20 transmits supply status to master.	

**Note 1:** For parasite-powered DS18B20s, 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.

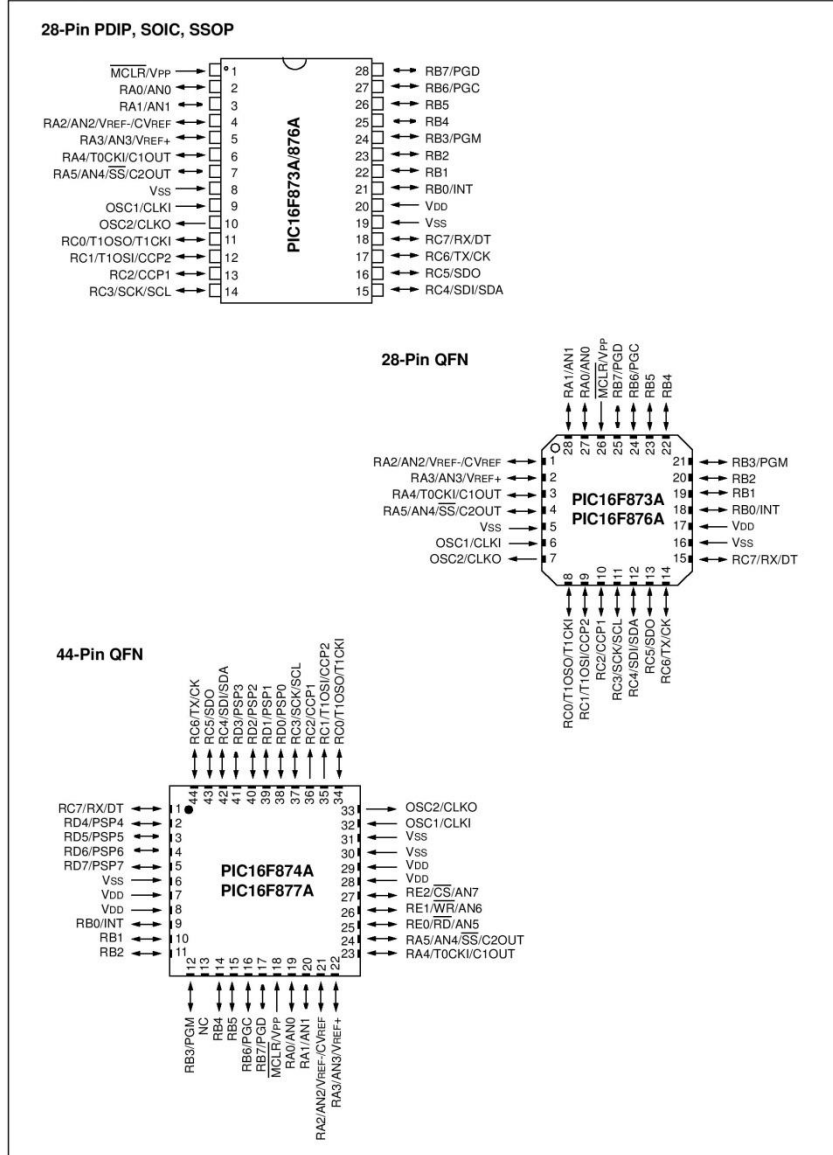
**Note 2:** The master can interrupt the transmission of data at any time by issuing a reset.

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

## Anexo 5.- Microprocesador 16f877A

### PIC16F87XA

#### Pin Diagrams







# PIC16F87XA

## 1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40-pin and 44-pin packages. All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight
- The Parallel Slave Port is implemented only on the 40/44-pin devices

The available features are summarized in Table 1-1. Block diagrams of the PIC16F873A/876A and PIC16F874A/877A devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

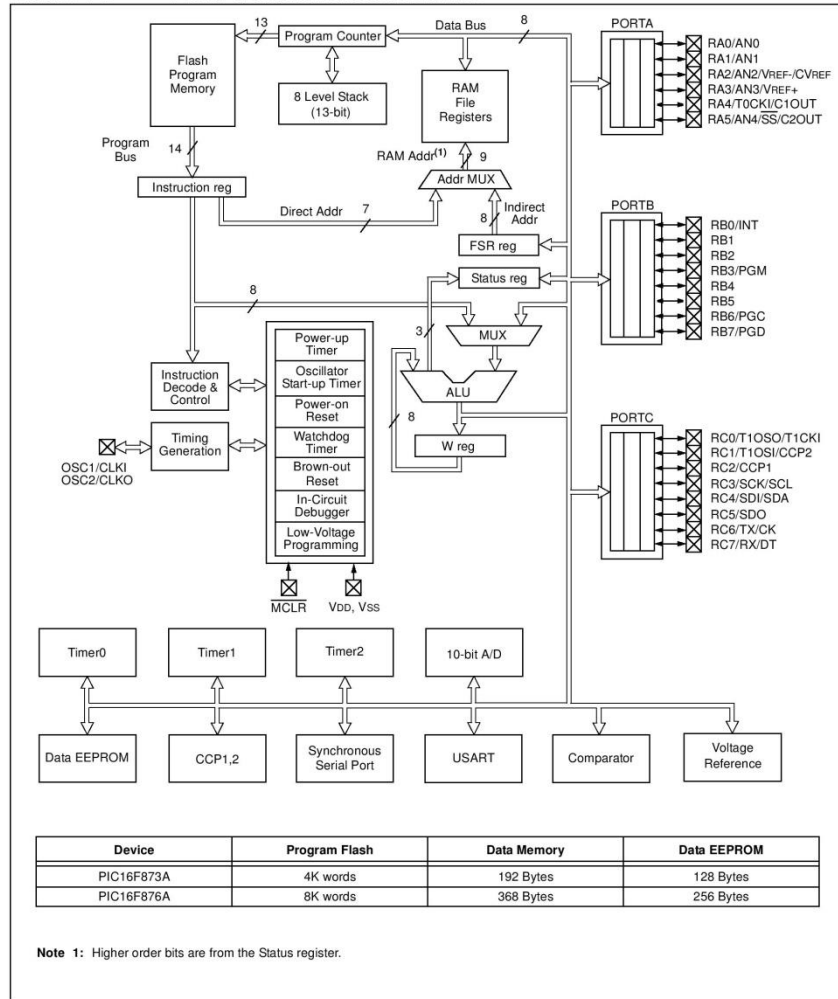
Additional information may be found in the PICmicro® Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

**TABLE 1-1: PIC16F87XA DEVICE FEATURES**

Key Features	PIC16F873A	PIC16F874A	PIC16F876A	PIC16F877A
Operating Frequency	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Flash Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory (bytes)	128	128	256	256
Interrupts	14	15	14	15
I/O Ports	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C	Ports A, B, C, D, E
Timers	3	3	3	3
Capture/Compare/PWM modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Analog Comparators	2	2	2	2
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN

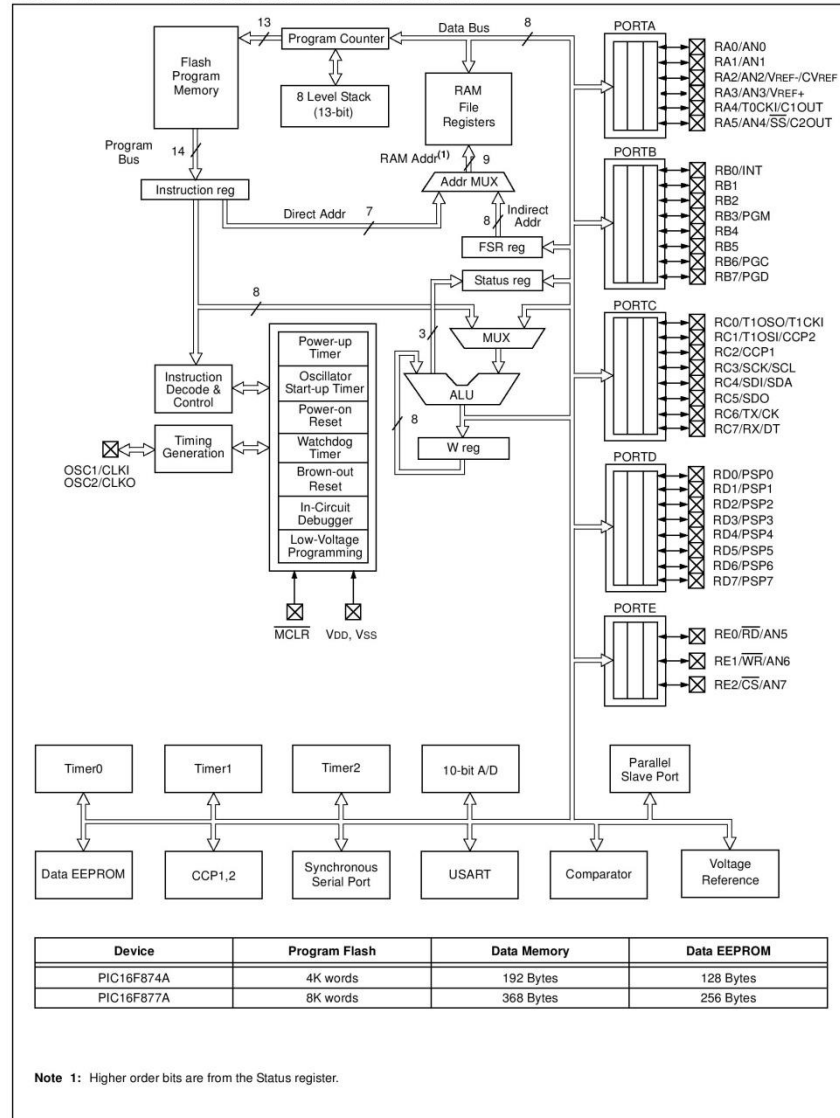
# PIC16F87XA

FIGURE 1-1: PIC16F873A/876A BLOCK DIAGRAM



# PIC16F87XA

FIGURE 1-2: PIC16F874A/877A BLOCK DIAGRAM



# PIC16F87XA

## 2.5 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF register itself, indirectly (FSR = 0) will read 00h. Writing to the INDF register indirectly results in a no operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (Status<7>) as shown in Figure 2-6.

