



UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIA Y TECNOLOGÍA

ESCUELA DE INGENIERÍA ELECTRÓNICA

“Diseño de una Red ZigBee Aplicada al Monitoreo de un Invernadero”

Trabajo de graduación previo a la obtención del título de Ingeniero Electrónico

AUTOR:

Jonathan Xavier Trelles Muñoz.

DIRECTOR:

Leonel Pérez Rodríguez.

CUENCA – ECUADOR

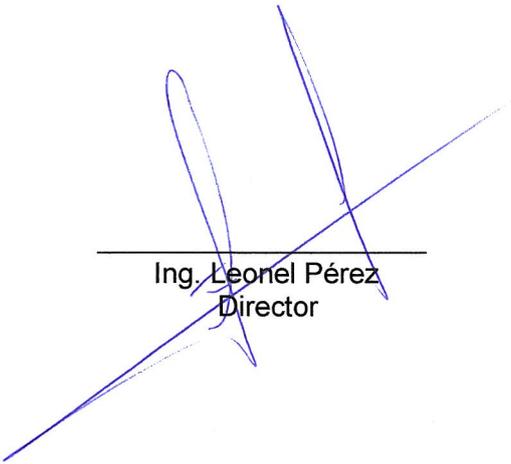
2012

Handwritten signature in red ink, dated 06/01/12.

Diseño de una Red ZigBee Aplicada al Monitoreo de un Invernadero

ABSTRACT

Designing a wireless sensor network applied to the monitoring of important parameters of a greenhouse using the ZigBee wireless communications standard begins with hardware and firmware to control ZigBee modules and sensors interfaces based on Microchip Technology Inc brand offered technology. Due to the characteristics of the greenhouse that was taken as reference, network topology that offers major advantages compared to other groups such as mesh and cluster tree is the star network topology. That is why ZigBee Router Devices take no part in design. Only one ZigBee Coordinator and several ZigBee End Devices do.



Ing. Leonel Pérez
Director



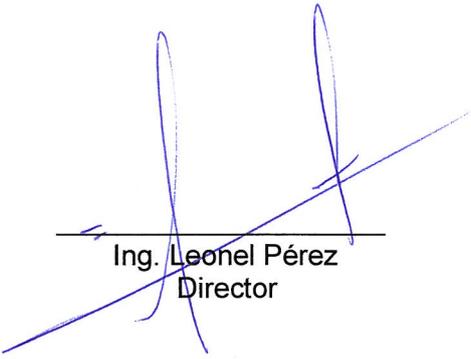
Jonathan Trelles
Autor

Ing. Néstor Bernal
Revisor

Diseño de una Red ZigBee Aplicada al Monitoreo de un Invernadero

RESUMEN

Para diseñar una red de sensores inalámbricos aplicada al monitoreo de los parámetros más importantes de un invernadero utilizando el estándar de comunicaciones inalámbricas ZigBee se empieza con el hardware y firmware para el control de los módulos ZigBee e interfaz con sensores basándose en la tecnología ofertada por la marca Microchip Technology Inc. Dadas las características del invernadero tomado como referencia la topología de red que mayores ventajas ofrece con respecto a otras como malla y grupos de árbol es la de red en estrella. Por tal motivo en el diseño no intervienen Dispositivos Enrutadores ZigBee. Tan sólo un Coordinador y varios Dispositivos Finales.



Ing. Leonel Pérez
Director



Jonathan Trelles
Autor

Ing. Néstor Bernal
Revisor

ÍNDICE DE CONTENIDOS

| | |
|---|-------|
| Resumen..... | ii |
| Abstract..... | iii |
| INTRODUCCIÓN..... | 1 |
| 1. CAPÍTULO 1: ZIGBEE | |
| 1.1. Alianza ZigBee..... | 2 |
| 1.2. Estándar IEEE 802.15.4..... | 2 |
| 1.2.1. Capa Física (PHY) | 3 |
| 1.2.2. Capa de Acceso al Medio (MAC)..... | 8 |
| 1.3. Especificaciones ZigBee..... | 9 |
| 1.3.1. Capa de Red (NWK) | 10 |
| 1.3.2. Capa de Aplicación (APL) | 11 |
| 1.3.2.1. Subcapa de Soporte de Aplicación (APS) | 11 |
| 1.3.2.2. Objetos de Dispositivos ZigBee (ZDO) | 11 |
| 1.3.2.2.1. Coordinadores ZigBee..... | 12 |
| 1.3.2.2.2. Enrutadores ZigBee..... | 12 |
| 1.3.2.2.3. Dispositivos Finales ZigBee..... | 12 |
| 1.3.2.2.4. Objetos de Aplicación Definidos por Fabricantes..... | 12 |
| 2. CAPÍTULO 2: DISEÑO DE HARDWARE PARA CONTROL DE UN MÓDULO ZIGBEE E INTERFAZ CON SENSORES | |
| 2.1. MRF24J40MA..... | 13 |
| 2.1.1. Características..... | 13 |
| 2.1.2. Construcción Interna..... | 16 |
| 2.1.3. Interfaz MRF24J40MA Microcontrolador..... | 18 |
| 2.2. Microcontrolador PIC18F252..... | 19 |
| 2.2.1. Características..... | 19 |
| 2.2.2. Circuito Oscilador Externo del PIC18F252..... | 23 |
| 2.2.3. Fuente de Alimentación del PIC18F252..... | 24 |
| 2.2.4. Conexión con LCD Matricial..... | 25 |
| 2.2.5. Puerto Serial RS232..... | 26 |
| 2.3. Interfaz con Sensores..... | 27 |
| 2.3.1. Sensor de Temperatura..... | 27 |
| 2.3.2. Sensor de Iluminación..... | 29 |

| | |
|---|----|
| 2.3.3. Sensor de Humedad Relativa..... | 32 |
| 2.3.4. Sensor de Humedad del Suelo..... | 34 |
| 2.4. Hardware basado en microcontrolador para el control de un módulo ZigBee..... | 36 |
| 2.4.1. Diagrama de Bloques..... | 37 |

3. CAPÍTULO 3: DISEÑO DE FIRMWARE PARA CONTROL DE UN MÓDULO ZIGBEE

| | |
|---|----|
| 3.1. Diseño del firmware para el control de un dispositivo final..... | 40 |
| 3.1.1. Configuraciones Iniciales..... | 41 |
| 3.1.1.1. Configuración Inicial de Puertos..... | 41 |
| 3.1.1.2. Configuración Inicial del Conversor Analógico Digital..... | 42 |
| 3.1.1.3. Configuración Inicial del Interfaz SPI..... | 42 |
| 3.1.1.4. Configuración Inicial de Parámetros ZigBee..... | 43 |
| 3.1.1.5. Configuración Inicial de Temporizadores..... | 44 |
| 3.1.1.6. Configuración Inicial de Interrupciones..... | 44 |
| 3.1.2. Programa Principal..... | 45 |
| 3.1.2.1. Primitiva 1: NLME_NETWORK_DISCOVERY_confirm..... | 47 |
| 3.1.2.2. Primitiva 2: NLME_JOIN_confirm..... | 47 |
| 3.1.2.3. Primitiva 3: APSDE_DATA_indication..... | 48 |
| 3.1.2.4. Primitiva 4: NO_PRIMITIVE..... | 50 |
| 3.1.2.5. Procesos que no son ZigBee..... | 51 |
| 3.2. Diseño del firmware para el control de un dispositivo coordinador..... | 51 |
| 3.2.1. Configuraciones Iniciales..... | 52 |
| 3.2.1.1. Configuración Inicial de Puertos..... | 53 |
| 3.2.1.2. Configuración Inicial de Interfaz SPI..... | 53 |
| 3.2.1.3. Configuración Inicial de Parámetros ZigBee..... | 54 |
| 3.2.1.4. Configuración Inicial de Interrupciones..... | 54 |
| 3.2.2. Programa Principal..... | 55 |
| 3.2.2.1. Primitiva 1: NLME_NETWORK_FORMATION_confirm..... | 56 |
| 3.2.2.2. Primitiva 2: NLME_PERMIT_JOINING_confirm..... | 56 |
| 3.2.2.3. Primitiva 3: NLME_JOIN_indication..... | 57 |
| 3.2.2.4. Primitiva 4: APSDE_DATA_indication..... | 57 |
| 3.2.2.4.1. Sentencias para endpoint0 (EPZDO) | 57 |
| 3.2.2.4.2. Sentencias para endpoint1 (EPTMP) | 58 |
| 3.2.2.4.3. Sentencias para endpoint2 (EPILUM) | 58 |

| | | |
|------------|--|----|
| 3.2.2.4.4. | Sentencias para endpoint3 (EPHURE) | 58 |
| 3.2.2.4.5. | Sentencias para endpoint4 (EPHUSU) | 58 |
| 3.2.2.5. | Primitiva 5: NO_PRIMITIVE..... | 59 |
| 3.2.2.6. | Procesos que no son ZigBee..... | 59 |

4. CAPÍTULO 4: DISEÑO DE RED ZIGBEE DE SENSORES INALÁMBRICOS

| | | |
|---------------------------------|-------------------------------------|----|
| 4.1. | Topología de Red..... | 61 |
| 4.2. | Elementos que conforman la Red..... | 62 |
| 4.3. | Perfil..... | 63 |
| 4.4. | Direccionamiento..... | 64 |
| 4.5. | Red ZigBee en el Invernadero..... | 65 |
| CONCLUSIONES..... | | 68 |
| RECOMENDACIONES..... | | 70 |
| REFERENCIAS BIBLIOGRÁFICAS..... | | 71 |

ÍNDICE DE TABLAS

| | |
|---|----|
| 1.1. Bandas operacionales de frecuencias de IEEE 802.15.4..... | 4 |
| 1.2. Descripciones de números de canal asociados a páginas de canal..... | 5 |
| 2.1. Descripción de pines de MRF24J40MA..... | 15 |
| 2.2. Tabla lógica de funciones de NC7SZ125P5X..... | 17 |
| 2.3. (a) Descripción de pines del PIC18F252..... | 21 |
| 2.3. (b) Descripción de pines del PIC18F252 (Continuación)..... | 22 |
| 3.1. Configuración de puertos en un dispositivo final..... | 41 |
| 3.2. Configuración inicial del Conversor A/D..... | 42 |
| 3.3. Configuración inicial del Interfaz SPI..... | 43 |
| 3.4. Configuración inicial de Parámetros ZigBee..... | 43 |
| 3.5. Configuración inicial del timer 0..... | 44 |
| 3.6. Configuración inicial de Interrupciones..... | 44 |
| 3.7. Estructura principal de configuraciones en un dispositivo coordinador..... | 53 |
| 3.8. Configuración inicial del Interfaz SPI..... | 53 |
| 3.9. Configuración inicial de Parámetros ZigBee..... | 54 |
| 3.10. Configuración inicial de Interrupciones..... | 54 |
| 4.1. Características de los dispositivos..... | 67 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| 1.1. Mapa de bits: asignación del canal 12 de la página de canal 2..... | 5 |
| 1.2. Selección de Canales de IEEE 802.15.4-2006 para 2,4 GHz..... | 6 |
| 1.3. Comparación entre selección de canales para 2,4 GHz de 802.11b y 802.15.4..... | 7 |
| 1.4. Formato General de Trama de Capa de Red..... | 10 |
| 2.1. Diagrama de pines de MRF24J40MA..... | 14 |
| 2.2. Diagrama de bloques del MRF24J40MA..... | 16 |
| 2.3. Construcción interna de MRF24J40MA..... | 17 |
| 2.4. Diagrama de bloques para la conexión entre el MRF24J40MA y el microcontrolador..... | 18 |
| 2.5. Diagrama de pines de PIC18F252..... | 20 |
| 2.6. Circuito oscilador externo del PIC18F252..... | 23 |
| 2.7. Circuito convertidor de 3VDC a 5VDC..... | 24 |
| 2.8. Conexión de LCD con PIC18F252..... | 25 |
| 2.9. Interfaz RS232-Microcontrolador..... | 26 |
| 2.10. Interfaz Sensor de temperatura-Microcontrolador..... | 28 |
| 2.11. Señal simulada de salida del interfaz de temperatura..... | 29 |
| 2.12. PCB del circuito de interfaz de temperatura..... | 29 |
| 2.13. Interfaz Sensor de iluminación-Microcontrolador..... | 30 |
| 2.14. Señal simulada de salida del interfaz de iluminación..... | 31 |
| 2.15. PCB del circuito de interfaz de iluminación..... | 32 |
| 2.16. Interfaz Sensor de humedad relativa-Microcontrolador..... | 32 |
| 2.17. Señal simulada de salida del interfaz de humedad relativa..... | 33 |
| 2.18. PCB del circuito de interfaz de humedad relativa..... | 34 |
| 2.19. Interfaz Sensor de humedad del suelo-Microcontrolador..... | 34 |
| 2.20. Señal simulada de salida del interfaz de humedad del suelo..... | 35 |
| 2.21. PCB del circuito de interfaz de humedad del suelo..... | 36 |
| 2.22. Hardware basado en microcontrolador para el control de un módulo ZigBee..... | 36 |
| 2.23. PCB del Hardware para control del módulo ZigBee..... | 37 |
| 2.24. Diagrama de Bloques del Hardware para control de un módulo ZigBee..... | 38 |
| 3.1. Estructura principal del firmware de un dispositivo final..... | 40 |
| 3.2. Estructura principal de configuraciones en un dispositivo final..... | 41 |
| 3.3. Estructura del programa principal de un dispositivo final..... | 46 |
| 3.4. Estructura principal del firmware de un dispositivo coordinador..... | 52 |

| | |
|---|----|
| 3.5. Estructura principal de configuraciones en un dispositivo coordinador..... | 52 |
| 3.6. Estructura del programa principal de un dispositivo coordinador..... | 55 |
| 4.1. Invernadero de “Productos Agropecuarios Don Luchito”..... | 61 |
| 4.2. Topología de red en estrella..... | 62 |
| 4.3. Perfil no certificado Invernadero..... | 63 |
| 4.4. Red ZigBee en el Invernadero de Solano..... | 66 |
| 4.5. Simbología del Plano de la figura 4.4..... | 66 |
| 4.6. Simbología del Plano de la figura 4.4..... | 67 |

Jonathan Xavier Trelles Muñoz.
Trabajo de Grado.
Ing. Leonel Pérez Rodríguez.
Enero 2012.

DISEÑO DE UNA RED ZIGBEE APLICADA AL MONITOREO DE UN INVERNADERO

INTRODUCCIÓN

El constante avance tecnológico en el área de las telecomunicaciones representa una gran ventaja para incontables aplicaciones en diversos ámbitos de la sociedad humana. Las redes de sensores inalámbricos son tan sólo una de esa gran variedad de aplicaciones que las telecomunicaciones brindan hoy en día. Dichas redes de sensores inalámbricos pueden ser usadas de varias formas en diferentes ambientes. Dada su precariedad en una gran medida, uno de los sectores de la región que mayores beneficios puede obtener con redes de sensores inalámbricos es el agropecuario. Principalmente en el área de cultivos en invernaderos.

Existen varios estándares de comunicaciones inalámbricas que se pudieran utilizar en una red de sensores. Uno sumamente atractivo por sus características es ZigBee. Bajo el estándar ZigBee se pueden desarrollar muchísimas aplicaciones de comunicaciones inalámbricas. Principalmente redes de sensores para ámbitos tales como domótica, control y bioelectrónica, entre otros. Su flexibilidad, alcance, bajo consumo y bajo costo hacen de ZigBee un estándar de redes inalámbricas de área personal que busca satisfacer necesidades que con algunos estándares tales como bluetooth y otros no se puede lograr. Es por los motivos antes mencionados que el presente trabajo de grado se centra en elaborar un diseño de una red ZigBee para monitorear los parámetros ambientales más importantes al interior de un invernadero.

CAPÍTULO 1

ZIGBEE

1.1 Alianza ZigBee

Establecida en 2002, la alianza ZigBee es una asociación abierta y sin ánimo de lucro de organizaciones de todo tipo de alrededor del mundo. La alianza ZigBee pretende desarrollar un estándar de comunicaciones inalámbricas de bajo consumo y bajo costo que permita optimizar la vida útil de baterías. Este estándar de comunicaciones inalámbricas se llama ZigBee. La alianza ZigBee trabaja conjuntamente con la IEEE para garantizar el desarrollo del estándar. De hecho, ZigBee se basa en el estándar IEEE 802.15.4. Con ZigBee se pretende satisfacer necesidades que protocolos como Bluetooth y otros no logran alcanzar.

En principio, la alianza ZigBee buscó diseñar redes ad hoc de auto organización para radios digitales. La misión actual de la alianza es trabajar para desarrollar estándares libres de comunicación inalámbrica cada vez más flexibles. Como resultado de ese pensamiento se obtienen características muy deseables para las comunicaciones inalámbricas por medio de ZigBee. IEEE 802.15.4 es el estándar de comunicaciones en el que se basa ZigBee ya que define las capas física (PHY) y de acceso al medio (MAC). Los niveles superiores del stack de protocolos no son definidos por el estándar IEEE 802.15.4. La alianza ZigBee tiene como objetivo desarrollarlos.

1.2 Estándar IEEE 802.15.4

Entre las características más importantes de IEEE 802.15.4 están el bajo consumo, bajo costo y flexibilidad de la red. Puede tener aplicaciones tan diversas como domóticas o industriales e incluso médicas. Cualquier aplicación que requiera una

baja tasa de transmisión inalámbrica de datos. ZigBee se basa en el estándar IEEE 802.15.4. Este estándar define las capas física (PHY) y de acceso al medio (MAC). Las capas de red (NWK) y de aplicación (APL) son definidas por la alianza.

IEEE 802.15.4 es un estándar de comunicaciones para Redes Inalámbricas de Área Personal de Baja Tasa de transmisión ó Low Rate Wireless Personal Area Network (LR-WPAN) enfocado al desarrollo de redes de sensores inalámbricos. La banda operacional de frecuencias del estándar es la banda ISM (Industrial, Scientific and Medical), principalmente la de 2.4 GHz que es una banda libre en todo el mundo. La banda de frecuencias ISM también es utilizada por otros estándares IEEE 802 de comunicación inalámbrica.

“IEEE 802.15.4 define tres bandas de frecuencia para su operación: 2.4 GHz, 915 MHz y 868 MHz. Cada banda de frecuencias ofrece un número fijo de canales, por ejemplo, la banda de frecuencias de 2.4 GHz ofrece 16 canales (canales del 11 al 26), la de 915 MHz ofrece 10 canales (canales del 1 al 10) y la de 868 MHz ofrece 1 canal (canal 0)”¹. El estándar IEEE 802.15.4 define sólo las capas Física (PHY) y de Acceso al Medio (MAC). PHY se encarga de establecer los mecanismos de transmisión y recepción de datos sobre el medio físico y MAC es responsable de una comunicación confiable entre dos dispositivos a través de un enlace físico directo, es decir, sin nodos intermedios.

1.2.1 Capa Física (PHY)

PHY es la encargada de activar y desactivar el radio transceptor, sintonizar el canal de la frecuencia, realizar la corrección de errores a nivel físico, realizar la detección de portadora, estimar la intensidad de la señal recibida y realizar la codificación y la modulación de los datos. La tabla 1.1 muestra las bandas de frecuencias disponibles para la capa física de IEEE 802.15.4 junto con sus parámetros más importantes. Se puede observar el número de canales por banda, sus tasas de transmisión y los lugares donde son bandas no licenciadas. La banda de 2,4GHz es comúnmente la más utilizada por ser una banda libre en todo el mundo.

