



**INSTITUTO POLITÉCNICO NACIONAL**



**Escuela superior de Ingeniería Mecánica y Eléctrica**

**Sistema de seguridad, control y monitoreo a distancia de un recinto aplicando un sistema inmerso de 32 bits e interfaces de comunicación Ethernet y ZigBee.**

**TESIS**

Que para obtener el título de:

**Ingeniero en Comunicaciones y Electrónica**

Presentan:

**Asbel Fuentes Vargas**

**Sergio Velasco López**

Asesor:

**Ing. Alejandro Vicente Lugo silva**

Ciudad de México  
Mayo, 2013

**INSTITUTO POLITÉCNICO NACIONAL**  
**ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELECTRICA**  
**UNIDAD PROFESIONAL “ADOLFO LÓPEZ MATEOS”**

**TEMA DE TESIS**

**QUE PARA OBTENER EL TITULO DE INGENIERO EN COMUNICACIONES Y ELECTRÓNICA  
POR LA OPCIÓN DE TITULACIÓN TESIS COLECTIVA Y EXAMEN ORAL INDIVIDUAL  
DEBERA(N) DESARROLLAR**

**C. ASBEL FUENTES VARGAS**

**C. SERGIO VELASCO LÓPEZ**

**“SISTEMA DE SEGURIDAD Y MONITOREO A DISTANCIA DE UN RECINTO APLICANDO  
UN SISTEMA INMERSO DE 32 BITS E INTERFACES DE COMUNICACIÓN ETHERNET Y  
ZIGBEE”**

DISEÑAR UN SISTEMA PARA LA ADQUISICIÓN Y EL CONTROL LOCAL Y REMOTO A TRAVÉS DE INTERNET, DE LAS CONDICIONES DE SEGURIDAD EN UN RECINTO CERRADO, APLICANDO A NIVEL PROTOTIPO UN SISTEMA INMERSO DE 32 BITS ACOPLADO A INTERFACES DE COMUNICACIÓN ETHERNET Y ZIGBEE.

- ❖ DISEÑAR Y PROGRAMAR EL SOFTWARE DEL SISTEMA INMERSO DE 32 BITS UTILIZANDO LENGUAJE C.
- ❖ DISEÑAR Y CONSTRUIR LAS INTERFACES DE ADQUISICIÓN DE DATOS Y DE CONTROL.
- ❖ DISEÑAR Y PROGRAMAR LAS REDES DE COMUNICACIÓN INALÁMBRICA (ZIG BEE) Y TCP IP (ETHERNET) DEL SISTEMA.
- ❖ CONSTRUIR EL SISTEMA DE INFORMACIÓN Y ADMINISTRACIÓN REMOTA DEL SISTEMA EN UNA PC.

**MÉXICO D. F., A 11 DE JUNIO DE 2012.**

**ASESOR**



**ING. ALEJANDRO VICENTE LUGO**



**M. EN C. DAVID VÁZQUEZ ÁLVAREZ**  
**JEFE DEL DEPARTAMENTO ACADÉMICO DE**  
**INGENIERÍA EN COMUNICACIONES Y ELECTRÓNICA**

## *Dedicatorias*

*A mis padres, por ser el pilar fundamental de todos mis logros tanto en la vida como en los académicos, por su incondicional apoyo perfectamente mantenido a través del tiempo.*

*A mi hermana Argel, por ser el ejemplo de una hermana mayor de la cual aprendí a valorar mis aciertos y levantarme de mis tropiezos.*

*Todo este trabajo ha sido posible gracias a ellos.*

**Asbel**

*A toda mi familia que estuvo al pendiente de mí guiándome con educación, buenos consejos y críticas que mejoraron mi persona, pero en especial a mis padres que me han dado todo lo que necesitaba para llegar a hasta este punto como lo es su apoyo, educación, valores y amor. A mi hermana que me ha motivado y para quien quiero ser un ejemplo. También para aquella persona que sé que me observa y cuida desde el cielo.*

*Muchas gracias sin ustedes no hubiera sido posible alcanzar mis metas.*

**Sergio**

### Objetivo general.

- Diseñar un sistema para la adquisición de datos, el control local y remoto a través de internet, de las condiciones de seguridad en un recinto cerrado, aplicando a nivel prototipo un sistema inmerso de 32 bits acoplado a interfaces de comunicación Ethernet y ZigBee.

### Objetivos particulares.

- Diseñar y programar el software del sistema inmerso de 32 bits utilizando lenguaje C.
- Diseñar y construir las interfaces de adquisición de datos y de control.
- Diseñar y programar las redes de comunicación inalámbrica (Zig Bee) y TCPI IP (Ethernet) del sistema.
- Construir el sistema de información y administración remota del sistema en una pc.

# Contenido

<b>Objetivo general.....</b>	<b>IV</b>
<b>Objetivos particulares. ....</b>	<b>IV</b>
<b>Introducción.....</b>	<b>VI</b>
<b>1 Domótica, protocolos y páginas web .....</b>	<b>1</b>
1.1 Antecedentes .....	1
1.2 Aplicaciones .....	2
1.3 Elementos de un sistema domótico .....	4
1.4 Normatividad y estandarización.....	6
1.5 Protocolos de comunicación .....	10
1.5.1 Protocolos utilizados en la pila TCP/IP de Microchip .....	10
1.5.2 Modelo de Referencia TCP/IP.....	11
1.5.3 Ethernet.....	12
1.5.4 Protocolo de resolución de direcciones (ARP).....	13
1.5.5 Protocolo de Internet (IP) .....	13
1.5.6 Protocolo de Mensajes de Control de Internet (ICMP) .....	13
1.5.7 Protocolo de Control de Transmisión (TCP) .....	14
1.5.8 Protocolo de Datagramas de Usuario (UDP) .....	14
1.5.9 Protocolo de configuración dinámica de host (DHCP).....	14
1.5.10 Protocolo de Administración de Red Simple (SNMP).....	14
1.5.11 Protocolo de Transferencia de Hipertexto (HTTP).....	15
1.5.12 Protocolo de Transferencia de Archivos (FTP) .....	15
1.5.13 Protocolo ZigBee.....	15
1.5.14 Recommended Standard 232 (RS-232).....	17
1.5.15 Transmisor-Receptor Asíncrono Universal (UART) .....	18
1.6 Páginas web.....	19
<b>2 Descripción del sistema propuesto .....</b>	<b>22</b>
2.1 PIC32 Ethernet Starter Kit.....	24
2.2 XBee.....	25
2.2.1 Configuración de módulos XBee.....	26

## Contenido

---

2.3 Interfaz de usuario .....	31
2.3.1 Pantalla LCD JHD 162 A.....	31
2.3.2 Microcontrolador PIC16F876A.....	32
2.4 Sensores.....	33
2.4.1 Sensor de temperatura .....	34
2.4.2 Sensor de gas.....	35
2.5 Actuadores.....	37
2.5.1 Motor de CD.....	37
2.5.2 Lámpara Fluorescente Compacta.....	38
2.5.3 Alarma.....	38
<b>3 Diseño de software .....</b>	<b>40</b>
3.1 Programación del Microcontrolador de 32 bits.....	40
3.2 Archivos principales de la Pila .....	47
3.3 Cabeceras .....	48
3.4 Programación de la página web.....	50
3.4.1 Diseño de la página web del microcontrolador .....	52
3.4.2 Comprimiendo páginas web con MPFS .....	57
3.5 Programación de la interfaz de usuario local .....	60
3.6 Programación de módulos XBee .....	66
<b>4 Diseño y desarrollo de las interfaces de monitoreo y control.....</b>	<b>72</b>
4.1 Diseño de PCB para módulo XBee.....	72
4.2 Sensor de Temperatura.....	74
4.3 Sensor de gas .....	77
4.4 Interfaz de iluminación .....	82
4.5 Alarma .....	84
4.6 Interfaz de Motor controlado por PWM.....	90
4.7 Interfaz de usuario local .....	93
<b>5 Resultados .....</b>	<b>100</b>
5.1 Manual de Usuario.....	100
5.2 Mejoras.....	111

## Contenido

---

<b>Conclusiones.....</b>	<b>112</b>
<b>Anexo A. Programación de la página web.....</b>	<b>113</b>
<b>Anexo B. Interrupción de la UART para procesar las tramas en la interfaz local.....</b>	<b>114</b>
<b>Glosario .....</b>	<b>115</b>
<b>Bibliografía .....</b>	<b>117</b>

### Introducción.

En la actualidad, la automatización de viviendas está en auge y es un paso hacia el futuro. Existen diferentes maneras de implementar la domótica, motivo por el cual, se ha decidido optar por una nueva tecnología, la de los PIC's con sistemas inmersos, obteniendo el desarrollo de una aplicación que reduzca en costo y tamaño lo que hoy se puede encontrar en los mercados.

Con la ayuda de Microchip y sus diferentes productos, se desarrolló un software que nos permitirá el monitoreo y control de las luces de una vivienda, un sensor de temperatura, de gas, control de la velocidad de un motor y una alarma.

Recalcando que el mismo proyecto, con sus respectivas modificaciones, podrá ser utilizado para diferentes aplicaciones eléctricas.

Esta aplicación consta de una tarjeta de control, PIC32MX512 con módulo Ethernet programado con el Ambiente de Desarrollo MPLAB utilizando la pila TCP/IP que nos brinda Microchip de manera gratuita.

También se desarrollarán las interfaces de control y de monitoreo que se comunicarán a la tarjeta de manera inalámbrica por medio del protocolo ZigBee y los componentes XBee.

El acceso al sistema será de manera remota a través de una página web y de manera local, por medio de una botonera que también permite modificar y revisar el estado de las variables antes mencionadas, reportándolo en la página web y en la pantalla de la botonera.

Todo el análisis y funcionamiento del proyecto, será demostrado a escala en una pequeña vivienda con su respectiva instalación eléctrica.

# 1

## Domótica, protocolos y páginas web

La evolución marca el ritmo de la vida y las casas tampoco pueden escapar a ella. De la cueva con fuego, para calentar e iluminar, a las antorchas, las velas, el candil y por último: la electricidad.

La electricidad nos ha permitido elevar el nivel de confort en nuestras casas y ha dado paso a la entrada de los electrodomésticos: lavadora, frigorífico, lavavajillas, horno, placas vitrocerámicas, máquinas capaces de realizar tareas cotidianas de forma casi autónoma (aún queda por solucionar la carga y descarga de las mismas), elevando nuestro nivel de confort a cotas en otro tiempo inimaginables.

Estas máquinas no existirían sin el desarrollo de una nueva evolución: la electrónica, permitiendo realizar programaciones (rutinas), que regulan cada proceso (lavado en frío, grabación de un video, etc.).

La siguiente evolución que ha llegado es la: Domótica, que se encarga de la integración y regulación de ambos sistemas (eléctricos y electrónicos), de tal manera que “la casa” es capaz de “sentir” (detectar la presencia de personas, la temperatura, el nivel de luz, etc.) y reaccionar por sí sola, a estos estímulos (regulando el clima, la iluminación, conectando la alarma, etc.), al mismo tiempo que es capaz de comunicarse e interactuar con nosotros (telecontrol) por multitud de medios (pantalla táctil, PC, móvil, etc.), llegando a elevadas cotas de confort, seguridad y sobretodo: ahorro energético.

La Historia de la domótica comprende una serie de etapas, desde los primeros protocolos orientados al "control remoto", hasta los grandes protocolos capaces de realizar "funciones lógicas complejas", para satisfacer las más exigentes programaciones de regulación y preparados para la verdadera Revolución Domótica: La autorregulación.

### 1.1 Antecedentes

La vivienda es el lugar donde satisfacemos nuestras necesidades así que debe ser confortable y segura. Lo cual ha generado que las viviendas sufran una transformación por la implementación de los diversos avances tecnológicos que se han llevado a lo largo del tiempo, maximizando el concepto de vivienda.

Esta transformación ha generado un nuevo concepto llamado domótica que proviene de la unión de las palabras *domus* (que significa *casa* en latín) y *tica* (de *automática*, palabra

en griego, 'que funciona por sí sola'). Se entiende por domótica al conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación, que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar. Se podría definir como la *integración de la tecnología en el diseño inteligente de una vivienda*.

Esta no solo va dirigida a las viviendas, sino también a los comercios, edificios, granjas. etc. La domótica se ha implantado desde hace décadas, pero, desde que se creó el Internet este ha tomado un giro controversial, los modelos tecnológicos relacionados a este han progresado y forman parte del futuro de la domótica. Las tecnologías inalámbricas WiFi y las redes de Internet, creen haberse constituido, como las tecnologías del entorno digital que evolucionarán, y sobre las cuales la domótica debería mantenerse para poder aumentar el uso de las tecnologías en los hogares figura 1.1. Debemos de tener en cuenta algunos elementos para poder instalar este sistema como es:

- El incremento en el confort
- Climatización del control de luces, ventanas, cortinas, persianas, etc.
- Uso de energía renovable como la energía solar, la geotérmica y la eólica.

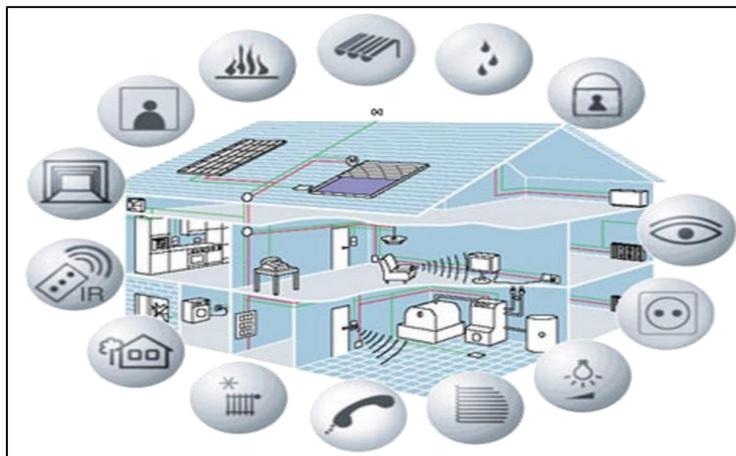


Figura 1.1 Integración de sistemas en una vivienda inteligente.

### 1.2 Aplicaciones

Los servicios que ofrece la domótica se pueden agrupar según cinco aspectos o ámbitos principales:

1. Ahorro energético. El ahorro energético no es algo tangible, sino un concepto al que se puede llegar de muchas maneras. En muchos casos no es necesario sustituir los

aparatos o sistemas del hogar por otros que consuman menos sino una *gestión eficiente* de los mismos.

- a) Climatización: programación y zonificación.
- b) Gestión eléctrica.
  - Racionalización de cargas eléctricas: desconexión de equipos de uso no prioritario en función del consumo eléctrico en un momento dado.
  - Gestión de tarifas, derivando el funcionamiento de algunos aparatos a horas de tarifa reducida.
- c) Uso de energías renovables.

2. Confort. Conlleva todas las actuaciones que se puedan llevar a cabo que mejoren el confort en una vivienda. Dichas actuaciones pueden ser de carácter tanto pasivo, como activo o mixtas.

- a) Iluminación.
  - Apagado general de todas las luces de la vivienda.
  - Automatización del apagado/ encendido en cada punto de luz.
  - Regulación de la iluminación según el nivel de luminosidad ambiente.
- b) Automatización de todos los distintos sistemas/ instalaciones / equipos dotándolos de control eficiente y de fácil manejo.
- c) Integración del portero al teléfono, o del videoportero al televisor.
- d) Control vía Internet.
- e) Gestión Multimedia y del ocio electrónico.
- f) Generación de macros y programas de forma sencilla para el usuario.

3. Seguridad. Consiste en una red de seguridad encargada de proteger tanto los bienes patrimoniales y la seguridad personal.

- a) Simulación de presencia.
- b) Alarmas de Detección de incendio, fugas de gas, escapes de agua, concentración de monóxido en garajes.
- c) Alerta médica. Teleasistencia.
- d) Cerramiento de persianas puntual y seguro.
- e) Acceso a Cámaras IP.

4. Comunicaciones. Son los sistemas o infraestructuras de comunicaciones que posee el hogar.

- a) Ubicuidad en el control tanto externo como interno, control remoto desde Internet, PC, mandos inalámbricos (p.ej. PDA con WiFi).
- b) Transmisión de alarmas.
- c) Intercomunicaciones.

5. Telegestión y Accesibilidad. Diseño para todos, un diseño accesible para la diversidad humana, la inclusión social y la igualdad. Este enfoque constituye un reto Ético y creativo. Donde las personas con discapacidad reducida puedan acceder a estas tecnologías sin temor a un obstáculo del tipo de tecnología o arquitectura.

### 1.3 Elementos de un sistema domótico

Son muchos los elementos que componen los distintos sistemas de automatización de viviendas y edificios, desde una central de gestión para sistemas integrados hasta un mando automático a distancia. De todos los elementos que los integran, existen cinco básicos: controladores, actuadores, sensores, conexión a Internet, sistemas de comunicación entre dispositivos y usuario.

#### a) Controladores.

Se trata de un equipo electrónico, que, tal como su mismo nombre lo indica, se ha diseñado para controlar procesos en tiempo real, el cual debe ser programado. Por lo general, es posible encontrar este tipo de equipos en ambientes donde se requiera manipular y automatizar.

En este caso se empleará como controlador un Sistema Inmerso el cual consiste en un sistema basado en microprocesador, cuyo hardware y software están específicamente diseñados y optimizados para resolver un problema concreto eficientemente, reduciendo costos, elevando el consumo y su fabricación. Un sistema inmerso utiliza la arquitectura de una computadora de forma simplificada, reduciendo las velocidades de su procesador y sus periféricos ajustándolas a las necesidades específicas de una aplicación. Las características que distinguen a estos sistemas son: tiempo de respuesta (Sistemas Inmersos en Tiempo Real), el tamaño, consumo de potencia y bajos costos.

#### b) Actuadores.

Por otra parte, tenemos los actuadores. Son elementos que utiliza el sistema para modificar el estado de ciertos equipos e instalaciones. Entre los más comúnmente utilizados se distinguen los siguientes:

- a) Relevadores (o relés de actuación) de carril DIN.

- b) Relevadores para base de enchufe.
- c) Electroválvulas de corte de suministro (gas y aguas).
- d) Válvulas para la zonificación de la calefacción por agua caliente.
- e) Sirenas o elementos zumbadores, para el aviso de alarmas en curso.

### c) Sensores.

Los sensores son los elementos que utiliza el sistema para conocer el estado de ciertos parámetros (la temperatura ambiente, la existencia de un escape de agua, gas, etc.).

Entre los más comúnmente utilizados se distinguen los siguientes:

- a) Termostato de ambiente, destinado a medir la temperatura de la estancia y permitir la modificación de parámetros de consigna por parte del usuario.
- b) Sensor de temperatura interior, destinado a medir únicamente la temperatura de la estancia.
- c) Sonda de humedad, destinada a detectar posibles escapes de agua en cocinas, aseos, etc.
- d) Detector de fugas de gas, para la detección de posibles fugas de gas en cocina, etc.
- e) Detector de humo y/o fuego, para la detección de conatos de incendio.
- f) Sensor de presencia, para detección de intrusiones no deseadas en la vivienda.
- g) Receptor de infrarrojos.

### d) Conexión a Internet.

La conexión es el elemento fundamental en la vivienda domótica. Por una parte, permite tener una conexión permanente con el exterior, lo que es imprescindible para disfrutar de servicios como la teleasistencia, la televigilancia o la gestión remota de los dispositivos. Por otra parte, permite disponer de la capacidad de transmisión necesaria para disfrutar de servicios como el vídeo bajo demanda o la videovigilancia del hogar.

### e) Comunicación entre dispositivos y usuario.

El conjunto de elementos que intervienen en el proceso de intercambio de información forman un sistema de comunicación. En todo sistema de comunicación podemos distinguir los siguientes componentes:

- a) Emisor: es el elemento que transmite la información.
- b) Receptor: es el elemento que recibe la información.

- c) Canal: es el medio a través del cual tiene lugar el intercambio de información entre el transmisor y receptor.

La información que se transmite entre el receptor y el emisor debe adaptarse al canal de transmisión. Ello implica la necesidad de disponer de un soporte adecuado a través del cual pueda viajar la información. Los sistemas de comunicaciones actuales utilizan básicamente dos tipos de soporte, lo que permite hablar de dos clases diferentes de comunicación:

- a) Comunicación alámbrica: también llamada comunicación por cable, pues tiene lugar a través de líneas o cables (tradicionalmente de cobre) que unen al emisor y al receptor. La información se transmite mediante impulsos eléctricos.
- DSL
  - Fibra óptica
  - Power Line Communications y X10
  - Cable (coaxial y par trenzado)
- b) Comunicación inalámbrica: en este caso el soporte material a través del cual tiene lugar la comunicación es el propio espacio, y concretamente en la atmósfera terrestre, el aire. La información se transmite mediante ondas de radio.
- Wifi
  - GPRS
  - Bluetooth
  - Radiofrecuencia
  - Infrarrojos
  - ZigBee

### 1.4 Normatividad y estandarización

En el mundo existen diversas organizaciones sin fines de lucro que se encargan de llevar a cabo un seguimiento, regulación, certificación y normalización de la domótica, que a su vez comparten información e interactúan entre sí para poder hacer de esta, compatible y universal.

A continuación se mencionan las organizaciones más representativas que hoy en día se encargan de esta regular y certificar alrededor del mundo.

### *IMEI.*

En México el encargado de evaluar los grados de inteligencia de un edificio es el IMEI, (Instituto Mexicano del Edificio Inteligente), y en resumen debe cumplir con los siguientes requisitos:

- Eficiencia en el uso de energéticos y consumibles, renovables (Máxima Economía).
- Adaptabilidad a un bajo costo a los continuos cambios tecnológicos requeridos por sus ocupantes y su entorno (Máxima Flexibilidad).
- Capacidad de proveer un entorno Ecológico interior y exterior respectivamente habitable y sustentable, altamente seguro que maximice la eficiencia en el trabajo a los niveles óptimos de confort de sus ocupantes según sea el caso (Máxima Seguridad para el entorno, usuario y patrimonial).
- Eficazmente comunicativo en su operación y mantenimiento, (Máxima automatización de la actividad).
- Operando y mantenido bajo estrictos métodos de optimización (Máxima predicción y prevención, refaccionamiento virtual).

### *KNX.*

KNX Association es el creador y propietario de la tecnología KNX – el único ESTÁNDAR abierto para todas las aplicaciones de control de la vivienda y el edificio, como por ejemplo el control de la iluminación y las persianas, así como variados sistemas de seguridad, calefacción, ventilación, aire acondicionado, monitorización, alarma, control de agua, gestión de energía, contador, así como electrodomésticos del hogar, audio/video y mucho más. Esta tecnología puede ser usada tanto en viviendas y edificios de nueva construcción, como en los ya existentes.

Para los miembros de la KNX Association el sistema está a su disposición gratis, además puede ser implementado sobre cualquier plataforma de microprocesador. Todos los productos que deseen llevar el logo KNX son certificados para garantizar la compatibilidad, interworking e interoperabilidad del sistema.

- KNX es el único estándar mundial para el control de la vivienda y del edificio con:
- Una única herramienta de puesta en marcha (ETS).
- Una completa gama de medios de transmisión (TP, PL, RF y IP).
- Una completa gama de modos de configuración (System y Easy modos).

KNX está aprobado como:

- Estándar Europeo (CENELEC EN 50090 y CEN EN 13321-1).
- Estándar Internacional (ISO/IEC 14543-3).
- Estándar Chino (GB/Z 20965).
- Estándar Norteamericano (ANSI/ASHRAE 135).

Este estándar se fundamenta en más de 15 años de experiencia en el mercado, que incluye a sus predecesores EIB, EHS y BatiBUS. Más de 100 compañías miembros por todo el mundo de diversas aplicaciones cuentan con casi 7.000 grupos de productos certificados KNX en sus catálogos. KNX Association tiene acuerdos de asociados con más de 21.000 compañías instaladoras en 70 países, más de 50 universidades, así como más de 100 centros de formación.

*CEDOM.*

La Asociación Española de Domótica, CEDOM, nació en 1992 como una iniciativa de un grupo de empresas fabricantes de material eléctrico que apostaron por el sector de la domótica, tratando de impulsar el mercado y facilitando la labor de venta de los productos de las empresas miembro.

Inicialmente fue creado como el Comité Español de la Domótica. En el año 2001, se decidió abrir la entidad al resto de colectivos presentes en el mercado de la Domótica, no sólo a fabricantes, y se vio la necesidad de cambiar la denominación de Comité a la de Asociación, aunque se mantuvo el nombre de CEDOM debido a que ya era un nombre reconocido en el sector eléctrico.

Desde entonces, CEDOM ha ido adaptándose a los cambios y dificultades que ha sufrido el sector de la Domótica. Así, en la actualidad, CEDOM es la única Asociación a nivel nacional que reúne a todos los agentes del sector de la Domótica en España: fabricantes de productos domóticos, fabricantes de sistemas, instaladores, integradores, arquitecturas e ingenierías, centros de formación, universidades, centros tecnológicos.

El principal objetivo de CEDOM es la promoción y difusión de la Domótica en general, sin diferenciación de sistemas, protocolos de comunicación, tipos de productos o empresas. El resto de objetivos de CEDOM son:

- Estudiar, investigar y promover el concepto de Domótica en general, su desarrollo tecnológico y su aplicación como Gestión Técnica de Edificios y Viviendas.
- Promover, activar, desarrollar, coordinar y proteger los intereses de sus asociados en la materia relacionada en el párrafo anterior.
- Desarrollar herramientas y plataformas para el desarrollo del mercado de la domótica en nuestro país.
- Informar y formar a todos los agentes que intervienen en la cadena de valor de la Domótica.

En resumen, CEDOM es una Asociación al servicio de sus asociados y con un objetivo claro, aumentar la implantación de la domótica en España.

### *AENOR.*

Fue designada por el Ministerio de Industria y Energía en 1986 como entidad para desarrollar las actividades de normalización y certificación (N+C) y fue reconocida como organismo de normalización y para actuar como entidad de certificación por el Real Decreto 2200/1995, en desarrollo de la Ley 21/1992 de Industria. Está acreditada por la Entidad Nacional de Acreditación (ENAC) como organismo de certificación de sistemas de la calidad ISO 9000, sistemas de gestión ambiental ISO 14000, verificación medioambiental y sistemas de la calidad ISO/TS 16949 para el sector automotriz y sistemas de calidad según el Esquema ATECMA (industria aeroespacial). También cuenta con la acreditación de ENAC para la actividad de certificación de proyectos de I+D+I y de certificación de productos de numerosos comités técnicos de certificación. En México cuenta con la Acreditación de la Entidad Mexicana de Acreditación (EMA), en ISO 9001, ISO 14000 y OHSAS 18001.

AENOR es Miembro de la red de IQNET, que está formado por organismos de certificación líderes en la certificación de empresas en sus respectivos países. En los ámbitos internacional y comunitario AENOR está presente en los principales foros internacionales, europeos y americanos, lo que garantiza la participación de nuestro país en el desarrollo de la normalización y el reconocimiento internacional de la certificación de AENOR.

Actualmente, AENOR cuenta con más de 890 miembros pertenecientes a la práctica totalidad del entramado industrial español. Este hecho permite afirmar la independencia y objetividad con que AENOR lleva a cabo el desarrollo de sus actividades. Como asociación, y de acuerdo con la legislación española vigente, AENOR puede constituir sociedades, dentro y fuera de España, de participación mayoritaria. En 1997 se creó AENOR MÉXICO,

Sociedad Anónima de Capital Variable, y en 2001 la sociedad anónima unipersonal AENOR Internacional, que a su vez ha constituido AENOR CHILE, AENOR ITALIA, LUSAENOR y AENOR EL SALVADOR.

Igualmente, participa como socio mayoritario en el Centro de Ensayos, Innovación y Servicios (CEIS, S.L.), creado en 1999.

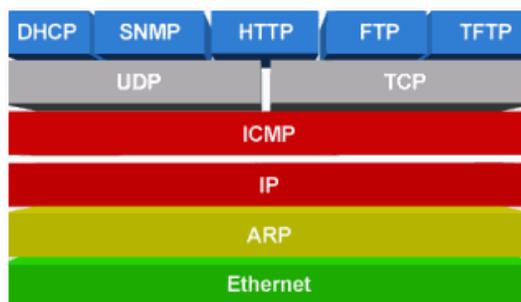
Siendo uno de los organismos de Certificación y Capacitación más importantes en México, con 450 certificados emitidos tanto en México como en países de Centroamérica. Los sectores en los que destaca la certificación de sistemas de gestión son el sector químico, eléctrico, industrial, de servicio, entre otros, y en el último año ha desempeñado un papel muy relevante en el sector de la construcción ya que 210 empresas verificadas y supervisoras de construcción de vivienda iniciaron el proceso de certificación de su sistema de gestión según la norma Mexicana NMX-C-442-ONNCCEE-2004.

### 1.5 Protocolos de comunicación

Los protocolos son reglas de comunicación que permiten el flujo de información entre equipos que manejan lenguajes distintos, por ejemplo, dos computadoras conectados en la misma red pero con protocolos diferentes no podrían comunicarse jamás, para ello, es necesario que ambas "hablen" el mismo idioma. El protocolo TCP/IP fue creado para las comunicaciones en Internet. Para que cualquier computadora se conecte a Internet es necesario que tenga instalado este protocolo de comunicación.

#### 1.5.1 Protocolos utilizados en la pila TCP/IP de Microchip

Microchip TCP/IP está formado por un conjunto de capas que dan acceso a distintos protocolos: ARP, IP, ICMP, UDP, TCP, DHCP, SNMP, HTTP, FTP y TFTP. Figura 2. Está programado en C, requiere aproximadamente 32K en memoria de programa y es portable a través de las distintas familias de Microchip. Esto quiere decir que funciona en micros de 8, 16 y 32 bits. Además, tiene soporte para encriptación de datos, transmisión de audio y conexión Wi-Fi.



**Figura 1.2** La pila está formada por varias capas, cada una brinda servicios a la superior.

En los siguientes temas se explicarán cada uno de los protocolos que han sido utilizados para el desarrollo del proyecto.

### 1.5.2 Modelo de Referencia TCP/IP

La familia de protocolos de Internet es un conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre redes de computadoras. En ocasiones se le denomina conjunto de protocolos TCP/IP, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP), que fueron los dos primeros en definirse, y que son los más utilizados de la familia. Existen tantos protocolos en este conjunto que llegan a ser más de 100 diferentes, entre ellos se encuentra el popular HTTP (HyperText Transfer Protocol), que es el que se utiliza para acceder a las páginas web, además de otros como el ARP (Address Resolution Protocol) para la resolución de direcciones, el FTP (File Transfer Protocol) para transferencia de archivos, y el SMTP (Simple Mail Transfer Protocol) y el POP (Post Office Protocol) para correo electrónico, TELNET para acceder a equipos remotos, entre otros.

El TCP/IP es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN) y área extensa (WAN), este protocolo consta de las siguientes capas:

- *Capa de Acceso a la Red.* La capa de acceso a la red es la primera capa de la pila TCP/IP. Ofrece la capacidad de acceder a cualquier red física, es decir, brinda los recursos que se deben implementar para transmitir datos a través de la red.
- *Capa de Internet.* La capa de Internet es la capa "más importante" (si bien todas son importantes a su manera), ya que es la que define los datagramas y administra las nociones de direcciones IP. Permite el enrutamiento de datagramas (paquetes de datos) a equipos remotos junto con la administración de su división y ensamblaje cuando se reciben.
- *Capa de Transporte.* Los protocolos de las capas anteriores permiten enviar información de un equipo a otro. La capa de transporte permite que las aplicaciones que se ejecutan en equipos remotos puedan comunicarse.
- *Capa de Aplicación.* La capa de aplicación se encuentra en la parte superior de las capas del protocolo TCP/IP. Contiene las aplicaciones de red que permiten la comunicación mediante las capas inferiores. Por lo tanto, el software en esta capa se comunica mediante uno o dos protocolos de la capa inferior (la capa de transporte), es decir, TCP o UDP.

