



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE ESTUDIOS SUPERIORES

ARAGÓN

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA
DE MONITOREO INALÁMBRICO PARA
CASA HABITACIÓN VÍA GSM**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO MECÁNICO ELECTRICISTA

P R E S E N T A N:

MARTÍNEZ HERNÁNDEZ YOLANDA

MARTÍNEZ PÉREZ DANIEL

ASESOR: ING. ADRIÁN PAREDES ROMERO



San Juan de Aragón, México

Enero de 2012



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

“Demos gracias a los hombres y a las mujeres que nos hacen felices,

ellos son los encantadores jardineros que hacen florecer

a nuestros espíritus”

A Dios, que me ayudo a realizar mis metas y me dio la oportunidad de vivir para lograrlos.

A mi madre, que luchaste por un sueño el cual compartimos juntas, no importando el sacrificio que realizaste, solo pensando en una meta la cual hoy podemos disfrutar. Acompañándome siempre con ese amor que me ayudo a vencer todos los problemas, depositando toda tu confianza en mí y creyendo que lo podíamos lograr. Por tal motivo, no solo es mi logro, es de las dos, porque sin ti nada de esto sería realidad.

A mi padre, que me brindo su apoyo, que creyó en mi y que hoy podemos disfrutar de este sueño que juntos empezamos.

A mi hermana, que caminamos juntas para llegar a nuestra meta, aportando esperanzas y fuerzas para afrontar todos los obstáculos que se presentaron en la vida, compartiendo alegrías y tristezas, pero sobre todo, por el cariño incondicional que me ofreces.

A Daniel, que llegaste a alegrar mi corazón con una sonrisa, ofreciéndome tu cariño, comprensión y amistad. Gracias a ello, realizamos juntos el principio de una gran historia.

A la Universidad Nacional Autónoma de México, en especial a la Facultad de Estudios Superiores Aragón y los maestros que la integran, por apoyarme en mi formación profesional, y dejarme ser parte de ellos.

POR SU CARÍÑO, COMPRENSIÓN Y APOYO

GRACIAS

YOLANDA MARTÍNEZ HERNÁNDEZ

CON AMOR, RESPETO Y ADMIRACIÓN

A DIOS

Por permitirme estar al lado de
grandes personas que me han visto crecer.

A MI FAMILIA

Quienes fueron, son y serán pilares
fundamentales en mi formación y educación como persona.

A MIS PADRES GABINA Y AMADO

Por su comprensión y ayuda en los momentos malos, menos malos y buenos.

A mi hermana BEATRIZ y mi hermano AMADO

Porque siempre me motivan y aconsejan, para que mis éxitos de hoy sean los suyos
mañana y siempre.

A MI SOBRINO ARTURO

Por la alegría que me da tu compañía "TUDITO".

A YOLANDA

Por quererme, aceptarme, cuidarme,
motivarme y aguantarme, por todo tu amor, por ser tal y como eres.

A DOÑA ROSA, YOLANDA, ANA, DON JERONIMO

Por permitirme formar parte en sus vidas, en las risas y en los llantos, en la valiosa
convivencia que tuve, tengo y tendré a su lado.

**GRACIAS A TODOS, PORQUE ME AYUDARON
A QUE ESTE GRAN ESFUERZO SE VOLVIERA REALIDAD.**

DANIEL MARTÍNEZ PÉREZ

OBJETIVO

Diseñar y construir un sistema inalámbrico para monitorear temperatura, día y noche, apertura de puertas y ventanas, cierre de puertas y ventanas, presencia humana y humedad, a través del sistema SMS/GSM para una casa habitación.

JUSTIFICACIÓN

Debido al constante crecimiento de la tecnología inalámbrica, es importante utilizar estas como un medio para generar aplicaciones que sean útiles y aplicables al mercado actual enfocando una comunicación no solamente maquina-maquina, si no también incluir una comunicación tipo maquina-usuario y usuario-maquina vía remota.

Con respecto a esto, la domótica abre nuevas posibilidades en cuanto a la integración del hogar, pues se convierte en una herramienta mediante la cual los habitantes de un hogar pueden controlar y administrar de manera eficiente su espacio en común.

Para lograr esto se debe relacionar la información de todos los dispositivos de control en un solo estándar que permita tenerlos a todos bajo un mismo control en una sola red inalámbrica (comunicación maquina-maquina) y mediante la aplicación de mensajes cortos de texto SMS se pueden controlar diferentes eventos vía remota (comunicación maquina-usuario y usuario maquina).

ÍNDICE

OBJETIVO

JUSTIFICACIÓN

CAPÍTULO I

UBICACIÓN Y DISTRIBUCIÓN DE LA CASA HABITACIÓN

1.1	Introducción	1
1.1.1	Sistemas de instalación	1
1.1.2	Arquitectura de los sistemas	1
1.1.3.	Elementos de la instalación	2
1.2	Ubicación	2
1.3	Aspecto original	3
1.4	Modificación	4
1.4.1	Dormitorio1	5
1.4.2	Sala	5
1.4.3	Dormitorio2	5
1.4.4	Jardín	6
1.4.5	Cocina y comedor	6

CAPÍTULO II

MICROCONTROLADOR PIC16F877A

2.1	Microcontrolador	7
2.1.1	Arquitectura VON NEUMANN	7
2.1.2	Arquitectura HARVARD	7
2.2	Organización de la Memoria	8
2.2.1	Organización de la memoria de programa	8
2.2.2	Organización de la memoria de datos	9
2.2.3	Registros de propósito general	10
2.2.4	Registros de funciones especiales	11
2.3	Puertos Digitales	12
2.4	Módulo Convertidor analógico/digital (A/D)	13
2.4.1	Módulo ADC del PIC16F877A	14
2.4.2	Configuración del convertidor A/D	14
2.4.3	Requerimientos para la adquisición A/D	16
2.4.4	Selección del clock de conversión Analógica Digital	16
2.4.5	Conversiones A/D	17
2.4.6	Registros De Resultado A/D	18
2.5	Módulo USART	18
2.5.1	Generador de Baudios	19
2.5.2	El Circuito De Muestreo	19
2.5.3	El Transmisor Asíncrono	20
2.5.4	Receptor Asíncrono	21

CAPÍTULO III

COMPONENTES DEL SISTEMA

3.1	Introducción	24
3.2	Sensor Temperatura Ambiente	24
3.2.1	Introducción	24
3.2.2	Medición de la temperatura ambiente	25
3.2.3	Sensor de Temperatura LM35	25
3.3	Sensor de Humedad Ambiente	26
3.3.1	Introducción	26
3.3.2	Medición de la humedad ambiente	27
3.3.2	Sensor de Humedad HMZ-433	28
3.4	Sensor de Efecto Hall	30
3.4.1	Introducción	30
3.4.2	Efecto Hall	30
3.4.3	Aplicaciones del sensor de efecto hall	31
3.4.4	Sensor Hall DN6851A	31
3.5	Sensor PIR (Passive infrared sensor)	32
3.5.1	Introducción	32
3.5.2	Lente de Fresnel Infrarrojo	34
3.5.3	Sensor movimiento PIR SE-10	36
3.6	Fotorresistencia	37
3.6.1	Introducción	37
3.6.2	Circuito Representativo	38
3.7	Relevador	39
3.7.1	Tipos de relés	40
3.7.1.1	Relés electromecánicos	40
3.7.1.2	Relé de estado solido	41
3.7.1.3	Relé de corriente alterna	41
3.7.1.4	Relé de láminas	41
3.7.2	Ventajas del uso de relés	41
3.8	LCD	42
3.8.1	Introducción	42
3.8.2	Los caracteres del LCD	43
3.8.3	Diversidad de módulos LCD	43
3.8.4	DD RAM (Display Data Ram)	43
3.8.5	CG RAM (Character Generator Ram)	44
3.8.6	Asignación de pines	45

CAPÍTULO IV

COMUNICACIÓN INALÁMBRICA XBEE

4.1	Módulo XBEE	46
4.1.1	Introducción	46
4.1.2	Comunicación Inalámbrica	47
4.1.3	Estándar IEE 802.15.4	50
4.2	Tipos de dispositivos en una red XBEE	51
4.2.1	El coordinador	51

4.2.2	Los routers	51
4.2.3	Los end devices	51
4.3	Topologías de red Xbee	51
4.3.1	Topología en estrella	51
4.3.2	Topología en árbol	52
4.3.3	Topologías en malla	52
4.4	Conexiones	53
4.4.1	Red Mesh	54
4.4.2	Circuito básico para el XBEE	54
4.4.3	Modos de operación	55
4.5	Configuración de los XBEE	57
4.5.1	Características del software	57
4.5.1.1	PC Settings	58
4.5.1.2	Configuración del Host	58
4.5.1.3	Prueba de rango	58
4.5.1.4	Terminal	59
4.5.1.5	Configuración del modem	60
4.6	Tipos de dispositivos Xbee	61
4.7	Conceptos del firmware	62
4.8	Ejemplos de transmisión	62
4.9	Modo API	63
4.10	Modulación QPSK	65

CAPÍTULO V

MODULO GSM/GPRS

5.1	Introducción	69
5.2	La telefonía celular	70
5.3	Conceptos básicos de GSM/GPRS	72
5.3.1	Frecuencia de transmisión	73
5.3.2	GPRS	74
5.3.3	Comandos AT	75
5.3.4	SMS	75
5.4	Funcionamiento GSM	76
5.4.1	El uso del espectro	77
5.4.2	Aprovechamiento del canal en la transmisión de voz	78
5.4.3	Aprovechamiento del canal en la transmisión de datos	78
5.4.4	Generaciones 3G, 3.5G y sucesivas	80
5.5	Módulo GSM/GPRS PLUTO	81
5.5.1	Tarjeta de Evaluación para PLUTO1	82
5.5.2	Descripción de la tarjeta de evaluación para PLUTO	83
5.5.3	Fuente de alimentación	83
5.5.4	Pines de comunicación serial	84
5.5.5	Comunicación serial RS-232	85
5.5.6	Tarjeta SIM	85
5.5.7	Entrada/Salida de Audio	86
5.6	Los comandos AT para modulo PLUTO	86
5.6.1	Comandos AT para propiedades del módulo	87
5.6.1.1	Comandos AT para Cambios en configuración	87

5.6.1.2	Comandos AT de Identificación	87
5.6.1.3	Comandos AT para Comunicación Serial	88
5.6.1.4	Comandos AT para Configuración del modulo GSM	88
5.6.1.5	Comandos AT para la Configuración del Hardware	88
5.6.1.6	Comandos AT Configuración Hardware	88
5.6.2	Calidad de señal	88
5.6.3	Servicios GSM	89
5.6.4	Comandos AT para control de llamada	89
5.6.5	Comandos AT para conexión SMS	89
5.6.6	Conexiones de hardware primarias	90
5.6.6.1	Configuración de comunicación serial	91

CAPÍTULO VI DISEÑO DEL SISTEMA

6.1	Módulo básico de temperatura	92
6.2	Módulo de humedad	99
6.3	Módulo completo	100
6.4	Módulo de control	104
6.5	Implementación	108

CONCLUSIONES

ANEXOS

BIBLIOGRAFÍA

CAPÍTULO 1 UBICACIÓN Y DISTRIBUCIÓN DE LA CASA HABITACIÓN

1.1 INTRODUCCIÓN

La tecnología ha generado diversas formas de cubrir las necesidades del ser humano, que al paso del tiempo, se van transformando para crear múltiples innovaciones. Un ejemplo de ello, es la domótica, que comúnmente se le llama “casa inteligente”. El objetivo de la domótica es el de proporcionarnos una mejor calidad de vida aumentando la comodidad, la seguridad y el confort, consiguiendo a su vez un mayor número de servicios de gran utilidad disponibles para el habitante de la casa

Varios investigadores y expertos del área de la domótica tratan la integración de sistemas inteligentes en el hogar y afirman que en el futuro todos los hogares estarán controlados por inteligencia artificial y que podríamos llegar a un mundo totalmente controlado y dominado por máquinas.

1.1.1 Sistemas de instalación

Existen una gran variedad de sistemas desarrollados que son adaptados de acuerdo a las necesidades del lugar o del propietario. Algunas de las áreas principales de la domótica son:

- Automatización.- incluye el control (abrir / cerrar, on / off y regulación) de la iluminación, climatización, etc.
- Seguridad.- incluye alarmas de intrusión, alarmas personales y alarmas técnicas.
- Telecomunicaciones.- los equipos incluyen transmisión de datos con redes locales para compartir acceso de alta velocidad a Internet. Además permite disfrutar de servicios como Telefonía (Figura 1.1).



Figura 1.1.- Integración de sistemas

1.1.2 Arquitectura de los Sistemas

La arquitectura de los sistemas hace referencia a la estructura de su red. La clasificación se realiza en base de donde reside el controlador del sistema. Las principales arquitecturas son:

- Arquitectura Centralizada. – un controlador centralizado es donde se recibe la información de múltiples sensores y, una vez procesada, genera información la cual envía a los actuadores e interfaces según el programa.
- Arquitectura Descentralizada. – existen varios controladores, interconectados por un bus, que envía información entre ellos y a los actuadores e interfaces

conectados a los controladores, según el programa, la configuración y la información que recibe de los sensores, sistemas interconectados y usuarios.

- **Arquitectura Distribuida.** - En un sistema de domótica de arquitectura distribuida, cada sensor y actuador es también un controlador capaz de actuar y enviar información al sistema según el programa, la configuración, la información que capta por sí mismo y la que recibe de los otros dispositivos del sistema. Suele ser típico de los sistemas de cableado en bus, o redes inalámbricas.
- **Arquitectura Híbrida / Mixta.** – En un sistema de domótica de arquitectura híbrida se combinan las arquitecturas de los sistemas centralizadas, descentralizadas y distribuidas. A la vez que puede disponer de un controlador central o varios controladores descentralizados, los dispositivos de interfaces, sensores y actuadores pueden también ser controladores (como en un sistema “distribuido”) y procesar la información según el programa, la configuración, la información que capta por sí mismo, y tanto actuar como enviarla a otros dispositivos de la red, sin que necesariamente pasa por otro controlador.

Al diseñar se tienen que tomar en cuenta ciertos elementos que no deben de faltar en nuestro diseño y que son parte fundamental de esta tecnología. Estos componentes son los siguientes:

- Tipo de arquitectura
- Medio de transmisión
- Protocolos de comunicaciones

1.1.3 Elementos de la Instalación

La amplitud de la solución puede variar desde un único dispositivo, que realiza una sola acción, hasta amplios sistemas que controlan prácticamente todas las instalaciones dentro de la vivienda. Los elementos de una instalación son:

- **Controlador.**- Los controladores son los dispositivos que gestionan el sistema según la programación y la información que reciben. Puede haber un controlador solo, o varios distribuidos por el sistema.
- **Sensores.**- El sensor es el dispositivo que monitorea el entorno captando información que transmite al sistema
- **Actuador.**- El actuador es un dispositivo capaz de ejecutar y/o recibir una orden del controlador y realizar una acción sobre un aparato o sistema
- **Soportes de comunicación (bus)**- Es el medio de transmisión que transporta la información entre los distintos dispositivos por un cableado propio, por la red de otros sistemas o de forma inalámbrica.
- **Interfaz.**- Las interfaces se refieren a los dispositivos y formatos en que se muestra la información del sistema para los usuarios y donde los mismos pueden interactuar con el sistema.

1.2 UBICACIÓN

La comunicación es lo más importante para un sistema inteligente. Para que la casa sea capaz de comunicarse debe de establecer diálogos con el medio y/u otros agentes. Dicha comunicación se obtiene a partir de lo que recibe el sistema a través de lo que

perciben los sensores. Su información se va retroalimentando con lo que va observando dentro de la casa y con las respuestas que recibe.

Para que la comunicación se pueda realizar hay que establecer patrones de comportamiento y planificación. El objetivo de estas técnicas es de obtener un conjunto de acciones que podemos asumir con cierta posibilidad que el usuario hará en un momento determinado, para tal fin, por supuesto el uso de sensores es imprescindible.

Antes de centrarnos en el sistema, primero se tiene que saber y observar el tipo de arquitectura donde se va a colocar la instalación, que medidas tiene, cuantos lugares diferentes existen, y que características tiene cada lugar. De esta manera conoceremos lo que se va a utilizar.

Para esto, se tiene una vivienda en la cual se realizara la instalación. Dicha vivienda es de una sola planta, mide 15 m de ancho por 20 m de largo. Cuenta con dos dormitorios, una sala, un comedor, una cocina, un jardín y un baño. La entrada principal se une con el jardín y las demás aéreas. Cada lugar cuenta con una entrada (Figura 1.2).

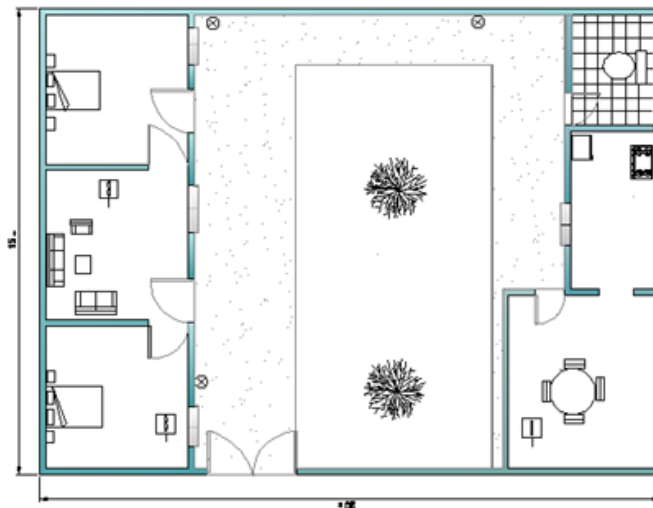


Figura 1.2.- Plano de la vivienda

Con el plano de vivienda, podemos observar las características del lugar que nos ayudara a realizar el “diseño e implementación de un sistema de monitoreo inalámbrico para casa habitación vía GSM”.

1.3 ASPECTO ORIGINAL

Para saber las características de cada área de la vivienda, vamos a realizar un breve recorrido por las estancias más relevantes de la vivienda.

Dormitorio.- Cuenta con un ventilador que se acciona manualmente, dependiendo de la temperatura deseada para el usuario. La iluminación se activa todas las noches por el usuario.

Sala.- En el se puede controlar varios dispositivos de la casa. Cuenta con un ventilador que se acciona manualmente. La iluminación se activa todas las noches por el usuario.

Comedor.- Cuenta con un ventilador que se acciona manualmente. La iluminación se activa todas las noches por el usuario.

Cocina.- La iluminación se activa todas las noches por el usuario.

Jardín.- La iluminación se activa todas las noches por el usuario. Cuenta con un sistema de riego manual.

Las características antes señaladas se utilizarán para realizar el control de cada área, según el usuario lo prefiera, desde un solo punto. En general, podemos considerar que las planificaciones que estamos considerando en el entorno del hogar son secuencias de acciones bastante simples y que no necesitan de una excesiva complejidad en el uso o desarrollo de técnicas.

1.4 MODIFICACIÓN

Las tareas que realizan el usuario para su seguridad serán ejecutadas por el sistema, que con los conocimientos extraídos de los patrones de comportamiento del usuario, se obtendrá el plan que mejor se adapte al usuario. Para observar las necesidades de cada área e identificarlas, a cada instalación de determinada área se le asignará un nombre, ya que también servirán para los datos que se mostrarán al usuario para su fácil localización (Figura 1.3). Los nombres que se les asignará a cada área son:

- Dormitorio 1 = D1
- Sala = S
- Dormitorio 2 = D2
- Jardín = J
- Cocina y Comedor = C

Ahora bien, observando que cada área cuenta con diferentes variables, y para cumplir la función de cada una, se escogerán los equipos y/o sensores que nos ayudarán para el cumplimiento de las tareas. Cabe señalar que la construcción de la vivienda ya se realizó, por lo tanto no se permite realizar un nuevo cableado, por lo cual se dedujo que el diseño será inalámbrico, ya que esto nos permitirá estar comunicados en toda la vivienda sin necesidad de modificar su arquitectura. Tomando en cuenta lo anterior, vamos a visualizar por área lo que se colocará para el funcionamiento del sistema.

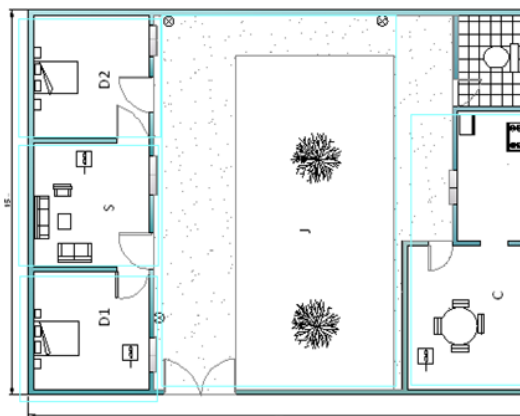


Figura 1.3.- Asignación de nombres

1.4.1 Dormitorio 1

- Sensor de Temperatura.- Accionara el ventilador que se encuentra en el lugar.
- Sensor de Luz.- Accionara la lámpara cuando se tenga ausencia de luz.

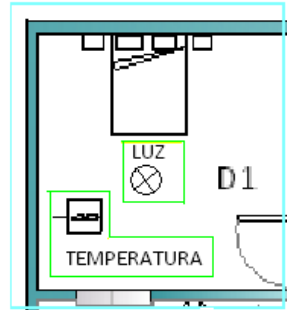


Figura 1.4.- Ubicación de sensores en D1.

1.4.2 Sala

- Sensor de Efecto Hall.- En este lugar se encuentra la entrada principal para esta parte de la casa, por lo cual se colocara un sensor en la puerta que alertara cuando se abra y/o se cierre la puerta.
- Sensor de Presencia.- Por el alcance del sensor, se colocara en esta área, pero en el exterior, y cubrirá la entrada principal y una pequeña área del jardín.
- Sensor de Temperatura.- Accionara el ventilador que se encuentra en el lugar.
- Sensor de Luz.- Accionara la lámpara cuando se tenga ausencia de luz.
- Caja de control.- En esta área se tendrá el control de todos los sensores de la casa.

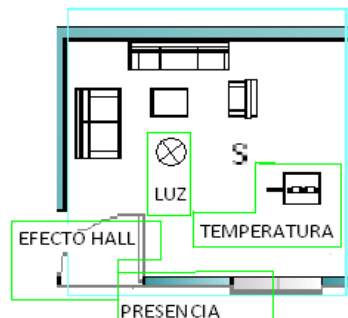


Figura 1.5.- Ubicación de sensores en S.

1.4.3 Dormitorio 2

- Sensor de Efecto Hall.- Encontramos otra entrada para el exterior en la cual se colocara un sensor en la puerta, el cual alertara cuando se abra y/o se cierre la puerta.
- Sensor de Presencia.- Por el alcance del sensor, se colocara en esta área, pero en el exterior, ya que cubrirá una parte del pasillo y una pequeña área del jardín.
- Sensor de Temperatura.- Accionara el ventilador que se encuentra en el lugar.
- Sensor de Luz.- Accionara la lámpara cuando se tenga ausencia de luz.

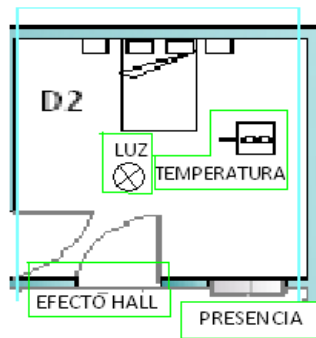


Figura 1.6.- Ubicación de sensores en D2.

1.4.4 Jardín

- Sensor de Humedad. Accionara el sistema de riego que se encuentra en el lugar.
- Sensor de Luz. Accionara la lámpara cuando se tenga ausencia de luz.

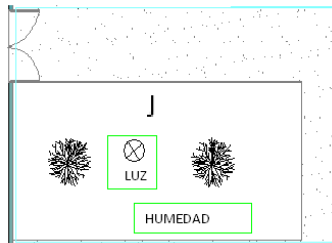


Figura 1.7.- Ubicación de sensores en J.

1.4.5 Cocina y comedor

- Sensor de Efecto Hall. En este lugar se encuentra la entrada principal para esta parte de la casa, por lo cual se colocara un sensor en la puerta que alertara cuando se abra y/o se cierre la puerta.
- Sensor de Presencia. Por el alcance del sensor, se colocara en esta área, pero en el exterior, y cubrirá el área faltante del pasillo.
- Sensor de Temperatura. Accionara el ventilador que se encuentra en el lugar.
- Sensor de Luz. Accionara las dos lámparas que se encuentran cuando se tenga ausencia de luz.

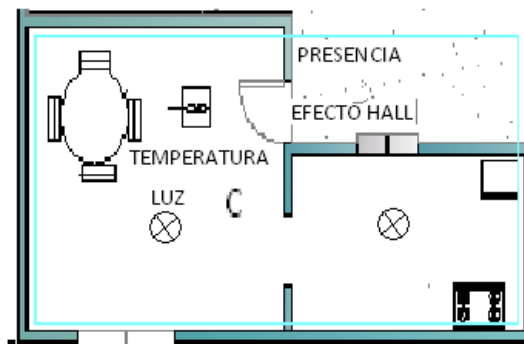


Figura 1.8.- Ubicación de sensores en C.

CAPÍTULO 2 MICROCONTROLADOR PIC16F877A

2.1 MICROCONTROLADOR

Un microcontrolador es un sistema cerrado, lo cual quiere decir que en un solo circuito integrado se encierra un sistema digital programable completo. Este dispositivo se destina a gobernar una sola tarea que no se puede modificar. Los microcontroladores disponen de los bloques esenciales: CPU, memorias de datos y programa, reloj, periféricos de entrada/salidas, etc.

2.1.1 Arquitectura De VON NEUMANN

La arquitectura tradicional de sistemas digitales programables se basa en el esquema propuesto por John Von Neumann. En este modelo la unidad central de proceso o CPU se conecta en una memoria única que contiene las instrucciones del programa y los datos (Figura 2.1). El tamaño de la unidad de datos o instrucciones esta fijado por el ancho del bus de datos de la memoria exterior utilizada, que es de 8 bits. Un microprocesador con un bus de 8 bits que lo conecta con la memoria deberá manejar datos e instrucciones de una o más unidades de 8 bits de longitud. Cuando deba acceder a una instrucción o dato de más de un byte de longitud, deberá realizar más de un acceso a la memoria. Por otro lado este bus único limita la velocidad de operación del microprocesador, ya que no se puede buscar en la memoria una nueva instrucción antes de que finalicen las transferencias de datos que pudieran resultar de la instrucción anterior.

Resumiendo, las dos principales limitaciones de la arquitectura tradicional o de Von Neumann son:

- La longitud de las instrucciones esta limitada por la unidad de longitud de los datos, por lo tanto el microprocesador debe hacer varios accesos a memoria para buscar instrucciones completas.
- La velocidad de operación esta limitada por el efecto de cuello de botella que significa un único bus para datos e instrucciones, que impide superponer ambos tiempos de acceso.

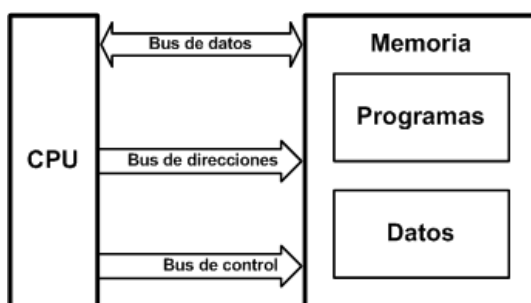


Figura 2.1.- Arquitectura Von Neumann

2.1.2 Arquitectura Harvard

La arquitectura conocida como Harvard, consiste simplemente en un esquema en el que el CPU esta conectado a dos memorias por intermedio de dos buses separados. Una de

las memorias contiene solamente las instrucciones del programa, y es llamada Memoria de Programa. La otra memoria solo almacena los datos y es llamada Memoria de Datos (Figura 2.2). Ambos buses son totalmente independientes y pueden ser de distintos anchos.

Para un procesador de Set de Instrucciones Reducido, el set de instrucciones y el bus de la memoria de programa pueden diseñarse de manera tal que todas las instrucciones tengan una sola posición de memoria de programa de longitud. Además, como los buses son independientes, el CPU puede estar accediendo a los datos para completar la ejecución de una instrucción, y al mismo tiempo estar leyendo la próxima instrucción a ejecutar. Se puede observar claramente que las principales ventajas de esta arquitectura son:

- El tamaño de las instrucciones no esta relacionado con el de los datos, por lo tanto, puede ser optimizado para que cualquier instrucción ocupe una sola posición de memoria. Así se logra una mayor velocidad y una menor longitud de programa.
- El tiempo de acceso a las instrucciones puede utilizarse con el de los datos, logrando una mayor velocidad de operación.

Una pequeña desventaja de los procesadores con arquitectura Harvard, es que deben poseer instrucciones especiales para acceder a tablas de valores constantes que puedan ser necesarios incluir en los programas, ya que estas tablas se encontraran físicamente en la memoria de programa (por ejemplo en la EPROM de un microprocesador).

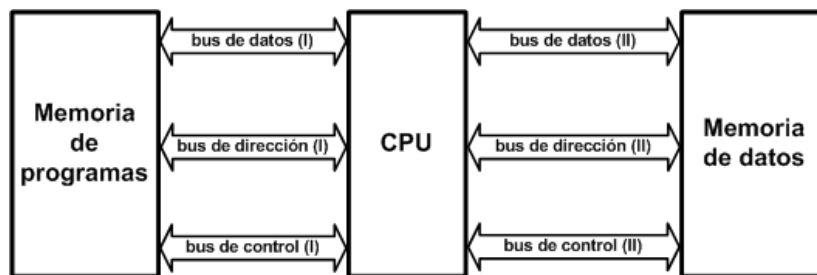


Figura 2.2.- Arquitectura Harvard

2.2 ORGANIZACIÓN DE LA MEMORIA

Esta familia de microcontroladores dispone de tres bloques de memoria. La Memoria de Programa y la Memoria de Datos RAM que tienen los buses separados para poder permitir el acceso simultáneo a estos dos bloques y la Memoria de datos EEPROM.

2.2.1 Organización de la Memoria de programa

Los dispositivos PIC16F87X tienen un contador de programa de 13 bits capaz de direccionar 8Kx14 posiciones de memoria. Los dispositivos PIC16F877A tienen 8Kx14 posiciones de memoria de programa tipo FLASH. Acceder a una posición de memoria no implementada, provoca la lectura o escritura de la posición de memoria envolvente. El vector de Reset está en la posición de memoria 0000h y el vector de Interrupción está en la posición de memoria 0004h (Figura 2.3).

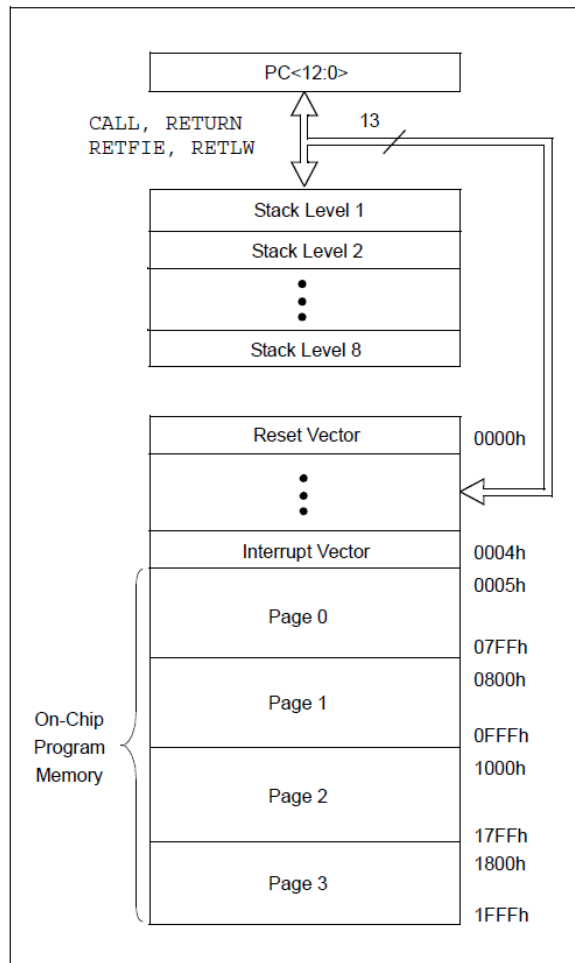


Figura 2.3.- Mapa de Memoria y STACK

2.2.2 Organización de la Memoria de datos

La memoria de datos se divide en cuatro Bancos que contienen los Registros de Propósito Generales y los Registros de la Funciones Especiales (FSR). Los bits RP1 (STATUS <6>) y RP0 (STATUS <5>) seleccionan cada uno de estos Bancos (Figura 2.4).

RP1	RP0	Banco
0	0	Banco 0
0	1	Banco 1
1	0	Banco 2
1	1	Banco 3

Figura 2.4.- Selección de los bancos de memoria

En cada Banco hay 7Fh posiciones de memoria (128 bytes). Las posiciones más bajas están reservadas para los Registros de Funciones Especiales. Por encima de los Registros de Funciones Especiales se encuentran los Registros de Propósito General, que se utilizan como posiciones de memoria RAM estática. Todos están estructurados en Bancos. Algunos Registros de Funciones Especiales están reflejados en varios Bancos para reducir el código y tener un acceso más rápido (Figura 2.5). Si analizamos estas figuras podemos ver como efectivamente después de cada grupo de registros especiales de cada Banco están las posiciones de memoria que pueden ser usadas para el almacenamiento de datos.

File Address		File Address		File Address		File Address	
Indirect addr. ^(*)	00h	Indirect addr. ^(*)	80h	Indirect addr. ^(*)	100h	Indirect addr. ^(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD ⁽¹⁾	08h	TRISD ⁽¹⁾	88h		108h		188h
PORTE ⁽¹⁾	09h	TRISE ⁽¹⁾	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved ⁽²⁾	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved ⁽²⁾	18Fh
T1CON	10h		90h	General Purpose Register 16 Bytes	110h	General Purpose Register 16 Bytes	190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah	11Ah	19Ah		
CCPR2L	1Bh		9Bh	11Bh	19Bh		
CCPR2H	1Ch	CMCON	9Ch	11Ch	19Ch		
CCP2CON	1Dh	CVRCON	9Dh	11Dh	19Dh		
ADRESH	1Eh	ADRESL	9Eh	11Eh	19Eh		
ADCON0	1Fh	ADCON1	9Fh	11Fh	19Fh		
General Purpose Register 96 Bytes	20h	General Purpose Register 80 Bytes	A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h
	7Fh		EFh		16Fh		1EFh
			accesses 70h-7Fh		F0h		accesses 70h-7Fh
Bank 0	Bank 1	Bank 2	Bank 3				

Figura 2.5.- Mapa de registros

2.2.3 REGISTROS DE PROPÓSITO GENERAL

Se puede acceder a cualquier registro utilizando el direccionamiento directo o indirecto, en este último caso a través del registro FSR.

2.2.4 REGISTROS DE FUNCIONES ESPECIALES

Los Registros de Funciones Especiales son registros usados por la CPU y los módulos periféricos para hacer trabajar el dispositivo como nosotros queremos. Estos registros están realizados como RAM estática. Los Registros de Funciones Especiales los podemos clasificar en dos tipos, los correspondientes a la CPU y los que controlan los periféricos. Los registros que afectan directamente a la CPU se describen seguidamente con detalle. Aquellos registros relacionados con el funcionamiento de los periféricos se describen en el apartado correspondiente al periférico (Figura 2.6).

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:	
Bank 0												
00h ⁽³⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	31, 150	
01h	TMR0	Timer0 Module Register								xxxx xxxx	55, 150	
02h ⁽³⁾	PCL	Program Counter (PC) Least Significant Byte								0000 0000	30, 150	
03h ⁽³⁾	STATUS	IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	0001 1xxxx	22, 150	
04h ⁽³⁾	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	31, 150	
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read							--0x 0000	43, 150
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	45, 150	
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	47, 150	
08h ⁽⁴⁾	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	48, 150	
09h ⁽⁴⁾	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxxx	49, 150	
0Ah ^(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	30, 150	
0Bh ⁽³⁾	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBFIF	0000 000x	24, 150	
0Ch	PIR1	PSPIF ⁽³⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	26, 150	
0Dh	PIR2	—	CMIF	—	EEIF	BCLIF	—	—	CCP2IF	-0-0 0--0	28, 150	
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	60, 150	
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	60, 150	
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYN \overline{C}	TMR1CS	TMR1ON	--00 0000	57, 150	
11h	TMR2	Timer2 Module Register								0000 0000	62, 150	
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	61, 150	
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	79, 150	
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	82, 82, 150	
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	63, 150	
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	63, 150	
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	64, 150	
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	112, 150	
19h	TXREG	USART Transmit Data Register								0000 0000	118, 150	
1Ah	RCREG	USART Receive Data Register								0000 0000	118, 150	
1Bh	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								xxxx xxxx	63, 150	
1Ch	CCPR2H	Capture/Compare/PWM Register 2 (MSB)								xxxx xxxx	63, 150	
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	64, 150	
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	133, 150	
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/ \overline{DONE}	—	ADON	0000 00-0	127, 150	

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:		
Bank 1													
80h ⁽³⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	31, 150		
81h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	23, 150		
82h ⁽³⁾	PCL	Program Counter (PC) Least Significant Byte								0000 0000	30, 150		
83h ⁽³⁾	STATUS	IRP	RP1	RP0	T0	PD	Z	DC	C	0001 1xxx	22, 150		
84h ⁽³⁾	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	31, 150		
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	43, 150		
86h	TRISB	PORTB Data Direction Register								1111 1111	45, 150		
87h	TRISC	PORTC Data Direction Register								1111 1111	47, 150		
88h ⁽⁴⁾	TRISD	PORTD Data Direction Register								1111 1111	48, 151		
89h ⁽⁴⁾	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction bits				0000 -111	50, 151	
8Ah ^(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter							---0 0000	30, 150
8Bh ⁽³⁾	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	24, 150		
8Ch	PIE1	PSPIE ⁽²⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	25, 151		
8Dh	PIE2	—	CMIE	—	EEIE	BCLIE	—	—	CCP2IE	-0-0 0--0	27, 151		
8Eh	PCON	—	—	—	—	—	—	POR	BOR	---- --gq	29, 151		
8Fh	—	Unimplemented								—	—		
90h	—	Unimplemented								—	—		
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	83, 151		
92h	PR2	Timer2 Period Register								1111 1111	62, 151		
93h	SSPADD	Synchronous Serial Port (I ² C mode) Address Register								0000 0000	79, 151		
94h	SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000	79, 151		
95h	—	Unimplemented								—	—		
96h	—	Unimplemented								—	—		
97h	—	Unimplemented								—	—		
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	111, 151		
99h	SPBRG	Baud Rate Generator Register								0000 0000	113, 151		
9Ah	—	Unimplemented								—	—		
9Bh	—	Unimplemented								—	—		
9Ch	CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	0000 0111	135, 151		
9Dh	CVRCON	CVREN	CVROE	CVRR	—	CVR3	CVR2	CVR1	CVR0	000- 0000	141, 151		
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	133, 151		
9Fh	ADCON1	ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0	00-- 0000	128, 151		

Figura 2.6.- Registros de Función Especiales

2.3 PUERTOS DIGITALES

El PIC16F877A contiene 5 puertos que pueden ser configurados como entrada o salida digitales (A, B, C, D, E). El puerto A contiene 6 bits (RA0-5). El puerto B (RB0-7), el puerto C (RC0-7) y el puerto D (RD0-7) tienen cada uno 8 líneas. El puerto E solo cuenta con 3 líneas (RE0-2).

La operación de configuración de los puertos en general implica la siguiente secuencia:

- Ingresar al banco 1.
- Configurar los puertos (registros TRISA, TRISB, TRISC, TRISD y TRISE)
- Regresar al banco 0.
- Escribir o leer datos desde los puertos. (registros PORTA, PORTB, PORTC, PORTD y PORTE).

En las posiciones inferiores de ambos bancos se encuentran los registros especiales de función. En la posición 0x05, 0x06, 0x07, 0x08 y 0x09 respectivamente se encuentran los registros PORTA, PORTB, PORTC, PORTD y PORTE que se usan para leer o escribir

datos, en tanto que en las posiciones 0x85, 0x86, 0x87, 0x88 y 0x89 se encuentran los registros TRISA, TRISB, TRISC, TRISD y TRISE respectivamente, es allí donde se configuran los puertos. Cabe señalar que el PORTB también aparece en el banco 2 en la posición de memoria 0x106 y el TRISB en la posición de memoria 0x186 (Figura 2.7).

Posmem	Banco 0	Banco 1	Posmem
	
0x05	PORTA	TRISA	0x85
0x06	PORTB	TRISB	0x86
0x07	PORTC	TRISC	0x87
0x08	PORTD	TRISD	0x88
0x09	PORTE	TRISE	0x89
	
		ADCON1	0x1F

Figura 2.7.- Puertos en la Posición de Memoria

Cada una de las líneas de los puertos puede ser configurada como entrada o como salida. En el registro TRIS determinamos la configuración de los puertos. Por ejemplo si escribimos un 0 en el bit 0 del TRISA la línea RA0 se comportará como una línea de salida. Si colocamos a 1 el bit 0 del TRISA a la línea RA0 se comportará como entrada.

A través de los valores que escribamos en los registros TRIS determinamos el comportamiento de los puertos. La escritura y lectura de valores desde los puertos se hace a través de los registros PORT que se encuentran en el Banco 0 (y banco 2 para el puerto B). Desde luego si configuramos un puerto como entrada (lectura) los valores que escribamos en el no tendrán efecto porque fue configurado como entrada y no como salida. A través de las instrucciones MOV podemos leer o escribir valores.

2.4 MÓDULO CONVERTIDOR ANALÓGICO/DIGITAL (ADC)

Un convertidor analógico digital tiene como entrada un nivel de voltaje (valor analógico) y produce en su salida un número binario de n bits proporcionales al nivel de la entrada (valor digital). Los convertidores de señal análoga a digital se abrevian ADC o A/D. Uno de los parámetros que definen al A/D es la resolución como la mínima variación de voltaje en la entrada que produce cambio del valor digital en la salida. Por ejemplo un convertidor de 10 bits tiene un total de 210 valores (1024 valores de 0 a 1023).

Si tenemos tv a la entrada la resolución sería de 4.882mV. En este caso el voltaje es de 5v a 0v pero pueden variar. Por ejemplo si tenemos de 5v a 2v la resolución será:

$$\text{Resolución} = (5v - 2v) / 1024 = 2.929 \text{ mV}$$

Una fórmula para el cálculo será:

$$\text{Resolución} = (V_{\text{ref2}} - V_{\text{ref1}}) / 1024$$

Donde las tensiones de referencia son 5v y 2V.

2.4.1 Módulo ADC del PIC16F877A

El PIC16F877A posee un módulo ADC interno que les permite manejar 8 entradas analógicas (Figura 2.8). La conversión de la señal analógica aplicada (a uno de los canales) se plasma en número binario de 10 dígitos. El módulo A/D posee voltajes de referencia que pueden ser seleccionados para emplear las tensiones VDD, VSS del microcontrolador o puede emplear tensiones aplicadas a los pines RA2 o RA3 (incluso es posible establecer combinaciones de los valores anteriores).

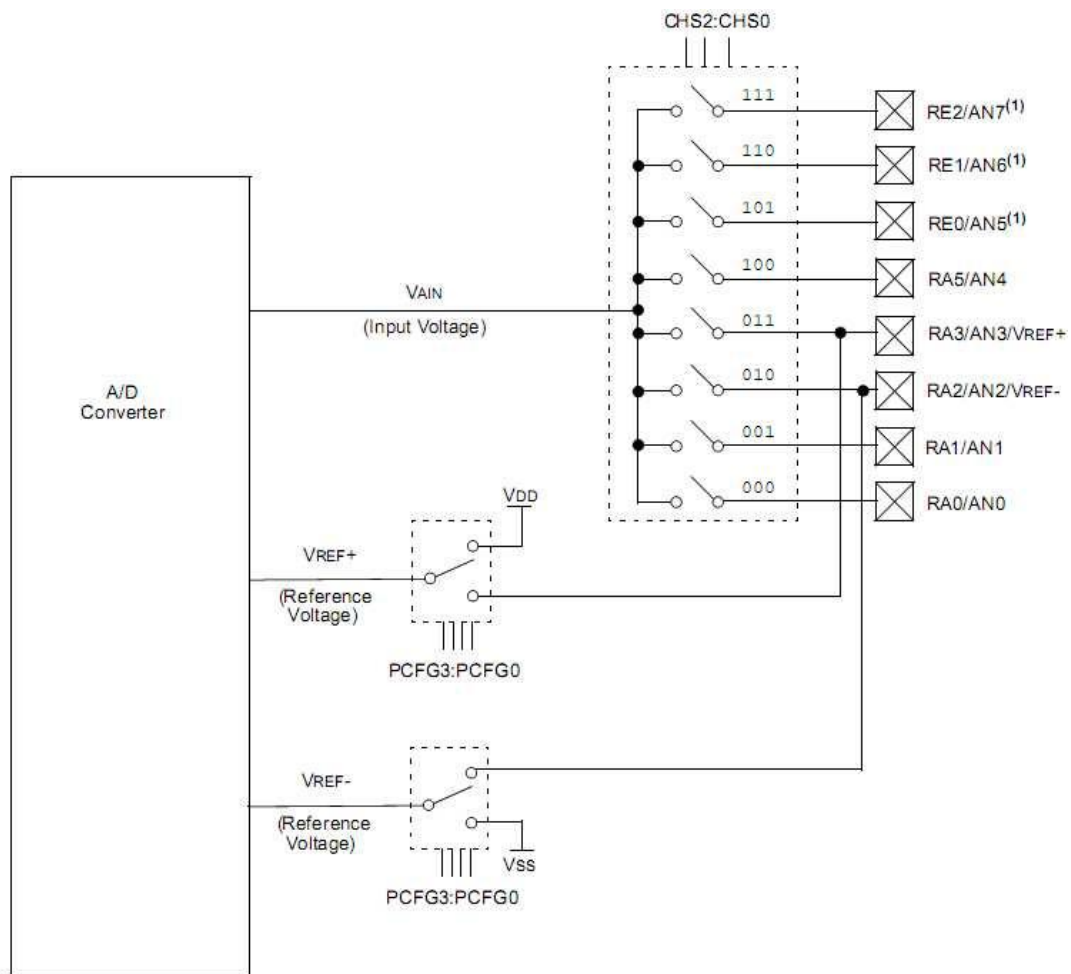


Figura 2.8.- Diagrama de bloques del módulo A/D

2.4.2 Configuración del convertidor A/D

Para operar el módulo ADC contamos con 4 registros:

- Registro de resultado de byte alto de la conversión A/D (ADRESH). Banco 0, 0x1E.
- Registro de resultado de byte bajo de la conversión A/D (ADRESL). Banco 1, 0x9E.

- Registro 0 de control del módulo A/D (ADCON0). Banco 0, 0x1F.
- Registro 1 de control del módulo A/D (ADCON1). Banco 1, 0x9F.

El registro ADCON0 controla la operación del módulo convertidor A/D. El registro ADCON1 configura las funciones de los pines de entrada al módulo, se puede configurar los pines del puerto A como entradas analógicas inclusive la línea RA3 puede funcionar como el voltaje de referencia.

Los registros ADRESH:ADRESL contienen el resultado de la conversión (10 bits). Cuando se ha completado una conversión el resultado es almacenado en ADRESH:ADRESL y además el bit GO/-DONE (registro ADCON bit 2) se pone a 0 lógico y el bit ADIF (registro PIR1 bit 7) se pone como 1 lógico. El registro PIR1 ocupa la posición 0x0C del banco 0.

Después que el módulo A/D ha sido configurado, es necesario esperar un periodo de tiempo para que la señal sea adquirida. Cada uno de los canales de entrada tiene su correspondiente bit de configuración en los registros TRIS y estos han de ser puestos como entradas.

Para lograr la conversión con el módulo A/D se debe seguir los siguientes pasos. Estos pasos son entregados por el fabricante.

1. Configurar el módulo A/D:

- Configurar pines analógicos, referencias de voltaje y entradas y salidas digitales a través del registro ADCON1.
- Seleccionar canales de entrada A/D en el registro ADCON0
- Seleccionar el reloj de conversión en el registro ADCON0.
- Encender el módulo A/D en el registro ADCON0.