¹ Nota de Aplicación AN1232 de Microchip en www.microchip.com

| Frecuencia | Número de Canales | Tasa de Transmisión (kbps) | Frecuencia no Licenciada |
|------------|-------------------|----------------------------|--------------------------|
| 2,4 GHz | 16 (11 al 26) | 250 | Todo el mundo |
| 915 MHz | 10 (1 al 10) | 40 ó 250 | Norteamérica y Australia |
| 868 MHz | 1(canal 0) | 20 ó 100 | Europa |

Tabla 1.1. Bandas operacionales de frecuencias de IEEE 802.15.4.

La última revisión del estándar fue realizada en el 2006. La versión revisada del estándar IEEE.802.15.4 de 2003 realizada en 2006 implementa mejoras importantes. Extiende la aplicabilidad del mercado, remueve ambigüedades encontradas en la edición de 2003, reduce complejidades innecesarias y realiza mejoras relacionadas con la implementación del estándar. La edición revisada del estándar ofrece cuatro especificaciones de capa física. La elección de una u otra depende de situaciones regulatorias y de las preferencias de cada usuario. Son las siguientes:

- Capa física de espectro ensanchado por secuencia directa (DSSS por sus siglas en inglés) de 915/868 MHz empleando modulación por desplazamiento binario de fase (BPSK).
- Capa física DSSS de 915/868 MHz empleando modulación por compensación de cuadratura de desplazamiento de fase (O-QPSK).
- Capa física de espectro extendido por secuencia paralela (PSSS) de 915/868 MHz empleando BPSK y ASK (modulación por desplazamiento de amplitud).
- Capa física DSSS de 2450 MHz empleando modulación O-QPSK.

De acuerdo a ambas ediciones del estándar, las bandas operacionales de 915 y 868 MHz, en el aire soportan tasas de transmisión de 20 y 10 Kbps y opcionalmente tasas de 250 y 100 Kbps respectivamente. La banda de 2,4 GHz soporta una tasa de transmisión de 250 Kbps en el aire. La edición de 2006 dice que la asignación de canales debe ser definida por una combinación de números de canal y páginas de canal.

“Los primeros cinco bits más significativos (MSBs) del mapa de bits de un canal de 32 bits en canales compatibles de capa física deben ser usados como un valor entero para especificar 32 posibles páginas de canal. Los siguientes 27 bits del mapa de bits del canal deben ser usados como una máscara de bits para

especificar los números de canal dentro de la página de canal². Esto quiere decir que existirán 27 canales numerados de 0 a 26 por cada página de canal y un total de 32 páginas de canal. Por ejemplo, el mapa de bits de la figura 1.1 representaría el canal 12 de la página de canal 2.

Figura 1.1. Mapa de bits que representa la asignación del canal 12 de la página de canal 2.

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

Dependiendo de la correlación entre los números de canal y las páginas de canal se asignan características específicas en cuanto a frecuencia y modulación para cada canal. La tabla 1.2 muestra las características de cada canal. Se puede notar que para la banda de 915 MHz, de acuerdo a lo establecido en la edición revisada del estándar, ahora se cuenta con 30 posibilidades de asignación de canales usando 3 diferentes tipos de modulación y para la banda de 868 MHz 3 posibilidades de asignación con 3 diferentes tipos de modulación. La banda de 2,4 GHz se mantiene con 16 canales y un solo tipo de modulación.

| Página de Canal (Decimal) | Página de Canal (Binario) (b31,b30,b29,b28,b27) | Número de Canal (Decimal) | Descripción de Número de Canal |
|---------------------------|---|---------------------------|---|
| 0 | 00000 | 0 | Canal 0 está en la banda de 868 MHz usando BPSK |
| | | 1 - 10 | Canales del 1 al 10 están en la banda de 915 MHz usando BPSK |
| | | 11 - 26 | Canales del 11 al 26 están en la banda de 2,4 GHz usando O-QPSK |
| 1 | 00001 | 0 | Canal 0 está en la banda de 868 MHz usando ASK |
| | | 1 - 10 | Canales del 1 al 10 están en la banda de 915 MHz usando ASK |
| | | 11 - 26 | Reservado |
| 2 | 00010 | 0 | Canal 0 está en la banda de 868 MHz usando O-QPSK |
| | | 1 - 10 | Canales del 1 al 10 están en la banda de 915 MHz usando O-QPSK |
| | | 11 - 26 | Reservado |
| 3 - 31 | 00011 - 11111 | Reservado | Reservado |

Tabla 1.2. Descripciones de números de canal asociados a páginas de canal.

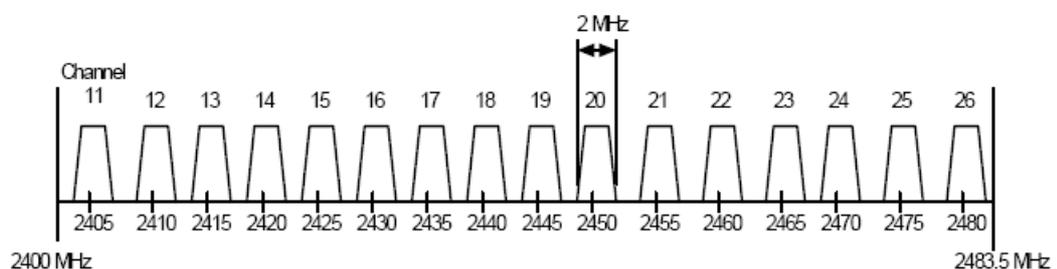
Fuente: IEEE Std 802.15.4™-2006.

La banda de 2,4 GHz es la banda más utilizada debido a que es libre en todo el mundo. Al ser la más utilizada resulta lógico deducir que es la banda que se toma como punto de partida para la elaboración de la mayoría de estudios de redes inalámbricas de área personal. El presente documento no es una excepción. La

² Tomado de IEEE Std 802.15.4™-2006 (Revision of IEEE Std 802.15.4-2003)

selección de canales para la banda de 2,4 GHz de acuerdo a lo establecido en la edición del estándar IEEE 802.15.4 de 2006 se muestra en la figura 1.2. Se puede observar que la banda de 2,4 GHz empieza en 2400 MHz y acaba en 2483,5 MHz. Existen 16 canales numerados del 11 al 26 en los que cada canal tiene una separación de 5 MHz entre las frecuencias centrales de cada canal y un ancho de banda de 2 MHz. La frecuencia central del primer canal de la banda de 2,4 GHz es de 2405 MHz y del último canal es de 2480 MHz.

Figura 1.2. Selección de Canales de IEEE 802.15.4-2006 para 2,4 GHz.



Fuente: IEEE Std 802.15.4™-2006.

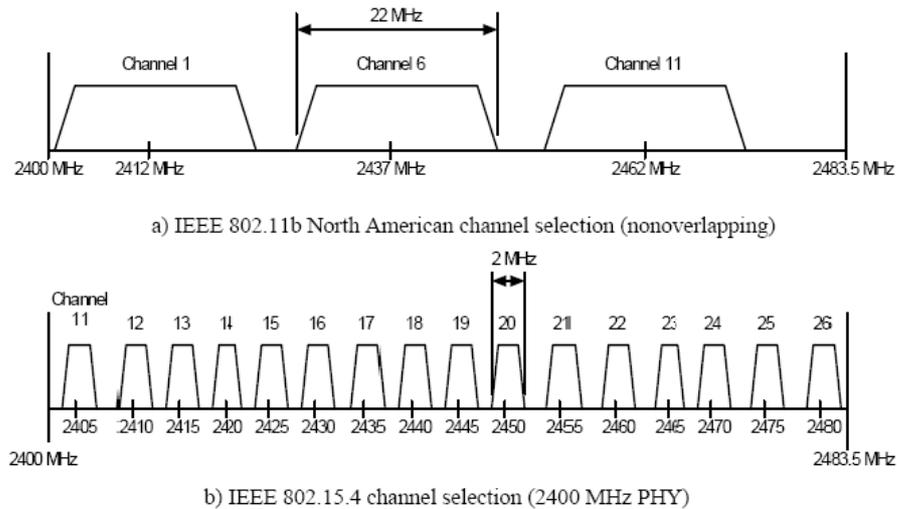
La selección adecuada de canales es sumamente importante al momento de reducir la probabilidad de interferencia. Algunos estándares de comunicación inalámbrica, entre ellos el IEEE 802.11 en todas sus especificaciones por ejemplo, pudieran causar problemas de interferencia debido a que también utilizan las bandas ISM para su operación. Cuando un sistema utiliza el estándar IEEE 802.11b definido para Norteamérica, que es el que se utiliza en Ecuador, para prevenir problemas de interferencia se suele seleccionar canales que no se traslapan.

Los canales de la banda de 2,4 GHz definidos en el estándar 802.11b que no se traslapan son los canales 1, 6 y 11. De acuerdo a la edición de 2006 del estándar IEEE 802.15.4, la probabilidad de interferencia entre sistemas que utilicen la banda de 2,4 GHz se reduce gracias a que existen 4 canales que se encuentran entre las bandas de guarda de 802.11b. Se realiza la comparación con IEEE 802.11b debido a que muchos de los estándares de comunicación inalámbrica realizan una selección de canales de manera similar.

Incluso el estándar IEEE 802.11n ratificado en 2008, que por haber sido ratificado más tarde que la elaboración de la edición del estándar IEEE 802.15.4 de 2006 no es considerado en dicho documento, lo hace. La figura 1.3 muestra una

comparación entre la banda de 2,4 GHz establecida en el estándar IEEE 802.11b definido para Norteamérica (a) y la establecida en el estándar IEEE 802.15.4 (b). Se puede observar que los canales 15, 20, 25 y 26 de 802.15.4 se encuentran entre las bandas de guarda de los canales 1, 6 y 11, que son canales que no se traslapan, en 802.11b.

Figura 1.3. Comparación entre selección de canales para 2,4 GHz de 802.11b y 802.15.4.



Fuente: IEEE Std 802.15.4™-2006.

IEEE 802.15.4 es un estándar de comunicaciones inalámbricas de bajo consumo. Es por eso que el estándar especifica que los dispositivos deben tener una potencia de transmisión de al menos -3dBm , lo que significa que, si se asume una ganancia de antena de 0dBi , un transmisor que cumpla con IEEE 802.15.4 operará 8dB menos que uno que cumpla con IEEE 802.15.3 y entre 12 y 18dB menos que otro que cumpla con IEEE 802.11b. La baja potencia de transmisión ayuda también a minimizar la interferencia. El nivel máximo de potencia de transmisión para la banda de $2,4\text{GHz}$, debido a situaciones regulatorias, en Ecuador es de 1000mW que equivale a 30dBm . El nivel de entrada máximo de recepción debe ser mayor o igual a -20dBm .

Los dispositivos creados para cumplir con este estándar, debido a su aplicabilidad, por lo general son alimentados por baterías. Con el fin de reducir el consumo de energía y aumentar la vida útil de las baterías de los dispositivos, el estándar especifica que éstos requieren duty cycling, a más de pasar la mayor parte de su

vida operacional en un estado de dormidos y escuchar periódicamente los canales de radiofrecuencia con el fin de determinar si existen mensajes pendientes.

1.2.2 Capa de Acceso al Medio (MAC)

MAC se encarga del direccionamiento en capa dos de los dispositivos, el entramado de los datos y la validación de las tramas recibidas, la gestión de los canales de acceso, la asociación y disociación de dispositivos y el envío de tramas de reconocimiento. En la capa de acceso al medio, la estructuración de las tramas se las va realizando con información que cada protocolo de cada subcapa específica va agregando.

El estándar define cuatro estructuras de tramas: tramas de guía, que son las encargadas de avisar a otros dispositivos que un dispositivo existe; tramas de datos, que se encargan de la transmisión de datos; tramas de reconocimiento, que se usan para confirmar la recepción de tramas; y tramas de comandos MAC, que sirven para el direccionamiento en capa 2 de los dispositivos. El estándar IEEE 802.15.4 utiliza el mecanismo de Acceso Múltiple por Detección de Portadora con Evasión de Colisiones (CSMA-CA por sus siglas en inglés), que es muy utilizado en comunicaciones inalámbricas, y lo puede utilizar en dos formas dependiendo de la configuración de la red. Cuando las Redes de Área Personal (PAN) tienen inhabilitado el envío de tramas de guía, el mecanismo de acceso al canal que se utiliza es el CSMA-CA sin espaciar. Si tienen habilitado el envío de tramas guía, se utiliza el CSMA-CA espaciado.

Según lo publicado en IEEE 802.15.4-2006, cuando se utiliza CSMA-CA sin espaciar, el dispositivo que desee enviar tramas de datos o de comandos MAC espera un espacio de tiempo aleatorio. Si se encuentra un canal libre, el dispositivo enviará la trama el siguiente espacio aleatorio. Si no se encuentra un canal libre, el dispositivo esperará al siguiente espacio aleatorio antes de volver a tratar de acceder al canal. En este caso las tramas de reconocimiento se envían sin utilizar el mecanismo CSMA-CA.

Cuando se utiliza CSMA-CA espaciado, los espacios de tiempo no son aleatorios y son alineados con el inicio de la transmisión de las tramas de guía. Una vez que se detectan dispositivos que intentan comunicarse, uno de ellos, llamado dispositivo

coordinador, sincroniza los espacios de tiempo. Si un dispositivo desea enviar una trama de datos, encuentra el límite al final del siguiente espacio sincronizado y espera un número aleatorio de espacios. Si el canal se encuentra disponible, el dispositivo empezará la transmisión cuando detecte el límite al final del siguiente espacio disponible. Si el canal está ocupado, el dispositivo esperará otra vez un número aleatorio de espacios antes de volver a tratar de acceder al canal. Este mecanismo ayuda a evitar la presencia de colisiones en la red.

1.3 Especificaciones ZigBee

La Alianza ZigBee ha desarrollado dos tipos de especificaciones: una para redes en malla y otra para aplicaciones de control remoto de dispositivos que no requieren de las capacidades que una red tipo malla ofrece (control remoto de un televisor por ejemplo). La especificación para aplicaciones de control remoto de dispositivos se llama ZigBee RF4CE Specification Version 1.00. RF4CE viene de Radio Frecuencia Para Consumidores Electrónicos. La especificación fue ratificada el 17 de marzo de 2009 y su análisis se desviaría de los objetivos que rigen la elaboración del presente trabajo de grado. La especificación que define la operación de las redes tipo malla de ZigBee se denomina Especificación ZigBee.

Existen dos opciones de implementación disponibles en la especificación ZigBee: ZigBee y ZigBee PRO. ZigBee PRO es la opción más usada gracias a su optimización en cuanto a su consumo energético y a que soporta redes muy grandes que pueden llegar a tener hasta miles de dispositivos. La última revisión de la especificación ZigBee fue ratificada el 17 de enero de 2008. Define las capas de Red (NWK) y de Aplicación (APL).

La especificación ZigBee dice que existen tres tipos de dispositivos ZigBee: coordinador ZigBee, enrutadores ZigBee y dispositivos finales ZigBee. Un coordinador ZigBee se encarga de la elección del canal y de la identificación de la red. Es el dispositivo que básicamente inicia la creación de la red y luego pasa a actuar como un enrutador. Un enrutador ZigBee se encarga del direccionamiento en capa 3 de los dispositivos finales ya que éstos no poseen ninguna tabla que contenga información de enrutamiento. Los dispositivos finales son administrados por los enrutadores y coordinadores y son libres de fijar su propio duty cycle con el

fin de ahorrar energía. Son dispositivos que dependen de enrutadores y coordinadores para su correcta comunicación.

1.3.1 Capa de Red (NWK)

Una de las capas que define la especificación ZigBee es la capa de red (NWK). La capa de red es la encargada del correcto funcionamiento del enrutamiento de los dispositivos de la red. Cada trama de la capa de red tiene dos componentes básicos: cabecera de trama y carga útil de trama. La cabecera de la trama contiene información tal como direccionamiento y secuencia. La carga útil de la trama contiene información específica del tipo de trama y es de longitud variable. La figura 1.4 muestra el formato general de una trama de la capa de red. En ella se pueden observar la cabecera y la carga útil.

Figura 1.4. Formato General de Trama de Capa de Red.

| Bytes: | 2 | 2 | 1 | 1 | 0/8 | 0/8 | 0/1 | Variable | Variable |
|--------|----------------------------------|-------------------|------------------|-------|---------------------|------------------------|-----------------------|----------------------|------------|
| | Control | Dirección Destino | Dirección Origen | Radio | Número de Secuencia | Dirección IEEE Destino | Dirección IEEE Origen | Control de Multicast | Carga Útil |
| | Cabecera de Trama de Capa de Red | | | | | | | | Carga Útil |

Fuente: Especificación ZigBee Documento 053474r17. ZigBee Standards Organization.

El campo control es de 16 bits (2 Bytes) y es en donde se establecen las configuraciones básicas de la trama de capa de red. El campo dirección destino contiene la dirección en capa 3 del dispositivo al que se desea llegar con la información. El campo dirección origen contiene la dirección en capa 3 del dispositivo en donde se origina el traslado de información. El campo radio determina el límite del rango de transmisión. Debe disminuirse en uno cada vez que la trama ha sido recibida por un dispositivo. El número de secuencia sirve para identificar una trama. Debe incrementarse en uno por cada trama nueva.

La dirección IEEE destino contiene la dirección en capa 2 del dispositivo al que se desea llegar con la información. La dirección IEEE origen contiene la dirección en capa 2 del dispositivo anterior que envía la trama. El campo control de multicast determina qué dispositivos son los que tienen que recibir la información en

multicast. Para activar la opción de multicast hay que activar una bandera en el campo control. El campo carga útil contiene información específica acerca del tipo de trama.

1.3.2 Capa de Aplicación (APL)

Otra de las capas definidas por la especificación ZigBee es la capa de aplicación (APL). La capa de aplicación consta de tres partes: la subcapa de soporte de aplicación (APS), los objetos de dispositivos ZigBee (ZDO por sus siglas en inglés) y los objetos de aplicación definidos por fabricantes.

1.3.2.1 Subcapa de Soporte de Aplicación (APS)

La subcapa de soporte de aplicación (APS) hace uso de un conjunto de servicios utilizados por los objetos de dispositivos ZigBee (ZDO) y los objetos de aplicación definidos por fabricantes con el fin de establecer un interfaz entre la capa de red (NWK) y la capa de aplicación (APL). Existen dos tipos de entidades de servicios: entidad de datos y entidad de mantenimiento. La primera se encarga de la transmisión de datos entre varias entidades de la capa de aplicación de una misma red. La segunda se encarga de proveer varios tipos de servicios, incluyendo los de seguridad, a los dispositivos de la capa de aplicación.

1.3.2.2 Objetos de Dispositivos ZigBee (ZDO)

Los objetos de dispositivos ZigBee (ZDO por sus siglas en inglés) son aplicaciones que emplean la capa de red y la subcapa de aplicación con el fin de presentar interfaces en la capa de aplicación para el control de dispositivos y de las funciones de red. Los objetos de dispositivos ZigBee se usan específicamente para cada tipo de dispositivo ZigBee. Existen tres tipos de dispositivos ZigBee: coordinadores ZigBee, enrutadores ZigBee y dispositivos finales ZigBee.

1.3.2.2.1 Coordinadores ZigBee

Se encargan de iniciar la formación de la red mediante el establecimiento de procesos que permitan realizar tareas como la identificación de la red y la elección del canal. Una vez iniciada la red pasaría a cumplir funciones de enrutador.

1.3.2.2.2 Enrutadores ZigBee

Se encargan de definir hacia dónde y cómo enviar el tráfico que llega a otros dispositivos de la red. Principalmente de los dispositivos finales ZigBee, los que no tienen ningún tipo de tabla de enrutamiento.