### 1.5.3 Ethernet

Las arquitecturas de red proveen diferentes medios para resolver un problema común (mover datos rápida y eficientemente sobre el medio de la red). La arquitectura de red en particular que se esté usando (ej. Ethernet), no sólo definirá la topología de la red, sino también define como el medio de comunicación es accesado por los nodos. Existen varias arquitecturas de red disponibles, tales como Ethernet (Xerox, Intel y DEC), Token Ring (IBM), FDDI, AppleTalk (Apple computers); todas con una estrategia diferente para mover la información en la red. A continuación describiremos la arquitectura más popular en la actualidad, ETHERNET.

El término "Ethernet" se refiere a la familia de implementaciones de Redes de Área Local (LAN, Local Área Network) que incluye tres principales categorías:

- 10 Mbps Ethernet e IEEE 802.3: Especificaciones LAN que operan a 10 Mbps sobre cable coaxial
- **100 Mbps Ethernet:** Especificación LAN, también conocida como "FAST ETHERNET", que opera a 100 Mbps sobre cable par trenzado.
- **1000 Mbps Ethernet:** Especificación LAN, también conocida como Gigabit Ethernet, que opera a 1000 Mbps (1 Gbps) sobre fibra óptica y cable par trenzado.

Ethernet no es una tecnología sino una familia de tecnologías de redes de computadoras de área local LAN del tipo de Acceso Múltiple con Detección de Portadora y Detección de Colisiones CSMA/CD, que suelen cumplir con las especificaciones Ethernet, incluyendo IEEE 802.3; basada en tramas de datos. Las estaciones en una LAN CSMA/CD pueden acceder a la red en cualquier momento y, antes de enviar los datos, las estaciones CSMA/CD escuchan la red para ver si ya es operativa. Si lo está, la estación que desea transmitir espera. Si la red no está en uso, la estación transmite. Decimos que se produce una colisión cuando dos estaciones que escuchan el tráfico en la red no oyen nada y transmiten simultáneamente. En este caso, ambas transmisiones quedan desbaratadas y las estaciones deben transmitir de nuevo en otro momento. Los algoritmos Backoff son los que determinan cuando deben retransmitir las estaciones que han colisionado. Ethernet está bien adaptada a las aplicaciones en que el soporte de comunicaciones local a menudo tiene que procesar un elevado tráfico con puntas elevadas de intercambio de datos. Los estándares Ethernet no necesitan especificar todos los aspectos y funciones necesarios en un Sistema Operativo de Red (NOS). Como ocurre con otros estándares de red, la especificación Ethernet se refiere solamente a las dos primeras capas del modelo de

referencia OSI. Estas son la capa física y la de enlace; que proporcionan direccionamiento local, detección de errores, control del acceso a la capa física, entre otras.

### 1.5.4 Protocolo de resolución de direcciones (ARP)

ARP Son las siglas en inglés de Address Resolution Protocol (Protocolo de resolución de direcciones). Es un protocolo de la capa de enlace de datos responsable de encontrar la dirección hardware (Ethernet MAC) que corresponde a una determinada dirección IP. Para ello se envía un paquete (ARP request) a la dirección de difusión de la red (broadcast (MAC = FF FF FF FF FF FF)) que contiene la dirección IP por la que se pregunta, y se espera a que esa máquina (u otra) responda (ARP reply) con la dirección Ethernet que le corresponde. Cada máquina mantiene una caché con las direcciones traducidas para reducir el retardo y la carga. ARP permite a la dirección de Internet ser independiente de la dirección Ethernet, pero esto sólo funciona si todas las máquinas lo soportan. En Ethernet, la capa de enlace trabaja con direcciones físicas. El protocolo ARP se encarga de traducir las direcciones IP a direcciones MAC (direcciones físicas). Para realizar esta conversión, el nivel de enlace utiliza las tablas ARP, cada interfaz tiene tanto una dirección IP como una dirección física MAC.

ARP se utiliza en 4 casos referentes a la comunicación entre 2 hosts:

1. Cuando 2 hosts están en la misma red y uno quiere enviar un paquete a otro.
2. Cuando 2 host están sobre redes diferentes y deben usar un gateway/router para alcanzar otro host.
3. Cuando un router necesita enviar un paquete a un host a través de otro router.
4. Cuando un router necesita enviar un paquete a un host de la misma red.

### 1.5.5 Protocolo de Internet (IP)

El protocolo IP es parte de la capa de Internet del conjunto de protocolos TCP/IP. Es uno de los protocolos de Internet más importantes ya que permite el desarrollo y transporte de datagramas de IP (paquetes de datos), aunque sin garantizar su "entrega". En realidad, el protocolo IP procesa datagramas de IP de manera independiente al definir su representación, ruta y envío.

### 1.5.6 Protocolo de Mensajes de Control de Internet (ICMP)

El Protocolo de Mensajes de Control de Internet (por sus siglas en inglés de Internet Control Message Protocol) es el sub protocolo de control y notificación de errores del Protocolo de Internet (IP). Como tal, se usa para enviar mensajes de error, indicando por ejemplo que un servicio determinado no está disponible o que un router o host no puede ser localizado.

ICMP difiere del propósito de TCP y UDP ya que generalmente no se utiliza directamente por las aplicaciones de usuario en la red. La única excepción es la herramienta ping y traceroute, que envían mensajes de petición Echo ICMP (y recibe mensajes de respuesta Echo) para determinar si un host está disponible, el tiempo que le toma a los paquetes en ir y regresar a ese host y cantidad de hosts por los que pasa.

### **1.5.7 Protocolo de Control de Transmisión (TCP)**

TCP (que significa *Protocolo de Control de Transmisión*) es uno de los principales protocolos de la capa de transporte del modelo TCP/IP. En el nivel de aplicación, posibilita la administración de datos que vienen del nivel más bajo del modelo, o van hacia él, (es decir, el protocolo IP). Cuando se proporcionan los datos al protocolo IP, los agrupa en datagramas IP, fijando el campo del protocolo en 6 (para que sepa con anticipación que el protocolo es TCP). TCP es un protocolo orientado a conexión, es decir, que permite que dos máquinas que están comunicadas controlen el estado de la transmisión.

### **1.5.8 Protocolo de Datagramas de Usuario (UDP)**

User Datagram Protocol (UDP) es un protocolo del nivel de transporte basado en el intercambio de datagramas (Paquete de datos). Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera. Tampoco tiene confirmación ni control de flujo, por lo que los paquetes pueden adelantarse unos a otros; y tampoco se sabe si ha llegado correctamente, ya que no hay confirmación de entrega o recepción. Su uso principal es para protocolos como DHCP, BOOTP, DNS.

### **1.5.9 Protocolo de configuración dinámica de host (DHCP)**

DHCP (*Dynamic Host Configuration Protocol*) es un protocolo de red que permite a los clientes de una red IP obtener sus parámetros de configuración automáticamente. Se trata de un protocolo de tipo cliente/servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas van estando libres, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después.

### **1.5.10 Protocolo de Administración de Red Simple (SNMP)**

El Protocolo Simple de Administración de Red o SNMP es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. Es parte de la familia de protocolos TCP/IP. SNMP permite a los administradores supervisar el funcionamiento de la red, buscar y resolver sus problemas, y planear su crecimiento.

### 1.5.11 Protocolo de Transferencia de Hipertexto (HTTP)

Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo usado en cada transacción de la World Wide Web.

HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador web o un spider) se lo conoce como "user agent" (agente del usuario). A la información transmitida se la llama recurso y se la identifica mediante un localizador uniforme de recursos (URL). Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.

### 1.5.12 Protocolo de Transferencia de Archivos (FTP)

FTP (File Transfer Protocol - Protocolo de Transferencia de Archivos) es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

### 1.5.13 Protocolo ZigBee

Las tecnologías inalámbricas han adoptado con el paso del tiempo una manera más sencilla y cómoda de utilizar toda clase de dispositivos con el fin de mejorar el confort y las comunicaciones en general. Ésta investigación aborda la tecnología inalámbrica ZigBee, basada en el estándar 802.15.4 que por su poca introducción al mercado no es muy conocida a pesar de que no es muy reciente.

ZigBee es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radios digitales de bajo consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (wireless personal area network, WPAN). Su objetivo son las aplicaciones para redes Wireless que requieran comunicaciones seguras y fiables con baja tasa de envío de datos y maximización de la vida útil de sus baterías.

Características.

- ZigBee, también conocido como "Home RF Lite", es una tecnología inalámbrica con velocidades comprendidas entre 20 kB/s y 250 kB/s.
- Los rangos de alcance son de 10 m a 75 m.

- Puede usar las bandas libres ISM (6) de 2,4 GHz (Mundial), 868 MHz ( Europa) y 915 MHz (EEUU).
- Una red ZigBee puede estar formada por hasta 255 nodos los cuales tienen la mayor parte del tiempo el transceiver ZigBee dormido con objeto de consumir menos que otras tecnologías inalámbricas.
- Un sensor equipado con un transceiver ZigBee pueda ser alimentado con dos pilas AA durante al menos 6 meses y hasta 2 años.
- La fabricación de un transmisor ZigBee consta de menos circuitos analógicos de los que se necesitan habitualmente.
- Diferentes tipos de topologías como estrella, punto a punto, malla, árbol.
- Acceso de canal mediante CSMA/CA(7) (acceso múltiple por detección de portadora con evasión de colisiones).
- Escalabilidad de red -- Un mejor soporte para las redes más grandes, ofreciendo más opciones de gestión, flexibilidad y desempeño.
- Fragmentación -- Nueva capacidad para dividir mensajes más largos y permitir la interacción con otros protocolos y sistemas.
- Agilidad de frecuencia -- Redes cambian los canales en forma dinámica en caso que ocurran interferencias.
- Gestión automatizada de direcciones de dispositivos - El conjunto fue optimizado para grandes redes con gestión de red agregada y herramientas de configuración.
- Localización grupal -- Ofrece una optimización adicional de tráfico necesaria para las grandes redes.
- Puesta de servicio inalámbrico -- El conjunto fue mejorado con capacidades seguras para poner en marcha el servicio inalámbrico.
- Recolección centralizada de datos -- El conjunto fue sintonizado específicamente para optimizar el flujo de información en las grandes redes.

### Ventajas

- Ideal para conexiones punto a punto y punto a multipunto.
- Diseñado para el direccionamiento de información y el refrescamiento de la red.
- Opera en la banda libre de ISM 2.4 Ghz para conexiones inalámbricas.
- Óptimo para redes de baja tasa de transferencia de datos.
- Alojamiento de 16 bits a 64 bits de dirección extendida.
- Reduce tiempos de espera en el envío y recepción de paquetes.
- Detección de Energía (ED).

- Baja ciclo de trabajo - Proporciona larga duración de la batería.
- Soporte para múltiples topologías de red: Estática, dinámica, estrella y malla.
- Hasta 65.000 nodos en una red.
- 128-bit AES de cifrado - Provee conexiones seguras entre dispositivos.
- Son más baratos y de construcción más sencilla.

Desventajas.

- La tasa de transferencia es muy baja.
- Solo manipula textos pequeños comparados con otras tecnologías.
- Zigbee trabaja de manera que no puede ser compatible con bluetooth en todos sus aspectos porque no llegan a tener las mismas tasas de transferencia, ni la misma capacidad de soporte para nodos.
- Tiene menor cobertura porque pertenece a redes inalámbricas de tipo WPAN.

Tipos de Dispositivos.

Se definen tres tipos distintos de dispositivo ZigBee según su papel en la red:

- Coordinador ZigBee (ZigBeeCoordinator, ZC): El tipo de dispositivo más completo. Debe existir uno por red. Sus funciones son las de encargarse de controlar la red y los caminos que deben seguir los dispositivos para conectarse entre ellos, requiere memoria y capacidad de computación.
- RouterZigBee (ZigBeeRouter, ZR): Interconecta dispositivos separados en la topología de la red, además de ofrecer un nivel de aplicación para la ejecución de código de usuario.
- Dispositivo final (ZigBeeEndDevice, ZED): Posee la funcionalidad necesaria para comunicarse con su nodo padre (el coordinador o un router), pero no puede transmitir información destinada a otros dispositivos. De esta forma, este tipo de nodo puede estar dormido la mayor parte del tiempo, aumentando la vida media de sus baterías. Un ZED tiene requerimientos mínimos de memoria y es por tanto significativamente más barato.

### **1.5.14 Recommended Standard 232 (RS-232)**

RS232 (Recommended Standard 232, también conocido como Electronic Industries Alliance RS-232C) es una interfaz que designa una norma para el intercambio serie de datos binarios entre un DTE (Data Terminal Equipment, Equipo terminal de datos) y un DCE (Data Communication Equipment, Equipo de Comunicación de datos), aunque existen otras en las que también se utiliza la interfaz RS-232.

El estándar define características de la señal eléctrica, características mecánicas de la conexión, descripción funcional de los circuitos de intercambio y secuencia de procedimientos.

En particular, existen ocasiones en que interesa conectar otro tipo de equipamientos con este mismo estándar, como pueden ser microcontroladores. Para la conexión de estos dispositivos es necesario un circuito especializado para realizar la conversión serie-paralelo y viceversa llamado UART.

### **1.5.15 Transmisor-Receptor Asíncrono Universal (UART)**

UART son las siglas de "Universal Asynchronous Receiver-Transmitter" (en español, Transmisor-Receptor Asíncrono Universal). Éste controla los puertos y dispositivos serie. Se encuentra integrado en la placa base, en la tarjeta adaptadora o en el mismo dispositivo. Un UART dual, o DUART, combina dos UARTs en un solo chip. Existe un dispositivo electrónico encargado de generar la UART en cada puerto serie.

El controlador del UART es el componente clave del subsistema de comunicaciones series de una computadora. El UART toma bytes de datos y transmite los bits individuales de forma secuencial. En el destino, un segundo UART re ensambla los bits en bytes completos. La transmisión serie de la información digital (bits) a través de un cable único u otros medios es mucho más efectiva en cuanto a costo que la transmisión en paralelo a través de múltiples cables. Se utiliza un UART para convertir la información transmitida entre su forma secuencial y paralela en cada terminal de enlace. Cada UART contiene un registro de desplazamiento que es el método fundamental de conversión entre las forma serie y paralelo.

El UART normalmente no genera directamente o recibe las señales externas entre los diferentes módulos del equipo. Usualmente se usan dispositivos de interfaz separados para convertir las señales de nivel lógico del UART hacia y desde los niveles de señalización externos.

Las señales externas pueden ser de variada índole. Ejemplos de estándares para señalización por voltaje son RS-232, RS-422 y RS-485 de la EIA. Históricamente, se usó la presencia o ausencia de corriente en circuitos telegráficos. Algunos esquemas de señalización no usan cables eléctricos. Ejemplo de esto son la fibra óptica, infrarrojo y Bluetooth (inalámbrico). Algunos esquemas de señalización emplean una modulación de señal portadora (con o sin cables). Ejemplos son la modulación de

señales de audio con módems de línea telefónica, modulación en radio frecuencia (RF) en radios de datos y la DC-LIN para la comunicación de línea eléctrica.

### 1.6 Páginas web

Una página web, es una fuente de información adaptada para la Red Mundial (World Wide Web) y accesible mediante un navegador de Internet. Esta información se presenta generalmente en formato HTML y puede contener hiper-enlaces a otras páginas web, constituyendo la red enlazada de la World Wide Web.

Las páginas web pueden ser cargadas por una computadora local o remota, llamada Servidor Web, la cual servirá de CLIENTE. El servidor web puede restringir las páginas a una red privada, por ejemplo, una intranet, o puede publicar las páginas en la Red Mundial. Las páginas web son solicitadas y transferidas de los servidores usando el Protocolo de Transferencia de Hipertexto (HTTP). La acción del Servidor CLIENTE de guardar la página web, se denomina "HOSTING". Las páginas web pueden consistir en archivos de texto estático, o se pueden leer una serie de archivos con código que instruya al servidor cómo construir el HTML para cada página que es solicitada, a esto se le conoce como Página Web Dinámica.

Las páginas estáticas, generalmente usan la extensión de archivo.htm o.html. Las páginas dinámicas, usan extensiones que generalmente reflejan el lenguaje o tecnología que se utilizó para crear el código, como.php (PHP), .jsp (JavaServer), etc. En estos casos, el servidor debe estar configurado para esperar y entender estas tecnologías.

Las páginas web dinámicas, son aquellas que pueden acceder a bases de datos para extraer información que pueda ser presentada al visitante dependiendo de determinados criterios. Ejemplo de esto son páginas que tienen sistemas de administración de contenido o CMS. Estos sistemas permiten cambiar el contenido de la página web sin tener que utilizar un programa de ftp para subir los cambios. Existen diversos lenguajes de programación, que permiten agregar dinamismo a una página web, tal es el caso de ASP, PHP, JSP y varios más.

Elementos de una página Web.

Una página web, tiene contenido que puede ser visto o escuchado por el usuario final. Estos elementos incluyen, pero no exclusivamente:

- Texto. El texto editable se muestra en pantalla con alguna de las fuentes que el usuario tiene instaladas.

- Imágenes. Son ficheros enlazados desde el fichero de la página propiamente dicho. Se puede hablar de tres formatos casi exclusivamente: GIF, JPG y PNG.
- Audio, generalmente en MIDI, WAV y MP3.
- Adobe Flash.
- Adobe Shockwave.
- Gráficas Vectoriales (SVG - Scalable Vector Graphics).
- Hipervínculos, Vínculos y Marcadores.

La página web, también puede traer contenido que es interpretado de forma diferente dependiendo del navegador y generalmente no es mostrado al usuario final. Estos elementos incluyen, pero no exclusivamente:

- Scripts, generalmente JavaScript.
- Meta tags.
- Hojas de Estilo (CSS - Cascading Style Sheets).

# 2

## Descripción del sistema propuesto

Como se ha mencionado al inicio de la documentación, la domótica es el resultado de la integración de los sistemas de gestión de seguridad, comunicaciones, gestión del confort y control energético. Estudiando las diferentes aplicaciones prácticas y características del PIC32 en especial la del módulo Ethernet, y tomando en cuenta la importancia que tiene hoy en día el Internet en la vida diaria del ser humano, se ha pensado en diseñar e implementar un sistema que integre estas tecnologías en una aplicación útil, que permita el monitoreo y control de las luminarias de una vivienda en forma remota.

Para conseguir desarrollar este proyecto, es necesario recurrir a la modularidad y dividir el sistema total en diferentes sistemas parciales, tales como: interfaz de usuario local, interfaces de control, interfaz de usuario remoto, sistema inmerso, módulos TX/RX e interfaz de usuario. Con estos módulos conseguimos simplificar el trabajo, los cuales serán explicados detalladamente a lo largo de todo este trabajo.

Como se puede observar en el diagrama de bloques que muestra en este capítulo, el sistema consta de seis partes fundamentales para su correcto funcionamiento, que son:

- Interfaz de usuario local
- Interfaces de control.
- Interfaz de usuario remoto.
- Sistema inmerso.
- Módulos TX/RX.
- Interfaz de usuario.

Para la computadora personal los requerimientos son mínimos, es suficiente con que esta pueda conectarse y trabajar a través de la red de internet con facilidad. El usuario tendrá la comodidad de transmitir las instrucciones mediante la computador y estas viajarán a través de la red cumpliendo los protocolos del modelo TCP/IP y siguiendo los estándares que involucran toda una conectividad. Además, por la misma vía se podrá recibir los paquetes de datos con el estado del microcontrolador, de esta manera podremos controlar y monitorear nuestra aplicación, ya sea de manera remota o local. A continuación en la figura 2.1 se muestra el diagrama de bloques describiendo cada una de las etapas del sistema.

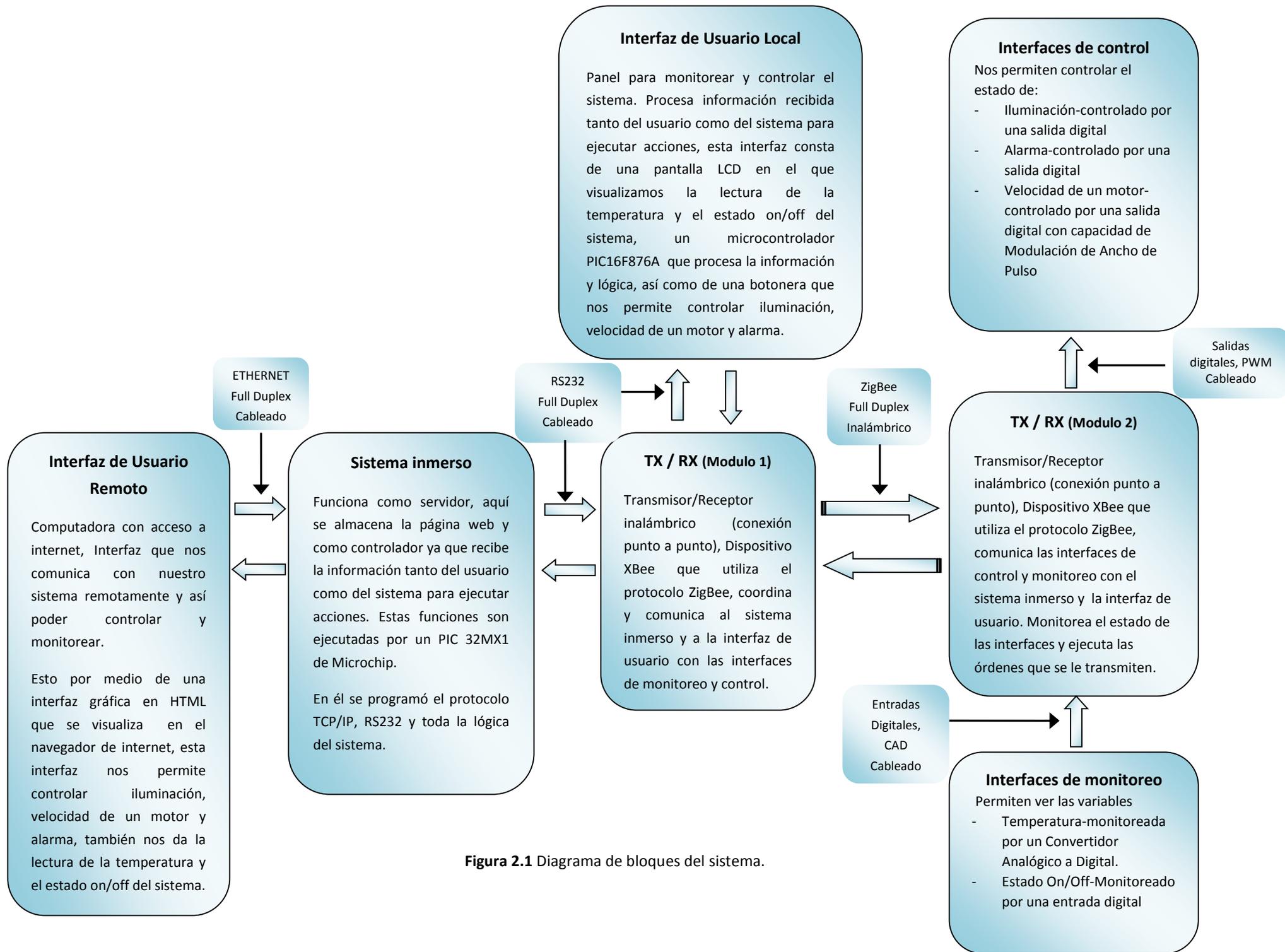


Figura 2.1 Diagrama de bloques del sistema.

---

## 2.1 PIC32 Ethernet Starter Kit

Hay muchos fabricantes de microcontroladores, los principales son: ST Microelectronics, Texas Instruments, Infineon Technologies, Intel Corp, Royal Philips Electronics, Analog Devices, National Semiconductors, Motorola Semiconductors, Toshiba, Freescale Semiconductor, Inc. Microchip, NEC Corp., Mitsubishi, Hitachi Corp y Atmel Corporation. Todos tienen sus ventajas e inconvenientes y se deben valorar varios factores desde prestaciones, relación calidad/precio, disponibilidad comercial y tipo de herramientas disponibles.

La compañía Microchip Technology Inc., presentó su nueva gama de Microcontroladores (MCU's) cuya principal característica es la integración del núcleo MIPS32 "M4K" de 72 MHz, 512 KB en memoria Flash y 32 KB en RAM.

De acuerdo a las declaraciones de Microchip Technology, su gama PIC32 se conforma de Microcontroladores de 64 y 100 pines que responden al desempeño antes señalado, además involucran una compatibilidad de funcionamiento con MCUs PIC de 8, 16 y 32 bits, debido a la flexibilidad y re-configuración de sus pines, herramientas de desarrollo y periféricos varios.

La gama PIC32 contiene el Entorno de Desarrollo Integrado (IDE), MPLAB de Microchip, el cual viene anexado en un kit de desarrollo y está encomendado para dar soporte a los nuevos Microcontroladores de la firma.

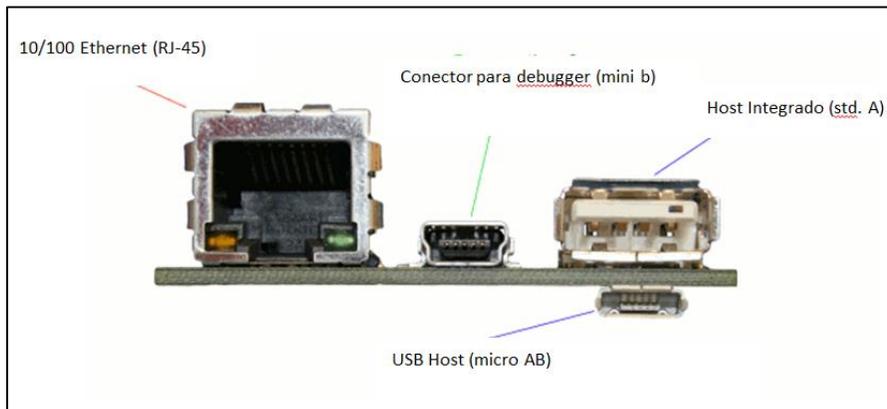
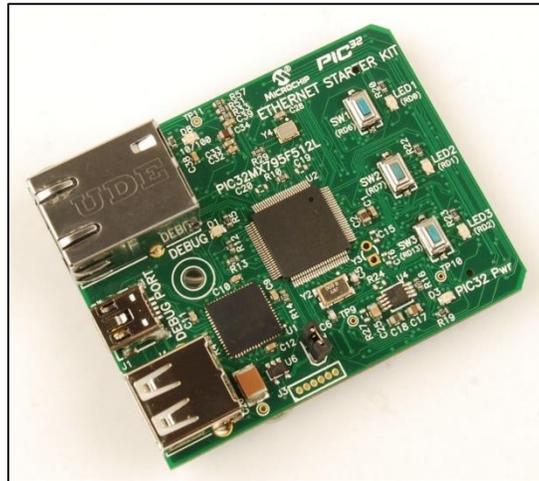
Estas nuevas familias MCU se complementan con las pilas de software libre de Microchip, disponible en forma de código fuente, por lo que es más fácil para los diseñadores de sistemas embebidos añadir conectividad a sus aplicaciones.

Entre los elementos que conforman los nuevos dispositivos se halla una variedad de periféricos destinados a las aplicaciones de comunicación y la adición de un Puerto Maestro Paralelo (PMP) de 16 bits con soporte a memoria y visualizadores. Asimismo el núcleo M4K se encuentra dentro del estándar de funcionamiento "MIPS32" y según precisan sus fabricantes, alcanza un "performance" optimizado, poseyendo la característica de bajo consumo de energía, amplio soporte de herramientas y respuesta de interrupción inmediata.

Bajo el protocolo industrial de ejecución MIPS32, el core de la familia PIC 32 registra un índice de procesamiento de 1.5 DMIPS/Mhz gracias al fraccionamiento en 5 episodios de su estructura y a la integración de una unidad denominada MIPS16e 16 bits ISA, la cual confiere la bondad de disminuir hasta un 40 por ciento el tamaño del código.

---

El PIC32 Ethernet Starter Kit ofrece el método de costo más bajo y más fácil de experimentar 10/100 Ethernet con el desarrollo PIC32. En combinación con Microchip gratis software TCP / IP , el proyecto se ejecutará en todo momento. El PIC32 tiene una disposición CAN 2.0b periféricos y USB host / dispositivo / OTG. Figura 2.2.



**Figura 2.2** PIC32 Ethernet Starter Kit. (Modelo DM320004).

El Ethernet Starter Kit tiene un factor de forma y el conector de expansión que son compatibles con otros PIC32 Starter Kits.

## 2.2 XBee

Para la implementación del proyecto se optó por módulos XBee compuestos por un microcontrolador el cual ya contiene el protocolo de comunicación ZigBee (IEEE 802.15.4) y la antena para transmitir.

XBee es el nombre comercial de Digi International para una familia de módulos compatibles para transmitir por RF. Los radios XBee primero fueron introducidos por la

---

marca MaxStream en 2005 y se basa en el estándar 802.15.4 -2003, estándar diseñado para conexiones punto-a-punto y comunicaciones punto a multipunto a través del aire a velocidades de 250kbps.

Las radios XBee se pueden usar con conexiones mínimas de alimentación (3,3 V), GND, DOUT y DIN (UART), con otras líneas que se recomienda Reset y sleep. Además, las familias más XBee algunas otras de control de flujo , I / O , A / D y las líneas indicador construido adentro una versión de la XBee llamado XBee programable tiene un procesador adicional a bordo para el código de usuario. El XBee programables y una nueva superficie de montaje (SMT), la versión de las radios XBee fueron introducidas en 2010.

El XBee puede funcionar en un modo transparente o modo API (Interfaz de Aplicación para programadores). Los datos en modo transparente entran en la DIN (pin del módulo de recepción de datos) y se transmiten directamente a las radios solo con la intención de recibir información, sin ninguna modificación. En modo API los datos se envuelven en una trama que permite configurar los parámetros del dispositivo y la retroalimentación de entrega de paquetes.



**Figura 2.3** Módulo XBee serie 1 de MaxStream.

Cada módulo tiene periféricos de entrada y salida digitales, convertidores analógicos digitales y viceversa que permiten la manipulación de dispositivos externos conectados a estos módulos con circuitería muy simple.

### **2.2.1 Configuración de módulos XBee**

Dentro del sistema se recurre a comunicarse con las interfaces y el controlador principal de manera inalámbrica, usando módulos XBee.

Al estarse utilizando dos dispositivos XBee, tenemos una red con conexión punto a punto, pero ya sea nuestro caso o donde se requiera conectar una red con más dispositivos,

existe la posibilidad de agregar a cada dispositivo XBee circuitería que nos ayude a monitorear y controlar el medio, así como de circuitos para interpretar la información que sea enviada o recibida.

Así que nuestra pequeña red consta de un Coordinator y un End Device, el Coordinator se encarga de manipular la información junto con el procesador principal que es el que tomara decisiones para ejecutar un proceso en un determinado momento, el End device envía el estado de cada sensor y también funciona como actuador ya que ejecutara las acciones que se le indiquen.