2. Configurar interrupciones A/D (si se desea):

- ADIF=0 lógico (bit que indica si se produjo una conversión)
- ADIE=1 lógico (habilitador de interrupción del módulo analógico)
- PEIE=1 lógico (habilitador de interrupción de periféricos)
- GIE=1 lógico (Habilitador general de interrupciones)

3. Esperar el tiempo de adquisición necesaria.

4. Iniciar la conversión:

- GO/-DONE=1 lógico (en el registro ADCON0)

5. Esperar que el tiempo de conversión este completado:

- Revisar el bit GO/-DONE esperando que sea de nuevo 0 -lógico
- Esperar al flag de ADIF sea 1-lógico (puede emplearse como interrupción)

6. Leer resultado A/D en el par de registro (ADRESH; ADRESL), limpiar el bit ADIF si es necesario.

7. Para la siguiente conversión seguir en el paso 1 o el paso 2. El tiempo de conversión por bit está definido por TAD, se exige un mínimo de $2 \cdot TAD$ para realizar una nueva conversión.

2.4.3 Requerimientos para la adquisición A/D

Para que el módulo convertidor A/D trabaje apropiadamente la carga del capacitor que muestrea la señal analógica debe ser la máxima. La impedancia de la fuente (R_s) y la impedancia del switch interno (R_{ss}) (Figura 2.9) afectan el tiempo requerido para la carga del capacitor. La máxima impedancia recomendada para la fuente analógica es de 10 K ohmios. Cuando la impedancia es menor el tiempo de adquisición es menor y por tanto la respuesta es mejor. Luego que el canal de entrada es seleccionado (o ha sido cambiado) no olvide esperar un tiempo a que la adquisición del dato sea hecha antes de que la conversión propiamente dicha empiece.

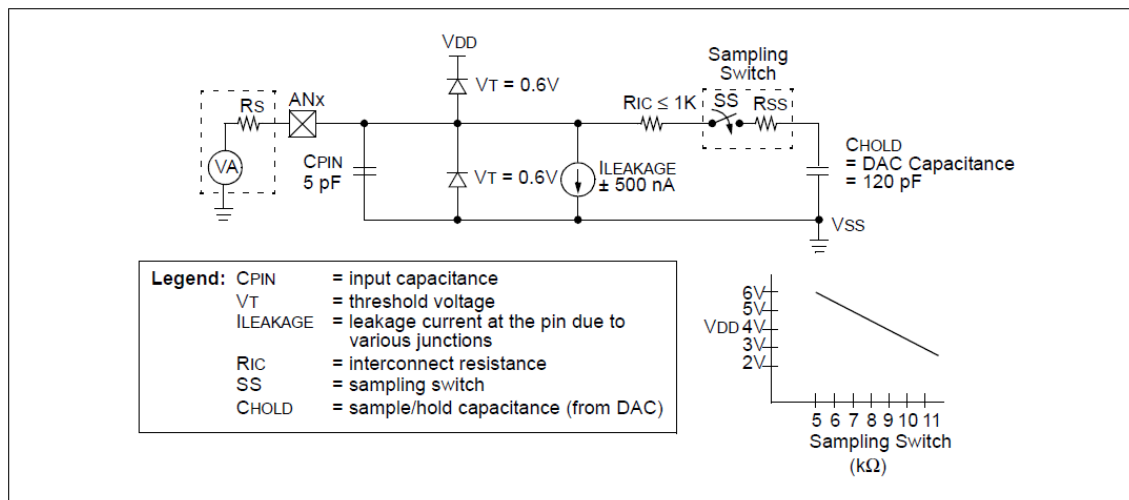


Figura 2.9.- Estructura de un pin de un canal Analógico

Para calcular el mínimo tiempo requerido en la adquisición podemos usar la ecuación:

TACQ	=	Amplifier Settling Time + Hold Capacitor Charging Time + Temperature Coefficient
	=	$TAMP + TC + TCOFF$
	=	$2 \mu s + TC + [(Temperature - 25^\circ C)(0.05 \mu s/^\circ C)]$
TC	=	$C_{HOLD} (R_{IC} + R_{SS} + R_s) \ln(1/2047)$
	=	$- 120 \text{ pF} (1 \text{ k}\Omega + 7 \text{ k}\Omega + 10 \text{ k}\Omega) \ln(0.0004885)$
	=	$16.47 \mu s$
TACQ	=	$2 \mu s + 16.47 \mu s + [(50^\circ C - 25^\circ C)(0.05 \mu s/^\circ C)]$
	=	$19.72 \mu s$

2.4.4 Selección del tiempo de conversión A/D

Existe otro parámetro que es importante mencionar y es el tiempo de conversión de A/D por BIT (definida como TAD). La conversión A/D requiere un mínimo de 12 TAD por 10 bits de conversión, La fuente para el clock en la conversión A/D se selecciona por software. Hay 7 posibles valores para la selección del TAD:

- 2Tosc
- 4Tosc
- 8Tosc
- 16Tosc
- 32Tosc
- 64Tosc
- Oscilador RC interno que tiene el modulo A/D (crea un retardo de 2µs a 6 µs).

Para el correcto funcionamiento del módulo el clock de conversión debe ser seleccionado para asegurar un TAD de 1.6 µs como mínimo (Figura 2.10).

AD Clock Source (TAD)		Maximum Device Frequency
Operation	ADCS2:ADCS1:ADCS0	
2 TOSC	000	1.25 MHz
4 TOSC	100	2.5 MHz
8 TOSC	001	5 MHz
16 TOSC	101	10 MHz
32 TOSC	010	20 MHz
64 TOSC	110	20 MHz
RC(1, 2, 3)	x11	(Note 1)

Figura 2.10.- Resultado de varios TAD calculados para diferentes clock

2.4.5 Conversiones A/D

Si iniciamos una conversión y colocamos a 0 lógico el bit GO/-DONE conseguimos abortar la conversión A/D que se este llevando a cabo en ese momento. El resultado de la conversión no aparecerá en los registros ADRESH:ADRESL y se mantendrá el último valor convertido. Después que la conversión A/D es abortada, la siguiente adquisición en el canal seleccionado empieza automáticamente. El bit GO/-DONE puede ser colocado a 1 para empezar la conversión.

Después que el bit GO es colocado a 1 lógico comienza la conversión y al inicio hay que desconectar el condensador lo cual demanda como máximo 1 TAD, luego vienen los 10TAD correspondientes a los 10 bits del A/D y se necesita un TAD mas para depositar el resultado en los registros ADRES, colocar el bit GO/-DONE de nuevo a 0-lógico y el bit ADIF a 1 lógico (Figura 2.11).

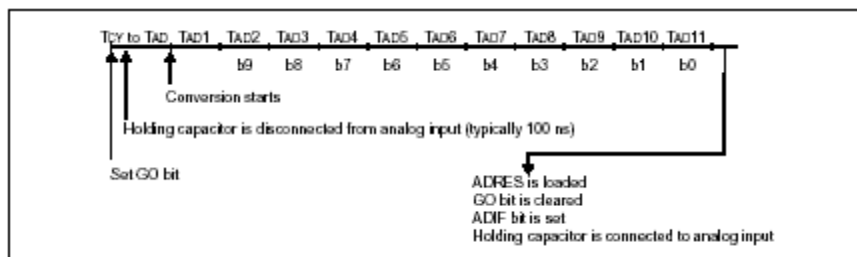


Figura 2.11.- Ciclos TAD del convertidor A/D

2.4.6 Registros De Resultado A/D

El par de registros ADRESH: ADRESL (1Eh:9Eh) son cargados con el dato resultante de una conversión analógico/digital al terminar ésta. Cada uno de estos registros es de 8 bits, por lo tanto, juntos pueden guardar hasta 16 bits. Sin embargo, como el resultado es de 10 bits, el módulo de conversión A/D permite justificarlo (alinearlo) en la parte izquierda o derecha de los 16 bits disponibles, para elegir alguna de las dos opciones se usa el bit ADFM (ADCON1<7>) (Figura 2.12).

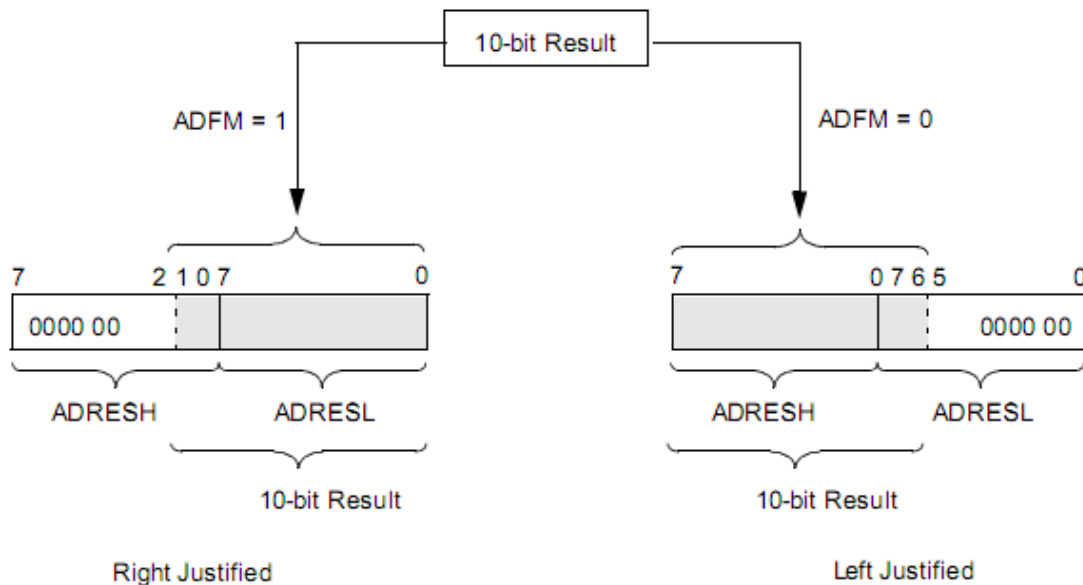


Figura 2.12.- Justificación de los resultados de la conversión A/D

Los bits extras (6 bits) son llenados con 0-lógicos. Si en un programa no se emplea el módulo A/D es posible usar los registros ADRESH :ADRESL como si fueran registros de propósito general (registros de 8 bits).

2.5 MÓDULO USART

La USART (Universal Synchronous Asynchronous Receiver Transmitter) también conocida como SCI (Serial Communications Interface) puede configurarse como una unidad de comunicación en serie para la transmisión de datos asíncrona con dispositivos tales como terminales de computadora o computadoras personales, o bien para comunicación síncrona con dispositivos tales como convertidores A/D o D/A, circuitos integrados o memorias EEPROM con comunicación serie, etc.

El USART puede trabajar de dos formas:

- Asíncrono (Bidireccional)
- Síncrono (Unidireccional)

En el modo asíncrono, la comunicación serie del USART esta soportada por las líneas RC6/TX/CK y RC7/RX/DT por las que se mueven los bits a la frecuencia interna de reloj. En el modo síncrono, los bits de información circulan en ambos sentidos por la línea DT a la frecuencia de los impulsos que genere el maestro por la línea CK.

En esta forma de comunicación serie, se usa la norma RS-232-C, donde cada palabra de información o dato se envía independientemente de los demás. Suele constatar de 8 o 9 bits y van precedidos por un bit de START (inicio) y detrás de ellos se coloca un bit de STOP (parada), de acuerdo con las normas del formato estándar NRZ (Non Return-to-Zero) Los bits se transfieren a una frecuencia fija y normalizada. La USART transmite y recibe primero el bit menos significativo.

La USART en modo asíncrono contiene los siguientes elementos:

- Generador de Baudios
- Circuito de Muestreo
- Transmisor Asíncrono
- Receptor Asíncrono

2.5.1 Generador de Baudios

Para el protocolo asíncrono RS-232-C, la frecuencia en baudios (bits por segundo) a la que se realiza la transferencia se debe efectuar a un valor normalizado: 330, 600, 1200, 2400, 4800, 9600, 19200, 38400, etc. Para generar esta frecuencia, el USART dispone de un generador de frecuencia en Baudios, BRG, cuyo valor es controlado por el contenido grabado en el registro SPBRG.

Aparte del valor X cargado en el registro SPBRG, la frecuencia en baudios del generador depende del bit BRGH del registro TXSTA <2>. En el caso de que BRGH = 0 se trabaja en baja velocidad y si BRGH = 1 se trabaja en alta velocidad. Según este bit se obtendrá el valor de una constante K necesaria en la determinación de la frecuencia de funcionamiento.

$$Frecuencia = \frac{Fosc}{(K * (X + 1))}$$

Donde:

Frecuencia = baudios

X = es el valor cargado en el registro SPBRG

BRG =

Si BRG = 0, baja velocidad y K = 64

Si BRG = 1, baja velocidad y K = 16

$$X = \frac{Fosc}{Frecuencia \times K} - 1$$

Debido a que el divisor es de 8 bits, no se puede tener cualquier velocidad de transmisión deseada, ya que X se deberá redondear al entero más cercano.

2.5.2 El Circuito De Muestreo

El dato en el pin de recepción (RC7/RX/DT) es muestreado tres veces para poder decidir mediante un circuito de mayoría, si se trata de un nivel alto o un nivel bajo.

2.5.3 El Transmisor Asíncrono

El dato que se desea transmitir por el USART se deposita en el registro TXREG y a continuación se traspassa al registro de desplazamiento TSR (Figura 2.13), que va sacando los bits secuencialmente y a la frecuencia establecida. Además, antes de los bits del dato de información incluye un bit de inicio y después de sacar todos los bits añade un bit de parada. El USART receptor recibe, uno a uno, los bits, elimina los de control y los de información una vez que han llenado el registro de desplazamiento RSR los traslada automáticamente al registro RCREG, donde quedan disponibles para su posterior procesamiento.

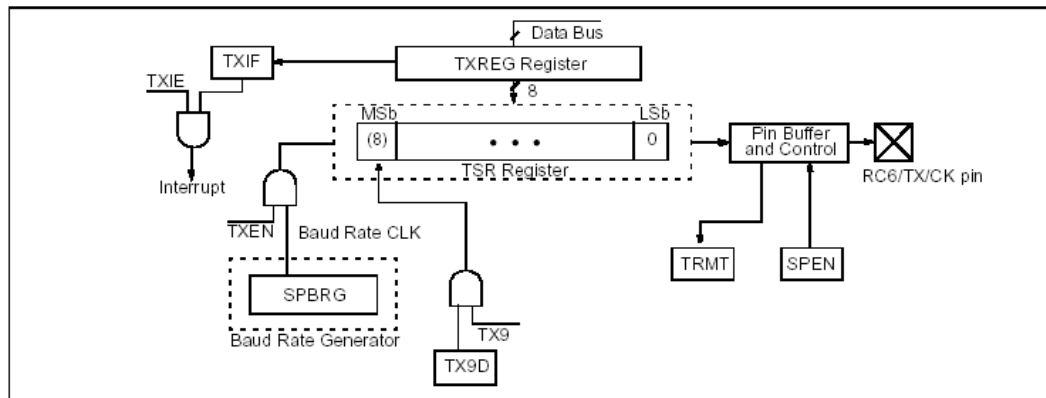


Figura 2.13 Diagrama por bloques de la sección de transmisión del USART

Si observamos el diagrama de bloques de la sección transmisora del USART (Figura 2.13). El núcleo está constituido por el registro de desplazamiento TSR, que obtiene el dato desde el registro TXREG y luego lo va desplazando y sacando bit a bit, en serie, por la línea RC6/TX/CK. El primer bit que sale es el de menos peso. El dato a transferir se carga por software en TXREG y se transfiere al TSR en cuanto se haya transmitido el bit de parada del dato anterior.

La transferencia entre los dos registros se realiza en un ciclo y entonces el señalizador TXIF se pone a 1, para advertir que el registro de transmisión se ha vaciado. También en este momento puede producirse una interrupción si se ha posibilitado el uso de interrupciones. Cuando se escribe otro dato sobre TXREG, el señalizador TXIF se pone a 0. El bit TRMT sirve para indicar el estado del registro TSR y vale 1 cuando está vacío.

La secuencia de pasos a seguir para una transmisión en el USART es la siguiente:

1. Configurar las líneas RC6/TX/CK como salida y RC7/RX/DT como entrada.
2. Asignar SYNC=0 y SPEN=1 para activar el USART como asíncrono.
3. Si se va a trabajar con interrupción, asignar TXIE=1, además de habilitar las interrupciones.
4. Si el dato consta de 9 bits, en lugar de los 8 típicos, asignar el bit TX9=1. El noveno bit se colocará en TX9D (TXSTA).
5. Se carga el valor adecuado en el registro SPBRG, para producir la frecuencia de trabajo deseada. Hay que controlar el bit BRGH (alta y baja velocidad).
6. Activar la transmisión con TXEN = 1. El bit TXIF tendrá valor 1; ya que TXREG se encuentra vacío.
7. Cargar en TXREG el dato a transmitir. Comienza la transmisión.

Registro de estado y transmisión TXTA (Dirección 98h)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0	
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	
bit 7								bit 0

- Bit 7 CSRC: bit de selección de reloj:
Modo asíncrono: no incluye.
Modo síncrono:
1 = Modo maestro (reloj generado internamente desde BRG).
0 = Modo esclavo (reloj generado por una fuente externa).
- Bit 6 TX9 : Habilita el bit 9 de transmisión:
1 = Selecciona transmisión de 9 bits.
0 = Selecciona transmisión de 8 bits.
- Bit 5 TXEN: Activa la transmisión:
1 = Transmisión activada.
0 = Transmisión desactivada.
Nota: SREN/CREN anula TXEN en modo síncrono.
- Bit 4 SYNC: Bit de selección del modo del USART:
1 = Modo síncrono.
0 = Modo asíncrono.
- Bit 3 No implementado: Leído como '0'.
- Bit 2 BRGH: Bit de selección de la velocidad de baudios:
Modo asíncrono:
1 = Alta velocidad.
0 = Baja velocidad.
Modo Síncrono:
No se usa en este modo.
- Bit 1 TRMT: Bit de estado del registro de desplazamiento de transmisión:
1= TSR vacío.
0 = TSR no vacío.
- Bit 0 TX9D: Bit 9 del dato a transmitir (puede ser el bit de paridad).

2.5.4 Receptor Asíncrono

Los datos se reciben en serie, bit a bit, por la línea RC7/RX/DT y se van introduciendo secuencialmente en el registro de desplazamiento RSR que funciona a una frecuencia 16 veces más rápida que la de trabajo (baud rate). Cuando el dato consta de 9 bits hay que programar el bit RX9 = 1 y el noveno bit de información se colocará en el bit RX9D del registro RCSTA.

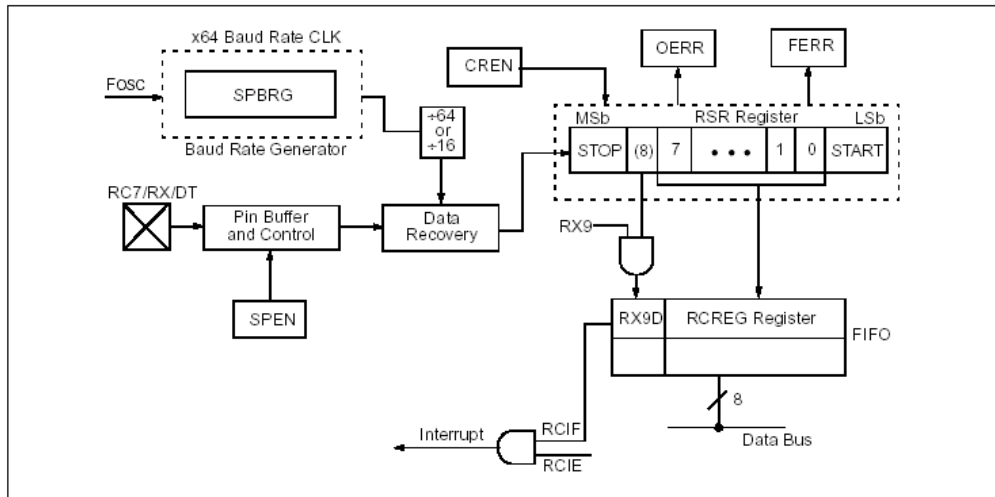


Figura 2.14.- Diagrama de bloques de la sección receptora del USART

Una vez que el modo asíncrono ha sido seleccionado, la recepción es habilitada colocando a 1 lógico el bit CREN (RCSTA). El corazón de la recepción serial es el registro de corrimiento de recepción (RSR). Después de detectar el bit de parada el dato presente en RSR es transferido al registro RCREG (si es que está vacío). Si la transferencia ha sido completada con éxito el RCIF (PIR1) es puesto a 1 lógico. Si lo deseamos podemos usar esta característica para trabajar interrupciones. Para ello deberemos habilitar el bit RCIE (PIE1). El bit RCIF solo es de lectura y colocado a 0 lógico por hardware; es decir cuando el registro RCREG esta vacío.

Al parecer RCREG es un solo registro pero no es así; el registro RCREG es una parte de pila de dos posiciones. Por tanto es capaz de almacenar 2 datos (bytes) y mantener un tercer dato en RSR. Si la pila está llena y se llena el RSR (tercer dato), es decir, llega el bit de stop de ese tercer dato, el microcontrolador procede a colocar a 1-lógico el bit OERR (over run error bit) y el valor presente en RSR se pierde.

De darse esta situación lo que debemos hacer es rescatar los dos datos que están en la pila de RCREG. Luego resetear el OERR, para eso es necesario resetear el bit CREN (ponerlo a 0 lógico y luego a 1 lógico). Cuando se activa el bit OERR las transferencias de RSR a RCREG son deshabilitadas por lo tanto debemos cumplir con hacer el reset del bit CREN para normalizar la recepción. Hay otro bit que también es importante señalar y es el bit FERR (RCSTA) error de frame o trama o marco. El bit FERR se coloca a 1 lógico si detecta un 0 lógico como bit de parada. Por lo tanto antes de leer lo que contiene el registro RCREG es necesario revisar el valor de los bits FERR y OERR.

La siguiente es la secuencia de pasos a realizar para configurar la recepción asíncrona:

1. Inicializar el registro SPBRG con el valor apropiado que genere los baudios necesarios. No olvide colocar un valor al bit BRGH en función a si va o no a transmitir en alta velocidad.
2. Habilite la puerta serial asíncrona colocando a 0 lógico el bit SYNC y a 1 lógico el bit SPEN.
3. Si se va a trabajar con interrupción, asignar RCIE=1, además de habilitar las interrupciones.

4. Habilite la recepción colocando a 1 lógico el bit CREN.
5. El bit RCIF se colocará a 1 lógico cuando un dato llegue completo al microcontrolador y una interrupción se generará si es que se ha reseteado el bit RCIE.
6. Lea el registro RCSTA para obtener el valor del noveno bit (si es que esta trabajando con 9 bits de datos) y determine si hubo error en la comunicación (revise los bits OERR y FERR).
7. Si no hubo error lea los 8 bits de datos del registro RCREG.
8. Si no hubo algún error, resetee el bit CREN.

Registro de estado y control de recepción RCSTA (Dirección 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7				bit 0			

- Bit 7 SPEN: Habilitación del puerto serie:
1 = Puerto serie habilitado (configura pines RC7/RX/DT y RC6/TX/CK).
0 = Puerto serie deshabilitado.
- Bit 6 RX9: Habilita el bit 9 de recepción:
1 = Selecciona recepción de 9 bits.
0 = Selecciona recepción de 8 bits.
- Bit 5 SREN: Configura la recepción sencilla:
Modo asíncrono no incluye.
Modo síncrono maestro:
1 = Habilita recepción sencilla.
0 = Deshabilita recepción sencilla.
Modo síncrono esclavo no se utiliza.
- Bit 4 CREN: Configura la recepción continua:
Modo asíncrono:
1 = Habilita modo de recepción continua.
0 = Deshabilita recepción continua.
Modo síncrono:
1 = Habilita recepción continua hasta que el bit CREN es borrado.
0 = Deshabilita recepción continua.
- Bit 3 ADDEN: Detección de dirección:
Modo asíncrono con 9 bits (RX9 = 1):
1=Activa la detección de dirección, activa la interrupción y descarga el buffer de recepción al activarse RSR.
0=Desactiva la detección de dirección, todos los bits son recibidos y el bit 9 puede ser utilizado como bit de paridad.
- Bit 2 FERR: Bit de error de trama:
1 = Error de trama (puede ser actualizado leyendo el registro RCREG y recibir el siguiente dato válido).
0 = No hay error de trama.
- Bit 1 OERR: Bit de error de sobrepasamiento:
1 = Error de sobrepasamiento (puede ser borrado escribiendo un 0 en el bit CREN).
0 = No hay error de sobrepasamiento.
- Bit 0 RX9D: Bit 9 del dato recibido (Puede ser el bit de paridad).

CAPÍTULO 3 COMPONENTES DEL SISTEMA

3.1 INTRODUCCIÓN

Los sensores son la interfaz con el mundo real, ellos recolectan la información necesaria y estos son monitoreados por la unidad de procesamiento central. Las plataformas pueden estar construidas en una forma modular de modo que una variedad de sensores puedan ser usados en la misma red.

La tecnología para el censado y control incluye sensores de campo eléctrico y campo magnético; sensores de radio frecuencia, sensores ópticos, opto-electrónicos e infrarrojos; radares, laceres; sensores de localización y navegación; sensores sísmicos; sensores ambientales (viento, humedad, calor); sensores bioquímicos. Los sensores de hoy pueden ser descritos como inteligentes, esto es, dispositivos baratos equipados con varios sensores, que son de bajo costo y bajo consumo.

3.2 SENSOR TEMPERATURA AMBIENTE

3.2.1 Introducción

Cualitativamente, la temperatura está asociada a la sensación de frío y calor. Sin embargo, para poder definir operacionalmente este concepto es necesario disponer de alguna propiedad termométrica, es decir, una propiedad que varíe con la temperatura. Luego, podemos definir la temperatura como la medida del contenido de calor de un cuerpo y calor es la energía calorífica que contienen los cuerpos.

En la actualidad existen numerosos tipos de termómetros o sensores de temperatura (Figura 3.1) que pueden ser usados para diferentes aplicaciones con los cuales se busca calibrar un termómetro o un sensor de temperatura respecto de un termómetro de referencia o patrón.

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

Figura 3.1.- Tipos de dispositivos para temperatura

Las aplicaciones para los LM35D incluyen casi cualquier tipo de temperatura que se encuentre por encima de los -55°C y por debajo de los $+150^{\circ}\text{C}$, cuyas características principales son:

- Factor de escala: $10\text{ mV}/^{\circ}\text{C}$ (garantizado entre 9.8 y $10.2\text{ mV}/^{\circ}\text{C}$)
- Rango de lectura de temperatura: $-55^{\circ}\text{C} < T < 150^{\circ}\text{C}$
- Precisión de: 1.5°C
- No linealidad: 0.5°C

Este integrado consta de 3 pines (Figura 3.3), uno de voltaje, el cual puede ser de 4 voltios a 20 voltios, este voltaje permite determinar la magnitud de la salida de voltaje del sensor, el pin del centro es la salida y esta puede variar de 0 a 1.5 milivoltios, variando en un rango de 10 milivoltios por grado centígrado de variación, para que este reflejo se de la entrada de voltaje deberá de estar por lo menos a 12 voltios. El otro pin del integrado va conectado a tierra (Figura 3.4).

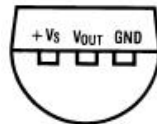


Figura 3.3.- Integrado LM35D

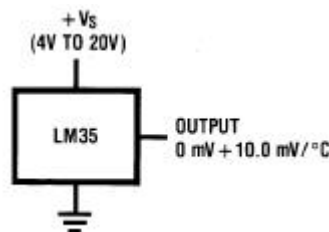


Figura 3.4.- Diagrama del sensor LM35D

El LM35D puede ser aplicado fácilmente usando algunos circuitos eléctricos, de la manera que se pueda registrar la temperatura correctamente, este sensor puede medir una temperatura de hasta 0.01 grado centígrado. Esto quiere decir que podemos medir la temperatura de cualquier nivel en un ambiente cerrado. Si la temperatura llegara a variar en algún momento debido a la circulación del aire, el sensor LM35D llega a registrar este margen de variación de temperatura.

3.3 SENSOR DE HUMEDAD AMBIENTE

3.3.1 Introducción

La humedad es un fenómeno natural, que se presenta a nivel molecular y se encuentra básicamente relacionada con la cantidad de moléculas de agua presentes en una determinada sustancia, la cual puede estar en estado sólido o gaseoso. Si bien el grado de concentración de agua en el ambiente, no influye mayormente sobre la vida normal de un ser humano (salvo en el confort), sabemos que sí resulta relevante para ciertos procesos, ya sean químicos, físicos o biológicos.

Podría decirse que la humedad juega un rol en todos los procesos industriales. El solo hecho de que la atmósfera contiene humedad hace que, por lo menos, se estudie su efecto en el almacenamiento y operación de los distintos productos y dispositivos. El alcance que la influencia de la humedad podría tener en cualquier proceso industrial puede variar pero es esencial que al menos sea monitoreada, y en muchos casos controlada. Se puede decir que la humedad es una propiedad más difícil de definir y medir que sus parámetros asociados como pueden ser la presión y temperatura. La medición de la humedad es un proceso verdaderamente analítico en el cual el sensor debe estar en contacto con el ambiente de proceso.

3.3.2 Medición de la humedad ambiente

El principio de funcionamiento esta basado en el cálculo de la Humedad Relativa que es una manera de expresar y cuantificar la cantidad de vapor de agua contenida en un determinado volumen de aire; por definición, se expresa en porcentaje (%) como la relación que hay entre la masa de agua contenida en el aire y la masa de agua que se requiere para producir la saturación del mismo aire. Cuando el aire está saturado, su contenido de agua es el máximo y se dice que su humedad relativa es también la máxima o del 100 %.

$$H_r = \frac{P_v}{P_{Sat}} \times 100\%$$

Tipos de Sensores de Humedad

- Sensor de polímero resistivo. Por su misma estructura este tipo de sensores son relativamente inmunes a la contaminación superficial ya que no afecta su precisión aunque si el tiempo de respuesta. Debido a los valores extremadamente altos de resistencia del sensor a niveles de humedad menores que 20% es apropiado para los rangos altos de humedad.
- Sensor de polímero capacitivo. Este tipo de sensor es especialmente apropiado para ambiente de alta temperatura porque el coeficiente de temperatura es bajo y el polímero dieléctrico puede soportar altas temperaturas. Estos sensores son también apropiados para aplicaciones que requieran un alto grado de sensibilidad a niveles bajos de humedad, donde proveen una respuesta relativamente rápida. A valores de humedad superiores al 85% sin embargo el sensor tiene una tendencia a saturar y se transforma en no lineal.
- Higrómetro óptico de condensación. Con el higrómetro óptico son posibles precisiones de +/- 0.2°. Ciertos equipos especiales pueden tener un rango completo desde -85° hasta casi 100° de punto de rocío. Los tiempos de respuesta son rápidos y la operación está relativamente libre de problemas de pérdida de calibración.
- Higrómetro electrolítico. El sensor electrolítico se utiliza en aplicaciones secas de hasta un máximo de 1000 PPM y es apropiado para el uso en procesos industriales tales como gases ultra-puros, química fina, y producción de circuitos integrados, etc.

Las aplicaciones típicas de los sensores de humedad son:

- Líneas de aire medicinal.
- Equipo electrónico refrigerado con líquido.
- Computadoras refrigeradas.
- Hornos de tratamiento térmico.
- Control ambiental de recintos.
- Secadores.
- Estándares de calibración de humedad Generadores de ozono.
- Líneas de aire seco.
- Hornos de fundición.
- Sistemas de transferencia de nitrógeno.
- Soldadura con gas inerte.

3.3.2 Sensor de Humedad HMZ-433

El HMZ-433A (Figura 3.5) es un módulo de detección de humedad y de temperatura que pueden ser instalados en una amplia gama de aplicaciones. Este módulo no requiere determinados componentes externos.

Algunas de las aplicaciones de este módulo son: sensor de Humedad para sistemas de aire acondicionado y monitorización, registradores de humedad, provee valores de humedad a través de una salida digital. Algunas de las características del sensor son:

- Es económico.
- Amplio rango de operación de la humedad.
- Entrega una salida lineal de voltaje para la lectura de humedad que va de 0 a 3.3 volts.
- Tiene un termistor integrado del tipo NTC para medir la temperatura.
- Tamaño reducido.
- Se alimenta con 5 voltios.
- Bajo consumo de corriente.
- No requiere componentes externos, y puede ser conectado directamente a un microcontrolador.

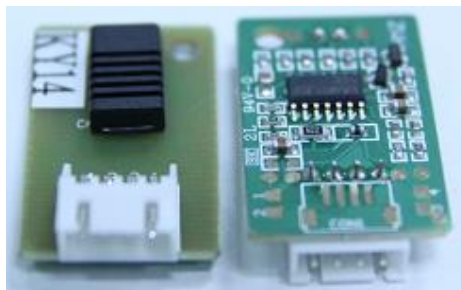
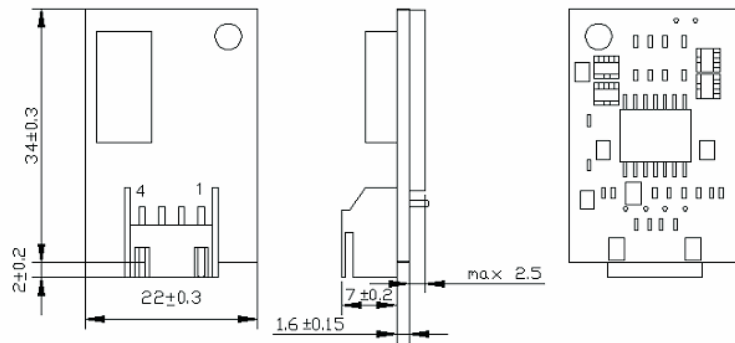


Figura 3.5.- Sensor HMZ-433

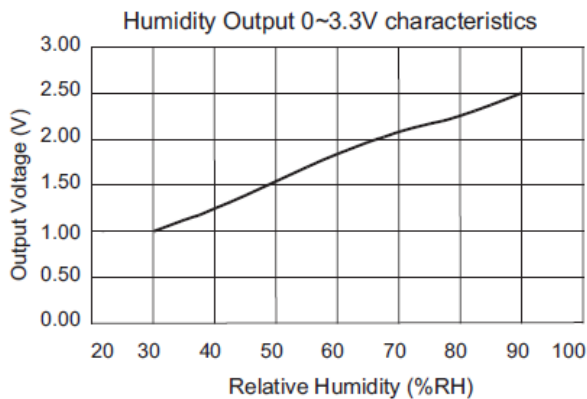
Sus dimensiones son:



La señal de salida esta dada por una relación de porcentaje de humedad contra una variación de voltaje:

% RH	30	40	50	60	70	80	90
V_o	1.00	1.32	1.65	1.98	2.31	2.64	2.97

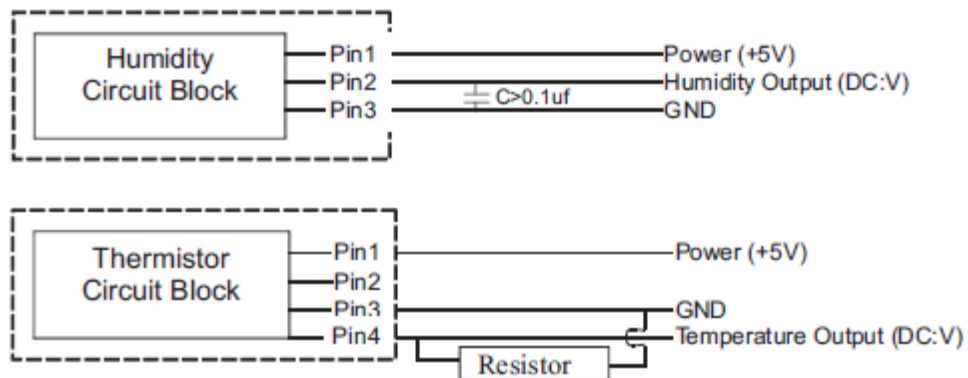
Cuya expresión grafica esta dada por:



	15°C	20°C	25°C	30°C	35°C	40°C
30% RH	1.23	1.09	1.02	0.94	0.83	0.78
40% RH	1.39	1.32	1.30	1.20	1.14	1.13
50% RH	1.65	1.65	1.64	1.60	1.58	1.62
60% RH	2.01	2.00	1.98	1.97	1.94	1.95
70% RH	2.28	2.21	2.25	2.22	2.15	2.16
80% RH	2.47	2.46	2.45	2.41	2.42	2.42
90% RH	2.82	2.77	2.73	2.71	2.70	2.67

HMZ Module Humidity Output 0~3.3V vs. Temperature Characteristics

Su circuito característico es:



3.4 SENSOR DE EFECTO HALL

3.4.1 Introducción

En el mercado existen una gran cantidad de sensores industriales para diversos usos, basados en el efecto que descubrió el científico Edwin Herbert Hall. El nombre de Hall, físico norteamericano, ha pasado a la posteridad debido a una singularidad electromagnética que descubrió por causalidad en el curso de un montaje eléctrico: el "Efecto Hall".

3.4.2 Efecto Hall

Cuando por una placa metálica circula una corriente eléctrica y ésta se halla situada en un campo magnético perpendicular a la dirección de la corriente, se desarrolla en la placa un campo eléctrico transversal, es decir, perpendicular al sentido de la corriente. Este campo, denominado Campo de Hall, es la resultante de fuerzas ejercidas por el campo magnético sobre las partículas de la corriente eléctrica, sean positivas o negativas (Figura 3.6).

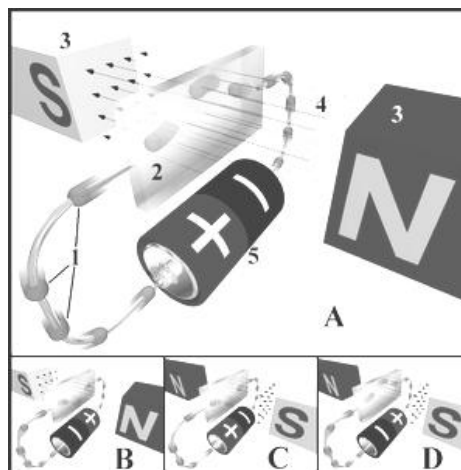


Figura 3.6.- Diagrama del efecto Hall, mostrando el flujo de electrones.

Leyendas:

1. Electrones.
2. Sensor o sonda Hall.
3. Imanes.
4. Campo magnético.
5. Fuente de energía.

Descripción:

En la Figura 3.6 A, una carga negativa aparece en el borde superior del sensor Hall, y una positiva en el borde inferior (color rojo). En B y C, el campo eléctrico o el magnético están invertidos, causando que la polaridad se invierta. Invertir tanto la corriente como el campo magnético (Figura 3.6 D) causa que la sonda asuma de nuevo una carga negativa en la esquina superior.

Este fenómeno tiene dos consecuencias principales. La primera es que la acumulación de cargas en un lado de la placa, en el campo así creado, implica que el otro lado tiene una carga opuesta, creándose entonces una diferencia de potencial; la segunda es que la carga positiva posee un potencial superior al de la carga negativa. La medida del potencial permite, por tanto, determinar si se trata de un campo positivo o negativo.

En la mayor parte de los metales, la carga es negativa, pero en algunos metales como el hierro, el zinc, el berilio y el cadmio son positivos, y en los semiconductores es positiva y negativa al mismo tiempo. Hay una desigualdad entre los intercambios negativos y los positivos; también en este caso, la medida del potencial permite saber cuál domina, el positivo o el negativo.

Los sensores basados en efecto Hall suelen constar de un elemento conductor o semiconductor y un imán. Cuando un objeto ferromagnético se aproxima al sensor, el campo que provoca el imán en el elemento se debilita. Así se puede determinar la proximidad de un objeto, siempre que sea ferromagnético.

3.4.3 Aplicaciones del sensor de efecto hall

Además de ser una herramienta básica para la caracterización eléctrica de materiales, muchas aplicaciones han surgido a partir del desarrollo de circuitos integrados basados en el efecto Hall y de amplia difusión en el mercado de la electrónica. Actualmente estas aplicaciones incluyen el uso en sistemas de ignición, controles de velocidad, sistemas de seguridad, controles de alineación, micrómetros, interruptores de límite mecánico, computadores, impresoras, unidades de disquete, teclados, herramientas de máquina, interruptores de clave, e interruptores de pulsador. Estos integrados también se usan como tacómetros, interruptores de límite de corriente, detectores de posición, interruptores de selección, sensores de corriente, potenciómetros lineales, y conmutadores de motores DC sin escobillas.

Los dispositivos de efecto Hall cuando son apropiadamente empaquetados son inmunes al polvo, suciedad, barro, y agua. Estas características hacen a los dispositivos de efecto Hall superiores, por ejemplo comparados con medios alternativos como sensores ópticos y electromecánicos para la lectura de posición.

3.4.4 Sensor Hall DN6851A

Este sensor DN6851 (Figura 3.7) es un circuito integrado que hace uso del efecto Hall (elemento que responde a cambios en el sentido y magnitud de un campo magnético) y un imán para su funcionamiento. Está diseñado especialmente para funcionar a una baja tensión de voltaje en la presencia del campo magnético. Es adecuado para su uso en diversos sensores e interruptores de contacto.

Sus características principales destacan:

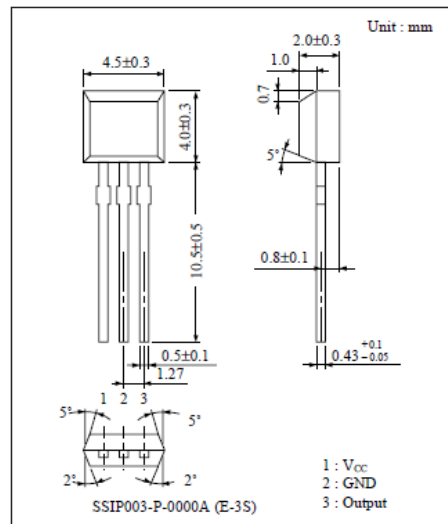
- Amplia gama en el voltaje de alimentación de 3.6 a 16v
- Opera en la presencia del campo magnético.

- Se pueden manejar directamente a la salida circuitos TTL y CMOS.
- La vida de servicio es semipermanente porque no hay partes de contacto.
- Inicia la conducción con el acercamiento de un pequeño imán.
- En su empaquetado contiene solo tres pines tipo SIP.
- Se caracteriza por tener una salida digital dependiente de la dirección del campo magnético.
- Al aplicarle un campo magnético el estado de la salida adquiere un nivel alto, en ausencia del campo magnético el estado de la salida adquiere un estado bajo.



Figura 3.7.- Sensor de efecto hall DN6851

Sus dimensiones son:



3.5 SENSOR PIR (Passive infrared sensor)

3.5.1 Introducción

El sensor piroeléctrico es un dispositivo que mide cambios en los niveles de radiación infrarroja emitida por los objetos a su alrededor. Está hecho de un material cristalino o cerámico, que genera una carga eléctrica en su superficie cuando se expone al calor en forma de radiación infrarroja. Cuando la temperatura del exterior cambia, la radiación infrarroja que recibe el PIR también lo hará, por tanto, la carga eléctrica sufrirá una variación que será detectada por un FET ubicado dentro del sensor.

Los elementos del sensor son sensitivos a un amplio rango de radiación infrarroja, por lo que se adhiere un filtro que limita el rango entre $5\mu\text{m}$ y $14\mu\text{m}$, en el cual, se ubica la radiación promedio de un cuerpo humano a 36 grados centígrados. Compuesto de dos

elementos que censan los cambios de radiación infrarroja, colocados en una polarización opuesta entre ellos (Figura 3.8), con el fin de anular las señales de interferencia causadas por la luz solar, vibraciones y variaciones de temperatura, mejorando en gran escala la estabilidad de funcionamiento del sensor.

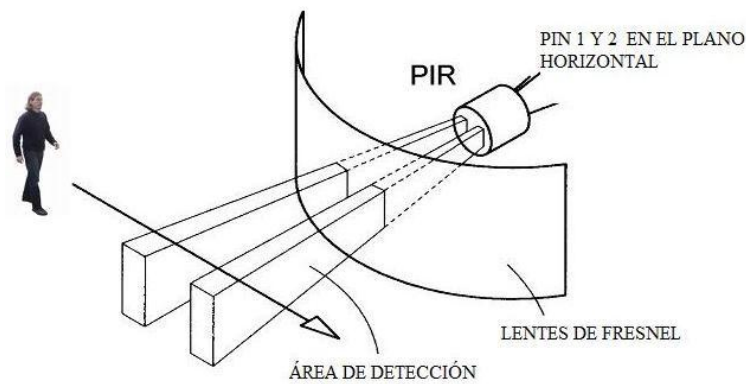


Figura 3.8.- Área de Detección del Sensor PIR

Cuando una persona pasa frente al sensor, cada uno de los elementos se activa en forma secuencial y generan un cambio de tensión eléctrica, el cual puede ser positivo o negativo dependiendo de la dirección en que se pase. Al ser opuestas las polaridades de los sensores, necesariamente los pulsos generados serán de diferente polaridad (Figura 3.9).

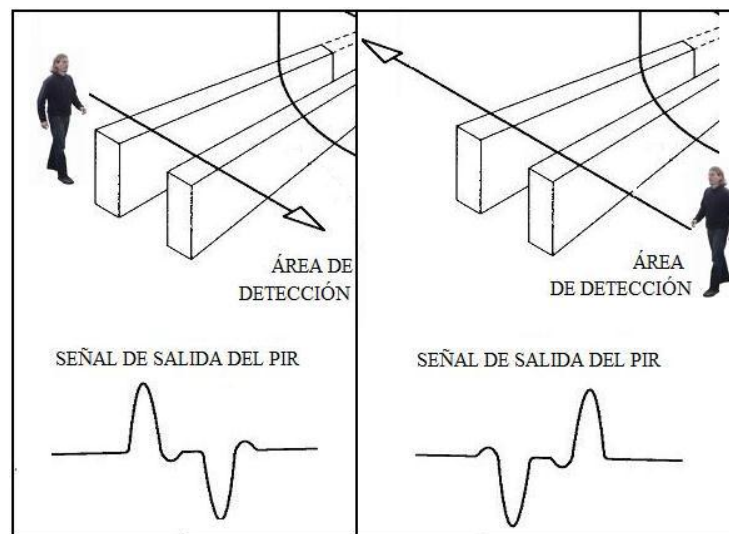


Figura 3.9.- Diferente polaridad del sensor PIR

A diferencia de otros sistemas, como los ultrasónicos o de microondas, los sensores PIR no emiten radiación; sino que, solo reciben "pasivamente" la radiación infrarroja proveniente de cuerpos a temperatura superior al ambiente (todo cuerpo caliente emite radiación infrarroja); además, presentan gran ahorro de energía y protegen el medio ambiente.

Los sensores PIR son, al igual que los infrarrojos, de una muy buena confiabilidad, ya que actúan siempre en conjunto con programas especializados que los hace más precisos; pero, los primeros son de bajo costo, reducido tamaño y si bien su implementación es más complicada, es mejor, comparado con el costo final de un contador de personas elaborado a través de la tecnología infrarroja, por ejemplo el DILAX.

3.5.2 Lente de Fresnel Infrarrojo

Son lentes de vidrio o plástico tallados o fabricados de tal forma que, cuando esté atravesado por un rayo de luz, se comporte de la misma forma que un lente plano convexo. Cada rayo que ingresa al lente cambia su dirección enfocándose en una región específica, obteniendo más direccionalidad y potencia. En la figura 3.10 se muestra como los rayos de una fuente de luz que están dispersos, se conglomeran después del paso a través de un lente de Fresnel:

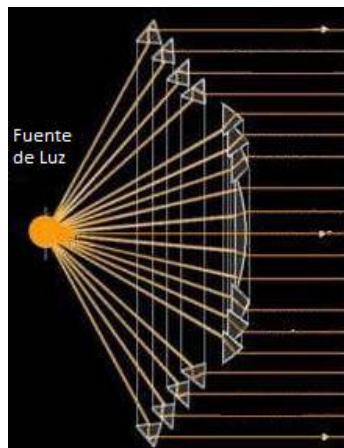


Figura 3.10.- Comportamiento de la Luz a través del Lente de Fresnel

Su diseño permite la construcción de lentes de gran apertura y una corta distancia focal, sin el peso y tamaño de una lente convencional, gracias a que se pueden mantener los radios de curvatura de las lentes, separándolas en anillos circulares concéntricos consecutivos, con lo cual el grosor de la lente en cada anillo es diferente, eliminando el espesor que tuviera si fuera una superficie continua.

Todos los lentes de Fresnel se componen de un conjunto de elementos reflectantes y refringentes (lentes, prismas y anillos) por medio de los cuales los rayos emitidos por una fuente de luz se desvían en la dirección deseada.

Si el lente de Fresnel se basa en la reflexión, constituye un catoptrio. Son espejos parabólicos que producen un haz reflejado de rayos paralelos, es decir un haz unidireccional, o bien espejos elíptico-parabólicos en que los haces reflejados sólo son paralelos en determinadas direcciones (figura 3.11).

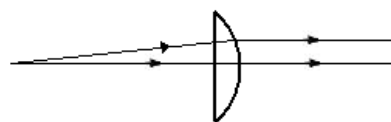


Figura 3.11.- Espejo Catoptrio

Si el lente de Fresnel se basa en una doble refracción, constituye un dioptrio (figura 3.12).