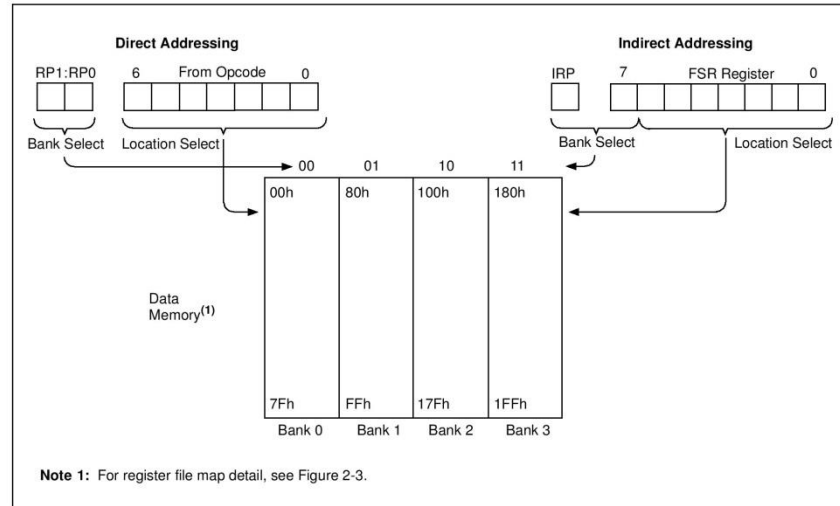
A simple program to clear RAM locations 20h-2Fh using indirect addressing is shown in Example 2-2.

### EXAMPLE 2-2: INDIRECT ADDRESSING

```

MOV LW 0x20 ;initialize pointer
MOV WF FSR ;to RAM
NEXT   CLRF INDF ;clear INDF register
       INC FSR,F ;inc pointer
       BTFS FSR,4 ;all done?
       GOTO NEXT ;no clear next
CONTINUE
       : ;yes continue
    
```

FIGURE 2-6: DIRECT/INDIRECT ADDRESSING

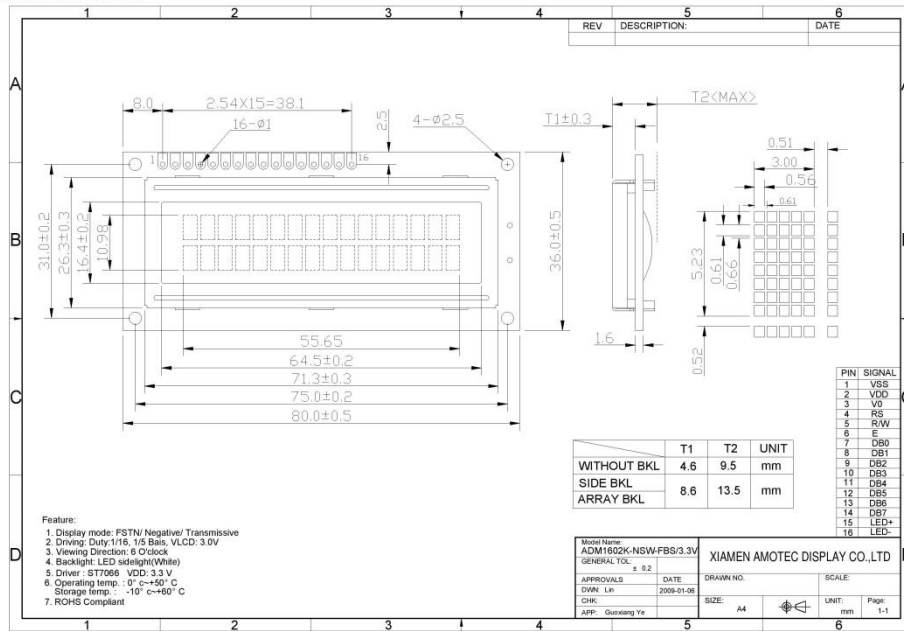






## Anexo 6.- LCD

### 3. Outline dimension

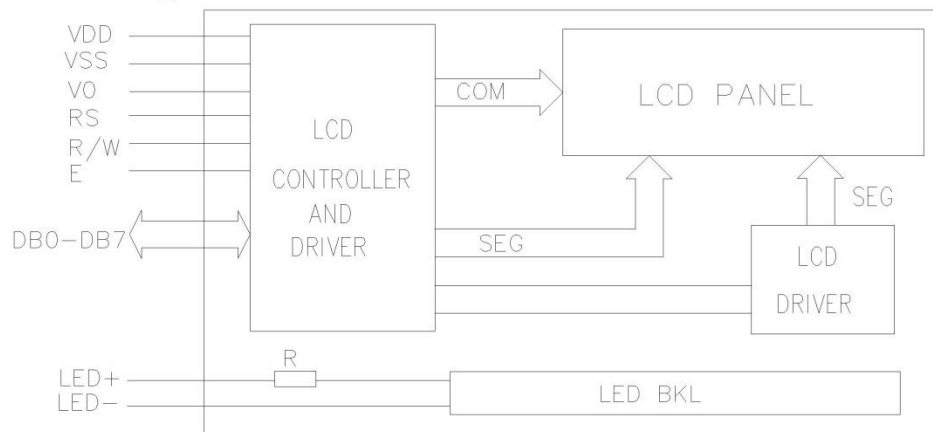




#### 4. Absolute maximum ratings

Item	Symbol	Standard			Unit
Power voltage	$V_{DD}-V_{SS}$	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	

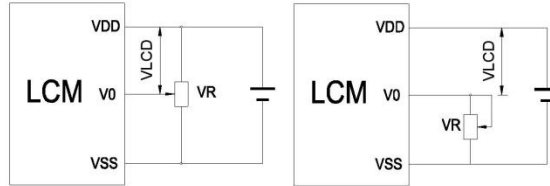
#### 5. Block diagram



#### 6. Interface pin description

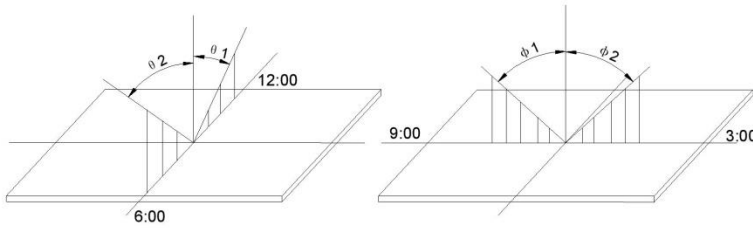
Pin no.	Symbol	External connection	Function
1	V <sub>SS</sub>	Power supply	Signal ground for LCM
2	V <sub>DD</sub>		Power supply for logic for LCM
3	V <sub>0</sub>		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7~10	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

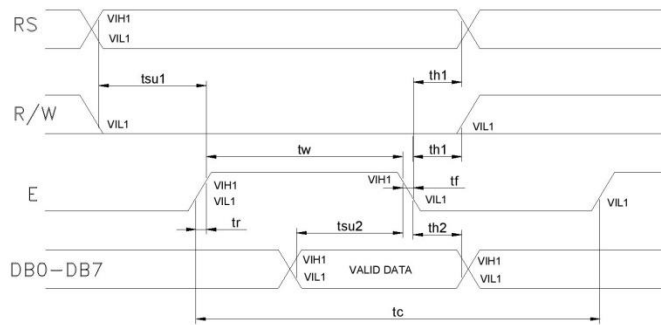
Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Supply voltage for LCD	$V_{DD}-V_0$	$T_a = 25^\circ\text{C}$	-	3.0	-	V
Input voltage	$V_{DD}$		3.1	3.3	3.5	
Supply current	$I_{DD}$	$T_a=25^\circ\text{C}$ , $V_{DD}=3.3\text{V}$	-	1.5	2.5	mA
Input leakage current	$I_{LKG}$		-	-	1.0	$\mu\text{A}$
"H" level 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 (Ta=25°C, VDD=3.3V)

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	-	-	

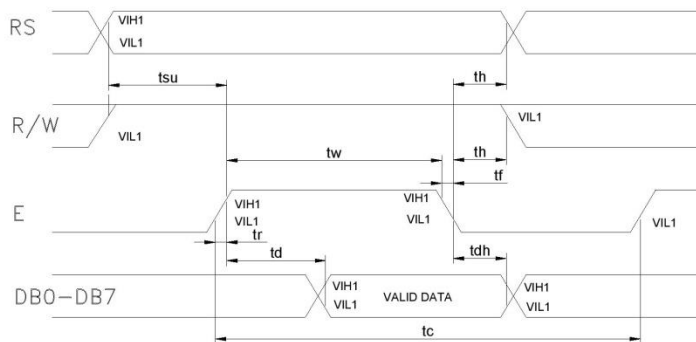
Write mode timing diagram



Read cycle (Ta=25°C, VDD=3.3V)

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	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	0				1	0	0	1	
						0	0	1				0	1	0				0	1	0	0	1
						0	0	1				0	1	1				0	1	1	0	0
						0	0	1				1	0	0				0	1	0	0	0
						0	0	1				1	0	0				0	1	0	0	0
						0	0	1				1	0	0				0	1	0	0	0
						0	0	1				1	0	0				0	1	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

11.9 Instruction Table

Instruction	Instruction code										Description	Execution time (fosc=270 KHZ)
	RS	R/M	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRA and set DDRAM address to "00H" from AC	1.53ms
Return Home	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".