Para la manipulación de estos dispositivos el fabricante proporciona manuales y software de manera gratuita, el hardware es lo único que se compra que en este caso son los módulos XBee y el programador.

Para que los módulos XBee se puedan comunicar inalámbricamente hay que programarlos, en esta programación se definen características como el canal de transmisión, si se trata de un Coordinator o un End Device, la dirección de cada módulo para que se puedan identificar dentro de una red, configuración de pines en los que se establece cuál será su función, por ejemplo si van a funcionar como entradas o salidas analógicas o digitales etc.

Para la programación es necesario utilizar comandos especiales denominados comandos AT, por practicidad no mencionamos todos solo se enlistan en la siguiente tabla los más importantes y su función.

**Tabla 2.1** Comandos AT para programar módulo XBee (módems).

Comando AT	Rango	Descripción
<b>A1</b>	0-0x0F	Describe el modo de asociación de un módulo utilizado como dispositivo terminal (CE=0) por defecto
<b>A2</b>	0-0x0F	Describe el modo de asociación de un módulo utilizado como coordinador (CE=1)
<b>AC</b>	-	Aplica los cambios realizados explícitamente en la configuración
<b>AP</b>	0-0x02	Habilita el modo de operación API. Defecto =0 0     Modo API deshabilitado 1     Modo API habilitado 2     Modo API habilitado con carácter de
<b>BD</b>	0-0x07	Velocidad de transmisión 0     1200 1     2400

		2	4800
		3	9600
		4	19200
		5	38400
		6	57600
		7	115200
<b>CC</b>	0-0xFF	Establece el carácter de secuencia a ser usado entre tiempos de espera para entrar al modo al modo de comandos. Defecto 0x2B (carácter ASCII +)	
<b>CH</b>	0x0B-0x1A	Establece el canal por el cual se realiza la conexión RF entre módulos. Defecto 0x0C	
<b>CE</b>	0-1	Indica el comportamiento del modulo. Defecto =0	
<b>CN</b>	-	Sale del modo comando	
<b>D0-D4</b>	0-5	Ajusta la configuración de los pines I/O	
		0	deshabilitado
		1	-
		2	CAD
		3	Entrada digital
		4	Salida digital LOW
		5	Salida digital HIGH
<b>D5</b>	0-5	Las mismas funciones que D0 – D4 exceptuando lo siguiente:	
		1	Indicador de asociación
<b>D6</b>	0-5	Las mismas funciones que D0 – D4 exceptuando lo siguiente:	
		1	Control de flujo RTS
		2	No tiene conversor ADC
<b>D7</b>	0-5	Las mismas funciones que D0 – D4 exceptuando lo siguiente:	
		1	Control de flujo CTS
		2	No tiene conversor ADC
<b>D8</b>	Solo 0 y 3	Ajusta la configuración del pin DI-8 (pin 9)	
		0	Deshabilitado
		3	Entrada digital
<b>DB</b>	0x17-0x5C (x-1 dBm)	Lee la potencia de la señal del módulo del cual provino el último paquete RF recibido	
<b>DL</b>	0-0xFFFFFFFF	Ajusta los 32 bits más significativos para direccionamiento. Defecto =0	
<b>DH</b>	0-0xFFFFFFFF	Ajusta los 32 bits más significativos para direccionamiento. Defecto =0	
<b>GT</b>	2-0x0CE4 (x 1 ms)	Tiempo de espera antes y después de ingresar el carácter de secuencia para entrar al modo de comandos. Defecto 0 0x3E8	
<b>IA</b>	0-0xFFFFFFFF	Utilizado para crear cable virtual. Indica la dirección del	

	FFFFFFFF	módulo de origen de los datos. Defecto =0FFFFFFFFFFFFFFFF (no permite el recibo de ningún paquete para cambiar las salidas)
<b>ID</b>	0-0xFFFF	Ajusta la dirección PAN del módulo. Defecto = 0x3332
<b>IR</b>	0-0xFFFF (x 1 ms)	Ajusta la tasa de muestreo de los pines I/O.
<b>IS</b>	1-0xFF	Le pide al módulo a leer todos sus pines I/O.
<b>IT</b>	1-0xFF	Numero de muestras DIO y ADC que se deben esperar antes de transmitir. Defecto =1
<b>IO</b>	8 bits	Ajusta los niveles de salidas digitales. Cada bit representa el nivel de los pines I/O configurados como salida
<b>M0 – M1</b>	0-0x03FF	Ajusta el ciclo de trabajo de salida PWM0 y PWM1.
<b>MY</b>	0-0xFFFF	Configura la dirección de 16 bits para el módulo
<b>NB</b>	0-4	Ajusta la paridad para la comunicación serial UART del módulo. Defecto =0
<b>P0 – P1</b>	0-2	Configura el pin PWM0 y PWM1. Defecto P0=1, P1=0 0      Deshabilitado 1      RSSI 2      PWM habilitado
<b>RE</b>	-	Restaura los valores de los parámetros a los valores por defecto que vienen de fábrica.
<b>SM</b>	0-6	Configura el modo de operación SLEEP. Defecto =0 0      deshabilitado 1      Pin de hibernado 2      Pin doze 3      (reservado) 4      Remote cyclic SLEEP 5      Remote cyclic SLEEP (con pin Wake – Up) 6      SLEEP Coordinador
<b>SL</b>	0-0xFFFFFFFF	Entrega los 32 bits menos significativos del numero serial del modulo
<b>SH</b>	0-0xFFFFFFFF	Entrega los 32 bits más significativos del numero serial del modulo
<b>SP</b>	1 – 0x68B0 (x 10 ms)	Ajusta el tiempo de duración en que un módulo duerme o se mantienen el modo SLEEP.
<b>ST</b>	1 – FFFF (x 1 ms)	Ajusta el tiempo de inactividad (datos ni recibidos ni enviados ya sea por RF o serial) antes de que el modulo ingrese al modo SLEEP
<b>T0 – T7</b>	0-0xFF (x 100 ms)	
<b>IU</b>	0-1	Habilita o no la salida I/O UART 0      Deshabilitado. No se reciben paquetes

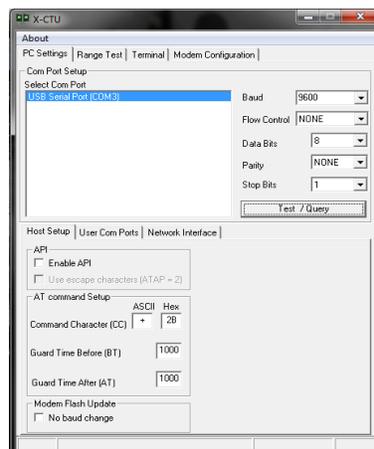
		RF por la UART 1 Habilitado. Se reciben paquetes de RF por la UART
<b>VR</b>	0-0xFFFF	Indica cual versión de firmware se encuentra actualmente en el módulo.
<b>WR</b>	-	Guarda en la memoria no volátil del módulo, todos los valores de los parámetros

También se usa un software específico para la programación de los XBee llamado X-CTU; este software nos permite manipular nuestros dispositivos de manera sencilla e intuitiva.

En la siguiente imagen se muestra la ventana principal de X-CTU en donde aplicamos configuraciones como velocidad de transmisión, si queremos un bit de control en nuestra trama, longitud de cada segmento de la trama, bit de paridad, bit de parada si queremos que funcione o no en modo API etc.

Este programa se divide en 4 pestañas:

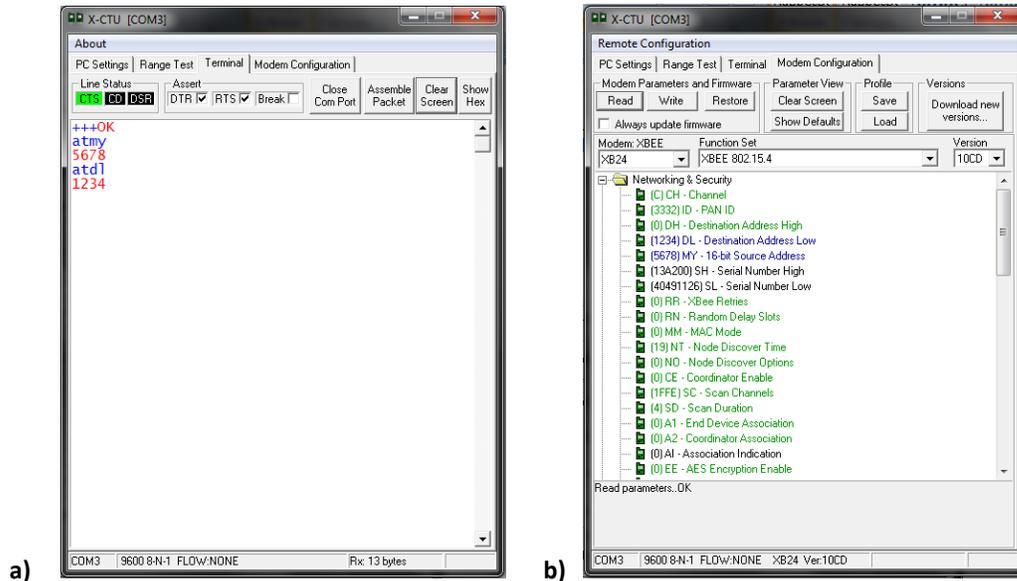
- PC Settings
- Range Test
- Terminal
- Modem configuration



**Figura 2.4** Programa X-CTU para simular y programar los dispositivos XBee.

En la pestaña Range Test, es posible enviar una cadena de datos de cualquier tipo para probar el rango de alcance de la señal. Este genera automáticamente datos y los envía por el módulo, de tal forma que permite verificar cuales datos llegan buenos y cuáles no y a partir de esa estadística determinar el rango o alcance de la señal.

Las pestañas más usadas serán la de Terminal, que es la que usamos para programar los XBee y la pestaña Modem Configuration que es aquí donde se observan los parámetros programados además de que también se puede programar de manera más intuitiva los dispositivos.



**Figura 2.5** a y b. a) Terminal, donde se lleva a cabo la programación; b) Modem configuration, aquí revisamos los parámetros y también se pueden programar los XBee.

## 2.3 Interfaz de usuario

La interfaz principal para manipular nuestro sistema es la computadora, sin embargo también es necesaria una interfaz que nos permita controlar por otro medio ya que no siempre estará disponible la computadora, para ello se diseñó una interfaz de usuario que este siempre disponible para manipular el sistema, consiste en una pantalla LCD y un microcontrolador.

### 2.3.1 Pantalla LCD JHD 162 A

La pantalla es el medio por el cual se visualizara la información del sistema, necesitamos que sea clara, de bajo consumo de energía y de fácil operación por ello optamos por una pantalla LCD JHD 162 que es muy popular.

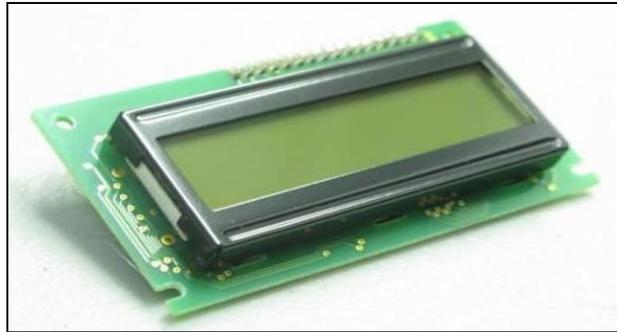
Las pantallas de cristal líquido (LCD) tienen la capacidad de mostrar cualquier carácter alfanumérico permitiendo representar la información que genera cualquier dispositivo electrónico de una forma fácil y económica.

Independientemente de la marca de LCD, la mayoría traen internamente un microprocesador Hitachi 44780 que se encarga de mostrar los caracteres.

---

## Características

- Contenido de display 16 caracteres \* 2 líneas
- Cada carácter compuesto de 5 columnas \* 8 filas
- Alimentación 5 Volts
- Muestreo de 400Hz
- Contraste ajustable



**Figura 2.6** Pantalla LCD JHD 162A.

### **2.3.2 Microcontrolador PIC16F876A**

Un microcontrolador es un circuito integrado que incluye en su interior las tres unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada y salida pero su aplicación es para procesos específicos el microcontrolador que ocuparemos deberá cumplir con ciertos requerimientos.

Cabe mencionar que un microcontrolador debe ser programado con ciertas instrucciones las cuales le dirán al que acciones debe ejecutar y en qué momento.

Para el manejo del display y poder interpretar los datos del sistema en nuestra interfaz de usuario es necesario un microcontrolador con periféricos de entrada y salida, velocidad de procesamiento rápido, con interrupciones y UART ya que el sistema nos entrega los datos de forma serial.

*PIC16F876A.*

Seleccionamos un microcontrolador de Microchip ya que cumple con los requisitos anteriores, esta familia de PIC's (Peripheral Interface Controller).

## Características

- Programación con solo 35 instrucciones
- Velocidad de funcionamiento: 20 Mhz de reloj de entrada DC-200 ns ciclo de instrucción.
- Hasta 8K x 14 palabras de memoria de programa Flash, hasta 368 x 8 bytes de memoria de datos (RAM) Hasta 256 x 8 bytes de memoria EEPROM de datos.
- Universal Synchronous receptor asíncrono Transmisor (USART/SCI) con detección de 9 bits
- Amplio rango de tensión: 2.0 V a 5.5 V.

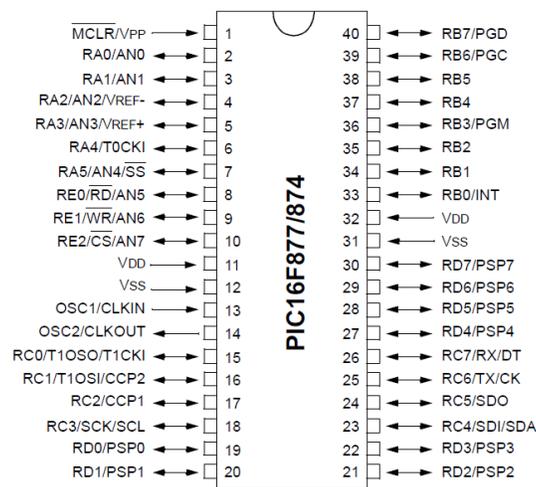


Figura 2.7 Microcontrolador PIC16f876A de Microchip Technology Inc.

## 2.4 Sensores

Son dispositivos diseñados para recibir información de una magnitud del exterior y transformarla en otra magnitud, normalmente eléctrica, que seamos capaces de cuantificar y manipular.

Normalmente estos dispositivos se encuentran realizados mediante la utilización de componentes pasivos (resistencias variables, PTC, NTC, LDR, etc... todos aquellos componentes que varían su magnitud en función de alguna variable), y la utilización de componentes activos.



---

Este se puede utilizar con fuentes de alimentación individuales, cabe mencionar que a solo 60 mA de operación tiene muy bajo auto calentamiento, menor a 0,1 C en aire en calma. El LM35 está diseñado y probado para funcionar en un rango de temperatura de -55 °C a 150 °C, esta serie de integrado viene en distintos tipos de encapsulado pudiendo tener muy diversas aplicaciones y variadas formas de montaje.

Características:

- Calibrados directamente en °C (grados centígrados)
- Salida lineal + 10,0 mV/°C, factor de escala
- 0,5 °C exactitud (a +25 °C)
- Diseñado para un correcto funcionamiento a un rango de -55 °C a +150 °C
- Adecuados para aplicaciones remotas
- De bajo costo debido a ajuste a nivel de oblea
- Opera desde 4 a 30 voltios
- Menos de 60 uA de corriente de drenaje
- bajo calentamiento espontáneo, 0,08 °C en aire inmóvil
- No lineal en sólo  $\pm 1/4$  °C típico
- De baja impedancia de salida, 0,1 ohm con una carga de 1 mA

#### **2.4.2 Sensor de gas**

Un componente muy importante respecto al monitoreo y sobre todo en seguridad es el sensor de gas que nos permitirá saber si tenemos alguna fuga o que por algún descuido se dejó escapar el gas, pero surge la pregunta que sensor de gas es el adecuado, primeramente el sensor debe cumplir con ciertas características.

Tiene que tener la capacidad de detectar solo el gas butano y propano que es el que comúnmente se usa en casa y que no lo vaya a confundir con otro tipo de gases que hay en el ambiente, que sea sensible y que tenga una respuesta rápida ya que al tratarse de un sensor de seguridad necesitamos la mayor rapidez para alertar al usuario de que hay una anomalía en nuestro hogar.

Después de revisar varios sensores se optó por uno en especial que cumple con estas características y que relativamente es económico ya que este tipo de sensores son de alto costo por la complejidad de los mismos, el TGS 2610 del que ahora procederemos a mencionar sus características.

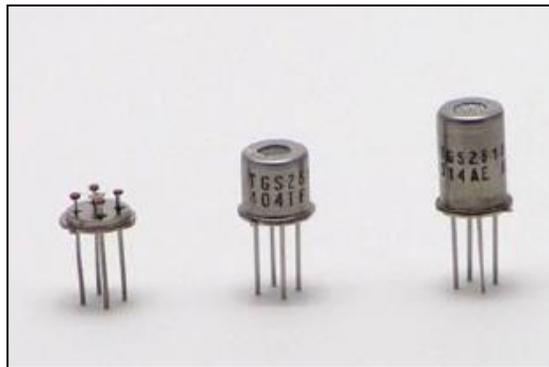
---

### *Elemento activo TGS 2610*

TGS2610 es un tipo de sensor de gas semiconductor que combina la alta sensibilidad al gas LP con bajo consumo de energía y larga vida. Debido a la miniaturización de los chips de detección, TGS2610 requiere una corriente del calentador de sólo 56mA y el dispositivo se encuentra en un nivel A-5 del paquete.

El TGS2610 está disponible en dos modelos diferentes que tienen diferentes carcasas externas, sino la sensibilidad idéntica a la del gas LP. Ambos modelos son capaces de satisfacer los requisitos de las normas de rendimiento, como UL1484 y EN50194.

TGS2610-C00 posee la pequeña y rápida respuesta con el gas, por lo que es conveniente para las alarma de fugas de gas.



**Figura 2.9** Sensor de Gas TGS 2610 y sus diferentes encapsulados.

TGS2610-D00 utiliza material de filtro en su vivienda, que elimina la influencia de los gases de interferencia, tales como el alcohol, que provoca una respuesta muy selectiva a gas LP. Esta característica hace que el sensor ideal para zonas residenciales detectores de fugas de gas que requieren durabilidad y resistencia a la interferencia de gas.

#### Características:

- Bajo consumo de energía
- Alta sensibilidad a LP y sus gases componentes (por ejemplo, propano y butano)
- Larga vida y bajo costo
- Utilización de circuitos electrónicos simples

#### Aplicaciones:

- Para detectores residenciales de gas LP de fugas y alarmas
- Para detectores de LP portátiles

- 
- Gas LP y de detección de vapores

## 2.5 Actuadores

Los actuadores son elementos que utiliza el sistema para modificar el estado de ciertos equipos e instalaciones de manera física como válvulas, interruptores, llaves de paso, chapas eléctricas, iluminación, etc.

En su mayoría son componentes que demandan potencia y por lo tanto es necesario aparte del actuador un sistema de potencia que controle al dispositivo.

### 2.5.1 Motor de CD

El motor de corriente continua es una máquina que convierte la energía eléctrica en mecánica, principalmente mediante el movimiento rotatorio.

En este caso usaremos uno de estos motores, al cual añadiremos circuitería para poder variar la velocidad de giro por medio de nuestro sistema. El motor simula el giro de diversos aparatos como una licuadora o un ventilador.

#### *Motor AMRF-300*

Su aplicación se enfoca a reproductores de CD, DVD por su diseño no hace ruido y está totalmente sellado.

- Motor de CC
- Motores Peso: aproximadamente 22g
- Diámetro: 24,4 mm
- Salida: aproximadamente: 0,3 W ~ 0,8 W, dependiendo voltaje
- Uso típico: Reproductor de CD / DVD, CD / DVD ROM



**Figura 2.10** Motor AMRF-300 de CC.

---

### 2.5.2 Lámpara Fluorescente Compacta

Un dispositivo muy común en el hogar es el uso de lámparas y por ello también es necesario poder controlarlas, hoy en día en su mayoría se usan lámparas ahorradoras de luz por ello usaremos un CFL.

La lámpara compacta fluorescente o CFL (sigla del inglés *compact fluorescent lamp*) es un tipo de lámpara fluorescente que se puede usar con casquillos de rosca Edison normal (E27) o pequeña (E14).

En comparación con las lámparas incandescentes, las CFL tienen una vida útil mayor y consumen menos energía eléctrica para producir la misma iluminación. De hecho, las lámparas CFL ayudan a ahorrar costes en facturas de electricidad, en compensación a su alto precio dentro de las primeras 500 horas de uso.



Figura 2.11 Lámpara fluorescente compacta de 25 Watts.

### 2.5.3 Alarma

Dentro de un sistema domótico es fundamental la seguridad así que implementamos una alarma que va a ligada a los sensores descritos, para que cuando haya una irregularidad detectada por los sensores dentro de nuestra casa el sistema actúe y pueda prevenir un accidente pero que también nos pueda avisar y podamos tomar las acciones necesarias.

Esta alarma se compone de un zumbador y un módulo LED para que el aviso sea tanto auditivo como visual.

#### *Zumbador*

Zumbador, buzzer en inglés, es un transductor electro acústico que produce un sonido o zumbido continuo o intermitente de un mismo tono. Sirve como mecanismo de señalización o aviso, y son utilizados en múltiples sistemas como en automóviles o en electrodomésticos.

---

### Zumbador BD3398

Se decidió usar un zumbador piezoeléctrico ya que no consumen mucha corriente y su sonido es lo suficientemente alto para ser alertados, sus características son las siguientes:

- Tensión de trabajo: 3 – 24 VDC
- Nivel de sonido mínimo: 90 dB a 30 cm
- Frecuencia de resonancia: 3 KHz
- Corriente máxima 10 mA
- Tipo de sonido continuo



**Figura 2.12** Zumbador piezoeléctrico BD3398.

### Módulo LED

El módulo LED consiste de 4 LEDs de alta luminancia en color azul conectados entre sí en una placa metálica de alta eficiencia energética

- Angulo de apertura 140 Grados
- Módulo de 4 LEDs
- Voltaje de operación: 12 Volts
- Uso tanto para interiores como exteriores
- Consumo mW/mA: 444/37
- Flujo luminoso. 2 a 3 lm
- Dimensiones: 35X35 mm



**Figura 2.13** Módulo LED color azul Modelo H32L32.

# 3

## Diseño de software

Este capítulo contiene el desarrollo que se realizó para la programación de la tarjeta microcontroladora (PIC32 Ethernet), así como los diagramas de flujo de cada uno de los programas, indicando el diseño e implementación necesaria para que la página web pueda ser almacenada en el microcontrolador y comunicarse con las interfaces de monitoreo y control. También se muestra la configuración e instrucciones necesarias para el funcionamiento de los módulos XBee.

### 3.1 Programación del Microcontrolador de 32 bits

Para la programación del Microcontrolador, se ha tomado como referencia la versión de programación TCP/IP que facilita el fabricante de forma libre. Para que el programa cumpla una de las normas de la programación (todo sistema debe ser estructurado), se ha dividido el código en funciones que serán utilizadas para el adecuado funcionamiento de nuestra aplicación.

A continuación se hará una descripción de los pasos que se realizaron para la programación del PIC32 y se mostrará su respectivo diagrama de flujo, se empieza por declarar e inicializar las variables locales, luego se procede a inicializar la tarjeta controladora haciendo una llamada a la función "InitializeBoard". Continuando con la configuración, se inicializa el tiempo (timer 0), la variable global "AppConfig", se inicializa la función para la página web; llamando a las funciones "TickInit", "Inicializar\_AppConfig" y "MPFSInit" respectivamente. En este punto realizamos una rutina para permitir que el usuario pueda borrar el contenido de la EEPROM, con un simple procedimiento que detallamos a continuación:

1. Mantener presionado el botón RB3.
2. Presionar y soltar el botón MCLR.
3. Continuar presionando el botón RB3, por aproximadamente 4 segundos.
4. Dejar de presionar el botón RB3.
5. Presionar y soltar nuevamente el botón MCLR, para restaurar la aplicación.

Como se observa en la figura 3.1, se muestra el procedimiento de la función principal "Main".

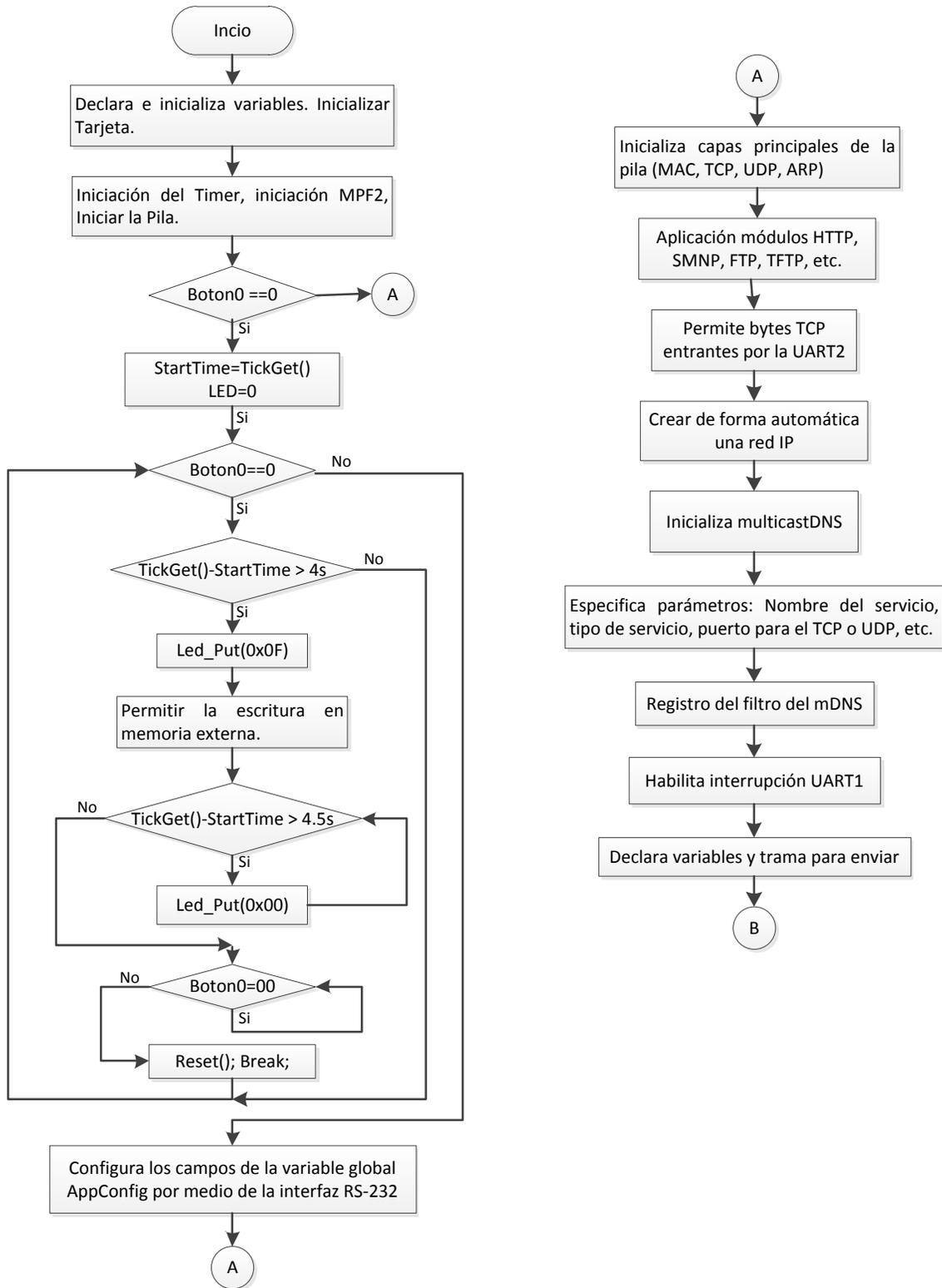


Figura 3.1 Diagrama de flujo de la función del Main.

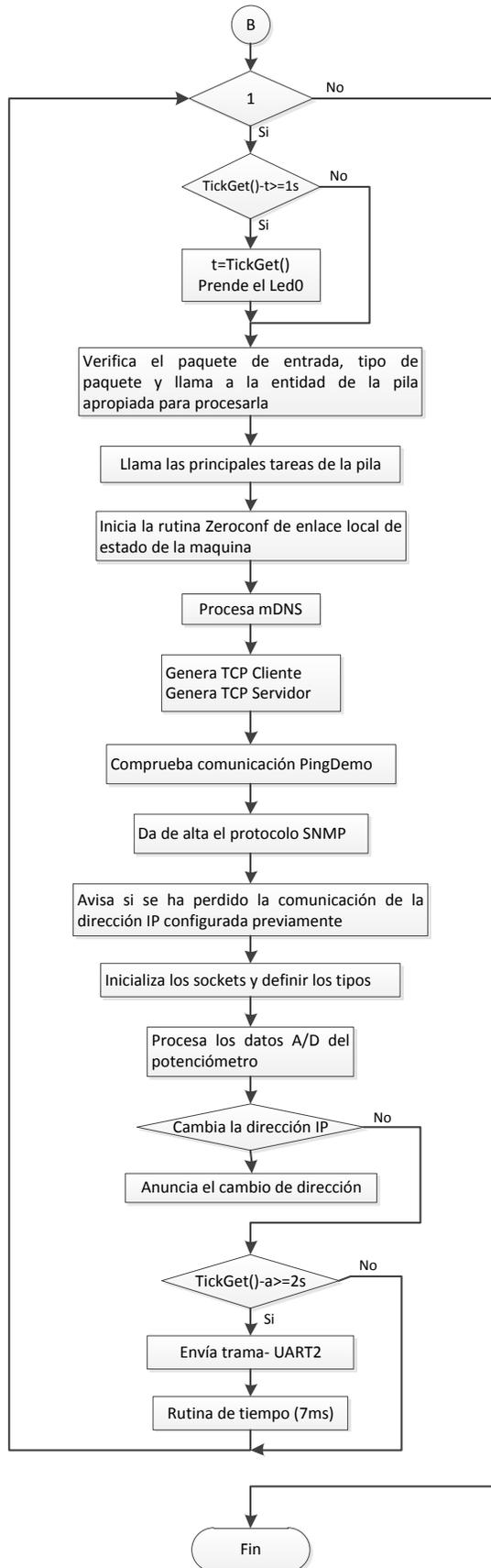


Figura 3.1 Diagrama de flujo de la función del Main (continuación).

En este mismo diagrama de flujo se puede observar que si se repite los pasos 1 y 2, pero se mantiene presionado el botón RB3 menos de 4 segundos, se procede a llamar a la función “DoUARTConfig”.

Una vez explicada esta rutina, el programa continúa con la inicialización del stack, del módulo EUSART y del protocolo HTTP, llamando a las funciones “StackInit”, “UART2TCPBridgeInit” y “HTTPInit” consecutivamente; también se inicializa el multicastDNS, se definen sus parámetros y se habilita la interrupción de la UART1 por medio de las funciones “mDNSInitialize”, “mDNSServiceRegister” y “ConfigIntUART1”.

En esta parte del programa tenemos un lazo infinito, dentro de este lazo se llaman a las diferentes funciones que ejecutan tareas específicas que hacen que nuestra aplicación funcione correctamente. Estas funciones las explicamos a continuación:

Tabla 2.1 Funciones TCP/IP Microchip.

