

VALIDACIÓN DE LA TECNOLOGÍA ZIGBEE DE MICROCHIP MEDIANTE
LA IMPLEMENTACIÓN DE UNA RED INALÁMBRICA DE SENSORES

CAMILO ANDRÉS MONTES MOSQUERA
COD: 2005102223

OMAR ANDRÉS ROJAS PALOMA
COD: 2004200826

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA – HUILA
2010

VALIDACIÓN DE LA TECNOLOGÍA ZIGBEE DE MICROCHIP MEDIANTE
LA IMPLEMENTACIÓN DE UNA RED INALÁMBRICA DE SENSORES

CAMILO ANDRÉS MONTES MOSQUERA
COD: 2005102223

OMAR ANDRÉS ROJAS PALOMA
COD: 2004200826

Proyecto de grado presentado para optar
al título de Ingeniero Electrónico

Director
JAVIER RUBIO
Ingeniero Electrónico.

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA, HUILA
2010

Nota de aceptación:

Firma del presidente del jurado

Firma del primer jurado

Firma del segundo jurado

Neiva, 22 de Septiembre de 2010

A Dios;
A mis padres Hemil Montes y Leidy Mosquera;
A mis hermanos Julieth y Leonardo;
Y a todas las personas que creyeron en mí este tiempo.

CAMILO ANDRÉS MONTES MOSQUERA

A Dios primero que todo;
A mis padres Luis Rojas y Adelaida Palomá;
A mis dos hermanas Luisa Rojas y Astrid Rojas;
Y a todas esas personas que me apoyaron en el transcurso de la carrera;

OMAR ANDRÉS ROJAS PALOMÁ

AGRADECIMIENTOS

Queremos agradecer primero que todo a Dios por dejarnos vivir todo este tiempo y lograr llegar a esta instancia, para poder culminar satisfactoriamente nuestros estudios.

A nuestras familias, que fueron un gran apoyo en el transcurso de nuestra carrera, que estuvieron ahí brindándonos su colaboración y su ánimo para que no desistiéramos en esos momentos que no fueron tan gratos.

A nuestros compañeros con los cuales compartimos traspasadas y muchas horas de estudio y proyectos con los cuales afianzamos nuestro amor por esta carrera, tan interesante y de gran desarrollo en el futuro.

A todos los ingenieros que de una u otra manera, nos brindaron un poco de su tiempo, proporcionándonos sus conocimientos, para que conociéramos muchos de esos aspectos importantes que tiene la carrera.

Al ingeniero Javier Rubio por dedicar parte de su tiempo en la dirección de este proyecto.

Y a todas esas personas que directamente o indirectamente colaboraron para la creación y desarrollo de este proyecto.

CONTENIDO

	Pág.
INTRODUCCIÓN	17
CAPÍTULO 1	
ANÁLISIS DEL PROTOCOLO ZIGBEE / IEEE 802.15.4	18
1.1 ZIGBEE	18
1.2 IEEE 802.15.4	18
1.3 CARACTERÍSTICA DEL PROTOCOLO ZIGBEE / IEEE 802.15.4	19
1.4 TIPOS DE DISPOSITIVOS ZIGBEE	19
1.5 TOPOLOGÍAS	20
1.5.1 Configuración de red en Estrella	20
1.5.2 Configuración de red en Árbol	20
1.5.3 Configuración de red en Malla	21
1.6 ARQUITECTURA DE LA PILA DE PROTOCOLOS ZIGBEE DE MICROCHIP	22
1.6.1 Capa Física	22
1.6.2 Capa MAC	22
1.6.3 Capa de Red	22
1.6.4 Capa de Aplicación	22
1.7 ARQUITECTURA DE LAS TRAMAS DE LAS DIFERENTES CAPAS DE LA PILA DE PROTOCOLO ZIGBEE.	24
1.7.1 Trama de la capa PHY	24
1.7.2 Trama de la capa MAC	24
1.7.2.1 Trama de datos	25
1.7.2.2 Trama ACK (Acknowledgment Frame)	25
1.7.2.3 Trama de comandos	25
1.7.2.4 Trama de beacon o balizas	26
1.7.2.5 Superframes (Supertramas)	26
1.7.3 PDU (Protocol Data Unit) NWK.	27
1.7.4 PDU (Protocol Data Unit) APS.	27
CAPÍTULO 2	
ADQUISICIÓN Y ANALISIS DE HARDWARE	28
2.1 REQUERIMIENTOS PARA HARDWARE	28
2.2 SELECCIÓN DEL HARDWARE	28
2.2.1 Kit de desarrollo picdem z	28

2.2.1.1 Motherboard picdem z	29
2.2.1.2 Mrf24j40ma picdem z 2.4 GHz RF board	31
2.2.1.3 Microcontrolador pic18f4620	32
2.2.1.4 Analizador de redes inalámbricas ZENA	33
2.2.1.5 Sensor de temperatura con interfaz SPI	34
2.3 ANALISIS DE COSTOS	34

CAPÍTULO 3

ANÁLISIS DEL SOFTWARE	36
3.1 PILA DE PROTOCOLOS DE LA TECNOLOGIA ZIGBEE DE MICROCHIP	36
3.1.1 Funcionamiento	36
3.2 ANALIZADOR INALÁMBRICO DE REDES ZENA	38
3.2.1 Análisis de los paquetes de datos utilizando el analizador ZENA	39
3.3 ORGANIZACIÓN DE ARCHIVOS FUENTE	42
3.4 APLICACIONES DE DEMOSTRACIÓN	42
3.4.1 Característica de la aplicación de demostración	43
3.5 DESCRIPCIÓN DE LOS ARCHIVOS FUENTES.	43
3.6 USO DE LA PILA DE PROTOCOLOS DE LA TECNOLOGIA ZIGBEE	46
3.6.1 Interfaz con la pila de protocolos de la tecnología zigbee de Microchip	46
3.6.2 Formando o uniéndose a una red	49
3.6.3 Captación de mensajes	49
3.6.4 Envío de mensajes	50
3.6.5 Solicitar y recibir datos en un dispositivo RFD	52
3.6.6 Transmisión segura	54

CAPÍTULO 4

PRUEBAS DE LA VALIDACIÓN DE LA TECNOLOGÍA ZIGBEE DE MICROCHIP	57
4.1 COMPROBACIÓN DE LA TOPOLOGÍA PUNTO A PUNTO	57
4.2 COMPROBACIÓN DE LA TOPOLOGÍA ÁRBOL	59
4.3 COMPROBACIÓN DE LA TOPOLOGÍA ESTRELLA	61
4.4 COMPROBACIÓN DE LA TOPOLOGÍA MALLA	63
4.5 PRUEBA DE DISTANCIA	65
4.6 PRUEBAS DE POTENCIA VS DISTANCIA	66
4.7 FOTOS DE PRUEBA (POTENCIA VS DISTANCIA)	71

CAPÍTULO 5	
CREACIÓN E IMPLEMENTACIÓN DEL SOFTWARE	72
5.1 SENSOR ELECTRÓNICO	72
5.2 DESCRIPCIÓN DEL SISTEMA IMPLEMENTADO	72
5.3 DESARROLLO DEL SOFTWARE EN EL ENTORNO GRAFICO DE PROGRAMACIÓN LABVIEW	74
5.3.1 Explicación por bloques de los elementos que se trabajaron en el ambiente de desarrollo grafico Labview, para la realización del software.	75
5.4 DESCRIPCIÓN DE LOS NODOS ZIGBEE	76
5.4.1 Coordinador	76
5.4.2 Router	76
5.5 CÓDIGO DE ADQUISICIÓN DE TEMPERATURA	80
CAPÍTULO 6	
CONCLUSIONES	82
CAPÍTULO 7	
RECOMENDACIONES	83
BIBLIOGRAFIA	84

ÍNDICE DE FIGURAS

CAPÍTULO 1

Figura 1.1 Topología en Estrella	20
Figura 1.2 Topología en Árbol	21
Figura 1.3 Topología en Malla	21
Figura 1.4 Arquitectura de la pila de protocolos de la tecnología ZigBee de Microchip	23
Figura 1.5 Estructura de la PDU PHY	24
Figura 1.6 Estructura general de la trama MAC	24
Figura 1.7 Estructura de Trama de Datos	25
Figura 1.8 Estructura de Trama ACK	25
Figura 1.9 Estructura de Trama de comandos	25
Figura 1.10 Estructura de Trama de Beacon	26
Figura 1.11 Estructura de la supertrama	26
Figura 1.12 Estructura PDU NWK	27
Figura 1.13 Estructura PDU APS	27

CAPÍTULO 2

Figura 2.1 Kit PICDEM Z	28
Figura 2.2 Elementos de la tarjeta madre	29
Figura 2.3 Transceiver MRF24j40MA	31
Figura 2.4 Diagrama de Bloques Del transceiver MRF24J40MA	32
Figura 2.5 Pines PIC 18f4620	32
Figura 2.6 Analizador Zena	33
Figura 2.7 Sensor de temperatura TC77	34

CAPÍTULO 3

Figura 3.1 Mensajes de asociación	39
Figura 3.2 Niveles de reconocimiento (acknowledgment)	41
Figura 3.3 Estructura Básica de la Aplicación.	47
Figura 3.4 Flujo general de una aplicación basada en la pila Zigbee de Microchip.	48
Figura 3.5 Recepción de Mensajes	50
Figura 3.6 Enviar un Mensaje	52
Figura 3.7 Solicitar y recibir datos en un dispositivo RFD.	53
Figura 3.8 Enviar un mensaje seguro.	56

CAPÍTULO 4

Figura 4.1 Topología punto a punto	57
Figura 4.2 Red punto a punto (Zena)	57
Figura 4.3 Topología Árbol	59
Figura 4.4 Red Árbol (Zena)	59
Figura 4.5 Trama red Árbol (Zena)	60
Figura 4.6 Topología estrella	61
Figura 4.7 Red Estrella (Zena)	61
Figura 4.8 Trama red Estrella (Zena)	62
Figura 4.9 Topología malla	63
Figura 4.10 Red malla (Zena)	64
Figura 4.11 Trama red malla (Zena)	64
Figura 4.12 Led indicador de la tarjeta madre	65
Figura 4.13 Máxima distancia alcanzada	66
Figura 4.14 RSSI vs. DISTANCIA	67
Figura 4.15 Gráfica típica de lectura RSSI VS Potencia de entrada	68
Figura 4.16 Potencia (dbm) vs. Distancia	69
Figura 4.17 Tramas a diferentes distancias	70
Figura 4.18 Prueba a 30 metros	71
Figura 4.19 Prueba a 50 metros	71
Figura 4.20 Prueba a 80 metros	71
Figura 4.21 Prueba a 100 metros	71

CAPÍTULO 5

Figura 5.1 Red implementada.	73
Figura 5.2 Monitoreo de temperatura en la red Zigbee (panel).	74
Figura 5.3 Monitoreo de temperatura en la red Zigbee (esquemático).	75
Figura 5.4 Diagrama de flujo simplificado del coordinador.	77
Figura 5.5 Diagrama de flujo simplificado del Primer Router.	78
Figura 5.6 Diagrama de flujo simplificado del Segundo Router	79

ÍNDICE DE CUADROS

CAPÍTULO 2

Cuadro 2.1 Características del PIC 18f4620	33
Cuadro 2.2 Costos de un nodo.	35
Cuadro 2.3 Costos de toda la implementación.	35
Cuadro 2.4 Costos de importación: importación y transporte.	35

CAPÍTULO 3

Cuadro 3.1 Lista de primitivas de la tecnología Zigbee.	37
Cuadro 3.2 Estructura de los directorios de los archivos fuentes	42
Cuadro 3.3 Archivos fuente del Stack de Microchip en el Subdirectorio <i>ZigBeeStack</i>	43
Cuadro 3.4 Archivos Fuente comunes de Microchip en el Subdirectorio <i>Common</i>	44
Cuadro 3.5 Subdirectorio de demostración de Coordinadores de la tecnología ZigBee en <i>DemoCoordinator</i> .	44
Cuadro 3.6 Subdirectorio de demostración de Routers de la tecnología ZigBee en <i>DemoRouter</i> .	45
Cuadro 3.7 Subdirectorio de demostración de End Device de la tecnología ZigBee en <i>DemoRFD</i>	45
Cuadro 3.8 Servicios de seguridad del Protocolo Zigbee	55
Cuadro 3.9 Formato de cabecera de seguridad auxiliar del Protocolo Zigbee.	55

CAPÍTULO 4

Cuadro 4.1 Características del Led indicador de la tarjeta madre	65
Cuadro 4.2 Datos RSSI adquiridos por el zena	67
Cuadro 4.3 Datos RSSI utilizando la ec. $P = \text{RSSI_VAL} + \text{RSSI_OFFSET}$	69

ÍNDICE DE ANEXOS

ANEXO A. MANUAL DE USUARIO.	85
ANEXO B. CIRCUITO DE TARJETA MADRE	103

GLOSARIO

ACK (ACKNOWLEDGMENT FRAME): Es una trama de reconocimiento la cual provee una *realimentación* desde el **receptor al emisor** para informar que el paquete ha sido recibido correctamente.

ATRIBUTO (ATTRIBUTE): Es una entidad de datos que representa una cantidad física o un estado.

BEACON (BALIZA): Es un mensaje diseñado intencionalmente para llamar la atención de todos los dispositivos que se encuentren en la red.

CLÚSTER (CLUSTER): Están formados por uno o más atributos. Los clusters se diferencian entre ellos por un identificador, el cual está asociado con el flujo de datos saliente o entrante del dispositivo.

DESCRIPCION DEL DISPOSITIVO (DEVICE DESCRIPTION): Es la delineación de la función de dispositivos específicos dentro de un segmento del profile. Por ejemplo en el perfil de “control de iluminación”, lámparas, switches y otros dispositivos tienen cada uno su descriptor.

FFD: Dispositivo con funciones reducidas.

FCS (FRAME CHECK SEQUENCE): Es la verificación de secuencia de trama y sirve para verificar la integridad de las tramas MAC.

MAC: Es la capa de Control de Acceso al medio.

MIC: Es un código de integridad de mensajes que se utiliza para darle seguridad a la red.

MULTICAST: Es el envío de la información en una red a múltiples destinos simultáneamente.

NCD: Es una ventana que utiliza el software Zena para la visualización de la configuración de red.

OBJETO (ENDPOINT): Es un elemento particular dentro de un dispositivo. Cada dispositivo Zigbee dispone de una única dirección de red, por lo tanto se requiere de una dirección de “endpoint” que diferencie cada componente (interruptores, lámparas, sensores, etc).

OSI: El **modelo de referencia de Interconexión de Sistemas Abiertos** es un marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones.

PAN: Es una Red de Área Personal.

PERFIL (PROFILE): Es un acuerdo en los mensajes, su formato y acciones que permiten a las aplicaciones residir en dispositivos separados y trabajar en forma cooperativa. Por ejemplo, un interruptor en un nodo puede comunicarse con una lámpara en otro nodo. Juntos, son parte de un perfil para “control de iluminación.

PRIMITIVAS: Son estados especificados en el estándar Zigbee/IEEE 802.15.4, que permiten el acceso a los servicios más importantes para realizar procesos como formación de la red, transmisión y recepción de comandos y datos.

RFD: Dispositivo con funciones completas.

RSSI: Es la abreviatura en inglés de *Receive Signal Strength Indication*, Indicador de fuerza de señal de recepción. Este término se utiliza para medir el nivel de fuerza de las señales recibidas en las redes inalámbricas.

SPI (SERIAL PERIPHERAL INTERFACE): Es una interfaz periférica serial la cual es un bus estándar de comunicaciones.

STRING: Es una cadena de caracteres que pueden ser letras, números u otros signos o símbolos.

UNICAST: es el envío de información desde un único emisor a un único receptor.

RESUMEN

El estándar ZigBee es una tecnología creada en el 2004 pero ha tomado furor en estos tiempos gracias a la simplicidad de sus redes y a su poco consumo de energía.

Esta tecnología es regida por el estándar IEEE 802.15.4 el cual está creado para redes de área personal y es administrada por la ZigBee Alliance, la cual es una asociación de empresas que tomó como base el estándar IEEE 802.15.4 para mejorarlo, expandirlo y así darle mayor flexibilidad y movilidad a las redes.

Gracias a este trabajo se logró acabar un poco con el desconocimiento de la tecnología ZigBee en la universidad y en la ciudad, también se contribuye con un material bibliográfico el cual servirá como ayuda a aquellas personas especialmente los estudiantes del programa de Ingeniería Electrónica pertenecientes a la Universidad Surcolombiana para que puedan estar en la capacidad de implementar y desarrollar aplicaciones con esta tecnología inalámbrica teniendo a MICROCHIP como la marca mas utilizada e implementada en el programa, desde hace ya mucho tiempo.

Los resultados hallados en este trabajo fueron los siguientes:

- la máxima distancia que se podría lograr en un red punto a punto la cual fue 120 mts
- la comprobación de las diferentes topologías soportadas por esta tecnología como lo es la árbol, estrella y malla
- las potencias en una red punto a punto a diferentes distancias.

ABSTRACT

The ZigBee standard is a technology invented in 2004 but has taken rage these days thanks to the simplicity of their networks and their low power consumption.

The technology is governed by the IEEE 802.15.4 standard which is designed for personal area networks and is managed by the ZigBee Alliance, which is an association of companies that took based on the IEEE 802.15.4 standard to improve, expand and and give greater flexibility and mobility to networks.

Thanks to this work is to finish a little ignorance of ZigBee technology in the university and the city also contributes a bibliography which will serve as an aid to those people especially students of Electronics Engineering program affiliated with the University Surcolombiana so they can be in the ability to implement and develop wireless applications with this technology having a microchip as the brand most used and implemented in the program, from long ago.

The findings of this study were as follows:

- The maximum distance that could be achieved in a peer to peer network which was 120 meters
- Verification of the different topologies supported by this technology such as the tree, star and mesh
- Powers in a peer to peer network at different distances.

INTRODUCCIÓN

En la actualidad se han diseñado e implementado gran cantidad de redes inalámbricas en el mundo, entre ellas se encuentra la tecnología Zigbee la cual se caracteriza por ser una red creada para el uso de sensores ya que maneja una tasa de envío de datos baja.

Esta investigación aborda la tecnología inalámbrica ZigBee, basada en el estándar 802.15.4 que por su poca introducción al mercado **Colombiano**, no es muy conocida a pesar de llevar bastante tiempo creada.

Este proyecto está elaborado en siete partes fundamentales; las cuales son: la introducción a la tecnología ZigBee, el análisis y adquisición del hardware, análisis del software, pruebas y resultados de la validación, creación e implementación de la tecnología ZigBee en los campos de la industria, conclusiones y recomendaciones y por último un pequeño manual para la implementación de la tecnología ZigBee.

En el primer capítulo se realizó un análisis del estándar IEEE 802.15.4 y una introducción a la tecnología ZigBee en el cual se hizo una descripción de este protocolo, sus aplicaciones, la estructura de la pila de protocolos y la estructura de las tramas para el envío de paquetes.

En el capítulo dos se realizó el análisis y adquisición del hardware, el cual contiene todo lo que tiene que ver con la compra y adquisición de los diferentes elementos a utilizar para poder realizar la validación de la tecnología.

En el capítulo tres se realizó el análisis del software, el cual contiene todo lo que tiene que ver con las herramientas de programación y el programa que se utilizó para poder crear la visualización de la información que emiten los sensores.

En el capítulo cuatro se realizaron las pruebas y resultados de la validación de esta tecnología, como por ejemplo, las distancias que se pueden alcanzar entre una red punto a punto o una red en malla, el funcionamiento de la red y el monitoreo de paquetes de datos que fluyan por ésta.

En el capítulo cinco se realizó la pequeña implementación de la tecnología ZigBee en una aplicación concreta como lo es la adquisición sensorial de datos de gran importancia para el futuro.

En los capítulos seis y siete se realizaron las conclusiones y recomendaciones que se deben tener en cuenta al momento de su implementación y puesta en funcionamiento.

En el anexo se encuentra el manual el cual muestra todas las características y procedimientos para crear e implementar una red inalámbrica ZigBee.

CAPITULO 1

ANÁLISIS DEL PROTOCOLO ZIGBEE / IEEE 802.15.4

1.1 ZIGBEE

Zigbee es un protocolo de red inalámbrica la cual está diseñada para el manejo de sensores con baja velocidad de datos y control de redes.

En esta tecnología se pueden descubrir gran variedad de aplicaciones las cuales encontramos muy fácilmente en la industria o en nuestros hogares como son:

- Medición de distancias, redes de automatización, periféricos para PC y en nuestros hogares la parte de domótica.

1.2 IEEE 802.15.4

La tecnología Zigbee se basa en el estándar IEEE 802.15.4 referente a la capa física (PHY) y a la capa de acceso al medio (MAC).

Este estándar define tres bandas de frecuencia para su operación, las cuales son:

- 868 MHz Está disponible en la mayoría de **países europeos**.
- 915 MHz Está disponible en **Norte América**, Nueva Zelanda, Australia.
- 2.4 GHz Está disponible en **la mayoría de países del mundo**.

Cada banda de frecuencia ofrece un arreglo de número de canales los cuales son:

- 868 MHz ofrece 1 canal (0).
- 915 MHz ofrece 10 canales (1-10).
- 2.4 GHz ofrece 16 canales (11-26).

Las tasas de bits que se manejan en la red están determinadas por las frecuencias de operación, estas son:

- 868 MHz ofrece una tasa de datos hasta 20 Kbps.
- 915 MHz ofrece una tasa de datos hasta 40 Kbps.
- 2.4 GHz ofrece una tasa de datos hasta 250 Kbps.

1.3 CARACTERÍSTICAS DEL PROTOCOLO ZIGBEE / IEEE 802.15.4.

El protocolo ZigBee/ IEEE 802.15.4 ofrece las siguientes características:

- Estándar para comunicación de datos a corto alcance.
- Bajo consumo de energía.
- Alcance de hasta 75 metros.
- Hasta 65,536 nodos por red, 255 por subred.
- Incorpora algoritmos de seguridad: de encriptación y de control de acceso.
- Basado la especificación IEEE 802.15.4 para redes PAN.
- Bajo costo del mantenimiento de la red.
- Utiliza bandas de 2,4 GHz, 868 MHz (Europa) y 915 MHz (EEUU).
- Velocidad de transmisión 20 a 250 kbps.
- Topología Maestro/Esclavo.

1.4 TIPOS DE DISPOSITIVOS ZIGBEE

Los tipos de dispositivos que trabaja la tecnología zigbee son 3, los cuales son:

- Coordinador.
- Router.
- Dispositivo final.

Estos dispositivos están regidos por las características del estándar IEEE 802.15.4 en los cuales se manejan dos tipos de dispositivos:

- Dispositivo con funciones completas (FFD).
- Dispositivo con funciones reducidas (RFD).

Las funciones que cumplen los dispositivos Zigbee son:

- Coordinador → Es un dispositivo FFD → * Se necesita uno por red.
* Se encarga de la forma de red.
* Asigna las direcciones de red.
- Router → Es un dispositivo FFD → * Es opcional.
* Amplia el rango físico de la red.
* Permite a más nodos unirse a la red.
* Puede realizar funciones de control.

- Dispositivo Final → Es un dispositivo RFD o FFD → * Realiza el seguimiento y/o funciones de control.

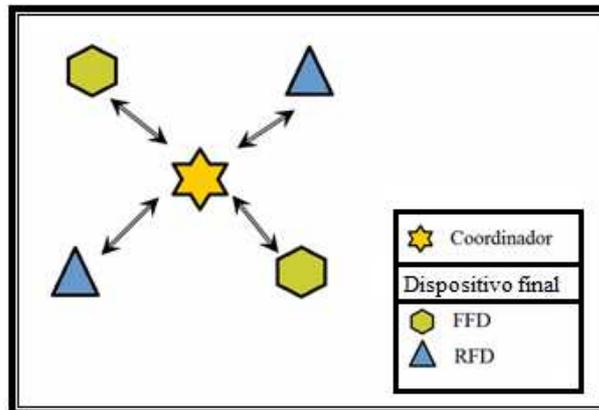
1.5 TOPOLOGÍAS

ZigBee permite implementar tres configuraciones básicas utilizando los dispositivos antes definidos.

1.5.1 Configuración de red en estrella

- Consiste en un protocolo Zigbee y un nodo Coordinador unido a más dispositivos finales, los cuales pueden ser RFD o FFD.
- Todos los dispositivos finales solo se comunican con el Coordinador.
- Si un dispositivo final necesita enviar un dato a otro dispositivo final, envía sus datos al Coordinador y luego éste los envía a su destinatario.

Figura 1.1 Topología en estrella

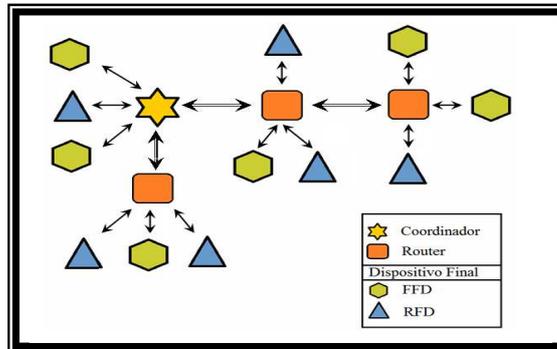


Fuente: <http://ww1.microchip.com/downloads/en/AppNotes/ZigBee2006%20Application%20Note%20AN1232A.pdf>

1.5.2 Configuración de red en árbol

- En esta configuración los dispositivos finales pueden unirse al Coordinador o al Router del protocolo Zigbee.
- Ingresando Routers en esta configuración de red, un dispositivo final no necesita estar a un rango determinado de radio del coordinador.
- Todos los mensajes en una red de topología árbol serán enrutados o distribuidos a lo largo del árbol.

Figura 1.2 Topología en árbol

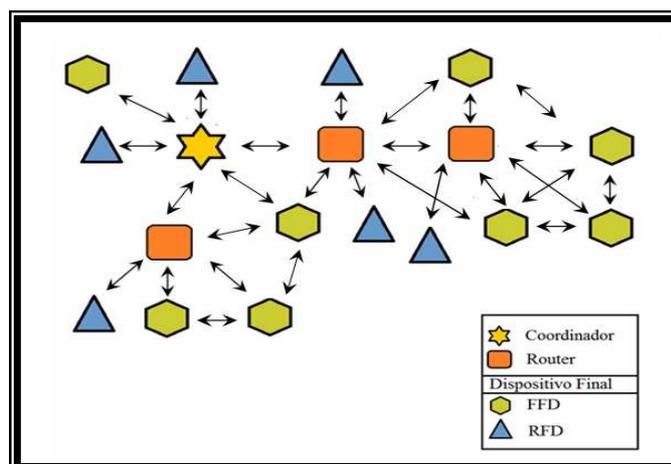


Fuente: <http://ww1.microchip.com/downloads/en/AppNotes/ZigBee2006%20Application%20Note%20AN1232A.pdf>

1.5.3 Configuración de red en malla

- Una red en malla es similar a la red árbol, excepto que los dispositivos FFDs pueden enrutar mensajes directamente con otros dispositivos FFDs, en lugar de seguir la estructura del árbol.
- Los dispositivos RFD no pueden interactuar con todos los dispositivos que tenga a su alrededor, si no únicamente con el dispositivo de mayor rango que tenga cerca.
- La ventaja de esta topología es que puede reducirse la latencia e incrementarse la confiabilidad.

Figura 1.3 Topología en malla



Fuente: <http://ww1.microchip.com/downloads/en/AppNotes/ZigBee2006%20Application%20Note%20AN1232A.pdf>

1.6 ARQUITECTURA DE LA PILA DE PROTOCOLOS ZIGBEE DE MICROCHIP

La arquitectura de la pila de protocolos Zigbee se fundamenta en modelo de referencia de Interconexión de Sistemas Abiertos (OSI), en la cual toma aquellas capas que son necesarias para la perfecta funcionalidad de la red Zigbee; estas capas son:

- Capa Física
- Capa MAC
- Capa de Red
- Capa de Aplicación

1.6.1 Capa física

Las principales funcionalidades son:

- Detección de energía del receptor.
- Indicación de calidad del enlace.
- Evaluación del estado del canal.
- Activación y desactivación del transceiver.
- Transmisión y recepción de datos.

1.6.2 Capa MAC

Las principales funcionalidades y responsabilidades de la capa MAC son:

- Transmisión de tramas, sincronización y provisión de un mecanismo de transmisión confiable.
- Asociación y disociación de dispositivos en la red.