Figura 3.12 Espejo Dioptrio

Si a través del prisma se producen dos refracciones y una reflexión total, constituye un catadioptrio (figura 3.13).

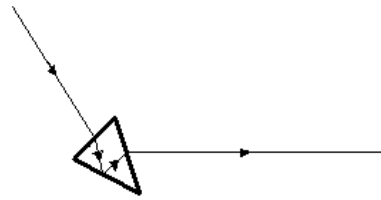


Figura 3.13.- Espejo Catadioptrio

Los lentes de Fresnel son diseñados teniendo muy en cuenta la aplicación en los que se le van a usar, el lente utilizado en el presente proyecto está compuesto de espejos dióptricos y catóptricos que concentrarán la luz en un solo punto focal.

Existen en el mercado una amplia gama de lentes de Fresnel para diferentes aplicaciones tales como, proyectores, detectores de presencia, aparatos para los vehículos, etc. Son elementos pasivos, es decir no necesitan de energía para funcionar. En este proyecto se utilizaran los lentes de Fresnel conjuntamente con los sensores PIR, creando un determinado ángulo de detección, dividiendo y amplificando, el haz de los sensores en zonas; existen dos alternativas de lentes que podemos ocupar, los lentes para sensores PIR, y los lentes planos. Se ha escogido los lentes planos por la facilidad de obtención, ya que se encuentran con cualquier distribuidor, debido a que los lentes para sensores PIR que solo se encuentran en el mercado extranjero bajo pedido.

Los lentes de Fresnel planos son flexibles, hechos de un material plástico de alta densidad, de color blanco lechoso. Al trabajar con lentes de Fresnel planos, no existe un tamaño estándar; el tamaño depende de la aplicación en la que se va a usar. En la figura 3.14 se muestra en gran escala la forma de un lente de Fresnel:



Figura 3.14.- Lente de Fresnel Plana

Los sensores PIR por su propia cuenta, no podrían detectar la presencia y movimiento de una persona que esté más lejos de 60 cm del contador. Para dar mayor alcance a los sensores, se utiliza un arreglo de lentes de Fresnel; que permite tener un haz de cobertura más directivo. Los lentes de Fresnel también permiten ajustar el haz de los sensores para adecuarlos según la aplicación a utilizarlos; por ejemplo, para detectar una persona a la vez es necesario que el haz de cobertura de los sensores sea muy directivo.

El haz de un sensor PIR es cónico (figura 3.15a) y abarca todo el diámetro posible, teniendo un gran campo de cobertura que no se puede controlar fácilmente; ya que está expuesto a censar personas que estén circundando el lugar a pesar de no haber atravesado el haz de cobertura del sensor PIR, incrementándose de manera significativa la

probabilidad de error. Con los lentes de Fresnel, el haz se reduce notablemente y la efectividad de las mediciones realizadas mejora en una proporción directa a la directividad del haz, como se muestra en la Figura. 3.15b.

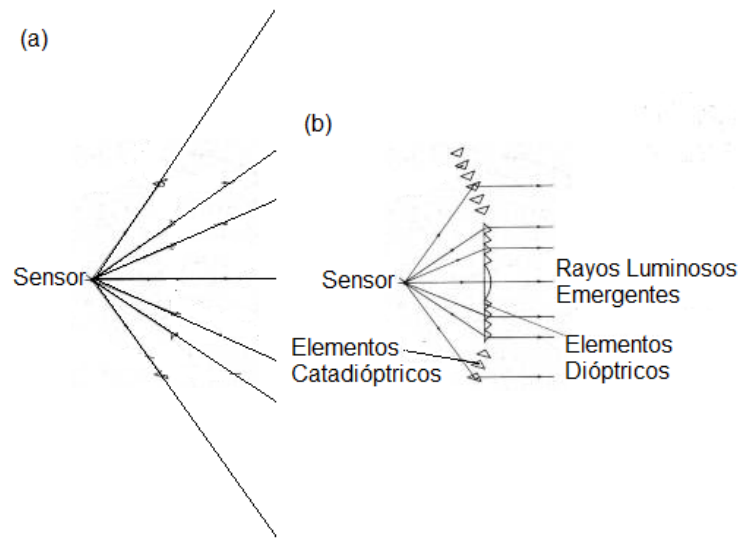


Figura. 3.15.- (a) Haz del sensor PIR sin los lentes de Fresnel. (b) Haz del sensor PIR acoplado a un lente de Fresnel

3.5.3 Sensor movimiento PIR SE-10

Este es un sencillo sensor de movimiento (Figura 3.16), es un dispositivo piroeléctrico que mide cambios en los niveles de radiación infrarroja emitida por los objetos a su alrededor a una distancia máxima de 6 metros. Como respuesta al movimiento, el sensor cambia el nivel lógico de un “pin”, por lo cual, es extremadamente simple. Adicionalmente es un sensor de bajo costo y reducido tamaño muy utilizado en sistemas de alarmas, iluminación controlada por movimiento y aplicaciones de robótica.



Figura 3.16.- Sensor SE-10

El sensor PIR cuenta con solamente tres terminales. Dos de ellos se utilizan para la alimentación y el restante es para la salida de detección de movimiento (el cable rojo es alimentación, el café es tierra GND y el negro es la alarma en colector abierto). El pin de alarma es de colector abierto, lo que implica que necesitará una resistencia de pull-up en ese pin. Múltiples sensores pueden ser conectados a un solo pin de entrada debido a esta configuración.

Al energizarse el sensor PIR requiere de un tiempo de preparación para comenzar a operar de forma adecuada. Esto se debe a que tiene que ocurrir la adaptación a las condiciones propias de operación del ambiente donde se instalara. Durante este periodo el sensor reconoce el estado de reposo o no movimiento del ambiente. La duración puede estar entre 1 o 2 segundos para que el sensor se calibre al espacio sin movimiento, es altamente recomendable la ausencia de personas. Si las señales infrarrojas del ambiente cambian o se mueven después de ese período, el pin de "alarma" cambiará a 0. Esta unidad funciona bien a 12v. También se puede puentear el regulador de 5v del sensor para hacerlo funcionar a 3.3v. El sensor utiliza 1.6mA a 3.3v.

Sus dimensiones son:

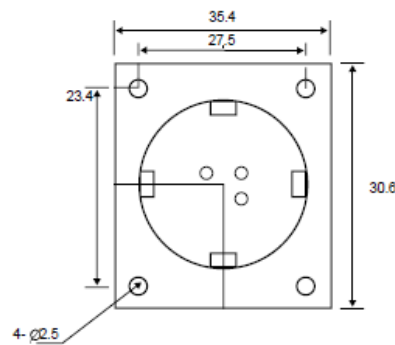


Figura 3.17.- Dimensiones del sensor PIR.

3.6 FOTORRESISTENCIA

3.6.1 Introducción

Una fotorresistencia, LDR o fotocelda, es un componente electrónico cuya resistencia disminuye con el aumento de la intensidad de luz incidente. Puede también ser llamado fotorresistor, fotoconductor, célula fotoeléctrica o resistor dependiente de la luz, cuyas siglas LDR se originan de su nombre en inglés *light-dependent-resistor*. Su cuerpo está formado por una célula o celda y dos patillas.

El valor de la resistencia eléctrica de un LDR es bajo cuando hay luz incidiendo en ella (puede descender hasta 50 ohms) y muy alto cuando está a oscuras (varios megaohms).

Un fotorresistor está hecho de un semiconductor de alta resistencia como el sulfuro de cadmio (CdS). Si la luz que incide en el dispositivo es de alta frecuencia, los fotones son absorbidos por la elasticidad del semiconductor dando a los electrones la suficiente energía para saltar la banda de conducción. El electrón libre que resulta y su hueco asociado conducen la electricidad, de tal modo que disminuye la resistencia. Los valores típicos varían entre 1 MΩ o más en la oscuridad y 100Ω con luz brillante.

Las células de sulfuro del cadmio se basan en la capacidad del cadmio de variar su resistencia según la cantidad de luz que incide la célula. Cuanta más luz incide, más baja

es la resistencia. Las células son también capaces de reaccionar a una amplia gama de frecuencias, incluyendo infrarrojo (IR), luz visible, y ultravioleta (UV).

La variación del valor de la resistencia tiene cierto retardo, diferente si se pasa de oscuro a iluminado o de iluminado a oscuro. Esto limita a no usar los LDR en aplicaciones en las que la señal luminosa varía con rapidez. El tiempo de respuesta típico de un LDR está en el orden de una décima de segundo. Esta lentitud da ventaja en algunas aplicaciones, ya que se filtran variaciones rápidas de iluminación que podrían hacer inestable un sensor (ej. tubo fluorescente alimentado por corriente alterna). En otras aplicaciones (saber si es de día o es de noche) la lentitud de la detección no es importante.

3.6.2 Circuito Representativo

Las LDR's son resistores que varían su valor de acuerdo a la intensidad de la luz, razón por la cual se trata de un sensor analógico, es decir, que siempre toma valores distintos, por lo cual no se puede tomar un valor lógico 1 o 0 como en lógica digital. La fotocelda en total oscuridad puede llegar a tomar valores de 1M si no es que más, y a plena iluminación unos pocos kΩ o quizás menos. Lo que se puede hacer, es un arreglo entre la fotocelda al polo (-) y una resistencia fija al polo (+), de esa manera el punto de unión entre estos dos componentes podrá tomar dos valores según la variación de la LDR, señal que se puede utilizar como salida del sensor, este tipo de circuitos es conocido como divisor de tensión.

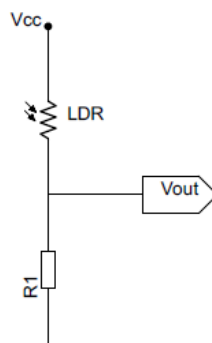


Figura 3.18.- Circuito divisor de tensión.

El voltaje que se tiene a la salida está dado por la ecuación del divisor de voltaje:

$$V_{OUT} = \left(\frac{R_1}{R_1 + LDR} \right) V_{CC}$$

La señal de la Figura 3.18 aun sigue siendo analógica, y para convertirla en señal digital se dispone a utilizar un disparador Schmitt como el HD74LS14P para convertir la salida a digital (Figura 3.19).

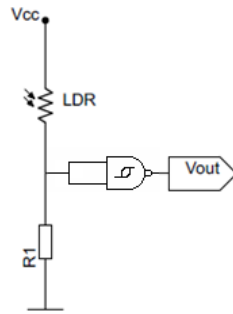


Figura 3.19.- Salida digital con LDR

3.7 RELEVADOR

Un relevador es un dispositivo electromecánico. Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes (Figura 3.20).

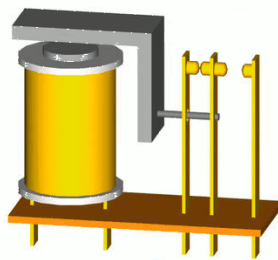


Figura 3.20.- Relevador

Un relé se representa de forma esquemática con la disposición de los distintos elementos (Figura 3.21) que forman un relé de un único contacto de trabajo y con su circuito representativo (Figura 3.22).

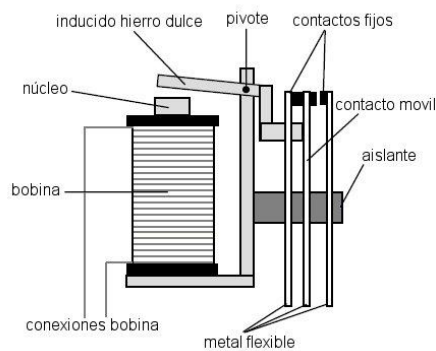


Figura 3.21.- Partes de de un relé

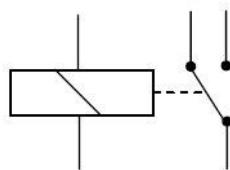


Figura 3.22.- Símbolo eléctrico de un relé de 1 circuito.

3.7.1 Tipos de relés

Existen multitud de tipos distintos de relés (Figura 3.23), dependiendo del número de contactos, de la intensidad admisible por los mismos, tipo de corriente de accionamiento, tiempo de activación y desactivación, etc. Cuando controlan grandes potencias se les llama contactores en lugar de relés.

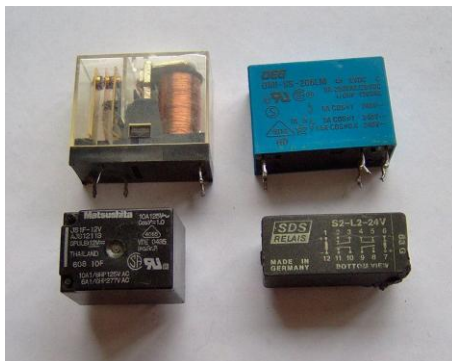


Figura 3.23.- Tipos de relés

3.7.1.1 Relés electromecánicos

- Relés de tipo armadura: pese a ser los más antiguos siguen siendo lo más utilizados en multitud de aplicaciones. Un electroimán provoca la basculación de una armadura al ser excitado, cerrando o abriendo los contactos dependiendo de si es NA o NC.
- Relés de núcleo móvil: a diferencia del anterior modelo estos están formados por un émbolo en lugar de una armadura. Debido su mayor fuerza de atracción, se utiliza un solenoide para cerrar sus contactos. Es muy utilizado cuando hay que controlar altas corrientes.
- Relé tipo reed o de lengüeta: están constituidos por una ampolla de vidrio, con contactos en su interior, montados sobre delgadas láminas de metal. Estos contactos conmutan por la excitación de una bobina, que se encuentra alrededor de la mencionada ampolla.
- Relés polarizados o biestables: se componen de una pequeña armadura, solidaria a un imán permanente. El extremo inferior gira dentro de los polos de un electroimán, mientras que el otro lleva una cabeza de contacto. Al excitar el electroimán, se mueve la armadura y provoca el cierre de los contactos. Si se polariza al revés, el giro será en sentido contrario, abriendo los contactos ó cerrando otro circuito.

3.7.1.2 Relé de estado sólido

Se llama relé de estado sólido a un circuito híbrido (Figura 3.24), normalmente compuesto por un optoacoplador que aísla la entrada, un circuito de disparo, que detecta el paso por cero de la corriente de línea y un triac o dispositivo similar que actúa de interruptor de potencia. Su nombre se debe a la similitud que presenta con un relé electromecánico; este dispositivo es usado generalmente para aplicaciones donde se presenta un uso continuo de los contactos del relé que en comparación con un relé convencional generaría un serio desgaste mecánico, además de poder conmutar altos amperajes que en el caso del relé electromecánico destruirían en poco tiempo los contactos. Estos relés permiten una velocidad de conmutación muy superior a la de los relés electromecánicos.



Figura 3.24.- Relé de estado sólido

3.7.1.3 Relé de corriente alterna

Cuando se excita la bobina de un relé con corriente alterna, el flujo magnético en el circuito magnético, también es alterno, produciendo una fuerza pulsante, con frecuencia doble, sobre los contactos. Es decir, los contactos de un relé conectado a la red, en algunos lugares, como varios países de Europa y Latinoamérica oscilarán a 50 Hz y en otros, como en Estados Unidos lo harán a 60 Hz. Este hecho se aprovecha en algunos timbres y zumbadores, como un activador a distancia. En un relé de corriente alterna se modifica la resonancia de los contactos para que no oscilen.

3.7.1.4 Relé de láminas

Este tipo de relé se utilizaba para discriminar distintas frecuencias. Consiste en un electroimán excitado con la corriente alterna de entrada que atrae varias varillas sintonizadas para resonar a ciertas frecuencias de interés. La varilla que resuena acciona su contacto; las demás, no. Los relés de láminas se utilizaron en aeromodelismo y otros sistemas de telecontrol.

3.7.2 Ventajas del uso de relés

La gran ventaja de los relés electromagnéticos es la completa separación eléctrica entre la corriente de accionamiento, la que circula por la bobina del electroimán, y los circuitos controlados por los contactos, lo que hace que se puedan manejar altos voltajes o

elevadas potencias con pequeñas tensiones de control. También ofrecen la posibilidad de control de un dispositivo a distancia mediante el uso de pequeñas señales de control.

En el caso presentado podemos ver un grupo de relés que son controlados por módulos digitales programables que permiten crear funciones de temporización y contador como si se tratase de un miniPLC. Con estos modernos sistemas los relés pueden actuar de forma programada e independiente, lo que supone grandes ventajas en su aplicación aumentando su uso en aplicaciones sin necesidad de utilizar controles como PLC's u otros medios para comandarlos (Figura 3.25).

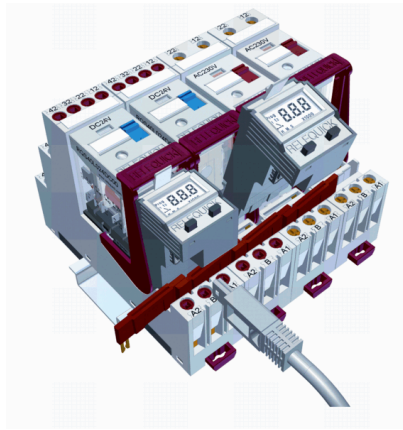


Figura 3.25.- Relé interface con módulo programable

3.8 LCD

3.8.1 Introducción

Antes de aparecer los módulos LCD, se utilizaban los displays de siete segmentos para poder mostrar la información. Tenían una gran limitación de poder mostrar los caracteres alfa numéricos y símbolos especiales, también consumían demasiada corriente y ocupaban demasiado espacio físico. Posteriormente aparecieron otros tipos de displays mas complejos que podían mostrar algunos caracteres y símbolos; pero tenían de igual manera mucho consumo de corriente y espacio físico, ocupaban también bastante espacio físico. Finalmente aparecieron los módulos LCD o pantallas de cristal líquido la cual tiene la capacidad de mostrar cualquier carácter alfa numérico. Estos dispositivos cuentan con una pantalla y la lógica de control preprogramada de fábrica y lo mejor de todo es que el consumo de corriente es mínimo y no hace falta realizar tablas especiales como se hacía anteriormente con los displays de siete segmentos. Las aplicaciones de los módulos LCD son infinitas ya que pueden ser aplicados en la informática, comunicaciones, telefonía, instrumentación, robótica, automóviles, equipos industriales, etc.

Para poder hacer funcionar un LCD, debe de estar conectado a un circuito impreso en el que estén integrados los controladores del display y los pines para la conexión del display. Sobre el circuito impreso se encuentra el LCD en sí, rodeado por una estructura metálica que lo protege. Tiene un consumo de energía de menos de 5mA y son ideales para dispositivos que requieran una visualización pequeña o media (Figura 3.26).

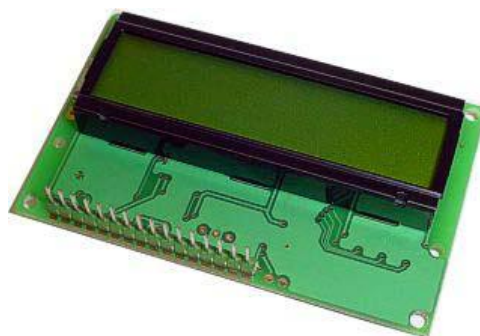


Figura 3.26.- Aspecto físico de un LCD

3.8.2 Los caracteres del LCD

El LCD dispone de una matriz de 5x8 puntos para representar cada carácter. En total se pueden representar 256 caracteres diferentes, 240 caracteres están grabados dentro del LCD y representan las letras mayúsculas, minúsculas, signos de puntuación, números, etc. Existen 8 caracteres que pueden ser definidos por el usuario (Figura 3.27).

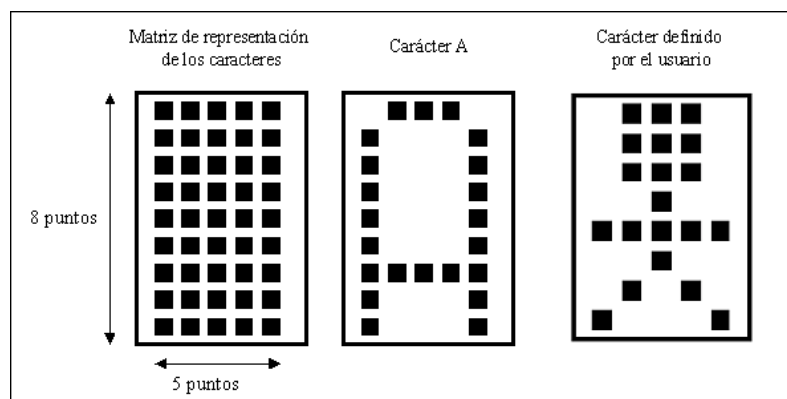


Figura 3.27.- Matriz de puntos

3.8.3 Diversidad de módulos LCD

En la actualidad existen una gran variedad de versiones, clasificadas en dos grupos. El primer grupo está referido a los módulos LCD de caracteres (solamente se podrán presentar caracteres y símbolos especiales en las líneas predefinidas en el modulo LCD) y el segundo grupo está referido a los módulos LCD matriciales (se podrán presentar caracteres, símbolos especiales y gráficos). Los módulos LCD varían su tamaño físico dependiendo de la marca, por lo tanto en la actualidad no existe un tamaño estándar para los módulos LCD.

3.8.4 DD RAM (Display Data Ram)

Es la memoria encargada de almacenar los caracteres de la pantalla que se estén visualizando en ese momento, o bien, que estén en una posición no visible. El display tiene una capacidad de 2 líneas horizontales por 40 caracteres cada una, de los cuales solo serán visibles 2 líneas de 16 caracteres cada una. La DDRAM tiene un tamaño de

2x40=80 bytes. Una vez conocida la disposición de almacenamiento del display, es fácil pensar en un display de 2 líneas de 40 caracteres sobre el que se desplaza una ventana de 2 líneas por 16 caracteres.

3.8.5 CG RAM (caracter generator RAM)

La CG RAM contiene los caracteres que pueden ser definidos por el usuario, es decir que pueden ser personalizados. Está formada por 64 posiciones, con direcciones \$00-\$3F. Cada posición es de 5 bits. La memoria está dividida en 8 bloques que corresponden a los posibles caracteres creados por el usuario que van del 0 al 7 (Figura 3.28).

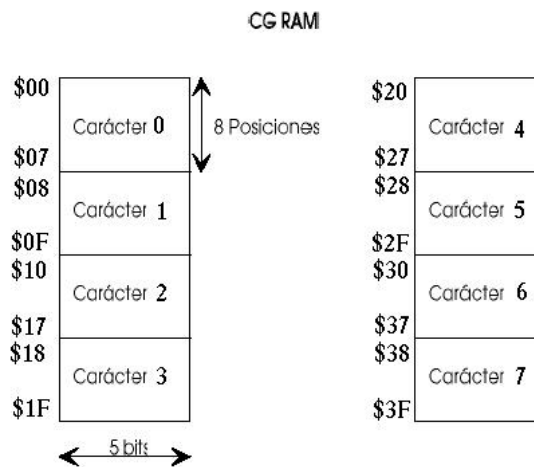


Figura 3.28.- Memoria CGRAM

Cada carácter está constituido por una matriz de 5 columnas por 8 filas. De este modo un 1 indica un punto de la matriz encendido y un 0 apagado (Figura 3.29).

Carácter definido por el usuario	Valores a almacenar en la CG RAM	
	BINARIO	HEXADECIMAL
	01110	\$0B
	01110	\$0B
	01110	\$0B
	00100	\$04
	11111	\$1F
	00100	\$04
	01010	\$0A
	10001	\$11

Figura 3.29.- Carácter definido por el usuario.

A continuación si quisiéramos almacenar este carácter en la posición 0 de la CG RAM, tendríamos que almacenar cada una de las líneas de 5 bits en las direcciones de la CG RAM que van de la \$00 hasta la \$07 como se observo en el mapa de memoria CG RAM.

3.8.6 Asignación de pines

A continuación podemos ver la asignación de pines en una pantalla LCD

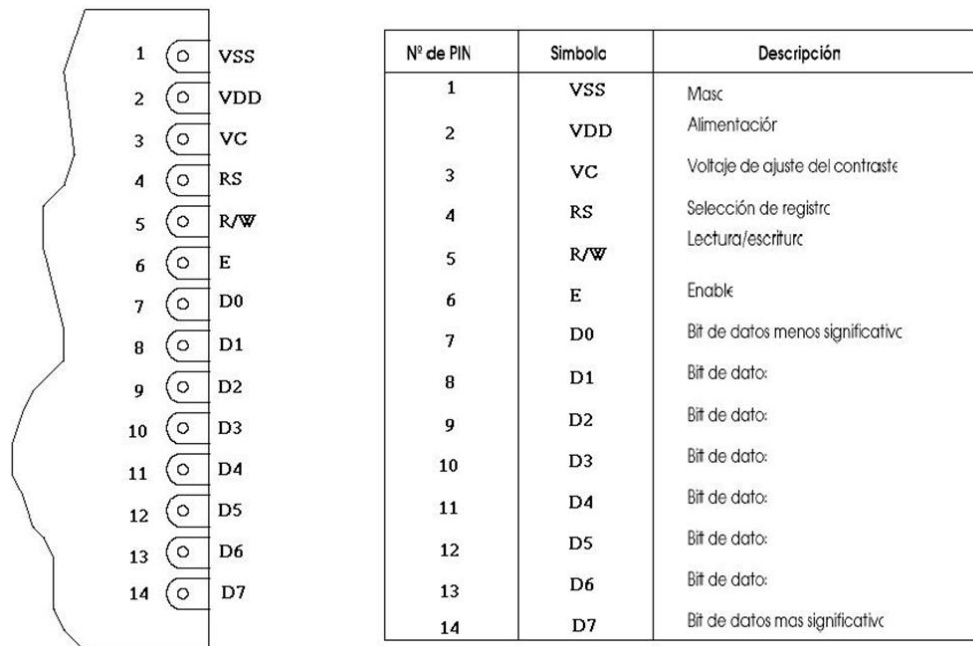


Figura 3.30.- Asignación de pines

Los pines 1 y 2, son los utilizados para la alimentación del módulo LCD. La tensión utilizada es de 5 voltios. El pin 3 se utiliza para ajustar el contraste de la pantalla LCD por medio de un potenciómetro que regula la intensidad de los caracteres (a mayor tensión mayor intensidad). El pin 4 se utiliza para indicar al bus de datos si la información que le llega es una instrucción o por el contrario si es un carácter. Si RS=0 indicara que en el bus de datos hay presente una instrucción, y si RD=1, indicará que tiene un carácter alfanumérico.

El pin 5 es el de escritura o lectura. Si esta a 0 el módulo escribe en pantalla el dato que se encuentre en el bus de datos, y si esta a 1 leeremos lo que se tenga en el bus de datos. El pin 6 es el indicado de hacer que el módulo LCD funcione, o por el contrario no acepte órdenes de funcionamiento. Cuando E=0 no se podrá utilizar el display y cuando E=1 se podrán transferir datos y realizar las demás operaciones. Las pines del 7 al 14 son los del bus de datos.

CAPÍTULO 4 COMUNICACIÓN INALÁMBRICA XBEE

4.1 MÓDULO XBEE

4.1.1 Introducción

Xbee es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas IEEE_802.15.4. Es especialmente útil para redes de sensores en entornos industriales, médicos y, sobre todo, domóticos. Las comunicaciones Xbee se realizan en la banda libre de 2.4GHz. A diferencia de bluetooth, este protocolo no utiliza FHSS (Frequency hopping), sino que realiza las comunicaciones a través de una única frecuencia, es decir, de un canal.

Normalmente puede escogerse un canal de entre 16 posibles. El alcance depende de la potencia de transmisión del dispositivo así como también del tipo de antenas utilizadas (cerámicas, dipolos, etc), el alcance normal con antena dipolo en línea vista es de aproximadamente (tomando como ejemplo el caso de MaxStream, en la versión de 1mW de potencia) de 100m y en interiores de unos 30m. La velocidad de transmisión de datos de una red Xbee es de hasta 256kbps. Una red Xbee la pueden formar, teóricamente, hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad enorme de dispositivos.

Entre las necesidades que satisface el módulo se encuentran:

- Bajo costo.
- Ultra-bajo consumo de potencia.
- Uso de bandas de radio libres y sin necesidad de licencias.
- Instalación barata y simple.
- Redes flexibles y extensibles.

El uso del protocolo Xbee va desde reemplazar un cable por una comunicación serial inalámbrica, hasta el desarrollo de configuraciones punto a punto, multipunto, nodo a nodo o redes complejas de sensores. Una conexión típica (Figura 4.1) es donde los datos son enviados a través de una red a un centro que administra la información.

Una red Xbee esta conformada principalmente por un Coordinador, un router y un dispositivo final (End Point).

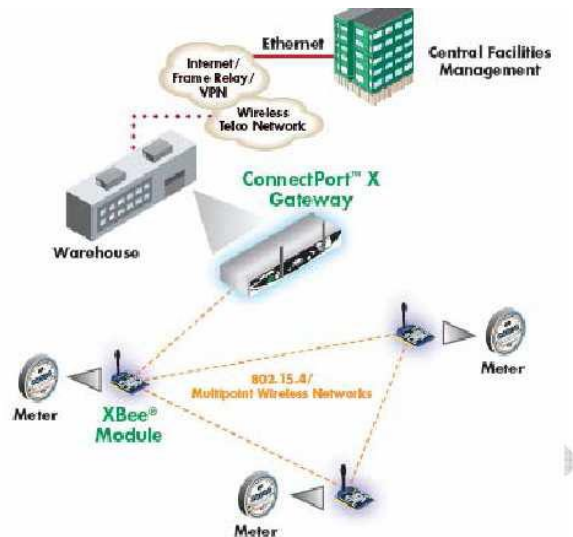


Figura 4.1.- Conexión típica de una red con Xbee.

4.1.2 Comunicación inalámbrica

Las comunicaciones inalámbricas son aquellas que se propagan en condiciones de espacio libre, por medio de ondas electromagnéticas, por lo tanto este tipo de red carece de cables. Entre algunas de las ventajas que proporcionan los sistemas radioelectrónicos, se pueden mencionar los siguientes:

- La capacidad de cruzar zonas de agua, en donde una instalación con cable requeriría de un trato especial para prevenir las filtraciones sobre los conductores de cobre.
- La capacidad de vencer obstáculos en la transmisión causados por la presencia de montañas o valles.
- La capacidad de evitar la interconexión al proveedor de telefonía local. Una conexión privada para evitar la ocupación de la central del operador local.

Debido a la gran variedad de aplicaciones de las comunicaciones inalámbricas, a las redes inalámbricas se les puede clasificar como (Figura 4.2): Redes de Área Personal (PAN), Redes de Área Local (LAN), Red de Área Metropolitana (MAN), Redes de Área Extensa (WAN), Wirelees BAN o Red de Área Corporal.

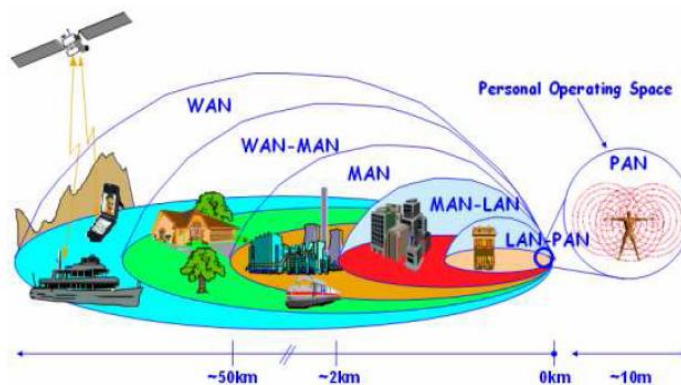


Figura 4.2.- Clasificación de las redes de acuerdo al área de cobertura

- Wireless BAN (Body Area Network)

Es una red de soporte de micro dispositivos, distribuidos por el cuerpo humano como cámaras, micrófonos, sensores, entre otros (Figura 4.3), sus características son bajo consumo de energía y baja capacidad. Entre sus aplicaciones podemos mencionar las siguientes: reconocimiento automático del usuario y configuración del entorno como redes en el hogar, redes de carros o redes de área personal (PAN); entornos inteligentes; ropa inteligente.

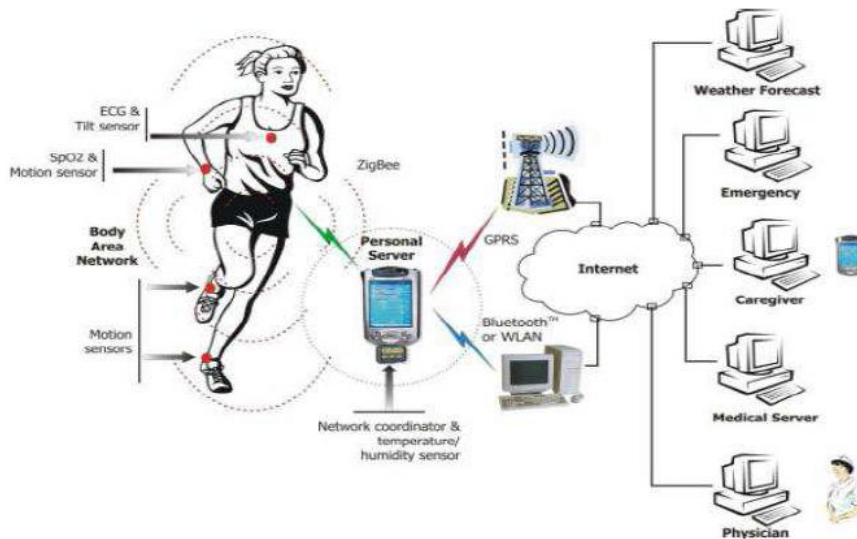


Figura 4.3.- Red de Área Corporal

- Wireless PAN (Personal Área Network)

Este tipo de redes tiene una cobertura entre 1 y 10 metros. Esta red se establece alrededor de personas y permite la comunicación entre dispositivos de uso personal (Figura 4.4). Permite la comunicación en un área reducida, la conexión puede ser con dispositivos propios o con otros dispositivos del entorno.

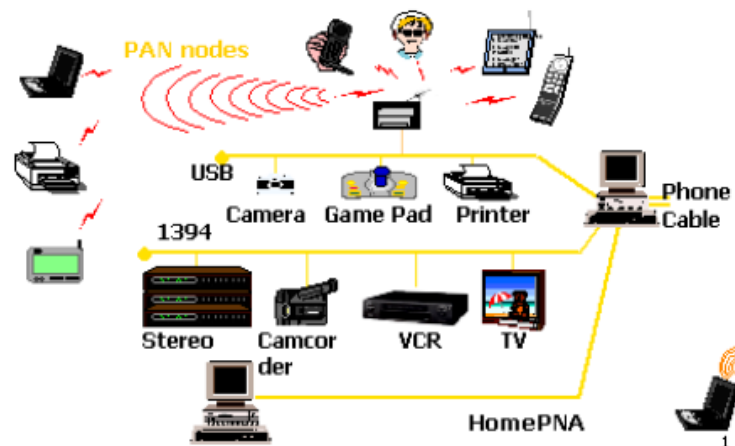


Figura 4.4.- Red de Área Personal

- Wireless LAN (Local Área Network)

Este tipo de redes tiene una cobertura de entre 100 a 500 metros. Son redes corporativas de acceso a datos (Figura 4.5). Permiten el acceso a estaciones móviles mediante tecnología inalámbrica utilizando puntos de acceso. Requieren de capacidad y conectividad al igual que las LAN.

Entre sus características especiales tenemos: Requiere un punto de acceso por celda; acceso inalámbrico; seguridad reducida; proporciona movilidad y el ancho de banda es reducido.



Figura 4.5.- Red Inalámbrica de Área Local

- Wireless WAN (Wide Area Network), Red Satelital

Un sistema de comunicaciones por satélites está conformado por estaciones terrenas y por uno o varios satélites los cuales se encuentran en órbita alrededor de la Tierra (Figura 4.6). La estación terrena, es una instalación diseñada para transmitir señales a un satélite o recibir señales de éste.



Figura 4.6.- Red Satelital

La transmisión de la estación terrena hacia el satélite se conoce como enlace ascendente, mientras que la transmisión desde el satélite hacia la estación terrena se denomina enlace descendente. Las características que se deben tomar en cuenta en las comunicaciones por satélite son el retardo, el ruido y el ancho de banda limitado. Una ventaja de la utilización de satélites es alcanzar zonas geográficas remotas.

4.1.3 Estándar IEEE 802.15.4

IEEE 802.15.4 es un estándar que define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos (low-rate wireless personal área network, LR-WPAN)

También es la base sobre la que se define la especificación de XBee, cuyo propósito es ofrecer una solución completa para este tipo de redes construyendo los niveles superiores de la pila de protocolos que el estándar no cubre.

El propósito del estándar es definir los niveles de red básicos para dar servicio a un tipo específico de red inalámbrica de área personal (WPAN) centrada en la habilitación de comunicación entre dispositivos con bajo coste y velocidad. Se enfatiza el bajo coste de comunicación con nodos cercanos y sin infraestructura o con muy poca, para favorecer aún más el bajo consumo.

En su forma básica se concibe un área de comunicación de 10 metros con una tasa de transferencia de 250Kbps. Se pueden realizar compromisos que favorezcan aproximaciones más radicales a los sistemas empotrados con requerimientos de consumo aún menores. Para ello se definen no uno, sino varios niveles físicos. Se definieron inicialmente tasas alternativas de 20 y 40Kbps; la versión actual añade una tasa adicional de 100Kbps. Se pueden lograr tasas aún menores con la consiguiente reducción de consumo de energía.

Como se ha indicado, la característica fundamental de 802.15.4 entre las WPAN's es la obtención de costes de fabricación excepcionalmente bajos por medio de la sencillez tecnológica, sin perjuicio de la generalidad o la adaptabilidad. Entre los aspectos más importantes se encuentra la adecuación de su uso para tiempo real por medio de slots de tiempo garantizados, evitación de colisiones por CSMA/CA y soporte integrado a las comunicaciones seguras. También se incluyen funciones de control del consumo de energía como calidad del enlace y detección de energía.

Un dispositivo que implementa el 802.15.4 puede transmitir en una de tres posibles bandas de frecuencia 2.4 GHz, 915 MHz y 868 MHz. Cada banda de frecuencia ofrece un determinado número de canales, la banda de frecuencia de 2.4 GHz ofrece 16 canales (11-26), 915 MHz ofrece 10 canales (1- 10) y 868 MHz ofrece un canal (0). La tasa de transmisión de bits de este estándar depende de la frecuencia de operación. La banda de 2.4 GHz provee 250 Kbps, 915 MHz provee 40 Kbps y 868 MHz provee 20 Kbps de datos de transmisión. La cantidad real de datos que pueden fluir en un determinado intervalo de tiempo (throughput) es menor que la tasa nominal de datos debido al empaquetamiento, encabezado y procesos de retardo. La máxima longitud de una trama IEEE 802.15.4 es de 127 bytes.

4.2 TIPOS DE DISPOSITIVOS EN UNA RED XBEE

Se definen tres tipos distintos de dispositivos Xbee según su papel en una red.

4.2.1 El Coordinador

Es el nodo de la red que tiene la única función de formar una red. Es el responsable de establecer el canal de conexión y del PAN ID (identificador de red) para toda la red. Una vez establecidos estos parámetros, el Coordinador puede formar una red, permitiendo unirse a él a dispositivos Routers y End Points. Una vez formada la red, el Coordinador hace las funciones de Router, esto es, participar en el enrutado de paquetes y ser origen y/o destinatario de información.

4.2.2 Los Routers

Es un nodo que crea y mantiene conexión sobre la red para determinar la mejor ruta para transmitir un paquete de información, lógicamente un router debe unirse a una red Xbee antes de poder actuar como Router retransmitiendo paquetes de otros routers o de End points.

4.2.3 Los End Device

Los dispositivos finales no tienen capacidad de enrutar paquetes. Deben interactuar siempre a través de un nodo, ya sea con un coordinador o un router, es decir, no puede enviar conexión directamente a otro end device.

4.3 TOPOLOGÍAS DE RED XBEE

La capa de red soporta múltiples configuraciones de red incluyendo estrella, árbol, malla y mesh.

4.3.1 Topología en Estrella

La configuración de red en estrella consta de un coordinador y uno o más dispositivos finales (Figura 4.7). En una red en estrella, todos los dispositivos finales se comunican únicamente con el coordinador. Si un dispositivo final necesita transferir datos a otro dispositivo final, este envía sus datos al coordinador. El coordinador selecciona y reenvía los datos al nodo destinatario.

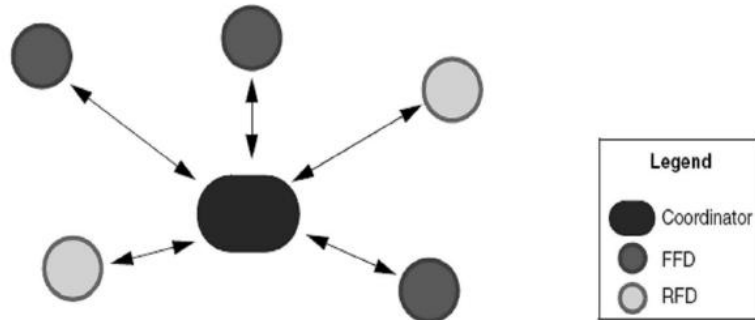


Figura 4.7.- Configuración de red en estrella

En la configuración en estrella, uno de los dispositivos Xbee asume el rol de coordinador de red y es responsable de inicializar y mantener los dispositivos en la red. Todos los demás dispositivos Xbee, conocidos con el nombre de dispositivos finales, hablan directamente con el coordinador.

4.3.2 Topología en Árbol

En esta configuración, los dispositivos finales pueden unirse unos con otros por medio de coordinadores o ruteadores (Figura 4.8). Los ruteadores ofrecen dos funciones. Uno es el incremento del número de nodos que puedan estar en la red. La otra es la ampliación física del rango de alcance de la red. Con el ingreso de un ruteador, un dispositivo final no necesita estar a un rango determinado de radio del coordinador. Todos los mensajes en la topología en árbol se enrutarían a lo largo del árbol.

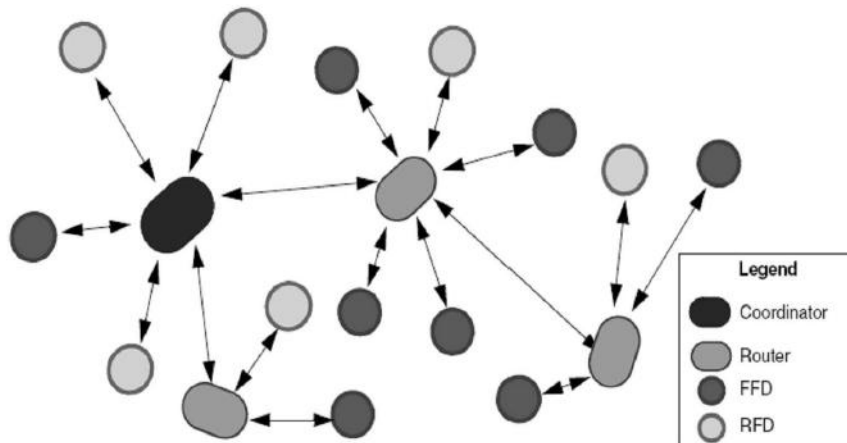


Figura 4.8.- Configuración de red en árbol

4.3.3 Topología en Malla.

Una red en malla es similar a la red ramas de árbol, excepto que los FFDs pueden comunicarse directamente con otros FFDs, lo que no ocurre en la topología anterior. La ventaja de esta topología es que puede reducirse la latencia e incrementarse la confiabilidad (Figura 4.9).

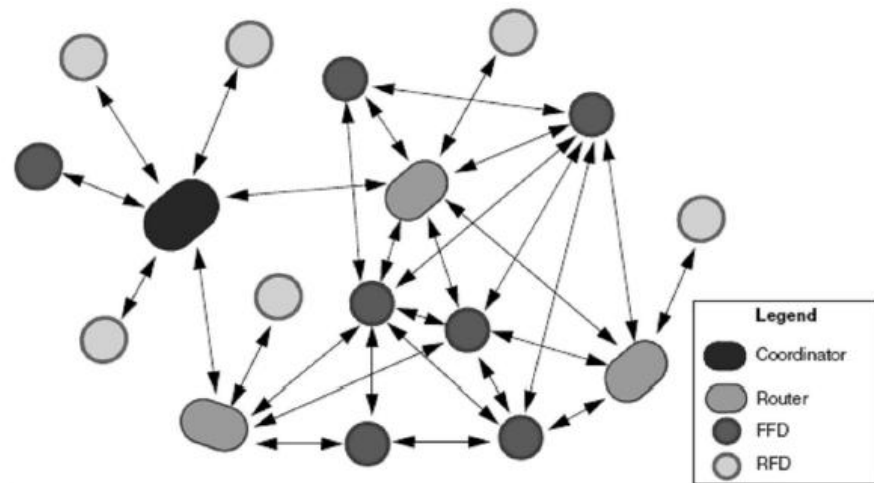


Figura 4.9.- Conexión de red en malla

Las topologías de red en árbol y malla se las conocen también como redes multisalto, debido a las habilidades de enrutar los paquetes a través de múltiples dispositivos, mientras que la topología en estrella se la conoce como red simple salto, debido a que el paquete realiza un único salto para llegar a su destino. La red XBee es una red multiacceso, principalmente porque todos los nodos en la red tienen igual acceso al medio de comunicación.

4.4 CONEXIONES

Cada módulo Xbee tiene una dirección única de 64bits que viene grabada de fábrica. Por otro lado, la red Xbee, utiliza para sus algoritmos de ruteo direcciones de 16 bits. Cada vez que un dispositivo se asocia a una red Xbee, el coordinador al cual se asocia le asigna una dirección única en toda la red de 16bits. Por eso el número máximo teórico de elementos que puede haber en una red Xbee es de $2^{16} = 65535$, que es el número máximo de direcciones de red que se pueden asignar. Estos módulos Xbee, pueden ser ajustados para usarse en redes de conexión punto a punto, punto-a-multipunto o nodo a nodo. Un ejemplo se muestra (Figura 4.10) una conexión multipunto, con un coordinador, conectado a varios nodos.

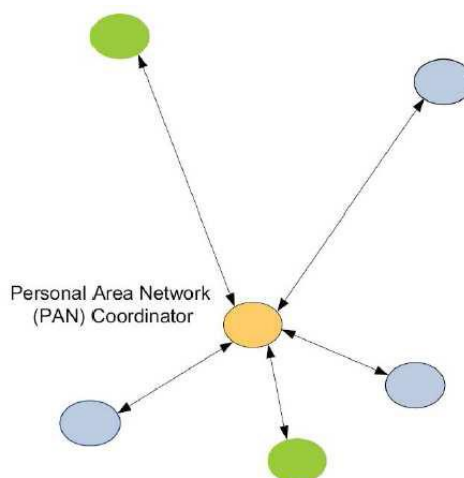


Figura 4.10.- Coordinador PAN con múltiples nodos

4.4.1 Red Mesh

Con los módulos Xbee PRO de la Serie2, es posible crear redes más complejas, como las llamadas MESH. Estas permiten acceder a un punto remoto, utilizando módulos intermedios para llegar como routers. Además los módulos automáticamente generaran la red entre ellos, sin intervención humana alguna, permitiendo la reparación de la red en caso de que algún nodo falle. Al mismo tiempo la red por sí sola resuelve la mejor ruta para un determinado paquete.

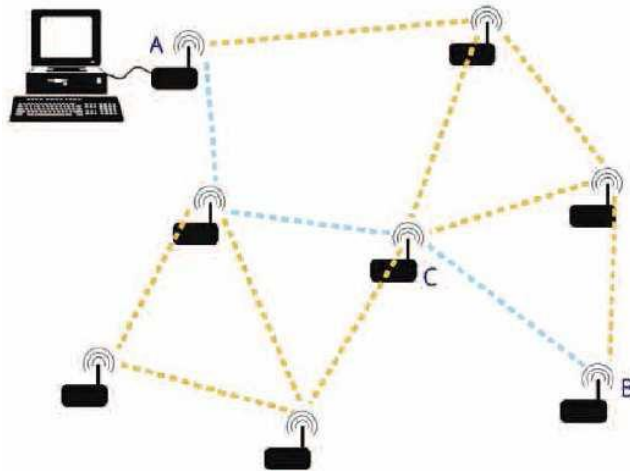


Figura 4.11.- Red Mech para módulos Xbee PRO Serie 2.

La figura anterior (Figura 4.11) muestra un ejemplo de una red MESH. Se observa que se desea acceder al punto B a partir del punto A. Suponiendo que la distancia entre A y B es demasiado para que alcance la señal, se utiliza la red MESH para poder alcanzarla, así cada nodo ubicado en medio del camino, funciona como módulo transparente, donde todo lo que le llega es retransmitido hacia el punto A. Esto funciona para cada uno de los módulos de la red.