11.3Contents

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 & 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

## 12. Standard character pattern

Upper 4bit Lower 4bit	LLLL	LLLH	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HLHL	HLHH	HHLH	HHHL	HHHH			
LLLL	CG RAM (1)			0	A	P	\	P					—	9	3	0	P	
LLLH	(2)	!	1	A	Q	a	a						。	7	7	△	△	9
LLHL	(3)	"	2	B	R	b	r						「	イ	ツ	×	β	θ
LLHH	(4)	#	3	C	S	c	s						」	ウ	テ	ε	ε	∞
LHLL	(5)	\$	4	D	T	d	t						√	エ	ト	ト	μ	σ
LHLH	(6)	%	5	E	U	e	u						。	オ	ナ	1	ε	U
LHHL	(7)	&	6	F	V	f	v						ヲ	カ	ニ	ヨ	ρ	Σ
LHHH	(8)	'	7	G	W	g	w						ア	チ	ア	ラ	9	π
HLLL	(1)	(	8	H	X	h	x						イ	ウ	カ	リ	リ	又
HLLH	(2)	)	9	I	Y	i	y						オ	ケ	ル	ル	リ	ウ
HLHL	(3)	*	:	J	Z	j	z						エ	コ	ン	ル	リ	チ
HLHH	(4)	+	;	K	L	k	l						オ	ケ	レ	ロ	リ	チ
HHLH	(5)	,	<	L	¥	l	l						オ	ケ	レ	ロ	リ	チ
HHLH	(6)	—	=	M	O	m	o						ユ	ズ	レ	ル	ル	チ
HHHL	(7)	.	>	N	^	n	^						ヨ	セ	レ	リ	リ	チ
HHHH	(8)	/	?	O	_	o	_						ウ	リ	マ	マ	マ	チ



## Anexo 7.- LM7805

MC78XX/LM78XX/MC78XXA

### Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Input Voltage (for $V_O = 5V$ to $18V$ ) (for $V_O = 24V$ )	$V_I$	35	V
	$V_I$	40	V
Thermal Resistance Junction-Cases (TO-220)	$R_{\theta JC}$	5	$^{\circ}C/W$
Thermal Resistance Junction-Air (TO-220)	$R_{\theta JA}$	65	$^{\circ}C/W$
Operating Temperature Range	$T_{OPR}$	0 ~ +125	$^{\circ}C$
Storage Temperature Range	$T_{STG}$	-65 ~ +150	$^{\circ}C$

### Electrical Characteristics (MC7805/LM7805)

(Refer to test circuit,  $0^{\circ}C < T_J < 125^{\circ}C$ ,  $I_O = 500mA$ ,  $V_I = 10V$ ,  $C_I = 0.33\mu F$ ,  $C_O = 0.1\mu F$ , unless otherwise specified)

Parameter	Symbol	Conditions	MC7805/LM7805			Unit	
			Min.	Typ.	Max.		
Output Voltage	$V_O$	$T_J = +25^{\circ}C$	4.8	5.0	5.2	V	
		$5.0mA \leq I_O \leq 1.0A$ , $P_O \leq 15W$ $V_I = 7V$ to $20V$	4.75	5.0	5.25		
Line Regulation (Note1)	Regline	$T_J = +25^{\circ}C$	$V_O = 7V$ to $25V$	-	4.0	100	mV
			$V_I = 8V$ to $12V$	-	1.6	50	
Load Regulation (Note1)	Regload	$T_J = +25^{\circ}C$	$I_O = 5.0mA$ to $1.5A$	-	9	100	mV
			$I_O = 250mA$ to $750mA$	-	4	50	
Quiescent Current	$I_Q$	$T_J = +25^{\circ}C$	-	5.0	8.0	mA	
Quiescent Current Change	$\Delta I_Q$	$I_O = 5mA$ to $1.0A$	-	0.03	0.5	mA	
		$V_I = 7V$ to $25V$	-	0.3	1.3		
Output Voltage Drift	$\Delta V_O / \Delta T$	$I_O = 5mA$	-	-0.8	-	mV/ $^{\circ}C$	
Output Noise Voltage	$V_N$	$f = 10Hz$ to $100KHz$ , $T_A = +25^{\circ}C$	-	42	-	$\mu V/V_O$	
Ripple Rejection	RR	$f = 120Hz$ $V_O = 8V$ to $18V$	62	73	-	dB	
Dropout Voltage	$V_{Drop}$	$I_O = 1A$ , $T_J = +25^{\circ}C$	-	2	-	V	
Output Resistance	$r_O$	$f = 1KHz$	-	15	-	$m\Omega$	
Short Circuit Current	$I_{SC}$	$V_I = 35V$ , $T_A = +25^{\circ}C$	-	230	-	mA	
Peak Current	$I_{PK}$	$T_J = +25^{\circ}C$	-	2.2	-	A	

**Note:**

1. Load and line regulation are specified at constant junction temperature. Changes in  $V_O$  due to heating effects must be taken into account separately. Pulse testing with low duty is used.

Typical Performance Characteristics

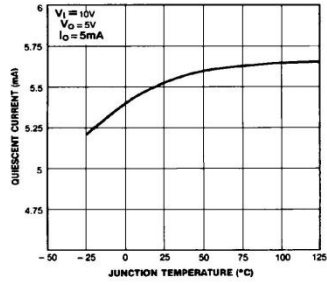


Figure 1. Quiescent Current

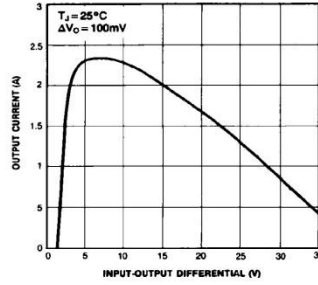


Figure 2. Peak Output Current

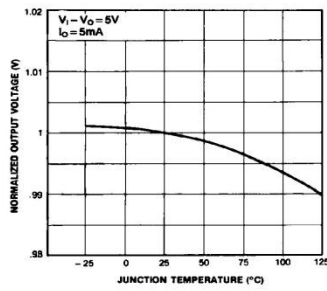


Figure 3. Output Voltage

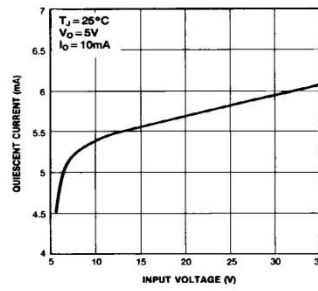


Figure 4. Quiescent Current

## Typical Applications

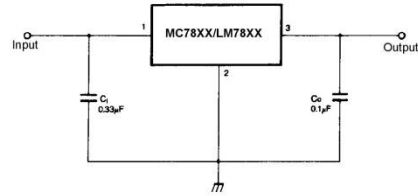


Figure 5. DC Parameters

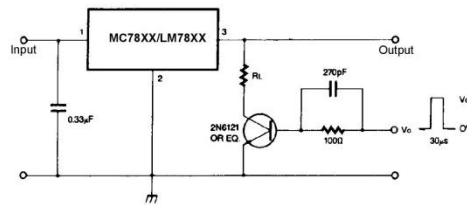


Figure 6. Load Regulation

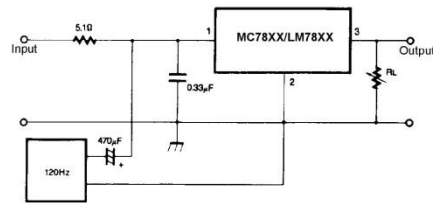


Figure 7. Ripple Rejection

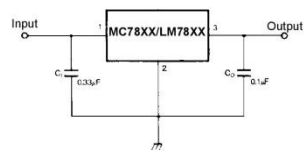


Figure 8. Fixed Output Regulator

## Anexo 8. - 74LS08

**FAIRCHILD**  
SEMICONDUCTOR™

August 1986  
Revised March 2000

### DM74LS08 Quad 2-Input AND Gates

#### General Description

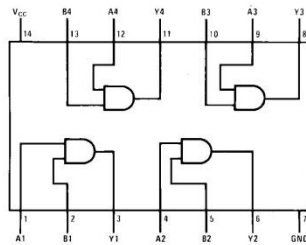
This device contains four independent gates each of which performs the logic AND function.

#### Ordering Code:

Order Number	Package Number	Package Description
DM74LS08M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS08SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS08N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter 'X' to the ordering code.