1.6.3 Capa de red

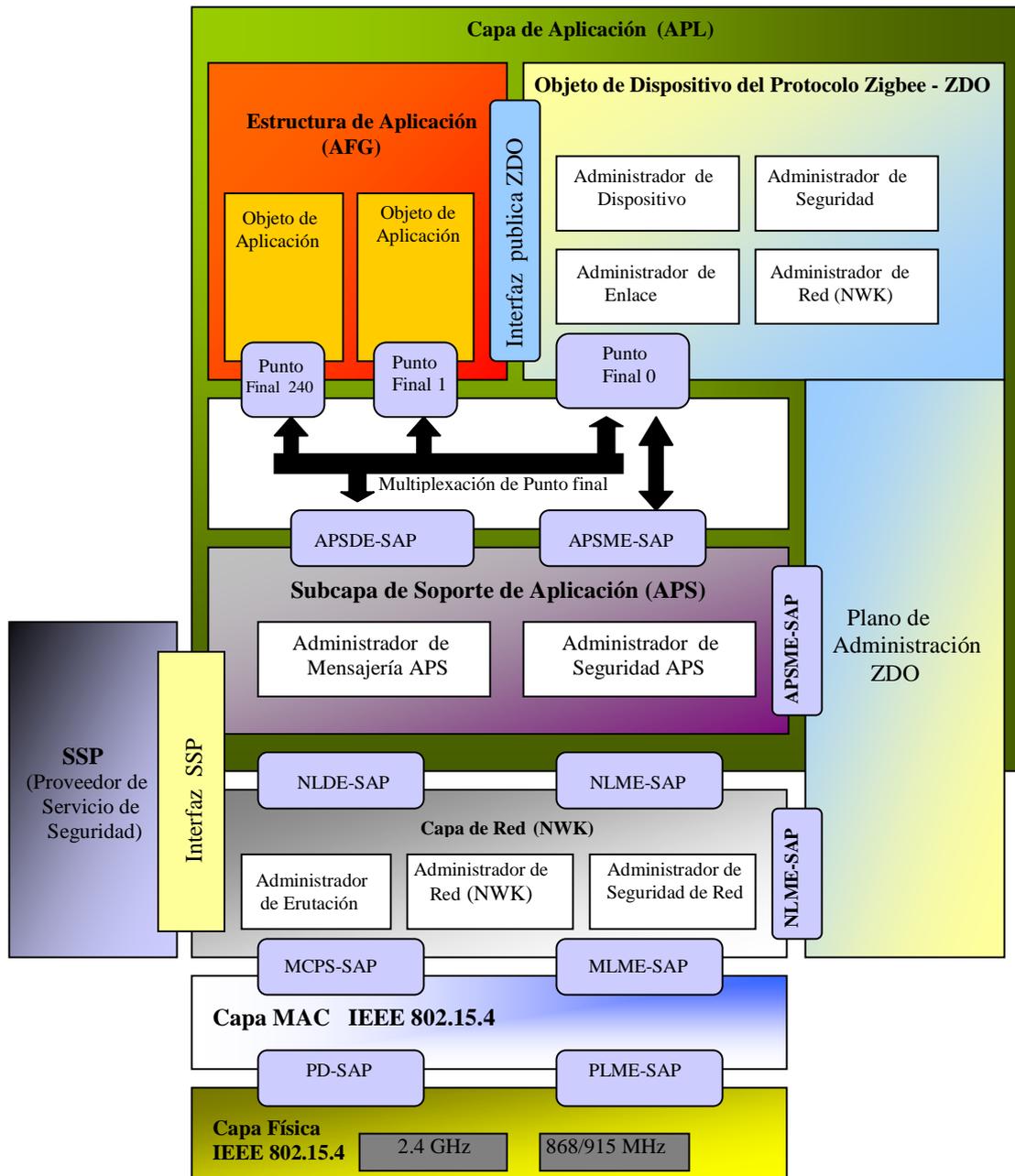
- Entre sus responsabilidades incluye mecanismos para unirse y dejar una red.
- Tiene como tarea el descubrimiento y mantenimiento de rutas entre los dispositivos de la red.
- Está a cargo de la labor de descubrimiento de vecinos y memorización de dicha información.

1.6.4 Capa de aplicación

- Esta capa está dividida en 2 subcapas las cuales son: soporte de aplicación (APS) y el entorno de aplicación (AF); la subcapa AF se divide nuevamente en 2 subcapas las cuales son: objetos para dispositivos ZigBee (ZDO) y los objetos definidos por el diseñador.
- La responsabilidad de la APS es brindar un conjunto general de servicios para el uso de las subcapas AF.

- Las responsabilidades del ZDO son definir el rol de un dispositivo en la red, iniciar y responder a un requerimiento de relación.
- El entorno de aplicación (AF) y sus objetos tienen la responsabilidad de procesar los datos propios de la aplicación.

Figura 1.4 Arquitectura de la pila de protocolos de la tecnología ZigBee de Microchip



Fuente: <http://ww1.microchip.com/downloads/en/AppNotes/ZigBee2006%20Application%20Note%20AN1232A.pdf>

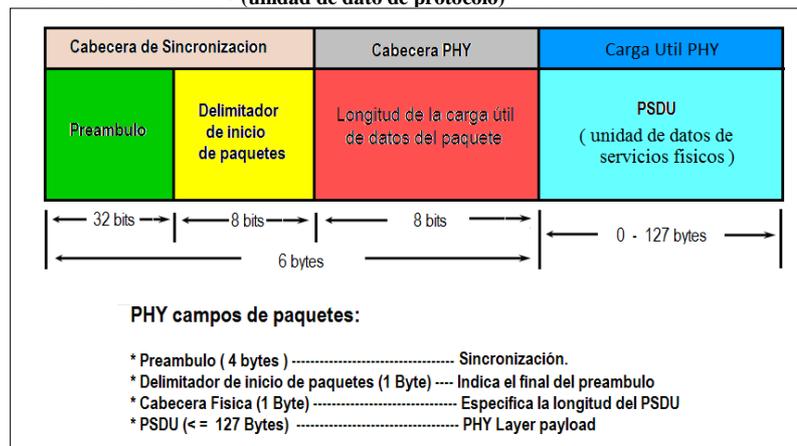
1.7 ARQUITECTURA DE LAS TRAMAS DE LAS DIFERENTES CAPAS DE LA PILA DE PROTOCOLO ZIGBEE.

1.7.1 Trama de la capa PHY

La estructura de la trama PHY Se fundamenta en 3 componentes:

*Cabecera de sincronización. * Cabecera de la capa física. * Carga útil PHY.

Figura 1.5 Estructura de la PDU_(unidad de dato de protocolo) PHY



Fuente:

<http://www.media.mit.edu/resenv/classes/MAS961/readings/embeddedmag/zigbee.html>

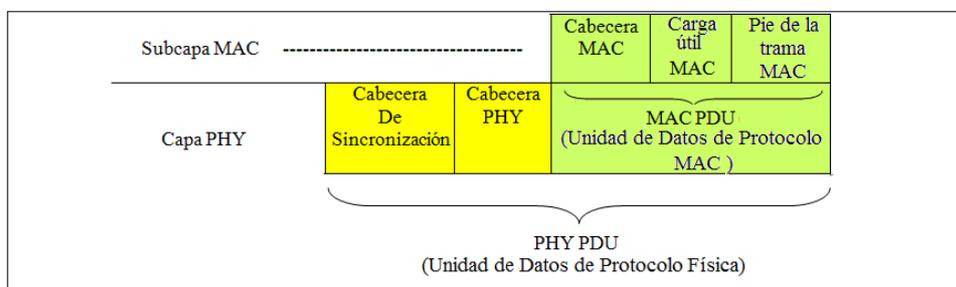
1.7.2 Trama de la capa MAC

La estructura de la trama MAC la cual es establecida por el estándar IEEE 802.15.4 está diseñada de modo que refleje la simplicidad del protocolo.

La trama MAC consta de 3 partes:

*Cabecera de la trama. * Carga útil de la trama. * Pie de la trama.

Figura 1.6 Estructura general de la trama MAC



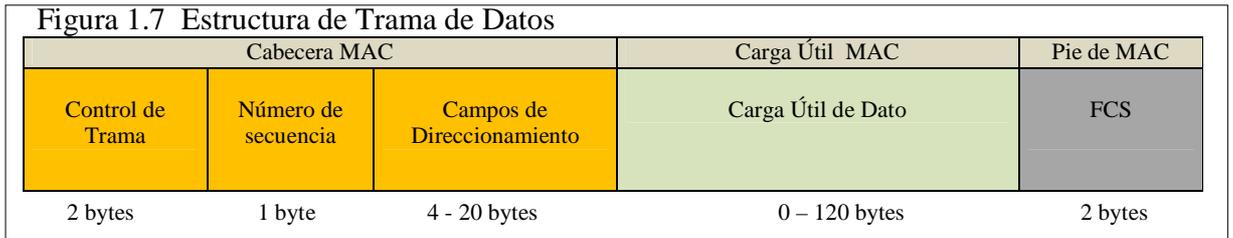
Fuente:

<http://www.media.mit.edu/resenv/classes/MAS961/readings/embeddedmag/zigbee.html>

- La trama MAC se subdivide en 5 tipos regidos por el estándar IEEE 802.15.4
 - * Trama de Datos. *Trama de Comandos MAC. *Trama Beacon. *Supertrama.
 - * Trama ACK.

1.7.2.1 Trama de datos

Esta trama provee hasta *104 bytes* para *información* en el payload, utiliza un campo para secuencia de datos y control de *errores FCS*

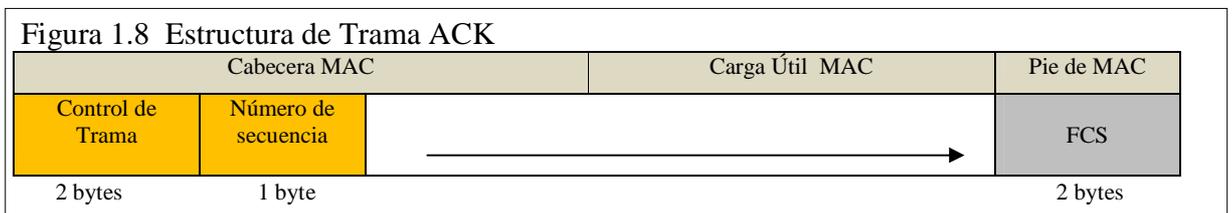


Fuente:

<http://www.media.mit.edu/resenv/classes/MAS961/readings/embeddedmag/zigbee.html>

1.7.2.2 Trama ACK (Reconocimiento de trama)

Esta trama es la encargada de informar si el paquete ha sido recibido correctamente, esto lo realiza por medio de una *realimentación* desde el receptor al emisor.

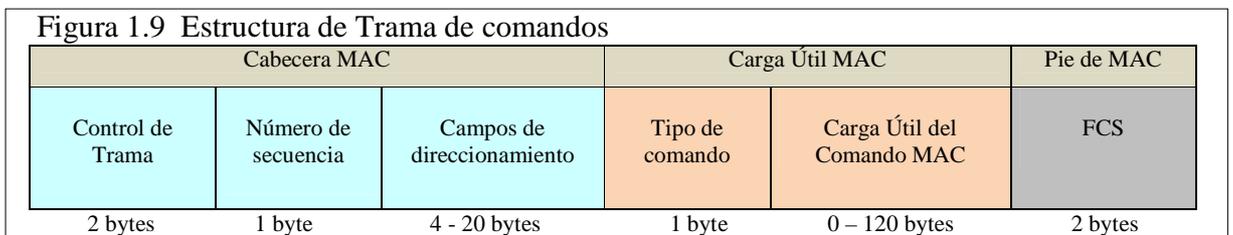


Fuente:

<http://www.media.mit.edu/resenv/classes/MAS961/readings/embeddedmag/zigbee.html>

1.7.2.3 Trama de comandos

La trama de comandos MAC es originada por la subcapa MAC y se encarga de todo el control de transferencia de la MAC.



Fuente:

<http://www.media.mit.edu/resenv/classes/MAS961/readings/embeddedmag/zigbee.html>

1.7.2.4 Trama de Beacon o Balizas

Esta trama es la encargada de enviar balizas de reconocimiento por toda la red.

Si se manda una baliza de datos por toda la red, los dispositivos que se encuentren en ella se despertarán si escuchan su dirección, si no seguirán dormidos.

Las balizas de datos son importantes en las redes estructuradas (malla, árbol), ya que son las encargadas de tener todos los nodos en sincronización.

Figura 1.10 Estructura de Trama de Beacon

Cabecera MAC			Carga Útil MAC				Pie de MAC
Control de Trama	Número de Secuencia	Campos de Direccionalamiento	Especificación de la Supertrama	Especificación de las Direcciones Pendientes	Lista de Direcciones	Carga Útil Beacon	FCS

Fuente:

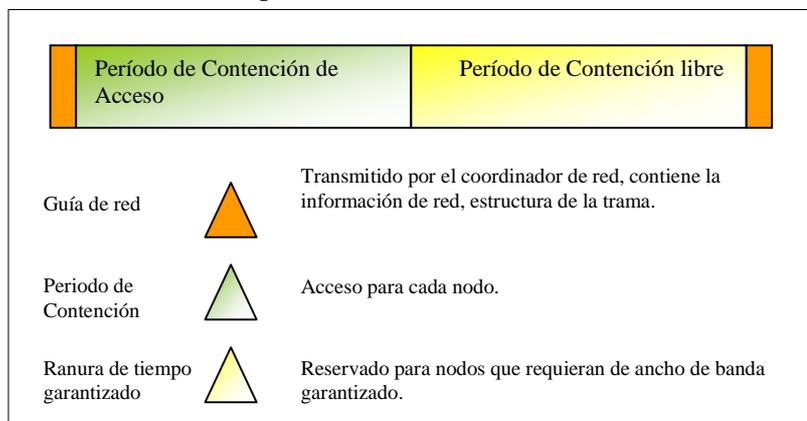
http://www.eng.yale.edu/enalab/courses/eeng460a/homeworks/hw1_results/zigbee.html

1.7.2.5 Superframes (Supertramas)

La siguiente figura corresponde a un Superframe; en donde un coordinador de red, denominado coordinador PAN, transmite guías de red (beacon) en intervalos definidos.

Estos intervalos pueden ser tan cortos como 15 ms. o tan largos como 245 s. El tiempo entre cada uno de ellos se divide en 16 ranuras de tiempo o “time slots”, independientes a la duración de cada Superframe.

Figura 1.11 Estructura de la supertrama



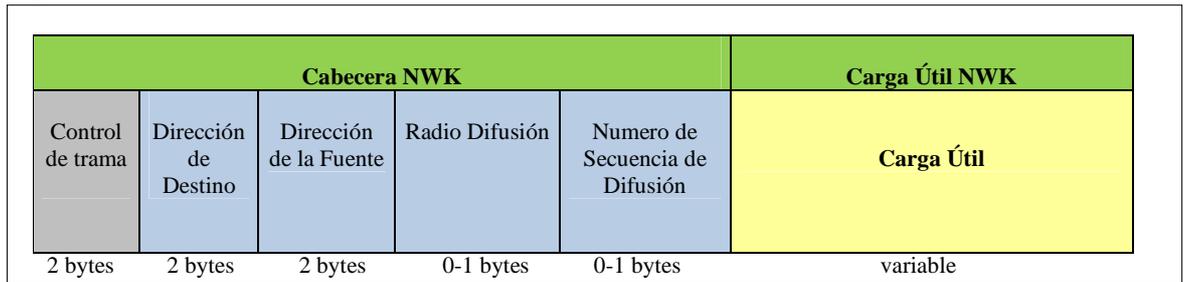
Fuente: <http://wiki.uni.lu/secan-lab/ZigBee+technology+in+sensor+network.html>

1.7.3 PDU (unidad de dato de protocolo) NWK.

Está compuesto principalmente por una *cabecera* y un *payload*.

Las PDUs pueden ser *de datos o de comandos*.

Figura 1.12 Estructura PDU NWK



Fuente:

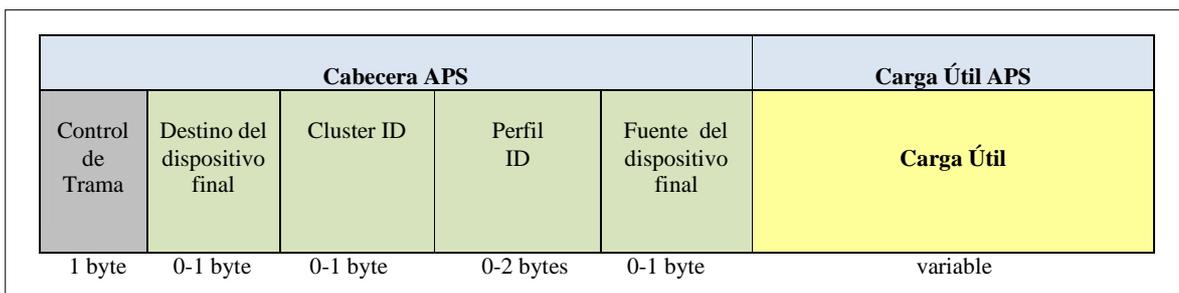
http://www.eng.yale.edu/enalab/courses/eeng460a/homeworks/hw1_results/zigbee.html

1.7.4 PDU (unidad de dato de protocolo) APS.

Está compuesto por una *cabecera* y una carga útil.

Los campos de la cabecera aparecen en un *orden fijo*, pudiendo no ser el de dirección incluida en todas las tramas.

Figura 1.13 Estructura PDU APS



Fuente:

http://www.eng.yale.edu/enalab/courses/eeng460a/homeworks/hw1_results/zigbee.html

CAPITULO 2

ADQUISICIÓN Y ANALISIS DEL HARDWARE

En este capítulo se presenta la selección y descripción de los dispositivos y componentes requeridos para el funcionamiento de la red Zigbee utilizando la pila de protocolos Zigbee de Microchip.

2.1 REQUERIMIENTOS PARA HARDWARE

- Incluir dispositivos para la transmisión y recepción de datos formando una red inalámbrica utilizando la pila de protocolos de Microchip en la tecnología Zigbee.
- permitir terminales de interacción con el usuario mediante el uso de dispositivos de entrada y salida (rs232, usb, spi).
- Proporcionar la interfaz necesaria para la programación del software requerido por los nodos (PIC).
- Suministro de energía para los dispositivos de un modo portátil que brinde un buen funcionamiento del mismo.

2.2 SELECCIÓN DEL HARDWARE

De acuerdo a los requerimientos anteriormente descritos, se decidió adquirir el módulo de desarrollo PICDEMZ de la Compañía Microchip para la implementación, programación y depuración del sistema.

2.2.1 Kit de desarrollo picdem z

El kit de evaluación PICDEM Z está diseñado para permitir a los desarrolladores experimentar y crear aplicaciones con las herramientas Microchip para el protocolo Zigbee. Cada nodo del kit contiene una tarjeta madre y un dispositivo RF. La tarjeta madre está diseñada para soportar diferentes tipos de transceivers (RF).

Figura 2.1 Kit PICDEM Z



Características

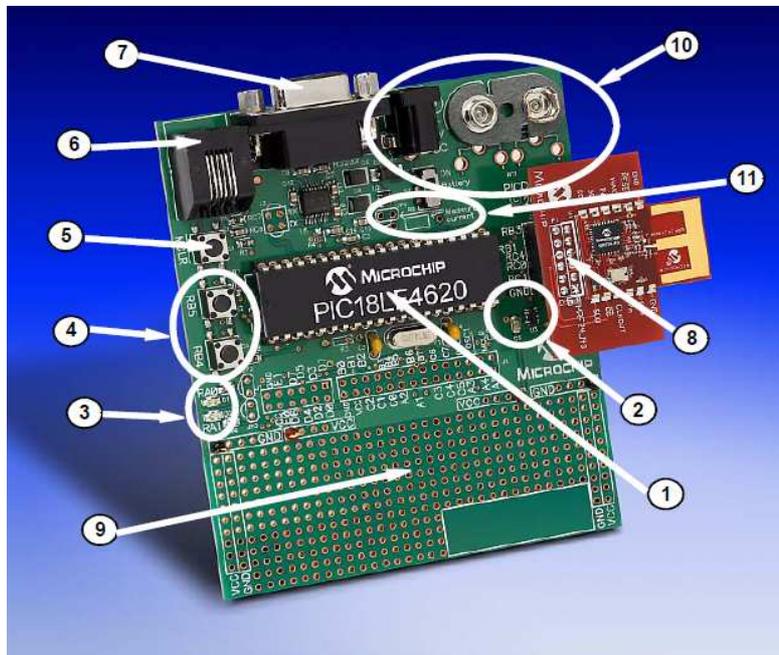
Las principales características del kit de demostración PICDEM Z son:

- El software de la pila de protocolos Zigbee para los dispositivos RFD (Reduced Function Device), FFD (Full Function Device) y Coordinador
- PIC18LF4620 MCU con tecnología del nanowatt, 64 KB de memoria Flash y periféricos integrados robustos.
- Microchip MRF24J40 Transceiver RF y antena con conexión para tarjeta madre.
- Analizador de red inalámbrica ZENA, incluyendo el software de configuración de la pila de protocolo.
- Conector de interfaz ICSP y MPLAB ICD 2.
- Interfaz RS-232.
- Regulador de 9 v DC a 3,3 V DC.
- Sensor de temperatura (Microchip TC77), LEDS y switches como apoyo a la demostración.

2.2.1.1 Motherboard PICDEM Z

El esquema de los circuitos y los componentes de la tarjeta madre PICDEM Z se muestra en las Figura 2.2, Figura 2.3 y Figura 2.4 y se describen a continuación.

Figura 2.2 Elementos de la tarjeta madre



Fuente: <http://www.microchipdirect.com/ProductSearch.aspx?keywords=AC163027-1>

1. **Zócalo para microcontrolador:** Esta diseñado para microcontroladores PIC de la familia 18 de 40 y 28 pines. La tarjeta contiene un PIC 18LF4620.
2. **Sensor de temperatura (TC77):** Sensor térmico con Interfaz Periférica Serial SPI de Microchip, el cual comparte el bus de comunicación SPI con el transceiver (U3).
3. **LEDs (D1, D2):** Estos dos LEDs son manejados por pines de (I/O) del microcontrolador, y se usan para simular salidas digitales. Estos LEDs pueden ser habilitados o deshabilitados por los jumpers JP2 y JP3. Por defecto están habilitados.
4. **Pulsadores (S2, S3):** Estos dos pulsadores están conectados a los pines de (I/O) del microcontrolador y se usan para simular entradas digitales.
5. **Pulsador de Reset (S1):** Está conectando al MCLR (Reset) del microcontrolador y se usa para resetear la tarjeta.
6. **Conector RJ11 (J5):** Este conector permite que la tarjeta de demostración se conecte al sistema Microchip MPLAB ICD2 para **programación** del microcontrolador.
7. **Conector RS-232 (P1):** Este conector se usa para conexión de la tarjeta con el puerto serial de la PC.
8. **Conector del transceiver (J2):** Se usa para conectar la tarjeta de RF. Este conector provee voltaje de 3.3V DC, una interfaz periférica serial SPI y señales discretas I/O de control.
9. **Área de Prototipo:** Conexión de circuitos adicionales.
10. **Conexión de energía:** Se puede energizar externamente a 9V DC por medio de una entrada (J1) o una batería de 9V (B1). La tarjeta tiene un regulador de voltaje a 3.3 V y 100mA y un diodo de protección para conexión accidental en polaridad inversa.
11. **Medidor de corriente (JP4):** Abriendo este Jumper se puede medir la corriente de la tarjeta.

2.2.1.2 MRF24J40MA PICDEM Z 2.4 GHZ RF Board

El transceptor (transceiver) es el encargado de implementar la capa PHY y MAC establecidas en el estándar IEEE 802.15.4.

Figura 2.3 Transceiver MRF24j40MA



Fuente: <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en535967>

Características:

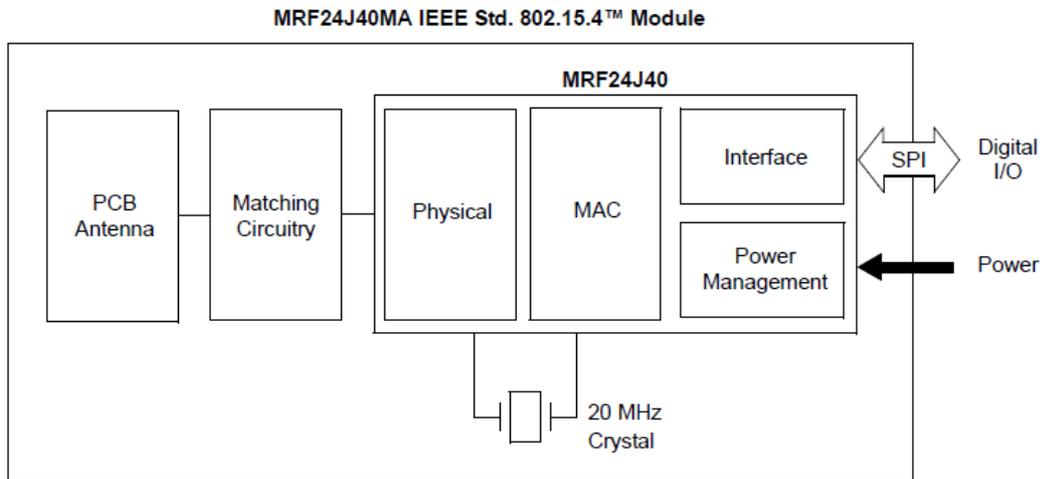
- Transceptor RF IEEE 802.15.4
- Compatible con ZigBee, MiWi, P2P MiWi y Protocolos de propiedad de redes Inalámbricas.
- Pequeño Tamaño: 0,7 "x 1,1" (17,8 mm x 27,9 mm), Montaje superficial
- Cristal integrado, regulador de voltaje interno, Coincidencia de los circuitos y Antena PCB.
- Compatible con microcontrolador de Microchip Familias (PIC16F, PIC18F, PIC24F / H, dsPIC33 y PIC32).
- Rango arriba de 400 ft.

Características analógica RF:

- Banda de Operación 2.405 - 2.48 GHz
- Velocidad de datos: 250 Kbps
- -94 dBm sensibilidad típica con +5 dBm Nivel máximo de entrada
- 0 dBm de potencia de salida típica con 36 dB TX rango de poder de control
- Alto recepcionamiento y rango dinámico RSSI.

Diagrama de bloques

Figura 2.4 Diagrama de bloques del transceiver MRF24J40MA



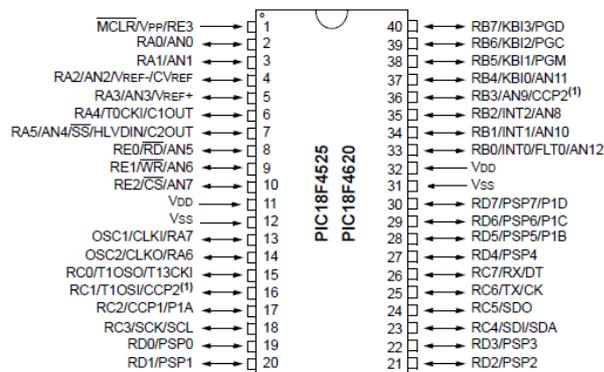
Fuente: <http://ww1.microchip.com/downloads/en/DeviceDoc/70329b.pdf>

La Figura 2.4 muestra un diagrama de bloques simplificado del MRF24J40MA módulo. El módulo se basa en la Tecnología Microchip MRF24J40 IEEE 802.15.4 de 2,4 GHz receptor RF.

2.2.1.3 Microcontrolador PIC18F4620

El microcontrolador PIC 18LF4620 es utilizado por Microchip en su kit de desarrollo Zigbee PICDEM Z. Este microcontrolador de 40 pines en empaque PDIP de la familia 18 dispone de 64kbytes de memoria de programa y 4 kbytes de memoria de datos. Además soporta opciones de ahorro de energía, recomendado para aplicaciones Zigbee.

Figura 2.5 Pines PIC 18F4620



Fuente: <http://www.datasheetcatalog.org/datasheet2/d/0j1script8ektr10e9szi820j36cy.pdf>

Cuadro 2.1 Características del PIC 18F4620

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ ECCP (PWM)	MSSP		EUSART	Comp.	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI	Master I ² C™			
PIC18F4620	64K	32768	3968	1024	36	13	1/1	Y	Y	1	2	1/3

Fuente: <http://www.datasheetcatalog.org/datasheet2/d/0j1scipt8ektr10e9szi820j36cy.pdf>

2.2.1.4 Analizador de redes inalámbricas ZENA

- El analizador ZENA ofrece tres herramientas principales para desarrollar soluciones IEEE 802.15.4 rápida y eficientemente con la pila de protocolos de Microchip zigbee la cual se obtiene de una manera gratuita.
- El analizador ZENA permite a los desarrolladores poder modificar rápidamente las pilas de protocolos, para adaptarse a las necesidades de aplicación.
- El analizador de ZENA es también un analizador de paquetes IEEE 802.15.4, que soporta actualmente 2,4 GHz de espectro.
- El analizador de ZENA señala a la topología de la red, ya que se forma y permite a los usuarios ver las transacciones de paquetes a medida que ocurren.