4.4.2 Circuito básico para el Xbee.

La Figura 4.12 muestra las conexiones mínimas que necesita el módulo Xbee para poder ser utilizado. Luego de esto, se debe configurar según el modo de operación adecuado para la aplicación requerida por el usuario.

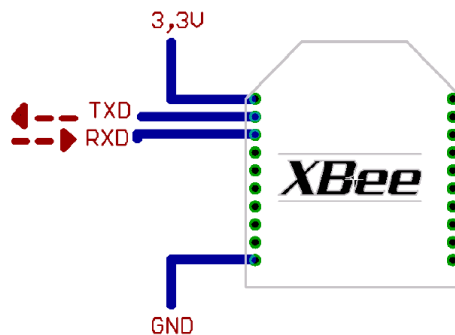


Figura 4.12.- Conexiones mínimas requeridas para el Xbee.

El módulo requiere una alimentación desde 2.8 a 3.4 V, la conexión a tierra y las líneas de transmisión de datos por medio del UART (TXD y RXD) para comunicarse con un microcontrolador, o directamente a un puerto serial utilizando algún conversor adecuado para los niveles de voltaje.

4.4.3 Modos de Operación.

Los módulos Xbee, pueden operar en los siguientes 6 modos:

1.- Modo recibir / transmitir.-Se encuentra en estos modos cuando el módulo recibe algún paquete RF a través de la antena (modo Receive) o cuando se envía información serial al buffer del pin 3 (UART Data in) que luego será transmitida (modo Transmit). La información transmitida puede ser Directa o Indirecta. En el modo directo la información se envía inmediatamente a la dirección de destino. En el modo Indirecto la información es retenida por el módulo durante un período de tiempo y es enviada sólo cuando la dirección de destino la solicita. Además es posible enviar información por dos formas diferentes. Unicast y Broadcast. Por la primera, la comunicación es desde un punto a otro, y es el único modo que permite respuesta de quien recibe el paquete RF, es decir, quien recibe debe enviar un ACK (paquete llamado así, y que indica que recibió el paquete, el usuario no puede verlo, es interno de los módulos) a la dirección de origen. Quien envió el paquete, espera recibir un ACK, en caso de que no le llegue, reenviará el paquete hasta 3 veces o hasta que reciba el ACK. En el modo Broadcast la comunicación es entre un nodo y a todos los nodos de la red. En este modo no hay confirmación por ACK.

2.- Modo de Bajo Consumo (Sleep Mode).- El modo de Bajo Consumo o modo Sleep, corresponde cuando el módulo entra en un estado de bajo consumo de energía. Esto depende de la configuración en la que se encuentra. Se debe indicar que el ahorro de energía depende enormemente del voltaje de alimentación.

Para entrar a este modo se debe configurar el comando SM (ATSM – Sleep Mode). Si SM=4 o SM=5, el módulo entra al modo SLEEP cuando transcurre un período dado por el comando ST (ATST-Time Before Sleeping-tiempo antes de dormir). Donde ST posee un rango entre 0 y 0xFFFF (x 1ms). Si SM=4, el módem despertará cada cierto tiempo, dado por SP (ATSP – Cyclic Sleep Period), cuyo rango es entre 0 y 0x68B0 (x10 ms). Cuando despierte buscará por datos entrantes desde un módulo configurado como Coordinador, en caso de no haber nada, el módem volverá al modo SLEEP. Si SM=5, el módulo realizará lo mismo que antes, pero despertará de acuerdo al estado del pin SLEEP_RQ (pin 9).

Cuando el pin pase a LOW (el pin es disparado por detección de borde, no por nivel), despertará, buscará por datos entrantes, y si no hay nada volverá a cero el timer del comando ST, por lo que sólo volverá a dormir cuando transcurra el período dado por éste. Mientras transcurra ese período, cualquier actividad en el pin SLEEP_RQ será ignorada, hasta que vuelva al modo SLEEP. Si SM=1, el módulo entrará al estado de hibernación. En este estado, si el pin SLEEP_RQ está en HIGH, el módulo cortará cualquier actividad entrante, ya sea de transmisión, recepción o de asociación (se verá más adelante) y entrará al modo SLEEP, y no saldrá de ahí hasta que el pin SLEEP_RQ vuelva a estar en LOW.

Mientras se encuentre en el estado de hibernación, no responderá a ninguna actividad serial ni a ningún paquete RF entrante, simplemente los desechará. En el modo

hibernación el sistema ahorra una gran cantidad de potencia. Si SM=2, el sistema se comporta igual que el modo de hibernación, pero sin tanto ahorro de energía y además despierta mucho más rápido que el modo anterior.

3.- Modo de Comando.- Este modo permite ingresar comandos AT al módulo Xbee, para configurar, ajustar o modificar parámetros. Permite ajustar parámetros como la dirección propia o la de destino, así como su modo de operación entre otras cosas.

4.- Modo Transparente.- En este modo todo lo que ingresa por el pin 3 (Data in) (Figura 4.13), es guardado en el buffer de entrada y luego transmitido y todo lo que ingresa como paquete RF, es guardado en el buffer de salida y luego enviado por el pin 2 (Data out). El modo Transparente viene por defecto en los módulos Xbee. Este modo está destinado principalmente a la comunicación punto a punto, donde no es necesario ningún tipo de control.

También se usa para reemplazar alguna conexión serial por cable, ya que es la configuración más sencilla posible y no requiere una mayor configuración. En este modo, la información es recibida por el pin 3 del módulo Xbee, y guardada en el buffer de entrada.

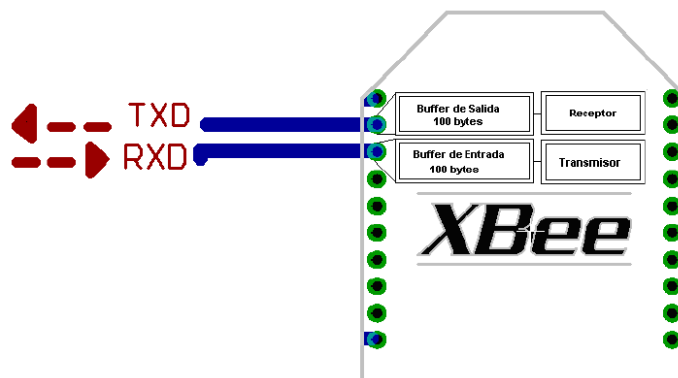


Figura 4.13.- Correspondencia de Buffer.

5.- Modo de operación API.- Este modo es más complejo, pero permite el uso de frames con cabeceras que aseguran la transmisión de los datos, al estilo TCP. Extiende el nivel en el cual la aplicación del cliente, puede interactuar con las capacidades de red del módulo.

Cuando el módulo Xbee se encuentra en este modo, toda la información que entra y sale, es empaquetada en frames, que definen operaciones y eventos dentro del módulo. Así, un frame de Transmisión de Información (información recibida por el pin 3 o DIN) incluye:

- Frame de información RF transmitida.
- Frame de comandos (equivalente a comandos AT).

Mientras que un Frame de Recepción de Información incluye:

- Frame de información RF recibida.
- Comando de respuesta.
- Notificaciones de eventos como Reset, _ ODEM_ do, Disassociate, etc.

Esta API, provee alternativas para la configuración del módulo y ruteo de la información en la capa de aplicación del cliente. Un cliente puede enviar información al módulo Xbee. Estos datos serán contenidos en un frame cuya cabecera tendrá información útil referente el módulo. Esta información además se podrá configurar, esto es, en vez de estar usando el modo de comandos para modificar las direcciones, la API lo realiza automáticamente. El módulo así enviará paquetes de datos contenidos en frames a otros módulos de destino, con información a sus respectivas aplicaciones, conteniendo paquetes de estado, así como el origen, RSSI (potencia de la señal de recepción) e información de la carga útil de los paquetes recibidos. Entre las opciones que permite la API, se tienen:

- Transmitir información a múltiples destinatarios, sin entrar al modo de Comandos.
- Recibir estado de éxito/falla de cada paquete RF transmitido.
- Identificar la dirección de origen de cada paquete recibido.

6.- IDLE.- Cuando el módulo no se está en ninguno de los otros modos, se encuentra en éste. Es decir, si no está ni transmitiendo ni recibiendo, ni ahorrando energía ni en el modo de comandos, entonces se dice que se encuentra en un estado al que se le llama IDLE.

4.5 CONFIGURACIÓN DE LOS XBEE

La aplicación X-CTU que provee la empresa Digi, para la configuración y prueba de los productos X-Bee, es una aplicación basada en Windows, hecha para interactuar con los productos que contienen archivos con firmware encontrados en los productos RF de Digi.

El software puede ser descargado de la página del fabricante, una vez hecho esto procedemos a una instalación sencilla. Para realizar la programación se requerirá el modulo de programación para la X-BEE (Figura 4.14).

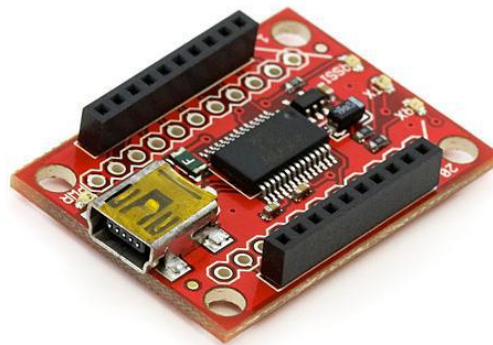


Figura 4.14.- Programador para X-BEE

4.5.1 Características del software.

Una vez instalado el software nos encontraremos con cuatro pestañas que nos permitirán manipular fácilmente nuestro producto X-Bee.

4.5.1.1 PC Settings.

Esta pestaña permite seleccionar al usuario el puerto de comunicaciones serial deseado y la configuración de este.

- Configuración del puerto Serial.

Aquí configuramos el puerto serial seleccionado y establecemos los parámetros de la comunicación como son:

1. Baud Rate: la velocidad a la cual nos vamos a comunicar.
2. Flow Control: Si queremos establecer un control de flujo ya sea por Hardware, por software o ninguno.
3. Parity: Si queremos poner un bit de paridad.
4. Stop Bit: si queremos un bit de parada.

Una vez configurado podemos probar la configuración con el botón “Test/Query”, y si esta correcta se recibe un mensaje similar a la de la Figura 4.15.

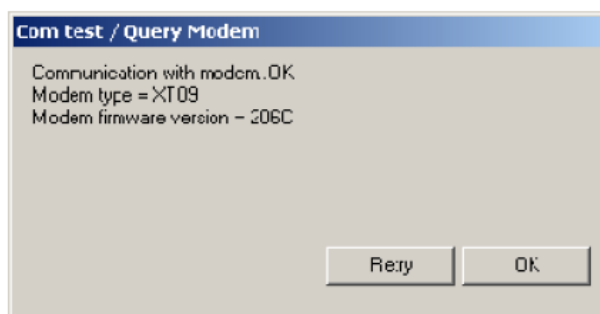


Figura 4.15.- Mensaje Test Query

4.5.1.2 Configuración del Host.

La configuración del host le permite al usuario configurar como el programa X-CTU interactúa con el firmware de los módulos de RF. Esto incluye determinar en que modo se desea trabajar, si en modo de comandos AT o modo de comandos API. O API mode: seleccione esta opción si tiene un modulo configurado en modo API, o si desea empezar a configurar un modulo con esta opción. Podremos agregar un puerto de comunicaciones pero solo es de uso temporal una vez cerrado el programa este desaparecerá.

4.5.1.3 Range Test.

Esta pestaña permite verificar el rango del radio enlace enviando un paquete de datos especificado por el usuario y verificando la respuesta del paquete mismo con el tiempo especificado. Por defecto el tamaño del paquete de datos enviado es de 32 bytes (Figura 4.16). Este tamaño del paquete de datos puede ser cambiado al igual que lo que se envía.

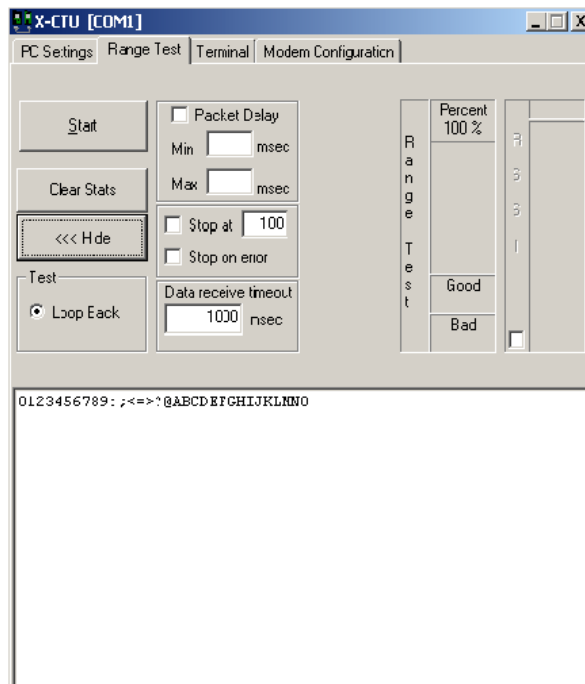


Figura 4.16.- Pestaña Range Test

Para cambiar el tamaño del paquete lo podemos hacer en el campo al lado del botón “Create Data”, para cambiar la información enviada, lo podemos hacer en la ventana de transmisión. En esta pestaña también podremos cambiar el retraso de los paquetes y el “timeout” para la recepción, lo cual le permite al usuario simular un amplio rango de escenarios.

Al habilitar la opción del RSSI (Received Signal Strength Indicator), nos permite ver con cuanto nivel de señal nos llega el paquete. Otro detalle importante es que si en la pestaña de “PC Settings” habilitamos la función de comandos API, podremos probar también estos parámetros en la pestaña de “Range Test”. Para realizar una prueba en este modo debemos haber configurado previamente otro dispositivo en modo API.

4.5.1.4 Terminal.

La pestaña terminal tiene tres funciones básicas: Un simulador de una terminal, similar al “hyperterminal” de Windows. Habilitar el envío y recepción de paquetes de datos predefinidos (Assemble Packet). Habilitar el envío y recepción de datos en formatos hexadecimal o ASCII (Show/Hide Hex) (Figura 4.17).

La porción en blanco de la pestaña terminal es donde habilitamos los comandos de comunicación y donde además leeremos los datos recibidos por el modulo. El texto en azul es lo que se copia y se envía al modulo de RF, el texto en rojo son los datos que nos envía el modulo por el puerto serial. La opción de “Assemble Packet” permite al usuario ensamblar un paquete de datos en formato ASCII o HEX.

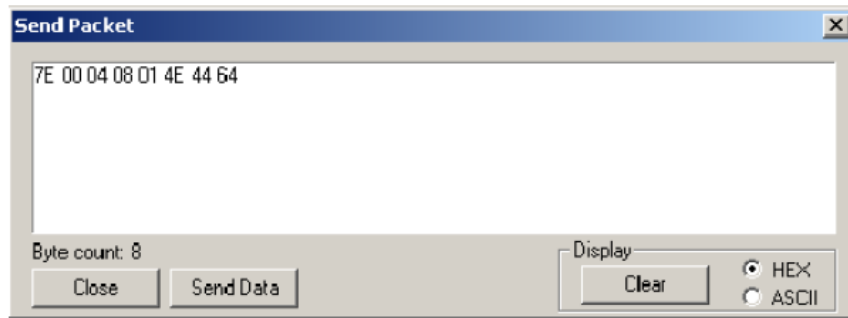


Figura 4.17.- Opción Assemble Packet

Esto hace que no se envíen inmediatamente los valores al modulo a medida que se van escribiendo, sino que primero creamos todo el paquete y luego lo enviamos, también es muy útil cuando estamos trabajando con el modulo en modo API.

4.5.1.5 Configuración de modem.

La pestaña de configuración (Figura 4.18) tiene cuatro funciones básicas:

1. Proveer una interfaz grafica del usuario con el Firmware.
2. Leer y escribir el firmware al micro controlador del módulo.
3. Descargar archivos actualizados del firmware para los módulos.
4. Grabar o cargar un perfil de modem.

Podremos leer el firmware que tiene el módulo, presionando el botón “Read” una vez tengamos conectado el modulo. Si no se lee intente habilitar el modo API en la pestaña “PC settings” y luego intente “leer” nuevamente el firmware. Una vez leídos los parámetros del firmware estos serán mostrados en tres diferentes colores.

- Negro: De solo lectura, no configurables.
- Verde: Valores por defecto.
- Azul: Valores puestos por el usuario.

Antes de cambiar cualquier valor del firmware lo mejor es actualizarlo a través del botón “Download new versions”, lo podremos hacer desde la web o desde un archivo previamente descargado. Una vez actualizado el firmware si podremos hacer los cambios deseados.

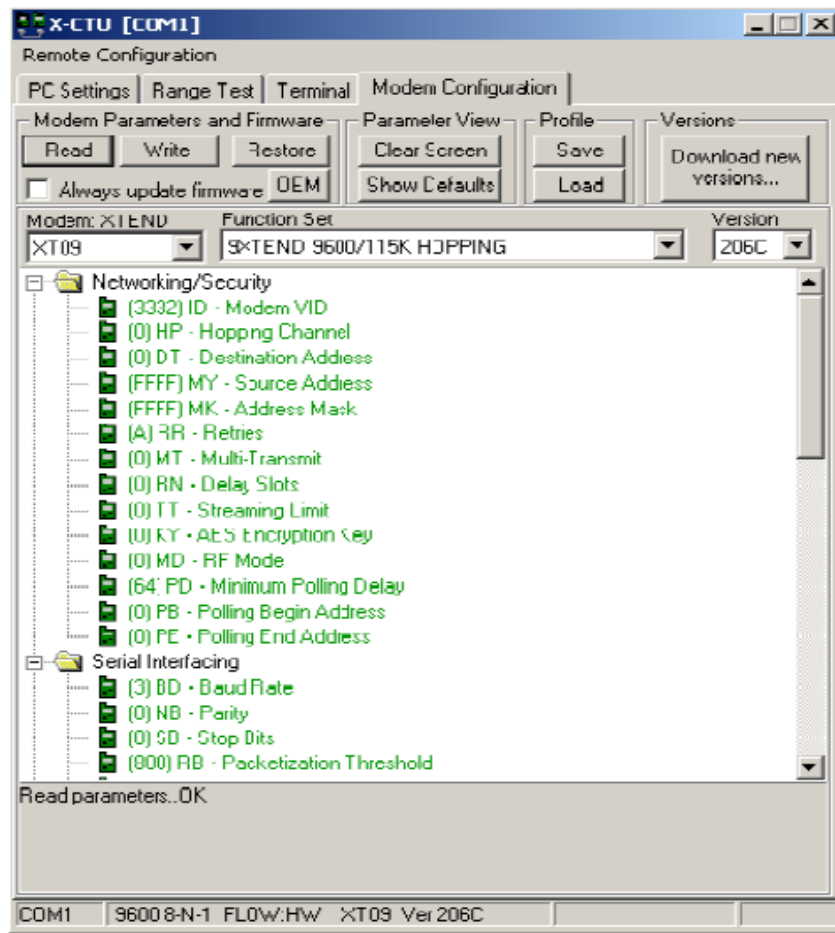


Figura 4.18.- Configuración de Modem

Una vez analizado el software para configurar pasaremos a analizar otros conceptos propios de los módulos Zigbee de Digi.

4.6 Tipos de dispositivos X-BEE

El estándar de Xbee define tres tipos de dispositivos (Figura 4.19).

- Xbee Coordinator: Encargado de establecer el canal y el PAN-ID (Personal Area Network Identifier). Puede permitir a los router y a los end devices unirse a la red. Puede asistir en el enrutamiento de datos. No puede dormirse. Puede Almacenar datos de los hijos que están dormidos.
- Xbee Router: Este debe unirse a una PAN antes de empezar a transmitir, enrutar o recibir datos. Después de unirse a la PAN permite que otros routers y end devices se unan a la red. Puede asistir en el enrutamiento de paquetes. Puede almacenar datos de otros dispositivos dormidos.
- Xbee end device: Debe unirse a una PAN antes de empezar a transmitir datos. No puede dejar que otros dispositivos se unan a la red. Puede enviar y recibir datos a través de sus padres. Puede entrar en modo Sleep para ahorro de energía.

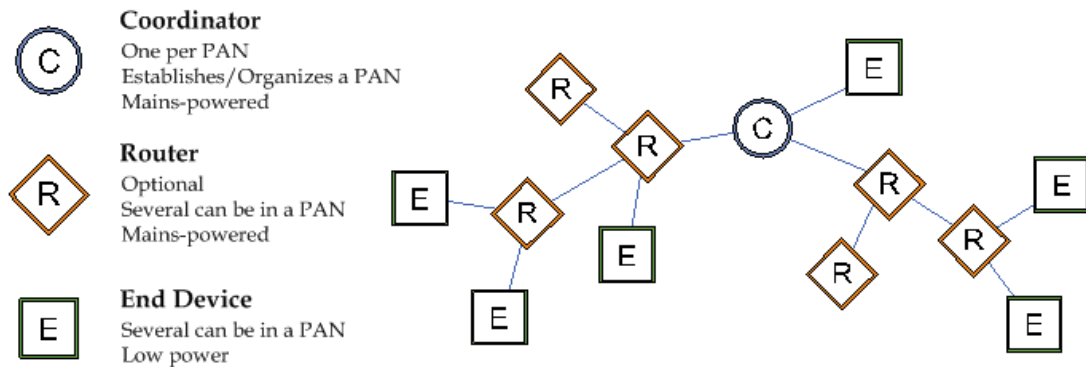
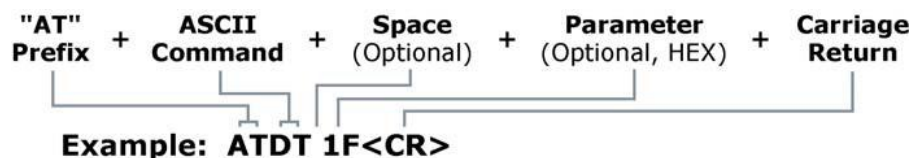


Figura 4.19.- Tipos de dispositivos Xbee

4.7 Conceptos del firmware

- PAN ID: el identificador de la red de área personal ya sea con una dirección de 16 o 64 bits. Este identificador es común a todos los dispositivos de la red.
- CHANNEL: El estándar IEEE 802.15.4 establece 16 canales de operación para los módulos Zigbee. X-bee soporta todos los 16 canales mientras que X-Bee Pro tan solo soporta 14 de los 16 canales.
- SC: Scan Channel establece la lista de canales a escanear.
- SD: Scan duration. Establece el tiempo de duración de escaneo del canal El tiempo de escaneo es medido por as: $(\# \text{ Channels to Scan}) * (2 \wedge \text{SD}) * 15.36\text{ms}$
- NJ: Node Join time. Establece el tiempo que un coordinador o un router se puede unir a una red. 0xFF permite que siempre se unan.
- SH: Serial Number High.
- SL: Serial Number Low. Establece los 32 bits altos y bajos de la única dirección de 64 bits.
- NI: Node identifier: Este establece un nombre al dispositivo en caracteres ASCII imprimibles, y puede ser utilizado para enviar datos.

Todos estos son comandos AT y pueden ser configurados tanto en la pestaña de Modem Configuration o a través de enviar comandos AT por la pestaña de terminal. Para entrar en el modo configuración con comandos AT a través del terminal enviamos la secuencia de tres caracteres “+++” y esperamos la confirmación del modulo ”OK”. Para enviar comandos AT se utiliza la siguiente sintaxis



Al final se anexan las tablas de los comandos AT para la configuración.

4.8 Ejemplos de transmisión de datos.

Antes que nada debemos haber conectado dos dispositivos al PC y haber actualizado el firmware. Después establecemos uno como coordinador ZB y esperamos a que se una a un canal. Luego procedemos a establecer el otro modulo como router o end device y esperamos a que se enganche a la misma red del coordinador, es decir que tengan

el mismo PAN-ID y el mismo canal. Después de haber verificado esto podremos enviar mensajes desde el router o el end device al coordinador de 2 formas.

La primera es enviando a la dirección de destino 0x00. Lo cual se realiza de la siguiente manera. Entramos en modo comandos en el terminal:

1. “+++” y esperamos confirmación.
 - a. ATDH0\r //DH=0
 - b. ATDL0\r //DL=0
 - c. ATCN\r //exit.

En cada uno de los comandos nos debe llegar un “ok”. Después de esto todos los mensajes serán enviados al coordinador. La segunda forma es conociendo la dirección de 64 bits del coordinador. Supongamos que la dirección del coordinador es 0x0013A200404A2244. Entramos en modo comandos:

1. “+++” y esperamos confirmación.
 - a. ATDH13A200\r
 - b. ATDL404A2244\r
 - c. ATCN\r //exit.

De esta manera lo que se envíe llegará solo al coordinador. Todo esto es una transmisión “unicast” es decir solo entre dos dispositivos de la red ZB, aunque solo tenemos dos dispositivos conectados, si más dispositivos se unieran a la red la comunicación solo llegaría al coordinador.

Ahora realicemos una transmisión “broadcast”:

1. “+++” y esperamos confirmación.
 - a. ATDH0\r
 - b. ATDLFFFF\r //quiere decir que la transmisión va para todos los dispositivos de la red.
 - c. ATCN\r //exit.

4.9 Modo API

Este modo es más complejo, pero permite el uso de frames con cabeceras que aseguran la transmisión de los datos, al estilo TCP. Extiende el nivel en el cual la aplicación del cliente, puede interactuar con las capacidades de red del módulo.

Cuando el módulo Xbee se encuentra en este modo, toda la información que entra y sale, es empaquetada en frames, que definen operaciones y eventos dentro del módulo.

Así, un frame de Transmisión de Información (información recibida por el pin 3 o DIN) incluye:

- Frame de información RF transmitida.
- Frame de comandos (equivalente a comandos AT).

Mientras que un Frame de Recepción de Información incluye:

- Frame de información RF recibida.
- Comando de respuesta.
- Notificaciones de eventos como Reset, _ ODEM_ do, Disassociate, etc.

Esta API, provee alternativas para la configuración del módulo y ruteo de la información en la capa de aplicación del cliente. Un cliente puede enviar información al módulo Xbee. Estos datos serán contenidos en un frame cuya cabecera tendrá información útil referente el módulo.

Esta información además se podrá configurar, esto es, en vez de estar usando el modo de comandos para modificar las direcciones, la API lo realiza automáticamente. El módulo así enviará paquetes de datos contenidos en frames a otros módulos de destino, con información a sus respectivas aplicaciones, conteniendo paquetes de estado, así como el origen, RSSI (potencia de la señal de recepción) e información de la carga útil de los paquetes recibidos.

Entre las opciones que permite la API, se tienen:

- Transmitir información a múltiples destinatarios, sin entrar al modo de Comandos
- Recibir estado de éxito/falla de cada paquete RF transmitido.
- Identificar la dirección de origen de cada paquete recibido.

Esta conexión, agrega información extra a los paquetes de datos RF. Ya no son enviados de forma transparente, sino que cada paquete de datos, son almacenados dentro de un frame, con una estructura definida que permite una forma más robusta para enviar datos. Esto permite entre otras cosas determinar el origen de algún paquete recibido dentro de la red. Cuando la configuración API está activada, cada paquete RF que se envía o recibe se encapsula en un frame de datos UART. Para esto se utiliza el comando AP. El frame se observa en la siguiente figura 4.20:



Figura 4.20.- Frame de datos UART

Existen tres posibilidades de configuración. Con AP=0, se deshabilita el frame API y el módulo trabaja en modo transparente. Con AP=1, el módulo trabaja en el modo API. Y con AP=2, el módulo trabaja en modo API, pero con Carácter de Escape. Este modo es necesario sólo cuando se envían bytes que interfieren con la estructura del Frame.

Éstos son:

- 0x7E – Delimitador de Frame.
- 0x7D – Escape
- 0x11 – XON
- 0x13 – XOFF

Este modo, ingresa un carácter de escape, además de otra operación sobre el bytes de interferencia. Esto hace que el frame sea más grande, al agregar bytes, pero evita que la cabecera del frame se confunda con los datos enviados. Otra ventaja es el Checksum, que permite verificar que los datos entregados no se hayan corrompido.

Entre las posibilidades que permite la API, es la posibilidad de cambiar parámetros a través de comandos AT, enviándolos al módulo de destino. Así, desde un módulo, es posible configurar otro utilizando el modo API. También es posible consultar sobre el estado de algún parámetro en otro módulo. Además se puede consultar sobre el estado del módem, como saber si está asociado a un coordinador, o si el módulo es o no un coordinador.

Como regla general el modo API se utiliza cuando:

- Se envían datos RF a múltiples destinos.
- Se envían comandos de configuración remota a otros dispositivos en lared.
- Se reciben muestras de una I/O de otro dispositivo.
- Se reciben datos de múltiples destinos y se necesita saber quien los envía.
- Se soportan modulos Zigbee con cluster ID, perfiles ID u otros dispositivos de otra marca (ZDO).

4.10 Modulación QPSK

El módulo XBee emplea la tecnología de secuencia directa en espectro disperso (DSSS, Direct Sequence Spread Spectrum) con transmisión por desplazamiento de fase cuaternaria (QPSK, Quadrature Phase-Shift Keying). En QPSK se tienen cuatro posibles fases de salida, para una sola frecuencia de portadora. Para seleccionar una de estas fases se emplean dos bits de entrada llamados dibits. Cada dibits genera una de las cuatro fases posibles. En la Figura 4.21 se muestra el diagrama de constelación de QPSK.

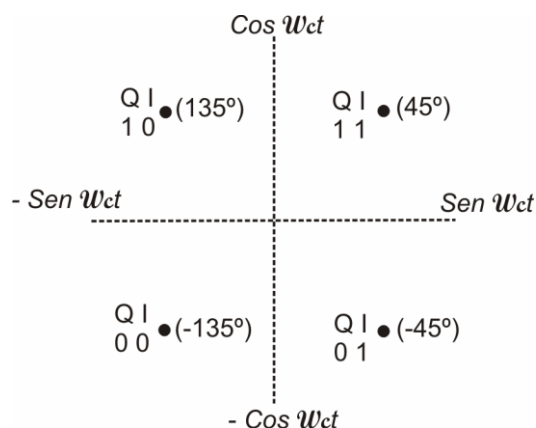


Figura 4.21.- Diagrama de constelación.

En la figura 4.22 se muestra el diagrama a bloques de un modulador QPSK. Los datos de entrada se separan en dibits los cuales ingresan en forma paralela, un bit al canal I, mientras que el otro bit entra al canal Q. El bit I modula una portadora que está en fase con el oscilador de referencia, el bit Q modula una portadora que está en cuadratura (desfasado 90°) con el oscilador de referencia. El sumador combina las dos señales teniendo como resultado cuatro posibles fases.

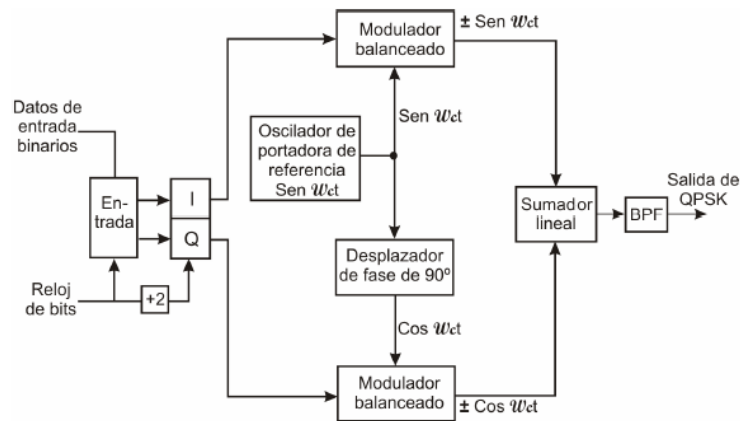


Figura 4.22.- Modulador QPSK.

Como los datos de entrada se dividen en dos canales, la frecuencia de bit en el canal I o en el canal Q es igual a la mitad de la frecuencia de los datos de entrada ($f_b/2$). El circuito QPSK alarga el tiempo de bit de Q e I, por lo tanto la frecuencia de Q o I es un cuarto de la frecuencia de los datos de entrada ($f_b/4$). En la figura 4.23 se esquematizan las señales de entrada y de los dibits para obtener el ancho de banda mínimo.

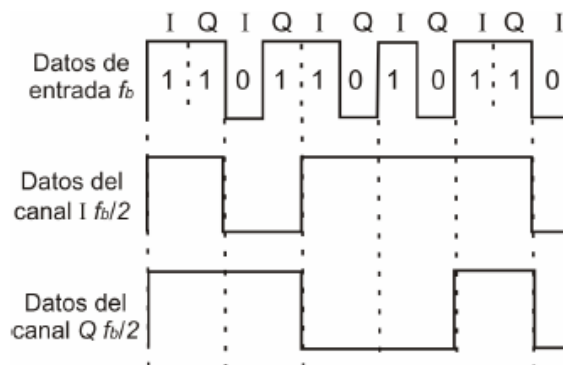


Figura 4.23.- Datos de entrada y los dibits.

Entonces el ancho de banda (B) mínimo requerido por los moduladores balanceados es dos veces la frecuencia de Q o I, por lo tanto se tiene que B esta dado por la siguiente ecuación:

$$B = 2 \left(\frac{f_b}{4} \right) = \frac{f_b}{2}$$

La señal periódica esta descrita como:

$$x(t) = x(t - T_0)$$

Con

$$\omega_o = \frac{2\pi}{T_0}$$

$$f_0 = \frac{1}{T_0}$$

Se tiene que la **Transformada de Fourier** de una función periódica $x(t)$ se puede escribir como:

$$F\{x(t)\} = X(\omega)$$

$$X(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$$

Dada el siguiente pulso rectangular:

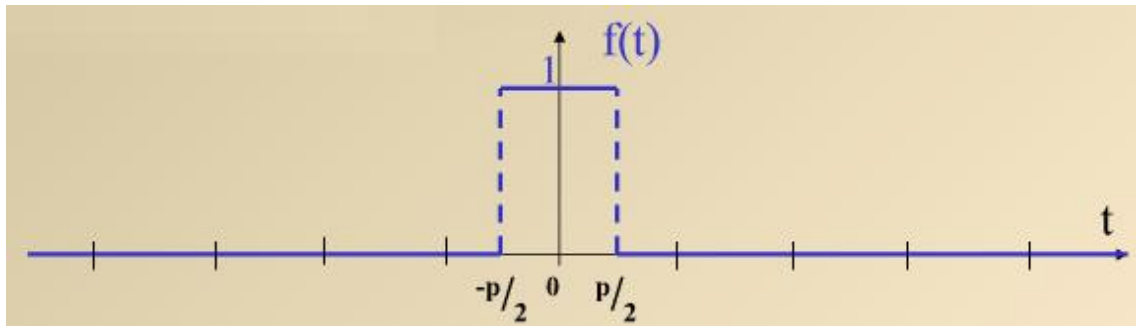


Figura 4.24.-Pulso rectangular no periódico en el tiempo.

Dada la señal cuadrada no periódica mostrada en la Figura 4.23 se tiene el siguiente cálculo de su Transformada de Fourier.

$$\begin{aligned}
 F(\omega) &= F\{f(t)\} = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \\
 &= \int_{-p/2}^{p/2} A e^{-j\omega t} dt = A \left[-\frac{1}{j\omega} e^{-j\omega t} \right] \Big|_{-p/2}^{p/2} \\
 &= A \left[\frac{1}{-j\omega} e^{-j\omega p/2} - \frac{1}{-j\omega} e^{j\omega p/2} \right] \\
 F(\omega) &= A \left[\frac{e^{j\omega p/2} - e^{-j\omega p/2}}{j\omega} \right] \\
 F(\omega) &= \frac{2A}{\omega} \left[\frac{e^{j\omega p/2} - e^{-j\omega p/2}}{2j} \right] \\
 F(\omega) &= \frac{2A}{\omega} \sin\left(\frac{\omega p}{2}\right) \\
 F(\omega) &= \frac{2A}{2\pi f} \sin\left(\frac{2\pi f p}{2}\right) \\
 F(\omega) &= \frac{A}{\pi f} \sin(\pi f p)
 \end{aligned}$$

$$F(\omega) = \frac{A \sin(\pi f p)}{\pi f}$$

$$F(\omega) = A p \operatorname{sinc}(p f)$$

Gráficamente su espectro está representado por:

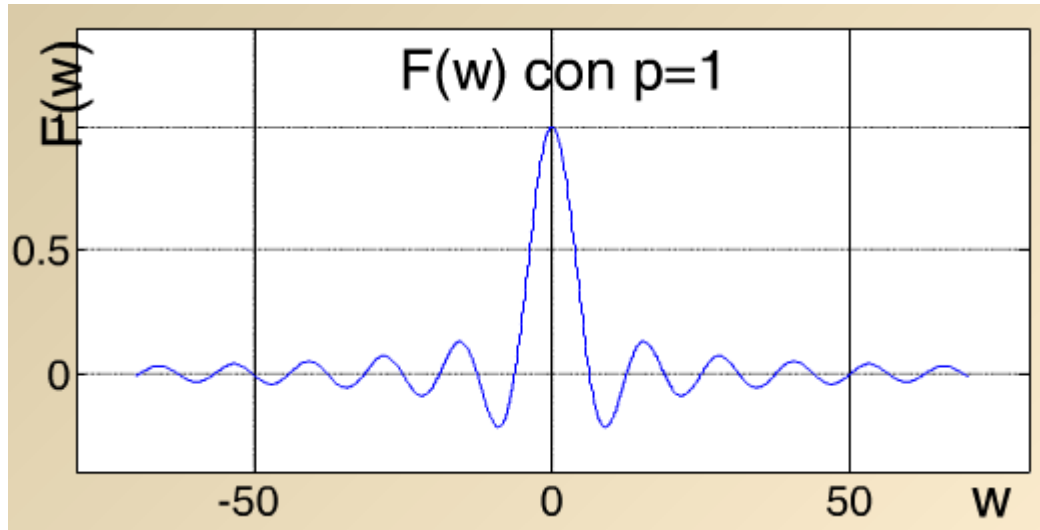


Figura 4.25.- Espectro de amplitud de un pulso rectangular.

NOTA: Para una mejor explicación sobre series trigonométricas de Fourier consulte en el apartado de la bibliografía las siguientes referencias 15, 16, 17, 18, 19.

CAPÍTULO 5 MÓDULO GSM

5.1 INTRODUCCIÓN

La red GSM empleada por los teléfonos celulares provee una comunicación inalámbrica de amplio rango y gracias a los servicios que ofrece es cada vez más común el empleo de módulos embebidos GSM/GPRS en aplicaciones de control remoto y monitoreo a través de de redes GSM.

El mundo de las telecomunicaciones es enormemente dinámico, y en los últimos años su exponente más notable ha sido el auge de la telefonía celular (aunque quizás sería más preciso llamarla comunicaciones personales móviles). La telefonía celular no es nueva, ya lleva más de 20 años en el mercado, y surgió como evolución de los tradicionales sistemas de RF por VHF o UHF al contarse con dispositivos semiconductores capaces de operar a frecuencias cada vez mayores, a la disponibilidad de redes de comunicaciones y de sistemas de cómputo más poderosos y económicos.

De los primeros y voluminosos teléfonos celulares a los modelos de hoy, la evolución ha sido constante. Pero con la aparición de nuevas normas de transmisión también aparecieron nuevas funcionalidades y por ende áreas de aplicación que han excedido a la simple telefonía. El caso del GSM, a través de sus facilidades de transmisión de datos, es un ejemplo claro. Actualmente se está comercializando un completo conjunto de productos GSM, en la forma de módulos de reducidísimas dimensiones (35x33x3 mm) (Figura 5.1), capaces de establecer no sólo comunicaciones de voz, sino también canales de datos para el envío de mensajes diferidos (SMS) o enlaces de datos de alta velocidad por conmutación de paquetes (GPRS, UMTS, HSDPA), en ciertos casos bajo protocolos de Internet (TCP/IP) e incluso ejecutando programas escritos en lenguaje Java.

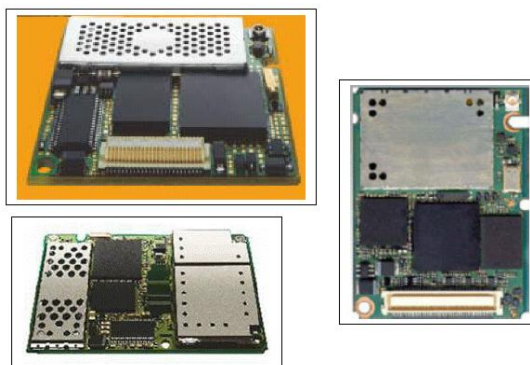


Figura 5.1.- Módulos GSM/GPRS

Estos dispositivos abren un nuevo mundo de aplicaciones, por ejemplo:

- Conexiones de datos entre máquinas (M2M: Machine to Machine), a través de telefonía: sistemas de medición, puestos de venta, seguridad, control de flotas, seguimiento de mercaderías. Por ejemplo, medición de equipos de red en CATV, de los transformadores en una red de alta tensión, terminales de lectura de tarjeta de débito móviles, contenedores inteligentes, monitoreo de flotas taxímetros o de camiones, alarmas domiciliarias, sistemas portátiles de monitoreo en línea de signos vitales.

- Aplicaciones automotrices: teléfono, correo electrónico y SMS, sistemas antirrobo, navegación.

Es decir, tanto la madura tecnología de seguimiento de vehículos, como la adquisición remota de datos, telemetría, y control a distancia, abren un nuevo universo de negocios en el que está todo por hacer e imaginar.

5.2 LA TELEFONÍA CELULAR

El uso de equipos de radio convencionales para comunicar a personas entre sí presenta enormes limitaciones cuando la cantidad de interlocutores es elevada; dado que un canal de voz transmitido “analógicamente” puede ocupar de 5 a 30kHz (según la voz se module en banda lateral, AM o FM), y las frecuencias disponibles en el espectro se ocuparían rápidamente. Este uso “compartido” del espectro entre muchas comunicaciones es llamado Frequency Division Multiple Access (FDMA).

Una primera alternativa puede ser no usar frecuencias fijas sino asignar a cada interlocutor las frecuencias de uso dinámicamente, a medida que las solicita, y liberarlas al cortarse la comunicación: esto es lo que hoy hacen los sistemas llamados de “trunking”. Lo interesante de esta solución es que aparece la idea de la existencia de un canal de control, a través del cual se negocia una transacción para el pedido de canal y de liberación de canal, y en la que no sólo se genera el permiso de uso sino que se define la frecuencia en que se realizará ese enlace por esa vez. Un canal de control, usado por todos los abonados, tiene el problema de que más de un abonado puede querer usarlo a la vez y colisionar, por lo que se hace necesario que los mensajes de control sean breves y usar métodos de acceso que traten de minimizar los perjuicios de esas colisiones.

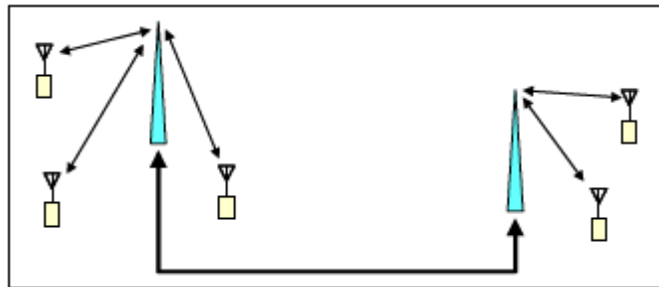


Figura 5.2.- Conexión de una red

Si en vez de realizar comunicaciones punto a punto éstas se realizan a través de centros de control que reciben y retransmiten los diálogos (Figura 5.2), es posible administrar las comunicaciones más eficientemente. Esta solución permite emplear mucha menos potencia de radio para el enlace entre las terminales (los abonados) y los centros de control próximos, y permitir que los centros de control se comuniquen a su vez entre sí. De este modo la señal de RF que emite cada abonado se extingue rápidamente y la frecuencia que le ha sido asignada puede volver a ser asignada a otro abonado que esté conectado a otro centro de control distante, sin que se produzca interferencia (esto es llamado frequency reuse).

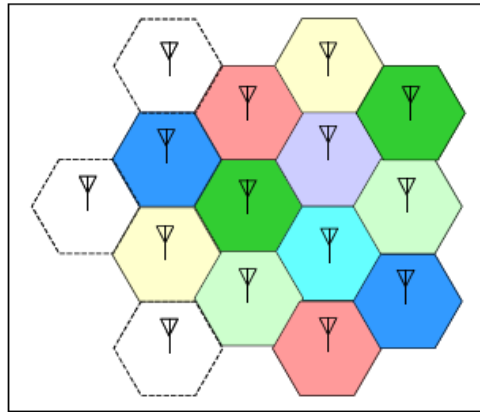


Figura 5.3.- Centros de control telefónicos

Esto da pie al concepto de “célula” usado en telefonía “celular”: una malla de centros de control distribuida (Figura 5.3), y comunicaciones en las que los abonados no dialogan directamente entre sí, sino a través de los centros de control. En esta malla (por ejemplo, es tradicional una red de hexágonos) las frecuencias de uso posibles entre un centro de control y los 6 vecinos son diferentes, de modo de no interferir.

Según la frecuencia de transmisión y la potencia empleada las celdas son más o menos grandes. En 800MHz y Clase 4 (potencia máxima de transmisión de 2W, o 33dBm) la separación entre antenas puede ser de hasta 10 km, en tanto en 1900MHz y Clase 1 (1W de potencia máxima, o 30dBm), esa separación es de sólo 3 a 4 km. Obviamente, una celda pequeña tiene como ventaja que al usar menos potencia las baterías de los móviles duran más, y que las frecuencias pueden reusarse rápidamente, y por tanto atender más abonados en una misma región; en contraparte, obligan a que el proveedor del servicio instale muchas más estaciones de control para poder garantizar una dada cobertura.

Una vez que esta estructura está armada, aparecen problemas y soluciones:

- ¿Qué pasa si un abonado se mueve de una célula a otra célula? Pues si el centro de control que lo atiende mide que la potencia de su señal va bajando, y algún centro de control vecino lo detecta con mayor nivel, por un canal de control se le da al equipo del abonado la orden de cambiar de frecuencia y el nuevo centro de control pasa a atenderlo (este “pasaje de manos” es llamado handoff), y la posibilidad de saltar dinámicamente de una frecuencia a otra es llamada Frequency Hopping (FH).
- A la vez, pasa a ser necesario que el centro de control pueda controlar la potencia con que cada abonado transmite, de modo que si dos abonados usan frecuencias vecinas, pero uno está mucho más cerca del centro de control, no interfiera con la señal más débil del abonado vecino.

La administración de todos los centros de control implica una tarea de computación importantísima, y este tipo de tareas es realizada por lo que es llamado MTSO (Mobile Telephone Switching Office). En el mundo analógico no existen muchas más alternativas que brindar el simple servicio de voz (POTS: Plain Old Telephone Service), pero al pasar al mundo digital aparecen muchas nuevas variaciones:

- Ya no es necesario transmitir la voz directamente, sino que puede ser convertida a digital, y emplear métodos de modulación para la transmisión de datos que son mucho más eficientes.
- En una misma frecuencia pueden asignarse espacios de tiempo (time-slots) para que cada abonado envíe o reciba sus datos, es decir compartir una misma frecuencia entre varios canales de voz dividiendo y repartiendo el uso del tiempo. Esto se llama Time Division Multiple Access (TDMA). En realidad es una combinación de FDMA (se asignan bandas de frecuencia móviles) con TDMA (se asignan porciones de uso del tiempo en esa banda).
- En vez de digitalizar la voz usando Pulse Code Modulation (PCM) que requiere 64 kbps (8 mil muestras por segundo, a 8 bits por muestra) es posible aprovechar la redundancia de la voz, y emplear métodos de compresión de datos que requieren menor bitrate. Este proceso de compresión y descompresión es llamado CODEC y en cada nueva generación de celulares es cada vez más eficiente (es decir se transmite igual calidad de voz empleando cada vez menos bits/segundo).

Hasta hace pocos años, la telefonía celular analógica y digital ha funcionado de este modo, con la misión fundamental de facilitar enlaces de voz entre abonados.

5.3 CONCEPTOS BÁSICOS DE GSM/GPRS.

GSM proviene de las siglas Global System for Mobile communications (Sistema Global para las comunicaciones Móviles), es el sistema de teléfono móvil digital más utilizado y el estándar para teléfonos móviles en Europa, ETSI (European Telecommunications Standards Institute) define las especificaciones de GSM (Figura 5.4). Originalmente como estándar Europeo abierto para que una red digital de teléfono móvil soporte voz, datos, mensajes de texto y roaming (el movimiento desde la red de un operador a otra) en varios países.

Su característica más visible es la tarjeta-chip SIM (Subscriber Identity Module) que se inserta en el aparato telefónico (Figura 5.5) y en el cual se almacenan los datos de identificación del usuario, así como datos propios del usuario, la tarjeta SIM tiene capacidades procesamiento y almacenamiento. Las especificaciones de dichas tarjetas inteligentes esta dada en el documento ISO-7816.