Funciones TCP/IP Pila Microchip.	Descripción.
<b>StackTask</b>	Maneja el paquete Rx interno para preprocesarlo previamente antes de ser utilizado por las capas superiores. Esta función checa los nuevos paquetes entrantes y los guía hacia los componentes apropiados de la pila.
<b>UART2TCPBridgeTask</b>	Transmite todos los bytes TCP entrantes en una ranura o conexión exterior del módulo USART.
<b>HTTPServer</b>	Servidor de páginas dinámicas para los navegadores web tales como el Explorador de Internet de Microsoft, Mozilla Firefox, etc.
<b>DiscoveryTask</b>	Examina los paquetes de broadcast para saber si se trata de una petición de descubrimiento, luego comenzamos el envío de nuestra dirección MAC en formato legible (convertimos la dirección MAC de bytes a hexadecimal), al último nodo de donde recibimos la petición.
<b>NBNSTask</b>	

	<p>Responde a la petición de nombre NBNS para permitir la asignación de un nombre (dado por el usuario), a la tarjeta. Es decir, permite a los nodos de la misma subred IP usar un nombre de cliente en lugar de una dirección IP para acceder a nuestra tarjeta.</p>
<p><b>DHCPSTask</b></p>	<p>Proporciona una dirección IP automática, máscara de subred, dirección de puerta de enlace, la dirección del servidor DNS, y otros parámetros de configuración DHCP utilizados en redes.</p>
<p><b>GenericTCPServer</b></p>	<p>Esta función implementa un servidor TCP simple. La función se invoca periódicamente por nuestra aplicación para escuchar las conexiones entrantes. Cuando una conexión es realizada, el servidor lee todos los datos que ingresan, lo transforma a mayúsculas, y este proceso se repite de regreso. Esto debe ser usado como una base para crear nuevas aplicaciones de servidor TCP.</p>
<p><b>TelnetTask</b></p>	<p>Proporciona el servicio de Telnet sobre el puerto TCP 23.</p>
<p><b>RebootTask</b></p>	<p>Permite reiniciar remotamente el PIC. Esta función revisa el tráfico entrante en el puerto 30304 y reinicia el PIC si una "R" es recibida.</p>
<p><b>SNTPCient</b></p>	<p>Esta función localiza un servidor NTP (Protocolo de Internet utilizado para sincronizar los relojes de los ordenadores a un tiempo de referencia), de un lugar público usando DNS. Pide el tiempo UTC (estándar oficial para el tiempo actual), usando Sntp y actualiza periódicamente la estructura SntpTime.</p>

Adicionalmente, en dicho lazo se tiene una rutina la cual sirve para actualizar la dirección IP en el momento que el usuario habilite el modo DHCP, una vez realizado este paso, cada

dos segundos envía una trama al componente Xbee para saber el estado de las variables a controlar y monitorear para después entrar a una rutina de retardo de 7 ms para esperar la respuesta de dicho componente

En la figura 3.2, se muestra la secuencia de la función una vez que ha ocurrido la interrupción en la UART1.

Se empieza por declarar e inicializar las variables para comenzar a recibir el primer byte y hacer la comparación si el dato es igual a 7EH, si esta comparación es verdadera se lee la trama que es mandada por el componente XBEE, una vez que se termina la transmisión de datos se obtiene la temperatura y se comparan los bytes recibidos para conocer el estado de las variables y así poder reportarlos a la página web almacenada en el microcontrolador y regresar al Main. Si la comparación es falsa inicializa la trama y regresa al Main.

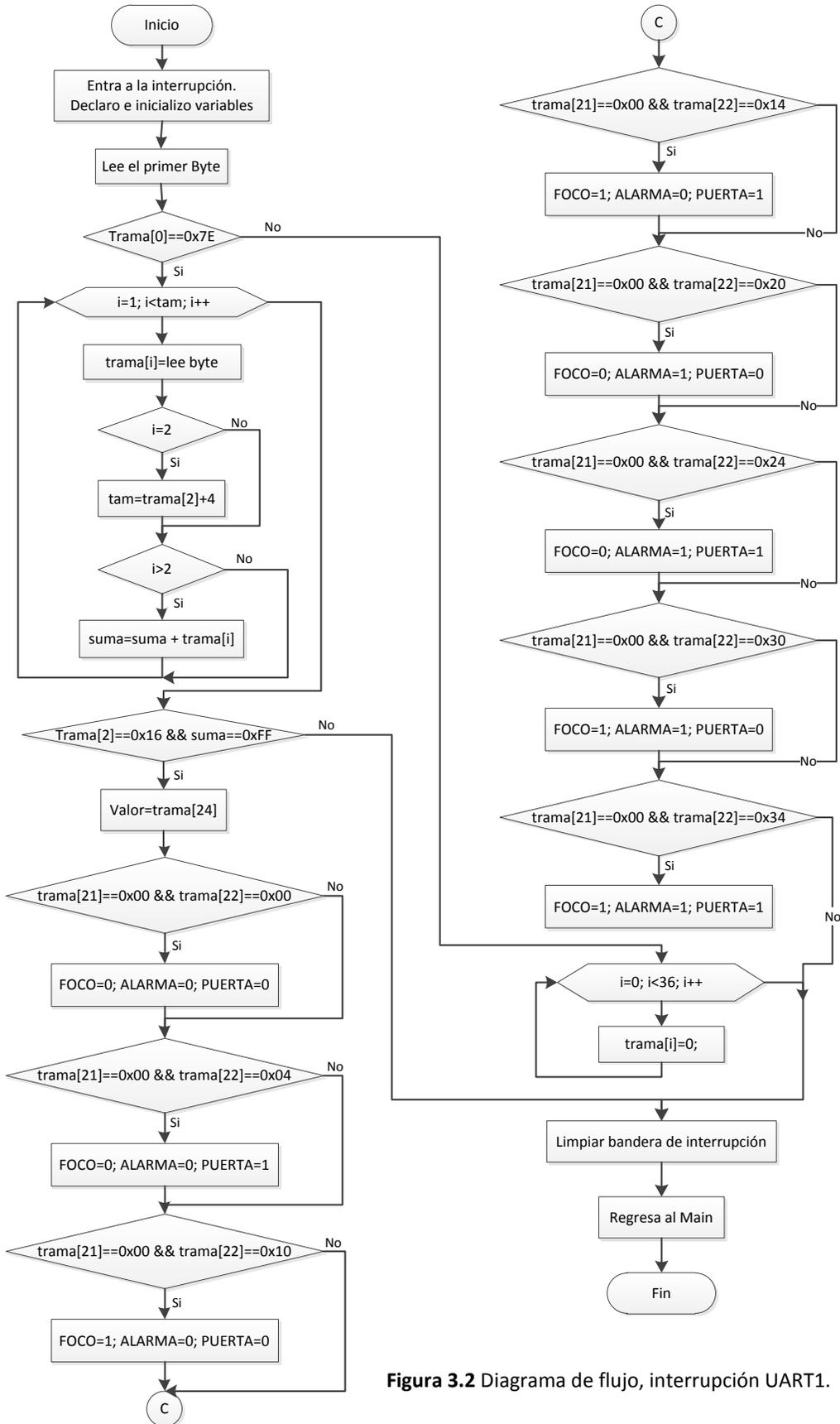
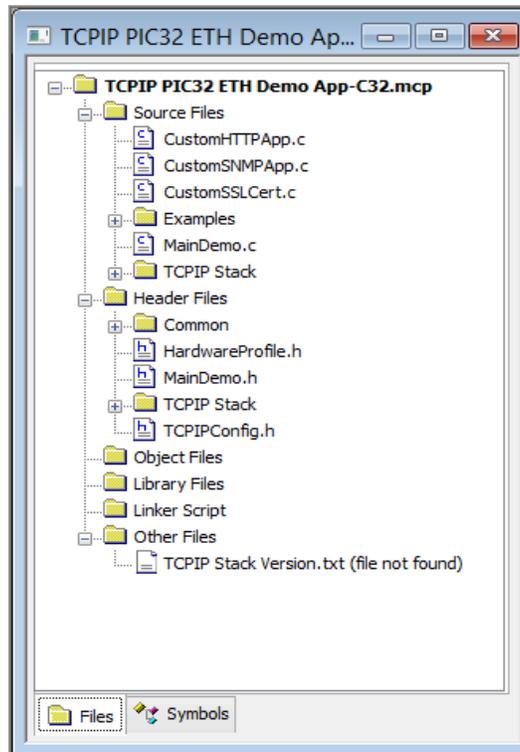


Figura 3.2 Diagrama de flujo, interrupción UART1.

### 3.2 Archivos principales de la Pila

En la figura 3.3 vemos la estructura general del TCP/IP Stack en el MPLAB. Se distingue la estructura dividida en archivos de cabecera (.h) y el código en los archivos (.c).



**Figura 3.3** Estructura de archivos de la pila.

Los archivos que es necesario conocer y que permiten personalizar la aplicación son los siguientes:

1. Mapear e inicializar el puerto en el micro. (HardwareProfile.h, MainDemo.c)
  - a. HardwareProfile.h. En este archivo simplemente se asocia un puerto con un nombre, por ejemplo si se coloca un LED en RD0 (pin 72 del microcontrolador):
 

```
#define LED0_TRIS (TRISDbits.TRISD0)
#define LED0_IO (LATDbits.LATD0)
```
  - b. MainDemo.c. Se definirá a RD0 como salida e inicializarlo en cero. Este proceso está fuera del loop principal, por lo que solo se va a ejecutar al reiniciar el microcontrolador.
 

```
static void InitializeBoard(void)
{
```

```
    LED0_TRIS = 0;
    LED_PUT(0x00);
```

```

    .
    .
}

```

2. Crear una función Callback. (CustomHTTPApp.c).

- a. La función callback contiene el nombre y el valor de la variable dinámica (led) que va a ser mostrada en la web. Un error muy común es enviar directamente al buffer Ethernet el estado del led mediante la variable int LED\_IO. Hay que tener en cuenta que solo se pueden strings por Ethernet, por lo que en numerosas ocasiones usamos itoa o ftoa (entero a ASCII y float a ASCII) para convertir los tipos de datos.

En este caso, en lugar de usar los conversores de tipo, se crean dos variables auxiliares ESTADO\_LED\_UP y ESTADO\_LED\_DOWN para mostrar el estado del led mediante ON, OFF.

```

ROM BYTE ESTADO_LED_UP[] = "ON";
ROM BYTE ESTADO_LED_DOWN[] = "OFF";

```

- b. La función void HTTPPrint\_led envía a la página web el valor de la variable dinámica LED. Para verificar el estado del puerto RDO (donde está mapeado el LED), se utiliza la sentencia condicional IF. Anteriormente se definió LED\_IO como estado del puerto RDO. El IF revisa el estado de LED\_IO. Si el puerto está en 1, entonces la sentencia es verdadera y envía, por HTTP, ESTADO\_LED\_UP; en cambio sí está en cero, la sentencia es falsa y envía ESTADO\_LED\_DOWN. De esta forma, tenemos un sistema que reporta el estado de las variables ante un llamado del servidor HTTP.

```

void HTTPPrint_led(void)
{
    TCPPutROMString(sktHTTP, (LED_IO?ESTADO_LED_UP:ESTADO_LED_DOWN));
}

```

3. MainDemo.c: Es el programa principal en el que encontramos el mapeo de puertos, la configuración de los registros de la USART, el ADC (Conversor Analógico Digital), la interfaz SPI y la inicialización de la placa.

### 3.3 Cabeceras

Debido a que el programa necesita una serie de funciones que se llaman de manera automática unas a otras, para cumplir con una tarea específica, se utilizan cabeceras o .h. Las cabeceras, contienen única y exclusivamente los prototipos de las funciones y en algunos casos contienen las estructuras de datos que ya hemos mencionado. En esta

sección se han puesto los archivos principales .h que se utilizan en el código para un mejor manejo del mismo.

**Tabla 3.2** Archivos .h TCP/IP.

Archivo .h	Descripción.
<b>ARP.h</b>	Funciones que permiten encontrar la dirección hardware o MAC correspondientes a una determinada dirección IP.
<b>Delay.h</b>	Declaraciones que establecen retardos específicos, que son de utilidad para el buen desempeño del programa.
<b>DHCP.h</b>	Funciones que permite a las estaciones de trabajo de una red IP obtener sus parámetros de configuración de manera automática.
<b>FTP.h</b>	Funciones que permiten la comunicación tipo FTP entre el usuario y el Microcontrolador.
<b>GenericTCPServer.h</b>	Función que utiliza el stack periódicamente para escuchar las conexiones entrantes.
<b>HTTP2.h</b>	Funciones y estructuras que permiten utilizar de forma adecuada el protocolo HTTP.
<b>ICMP.h</b>	Definiciones que se establecen para el control y notificación de errores del protocolo IP.
<b>IP.h</b>	Estructuras y funciones del protocolo IP, usadas por el programa, para la comunicación de datos a través de una red de paquetes conmutados.
<b>MAC.h</b>	Estas funciones proporcionan acceso al controlador Ethernet del PIC32MX795F512L.
<b>MPFS.h</b>	Funciones utilizadas para acceder a la página web, y a otros archivos de la memoria de programas internos, y de una memoria serial EEPROM externa.
<b>Random.h</b>	Función que permite reiniciar el microcontrolador de forma remota.

<b>SNTP.h</b>	Definiciones que actualizan el tiempo interno del programa cada 10 minutos usando la dirección “pool.ntp.org”, que es un servidor de tiempo de red.
<b>StackTsk.h</b>	Funciones que permiten manejar los paquetes de recepción realizando un pre procesamiento para luego ser enviados a las capas superiores.
<b>TCP.h</b>	Funciones que proporcionan confiabilidad, ofreciendo transporte en la transferencia de aplicaciones orientadas a datos con control de flujo.
<b>TCPIP.h</b>	En esta cabecera se encuentran declaradas todos los “.h” del programa, permitiendo acceder a todos las cabeceras mediante una sola llamada.
<b>TCPPerformance.h</b>	Funciones que realizan únicamente pruebas, para establecer una conexión, enviando imitaciones de paquetes de la memoria ROM.
<b>Telnet.h</b>	Declaraciones que establecen los servicios de comunicación de telnet.
<b>TFTPC.h</b>	Funciones que establecen servicios poco confiables de carga y descarga de archivos a otras aplicaciones.
<b>Tick.h</b>	Funciones útiles para la administración del cronómetro interno del microcontrolador, así como la inicialización de los Relojes (Timers), conversión a escala del tiempo, entre otras.
<b>UART2TCPBridge.h</b>	Declaraciones útiles que permiten controlar y configurar los periféricos UART para esta aplicación.
<b>UDP.h</b>	Funciones que establecen un transporte de paquetes de forma rápida pero de poca confiabilidad.

### 3.4 Programación de la página web

En la figura 3.4 se muestra la estructura que llevará la página web, indicando cada una de las páginas que tendrá el archivo, el cual será almacenado en la memoria del programa del microcontrolador.

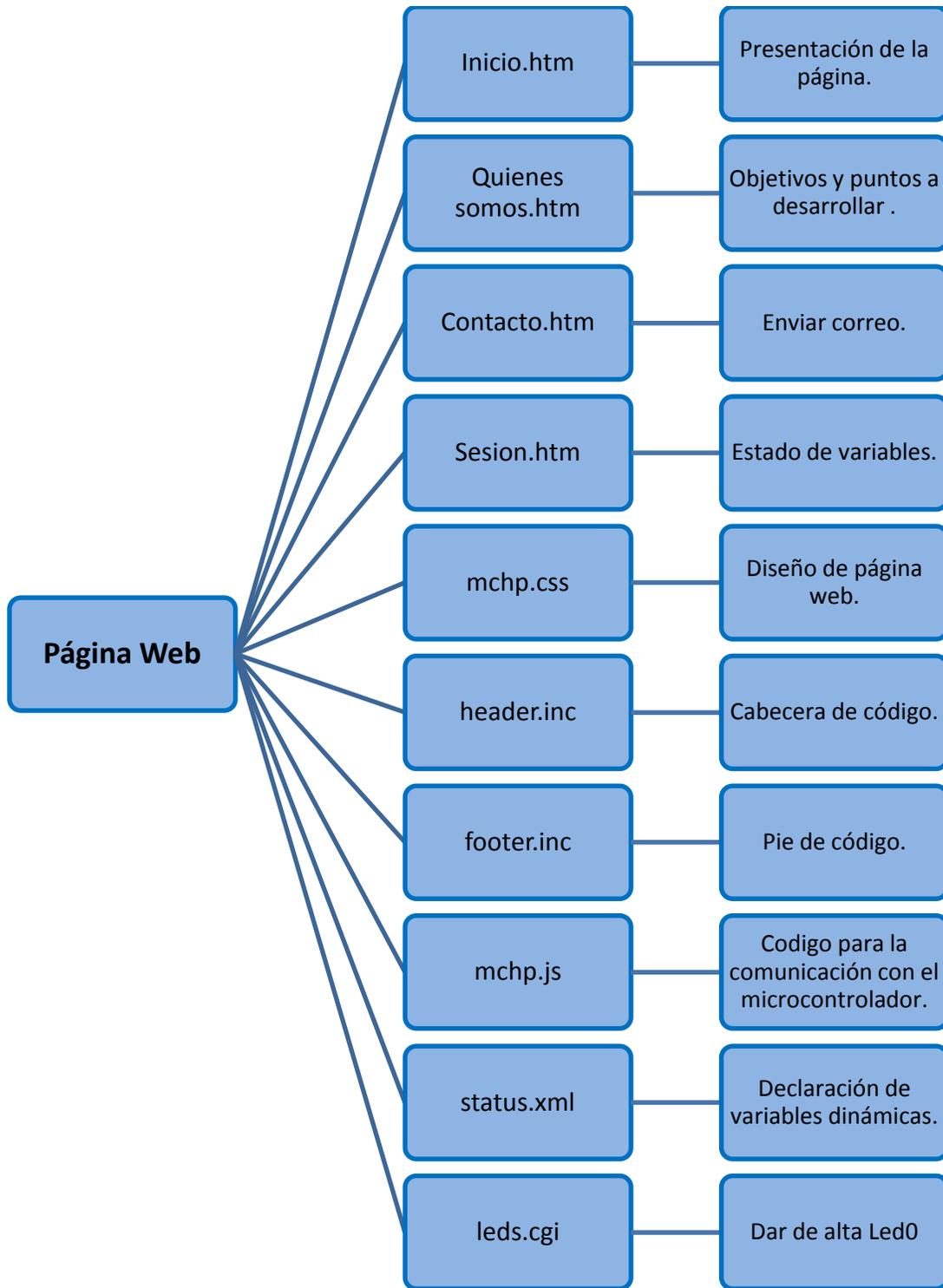
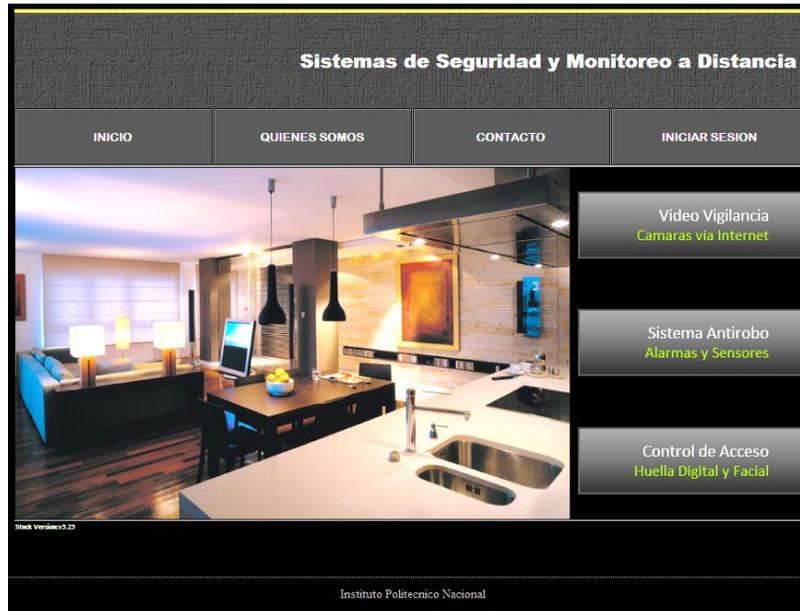


Figura 3.4 Estructura de la página web.

### 3.4.1 Diseño de la página web del microcontrolador

El diseño de la página web del microcontrolador que será grabada en la memoria interna del programa consta de las siguientes partes:

- Hipervínculos. Al utilizar solo una página web, haremos un hipervínculo entre ellas para pasar de una a otra sin ningún problema. Figura 3.5.



a) Página de inicio.



b) Página de contacto.

Figura 3.5 Hipervínculos dentro de la página web.

- Objetivos y puntos a desarrollar del Proyecto: Debido a que es una aplicación académica, se decidió nombrar los objetivos del proyecto para verificar su

cumplimiento y el buen funcionamiento del mismo como se observa en la Figura 3.6.



**Figura 3.6** Objetivo y Puntos a desarrollar.

- Estado de las variables a controlar y monitorear: En esta página se muestra el estado de la iluminación, gas, temperatura, alarma y el control de velocidades de un ventilador figura 3.8. En la sección de iluminación se podrá controlar el encendido o apagado según sea el caso, y de igual manera la página mostrará si el foco ha sido prendido o apagado de manera local por medio de la botonera. En el apartado del Estado de Gas, únicamente se monitorea si hay o no fuga de gas, si la hay el botón de la alarma se encenderá de color verde indicando el peligro. En la sección temperatura se despliega el valor de la temperatura que se obtiene a través del sensor, si el valor recibido es mayor a 50°C el botón de la alarma se encenderá de color verde indicando el peligro. El botón de la alarma controla y monitorea su estado, esta puede ser encendida o apagada de manera local (por medio de la botonera) o de manera remota (por la página web). Y por último en la sección del ventilador se muestran cinco botones de los cuales el primero se utiliza para prender o apagar el ventilador, y los otros cuatro para variar la velocidad del ventilador. En la parte superior derecha debajo del menú se encuentra un pequeño círculo representando un led que prende y apaga indicando que la conexión vía Ethernet se está realizando de manera correcta con el microcontrolador.

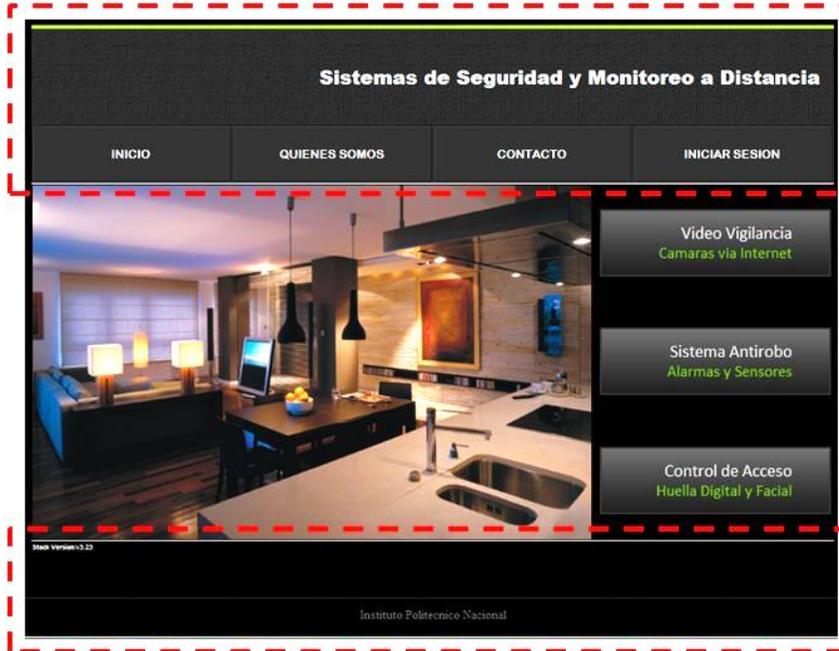


Figura 3.7 Estado de las variables a controlar y monitorear.

Microchip organiza los archivos del sitio web de tal forma de ahorrar espacio (anexo A figura A.3). Se trata de no repetir la información, por ejemplo, al inicio de cada página se encuentra el menú el cual es el mismo para cada una de las páginas del proyecto, por lo tanto para reducir el tamaño del archivo se crea un archivo header.inc que se manda a llamar en cada página (anexo A figura A.1).

Lo mismo ocurre con el pie de todas las páginas aparece la misma leyenda, entonces, se utiliza el archivo footer.inc que será llamado desde cada página, en lugar de escribir el texto nuevamente (anexo A figura A.2). Todo esto permite reducir el tamaño total del sitio.

Si ejecutamos los cambios que se han realizado en la página web, se obtendrá el siguiente resultado de manera gráfica, figura 3.8.



Encabezado de la página web (menú, generado en el archivo header.inc)

Pie de contenido de la página web (generado en el archivo footer.inc)

**Figura 3.8** Encabezado y pie de la página web de manera gráfica.

Una vez que se hacen los ajustes necesarios para que la página web sea de menor tamaño y quede espacio suficiente en la memoria del programa se prosigue a realizar el código necesario para poder tener la comunicación entre la tarjeta PIC32 Ethernet y la página web. Para monitorear el estado de un puerto desde la web, es necesario hacer lo siguiente:

Insertar una variable dinámica en la página web. (status.xml, index.html). Los archivos que deberán modificarse en la página web son los siguientes.

- a. status.xml. Contiene todas la variables dinámicas que se actualizan (leds, temperatura, botones, etc.). Se agrega la línea correspondiente a la nueva variable dinámica:

```
<led>~led~/led>
```

Entre <> se indica el nombre del divisor que se va a utilizar en el index para definir la posición de la variable.

- b. Index.html. Finalmente es el archivo que ve el usuario. Donde aparezca <~led~> el webserver lo va a remplazar con ON, OFF según el valor de LED\_IO.

```
<span id="led">?</span>;
```

En la función javascript function pdateStatus(xmlData) que viene incluida en este mismo archivo se debe indicar el color de los botones que se le presentaran al usuario por medio de un ciclo for como se ve en las siguientes líneas.

```
for (i = 0; i < 8; i++) {  
    document.getElementById('led' + i).style.color =  
        (getXMLValue(xmlData, 'led' + i) == '1') ? '#090' : '#ddd';  
}
```

Es importante destacar que este ejemplo utiliza AJAX para actualizar la información sin tener que apretar F5 en el explorador. Por lo tanto debemos agregar la siguiente sentencia al final del index:

```
Document.getElementById('led').innerHTML=getXMLValue(xmlData,'led');
```

Para la variable que muestra el estado de gas, la página web recibe un uno o un cero por lo tanto la lógica de programación es la misma, el único cambio que se realiza es en el archivo Index.html en la sección de diseño y en la función JavaScript function pdateStatus(xmlData).

- Diseño. En donde aparece el signo “?” será remplazado por la frase “FUGA DE GAS” o “SIN FUGA”, esto se debe a que el id se llama gas y en nuestra función JavaScript también se ha nombrado así.

```
<legend style="color:#5FB404"><b>Estado de Gas.</b></legend>  
<p><span id="gas" style="font-family:Calibri; font-size:14pt; color:white">  
<a style="text-align:center">?</a></span></p>
```

- Función JavaScript function pdateStatus(xmlData). En la función se realiza la evaluación para saber que letrero es el que debe mostrar, como se muestra en el siguiente código.

```
document.getElementById('gas').innerHTML = (getXMLValue(xmlData, 'gas') == '1') ?  
'FUGA DE GAS' : 'SIN FUGA DE GAS';
```

Finalmente para la variable de la temperatura únicamente se recibe el número y se despliega en la pantalla, por lo tanto, el código es muy similar a la variable que monitorea el gas en cuestión de diseño, únicamente se necesita cambiar el nombre del id, en este caso será “id=temperatura”.

En la función JavaScript cambia el código con respecto a la función para monitorear el gas debido a que para reportar la temperatura no es necesario realizar ninguna evaluación, ya

que esa acción la realiza el microcontrolador y no la página web, por lo tanto el código quedará de la siguiente manera:

```
document.getElementById('temp').innerHTML = getXMLValue(xmlData, 'temp');
```

Esta modificación se deberá realizar para cada una de las variables que se desee monitorear, en este caso, para este proyecto se declararan 9 variables:

- Iluminación.
- Estado de Gas.
- Temperatura.
- Alarma.
- Motor.
  - ON/OFF.
  - 25%
  - 50%
  - 75%
  - 100%

#### **3.4.2 Comprimiendo páginas web con MPFS**

La utilidad MPFS2 tiene muchas características. El propósito principal es empaquetar páginas web en un formato para el almacenamiento eficiente en un sistema embebido. Esto también adhiere variables dinámicas y genera el archivo HTTPPrint.h para asegurar que las llamadas a las mismas se realicen el número necesario de veces.

Debemos tener en cuenta que cada variable que agreguemos en la web requiere su correspondiente variable en el Stack, por lo tanto, es necesario recompilar el Stack y cargarlo en el PIC antes de compactarlo a la página y transferirla a la memoria. La secuencia correcta es la siguiente:

1. Modificar el Stack según las necesidades del proyecto requeridas.
2. Agregar las variables dinámicas correspondientes en status.xml e index.html.
3. Compactar la web con MPFS.
4. Compilar el Stack con el MPLAB.
5. Cargar el Stack en el PIC.
6. Transferir la página a la memoria externa.

Es necesario respetar este procedimiento porque, si las variables de la web no corresponden con las del Stack, la página no se cargará en el explorador.

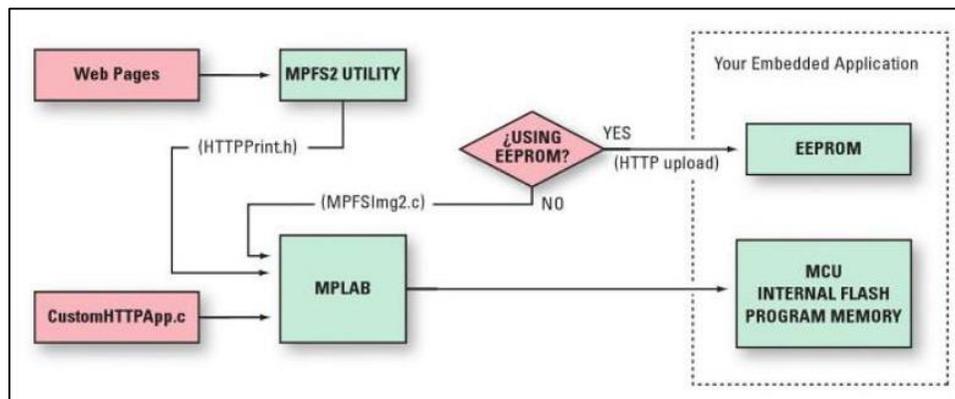
Lo que hace internamente este conversor es juntar todos los archivos en uno solo, de hecho, podemos abrir el archivo resultante con el bloc de notas y veremos, ordenado secuencialmente, el contenido de cada archivo que compone el website.

La aplicación MPFS.exe convierte el conjunto de archivos que componen la página web a dos formatos posibles:

- .bin (para cargar el website en la memoria EEPROM externa).
- .c (para cargar el website en la memoria del programa del PIC).