1.3.2.2.3 Dispositivos Finales ZigBee

Son los dispositivos que se encuentran en el extremo lógico más lejano de la red. Son administrados por los coordinadores y enrutadores ZigBee. No tienen ningún tipo de información de enrutamiento. Pueden entrar en un estado de dormidos y establecer su propio duty cycle con el fin de ahorrar energía.

1.3.2.3 Objetos de Aplicación Definidos por Fabricantes

Los objetos de aplicación definidos por fabricantes comparten la subcapa de aplicación y los servicios de seguridad con los objetos de dispositivos ZigBee (ZDO). Pero además, utilizan el marco de aplicación. El marco de aplicación es un ambiente en el que los objetos de aplicación se organizan. Los fabricantes que finalmente son los que implementan la aplicación pueden definir hasta 240 distintos objetos de aplicación.

CAPITULO 2

DISEÑO DE HARDWARE PARA CONTROL DE UN MÓDULO ZIGBEE E INTERFAZ CON SENSORES

Este capítulo presenta el diseño del hardware para el control de un módulo ZigBee e interfaz con sensores basado en microcontrolador. Entre los componentes más destacados se encuentran el módulo de radio frecuencia de Microchip MRF24J40MA (Módulo ZigBee) y el microcontrolador que permitirá realizar el control del mismo.

2.1 MRF24J40MA

El módulo MRF24J40MA es un transceptor de radiofrecuencia basado en el módulo MRF24J40. Tiene ciertas características que representan una ventaja sobre este último. Entre ellas está que el módulo MRF24J40MA posee circuitería de acoplamiento y antena PCB integrados en él. Las características principales del módulo MRF24J40MA (llamado también módulo ZigBee) se describen a continuación.

2.1.1 Características

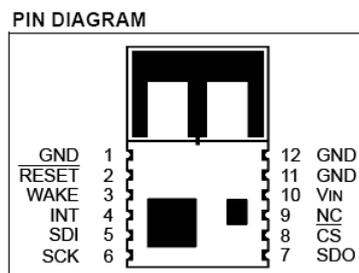
El módulo MRF24J40MA (llamado también módulo ZigBee por soportar el estándar) es un módulo transceptor de radiofrecuencia de bajo costo y consumo fabricado por Microchip. Usa el estándar para redes inalámbricas de área personal IEEE 802.15.4 en el que se fundamenta ZigBee. Lo que quiere decir que integra funcionalidades de capa física y de capa de acceso al medio en un solo circuito integrado. Trabaja en la banda ISM de 2,4GHz. Soporta protocolos tales como MiWi, MiWi P2P, ZigBee y otros protocolos propietarios de redes inalámbricas. Posee bajo consumo

de corriente: típicamente 19mA como receptor, 23mA como transmisor y 2uA en modo dormido. Posee un interfaz periférico serial (SPI por sus siglas en inglés) que permitirá la comunicación con el microcontrolador. Su velocidad de transmisión es de 250kbps (correspondiente a lo dictaminado por IEEE 802.15.4).

Es compatible con los microcontroladores de Microchip de las familias PIC16F, PIC18F, PIC24F/H, dsPIC33 y PIC32. A diferencia del módulo MRF24J40, que es de 40 pines, el módulo MRF24J40MA posee tan solo 12. Sin embargo, su funcionalidad no se ve afectada por esta situación. Aunque la diferencia de la cantidad de pines entre uno y otro parezca excesiva, la única ventaja que tiene el primero sobre el segundo es el poseer 6 entradas / salidas digitales de propósito general, las cuales no son estrictamente necesarias para el desarrollo de este capítulo. Otra diferencia, y a su vez una ventaja sobresaliente, es que el módulo MRF24J40MA tiene integrados una antena PCB, circuitería de acoplamiento, cristal oscilador y regulador de voltaje internamente, elementos que el módulo MRF24J40 no incluye. El módulo MRF24J40MA puede llegar a tener un rango de cobertura, teóricamente, de hasta 400 pies (121,92m).

En la capa de acceso al medio, el módulo ZigBee MRF24J40MA utiliza el mecanismo CSMA-CA (Acceso Múltiple por Detección de Portadora con Evasión de Colisiones), respuesta a tramas de reconocimiento automática y secuencia de verificación de trama incorrecta (FCS por sus siglas en inglés). Es capaz de retransmitir paquetes automáticamente. También posee tramas beacon independientes y utiliza como sistema de seguridad el algoritmo del estándar de encriptado avanzado de 128 bits (AES-128 por sus siglas en inglés). La figura 2.1 muestra el diagrama de pines del módulo ZigBee MRF24J40MA. La tabla 2.1 muestra la descripción de los pines del módulo ZigBee. La descripción fue tomada de la hoja de datos del módulo.

Figura 2.1. Diagrama de pines de MRF24J40MA.



Fuente: MRF24J40MA Data Sheet. Microchip INC.

PIN DESCRIPTION

| Pin | Symbol | Type | Description |
|-----|--------|--------|---|
| 1 | GND | Power | Ground |
| 2 | RESET | DI | Global hardware Reset pin |
| 3 | WAKE | DI | External wake-up trigger |
| 4 | INT | DO | Interrupt pin to microcontroller |
| 5 | SDI | DI | Serial interface data input |
| 6 | SCK | DI | Serial interface clock |
| 7 | SDO | DO | Serial interface data output from MRF24J40 |
| 8 | CS | DI | Serial interface enable |
| 9 | NC | — | No connection (allow pin to float; do not connect signal) |
| 10 | VIN | Power | Power supply |
| 11 | GND | Ground | Ground |
| 12 | GND | Ground | Ground |

Legend: Pin type abbreviation: D = Digital, I = Input, O = Output

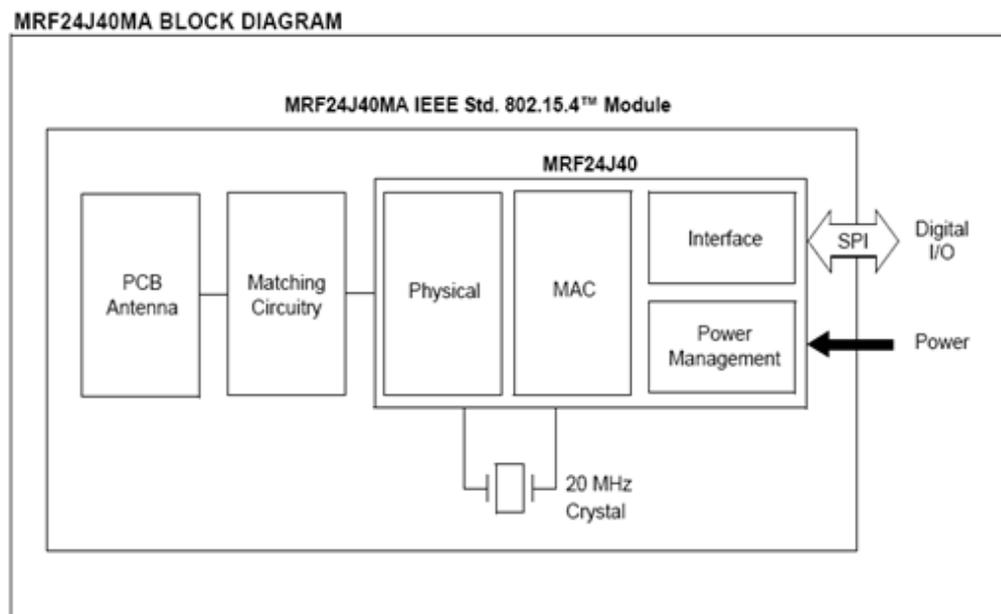
Tabla 2.1. Descripción de pines de MRF24J40MA.

Fuente: MRF24J40MA Data Sheet. Microchip INC.

Se puede observar en la tabla 2.1 que el módulo consta de 12 pines. De los cuales 1 no tiene conexión: pin 9. El pin 10 es destinado a alimentación. Los pines 1, 11 y 12 son tierra. El pin 2 permite establecer un reset por hardware. El pin 3 es una entrada digital que permite despertar el módulo ZigBee por medio de un disparo cuando éste está en modo dormido. El pin 4 es una salida digital que permite establecer una interrupción al microcontrolador cuando el módulo ZigBee lo requiera.

Para poder controlar el módulo con un microcontrolador existe lo que se denomina el interfaz periférico serial (SPI). El SPI es el medio por el que se comunican el módulo y el microcontrolador. El microcontrolador también debe poseer un SPI para conectarse con el del módulo. En el módulo ZigBee, los pines que cumplen la función de SPI son: pines del 5 al 8. El pin 5 (SDI) es la entrada de datos del interfaz serial del módulo (SPI). El pin 6 (SCK) es la entrada del reloj del SPI. El pin 7 (SDO) es la salida de datos del módulo. El pin 8 (CS negado) es una entrada digital que permite habilitar el SPI. La figura 2.2 muestra el diagrama de bloques del MRF24J40MA.

Figura 2.2. Diagrama de bloques del MRF24J40MA.



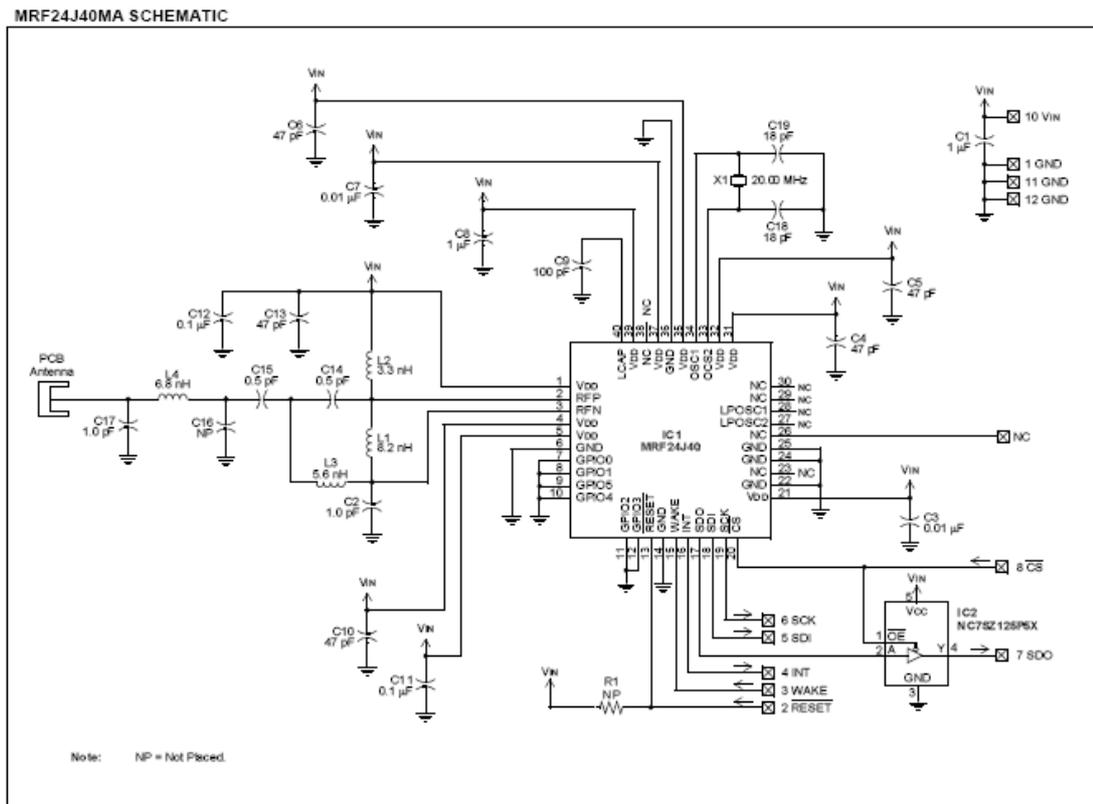
Fuente: MRF24J40MA Data Sheet. Microchip INC.

En la figura 2.2 se puede observar que el módulo MRF24J40MA incluye circuitería de acoplamiento entre la antena y el módulo, el cristal oscilador de 20MHz y una antena PCB integrada dentro del mismo módulo. También se puede observar que la base de su construcción es el MRF24J40.

2.1.2 Construcción Interna

La figura 2.3 muestra la construcción interna del módulo ZigBee. En la figura 2.3 se puede observar que la base del módulo MRF24J40MA es el módulo MRF24J40. El pin 7 (SDO), que es la salida de datos del interfaz serial, se conecta internamente a través de un circuito integrado pequeño denominado NC7SZ125P5X. Este circuito integrado es un buffer (seguidor) de ultra alta velocidad (UHS por sus siglas en inglés) con tres estados de salida. Sus estados de salida pueden ser: nivel lógico alto, nivel lógico bajo y estado de alta impedancia. La tabla 2.2 muestra la tabla de funciones del NC7SZ125P5X.

Figura 2.3. Construcción interna de MRF24J40MA.



Fuente: MRF24J40MA Data Sheet. Microchip INC.

| Inputs | | Output |
|--------|------|--------|
| OE | In A | Out Y |
| L | L | L |
| L | H | H |
| H | X | Z |

H = HIGH Logic Level
L = LOW Logic Level
X = HIGH or LOW Logic Level
Z = HIGH Impedance State

Tabla 2.2. Tabla lógica de funciones de NC7SZ125P5X.
Fuente: NC7SZ125 Data Sheet. Fairchild Semiconductor INC.

De acuerdo a las conexiones del buffer y a la tabla 2.2, se puede observar que la función del NC7SZ125P5X es establecer un estado de alta impedancia en el ingreso del interfaz serial del microcontrolador cuando el pin que habilita el interfaz serial del módulo ZigBee (pin 8) recibe un pulso alto. En otras palabras, el buffer tiene como función impedir que el ingreso del interfaz serial del microcontrolador reciba niveles altos de corriente, sino tan solo los niveles lógicos que el módulo transceptor envía. De esa forma se logra proteger el pin de ingreso del interfaz serial del microcontrolador y también evitar posibles fallas, las cuales se pudieran

dar por características propias del material con el que se fabrica el componente (silicio), en la lectura de datos del microcontrolador.

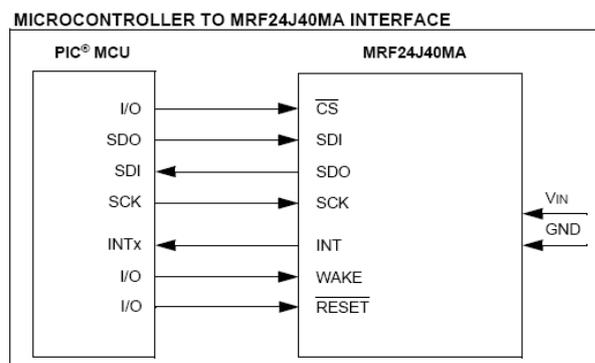
Los otros componentes de la construcción interna del módulo ZigBee se describen a continuación. El componente X1 del circuito mostrado en la figura 2.3 es el cristal oscilador de cuarzo de 20 MHz. El cristal oscilador tiene como función definir la frecuencia a la que va trabajar el módulo. Los componentes L1, L3, C2 y C14 forman un circuito adaptador de impedancias con el fin de transformar la impedancia de entrada no balanceada de la antena (para este caso una antena PCB) en una impedancia de entrada balanceada del módulo.

L2 es un inductor de choque, es decir, impide el paso de corriente alterna, por mínima que sea, y deja pasar sólo corriente continua a los pines RFP y RFN del MRF24J40. C15 es un capacitor que bloquea el ingreso de corriente continua. L4, C16 y C17 forman un filtro pasa bajo con el fin de dejar pasar sólo las frecuencias de interés. Dado que el módulo ZigBee es un transceptor de radiofrecuencia, es decir, se comunica inalámbricamente, los capacitores restantes forman filtros que tienen como fin disminuir en la mayor medida posible los diferentes tipos de ruido que se puedan encontrar en el ambiente.

2.1.3. Interfaz MRF24J40MA Microcontrolador

La figura 2.4 muestra un diagrama de bloques para la conexión entre el módulo ZigBee y el microcontrolador.

Figura 2.4. Diagrama de bloques para la conexión entre el MRF24J40MA y el microcontrolador.



Fuente: MRF24J40MA Data Sheet. Microchip INC.

En la figura 2.4 se puede observar que además del SPI, el cual se explicó previamente, es necesario conectar los pines INT, WAKE y RESET negado al microcontrolador. INT es una salida digital que tiene como objetivo establecer una interrupción en el microcontrolador cuando el módulo necesite enviarle sus datos. WAKE es una entrada digital que despierta el módulo cuando el microcontrolador requiere enviarle sus datos. RESET negado es una entrada digital que restablece las configuraciones por defecto del módulo ZigBee.

2.2 Microcontrolador PIC18F252

El módulo ZigBee MRF24J40MA es compatible con los microcontroladores de Microchip de las familias PIC16F, PIC18F, PIC24F/H, dsPIC33 y PIC32. La elección del microcontrolador más apto para el control del módulo depende de la aplicación para la que se diseña la red ZigBee. La aplicación que propone el presente trabajo de grado es el monitoreo de parámetros ambientales al interior de un invernadero. Las señales que por lo general se monitorean en un invernadero no requieren un procesamiento digital tan agudo como pudieran ofrecer los microcontroladores de la familia dsPIC33.

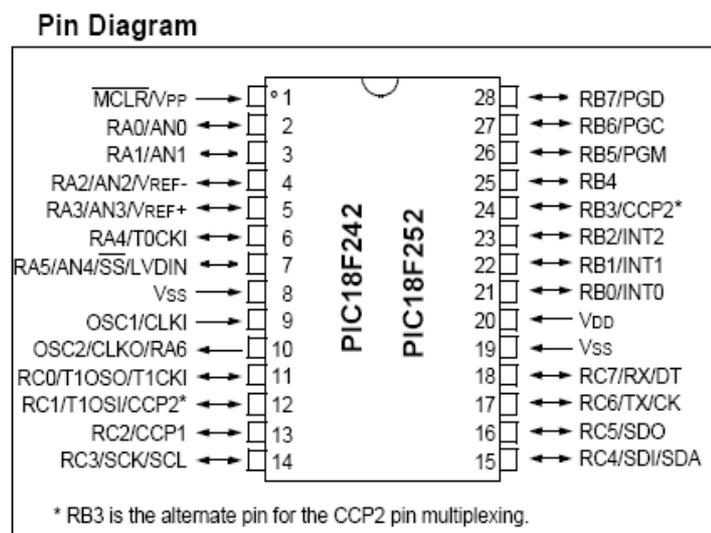
Tampoco se requiere de características demasiado sobresalientes como las que poseen los microcontroladores de las familias PIC24F/H y PIC32. La familia PIC16F es un poco básica con respecto a las demás. Es por tal motivo que los microcontroladores de la familia PIC18F son ideales para el control del módulo ZigBee. Los microcontroladores PIC18F252 poseen un interfaz periférico serial (SPI) y por lo tanto pueden comunicarse con el módulo ZigBee.

2.2.1 Características

El PIC18F252 es un microcontrolador de 28 pines de alto rendimiento con memoria flash mejorada de 32 Kbyte que posee un conversor analógico digital de 10 bits con alta tasa de muestreo. Es capaz de manejar hasta 25mA. Posee tres pines para interrupciones externas. También tiene 4 temporizadores/contadores numerados del 0 al 3. Posee 2 módulos Compara/Captura/PWM (CCP). Además, tiene un

Puerto Serial Síncrono Maestro (MSSP por sus siglas en inglés) que puede trabajar en 2 modos: SPI e I²C. Esta última característica permite al microcontrolador comunicarse mediante SPI con el módulo ZigBee. Su rango de operación es de 2V a 5.5V. Otra característica que lo hace ideal para controlar el módulo ZigBee es su bajo consumo pues típicamente consume menos de 1.6mA trabajando a 5V y 4MHz. La figura 2.5 muestra el diagrama de pines del microcontrolador PIC18F252 de Microchip.