Figura 2.6 Analizador Zena

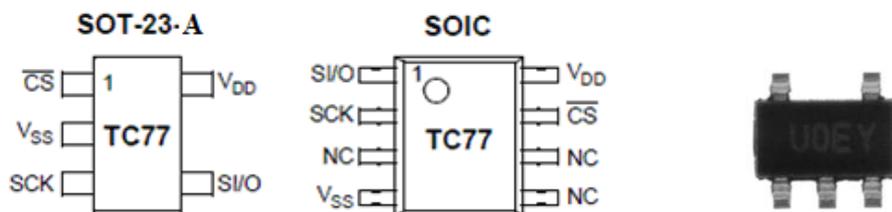


- Tiene una antena PCB (Printed Circuit Board) la cual recibe los paquetes en el canal especificado y envía la información a través del cable USB al PC.

2.2.1.5 Sensor de temperatura con interfaz SPI

Se utilizó este sensor de temperatura ya que viene incluido en la tarjeta madre, y es muy eficiente ya que es digital, esto le da mayor precisión, y su comunicación es por medio de SPI (Serial Peripheral Interface), el cual es un bus estándar de comunicaciones.

Figura 2.7 Sensor de temperatura TC77



Fuente: http://www.datasheetcatalog.org/datasheets2/38/383989_1.pdf

Características

- Sensamiento de temperatura digital en 5-Pines para los modelos SOT -23-A.
- Salidas de la temperatura como una palabra digital de 13-Bit
- SPI y compatible con la interfaz Microwire
- La temperatura de estado sólido de detección
- $\pm 1^\circ \text{C}$ (máx.) exactitud de $+25^\circ \text{C}$ a $+65^\circ \text{C}$
- $\pm 2^\circ \text{C}$ (máx.) Precisión de -40°C a $+85^\circ \text{C}$
- $\pm 3^\circ \text{C}$ (máx.) Precisión de -55°C a $+125^\circ \text{C}$
- Rango de funcionamiento 2.7V a 5.5V
- Baja Potencia
- 250 μA (typ.) en modo de conversión continua
- 0.1 μA (typ.) en modo de apagado

2.3 ANALISIS DE COSTOS

Los precios descritos en las tablas 2.2 y 2.3 exponen los costos de los elementos que se adquirieron para construcción de 1 nodo o para la construcción de toda la implementación. En el cuadro 2.4 muestra los valores de las obligaciones arancelarias, impuestos de ley y costos de transporte para importar los dispositivos requeridos.

Cuadro 2.2 Costos de un nodo.

Componente	Referencia	Cantidad	Precio USD
PICDEM Z Motherboard	AC163027-1	1	89.99
PICDEM Z RF Board	AC163028	1	18.95
Batería 9 Volt.		1	4.0
TOTAL			USD \$ 112.94

Cuadro 2.3 Costos de toda la implementación.

Componente	Referencia	Cantidad	Precio USD
Kit PICDEM Z	DM163027-5	1	269.99
PICDEM Z Motherboard	AC163027-1	1	89.99
PICDEM Z RF Board	AC163028	1	18.95
TOTAL			USD \$478.93

NOTA: Estos productos fueron adquiridos en la página oficial de MOUSER ELECTRONICS la cual es WWW.MOUSER.COM .

Cuadro 2.4 Costos de importación: importación y transporte.

Detalle	Valor USD
Compra total Kit PICDEM Z PICDEM Z Motherboard PICDEM Z RF Board	478.93
Transporte	100
Importación	
Arancel 10%	38.37
I.V.A 16%	67.42
Otros Cargos HANDLING FEE IVA HANDLING	20.96 3.33
TOTAL	USD \$ 709.01

CAPITULO 3

ANÁLISIS DEL SOFTWARE

3.1 PILA DE PROTOCOLOS DE LA TECNOLOGIA ZIGBEE DE MICROCHIP

El protocolo ZigBee es un protocolo de red inalámbrica diseñado específicamente para sensores de baja velocidad de datos y redes de control. Hay una serie de aplicaciones que pueden beneficiarse del protocolo ZigBee, incluyéndola pero sin limitarse a las redes de automatización de edificios, sistemas de seguridad, redes de control industrial, medición remota y periféricos de PC.

Las capas superiores son especificadas por el estándar IEEE 802.15.4 las cuales proporcionan una metodología para realizar las funciones de red tales como la formación de redes, tareas dirección de dispositivo, encaminamiento, mensajería y búsqueda de dispositivos.

3.1.1 Funcionamiento

La pila se encuentra desarrollada en lenguaje C y está diseñado para compilarse mediante MPLAB C18 y ejecutarse en microcontroladores PIC de la familia 18 y de la familia 24. Utiliza memoria interna de programa tipo FLASH para guardar una gran cantidad de parámetros, incluyendo dirección MAC, tabla de vecindades y tabla de relaciones (bindings). La pila está diseñada para correr en el Kit de desarrollo PICDEM Z. Sin embargo, puede ser trasladado fácilmente a cualquier hardware equipado con un microcontrolador PIC compatible y el transceiver RF soportado.

La pila y su código están organizados mediante primitivas o estados para su ejecución, especificados en el estándar Zigbee/IEEE 802.15.4, que permiten el acceso a los servicios más importantes para realizar procesos como formación de la red, transmisión y recepción de comandos y datos.

Las primitivas Zigbee utilizadas se clasifican en:

- ◆ Primitivas de petición:

Utilizadas por la pila con el objeto de realizar los diferentes procesos de comunicación (formar la red, enviar datos, establecer relaciones, descubrir direcciones, etc.). En la pila se implementa cada una de estas primitivas, el desarrollador sólo es responsable de llamarlo y configurar previamente los valores necesarios para que el Stack la procese.

◆ Primitivas de respuesta/indicación:

Son utilizadas por la pila para entregar una contestación a una petición o para indicar la recepción de información. El desarrollador es responsable de implementar el código necesario para procesar los datos o la respuesta recibida para proseguir o corregir errores si es necesario.

◆ No Primitive:

En este estado la pila no realiza ningún proceso Zigbee, es utilizada por el desarrollador para preparar la próxima primitiva de tipo petición y cargar los valores que se necesiten en los registros requeridos para su ejecución.

Cuadro 3.1 Lista de primitivas de la tecnología Zigbee.

Primitivas de petición	Primitivas de respuesta	Descripción
APSDE_DATA_request	APSDE_DATA_confirm	Para enviar datos a otros dispositivos
APSME_BIND_request	APSME_BIND_confirm	Obliga a crear un enlace o binding.
APSME_UNBIND_request	APSME_UNBIND_confirm	Obliga a eliminar un enlace o binding.
NLME_NETWORK_DISCOVERY_request	NLME_NETWORK_DISCOVERY_Confirm	Descubre las redes disponibles para unirse. No es usada por el coordinador.
NLME_NETWORK_FORMATION_request	NLME_NETWORK_FORMATION_Confirm	Inicia una red en uno de los canales especificados
NLME_PERMIT_JOINING_request	NLME_PERMIT_JOINING_Confirm	Permite a otros nodos unirse a la red como esclavos.
NLME_START_ROUTER_request	NLME_START_ROUTER_Confirm	Inicializa las funcionalidades de ruteo. (Solo Routers)
NLME_JOIN_request	NLME_JOIN_confirm	Intenta unirse o volver a una red especificada.
NLME_DIRECT_JOIN_request	NLME_DIRECT_JOIN_Confirm	Añade un dispositivo como esclavo. solo coordinadores y routers
NLME_LEAVE_request	NLME_LEAVE_confirm	Dejar la red o forzar a un esclavo a dejarla.

NLME_SYNC_request	NLME_SYNC_confirm	Petición de mensajes guardados por el dispositivo padre. Solo RFD
APSME_ADD_GROUP_request	APSME_ADD_GROUP_confirm	Solicitud de afiliación en determinado grupo a un endpoint. Sólo se puede utilizar en dispositivos que admiten multicast addressing.
APSME_REMOVE_GROUP_request	APSME_REMOVE_GROUP_Confirm	Remove afiliación en determinado grupo de un endpoint. Sólo se puede utilizar en dispositivos que admiten multicast addressing.
APSME_REMOVE_ALL_GROUPS_request	APSME_REMOVE_ALL_GROUPS_Confirm	Remove afiliación en todos los grupos de un endpoint. Sólo se puede utilizar en dispositivos que admiten multicast addressing.
NETWORK_ROUTE_DISCOVERY_request	NETWORK_ROUTE_DISCOVERY_Confirm	Iniciar un descubrimiento de ruta a otro dispositivo. Solo coordinador y router.

3.2 ANALIZADOR INALÁMBRICO DE REDES ZENA

Para ayudar en el desarrollo de aplicaciones de protocolo ZigBee, microchip proporciona un software analizador de red de bajo costo, llamado ZENA. El software de PC ZENA también contiene una herramienta para crear archivos de configuración específica de la aplicación y scripts enlazador para aplicaciones de protocolo ZigBee. El software de demostración ZENA es suministrado de forma gratuita como parte de la pila de Microchip para la instalación de ZigBee Protocolo y se encuentra en el directorio ZigBee2006Res.

Cuando el software ZENA se utiliza para configurar una aplicación de protocolo ZigBee, creará tres archivos de aplicación: zigbee.def, myZigBee.c y zLink.lkr. El zigbee.def y archivos myZigBee.c contienen información crítica para la configuración de la pila. El zLink.lkr es el script del enlazador para la aplicación. Es altamente recomendable que utilice el software ZENA para generar estos archivos, en lugar de editar los archivos manualmente, ya que los archivos son interdependientes. Una secuencia de comandos del vinculador no es necesaria para las versiones de compilador C30 PIC24 del protocolo ZigBee Stack.

Para mirar como configurar y generar los archivos zigbee.def, myZigBee.c y zLink.lkr dirigirse al archivo ZENA™ Wireless Network Analyzer User's Guide (DS51506) que se encuentra en la pagina oficial de Microchip.

3.2.1 Análisis de los paquetes de datos utilizando el analizador ZENA

Los mensajes que aparecen cuando un dispositivo se une a una red de protocolo ZigBee nos dicen mucho acerca de la red. Cuando un dispositivo intenta unirse a una red, se emite una solicitud Beacon. Los coordinadores de los alrededores y routers responderán mediante la emisión de las señales. El nuevo dispositivo seleccionará un Beacon y enviará una solicitud de asociación al dispositivo, pidiendo unirse a la red.

Después de un breve retraso, el nuevo dispositivo enviará una solicitud de datos, pidiendo una respuesta. El dispositivo de red entonces enviará una respuesta de asociación, la cual indica si el nuevo dispositivo ha sido aceptado en la red. Estos mensajes pueden aparecer como se ilustra en la Figura 3.1.

Figura 3.1 Mensajes de asociación.

est	Dest	Beacon	FCS																
All	Addr	Request	RSSI	Corr	CRC														
FFFF	0xFFFF		-15	0x6C	OK														
Source	Source	SuperFrame Specification			GTS Specification	PendAddr Spec	Beacon Payload			FCS									
All	Addr	BO	SO	CAP	Batt	Coord	Assoc	Permit	Count	ExtAddr	ShortAddr	DevCap	Depth	RtrCap	NWKVer	StkProf	ProtID	RSSI	C
1234	0x143E	None	None	0xF	N	N	Y	N	0x0	0x0	0x0	Y	0x1	Y	0x1	0x1	0x00	-10	0:
Source	Source	SuperFrame Specification			GTS Specification	PendAddr Spec	Beacon Payload			FCS									
All	Addr	BO	SO	CAP	Batt	Coord	Assoc	Permit	Count	ExtAddr	ShortAddr	DevCap	Depth	RtrCap	NWKVer	StkProf	ProtID	RSSI	C
1234	0x0000	None	None	0xF	N	Y	Y	N	0x0	0x0	0x0	N	0x0	N	0x1	0x1	0x00	-16	0:
est	Dest	Source	Association Request			FCS													
All	Addr	PAH	Source Address	Alloc	Sec	RxOn	Power	Dev	AltCoord	RSSI	Corr	CRC							
1234	0x143E	0xFFFF	0x0004A30000000088	Y	N	On	Main	END	N	-15	0x6B	OK							
FCS																			
SI			Corr CRC																
1			0x6B OK																
est	Dest	Source	Association Request			FCS													
All	Addr	PAH	Source Address	Data	RSSI	Corr	CRC												
1234	0x143E	0x1234	0x0004A30000000088		-14	0x6C	OK												
FCS																			
SI			Corr CRC																
1			0x6B OK																
est	Destination Address	Source	Association Response		FCS														
All		PAH	Status	Address	RSSI	Corr	CRC												
1234	0x0004A30000000088	0x1234	Success	0x1AF9	-11	0x6C	OK												
FCS																			
SI			Corr CRC																
1			0x6B OK																

El primer mensaje es del nuevo dispositivo, solicitando balizas. Los siguientes dos mensajes son Beacons.

Se pueden decir algunas cosas interesantes acerca de las balizas de los dispositivos. En primer lugar, se puede observar en la Figura 3.1 que la segunda baliza es la del coordinador, porque la dirección de origen es 0x0000 y la profundidad (depth) especificada en la carga útil del beacon es 0.

El primer beacon es de un router, ya que su dirección de origen es distinta de cero.

Este router está conectado a la red a través del coordinador ya que se puede observar en la Figura 3.1 que pertenece a la misma red PAN 0x1234 formada por dicho coordinador y la carga útil del beacon de profundidad (depth) del router es igual a 1.

Se sabe que en la capa de aplicación de ambos dispositivos se tiene la capacidad de aceptación de nuevos dispositivos para unirse a la red, como se observa en las tramas dos y tres de la Figura 3.1 en la parte de superframe specification en la opción assoc la cual se encuentra habilitada.

Sin embargo, si se observa en las cargas útiles beacon, se puede ver que el coordinador no tiene capacidad para ningún router adicional o dispositivo final, ya que el **RtrCap** y **DEVCAP** son 0.

El router por su parte, todavía tiene capacidad para dos routers y dispositivos finales.

En el cuarto mensaje, el nuevo dispositivo ha decidido tratar de unirse a la red. Mediante la observación de la dirección de destino de la solicitud de asociación, se puede decir que el nuevo dispositivo de la red está tratando de unirse al router.

La petición de la asociación dice mucho sobre el nuevo dispositivo. Los elementos críticos son los bits de **Dev** y **RxOn**.

El bit Dev indica que el dispositivo está tratando de unirse como un dispositivo final. Además, se sabe que este dispositivo siempre tendrá su transceiver encendido, así que sus padres no tendrán que búfear mensajes para él.

Este bit es fundamental para la comunicación posterior. Si RxOn está desactivado, entonces el padre supone que el nuevo dispositivo mantiene su transceptor apagado y búfeará mensajes para él. Si el nuevo dispositivo no pide mensajes, entonces el padre nunca le envía mensajes.

Por último, puede verse que la solicitud de asociación fue un éxito. Ahora se tiene un nuevo FFD (Full Function Device) dispositivo final en la red, con una dirección de 0x1AF9

En la figura 3.2, el dispositivo 0x0001 le dice al dispositivo 0x0000 establecer un atributo a un valor específico usando los tres niveles de reconocimiento (acknowledgment).

Figura 3.2 Niveles de reconocimiento (acknowledgment)

n	MAC Frame Control	Seq Num	Dest PAN	Dest Addr	Source PAN	Source Addr	HWK Header	APS Frame Control	Dest EP	Cluster ID	Profile ID	Source EP	AF Header	Transaction 1	Data 1	RS
1	Type: DATA, Sec: N, Pend: N, ACK: Y, IPAN: N	0x08	0x1234	0x0000	0x1234	0x0001	0x04 0x00 0x00 0x00	Type: DAT, Deliv: UNI, Mode: N/A, Sec: N, ACK: Y	0x08	0x13	0x0100	0x08	Cnt: 0x01, Type: KYF, SN: 0xA5, Cmd: SetAck, Type: UINT8, Attrib: 0x0000	0x00	-1	
2	Type: ACK, Sec: N, Pend: Y, ACK: N, IPAN: N	0x08	FCS		RSSI: -13, Corr: OK, CRC: OK											
3	Type: DATA, Sec: N, Pend: N, ACK: Y, IPAN: N	0x01	0x1234	0x0001	0x1234	0x0000	0x44 0x00 0x01 0x00	Type: ACK, Deliv: UNI, Mode: N/A, Sec: N, ACK: N	0x08	0x13	0x0100	0x08	FCS		RSSI: -09, Corr: OK, CRC: OK	
4	Type: ACK, Sec: N, Pend: N, ACK: N, IPAN: N	0x01	FCS		RSSI: -17, Corr: OK, CRC: OK											
5	Type: DATA, Sec: N, Pend: N, ACK: Y, IPAN: N	0xE2	0x1234	0x0001	0x1234	0x0000	0x00 0x00 0x0A 0x71	Type: DAT, Deliv: UNI, Mode: N/A, Sec: N, ACK: Y	0x08	0x13	0x0100	0x08	Cnt: 0x01, Type: KYF, SN: 0xA5, Cmd: SetResp, Type: UINT8, Attrib: 0x0000	0x00	Success	
6	Type: ACK, Sec: N, Pend: Y, ACK: N, IPAN: N	0xE2	FCS		RSSI: -13, Corr: OK, CRC: OK											
7	Type: DATA, Sec: N, Pend: N, ACK: Y, IPAN: N	0x09	0x1234	0x0000	0x1234	0x0001	0x44 0x00 0x00 0x00	Type: ACK, Deliv: UNI, Mode: N/A, Sec: N, ACK: N	0x08	0x13	0x0100	0x08	FCS		RSSI: -09, Corr: OK, CRC: OK	
8	Type: ACK, Sec: N, Pend: N, ACK: N, IPAN: N	0x09	FCS		RSSI: -17, Corr: OK, CRC: OK											

El primer mensaje es el comando para el dispositivo 0x0000 a establecer un atributo. Esto es seguido por un corto reconocimiento MAC. Este reconocimiento indica que el transceiver del dispositivo receptor recibe correctamente el mensaje. Estos son a menudo generados automáticamente por el transceiver. El protocolo ZigBee especifica que todos los mensajes deben solicitar un reconocimiento MAC.

En el campo del APS- frame control del primer mensaje, hay un bit que indica si existe o no una solicitud de reconocimiento APS. Si este bit se establece, entonces la capa de APS de la pila ZigBee del destinatario generará automáticamente un APS de reconocimiento, como lo indica el tercer mensaje.

Este reconocimiento indica que la pila del dispositivo de destino recibió el mensaje. La Aps ack generará otra Mac de reconocimiento, como lo demuestra el cuarto mensaje.

En cuanto a la operación de decodificación del mensaje en primer lugar, podemos observar que se ha utilizado el set con el comando de Reconocimiento. Esta es una indicación de la capa de aplicación, la capa de aplicación debe reconocer el comando con un comando de set de respuesta, como indica el quinto mensaje

Este reconocimiento indica que la capa de aplicación recibo el mensaje y el código de error de transmisión indicará si algún problema se produjo cuando se estableció el atributo con el valor solicitado.

El acknowledge AF reconocer generará otro acknowledgment Mac, como lo demuestra el sexto mensaje. Además, el acknowledge AF fue enviado con un APS ack requested, que va a generar el séptimo y octavo mensaje.

3.3 ORGANIZACIÓN DE ARCHIVOS FUENTE

La pila de Microchip consta de múltiples archivos fuentes. Para compatibilidad con otras aplicaciones de Microchip, archivos que son comunes para las aplicaciones son almacenados en un directorio específico, los archivos específicos de la pila del estándar ZigBee son almacenados en otros directorios. La siguiente tabla muestra la estructura de los directorios.

Cuadro 3.2 Estructura de los directorios de los archivos fuentes

Nombre de el Directorio	Contenido
DemoPIC18Coordinator DemoPIC24Coordinator	Código fuente para la aplicación de demostración ZigBee protocolo Coordinador. Estos archivos se pueden cambiar para crear una aplicación personalizada.
DemoPIC18RFD DemoPIC24FRFD	Código fuente para la aplicación de demostración ZigBee protocolo RFD. Estos archivos se pueden cambiar para crear una aplicación personalizada.
DemoPIC18Router DemoPIC24FRouter	Código fuente para la aplicación de demostración ZigBee protocolo Router. Estos archivos se pueden cambiar para crear una aplicación personalizada.
Documentation	Contiene documentación de la pila de Estándar ZigBee de Microchip.
Microchip	Contiene los archivos fuentes de la pila del Estándar ZigBee de Microchip. Los archivos contenidos en este directorio no deberán cambiarse.

3.4 APLICACIONES DE DEMOSTRACIÓN

Las versiones 2.0 - 2.6 de la pila de Microchip incluyen tres principales aplicaciones de demostración:

- ◆ DemoPIC18Coordinator /DemoPIC24Coordinator - Muestra una típica aplicación de el dispositivo ZigBee Coordinador.
- ◆ DemoPIC18RFD/DemoPIC24FRFD - Muestra una típica aplicación de el dispositivo ZigBee RFD.

- ◆ DemoPIC18Router/DemoPIC24FRouter - Muestra una típica aplicación de el dispositivo ZigBee Router.

3.4.1 Característica de la aplicación de demostración

Las aplicaciones de demostración cumplen las siguientes características:

- ◆ Están dirigidas para usar con las tarjetas PICDEM Z y EXPLORE 16 DEMO.
- ◆ Sondeo de operación de el dispositivo final.
- ◆ Salida RS-232 para ver la operación del dispositivo, así como un sistema de menú para enviar comandos a los dispositivos de funcionamiento.
- ◆ Funciona como una aplicación simple de direccionamiento multicast.

Una tarjeta PICDEM Z ó EXPLORE 16 DEMO debe ser programada como Zigbee Protocolo Coordinador con el proyecto DemoCoordinator. Una segunda tarjeta debe ser programada como un router mediante el correspondiente proyecto DemoPICXXRouter. Si más tarjetas PICDEM Z ó EXPLORE 16 DEMO están disponibles, se pueden programar ya sea como dispositivos finales o como routers con el proyecto adecuado.

3.5 DESCRIPCIÓN DE LOS ARCHIVOS FUENTES.

Las siguientes tablas, describen la implementación de los archivos del stack ZigBee y sus aplicaciones de demostración.

Cuadro 3.3 Archivos fuente del Stack de Microchip en el Subdirectorio *ZigBeeStack*.

Nombre del Archivo	Descripción
SymbolTime.c,.h	Ejecuta funciones de tiempo para la pila del Estándar ZigBee de Microchip.
zAPL.h	Archivo de cabecera de interfaz del nivel de aplicación para la pila.
zAPS.c, .h	Capa APS del estándar ZigBee.
zTest.h	Información del profile ZigBee ZCP. Este puede cambiar dependiendo del profile.
zigbee.h	Genera las constantes de el Zigbee protocolo
ZigBeeTasks.c, .h	Direcciona el programa a través de las capas de la pila
zMAC.h	Archivo de cabecera de la capa MAC genérica IEEE 802.15.4.

zMAC_MRF24J40.c, .h	Capa MAC IEEE 802.15.4 para el transmisor-receptor Microchip MRF24J40
zNVM.c, .h	Ejecuta funciones de almacenamiento en memoria no volátil.
zNWK.c, .h	Capa NWK del estándar ZigBee
zPHY.h	Archivo de cabecera de la capa PHY genérica IEEE 802.15.4.
zPHY_MRF24J40.c, .h	Capa PHY para el transmisor-receptor Microchip MRF24J40 IEEE 802.15.4.
zSecurity.h	Archivo de cabecera ZigBee protocol security.
zSecurity_MRF24J40.c, .h	Archivo de cabecera ZigBee protocol security para el transmisor-receptor Microchip MRF24J40.
zZDO.c, .h	Capa ZDO (ZDP) del Estándar ZigBee.

Cuadro 3.4 Archivos Fuente comunes de Microchip en el Subdirectorío *Common*.

Nombre de el archivo	Descripción
Compiler.h	Define especificaciones de compilación.
Console.c, .h	Código de interface USART (opcional)
Generic.h	Constantes genéricas y definición de variables.
MSPI.c, .h	Código de interface SPI
sralloc.c, .h	Código de distribución de memoria dinámica.

Cuadro 3.5 Subdirectorío de demostración de Coordinadores del Estándar ZigBee en *DemoCoordinator*.

Nombre de el archivo	Descripción
Coordinator.c	Archivo fuente de aplicación principal.
DemoPIC18Coordinator.mcp DemoPIC24Coordinator.mcp	Archivo del proyecto.
DemoPIC18Coordinator.mcw DemoPIC24Coordinator.mcw	Espacio de trabajo.
myZigBee.c	Contiene información específica de aplicación. Es generada por el software ZENA.
zigbee.def	Contiene información específica de aplicación. Es generada por el software ZENA.
zLink.lkr	Enlace manuscrito del proyecto. Es generada por el software ZENA.

Cuadro 3.6 Subdirectorio de demostración de Routers de la tecnología ZigBee en *DemoRouter*.

Nombre de el archivo	Descripción
Router.c	Archivo fuente de aplicación principal.
DemoPIC18Router.mcp DemoPIC24FRouter.mcp	Archivo del proyecto.
DemoPIC18Router.mcw DemoPIC24FRouter.mcw	Espacio de trabajo.
myZigBee.c	Contiene información específica de aplicación. Es generada por el software ZENA.
zigbee.def	Contiene información específica de aplicación. Es generada por el software ZENA.
zLink.lkr	Enlace manuscrito del proyecto. Es generada por el software ZENA.

Cuadro 3.7 Subdirectorio de demostración de End Device de la tecnología ZigBee en *DemoRFD*.

Nombre de el archivo	Descripción
RFD.c	Archivo fuente de aplicación principal.
DemoPIC18RFD.mcp DemoPIC24FRFD.mcp	Archivo del proyecto.
DemoPIC18RFD.mcw DemoPIC24FRFD.mcw	Espacio de trabajo.
myZigBee.c	Contiene información específica de aplicación. Es generada por el software ZENA.
zigbee.def	Contiene información específica de aplicación. Es generada por el software ZENA.
zLink.lkr	Enlace manuscrito del proyecto. Es generada por el software ZENA.

3.6 USO DE LA PILA DE PROTOCOLOS DE LA TECNOLOGIA ZIGBEE

Siga estos pasos básicos para desarrollar una aplicación del protocolo Zigbee:

1. Determinar el profile que el sistema va utilizar.
2. Determinar la estructura de los Endpoint que el sistema va utilizar.
3. Crear un nuevo proyecto. Coloque todo el código fuente especificado de la aplicación y los archivos de proyecto en este directorio.
4. Utilice el software ZENA para generar los archivos de configuración basado en el tipo de dispositivo, la configuración del dispositivo y la estructura de EndPoint. Comparar los archivos generados por el ZENA con los que vienen con la pila y hacer los cambios apropiados según sea necesario, basado en sus requisitos de hardware específicos.
5. Utilice la aplicación de ejemplo que viene con la pila como una guía en la creación de una nueva aplicación.
6. Agregar el código para la nueva aplicación, incluyendo las inicializaciones extras, cualquier manipulación del ZDO, recepción de mensajes de EndPoint y transmisión, todo tratamiento que no pertenezca al protocolo y manejo de interrupciones.

3.6.1 Interfaz con la pila de protocolos de la tecnología zigbee de Microchip

El código fuente de aplicación debe incluir el encabezado del archivo, zAPL.h, para acceder a las funciones de protocolo ZigBee.

```
#include "zAPL.h"
```

Un Coordinador tendrá que tener una variable de apoyo para mantener un seguimiento de la actual primitiva que está siendo ejecutada por la pila.

```
ZIGBEE_PRIMITIVE currentPrimitive;
```

Un Router o un End Device también tendrán que hacer un seguimiento de la actual primitiva, pero, además, se necesitan dos variables de otro tipo de apoyo para ayudar en la detección de redes y de unión.

```
NETWORK_DESCRIPTOR * currentNetworkDescriptor;
```

```
ZIGBEE_PRIMITIVE currentPrimitive;
```

```
NETWORK_DESCRIPTOR * NetworkDescriptor;
```

En la aplicación se deben configurar todos los pines necesarios para interactuar con el transceiver. El analizador ZENA creará varias etiquetas que se pueden utilizar para acceder a los bits de requerimiento LAT y TRIS.