Figura 5.4.- Logotipo de GSM y ETSI.

Por ser digital y tener un rendimiento máximo de 9.6kbps, cualquier cliente de GSM puede conectarse a través de su teléfono con su ordenador y puede hacer, enviar y recibir mensajes por e-mail, fax, navegar por Internet,, así como utilizar otras funciones digitales de transmisión de datos, incluyendo el Servicio de Mensajes Cortos (SMS) (si el hardware del equipo lo permite, claro).

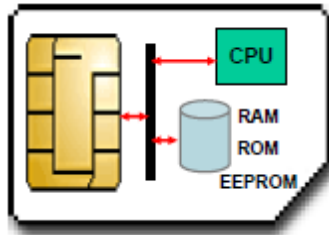


Figura 5.5.- Tarjeta SIM.

GSM es considerado un estándar de segunda generación (2G-comunicación completamente digital). Básicamente la denominación de las generaciones (Figura 5.6) contempla cambios en modo de transmisión (analógica, digital), servicios ofrecidos y sobre todo velocidad de transmisión de datos.

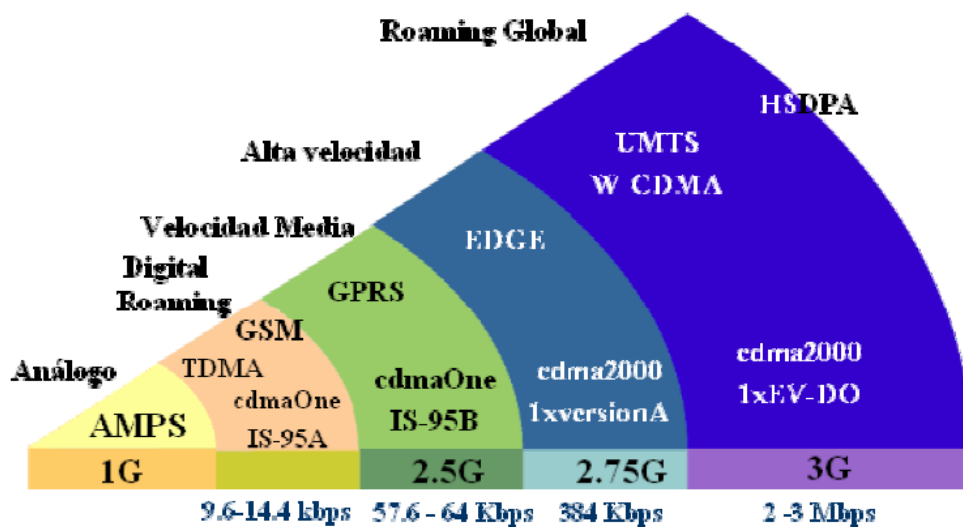


Figura 5.6.- Generaciones GSM

La tecnología 2G puede ser dividida en dos grupos de acuerdo al estándar de acceso al medio que ocupen, los cuales pueden ser TDMA (Time division multiple access) y CDMA (Code division multiple access), los principales estándares 2G que se adoptaron en México son GSM, iDEN (basados en TDMA) e IS95A – cdmaOne (basado en cdma).

5.3.1 Frecuencia de transmisión.

En Europa, el estándar GSM emplea las bandas de frecuencia de 900Mhz y 1800Mhz, llamadas GSM- 900 y GSM-1800. En Estados Unidos se emplean las bandas 850Mhz y 1900MHz. En el caso de las bandas el cambio para los teléfonos es automático (Figura 5.7), así para un teléfono de banda dual conmutará de una banda a otra, un teléfono tribanda (900/1800/1900) podrá trabajar en las tres bandas y un cuatribanda podrá configurarse generalmente para trabajar en una u otra región.

A menudo se hace referencia a las bandas de frecuencia con los nombres E-GSM 900, DCS 1800, GSM 850 y PCS1900.

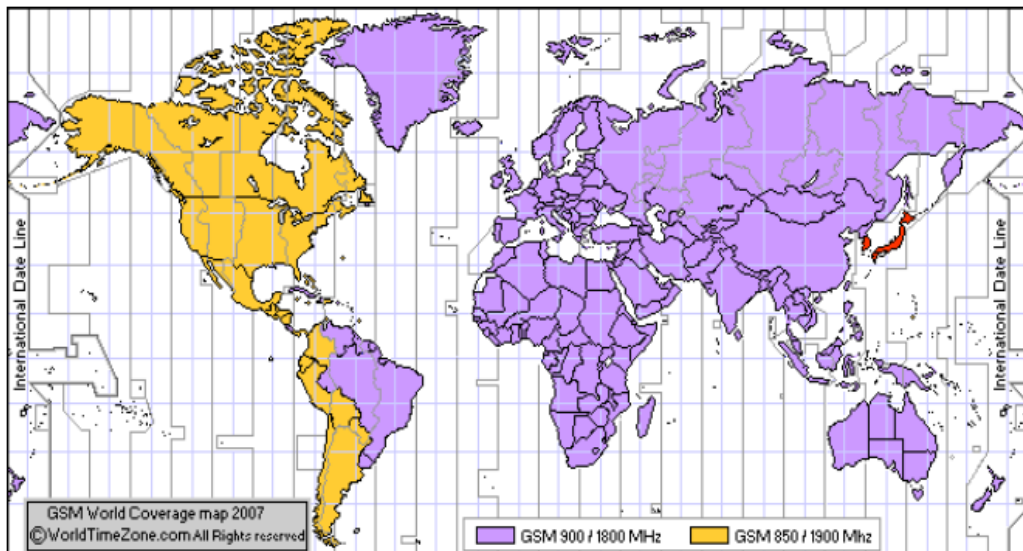


Figura 5.7.- Bandas de frecuencia GSM.

Por cada banda de frecuencia, GSM las divide en 2 sub-bandas (Figura 5.8) una para transmitir y otra para recibir. Por ejemplo en el estándar GSM-900 la sub-banda de 890-915 MHz se usa para las transmisiones desde la MS (movil system) hasta el BTS (Enlace móvil a estación base o ascendente "uplink") y la banda de 935-960 MHz se usa para las transmisiones entre el BTS y la MS ("Enlace estación base a móvil o descendente downlink").

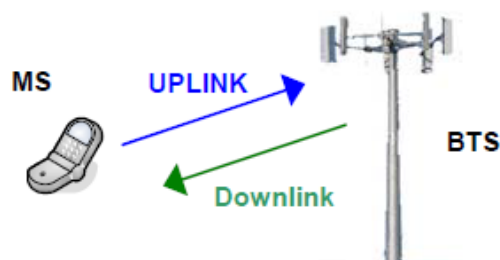


Figura 5.8.- Sub-bandas de transmisión/recepción.

5.3.2 GPRS.

GPRS (General Packet Radio Service – Servicio general de radiocomunicaciones por paquetes), GPRS es sólo una modificación de la forma de transmitir datos en una red GSM a partir de la superposición de hardware y software a una red GSM (Figura 5.9). Precisamente de aquí se desprenden dos términos muy importantes, la conmutación de circuitos en GSM y la conmutación de paquetes en GPRS. En la conmutación de circuitos el sistema reserva un camino “circuito” permanentemente mientras dure la comunicación, y aunque no se envíe información en un momento dado, el servicio de transmisión de datos por este medio es llamado Datos por Conmutación de circuitos (CSD).

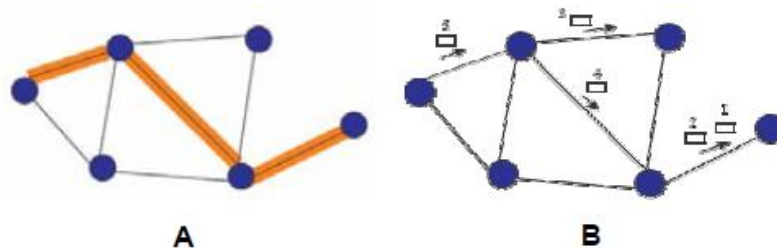


Figura 5.9.- A) Conmutación de circuitos y B) paquetes.

En el caso de la conmutación de paquetes los recursos de la red son compartidos teniendo una conexión permanente, la facturación es un punto muy importante, mientras en CSD se factura por tiempo de conexión, en gprs se factura por evento o por volumen de información intercambiada. Los tiempos de espera o los dedicados a leer una página no le cuestan nada al cliente.

5.3.3 Comandos AT.

El protocolo empleado por los módems GSM para configurarlos y controlarlos está basado en el set de comandos AT Hayes, pero hay comandos específicos que han sido adaptados para los servicios que ofrece GSM tal es el caso de los mensajes de texto, manejo de memoria, funciones especiales de modem. Los comandos AT para módems compatibles están definidos en los documentos ETSI GSM 07.07 y GSM 07.05. GSM 07.07 define los comandos generales de GSM y GSM 07.05 define los comandos para manejo de mensajes escritos.

5.3.4 SMS.

La especificación para SMS está dada en los documentos GSM 03.40 y GSM 03.38. Un SM puede tener hasta 160 caracteres de longitud, donde cada carácter es de 7 bits y corresponde a un carácter del alfabeto de 7 bits. En el caso de 8 bits el SM puede tener un máximo de 140 caracteres, empleado generalmente para mensajes inteligentes (imágenes y ring tones).

La estructura dentro del cual viaja un SM se le denomina PDU (Protocol Data Unit) la cual además de llevar información del propio texto lleva otra serie de caracteres con las que se pueden hacer funciones de control en la presentación del mensaje. La principal ventaja de este modo es que el mensaje antes de ser enviado a la red, debe pasar por un algoritmo el cual hace una codificación de 7 a 8 bits.

Al momento de emplear comandos AT los módems dan la opción de emplear el modo de envío/recepción PDU ó TEXTO, lo más sencillo es manejar modo texto ya que en modo PDU hay que armar la cadena en forma comprimida. En el envío de Mensajes cortos esta involucrado el llamado SMSC (Centro de servicio de Mensajes Cortos) el cual se encarga del almacenamiento y envío de dicho mensaje, este se puede configurar por comandos AT, pero generalmente el modem GSM lo configura al conectarse a la red que le es asignada.

5.4 FUNCIONAMIENTO GSM

Desde el punto de vista técnico, los terminales GSM suelen poder operar en al menos dos bandas de frecuencia, una casi el doble que la otra, lo que les da mejores posibilidades de comunicación. Las frecuencias hoy en uso son de 850MHz y 900 MHz (frecuencias bajas, con hasta 2W de potencia) y 1.8GHz y 1.9GHz en frecuencias altas (con hasta 1W de potencia), no existiendo un uso de frecuencias común a todos los países (esto plantea una limitación de compatibilidad para terminales bi-banda o tri-banda, por lo que la tendencia es hacia módulos cuadri-banda).

El planteo de GSM fue revolucionario, en cuanto propuso el uso de técnicas de compresión de voz y de transmisión de datos en forma digital que no estaban aún disponibles en el momento de la especificación, y que significaban un enorme cambio respecto a los sistemas preexistentes analógicos como AMPS (Advanced Mobile Phone Service, utilizado entonces en USA) y TACS (Total Access Communication System, usado entonces en Inglaterra).

GSM usa TDMA (en el que cada canal de RF de 200kHz es dividido en 8 períodos de tiempo, permitiendo de este modo 8 conversaciones simultáneas en esa misma frecuencia de radio; también permite dividir ese tiempo en 16 rodajas (time-slots) con un régimen de datos inferior (half-rate). En las frecuencias bajas, para GSM se asignan 50MHz de ancho de banda, 25MHz para el enlace de subida (uplink) y otros 25MHz para el enlace de baja (downlink), es decir 125 canales de subida y otro tanto de bajada; en las frecuencias altas se asignan 100MHz, 50MHz para el uplink y 50MHz para el downlink, lo que resulta en el doble de canales disponibles.

En GSM, la estación móvil (MS: Mobile Station) consiste en la terminal móvil y un módulo de memoria denominado SIM (Subscriber Identification Module) que contiene información básica del abonado y de la terminal. De este modo, un SIM puede ser pasado de una terminal a otra, que instantáneamente puede utilizar esa información. El equipo terminal es identificado por un código denominado International Mobile Equipment Identity (IMEI), que puede servir para identificar equipos robados, o de contrabando, en tanto que la tarjeta SIM contiene el código International Mobile Subscriber Identity (IMSI), utilizado para identificar al abonado en el sistema, claves secretas de autenticación y mucha otra información adicional.

El diseño puramente digital de GSM y la forma dinámica de asignación de los “time-slots” permite ofrecer servicios no sólo de voz sino también de datos entrantes y salientes, como email, fax, e internet, donde el flujo de datos no es instantáneo (como requiere una comunicación de voz) sino que puede ser en forma de paquetes que se almacenan y retransmiten múltiples veces para llegar desde el origen al destino. El ancho de banda limitado de estos enlaces de datos (en comparación a los enlaces de banda ancha disponibles en ADSL o CATV) determina el uso de variaciones de ciertos protocolos, tal es el caso del WAP (Wireless Access Protocol) que permite la realización de páginas web simples de actualizar y aptas para pequeñas pantallas.

Entre los servicios digitales típicos están:

- GPRS (General Packet Radio Service), que en función de los “time-slots” disponibles y la forma de modulación permiten enlaces de decenas de kbps (es

normal hablar de regímenes máximos de más de 80kbps, y típicos de 10 a 20 kbps).

- Como alternativa a GPRS, en GSM también existe la conmutación de circuitos (llamada CSD, por Circuit Switched Data), donde un canal de voz es empleado para transmitir datos, lo que suele permitir enlaces de hasta 14.4kbps, y donde se realiza control de errores y de retransmisión, es decir se establece un “circuito virtual” de datos.
- SMS (Short Message Service), apto para el envío y recepción de mensajes de hasta 256 caracteres empleando facilidades de GPRS o de CSD. Este es un servicio donde los mensajes son transportados en modo “store-and-forward”, es decir, “dando saltos” de computadora a computadora del sistema, desde que salen del origen hasta que llegan al destino. Estos mensajes pueden tener un único destinatario (punto a punto) o ser de información general (por ejemplo, información de tráfico o noticias). Los mensajes SMS forman parte de la información que puede ser almacenada en el módulo SIM para su posterior consulta.

5.4.1 El uso del espectro

Una característica limitante de cualquier sistema de comunicación inalámbrica es el ancho de banda asignado. Se dijo que en la banda baja se asignan 50MHz, 25MHz para el uplink y 25MHz para el downlink. Estos 25 MHz en cada sentido son divididos en 125 canales de 200kHz cada uno, y un subgrupo de estos 125 señales es asignado a cada estación de control de cada celda, de modo de no colisionar con las frecuencias en uso por las celdas vecinas. En el caso de las frecuencias altas (1800/1900MHz) se dispone del doble de canales.

A su vez, cada canal de frecuencia es dividido en el tiempo en 8 canales físicos (TDMA), y para esta división en tiempo, a cada canal físico se le asignan períodos de 15/26 milisegundos (0.577ms, o burst time), y ocho de estos períodos son agrupados en el llamado TDMA frame, de $120/26 = 4.615$ ms de duración. Entonces, cada canal físico tiene asignado un burst time en cada TDMA frame.

Los canales físicos pueden ser asignados para tráfico a un dado móvil mientras dura un enlace (dedicated channels), o ser usados por los móviles en modo pasivo (idle mode) para tareas de control (common channels).

- Un canal de tráfico (traffic channel) emplea bloques de 26 TDMA frames, que se repiten cada 120ms; de éstos, 24 son usados para tráfico y los dos restantes para control. Los canales de tráfico de subida y bajada entre la estación de control y cada abonado están desfasados en el tiempo, de modo que un móvil no deba estar transmitiendo y recibiendo a la vez, lo que simplifica la electrónica y permite compartir una única antena para transmisión y recepción. Yendo hacia el detalle, en cada burst time normal se transmiten 156,25 bits, de los cuales son 114 de datos (separados en dos grupos de 57 bits, y encriptados para garantizar la privacidad y seguridad mediante claves que envía la central y claves guardadas en el SIM), lo que implica una tasa de transmisión pulsada de casi 271 kbps.
- Un canal de control emplea por su parte bloques de 51 TDMA frames, y los burst time empleados son similares al caso de los canales de tráfico, excepto un tipo

especial de burst llamado access burst que tiene menos bits, y sólo es usado cuando un móvil desea solicitar conexión a la estación de control (RACH: Random Access Channel). En este caso, cada burst time tiene varios momentos precisos en los que una terminal puede intentar pedir ser atendida; dado que existe el riesgo que dos terminales colisionen entre sí, una terminal no atendida insiste luego de un cierto tiempo aleatorio: este método de acceso usado en RACH es llamado “slotted aloha”.

Para transmitir los datos a 271kbps GSM emplea un método de modulación digital llamado GMSK (Gaussianfiltered Minimum Shift Keying), resultante de una elección de compromiso entre uso eficiente del espectro, facilidad de realización, y mínima interferencia sobre canales adyacentes.

5.4.2 Aprovechamiento del canal en la transmisión de voz

Cuando GSM es empleado para transmitir voz, los 271kbps son distribuidos entre los ocho canales físicos, lo que permitiría casi 34kbps por canal. Pero desde su inicio la norma GSM fue sumamente ambiciosa:

- A través de técnicas de DSP (Digital Signal Processing) la voz se comprime a 13kbps usando un método llamado RPE-LPC (Regular Pulse Excited-Linear Predictive Coder), donde se modela matemáticamente la voz y sus cambios cada 20ms, codificándola con 260 bits. Este es el método inicial de GSM, aunque en este momento existen varias alternativas de CODEC con distinta eficiencia, tales como Half-Rate (HR), Full Rate (FR), Enhanced Full Rate (EFR), Adaptive Multi-Rate (AMR) y donde el procesador DSP puede realizar cancelación de eco y reducción dinámica de ruido. Claramente, cuanto más eficiente es el CODEC más comunicaciones simultáneas pueden realizarse en una misma frecuencia
- Pero luego, para asegurar que los errores de recepción del enlace no afecten la calidad de la voz, se agrega información para corrección de errores (FEC: Forward Error Correction), y los bits resultantes son mezclados (interleaved) de modo que un error típico de un enlace (donde varios bits sucesivos son erróneos, o error burst) pueda ser corregido.
- Y aprovechando la circunstancia que una persona sólo está hablando una fracción del tiempo (estadísticamente, menos del 40%), se incluye un detector de actividad de voz (VAD: Voice Activity Detection) para que mientras que no hay voz no se gaste energía de las baterías en transmitir ruido de fondo (DTX: Discontinuous Transmission); a la vez, en el extremo receptor se suspende la recepción (DRX) pero se incluye un generador de ruido (comfort noise) de modo que quien escucha no sienta que la comunicación se interrumpió.

5.4.3 Aprovechamiento del canal en la transmisión de datos

Cuando GSM es empleado para transmitir datos por conmutación de paquetes, los 114 bits útiles de cada burst time pueden ser usados para transportar datos en varios de los 8 burst time de cada TDMA frame, (esto se llama uso multi-slot del canal de tráfico), en los 24 de los 26 TDMA frame que dura un canal de tráfico.

Clase Multi-slot	Slots de bajada	Slots de subida	Slots totales
1	1	1	2
2	2	1	3
3	2	2	3
4	3	1	4
5	2	2	4
6	3	2	4
7	3	3	4
8	4	1	5
9	3	2	5
10	4	2	5
11	4	3	5
12	4	4	5

Tabla 5.10.- Slots de bajada y de subida

Según cuántos slots sean usados para subir datos del móvil a la estación de control, cuántos para bajar datos desde la estación de control al móvil y la suma de ambos se definen clases de GPRS, entre 1 y 12, como muestra la tabla 5.10.

Considerando ciertos bits “de guarda” que también ocupan tiempo de transmisión, en GSM-GPRS el límite máximo teórico es de 171.2kbps para transmitir datos sin ningún tipo de corrección de errores; según el ruido del canal de radio esto es inaceptable, y por eso se definen 4 tipos de codificación con agregado de facilidades de detección de errores llamados CS1 a CS4.

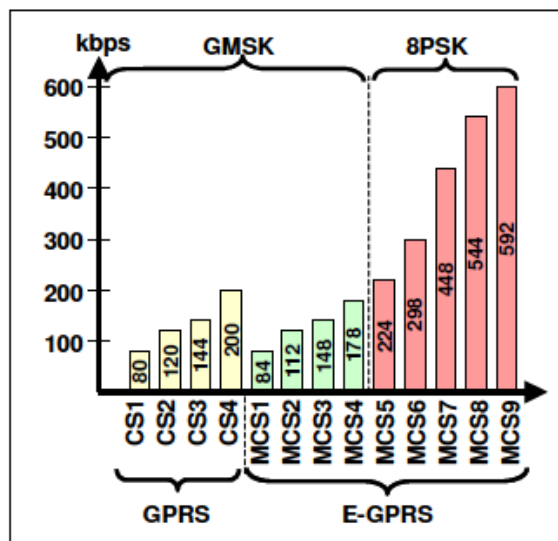


Figura 5.11.- Tipos de codificación

EDGE: Enhanced Data Rate for GSM Evolution ha sido agregado en GSM Fase 2+, y en este caso el cambio es el uso alternativo de una forma de modulación llamada 8PSK (Phase Shift Modulation en 8 pasos) en vez de GMSK, que aunque más compleja, permite usar la misma infraestructura de GSM pero obteniendo en teoría tasas máximas de transmisión de casi 600kbps. En este caso existen 9 formas de codificación con distintos niveles de detección de errores, denominados desde MCS1 a MCS9, donde MCS1 a MCS4 usa GMSK y MCS5 a MCS9 usan 8PSK (Figura 5.11).

Claramente, acá la limitación real no es tecnológica, sino que depende del interés de cada operadora de GSM de usar qué porcentaje de canal de tráfico para transmitir datos en vez de voz. Dentro de los modelos de negocio posibles, la cantidad de burst time asignados a

un único canal de datos probablemente dependan del grado de ocupación de la red, del horario en que se realiza el intercambio de datos, y esté sujeta a tarifas variables según el uso sea ocasional y se facture por bit transmitido, o por banda horaria, por un volumen de datos periódico (uno de los modelos de negocio ofrecido hoy en la Argentina), o por otro criterio comercial.

Otra especificación que hace al móvil es la llamada clase A, B o C:

- Los de Clase A pueden usar servicios de voz y datos simultáneamente
- En los de Clase B, se dispone de todos los servicios entre los que se puede conmutar dinámicamente, pero sólo un servicio puede ser usado a la vez, voz o datos, e incluso en el caso de datos, GPRS o SMS, pero no ambos simultáneamente
- En los de Clase C, existe una asociación predeterminada a voz o datos, entre los que debe conmutarse manualmente.

5.4.4 Generaciones 3G, 3.5G y sucesivas

Para una clara comprensión de las técnicas en evolución, es necesario describir el significado de ciertas siglas: Code Division Multiple Access (CDMA): es una tecnología de origen militar difícil de explicar en pocas palabras.

Para dar una analogía: si en un diálogo hay cinco personas hablando a la vez, con volumen parecido, cada una en un idioma distinto, a quien escucha esta cacofonía pero sabe uno de los idiomas le resulta posible diferenciar el dialogo en ese idioma de los demás. De igual modo, en CDMA varios abonados comparten el mismo espectro de frecuencia (se superponen) pero la señal de cada uno es modulada con una secuencia especial (chip) que permite separar luego esa señal de las demás, demodulándola con esa misma secuencia especial. Este proceso, que comparte un gran ancho de banda entre varios usuarios es llamado “de espectro ampliado”, o SS (Spread Spectrum).

Wideband Code Division Multiple Access (W-CDMA): es una ampliación de CDMA de gran ancho de banda empleada en redes celulares de tercera generación (3G), y es usada en el sistema denominado UMTS, que se describe más abajo. Sus características clave son:

- Los canales de radio de 5MHz de ancho de banda, con una tasa de chip de hasta 3.84 Mbps
- El enlace es dúplex, pudiendo ser por división de frecuencia o de tiempo. En el primer caso el enlace de subida (uplink) y el de bajada (downlink) usan canales distintos; en el segundo caso, se usa una única frecuencia y el tiempo se subdivide en 15 slots por trama (caso de UMTS).
- El proceso de recepción emplea detección coherente, lo que reduce la tasa de errores, gracias al agregado de símbolos y canales piloto. En función de la relación de interferencias (SIR) la potencia es controlada dinámicamente.
- A su vez, el tipo de enlace es dinámicamente renegociado de modo permanente (variable mission).
- El tipo de antenas y la negociación de enlace entre celdas es también redefinido, de modo de mejorar la performance de tasa de bits.

Universal Mobile Telecommunications System (UMTS): es una de las tecnologías usadas en 3G. Basada en WCDMA, es a veces llamada 3GSM, para enfatizar la combinación de de tecnología 3G y metodologías de conformación de paquetes propias de GSM, al que está reemplazando. Si bien en teoría puede llegar a obtenerse una tasa de bits de hasta 14Mbps, en el estado actual de la técnica 3G sólo llega a cerca de 400kbps, con lo que triplica a la tasa de bits obtenible mediante GPRS. Una ampliación de este sistema llamada HSDPA, por High Speed Downlik Packet Access, que usa modulación 16QAM, llega a los límites cercanos a 7Mbps en el enlace hacia la terminal.

5.5 MÓDULO GSM/GPRS PLUTO.

Pluto es un módulo GSM/GPRS basado en ARM9 (Figura 5.12), el sistema operativo es Nucleus Plus el cual implementa un set de comandos AT compatibles con GSM y una pila TCP/IP.

Las características principales son:

- Módulo cuatribanda GSM850, E-GSM 900, DCS 1800, PCS1900.
- GPRS Clase B, Clase 1~12, con Esquemas de codificación CS1 a CS4.
- SMS soporta modo Texto y PDU.
- Interfaces:
 - Líneas de comunicación para tarjeta SIM 1.8/3v
 - Teclado de 6x4
 - 1 UART 115200 bps (máximo).
 - 5 GPIO.
 - Pin de reset
 - Pin de Power On
 - Entrada para micrófono de electret.
 - Salida para altavoz.
 - 1 Salida 26 MHZ o 32.768KHz.
 - Alimentación de RTC.
- Antena helicoidal integrada.
- La pila TCP/IP no contempla APIs para SMTP, PoP3 o FTP.
- No funciona en modo CSD



Figura 5.12.- Módulo PLUTO-PCBA-GSM.

5.5.1 Tarjeta de Evaluación para PLUTO1.

La tarjeta de evaluación para el módulo PLUTOPCBA- GSM está diseñada para integrar de forma inmediata una comunicación telefónica GSM/GPRS, a bajo costo (Figuras 5.13 y 5.14).



Figura 5.13.- Vista Superior, tarjeta de evaluación.

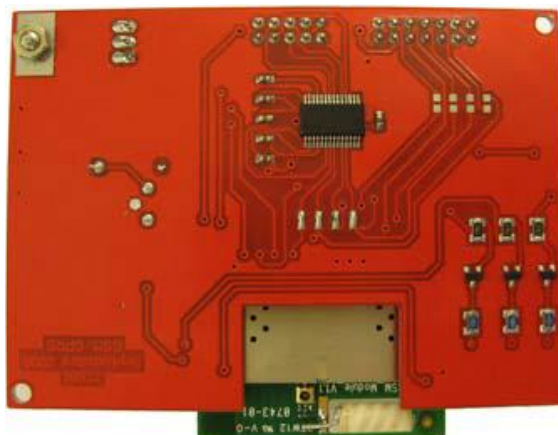


Figura 5.14.- Vista Inferior, tarjeta de evaluación.

Las características de la tarjeta de evaluación son:

- Fuente de alimentación regulada 5v y 2.8v a 80mA integrada en la tarjeta, para una entrada de C.D. de 6v a 12v (mínimo).
- Comunicación Serial, Niveles 2.8v y RS-232 (Rx, Tx, CTS y RTS).
- Salida para altavoz de 16Ω 1W.
- Entrada para micrófono de tipo electret.
- SIM holder.
- Leds monitores. Alimentación (PWR), Estado (STD) y Conexión a red (GSM_OK).
- Led monitor de GPIO (RST).
- Pin de RESET.
- Entrada para pila de RTC.

- Pin de PowerOn no disponible.
- Pines de teclado no disponible.
- No contempla protección ESD.

5.5.2 Descripción de la tarjeta de evaluación para PLUTO.

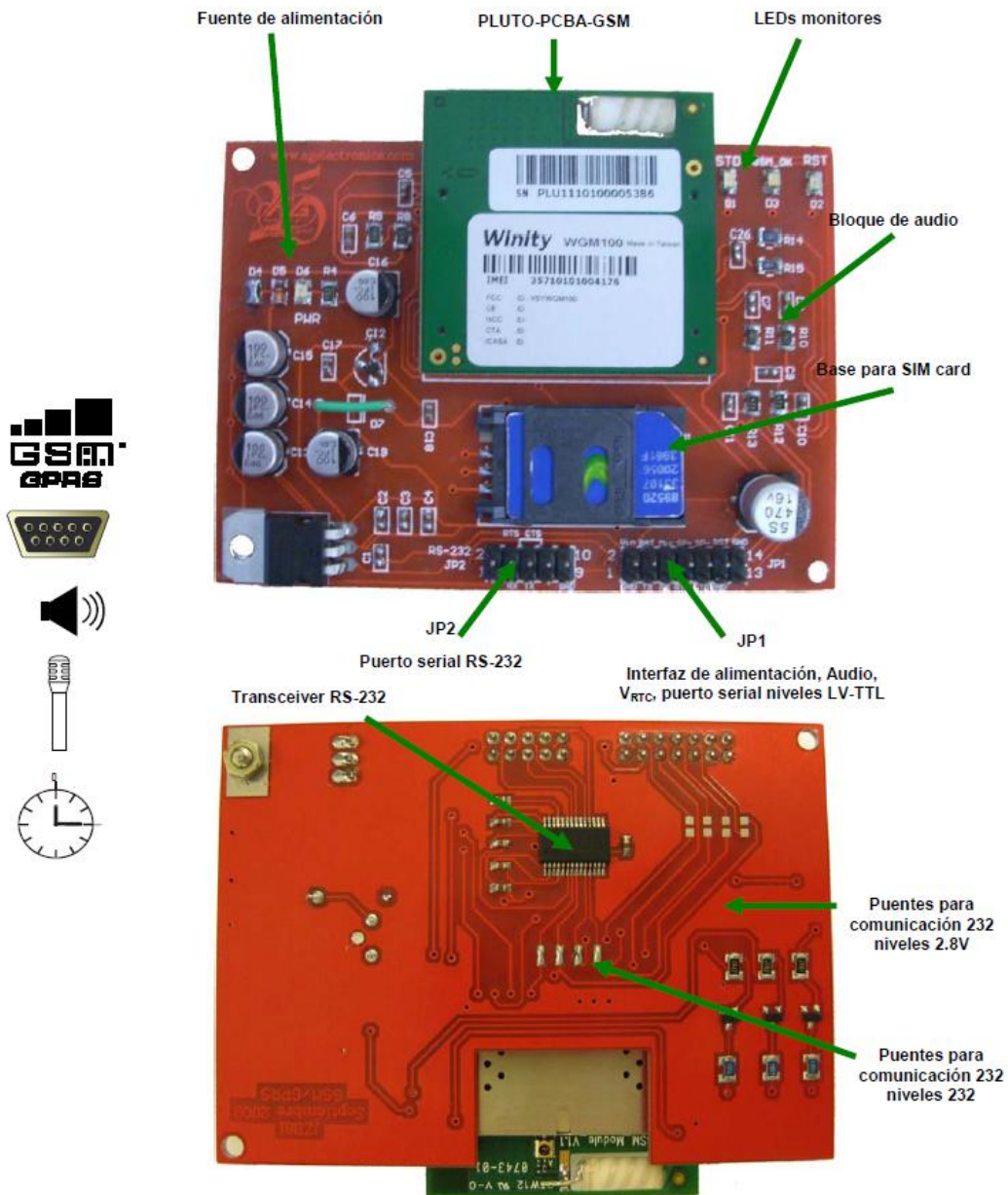


Figura 5.15.- Bloques de la tarjeta de evaluación.

5.5.3 Fuente de alimentación.

La alimentación deberá de aplicarse a través del conector JP1, pines 1 y 2, empleando una fuente de alimentación de corriente directa (C.D.) con un rango de voltaje de 6v a 12v y corriente de 1A (mínimo). Al energizar la tarjeta deberá encender el LED indicador de “energizado” (PWR) (Figura 5.16). Se recomienda no montar o desmontar el módulo mientras esté energizada la tarjeta ya que puede dañarse.



Figura 5.16.- Fuente de alimentación.

5.5.4 Pines de comunicación serial.

En JP1 (Figura 5.17) cuenta con pines de comunicación serial, el nivel de trabajo es 2.8v, por lo que a la salida de TX y RTS se tendrá como máximo este voltaje en nivel alto, RX y CTS son entadas tolerantes a 3.0v. JP2 es salida de comunicación serial a niveles RS- 232, por lo que se debe seleccionar una u otra salida mediante puentes en la parte inferior del PCB. Para comunicación con PC se debe dar salida a las terminales de comunicación serial del módulo PLUTO hacia el transceiver 232, y por tanto hacia JP2, se debe de hacer corto circuito en los puentes cercanos al transceiver (Figura 5.18). En caso contrario de comunicación con microcontrolador se deben de cortocircuitar los puentes cercanos a JP1.

ASIGNACIÓN DE PINES DEL CONECTOR JP1.

Pin No.	Nombre	Tipo	Descripción
1, 14, 11	GND	-	Tierra.
2	Vin	Ent.	Alimentación $V_{min}= 6V$, $V_{max}= 15V$.
3	TX	Sal.	Comunicación serial 232 TX nivel 2.8V.
4	BAT	Ent.	Alimentación RTC.
5	RX	Ent.	Comunicación serial 232 RX nivel 2.8V.
6	Mic	Ent.	Terminal positiva de micrófono de electret.
7	RTS	Ent.	Comunicación serial 232 RTS nivel 2.8V.
8	SP+	Sal.	Terminal positiva de altavoz.
9	CTS	Ent.	Comunicación serial 232 CTS nivel 2.8V.
0	SP-	Sal.	Terminal negativa de altavoz.
12	RST	Ent.	Pin de reset.



Figura 5.17.- Conector JP1 y asignación de pines

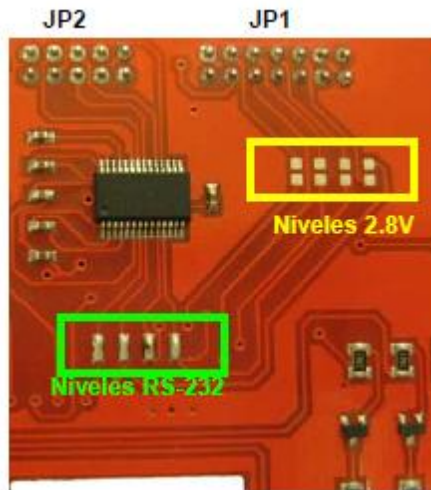


Figura 5.18.- Configuración de puentes para comunicación serial

5.5.5 Comunicación serial RS-232

La comunicación serial es de tipo DCE (conector DB9 hembra- Data Communication Equipment), por lo cual dependiendo del tipo de conexión se debe seleccionar adecuadamente el cable de conexión. JP2 esta adecuada para ser conectada a un cable serial con adaptador DB9 a cable plano y conector para cable plano de 10 terminales (Figura 5.19).

ASIGNACIÓN DE PINES DEL CONECTOR JP2.

Pin No.	Nombre	Tipo	Descripción
1, 2, 7, 8	NC	-	No conexión.
3	RX	Ent.	Línea de recepción en PC.
4	RTS	Sal.	Línea petición para enviar en PC.
5	TX	Sal.	Línea para enviar en PC.
6	CTS	Ent.	Línea listo para recibir en PC.
9	GND	-	Tierra
10	NC	-	No conexión.

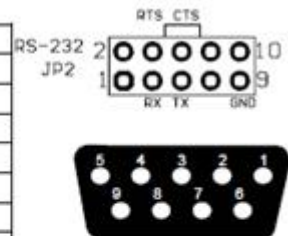


Figura 5.19.- JP2 y conector para DB9 con cable plano.

5.5.6 Tarjeta SIM.

SIM1 es un conector estándar para SIM card de 8 terminales, para operar correctamente y conectarse a la red GSM debe insertarse una SIM válida para la red de su preferencia. Las terminales de detección de SIM insertada no están conectadas hacia el módulo PLUTO, pero la detección se hace también por medio de software (Figura 5.20).

ASIGNACIÓN DE PINES SIM.

Pin No.	Nombre
CS1	$V_{SIM} 1.8V / 3V$
CS2	Reset
CS3	Reloj
CS4, CS8	Detección de SIM insertada
CS5	Tierra
CS6	No Conectar
CS7	Entrada/salida

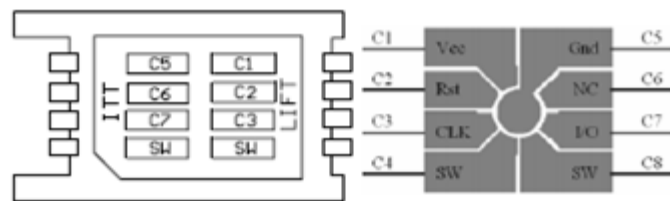


Figura 5.20.- Asignación de pines de conector SIM.

5.5.7 Entrada/Salida de Audio

La entrada de audio para acoplar un micrófono de electret es el pin 6 de JP1 respecto a tierra, no necesita de ningún circuito de polarización. En el caso de la salida de audio se debe conectar un altavoz en los pines 8 (SP+) y 10 (SP-) de JP1 (salida diferencial). Se puede emplear el auricular de un teléfono convencional, se recomienda el empleo de un jack RJ22, por ejemplo un 520249-2 de Tyco Electronics, para hacer el acoplamiento directo (Figura 5.21).

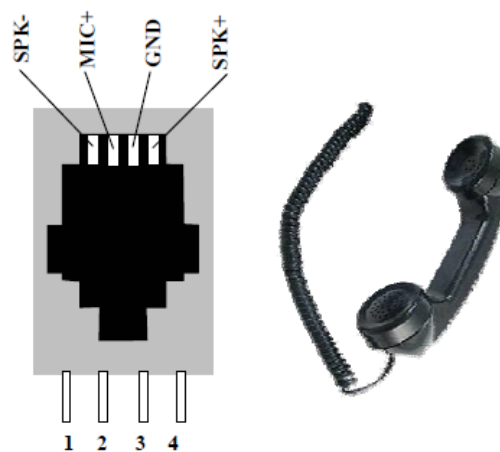


Figura 5.21.- Configuración de RJ22 y Haset de teléfono.

5.6 LOS COMANDOS AT PARA MÓDULO PLUTO

European Telecommunication Standart Institute (ETSI) GSM 07.05 define los comandos AT para módems compatibles con GSM (Figura 5.22).

5.6.1.3 Comandos AT de Configuración para Comunicación Serial

- AT+IPR: Velocidad Serial (autobaud).
- A+ICF: Configuración formato datos serial.
- AT&S1: Función DSR.
- AT&C: Función DCD.
- AT&D: Función DTR (modo datos).
- A+IFC: Configuración control de flujo.

5.6.1.4 Comandos AT para Configuración del módulo GSM

- AT+FCLASS: Configura el modo de llamada.
- AT+CCWW-BAND: Selección de banda de operación
- AT+COPS: Red (selección de operadora).
- AT+CREG: Registrar a la red.
- AT+CPAS: Estado de actividad.
- AT+CBST: Características de portadora RF.

5.6.1.5 Comandos AT para la Configuración del Hardware

- AT!GPIOCFG?: Lista de configuración
- AT!GPIO: Lectura/escritura de GPIO
- AT!GPIOMON: Monitor de entrada.

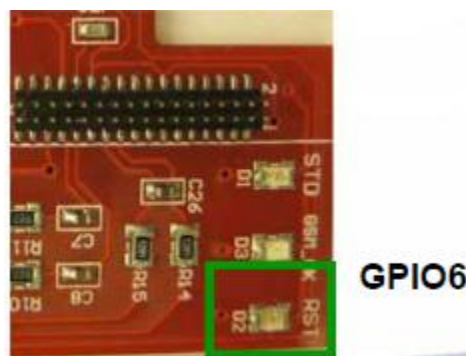


Figura 5.23.- Indicador para GPIO6

5.6.1.6 Comandos AT Configuración Hardware

- AT+CCLK: Ajuste de RTC
- AT+CFUN: Funcionalidades (CPOF)
- AT+CPWROFF: Apagar módulo. Y no puede ser encendido vía comando.
- AT+CSQ: Calidad de señal.

5.6.2 Calidad de señal

- Rssi: received signal strength indication.
0 – 6 (-113dBm a -100dBm)
7 – 11 (-99dBm a -90dBm)
12 – 16 (-89dBm a -80dBm)

17 – 21 (-79dBm a -70dBm)
22 – 26 (-69dBm a -60dBm)
27 – 31 (-59dBm a -50dBm)



Figura 5.24. - Calidad de señal

- Ber: channel bit error rate. Valores 0 – 7, 0 es el menor porcentaje de error, 99 es desconocido.

5.6.3 Servicios GSM

- Transmisión Digital: voz y datos (9.6kbps)
- FAX
- SMS de 160 caracteres en modo PDU.
- Identificación de número entrante.

5.6.4 Comandos AT para control de llamada

- ATD: Llamada (Voz/datos)
- ATA: Contestar llamada
- +++: Cadena de escape (por defecto)
- ATO: Regresar a modo datos en línea.
- ATH: Terminar llamada (colgar)ç
- ATSO: habilitar (0)/deshabilitar (x) autocontestación.
- AT+CLIP: Identificador de llamada.
- AT+CRC: Reporte tipo de llamada.
- AT+CLIP=1
- AT+CRC +CRING:ASYNC Asíncrono transparente
 +CRING:REL ASYNC Asíncrono no transparente
 +CRING:VOICE
 +CRING:FAX

5.6.5 Comandos AT para conexión SMS

Emplea PDU (Protocol Data Unit) para envío y recepción de SM. Emplea un SC (centro de servicio) para recepción/envío de SM. SM160 caracteres de 7 bits (Texto), 140 de 8 bits + 20 Bytes por compresión PDU. SM, 140 caracteres de 8 bits (imágenes y ringtones). Soporta hasta 6 cadenas concatenadas.

Para enviar un SMS:

- AT+CSCA: Centro de servicio para enviar SM (número telefónico)
- AT+CPMS: Selección de Memoria leer/escribir “SM” = SIM.
- AT+CSMP: Parámetro de modo de mensaje. Enviar, 24Hrs (periodo valido), GSM 8 bit.
- AT+CMGF: Cambia de modo, texto (1) o PDU (0).
- AT+CMGS: Enviar un mensaje (^Z=0x1A)
- AT+CMGW: Escribir mensaje en memoria. Almacenado por defecto como no enviado.
- AT+CMSS: Enviar un mensaje guardado en memoria
- AT+CMGD: Borrar mensajes

Para recibir un SMS:

- AT+CNMI: Desplegar mensaje nuevo.
- AT+CMGR: Leer un mensaje en SIM.
- AT+CMGL: lista los mensajes de acuerdo a parámetros (leído, no leído, ambos)

5.6.6 Conexiones de hardware primarias.

Para iniciar rápidamente con el kit de evaluación seguir las siguientes conexiones de hardware (Figura 5.25).

1. Insertar SIM card.
2. JP1 pin 1 (GND), pin 2 (+6v a +12v) Entrada de fuente D.C.
3. JP2 UART 0 para comandar al módulo.
4. D6 Led de power on.
5. D1 Led monitor STD. Parpadea al estar el módulo listo.
6. D3 Led monitor GSM_OK. Enciende al conectarse a red.



Figura 5.25.- Conexiones de hardware primarias.

5.6.6.1 Configuración de comunicación serial.

De fábrica la comunicación serial del módulo PLUTO esta configurado como 115200, 8, N, 1, sin control de flujo. El modo de llamada al momento de encendido es modo datos, cabe resaltar que el modo CDS no esta funcional en esta versión de firmware. La conexión aérea (RLP – Radio Link Protocol) predeterminada es 9600, asíncrono, no transparente. Para la comunicación con PC emplear cualquier software de comunicación serial como hyperterminal, XCTU, Teraterm, etc. (Figura 5.26).

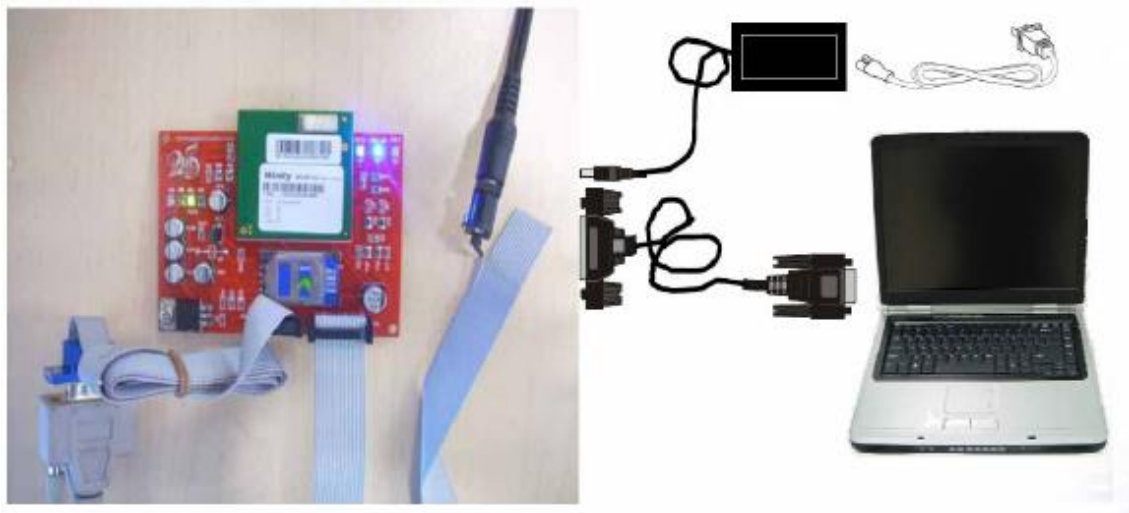


Figura 5.26.- Conexión con PC

CAPÍTULO 6 DESARROLLO DEL PROYECTO

El sistema contempla los siguientes módulos:

- 1.- Módulo básico de temperatura
- 2.- Módulo completo
- 3.- Módulo básico de Humedad
- 4.- Módulo de control

Los módulos son muy parecidos en algunas partes del hardware, y en algunos casos son un complemento uno de otro, esto no nos impide volver a recordar en cada apartado los mismos datos ya escritos en anteriores casos, debido a que cada módulo posee una tarea distinta que nos permite realizar estas repeticiones en cada caso.

6.1 MÓDULO BÁSICO DE TEMPERATURA

El módulo contempla como función el monitoreo (PIC16F877A) de la temperatura (LM35DZ) accionando una carga al tener una temperatura elevada y apagándola cuando esta llegue a una temperatura estable, a su vez realiza la detección del día o de la noche (Fotorresistencia) según sea el caso y activa o desactiva una carga luminosa. Para este caso el módulo se divide en dos partes: hardware (circuito) y software (programa fuente del pic).

Dadas las especificaciones del fabricante de los componentes, se tiene un circuito referente para el sensor LM35DZ mostrado en la figura 6.1

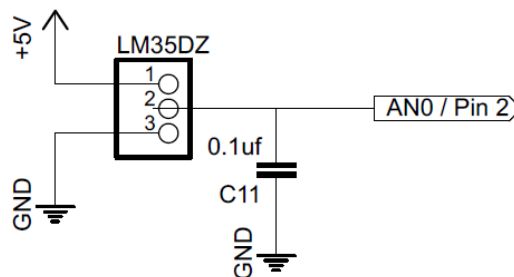


Figura 6.1.- Circuito para sensor LM35DZ

Para evitar que el convertidor este oscilando por las variaciones de voltaje, se ha puesto un capacitor (C11) de tantalio de 0.1uf para eliminar los rizados y estabilizar la lectura del convertidor analógico digital.

Para poder identificar el día o la noche se ha utilizado una fotorresistencia que varía su valor óhmico (50Ω a $1.6M\Omega$) con respecto a la variación de la luz, se utilizara en una configuración de divisor de voltaje con otra resistencia para poder tener una variación de voltaje de 0v a 5v (Figura 6.2).

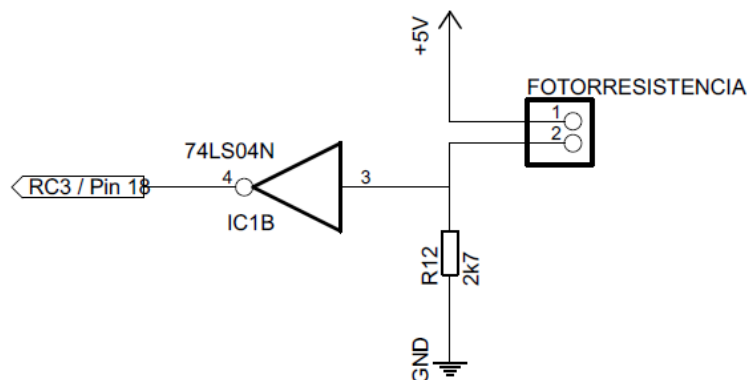


Figura 6.2.- Circuito de detección de día o noche.