Figura 2.5. Diagrama de pines de PIC18F252.



Fuente: PIC18Fxx2 Data Sheet. Microchip INC.

En la figura 2.5 se puede apreciar que el PIC18F252 es un microcontrolador de 28 pines. Se observan los nombres de las posibles funciones que pudiera tener cada pin. Las tablas 2.3-a y 2.3-b muestran la descripción de los pines del PIC18F252.

PIC18F2X2 PINOUT I/O DESCRIPTIONS

| Pin Name | Pin Number | | Pin Type | Buffer Type | Description |
|--|------------|------|--------------------|-------------------------------|--|
| | DIP | SOIC | | | |
| MCLR/VPP MCLR VPP | 1 | 1 | I I | ST ST | Master Clear (input) or high voltage ICSP programming enable pin. Master Clear (Reset) input. This pin is an active low RESET to the device. High voltage ICSP programming enable pin. |
| NC | — | — | — | — | These pins should be left unconnected. |
| OSC1/CLKI OSC1 CLKI | 9 | 9 | I I | ST CMOS | Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. ST buffer when configured in RC mode, CMOS otherwise. External clock source input. Always associated with pin function OSC1. (See related OSC1/CLKI, OSC2/CLKO pins.) |
| OSC2/CLKO/RA6 OSC2 CLKO RA6 | 10 | 10 | O O I/O | — — TTL | Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate. General Purpose I/O pin. |
| RA0/AN0 RA0 AN0 | 2 | 2 | I/O I | TTL Analog | PORTA is a bi-directional I/O port. Digital I/O. Analog input 0. |
| RA1/AN1 RA1 AN1 | 3 | 3 | I/O I | TTL Analog | Digital I/O. Analog input 1. |
| RA2/AN2/VREF- RA2 AN2 VREF- | 4 | 4 | I/O I I | TTL Analog Analog | Digital I/O. Analog input 2. A/D Reference Voltage (Low) input. |
| RA3/AN3/VREF+ RA3 AN3 VREF+ | 5 | 5 | I/O I I | TTL Analog Analog | Digital I/O. Analog input 3. A/D Reference Voltage (High) input. |
| RA4/T0CKI RA4 T0CKI | 8 | 8 | I/O I | ST/OD ST | Digital I/O. Open drain when configured as output. Timer0 external clock input. |
| RA5/AN4/SS/LVDIN RA5 AN4 SS LVDIN RA6 | 7 | 7 | I/O I I I | TTL Analog ST Analog | Digital I/O. Analog input 4. SPI Slave Select input. Low Voltage Detect Input. See the OSC2/CLKO/RA6 pin. |

Tabla 2.3-a. Descripción de pines del PIC18F252.

Fuente: PIC18Fxx2 Data Sheet. Microchip INC.

PIC18F2X2 PINOUT I/O DESCRIPTIONS (CONTINUED)

| Pin Name | Pin Number | | Pin Type | Buffer Type | Description |
|--|------------|-------|-------------------|------------------|---|
| | DIP | SOIC | | | |
| RB0/INT0 RB0 INT0 | 21 | 21 | I/O I | TTL ST | PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs. Digital I/O. External Interrupt 0. |
| RB1/INT1 RB1 INT1 | 22 | 22 | I/O I | TTL ST | External Interrupt 1. |
| RB2/INT2 RB2 INT2 | 23 | 23 | I/O I | TTL ST | Digital I/O. External Interrupt 2. |
| RB3/CCP2 RB3 CCP2 | 24 | 24 | I/O I/O | TTL ST | Digital I/O. Capture2 input, Compare2 output, PWM2 output. |
| RB4 | 25 | 25 | I/O | TTL | Digital I/O. Interrupt-on-change pin. |
| RB5/PGM RB5 PGM | 26 | 26 | I/O I/O | TTL ST | Digital I/O. Interrupt-on-change pin. Low Voltage ICSP programming enable pin. |
| RB6/PGC RB6 PGC | 27 | 27 | I/O I/O | TTL ST | Digital I/O. Interrupt-on-change pin. In-Circuit Debugger and ICSP programming clock pin. |
| RB7/PGD RB7 PGD | 28 | 28 | I/O I/O | TTL ST | Digital I/O. Interrupt-on-change pin. In-Circuit Debugger and ICSP programming data pin. |
| RC0/T1OSO/T1CKI RC0 T1OSO T1CKI | 11 | 11 | I/O O I | ST — ST | PORTC is a bi-directional I/O port. Digital I/O. Timer1 oscillator output. Timer1/Timer3 external clock input. |
| RC1/T1OSI/CCP2 RC1 T1OSI CCP2 | 12 | 12 | I/O I I/O | ST CMOS ST | Digital I/O. Timer1 oscillator input. Capture2 input, Compare2 output, PWM2 output. |
| RC2/CCP1 RC2 CCP1 | 13 | 13 | I/O I/O | ST ST | Digital I/O. Capture1 input/Compare1 output/PWM1 output. |
| RC3/SCK/SCL RC3 SCK SCL | 14 | 14 | I/O I/O I/O | ST ST ST | Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I ² C mode |
| RC4/SDI/SDA RC4 SDI SDA | 15 | 15 | I/O I I/O | ST ST ST | Digital I/O. SPI Data In. I ² C Data I/O. |
| RC5/SDO RC5 SDO | 16 | 16 | I/O O | ST — | Digital I/O. SPI Data Out. |
| RC6/TX/CK RC6 TX CK | 17 | 17 | I/O O I/O | ST — ST | Digital I/O. USART Asynchronous Transmit. USART Synchronous Clock (see related RX/DT). |
| RC7/RX/DT RC7 RX DT | 18 | 18 | I/O I I/O | ST ST ST | Digital I/O. USART Asynchronous Receive. USART Synchronous Data (see related TX/CK). |
| Vss | 8, 19 | 8, 19 | P | — | Ground reference for logic and I/O pins. |
| VDD | 20 | 20 | P | — | Positive supply for logic and I/O pins. |

Legend: TTL = TTL compatible input
 ST = Schmitt Trigger input with CMOS levels
 O = Output
 OD = Open Drain (no P diode to VDD)
 CMOS = CMOS compatible input or output
 I = Input
 P = Power

Tabla 2.3-b. Descripción de pines del PIC18F252 (Continuación).
 Fuente: PIC18Fxx2 Data Sheet. Microchip INC.

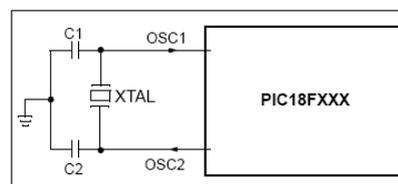
En las tablas 2.3-a y 2.3-b se pueden observar las posibles funciones que pudiera tener cada pin del microcontrolador PIC18F252. Se observa que el interfaz SPI utiliza los pines 14 (SCK), 15 (SDI) y 16 (SDO), todos pertenecientes al puerto C. El microcontrolador cuenta con 3 pines que pueden establecer interrupciones externas. Los pines son: 21, 22 y 23. Todos pertenecientes al puerto B. El módulo ZigBee enviará una señal a uno de estos pines (dependiendo de su conexión) cuando requiera establecer una interrupción en el microcontrolador.

El pin 1 se utiliza para generar un reset externo en el microcontrolador. Los pines del 2 al 7 forman la mayor parte del puerto A (RA0:RA5). RA6 es el pin 10. Sin embargo, este pin es comúnmente utilizado para la conexión del cristal oscilador (OSC2). Los pines del 21 al 28 forman el puerto B (RB0:RB6) y del 11 al 18 el puerto C (RC0:RC6). Cada pin puede tener diferentes opciones de funcionamiento pero sólo una opción a la vez. Las diferentes opciones se muestran en las tablas 2.3-a y 2.3-b. Los pines 8 y 19 se conectan a tierra. El pin 20 es la alimentación del microcontrolador y el pin 9 es la entrada del cristal oscilador (OSC1).

2.2.2 Circuito oscilador externo del PIC18F252

El microcontrolador PIC18F252 requiere de algún medio para establecer la frecuencia a la que va a trabajar. Se propone utilizar un cristal oscilador de cuarzo. El circuito oscilador externo y su forma de conexión con el microcontrolador se muestran en la figura 2.6.

Figura 2.6. Circuito oscilador externo del PIC18F252.



Fuente: PIC18Fxx2 Data Sheet. Microchip INC.

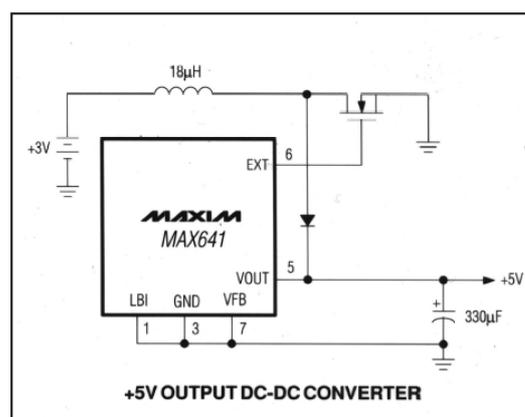
En la figura 2.6 se puede observar que el cristal oscilador de cuarzo se conecta a los pines OSC1 (pin 9) y OSC2 (pin 10) del microcontrolador. También es necesario

conectar 2 capacitores cuyos valores dependen de la frecuencia de oscilación del cristal. Mientras mayor es la frecuencia del cristal oscilador menor es el ciclo de máquina. El ciclo de máquina es el tiempo mínimo en el que el microcontrolador realiza una acción. El ciclo de máquina es igual a 4 períodos del circuito oscilador externo. Se propone utilizar un cristal de 16MHz. Los valores de los capacitores para una frecuencia del oscilador de 16MHz, de acuerdo a la hoja de datos del microcontrolador, son de: entre 10pF a 22pF tanto para C1 (figura 2.6) como para C2 (figura 2.6). Se propone utilizar capacitores de 22pF.

2.2.3 Fuente de alimentación del PIC18F252

Al tratarse de comunicación inalámbrica se requiere que el PIC18F252 sea alimentado con baterías. ZigBee es un estándar de comunicaciones inalámbricas que optimiza la vida útil de las mismas. Las características eléctricas del microcontrolador PIC18F252 especifican un mínimo de 4.2V y un máximo de 5.5V como voltaje de alimentación. La figura 2.7 muestra un circuito que permite obtener 5VDC para la alimentación del microcontrolador. El circuito es alimentado con 2 baterías alcalinas AA de 1.5V.

Figura 2.7. Circuito convertidor de 3VDC a 5VDC.



Fuente: MAX641 Data Sheet. MAXIM.

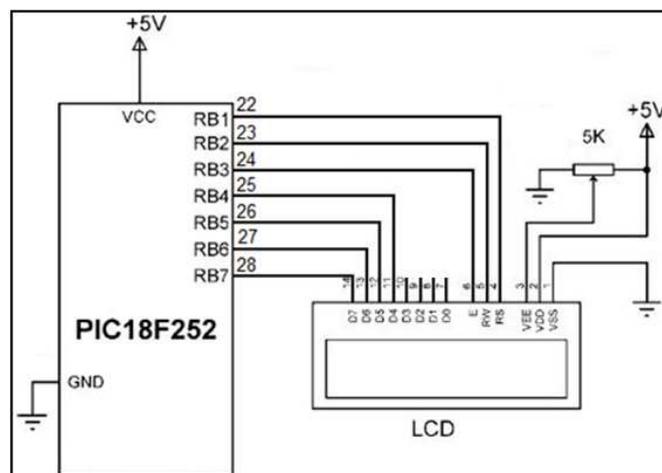
En la figura 2.7 se puede observar un circuito convertidor de 3VDC a 5VDC. La batería de +3V de la figura 2.7 se consigue al conectar 2 baterías alcalinas AA de

1.5V en serie. El pin 5 del MAX641 se conecta a la alimentación (VDD) del microcontrolador PIC18F252.

2.2.4 Conexión con LCD matricial

Con el fin de mostrar en pantalla las lecturas de las mediciones de los parámetros ambientales del invernadero, se considera la opción de conectar un LCD matricial al microcontrolador. Se considera un LCD matricial L1672 que es un LCD de 2 líneas y 16 posibles caracteres por cada línea. Existen 3 posibles formas de conectar un LCD matricial a un microcontrolador dependiendo de su modo de operación: 4 bits utilizando la parte baja del puerto, 4 bits utilizando la parte alta del puerto y 8 bits utilizando todo el puerto. A fin de optimizar los recursos del microcontrolador se propone conectar el LCD matricial en modo de operación de 4 bits utilizando la parte alta del puerto B. La figura 2.8 muestra la conexión entre el PIC18F252 y el LCD matricial L1672.

Figura 2.8. Conexión de LCD con PIC18F252.



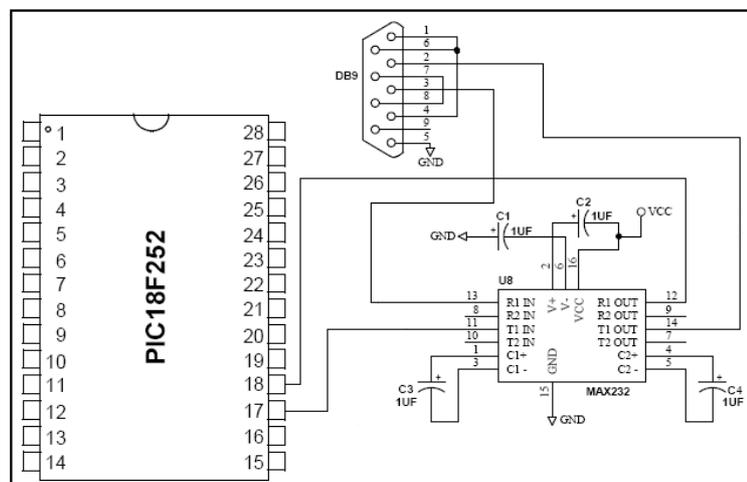
En la figura 2.8 se puede observar la conexión entre el PIC18F252 y el LCD matricial L1672. Se observa que se lo conecta en modo de operación de 4 bits usando la parte alta del puerto B. Los pines desde D4 hasta D7 del LCD se conectan a los pines RB4 a RB7 del PIC18F252 y es en donde se colocará ya sea el dato o la instrucción del LCD. Los pines de RB1 a RB3 del PIC18F252 se

conectan a los pines de control del LCD: pines del 4 al 6. El pin 4 (E) habilita la operación del LCD. El pin 5 (R/W) permite indicar si se va a escribir en el LCD o si se va a leer de él. El pin 4 (RS) indica si al LCD llega un dato o una instrucción. Los pines del 1 al 3 del LCD sirven para su alimentación y para la regulación del contraste del mismo. Su forma de conectar se muestra en la figura 2.8.

2.2.5 Puerto Serial RS232

A fin de poder establecer una forma de comunicación entre el hardware basado en microcontrolador para el control de un módulo ZigBee (de ahora en adelante llamado circuito controlador) y un dispositivo que permita el monitoreo de los parámetros medidos por los sensores en el invernadero, se propone la conexión entre ambos mediante el módulo transmisor receptor síncrono asíncrono direccionable universal (USART por sus siglas en inglés) del microcontrolador. De esta forma, cualquier circuito controlador usado en el diseño de la red ZigBee podrá ser programado como dispositivo coordinador ZigBee y comunicarse con una PC. El módulo USART es uno de los dos módulos seriales del microcontrolador PIC18F252. Los pines asociados al módulo son RC6/TX/CK (pin 17) y RC7/RX/DT (pin 16). Se puede establecer un interfaz con una computadora mediante el puerto serial RS232 de ésta. La figura 2.9 muestra el interfaz RS232-microcontrolador.

Figura 2.9. Interfaz RS232-Microcontrolador.



En la figura 2.9 se observa que el PIC18F252 se conecta al conector RS232 (Conector DB9) a través de un circuito integrado MAX232. MAX232 es un driver que permite adaptar niveles de tensión TTL o CMOS a niveles de la norma EIA-232 (norma para comunicación serial), o viceversa. Los valores de los capacitores de la figura 2.9 y su forma de conectarlos son proporcionados por la hoja de datos del circuito integrado MAX232.

2.3 Interfaz con sensores

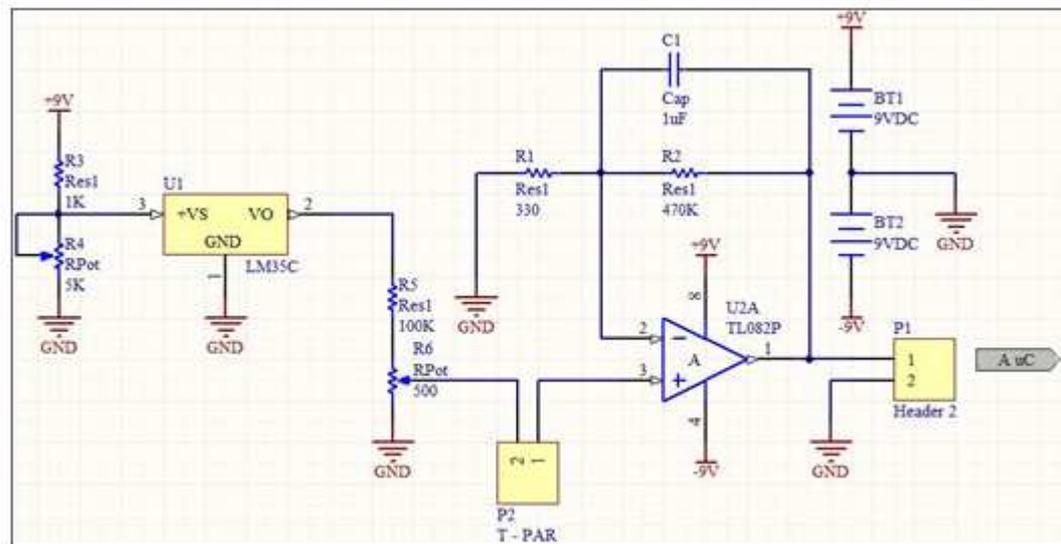
Los sensores a utilizar en el monitoreo de parámetros ambientales del invernadero requieren de circuitería de acoplamiento y linealización con el fin de establecer una variación del voltaje de entrada común en el microcontrolador. Esta variación debe estar comprendida dentro de un rango de 0V a 5V. A continuación se describen los sensores y sus circuitos de acoplamiento y linealización.

2.3.1 Sensor de temperatura

Una forma bastante común de realizar mediciones de temperatura es haciendo uso de termopares. Pero para hacer uso de los mismos hay que recordar que se requiere compensar su unión fría, la cual podría provocar lecturas incorrectas de temperatura. Para realizar la compensación de la unión fría se hace uso de otro sensor de temperatura. Uno de los sensores de temperatura más utilizados para este propósito es el circuito integrado LM35C. El LM35C es un sensor integrado de temperatura que varía linealmente 10mV por cada grado centígrado.

Para que el microcontrolador PIC18F252 pueda realizar la conversión de los valores analógicos entregados por el termopar mediante su conversor analógico digital en valores digitales para ser procesados por él es necesario adaptar su señal para que la temperatura tenga una variación entre 0V y 5V dentro de un rango de interés luego de haber compensado la unión fría. La figura 2.10 muestra el circuito que logra adaptar dicha señal a los valores de interés compensando la unión fría de un termopar tipo k.

Figura 2.10. Interfaz Sensor de temperatura-Microcontrolador.