Antes de que la pila pueda ser utilizada, debe ser inicializada. Para poder inicializarla todas las Interrupciones deben estar habilitadas. Por ejemplo, en la plataforma PIC18, las siguientes líneas de código pueden habilitar las interrupciones:

```
ZigBeeInit (); INTCONbits.RBIE = 1  
RCONbits.IPEN = 1;  
INTCONbits.GIEH = 1;
```

La aplicación interactúa con la pila a través de las primitivas definidas en el protocolo ZigBee y especificaciones IEEE 802.15.4. Las operaciones de la pila se activan llamando a la función, ZigBeeTasks (). La operación de la pila continuará hasta que la primitiva solicitada se complete o en necesidad de que una primitiva de nivel de aplicación necesite ser procesada.

Puesto que solamente una primitiva se puede procesar a la vez, una única estructura de datos se utiliza para guardar todos los parámetros de las primitivas. Esta estructura se puede ver en el archivo, ZigBeeTasks.h. Se debe tener cuidado cuando se accede a esta estructura para evitar sobrescribir un parámetro antes de usarlo. Después de procesar una primitiva, es necesario que la actual primitiva se debe establecer con la próxima primitiva a ejecutar (o NO_PRIMITIVE) para evitar un bucle infinito (Figura 3.3). El flujo general de una aplicación se puede observar en la Figura 3.4

El procesamiento por defecto para la mayoría de las primitivas se incluye en los archivos de aplicación de los ejemplos. Dos primitivas que requieren código adicional específico según la aplicación son:

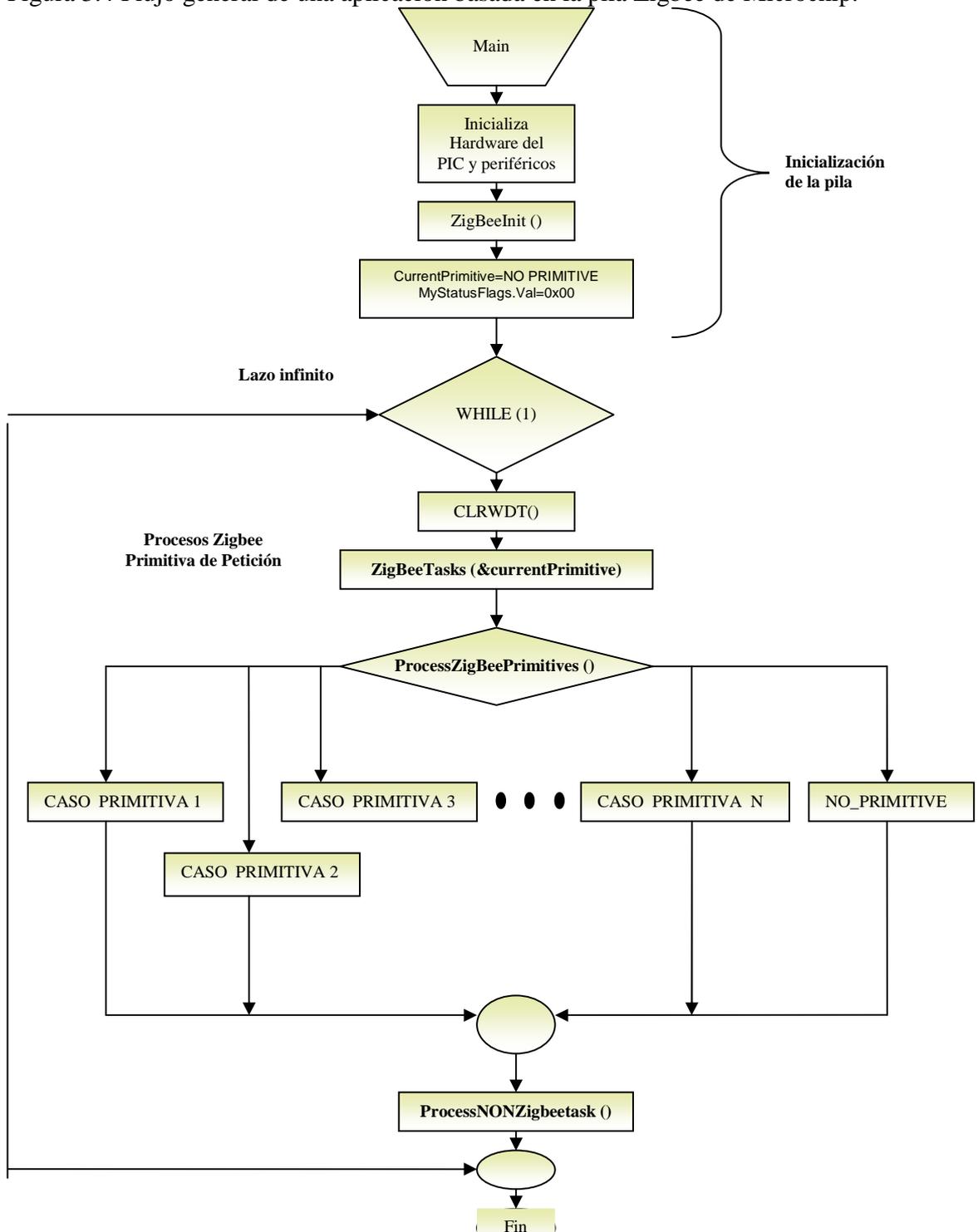
APSDE_DATA_indication y NO_PRIMITIVE.

Figura 3.3 Estructura básica de la aplicación.

```
while (1)  
{  
    /* Limpie el watch dog timer (perro guardian) */  
    CLRWDT();  
  
    /* Procese la actual primitiva del protocolo zigbee */  
    ZigBeeTasks ( &currentPrimitive );  
  
    /* Determine la próxima primitiva del protocolo zigbee para  
    procesar */  
    ProcessZigBeePrimitives();  
  
    /* No realizar ninguna tarea relacionada con el protocolo  
    zigbee aqui*/  
    ProcessNONZigBeeTasks ();  
}
```

Fuente: <http://ww1.microchip.com/downloads/en/AppNotes/ZigBee2006%20Application%20Note%20AN1232A.pdf>

Figura 3.4 Flujo general de una aplicación basada en la pila Zigbee de Microchip.



Fuente: <http://ww1.microchip.com/downloads/en/AppNotes/ZigBee2006%20Application%20Note%20AN1232A.pdf>

3.6.2 Formando o Uniéndose a una Red

El proceso de formar o unirse a una red se muestra en las aplicaciones de demostración. El proceso se inicia en la primitiva NO_PRIMITIVE. Si el dispositivo es un Coordinador, y si no se ha formado una red, entonces iniciará el proceso de intentar formar una red mediante la primitiva NLME_NETWORK_FORMATION_request.

Si el dispositivo no es un coordinador y no se encuentra actualmente en una red, intentará unirse a una. Si el dispositivo ha determinado que estaba anteriormente en una red, entonces intentará unirse como huérfano mediante la primitiva NLME_JOIN_request con el parámetro RejoinNetwork establecido en TRUE.

Si eso falla, o si el dispositivo no estaba previamente en una red, entonces intentará unirse como un nuevo nodo. En primer lugar, éste usará la primitiva NLME_NETWORK_DISCOVERY_request para descubrir qué redes están disponibles.

A continuación selecciona una de las redes descubiertas y tratará de unirse a ella al emitir la primitiva NLME_JOIN_request con el parámetro RejoinNetwork establecido en FALSE.

3.6.3 Captación de mensajes

La pila notifica a la aplicación de los mensajes recibidos a través de la primitiva APSDE_DATA_indication. Cuando esta primitiva es devuelta, los parámetros de la primitiva APSDE_DATA_indication se rellenan con información sobre el mensaje y el mensaje recibido se encuentra en un búfer.

La función, APLGet (), es utilizada para extraer cada byte del mensaje de la memoria intermedia. El parámetro DstEndpoint indica el EndPoint de destino para el mensaje. Si se trata de un EndPoint válido, el mensaje puede ser procesado (ver Figura 3.5).

Para el caso del EndPoint ZDO (EndPoint 0) debe ser incluido para manejar las respuestas a todos los mensajes ZDO enviados por la aplicación.

Después de que el mensaje se procesa, hay que eliminar el mensaje mediante la función APLDiscard (). La falta de eliminación del mensaje no dará lugar a que más mensajes sean procesados.

Figura 3.5 Recepción de Mensajes

```
case APSDE_DATA_indication:
{
    // Declare variables utilizadas por esta primitiva.
    currentPrimitive = NO_PRIMITIVE; // Esto puede variar durante
    // el procesamiento.
    frameHeader = APLGet ();
    switch (params.APSDE_DATA_indication.DstEndpoint)
    {
        case EP_ZDO;
            // Manejo de todas las respuestas de ZDO las cuales se
            // envían.
            break;

            // Incluye los casos para todas las aplicaciones de
            // los dispositivos finales.
    }
    APLDiscard();
}
break;
```

Fuente: <http://ww1.microchip.com/downloads/en/AppNotes/ZigBee2006%20Application%20Note%20AN1232A.pdf>

3.6.4 Envío de mensajes

La pila de Microchip para el Protocolo de ZigBee permite un mensaje de salida en la capa de aplicación a la vez.

Los mensajes se envían mediante los siguientes pasos:

1. Comprobar que la capa de aplicación esté lista para un mensaje saliente, comprobando que ZigBeeReady () es TRUE.
2. Bloquear el sistema con ZigBeeBlockTx () así las llamadas posteriores de ZigBeeReady () devolverá FALSE.
3. Cargar el mensaje en la matriz TxBuffer, utilizando TxData para acceder a través de la matriz. Cuando se complete, TxData debe apuntar a la primera ubicación después del mensaje (es decir, TxData es igual a la longitud de los datos).
4. Cargar los parámetros de la primitiva APSDE_DATA_request.
5. Establecer currentPrimitive como APSDE_DATA_request y llamar la función ZigBeeTasks () .

Los mensajes suelen ser enviados por la aplicación en dos lugares:

- En el procesamiento de la primitiva `APSDE_DATA_indication`, en respuesta a un mensaje recibido.
- En el procesamiento de `NO_PRIMITIVE`, en respuesta a un evento de aplicación.

El proceso de enviar un mensaje es idéntico para ambos lugares. La Figura 3.6 muestra cómo enviar un mensaje de unidifusión a un dispositivo específico de interés. Hay que señalar:

- Cada APS frame debe dirigirse a un EndPoint y Cluster específico dentro del profile.
- TxData debe apuntar a la siguiente ubicación disponible, por lo que se carga utilizando TxBuffer post incremento de direccionamiento.
- Dirigir el mensaje a un EndPoint específico (applet) en el dispositivo de destino.
- Forzar a la utilización de árboles de enrutamiento. De lo contrario, la aplicación podría desencadenar una petición de ruta antes de enviar el mensaje.
- Se solicita el enrutamiento del mensaje, si es posible.

El estado del mensaje transmitido será devuelto a través de la primitiva `APSDE_DATA_confirm`. Se debe tener en cuenta que si el mensaje no se puede transmitir, la pila de forma automática se encargará de volver a intentar transmitir el mensaje.

Figura 3.6 Enviar un Mensaje.

```
if (ZigBeeReady () )
{
    /* Envíe un mensaje a un dispositivo cuya dirección de red es la
    dirección del destinatario. */
    if ( myStatusFlags.bits.bmyBottonWasPressed )
    {
        myStatusFlags.bits.bmyBottonWasPressed = FALSE;

        /* Bloquee la ruta de transmisión antes de cargar el
        payload del mensaje*/
        ZigBeeBlockTx ();

        /* Solicite 10-bytes de datos del dispositivo receptor */
        TxBuffer [TxData ++] = 0x0a;

        /* Utilice unicast, 16-bits de direcciones de red */
        params.APSDE_DATA_request.DstAddrMode = APS_ADDRESS_16_BIT;

        /* Dirección de destino de los dispositivos de interés,
        incluyendo los dispositivos finales */
        params.APSDE_DATA_request.DstAddress.ShortAddr = destinationAddress;
        params.APSDE_DATA_request.DstEndPoint = destinationEndPoint;

        /* Usando el profileID de ZCP y ClusterIDs como un ejemplo*/
        params.APSDE_DATA_request.ProfileId.Val = MY_PROFILE_ID;
        params.APSDE_DATA_request.ClusterId.val = BUFFER_TEST_REQUEST_CLUSTER;

        params.APSDE_DATA_request.SrcEndpoint = EP_BUTTON_APPLET;

        params.APSDE_DATA_request.RadiusCounter = DEFAULT_RADIUS;

        /* Encuentre una ruta si es posible */
        params.APSDE_DATA_request.DiscoverRoute = ROUTE_DISCOVERY_ENABLE;
        params.APSDE_DATA_request.TxOptions.Val = 0;

        /* Solicite un ACK de aplicación en el dispositivo receptor */
        params.APSDE_DATA_request.TxOptions.bits.acknowledged = 1;

        /* Inicie la próxima primitiva para ser ejecutado */
        currentPrimitive = APSDE_DATA_request;
    }
}
```

Fuente: <http://ww1.microchip.com/downloads/en/AppNotes/ZigBee2006%20Application%20Note%20AN1232A.pdf>

3.6.5 Solicitar y recibir datos en un dispositivo RFD

Normalmente los Dispositivos RFD apagan su transceiver cuando están inactivos para conservar la energía, estos dispositivos solicitan sus mensajes cuando su transceiver está encendido mediante la emisión de la primitiva NLME_SYNC_request. La Figura 3.7 muestra una secuencia típica para ir a dormir (apagar) y despertar (encender) utilizando el temporizador Watchdog, o presionando un botón para despertar.

Puede el dispositivo RFD ir a dormir si todas las siguientes condiciones se cumplen:

- No existe ninguna primitiva para ser procesada.
- La pila no está realizando tareas de segundo plano.
- La anterior petición de datos se ha completado.
- Todos los procesos específicos de la aplicación están completos.

Figura 3.7 Solicitar y recibir datos en un dispositivo RFD.

```
// Si no tenemos que ejecutar una primitiva, se ve si necesitamos solicitar
// datos de nuestros padres, o si podemos ir a dormir.
if (currentPrimitive == NO_PRIMITIVE)
{
    if (!ZigBeeStatus.flags.bits.bDataRequestComplete)
    {
        // si nosotros no hemos recibido todos los datos de nuestros padres y no estamos
        // esperando una respuesta de una solicitud de datos, enviamos una solicitud de datos.

        if (!ZigBeeStatus.flags.bits.bRequestingData)
        {
            if (ZigBeeReady() )
            {
                // nuestros padres aún pueden tener datos para nosotros.

                params.NLME_SYNC_request.Track = FALSE;
                currentPrimitive = NLME_SYNC_request;
            }
        }
    }
    else
    {
        if (!ZigBeeStatus.flags.bits.bHasBackgroundTasks &&
            !ZigBeeStatus.flags.bits.bRadioIsSleeping)
        {
            //cuando no se tiene una primitiva a ejecutar, y se haya extraído todos los
            //mensajes que nuestros padres nos tienen y la pila no tiene tareas de segundo
            //plano y todos los procesos de las aplicaciones específicas están completos.

            //Ahora se puede ir a dormir. Asegurándose que la UART haya terminado, apague el
            //Transmisor y asegúrese de despertar al pulsar una tecla.

            while (!ConsoleIsPutReady ());
            MRF24J40Sleep();
            INTCONbits.RBIE = 1;
            SLEEP ();
            NOP();
            //cuando se despierta. Se encienda el transmisor y solicitamos
            //datos de nuestros padres.
            MRF24J40Wake();
            params.NLME_SYNC_request;
        }
    }
}
}
```

Fuente: <http://ww1.microchip.com/downloads/en/AppNotes/ZigBee2006%20Application%20Note%20AN1232A.pdf>

Después de despertar, el RFD debe solicitar los datos de su padre con la primitiva NLME_SYNC_request. El RFD recibirá una de las siguientes respuestas de la emisión de un NLME_SYNC_request:

- Si el padre del RFD ha recibido mensajes para el dispositivo, él los enviará, y el RFD generará una primitiva APSDE_DATA_indication.
- Si el dispositivo principal no tiene ningún mensaje para el RFD, el RFD generará una primitiva NLME_SYNC_confirm con un estado de éxito. Si el RFD no recibe respuesta de su padre, el RFD generará una primitiva NLME_SYNC_confirm con un estado de NWK_SYNC_FAILURE.

3.6.6 Transmisión segura

La pila de Microchip para el Protocolo ZigBee soporta todos los siete modos de seguridad definidos en la especificación del protocolo ZigBee para proteger los paquetes de salida.

Los modos de seguridad se pueden clasificar en tres grupos:

1. Código de integridad del mensaje (MIC) (Message Integrity Code) – Este modo de seguridad garantiza la integridad del paquete. El MIC es adjuntado al paquete (cuyo tamaño está determinado por el modo particular) para asegurar que éste, incluya la cabecera y la carga útil para que no sea modificado en modo alguno durante la transmisión. La carga útil del paquete no se cifra en estos modos.
2. Encriptación (ENC)- Modo de seguridad que encripta la carga útil. El contenido de texto claro de la carga útil no puede ser expuesta sin una clave de seguridad válida. Este modo no puede verificar la integridad del frame o el contenido de la cabecera, incluida la fuente del paquete original y el frame counter.
3. ENC-MIC – Este modo de Seguridad es una combinación de los dos grupos anteriores. En este modo, la carga útil se encripta. Al mismo tiempo, el encabezado y la integridad de la carga útil está protegida por el MIC adjuntado al final del paquete.

Además, también hay modo de seguridad, 0x00, que especifica que no hay seguridad. En esencia, este es el funcionamiento de la pila con el módulo de seguridad desactivado. La capacidad de cada uno de los modos de seguridad se puede encontrar en el cuadro 3.8.

La especificación del protocolo ZigBee también define el apoyo a modos de seguridad Residencial y Comercial, basado en el uso de claves de seguridad. La principal diferencia entre ambos es que el modo comercial requiere la generación de una clave de seguridad individual entre dos nodos durante la comunicación, mientras que el modo residencial

utiliza la clave de red única dentro de la red para proteger los paquetes. Actualmente, la pila de Microchip para el Protocolo ZigBee sólo es compatible con el modo residencial.

La pila soporta redes con o sin una clave de seguridad preconfigurada. La seguridad es apoyada tanto en la NWK o en la capa APL, dependiendo de los requisitos del perfil de aplicación. La seguridad para la capa MAC también se puede habilitar.

La pila añade una cabecera de seguridad auxiliar antes de la carga útil de seguridad de cada paquete asegurado. El formato del encabezado de seguridad auxiliar se puede encontrar en el cuadro 3.9.

Cuadro 3.8 Servicios de seguridad del Protocolo Zigbee.

Modo de seguridad		Servicio de seguridad			Longitud MIC (Bytes)
Identificador	Nombre	Control de Acceso	Encriptación de Datos	Integridad de Trama	
0x01	MIC-32	X		X	4
0x02	MIC-64	X		X	8
0x03	MIC-128	X		X	16
0x04	ENC	X	X		0
0x05	ENC-MIC-32	X	X	X	4
0x06	ENC-MIC-64	X	X	X	8
0x07	ENC-MIC-128	X	X	X	16

Cuadro 3.9 Formato de cabecera de seguridad auxiliar del Protocolo Zigbee.

Ubicación de la seguridad	Paquete de funciones de cabecera			
	Control de Seguridad (1 Byte)	Contador de Trama (4 Bytes)	Dirección extendida de fuente (8 Bytes)	Numero de secuencia de llave (1 Byte)
MAC Layer Security		X		X
NWK Layer Security	X	X	X	X
APL Layer Security	X	X		X

La pila es capaz de garantizar la frescura secuencial comprobando la transmisión del contador de trama. Solo se comprobará el contador de trama de los familiares (padres o hijos), ya que solo los miembros de la familia saben cuando un dispositivo se une a la red. Los paquetes que provienen de miembros de la familia pero que no cumplen el requisito de frescura secuencial serán descartados.

La longitud máxima de un mensaje transmitido es de 127 bytes. Cuando el módulo de seguridad está activado, entre 5 y 29 bytes adicionales son necesarios para la encabezado de seguridad auxiliar y el MCI, dependiendo de el modo de seguridad y de la capa de seguridad. Los usuarios tendrán que equilibrar las necesidades de seguridad y el impacto en el tamaño de carga de datos (y los resultados de impacto) asociados con la combinación de las opciones de seguridad.

El modo de configuración de seguridad y secured layer se definen en el perfil de aplicación. Usar el analizador inalámbrico de redes ZENA herramienta de configuración para crear todas las demás opciones de seguridad.

Una vez que el modo de seguridad se ha definido, para enviar el paquete asegurado es sencillo, sólo una modificación en el código de la aplicación. En la Figura 3.7 se muestra el mismo código que en la Figura 3.9, el código adicional para permitir la transmisión segura se muestra en negrita.

Figura 3.8 Enviar un mensaje seguro.

```
if (ZigBeeReady() )
{
    if (bLightSwitchToggled)
    {
        bLightSwitchToggled = FALSE;
        ZigBeeBlockTx ();

        TxBuffer [TxData ++] = 0x08 /* solicitar 8 bits */

        params.APSDE_DATA_request.DstAddrMode = APS_ADDRESS_16_BIT;
        params.APSDE_DATA_request.DstEndpoint = destinationEndpoint;
        params.APSDE_DATA_request.DstAddress.ShortAddr = destinationAddress;

        params.APSDE_DATA_request.ProfileId.Val = MY_PROFILE_ID;
        params.APSDE_DATA_request.RadiusCounter = DEFAULT_RADIUS;
        params.APSDE_DATA_request.DiscoverRoute = ROUTE_DISCOVERY_ENABLE;
        params.APSDE_DATA_request.TxOptions.Val = 0;
params.APSDE_DATA_request.TxOptions.bits.securityEnabled = 1;
        params.APSDE_DATA_request.SrcEndpoint = EP_SWITCH;
        params.APSDE_DATA_request.ClusterId.Val = BUFFER_TEST_REQUEST_CLUSTER;

        currentPrimitive = APSDE_DATA_request;
    }
}
```

Fuente: <http://ww1.microchip.com/downloads/en/AppNotes/ZigBee2006%20Application%20Note%20AN1232A.pdf>

CAPITULO 4

PRUEBAS DE LA VALIDACIÓN DE LA TECNOLOGÍA ZIGBEE DE MICROCHIP

La primera prueba que se realizó fue comprobar la topología que maneja la tecnología ZigBee de Microchip.

4.1 COMPROBACIÓN DE LA TOPOLOGÍA PUNTO A PUNTO

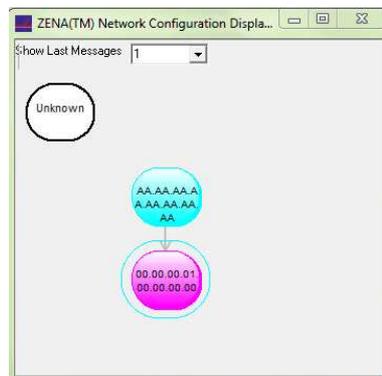
La primera topología que se comprobó fue la topología punto a punto.

Figura 4.1 Topología punto a punto.



Para comprobar esta topología se utilizó la ventana NCD del programa zena, el cual muestra la formación de la red que hay en ese momento, y el hyperterminal de Windows.

Figura 4.2 Red punto a punto (Zena).



En la Figura 4.2 se observa la red punto a punto que capta el analizador de redes inalámbricas ZENA; para comprobar como se forma esta red, se utilizó el Hyperterminal de windows conectado al coordinador, en la pantalla de la sesión del Hyperterminal se pudo observar cuando se une el nuevo nodo (dispositivo) al coordinador, se realizó un envío de datos hacia el dispositivo RFD para comprobar que estuviera transmitiendo datos el dispositivo, del mismo modo se envió una solicitud de datos desde el coordinador hacia el RFD para que éste enviara una trama de datos, todo este proceso ocurrido en el Hyperterminal se observa a continuación:

Microchip ZigBee2006(TM) Stack v2.0-2.6.0a Coordinator
Transceiver-MRF24J40

Trying to start network...
PAN 1AAA started successfully.
Joining permitted.

- 1: Enable/Disable Joining by Other Devices
- 2: Request Data From Another Device
- 3: Request Data From a Group of Devices
- 4: Send Data To Another Device
- 5: Send Data To a Group of Devices
- 6: Add/Remove Device to/from a Group
- 7: Dump Neighborhood Information

Enter a menu choice:

Node 0001 With MAC Address 0000000100000000 just joined. */ Se unió el dispositivo
A la red

(2) */ se realizó una solicitud desde el otro dispositivo

How many bytes are you requesting(hex): 0a

What is the short address of device you want data from: 0001

- 1: Enable/Disable Joining by Other Devices
- 2: Request Data From Another Device
- 3: Request Data From a Group of Devices
- 4: Send Data To Another Device
- 5: Send Data To a Group of Devices
- 6: Add/Remove Device to/from a Group
- 7: Dump Neighborhood Information

Enter a menu choice:

Message sent successfully. */ La solicitud del dato que se hizo fue exitosa

Len: 0A

From Address: 0001

0000010203040506070809

(4) */ se realizó un envío de datos hacia el otro dispositivo

Please enter the number of bytes to send (hex): 0a

Please enter the short address of the destination device: 0001

- 1: Enable/Disable Joining by Other Devices
- 2: Request Data From Another Device
- 3: Request Data From a Group of Devices
- 4: Send Data To Another Device
- 5: Send Data To a Group of Devices
- 6: Add/Remove Device to/from a Group
- 7: Dump Neighborhood Information

Enter a menu choice: Message sent successfully. */ El envío del dato fue exitoso

4.2 COMPROBACIÓN DE LA TOPOLOGÍA ÁRBOL

Figura 4.3 Topología Árbol.

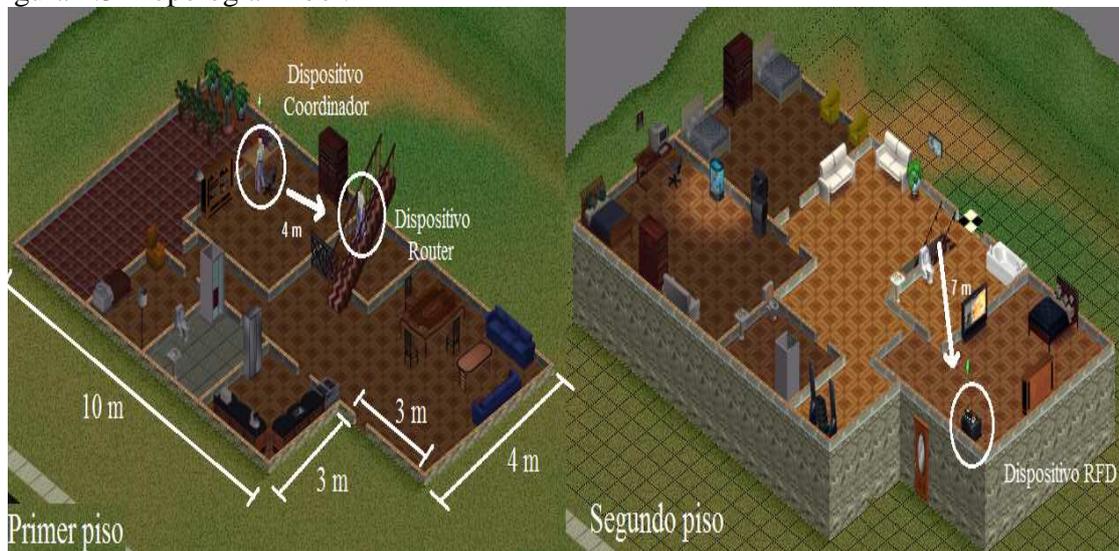


Figura 4.4 Red Árbol (Zena).

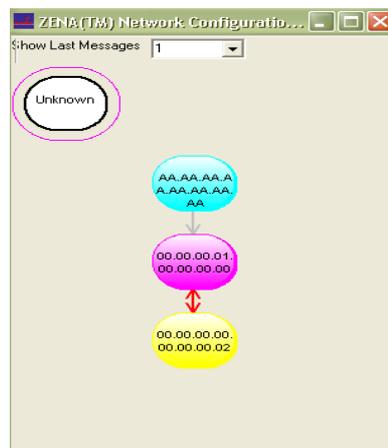


Figura 4.5 Trama red Árbol (Zena).