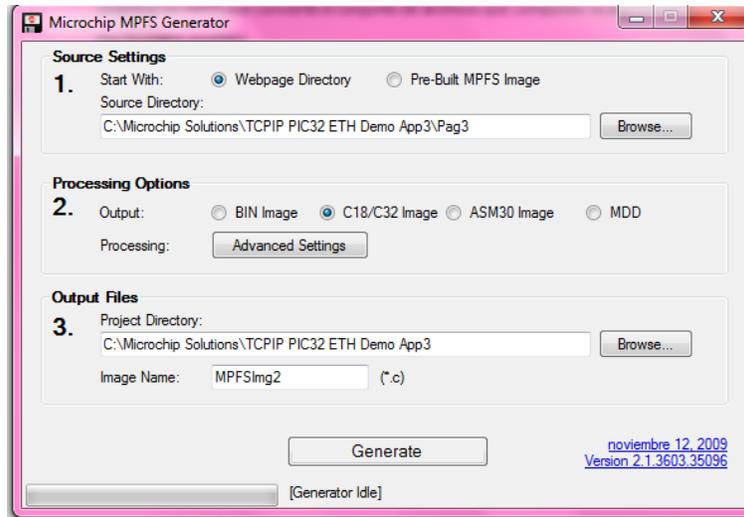
La utilidad MPFS tiene tres pasos, denotados en el margen izquierdo de la ventana de diálogo, figura 3.10. Para construir una imagen MPFS, seleccionamos “Comenzar con: Directorio de Página Web” en el paso uno y seleccionamos el directorio en que la página se almacenará.

El segundo paso, selecciona el formato de salida. Si el almacenamiento es en una EEPROM, seleccionamos “BIN Image” como formato de salida. Si utilizamos la memoria interna de programa, indicamos “C18/C32 Image” para usar con secciones de 8 bits, y “ASM 30 Image” para aplicaciones de 16 bits (que es nuestro caso) figura 3.9.



**Figura 3.9** En este diagrama se muestran las distintas opciones para almacenar la página.

El tercer paso, controla las características de carga. La opción para cargar la página web seleccionada estará disponible.



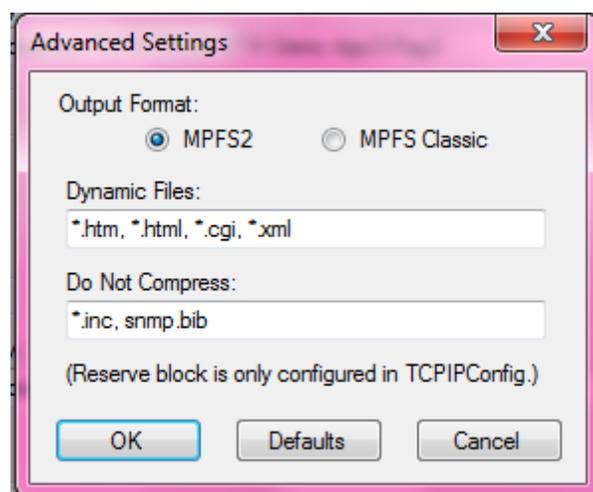
**Figura 3.10** Pantalla principal del MPFS.exe con las opciones necesarias para compactar la página web y cargarla en la memoria.

Si se está siendo usada la memoria interna del programa, la imagen será compilada con el proyecto y debido a eso las cargas no estarán disponibles. Hay que asegurarse que la salida del archivo fuente indicado, está incluida en el proyecto.

Una vez listo todo, se selecciona el botón “Generate” para crear la imagen y cargarla en la tarjeta destino.

Características Avanzadas de Configuración.

Las características avanzadas de procesamiento encontradas en el segundo paso, nos proveen de mayor control sobre el procesamiento de archivos figura 3.11.



**Figura 3.11** Ventana de Opciones de Procesamiento.

El formato de salida permite seleccionar entre los formatos MPFS2 y MPFS clásico. El servidor HTTP2 es solamente compatible con MPFS2. Los tipos de archivo HTML indican los tipos de archivos a ser analizados para variables dinámicas. Por defecto, todos los archivos con extensión htm, html, cgi o xml son analizados. Si una aplicación contiene variables dinámicas en otros tipos de archivo, estos tipos deben ser añadidos a la lista.

En el apartado “Do Not Compress” indica que tipos de archivos nunca deberían ser comprimidos. Los archivos con variables dinámicas serán excluidos automáticamente. Los archivos incluidos vía `~inc:filename~` no deberían ser comprimidos, no se debería usar ningún archivo BIB para el módulo SNMP. Los tipos de archivo adicionales pueden ser añadidos a las lista si alguna aplicación accede a MPFS.

#### **3.5 Programación de la interfaz de usuario local**

Esta interfaz es parte fundamental del sistema ya que permite controlar y monitorear localmente las interfaces también se encarga de comunicar los XBee y el PIC32 por ello necesita ser programado con un algoritmo que le diga que hacer y también cómo hacer lo que el usuario le pida. En esta interfaz se programa lo que desplegara el LCD, la función que debe realizar cada uno de los botones al ser oprimidos y al variar el potenciómetro, activar o desactivar alguna interfaz, recibir y enviar tramas.

Para ello se programa el microcontrolador PIC16F876A con un software especializado, el compilador PCWHD Compiler o más conocido como CCS, básicamente la función de este programa es traducir un programa escrito en lenguaje de programación de alto nivel a otro lenguaje de programación equivalente para que una maquina o en este caso el PIC pueda interpretar.

Para programar se usa el lenguaje C, con el que se realiza el algoritmo que se necesita para el PIC, este programa se escribe en lenguaje C, se compilara para traducir nuestro programa a un lenguaje que podrá interpretar nuestro PIC pero también nos ayudara a encontrar algunos errores o advertencias en nuestro código para corregirlos, por ultimo al tener lista esta traducción la pasaremos al PIC con un programador y después probaremos físicamente que funcione nuestro código. Programaremos y compilaremos tantas veces sea necesario hasta obtener los resultados deseados.



**Tabla 3.3** Rango de valores asignados a cada condicional if para la comparación con el valor del CAD, así como la función de cada botón al ser oprimido.

Botón	Rango de valores asignados a cada if	Función del botón
1	5-50	Funcion no definida (futura aplicación)
2	51-100	Salir de rutina para controlar la velocidad del motor
3	101-150	Entrar a rutina para controlar la velocidad del motor
4	151-200	Encender/Apagar la alarma
5	201-250	Encender/Apagar el foco

Como se mencionaba toda la información enviada y recibida de las interfaces pasa por los puertos digitales y analógicos del XBee End Device el cual se comunica inalámbricamente con el XBee Coordinator, después pasa la información del XBee Coordinator por su UART a la UART del PIC, para que haya una correcta comunicación se programa la UART con la siguiente línea de código para trabajar en modo estándar:

```
#use rs232 (baud=9600, xmit=PIN_C6, rcv=PIN_C7)
```

La siguiente sección del programa en la interrupción de la UART que se programó para que la UART del PIC siempre esté atento a la información que le envíe el XBee y así sin importar que esté ejecutando el PIC en el momento que el XBee mande información, el PIC deje de ejecutar lo que esté haciendo y de prioridad a la información que está recibiendo y así siempre tener el estado de las interfaces actualizado.

El código de la lectura de la trama transmitida por el XBee funciona de la siguiente manera:

Cuando ocurre la interrupción en la UART el PIC inmediatamente entrara en la función *serial\_isr()* y ejecutara el código que se muestra abajo, este código guarda cada byte en un arreglo que llamamos *cbuff*, al finalizar la trama se revisara el byte Checksum para saber si la trama llego correctamente de ser así entrara a la función *procesa\_comando()* que procesara la cadena y mostrara la información en la pantalla LCD y así saber el estado de las interfaces.

Las tramas enviadas como se menciona básicamente son 4 pero de estas se derivan otras para controlar en su totalidad el sistema, ver tabla 3.4.

Estas tramas son transmitidas de la UART del PIC a la UART del XBee Coordinator con la siguiente expresión:

```
printf("%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c%c",126,0,16,23,18,0,19,162,0,64,73,6,251,255,254,2,73,79,48,208);
```

La trama que vemos es un ejemplo, con esta trama encendemos el foco, vemos que es una simple función de impresión de texto que maneja el lenguaje C, las tramas siempre se manejan en forma hexadecimal, en la función la trama se pone en decimal pero la expresión %c cambia el formato a hexadecimal para que el XBee pueda interpretarlo. Así que cada que oprimimos un botón se enviara una trama distinta dependiendo de lo que se requiera.

La pantalla LCD se inicializada con programación, el compilador CCS tiene una librería específica para controlar el display llamada flex\_lcd.c, únicamente es necesario usar las funciones para desplegar texto como se ve en el código de abajo

```
lcd_init(); // Inicializo el LCD
printf(lcd_putc," -Ilum:%c -Gas:%c",f,p); //Escribo en la primera fila
lcd_gotoxy(0,2); //Columna 0, Fila 2
printf(lcd_putc,"-Alar:%c -Temp:%ldG",a,t); //Escribo en la segunda fila
```

La función *lcd\_init()* inicializa el display, la función *printf()* tiene como argumento *lcd\_putc* que imprime texto en la pantalla junto con variables de estado del sistema y *lcd\_gotoxy()* ubica el texto en la fila y columna que se especifique.

**Tabla 3.4** Tramas enviadas al XBee Coordinator al oprimir un botón. Tramas enviadas con la función *printf()*.

Función	Trama hexadecimal
Estado de pines	7E 00 0F 17 12 00 13 A2 00 40 49 06 FB FF FE 00 49 53 FE
Foco On	7E 00 10 17 12 00 13 A2 00 40 49 06 FB FF FE 02 49 4F 10 F0
Alarma On	7E 00 10 17 12 00 13 A2 00 40 49 06 FB FF FE 02 49 4F 20 E0
Foco/Alarma On	7E 00 10 17 12 00 13 A2 00 40 49 06 FB FF FE 02 49 4F 30 D0
Foco/Alarma Off	7E 00 10 17 12 00 13 A2 00 40 49 06 FB FF FE 02 49 4F 00 00
PWM	7E 00 11 17 12 00 13 A2 00 40 49 06 FB FF FE 02 4D 30 00 FF 1C Valores que puede tomar el byte #19 y #20: 00 00 -a- 03 FF El byte #21 cambiara de acuerdo a la regla del Checksum

A continuación se muestran dos diagramas de flujo del programa en el PIC, uno del Main y otro de la Interrupción de la UART.

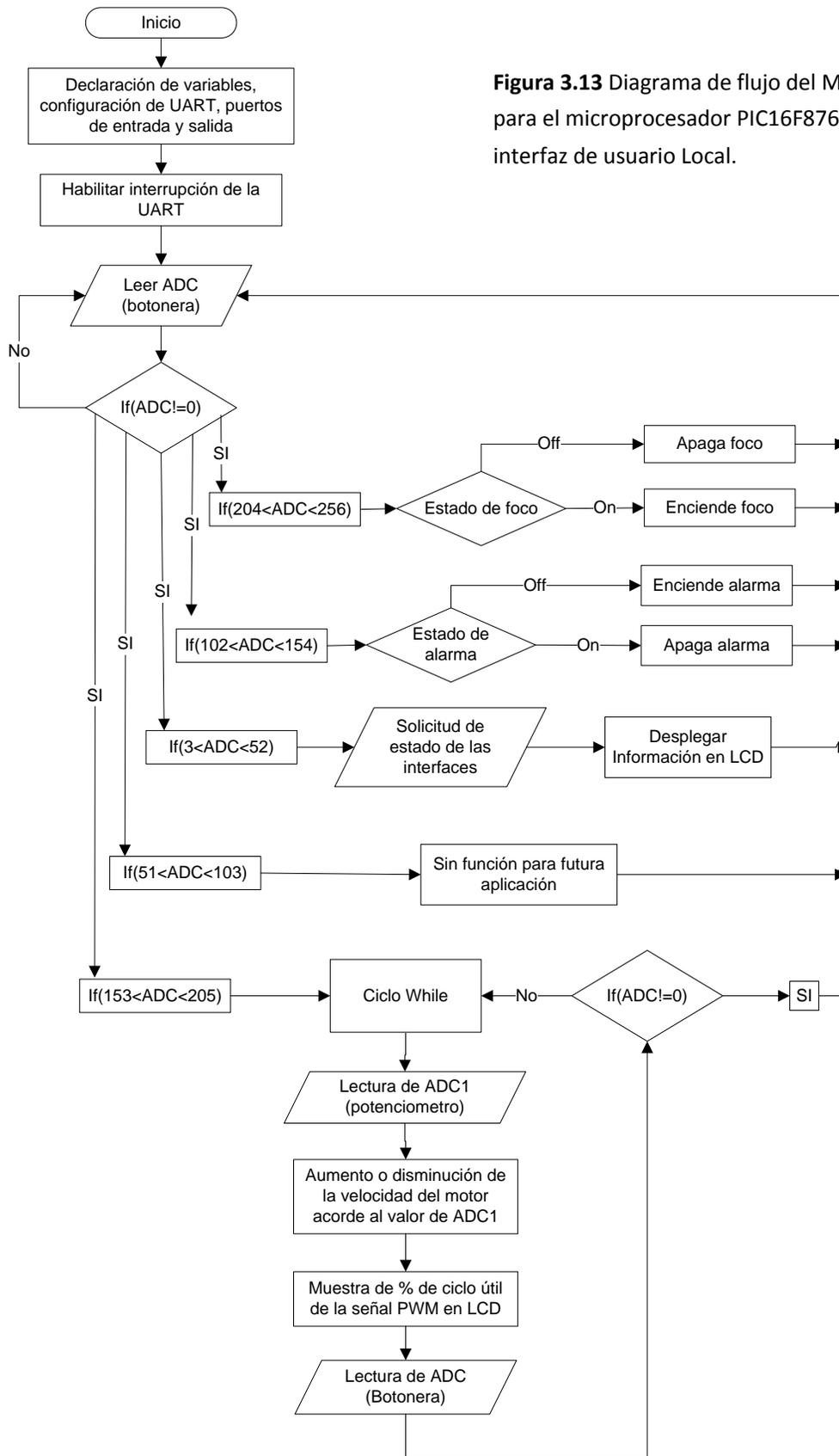
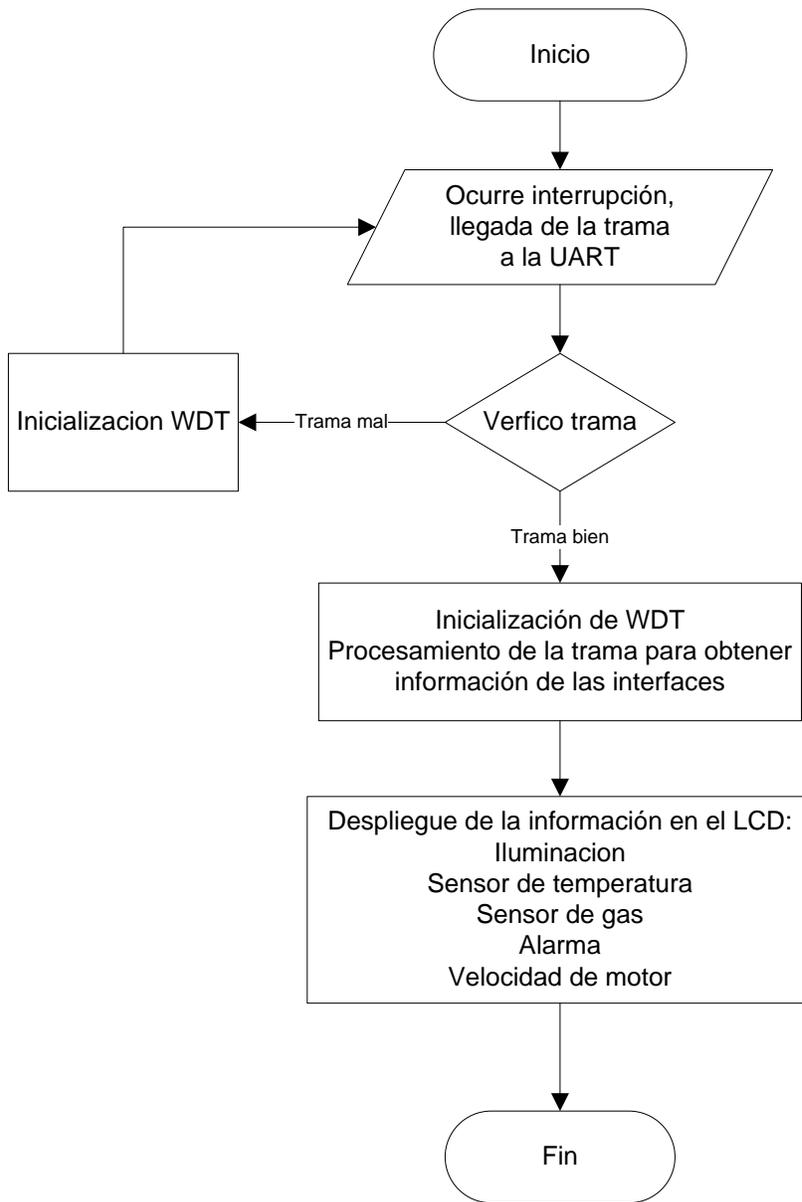


Figura 3.13 Diagrama de flujo del Main, programa para el microprocesador PIC16F876A para la interfaz de usuario Local.



**Figura 3.14** Diagrama de flujo de la interrupción de la UART, programa para el microprocesador PIC16F876A para la interfaz de usuario Local.

### 3.6 Programación de módulos XBee

La programación de los XBee se basa en asignar un valor en específico para cada comando AT dependiendo de la función que va a desempeñar cada módulo, a continuación se explica más a detalle qué función tiene cada comando AT y el respectivo valor que le fue asignado.

Programación para que exista comunicación entre los XBee:

El valor del comando, canal de transmisión (CH): 0C; este valor se selecciona arbitrariamente, en nuestro caso como el valor es C, nuestros dispositivos estarán trabajando a una frecuencia central 2.41 GHz, cabe mencionar que al seleccionar este valor todos los demás dispositivos deberán contener el mismo canal de no ser así simplemente no habrá comunicación de los XBee.

**Tabla 3.5** Frecuencias de los 16 canales que pueden manejar los XBee.

Frecuencia (GHz)					
Canal	Hexadecimal	Lim. Inferior	Central	Lim. Superior	Comando AT
11	0B	2.4025	2.4050	2.4075	ATCH0B
12	0C	2.4075	2.4100	2.4125	ATCH0C
13	0D	2.4125	2.4150	2.4175	ATCH0D
14	0E	2.4175	2.4200	2.4225	ATCH0E
15	0F	2.4225	2.4250	2.4275	ATCH0F
16	10	2.4275	2.4300	2.4325	ATCH10
17	11	2.4325	2.4350	2.4375	ATCH11
18	12	2.4375	2.4400	2.4425	ATCH12
19	13	2.4425	2.4450	2.4475	ATCH13
20	14	2.4475	2.4500	2.4525	ATCH14
21	15	2.4525	2.4550	2.4575	ATCH15
22	16	2.4575	2.4600	2.4625	ATCH16
23	17	2.4625	2.4650	2.4675	ATCH17
24	18	2.4675	2.4700	2.4725	ATCH18
25	19	2.4725	2.4750	2.4775	ATCH19
26	1A	2.4775	2.4800	2.4825	ATCH1A

Después seleccionamos un número de PAN ID, este valor permite identificar toda la red, es decir permite que la información enviada por un dispositivo sea recibida por todos los dispositivos que tengan este mismo número de ID.

En este caso se selecciona el valor que trae configurado por default 3332, el valor de este parámetro puede ser arbitrario siempre y cuando este en el rango (véase tabla de parámetros AT)

Posteriormente se asignan los valores para los parámetros de direccionamiento de los XBee DH, DL y MY; DH tiene el valor de 0 ya que este se usa en lo que llamamos direccionamiento de 64 bits esto es para conectar los módulos mediante su número de serie pre asignado de fábrica, en nuestro caso solo usaremos el direccionamiento de 16 bits, al usar un transmisor y un receptor se dice que establecemos una conexión punto a punto así que para los comandos DL y MY si es importante asignar un valor, por lo tanto el valor de MY del receptor que está funcionando como Coordinator se asigna a DL del transmisor que funciona como End Device y viceversa el valor de MY del End Device se asigna a DL del Coordinator para establecer la comunicación entre ellos.

Los comandos SH y SL son únicos y vienen pregrabados de fábrica de igual manera identifican al módulo al activar el direccionamiento de 64 bits que en este caso no usaremos.

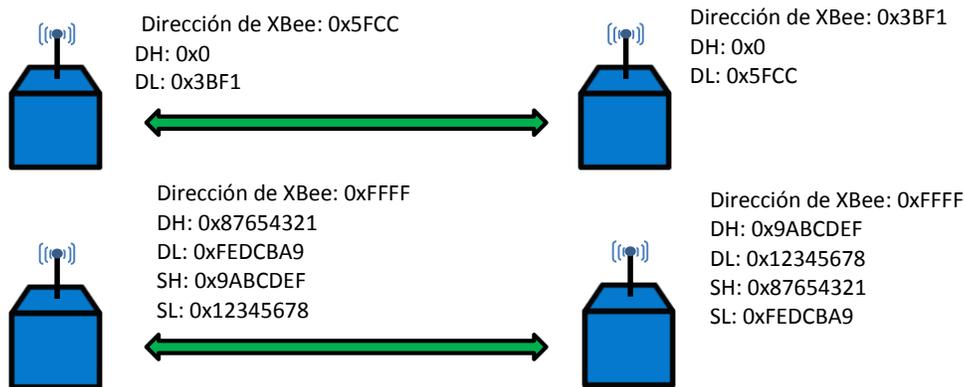


Figura 3.15 Direccionamiento de 16 y 64 bits para conexión punto a punto

Como siguiente parámetro a configurar es la velocidad de transmisión entre los módulos, para el proyecto se eligió una velocidad de 9600 bps así que al comando BD se le asigna el valor 3.

**Tabla 3.6** Velocidad de transmisión para XBee.

Parámetro	Configuración (bps)
0	1200
1	2400
2	4800
3	9600
4	19200
5	38400
6	57600
7	115200

Al trabajar con tramas que contendrán información tanto de configuración como del estado de las interfaces de monitoreo y control es necesario trabajar en modo API, este modo es el que nos permite manejar tramas de información.

El modo API se configura con el comando AP al cual simplemente al asignarle el valor de 1, le decimos al dispositivo que encienda este modo, más adelante explicaremos cómo funciona el modo API.

Todos los comandos mencionados se aplican de igual manera al Coordinator y al End Device

Al transmitir tramas es necesario activar la salida de la UART del Coordinator para que así este pueda enviar la información al procesador principal esto se activa con el comando IU al asignarle el valor de 1.

Dependiendo de la interfaz que se conecte al XBee End Device, se configuran los pines para poder interpretar la información, las interfaces que se conectaran son las siguientes:

- Un sensor de temperatura LM35
- Un sensor de gas TGS-2610
- Un foco de 127 Volts
- Una alarma que consta de un Buzzer y LED's
- Un motor de CD que varía su velocidad con PWM

Cada interfaz requiere o proporciona una señal para su control o lectura, por lo que cada una de estas se conecta a una patilla programada para una función específica.

Dentro de la misma ventana de X-CTU Modem Configuration se programan los pines de la siguiente manera, el sensor de temperatura entrega una señal analógica así que para interpretar esta señal se programa un CAD con el comando D0 asignando el valor 2, el sensor de gas entrega una señal digital así que se programa otro pin con el comando D2

como entrada digital asignando el valor 3, para encender el foco necesitamos proporcionar una señal digital a la interfaz con lo que se tiene que programar otro pin con el comando D4 asignando el valor 4, para la alarma se requiere proporcionar una señal digital a la interfaz por lo que se programa un pin como salida digital con el comando D5 asignando el valor de 4 por último el control de velocidad para el motor requiere que se le proporcione una señal analógica por lo que se conecta a un pin PWM0 programando con el comando P0 asignando el valor 2.

Toda la información se maneja a través de tramas que contienen la información leída por los pines del XBee, la trama también contiene un byte Checksum (byte de verificación de error), de longitud de trama, dirección del XBee que envía o recibirá la información e instrucciones a ejecutar; el protocolo ZigBee define un tipo estructura para las tramas soportada por los XBee.

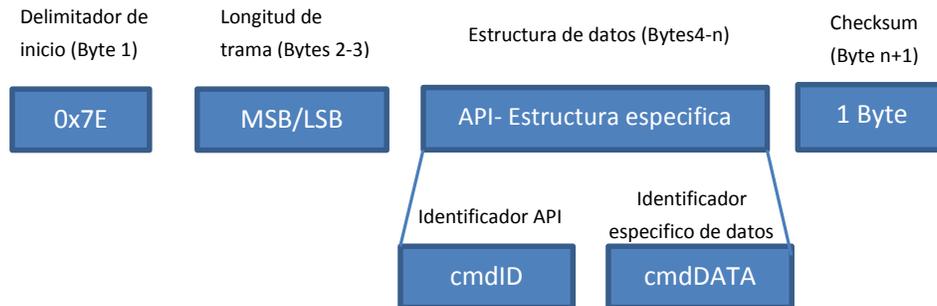


Figura 3.16. Estructura de la trama, modo API.

El Checksum es un byte que siempre estará al final de la trama y es para saber si la trama llegó sin errores, este byte siempre se calcula sumando todos los bytes de la trama excepto el byte delimitador de inicio, los bytes de longitud de trama y el Checksum. Del resultado de la suma solo se tomara el byte menos significativo, este valor en hexadecimal será restado a el valor FF el resultado de esta resta será Checksum.

Básicamente se manejarán 4 tipos de tramas:

- a) Para saber en cada momento el estado de las interfaces se envía una trama de solicitud, esta trama será enviada por el XBee Coordinador, si la petición llegó correctamente el XBee End Device deberá responder con la información leída en los pines asignados a cada interfaz.

**7E 00 0F 17 12 00 13 A2 00 40 49 06 FB FF FE 00 49 53 FE**

Tabla 3.7 Explicación de los bytes que componen la trama de solicitud de lectura de pines.

Bytes	Significado de byte
7E	Inicio de la trama byte delimitador
00 0F	Longitud de la trama sin contar el byte delimitador de la trama, los dos bytes de longitud y el byte Checksum
17	Comando remoto de solicitud, Identificador API
12	Identificador de la trama (cualquier numero distinto de cero)
00 13 A2 00 40 49 06 FB FF FE	Dirección IEEE del dispositivo al que se le va a solicitar la información
00	Opciones (no usadas)
49 53	Comando AT IS de solicitud de lectura de pines
FE	Checksum (byte para comprobar si la trama llego correctamente)

b) Para encender/apagar la alarma y el foco usamos la siguiente trama:

**7E 00 10 17 12 00 13 A2 00 40 49 06 FB FF FE 02 49 4F 10 F0**

Tabla 3.8 Explicación de los bytes que componen la trama del encendido y apagado del foco.

Bytes	Significado de byte
7E	Inicio de la trama byte delimitador
00 10	Longitud de la trama sin contar el byte delimitador de la trama, los dos bytes de longitud y el byte Checksum
17	Comando remoto de solicitud, Identificador API
12	Identificador de la trama (cualquier numero distinto de cero)
00 13 A2 00 40 49 06 FB FF FE	Dirección IEEE del dispositivo al que se le va a solicitar la información
02	Aplicar cambios en el dispositivo XBee
49 4F	Comando AT IO de solicitud de cambio de estado de los pines
10	Indicador de que pines serán cambiados de estado
F0	Checksum (byte para comprobar si la trama llego correctamente)

El byte #19 va a variar dependiendo de qué interfaz se encienda o apague y también el byte Checksum cambiara.

c) Para variar la velocidad del motor se usa la siguiente trama.

**7E 00 11 17 12 00 13 A2 00 40 49 06 FB FF FE 02 4D 30 00 FF 1C**

Como se ve la trama es muy similar a las anteriores solo cambia el byte de longitud los bytes #17 y #18 que son el comando AT M0 para modificar el ciclo útil de la señal PWM,

este ciclo variara de acuerdo al valor asignado a los bytes #19 y #20 (ver tabla de comandos AT, Tabla 3.7) y el byte #21 que es el Checksum.

- d) Por último la trama que recibiremos como resultado de la petición de la lectura de los pines tiene la siguiente estructura:

**7E 00 16 97 12 00 13 A2 00 40 49 06 FB 12 34 49 53 00 01 02 14 00 14 00 51 B9**

La estructura es similar a las demás tramas, el byte #18 indica que la lectura de los pines se ha hecho correctamente el byte #19 indica que solo se ha pedido una muestra de la lectura de los pines, el byte #20 y #21 indican los pines que inicialmente fueron habilitados y cuál es su función, el byte #22 y #23 indican el estado de los pines, los bytes #24 y #25 indican el valor del CAD.

Los valores de los bytes de estado de los pines, los bytes del CAD y Checksum variaran dependiendo de los cambios que se presenten en el sistema pero la estructura siempre será similar así que cuando se programe el sistema se debe tener cuidado que el programa del PIC tenga la capacidad de reconocer los cambios para ejecutar las acciones que se soliciten o bien para mostrar la información correcta del estado de las interfaces.

# 4

## Diseño y desarrollo de las interfaces de monitoreo y control

Las interfaces del sistema nos comunican con el medio físico y nos darán la capacidad de conocer variables como la temperatura y presencia de gas LP así como de manipular dispositivos como son la iluminación, velocidad de un motor y una alarma.

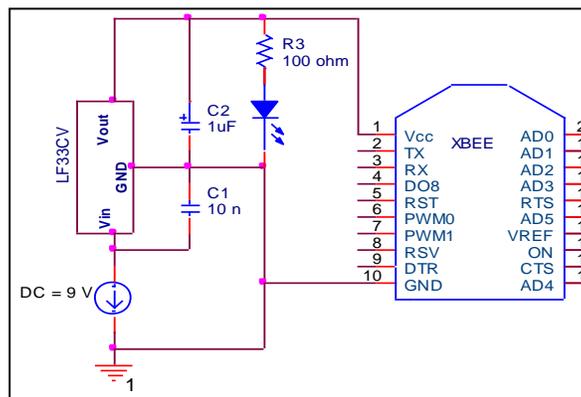
El diseño y elaboración de las interfaces debe ser precisa, buscando que sean eficientes energéticamente, que no presenten fallos con el tiempo, se adapten al sistema y bajo costo.

### 4.1 Diseño de PCB para módulo XBee

Las interfaces se conectan inalámbricamente al sistema por medio de un Módulo XBee, este módulo requiere de un circuito de alimentación así como de headers para conectar las interfaces al XBee, los pines que se utilizaran del módulo dependerá de la señal entregada o en su caso que necesite recibir cada interfaz.

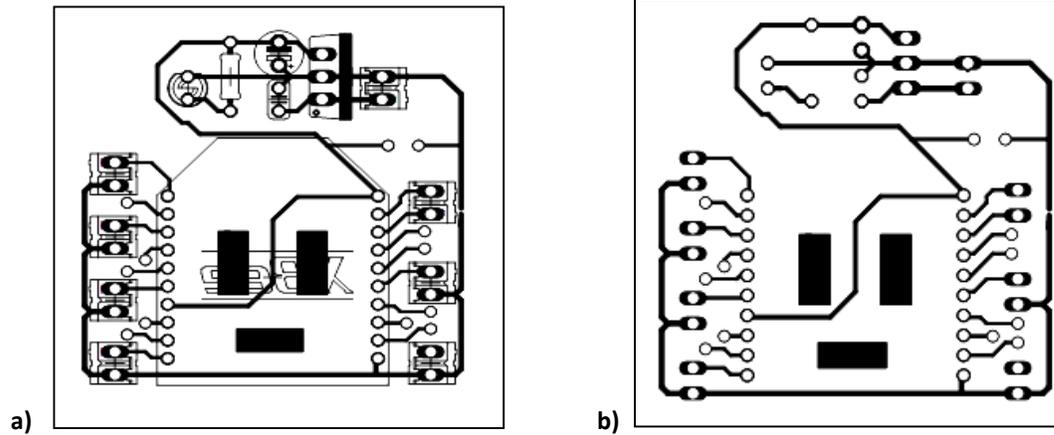
El circuito es muy simple consta de un regulador de voltaje de 3.3 Volts, un LED que indica que el circuito está alimentado y el módulo XBee.