El componente P2 del circuito de la figura 2.10 sirve para alojar en la tarjeta de circuito impreso (PCB por sus siglas en inglés) una bornera doble en la que se conecta un termopar tipo K. A fin de compensar la unión fría del termopar se utiliza un sensor integrado de temperatura LM35C. Para compensar la unión fría del termopar se requiere que la variación del voltaje de salida del LM35C con la temperatura sea igual a la variación del termopar. Esta variación recibe el nombre α . En el termopar tipo K α es igual a $31\mu\text{V}/^\circ\text{C}$ y en el LM35C es $10\text{mV}/^\circ\text{C}$. Para igualar el valor de α del LM35C y el termopar se ubica un potenciómetro a la salida del LM35C. El potenciómetro R4 debe ajustarse en un valor de resistencia de 311Ω para lograrlo.

Dado que α es un valor muy pequeño se requiere amplificar la señal de medida. Es por esto que se conecta a un amplificador no inversor formado por los componentes U2A, R1, R2. C1 es un capacitor que tiene como objetivo filtrar el ruido del ambiente. El voltaje de salida del amplificador operacional variará linealmente entre 0V y 5V con un cambio de temperatura entre 0°C y aproximadamente 56°C . De esta forma la señal queda lista para ingresar al microcontrolador. La figura 2.11 muestra señal de salida del interfaz del sensor de temperatura simulada en Altium Designer Winter 09. La figura 2.12 muestra el PCB del circuito de la figura 2.10.

Figura 2.11. Señal simulada de salida del interfaz de temperatura.

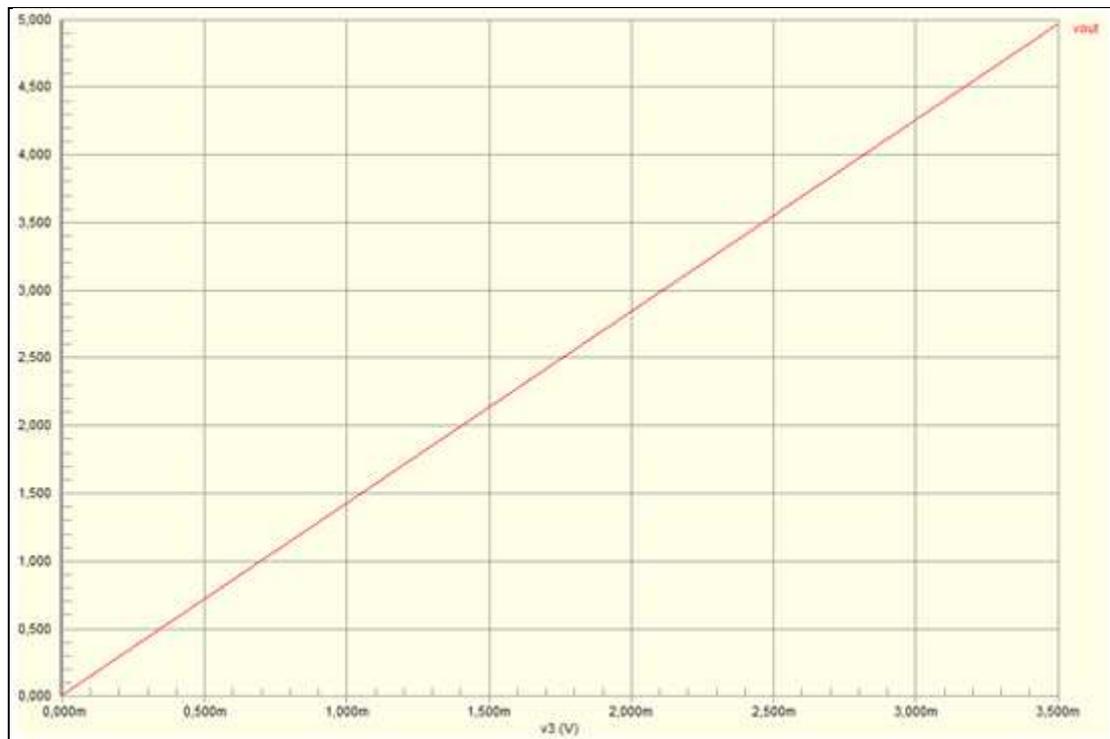
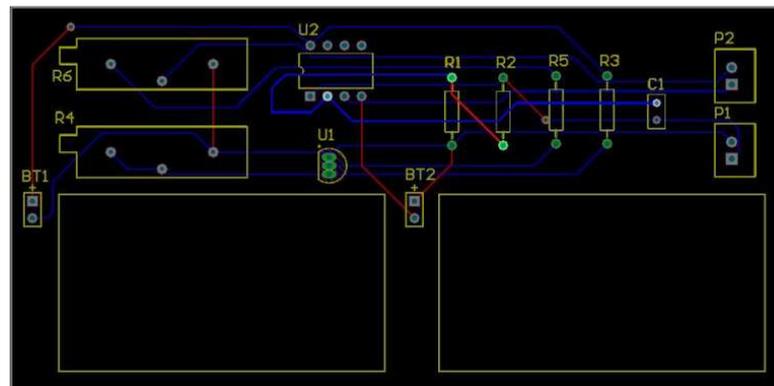


Figura 2.12. PCB del circuito de interfaz de temperatura.



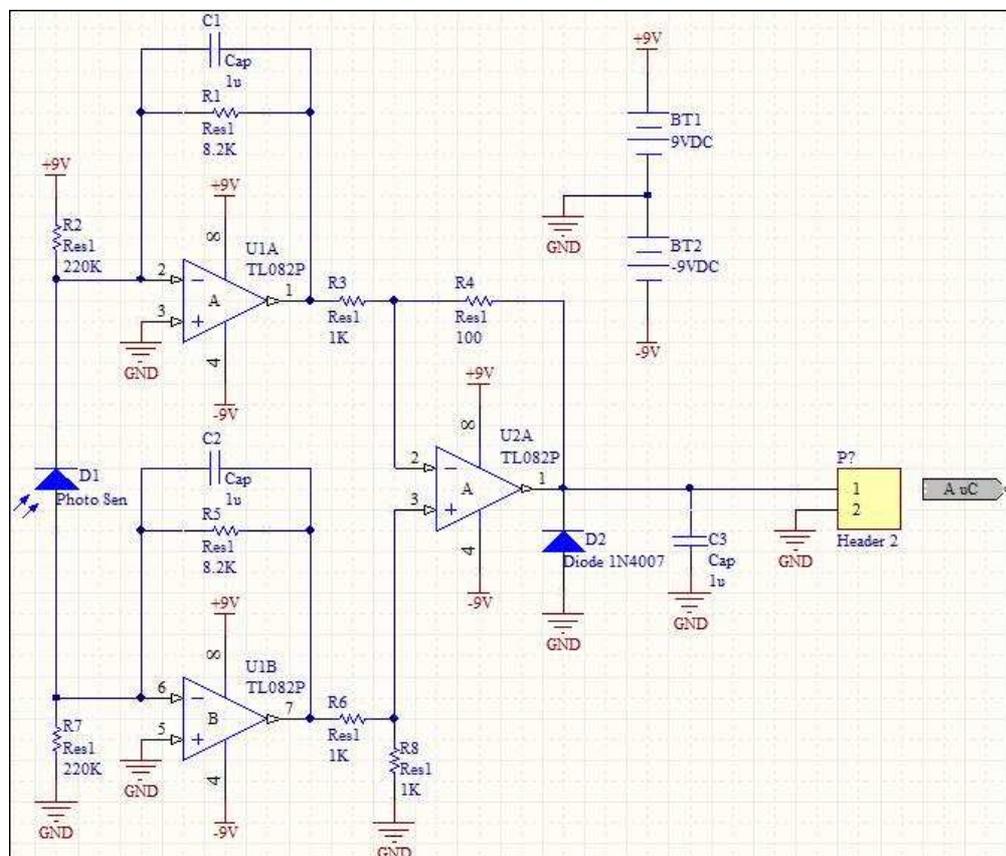
2.3.2 Sensor de iluminación

Existen muchas formas de medir la incidencia de la luz visible. Una de las mejores y más comunes es con un fotodiodo. Sin embargo, al igual que con la temperatura, un fotodiodo requiere de circuitería de acoplamiento antes de que su señal de

salida pueda ingresar al microcontrolador. Por un fotodiodo circula una pequeña corriente el momento que se inciden fotones sobre él. Pero además es un elemento sensible a los cambios de temperatura. Esto representa un problema al momento de realizar la medición puesto que se pudieran obtener lecturas erróneas.

Además, siempre está presente el problema del ruido en el ambiente. Una forma de evitar los efectos del cambio de temperatura sobre el fotodiodo es conectándolo en configuración fotovoltaica en cortocircuito. Esto quiere decir cortocircuitar el fotodiodo sin unir sus pines. Se lo puede hacer con amplificadores operacionales ya que por su alta impedancia de ingreso poseen un cortocircuito virtual entre sus entradas. También se puede utilizar amplificadores operacionales para solucionar el problema del ruido. Para lograrlo hay que conectarlo como amplificador restador. Se debe tener en cuenta que lo único que se desea restar es el ruido. Al hacer uso de amplificadores operacionales también se logra obtener una variación de voltaje con la variación de corriente del fotodiodo. Al amplificarla lo suficiente la señal quedaría lista para ser enviada al microcontrolador. La figura 2.13 muestra el circuito de acoplamiento del fotodiodo.

Figura 2.13. Interfaz Sensor de iluminación-Microcontrolador.



En el circuito de la figura 2.13 se puede observar la forma de conectar un fotodiodo en configuración fotovoltaica en cortocircuito haciendo uso de los amplificadores operacionales conectados en configuración inversora marcados como U1A y U1B. El amplificador operacional marcado como U2A es un amplificador restador que logra restar sólo la señal de ruido gracias a que este ingresa por los amplificadores U1A y U1B pero la corriente que atraviesa el fotodiodo circula en un solo sentido. El fotodiodo BPW21R es un sensor para luz visible capaz de sensor entre 0.01 luxes y 100000 luxes. La variación de corriente que entrega el BPW21R con cada lux que cambia es de 9nA/lux. Esto quiere decir que el fotodiodo entregará una variación de corriente comprendida entre 0.09nA y 900000nA ($\approx 1\text{mA}$). La figura 2.14 muestra la señal de salida del circuito de interfaz del sensor de iluminación simulada en Altium Designer Winter 09. La figura 2.15 muestra el PCB del circuito de la figura 2.13.

Figura 2.14. Señal simulada de salida del interfaz de iluminación.

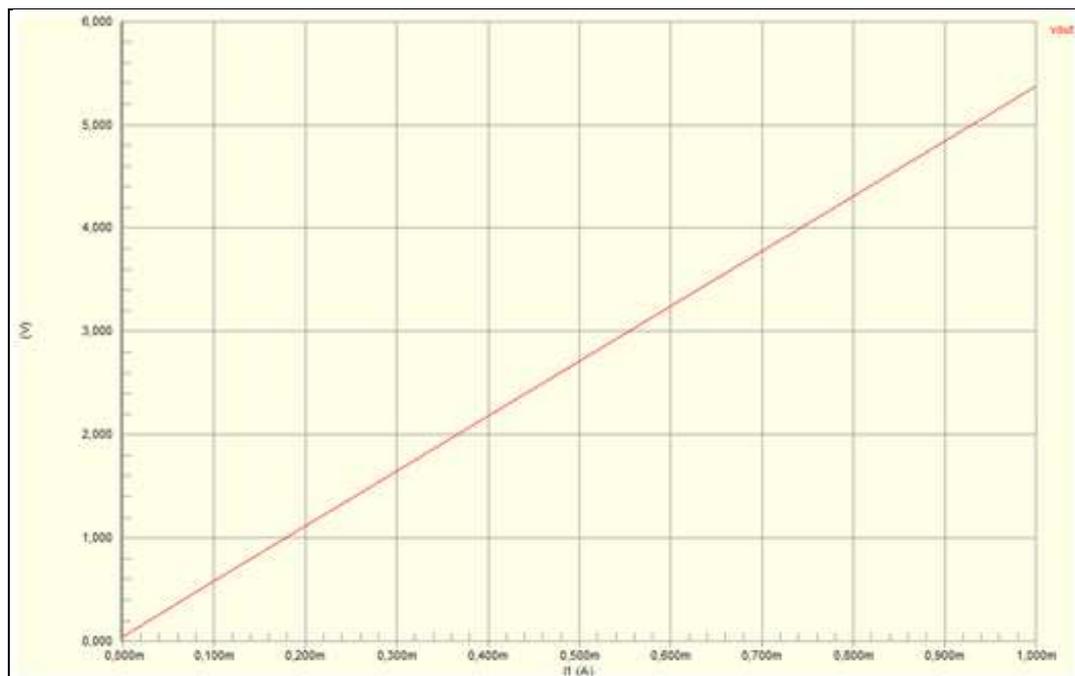
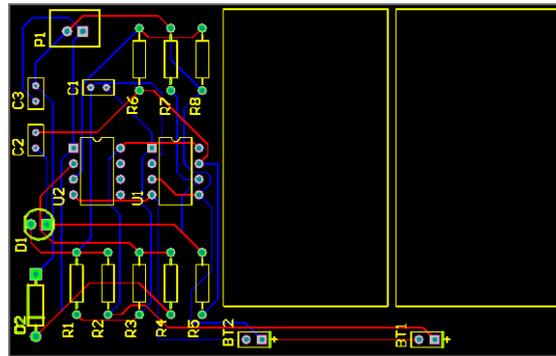


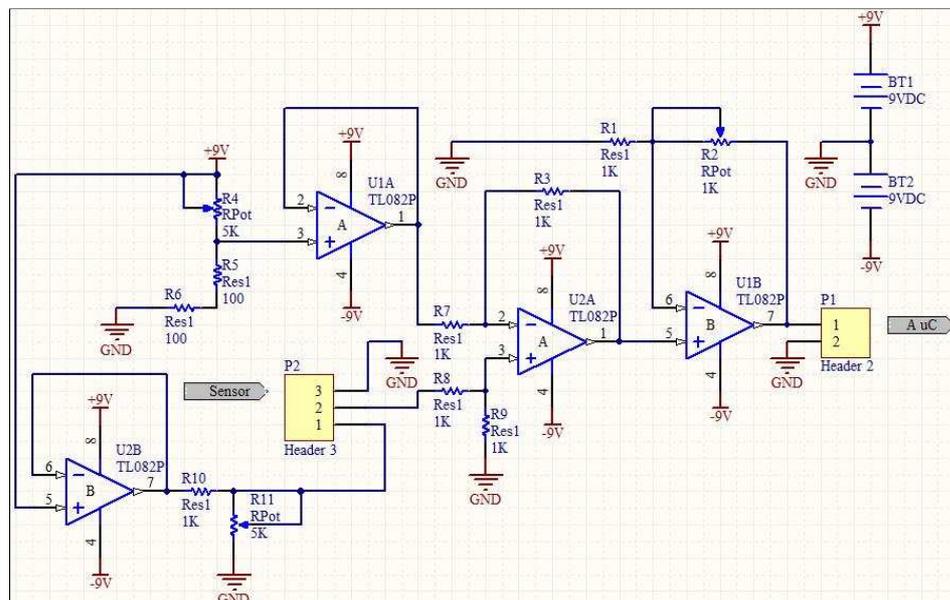
Figura 2.15. PCB del circuito de interfaz de iluminación.



2.3.3 Sensor de humedad relativa

Existen diversos tipos de sensores de humedad relativa. Entre los más usados están los de tipo capacitivo y resistivo. El sensor HIH-4000-001 es un sensor de tipo resistivo que presenta una señal de salida que varía linealmente entre 0.8V y 3.9V con una variación de humedad relativa entre 0% y 100% cuando su voltaje de alimentación es de 5V. Para acoplarlo antes del ingreso al microcontrolador se requiere restar los 0.8V y posteriormente amplificar su señal hasta que varíe en un rango entre 0V y 5V. La figura 2.16 muestra el circuito que logra acoplar su señal de esa manera.

Figura 2.16. Interfaz Sensor de humedad relativa-Microcontrolador.



La figura 2.16 muestra el circuito de acoplamiento del sensor de humedad relativa. El componente marcado como P2 representa una bornera triple en la que se alojará el sensor HIH-4000-001. El amplificador operacional marcado como U1A es un amplificador seguidor que deja pasar 0.8V obtenidos del partidor de tensión formado por los componentes R4, R5 y R6. Las señales de salida tanto del sensor como amplificador seguidor son enviadas a las entradas del amplificador U2A, el cual está conectado en configuración de restador.

A la salida de U2A se obtendrá lo que entregue el sensor restado un valor de 0.8V. El amplificador U1B se conecta en configuración de amplificador no inversor con una ganancia de 1.29 con el fin de obtener una variación entre 0V y 5V. La ganancia se determina por los componentes R1 y R2. El amplificador marcado como U2B, junto con los componentes R10 y R11, tienen como objetivo entregar 5V para la alimentación del sensor. La figura 2.17 muestra la señal de salida del circuito de acoplamiento del sensor de humedad relativa simulada en Altium Designer Winter 09. La figura 2.18 muestra el PCB del circuito de la figura 2.16.

Figura 2.17. Señal simulada de salida del interfaz de humedad relativa.

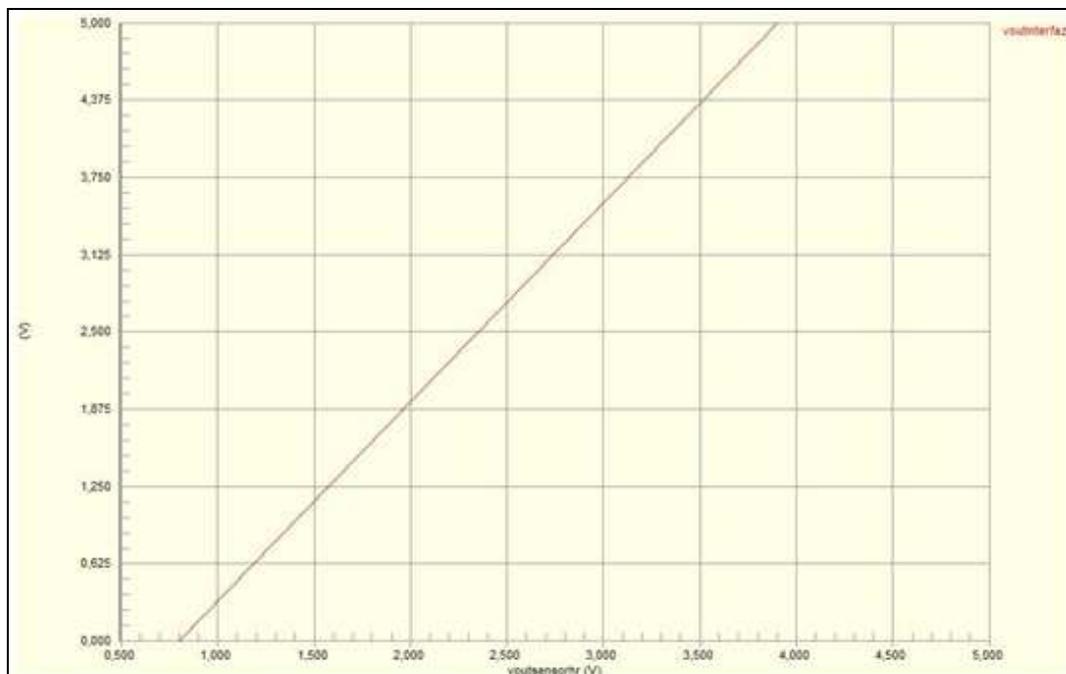
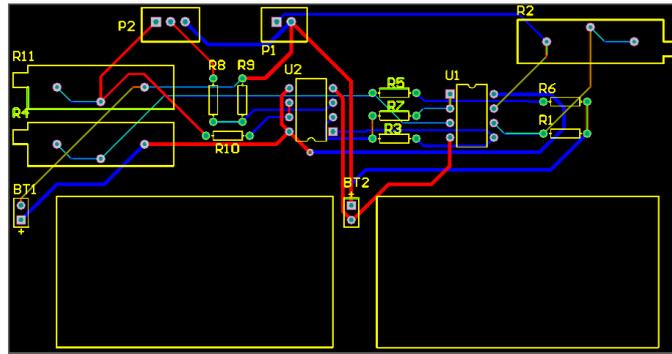


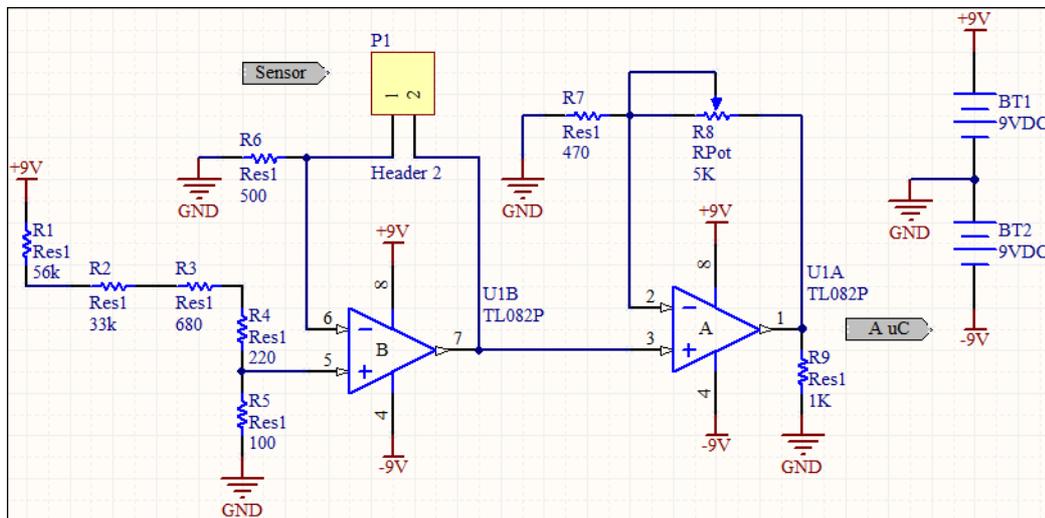
Figura 2.18. PCB del circuito de interfaz de humedad relativa.