Frame	Time(us)	Len	MAC Frame Control	Type	Sec	Pend	ACK	IPAN	Seq Num	Dest PAN	Dest Addr	Source Addr	NWK Frame Control	Type	Ver	Route	Sec	Dest Addr	Source Addr	Radius	Seq Num	APS Header	Seq Num	APS Payload	FCS	RSI	Corr	CRC	
00071	+442816 +63180592	29	DATA	N	N	Y	Y		0x55	0x1AAA	0x0001	0x1430	DAT	0x2	SUP	N	0xFFFF	0x1430	0x0A	0x02	0x0C	0x04	0x00	0x1C	0x0A	+18	0x6B	OK	
00072	+1728 +63182320	5	ACK	N	N	N	N		0x55	+02	0x69	OK																	
00073	+489568 +63671888	29	DATA	N	N	Y	Y		0x39	0x1AAA	0xFFFF	0x0001	DAT	0x2	SUP	N	0xFFFF	0x1430	0x09	0x02	0x0C	0x04	0x00	0x1C	0x0A	+02	0x6B	OK	
00074	+18240 +63690128	39	DATA	N	N	Y	Y		0x1A	0x1AAA	0x0001	0x0000	DAT	0x2	SUP	N	0x1430	0x0000	0x0A	0x01	0x00	0x04	0x54	0x00	0x0A	0x00	0x01	0x02	0x03
00075	+2256 +63692384	5	ACK	N	N	N	N		0x1A	+02	0x6B	OK																	
00076	+539360 +64231744	12	CHD	N	N	Y	Y		0x56	0x1AAA	0x0001	0x1430	Data Request												+19	0x6B	OK		
00077	+832 +64232576	5	ACK	N	Y	N	N		0x56	+00	0x6B	OK																	
00078	+9616 +64242192	39	DATA	N	N	Y	Y		0x38	0x1AAA	0x1430	0x0001	DAT	0x2	SUP	N	0x1430	0x0001	0x0A	0x02	0x00	0x04	0x54	0x00	0x0A	0x00	0x01	0x02	0x03
00079	+2256 +64244448	5	ACK	N	N	N	N		0x38	+19	0x6A	OK																	
00080	+39680 +64284128	12	CHD	N	N	Y	Y		0x57	0x1AAA	0x0001	0x1430	Data Request													+19	0x6B	OK	
00081	+848 +64284976	5	ACK	N	Y	N	N		0x57	+00	0x6B	OK																	
00082	+8464 +64293440	39	DATA	N	N	Y	Y		0x3A	0x1AAA	0x1430	0x0001	DAT	0x2	SUP	N	0x1430	0x0000	0x09	0x01	0x00	0x04	0x54	0x00	0x0A	0x00	0x01	0x02	0x03
00083	+2256 +64293696	5	ACK	N	Y	N	N		0x3A	+19	0x6A	OK																	

Para la comprobación de la topología de red Árbol se utilizaron 3 dispositivos (coordinador, router y dispositivo final) los cuales formaron la red de la Figura 4.4 la cual fue implementada en un ambiente con obstáculos (paredes) comprobando su funcionalidad; del mismo modo se implementó en un ambiente abierto. Para mirar su funcionamiento y que esté cumpliendo con los principios de esta red, se utilizó el analizador de redes inalámbricas ZENA el cual entregó un resultado de transmisión de paquetes como lo muestra la Figura 4.5.

Enviando una solicitud desde el dispositivo final, con dirección 1430, hacia los demás dispositivos que se encuentran en la red como lo son el coordinador y router, para el envío de 10 bits desde éstos al dispositivo final.

Si se observa la Figura 4.5, el Frame 71 muestra la petición o solicitud del dispositivo final, si se observa el NWK Frame Control, la dirección de origen de la solicitud es 1430, la cual pertenece al dispositivo final, hacia los demás dispositivos de la red, ahora si se observa el MAC Frame Control, el cual decide a que dispositivo va dirigido el mensaje, al ser el 1430 un dispositivo RFD no puede hacer conexión directa con el coordinador (0000) por que éste tiene una vinculación directa con el router, por con siguiente la dirección de destino va ser 0001 la cual pertenece al router.

Una vez el router recibe esta trama (Frame) observa el NWK Frame Control y determina el destino final de la solicitud, la cual es FFFF que quiere decir que va dirigida a todos los dispositivos de la red, y enruta el mensaje hacia los dispositivos de la red para que reciban el Frame generado por 1430, este proceso se ve en el Frame 73 la figura 4.5.

El coordinador (0000) al recibir el mensaje del router, envía los datos solicitados, estos datos se ven en la capa APS de el mensaje (APS Payload), hacia el dispositivo que lo solicitó, esto se ve descrito en el NWK Frame Control de el Frame 74 de la Figura 4.5. Como el dispositivo de donde se recibió el mensaje fue el router y éste pertenece al árbol del dispositivo final 1430, entonces la capa MAC direcciona este mensaje hacia la dirección 0001 la cual pertenece al router, esto se puede ver en el MAC Frame Control de el Frame 74 de la Figura 4.5.

Una vez el coordinador termina de enviar el mensaje, el dispositivo final envía un Beacon Request solicitando a su padre (router) si tiene algún mensaje para él. Esto se puede ver en el Frame 76, el router responde con el mensaje que fue solicitado a él y luego de otro Beacon Request del dispositivo final envía el del coordinador, esto se puede observar en los Frames 78 y 82. Se comprobó así que esta topología es soportada por la pila de protocolos de la tecnología Zigbee de Microchip.

4.3 COMPROBACIÓN DE LA TOPOLOGÍA ESTRELLA

Figura 4.6 Topología estrella.

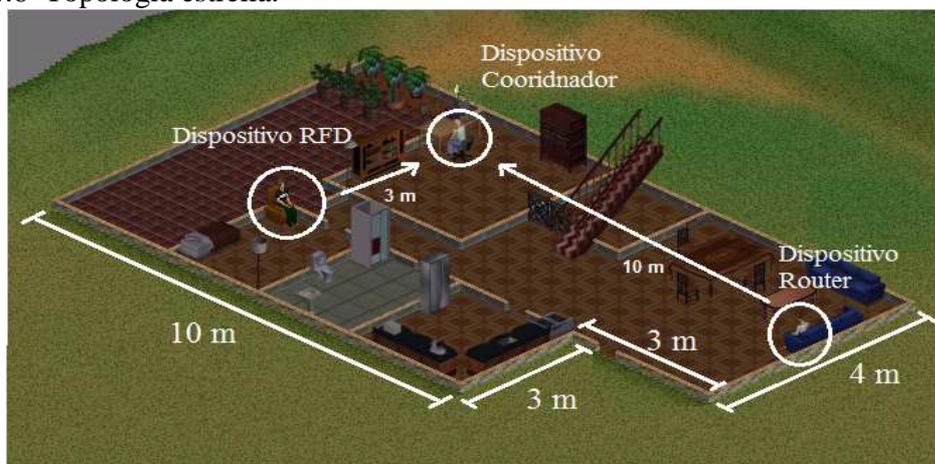


Figura 4.7 Red Estrella (Zena).

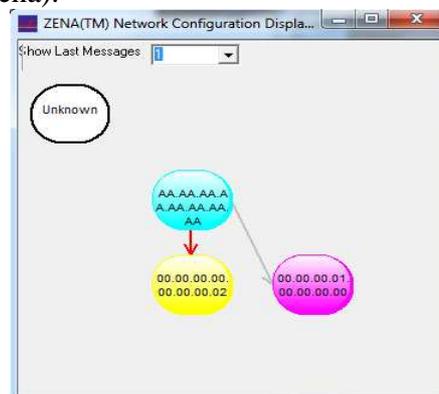


Figura 4.8 Trama red Estrella (Zena).

Frame	Time(us)	Len	MAC Frame Control	Seq Num	Dest PAN	Dest Addr	Source Addr	NWK Frame Control	Dest Addr	Source Addr	Radius	Seq Num	APS Header	APS Payload	FCS
00044	+537088 +22262704	29	DATA N N Y Y	0x50	0x1AAA	0x0000	0x796F	DATA 0x2 SUP N	0xFFFF	0x796F	0x0A	0x00	0x0C 0x04 0x00 0x1C	0x0A	-13 0x6B OK
00045	+1729 +22264432	5	ACK N N N N	0x50											
00046	+639136 +22903568	29	DATA N N Y Y	0x22	0x1AAA	0xFFFF	0x0000	DATA 0x2 SUP N	0xFFFF	0x796F	0x03	0x00	0x0C 0x04 0x00 0x1C	0x0A	-07 0x6B OK
00047	+317936 +23221504	12	CMD N N Y Y	0x51	0x1AAA	0x0000	0x796F	DATA Request							-17 0x6A OK
00048	+892 +23222336	5	ACK N Y N N	0x51											
00049	+7760 +23230096	39	DATA N N Y Y	0x21	0x1AAA	0x0000	0x796F	DATA 0x2 SUP N	0xFFFF	0x0000	0x0A	0x00	0x00 0x04 0x54 0x00	0x0A 0x00 0x00 0x01 0x00	0x07 0x00
00050	+2256 +23232352	5	ACK N N N N	0x21											
00051	+39968 +23272320	12	CMD N N Y Y	0x52	0x1AAA	0x0000	0x796F	DATA Request							-17 0x67 OK
00052	+848 +23273168	5	ACK N Y N N	0x52											
00053	+5232 +23278400	11	DATA N N N Y	0x23	0x1AAA	0x796F	0x0000								-08 0x68 OK
00061	+741088 +31089808	18	CMD N N Y Y	0x35	0x1AAA	0x00000000	0x00000000	DATA Request							-07 0x6B OK
00062	+1152 +31090960	5	ACK N Y N N	0x35											
00063	+6976 +31097936	27	CMD N N Y Y	0x26	0x1AAA	0x00000000	0x00000000	Association Response	Success 0x001						-13 0x69 OK
00064	+1616 +31099552	5	ACK N N N N	0x26											
00065	+64160 +31163712	39	DATA N N N Y	0x36	0x1AAA	0xFFFF	0x0001	DATA 0x2 SUP N	0xFFFF	0x0001	0x0A	0x21	0x08 0x00 0x13 0x00	0x00 0x00 0x01 0x00 0x00 0x00 0x00	
00066	+43376 +31507088	39	DATA N N N Y	0x27	0x1AAA	0xFFFF	0x0000	DATA 0x2 SUP N	0xFFFF	0x0000	0x0A	0x21	0x08 0x00 0x13 0x00	0x00 0x00 0x01 0x00 0x00 0x00 0x00	
00067	+5816528 +37623616	29	DATA N N N Y	0x37	0x1AAA	0xFFFF	0x0001	DATA 0x2 SUP N	0xFFFF	0x0001	0x0A	0x22	0x0C 0x04 0x00 0x1C	0x0A	-08 0x6E OK
00068	+17040 +37640656	39	DATA N N Y Y	0x29	0x1AAA	0x0000	0x0000	DATA 0x2 SUP N	0xFFFF	0x0000	0x0A	0x02	0x00 0x04 0x54 0x00	0x0A 0x00 0x00 0x01 0x00	
00069	+2256 +37642912	5	ACK N N N N	0x29											

Para la comprobación de la topología de red Estrella se utilizaron 3 dispositivos (coordinador, router y dispositivo final) los cuales formaron la red de la Figura 4.7 la cual fue implementada en un ambiente con obstáculos (paredes) comprobando su funcionalidad; del mismo modo se implementó en un ambiente abierto. Para mirar su funcionamiento y que esté cumpliendo con los principios de esta red, se utilizó el analizador de redes inalámbricas ZENA el cual nos entregó un resultado de transmisión de paquetes como lo muestra la Figura 4.8.

El proceso que se describe en la Figura 4.8, son enviadas solicitudes desde el dispositivo final y desde el router, con direcciones 796F y 0001 respectivamente, hacia el coordinador para el envío de una trama de 10 bits.

Si se observa la Figura 4.8, el Frame 44 muestra la petición o solicitud del dispositivo final, si se observa el NWK Frame Control, la dirección de origen de la solicitud es 796F, la cual pertenece al dispositivo final, hacia los demás dispositivos de la red, ahora si se observa el MAC Frame Control, el cual decide a qué dispositivo va dirigido el mensaje, debido a que

el dispositivo final está unido directamente a el coordinador este mensaje es enviado hacia él.

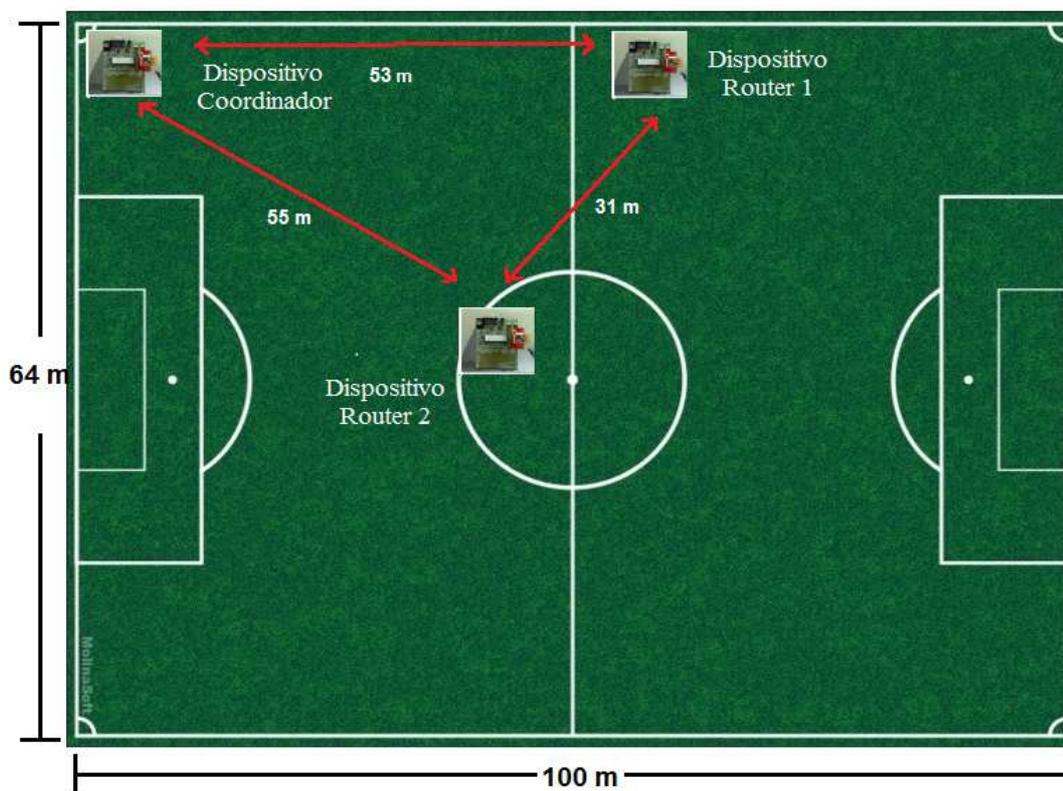
Una vez el coordinador recibe la solicitud de envío por el dispositivo final, éste envía el mensaje solicitado hacia el destino de origen, esto se puede observar en el Frame 49 de la Figura 4.8.

Ahora si la solicitud es enviada por el router como lo muestra el Frame 67 de la Figura 4.8, si se observa el NWK Frame Control, la dirección de origen es 0001 la cual pertenece al router y va dirigida a el coordinador, una vez el coordinador recibe esta solicitud, el envía el mensaje a el coordinador como se ve en el Frame 68 terminando así el proceso del mensaje.

Se Comprobó así que la pila de protocolos Zigbee de Microchip soporta la formación de esta topología, como ya se había comprobado con la red punto a punto la cual es una red de topología Estrella que utiliza dos dispositivos (coordinador y router (FFD) ó dispositivo final (RFD)).

4.4 COMPROBACIÓN DE LA TOPOLOGÍA MALLA

Figura 4.9 Topología malla.



Enviando una solicitud desde el router 1, con dirección 0001, hacia los demás dispositivos que se encuentran en la red como lo son el coordinador y router 2, para el envío de 10 bits desde éstos al router 1.

Si se observa la Figura 4.11, el Frame 24 muestra la petición o solicitud del router 1, si se observa el NWK Frame Control, la dirección de origen de la solicitud es 0001, la cual pertenece al router 1, hacia los demás dispositivos de la red, ahora si se observa el MAC Frame Control, el cual decide a qué dispositivo va dirigido el mensaje, al ser un dispositivo FFD éste puede direccionar directamente el mensaje a toda la red.

Una vez el coordinador (0000) y el router 2 (0002) al recibir el mensaje del router 1, envían los datos solicitados, estos datos se ven en la capa APS de el mensaje (APS Payload), hacia el dispositivo que lo solicitó, esto se ve descrito en el NWK Frame Control de los Frames 25y 28 de la Figura 4.11. Llegando con éxito a su destino.

Se Comprobó así que la pila de protocolos Zigbee de Microchip soporta la formación de esta topología

4.5 PRUEBA DE DISTANCIA

Para la prueba de distancia se trabajó con una red Punto a Punto, en este caso con las dos Tarjetas madres que tiene el Kit de desarrollo PICDEM Z.

Gracias a estas Tarjetas madres que contienen LEDs indicadores, se pudo realizar la prueba de máxima distancia en el cual se podría obtener una comunicación entre estos dos dispositivos.

Figura 4.12 Led indicador de la tarjeta madre



Cuadro 4.1 Características del Led indicador de la tarjeta madre

Led	Estado	Función
D1	ON	El nodo se encuentra con alimentación y se realiza la inicialización de este.
D2	ON	El nodo pertenece al grupo asignado, en este caso el grupo 4.
	OFF	El nodo no pertenece al grupo asignado.

Los pasos que se realizaron fueron:

- 1) inicializar el coordinador
- 2) inicializar el router
- 3) unir el coordinador al grupo, pulsando el pushbotton RB4

- 4) unir el router al grupo, pulsando el pushbotton RB4
 - 5) enviar un paquete de prueba pulsando el pushbotton RB5 en la tarjeta madre del coordinador.
 - 6) Con este proceso se debería apagar el led RA1 como se observa en la grafica anterior.
 - 7) Si se realiza nuevamente el paso 5, el led RA1 debería encenderse nuevamente.
 - 8) El coordinador se dejó fijo y se desplazó el router realizando los pasos 5,6,7 hasta que se perdiera la comunicación entre estos dos dispositivos, así se halló la máxima distancia
- La máxima distancia obtenida fue 120 metros

Figura 4.13 Máxima distancia alcanzada.



4.6 PRUEBAS DE POTENCIA VS DISTANCIA

- El objetivo de esta prueba era medir la potencia de transmisión en un Nodo.
- Se utilizó la herramienta proporcionada por MICROCHIP llamada Zena, la cual puede capturar, filtrar y decodificar paquetes **MAC IEEE 802.15.4** y los visualiza gráficamente en pantalla.

Como el elemento MRF24J40MA tiene incorporado un **RSSI** (Received Signal Strength Indicator), el cual entrega un valor digital de 8 bits, señalizado por el registro **RSSI.RSSI_VAL**, el **analizador Zena** lo leerá y visualizará su valor.

Para la realización de esta prueba se realizaron los siguientes pasos:

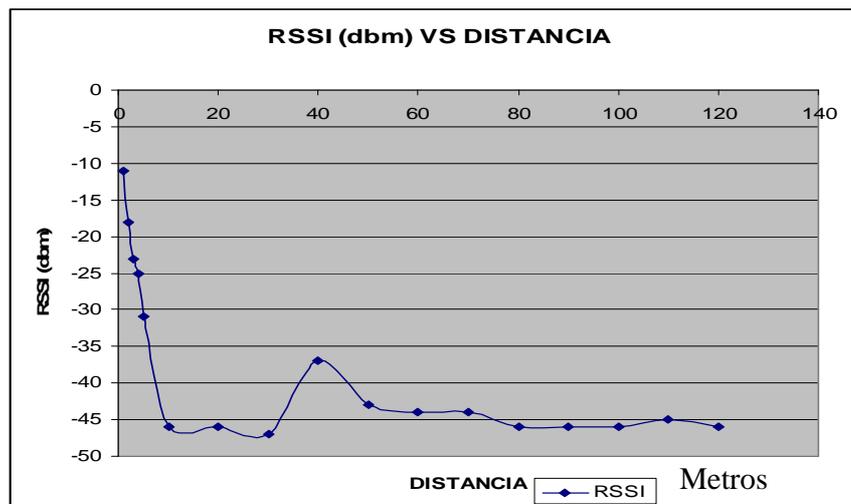
- 1) Se introdujo el programa Demo18FCoordinador.Hex en el PIC de una tarjeta madre.
- 2) Se conectó el hardware Zena en la PC, y se abrió el programa Microchip ZENA™.
- 3) Se ingresó al monitor de tráfico de red.
- 4) Se envió un mensaje de prueba desde el coordinador pulsando el pushbotton RB5
- 5) Desde el monitor de tráfico de red, se observaron los datos que se enviaron desde el dispositivo coordinador.
- 6) La primera prueba se realizó desde 1 metro, y se fue alejando hasta perder la comunicación entre la tarjeta madre y el Hardware Zena.

- Los datos de potencia Vs distancia fueron:

Cuadro 4.2 Datos RSSI adquiridos por el zena

Distancia (metros)	RSSI (dBm)	Distancia (metros)	RSSI (dBm)
1	-11	50	-43
2	-18	60	-44
3	-23	70	-44
4	-25	80	-46
5	-31	90	-46
10	-46	100	-46
20	-46	110	-45
30	-47	120	-46
40	-37		

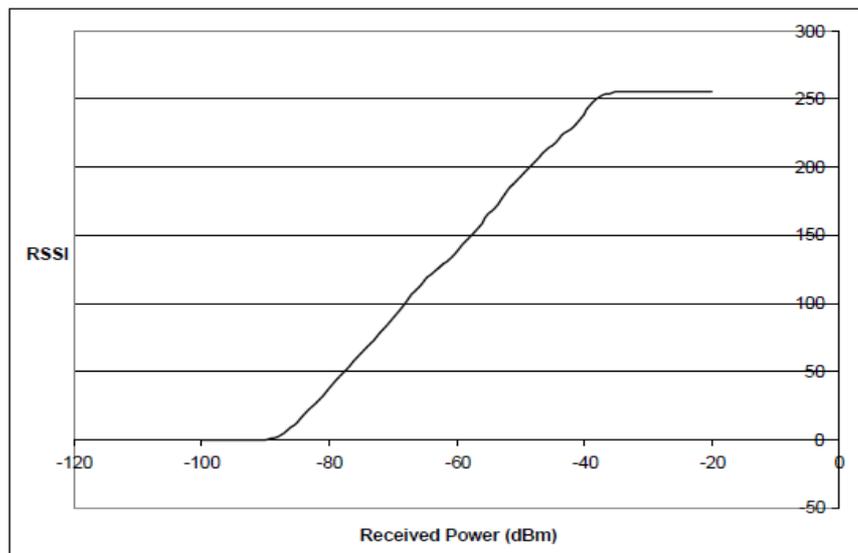
Figura 4.14 RSSI vs. DISTANCIA



El valor **RSSI** del registro **RSSI.RSSI_VAL** puede ser referido a la potencia **P** de los pines RF, usando la ecuación: $P = \text{RSSI_VAL} + \text{RSSI_OFFSET}$ [dBm]; donde el **RSSI_OFFSET** se halla empíricamente durante el desarrollo del sistema, a partir de la ganancia final.

RSSI_OFFSET tiene aproximadamente -45 dBm. Por ejemplo si la lectura es un valor de -11 dBm desde el registro **RSSI**, la potencia de entrada RF es de aproximadamente -56 dBm. Una gráfica típica de la lectura **RSSI_VAL** como función de la potencia de entrada se muestra en la Figura 4.15.

Figura 4.15 Gráfica típica de lectura RSSI VS Potencia de entrada.



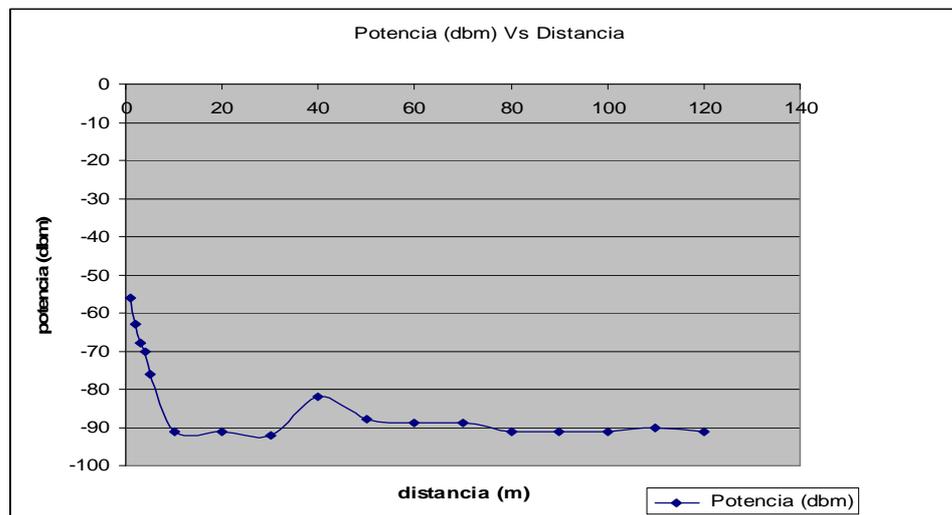
Fuente: <http://ww1.microchip.com/downloads/en/DeviceDoc/DS-39776b.pdf>

- Por tal razón los nuevos valores de potencia son:

Cuadro 4.3 Datos RSSI utilizando la ecuación $P = \text{RSSI_VAL} + \text{RSSI_OFFSET}$

Distancia (metros)	RSSI (dBm)	Distancia (metros)	RSSI (dBm)
1	-56	50	-88
2	-63	60	-89
3	-68	70	-89
4	-70	80	-91
5	-76	90	-91
10	-91	100	-91
20	-91	110	-90
30	-92	120	-91
40	-82		

Figura 4.16 Potencia (dbm) vs Distancia.



- En esta parte se mostrarán las tramas adquiridas por el analizador Zena a diferentes distancias y se mostrarán las fotos de las pruebas.