Únicamente se muestra el circuito y su PCB, como se ha explicado el módulo XBee se debe programar para que se comunique con la interfaz de usuario local inalámbricamente y las interfaces de monitoreo y control deben ser adaptadas para que las señales que sean recibidas o enviadas puedan ser interpretadas por el XBee.



**Figura 4.1** Circuito del módulo inalámbrico XBee, a este se conectarán las interfaces de monitoreo y control.

Para el diseño del PCB y la placa fenólica se toma de referencia la figura 4.1:

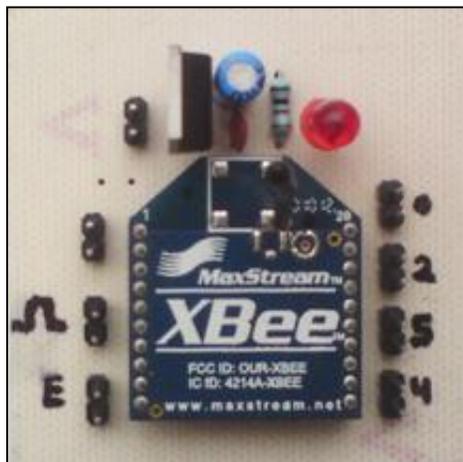


**Figura 4.2 a)** PCB de Modulo XBee con componentes electrónicos.

**b)** PCB de módulo XBee impreso en la placa fenólica.

Lista de componentes de la interfaz:

- 1 Regulador de Voltaje LF33CV
- 1 XBee
- 1 Capacitor cerámico de  $0.01\mu\text{F}$
- 1 Capacitor electrolítico de  $10\mu\text{F}$
- 1 LED rojo
- 1 resistencia de  $100\Omega$
- Header y placa fenólica



**Figura 4.3** Placa física del Dispositivo XBee que funciona como transmisor y receptor al cual se conectan todas las interfaces de control y monitoreo. Dimensiones 5cm. X 5cm.

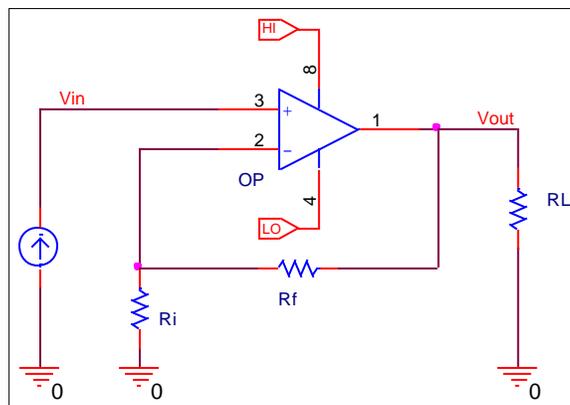
### 4.2 Sensor de Temperatura

El sensor de temperatura entrega una salida analógica así que necesitamos activar un convertidor analógico digital, pero antes de mandar esta señal analógica al dispositivo XBee tiene que ser adaptada a niveles que puedan ser interpretados por el CAD del XBee agregando circuitería al sensor.

El sensor LM35 esta calibrado para responder de manera lineal al cambio de temperatura, este nos entregara una salida de voltaje de  $10 \text{ mV} / ^\circ\text{C}$ , el CAD del XBee admite una variación de voltaje de  $0 \text{ V}$  a  $3.3 \text{ V}$ , por lo que tenemos que acoplar la salida del sensor a una variación igual, cuando el medio ambiente este a  $0^\circ$  se debe tener una salida de  $0 \text{ V}$  y cuando haya una temperatura de  $60^\circ$  obtengamos  $3 \text{ Volts}$  aproximadamente, cabe mencionar que no consideramos una temperatura mayor ya que no existe la probabilidad de que el medio ambiente alcance una temperatura de esta magnitud, si en dado caso se registra un valor de muy alta de temperatura significa que algo anda fuera de lo normal por ejemplo podría estar sucediendo un incendio por lo que el sistema procederá a activar una alarma.

Como se mencionó el sensor nos entregara una salida de voltaje de  $10 \text{ mV} / ^\circ\text{C}$ , así que si se registran  $0^\circ\text{C}$  el sensor entregara un voltaje de  $0 \text{ volts}$  y si se registran  $60^\circ\text{C}$  el sensor entregara un voltaje de  $600\text{mV}$ , para incrementar el voltaje a un nivel adecuado que pueda ser leído por el CAD del XBee, se conecta a la salida del sensor un amplificador operacional, en específico el LM358, se eligió este porque solo requiere de una fuente positiva para su funcionamiento.

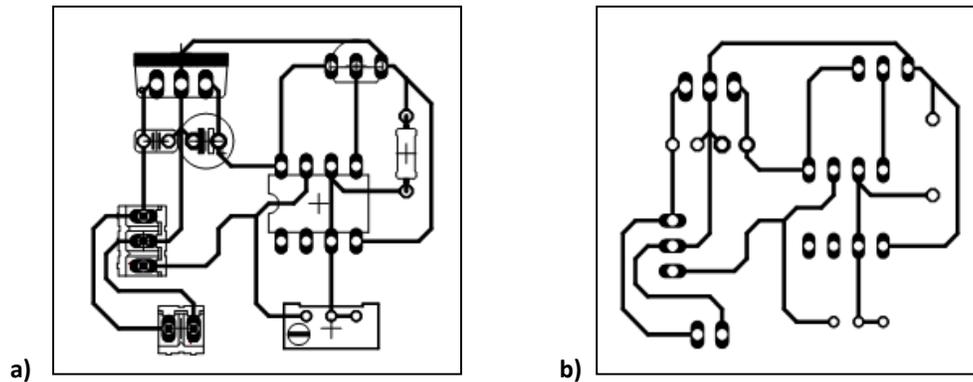
El amplificador operacional se conecta en configuración Amplificador no inversor como se muestra en el circuito de la figura 4.4:



**Figura 4.4** Configuración del circuito Amplificador no inversor utilizando un integrado LM358.



Para el diseño del PCB y la placa fenólica se toma de referencia la figura 4.5:

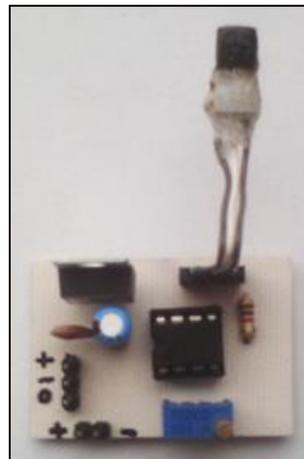


**Figura 4.6 a)** PCB de la interfaz de monitoreo, sensor de temperatura con componentes electrónicos.

**b)** PCB del sensor de temperatura impreso en la placa fenólica.

Lista de componentes de la interfaz:

- 1 Regulador de Voltaje 7805
- 1 Sensor de temperatura LM35
- 1 Amplificador operacional LM358
- 1 Capacitor cerámico de 0.01μF
- 1 Capacitor electrolítico de 10μF
- 1 Trimpot de 10KΩ
- 1 Resistencia de 1 KΩ
- Header y placa fenólica



**Figura 4.7** Placa física de la interfaz de monitoreo, sensor de temperatura. Dimensiones 3.8cm. X 2.5cm.

### 4.3 Sensor de gas

El sensor de gas de igual manera que el sensor de temperatura entrega una señal analógica la cual también adaptamos con circuitería para conectar al XBee, pero para este caso este sensor mide la concentración de gas LP en un determinado volumen de aire.

La hoja de datos del fabricante nos da recomendaciones, condiciones para operar el sensor y un modelo del sensor:

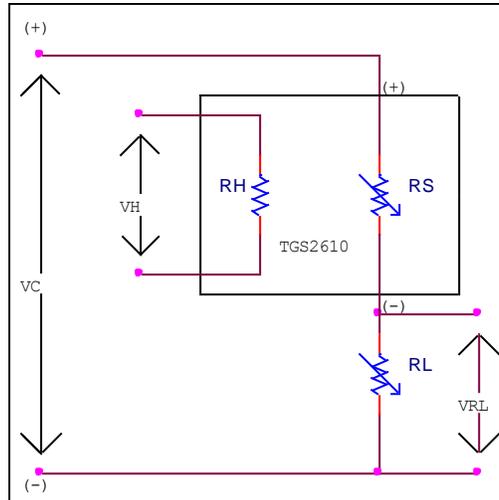


Figura 4.8 Modelo del sensor TGS2610.

El sensor requiere de dos voltajes de entrada, un voltaje para el calentador ( $V_H$ ) y un voltaje de circuito ( $V_C$ ). El voltaje de calentador es aplicado al calentador integrado en el circuito para mantener el elemento a una temperatura específica la cual es óptima para sensor. El voltaje de circuito es aplicado para permitir mediciones de voltaje ( $V_{RL}$ ) a través de la resistencia de carga ( $R_L$ ) la cual es conectada en serie con el sensor este al ser expuesto al gas este se comporta como una resistencia variable, a mayor concentración de gas menor será la resistencia y viceversa.

Para el buen funcionamiento el sensor debe cumplir tres condiciones:

1. Selección de resistencia de carga de acuerdo al ID del sensor.
2. En el momento que se alcanza una relación de  $R_s/R_L = 1$  existe el riesgo de explosión.
3. La potencia en el sensor ( $P_s$ ) debe ser menor a 15 mW

El sensor se alimenta a 5 V tanto para ( $V_C$ ) como para ( $V_H$ ). Para este voltaje el fabricante nos recomienda usar una resistencia de carga de acuerdo a un numero ID que viene marcado en cada sensor y que facilita la calibración, en este caso el ID del sensor es

18. De acuerdo a la siguiente tabla, se puede elegir diferentes valores de resistencia de carga para un determinado LEL (Limite de explosión inferior = 1800 ppm).

Tabla 4.1 Tabla de ID's para el sensor TGS2610

Sensor #ID	RL (kΩ) con ±1% de tolerancia			
	5% LEL	10% LEL	15% LEL	20% LEL
17	4.64	3.32	2.74	2.43
18	5.11	3.65	3.01	2.67
19	5.62	4.02	3.32	2.94

Así que para un ID=18 elegimos una resistencia de 3.65 kΩ, teniendo un 10% de LEL. Con esta selección cumpliremos que cuando  $R_S/R_L = 1$ , podremos hacer una lectura del voltaje umbral, cómo se menciona el sensor actúa como una resistencia variable que al ser expuesto al gas disminuye y con la resistencia de carga en serie se forma un divisor de voltaje así que cuando haya una caída de tensión de 2.5 volts en ( $R_L$ ) se cumplirá la condición 2 y estaremos en el voltaje umbral en el cual se deberá activar la alarma ya que existirá un riesgo de explosión.

Con el valor de la resistencia de carga ( $R_L$ ) ya seleccionada, se debe verificar la potencia de disipación ( $P_S$ ) del semiconductor bajo un límite de 15 mW para no dañar el sensor. La potencia en el sensor será mayor cuando el valor de ( $R_S$ ) es igual a ( $R_L$ ) al exponerse al gas.

El fabricante proporciona las siguientes fórmulas para calcular la potencia en el sensor.

$$P_S = \frac{(V_C - V_{RL})^2}{R_S} \quad (4)$$

$$R_S = \frac{V_C - V_{RL}}{V_{RL}} * R_L \quad (5)$$

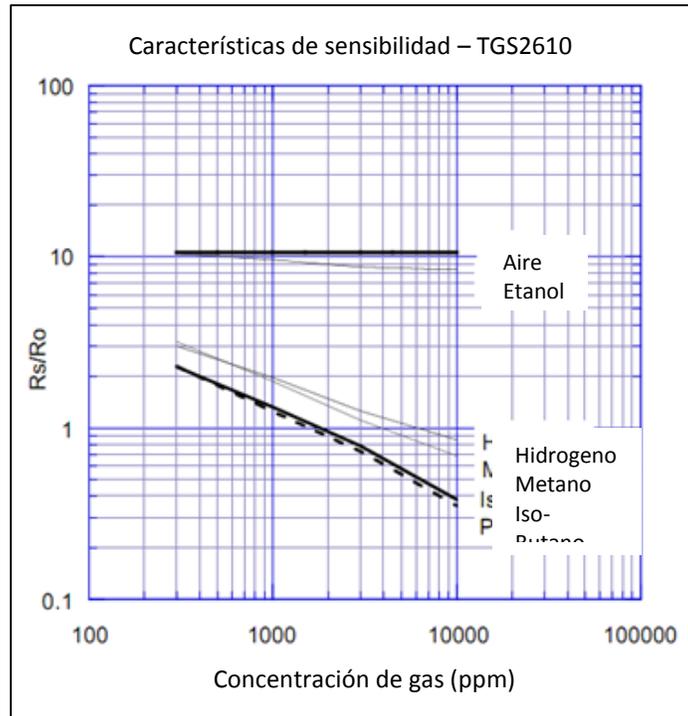
Primero se resuelve  $R_S$

$$R_S = \frac{5 - 2.5}{2.5} * 3650 = 3650 \quad (6)$$

Se sustituye  $R_S$  en la Ec. 4:

$$P_S = \frac{(5 - 2.5)^2}{3650} = 1.71 \text{ mW} \quad (7)$$

El valor obtenido de  $P_S$  es menor a 15 mW, así que el sensor funcionara correctamente sin riesgo de dañarse.



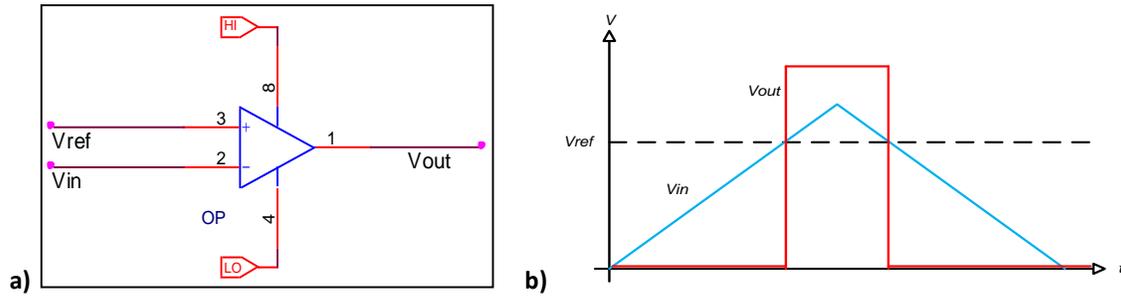
**Figura 4.9** Curva de comportamiento del sensor TGS2610 al detectar diferentes niveles de gas.

Revisando la gráfica de respuesta del sensor a distintas cantidades de gas, el eje horizontal indica la concentración de gas por partes de millón y el eje vertical indica la relación de  $R_s/R_L$ , así que la concentración de riesgo se da cuando hay 1800 ppm, ubicamos esa cantidad trazamos una línea vertical hasta cruzar con la curva de respuesta del sensor y trazamos otra línea horizontal partiendo de ese punto de cruce hacia el eje vertical y obtenemos un valor para la relación de  $R_s/R_L = 1$  tal y como el fabricante nos menciona.

Para activar la alarma, el sensor tiene que llegar al voltaje umbral, este voltaje es comparado con un voltaje de referencia, cuando el voltaje umbral supera el voltaje de referencia nos indicara que el sensor detecto una concentración de gas peligrosa por lo que se deberá activar la alarma.

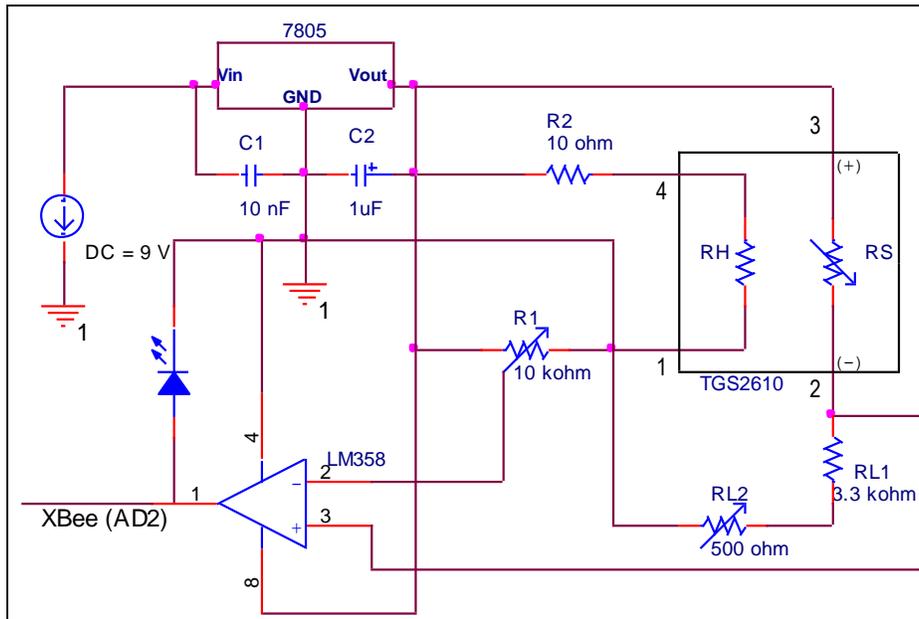
## 4 Diseño y desarrollo de las interfaces de monitoreo y control

Para hacer esta comparación usamos un amplificador operacional LM358 en configuración de comparador de voltaje no inversor.



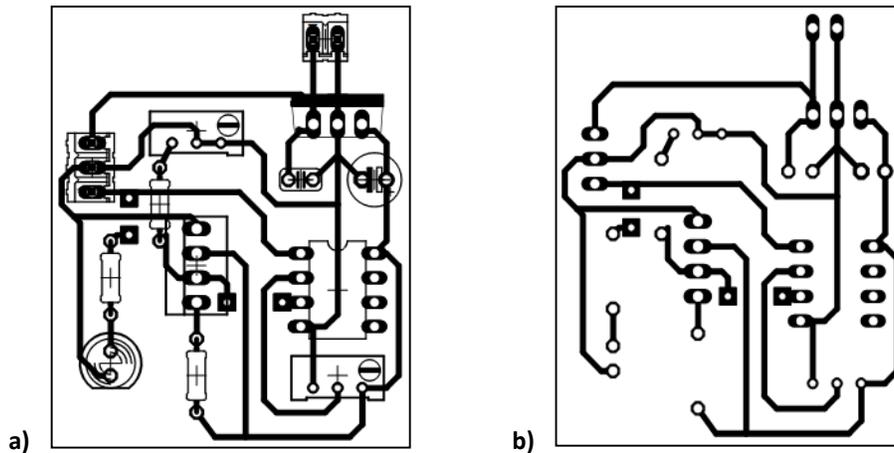
**Figura 4.10** a) Circuito comparador de voltaje. b) Imagen que explica el principio de funcionamiento del comparador de voltaje.

La salida del sensor se conecta al circuito comparador, en la siguiente imagen se muestra el circuito resultante, en este se observa que la resistencia de carga se compone de una resistencia  $RL1$  conectada en serie con una resistencia variable  $RL2$  esto para poder afinar la resistencia requerida de  $3.65\text{ k}\Omega$  ya que un valor comercial de este no existe, también cabe mencionar que para poder establecer el voltaje de referencia del circuito comparador se usa un potenciómetro  $R1$  y si este voltaje de referencia es disminuido por debajo de  $2.5\text{ volts}$  que es el voltaje umbral podemos hacer que el circuito sea más sensible.



**Figura 4.11** Circuito de la interfaz – Sensor de gas.

Para el diseño del PCB y la placa fenólica se toma de referencia la figura 4.11:

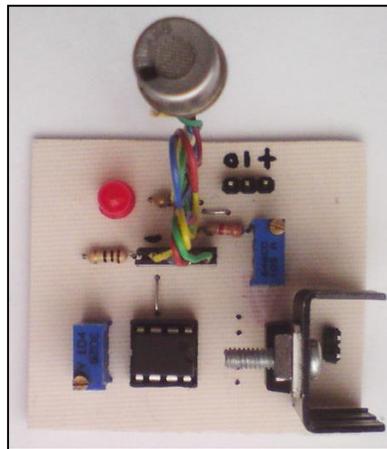


**Figura 4.12 a)** PCB de la interfaz de monitoreo, sensor de gas con componentes electrónicos.

**b)** PCB del sensor de gas impreso en la placa fenólica.

Lista de componentes de la interfaz:

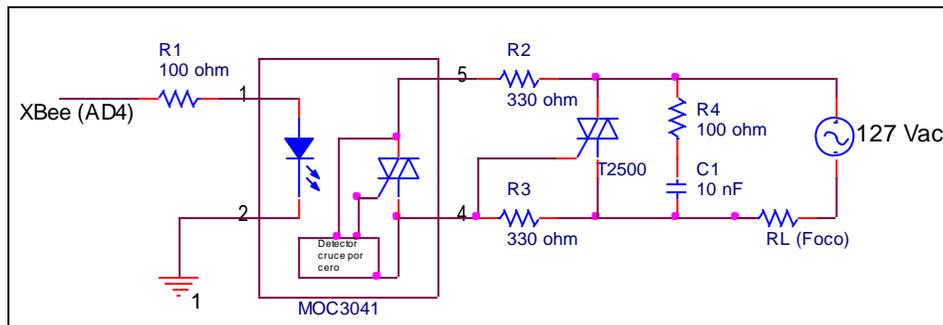
- 1 Regular de voltaje 7805
- 1 Sensor de gas TGS2610
- 1 Amplificador operacional LM358
- 1 Capacitor cerámico de 0.01 $\mu$ F
- 1 Capacitor electrolítico de 10 $\mu$ F
- 1 Trimpot de 10K $\Omega$
- 1 Trimpot de 500 $\Omega$
- 1 Resistencia de 10 $\Omega$
- 1 Resistencia de 3.3K



**Figura 4.13** Placa física de la interfaz de monitoreo, sensor de gas. Dimensiones 4.5cm. X 5cm.



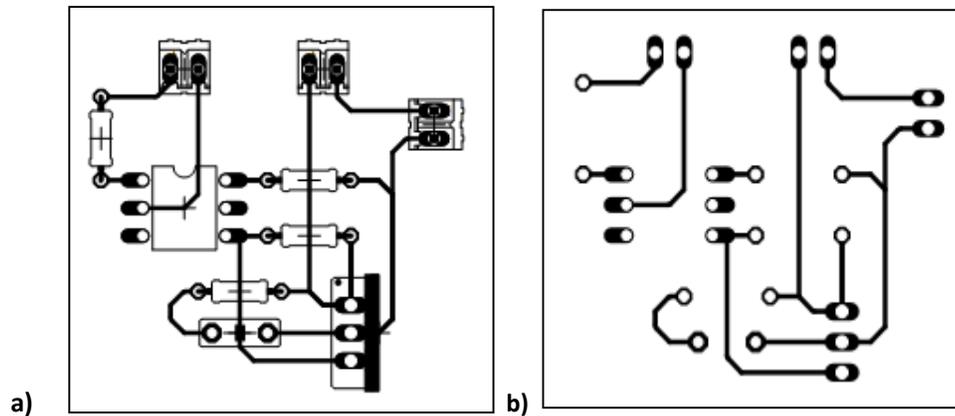
La implementación del circuito resultante es muy similar al recomendado por el fabricante:



**Figura 4.16** Circuito de la interfaz controladora de iluminación.

Cabe mencionar que este circuito soporta cargas que demanden un voltaje de 127V y hasta 6 Amperes, podríamos conectar un motor, focos de diferentes potencias etc.

Para el diseño del PCB y la placa fenólica se toma de referencia la figura 4.16:

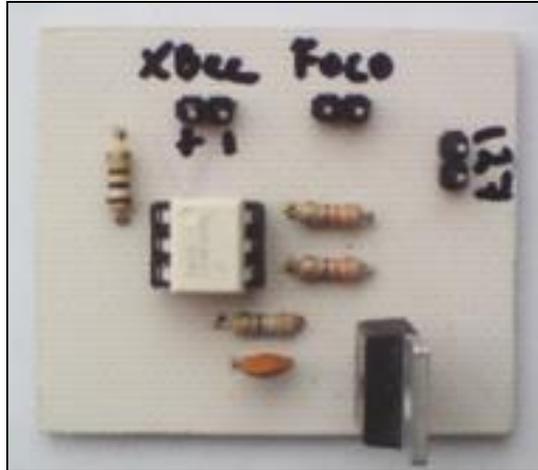


**Figura 4.17 a)** PCB de la interfaz de control para cargas de 127 V con componentes electrónicos.

**b)** PCB del sensor de temperatura impreso en la placa fenólica.

Listas de componentes de la interfaz:

- MOC3041
- Triac T2500
- 2 Resistencias de 330 $\Omega$
- 2 Resistencia de 100  $\Omega$
- 1 capacitor cerámico de 0.01 $\mu$ F
- Header y placa fenólica



**Figura 4.18** Placa física de la interfaz para controlar cargas de 127V on/off, controla foco incandescente. Dimensiones 3.7cm. X 4.4cm.

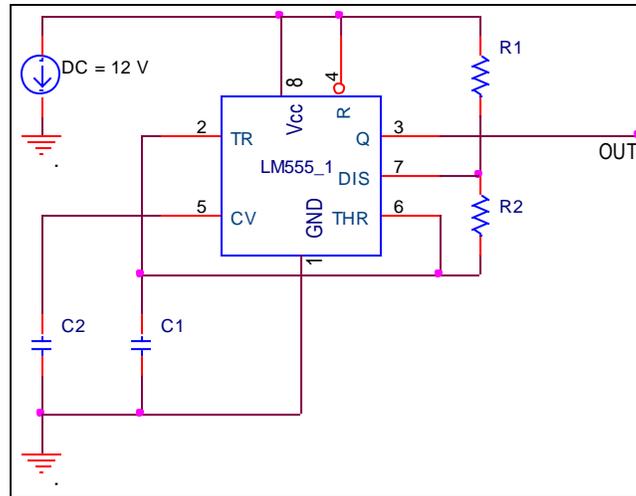
### 4.5 Alarma

La alarma consiste en un Buzzer y un arreglo de 4 LED's que darán una señal intermitente cuando el sensor de gas y/o de temperatura indique mediciones fuera de lo normal.

La alarma (Buzzer y LED's) funcionan a 12 Volts, esta se activara con una señal de 3 Volts así que para adaptar este circuito usaremos un opto acoplador 4N25 que aislara los circuitos que funcionan a distintos voltajes para su protección, cuando se envié la señal de 3 volts al opto acoplador encenderá un LED infrarrojo interno y a su vez esta luz incidirá sobre el silicio y excitara la base de un fototransistor por el cual comenzara a fluir corriente que alimentara a un circuito integrado que hará parpadear la alarma.

Para que la alarma parpadee se usa el circuito integrado LM555 al cual agregándole circuitería externa entregara una oscilación de determinada frecuencia que puede ser calculada y dependerá de los elementos que se conecten a este. La configuración de este circuito se le denomina modo estable porque su oscilación es constante mientras el circuito este alimentado.

Las fórmulas para calcular la frecuencia y ciclo útil no las proporciona el fabricante en su hoja de datos así como la conexión del mismo:



**Figura 4.19** Conexión del circuito integrado LM555 para funcionar en modo astable

$$f = \frac{1.44}{(R_1 + 2R_2)C_1} \quad (8)$$

$$t_1 = \frac{1}{1.44}(R_1 + R_2)C_1$$

$$t_2 = \frac{1}{1.44} * R_2 * C_1 \quad (9)$$

$$D = \frac{R_2}{R_1 + 2R_2} \quad (10)$$

Se elige que el circuito parpadee una vez por segundo, por lo que la frecuencia será de 1Hz,  $R_2$  se propone de  $22K\Omega$  y  $C_1$  también se propone de  $10\mu F$  así que la incógnita faltante es  $R_1$  la cual se despeja:

$$R_1 = \frac{1.44}{C_1 * f} - 2R_2 \quad (11)$$

Teniendo la formula se sustituyen los valores

$$R_1 = \frac{1.44}{10\mu F * 1Hz} - 2 * 22K\Omega$$

$$R_1 = 100K\Omega$$

Ya se tienen todos los valores de los componentes así que en el circuito de la figura sustuiremos los componentes resultantes para obtener el oscilador deseado, ahora se calcula el ciclo útil con las siguientes formulas:

$$t_1 = \frac{1}{1.44} (R_1 + R_2) C_1 \quad (12)$$

$$t_2 = \frac{1}{1.44} * R_2 * C_1 \quad (13)$$

La fórmula determina el ciclo útil es decir el tiempo que se mantiene encendida la alarma en el periodo y la formula determina el tiempo que estará apagado, al sustituir el valor de los componentes se tienen los siguientes tiempos:

$$t_1 = \frac{1}{1.44} (100K\Omega + 22K\Omega) * 10\mu F = 847.22mS$$

$$t_2 = \frac{1}{1.44} * 22K\Omega * 10\mu F = 152.78mS$$

Se tiene que la alarma por cada segundo que este activada se encenderá por un tiempo de  $847.22mS$  y estará apagada por  $152.78mS$  si sumamos estos tiempos nos da un tiempo total de 1 segundo.

Como la señal del oscilador no tiene la suficiente potencia para alimentar el Buzzer y los LED's, esta señal se envía a un opto acoplador para que excite el diodo infrarrojo y a su vez la luz infrarroja emitida por el diodo excite la base del fototransistor el cual permita un flujo de corriente que alimentara la compuerta de un transistor Mosfet, al excitar la compuerta se crea un canal dentro del Mosfet que permite fluir la corriente a través de este proporcionado la suficiente potencia para alimentar el Buzzer y los LED's.

El voltaje que alimenta la compuerta del transistor Mosfet proviene de un multiplicador de voltaje, este circuito consiste de un oscilador y una etapa de diodos y capacitores que entregan el doble del voltaje de alimentación del circuito esto para poder excitar la compuerta adecuadamente y así evitar que se caliente y trabaje correctamente el transistor Mosfet.

El oscilador del multiplicador de voltaje es muy similar al explicado anteriormente también se construyó con un LM555 la diferencia es que este oscilador trabaja a una frecuencia mucho mayor para poder cargar la red multiplicadora de voltaje. En la figura ( ) vemos la red multiplicadora junto con el oscilador y funciona de la siguiente manera: cuando la salida del oscilador se encuentra en nivel alto (12 Volts) el primer diodo (D1) está en polarización inversa (circuito abierto) y el segundo diodo (D2) está en polarización directa (circuito cerrado) por lo que los dos capacitores se cargaran (C3, C4) al valor de alimentación del circuito (Vcc). Cuando el oscilador se encuentre en estado bajo (0 Volts)

el primer diodo se encontrara en polarización directa y el segundo diodo en polarización inversa esto provocara que el voltaje almacenado en el primer capacitor se suma a la carga almacenada en el segundo capacitor teniendo así a la salida  $2V_{cc}$  aproximadamente, este voltaje no es exactamente el doble del voltaje de alimentación debido a la caída de tensión en los diodos y la corriente de fuga en los capacitores.