La configuración del divisor de voltaje tiene dos estados, cuando hay luz la fotorresistencia (LDR) alcanza un valor medido de 50Ω y en ausencia de luz se tiene un valor medido de $1.6M\Omega$, así que por ley de ohm se tiene:

$$V = R * I$$

Siendo:

R= valor de la resistencia en Ω

I= valor de la corriente en Amp

El voltaje en cada resistencia para la configuración en serie es:

$$V_{R12} = i * R_{12}$$

$$V_{LDR} = i * LDR$$

Aplicando la ley de las tensiones de Kirchhoff en la imagen de la Figura 6.2, se tiene lo siguiente:

$$-V + V_{R12} + V_{LDR} = 0$$

$$V = V_{R12} + V_{LDR}$$

$$V = i(R_{12} + LDR)$$

Para determinar la tensión a lo largo de cada resistor de la Figura 6.2 reemplazamos y se obtiene:

$$V_{R12} = V \frac{R_{12}}{R_{12} + LDR} \quad V_{LDR} = V \frac{LDR}{LDR + R_{12}}$$

Para conocer el valor de R_{12} cuando es de día se tiene que el valor de la LDR es igual a 50Ω y se propone un voltaje de salida (V_{R12}) de 4.9 v se tiene que:

$$R_{12} = \frac{50\Omega (4.9v)}{(5V - 4.9v)} = 2450\Omega$$

Observando que el resultado anterior no es un valor comercial, existen dos posibilidades para la resistencia, la de menor valor es $2.2k\Omega$ y la de mayor valor es de $2.7k\Omega$. Para asegurar el voltaje se utilizara la de $2.7k\Omega$.

En base a lo anterior, se calcula el nuevo voltaje de salida para la resistencia de $2.7k\Omega$

$$V_{R12} = 5v \frac{2.7K\Omega}{2.7K\Omega + 50\Omega} = 4.9090v$$

Ahora bien, para conocer el voltaje de salida cuando es de noche, considerando que el valor de LDR es de $1.6M\Omega$, se tiene lo siguiente:

$$V_{R12} = 5v \frac{2.7K\Omega}{2.7K\Omega + 1.6M\Omega} = 0.0084v$$

Debido a que a la salida de V_{R12} hay una compuerta inversora, cuando sea de día existe un nivel lógico bajo, así mismo cuando sea de noche existirá un nivel lógico alto.

Para conectar el pin de transmisión del microcontrolador al pin de recepción del xbee se requiere bajar el voltaje de salida del microcontrolador de $5v$ a un voltaje de $3.3v$ apto para la xbee, por lo tanto, se utiliza un divisor de voltaje como se muestra a continuación (Figura 6.3).

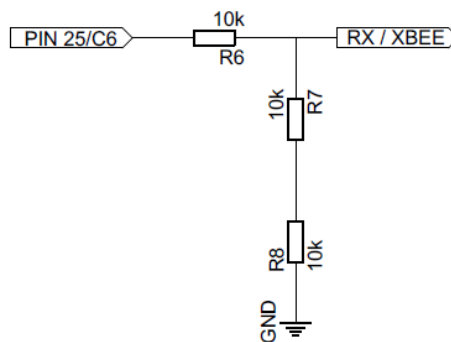


Figura 6.3.- Divisor de voltaje para xbee

Del circuito anterior se tiene que las resistencias $R8$ y $R7$ están en serie, por lo tanto tendrán una resistencia equivalente (R_{eq}), proponiendo el valor de $R6 = 10k\Omega$ se tiene:

$$V_{R6} = 5 - 3.3 = 1.7v$$

$$V_{R6} = V \frac{R_6}{R_{eq} + R_6}$$

$$(V_{R6})(R_{eq} + R_6) = V(R_6)$$

$$R_{eq} = \frac{V(R_6) - (V_{R6})(R_6)}{V_{R6}}$$

$$R_{eq} = \frac{(5)(10000) - (1.7)(10000)}{(1.7)} = 19412\Omega$$

Se decide poner dos resistencias en serie de 10KΩ para poder redondear a 20kΩ, con este nuevo valor se tiene un voltaje de:

$$V_{Req} = 5v \frac{20K\Omega}{20K\Omega + 10k\Omega} = 3.3333v$$

Por lo tanto R8 y R7 tendrán el valor de 10kΩ.

Para poder utilizar la xbee se requiere que sea alimentada con 3.3v, para ello se utiliza el regulador de voltaje LF33CV (Figura 6.4), con el cual reduciremos el voltaje de alimentación de 5v a 3.3v.

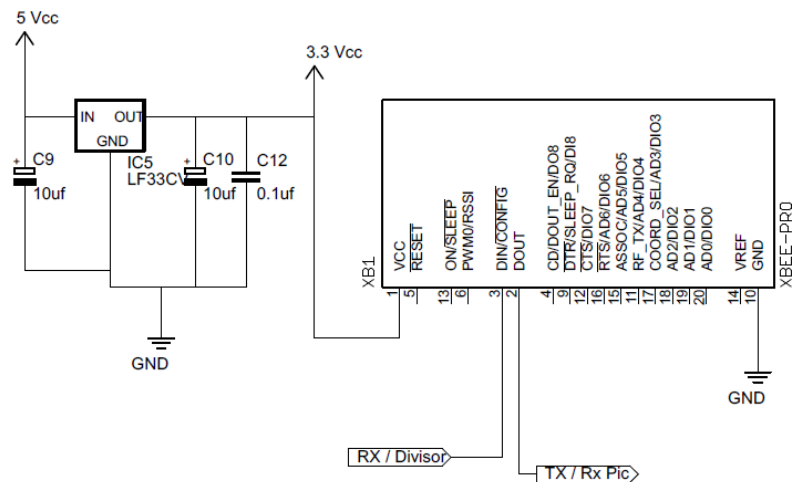


Figura 6.4.- Alimentación y conexión de xbee

Como se va a utilizar la configuración del convertidor analógico digital con voltaje de referencia, se utiliza otra vez el divisor de voltaje para poder obtener un voltaje de 2.56v que se conectara al Vref+ del microcontrolador (Figura 6.5).

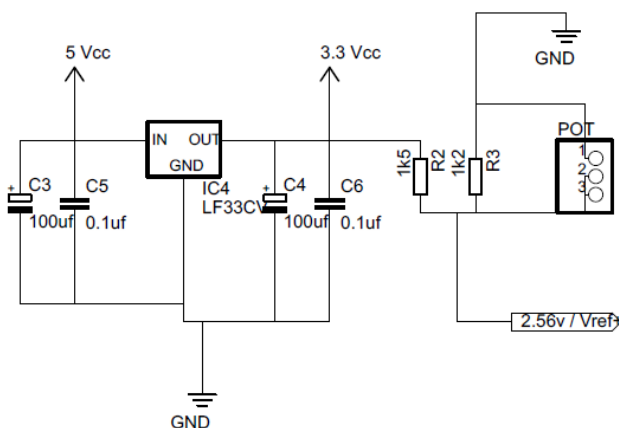


Figura 6.5.- Configuración para el voltaje de referencia de 2.56v

De la fórmula se tiene:

$$V_{R3} = V \frac{R_3}{R_3 + R_2}$$

Se propone el valor de la resistencia $R_3=1.5K\Omega$ con un voltaje de $V_{R3}=2.56v$, por lo tanto se procede a despejar R_2 :

$$R_2 = \frac{(V)(R_3) - (V_{R3})(R_3)}{V_{R3}}$$

$$R_2 = \frac{(3.3)(1500) - (2.56)(1500)}{2.56} = 433.6\Omega$$

Como R_2 debe de ser un valor exacto, se procede a sustituir a R_2 por un potenciómetro para asegurar el voltaje de 2.56 volts.

Los datos se mostraran en un LCD de 16 líneas y 2 renglones (16x2) cuyos mensajes corresponderán a la temperatura o humedad de cada uno de los módulos (Figura 6.6)

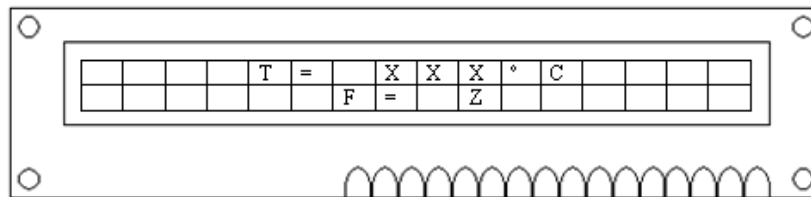


Figura 6.6.- Posición de los datos en pantalla LCD

Debido a que es el módulo básico de temperatura, solo tendrá dos parámetros en monitoreo. La “T” corresponde a la temperatura que será de 0 a 150°C (expresado en XXX para fin práctico), en el segundo renglón se muestra el monitoreo referente a la Fotorresistencia “F” que nos indica con una “D” si es día o una “N” si es de noche (expresado con una Z para fin práctico).

Así, cada salida activara un relevador para poder manejar una amplia gama de voltajes, además el microcontrolador requiere de un cristal con sus respectivos capacitores, tal circuito es mostrado a continuación (Figura 6.7).

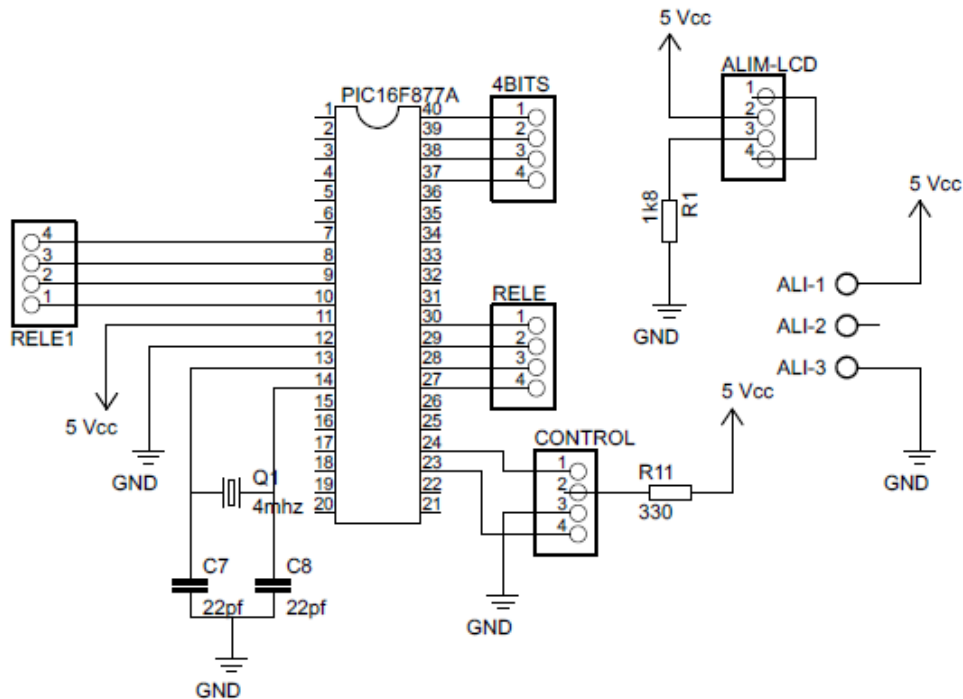


Figura 6.7.- Resto del circuito para módulo básico de temperatura

Los conectores RELE y RELE2 son para la activación de la carga (ventiladores, focos, etc). El conector 4BITS es donde se colocaran los 4 bits más significativos de la LCD, mientras que los pines RS y E se conectaran en CONTROL, por último los conectores ALI y ALI-LCD son para conectarlos a la alimentación correspondiente.

Para este caso utilizaremos los relevadores de la marca SUN HOLD, específicamente utilizaremos el relevador con número de parte RAS-0510. Si la bobina del relé tiene 69Ω de resistencia y funciona a 5 vcc, se puede calcular el consumo de corriente que tiene el relé.

Por ley de ohm se tiene:

$$V = I_{rele} * R_{rele}$$

$$I_{rele} = \frac{V_{cc}}{R_{rele}} = \frac{5}{69} = 72 \text{ mA}$$

Una corriente de 72 mA es demasiado elevada para conectar directamente a una salida del microcontrolador. Por lo tanto es necesario agregar un transistor que pueda manejar la corriente que el relé necesita. El transistor es del tipo NPN tenemos la ventaja de poder usar relé de cualquier tensión no obstante el circuito lógico sea de 5v, es decir, el transistor sirve también para adaptar los niveles de tensión (Figura 6.8).

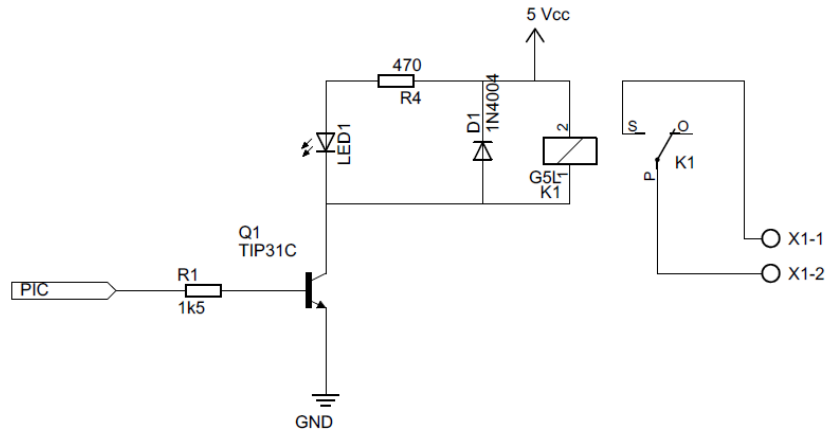


Figura 6.8.- Conexión de un relevador a un microcontrolador

Para este caso se ha conectado un led como indicador visual en paralelo con el relevador, este led servirá como indicador para saber en qué momento debe de estar o no activado el relevador.

De acuerdo a Ley de corrientes de Kirchoff tenemos que en el circuito:

$$I_c = I_{led} + I_{relay}$$

Si se coloca una resistencia de 470Ω al led, su corriente será de:

$$I_{led} = \frac{V}{R_{rele}} = \frac{5}{470} = 10.6 \text{ mA}$$

Sustituyendo dichas corrientes tenemos que la corriente de I_c será:

$$I_c = 72\text{mA} + 10.6\text{mA} = 82.6 \text{ mA}$$

Para calcular la resistencia de base del transistor, se tiene que las características del TIP31C tiene una ganancia estática (h_{FE}) de 50. Por lo tanto, para obtener la corriente de salida de 82.6 mA necesaria, la corriente de entrada debería ser:

$$I_b = \frac{I_c}{h_{FE}} = \frac{82.6 \text{ mA}}{50} = 1.65 \text{ mA}$$

Pero para asegurarnos que el transistor va a conmutar (va a entrar en saturación) vamos a hacer que este suministro alcance, por lo tanto $I_b = 3 \text{ mA}$.

De acuerdo al circuito se tiene que la ecuación característica para la base del transistor es:

$$R_b = \frac{V - V_{BE}}{I_b}$$

Sustituyendo valores tenemos que el valor de la resistencia es:

$$R_b = \frac{5 - 0.7}{0.003} = 1433.33 \Omega$$

Dado que el valor comercial de la resistencia más próxima es de 1.5 kΩ, se vuelven a realizar los cálculos para saber cuál es el nuevo valor de la corriente de base y de colector a fin de disminuir la que requiere el relevador y el led.

$$I_b = \frac{V - V_{BE}}{R_b} = \frac{5 - 0.7}{1500} = 2.86 \text{ mA}$$

Por lo tanto la corriente de colector será:

$$I_c = I_b * h_{FE} = (2.86 \text{ mA})(50) = 143 \text{ mA}$$

El diodo en paralelo con la bobina del relé cumple la función de absorber las tensiones que se generan en todos los circuitos inductivos. Por otro lado el conector X1 es una bornera que nos permitirá conectar las entradas y salidas a nuestro relevador. Hasta este punto se ha culminado el diseño de todo el circuito a utilizar, ahora toca el diseño de la parte del software. El programa fuente del microcontrolador esta hecho en código ensamblador utilizando el programa MPLAB v8.56 de la empresa Microchip. El programa fuente se puede consultar en la sección de Anexos parte A.

6.2 MÓDULO DE HUMEDAD

La principal función de este módulo es monitorear constantemente el cambio de humedad en el ambiente de un jardín, dependiendo del estado en que se encuentre (nivel bajo de humedad) va a activar una regadera para aumentar la humedad y se detendrá cuando esta llegue a un nivel de humedad alto. Además de tener una salida luminosa en caso de que sea de noche y apagándola en caso que sea de día. El hardware de este módulo es parecido en gran medida que el módulo básico de temperatura, a diferencia que este módulo ya no posee divisor de voltaje para el voltaje de referencia del microcontrolador, debido a que el sensor de humedad utilizado tiene un rango de 0 a 3.3 volts en la salida, por tal, se seleccionara el voltaje de referencia de 3.3 volts, así que la R2 será sustituida por un cable de 0Ω y la R3 en conjunto con el POT no serán colocados en este circuito (Figura 6.9).

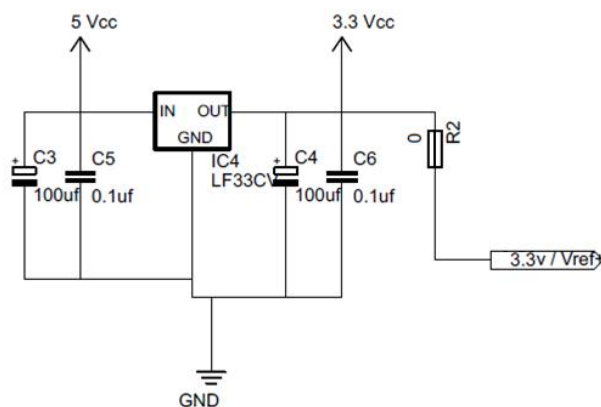


Figura 6.9.- Circuito para voltaje de referencia de 3.3v

Otro aspecto que cambia es los datos desplegados en el LCD (Figura 6.10), ya que esta vez se cambia la temperatura por la humedad, se sigue conservando el parámetro del día y de la noche.

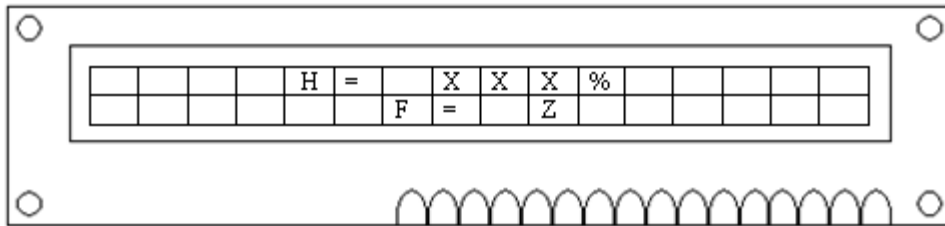


Figura 6.10.- Posición de los datos en pantalla LCD

Debido a que es el módulo básico de humedad, solo tendrá dos parámetros en monitoreo. La “H” corresponde a la humedad que será de 0 a 100 % (expresado en XXX para fin práctico), en el segundo renglón se muestra el monitoreo referente a la Fotorresistencia “F” que nos indica con una “D” si es día o una “N” si es de noche (expresado con una Z para fin práctico).

En cuanto al resto del hardware, será el mismo que el módulo anterior.

Hasta este punto se ha culminado el diseño de todo el circuito a utilizar, ahora toca el diseño de la parte del software. El programa fuente del microcontrolador esta hecho en código ensamblador utilizando el programa MPLAB v8.56 de la empresa Microchip. El programa fuente se puede consultar en la sección de Anexos parte B.

6.3 MÓDULO COMPLETO

Este módulo posee casi todas las aplicaciones de los sensores a excepción del sensor de humedad (ya que solo es para el control de humedad en jardines). Se le han incorporado el monitoreo de la apertura de una puerta mediante el uso de un sensor de efecto hall, así mismo se le incorporo la función de la detección de las personas a través de un sensor PIR. A excepción del control del sensor de efecto hall y PIR, el hardware es el mismo para el sensor de temperatura, la fotorresistencia, la conexión de la xbee, la alimentación de la xbee, el circuito de referencia de voltaje y la conexión de la LCD. Por tal motivo solo describiremos el hardware del sensor PIR y de efecto hall. Dadas las características del sensor de efecto hall descritas en un capítulo anterior, se tiene el siguiente circuito característico (Figura 6.11).

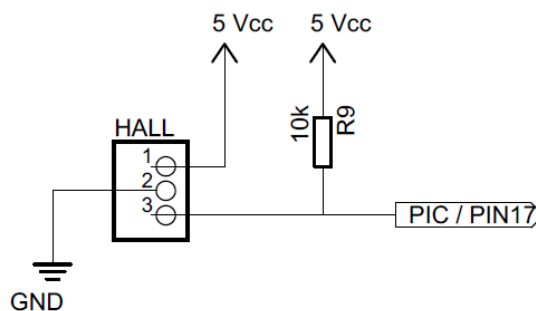


Figura 6.11.- Conexión del sensor de efecto hall

La conexión de este sensor solo es muy simple debido a que solo tiene dos estados, uno bajo y otro alto. Cuando el sensor esta en ausencia de un campo magnético este no adquiere ningún valor en su pin de salida, por ello es importante conectar a su salida una resistencia de pull-up para tener un nivel alto a la salida y para asegurar el cambio de nivel lógico, cuando un campo magnético entra en contacto con el sensor, este cambia el estado lógico de su pin de salida a un nivel bajo.

Para la detección de personas aremos uso del sensor PIR. El sensor solo requiere de componentes mínimos para su funcionamiento, al igual que el sensor de efecto hall, el sensor pir requiere de una resistencia de pull-up para asegurar el cambio de niveles lógicos de un nivel bajo a un nivel alto (Figura 6.12).

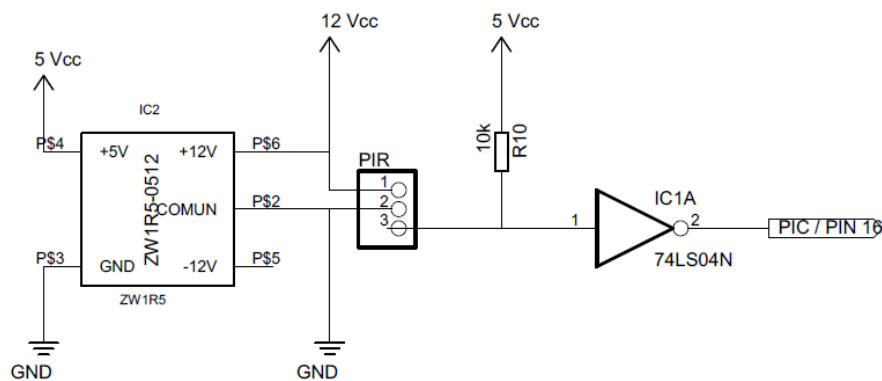


Figura 6.12.- Esquema para el funcionamiento del sensor PIR.

De acuerdo a la figura 6.12 el sensor PIR funciona a 12 volts, por lo tanto aremos uso de un convertidor de voltaje DC-DC con el fin de usar una sola alimentación (5 Vcc) para todo el modulo, el convertidor de voltaje DC-DC que utilizamos es el ZW1R5-0512 (Figura 6.13).



Figura 6.13.- Convertidor de voltaje DC-DC ZW1R5-0512

Este circuito posee la característica de ser alimentado con un voltaje de 5 volts y tener a su salida un voltaje de 12 volts simétricos y entregar 60 mA. Por otro lado se tiene una compuerta inversora conectada a la salida del PIR, por lo que al estar el sensor PIR inactivo se tendrá un nivel alto debido a su resistencia de pull-up, este voltaje entra a la compuerta inversora y a su salida nos entrega un nivel lógico bajo, por el lado contrario, si el sensor es activado, su salida pasa a nivel bajo y por lo tanto a la salida de la compuerta inversoras tendremos un nivel lógico alto.

El resto del circuito ya ha sido descrito en el capítulo anterior. Con respecto a la LCD, seguiremos usando una de 16x2, solo cambiando la forma en que se muestran los datos de los sensores (Figura 6.14).

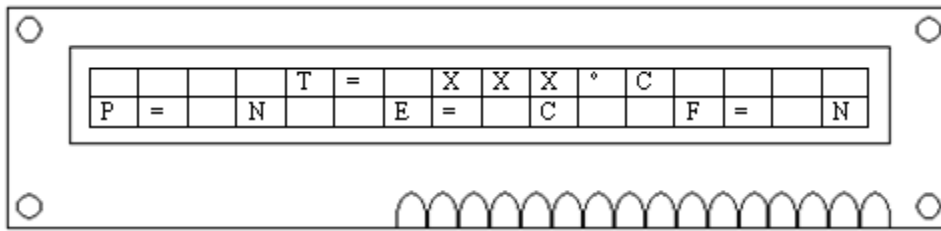


Figura 6.14.- Posición de los datos en pantalla LCD

Donde las “X” indican la temperatura en tiempo real, la “F” nos indica la función de la fotorresistencia con una “N” para noche y una “D” para día, los otros dos datos (P, E) comienzan a funcionar cuando se ha activado la función de alarma, una vez activada, la “P” nos indica la función del sensor PIR, con una “P” nos indica la presencia humana en la zona (5 metros radiales) y una “N” nos indica que no hay presencia humana en la zona. La letra “E” nos hace referencia al sensor de efecto hall, con una “C” nos indica que las puertas están cerradas y con una “A” que han sido abiertas. Cuando los sensores (PIR y efecto hall) han sido activados, el microcontrolador envía la alarma vía inalámbrica a través de la xbee al módulo de control para indicarle la ubicación del intruso y activar las alarmas sonoras y silenciosas.

Hasta este punto se ha culminado el diseño de todo el circuito a utilizar, ahora toca el diseño de la parte del software. El programa fuente del microcontrolador esta hecho en código ensamblador utilizando el programa MPLAB v8.56 de la empresa Microchip. El programa fuente se puede consultar en la sección de Anexos parte C.

Debido a que este mismo módulo se colocara en tres ubicaciones diferentes, no se repetirá el hardware y el software de nuevo, solo se hará la corrección de las ubicaciones de las cajas y de algunos parámetros que se han de cambiar para los otros dos módulos.

De acuerdo ala caja ubicada en el dormitorio 2 y a la ubicación del módulo tenemos:

```

.....
; CAJA CON NUMERO: 223
; UBICACIÓN: "D2"
; PROTOCOLO: "W"
.....

.....
; MODO ALARMA.- EN CASO DE PRESENCIA SE REALIZA LO SIGUIENTE
.....
activa1
.....
call    T_20ms
movlw  d'87'           ; carga una W de la ubicación de la caja
call    envia
call    Retardo_1s
bcf     sirena
return
.....
;MODO ALARMA.- EN CASO DE HALL SE REALIZA

```



```

.....
.....
activa2
.....
.....
call    T_20ms
movlw  d'87'           ; carga una W de la ubicación de la caja
call    envia
call    Retardo_1s
bcf     sirena
return

```

```

.....
; PARTE DEL PROGRAMA PARA LA COMUNICACIÓN MAIK/ESCLAVO PARA
; OBTENER TEMPERATURA DESEADA Y MODO NORMAL O ALARMA
.....

```

```

comunica
btfss  PIR1,RCIF      ;Pregunta si el buffer de RX es full
goto   esperemos     ; repetimos 50 veces y después abortamos
movf   RCREG,W        ;Si, W=RCREG y pone a cero el RCIF
sublw  d'223'        ; si es igual al # de la caja
btfss  STATUS,Z      ; no es la caja, así que sigue en funcionando
goto   sigueaqui
call   T_20ms
movlw  'W'           ; responde que si es la caja y espera a recibir
call   envia

```

De acuerdo a la caja ubicada en la cocina, los parámetros siguientes son los que se cambian:

```

.....
; CAJA CON NUMERO: 224
; UBICACIÓN: "C"
; PROTOCOLO: "C"
.....

```

```

.....
; MODO ALARMA.- EN CASO DE PRESENCIA SE REALIZA LO SIGUIENTE
.....

```

```

activa1
.....
.....
call    T_20ms
movlw  d'67'           ; carga una C de la ubicación de la caja
call    envia
call    Retardo_1s
bcf     sirena
return

```

```

.....
;MODO ALARMA.- EN CASO DE HALL SE REALIZA
.....

```

```

activa2
.....
.....
call    T_20ms

```

```

movlw d'67'           ; carga una C de la ubicación de la caja
call  envia
call  Retardo_1s
bcf   sirena
return

```

```

; PARTI DEL PROGRAMA PARA LA COMUNICACIÓN MAIK/ESCLAVO PARA
; OBTENER TEMPERATURA DESEADA Y MODO NORMAL O ALARMA

```

comunica

```

btfss PIR1,RCIF      ;Pregunta si el buffer de RX es full
goto  esperamos     ; repetimos 50 veces y después abortamos
movf  RCREG,W        ;Si, W=RCREG y pone a cero el RCIF
sublw d'224'        ; si es igual al # de la caja
btfss STATUS,Z      ;
goto  sigueaqui     ; no es la caja, así que sigue en funcionando
call  T_20ms
movlw 'C'           ; responde que si es la caja y espera a recibir
call  envia

```

6.4 MÓDULO DE CONTROL

El módulo contempla la función de comunicarse con los demás módulos para poder proporcionar a cada uno los parámetros de trabajo, así mismo el poder recibir cualquier información proveniente de los módulos. Posee la cualidad de controlar al modulo GSM y la información proveniente de él. Al igual que los casos anteriores se divide en dos partes: hardware (circuito) y software (programa fuente del pic).

Debido a que el microcontrolador solo tiene un puerto para comunicación serial, y para este caso se requiere el control del xbee y del modulo GSM, se hará un multiplexaje para el uso por un tiempo de trabajo de cada uno, Para lograr este fin se utilizara el circuito integrado 74LS126N, que es una compuerta triestado que usaremos como multiplexor (Figura 6.15).

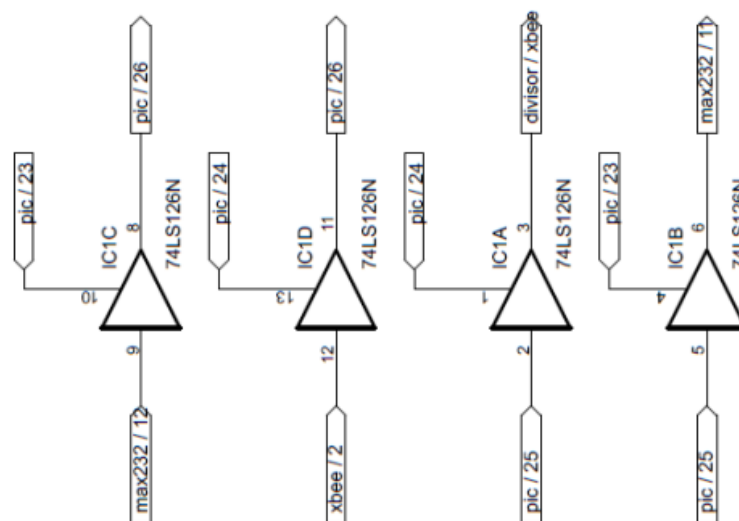


Figura 6.15.- Multiplexaje para xbee y modulo GSM Pluto

Donde los pines 4, 1, 13, 10 serán habilitados dependiendo de la tarea a realizar, serán básicamente los pines de control entre la xbee y el modulo GSM.

Para conectar el pin de transmisión del microcontrolador al pin de recepción del xbee se requiere bajar el voltaje de salida del microcontrolador de 5v a un voltaje de 3.3v apto para la xbee, por lo tanto se utiliza un divisor de voltaje como se muestra a continuación (Figura 6.16)

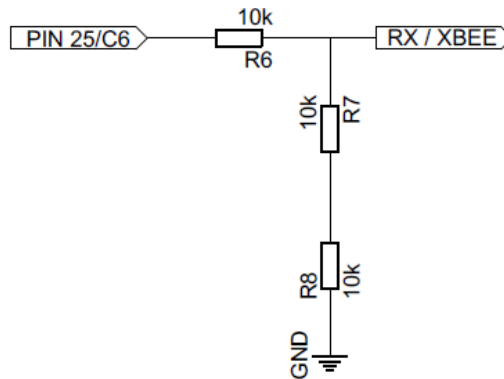


Figura 6.16.- divisor de voltaje para xbee

Del circuito anterior se tiene que las resistencias R8 y R7 están en serie, por lo tanto tendrán una resistencia equivalente (R_{eq}), proponiendo el valor de $R_6 = 10k\Omega$ se tiene:

$$V_{R_6} = 5 - 3.3 = 1.7v$$

$$V_{R_6} = V \frac{R_6}{R_{eq} + R_6}$$

$$(V_{R_6})(R_{eq} + R_6) = V(R_6)$$

$$R_{eq} = \frac{V(R_6) - (V_{R_6})(R_6)}{V_{R_6}}$$

$$R_{eq} = \frac{(5)(10000) - (1.7)(10000)}{(1.7)} = 19412\Omega$$

Se decide poner dos resistencias en serie de $10K\Omega$ para poder redondear a $20k\Omega$, con este nuevo valor se tiene un voltaje de:

$$V_{Req} = 5v \frac{20K\Omega}{20K\Omega + 10k\Omega} = 3.3333v$$

Por lo tanto R8 y R7 tendrán el valor de $10k\Omega$.

Para poder utilizar la xbee se requiere que sea alimentada con 3.3v, para ello se utiliza el regulador de voltaje LF33CV (Figura 6.17), con el cual reduciremos el voltaje de alimentación de 5v a 3.3v.

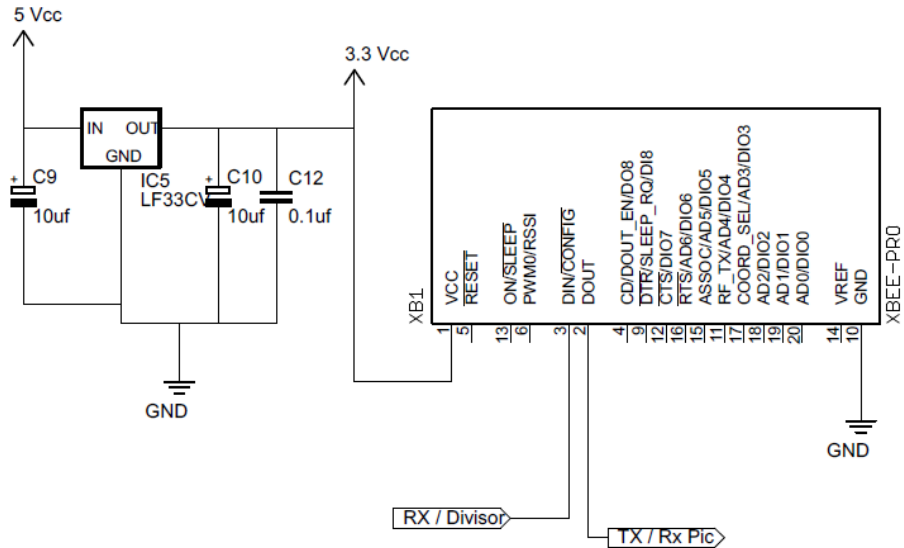


Figura 6.17.- Alimentación y conexión de xbee

Para poderse comunicar el microcontrolador con el módulo GSM hay que cambiar los niveles TTL que salen del microcontrolador a niveles RS232, por tal razón se debe introducir en el circuito un interface que traduzca los datos del microcontrolador al puerto y viceversa (Figura 6.18)

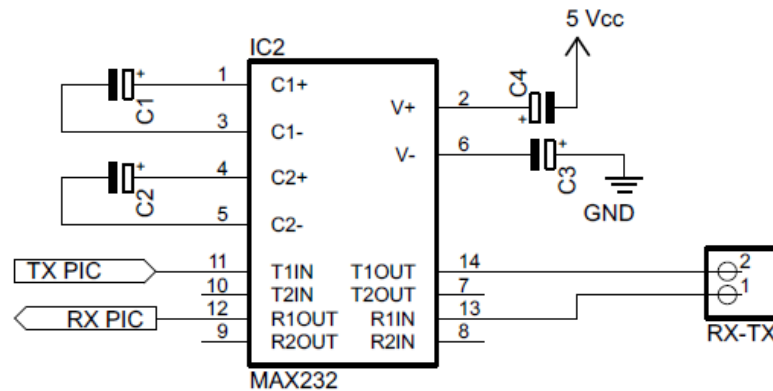


Figura 6.18.- Interface niveles TTL a RS232.

Los datos se mostraran en un LCD de 20 líneas y 2 renglones (20x2) cuyos mensajes corresponderán a las temperaturas y humedad de todos los módulos (Figura 6.19) y al menú de selección de temperatura (Figura 6.20a) y al menú de selección de humedad (Figura 6.20b).

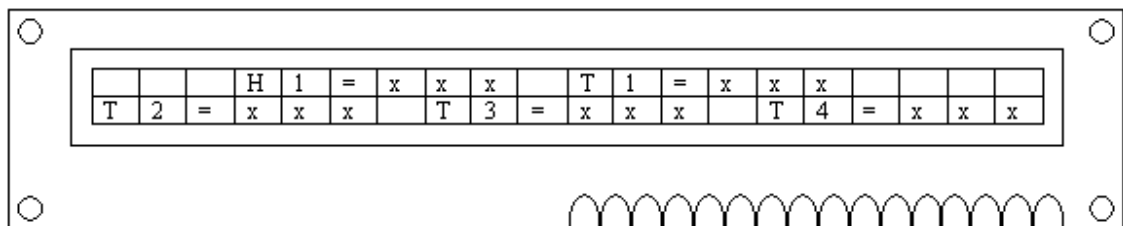


Figura 6.19.- Resultado de solicitud de temperaturas y humedad

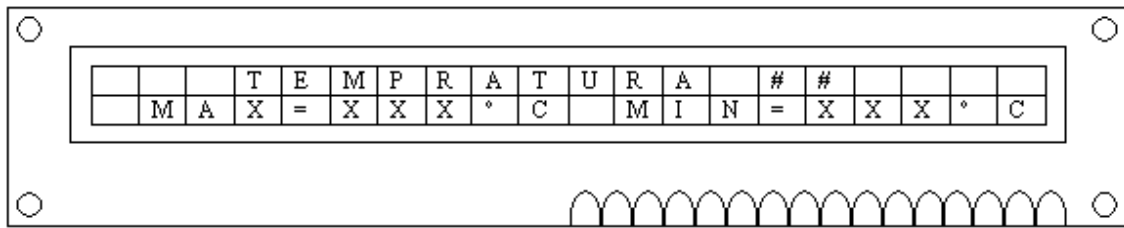


Figura 6.20a.- Menú de selección de temperaturas

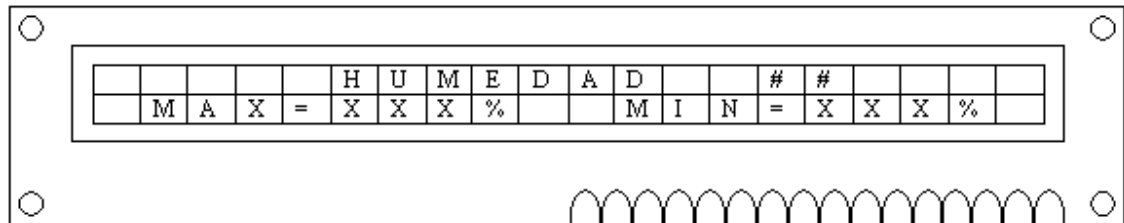


Figura 6.20b.- Menú de selección de humedad

De las ubicaciones anteriores para la LCD, las “x” corresponden al valor numérico de la temperatura (0 a 150°C) y al valor de la humedad (0 a 100 %) y los “#” la ubicación del módulo (dormitorio, jardín, cocina, etc).

Se requiere de 2 botones para ingresar la temperatura/humedad máxima y mínima, un botón para solicitar a todos los módulos sus temperaturas en tiempo real, y un último botón para el menú. Además el microcontrolador requiere de un cristal con sus respectivos capacitores, y se han de colocar indicadores visuales que permitan identificar las diferentes tareas a realizar (Figura 6.21).

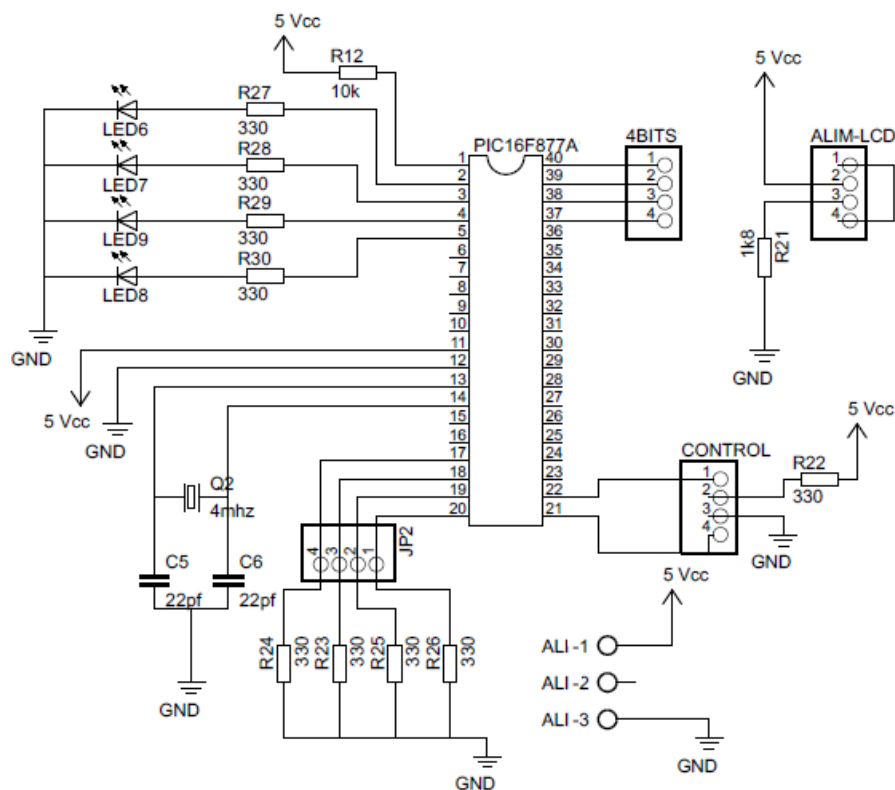


Figura 6.21.- Conexión del microcontrolador, botones y lcd.

El conector JP2 es donde se colocan los 4 botones para la manipulación del menú, solicitud de datos y selección de parámetros. Se colocan además 4 leds que nos van a permitir conocer el estado de los procesos. El conector 4BITS es donde se colocaran los 4 bits más significativos de la LCD, mientras que los pines RS y E se conectaran en CONTROL, por último los conectores ALI y ALI-LCD son para conectarlos a la alimentación correspondiente.

Hasta este punto se ha culminado el diseño de todo el circuito a utilizar, ahora toca el diseño de la parte del software. El programa fuente del microcontrolador esta hecho en código ensamblador utilizando el programa MPLAB v8.56 de la empresa Microchip. El programa fuente se puede consultar en la sección de Anexos parte D.

6.5 IMPLEMENTACIÓN.

Dado que los módulos de monitoreo (humedad y temperatura) tienen la misma estructura básica, se hará el modelo del módulo completo para los tres casos, y posteriormente no se colocaran los componentes que salgan sobrando en la aplicación, tales cambios ya han sido descrito en capítulos anteriores. Dicho circuito representativo es el descrito en la figura 6.22.

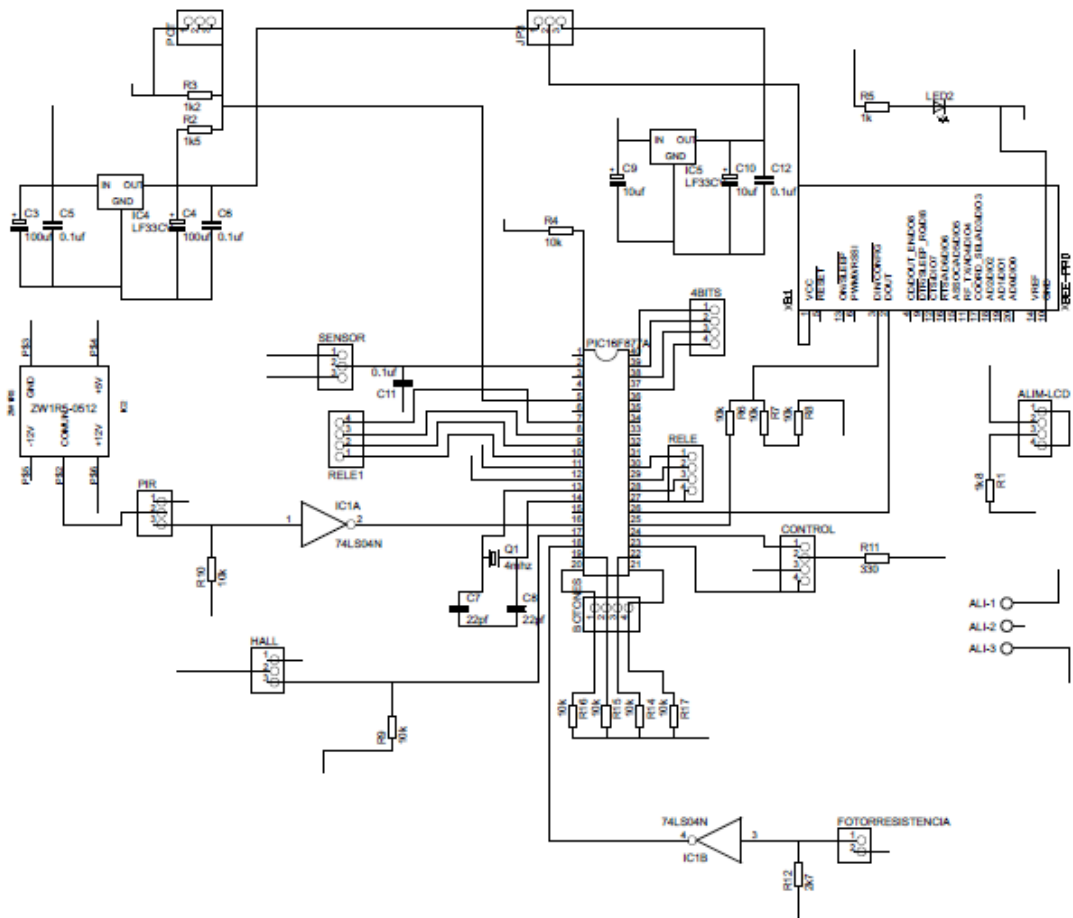


Figura 6.22.- Circuito del módulo completo.

A continuación se procede a realizar el circuito impreso, para lo cual se utilizó el software EAGLE v5.6 profesional, quedando el impreso de la siguiente forma:

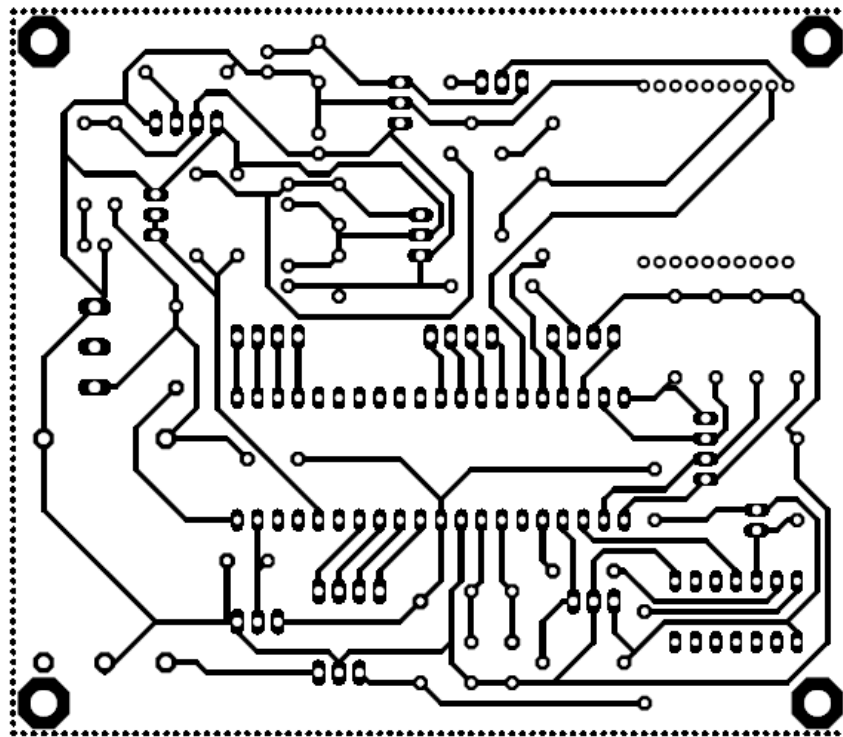


Figura 6.23.- Pistas circuito impreso.