#### Connection Diagram



#### Function Table

$$Y = AB$$

Inputs		Output
A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

H = HIGH Logic Level  
L = LOW Logic Level

DM74LS08 Quad 2-Input AND Gates

**Absolute Maximum Ratings**(Note 1)

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	0°C to +70°C
Storage Temperature Range	-65°C to +150°C

**Note 1:** The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the Electrical Characteristics tables are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

**Recommended Operating Conditions**

Symbol	Parameter	Min	Nom	Max	Units
V <sub>CC</sub>	Supply Voltage	4.75	5	5.25	V
V <sub>IH</sub>	HIGH Level Input Voltage	2			V
V <sub>IL</sub>	LOW Level Input Voltage			0.8	V
I <sub>OH</sub>	HIGH Level Output Current			-0.4	mA
I <sub>OL</sub>	LOW Level Output Current			8	mA
T <sub>A</sub>	Free Air Operating Temperature	0		70	°C

**Electrical Characteristics**

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
V <sub>I</sub>	Input Clamp Voltage	V <sub>CC</sub> = Min, I <sub>I</sub> = -18 mA			-1.5	V
V <sub>OH</sub>	HIGH Level Output Voltage	V <sub>CC</sub> = Min, I <sub>OH</sub> = Max, V <sub>IH</sub> = Min	2.7	3.4		V
V <sub>OL</sub>	LOW Level Output Voltage	V <sub>CC</sub> = Min, I <sub>OL</sub> = Max, V <sub>IL</sub> = Max		0.35	0.5	V
		I <sub>OL</sub> = 4 mA, V <sub>CC</sub> = Min		0.25	0.4	
I <sub>I</sub>	Input Current @ Max Input Voltage	V <sub>CC</sub> = Max, V <sub>I</sub> = 7V			0.1	mA
I <sub>IH</sub>	HIGH Level Input Current	V <sub>CC</sub> = Max, V <sub>I</sub> = 2.7V			20	μA
I <sub>IL</sub>	LOW Level Input Current	V <sub>CC</sub> = Max, V <sub>I</sub> = 0.4V			-0.36	mA
I <sub>OS</sub>	Short Circuit Output Current	V <sub>CC</sub> = Max (Note 3)	-20		-100	mA
I <sub>CC</sub>	Supply Current with Outputs HIGH	V <sub>CC</sub> = Max		2.4	4.8	mA
I <sub>CC</sub>	Supply Current with Outputs LOW	V <sub>CC</sub> = Max		4.4	8.8	mA

**Switching Characteristics**

at V<sub>CC</sub> = 5V and T<sub>A</sub> = 25°C

Symbol	Parameter	R <sub>L</sub> = 2 kΩ				Units
		C <sub>L</sub> = 15 pF		C <sub>L</sub> = 50 pF		
		Min	Max	Min	Max	
t <sub>PLH</sub>	Propagation Delay Time LOW-to-HIGH Level Output	4	13	6	18	ns
t <sub>PHL</sub>	Propagation Delay Time HIGH-to-LOW Level Output	3	11	5	18	ns

**Note 2:** All typicals are at V<sub>CC</sub> = 5V, T<sub>A</sub> = 25°C.

**Note 3:** Not more than one output should be shorted at a time, and the duration should not exceed one second.

## Anexo 9.- XBee

### **XBee®/XBee-PRO® ZB RF Modules**

---

#### **ZigBee RF Modules by Digi International**

Models: XBEE2, XBEEPRO2, PRO S2B

Hardware: S2 and S2B

Firmware Versions:

- 20xx - Coordinator - AT/Transparent Operation
- 21xx - Coordinator - API Operation
- 22xx - Router - AT/Transparent Operation
- 23xx - Router - API Operation
- 28xx - End Device - AT/Transparent Operation
- 29xx - End Device - API Operation



Digi International Inc.  
11001 Bren Road East  
Minnetonka, MN 55343  
877 912-3444 or 952 912-3444  
<http://www.digi.com>

90000976\_K  
March 2012

## Specifications

Specifications of the XBee®/XBee-PRO® ZB RF Module

Specification	XBee	XBee-PRO (S2)	XBee-PRO (S2B)
<b>Performance</b>			
Indoor/Urban Range	up to 133 ft. (40 m)	Up to 300 ft. (90 m), up to 200 ft (60 m) international variant	Up to 300 ft. (90 m), up to 200 ft (60 m) international variant.
Outdoor RF line-of-sight Range	up to 400 ft. (120 m)	Up to 2 miles (3200 m), up to 5000 ft (1500 m) international variant	Up to 2 miles (3200 m), up to 5000 ft (1500 m) international variant
Transmit Power Output	2mW (+3dBm), boost mode enabled 1.25mW (+1dBm), boost mode disabled	50mW (+17 dBm) 10mW (+10 dBm) for international variant	63mW (+18 dBm) 10mW (+10 dBm) for international variant
RF Data Rate	250,000 bps	250,000 bps	250,000 bps
Data Throughput	up to 35000 bps (see chapter 4)	up to 35000 bps (see chapter 4)	up to 35000 bps (see chapter 4)
Serial Interface Data Rate (software selectable)	1200 bps - 1 Mbps (non-standard baud rates also supported)	1200 bps - 1 Mbps (non-standard baud rates also supported)	1200 bps - 1 Mbps (non-standard baud rates also supported)
Receiver Sensitivity	-96 dBm, boost mode enabled -95 dBm, boost mode disabled	-102 dBm	-102 dBm
<b>Power Requirements</b>			
Supply Voltage	2.1 - 3.6 V	3.0 - 3.4 V	2.7 - 3.6 V
Operating Current (Transmit, max output power)	40mA (@ 3.3 V, boost mode enabled) 35mA (@ 3.3 V, boost mode disabled)	295mA (@3.3 V) 170mA (@3.3 V) international variant	205mA, up to 220 mA with programmable variant (@3.3 V) 117mA, up to 132 mA with programmable variant (@3.3 V), International variant
Operating Current (Receive)	40mA (@ 3.3 V, boost mode enabled) 38mA (@ 3.3 V, boost mode disabled)	45 mA (@3.3 V)	47 mA, up to 62 mA with programmable variant (@3.3 V)
Idle Current (Receiver off)	15mA	15mA	15mA
Power-down Current	< 1 $\mu$ A @ 25°C	3.5 $\mu$ A typical @ 25°C	3.5 $\mu$ A typical @ 25°C
<b>General</b>			
Operating Frequency Band	ISM 2.4 GHz	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960 x 1.297 (2.438cm x 3.294cm)	0.960 x 1.297 (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip Antenna, Embedded PCB Antenna, RPSMA, or U.FL Connector	Integrated Whip Antenna, Embedded PCB Antenna, RPSMA or U.FL Connector	Integrated Whip Antenna, Embedded PCB Antenna, RPSMA or U.FL Connector
<b>Networking &amp; Security</b>			
Supported Network Topologies	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh
Number of Channels	16 Direct Sequence Channels	14 Direct Sequence Channels	15 Direct Sequence Channels
Channels	11 to 26	11 to 24	11 to 25
Addressing Options	PAN ID and Addresses, Cluster IDs and Endpoints (optional)	PAN ID and Addresses, Cluster IDs and Endpoints (optional)	PAN ID and Addresses, Cluster IDs and Endpoints (optional)
<b>Agency Approvals</b>			
United States (FCC Part 15.247)	FCC ID: OUR-XBEE2	FCC ID: MCQ-XBEEPRO2	FCC ID: MCQ-PROS2B
Industry Canada (IC)	IC: 4214A-XBEE2	IC: 1846A-XBEEPRO2	IC: 1846A-PROS2B
Europe (CE)	ETSI	ETSI (International variant)	ETSI (10 mW max)

Specifications of the XBee®/XBee-PRO® ZB RF Module

Specification	XBee	XBee-PRO (S2)	XBee-PRO (S2B)
Australia	C-Tick	C-Tick	C-Tick
Japan	R201WW07215215 Wire, chip, RPSMA, and U.FL versions are certified for Japan. The PCB antenna version is not.	R201WW08215142 (international variant) Wire, chip, RPSMA, and U.FL versions are certified for Japan. PCB antenna version is not.	R201WW10215062 (international variant)
RoHS	Compliant	Compliant	Compliant

### Hardware Specs for Programmable Variant

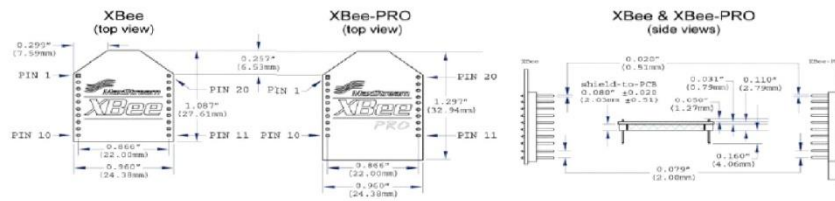
The following specifications need to be added to the current measurement of the previous table if the module has the programmable secondary processor. For example, if the secondary processor is running and constantly collecting DIO samples at a rate while having the RF portion of the XBEE sleeping the new current will be  $I_{total} = I_{r2} + I_0$ , where  $I_{r2}$  is the runtime current of the secondary processor and  $I_s$  is the sleep current of the RF portion of the module of the XBEE-PRO (S2B) listed in the table below.