El voltaje resultante de este circuito es el que se suministra a la compuerta de transistor Mosfet.

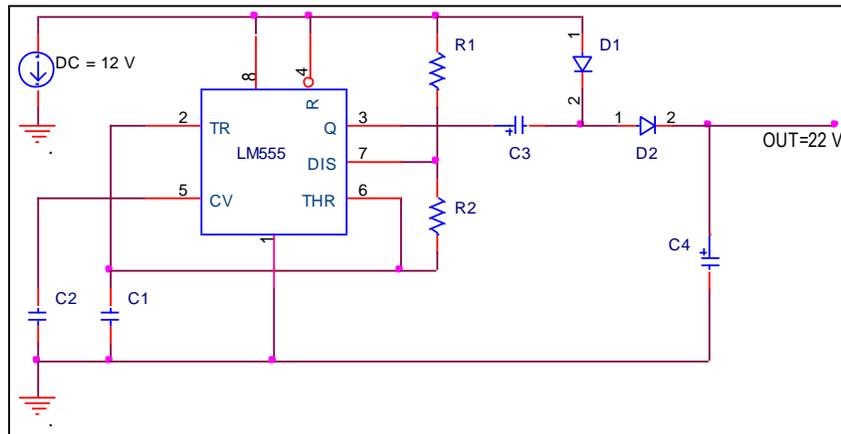


Figura 4.20 Configuración del circuito doblador de voltaje.

El circuito resultante de la alarma se muestra en la siguiente figura:

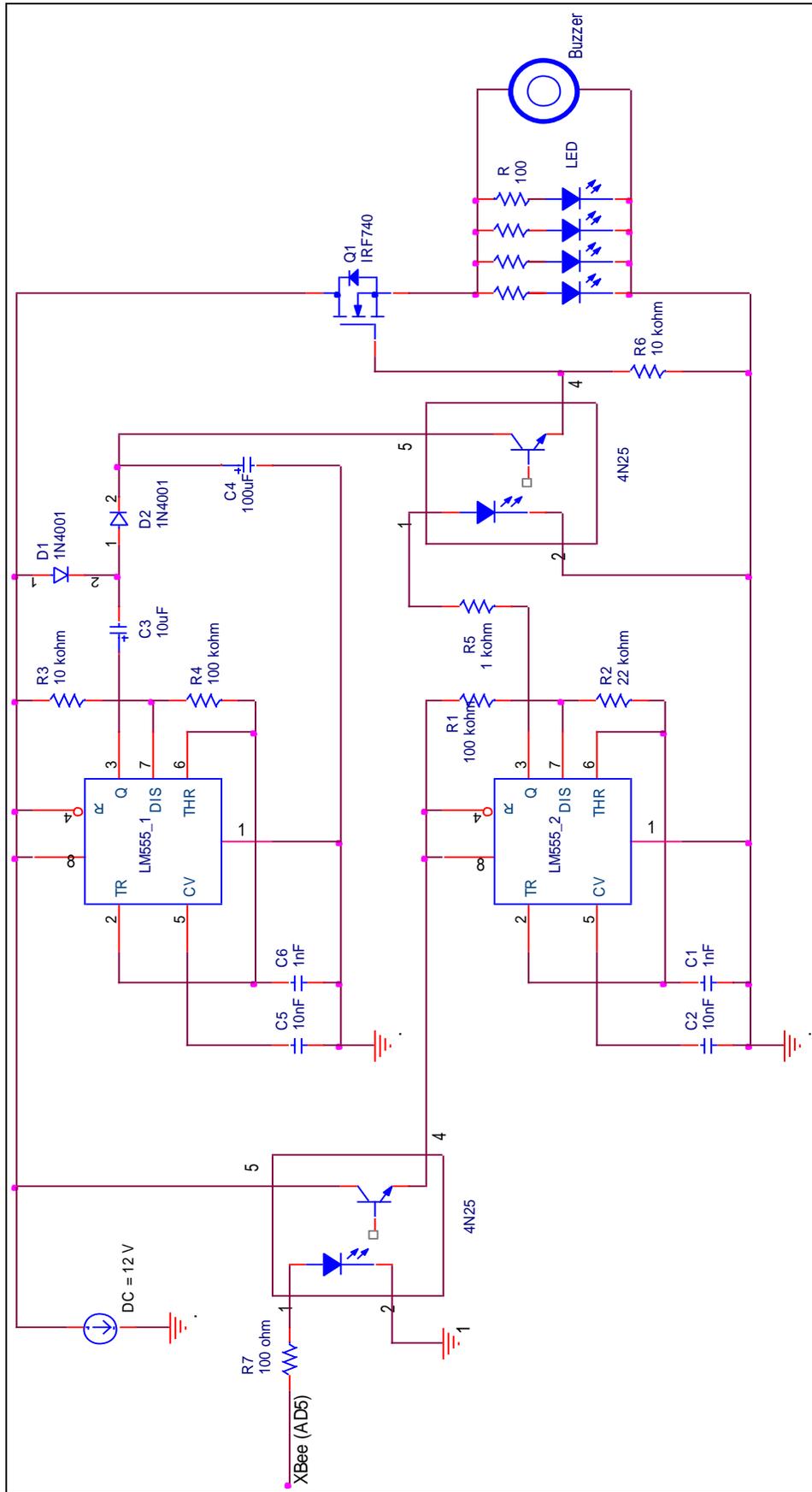
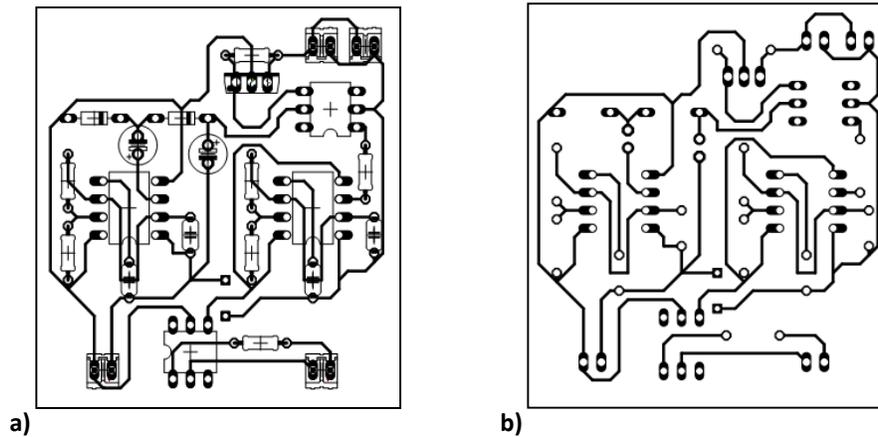


Figura 4.21 Circuito de la interfaz de control de la alarma

Para el diseño del PCB y la placa fenólica se toma de referencia la figura 4.21:

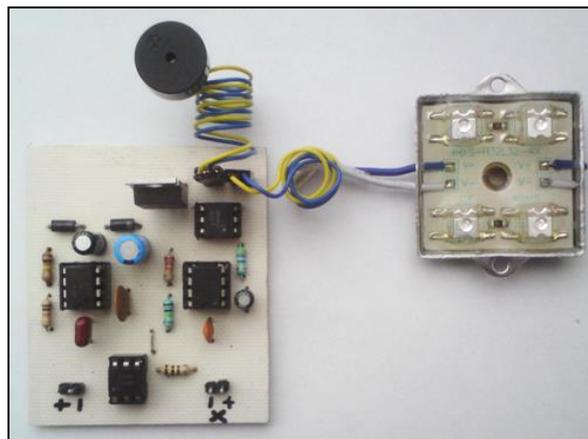


**Figura 4.22 a)** PCB de la interfaz de control para la alarma con componentes electrónicos.

**b)** PCB de la interfaz de control para la alarma impreso en la placa fenólica.

Lista de componentes de la interfaz:

- 2 Opto acopladores 4N25
- 2 Circuitos integrados LM555
- 1 Mosfet IRF740
- 1 Buzzer
- 1 Modulo con 4 LEDs azules
- 2 Diodos 1N4001
- 2 Capacitores cerámicos de 10nF
- 2 Capacitores cerámicos de 1nF
- 1 Capacitor electrolítico de 10 $\mu$ F
- 1 Capacitor electrolítico de 100 $\mu$ F
- 1 Resistencia de 100 $\Omega$
- 1 Resistencias de 1K $\Omega$
- 2 Resistencias de 10K $\Omega$
- 1 Resistencias de 22K $\Omega$
- 2 Resistencias de 100K $\Omega$



**Figura 4.23** Placa física de la interfaz de control para la alarma consta de un módulo LED y un Buzzer. Dimensiones 6cm X 5.2 cm.

### 4.6 Interfaz de Motor controlado por PWM

Este circuito consiste en un control de velocidad por Modulación de Ancho de Pulso (PWM: Modulation Wide Pulse) para un motor de Corriente Directa.

La señal PWM será proporcionada por el dispositivo XBee como se había mencionado anteriormente, esta señal tiene un voltaje de 3 Volts con una frecuencia de 15.6 KHz a la cual se puede variar su ciclo útil desde 0 % hasta el 100 % este ciclo es manipulado mediante tramas que son enviadas por el microcontrolador de la interfaz local y remota.

Como en los anteriores circuitos debemos aislar el circuito de control de velocidad del circuito XBee esto lo hacemos con un opto acoplador, se requiere que este opto acoplador sea mucho más rápido es decir que se pueda encender y apagar con mayor velocidad ya que la señal PWM así lo requiere para eso usamos el circuito integrado 6N137.

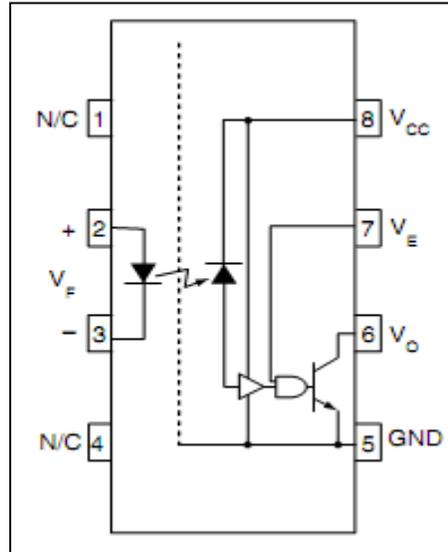


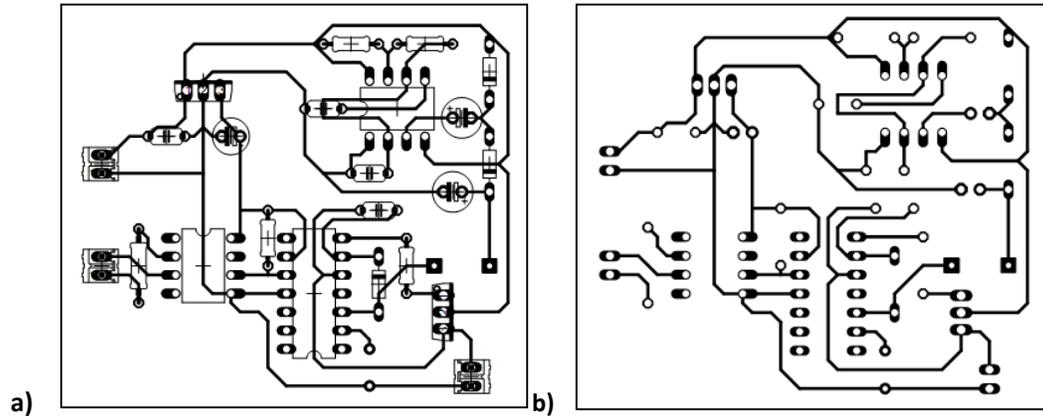
Figura 4.24 Opto acoplador 6N137.

La señal entregada por este circuito es lógica TTL (0V a 5V, con salida a transistor) y como se puede ver en la configuración interna del circuito integrado la señal de salida será la negación de la señal entrada, así que cuando la señal de entrada tenga un ciclo útil del 100% la señal de salida del circuito tendrá un ciclo útil del 0% y así sucesivamente para cualquier porcentaje del ciclo útil.

Posteriormente esta señal ya aislada es enviada a un driver IR2110, este es un circuito integrado muy común para puentes H, inversores, etc. que son controlados por PWM y que su principal función es alimentar IGBT's y Mosfet's de manera que a elevada frecuencia funcionen correctamente como es nuestro caso.



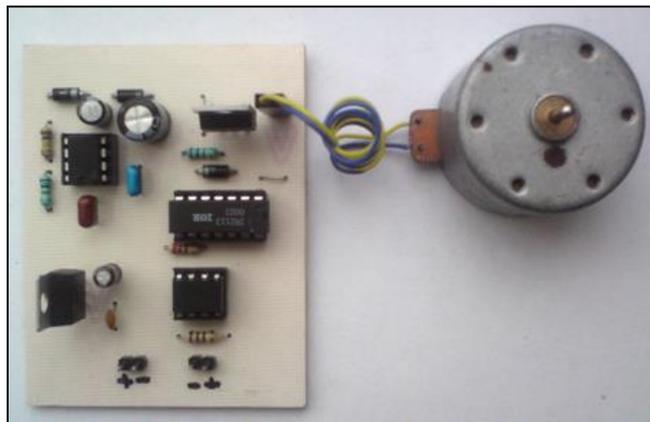
Para el diseño del PCB y la placa fenólica se toma de referencia la figura 4.25:



**Figura 4.26** a) PCB de la interfaz de control para variar la velocidad de un motor de DC con PWM con componentes electrónicos. b) PCB de la interfaz de control para variar la velocidad de un motor de DC con PWM impreso en la placa fenólica.

Lista de componentes de la interfaz:

- 1 Opto acoplador 6N137
- 1 Driver IR2110
- 1 Circuito integrado LM555
- 1 Regular de voltaje 7805
- 1 Mosfet IRF740
- 1 Motor DC de 12V
- 2 Diodos 1N4001
- 1 Diodo schotky
- 2 Capacitores cerámicos de 10nF
- 1 Capacitor cerámico de 1nF
- 1 Capacitor electrolítico de 10 $\mu$ F
- 1 Capacitor electrolítico de 100 $\mu$ F
- 1 Resistencia de 100 $\Omega$
- 1 Resistencias de 220 $\Omega$
- 1 Resistencia de 10K $\Omega$
- 2 Resistencias de 100K



**Figura 4.27** Placa física de la interfaz de control para variar la velocidad de un motor de DC con PWM. Dimensiones 7.2cm. X 5.5cm.

### 4.7 Interfaz de usuario local

Esta interfaz permite comunicarse con todas las interfaces anteriormente descritas tanto para su control como monitoreo y de igual manera permite la comunicación de las interfaces con la tarjeta PIC 32 Ethernet.

Esta interfaz consta de software y hardware que en conjunto permiten manipular el sistema y administrar la información. La conexión de los componentes resulta ser sencilla lo interesante es la programación.

Un microcontrolador PIC16F876A será el cerebro de la interfaz, se encargara de enviar y recibir tramas de información a través del dispositivo XBee, la información se actualizara cada segundo con el propósito de saber el estado de cada interfaz la cual se visualizara en un display LCD modelo JHD162A. Localmente podemos manipular el sistema para ello se agregaron 5 botones y un potenciómetro que se programaron para cumplir distintas funciones.

Todos los componentes de la interface local están conectados al microcontrolador, este es un circuito integrado que puede ser usado para muy diversas aplicaciones todo dependerá de cómo se programe y la capacidad del mismo, más adelante hablaremos de cómo funciona; el circuito tiene un amplio patillaje que está dividido en grupos de patillas llamados puertos. La configuración del display LCD requiere que se conecte a un puerto para poder desplegar texto en la pantalla. Como se muestra en la figura 4.28.

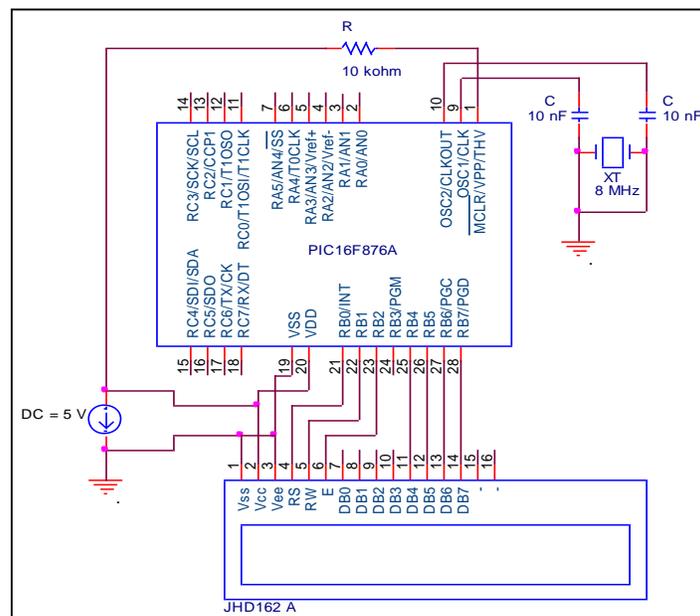


Figura 4.28 Circuito de la pantalla LCD conectada al microcontrolador.

Los 5 botones van conectados a un pin del microcontrolador que está programado para funcionar como CAD (convertidor analógico digital), cada botón al ser presionado deberá proporcionar un valor distinto de voltaje que será leído por el CAD y así poder asignar a cada botón una función, los botones se conectan de tal manera que funcionen como divisores del voltaje.

El divisor de voltaje consiste en repartir una tensión de una fuente entre dos o más resistencias conectadas en serie como se observa en la figura 4.29 si variamos los valores de las resistencias podemos obtener a la salida el voltaje que deseemos.

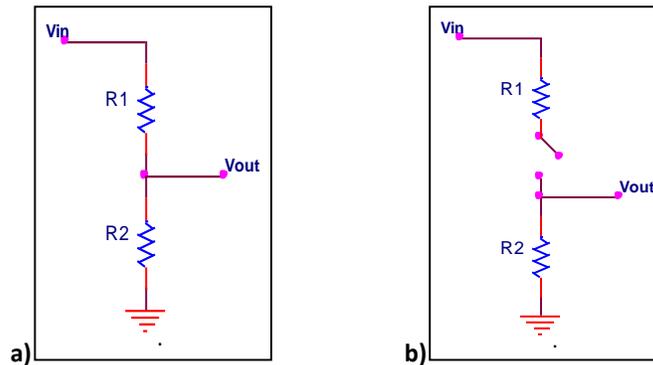


Figura 4.29 a) Divisor de voltaje. b) Divisor de voltaje con interruptor.

Este circuito se ve regido por la ecuación 14

$$V_{out} = \frac{R_2}{R_1 + R_2} * V_{in} \quad (14)$$

Al agregar a este divisor un botón, el circuito queda como se ve en la figura 4.29 – b cuando el botón este abierto el  $V_{out}$  siempre será 0 Volts pero al cerrar el circuito la tensión será el resultado de la ecuación 14.

El CAD del microcontrolador puede interpretar voltajes que van de 0 V a 5 V, el CAD convertirá este voltaje a un valor digital binario de 8 bits que puede variar de 0 a 255 así que para cada botón asignamos un voltaje y a su vez este voltaje se le asigna un rango de valores digitales para así determinar que botón fue oprimido y ejecutar la acción para el que fue programado.

Para el cálculo de los divisores de voltaje dejamos fijo el valor de  $R_2 = 2.2 \text{ k}\Omega$  que elegimos arbitrariamente. Despejamos el valor de R1 de la ecuación 14 y resolvemos para cada valor de voltaje para obtener los valores de R1 para cada circuito como se observa en la tabla 4.2.

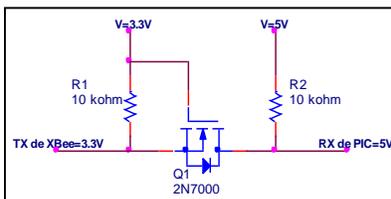
**Tabla 4.2** Valores calculados para los divisores de voltaje.

Botón	Vout	R1c ( $\Omega$ )	R1r ( $\Omega$ )	R2 ( $\Omega$ )	Valor de CAD
1	0.5	244.44	470//560	2200	5-50
2	1.5	942.85	1000	2200	51-100
3	2.5	2200	2200	2200	101-150
4	3.5	5130	10000//10000	2200	151-200
5	4.5	19800	10000+10000	2200	201-250

Se aprecia en la tabla que hay valores de resistencia R1c que significan -resistencia 1 calculado- y R1r que significa -resistencia 1 real- como se sabe no hay resistencias de todos los valores se tienen que hacer arreglos de resistencias comerciales ya sea en paralelo o serie para obtener una que se aproxime a la calculada, Sustituyendo los valores de resistencia R1r ( $\Omega$ ) en el circuito de la figura 4.29 - b obtendremos el  $V_{out}$  deseado al oprimir cada botón.

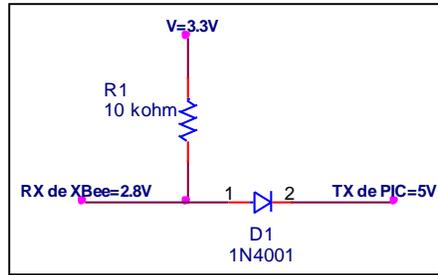
También se conecta al microcontrolador el XBee para poder comunicarnos con las interfaces, las tramas de información enviadas y recibidas por el XBee se transmiten por medio de una UART usando el protocolo RS232 que viene integrado en el XBee de igual manera el microcontrolador dispone de una UART que usa el mismo protocolo que permitirá comunicarse con el XBee, sin embargo la UART del XBee funciona con un nivel bajo de 0 V y un nivel alto de 3.3 V y la UART del microcontrolador funciona con un nivel bajo de 0 V y un nivel alto de 5V por lo tanto debemos adecuar los niveles lógicos para que las señales enviadas y recibidas entre las dos UARTs puedan ser interpretadas en su respectivo nivel, como se sabe la comunicación se lleva de manera Full Duplex por lo que cada UART tiene dos patillas designadas como TX (línea de transmisión) y RX (línea de recepción) su conexión se realiza cruzada es decir patilla TX del PIC se conecta al RX del XBee y la patilla RX del PIC se conecta al TX del XBee.

Para adaptar el nivel de 3.3 V a 5 V se optó como solución usar un Mosfet que permitirá conectar cualquier señal de bajo voltaje a uno de alto voltaje. Si 3.3V es demasiado bajo para registrar un '1' lógico en la UART del microcontrolador que es un dispositivo de 5V, usamos la siguiente configuración para interconectar los dos dispositivos.



**Figura 4.30** Interfaz para adaptar los niveles lógicos entre UARTs (TX-RX).

En el circuito de la figura 4.30 se observa que el XBee de 3.3V está transmitiendo al dispositivo de 5V. Cuando el XBee de 3.3V transmita un '1' lógico, el MOSFET se pondrá en alto (HIGH) y el pin TX-5V verá 5V a través de la resistencia de pull-up R2. Cuando el dispositivo de 3.3V transmita un '0' lógico (0 V), el MOSFET irá a tierra y el pin TX-5V verá 0 V. Hay que asegurarse de que el dispositivo de bajo voltaje se encuentre en el lado izquierdo del Mosfet. Este circuito no trabajará en la otra dirección (desde un alto voltaje a uno bajo) debido al diodo interno dentro del Mosfet.



**Figura 4.31** Interfaz para adaptar los niveles lógicos entre UARTs (RX-TX).

Cuando el dispositivo de 5V transmite un '1' lógico, el diodo bloqueará cualquier flujo de corriente que pase al dispositivo de 3.3V, por lo que éste no verá cambio alguno y permanecerá con el pull-up a 3.3V por la resistencia R1. Cuando el dispositivo de 5V transmita un '0' lógico, el diodo irá a tierra, esto permitirá el flujo de corriente a través del diodo desde la línea de 3.3V. La línea de entrada en el dispositivo de 3.3V verá la caída de voltaje del diodo (0.6V), por lo que en vez de tener 0V en el pin RX, el dispositivo de 3.3V tendrá 0.6V. De todas formas un voltaje de 0.6V es considerado lo suficientemente bajo como para que el dispositivo de 3.3V lo entienda como un '0' lógico.

Los diodos estándar de silicio tienen una caída de voltaje de aproximadamente 0.6V. Mientras más bajo es ese voltaje, mejor es para la aplicación, por lo que se usó un diodo Schottky, los cuales tienen una caída de voltaje cercano a los 0.2V.

Esta interfaz se alimenta de dos voltajes 3.3V y 5V por lo que se usaron circuitos integrados para regular el voltaje suministrado a cada dispositivo de la interfaz para su correcto funcionamiento. Figura 4.32.

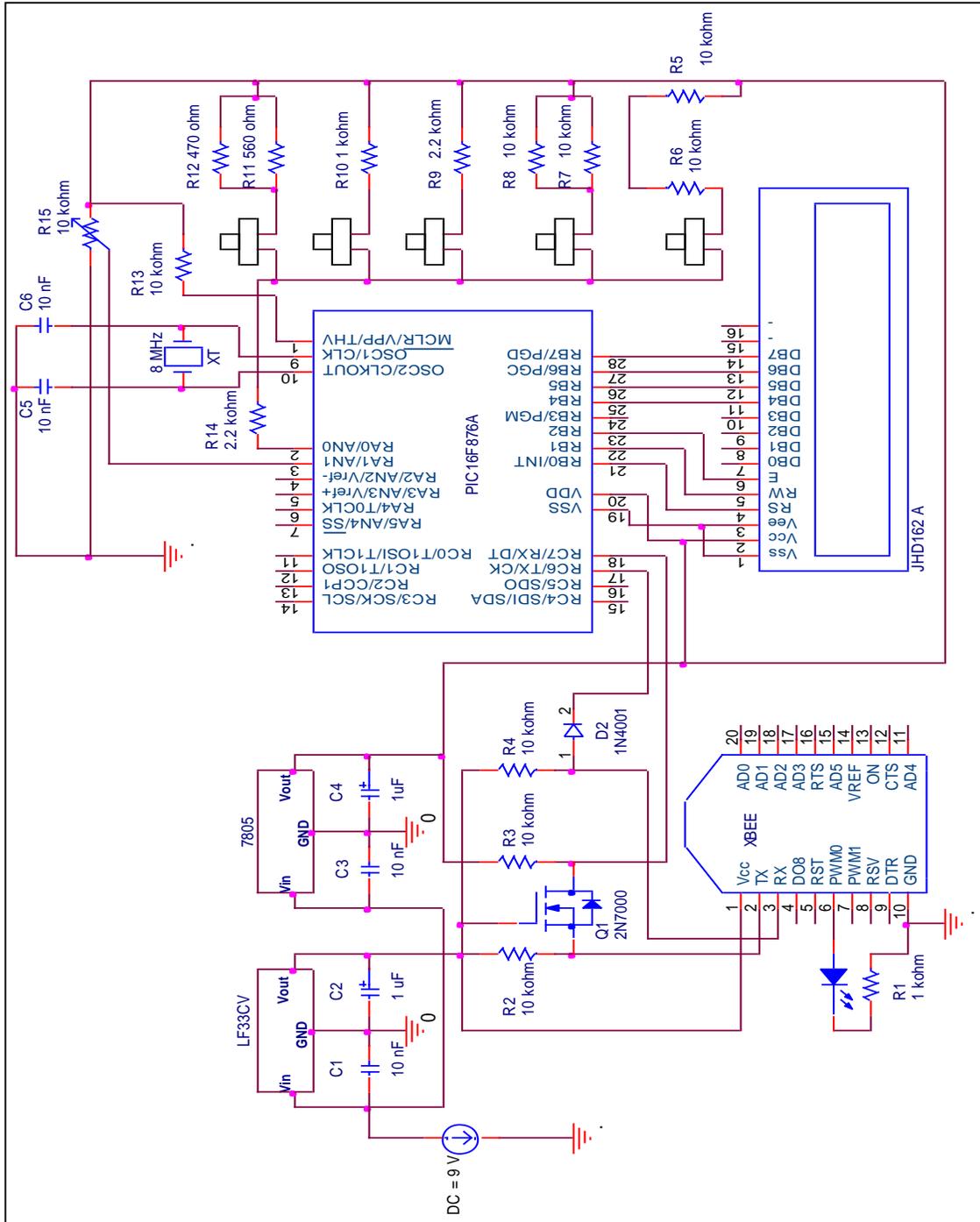
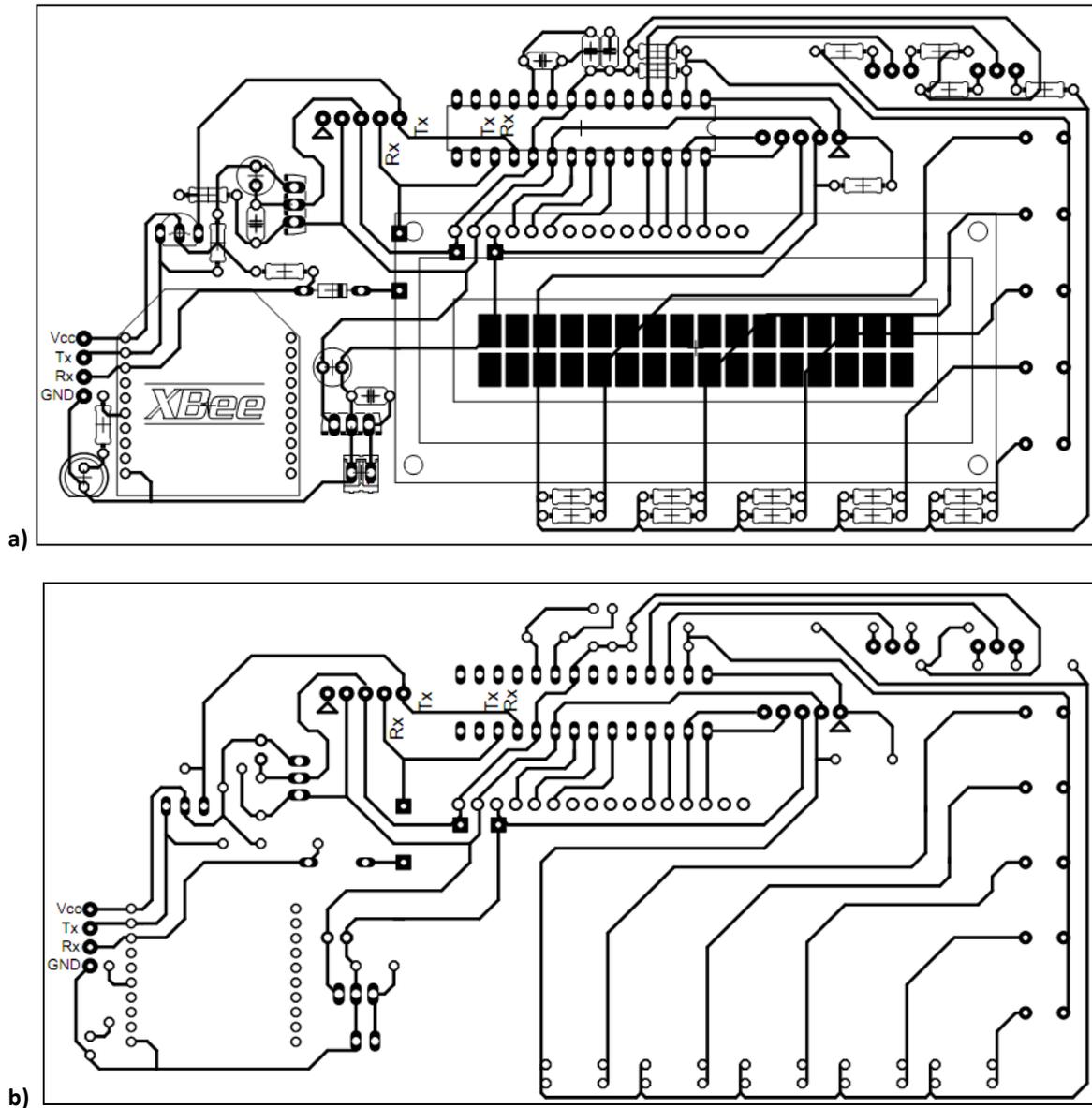


Figura 4.32 Circuito de la interfaz de usuario local constituido por XBee, PIC16F876A, pantalla LCD y botonera.