2.3.4 Sensor de humedad del suelo

La medición de humedad del suelo es utilizada para el control de riego. La humedad del suelo se mide de acuerdo a la tensión que este presenta. La tensión es una medida que determina la fuerza con que las partículas del suelo retienen moléculas de agua. El sensor Watermarks 200 SS-X fabricado por Irrrometer Company es de bajo costo y presenta una variación de resistencia eléctrica comprendida entre 500Ω y $30k\Omega$ con una variación de tensión entre 0 y 239 centibars (Cb). La figura 2.19 muestra el circuito de acoplamiento del sensor de humedad del suelo.

Figura 2.19. Interfaz Sensor de humedad del suelo-Microcontrolador.



El componente marcado como P1 en el circuito de la figura 2.19 sirve para colocar una bornera doble en la que se alojarán los terminales del sensor de humedad del suelo. Las resistencias R1, R2, R3 y R4 sirven para formar una resistencia equivalente de 89,9k y junto con la resistencia R5 forman un partidor de tensión para obtener 10mV. Esa tensión ingresa al amplificador no inversor marcado como U1A. Al variar la resistencia del sensor variará también la ganancia del amplificador. Esta variación es lineal. El amplificador operacional marcado como U1B amplifica la señal hasta obtener una variación entre 10mV y 5V. El potenciómetro marcado como R8 se utiliza para regular la ganancia del amplificado marcado como U1A. Su valor debe establecerse en 3.38k Ω . La señal de salida del amplificador U1A está lista para ser enviada al microcontrolador.

La figura 2.20 muestra la señal de salida del circuito de acoplamiento del sensor de humedad del suelo simulada en Altium Designer Winter 09. Para realizar la simulación se reemplaza el componente marcado como P1 en el circuito de la figura 2.19 por una resistencia y se realiza un barrido sobre ella con una variación entre 500 Ω y 30k Ω con pasos de 5k Ω . La figura 2.20 muestra los diferentes valores de voltaje de salida del circuito de la figura 2.19 para los valores definidos previamente. La figura 2.21 muestra el PCB del circuito de la figura 2.19.

Figura 2.20. Señal simulada de salida del interfaz de humedad del suelo.

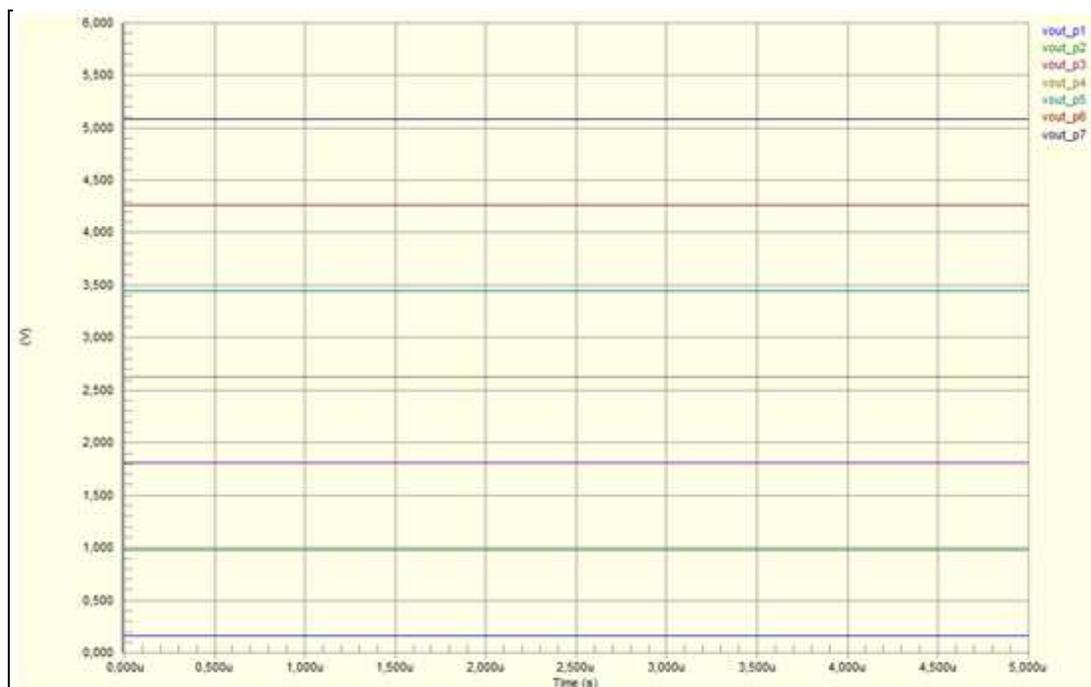
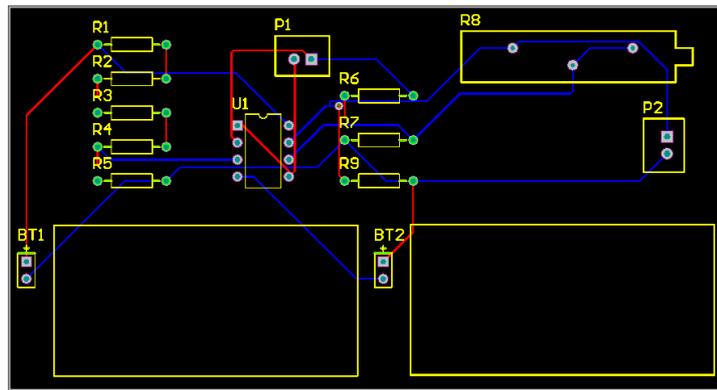


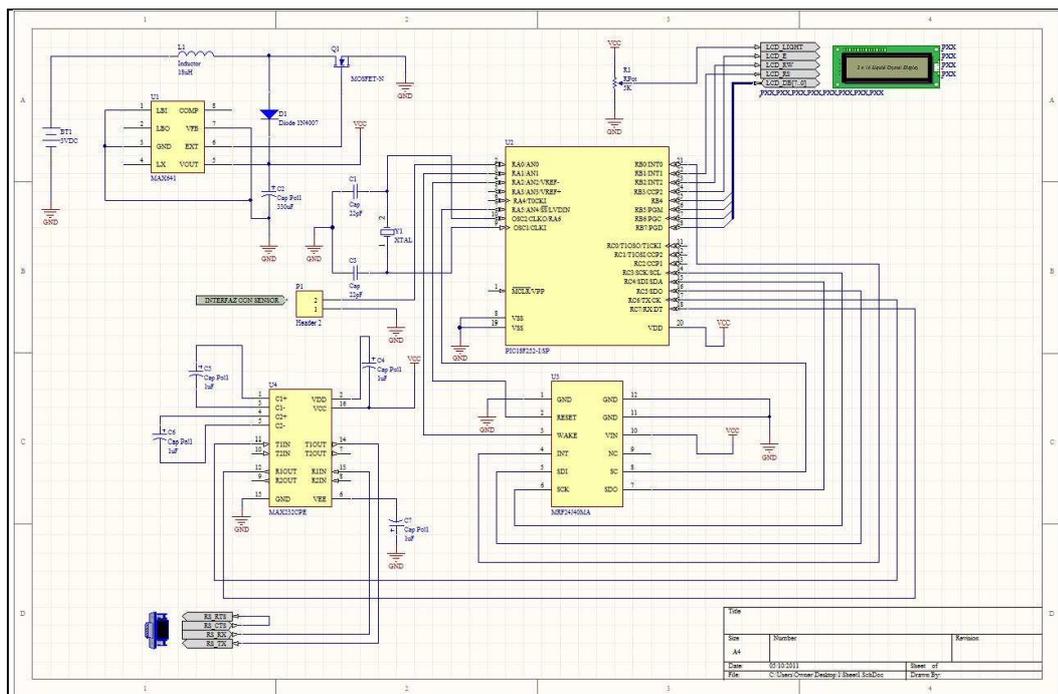
Figura 2.21. PCB del circuito de interfaz de humedad del suelo.



2.4 Hardware basado en microcontrolador para el control de un módulo ZigBee

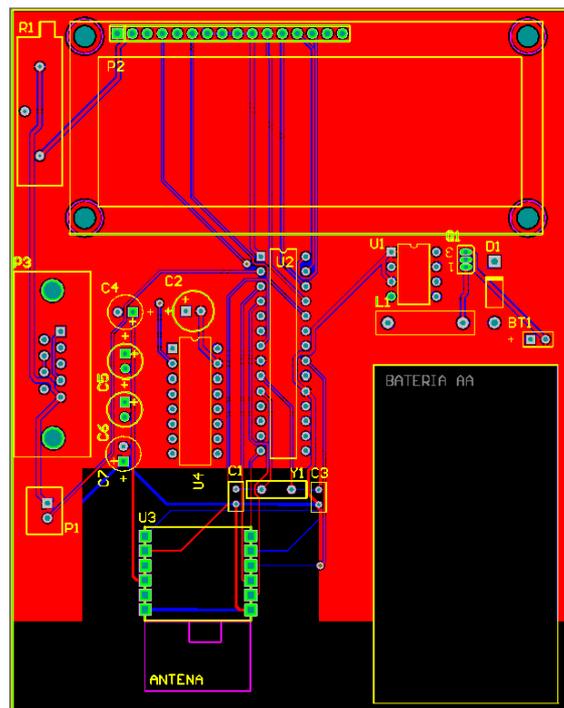
Una vez definidas todas las conexiones requeridas por el módulo ZigBee y el microcontrolador, el hardware para el control del primero quedaría como se muestra en la figura 2.22.

Figura 2.22. Hardware basado en microcontrolador para el control de un módulo ZigBee.



Cabe anotar que por motivos de optimización de recursos, ya que los distintos tipos de sensores a conectarse en el invernadero van en lugares específicos y separados uno de otro, se puede conectar un solo sensor (ya sea de temperatura, iluminación o humedad) al circuito de la figura. Es por tal motivo que los circuitos de interfaz con sensores se detallan por separado. Dichos circuitos se detallan en el apartado 2.3. La figura 2.23 muestra el PCB del circuito de la figura 2.22.

Figura 2.23. PCB del Hardware para control del módulo ZigBee.

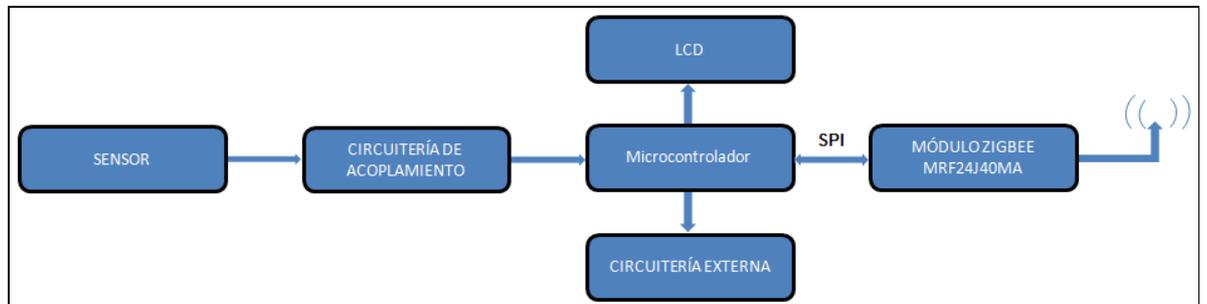


2.4.1 Diagrama de bloques

A fin de contar con una forma más comprensible del hardware para control del módulo ZigBee se establece una forma más general de representarlo. La forma de representar el hardware para el control del módulo es en diagrama de bloques. El diagrama de bloques del hardware basado en microcontrolador se muestra en la figura 2.24. En la figura 2.24 se puede observar que los sensores se conectan al microcontrolador por medio de sus circuitos de acoplamiento. También se observa que el microcontrolador y el módulo ZigBee interactúan a través del interfaz SPI.

También se observa que el microcontrolador requiere de circuitería externa para su funcionamiento. Adicionalmente se conecta un LCD para visualización de datos.

Figura 2.24. Diagrama de Bloques del Hardware para control de un módulo ZigBee.



CAPITULO 3

DISEÑO DE FIRMWARE PARA CONTROL DE UN MÓDULO ZIGBEE

En el protocolo ZigBee existen ciertas definiciones tales como perfiles, clusters, atributos y endpoints. A continuación se los describe:

Perfiles: son descripciones de los dispositivos y sus interfaces.

Atributo: es una porción de información que se puede transmitir de un dispositivo a otro. Posee un identificador único.

Cluster: es un grupo de atributos. Posee un identificador único.

Endpoint: es una porción de código que soporta uno o más clusters. Los dispositivos se comunican enviándose endpoints.

También existe la definición de primitiva. Una primitiva es soportada en un punto de acceso al servicio (SAP por sus siglas en inglés). Cada SAP sirve para que una entidad de servicio pueda acceder a un nivel de capa superior. En fin, una aplicación puede interactuar con el stack ZigBee de Microchip accediendo a sus servicios llamando funciones definidas en él por medio de primitivas. Esto es gracias a que el Stack ZigBee provisto por Microchip es escrito en lenguaje de programación C. Para configurar los módulos ZigBee hace falta cargarlos con el código fuente del stack ZigBee provisto por Microchip en su página Web para crear una aplicación personalizada. Una vez que se obtiene el stack de ZigBee provisto por Microchip se define el firmware para el control de un módulo ZigBee.

En el presente trabajo de grado se propone el diseño de una red ZigBee aplicada al monitoreo de un invernadero. Dada la aplicabilidad del mismo se requiere tan sólo de un dispositivo coordinador y varios dispositivos finales como se podrá observar en el capítulo 4. No hace falta la presencia de dispositivos enrutadores. A fin de establecer una nueva red, el dispositivo coordinador debe ser encendido primero. Cuando este haya establecido la red se puede encender los dispositivos finales, los cuales se unirán a la red.

El dispositivo coordinador solicitará el envío de datos de los dispositivos finales, los que enviarán sus lecturas de temperatura, iluminación, humedad relativa y humedad del suelo. El dispositivo coordinador recibirá esta información y la enviará por medio del interfaz serial USART por el puerto RS232. De esta forma se pueden obtener los datos en una PC para poder monitorearlos. A continuación se describe el diseño del firmware para el control de un dispositivo final y posteriormente de un dispositivo coordinador.

3.1 Diseño del firmware para el control de un dispositivo final

Una vez que se descarga e instala el stack ZigBee de Microchip se puede proseguir con el diseño del firmware. La figura 3.1 muestra la estructura principal del firmware para un dispositivo final. Hay que tener cuidado de incluir todos los archivos de cabecera. Además de los del microcontrolador hay que incluir el archivo de cabecera para ZigBee “zAPL.h” a fin de poder hacer uso del stack ZigBee provisto por Microchip.

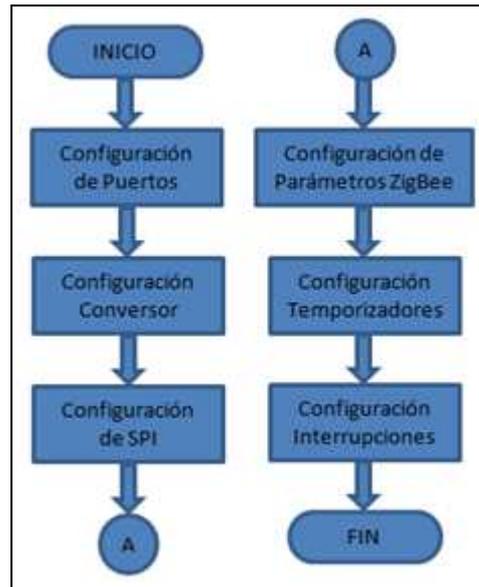
Figura 3.1. Estructura principal del firmware de un dispositivo final.



3.1.1 Configuraciones Iniciales

La figura 3.2 muestra las configuraciones necesarias en un dispositivo final.

Figura 3.2. Estructura principal de configuraciones en un dispositivo final.



3.1.1.1 Configuración Inicial de Puertos

| Pin | Tipo | Configuración Inicial | Aplicación |
|---------|-------------------|-----------------------|--|
| RA0 | Entrada Analógica | - | AN0: Entrada de Sensor |
| RA1 | Salida Digital | 0 | Envía un pulso a WAKE para despertar el módulo |
| RA2 | Salida Digital | 1 | Envía un pulso a Reset para restablecer el módulo |
| RA5 | Salida Digital | 1 | CS: Sirve para habilitar el interfaz serial del módulo |
| RB0 | Entrada Digital | 0 | Interrupción proveniente del módulo |
| RB1:RB7 | Salida Digital | 0 | LCD |
| RC3 | Salida Digital | - | SCK: Reloj del Interfaz serial |
| RC4 | Entrada Digital | - | SDO: Salida del Interfaz serial del microcontrolador |
| RC5 | Salida Digital | - | SDI: Entrada del Interfaz serial del microcontrolador |

Tabla 3.1. Configuración de puertos en un dispositivo final.

Los puertos requieren ser configurados como se muestra en la tabla 3.1. Los puertos se configuran por medio de los registros PORT, TRIS y LAT. También se

configura el registro ADCON0 a fin de establecer únicamente el pin RA0 como canal analógico y los demás como digitales.