Specifications of the programmable secondary processor

Optional Secondary Processor Specification	These numbers add to S2B specifications (Add to RX, TX, and sleep currents depending on mode of operation)
Runtime current for 32k running at 20MHz	+14mA
Runtime current for 32k running at 1MHz	+1mA
Sleep current	+0.5uA typical
For additional specifications see Freescale Datasheet and Manual	MC9S08QE32
Minimum Reset low pulse time for EM250	+50 nS (additional resistor increases minimum time)
VREF Range	1.8VDC to VCC

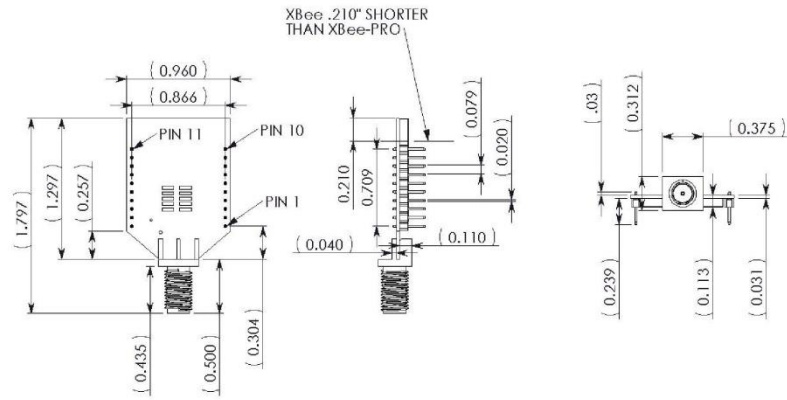
### Mechanical Drawings

Mechanical drawings of the XBee®/XBee-PRO® ZB RF Modules (antenna options not shown)



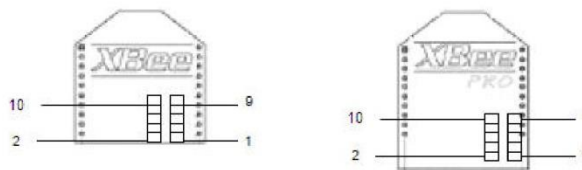


Mechanical Drawings for the RPSMA Variant



SIF Header Interface

The XBee/XBee-PRO ZB modules include a SIF programming header that can be used with Ember's programming tools to upload custom firmware images onto the XBee module. The SIF header orientation and pinout are shown below.



Pin Number	Pin Name
1	VBRD
2	SIF-MISO
3	Ground
4	SIF-MOSI
5	Ground
6	SIF-CLOCK
7	SIF-LOAD
8	RESET
9	PTI-EN
10	PTI-DATA

This figure shows the orientation of the insight port header .

A male header can be populated on the XBee that mates with Ember's 2x5 ribbon cable. The male header and ribbon cables are available from Samtec:

2x5 Male Header - FTSH-105-01-F-DV-K

2x5 Ribbon Cable - FFSD-05-D-12.00-01-N

## Pin Signals

### Pin Assignments for the XBee/XBee-PRO Modules

(Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Default State	Description
1	VCC	-	-	Power supply
2	DOUT	Output	Output	UART Data Out
3	DIN / <u>CONFIG</u>	Input	Input	UART Data In
4	DIO12	Both	Disabled	Digital I/O 12
5	<u>RESET</u>	Both	Open-Collector with pull-up	Module Reset (reset pulse must be at least 200 ns)
6	RSSI PWM / DIO10	Both	Output	RX Signal Strength Indicator / Digital IO
7	DIO11	Both	Input	Digital I/O 11
8	[reserved]	-	Disabled	Do not connect
9	<u>DTR</u> / SLEEP_RQ/ DIO8	Both	Input	Pin Sleep Control Line or Digital IO 8
10	GND	-	-	Ground
11	DIO4	Both	Disabled	Digital I/O 4
12	<u>CTS</u> / DIO7	Both	Output	Clear-to-Send Flow Control or Digital I/O 7. CTS, if enabled, is an output.
13	ON / <u>SLEEP</u>	Output	Output	Module Status Indicator or Digital I/O 9
14	VREF	Input	-	Not used for EM250. Used for programmable secondary processor. For compatibility with other XBEE modules, we recommend connecting this pin voltage reference if Analog sampling is desired. Otherwise, connect to GND.
15	Associate / DIO5	Both	Output	Associated Indicator, Digital I/O 5
16	<u>RTS</u> / DIO6	Both	Input	Request-to-Send Flow Control, Digital I/O 6. RTS, if enabled, is an input.
17	AD3 / DIO3	Both	Disabled	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Both	Disabled	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Both	Disabled	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0 / Commissioning Button	Both	Disabled	Analog Input 0, Digital IO 0, or Commissioning Button

- Signal Direction is specified with respect to the module
- See Design Notes section below for details on pin connections.

## Electrical Characteristics

DC Characteristics of the XBee/XBee-PRO

Symbol	Parameter	Condition	Min	Typical	Max	Units
$V_{IL}$	Input Low Voltage	All Digital Inputs	-	-	$0.2 * V_{CC}$	V
$V_{IH}$	Input High Voltage	All Digital Inputs	$0.8 * V_{CC}$	-	-	V
$V_{OL}$	Output Low Voltage	$V_{CC} \geq 2.7 V$	-	-	$0.18 * V_{CC}$	V
$V_{OH}$	Output High Voltage	$V_{CC} \geq 2.7 V$	$0.82 * V_{CC}$	-	-	V
$I_{IN}$	Input Leakage Current	$V_{IN} = V_{CC}$ or GND, all inputs, per pin	-	-	0.5uA	uA
$I_{OHS}$	Output source current (standard)	All digital outputs except RSSI/PWM, DIO10, DIO4			4	mA
$I_{OHH}$	Output source current (high current)	RSSI/PWM, DIO10, DIO4 digital outputs			8	mA
$I_{OLS}$	Output sink current (standard)	All digital inputs except RSSI/PWM, DIO10, DIO4			4	mA
$I_{OLH}$	Output sink current (high current)	RSSI/PWM, DIO10, DIO4 digital outputs			8	mA
$I_{OH} + I_{OL}$	Total output current for all I/O pins	All digital outputs			40	mA
$V_{REFI}$	VREF Internal	EM250 has an internal reference that is fixed	1.19	1.2	1.21	V
$V_{ADC}$	ADC input voltage range		0		$V_{REFI}$	V
$R_{IS}$	Input impedance	When taking a sample	1			M Ohm
$R_I$	Input impedance	When not taking a sample	10			M Ohm

Note – The signal-ended ADC measurements are limited in their range and only guaranteed for accuracy in the range 0 to  $V_{REFI}$ . The nature of the ADC's internal design allows for measurements outside of this range (+/- 200mV), but the accuracy of such measurements are not guaranteed.

## Module Operation for Programmable Variant

The S2B modules that have the programmable option populated have a secondary processor with 32k of flash and 2k of RAM. This allows module integrators to put custom code on the XBEE module to fit their own unique needs. The DIN, DOUT, RTS, CTS, and RESET lines are intercepted by the secondary processor to allow it to be in control of the data transmitted and received. All other lines are in parallel and can be controlled by either the EM250 or the MC9S08QE micro (see Block Diagram for details). The EM250 by default has control of certain lines. These lines can be released by the EM250 by sending the proper command(s) to disable the desired DIO line(s) (see XBEE Command Reference Tables).

In order for the secondary processor to sample with ADCs, the XBEE pin 14 ( $V_{REF}$ ) needs to be connected to a reference voltage.

Digi provides a bootloader that can take care of programming the processor over the air or through the serial interface. This means that over the air updates can be supported through an XMODEM protocol. The processor can also be programmed and debugged through a one wire interface BKGD (Pin 8).



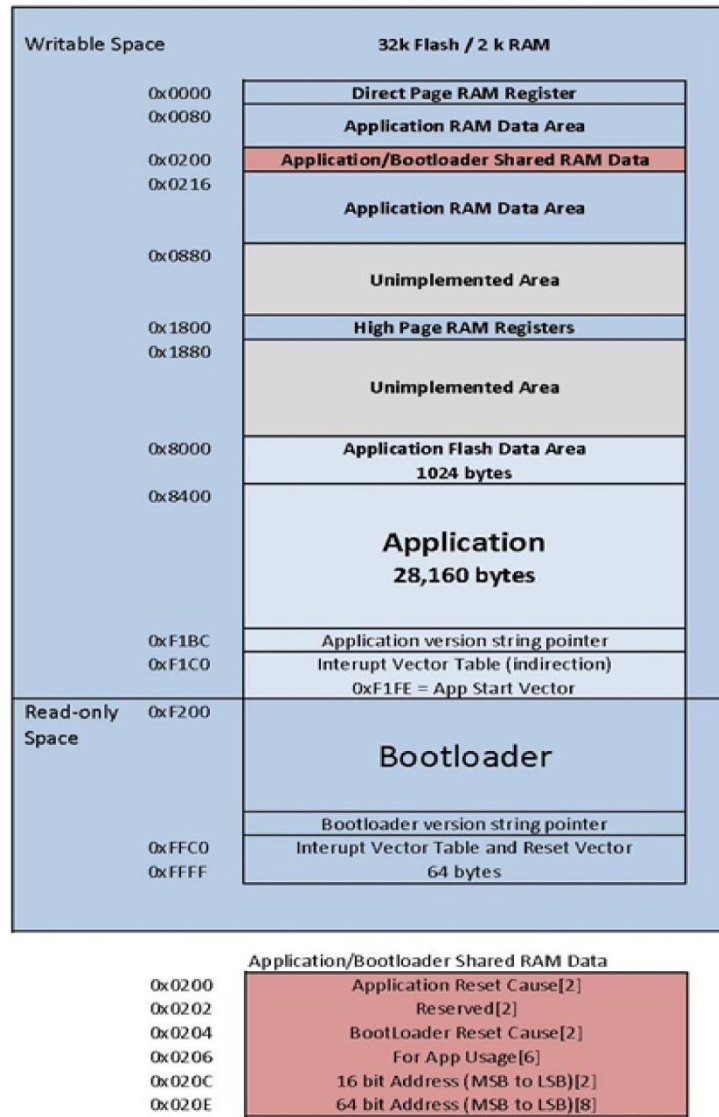


Figure 1: MC9S08QE32 Memory Map

## 2. RF Module Operation

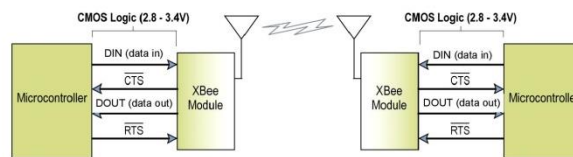
### Serial Communications

The XBee RF Modules interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device (for example: through a RS-232 or USB interface board).