Para el diseño del PCB y la placa fenólica se toma de referencia la figura 4.32:



**Figura 4.33 a)** PCB de la interfaz de usuario local componentes electrónicos.

**b)** PCB de la interfaz de usuario local impreso en la placa fenólica.

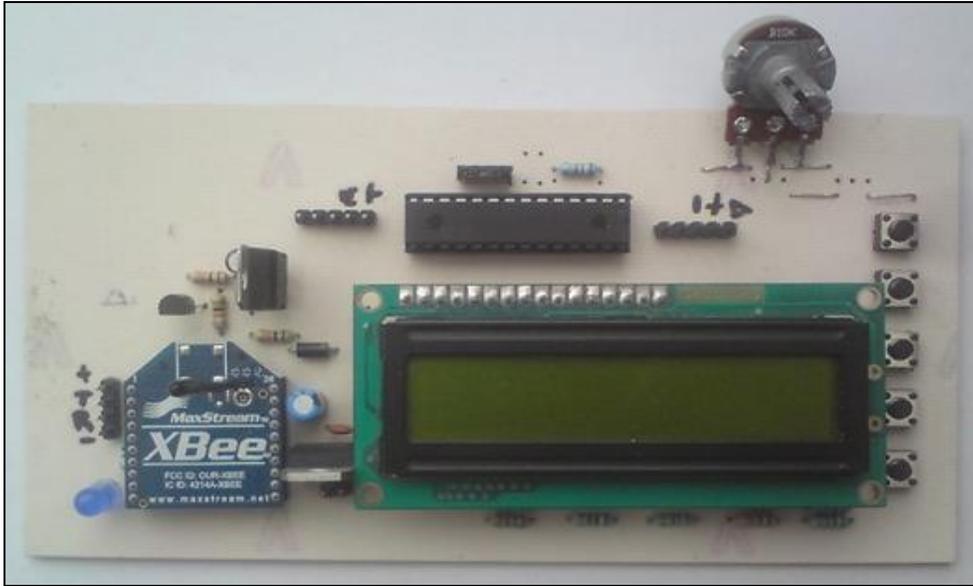
Lista de componentes de la interfaz:

- 1 Microcontrolador PIC16F876A
- 1 Pantalla LCD JHD162A
- 1 XBee
- 5 Push Boton
- 1 Regulador de voltaje 7805
- 1 Regulador de voltaje LF33CV
- 1 Mosfet 2N7000
- 1 Cristal de 8 Mhz
- 1 Diodo schotky
- 1 LED azul
- 4 Capacitores cerámicos de 10nF

#### 4 Diseño y desarrollo de las interfaces de monitoreo y control

---

- 2 Capacitores electrolíticos de 10 $\mu$ F
- 1 Potenciómetro de 10K $\Omega$
- 1 Resistencia de 470 $\Omega$
- 1 Resistencia de 560 $\Omega$
- 1 Resistencia de 1 $\Omega$
- 1 Resistencia de 2.2K $\Omega$
- 8 Resistencias de 10K $\Omega$
- Header y placa fenólica



**Figura 4.34** Placa física de la interfaz de usuario local para controlar y monitorear el estado de las interfaces así como el control de la comunicación con las interfaces y el PIC 32. Dimensiones 7.7cm. X 15cm.

# 5

## Resultados

El sistema se compone de diferentes secciones que son interfaces de control y monitoreo, módulos inalámbricos, interfaz de usuario local y la tarjeta que aloja la página web; cada sección desempeña una función por lo que se dedica este apartado para ver su conexión, funcionamiento y resultados.

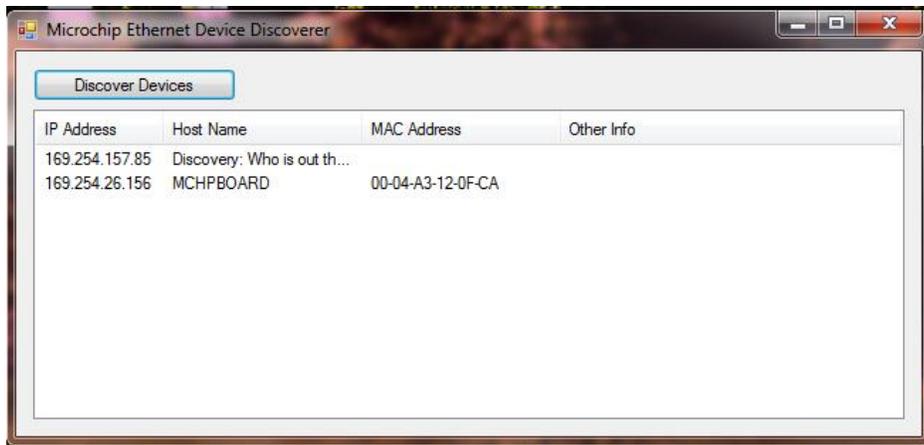
### 5.1 Manual de Usuario

Primeramente se conecta el cable de alimentación a la tarjeta y a la computadora, se observara que tres LED`s se encienden, dos en color verde y uno en color rojo (LED1), este último prende y apaga cada segundo para comprobar que la comunicación se ha establecido de manera correcta. Después se conecta el cable RJ-45 al puerto Ethernet de la PC y del microcontrolador. Figura 5.1



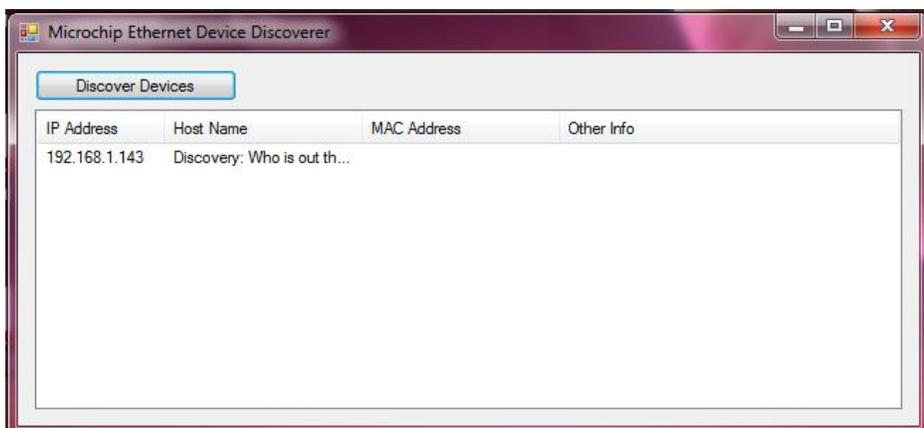
**Figura 5.1** Conexión de la tarjeta PIC32 a la PC.

Una vez que se ha conectado todo, se da doble clic a la aplicación *Microchip Ethernet Device Discoverer.exe*, la cual mostrara una ventana en donde indicará la dirección IP que tendrá la página web, para acceder a la página se deberá dar doble clic en la opción que tiene como Host Name MCHPBOARD. Figura 5.2



**Figura 5.2** Aplicación Microchip Ethernet Device Discoverer.exe con dirección IP generada.

Nota. Algunas veces cuando se abre esta aplicación no muestra la dirección IP asignada para la página Figura 5.3, entonces deberá darse un clic en el botón Discover Devices para generar la dirección IP.



**Figura 5.3** Aplicación Microchip Ethernet Device Discoverer.exe con dirección IP no generada.

Después de esto se abrirá la página mostrando la primera página inicio.html. Figura 5.4



Figura 5.4 Página inicio.html

Si se desea ir a la página en donde se mostrara el estado de la variables que se controlan y monitorean, solo se necesita dar clic en el menú en la sección que dice INICIAR SESIÓN, en esta página aparecerá el estado en tiempo real en el que se encuentran las variables, figura 5.5, en esta imagen se observa que la iluminación se ha encendido de manera local por medio de la botonera es por eso que al momento de entrar a la página el botón de iluminación está en color verde, también se muestra la temperatura en este caso 23° Centígrados.



Figura 5.5 Estado de variables a controlar y monitorear.

Para corroborar que la iluminación está encendida se observa que en la tarjeta esta prendido el LED2 de color naranja. Figura 5.6



**Figura 5.6** LED2 encendido indicando que el foco ha sido prendido.

Si se desea apagar el foco únicamente se necesita dar clic en el botón de la página y este se pondrá color gris, también se comprueba que el foco ha sido apagado porque en la tarjeta el LED2 aparecerá inactivo. Figura 5.7.



**Figura 5.7** LED2 apagado, indica que el foco está apagado.

En caso de que ocurra una fuga de gas, en la sección de gas en la página, el botón se pondrá color verde y la alarma se accionará, es decir también se pondrá en color verde el botón en el apartado de la alarma figura 5.8 - a. En la tarjeta se prenderá el LED3 para indicar que la alarma está encendida figura 5.8 - b.



- a) Alarma encendida por fuga de gas en la página web.



- b) Alarma encendida por fuga de gas en la tarjeta PIC32.

**Figura 5.8.** Fuga de gas en el sistema de monitoreo.

En la sección de temperatura se despliega el valor que indica el sensor y si este valor es mayor a los 70° Centígrados el botón de la alarma se pondrá en color verde, **figura 20 a**, y como ya se había mencionado anteriormente en la tarjeta se observará que le LED3 está encendido. **Figura 5.9 - b**.



a) Temperatura mayor a 70°C, alarma encendida.



b) Indicador de alarma encendida.

**Figura 5.9** Encendido de alarma debido al exceso de temperatura.

Cuando la alarma es activada por fuga de gas o elevación de temperatura, y se desea apagar es necesario que se haga manualmente, es decir, por medio de la botonera o la página web, pues aunque se hayan estabilizado los niveles de gas o temperatura esta permanecerá encendida. Figura 5.10.



**Figura 5.10** Alarma encendida.

En el espacio del ventilador cuando se presiona el botón ON/OFF el ventilador se enciende con una velocidad del 100%, figura 5.11, si se desea que gire a una velocidad menor únicamente necesita presionar alguno de los botones que se encuentran debajo de este botón. Figura 5.12 Para apagar el ventilador únicamente se vuelve a presionar el botón.



Figura 5.11 Ventilador encendido.



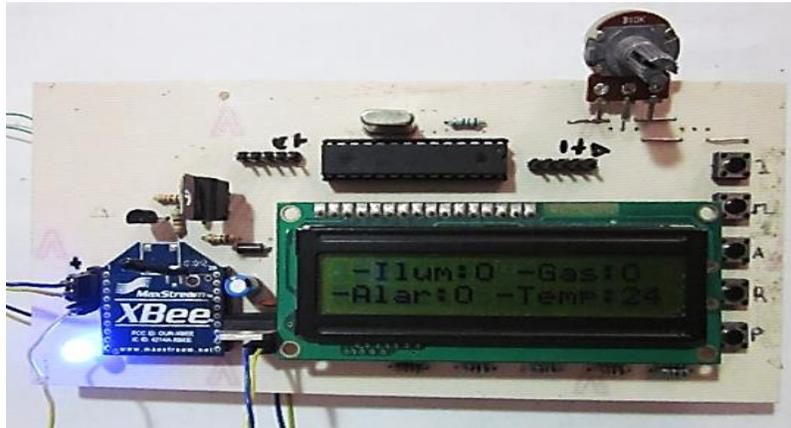
Figura 5.12 Ventilador con una velocidad del 100%.

La interfaz de usuario local nos permite controlar las diferentes interfaces sin la necesidad de entrar a la página web, su uso se dará cuando el usuario se encuentre en el hogar así evitara el uso de la computadora para manipular o saber el estado del sistema.

En la figura 5.13 de abajo vemos la interfaz con una botonera del lado izquierdo, cada botón tiene una función que permiten manipular el estado de las interfaces y en el centro hay una pantalla LCD en donde se despliega el estado del sistema.

La interfaz funciona de la siguiente manera, al oprimir el botón 1 una vez el foco enciende y en display se despliega un "1" que indica que esta encendido al volver a oprimir el

mismo botón el foco se apaga y en el display se muestra un "0" que indica que el foco se encuentra apagado.



**Figura 5.13** Interfaz de usuario local. En el display se muestra la información del sistema y del lado izquierdo se observa la botonera.

El botón 2 nos envía al modo de control de velocidad del motor como se ve en la figura 5.14 al estar en este modo podemos variar la velocidad; el potenciómetro variara de 0% al 100% el ciclo útil de la señal PWM y así aumentar o disminuir la velocidad, en el display se despliega el valor del ciclo útil, cuando se quiere salir de este modo y regresar a la pantalla principal se deberá oprimir el botón 4, el motor se mantendrá girando hasta que apaguemos el motor esto se hace poniendo el ciclo útil en 0%.



**Figura 5.14.** Interfaz en modo para variar la velocidad del motor.

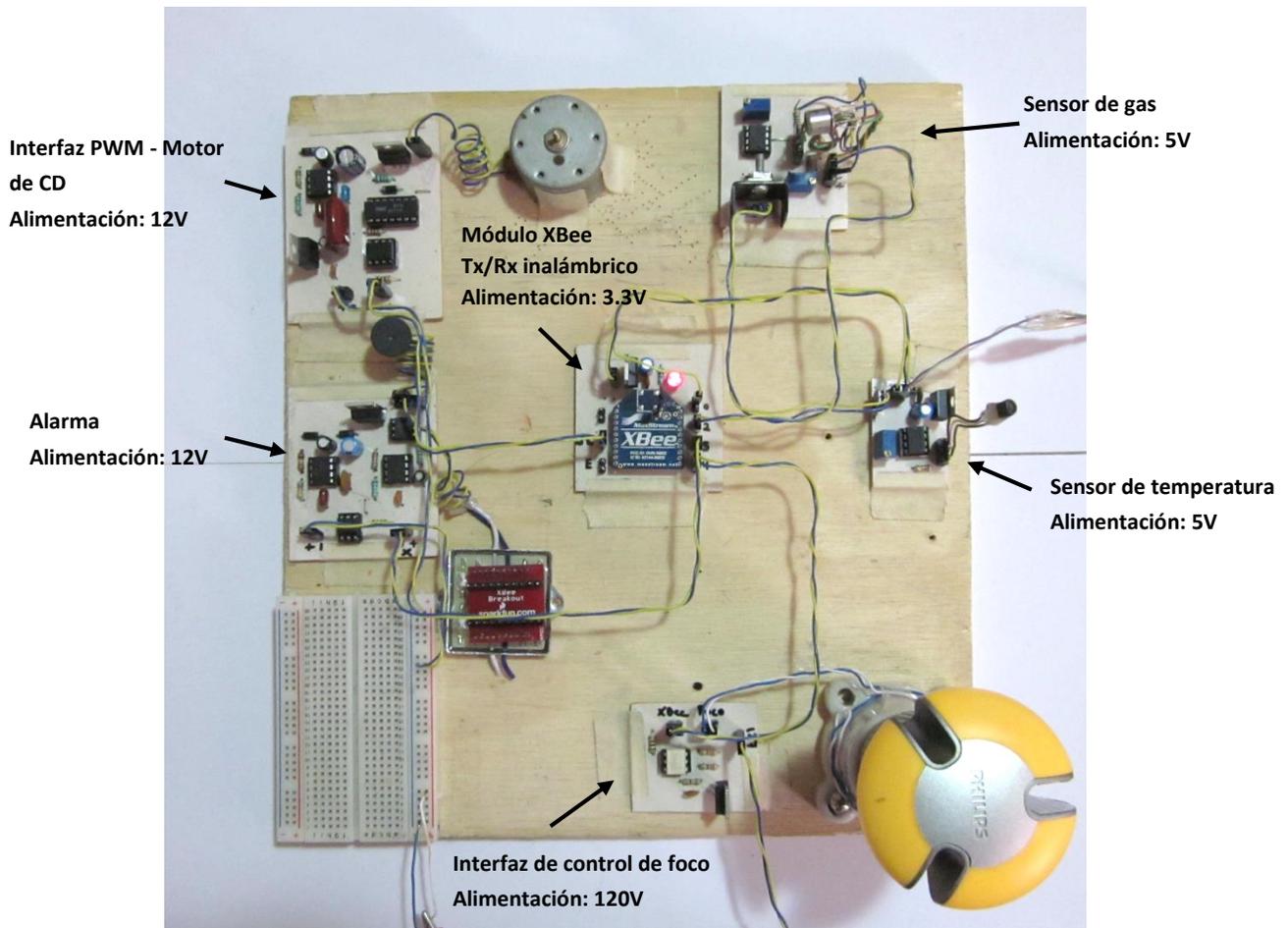
El botón 3 permite controlar la alarma y su funcionamiento es igual a la del encendido y apagado del foco, se desplegara en el display el estado de la misma.

Al activarse el sensor de gas se desplegara en el display el valor "1" indicando que el sensor ha detectado gas y se mantendrá un "0" en el display al no detectar gas. El valor

del sensor temperatura de igual manera se despliega en el display su valor esta dado en grados centígrados.

Cabe mencionar que la alarma está ligada a los sensores, cuando el sensor de gas se activa por la detección de gas o el sensor de temperatura registra temperaturas muy altas inmediatamente se activara la alarma alertando al usuario de que existe un riesgo. Toda la información desplegada en el LCD se actualiza cada segundo.

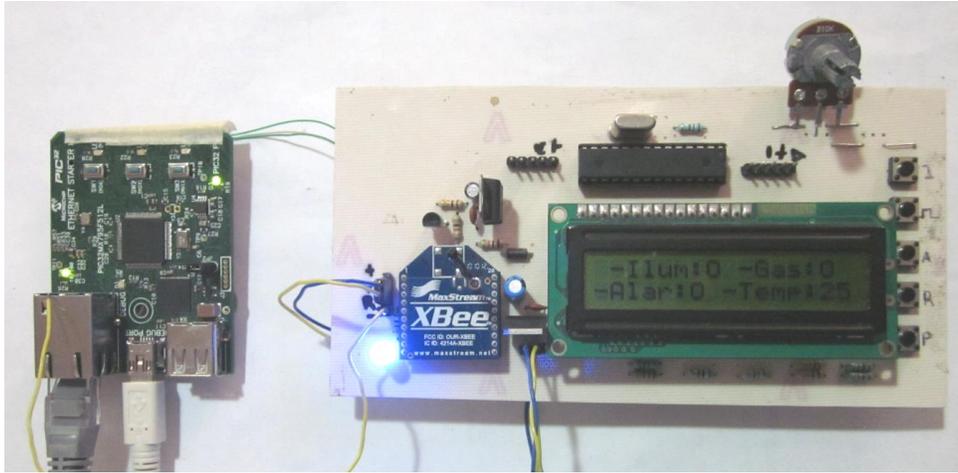
La conexión de todas las interfaces de control y monitoreo llevan su propia alimentación y todas ellas van conectadas al módulo inalámbrico XBee el cual será el encargado de recibir y ejecutar las acciones que el usuario requiera así como de enviar la información de los sensores tanto a la página web como a la interfaz de usuario, ver figura 5.15.



**Figura 5.15** Interfaces de monitoreo y control con su respectiva alimentación todas las interfaces van conectadas al módulo XBee inalámbrico.

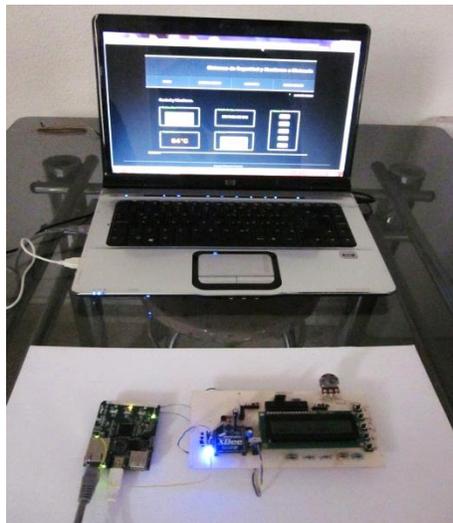
Como se ve en la figura 5.15 al estar alimentadas las interfaces y conectadas al módulo XBee es lo único que se requiere para su funcionamiento, el control de estas se dará de manera inalámbrica por medio de la página web o de la interfaz de usuario local.

El PIC 32 se conecta a la interfaz de usuario local así la información que deseemos enviar y recibir a través del XBee pasa por estos dos dispositivos; el PIC 32 aloja la página web a la cual se tiene acceso a través de una conexión Ethernet integrada en el PIC32.



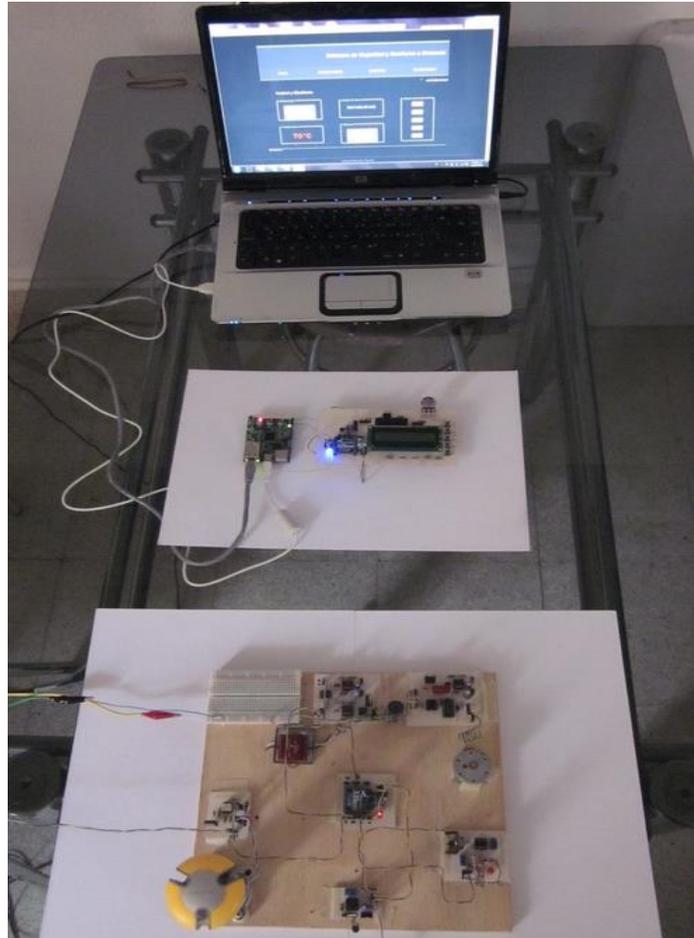
**Figura 5.16** Interfaz de usuario local conectado al PIC32, se usa un cable Ethernet para conectar el PIC32 a la PC o un a un Router.

El PIC32 se conecta directamente a la PC vía Ethernet como se ve en la figura 5.17, tanto la interfaz de usuario local como el PIC32 requieren una alimentación de 5V para funcionar lo que se requiere para poner en marcha el sistema.



**Figura 5.17.** Interfaz de usuario local y PIC32 conectados a la PC vía Ethernet.

En la figura 5.18 se observa todo el sistema conectado y funcionando, el PIC32 está conectado directamente a la computadora vía Ethernet, pero también se puede conectar a un router o al módem de nuestro proveedor de internet de esta manera se puede tener acceso al sistema de manera remota y sin la necesidad de conectarlo a la PC.



**Figura 5.18** Sistema completo conectado y funcionando. Interfaces de control y monitoreo con sus respectivas alimentaciones, interfaz de usuario local conectado al PIC32 y a su vez esté conectado a la PC vía Ethernet.

## 5.2 Mejoras

El proyecto puede ser extendido adaptando más interfaces como una cámara de video, sensores de presencia, actuadores de agua y gas, persianas y calefacción automática, etc. todo esto dependerá de las necesidades que se presenten en la vivienda partiendo del prototipo desarrollado.



**Figura 5.19** Mejoras que se pueden implementar al prototipo.

Para implementar las mejoras solo se requiere de una interfaz diseñada o comprada modificándola para que sea compatible con el sistema y realizar modificaciones mínimas en la programación que incluyan a la interfaz que se desea añadir.

## Conclusiones

La tendencia por mejorar la calidad de vida nos lleva investigar y diseñar un sistema domótico que proporcione seguridad y confort en el hogar dando como resultado un sistema totalmente funcional con capacidad de conexión a internet, con interfaces de control capaces de ejecutar instrucciones dadas por un usuario e interfaces de monitoreo que censan el medio obteniendo información que se usa para la seguridad.

El sistema se compone de hardware y software; por parte del hardware se encuentran diversos circuitos electrónicos como son módulos de comunicación inalámbrica llamados XBee de Digi International Inc., un microcontrolador PIC18 de Microchip para gestar la comunicación de los módulos inalámbricos así como de la interfaz de usuario que nos permite controlar y monitorear el sistema, un microcontrolador PIC32 de igual manera Microchip para proporcionar comunicación Ethernet y diseño de circuitos para sensores de gas y temperatura así como circuitos actuadores de control de iluminación, velocidad y alarma; por la parte de software los módulos XBee se configuraron de acuerdo a las necesidades del sistema usando el programa X-CTU que usa comandos AT cabe mencionar que los módulos vienen programados con el protocolo ZigBee, para la programación del PIC 32 se usó el compilador MPLab en el que se elaboró el programa para la comunicación Ethernet usando lenguaje C.

El sistema cumple con los objetivos planteados, nos permite controlar iluminación, velocidad de un motor y una alarma así como censar la temperatura y detectar la presencia de gas LP ya sea a través de la interfaz de usuario local o por la página web, el control es rápido después de dar la orden, los sensores responden inmediatamente al cambio de las variables que censan lo cual resulta de gran utilidad para la seguridad, la comunicación inalámbrica le da una gran adaptabilidad al sistema ya que evita de un cableado que complicaría su conexión y la página web resulta muy útil por el control remoto que nos permite dando así versatilidad al y eficacia en el sistema.

El sistema puede tener mayores alcances al adaptar más interfaces compatibles y realizando las modificaciones necesarias al software, todo esto es posible ya que se usan los protocolos de comunicaciones más usadas y circuitos integrados comunes y de bajo costo, todo dependerá de las necesidades que requiera el recinto y del desarrollo que se realice.



## Anexo B. Interrupción de la UART para procesar las tramas en la interfaz local.

```
#int_rda // Etiqueta de salto para la interrupción
void serial_isr() { // Función de la Interrupción UART
    rcvchar=0x00; // Inicializo caracter recibido
    if(kbhit()){ // En espera, Si hay algo pendiente de recibir
        rcvchar=getc(); // Guardo el byte recibido
        cbuff[xbuff++]=rcvchar; // Añade caracter recibido al Buffer
        if(xbuff==1 && rcvchar!=0x7E) { //Condicional para revisar y saber el fin de la trama
            int i;
            for(i=0;i<lenbuff;i++){ // Bucle que pone a 0 todos los elementos del buffer
                cbuff[ i ]=0x00; }
                xbuff=0x00
                sum=0;
                l=0; }
            if(xbuff==3) { //Condicional para contar bytes de trama
                l=rcvchar+4; }
            if(xbuff>3) {
                sum=sum+rcvchar; } //Suma de los bytes para Checksum
            if(xbuff==1) {
                if(sum==0xFF){
                    procesa_comando(); } //Función para procesar la trama
                else {
                    inicbuff(); }
                } } }
        } } }
```

## Glosario

**AJAX (Asynchronous JavaScript And XML).** Es una técnica de desarrollo web destinada a crear aplicaciones interactivas que se ejecutan en el navegador del cliente mientras se mantiene la conexión asíncrona con el servidor. Permite actualizar la información de la web sin necesidad de refrescar la pantalla, y es soportado por la mayoría de los navegadores.

**ADC - Convertidor Analógico a Digital,** este se encarga de llevar a cabo la conversión de una señal analógica a un equivalente binario.

**Funciones Callback.** Se encuentran en el archivo CustomHTTPApp.c, dentro del proyecto del TCP/IP Stack. Son las funciones que nos permiten, por ejemplo, encender un Led, monitorear la temperatura o controlar un motor a través de Ethernet. Veremos que necesariamente, debemos programarlas en el Stack y vincularlas a través de variables dinámicas a la interfaz web.

**HTTPPrint.h.** Es importante tener en cuenta que, al agregar o quitar variables de nuestra web (por medio del status.xml), se modifica el archivo HTTPPrint.h, por lo que hay que volver a compilar el Stack antes de cargar la aplicación en el micro. Es recomendable seguir la secuencia mencionada para modificar, compilar y, finalmente, cargar el Stack en el PIC.

**PCB.** Un circuito impreso (Printed Circuit Board), es una tarjeta o placa utilizada para realizar el montaje de los distintos elementos electrónicos que conforman el circuito y las interconexiones eléctricas entre ellos.

**PWM.** Modulación por ancho de pulsos (PWM, Pulse Width Modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (senoidal o cuadrada), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

**RS-232.** Es un protocolo o estándar mundial que rige los parámetros de uno de los modos de comunicación serial. Por medio de este protocolo se estandarizan las velocidades de transferencia de datos, la forma de control que utiliza dicha transferencia, los niveles de voltajes utilizados, el tipo de cable permitido, las distancias entre equipos, los conectores, etc.

Además de las líneas de transmisión (Tx) y recepción (Rx), las comunicaciones seriales

poseen otras líneas de control de flujo (*Hands-hake*), donde su uso es opcional dependiendo del dispositivo a conectar.

**TTL.** Es la sigla en inglés de *transistor-transistor logic*, es decir, "lógica transistor a transistor". Es una familia lógica o lo que es lo mismo, una tecnología de construcción de circuitos electrónicos digitales. En los componentes fabricados con tecnología TTL los elementos de entrada y salida del dispositivo son transistores bipolares.

**UART.** Circuito que se encarga de la comunicación serial (Universal Asynchronous Receiver-Transmitter) su propósito es convertir los datos recibidos de un bus en formato paralelo, a un formato serie que será utilizado en la transmisión hacia el exterior. También realiza el proceso contrario: transformar los datos serie recibidos del exterior en un formato paralelo entendible por el bus.

**XBee.** Módulos integrados inalámbricos fabricados especialmente para funcionar con el estándar IEEE 802.15.4 para redes punto a multipunto

**ZigBee.** Es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (*wireless personal area network*, WPAN). Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías.

## Bibliografía

*Introducción al análisis de circuitos*

Pearson

Robert L. Boylestad, Louis Nashelsky

2004

1230 pags.

*Teoría de circuitos y dispositivos electrónicos*

Pearson

Robert L. Boylestad, Louis Nashelsky

2003

1032 pags.

*Electrónica de potencia*

Pearson

Muhammad H. Rashid

2005

*Compilador CCS y simulador PROTEUS para microcontroladores PIC*

Alfaomega

Eduardo García Breijo

2008

263 pags.

*Redes de Computadoras*

UOC

José María Barceló Ordinas, Jordi Íñigo Griera, Ramón Martí Escalé, Enric Peig Olivé, Xavier

Perramon Tornil

2004

69 pags.

*Redes de Computadoras*

Pearson

Andrew S. Tanenbaum

Cuarta Edición, 2003

37 pags.