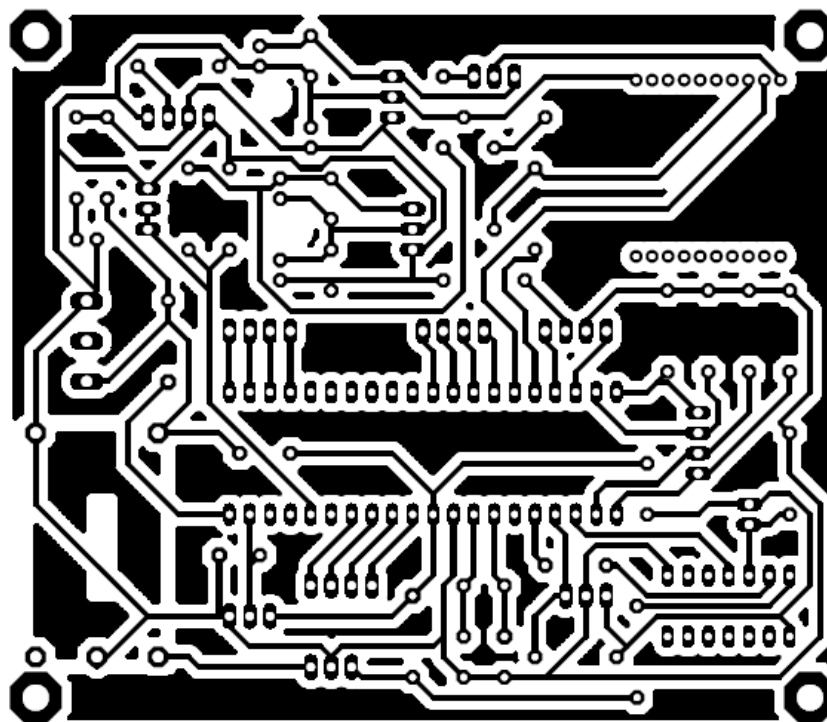


Figura 6.24.- Plano de tierra.

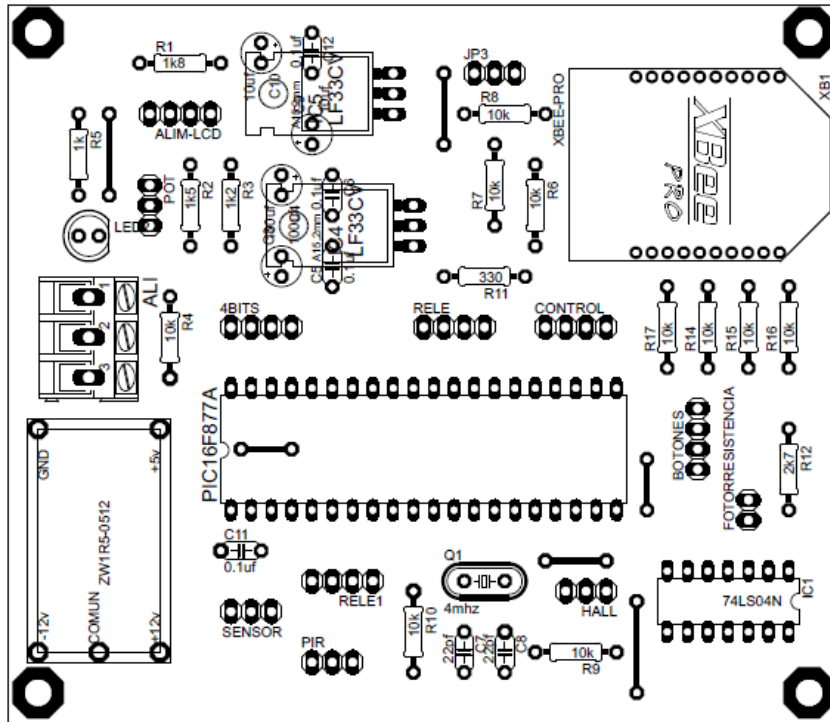


Figura 6.25.- Cara de componentes.

Con respecto al módulo de control, su circuito esta descrito en la figura 6.26.

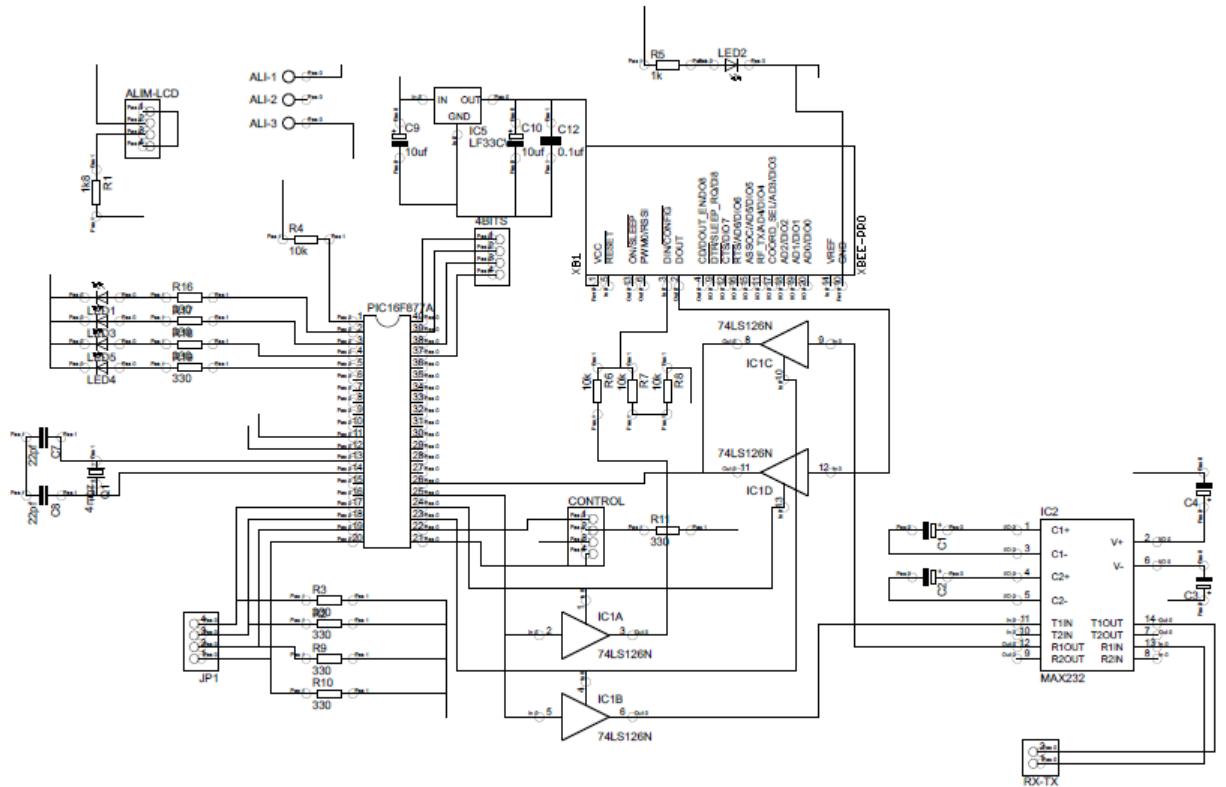


Figura 6.26.- Esquemático para módulo de control.

A continuación se procede a realizar el circuito impreso, para lo cual se utilizó el software EAGLE v5.6 profesional, quedando el impreso de la siguiente forma:

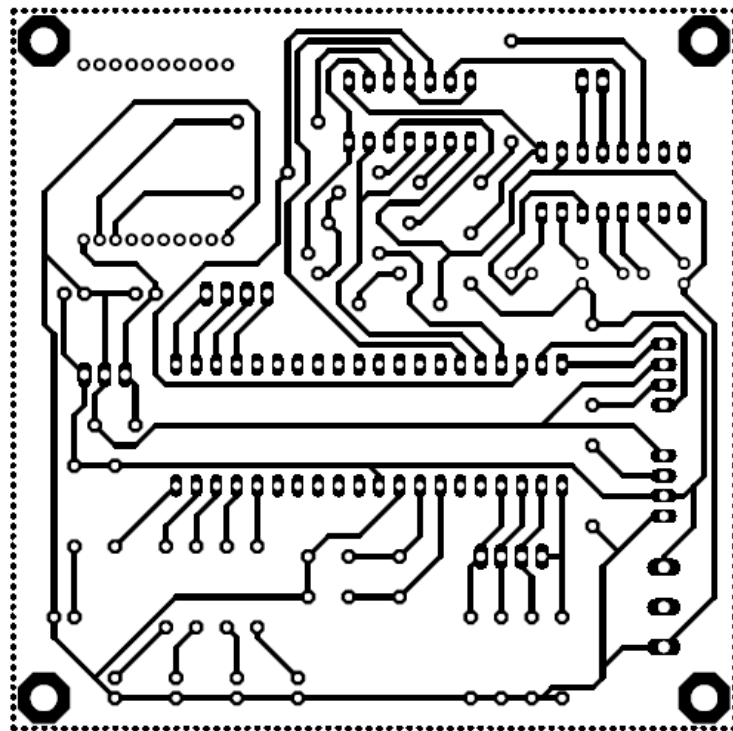


Figura 6.23.- Pistas circuito impreso.

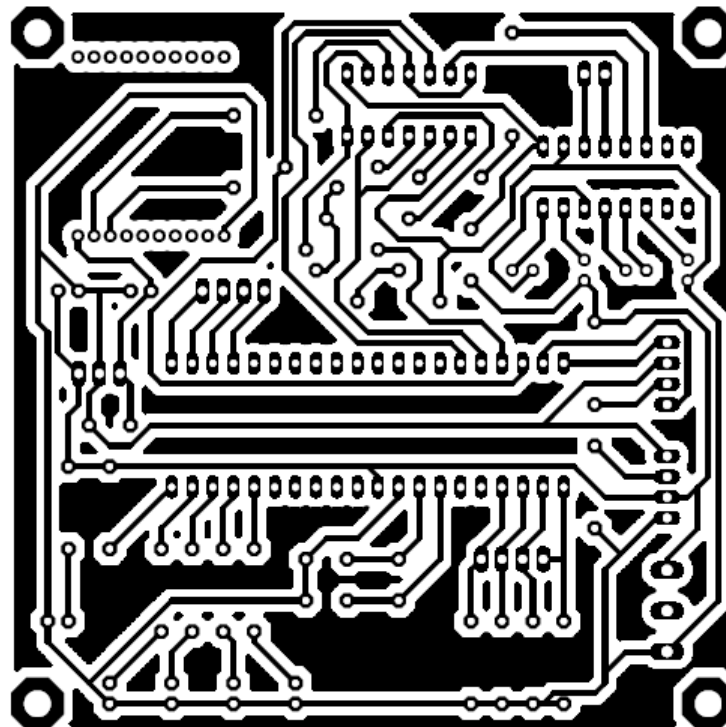


Figura 6.24.- Plano de tierra.

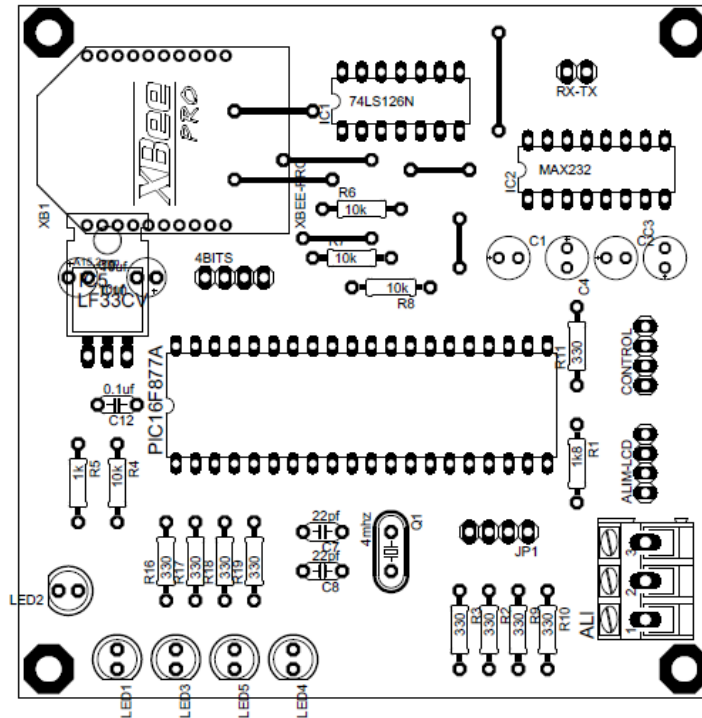


Figura 6.25.- Cara de componentes.

Una vez realizados los diseños finales procedemos a elaborar los PCB de los circuitos con la técnica de transferencia de calor de una impresión a laser en papel cuche sobre una placa de baquelita. Y después del decapado con el cloruro férrico obtenemos la cara de pistas (Figuras 6.26, 6.27, 6.28)

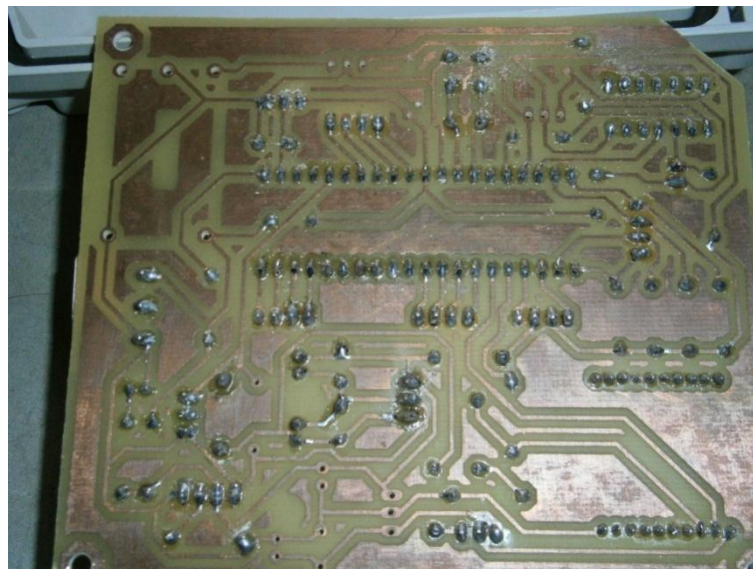


Figura 6.26.- Cara de pistas y soldado de componentes para los módulos de monitoreo.

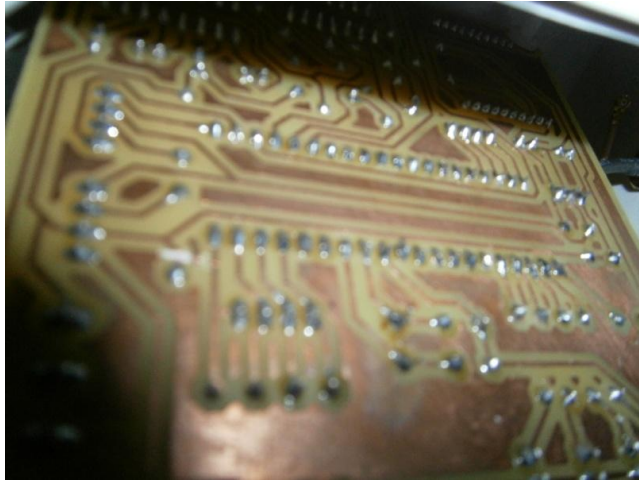


Figura 6.27.- Cara de pistas y soldado de componentes para los módulos de control.

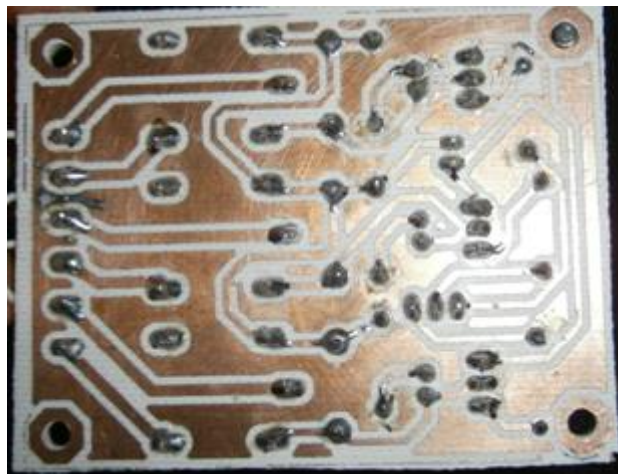


Figura 6.28.- Cara de pistas y soldado de componentes para el banco de relés

Y las respectivas caras de componentes soldados a las mismas (Figuras 6.29, 6.30, 6.31, 6.32, 6.33).

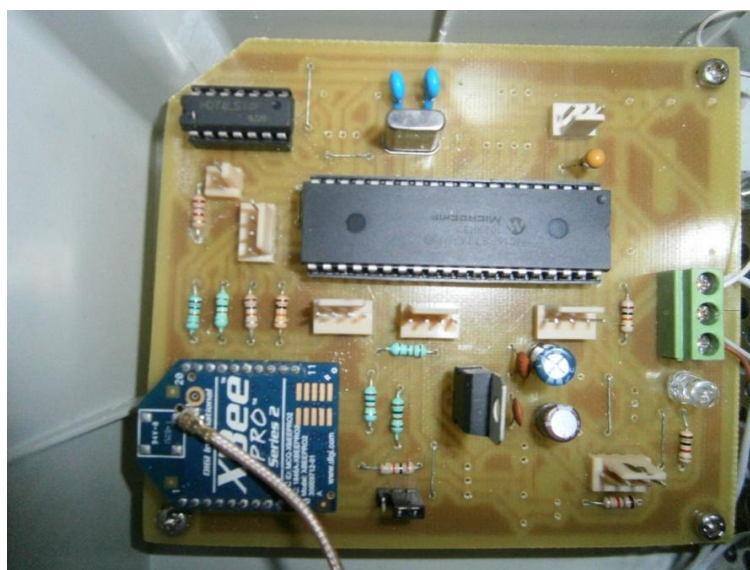


Figura 6.29.- Cara de componentes para el módulo de Humedad.



Figura 6.30.- Cara de componentes para el módulo de Temperatura.

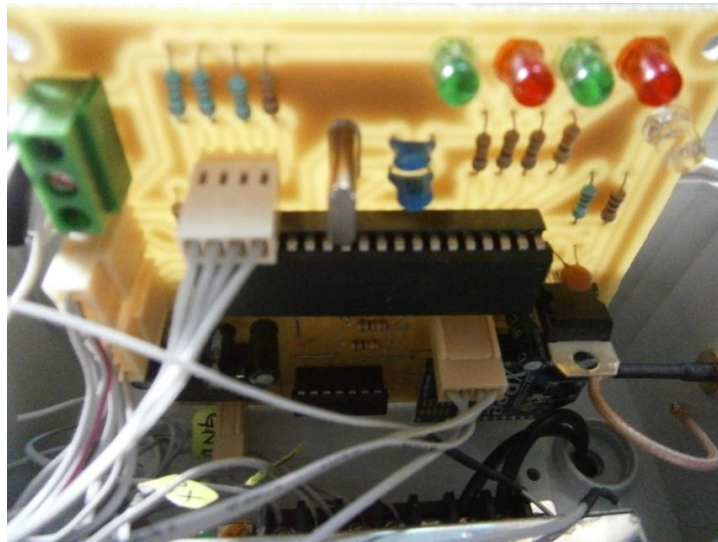


Figura 6.31.- Cara de componentes para el módulo de Control.

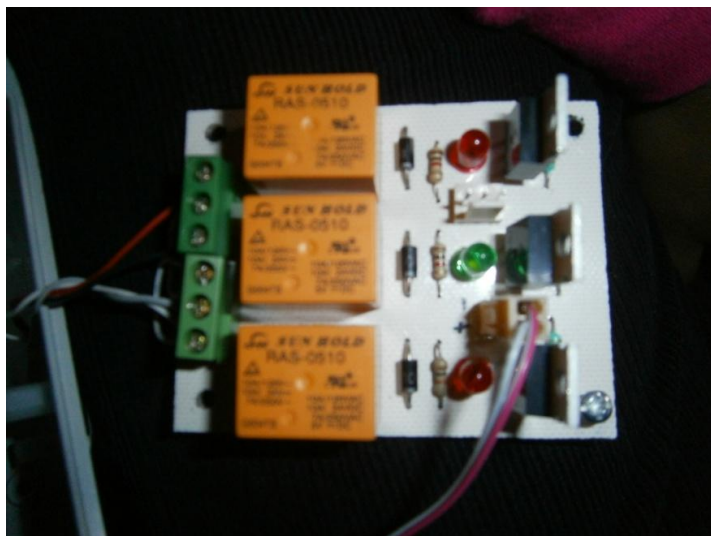


Figura 6.31.- Cara de componentes para el banco de relevadores.

Al finalizar el decapado y el soldado de los módulos se procede a integrar las tarjetas con los sensores y las respectivas fuentes de alimentación en una caja tipo nema para su instalación (Figuras 6.32, 6.33, 6.34, 6.35)

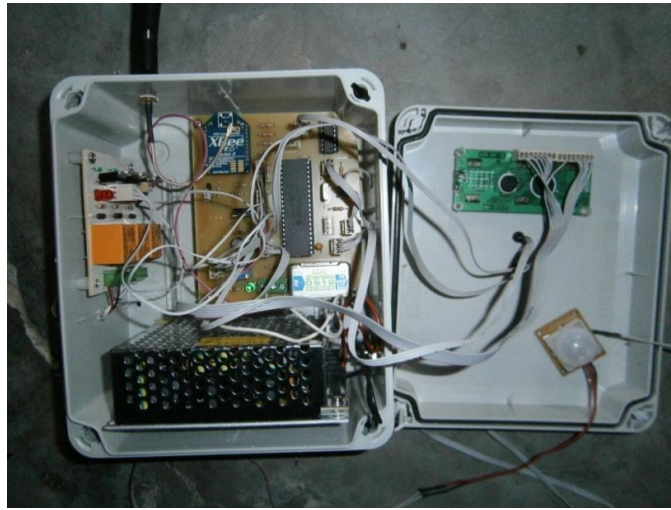


Figura 6.32.- Acoplamiento final del módulo completo de temperatura.



Figura 6.33.- Acoplamiento final del módulo de humedad.



Figura 6.34.- Acoplamiento final del módulo básico de temperatura.



Figura 6.35.- Acoplamiento final del módulo de control y módulo GSM.

Hasta este punto se a realizado la elaboración y de los módulos y se a verificado la funcionalidad de los mismos. La instalación de los módulos será con respecto a la ubicación descrita en el primer capítulo, dichos módulos deberán de automatizar varios elementos de uso manual tales como el encendido y apagado de la luz de un pasillo, el funcionamiento de un ventilador, el riego de un jardín y el estado de una alarma. Dichos módulos se colocaron de la siguiente manera:



Figura 6.36.- Módulo de control instalado.



Figura 6.37.- Módulo de temperatura instalado



Figura 6.38.- Módulo de humedad instalado

Gracias a que cada módulo posee una fotorresistencia, se a logrado eliminar un par de contactos que se utilizaban para el encendido y apagado de las luces de los pasillos, logrando automatizar este proceso (Figura 6.39).



Figura 6.39.- Automatización de luz de pasillo.

Por su parte el sensor de temperatura logra encender y apagar un ventilador dependiendo de la temperatura deseada por el usuario final (Figura 6.40).



Figura 6.40.- Automatización de un ventilador.

Con el control de humedad se logra automatizar el riego del jardín a través de una electroválvula (Figura 6.41) a 24v con conectores hembra de 3/4" para conectarla directo a la manguera. Esto permitirá controlar eléctricamente la salida de agua más fácilmente.



6.41.- Instalación de la electroválvula para automatizar riego.

Y finalmente se tiene una pequeña alarma (Figura 6.42) que es activada por la presencia humana (Figura 6.43) y por la apertura de las puertas.



Figura 6.42.- Alarma tipo sirena de 30 watt



Figura 6.43.- instalación del sensor PIR

CONCLUSIONES

El diseño y la implementación de este proyecto comprende de los siguientes módulos: modulo básico de temperatura, modulo básico de humedad, modulo completo y modulo de control, los cuales se componen de diferentes circuitos, que junto con sensores, nos ayudan a tener un control de variables, ya que se encuentran en constante comunicación a través de vía inalámbrica y vía GSM con ayuda de protocolos y comandos AT.

La integración de la vía inalámbrica, da como resultado la facilidad de colocar el proyecto sin grandes cambios en la arquitectura que se tenga, y a su vez da la facilidad de estar comunicado desde un solo punto. En el caso de la vía GSM, el usuario podrá comunicarse y controlar desde diferentes lugares, por medio de mensajes SMS, las variables de los distintos puntos.

El campo de aplicación de estos módulos se puede utilizar en la industria, así como también en aplicaciones de uso común. Su utilidad va desde realizar una tarea para comodidad del usuario, hasta la seguridad del mismo, por ello se realizo una investigación de las variables que se encuentran en diferentes puntos de la vivienda para determinar los elementos que se requiere para su aplicación.

Actualmente existen en el mercado una gran variedad de productos que ofrecen condiciones similares, pero con menores aplicaciones y/o costos elevados. El proyecto realizado, y a pesar de las aplicaciones que abarca, se realizo con un presupuesto bajo, en comparación con los productos existentes, lo cual permite una fácil adquisición de dicho producto.

Por ultimo, se puede afirmar que la realización del proyecto pone de manifiesto que es viable su colocación para beneficio del usuario, ya que el comportamiento de los elementos que lo conforman ha sido comprobado en la vivienda donde se coloco el proyecto, obteniendo resultados que afirman su eficiencia.

ANEXOS

ANEXO A: PROGRAMA FUENTE DEL MÓDULO BÁSICO DE TEMPERATURA.

```
.....
; CAJA CON NÚMERO: 221
; UBICACIÓN: "D1"
; PROTOCOLO: "Y"
.....

LIST P=16F877A
#include "P16F877A.INC"
__CONFIG 0x1939

CBLOCK 0X20
ENDC

CBLOCK 0X34
datotl
ENDC
CBLOCK 0X35
datoth
ENDC
CBLOCK 0X36
calor
ENDC
CBLOCK 0X37
dato
ENDC

CBLOCK 0X3A
preg3
ENDC
CBLOCK 0X3B
tipomodo
ENDC
CBLOCK 0X3D
centenas
decenas
unidades
ENDC

numero EQU d'50'
#define foto PORTC,3
#define reluz PORTD,5
#define ventilador PORTD,4
#define led PORTD,7

ORG 0

.....
; CONFIGURACIONES GENERALES
.....
configuraciones
call LCD_Inicializa
bsf STATUS,RP0 ; acceso al banco 1
bsf foto
bcf reluz
bcf ventilador
bcf led
```

```

        clrf    PORTB                ;PORTB salida
;~~~~~
; CONFIGURACIÓN DEL CONVERTIDOR ANALÓGICO
;~~~~~
        movlw  b'0000101'          ;A/D Port AN0/RA0
        movwf  ADCON1
        bsf    PORTA,0              ;Canal AN0 como entrada
        bcf    STATUS,RP0           ;Ir banco 0
        bcf    STATUS,RP1
        movlw  b'01000001'         ;A/D conversión Fosc/8
        movwf  ADCON0
        bsf    STATUS,RP0           ; acceso al banco 1
;~~~~~
; CONFIGURACIÓN DE LA USART PARA MODO RS_232; 115200,8,none,1,none
;~~~~~
        bcf    TRISC,6              ;RC6/TX salida, pin de transmisión
        bsf    TRISC,7              ;RC7/Rx entrada, pin de recepción
        movlw  d'1'                 ;115200 baud rate Xtal=3.9865Mhz
        movwf  SPBRG
        bsf    TXSTA,BRGH           ;Selección de alta velocidad
        bcf    TXSTA,SYNC           ;Modo asíncrono
        bcf    STATUS,RP0           ;Ir banco 0
        bsf    RCSTA,SPEN           ;habilita el puerto serie
        bsf    STATUS,RP0           ;Ir banco 1
        bcf    STATUS,RP1
        bcf    TXSTA,TX9            ;8 bits de datos a transmitir
        bsf    TXSTA,TXEN           ;Activa la transmisión serial, TXIF = 1
        bcf    STATUS,RP0           ;Ir banco 0
        bcf    RCSTA,RX9            ;8 Bits de datos
        bsf    RCSTA,CREN           ;Para Rx Continuo
;~~~~~
; AQUÍ ES UN MENÚ DE SENSORES HABILITADOS Y
; TOMA CONTINUA DE MUESTRAS ANALÓGICAS
;~~~~~
        bcf    reluz
        bcf    ventilador
        bcf    led
        movlw  d'24'
        movwf  datoth
        movlw  d'22'
        movwf  datotl
        call   LCD_Borra

activaalarma
        goto   comunica

sigueaquí
        goto   modonormal

modoalarma
        bcf    led
        call   adquisicion
        call   ldr
        call   checaven
        goto   muestra

modonormal

```

```

call    adquisicion
call    ldr
call    checaven
goto    muestra

```

```

; INICIA EL CONVERTIDOR ANALÓGICO AN0= TEMPERATURA

```

```

adquisicion

```

```

    clrf    calor
    call   T_20micros
    bsf    ADCON0,GO           ;inicia A/D conversión

```

```

momento

```

```

    btfsc  ADCON0,GO           ;ADCON0 es 0? (la conversión esta completa?)
    goto   momento            ;No, ir _espera
    movf   ADRESH,W           ;Si, W=ADRESH
    movwf  calor
    return

```

```

; COMPARA PARA ACTIVAR O DESACTIVAR UN VENTILADOR

```

```

checaven

```

```

    movf   datoth,W
    subwf  calor,W
    btfsc  STATUS,C
    bsf    ventilador
    movf   datotl,W
    subwf  calor,W
    btfss  STATUS,C
    bcf    ventilador
    return

```

```

; MODO NORMAL PARA LDR

```

```

ldr

```

```

    btfsc  foto
    bsf    reluz
    btfsc  foto
    movlw  'N'
    btfss  foto
    bcf    reluz
    btfss  foto
    movlw  'D'
    movwf  preg3
    return

```

```

; ESTA PARTE ES PARA VISUALIZAR DATOS FINALES EN EL LCD

```

```

muestra

```

```

    call   LCD_Lineal           ; Pasa a la segunda línea del LCD.
    call   LCD_DosEspaciosBlancos
    call   LCD_DosEspaciosBlancos
    movlw  'T'                   ; Visualiza carácter de temperatura
    call   LCD_Caracter
    movlw  '='                   ; Visualiza carácter
    call   LCD_Caracter
    call   LCD_UnEspacioBlanco
    movf   calor,W

```

```

call    convdec
call    LCD_Byte
movlw  '0'                ; Visualiza carácter °
call    LCD_Character
movlw  'C'                ; Visualiza carácter C
call    LCD_Character
call    LCD_Linea2
call    LCD_DosEspaciosBlancos
call    LCD_DosEspaciosBlancos
call    LCD_DosEspaciosBlancos ; Pasa a la segunda línea del LCD.
movlw  'F'                ; Visualiza carácter =
call    LCD_Character
movlw  '='                ; Visualiza carácter
call    LCD_Character
call    LCD_UnEspacioBlanco
movf   preg3,W            ; visualiza si es ? o A
call    LCD_Character
call    T_20ms
goto   activaalarma
.....
; PARTE DEL PROGRAMA PARA LA TRANSMISIÓN VÍA SERIE DE
; CUALQUIER DATO
.....
envia
    movwf TXREG            ;Transmitir dato
esperatx
    btfss PIR1,TXIF        ;Espera hasta que transmisión culminó
    goto  esperatx        ;No, ir _esperatx
    call  T_2ms
    return
.....
; PARTE DEL PROGRAMA PARA LA COMUNICACIÓN MAIK/ESCLAVO PARA
; OBTENER TEMPERATURA DESEADA Y MODO NORMAL O ALARMA
.....
comunica
    btfss PIR1,RCIF        ;Pregunta si el buffer de RX es full
    goto  esperemos        ; repetimos 50 veces y después abortamos
    movf  RCREG,W          ;Si, W=RCREG y pone a cero el RCIF
    sublw d'221'          ; si es igual al # de la caja
    btfss STATUS,Z
    goto  sigueaqui        ; no es la caja, así que sigue en funcionando
    call  T_20ms
    movlw 'Y'              ; responde que si es la caja y espera a recibir
    call  envia
recive
    btfss PIR1,RCIF        ;Pregunta si el buffer de RX es full
    goto  recive           ; repetimos 50 veces y después abortamos
    movf  RCREG,W          ;Si, W=RCREG y pone a cero el RCIF
    movwf tipomodo
    movf  tipomodo,W
    sublw d'73'            ; es igual a I ?
    btfsc STATUS,Z
    goto  limites         ; si toma los rangos de las temperaturas
    movf  tipomodo,W
    sublw d'84'            ; es igual a T ?
    btfsc STATUS,Z
    goto  enviamet        ; si envia la temperatura
    goto  sigueaqui
.....

```

.....
; PARTE DEL PROGRAMA PARA RECIBIR LA TEMPERATURA MÁXIMA Y MÍNIMA
.....

```

limites
    call    T_20ms
    movlw  'h'
    call    envia
reciveth
    btfss  PIR1,RCIF          ;Pregunta si el buffer de RX es full
    goto   reciveth          ;No, ir _espera
    movf   RCREG,W           ;Si, W=RCREG y pone a cero el RCIF
    movwf  datoth
    movlw  'l'
    call    envia
recivetl
    btfss  PIR1,RCIF          ;Pregunta si el buffer de RX es full
    goto   recivetl          ;No, ir _espera
    movf   RCREG,W           ;Si, W=RCREG y pone a cero el RCIF
    movwf  datotl
    goto   sigueaqui

```

.....
; PARTE PARA ENVIAR LOS DATOS DE LA TEMPERATURA VÍA SERIAL
.....

```

enviamet
    movf   calor,W
    call   envia
    goto   sigueaqui

```

.....
; PARTE PARA ESPERAR SI SE A RECIBIDO ALGÚN DATO DE LA CAJA DE
; CONTROL
.....

```

esperemos
    call    T_20ms
    incf   dato,F
    movf   dato,W
    sublw  numero
    btfss  STATUS,C
    goto   sigueaqui          ; abortamos ya que no hay dato a recibir
    goto   comunica           ; todavía vuelve a recibir

```

.....
; PARTE DEL PROGRAMA PARA CONVERTIR EL NUMERO EN DECIMAL
.....

```

convdec
    clrf   centenas          ; Carga los registros con el resultado inicial.
    clrf   decenas           ; En principio las centenas y decenas a cero.
    movwf  unidades         ; Se carga el número binario a convertir.
numuno
    movlw  .10               ; A las unidades se les va restando 10 en cada
    subwf  unidades,W        ; pasada. (W)=(BCD_Unidades) -10.
    btfss  STATUS,C         ; ¿C = 1?, ¿(W) positivo?, ¿(BCD_Unidades)>=10?
    goto   seacabo          ; No, es menor de 10. Se acabó.
numdos
    movwf  unidades         ; Recupera lo que queda por restar.
    incf   decenas,F        ; Incrementa las decenas y comprueba si ha llegado
    movlw  .10               ; a 10. Lo hace mediante una resta.
    subwf  decenas,W        ; (W)=(BCD_Decenas)-10).
    btfss  STATUS,C         ; ¿C = 1?, ¿(W) positivo?, ¿(BCD_Decenas)>=10?
    goto   numuno           ; No. Vuelve a dar otra pasada, restándole 10 a
numtres
    clrf   decenas          ; las unidades.
                                ; Pone a cero las decenas

```

```

    incf   centenas,F      ; e incrementa las centenas.
    goto   numuno         ; Otra pasada: Resta 10 al número a convertir.
seacabo
    swapf  decenas,W      ; En el nibble alto de (W) también las decenas.
    addwf  unidades,W     ; En el nibble bajo de (W) las unidades.
    return

```

```

INCLUDE <TIEMPOS.INC>
INCLUDE <LCD.INC>

```

```

END

```

Para la descripción de los archivos TIEMPOS.INC y LCD.INC ver anexo E y F respectivamente.

ANEXO B: PROGRAMA FUENTE DEL MÓDULO DE HUMEDAD.

```
.....  
; CAJA CON NUMERO: 222  
; UBICACIÓN: "J"  
; PROTOCOLO: "J"  
.....
```

```
LIST P=16F877A  
  #INCLUDE "P16F877A.INC"  
  __CONFIG 0x1939  
  
  CBLOCK 0X20  
  ENDC  
  
  CBLOCK 0X72  
  datotl  
  ENDC  
  CBLOCK 0X71  
  datoth  
  ENDC  
  CBLOCK 0X70  
  humedad  
  ENDC  
  CBLOCK 0X37  
  dato  
  ENDC  
  CBLOCK 0X38  
  Multiplicando  
  ENDC  
  CBLOCK 0X39  
  Contador  
  ENDC  
  CBLOCK 0X3A  
  preg3  
  ENDC  
  CBLOCK 0X3B  
  tipomodo  
  ENDC  
  CBLOCK 0X3D  
  hume_unidad  
  hume_decena  
  hume_centena  
  hume_millar  
  hume_decenamillar  
  ENDC  
  CBLOCK 0x50  
  resulH  
  resulL  
  cuentax  
  vecesxx  
  datoxx  
  sumaunidad  
  sumadecena  
  sumacentena  
  finalhume  
  divideH  
  divideL
```

```

divisor
cocienteH
cocienteL
sustra
unidades
decenas
centenas
ENDC

```

```

diez EQU d'10'
cien EQU d'100'
numero EQU d'50'
Multiplica EQU d'39'
#define foto PORTC,3
#define reluz PORTD,5
#define riego PORTD,4
#define led PORTD,7

```

```
ORG 0
```

```

.....
; CONFIGURACIONES GENERALES
.....

```

```
configuraciones
```

```

call LCD_Inicializa
bsf STATUS,RP0 ; acceso al banco 1
bsf foto
bcf reluz
bcf riego
bcf led
clrf PORTB ;PORTB salida

```

```

.....
; CONFIGURACIÓN DEL CONVERTIDOR ANALÓGICO
.....

```

```

movlw b'00000101' ;A/D Port AN0/RA0
movwf ADCON1
bsf PORTA,0 ;Canal AN0 como entrada
bcf STATUS,RP0 ;Ir banco 0
bcf STATUS,RP1
movlw b'01000001' ;A/D conversion Fosc/8
movwf ADCON0
bsf STATUS,RP0 ; acceso al banco 1

```

```

.....
; CONFIGURACIÓN DE LA USART PARA MODO RS_232; 115200,8,none,1,none
.....

```

```

bcf TRISC,6 ;RC6/TX salida, pin de transmisión
bsf TRISC,7 ;RC7/Rx entrada, pin de recepción
movlw d'1' ;115200 baud rate Xtal=3.9865Mhz
movwf SPBRG
bsf TXSTA,BRGH ;Selección de alta velocidad
bcf TXSTA,SYNC ;Modo asíncrono
bcf STATUS,RP0 ;Ir banco 0
bsf RCSTA,SPEN ;habilita el puerto serie
bsf STATUS,RP0 ;Ir banco 1
bcf STATUS,RP1
bcf TXSTA,TX9 ;8 bits de datos a transmitir
bsf TXSTA,TXEN ;Activa la transmisión serial, TXIF = 1
bcf STATUS,RP0 ;Ir banco 0
bcf RCSTA,RX9 ;8 Bits de datos

```

```

        bsf    RCSTA,CREN           ;Para Rx Continuo
;~~~~~
; AQUÍ ES UN MENÚ DE SENSORES HABILITADOS Y DE
; TOMA CONTINUA DE MUESTRAS ANALÓGICAS
;~~~~~
        bcf    reluz
        bcf    riego
        bcf    led
        movlw  d'72'
        movwf  datoth
        movlw  d'60'
        movwf  datotl
        call   LCD_Borra

activaalarma
        goto   comunica
sigueaquí
        call   adquisicion
        call   calculame
        call   ldr
        call   conversion
        call   finales
        call   checarie
        goto   muestra

;~~~~~
; INICIA EL CONVERTIDOR ANALÓGICO AN0= TEMPERATURA
;~~~~~
adquisicion
        clrf   humedad
        call   T_20micros
        bsf    ADCON0,GO           ;inicia A/D conversión

momento
        btfsc  ADCON0,GO           ;ADCON0 es 0? (la conversión esta completa?)
        goto   momento           ;No, ir _espera
        movf   ADRESH,W           ;Si, W=ADRESH
        movwf  humedad
        return

;~~~~~
;ESTA PARTE ES PARA CONVERTIR LOS 8 BITS A UNA ESCALA DE HUMEDAD (%) EN
; BINARIO
;~~~~~
calculame
        clrf   datoth
        clrf   datotl
        movf   humedad,Multiplicando
        movwf  datox
        movlw  Multiplica
        movwf  vecesxx
        call   producto8bit
        movf   resulH,W           ; Carga todos los registros y visualiza en hexadecimal.
        movwf  datoth
        movf   resulL,W
        movwf  datotl
        return

;~~~~~
;ESTA PARTE ES PARA CONVERTIR LOS 16 BITS A UN NÚMERO DECIMAL
;~~~~~

```

```

conversion
    clrf    hume_unidad
    clrf    hume_decena
    clrf    hume_centena
    clrf    hume_millar
    clrf    hume_decenamillar
    movf    datoH,W           ; Carga el número de entrada como primer dividendo
    movwf   divideH          ; parcial para realizar las sucesivas divisiones
    movf    datoL,W           ; entre 10.
    movwf   divideL
    movlw   .5                ; El número decimal tendrá 5 dígitos.
    movwf   Contador
    movlw   hume_unidad       ; Se va a utilizar direccionamiento indirecto.
    movwf   FSR               ; Apunta al primer dígito BCD, las unidades.

DigitoBCD2
    movlw   .10               ; Carga 10 en el divisor.
    movwf   divisor
    call    divide16          ; Divide el cociente anterior entre 10.
    movwf   INDF              ; El resto producido en la división es el dígito BCD.
    incf    FSR,F             ; Apunta a siguiente posición de registros de salida.
    movf    cocienteH,W       ; El cociente anterior será el dividendo de la
    movwf   divideH           ; próxima división de 16 bits.
    movf    cocienteL,W
    movwf   divideL
    decfsz  Contador,F
    goto    DigitoBCD2
    return

divide16
    clrf    cocienteH         ; En principio el resultado es cero.
    clrf    cocienteL
    clrf    sustra
    movlw   .16               ; Carga el contador.
    movwf   cuentax

DivideBit
    rlf     divideL,F
    rlf     divideH,F
    rlf     sustra,F          ; (Arit_Resto) contiene el dividendo parcial.
    movf    divisor,W
    subwf   sustra,W          ; Compara dividendo parcial y divisor.
    btfsc  STATUS,C          ; Si (dividendo parcial)>(divisor)
    movwf   sustra           ; (dividendo parcial) - (divisor) --> (dividendo parcial)
    rlf     cocienteL,F      ; Desplaza el cociente introduciendo el bit apropiado
    rlf     cocienteH,F      ; "0" ó "1" según corresponda.
    decfsz  cuentax,F
    goto    DivideBit
    movf    sustra,W         ; El resto también en (W).
    return

;*****
; MULTIPLICACIÓN DE (UNIDADES)(1), (DECENAS)(10), (CENTENAS)(100)
; SUMA DE TODOS LOS RESULTADOS PARA TENER EL VALOR DE LA HUMEDAD EN UN
; REGISTRO DE 8 BITS
;*****
finales
; PARA TENER UNIDADES
    movf    hume_centena,W
    movwf   sumaunidad
; PARA TENER DECENAS

```

```

    clrf    sumadecena
    movf   hume_millar,Multiplicando
    movwf  datoxx
    movlw  diez
    movwf  vecesxx
    call   producto8bit
    movf   resulL,W
    movwf  sumadecena
; PARA TENER CENTENAS
    clrf   sumacentena
    movf   hume_decenamillar,Multiplicando
    movwf  datoxx
    movlw  cien
    movwf  vecesxx
    call   producto8bit
    movf   resulL,W
    movwf  sumacentena
; HACE LA SUMA DE LOS TRES DATOS
    clrf   finalhume
    movf   sumaunidad,W
    addwf  sumadecena,F
    movf   sumadecena,W
    addwf  sumacentena,F
    movf   sumacentena,W
    movwf  finalhume
    return

```

producto8bit

```

    clrf   resulH           ; En principio el resultado es cero.
    clrf   resulL
    movlw  .8               ; Carga el contador de 8 bits.
    movwf  cuentax
    movf   datoxx,W        ; Carga el multiplicando.

```

multiplica8

```

    rrf    vecesxx,F        ; Va a comprobar el siguiente bit del multiplicador.
    btfsc  STATUS,C        ; Si el bit del multiplicador es "1"
    addwf  resulH,F        ; suma el resultado parcial anterior más el multiplicando.
    rrf    resulH,F        ; Desplaza el resultado parcial un lugar hacia la
    rrf    resulL,F        ; derecha, introduciendo un "0" por la izquierda.
    decfsz cuentax,F      ; ¿Ha multiplicado los 8 bits?
    goto   multiplica8    ; Repite la operación.
    return

```

```

.....
; COMPARA PARA ACTIVAR O DESACTIVAR EL RIEGO
.....

```

checarie

```

    movf   datoth,W
    subwf  finalhume,W
    btfsc  STATUS,C
    bcf    riego
    movf   datotl,W
    subwf  finalhume,W
    btfss  STATUS,C
    bsf    riego
    return

```

```

.....
; MODO NORMAL PARA LDR
.....

```

```

ldr
    btfsc    foto
    bsf      reluz
    btfsc    foto
    movlw   'N'
    btfss    foto
    bcf      reluz
    btfss    foto
    movlw   'D'
    movwf   preg3
    return

```

```

; ESTA PARTE ES PARA VISUALIZAR DATOS FINALES EN EL LCD

```

```

muestra
    call     LCD_Linea1           ; Pasa a la segunda línea del LCD.
    call     LCD_DosEspaciosBlancos
    call     LCD_DosEspaciosBlancos
    call     LCD_UnEspacioBlanco
    movlw   'H'                   ; Visualiza carácter de temperatura
    call     LCD_Caracter
    movlw   '='                   ; Visualiza caracter
    call     LCD_Caracter
    call     LCD_UnEspacioBlanco
    movf    finalhume,W
    call     convdec
    call     LCD_Byte
    movlw   '%'                   ; Visualiza caracter %
    call     LCD_Caracter
    call     LCD_Linea2
    call     LCD_DosEspaciosBlancos
    call     LCD_DosEspaciosBlancos
    call     LCD_DosEspaciosBlancos ; Pasa a la segunda línea del LCD.
    movlw   'F'                   ; Visualiza carácter F
    call     LCD_Caracter
    movlw   '='                   ; Visualiza carácter
    call     LCD_Caracter
    call     LCD_UnEspacioBlanco
    movf    preg3,W               ; visualiza si es ? o A
    call     LCD_Caracter
    call     T_20ms
    goto    activaalarma

```

```

; PARTE DEL PROGRAMA PARA LA TRANSMISIÓN VÍA SERIE DE
; CUALQUIER DATO

```

```

envia
    movwf   TXREG                 ;Transmitir dato
    esperatx
    btfss   PIR1,TXIF            ;Espera hasta que transmisión culminó
    goto    esperatx             ;No, ir _esperatx
    call    T_2ms
    return

```

```

; PARTE DEL PROGRAMA PARA LA COMUNICACIÓN MAIK/ESCLAVO PARA
; OBTENER TEMPERATURA DESEADA Y MODO NORMAL O ALARMA

```

```

comunica
    btfss   PIR1,RCIF            ;Pregunta si el buffer de RX es full

```

```

    goto    esperamos                ; repetimos 50 veces y después abortamos
    movf   RCREG,W                  ; Si, W=RCREG y pone a cero el RCIF
    sublw  d'222'                   ; si es igual al # de la caja
    btfss  STATUS,Z
    goto   sigueaqui                ; no es la caja, así que sigue en funcionando
    call   T_20ms
    movlw  'J'                       ; responde que si es la caja y espera a recibir
    call   envia

recive
    btfss  PIR1,RCIF                ;Pregunta si el buffer de RX es full
    goto   recive                   ; repetimos 50 veces y después abortamos
    movf   RCREG,W                  ; Si, W=RCREG y pone a cero el RCIF
    movwf  tipomodo
    movf   tipomodo,W
    sublw  d'73'                    ; es igual a I ?
    btfsc  STATUS,Z
    goto   limites                  ; si toma los rangos de las temperaturas
    movf   tipomodo,W
    sublw  d'72'                    ; es igual a H ?
    btfsc  STATUS,Z
    goto   enviament                ; si envia la temperatura
    goto   sigueaqui

```

```

.....
; PARTE DEL PROGRAMA PARA RECIBIR LA HUMEDAD MÁXIMA Y MÍNIMA
.....