#### UART Data Flow

Devices that have a UART interface can connect directly to the pins of the RF module as shown in the figure below.

System Data Flow Diagram in a UART-interfaced environment  
(Low-asserted signals distinguished with horizontal line over signal name.)

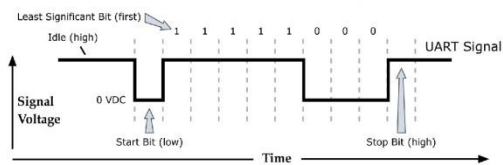


#### Serial Data

Data enters the module UART through the DIN (pin 3) as an asynchronous serial signal. The signal should idle high when no data is being transmitted.

Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high). The following figure illustrates the serial bit pattern of data passing through the module.

UART data packet 0x1F (decimal number "31") as transmitted through the RF module  
Example Data Format is 8-N-1 (bits - parity - # of stop bits)

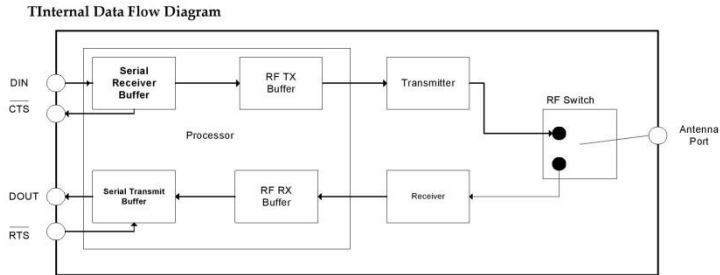


Serial communications depend on the two UARTs (the microcontroller's and the RF module's) to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits).

The UART baud rate, parity, and stop bits settings on the XBee module can be configured with the BD, NB, and SB commands respectively. See the command table in chapter 10 for details.

#### Serial Buffers

The XBee modules maintain small buffers to collect received serial and RF data, which is illustrated in the figure below. The serial receive buffer collects incoming serial characters and holds them until they can be processed. The serial transmit buffer collects data that is received via the RF link that will be transmitted out the UART.



### Serial Receive Buffer

When serial data enters the RF module through the DIN Pin (pin 3), the data is stored in the serial receive buffer until it can be processed. Under certain conditions, the module may not be able to process data in the serial receive buffer immediately. If large amounts of serial data are sent to the module,  $\overline{CTS}$  flow control may be required to avoid overflowing the serial receive buffer.

#### Cases in which the serial receive buffer may become full and possibly overflow:

1. If the module is receiving a continuous stream of RF data, the data in the serial receive buffer will not be transmitted until the module is no longer receiving RF data.
2. If the module is transmitting an RF data packet, the module may need to discover the destination address or establish a route to the destination. After transmitting the data, the module may need to retransmit the data if an acknowledgment is not received, or if the transmission is a broadcast. These issues could delay the processing of data in the serial receive buffer.

### Serial Transmit Buffer

When RF data is received, the data is moved into the serial transmit buffer and sent out the UART. If the serial transmit buffer becomes full enough such that all data in a received RF packet won't fit in the serial transmit buffer, the entire RF data packet is dropped.

#### Cases in which the serial transmit buffer may become full resulting in dropped RF packets

1. If the RF data rate is set higher than the interface data rate of the module, the module could receive data faster than it can send the data to the host.
2. If the host does not allow the module to transmit data out from the serial transmit buffer because of being held off by hardware flow control.

### Serial Flow Control

The  $\overline{RTS}$  and  $\overline{CTS}$  module pins can be used to provide  $\overline{RTS}$  and/or  $\overline{CTS}$  flow control.  $\overline{CTS}$  flow control provides an indication to the host to stop sending serial data to the module.  $\overline{RTS}$  flow control allows the host to signal the module to not send data in the serial transmit buffer out the uart.  $\overline{RTS}$  and  $\overline{CTS}$  flow control are enabled using the D6 and D7 commands.

#### $\overline{CTS}$ Flow Control

If  $\overline{CTS}$  flow control is enabled (D7 command), when the serial receive buffer is 17 bytes away from being full, the module de-asserts  $\overline{CTS}$  (sets it high) to signal to the host device to stop sending serial data.  $\overline{CTS}$  is re-asserted after the serial receive buffer has 34 bytes of space.

#### $\overline{RTS}$ Flow Control

If  $\overline{RTS}$  flow control is enabled (D6 command), data in the serial transmit buffer will not be sent out the DOUT pin as long as  $\overline{RTS}$  is de-asserted (set high). The host device should not de-assert  $\overline{RTS}$  for long

When data is transmitted from one node to another, a network-level acknowledgement is transmitted back across the established route to the source node. This acknowledgement packet indicates to the source node that the data packet was received by the destination node. If a network acknowledgement is not received, the source node will re-transmit the data.

It is possible in rare circumstances for the destination to receive a data packet, but for the source to not receive the network acknowledgment. In this case, the source will retransmit the data, which could cause the destination to receive the same data packet multiple times. The XBee modules do not filter out duplicate packets. The application should include provisions to address this potential issue

See Data Transmission and Routing in chapter 4 for more information.

### Receive Mode

If a valid RF packet is received, the data is transferred to the serial transmit buffer.

### Command Mode

To modify or read RF Module parameters, the module must first enter into Command Mode - a state in which incoming serial characters are interpreted as commands. Refer to the API Mode section in chapter 9 for an alternate means of configuring modules.

#### AT Command Mode

##### To Enter AT Command Mode:

Send the 3-character command sequence "+++" and observe guard times before and after the command characters. [Refer to the "Default AT Command Mode Sequence" below.]

Default AT Command Mode Sequence (for transition to Command Mode):

- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]
- Input three plus characters ("+++") within one second [CC (Command Sequence Character) parameter = 0x2B.]
- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]

Once the AT command mode sequence has been issued, the module sends an "OK\r" out the DOUT pin. The "OK\r" characters can be delayed if the module has not finished transmitting received serial data.

When command mode has been entered, the command mode timer is started (CT command), and the module is able to receive AT commands on the DIN pin.

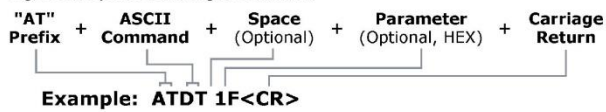
All of the parameter values in the sequence can be modified to reflect user preferences.

NOTE: Failure to enter AT Command Mode is most commonly due to baud rate mismatch. By default, the BD (Baud Rate) parameter = 3 (9600 bps).

##### To Send AT Commands:

Send AT commands and parameters using the syntax shown below.

Figure 2-01. Syntax for sending AT Commands



To read a parameter value stored in the RF module's register, omit the parameter field.

The preceding example would change the RF module Destination Address (Low) to "0x1F". To store the new value to non-volatile (long term) memory, subsequently send the WR (Write) command.

For modified parameter values to persist in the module's registry after a reset, changes must be saved to non-volatile memory using the WR (Write) Command. Otherwise, parameters are restored to previously saved values after the module is reset.

#### Command Response



## 3. XBee ZigBee Networks

---

### Introduction to ZigBee

---

ZigBee is an open global standard built on the IEEE 802.15.4 MAC/PHY. ZigBee defines a network layer above the 802.15.4 layers to support advanced mesh routing capabilities. The ZigBee specification is developed by a growing consortium of companies that make up the ZigBee Alliance. The Alliance is made up of over 300 members, including semiconductor, module, stack, and software developers.

### ZigBee Stack Layers

---

The ZigBee stack consists of several layers including the PHY, MAC, Network, Application Support Sublayer (APS), and ZigBee Device Objects (ZDO) layers. Technically, an Application Framework (AF) layer also exists, but will be grouped with the APS layer in remaining discussions. The ZigBee layers are shown in the figure below.

A description of each layer appears in the following table:

ZigBee Layer	Description
PHY	Defines the physical operation of the ZigBee device including receive sensitivity, channel rejection, output power, number of channels, chip modulation, and transmission rate specifications. Most ZigBee applications operate on the 2.4 GHz ISM band at a 250kbps data rate. See the IEEE 802.15.4 specification for details.
MAC	Manages RF data transactions between neighboring devices (point to point). The MAC includes services such as transmission retry and acknowledgment management, and collision avoidance techniques (CSMA-CA).
Network	Adds routing capabilities that allows RF data packets to traverse multiple devices (multiple "hops") to route data from source to destination (peer to peer).
APS (AF)	Application layer that defines various addressing objects including profiles, clusters, and endpoints.
ZDO	Application layer that provides device and service discovery features and advanced network management capabilities.