Figura 4.17 tramas a diferentes distancias

Frame	Time(us)	Len	MAC Frame Control	Type	Sec	Pend	ACK	IPAN	Seq Num	Dest PAN	Dest Addr	Reason	FCS			
00003	+334848 -9741040	10	MAC Frame Control	CMD	N	N	N	N	0x16	0x1AAA	0xFFFF	Request	RSSI Corr CRC -15 0x68 OK			
00004	+8380720 -18121760	29	MAC Frame Control	DATA	N	N	N	Y	0x17	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x63	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x00	APS Payload 0x0A	FCS RSSI Corr CRC -11 0x6A OK
00005	+62237408 -80359168	29	MAC Frame Control	DATA	N	N	N	Y	0x17	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x64	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x01	APS Payload 0x0A	FCS RSSI Corr CRC -18 0x69 OK
00006	+20750496 -101109664	29	MAC Frame Control	DATA	N	N	N	Y	0x18	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x65	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x02	APS Payload 0x0A	FCS RSSI Corr CRC -23 0x6C OK
00007	+13884272 -114993936	29	MAC Frame Control	DATA	N	N	N	Y	0x1A	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x66	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x03	APS Payload 0x0A	FCS RSSI Corr CRC -25 0x66 OK
00008	+21747536 -136741472	29	MAC Frame Control	DATA	N	N	N	Y	0x1A	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x67	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x04	APS Payload 0x0A	FCS RSSI Corr CRC -31 0x67 OK
00009	+21207696 -157949168	29	MAC Frame Control	DATA	N	N	N	Y	0x1C	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x68	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x05	APS Payload 0x0A	FCS RSSI Corr CRC -46 0x49 OK
00010	+43943472 -201892640	29	MAC Frame Control	DATA	N	N	N	Y	0x1E	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x6A	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x07	APS Payload 0x0A	FCS RSSI Corr CRC -46 0x53 OK
00011	+53957856 -255805496	29	MAC Frame Control	DATA	N	N	N	Y	0x20	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x6C	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x09	APS Payload 0x0A	FCS RSSI Corr CRC -47 0x39 OK
00012	+7786848 -674035744	29	MAC Frame Control	DATA	N	N	N	Y	0x1E	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x63	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x08	APS Payload 0x0A	FCS RSSI Corr CRC -37 0x69 OK
00013	+52981536 -762020448	29	MAC Frame Control	DATA	N	N	N	Y	0x22	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x67	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x0C	APS Payload 0x0A	FCS RSSI Corr CRC -43 0x66 OK
00014	+18228384 -845171776	29	MAC Frame Control	DATA	N	N	N	Y	0x26	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x6E	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x10	APS Payload 0x0A	FCS RSSI Corr CRC -44 0x51 OK
00015	+261158944 -1110330720	29	MAC Frame Control	DATA	N	N	N	Y	0x29	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0xFE	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x13	APS Payload 0x0A	FCS RSSI Corr CRC -44 0x65 OK
00016	+86416048 -1196746768	29	MAC Frame Control	DATA	N	N	N	Y	0x33	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x09	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x1D	APS Payload 0x0A	FCS RSSI Corr CRC -46 0x50 OK
00017	+84347760 -1281094528	29	MAC Frame Control	DATA	N	N	N	Y	0x39	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x0F	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x23	APS Payload 0x0A	FCS RSSI Corr CRC -46 0x51 OK
00018	+16600320 -1640514592	29	MAC Frame Control	DATA	N	N	N	Y	0x4E	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x21	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x35	APS Payload 0x0A	FCS RSSI Corr CRC -46 0x49 OK
00019	+20831216 -1717075120	29	MAC Frame Control	DATA	N	N	N	Y	0x51	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x27	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x3B	APS Payload 0x0A	FCS RSSI Corr CRC -45 0x5A OK
00020	+73318992 -1790394112	29	MAC Frame Control	DATA	N	N	N	Y	0x60	0x1AAA	0xFFFF	0x0000	NWK Header 0x08 0x00 0xFF 0xFF 0x00 0x0A 0x36	APS Header 0x0C 0x04 0x00 0x1C 0x01 0x7F 0x04 0x4A	APS Payload 0x0A	FCS RSSI Corr CRC -46 0x51 OK

4.7 FOTOS DE PRUEBA (POTENCIA VS DISTANCIA)

Figura 4.18 prueba a 30 metros

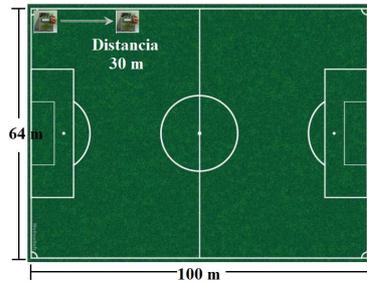


Figura 4.19 prueba a 50 metros

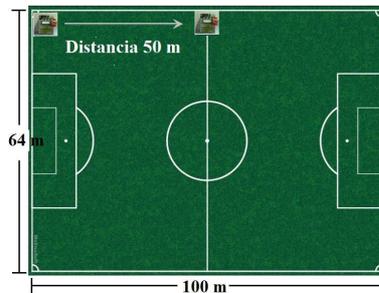


Figura 4.20 prueba a 80 metros

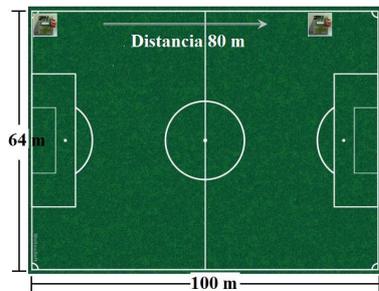
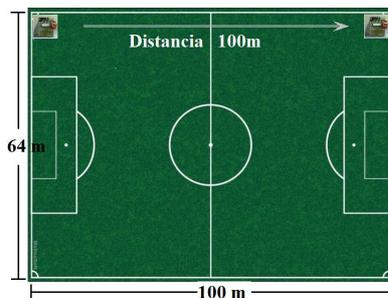


Figura 4.21 prueba a 100 metros



CAPÍTULO 5

CREACIÓN E IMPLEMENTACIÓN DEL SOFTWARE

El presente capítulo justifica la creación e implementación del software utilizando el ambiente de desarrollo gráfico LABVIEW para la tecnología Zigbee.

Se relacionan los requerimientos de un sistema de monitoreo de temperatura con las características brindadas por la tecnología Zigbee, permitiendo con ello generar un bosquejo general del sistema, válido para el diseño e implementación del hardware y software del proyecto.

5.1 SENSOR ELECTRÓNICO

Un **sensor**¹ es un aparato capaz de transformar magnitudes físicas o químicas, llamadas variables de instrumentación, en magnitudes eléctricas. Las variables de instrumentación pueden ser por ejemplo: **temperatura**, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, pH, etc.

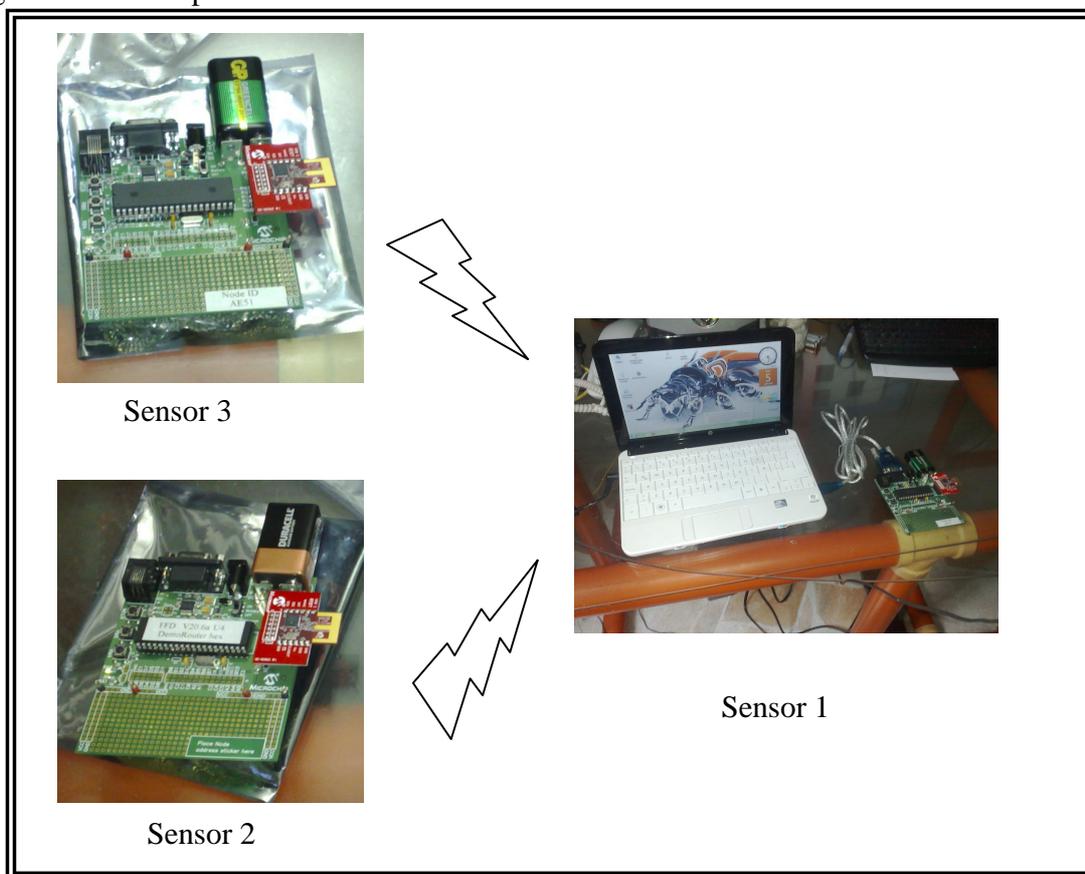
Un sensor se diferencia de un transductor en que el sensor está siempre en contacto con la variable de instrumentación con lo que puede decirse también que es un dispositivo que aprovecha una de sus propiedades con el fin de adaptar la señal que mide para que la pueda interpretar otro dispositivo. Como por ejemplo el termómetro de mercurio que aprovecha la propiedad que posee el mercurio de dilatarse o contraerse por la acción de la temperatura. Un sensor también puede decirse que es un dispositivo que convierte una forma de energía en otra.

5.2 DESCRIPCIÓN DEL SISTEMA DE RED IMPLEMENTADO

La pequeña red Zigbee a implementar para el monitoreo de temperatura consta de tres nodos (un nodo coordinador y dos nodos routers) que se comunican inalámbricamente, a su vez el nodo coordinador, quien inicializa y mantiene la red, transfiere los datos de las temperaturas al computador por medio de comunicación serial; la información que envía el coordinador al computador es procesada por una interfaz gráfica diseñada en LABVIEW la cual grafica y muestra la temperatura de los nodos, al mismo tiempo la almacena en un archivo Excel.

¹ <http://es.wikipedia.org/wiki/Sensor> Sensor electrónico

Figura 5.1 Red implementada.



Los procesos de adquisición de datos de temperatura son de flujo continuo, ya que se maneja el proceso de sensado en tiempo real, siendo el coordinador quien solicita el sensado de la temperatura a los routers para que ellos respondan con el envío de la temperatura y así el coordinador forme un paquete de información de temperatura el cual se envía al computador para su proceso;

Al ser un modelo prototipo y teniendo en cuenta distintas limitaciones de accesibilidad y costo para conseguir los dispositivos y herramientas necesarias (microcontroladores, programador, software, etc.), se planteó como objetivo demostrar la funcionalidad de la pila de protocolos de la tecnología Zigbee de Microchip a través de la implementación de una red básica la cual está compuesta por 1 coordinador y 2 routers, por esta razón se ha escogido la topología malla.

El suministro de energía DC a cualquiera de los nodos se realiza mediante una batería o una fuente regulada del mismo voltaje. Primordialmente, sólo el coordinador utiliza una fuente regulada.

El computador utilizado para la recolección de datos del coordinador debe cumplir con algunos requerimientos básicos:

Disponer de un puerto serial RS232 para la comunicación con el coordinador, si no lo posee debe tener un puerto USB para conectar a este un cable conversor de USB a RS232.
Trabajar con sistema operativo Windows xp o vista para la ejecución del programa a crear en el ambiente de desarrollo grafico **LABVIEW 8.6**.

El software desarrollado tiene como objetivos:

- Capturar los datos, que son enviados provenientes del coordinador de la red Zigbee y realizar una base de datos la cual contenga la fecha, temperatura de cada sensor y la hora de sensado.
- Mostrar la temperatura de cada sensor en una grafica.
- Generar una alarma de forma grafica cuando se sobrepasen los valores mayores o menores a los cuales se hayan calculado en los sistemas de sensado.
- Crear una grafica la cual genere los valores de las temperaturas sensadas en la variación del tiempo (Temperatura vs. Tiempo).
- Crear un string en el cual se pueda seleccionar el puerto COM para la comunicación entre el dispositivo Zigbee (coordinador), con el PC.

5.3 DESARROLLO DEL SOFTWARE EN EL ENTORNO GRAFICO DE PROGRAMACIÓN LABVIEW

Teniendo en cuenta las necesidades que se poseían, se desarrolló un software en el ambiente de desarrollo grafico (Labview).

Figura 5.2 Monitoreo de temperatura en la red Zigbee (panel).

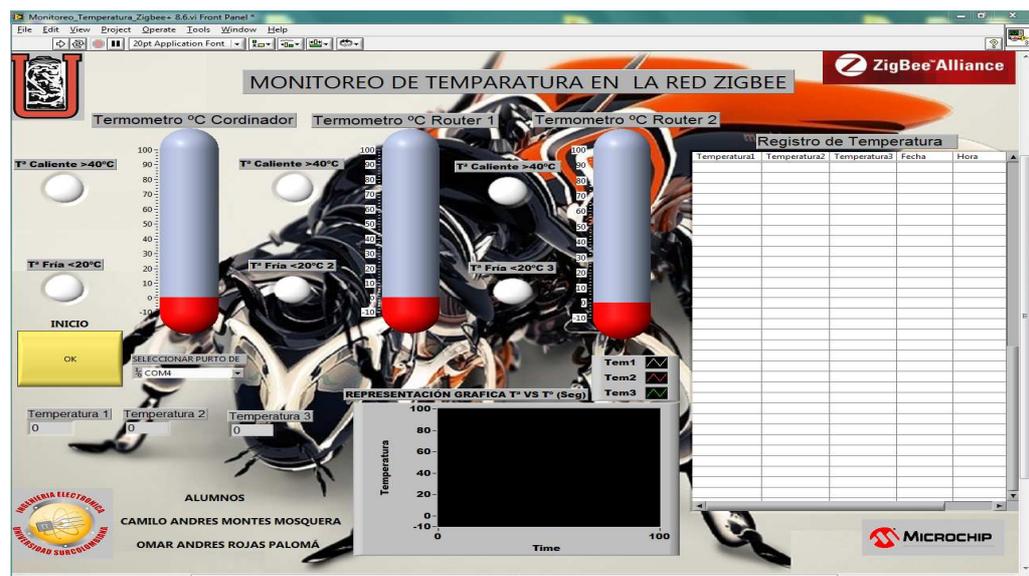
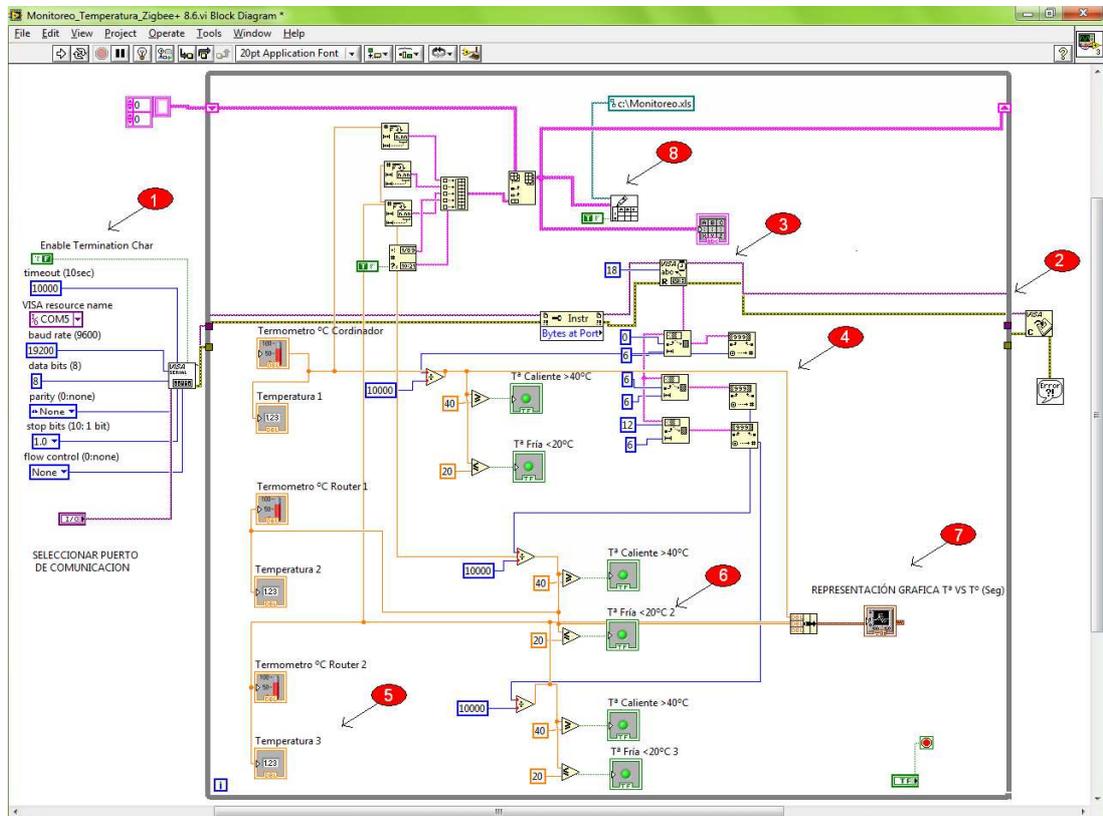


Figura 5.3 Monitoreo de temperatura en la red Zigbee (esquemático).



5.3.1 Explicación por bloques de los elementos que se trabajaron en el ambiente de desarrollo grafico Labview, para la realización del software.

1. En este bloque se pueden observar todos los parámetros del puerto de comunicación; se utiliza la consola VISA para programarlos y habilitar el puerto.
2. En este bloque se observa cuando se cierra la interfaz de comunicaciones VISA.
3. Con este elemento se pueden obtener los datos que se encuentran en el puerto serial, en este caso 18 bit.
4. En este bloque se realiza la separación de esos 18 bits capturados, con el elemento anterior.

El primer string subset inicia en la posición 0 y toma una longitud de 6 bit
 El segundo string subset inicia en la posición 6 y toma una longitud de 6 bit
 El tercer string subset inicia en la posición 12 y toma una longitud de 6 bit

Los **decimal string to Number**, como su nombre lo dice, pasa los valores de string a valores numéricos enteros.

5. En este bloque se realiza la creación de los termómetros gráficos y su valor numérico.
6. En este bloque se crean las alarmas graficas, cuando se sobrepasa o disminuyen los valores sensados a los valores especificados.
7. En este bloque se crea la grafica de todos los valores sensados.
8. Con este último bloque se realiza la base de datos la cual quedará guardada en la PC en formato .xls (Excel).

5.4 DESCRIPCIÓN DE LOS NODOS ZIGBEE

Para la realización de la pequeña red se utilizaron tres nodos, un coordinador y dos routers, los cuales recolectan la información de la temperatura de los sensores que poseen.

El lenguaje de programación en el cual fueron programados estos dispositivos fue el lenguaje C, utilizando el entorno de programación MPLAB IDE y con el compilador MPLAB C18 (otorgado por dos meses para uso sin licencia) los cuales se encuentran en la página oficial de Microchip para ser descargados.

A continuación se explica el funcionamiento de los dispositivos utilizados en la red.

5.4.1 Coordinador

Como ya se ha mencionado anteriormente el coordinador es el dispositivo que forma la red y permite la unión de otros dispositivos como routers y dispositivos finales. Además de esto, el coordinador fue diseñado para que solicitara a un grupo de dispositivos los datos de los sensores de temperatura, después de solicitar estos datos, el coordinador lee la temperatura de su sensor y lo agrega a una variable con los demás datos solicitados, éstos son enviados a través del puerto serial del coordinador hacia el computador para su posterior análisis.

El diagrama de flujo simplificado del coordinador se puede ver en la Figura 5.4.

5.4.2 Router

El router está diseñado para luego de unirse a la red y agregarse a un grupo de dispositivos, éste esperará la solicitud de adquisición de la temperatura por parte del coordinador, una vez el router la recibe, procede a capturar la información del sensor y enviarla hacia el coordinador para su correspondiente análisis.

El diagrama de flujo simplificado de los routers se puede ver en las Figura 5.5 y 5.6

Figura 5.4 Diagrama de flujo simplificado del coordinador.

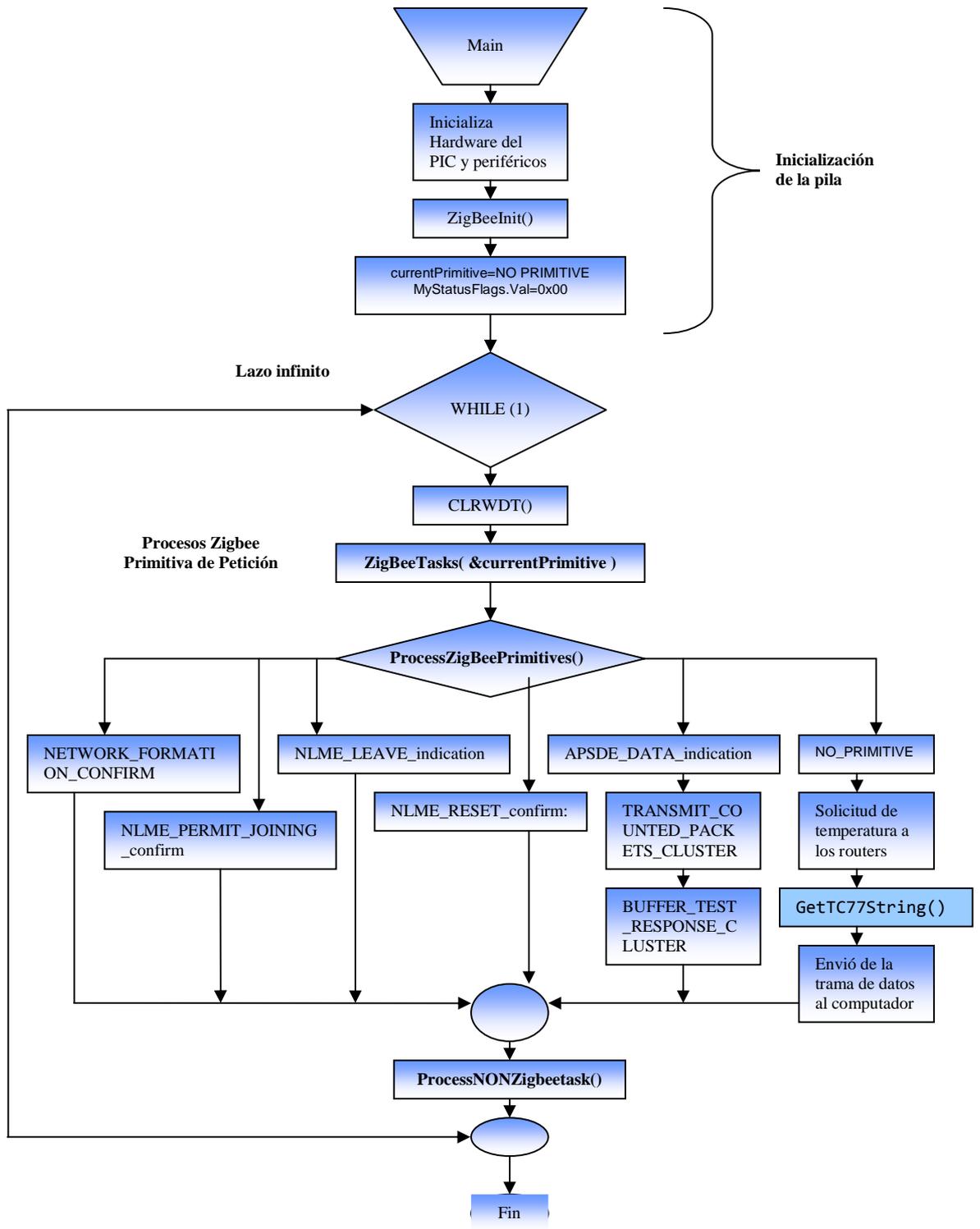


Figura 5.5 Diagrama de flujo simplificado del Primer Router.

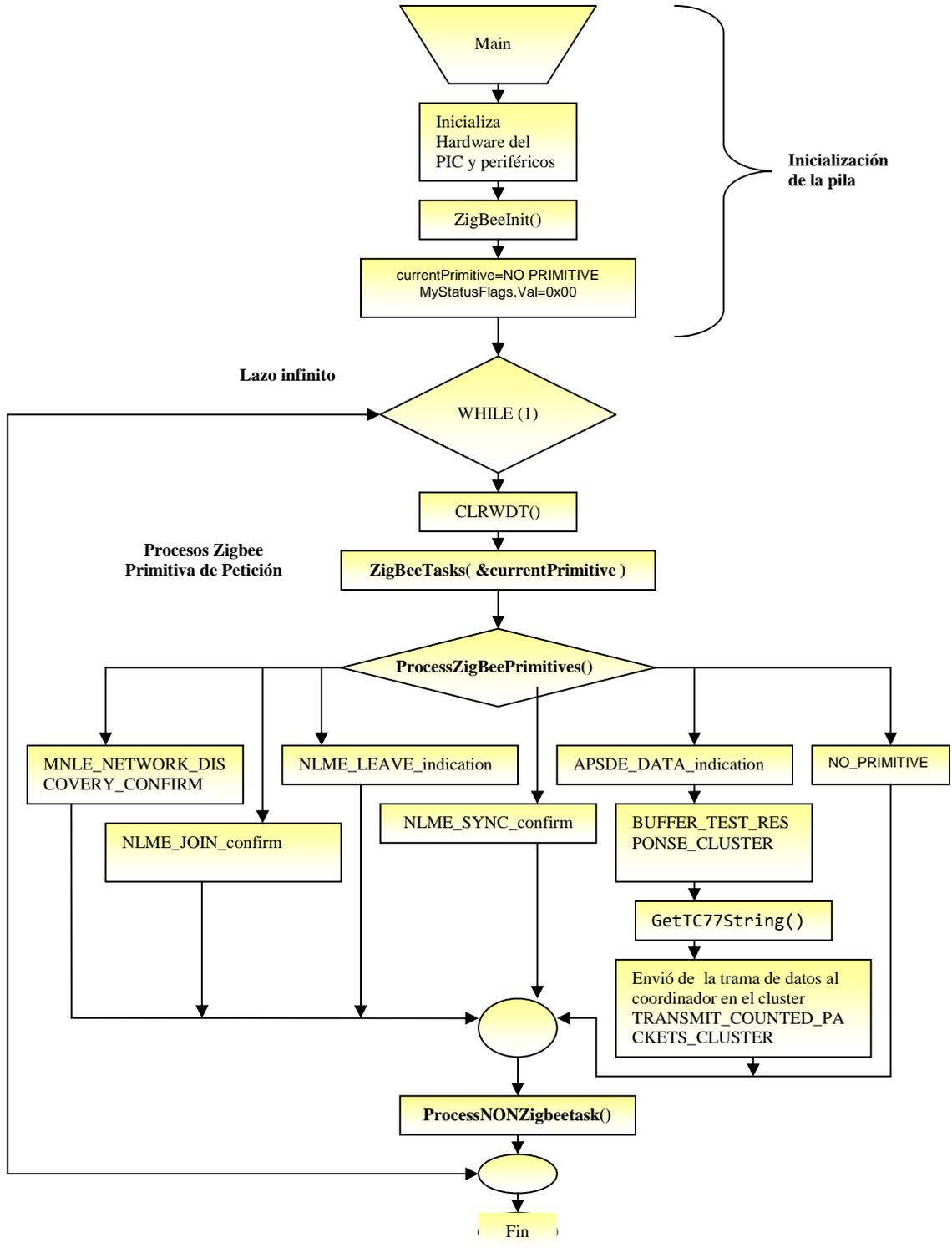
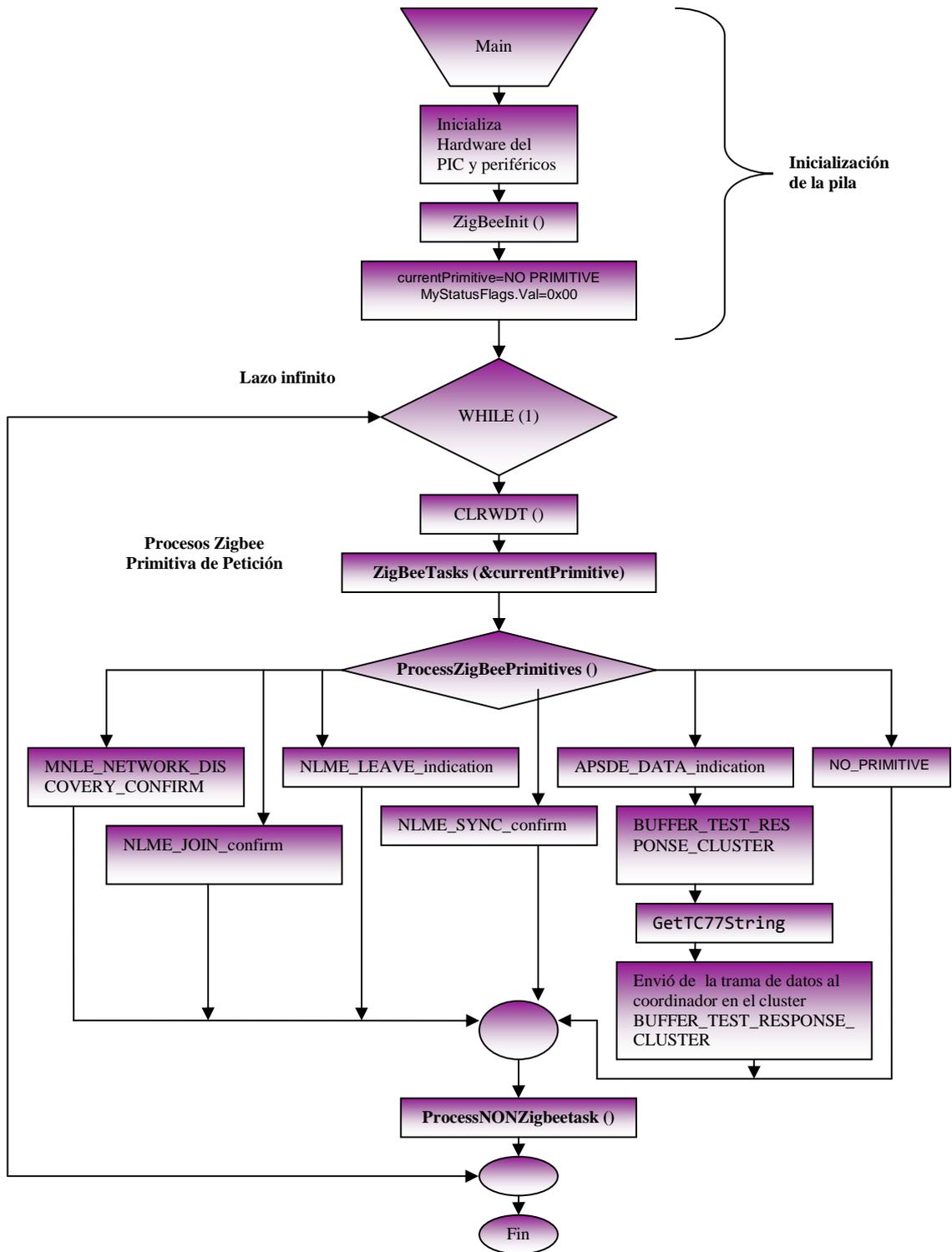


Figura 5.6 Diagrama de flujo simplificado del Segundo Router.