3.1.1.2 Configuración Inicial del Conversor Analógico Digital

El conversor analógico digital se utiliza para convertir a digital el dato analógico de la medición del sensor que ingresa al microcontrolador por RA0 (AN0). El conversor es configurado en los registros ADCON0 y ADCON1. Debe ser configurado para que tenga como referencia los niveles de voltaje de VDD y VSS. También se selecciona un reloj de conversión igual a la $F_{osc}/64$. El conversor debe estar apagado inicialmente. Se establece también el formato del resultado de la conversión. En este caso justificado a la izquierda para posteriormente, cuando se necesite, tomar el dato del registro ADRESH y los dos bits más significativos (MSB) del registro ADRESL. La tabla 3.2 muestra la configuración inicial del conversor A/D.

| Conversor A/D | |
|------------------------|----------------------------|
| Reloj de Conversión | $F_{osc}/64$ |
| Voltajes de Referencia | VDD (5V) y VSS (GND) |
| Canales Analógicos | Sólo AN0 (RA0) |
| Estado Inicial | Apagado |
| Formato del Resultado | Justificado a la izquierda |

Tabla 3.2. Configuración inicial del Conversor A/D.

3.1.1.3 Configuración Inicial de Interfaz SPI

El módulo ZigBee se comunica con el microcontrolador mediante el interfaz SPI. Las configuraciones iniciales del SPI se establecen en los registros SSPCON1 y SSPSTAT. Se selecciona al microcontrolador como maestro y al módulo como esclavo. Es por eso que el microcontrolador será el que establezca el reloj para la comunicación SPI. Se establece un reloj igual $F_{osc}/4$. El módulo MRF24J40 soporta el modo de bus SPI 0,0. Por tal motivo se selecciona ese modo. El modo 0,0 quiere decir que se selecciona un estado bajo para el estado libre del reloj y el dato se

transmite en el flanco de subida. También se habilita el puerto SPI. La tabla 3.3 muestra la configuración inicial del interfaz SPI.

| SPI | |
|----------------------|------------|
| Modo de Bus SPI | 0,0 |
| Puerto serial | Habilitado |
| Modo de comunicación | Maestro |
| Reloj | Fosc/4 |

Tabla 3.3. Configuración inicial del Interfaz SPI.

3.1.1.4 Configuración Inicial de Parámetros ZigBee

ZigBee hace uso de estructuras de datos llamadas descriptores para definir características de los dispositivos. Existen descriptores en cada capa del estándar. En la configuración inicial de parámetros ZigBee se establece que no se hace uso de ningún descriptor aún. Los descriptores se encuentran en las funciones que son llamadas por las primitivas. También se establece que se hace uso de la primitiva NO_PRIMITIVE que es usada para preparar la próxima primitiva. Se puede acceder a los servicios del stack de ZigBee llamando funciones definidas en él por medio de primitivas.

Otro parámetro a establecer es la cantidad de veces en las que el dispositivo final puede realizar intentos de unirse a una red como huérfano de la red. Se establece un número de tres intentos. Cuando un dispositivo no se encuentra enlazado a ninguna red tiene la posibilidad de unirse a ella por dos métodos: orfandad y dispositivo nuevo. Un dispositivo huérfano es aquel que perdió la conexión con su dispositivo padre. El dispositivo padre es aquel al que se conectan los dispositivos finales. En este caso el dispositivo padre es el dispositivo coordinador. Un dispositivo nuevo es aquel que trata de conectarse a una nueva red. La tabla 3.4 muestra la configuración Inicial de Parámetros ZigBee.

| Parámetros ZigBee | |
|-----------------------------------|--------------|
| Primitiva Actual | NO_PRIMITIVE |
| Descriptor Actual | Ninguno |
| Intentos de conexión por orfandad | 3 |

Tabla 3.4. Configuración inicial de Parámetros ZigBee.

3.1.1.5 Configuración Inicial de Temporizadores

El conversor A/D realizará la conversión cuando interrumpa el timer0. Otros timers no son utilizados. Se configura el timer 0 en el registro T0CON. En la configuración inicial se selecciona el timer 0 como temporizador de 8 bits. Se define que la fuente del reloj es un ciclo de instrucción interno y se le asigna un prescaler de 1:256. La tabla 3.5 muestra la configuración inicial del timer 0.

| Timer 0 | |
|---------------|------------------------------|
| Estado | Habilitado |
| Configuración | Timer de 8 bits |
| Fuente | Ciclo de instrucción Interno |
| Prescaler | 1:256 |

Tabla 3.5. Configuración inicial del timer 0.

3.1.1.6 Configuración Inicial de Interrupciones

Las interrupciones se configuran en los registros INTCON. Las prioridades para la atención a interrupciones se configuran en el registro RCON. Se habilitan las interrupciones por desbordamiento del timer 0 y se establece que no habrá prioridades. En la configuración inicial se habilitan las interrupciones en el bit GIE del registro INTCON. La tabla 3.6 muestra la configuración inicial de interrupciones.

| Interrupciones | |
|----------------|---------------------------|
| Estado | Habilitado |
| Tipo | Desbordamiento de timer 0 |
| Prioridades | No |

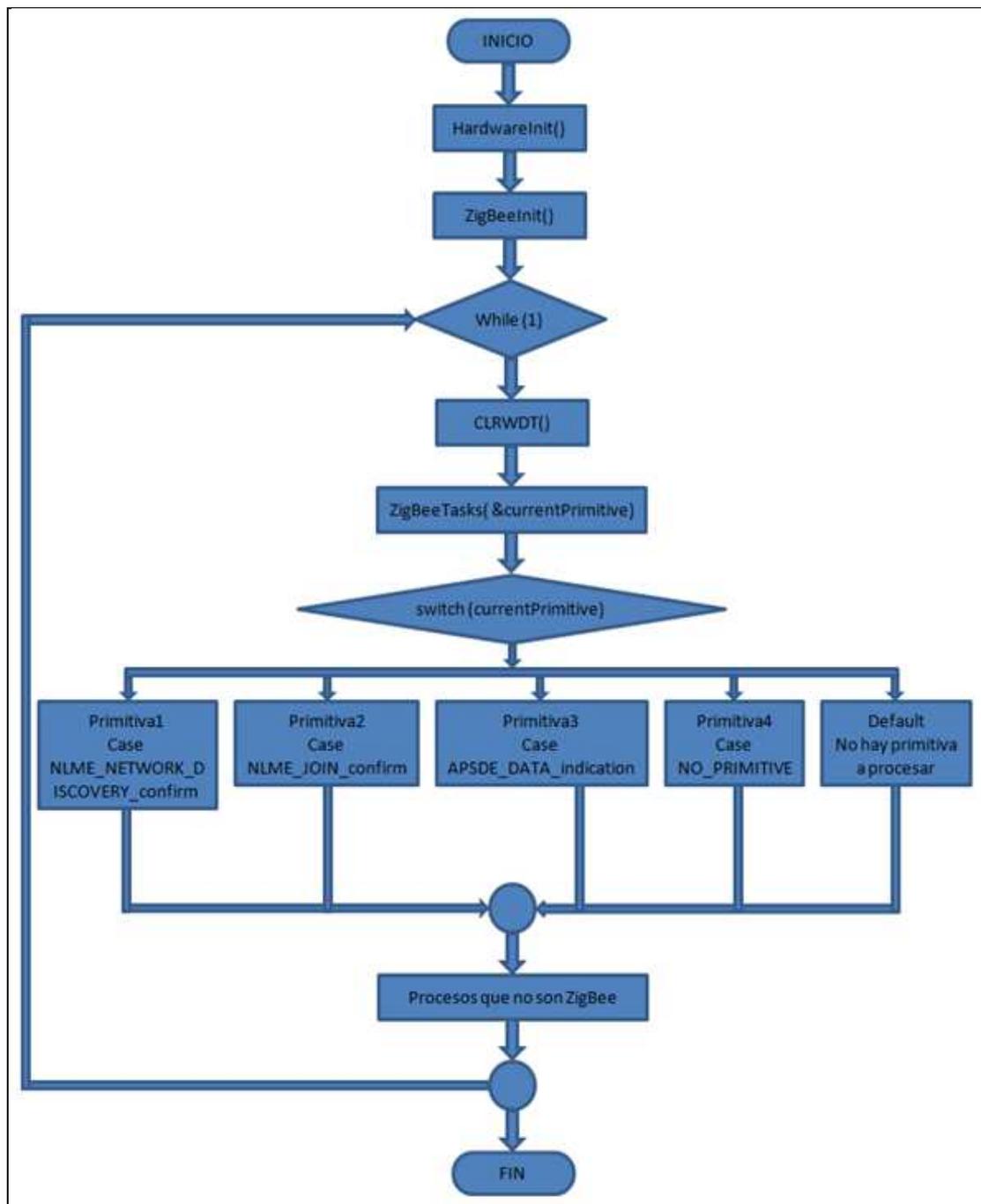
Tabla 3.6. Configuración inicial de Interrupciones.

3.1.2 Programa Principal

En el programa principal se establece todo el mecanismo de conexión a una red y el intercambio de datos entre el dispositivo coordinador y los dispositivos finales. Se accede a los servicios de ZigBee por medio del archivo ZigBeeTasks.h del stack provisto por Microchip. El archivo ZigBeeTasks.h contiene las funciones necesarias para el establecimiento de la red. Se llama a las funciones del archivo ZigBeeTasks.h por medio de primitivas. Sólo se puede ejecutar una primitiva a la vez. Por medio de las primitivas se van procesando las funciones para poder establecer la conexión a la red. Entre las primitivas existen las de petición que son aquellas que llaman funciones que permiten empezar el establecimiento de una red.

También existen primitivas de respuesta que llaman funciones que permiten indicar si se recibió o no información deseada. Además existe la primitiva NO_PRIMITIVE que sirve para preparar la próxima primitiva. La primitiva NO_RIMITIVE es utilizada al inicio del firmware en la configuración inicial de parámetros ZigBee y también antes de procesar cualquier otra primitiva. La figura 3.3 muestra la estructura del programa principal para un dispositivo final.

Figura 3.3. Estructura del programa principal de un dispositivo final.



En la figura 3.3 se puede observar que el programa principal empieza por inicializar el hardware por medio de la función `HardwareInit()`. En esta función se coloca RA1 en "1" para despertar al módulo ya que se conecta a su pin WAKE. También se coloca RA5 en "0" para habilitar el interfaz SPI ya que se conecta al pin CS del módulo. El pin RA2 se mantiene en "1" a para no generar un reset por hardware. Posteriormente se inicializa el stack de ZigBee por medio de la función `ZigBeeInit()`.

El programa ingresa en un lazo infinito para ejecutar los procesos gracias a la secuencia while (1). Dentro del lazo infinito se pone en cero el Watch Dog Timer [CLRWDT()] a fin de evitar un reset indeseado. Posteriormente se procesan las primitivas necesarias para la conexión de red. A continuación se las describe.

3.1.2.1 Primitiva 1: NLME_NETWORK_DISCOVERY_confirm

Cuando un dispositivo trata de conectarse a la red como un nodo nuevo envía la primitiva 'NLME_NETWORK_DISCOVERY_request' para buscar redes disponibles. La primitiva 'NLME_NETWORK_DISCOVERY_confirm' es utilizada para comprobar la presencia de redes disponibles. Para saber si existen redes disponibles se comprueba el parámetro que indica el estado de redes descubiertas (params.NLME_NETWORK_DISCOVERY_confirm.Status). Se comprueba este parámetro hasta que el parámetro 'params.NLME_NETWORK_DISCOVERY_confirm.NetworkCount', que es usado para contar redes, haya finalizado el conteo.

Los parámetros se encuentran en las funciones que son llamadas por las primitivas. Si los parámetros indican que sí existe al menos una red disponible, el dispositivo debe guardar el descriptor de la red que se obtiene del parámetro 'params.NLME_NETWORK_DISCOVERY_confirm.NetworkDescriptor' en la variable NetworkDescriptor y utilizarlo como descriptor actual a fin de saber las opciones que tiene para seleccionar una red y tratar de unirse a ella. Para tratar de unirse a una red el dispositivo final enviará la primitiva 'NLME_JOIN_request', en cuyos parámetros se solicita el descriptor que indica el identificador de red (PANID) y se lo debe guardar en el descriptor actual. Se solicita el identificador de red en los parámetros 'params.NLME_JOIN_request.PANId.byte.MSB' y 'params.NLME_JOIN_request.PANId.byte.LSB'.

3.1.2.2 Primitiva 2: NLME_JOIN_confirm

La primitiva 'NLME_JOIN_confirm' se utiliza para comprobar que el dispositivo final se ha unido a la red. Se comprueba el parámetro 'params.NLME_JOIN_confirm.Status'. Si este parámetro indica que el dispositivo final sí se unió a la red hay que vaciar el

contenido del descriptor actual para poder hacer uso de otros descriptores. Si el parámetro indica que no se unió a la red, el dispositivo debe comprobar si está tratando de unirse a ella como dispositivo huérfano. Para hacerlo se revisa la bandera 'ZigBeeStatus.flags.bits.bTryOrphanJoin'.

Si no se une a la red como dispositivo huérfano tiene que tratarlo otra vez como dispositivo nuevo a la misma red. Para hacerlo hay que limpiar la bandera 'ZigBeeStatus.flags.bits.bTryOrphanJoin'. Si no lo logra debe tratar de unirse a la siguiente red descubierta estableciendo en el descriptor actual que se intentará en el siguiente parámetro del descriptor. Esto se logra con la sentencia 'currentNetworkDescriptor = currentNetworkDescriptor->next'. Si no se une a ninguna red debe tratar de hacerlo de nuevo desde el principio con la sentencia goto para llevar el programa a la parte en la que se solicitan los parámetros del descriptor en la primitiva 1. Al final hay que vaciar el contenido del descriptor actual.

3.1.2.3 Primitiva 3: APSDE_DATA_indication

Esta primitiva es procesada a fin de avisar que se han recibido mensajes. Los parámetros de la primitiva contienen información del mensaje tal como el endpoint de destino por ejemplo. El mensaje se almacena en un buffer y se puede extraer todos sus bytes haciendo uso de la función APLGet(). Al procesar esta primitiva se puede saber qué endpoint se está solicitando y hacer el envío del mismo. En esta primitiva se ejecuta una sentencia para el endpoint0 (EPZDO) y otra para el endpoint que se desea transmitir.

Cada dispositivo, dependiendo del sensor que se conecte a él debe enviar un endpoint diferente. De esta forma, un dispositivo al que esté conectado un sensor de temperatura debe enviar el endpoint temperatura (EP_TEMP), si se trata de un sensor de iluminación será el endpoint iluminación (EP_ILUM), un sensor de humedad relativa será EP_HURE y si es un sensor de humedad del suelo será EP_HUSU. En la sentencia para EPZDO, el cual deber ir obligatoriamente para poder saber qué tipo de endpoint está solicitando el coordinador se realiza lo siguiente:

- Obtener el identificador del cluster (cluster id) solicitado en el parámetro 'params.APSDE_DATA_indication.ClusterId'.

- Obtener la cabecera de la trama (número de secuencia y longitud del dato) mediante la función `APLGet()` y determinando la longitud de la cabecera mediante `APL_FRAME_COUNT_MASK`.
- Comprobar el valor `END_DEVICE_BIND_rsp` del parámetro `'params.APSDE_DATA_indication.ClusterId'` a fin de determinar si se puede establecer 'binding'. Binding es una característica que implementa ZigBee a fin de hacer más manejable el descubrimiento y mantenimiento de las direcciones de red de otros dispositivos gracias a la creación de una tabla de concordancias creada por el coordinador. Si el valor de `END_DEVICE_BIND_rsp` indica binding exitoso (SUCCESS) hay que colocar el valor de la bandera `TOGGLE_BOUND_FLAG` en `'myStatusFlags.bits.bIsBound'` y respaldar ésta en `BIND_INDICATION`. Si `END_DEVICE_BIND_rsp` no indica ningún valor hay que colocar la bandera `myStatusFlags.bits.bTryingToBind` en 0.

Posteriormente se ejecuta la sentencia para el siguiente endpoint. A continuación se expondrán los procesos del siguiente endpoint. Para definir los procesos para los distintos endpoints, ya sean de temperatura, iluminación, humedad relativa o humedad del suelo, se siguen los mismos pasos con sus respectivas adaptaciones acorde al tipo de endpoint. En el presente trabajo de grado, a más del endpoint0, cada dispositivo podrá transmitir un solo endpoint dependiendo del sensor que esté conectado a él.

- Formar la cabecera de la trama (número de secuencia, comando e identificador de atributo). La trama tiene una longitud definida en `APL_FRAME_COUNT_MASK`. El identificador de atributo se define en `attributeld.byte.LSB` y en `attributeld.byte.MSB`.
- Comprobar si el identificador de cluster corresponde al cluster solicitado y el identificador de atributo es el indicado.
- Comprobar si el comando es el que indica el valor `APL_FRAME_COMMAND_GETACK`.
- Si el stack ZigBee está listo se van llenando los buffers de transmisión con los siguientes datos: tipo de trama, número de secuencia, valor `APL_FRAME_COMMAND_GET_RES`, identificador LSB de atributo, identificador MSB de atributo, valor `KVP_SUCCESS`, valor obtenido del conversor analógico digital (Respaldo del registro `ADRESH`).
- Indicar el tipo de dirección: 16 bits.

- Llenar el parámetro de solicitud de endpoint de destino con el valor del parámetro que indica el endpoint fuente. El parámetro que indica el endpoint fuente debe indicar el endpoint a enviar: EP_TEMP, EP_ILUM, EP_HURE ó EP_HUSU
- Llenar el parámetro de solicitud de dirección de destino con el valor del parámetro que indica la dirección fuente.
- Habilitar el descubrimiento de rutas.
- Seleccionar como primitiva actual 'APSDE_DATA_request'.

Al final del proceso de la primitiva se llama a la función APLDiscardRx().

3.1.2.4 Primitiva 4: NO_PRIMITIVE

Esta primitiva es utilizada antes de procesar cualquier otra primitiva. Sirve para preparar a las otras primitivas ya que en ella se definen características básicas del funcionamiento de cada dispositivo. En la primitiva NO_PRIMITIVE se realizan tests a varias banderas para establecer el comportamiento inicial de cada dispositivo. Si el dispositivo trata de unirse a la red como huérfano se establecen los siguientes parámetros:

- `params.NLME_JOIN_request.JoinAsRouter=FALSE;`
- `params.NLME_JOIN_request.RejoinNetwork=TRUE;`
- `params.NLME_JOIN_request.PowerSource=NOT_MAINS_POWERED;`
- `params.NLME_JOIN_request.RxOnWhenIdle=FALSE;`
- `params.NLME_JOIN_request.MACSecurity=FALSE;`
- `params.NLME_JOIN_request.ScanDuration=8;`
- `params.NLME_JOIN_request.ScanChannels.Val= ALLOWED_CHANNELS;`
- `currentPrimitive = NLME_JOIN_request;`

Si el dispositivo trata de unirse a la red como dispositivo nuevo se establecen los siguientes parámetros:

- `params.NLME_NETWORK_DISCOVERY_request.ScanDuration=8;`
- `params.NLME_NETWORK_DISCOVERY_request.ScanChannels.Va=ALLOWED_CHANNELS;`
- `currentPrimitive = NLME_NETWORK_DISCOVERY_request;`

Si el dispositivo ya se unió a la red se comprueba si se completó la transmisión del dato requerido por medio de la bandera 'ZigBeeStatus.flags.bits.bDataRequestComplete'. Si

ese es el caso se comprueba si se pudo establecer binding. Si no se lo pudo establecer se lo solicita de nuevo colocando el parámetro 'params.APSDE_DATA_request.ClusterId' con el valor END_DEVICE_BIND_req. Posteriormente se revisa si no hay ninguna primitiva que procesar. Si ese es el caso se solicita al dispositivo padre el envío de datos si los tiene. Para hacerlo hay que establecer como primitiva actual la primitiva 'NLME_SYNC_request'. Si el coordinador no tiene datos para el dispositivo final, éste puede ir a dormir. Se llama a la función SLEEP().