### Networking Concepts

---

#### Device Types

---

ZigBee defines three different device types: coordinator, router, and end device.

Node Types / Sample of a Basic ZigBee Network Topology

A **coordinator** has the following characteristics: it

- Selects a channel and PAN ID (both 64-bit and 16-bit) to start the network
- Can allow routers and end devices to join the network
- Can assist in routing data
- Cannot sleep--should be mains powered
- Can buffer RF data packets for sleeping end device children.

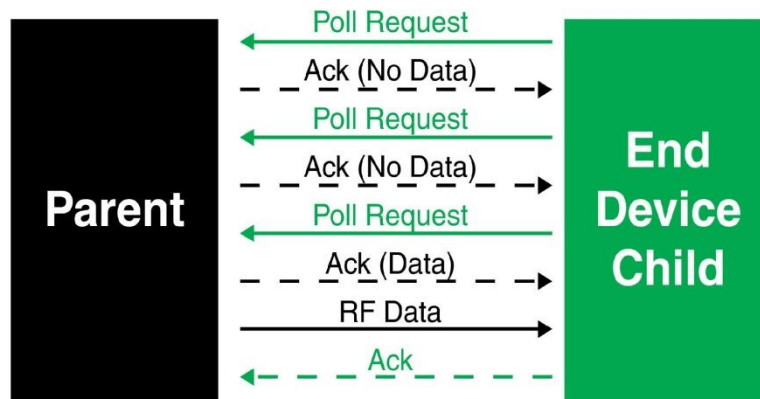
## 7. Managing End Devices

ZigBee end devices are intended to be battery-powered devices capable of sleeping for extended periods of time. Since end devices may not be awake to receive RF data at a given time, routers and coordinators are equipped with additional capabilities (including packet buffering and extended transmission timeouts) to ensure reliable data delivery to end devices.

### End Device Operation

When an end device joins a ZigBee network, it must find a router or coordinator device that is allowing end devices to join. Once the end device joins a network, a parent-child relationship is formed between the end device and the router or coordinator that allowed it to join. See chapter 3 for details.

When the end device is awake, it sends poll request messages to its parent. When the parent receives a poll request, it checks a packet queue to see if it has any buffered messages for the end device. It then sends a MAC layer acknowledgment back to the end device that indicates if it has data to send to the end device or not.



If the end device receives the acknowledgment and finds that the parent has no data for it, the end device can return to idle mode or sleep. Otherwise, it will remain awake to receive the data. This polling mechanism allows the end device to enter idle mode and turn its receiver off when RF data is not expected in order to reduce current consumption and conserve battery life.

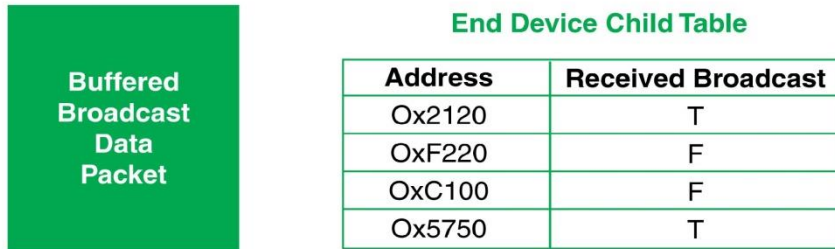
The end device can only send data directly to its parent. If an end device must send a broadcast or a unicast transmission to other devices in the network, it sends the message directly to its parent and the parent performs any necessary route or address discoveries to route the packet to the final destination.

### Parent Operation

Each router or coordinator maintains a child table that contains the addresses of its end device children. A router or coordinator that has unused entries in its child table is said to have end device capacity, or the ability to allow new end devices to join. If the child table is completely filled (such that the number of its end device children matches the number of child table entries), the device cannot allow any more end devices to join to it.

Since the end device children are not guaranteed to be awake at a given time, the parent is responsible for managing incoming data packets in behalf of its end device children. If a parent receives an RF data transmission destined for one of its end device children, and if the parent has enough unused buffer space, it will buffer the packet. The data packet will remain buffered until a timeout expires, or until the end device sends a poll request to retrieve the data.

The parent can buffer one broadcast transmission for all of its end device children. When a broadcast transmission is received and buffered, the parent sets a flag in its child table when each child polls and retrieves the packet. Once all children have received the broadcast packet, the buffered broadcast packet is discarded. If all children have not received a buffered broadcast packet and a new broadcast is received, the old broadcast packet is discarded, the child table flags are cleared, and the new broadcast packet is buffered for the end device children. This is demonstrated in the figure below.



When an end device sends data to its parent that is destined for a remote device in the network, the parent buffers the data packet until it can establish a route to the destination. The parent may perform a route or 16-bit address discovery in behalf of its end device children. Once a route is established, the parent sends the data transmission to the remote device.

#### End Device Poll Timeouts

To better support mobile end devices (end devices that can move around in a network), parent router and coordinator devices have a poll timeout for each end device child. If an end device does not send a poll request to its parent within the poll timeout, the parent will remove the end device from its child table. This allows the child table on a router or coordinator to better accommodate mobile end devices in the network.

#### Packet Buffer Usage

Packet buffer usage on a router or coordinator varies depending on the application. The following activities can require use of packet buffers for up to several seconds:

- Route and address discoveries
- Application broadcast transmissions
- Stack broadcasts (e.g. ZDO "Device Announce" messages when devices join a network)
- Unicast transmissions (buffered until acknowledgment is received from destination or retries exhausted)
- Unicast messages waiting for end device to wake.

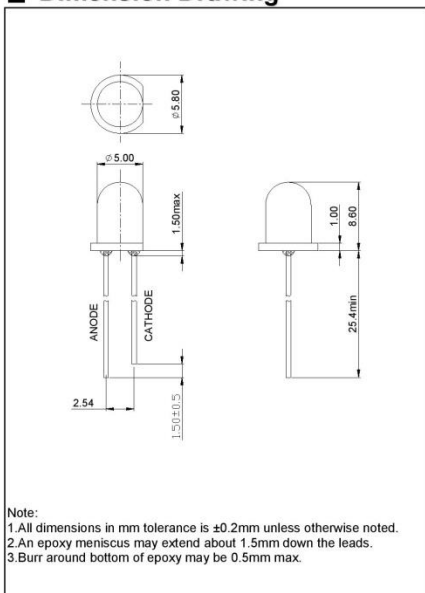
Applications that use regular broadcasting or that require regular address or route discoveries will use up a significant number of buffers, reducing the buffer availability for managing packets for end device children. Applications should reduce the number of required application broadcasts, and consider implementing an external address table or many-to-one and source routing if necessary to improve routing efficiency.

#### Non-Parent Device Operation

Devices in the ZigBee network treat data transmissions to end devices differently than transmissions to other routers and coordinators. Recall that when a unicast transmission is sent, if a network acknowledgment is not received within a timeout, the device resends the transmission. When transmitting data to remote coordinator or router devices, the transmission timeout is relatively short since these devices are powered and responsive. However, since end devices may sleep for some time, unicast transmissions to end devices use an extended timeout mechanism in order to allow enough time for the end device to wake and receive the data transmission from its parent.

## Anexo 10. LED

### Dimension Drawing



Note:  
 1. All dimensions in mm tolerance is  $\pm 0.2$ mm unless otherwise noted.  
 2. An epoxy meniscus may extend about 1.5mm down the leads.  
 3. Burr around bottom of epoxy may be 0.5mm max.

### Applications:

- Toys
- Lighting
- Traffic light
- Automotive
- Commercial Outdoor Advertising
- Front Panel Indicator

### Absolute Maximum Ratings ( $T_a = 25^\circ\text{C}$ )

Items	Symbol	Absolute maximum Rating	Unit
Forward Current( DC )	$I_F$	50	mA
Peak Forward Current*	$I_{FP}$	100	mA
Reverse Voltage	$V_R$	5	V
Power Dissipation	$P_D$	150	mW
Operation Temperature	$T_{opr}$	-20 ~ +95	$^\circ\text{C}$
Storage Temperature	$T_{sta}$	-40 ~ +100	$^\circ\text{C}$
Lead Soldering Temperature	$T_{sol}$	Max.260 $^\circ\text{C}$ for 5 sec Max. (3mm from the base of the epoxy bulb)	

\*pulse width  $\leq 0.1$ msec duty  $\leq 1/10$

### Typical Electrical & Optical Characteristics ( $T_a = 25^\circ\text{C}$ )

Items	Symbol	Condition	Min.	Typ.	Max.	Unit
Forward Voltage	$V_F$	$I_F = 20\text{mA}$	1.8	---	2.4	V
Reverse Current	$I_R$	$V_R = 5\text{V}$	---	---	10	$\mu\text{A}$
Dominant Wavelength	$\lambda_D$	$I_F = 20\text{mA}$	618	---	630	nm
Luminous Intensity	$I_V$	$I_F = 20\text{mA}$	1700	---	3500	mcd
50% Power Angle	$2\theta_{1/2}$	$I_F = 20\text{mA}$	---	30	---	deg