5.5 Código de adquisición de temperatura

Para la adquisición de la temperatura del sensor TC77 se implementó el siguiente código, el cual se explicará con detenimiento.

```
1.  /*****
2.  * Funcion:          BYTE GetTC77String( char *buffer )
3.  *
4.  * Precondiciones:  Ninguna
5.  *
6.  * Input:           buffer - puntero donde copilar el string.
7.  *
8.  * Output:          Retorna el tamaño de el string.
9.  *
10. * Efectos secundarios: Ninguno
11. *
12. * Información general: Esta función lee el sensor TC77 y entrega un valor
13. *                       de temperatura al puntero *buffer. Si se recibe 0x00
14. *                       ó 0xFF, indicara que tenemos un problema con el sensor.
15. *
16. *
17. *
18. *****/
19. BYTE GetTC77String( char *buffer )
20. {
21.     typedef union _SIGNED_WORD_VAL
22.     {
23.         unsigned int uVal;
24.         int Val;
25.         struct
26.         {
27.             BYTE LSB;
28.             BYTE MSB;
29.         } byte;
30.     } SIGNED_WORD_VAL;
31.
32.     BYTE        GIEHsave;
33.     char         *ptr;
34.     char         *pTempString;
35.     SIGNED_WORD_VAL RawTemp;
36.     signed long  ScaledTemp;
37.     BYTE         strLen;
38.     char         tempString[10];
39.
40.     // Leer la temperature de TC77
41.     //
42.     // El sensor TC77 entrega una línea 13 bits son signo
43.     // donde 0 = 0 C°, 1 = 0.0625 C°,
44.     // 2 = 0.125 C°, -1 = -0.0625 C°,
45.     // -2 = -0.125 C°, 400 = 25 C°, etc.
46.     // Para convertir a un valor que se puede trabajar.
47.     // Para poderlo visualizar se puede convertir este valor
48.     // a coma flotante y multiplicarlo por 0.0625 C°
49.     //
50.     // Otra alternativa es dejarlo como entero y multiplicarlo
51.     // por 625. El resultado será escalonado por un
52.     // un factor de 10,000.
53.     // Deshabilitados las interrupciones mientras se habla con el sensor TC77.
54.     GIEHsave = GIEH;
55.     GIEH = 0;
```

```

56.
57. // Leer dato de el TC77
58. TC77_CS = 0; // Seleccionamos el TC77
59. RawTemp.byte.MSB = SPIGet(); // Obtenemos los primeros 8 bits
60. RawTemp.byte.LSB = SPIGet(); // Obtenemos los 5 bits restantes
61. TC77_CS = 1; // deshabilitamos el TC77
62. //habilitamos las interrupciones.
63. GIEH = GIEHsave;
64. if ((RawTemp.uVal == 0x0000) || (RawTemp.uVal == 0xFFFF))
65. {
66.     strcpypgm2ram( buffer, (rom char *)"Problem reading TC77." );
67. }
68. else
69. {
70.     // Right align the 13 bit temperature reading
71.     RawTemp.uVal >>= 3;
72.     // Extender el valor de 13 bits a 16 bits
73.     if(((BYTE_VAL*)&RawTemp.byte.MSB)->bits.b4)
74.     {
75.         RawTemp.byte.MSB |= 0xE0;
76.     }
77.     // convertir el valor entero del TCC77 a escalonado con 0.0001 C°.
78.     ScaledTemp = (long)RawTemp.Val * (long)625;
79.     // Convertir el valor de entero a string y encontrar su tamaño
80.     pTempString = ltoa( ScaledTemp, tempString );
81.     strLen = strlen( tempString );
82.     // Escribimos el signo negativo si es necesario.
83.     ptr = buffer;
84.     if(tempString[0] == '-')
85.     {
86.         *ptr++ = '-';
87.         strLen--;
88.         pTempString++;
89.     }
90.     // escribir "0."y luego el numero si es cercano a cero, Ej.: 0, 0.0625
91.     if(strLen < 5)
92.     {
93.         *ptr++ = '0';
94.         *ptr++ = '.';
95.         memcpy( (void *)ptr, (void *)tempString, strLen );
96.         *ptr += strLen;
97.     }
98.     // de lo contrario, escribir el numero completo
99.     // y poner el punto decimal en la quinta posición;
100.    // Ej.: 24.1250
101.    else
102.    {
103.        for(;strLen >= 5; strLen--)
104.            *ptr++ = *pTempString++;
105.        *ptr++ = '.';
106.        for(;strLen > 0; strLen--)
107.            *ptr++ = *pTempString++;
108.    }
109.
110.    // escribir °C"
111.    *ptr++ = ' ';
112.    *ptr++ = 'C';
113.    *ptr = '\0';
114. }
115.
116. return strlen( buffer );
117. }

```

CAPÍTULO 6

CONCLUSIONES

En este proyecto se realizó una validación de la tecnología Zigbee de Microchip, además se diseñó e implementó un sistema de adquisición y control de temperatura a través de una pequeña red ZigBee (coordinador y dos routers), logrando una red de topología en malla.

El protocolo de la tecnología ZigBee es más simple que muchos protocolos inalámbricos, pero todavía son necesarias herramientas para desarrollar aplicaciones de manera eficiente y eficaz. Para confirmar que los valores de los datos están siendo enviados correctamente y comprender el comportamiento de la red, un analizador de red inalámbrica es una herramienta esencial para cualquiera diseñador de redes inalámbricas.

Se comprobó que la pila de protocolos de la tecnología Zigbee de Microchip soporta los dispositivos descritos por el estándar IEEE 802.15.4 y Zigbee Alliance como los son FFD y RFD; coordinadores, routers y dispositivos finales.

En la captación de la temperatura el sistema presenta una alta confiabilidad pero a la hora de la comunicación se deben realizar varios intentos debido a que la tecnología Zigbee al tener una tasa de transmisión de datos baja y una baja calidad de servicio se hace un poco difícil la adquisición, además se deben orientar las antenas de los dispositivos correctamente para tener un mejor desempeño del sistema.

Los dispositivos RFD tienen la capacidad de alternar en dos estados, en modo pasivo (idle) y en modo desactivado (sleep) haciendo que el consumo de energía sea poco; lo cual diferencia este protocolo con otros protocolos de redes inalámbricas que no poseen estas características.

Los resultados obtenidos demuestran que los nodos son capaces de enviar paquetes de información a una distancia considerable, tanto en exteriores (100m), como en interiores (30m), además se demostró que la potencia de emisión decrece de forma no lineal y a su vez la potencia mínima para que la información aún fuera útil fue de -92dbm.

Se comprobó la utilidad de las herramientas de análisis y monitoreo como lo es el analizador de redes inalámbricas ZENA para verificar la correcta conformación de los mensajes intercambiados por los nodos de una red y depurar con ello los códigos de las aplicaciones en el microcontrolador y el ordenador.

Se debe tener en cuenta que Colombia al no ser productor de dispositivos electrónicos y el escaso mercado del diseño provoca que los precios a nivel interno sean extremadamente altos. A esto se suman los impuestos, los aranceles y el transporte que agregan cerca del 30% del total del costo, lo cual desmotiva el diseño de soluciones para el mercado masivo y limitan a crear aplicaciones personalizadas para la industria.

CAPITULO 7

RECOMENDACIONES

Se recomienda perfeccionar y ampliar la red mediante enrutadores que mejoren el alcance del sistema y permitan incrementar el número de posibles nodos dentro de la red. Con ello, se pueden realizar pruebas de tráfico en un sistema más complejo.

A pesar de que el sensor de temperatura TC77 tiene un rango de medición desde -55°C hasta 125°C , se recomienda no exponer a los módulos a temperaturas bajo 0°C o sobre 70°C debido a que es un kit de demostración y el resto de sus componentes no soportan el mismo rango de temperatura del TC77.

Se recomienda colocar el analizador de red inalámbrica ZENA a una altura predeterminada por el usuario para la buena adquisición de los paquetes de la red observada.

Se recomienda para trabajos futuros, la utilización de la tecnología MIWI la cual fue diseñada por la empresa Microchip basándose en el estándar IEEE 802.15.4. Esta tecnología es muy similar a la tecnología Zigbee pero con la diferencia de manejar menos aspectos de seguridad en las capas MAC y APS, enfocándose más para el manejo de redes pequeñas PAN y redes de poca transmisión de datos.

Para el aprendizaje de la pila de protocolos de la tecnología Zigbee de Microchip, se recomienda la adquisición del kit de desarrollo PICDEM Z el cual cuenta con dos tarjetas madres PICDEM Z , dos trancivers y el analizador de redes inalámbricas ZENA para el monitoreo de paquetes. Siendo ésta una buena opción para la implementación de proyectos educativos basándose en la tecnología Zigbee.

BIBLIOGRAFÍA

Comunicación Serial Utilizando LabVIEW con un Microcontrolador
<http://zone.ni.com/devzone/cda/tut/p/id/7907>.

Getting Started: Running the ZigBee -2006 Demo
<http://www.microchip.com>

IEEE 802.15.4™ Specification
<http://www.ieee.org>

License Stack Software for ZigBee
<http://www.microchip.com>

Manejo de la sesión VISA de LabVIEW
<http://zone.ni.com/devzone/cda/tut/p/id/2897>

Microchip ZigBee-2006 Residential Stack Protocol (AN1232)
<http://www.microchip.com>

MRF24J40MA Datasheet (DS70329b)
<http://www.microchip.com>

PIC18F4620 Datasheet” (DS39626e)
<http://www.microchip.com>

PICDEM™ Z Demonstration Kit User’s Guide (DS51524)
<http://www.microchip.com>

TC77 TemperatureSensor Datasheet (DS 20092a)
<http://www.microchip.com>

ZENA™ Wireless Network Analyzer User’s Guide (DS51506)
<http://www.microchip.com>

ZIGBEE ALLIANCE
<http://www.zigbee.org/>

ZigBee™ Protocol Specification
<http://www.zigbee.org>

ZigBee Security” Robert Cragie Chair, ZigBee Alliance ZARC Security Task Group
Principal Engineer, Jennic Ltd.
<http://www.zigbee.org/>

ANEXO A

MANUAL DE USUARIO

INTRODUCCIÓN

El protocolo ZigBee es un protocolo de red inalámbrica diseñado específicamente para sensores de baja velocidad de datos y redes de control. Hay una serie de aplicaciones que pueden beneficiarse del protocolo ZigBee como lo son las siguientes: las redes de automatización de edificios, sistemas de seguridad, redes de control industrial, medición remota y periféricos de PC.

Cuando se compara contra otros protocolos inalámbricos, el protocolo ZigBee ofrece baja complejidad, la reducción de las necesidades de recursos y sobre todo, un conjunto estándar de especificaciones. También ofrece tres bandas de frecuencia de operación junto con un número de configuraciones de red y la capacidad de seguridad opcional.

La pila de protocolos de la tecnología ZigBee de Microchip versión v2.0-2.6.xx proporciona una conectividad inalámbrica basada en las normas establecidas por la ZigBee Alliance, esta tecnología utiliza el estándar IEEE 802.15.4 para las capas física y MAC, las capas restantes son definidas por la ZigBee Alliance.

La personalización de la capa de aplicación permite a los usuarios desarrollar una amplia gama de aplicaciones basadas en la pila de protocolos de la tecnología ZigBee, la aplicación que se incluye como parte de la versión v2.0-2.6.xx de la pila, muestra varias características de conectividad de red.

Estas características incluyen:

- La capacidad de activar o desactivar los dispositivos que pueden unirse a otros dispositivos que están en una red. Esta función es útil para ayudar a la configuración de una topología de red específica la primera vez que se forma una red.
- La capacidad para enviar o solicitar los datos de un dispositivo específico ya sea por **direccionamiento unicast** o por **direccionamiento broadcast**.
- **Multicasting.** La capacidad para enviar o solicitar datos a otros dispositivos que son miembros de un grupo específico. Un **grupo ID** se utiliza para referenciar los dispositivos que forman parte de un grupo determinado.
- La capacidad para agregar o quitar un dispositivo de un grupo específico.
- La capacidad para mostrar la información de los vecinos para cada dispositivo.

Las siguientes funciones están disponibles a través de dos botones en la tarjeta madre PICDEM Z

- Multicasting - El envío o solicitud de datos a los dispositivos que sean miembros del **Grupo 0x0004**.
- Agregar o quitar un dispositivo que pertenezca al **Grupo 0x0004**.

El grupo 0x0004 está predeterminado en el código de la aplicación para cuando se presione el botón se agregue a este grupo, esto se hace con el fin de simplificar el código e ilustrar la función de multidifusión (multicasting) de la forma más sencilla posible. Sin embargo, desde el menú del hyperterminal, el usuario tiene la opción de elegir el **grupo** dentro del **rango de 0x0001 a 0x00f0**.

Cada nodo puede incluirse como miembro de un grupo mediante la creación y el mantenimiento de una tabla de grupo. La tabla del grupo contiene el ID del grupo y el end point asociado del grupo al cual es miembro el nodo.

REQUERIMIENTOS DE HARDWARE

Con el fin de crear una red simple que conste de dos nodos, son necesarias dos plataformas de hardware las cuales contengan los siguientes dispositivos.

Plataforma de hardware (plataforma de 8-bits):

- PICDEM Z 2,4 Kit Demo (DM163027 DM163027-4 O-5)
- MRF24J40MA PICDEM Z de 2.4GHz Tarjeta RF (AC163028)

Otros Hardwares

- Cable RS-232 o adaptador de RS232 a USB
- PC con puerto RS-232 COM o USB

Software

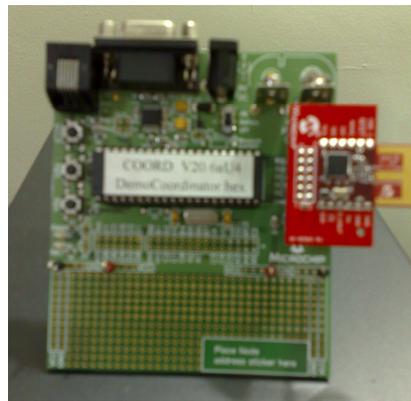
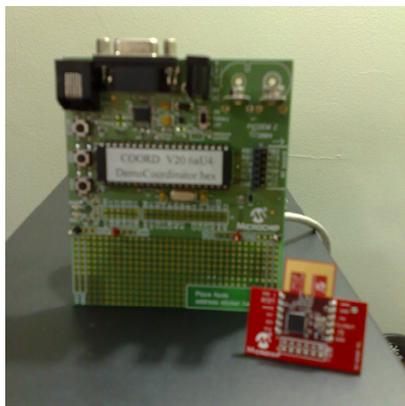
- MPLAB C18 compilador v3.20 o superior para la Plataforma PIC18F PICDEM Z
- Un programador como MPLAB REAL ICE o ICD 2 o que pueda quemar pic de la familia 18f.

CONFIGURACION DEL HARDWARE

En esta sección se describe cómo establecer la configuración de hardware para ejecutar esta demostración.

Configuración PICDEM Z

Conectar el MRF24J40 2.4GHz RF Card a la tarjeta de demostración PICDEM Z como se muestra a continuación.



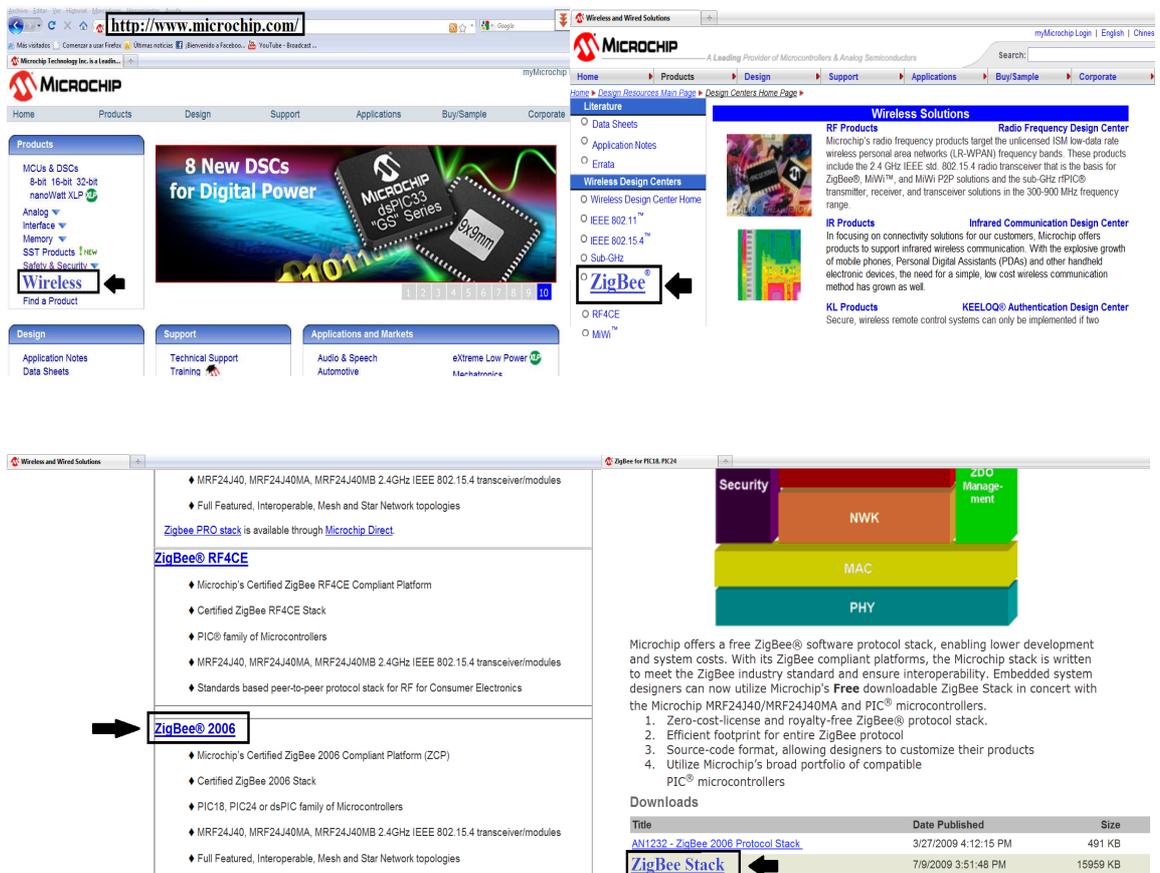
FIRMWARE

Todo el código fuente de la Pila ZigBee2006 se puede encontrar en la página oficial de Microchip. www.microchip.com

Las imágenes de abajo muestran la secuencia de cómo encontrar los códigos de la pila de protocolo para los dispositivos zigbee.

Los archivos ejecutables (.hex) de los tipos de dispositivos de ZigBee (Coordinador, Router y RFD) se consiguen en la página principal.

Cada archivo del proyecto viene separado para cada tipo de dispositivo de forma que **cada dispositivo se puede construir y modificar** para requisitos particulares o para operaciones específicas.



Fuente: www.microchip.com

En esta aplicación cuando usted:

- **Presiona el botón RB4/RD6** - se utiliza para agregar o quitar el nodo del grupo 0x0004 (el cual está predeterminado en el demo).
- **Presiona el botón RB5/RD7**- se utiliza para enviar un mensaje de difusión del grupo a cualquier dispositivo que sea un miembro del grupo 0x0004. El mensaje difundido solicita a cada uno de los dispositivos en el grupo devolver diez bytes de datos al solicitante. Los datos que se devuelven al solicitante serán del siguiente patrón de byte [00 01 02 03 04 05 06 07 08 09 0A]
- **RA0/D10 LED** - se enciende cuando el dispositivo es un miembro del Grupo 4 y se apaga cuando no lo es.
- **RA1/D09 LED** - se utiliza como un indicador. Indica que el nodo recibió y respondió a los mensajes dirigidos a él, cuando se trata de un miembro del Grupo de

0x0004. Por lo tanto, cada vez que un nodo sea un miembro del Grupo 0x0004 y reciba una solicitud de otro dispositivo para enviar sus datos; este led se activará después de que se hayan enviado los datos. Esta es una manera útil y sencilla para asegurarse visualmente de que el enlace de comunicación entre los dispositivos está trabajando.

Cuando el nodo está conectado a una consola de Hyper terminal, el contenido de los mensajes recibidos se muestran cada vez que el nodo sea un miembro del Grupo de 0x0004.

PROCESO PARA EL QUEMADO DEL PIC, CON LOS .HEX DESCARGADOS EN LA PÁGINA PRINCIPAL.

- Después de descargar los archivos .hex de la página principal, como se mostró en las graficas anteriores, se dispondrá a quemarlos en cada pic de cada tarjeta de nodo, dependiendo del dispositivo que se necesite (**coordinador, router, dispositivo final**).

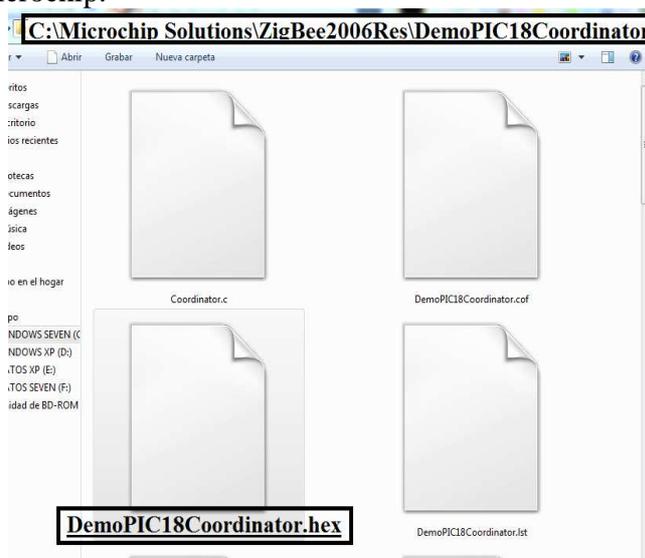
C: MicrochipSolution \ZigBee2006Res \DemoPIC18FCoordinator\ DemoPIC18FCoordinator.hex

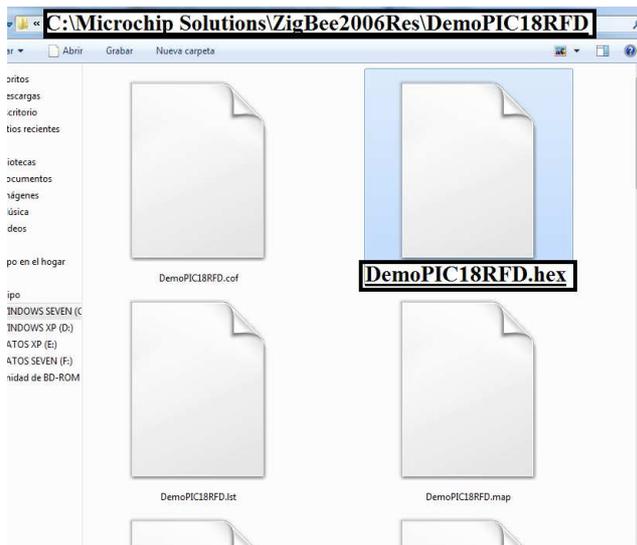
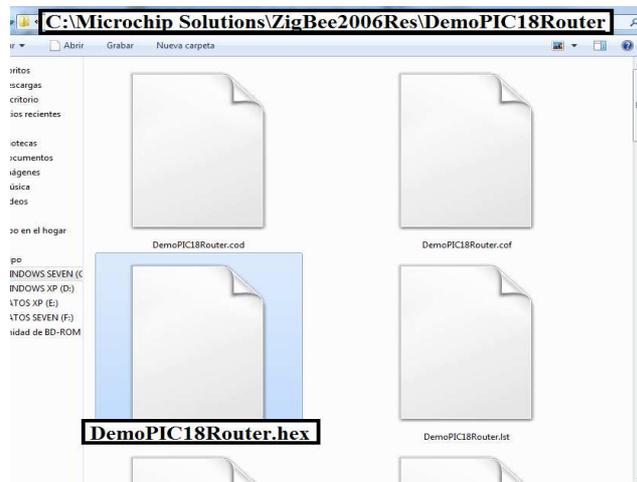
C: MicrochipSolution \ ZigBee2006Res \ DemoPIC18FRouter \DemoPIC18FRouter.hex

C: MicrochipSolution \ ZigBee2006Res \ DemoPIC18FRFD\ DemoPIC18FRFD.hex

Con estos archivos .hex se puede quemar directamente al Pic 18f4620, ya que está compilado específicamente para este microcontrolador.

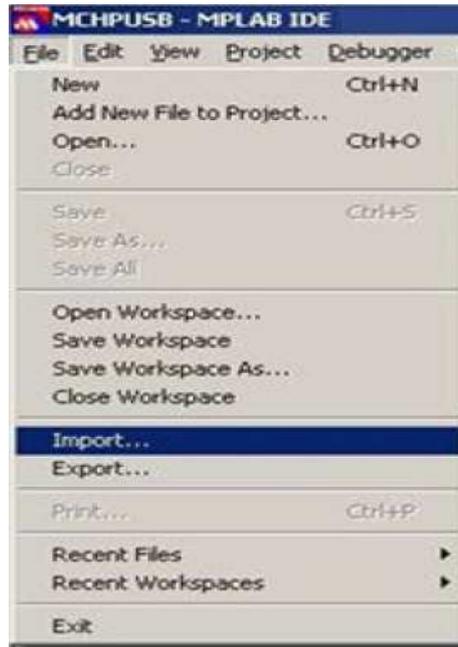
- Las siguientes imágenes muestran la ubicación de los archivos que han sido instalados después de haber descargado el paquete Zigbee2006Res de la página oficial de microchip.





- Para poder trabajar y modificar la pila de protocolos hecha por Microchip se realizan los siguientes pasos:

Utilice el menú Archivo → Importar desde MPLAB, para importar el archivo .hex. A continuación utilizar el programador MPLAB para programar las aplicaciones de demostración, iniciando con en el Coordinador y después ya sea bien con en el router o con el dispositivo RFD.



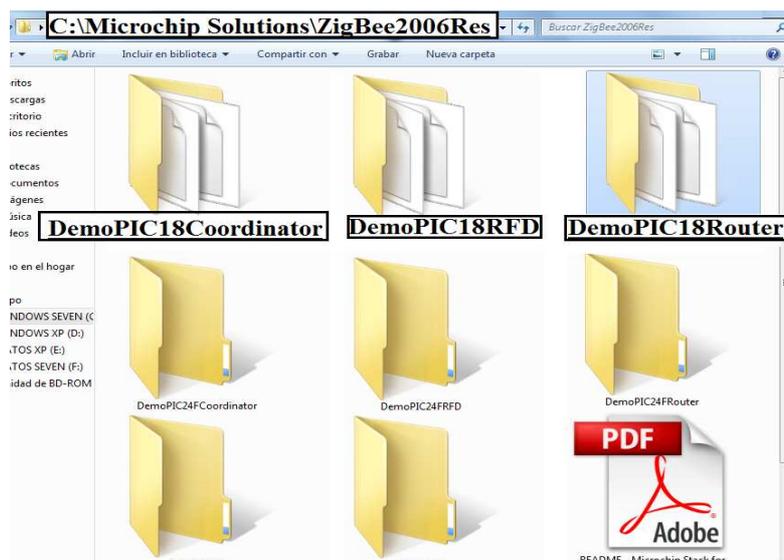
Para facilidad de la guía se realizará la comunicación entre el Coordinador y el Router.

El código fuente de estas aplicaciones demo está disponible en las siguientes carpetas:

C: MicrochipSolution\ ZigBee2006Res \ DemoPIC18FCoordinator

C: MicrochipSolution \ ZigBee2006Res \ DemoPIC18FRouter

C: MicrochipSolution \ ZigBee2006Res \ DemoPIC18FRFD



En estos directorios, usted encontrará todos los archivos fuente a nivel de aplicación y los archivos de cabecera (header), así como el proyecto y los archivos de espacio de trabajo (workspace) para cada una de las plataformas de hardware.

Haga doble clic en el archivo fuente .mcw en las carpetas correspondientes para poner en marcha MPLAB® IDE.

Una vez concluya con los cambios necesarios al programa y después de seleccionar el programador adecuado que se va a utilizar, seleccione proyecto → build-all opción de compilar.