```

```

limites
    call   T_20ms
    movlw  'h'
    call   envia

reciveth
    btfss  PIR1,RCIF                ;Pregunta si el buffer de RX es full
    goto   reciveth                ;No, ir _espera
    movf   RCREG,W                  ; Si, W=RCREG y pone a cero el RCIF
    movwf  datoht
    movlw  'I'
    call   envia

recivetl
    btfss  PIR1,RCIF                ;Pregunta si el buffer de RX es full
    goto   recivetl                ;No, ir _espera
    movf   RCREG,W                  ; Si, W=RCREG y pone a cero el RCIF
    movwf  datotl
    goto   sigueaqui

```

```

.....
; PARTE PARA ENVIAR LOS DATOS DE LA HUMEDAD VÍA SERIAL
.....

```

```

enviament
    movf   humedad,W
    call   envia
    movf   datoht,W
    call   envia
    movf   datotl,W
    call   envia
    goto   sigueaqui

```

```

.....
; PARTE PARA ESPERAR SI SE A RECIBIDO ALGÚN DATO DE LA CAJA DE
; CONTROL
.....

```

```

esperemos
    call   T_20ms

```

```

incf    dato,F
movf    dato,W
sublw   numero
btfss   STATUS,C
goto    sigueaqui           ; abortamos ya que no hay dato a recibir
goto    comunica            ; todavía vuelve a recibir

```

```

; PARTI DEL PROGRAMA PARA CONVERTIR EL NUMERO EN DECIMAL

```

```

convdec

```

```

clrf    centenas           ; Carga los registros con el resultado inicial.
clrf    decenas            ; En principio las centenas y decenas a cero.
movwf   unidades          ; Se carga el número binario a convertir.

```

```

numuno

```

```

movlw   .10                ; A las unidades se les va restando 10 en cada
subwf   unidades,W         ; pasada. (W)=(BCD_Unidades) -10.
btfss   STATUS,C           ; ¿C = 1?, ¿(W) positivo?, ¿(BCD_Unidades)>=10?
goto    seacabo            ; No, es menor de 10. Se acabó.

```

```

numdos

```

```

movwf   unidades          ; Recupera lo que queda por restar.
incf    decenas,F         ; Incrementa las decenas y comprueba si ha llegado
movlw   .10                ; a 10. Lo hace mediante una resta.
subwf   decenas,W         ; (W)=(BCD_Decenas)-10).
btfss   STATUS,C           ; ¿C = 1?, ¿(W) positivo?, ¿(BCD_Decenas)>=10?
goto    numuno            ; No. Vuelve a dar otra pasada, restándole 10 a

```

```

numtres

```

```

clrf    decenas           ; Ponc a cero las decenas
incf    centenas,F        ; e incrementa las centenas.
goto    numuno            ; Otra pasada: Resta 10 al número a convertir.

```

```

seacabo

```

```

swapf   decenas,W         ; En el nibble alto de (W) también las decenas.
addwf   unidades,W        ; En el nibble bajo de (W) las unidades.
return

```

```

INCLUDE <TIEMPOS.INC>

```

```

INCLUDE <LCD.INC>

```

```

END

```

Para la descripción de los archivos TIEMPOS.INC y LCD.INC ver anexo E y F respectivamente.

ANEXO C: PROGRAMA FUENTE DEL MÓDULO COMPLETO.

```
.....  
; CAJA CON NUMERO: 220  
; UBICACIÓN: "S"  
; PROTOCOLO: "S"  
.....
```

```
LIST P=16F877A  
#INCLUDE "P16F877A.INC"  
__CONFIG 0x1939
```

```
CBLOCK 0X20  
ENDC
```

```
CBLOCK 0X33  
siono  
ENDC
```

```
CBLOCK 0X34  
datotl  
ENDC
```

```
CBLOCK 0X35  
datoth  
ENDC
```

```
CBLOCK 0X36  
calor  
ENDC
```

```
CBLOCK 0X37  
dato  
ENDC
```

```
CBLOCK 0X38  
preg1  
ENDC
```

```
CBLOCK 0X39  
preg2  
ENDC
```

```
CBLOCK 0X3A  
preg3  
ENDC
```

```
CBLOCK 0X3B  
tipomodo  
ENDC
```

```
CBLOCK 0X3C  
modus  
ENDC
```

```
CBLOCK 0X3D  
centenas  
decenas  
unidades  
ENDC
```

```
numero EQU d'50'  
#define hall PORTC,2  
#define presencia PORTC,1  
#define foto PORTC,3  
#define reluz PORTD,5  
#define ventilador PORTD,4  
#define sirena PORTD,6  
#DEFINE led PORTD,7
```

ORG 0

.....
; CONFIGURACIONES GENERALES
.....

configuraciones

```
call    LCD_Inicializa
bsf     STATUS,RP0           ; acceso al banco 1
bsf     hall
bsf     presencia
bsf     foto
bcf     reluz
bcf     ventilador
bcf     led
bcf     sirena
clrf    PORTB                ;PORTB salida
```

.....
; CONFIGURACIÓN DEL CONVERTIDOR ANALÓGICO
.....

```
movlw  b'0000101'           ;A/D Port AN0/RA0
movwf   ADCON1
bsf     PORTA,0              ;Canal AN0 como entrada
bcf     STATUS,RP0          ;Ir banco 0
bcf     STATUS,RP1
movlw  b'01000001'          ;A/D conversion Fosc/8
movwf   ADCON0
bsf     STATUS,RP0          ; acceso al banco 1
```

.....
; CONFIGURACIÓN DE LA USART PARA MODO RS_232; 115200,8,none,1,none
.....

```
bcf     TRISC,6              ;RC6/TX salida, pin de transmisión
bsf     TRISC,7              ;RC7/Rx entrada, pin de recepción
movlw  d'1'                  ;115200 baud rate Xtal=3.9865Mhz
movwf   SPBRG
bsf     TXSTA,BRGH           ;Selección de alta velocidad
bcf     TXSTA,SYNC           ;Modo asíncrono
bcf     STATUS,RP0          ;Ir banco 0
bsf     RCSTA,SPEN           ;habilita el puerto serie
bsf     STATUS,RP0          ;Ir banco 1
bcf     STATUS,RP1
bcf     TXSTA,TX9            ;8 bits de datos a transmitir
bsf     TXSTA,TXEN           ;Activa la transmisión serial, TXIF = 1
bcf     STATUS,RP0          ;Ir banco 0
bcf     RCSTA,RX9            ;8 Bits de datos
bsf     RCSTA,CREN           ;Para Rx Continuo
```

.....
; AQUÍ ES UN MENÚ DE SENSORES HABILITADOS Y DE
; TOMA CONTINUA DE MUESTRAS ANALÓGICAS
.....

```
bcf     reluz
bcf     ventilador
bcf     sirena
bcf     led
clrf    modus
clrf    siono
movlw  d'24'
movwf   datoth
movlw  d'22'
```

```

movwf datotl
call LCD_Borra

activaalarma
goto comunica
sigueaquí
movf modus,W ; compara para saber que tipo de modo es
sublw d'0' ; 1=alarma 0=normal
btfsc STATUS,Z
goto modonormal
goto modoalarma

modoalarma
bcf led
call adquisicion
call actpir
call acthall
call ldr
call checaven
goto muestra

modonormal
bcf led
call adquisicion
call prehumana
call efectohall
call ldr
call checaven
goto muestra

; INICIA EL CONVERTIDOR ANALÓGICO AN0= TEMPERATURA
adquisicion
clrf calor
call T_20micros
bsf ADCON0,GO ;inicia A/D conversión
momento
btfsc ADCON0,GO ;ADCON0 es 0? (la conversión esta completa?)
goto momento ;No, ir _espera
movf ADRESH,W ;Si, W=ADRESH
movwf calor
return

; COMPARA PARA ACTIVAR O DESACTIVAR UN VENTILADOR
checaven
movf datoth,W
subwf calor,W
btfsc STATUS,C
bsf ventilador
movf datotl,W
subwf calor,W
btfss STATUS,C
bcf ventilador
return

```

```

; MODO NORMAL PARA PIR
prehumana
    btfsc    presencia
    movlw   'P'
    btfss    presencia
    movlw   'N'
    movwf   preg1
    return

; MODO NORMAL PARA EFECTO HALL
efectohall
    btfss    hall
    movlw   'A'
    btfsc    hall
    movlw   'C'
    movwf   preg2
    return

; MODO NORMAL PARA LDR
ldr
    btfsc    foto
    bsf      reluz
    btfsc    foto
    movlw   'N'
    btfss    foto
    bcf      reluz
    btfss    foto
    movlw   'D'
    movwf   preg3
    return

; MODO ALARMA.- EN CASO DE PRESENCIA SE REALIZA LO SIGUIENTE
actpir
    btfsc    presencia
    call     activa1
    return
activa1
    bsf      sirena
    movlw   'P'                ; muestra definitivamente P activación de alarma
    movwf   preg1
    movlw   d'80'              ; carga una P de la alarma porque hay intruso
    call     envia
    call     T_20ms
    movlw   d'83'              ; carga una S de la ubicación de la caja
    call     envia
    call     Retardo_1s
    bcf      sirena
    return

;MODO ALARMA.- EN CASO DE HALL SE REALIZA
acthall
    btfss    hall
    call     activa2
    return

```

```

activa2
    bsf    sirena
    movlw  'A'                ; muestra definitivamente P activación de alarma
    movwf  preg2
    movlw  d'72'              ; carga una H de la alarma porque hay intruso
    call   envia
    call   T_20ms
    movlw  d'83'              ; carga una S de la ubicación de la caja
    call   envia
    call   Retardo_1s
    bcf    sirena
    return

```

```

; ESTA PARTE ES PARA VISUALIZAR DATOS FINALES EN EL LCD

```

```

muestra
    call   LCD_Lineal          ; Pasa a la segunda línea del LCD.
    call   LCD_DosEspaciosBlancos
    movlw  'T'                ; Visualiza carácter de temperatura
    call   LCD_Character
    movlw  '='                 ; Visualiza caracter
    call   LCD_Character
    call   LCD_UnEspacioBlanco
    movf   calor,W
    call   convdec
    call   LCD_Byte
    movlw  "°"                 ; Visualiza caracter °
    call   LCD_Character
    movlw  'C'                 ; Visualiza caracter C
    call   LCD_Character
    call   LCD_Linea2         ; Pasa a la segunda línea del LCD.
    movlw  'P'                 ; Visualiza carácter de presencia
    call   LCD_Character
    movlw  '='                 ; Visualiza carácter
    call   LCD_Character
    call   LCD_UnEspacioBlanco
    movf   preg1,W
    call   LCD_Character
    call   LCD_DosEspaciosBlancos
    movlw  'E'
    call   LCD_Character
    movlw  '='                 ; Visualiza carácter
    call   LCD_Character
    call   LCD_UnEspacioBlanco
    movf   preg2,W            ; visualiza si es ? o A
    call   LCD_Character
    call   LCD_DosEspaciosBlancos
    movlw  'F'                 ; Visualiza carácter =
    call   LCD_Character
    movlw  '='                 ; Visualiza carácter
    call   LCD_Character
    call   LCD_UnEspacioBlanco
    movf   preg3,W            ; visualiza si es ? o A
    call   LCD_Character
    call   Retardo_20ms
;
return
goto    activaalarma

```

```

; PARTE DEL PROGRAMA PARA LA TRANSMISIÓN VÍA SERIE DE
; CUALQUIER DATO

```

```

.....
envia
    movwf TXREG                ;Transmitir dato
esperatx
    btfss PIR1,TXIF           ;Espera hasta que transmisión culminó
    goto  esperatx            ;No, ir _esperatx
    call  T_2ms
    return

```

```

.....
; PARTE DEL PROGRAMA PARA LA COMUNICACIÓN MAIK/ESCLAVO PARA
; OBTENER TEMPERATURA DESEADA Y MODO NORMAL O ALARMA
.....

```

```

comunica
    btfss PIR1,RCIF           ;Pregunta si el buffer de RX es full
    goto  esperemos           ;repetimos 50 veces y después abortamos
    movf  RCREG,W              ;Si, W=RCREG y pone a cero el RCIF
    sublw d'220'               ; si es igual al # de la caja
    btfss STATUS,Z
    goto  sigueaqui           ; no es la caja, así que sigue en funcionando
    call  T_20ms
    movlw 'S'                  ; responde que si es la caja y espera a recibir
    call  envia
recive
    btfss PIR1,RCIF           ;Pregunta si el buffer de RX es full
    goto  recive               ;repetimos 50 veces y después abortamos
    movf  RCREG,W              ;Si, W=RCREG y pone a cero el RCIF
    movwf tipomodo
    movf  tipomodo,W
    sublw d'73'                ; es igual a I ?
    btfsc STATUS,Z
    goto  limites             ; si toma los rangos de las temperaturas
    movf  tipomodo,W
    sublw d'84'                ; es igual a T ?
    btfsc STATUS,Z
    goto  enviamet            ; si envia la temperatura
    movf  tipomodo,W
    sublw d'65'                ; es igual a A ?
    btfsc STATUS,Z
    goto  elige                ; si recibe el tipo de modo de operación
    movf  tipomodo,W
    goto  sigueaqui

```

```

.....
; PARTE DEL PROGRAMA PARA RECIBIR LA TEMPERATURA MÁXIMA Y MÍNIMA
.....

```

```

limites
    call  T_20ms
    movlw 'h'
    call  envia
reciveth
    btfss PIR1,RCIF           ;Pregunta si el buffer de RX es full
    goto  reciveth            ;No, ir _espera
    movf  RCREG,W              ;Si, W=RCREG y pone a cero el RCIF
    movwf datoth
    movlw 'l'
    call  envia
recivetl
    btfss PIR1,RCIF           ;Pregunta si el buffer de RX es full
    goto  recivetl            ;No, ir _espera
    movf  RCREG,W              ;Si, W=RCREG y pone a cero el RCIF
    movwf datotl

```

```

        goto    sigueaqui
; ~~~~~
; PARTE PARA ENVIAR LOS DATOS DE LA TEMPERATURA VÍA SERIAL
; ~~~~~
enviamet
        movf   calor,W
        call   envia
        movf   datoht,W
        call   envia
        movf   datotl,W
        call   envia
        goto   sigueaqui
; ~~~~~
; PARTE PARA DEFINIR EL TIPO DE MODO EN QUE TRABAJA LA CAJA
; ~~~~~
elige
        movlw  'A'
        call   envia
elegido
        btfs   PIR1,RCIF           ;Pregunta si el buffer de RX es full
        goto   elegido             ; repetimos 50 veces y después abortamos
        movf   RCREG,W             ;Si, W=RCREG y pone a cero el RCIF
        movwf  modus
        goto   sigueaqui
; ~~~~~
; PARTE PARA ESPERAR SI SE A RECIBIDO ALGÚN DATO DE LA CAJA DE
; CONTROL
; ~~~~~
esperemos
        call   T_20ms
        incf   dato,F
        movf   dato,W
        sublw  numero
        btfs   STATUS,C
        goto   sigueaqui           ; abortamos ya que no hay dato a recibir
        goto   comunica            ; todavía vuelve a recibir
; ~~~~~
; PARTE DEL PROGRAMA PARA CONVERTIR EL NUMERO EN DECIMAL
; ~~~~~
convdec
        clrf   centenas            ; Carga los registros con el resultado inicial.
        clrf   decenas             ; En principio las centenas y decenas a cero.
        movwf  unidades            ; Se carga el número binario a convertir.
numuno
        movlw  .10                 ; A las unidades se les va restando 10 en cada
        subwf  unidades,W          ; pasada. (W)=(BCD_Unidades) -10.
        btfs   STATUS,C            ; ¿C = 1?, ¿(W) positivo?, ¿(BCD_Unidades)>=10?
        goto   seacabo             ; No, es menor de 10. Se acabó.
numdos
        movwf  unidades            ; Recupera lo que queda por restar.
        incf   decenas,F           ; Incrementa las decenas y comprueba si ha llegado
        movlw  .10                 ; a 10. Lo hace mediante una resta.
        subwf  decenas,W           ; (W)=(BCD_Decenas)-10).
        btfs   STATUS,C            ; ¿C = 1?, ¿(W) positivo?, ¿(BCD_Decenas)>=10?
        goto   numuno             ; No. Vuelve a dar otra pasada, restándole 10 a
numtres
        clrf   decenas             ; las unidades.
        incf   centenas,F          ; Pone a cero las decenas
        goto   numuno             ; e incrementa las centenas.
seacabo
        goto   numuno             ; Otra pasada: Resta 10 al número a convertir.

```

```
swapf decenas,W           ; En el nibble alto de (W) también las decenas.  
addwf unidades,W         ; En el nibble bajo de (W) las unidades.  
return
```

```
INCLUDE <TIEMPOS.INC>  
INCLUDE <LCD.INC>
```

```
END
```

Para la descripción de los archivos TIEMPOS.INC y LCD.INC ver anexo E y F respectivamente.

ANEXO D: PROGRAMA FUENTE DEL MÓDULO DE CONTROL.

```
.....  
: MÓDULO DE CONTROL  
.....
```

```
LIST P=16F877A  
#INCLUDE "P16F877A.INC"  
__CONFIG 0x1939
```

```
CBLOCK 0X20  
ENDC
```

```
CBLOCK 0X33  
tempex  
ENDC
```

```
CBLOCK 0X34  
tempei  
ENDC
```

```
CBLOCK 0X35  
temp1  
ENDC
```

```
CBLOCK 0X36  
temp2  
ENDC
```

```
CBLOCK 0X37  
temp3  
ENDC
```

```
CBLOCK 0X38  
temp4  
ENDC
```

```
CBLOCK 0X47  
temp5  
ENDC
```

```
CBLOCK 0X39  
hume1  
ENDC
```

```
CBLOCK 0X3A  
valorcaja  
ENDC
```

```
CBLOCK 0X3B  
valorcaja2  
ENDC
```

```
CBLOCK 0X3C  
dato  
ENDC
```

```
CBLOCK 0X3D  
alarma  
ENDC
```

```
CBLOCK 0X3E  
doble  
ENDC
```

```
CBLOCK 0X3F  
monitor  
ENDC
```

```
CBLOCK 0X40  
tipoala  
ENDC
```

```
CBLOCK 0X41  
lugarala
```

ENDC

CBLOCK 0X40

centenas

decenas

unidades

ENDC

numero EQU d'100' ; es para la máxima temperatura a sele, después inicia a cero
activam EQU d'1' ; es para seleccionar el modo alarma
repite EQU d'50' ; es el núm. de veces que espera para recibir

#define sele PORTC,2
#define vmax PORTC,3
#define vmin PORTD,0
#define soli PORTD,1
#define xbee PORTC,5
#define gsm PORTC,4
#define foco0 PORTA,0
#define foco1 PORTA,1
#define foco2 PORTA,2
#define foco3 PORTA,3

ORG 0

.....
; CONFIGURACIONES GENERALES
.....

configuraciones

call LCD_Inicializa
bsf STATUS,RP0 ; acceso al banco 1
bsf sele
bsf vmax
bsf vmin
bcf xbee
bcf gsm
movlw b'00000110'
movwf ADCON1
movlw b'00000000'
movwf TRISA

.....
; CONFIGURACIÓN DE LA USART PARA MODO RS_232; 115200,8,none,1,none
.....

bcf TRISC,6 ;RC6/TX salida, pin de transmisión
bsf TRISC,7 ;RC7/Rx entrada, pin de recepción
movlw d'1' ;115200 baud rate Xtal=3.9865Mhz
movwf SPBRG
bsf TXSTA,BRGH ;Selección de alta velocidad
bcf TXSTA,SYNC ;Modo asíncrono
bcf STATUS,RP0 ;Ir banco 0
bsf RCSTA,SPEN ;habilita el puerto serie
bsf STATUS,RP0 ;Ir banco 1
bcf STATUS,RP1
bcf TXSTA,TX9 ;8 bits de datos a transmitir
bsf TXSTA,TXEN ;Activa la transmisión serial, TXIF = 1
bcf STATUS,RP0 ;Ir banco 0
bcf RCSTA,RX9 ;8 Bits de datos
bsf RCSTA,CREN ;Para Rx Continuo

```

clrf    tempex
clrf    tempei
clrf    temp1
clrf    temp2
clrf    temp3
clrf    temp4
clrf    hume1
clrf    valorcaja
clrf    dato
clrf    alarma
clrf    doble
call    LCD_Borra
bcf     xbee
bcf     gsm
bcf     foco0
bcf     foco1
bcf     foco2
bcf     foco3

```

```

; MENU

```

```

call    confinic
comienzo
    btfsc    sele
    goto    masdatos
    btfsc    soli
    goto    pidetem
    goto    modulogsm

```

```

; PARTE DEL PROGRAMA PARA EL ENVIÓ DE LAS TEMPERATURAS EN LAS CAJAS
; Y DE LA HUMEDAD

```

```

masdatos
    bsf     xbee
    btfsc    sele
    goto    masdatos
caja1
    movlw   'S'
    movwf   valorcaja
    movlw   ''
    movwf   valorcaja2
    call    muestradatos
    btfsc    sele
    goto    manda1
    btfsc    vmax
    call    contmax
    btfsc    vmin
    call    contmin
    goto    caja1

```

```

manda1
    movlw   d'220'
    call    envia
    call    recibe
    sublw   'S' ; si es igual al # de la caja
    btfss   STATUS,Z
    goto    manda1
    call    terminatx

```

```

masdatos2
    btfsc    sele
    goto    masdatos2
caja2
    movlw   'D'
    movwf   valorcaja
    movlw   '1'
    movwf   valorcaja2
    call    muestradatos
    btfsc   sele
    goto    manda2
    btfsc   vmax
    call    contmax
    btfsc   vmin
    call    contmin
    goto    caja2
manda2
    movlw   d'221'
    call    envia
    call    recibe
    sublw   'Y'
    btfss   STATUS,Z
    goto    manda2
    call    terminatx
masdatos3
    btfsc    sele
    goto    masdatos3
caja4
    movlw   'D'
    movwf   valorcaja
    movlw   '2'
    movwf   valorcaja2
    call    muestradatos
    btfsc   sele
    goto    manda4
    btfsc   vmax
    call    contmax
    btfsc   vmin
    call    contmin
    goto    caja4
manda4
    movlw   d'223'
    call    envia
    call    recibe
    sublw   'W'
    btfss   STATUS,Z
    goto    manda4
    call    terminatx
masdatos5
    btfsc    sele
    goto    masdatos5
caja5
    movlw   'C'
    movwf   valorcaja
    movlw   ''
    movwf   valorcaja2
    call    muestradatos
    btfsc   sele
    goto    manda1

```

; si es igual al # de la caja

; si es igual al # de la caja

```

        btfsc    vmax
        call     contmax
        btfsc    vmin
        call     contmin
        goto     caja5
manda5
        movlw   d'224'
        call     envia
        call     recibe
        sublw   'C'                ; si es igual al # de la caja
        btfss   STATUS,Z
        goto     manda5
        call     terminatx
masdatos6
        btfsc    sele
        goto     masdatos6
caja3
        movlw   'J'
        movwf   valorcaja
        movlw   ''
        movwf   valorcaja2
        call    muestradatos
        btfsc    sele
        goto     manda3
        btfsc    vmax
        call     contmax
        btfsc    vmin
        call     contmin
        goto     caja3
manda3
        movlw   d'222'
        call     envia
        call     recibe
        sublw   'D'                ; si es igual al # de la caja
        btfss   STATUS,Z
        goto     manda3
        call     terminatx
        bcf     xbee
        goto     comienzo

contmax
        call    Retardo_200ms
        incf   tempex,F
        movf   tempex,W
        sublw  numero
        btfss  STATUS,C
        clrf   tempex
        return

contmin
        call    Retardo_200ms
        incf   tempei,F
        movf   tempei,W
        sublw  numero
        btfss  STATUS,C
        clrf   tempei
        return

terminatx
        call    Retardo_20ms
        movlw   'T'
        call     envia

```

```

call    recibe
sublw   'h'                ; si es igual al # de la caja
btfss   STATUS,Z
nop
movlw   tempex
call    envia
call    recibe
sublw   'l'                ; si es igual al # de la caja
btfss   STATUS,Z
nop
movlw   tempei
call    envia
return

```

```

.....
; PARTE DEL PROGRAMA PARA SELECCIONAR LAS TEMPERATURAS Y LA HUMEDAD DE
; TODAS LAS CAJAS EXISTENTES
.....

```

```

pidetem
    btfsc   soli
    goto    pidetem
    bsf     xbee

solicita1
    movlw   d'220'
    call    envia
    call    recibe
    sublw   'S'            ; si es igual al # de la caja
    btfss   STATUS,Z
    goto    solicita1
    movlw   'T'
    call    envia
    call    recibe
    movwf   temp1

solicita2
    movlw   d'221'
    call    envia
    call    recibe
    sublw   'Y'            ; si es igual al # de la caja
    btfss   STATUS,Z
    goto    solicita2
    movlw   'T'
    call    envia
    call    recibe
    movwf   temp2

solicita4
    movlw   d'223'
    call    envia
    call    recibe
    sublw   'W'            ; si es igual al # de la caja
    btfss   STATUS,Z
    goto    solicita4
    movlw   'T'
    call    envia
    call    recibe
    movwf   temp4

solicita5
    movlw   d'224'
    call    envia
    call    recibe
    sublw   'C'            ; si es igual al # de la caja

```

```

        btfss STATUS,Z
        goto solicita5
        movlw 'T'
        call envia
        call recibe
        movwf temp5
solicita3
        movlw d'222'
        call envia
        call recibe
        sublw 'D' ; si es igual al # de la caja
        btfss STATUS,Z
        goto solicita3
        movlw 'T'
        call envia
        call recibe
        movwf temp3
        call muestrasol
        bcf xbee
        goto comienzo

```

```

.....
; PARTE DEL PROGRAMA PARA ESPERAR RESPUESTA DEL MODULO GSM, ES PARA
; ACTIVAR O DESACTIVAR ALARMAS
.....

```

```

modulogsm
        bsf      foco3
        bsf      gsm
recivel
        btfss   PIR1,RCIF ;Pregunta si el buffer de RX es full
;         goto   recivel ;No, ir _espera
        goto   esperemos
        movf   RCREG,W ;Si, W=RCREG y pone a cero el RCIF
        movwf  dato
        bcf    gsm
        bcf    foco3
        movf   dato,W
        sublw  d'82' ; si es igual al R
        btfsc  STATUS,Z
        goto   compara
        goto   alarsss

compara
        movf   alarma,W
        sublw  d'0' ; si es igual a cero
        btfsc  STATUS,Z
        goto   carga1
        clrf   alarma
        call   envialar
        bcf    foco2
        call   Retardo_20s
        goto   alarsss

carga1
        movf   activam,W
        movwf  alarma
        call   envialar
        bsf    foco2
        call   Retardo_20s
        goto   alarsss

```

```
esperemos
;   call   Retardo_20ms
   incf   doble,F
   movf   doble,W
   sublw  repite
   btfss  STATUS,C
   goto   alarsss
   goto   recibe1
```

```
.....
; ESTA PARTE ES PARA VISUALIZAR TEMPERATURAS Y HUMEDAD A ENVIAR EN EL LCD
.....
```

```
muestradatos
   call   LCD_Linea1           ; Pasa a la segunda línea del LCD.
   call   LCD_DosEspaciosBlancos
   call   LCD_UnEspacioBlanco
   movlw  'T'                 ; Visualiza carácter de temperatura
   call   LCD_Character
   movlw  'E'                 ; Visualiza carácter de temperatura
   call   LCD_Character
   movlw  'M'                 ; Visualiza carácter
   call   LCD_Character
   movlw  'P'                 ; Visualiza carácter de temperatura
   call   LCD_Character
   movlw  'E'                 ; Visualiza carácter de temperatura
   call   LCD_Character
   movlw  'R'                 ; Visualiza carácter
   call   LCD_Character
   movlw  'A'                 ; Visualiza carácter de temperatura
   call   LCD_Character
   movlw  'T'                 ; Visualiza carácter de temperatura
   call   LCD_Character
   movlw  'U'                 ; Visualiza carácter
   call   LCD_Character
   movlw  'R'                 ; Visualiza carácter de temperatura
   call   LCD_Character
   movlw  'A'                 ; Visualiza carácter de temperatura
   call   LCD_Character
   call   LCD_UnEspacioBlanco
   movf   valorcaja,W
   call   LCD_Character
   movf   valorcaja2,W
   call   LCD_Character
   call   LCD_Linea2
   movlw  'M'                 ; Visualiza carácter
   call   LCD_Character
   movlw  'A'                 ; Visualiza carácter de temperatura
   call   LCD_Character
   movlw  'X'                 ; Visualiza carácter de temperatura
   call   LCD_Character
   movlw  '='                 ; Visualiza carácter de temperatura
   call   LCD_Character
   movf   tempex,W
   call   convdec
   call   LCD_Byte
   movlw  '0'                 ; Visualiza carácter de temperatura
   call   LCD_Character
   movlw  'C'                 ; Visualiza carácter de temperatura
```



```

call    LCD_Character
call    LCD_DosEspaciosBlancos
movlw  'M'                ; Visualiza carácter
call    LCD_Character
movlw  'T'                ; Visualiza carácter de temperatura
call    LCD_Character
movlw  'N'                ; Visualiza carácter de temperatura
call    LCD_Character
movlw  '='                ; Visualiza carácter de temperatura
call    LCD_Character
movf   tempei,W
call   convdec
call   LCD_Byte
movlw  '0'                ; Visualiza carácter de temperatura
call   LCD_Character
movlw  'C'                ; Visualiza carácter de temperatura
call   LCD_Character
return

```

```

.....
;ESTA PARTE ES PARA VISUALIZAR TEMPERATURAS DE LAS DIFERENTES CAJAS EN EL
;LCD
.....

```

```

muestrasol
call    LCD_Linea1        ; Pasa a la segunda línea del LCD.
call    LCD_DosEspaciosBlancos
call    LCD_UnEspacioBlanco
movlw  'H'                ; Visualiza carácter de temperatura
call    LCD_Character
movlw  '1'                ; Visualiza carácter de temperatura
call    LCD_Character
movlw  '='                ; Visualiza carácter
call    LCD_Character
movf   hume1,W
call   convdec
call   LCD_Byte
call   LCD_DosEspaciosBlancos
movlw  'T'                ; Visualiza carácter de temperatura
call   LCD_Character
movlw  '1'                ; Visualiza carácter de temperatura
call   LCD_Character
movlw  '='                ; Visualiza carácter
call   LCD_Character
movf   temp1,W
call   convdec
call   LCD_Byte
call   LCD_Linea2        ; Pasa a la segunda línea del LCD.
movlw  'T'                ; Visualiza carácter de temperatura
call   LCD_Character
movlw  '2'                ; Visualiza carácter de temperatura
call   LCD_Character
movlw  '='                ; Visualiza carácter
call   LCD_Character
movf   temp2,W
call   convdec
call   LCD_Byte
call   LCD_UnEspacioBlanco
movlw  'T'                ; Visualiza carácter de temperatura
call   LCD_Character

```

```

movlw '3' ; Visualiza carácter de temperatura
call LCD_Caracter
movlw '=' ; Visualiza carácter
call LCD_Caracter
movf temp3,W
call convdec
call LCD_Byte
call LCD_UnEspacioBlanco
movlw 'T' ; Visualiza carácter de temperatura
call LCD_Caracter
movlw '4' ; Visualiza carácter de temperatura
call LCD_Caracter
movlw '=' ; Visualiza carácter
call LCD_Caracter
movf temp4,W
call convdec
call LCD_Byte
return

```

```

; PARTE DEL PROGRAMA PARA LA TRANSMISIÓN VÍA SERIE DE
; CUALQUIER DATO

```

envía

```

movwf TXREG ;Transmitir dato
esperatx
btfss PIR1,TXIF ;Espera hasta que transmisión culminó
goto esperatx ;No, ir _esperatx
call Retardo_2ms
return

```

recive

```

esperarx
btfss PIR1,RCIF ;Pregunta si el buffer de RX es full
goto esperarx
movf RCREG,W ;Si, W=RCREG y pone a cero el RCIF
return

```

```

; PARTE DEL PROGRAMA PARA CONVERTIR EL NÚMERO EN DECIMAL

```

convdec

```

clrf centenas ; Carga los registros con el resultado inicial.
clrf decenas ; En principio las centenas y decenas a cero.
movwf unidades ; Se carga el número binario a convertir.
numuno
movlw .10 ; A las unidades se les va restando 10 en cada
subwf unidades,W ; pasada. (W)=(BCD_Unidades) -10.
btfss STATUS,C ; ¿C = 1?, ¿(W) positivo?, ¿(BCD_Unidades)>=10?
goto seacabo ; No, es menor de 10. Se acabó.
numdos
movwf unidades ; Recupera lo que queda por restar.
incf decenas,F ; Incrementa las decenas y comprueba si ha llegado
movlw .10 ; a 10. Lo hace mediante una resta.
subwf decenas,W ; (W)=(BCD_Decenas)-10).
btfss STATUS,C ; ¿C = 1?, ¿(W) positivo?, ¿(BCD_Decenas)>=10?
goto numuno ; No. Vuelve a dar otra pasada, restándole 10 a
numtres ; las unidades.
clrf decenas ; Pone a cero las decenas
incf centenas,F ; e incrementa las centenas.

```

```

goto numuno ; Otra pasada: Resta 10 al número a convertir.
seacabo
swapf decenas,W ; En el nibble alto de (W) también las decenas.
addwf unidades,W ; En el nibble bajo de (W) las unidades.
return

```

```

; CONFIGURACIONES INICIALES PARA EL MÓDULO GSM

```

```

confinic
call Retardo_10s
call Retardo_5s
call Retardo_2s
call Retardo_2s
bsf foco3
bsf gsm
call Retardo_1s
movlw 'A'
call envia
movlw 'T'
call envia
movlw d'13'
call envia
call Retardo_20ms
movlw 'A'
call envia
movlw 'T'
call envia
movlw 'S'
call envia
movlw '0'
call envia
movlw '='
call envia
movlw '0'
call envia
movlw d'13'
call envia
bcf gsm
bcf foco3
call Retardo_1s
return

```

```

; PARTE DEL PROGRAMA PARA ENVIAR EL MODO ALARMA

```

```

envialar
movlw d'220'
call envia
call recibe
sublw 'S' ; si es igual al # de la caja
btfss STATUS,Z
goto envialar
call culminala
envialar2
movlw d'221'
call envia
call recibe
sublw 'Y' ; si es igual al # de la caja
btfss STATUS,Z

```

```

        goto    enviar2
        call    culminala
encialar3
        movlw  d'223'
        call    envia
        call    recibe
        subl   'W'                                ; si es igual al # de la caja
        btfss  STATUS,Z
        goto    enviar
        call    culminala
        return

```

```

culminala
        call    Retardo_20ms
        movlw  'A'
        call    envia
        movlw  alarma
        call    envia
        return

```

```

.....
; PARTE DEL PROGRAMA PARA ESTAR AL PENDIENTE DE LAS ALARMAS
.....

```

```

alarsss

```

```

        bsf    xbee
        call    recibe
        movwf  monitor
;        bcf    xbee

        movf   monitor,W
        subl   'P'                                ; es igual a P ?
        btfsc  STATUS,Z
        goto   espresen                            ; si manda alarma de presencia y checa el lugar
        movf   monitor,W
        subl   'H'                                ; es igual a H ?
        btfsc  STATUS,Z
        goto   eshume                               ; si manda alarma de hall y checa el lugar
        goto   comienzo

```

```

espresen
        movf   'P',W
        movwf  tipoala
        goto   checalugar

```

```

eshume
        movf   'T',W
        movwf  tipoala
        goto   checalugar

```

```

checalugar
        call    recibe
        movwf  monitor
        bcf    xbee
        movf   monitor,W
        subl   'S'                                ; es igual a S ?
        btfsc  STATUS,Z
        goto   cargas                              ; si recibe el tipo de modo de operación
        movf   monitor,W
        subl   'W'                                ; es igual a W ?
        btfsc  STATUS,Z
        goto   cargad                              ; si recibe el tipo de modo de operación

```

```

movf   monitor,W
sublw  'C'
btfsc  STATUS,Z
goto   cargac
goto   comienzo

```

; es igual a C ?

; si recibe el tipo de modo de operación

cargas

```

movf   'S',W
movwf  lugarala
goto   avisaala

```

cargad

```

movf   'W',W
movwf  lugarala
goto   avisaala

```

cargac

```

movf   'C',W
movwf  lugarala
goto   avisaala

```

avisaala

```

movlw  'a'
call   envia
movlw  't'
call   envia
movlw  '+'
call   envia
movlw  'c'
call   envia
movlw  'm'
call   envia
movlw  'g'
call   envia
movlw  's'
call   envia
movlw  '='
call   envia
movlw  ""
call   envia
movlw  '5'
call   envia
movlw  '5'
call   envia
movlw  '1'
call   envia
movlw  '3'
call   envia
movlw  '8'
call   envia
movlw  '0'
call   envia
movlw  '6'
call   envia
movlw  '9'
call   envia
movlw  '5'
call   envia
movlw  '8'
call   envia
movlw  ""
call   envia

```

```

movlw d'13'
call  envia
call  Retardo_5s
movlw 'E'
call  envia
movlw 'I'
call  envia
movlw ''
call  envia
movlw 'i'
call  envia
movlw 'n'
call  envia
movlw 't'
call  envia
movlw 'r'
call  envia
movlw 'u'
call  envia
movlw 's'
call  envia
movlw 'o'
call  envia
movlw ''
call  envia
movlw 'e'
call  envia
movlw 's'
call  envia
movlw 't'
call  envia
movlw 'a'
call  envia
movlw ''
call  envia
movlw 'e'
call  envia
movlw 'n'
call  envia
movlw ''
movlw 'e'
call  envia
movlw 'I'
call  envia
movlw ''
call  envia
movlw tipoala
call  envia
movlw ''
call  envia
movlw 'd'
call  envia
movlw 'e'
call  envia
movlw ''
call  envia
movlw 'I'
call  envia
movlw 'a'
movlw ''

```

```
call    envia
movlw   lugarala
call    envia
movlw   d'26'
call    envia
goto    comienzo
```

```
INCLUDE <TIEMPOS.INC>
INCLUDE <LCD.INC>
```

```
END
```

Para la descripción de los archivos TIEMPOS.INC y LCD.INC ver anexo E y F respectivamente.

ANEXO E: PROGRAMA “TIEMPOS.INC” UTILIZADO EN EL PROYECTO.

```
CBLOCK
R_ContA
R_ContB
R_ContC
ENDC
Retardo_10micros
  nop
  nop
  nop
  nop
  nop
Retardo_5micros
  nop
Retardo_4micros
  return
Retardo_500micros
  nop
  movlw    d'164'
  goto     RetardoMicros
Retardo_200micros
  nop
  movlw    d'64'
  goto     RetardoMicros
Retardo_100micros
  movlw    d'31'
  goto     RetardoMicros
Retardo_50micros
  nop
  movlw    d'14'
  goto     RetardoMicros
Retardo_20micros
  movlw    d'5'
RetardoMicros
  movwf    R_ContA
Rmicros_Bucle
  decfsz   R_ContA,F
  goto     Rmicros_Bucle
  return
Retardo_200ms
  movlw    d'200'
  goto     Retardos_ms
Retardo_100ms
  movlw    d'100'
  goto     Retardos_ms
Retardo_50ms
  movlw    d'50'
  goto     Retardos_ms
Retardo_20ms
  movlw    d'20'
  goto     Retardos_ms
Retardo_10ms
  movlw    d'10'
  goto     Retardos_ms
```



```

Retardo_5ms
    movlw    d'5'
    goto    Retardos_ms
Retardo_2ms
    movlw    d'2'
    goto    Retardos_ms
Retardo_1ms
    movlw    d'1'
Retardos_ms
    Movwf    R_ContB
R1ms_BucleExterno
    movlw    d'249'
    movwf    R_ContA
R1ms_BucleInterno
    nop
    decfsz   R_ContA,F
    goto    R1ms_BucleInterno
    decfsz   R_ContB,F
    goto    R1ms_BucleExterno
    return
Retardo_20s
    movlw    d'200'
    goto    Retardo_1Decima
Retardo_10s
    movlw    d'100'
    goto    Retardo_1Decima
Retardo_5s
    movlw    d'50'
    goto    Retardo_1Decima
Retardo_2s
    movlw    d'20'
    goto    Retardo_1Decima
Retardo_1s
    movlw    d'10'
    goto    Retardo_1Decima
Retardo_500ms
    movlw    d'5'
Retardo_1Decima
    movwf    R_ContC
R1Decima_BucleExterno2
    movlw    d'100'
    movwf    R_ContB
R1Decima_BucleExterno
    movlw    d'249'
    movwf    R_ContA
R1Decima_BucleInterno
    nop
    decfsz   R_ContA,F
    goto    R1Decima_BucleInterno
    decfsz   R_ContB,F
    goto    R1Decima_BucleExterno
    decfsz   R_ContC,F
    goto    R1Decima_BucleExterno2
    return

```

ANEXO F: PROGRAMA “LCD.INC” UTILIZADO EN EL PROYECTO.

```
CBLOCK
LCD_Dato
LCD_GuardaDato
LCD_GuardaTRISB
LCD_Auxiliar1
LCD_Auxiliar2
ENDC

LCD_CaracteresPorLinea EQU .16
#define LCD_PinRSPORTC,5

#define LCD_PinEnable PORTC,4
#define LCD_BusDatos PORTB

LCD_Inicializa
    bsf STATUS,RP0
    bcf LCD_PinRS
    bcf LCD_PinEnable
    bcf STATUS,RP0
    bcf LCD_PinEnable
    bcf LCD_PinRS
    call Retardo_20ms
    movlw b'00110000'
    call LCD_EscribeLCD
    call Retardo_5ms
    movlw b'00110000'
    call LCD_EscribeLCD
    call Retardo_200micros
    movlw b'00110000'
    call LCD_EscribeLCD
    call Retardo_20micros
    movlw b'00100000'
    call LCD_EscribeLCD
    call Retardo_20micros
    call LCD_2Lineas4Bits5x7
    call LCD_Borra
    call LCD_CursorOFF
    call LCD_CursorIncr
    return
LCD_EscribeLCD
    andlw b'11110000'
    movwf LCD_Dato
    movf LCD_BusDatos,W
    andlw b'00001111'
    iorwf LCD_Dato,F
    bsf STATUS,RP0
    movf TRISB,W
    movwf LCD_GuardaTRISB
    movlw b'00001111'
    andwf PORTB,F
    bcf STATUS,RP0
```

```

        movf          LCD_Dato,W
        movwf        LCD_BusDatos
        bsf          LCD_PinEnable
        bcf          LCD_PinEnable
        bsf          STATUS,RP0
        movf          LCD_GuardaTRISB,W
        movwf        TRISB
        bcf          STATUS,RP0
        return
LCD_CursorIncr
        movlw        b'00000110'
        goto         LCD_EnviaComando
LCD_Linea1
        movlw        b'10000000'
        goto         LCD_EnviaComando
LCD_Linea2
        Movlw        b'11000000'
        goto         LCD_EnviaComando
LCD_Linea3
        movlw        b'10010100'
        goto         LCD_EnviaComando
LCD_Linea4
        movlw        b'11010100'
        goto         LCD_EnviaComando
LCD_PosicionLinea1
        iorlw        b'10000000'
        goto         LCD_EnviaComando
LCD_PosicionLinea2
        Iorlw        b'11000000'
        goto         LCD_EnviaComando
LCD_OFF
        movlw        b'00001000'
        goto         LCD_EnviaComando
LCD_CursorON
        movlw        b'00001110'
        goto         LCD_EnviaComando
LCD_CursorOFF
        movlw        b'00001100'
        goto         LCD_EnviaComando
LCD_Borra
        movlw        b'00000001'
        goto         LCD_EnviaComando
LCD_2Lineas4Bits5x7
        movlw        b'00101000'
LCD_EnviaComando
        bcf          LCD_PinRS
        goto         LCD_Envia
LCD_Caracter
        bsf          LCD_PinRS
        call         LCD_CodigoCGROM
LCD_Envia
        Movwf        LCD_GuardaDato
        call         LCD_EscribeLCD
        swapf        LCD_GuardaDato,W
        call         LCD_EscribeLCD

```

```

        btfss      LCD_PinRS
        call       Retardo_2ms
        call       Retardo_50micros
        return
LCD_CodigoCGROM
        movwf     LCD_Dato
LCD_EnheMinuscula
        sublw     'ñ'
        btfss     STATUS,Z
        goto      LCD_EnheMayuscula
        movlw     b'11101110'
        movwf     LCD_Dato
        goto      LCD_FinCGROM
LCD_EnheMayuscula
        movf      LCD_Dato,W
        sublw     'Ñ'
        btfss     STATUS,Z
        goto      LCD_Grado
        movlw     b'11101110'
        movwf     LCD_Dato
        goto      LCD_FinCGROM
LCD_Grado
        movf      LCD_Dato,W
        sublw     'º'
        btfss     STATUS,Z
        goto      LCD_FinCGROM
        movlw     b'11011111'
        movwf     LCD_Dato
LCD_FinCGROM
        movf      LCD_Dato,W
        return
LCD_LineaEnBlanco
        movlw     LCD_CaracteresPorLinea
        goto      LCD_EnviaBlancos
LCD_UnEspacioBlanco
        movlw     .1
        goto      LCD_EnviaBlancos
LCD_DosEspaciosBlancos
        movlw     .2
        goto      LCD_EnviaBlancos
LCD_TresEspaciosBlancos
        movlw     .3
LCD_EnviaBlancos
        Movwf     LCD_Auxiliar1
LCD_EnviaOtroBlanco
        movlw     ''
        call      LCD_Caracter
        decfsz    LCD_Auxiliar1,F
        goto      LCD_EnviaOtroBlanco
        return
LCD_Byte
        movwf     LCD_Auxiliar2
        andlw     b'11110000'
        btfss     STATUS,Z
        goto      LCD_VisualizaAlto

```

```

        movlw ''
        call    LCD_Character
        goto   LCD_VisualizaBajo
LCD_ByteCompleto
        movwf  LCD_Auxiliar2
LCD_VisualizaAlto
        swapf  LCD_Auxiliar2,W
        call   LCD_Nibble
LCD_VisualizaBajo
        movf   LCD_Auxiliar2,W
LCD_Nibble
        andlw  b'00001111'
        movwf  LCD_Auxiliar1
        sublw  0x09
        btfss STATUS,C
        goto   LCD_EnviaByteLetra
        movf   LCD_Auxiliar1,W
        addlw  '0'
        goto   LCD_FinVisualizaDigito
LCD_EnviaByteLetra
        movf   LCD_Auxiliar1,W
        addlw  'A'-0x0A
LCD_FinVisualizaDigito
        goto   LCD_Character

```

BIBLIOGRAFÍA

1. Fernando E. Valdes Pérez. Microcontrollers Fundamentals and Applications with PIC. CRC Press, Taylor and Francis Group.
2. Julio Sánchez. Microcontrollers Programming The Microchip PIC. CRC Press, Taylor and Francis Group.
3. Enrique Palacios. Microcontrolador PIC16F84A. Alfaomega.
4. David Muñoz Rodríguez. Sistemas Inalámbricos de Comunicación Personal. Alfaomega.
5. Jose M. Huidobro Moya. Comunicaciones en Redes WLAN. Limusa, Noriega Editores.
6. Javier Andres Alonso. Redes Privadas Virtuales. Alfaomega.
7. Boylestad Nashelsky. Electronica: teoría de circuitos y dispositivos electrónicos. Prentice Hall.
8. Albert Paul Malvino. Principios de Electronica. Mcgraw-Hill.
9. C.J Savant. Diseño Electronico. Prentice Hall.
10. Muhammad Rashid. Electronica de Potencia. Prentice Hall.
11. Hamley Aban. Electrónica. Prentice Hall.
12. <http://www.eveliux.com/mx/la-evolucion-de-la-telefonía-movil.php>
13. http://es.wikipedia.org/wiki/Tarjeta_SIM
14. <http://dspace.epn.edu.ec/bitstream/15000/8638/8/T10696CAP3.pdf>
15. D. Sundararajan. The Discrete Fourier Transform: Theory, Algorithms and applications. Singapore: World Scientific.
16. Kamisetty Ramamohan Rao. Fast Fourier Transform: algorithms and applications. Dordrecht: Springer Verlag.
17. Ronald Newbold Bracewell. The Fourier Transform and its applications. McGraw-Hill.
18. Enrique Ras Oliva. Análisis de fourier y cálculo operacional aplicados a la electrotecnia. Marcombo.
19. George Kennedy. Electronic Communication System. McGraw-Hill.