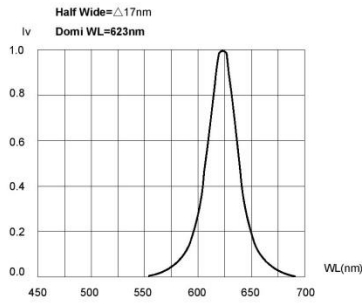
### Ranks Combination ( $I_F = 20\text{mA}$ )

Rank	0D	0E	0F	0G	---	
Dominant Wavelength (nm)	618-621	621-624	624-627	627-630	---	
Rank	0Q	1R	2R	0S	---	
Luminous Intensity (mcd)	1700-2000	2000-2500	2500-3000	3000-3500	---	
Rank	0G	0H	0J	0K	0L	0M
Forward Voltage(V)	1.8-1.9	1.9-2.0	2.0-2.1	2.1-2.2	2.2-2.3	2.3-2.4

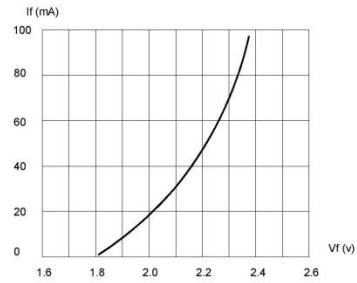
### Important Notes:

- 1) All ranks will be included per delivery.
- 2) Tolerance of measurement of luminous intensity is  $\pm 15\%$ .
- 3) Tolerance of measurement of dominant wavelength is  $\pm 1\text{nm}$ .
- 4) Tolerance of measurement of forward voltage is  $\pm 0.05\text{V}$ .
- 5) Pb content  $< 1000\text{PPM}$ .

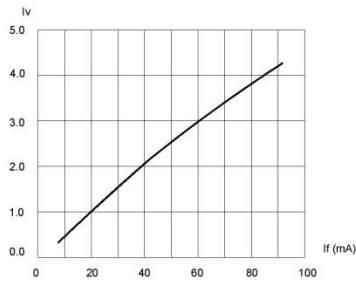
**Typical Electrical/ Optical Characteristics Curves**  
 (Ta=25°C Unless Otherwise Noted)



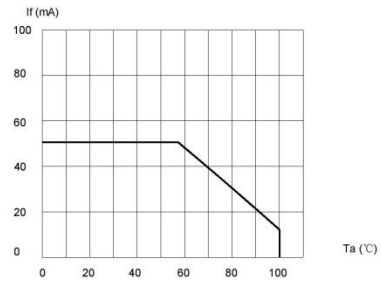
**Fig.1 Relative Luminous Intensity vs. Wavelength**



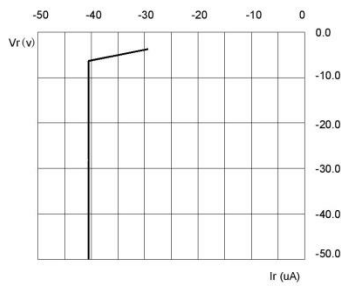
**Fig.2 Forward Current vs. Forward Voltage**



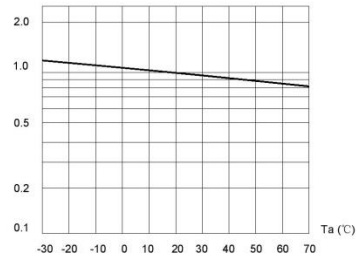
**Fig.3 Relative Luminous Intensity vs. Forward Current**



**Fig.4 Maximum Forward Current vs. Ambient Temperature**



**Fig.5 Reverse Current vs. Reverse Voltage**



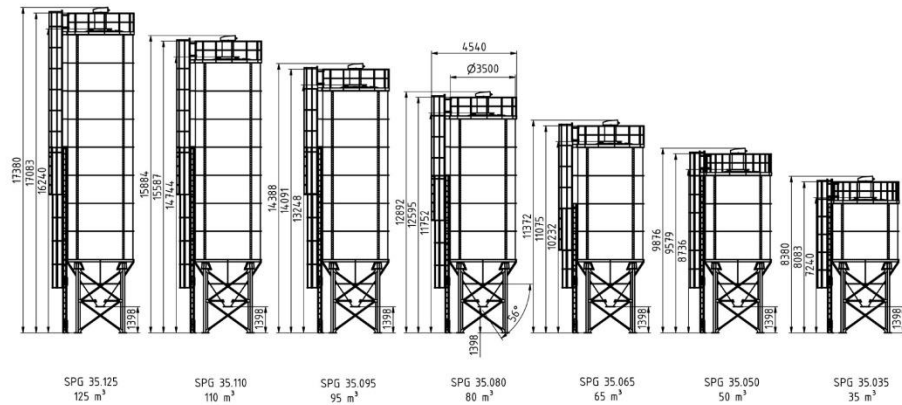
**Fig.6 Relative Luminous Intensity vs. Ambient Temperature**

## Anexo 11. Silos

### VERTIKALE SILOSYSTEME DATENBLATT / VERTICAL SILO SYSTEMS DATA SHEET

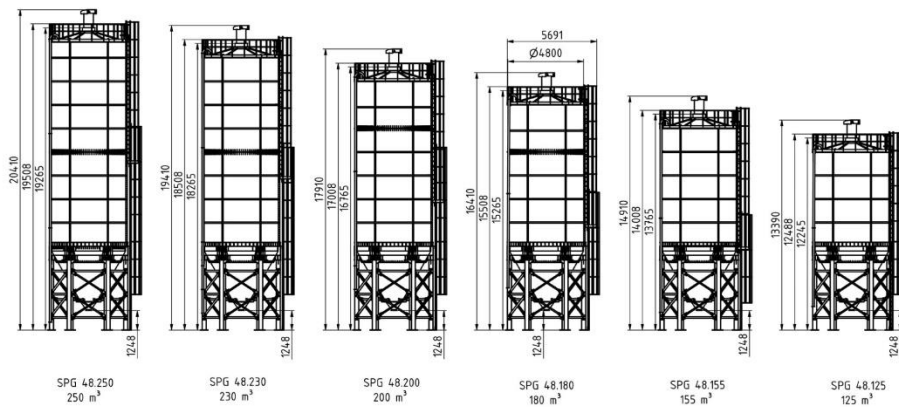
■ GESCHRAUBTE SILOS Ø 3500 mm / 35 - 125 m³ /  
BOLTED SILOS Ø 3500 mm / 35 - 125 m³

TYP / TYPE	Behältervolumen in m³ / Container volume in m³	Füllvolumen bei einer Dichte von 1,3 t/m³ / Filling capacity with a density of 1,3 t/m³	Abmessungen / Dimensions	
			Gesamthöhe / Total height	Zylinderhöhe / Cylinder height
SPG.35.035	35 m³	ca. 45 t	8380 mm	3000 mm
SPG.35.050	50 m³	ca. 65 t	9876 mm	4500 mm
SPG.35.065	65 m³	ca. 84 t	11372 mm	6000 mm
SPG.35.080	80 m³	ca. 104 t	12892 mm	7530 mm
SPG.35.095	95 m³	ca. 123 t	14388 mm	9030 mm
SPG.35.110	110 m³	ca. 143 t	15884 mm	10500 mm
SPG.35.125	125 m³	ca. 162 t	17380 mm	12000 mm



■ GESCHRAUBTE SILOS Ø 4800 mm / 125 – 250 m<sup>3</sup> /  
BOLTED SILOS Ø 4800 mm / 125 – 250 m<sup>3</sup>

TYP / TYPE	Behältervolumen in m <sup>3</sup> / Container volume in m <sup>3</sup>	Füllvolumen bei einer Dichte von 1,3 t/m <sup>3</sup> / Filling capacity with a density of 1,3 t/m <sup>3</sup>	Abmessungen / Dimensions	
			Gesamthöhe / Total height	Zylinderhöhe / Cylinder height
SPG 48.125	125 m <sup>3</sup>	ca. 162 t	13390 mm	5980 mm
SPG 48.155	155 m <sup>3</sup>	ca. 201 t	14910 mm	7500 mm
SPG 48.180	180 m <sup>3</sup>	ca. 234 t	16410 mm	9000 mm
SPG 48.200	200 m <sup>3</sup>	ca. 260 t	17910 mm	10500 mm
SPG 48.230	230 m <sup>3</sup>	ca. 299 t	19410 mm	12000 mm
SPG 48.250	250 m <sup>3</sup>	ca. 325 t	20410 mm	13000 mm



■ GESCHRAUBTE SILOS Ø 6100 mm / 215 – 400 m<sup>3</sup> /  
BOLTED SILOS Ø 6100 mm / 215 – 400 m<sup>3</sup>

TYP / TYPE	Behältervolumen in m <sup>3</sup> / Container volume in m <sup>3</sup>	Füllvolumen bei einer Dichte von 1,3 t/m <sup>3</sup> / Filling capacity with a density of 1,3 t/m <sup>3</sup>	Abmessungen / Dimensions	
			Gesamthöhe / Total height	Zylinderhöhe / Cylinder height
SPG 61.215	215 m <sup>3</sup>	ca. 279 t	14436 mm	6000 mm
SPG 61.260	260 m <sup>3</sup>	ca. 338 t	15936 mm	7500 mm
SPG 61.300	300 m <sup>3</sup>	ca. 390 t	17667 mm	9000 mm
SPG 61.350	350 m <sup>3</sup>	ca. 455 t	19167 mm	10500 mm
SPG 61.400	400 m <sup>3</sup>	ca. 520 t	20667 mm	12000 mm