EJECUTAR LA APLICACIÓN UTILIZANDO LOS BOTONES

Usted necesitará:

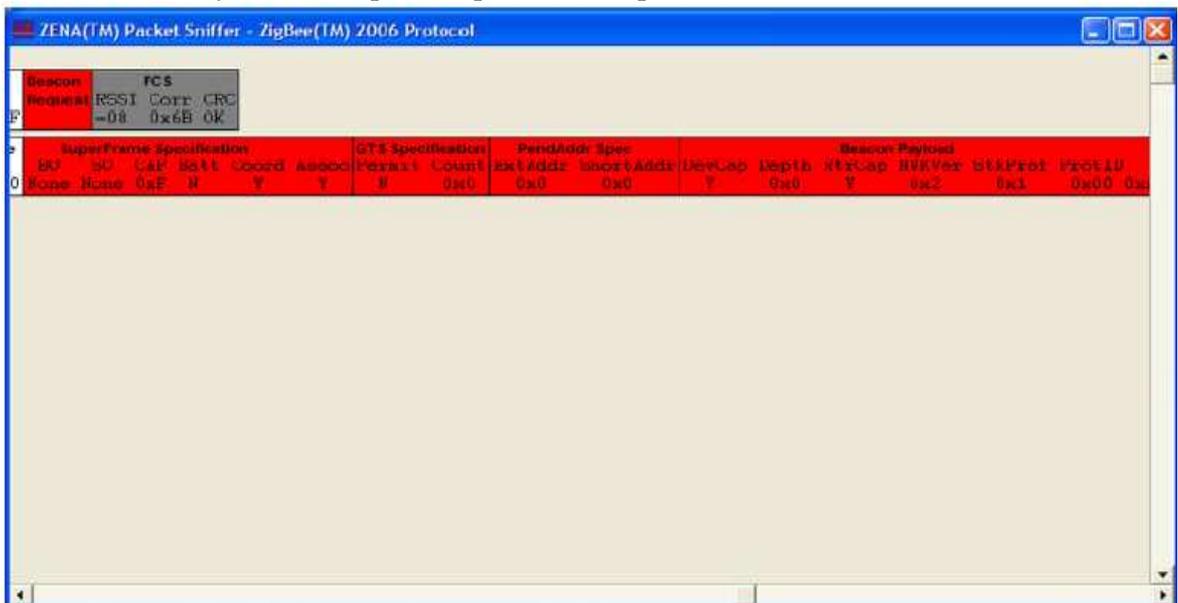
Un par de PICDEM Z, que se cargan con la aplicación de demostración correcta.

- Uno de los dispositivos debe ser un coordinador.
- El otro dispositivo puede ser un router o un RFD.

1. Conecte el analizador de red ZENA en el puerto USB del computador y configúrelo para controlar el tráfico de red en el canal 26.

2. Aplique alimentación al Coordinador con un adaptador de 9V o una de pila de 9V. Pulse y suelte el botón de etiqueta MCLR.

Los dos LEDs etiquetados RA0/D10 y RA1/D09 se encienden, indicando que el dispositivo ha formado una red. Observe el paquete de solicitud de Beacon Zena. El Coordinador ya está listo para aceptar otros dispositivos en su red.



3. Aplique energía al dispositivo Router con un adaptador de pared 9V o una batería de 9V. Pulse y suelte el botón etiquetado MCLR. Dos indicadores LED identificados RA0/D10 y RA1/D09 se encienden, indicando que el dispositivo se ha unido a la red.

Observe en el Zena el paquete de Solicitud de la **asociación** del router y la respuesta de **asociación** exitosa de paquetes del Coordinador.

El router se ha incorporado correctamente a la red.

ZENA(TM) Packet Sniffer - ZigBee(TM) 2006 Protocol																	
MAC Frame Control			Seq Num	Source PAN	Source Addr	SuperFrame Specification				GTS Specification		PendAddr Spec					
Type	Sec	Pend	ACK	IPAN		BO	SO	CAP	Start	Coord	Assoc	Permit	Count	ExtAddr	ShortAddr	Dev	
BCN	N	N	N	N	0x10	0x1AAA	0x0000	None	None	0xF	N	7	Y	N	0x0	0x0	Y
MAC Frame Control			Seq Num	Source PAN	Source Addr	SuperFrame Specification				GTS Specification		PendAddr Spec					
Type	Sec	Pend	ACK	IPAN		BO	SO	CAP	Start	Coord	Assoc	Permit	Count	ExtAddr	ShortAddr	Dev	
BCN	N	N	N	N	0x99	0x276C	0x0000	None	None	0xF	N	7	Y	N	0x0	0x0	Y
MAC Frame Control			Seq Num	Dest PAN	Dest Addr	Source PAN	Source Address	Association Request									
Type	Sec	Pend	ACK	IPAN				Align	Sec	ExtAddr	Frame	Dev	AltCoord	RSSI			
CMD	N	N	Y	N	0xC7	0x276C	0x0000	0xFFFF	0x00000000	2000000000	Y	Y	0x	None	FFU	N	-06
MAC Frame Control			Seq Num	FCS													
Type	Sec	Pend	ACK	IPAN		RSSI	Corr	CRC									
ACK	N	N	N	N	0xC7	-10	0x6B	OK									
MAC Frame Control			Seq Num	Dest PAN	Dest Addr	Source PAN	Source Address	Data Request	FCS								
Type	Sec	Pend	ACK	IPAN					RSSI	Corr	CRC						
CMD	N	N	Y	Y	0xC8	0x276C	0x0000	0x00000000	2000000000	-04	0x6B	OK					
MAC Frame Control			Seq Num	FCS													
Type	Sec	Pend	ACK	IPAN		RSSI	Corr	CRC									
ACK	N	Y	N	N	0xC8	-09	0x6B	OK									
MAC Frame Control			Seq Num	Dest PAN	Destination Address	Source PAN	Source Address	Association Response		FCS							
Type	Sec	Pend	ACK	IPAN				Status	Address	RSSI	Corr	CRC					
CMD	N	N	Y	Y	0x98	0x276C	0x00000000	2000000000	0xAAAAAAAAAAAAAAAA	Success	0x0001	-00	0x6B	OK			
MAC Frame Control			Seq Num	FCS													
Type	Sec	Pend	ACK	IPAN		RSSI	Corr	CRC									
ACK	N	N	N	N	0x98	-0A	0x6B	OK									

Ambos dispositivos están funcionando pero todavía ninguno es miembro del grupo 0x0004. Cuando un router recibe un mensaje de difusión del grupo, en primer lugar, se retransmite este mensaje para que otros nodos de la red puedan recibir el mensaje de difusión. En segundo lugar, también se comprueba si uno de sus extremos se encuentra en el mismo grupo y si es así, responde a la solicitud de difusión.

Como ninguno de los nodos están en ningún grupo específico en el reset, si el Coordinador envía Mensajes al Grupo 0x0004 el router sólo retransmite estos mensajes, pero no responde con el pedido de 10 bytes que el Coordinador hace.

4. En el Coordinador, presione y suelte el botón RB5/RD7. Observe el tráfico en el analizador Zena. Los mensajes serán transmitidos con dirección destino 0xffff; Dirección de origen 0x0000 (del Coordinador) y una etiqueta de los mensajes como mensajes de información de grupo dirigida. El router retransmite los mensajes pero no responde al Coordinador, ya que todavía no es miembro del Grupo de 0x0004.

Ahora agregue el router al Grupo 0x0004 haciendo lo siguiente:

5. Pulse y suelte el botón **RB4/RD6** en el dispositivo router y observe el **LED RA0/D10** parpadear, RB4/RD6 se presiona y se suelta.

Pulse y suelte el pulsador RB4/RD6 tal que RA0/D10 permanezca encendido después de soltar el botón. Cuando RA0/D10 está encendido, esto indica que el router se ha añadido como miembro del Grupo de 0x0004. **No hay tráfico asociado en el analizador ZENA ya que ésta es una operación interna del dispositivo y no se transmite ningún paquete por el aire.**

6. Pulse y suelte RB5/RD7 en el coordinador. Esto enviará un grupo de mensaje de difusión a todos los nodos que están en el Grupo 0x0004.

Observe que el mensaje se transmite a la dirección de destino 0xFFFF, Dirección Grupo 0x0004 en el analizador Zena.

Observe que el LED RA1/D09 en el router activa y desactiva cada vez que RB5/RD7 en el Coordinador se presiona.

Esto indica que el router está ahora respondiendo a los mensajes enviados por el Coordinador de Grupo 0x0004 porque ahora es un miembro de ese grupo.

ZENA(TM) Packet Sniffer - ZigBee(TM) 2006 Protocol																								
1	CMD	0x2	SUP	N	0xFFFF	0x0001	0x0A	0x03	Route Request				FCS											
0	CMD	0x2	SUP	N	0x0001	0x0000	0x0A	0x25	Route Reply				FCS											
00	DAT	0x2	SUP	N	0xFFFF	0x0000	0x0A	0x26	APS Frame Control				Group		Cluster		Profile		Source		AF Data		FCS	
01	DAT	0x2	EN	N	0x0000	0x0001	0x0A	0x04	APS Frame Control				Dest		Cluster		Profile		Source		AF Data		FCS	
01	DAT	0x2	SUP	N	0xFFFF	0x0000	0x09	0x26	APS Frame Control				Group		Cluster		Profile		Source		AF Data		FCS	

7. Pulse y suelte el botón RB4/RD6 en el router una vez más hasta que el led RA0/D10 esté apagado. Esto indica que el router ya no está en el Grupo 0x0004.

Mensajes En relación con el Grupo 4 del Coordinador No serán respondidos.

8. El Usuario puede añadir el Coordinador al Grupo 0x0004, repitiendo los pasos anteriormente descritos y enviar mensajes a el Coordinador desde el router presionando RB5/RD7.

La aplicación funciona de forma bidireccional, es decir, de manera exactamente igual en todos los tipos de dispositivos de la red ZigBee (Coordinadores, Routers y RFD), se puede enviar una solicitud desde el coordinador a un router o desde un router a un coordinador.

Resumen de la Operación Push Button:

- **Botón RB4/RD6:** Agrega un nodo o elimina un nodo del Grupo 0x0004. El Grupo ID 0x0004 está previamente configurado en el código de la aplicación.
- **Led RA0/D10:** Se utiliza para indicar si un nodo es un miembro del Grupo de 0x0004 (ON - sí; OFF - no)

Al inicio los dispositivos no están en ningún grupo, y los LEDs están encendidos para indicar que el dispositivo están en la red.

- **Botón: RB5/RD7:** se utiliza para enviar mensajes a un grupo de nodos en 0x0004. El mensaje real es una solicitud para todos los nodos del Grupo 0x0004 para enviar al solicitante 10 bytes.
- **Led: RA1/D09:** se enciende y apaga para indicar que el nodo receptor está en el Grupo 0x0004 y recibe solicitud de envío de datos.

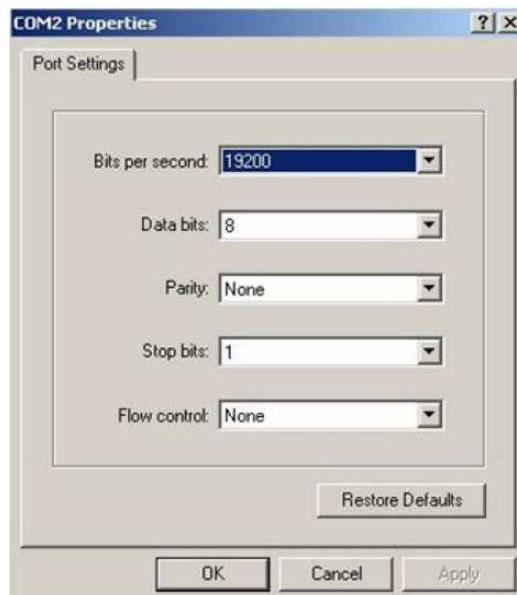
EJECUTAR LAS APLICACIONES A TRAVÉS DE LA CONSOLA HYPER TERMINAL

Para utilizar la aplicación desde el menú de sistema hyper terminal se realiza lo siguiente:

1. Siga los pasos del 1 al 3 de la sección de funcionamiento a través de pulsador para obtener la carga de dispositivos con el firmware y union a una red.
2. Conecte un cable RS-232 desde el dispositivo Coordinador al PC.
3. Abra una ventana de Hyper Terminal Server en el PC con el puerto COM al cual el cable RS-232 está conectado.

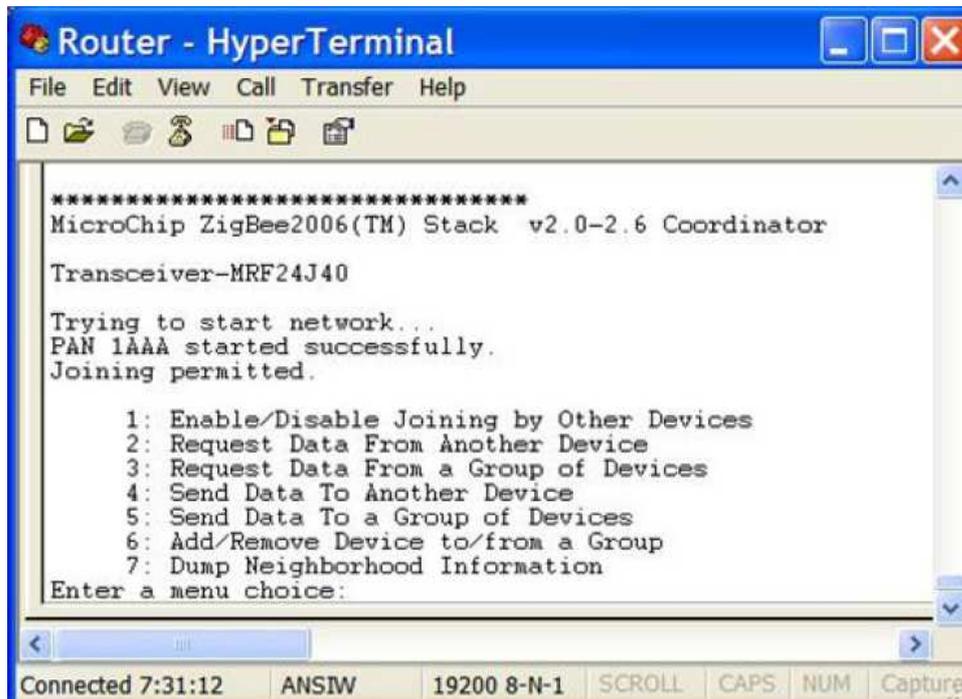
La configuración de Hyper-terminal debe ser:

- Baud Rate: 19200
- Bits de datos: 8-bits
- Paridad: Ninguna
- Bits de parada 1
- Control de flujo: Ninguno



4. Oprima el botón OK y coloque aceptar. Luego saldrá el siguiente menú que se visualiza en la consola del PC.

Tenga en cuenta las opciones del menú del 1 al 7 que están disponibles.



5. Pulse el botón <MCLR> en el dispositivo router. Esto forzará al Router a unirse a la red. Se puede observar en la consola del Coordinador los 16-bits de dirección de red que se le asignó al router y los 64-bits de la MAC del mismo. En una red típica de dos dispositivos, la dirección de red del router es 0x0001.

6. Para solicitar los datos del router a través del menú de Coordinador, haga lo siguiente:
Desde el menú de opciones seleccione <2>:

2: Solicitud de datos desde otro dispositivo

¿Cuántos bytes desea solicitar (hex)? :

0a

¿Cuál es la dirección corta del dispositivo del cual desea los datos? :

0001

Esto enviará una solicitud de datos al router a través de un unicast - directamente y sólo a el router pidiendo 0x0a bytes de datos que deben declararse.

La respuesta del router es:

Len:

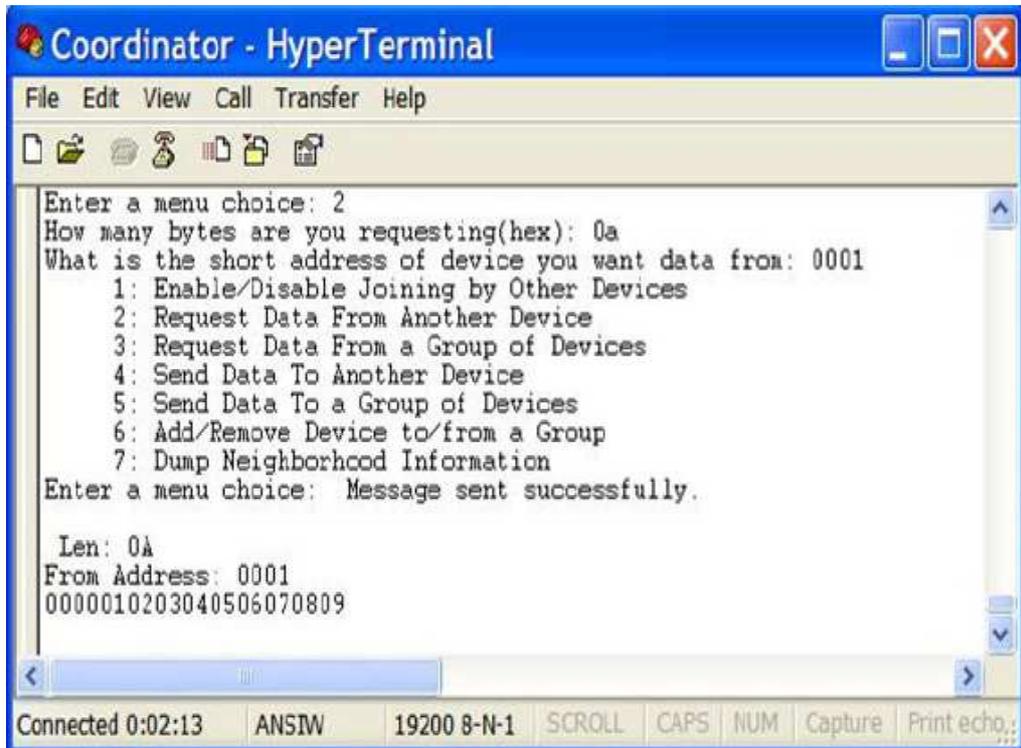
0a

De la Dirección:

0001

00000102030405060708090

Es decir, una carga útil de longitud 0x0a bytes se devuelve desde el dispositivo en la dirección de red 0001, seguida de los bytes de carga útil en bruto que se solicitaron. La carga útil es precedida por un byte de estado que tiene un valor 00.



7. Para solicitar datos de dispositivos que son miembros de un grupo particular, haga lo siguiente:

Pulse y suelte varias veces el botón RB4/RD6 en el dispositivo router. Observe que alterna el LED RA0/D10 de encendido / apagado mientras RB4/RD6 se presiona y se suelta, esta operación nos permite conocer si el dispositivo pertenece o no al grupo 0x0004. Pulse y suelte el botón RB4/RD6 tal que RA0/D10 esté encendido después de soltar el botón. Esto pondrá el Router en el Grupo 0x0004.

En el menú del Coordinador seleccione la opción<3> .

3. Introduzca el ID del grupo al cual quiere solicitar datos: 0004

La respuesta del router es:

Len: 0a
De Dirección: 0001

0000010203040506070809

El router es parte del Grupo 0x0004 por lo que responde con la información anterior.

8. Para enviar los datos directamente al router en el menú de uso del Coordinador coloque la **opción < 4 >** :

4: Enviar datos a otro dispositivo

Introduzca el número de bytes a enviar (hex): **0a**
Introduzca la dirección corta del dispositivo de destino: **0001**

El coordinador enviará diez bytes de datos directamente al router, el mensaje en la pantalla del hyperterminal del Coordinador será: **Mensaje enviado con éxito.**
Observe el tráfico en Zena para confirmar que el mensaje se envía correctamente.

9. Para agregar el Coordinador al Grupo 0x0004, haga lo siguiente desde el menú del Coordinador.

Seleccione la opción < 6 > del menú.

6: Agregar o quitar dispositivos a / de un Grupo
¿Qué actividades del grupo quiere hacer?

0 = Unirse a un grupo de dispositivos
1 = Eliminar dispositivo de un grupo
2 = Eliminar dispositivo de todos los grupos: **0**

Introduzca 16-bit del Grupo ID (Hex): **0004**

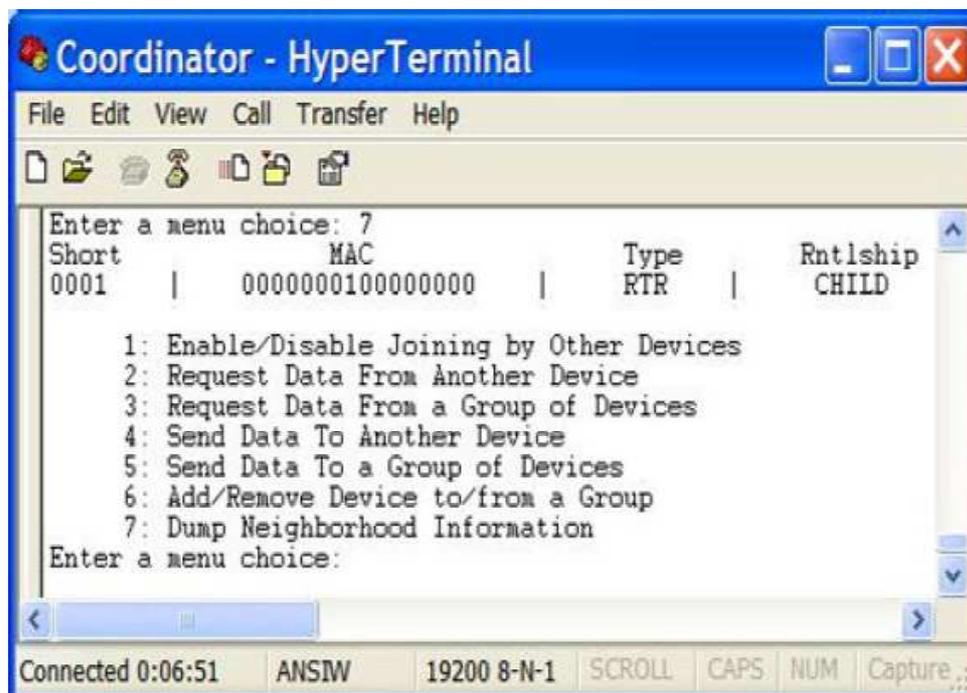
Esto agregará el Coordinador al Grupo ID 0x0004.
Ahora pulsando el botón RB5/RD7 en el router, enviará una solicitud al Coordinador para que este responda mediante el envío de 10 bytes de datos al router.

Utilice el mismo menú y elija la sub-opción 1 o 2 para quitar el Coordinador de un grupo.

10. Para ver qué dispositivos están actualmente almacenados en la tabla de Vecinos del Coordinador elija la **opción 7 en el menú** del Coordinador:

7: Los vecinos del coordinador son:
La información que se mostrará es la siguiente

short	MAC	type	Rntlship
0001	0000000100000000	Router	Child



Esto significa que el router con dirección de red 0001 y dirección MAC 0000000100000000 es un hijo (child) del Coordinador. A medida que haya más dispositivos en la red, esta tabla contendrá más entradas.

FAQs (preguntas frecuentes)

Índice	Síntoma	Solución
1	Los LEDs están apagados en el Coordinador	Pulse el botón restablecer <mclr> en el nodo Coordinador
2	Los LEDs están apagados en el Router / nodo RFD	En primer lugar reiniciar el Coordinador y a continuación reinicie el nodo en cuestión.
3	No hay tráfico en Zena	Asegúrese de que está funcionando el Zena en canal 26. Asegúrese de que el Coordinador ha formado la red para conectar el hyper terminal y pulsando el <CR> <LF> varias veces hasta que se muestre el menú en el Coordinador. Pulse y suelte el botón RB5/RD6
4	No se reciben datos desde otro nodo cuando RB5/RD6 se presiona y se suelta	Asegúrese de que al menos un nodo en la red es un miembro del Grupo 0x0004
5	Los LEDs en el RFD no están parpadeando	El RFD ha perdido el contacto con su padre. Mover el nodo RFD más cercano a un nodo Coordinador / Router que está en la red y presione el botón <mclr>.

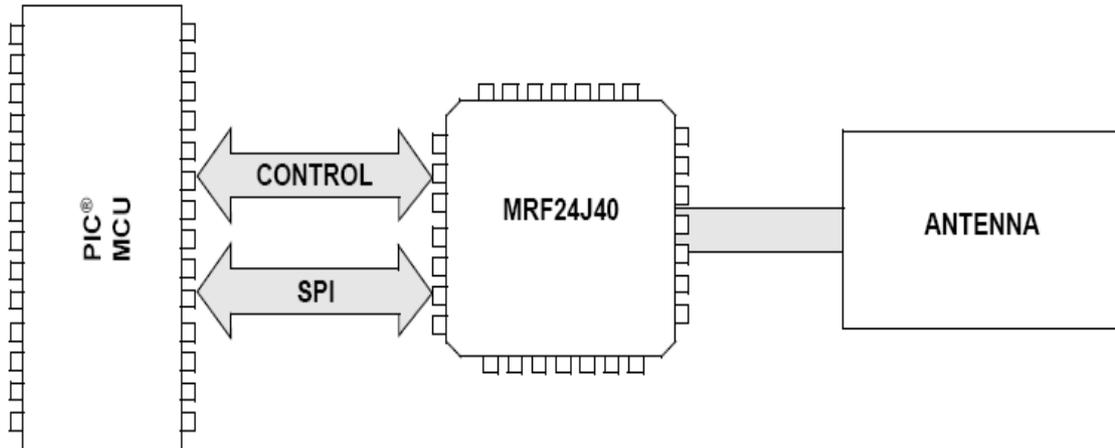
HARDWARE PARA UN NODO CON EL PROTOCOLO ZIGBEE

Para crear un típico nodo ZigBee usando la pila de protocolos de Microchip se necesitan como mínimo los siguientes componentes:

- Un microcontrolador de Microchip con una interfaz de SPI.
- Microchip MFR24J40 RF Transceiver con componentes externos requeridos.
- Una antena, puede ser una antena PCB o antena monopolo.
- Factores externos serial EEPROM (opcional).

Como se muestra en la Figura, el microcontrolador se conecta al Transceiver MRF24J40 a través del bus SPI y unas pocas señales discretas de control. El microcontrolador es el maestro SPI y el Transceiver MRF24J40 actúa como esclavo.

El controlador aplica el estándar IEEE 802.15.4 Medium Access Control (MAC) y las capas de protocolo ZigBee. También contiene la lógica específica de la aplicación. Se utiliza el bus SPI que le permitirá interactuar con el RF Transceiver.



La pila de Microchip proporciona un controlador totalmente integrado, que alivia la aplicación principal de gestión de las funciones del RF Transceiver. Los recursos hardware requeridos por las familias de microcontroladores PIC18F/PIC24F para conducir el RF Transceiver en la aplicación por defecto (proporcionado en el Kit de Demostración PICDEM Z o la plataforma de Explorer 16) se enumeran en la siguiente tabla.

Recursos necesarios del microcontrolador PIC18F para la pila de protocolos Zigbee

PIC18F	PIC24F	Descripción	MRF24J40
INT0	INT1	Usado para aceptar interrupciones del transceiver MRF24J40	INT
TMR0	TMR2	Usado para el símbolo timer	-
RC0	RB2	Chip selection	\overline{CS}
RC1	RG3	Wake-up pin	WAKE
RC2	RG2	Transceiver Reset	\overline{RESET}
RC3	RF6	SPI SCK	SCK
RC4	RF7	SPI SDI	SDO
RC5	RF8	SPI SDO	SDI

Si está utilizando como referencia un esquemático de un nodo ZigBee de la pila de protocolos de microchip, puede empezar a utilizar la pila de Microchip, sin ninguna modificación. Si es necesario, es posible trasladar algunas de las señales de control SPI a los pines de otro puerto para adaptarse a su aplicación de hardware. En este caso, tendrá que modificar las definiciones de la interfaz para poder incluir correctamente la asignación de pines.

El diseño de referencia de microchip para el protocolo ZigBee implementa una antena PCB y un diseño de antena monopolo. Dependiendo de la elección de la antena, usted tendrá que quitar y soldar unos pocos componentes.

El RF Transceiver de microchip es un dispositivo de 3.3V. Dependiendo de los requerimientos, el diseñador puede utilizar ya sea un adaptador o una batería para el suministro de energía. Por lo general, los coordinadores de protocolo de ZigBee y routers se energizan con fuente de alimentación (adaptador) y los dispositivos finales con una batería. Cuando se utiliza como fuente de alimentación la batería, se debe tener cuidado al operar el Transceiver en el rango de tensión especificado.

ANEXO B

CIRCUITO TARJETA MADRE