3.1.2.5 Procesos que no son ZigBee

Entre los procesos que no son ZigBee están la conversión del dato analógico en digital y la visualización de datos en el LCD que se conectó en el capítulo anterior en modo de operación de 4 bits utilizando la parte alta del puerto B. Para realizar la conversión analógica digital se hace uso de interrupciones de baja prioridad (vector 0x18). Se pregunta si interrumpe el timer 0. Cuando interrumpe el timer 0 se realiza la conversión.

En el presente trabajo de grado se convierte a digital el dato analógico obtenido de un sensor (ya sea de temperatura, iluminación, humedad relativa o humedad del suelo) en RA0 del microcontrolador. El resultado de la conversión será enviado al módulo al procesar la primitiva 'APSDE_DATA_indication'. Para realizar la conversión se realiza un test a la bandera TMR0IF. Esta bandera indica si se estableció una interrupción por desbordamiento del timer0. Si ese es el caso se prende e inicializa el conversor en el registro ADCON0. Una vez que se ha realizado la conversión se respalda el dato del registro ADRESH para poder enviarlo al módulo cuando se necesite.

3.2 Diseño del firmware para el control de un dispositivo coordinador

La estructura principal del firmware para un dispositivo coordinador se muestra en la figura 3.4. Al igual que para un dispositivo final, hay que tener cuidado de incluir

todos los archivos de cabecera. En especial el archivo “zAPL.h”. Posteriormente se describen las partes de la estructura de la figura 3.4.

Figura 3.4. Estructura principal del firmware de un dispositivo coordinador.



3.2.1 Configuraciones Iniciales

La figura 3.5 muestra las configuraciones necesarias en un dispositivo coordinador.

Figura 3.5. Estructura principal de configuraciones en un dispositivo coordinador.



3.2.1.1 Configuración Inicial de Puertos

| Pin | Tipo | Configuración Inicial | Aplicación |
|-----|-----------------|-----------------------|--|
| RA1 | Salida Digital | 0 | Envía un pulso a WAKE para despertar el módulo |
| RA2 | Salida Digital | 1 | Envía un pulso a Reset para restablecer el módulo |
| RA5 | Salida Digital | 1 | CS: Sirve para habilitar el interfaz serial del módulo |
| RB0 | Entrada Digital | 0 | Interrupción proveniente del módulo |
| RC3 | Salida Digital | - | SCK: Reloj del Interfaz serial |
| RC4 | Entrada Digital | - | SDO: Salida del Interfaz serial del microcontrolador |
| RC5 | Salida Digital | - | SDI: Entrada del Interfaz serial del microcontrolador |
| RC6 | Salida Digital | - | RS232: Salida de datos del módulo USART |
| RC7 | Entrada Digital | - | RS232: Entrada de datos del módulo USART |

Tabla 3.7. Estructura principal de configuraciones en un dispositivo coordinador.

Los puertos requieren ser configurados como se muestra en la tabla 3.7. Se configuran por medio de los registros PORT, TRIS y LAT. También se configura el registro ADCON0 a fin de establecer todos los puertos como digitales. Además se establece el bit SPEN del registro RCSTA en "1" para habilitar el puerto serial y poder hacer uso del módulo USART para enviar los datos a una PC.

3.2.1.2 Configuración Inicial de Interfaz SPI

Las configuraciones iniciales para el SPI son las mismas que para un dispositivo final. Se las establecen en los registros SSPCON1 y SSPSTAT. Se selecciona al microcontrolador como maestro y al módulo como esclavo, reloj igual $F_{osc}/4$ y de bus SPI 0,0. La tabla 3.8 muestra la configuración inicial del interfaz SPI.

| SPI | |
|----------------------|-------------|
| Modo de Bus SPI | 0,0 |
| Puerto serial | Habilitado |
| Modo de comunicación | Maestro |
| Reloj | $F_{osc}/4$ |

Tabla 3.8. Configuración inicial del Interfaz SPI.

3.2.1.3 Configuración Inicial de Parámetros ZigBee

En un dispositivo coordinador, las configuraciones iniciales de parámetros ZigBee tan solo definen como primitiva actual a la primitiva NO_PRIMITIVE. Dado que es el dispositivo que creará la red no hace falta establecer ninguna otra configuración inicial de este tipo. La tabla 3.9 muestra la configuración inicial de parámetros ZigBee para un dispositivo coordinador.

| Parámetros ZigBee | |
|-------------------|--------------|
| Primitiva Actual | NO_PRIMITIVE |

Tabla 3.9. Configuración inicial de Parámetros ZigBee.

3.2.1.4 Configuración Inicial de Interrupciones

Las interrupciones se configuran en los registros INTCON. Las prioridades para la atención a interrupciones se configuran en el registro RCON. Se habilitan las interrupciones por eventos externos ocurridos en INT0 (RB0) que establecerá una interrupción del módulo. Se establece que no habrá prioridades. En la configuración inicial se habilitan las interrupciones por eventos externos provenientes en INT0 en el bit INTOIE del registro INTCON. La tabla 3.10 muestra la configuración inicial de interrupciones.

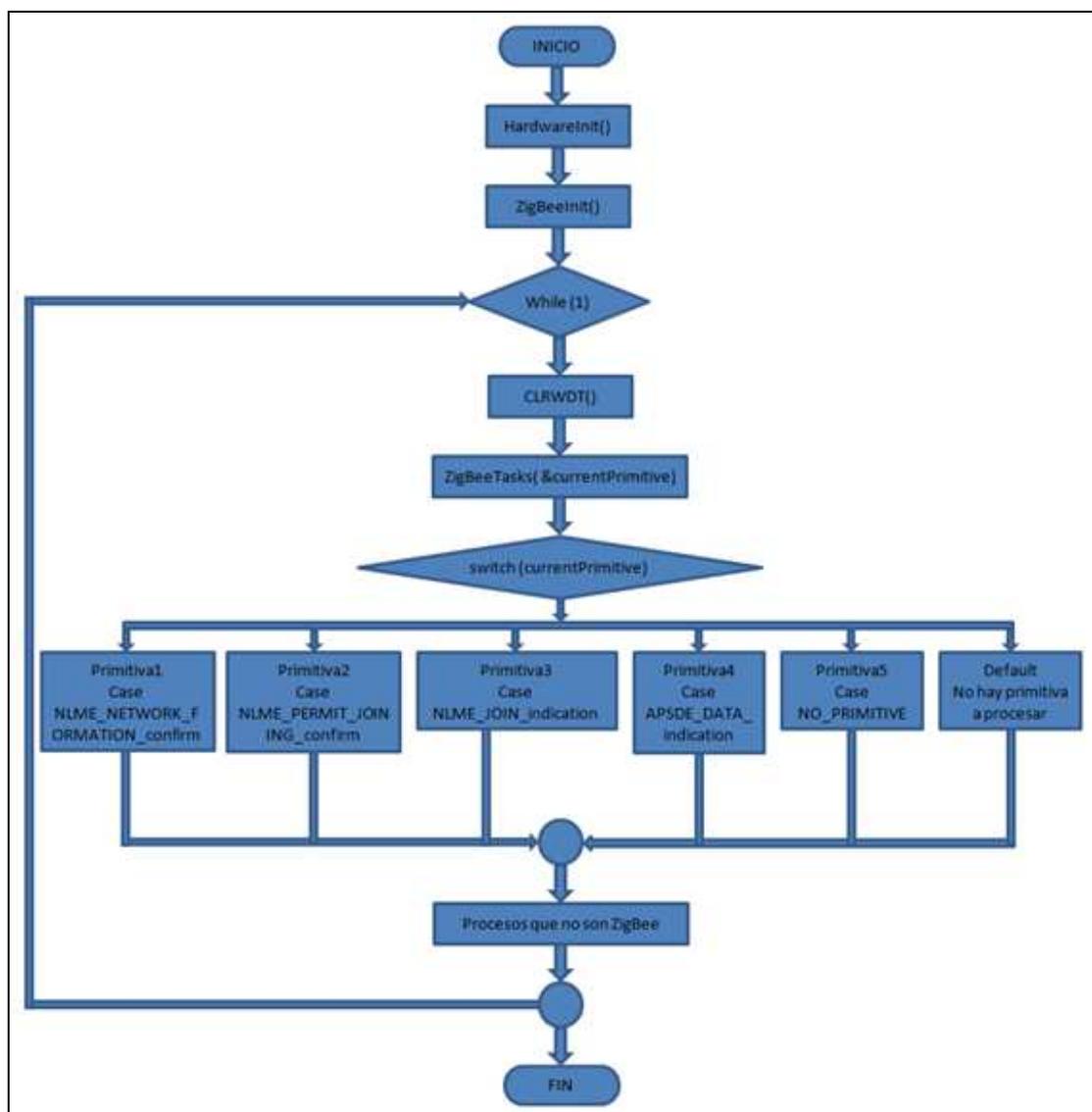
| Interrupciones | |
|----------------|--------------------------------|
| Estado | Habilitado |
| Tipo | Eventos externos en INT0 (RB0) |
| Prioridades | No |

Tabla 3.10. Configuración inicial de Interrupciones.

3.2.2 Programa Principal

En el programa principal de un dispositivo coordinador se realizan las tareas necesarias para la formación y mantenimiento de la red. Al igual que en un dispositivo final, en un dispositivo coordinador se realizan las tareas llamando a las funciones definidas en ZigBeeTasks.h que es provisto por Microchip en su stack ZigBee. La figura 3.6 muestra la estructura del programa principal para un dispositivo coordinador.

Figura 3.6. Estructura del programa principal de un dispositivo coordinador.



En la figura 3.6 se puede observar que el programa principal, al igual que para un dispositivo final, empieza por inicializar el hardware por medio de la función `HardwareInit()`. En esta función se coloca RA1 en "1" para despertar al módulo por medio de su pin WAKE. También se coloca RA5 en "0" para habilitar el interfaz SPI. El pin RA2 se mantiene en "1" a fin de evitar un reset. Posteriormente se inicializa el stack de ZigBee por medio de la función `ZigBeelnit()`. El programa ingresa en un lazo infinito para ejecutar los procesos gracias a la secuencia `while (1)`. Dentro del lazo infinito se pone en cero el Watch Dog Timer [`CLRWDT()`] a fin de evitar un reset indeseado. Posteriormente se procesan las primitivas necesarias para el establecimiento y mantenimiento de la red. A continuación se las describe.

3.2.2.1 Primitiva 1: NLME_NETWORK_FORMATION_confirm

Esta primitiva es procesada a fin de formar una nueva red. En ella se realiza un test al parámetro `'params.NLME_NETWORK_FORMATION_confirm.Status'` para verificar si comenzó el establecimiento de la red. Si ese es el caso se define el tiempo que poseen los otros dispositivos para unirse a ella. Este tiempo se define en el parámetro `'params.NLME_PERMIT_JOINING_request.PermitDuration'`. Se establece el valor de este parámetro en `0xFF`. Posteriormente se establece como primitiva actual la primitiva `'NLME_PERMIT_JOINING_request'`. Si aún no comienza el establecimiento de la red se establece como primitiva actual la primitiva `NO_PRIMITIVE`.

3.2.2.2 Primitiva 2: NLME_PERMIT_JOINING_confirm

En esta primitiva se comprueba si un dispositivo tiene permiso o no para unirse a la red haciendo uso del parámetro `'params.NLME_PERMIT_JOINING_confirm.Status'`. En cualquier caso se establece como primitiva actual la primitiva `NO_PRIMITIVE`.

3.2.2.3 Primitiva 3: NLME_JOIN_indication

Esta primitiva es procesada a fin de indicar si un dispositivo se unió a la red. Si es así se puede obtener la dirección del dispositivo que se une a la red en los parámetros `'params.NLME_JOIN_indication.ShortAddress.byte.MSB'` y `'params.NLME_JOIN_indication.ShortAddress.byte.LSB'`. Al final se establece como primitiva actual la primitiva NO_PRIMITIVE.

3.2.2.4 Primitiva 4: APSDE_DATA_indication

En esta primitiva se realizan los procesos para los distintos endpoints. Se debe incluir una secuencia de sentencias para cada uno de los endpoints (incluye el endpoint0). Los procesos a realizar comprueban el valor de la expresión indicada en el parámetro `'params.APSDE_DATA_indication.DstEndpoint'` [switch (parámetro)]. Las secuencias de sentencias para procesar los diferentes endpoints se exponen a continuación.

3.2.2.4.1 Sentencias para endpoint0 (EPZDO)

Primero hay que comprobar si la cabecera de la trama es del tipo adecuado. Se comprueba con el valor `APL_FRAME_TYPE_MSG`. Si es así se obtiene la cabecera de la trama (número de secuencia y longitud de trama). Posteriormente se comprueba el parámetro `'params.APSDE_DATA_indication.ClusterId'` con los valores `NWK_ADDR_rsp` y `END_DEVICE_BIND_rsp`. En el primer caso para obtener la dirección de red en `'destinationAddress.byte.LSB'` y `'destinationAddress.byte.MSB'`. En el segundo caso para determinar si se puede establecer binding. Si se obtiene la dirección de red hay que poner la bandera `'myStatusFlags.bits.bDestinationAddressKnown'` en "1". Si no se puede establecer binding hay que poner la bandera `'myStatusFlags.bits.bTryingToBind'` en 0. Hay que hacer esto para cada trama de EPZDO.

3.2.2.4.2 Sentencias para endpoint1 (EPTEMP)

Primero hay que comprobar si la cabecera de la trama es del tipo adecuado. Se comprueba con el valor `APL_FRAME_TYPE_KVP`. Si es así se obtiene la cabecera de la trama (número de secuencia, comando, LSB del identificador de atributo, MSB del identificador de atributo y código de error). Si el código de error indica que no existe error se comprueba si el identificador de cluster y el identificador de atributo corresponden a temperatura. Si es así se obtiene el dato por medio de la función `APLGert()` si comando es igual a `APL_FRAME_COMMAND_GET_RES`. El dato obtenido será el que envíe el dispositivo final con el endpoint indicado.

3.2.2.4.3 Sentencias para endpoint2 (EPILUM)

Estas secuencias se tratan de la misma manera que para el endpoint1. La diferencia está en la comprobación del identificador de cluster y del identificador de atributo. Se comprueba que correspondan a iluminación.

3.2.2.4.4 Sentencias para endpoint3 (EPHURE)

Al igual que las sentencias para el endpoint2, estas sentencias se tratan igual que para el endpoint1. La diferencia está en comprobar que el identificador de cluster y el identificador de atributo sean los de humedad relativa.

3.2.2.4.5 Sentencias para endpoint4 (EPHUSU)

La diferencia con las sentencias anteriores está en que se comprueba que los identificadores de cluster y de atributo correspondan a los de humedad del suelo.

3.2.2.5 Primitiva 5: NO_PRIMITIVE

En esta primitiva, si la red aún no se ha formado completamente y el coordinador está tratando de formarla, se establecen los siguientes valores para los distintos parámetros:

- params.NLME_NETWORK_FORMATION_request.ScanDuration= 8
- params.NLME_NETWORK_FORMATION_request.ScanChannels.Val= ALLOWED_CHANNELS
- params.NLME_NETWORK_FORMATION_request.PANId.Val=0xFFFF
- params.NLME_NETWORK_FORMATION_request.BeaconOrder=MAC_PIB_macBeaconOrder
- params.NLME_NETWORK_FORMATION_request.SuperframeOrder=MAC_PIB_macSuperframeOrder
- params.NLME_NETWORK_FORMATION_request.BatteryLifeExtension=MAC_PIB_macBattLifeExt
- currentPrimitive = NLME_NETWORK_FORMATION_request

Si la red ya se formó se establecen todos los procesos que pueden enviar mensajes. Entre ellos los que sirven para indicar a los dispositivos finales qué endpoints se pudieran solicitar. En estos procesos se establecen los siguientes parámetros con los siguientes valores.

Para el caso de temperatura:

- params.APSDE_DATA_request.DstAddrMode=APS_ADDRESS_16_BIT
- params.APSDE_DATA_request.DstEndpoint=EP_TEMP_RFD
- 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.SrcEndpoint=EP_TEMP
- params.APSDE_DATA_request.ClusterId=Temp_CLUSTER
- currentPrimitive = APSDE_DATA_request

Para otros endpoints se reemplaza “temp” por el endpoint indicado. Siempre se establece como primitiva actual la primitiva APSDE_DATA_request al final.

3.2.2.6 Procesos que no son ZigBee

En los procesos que no son ZigBee se encuentra el envío de datos por el módulo USART del microcontrolador a fin de poder realizar la lectura de los datos obtenidos

de los sensores en algún dispositivo tal como una PC por ejemplo. Se obtienen los datos del módulo leyendo el registro de interrupciones del mismo. Si el registro indica que existe una interrupción por dato recibido se deshabilitan momentáneamente las interrupciones del microcontrolador y se van recibiendo los datos en el registro SSPBUF por medio del interfaz serial SPI. Se mueven los datos recibidos al banco 2 de la RAM del microcontrolador y posteriormente se los mueve al registro TXREG y se habilita la transmisión usando el módulo USART en el bit TXEN del registro TXSTA. Se habilitan nuevamente las interrupciones.

CAPITULO 4

DISEÑO DE RED ZIGBEE DE SENSORES INALÁMBRICOS

El presente capítulo presenta el diseño de red de sensores inalámbricos tomando como referencia el invernadero de 1200m² perteneciente a “Productos Agropecuarios Don Luchito”. El invernadero se encuentra ubicado en la parroquia Solano del Cantón Déleg en la provincia de Cañar. La figura 4.1 muestra una fotografía del invernadero en cuestión.

Figura 4.1. Invernadero de “Productos Agropecuarios Don Luchito”.

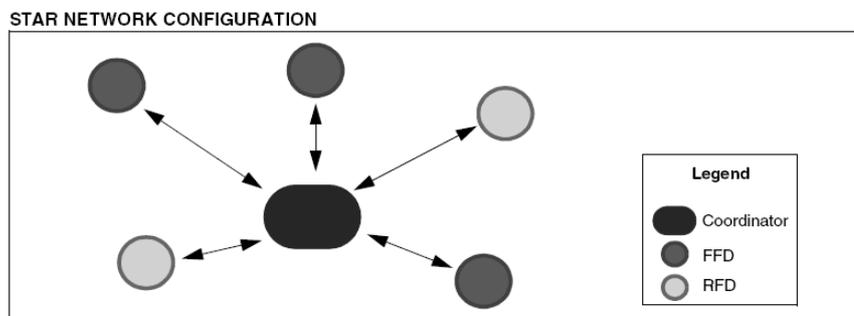


4.1 Topología de Red

En ZigBee existen 3 diferentes topologías de red: red en estrella, grupos de árbol y topología en malla. En el presente trabajo de grado se considera la utilización de

topología de red en estrella. En esta topología existe un coordinador y varios dispositivos finales que se comunican con el coordinador. Los dispositivos finales pueden ser dispositivos de funciones completas (FFD por sus siglas en inglés) o dispositivos de funciones reducidas (RFD por sus siglas en inglés). Un dispositivo FFD se diferencia de un RFD en que este último ofrece una cantidad limitada de servicios con respecto al primero. En el presente diseño no hace falta la utilización de dispositivos finales FFD. Cada uno de los dispositivos finales puede establecerse en modo dormir (Sleep) si sus procesos no son requeridos. La figura 4.2 muestra la topología de red en estrella.

Figura 4.2. Topología de red en estrella.



Fuente: Nota de Aplicación AN965. Microchip INC.

Dado que el módulo MRF24J40MA posee un alcance de hasta 400pies (121,92m) según se especifica en su hoja de datos, y en el invernadero no se necesita de un alcance más allá de 45m, la topología en estrella es más que suficiente. Otras topologías son utilizadas a fin de expandir la red o por requerimientos propios de cada diseño.

4.2 Elementos que conforman la red

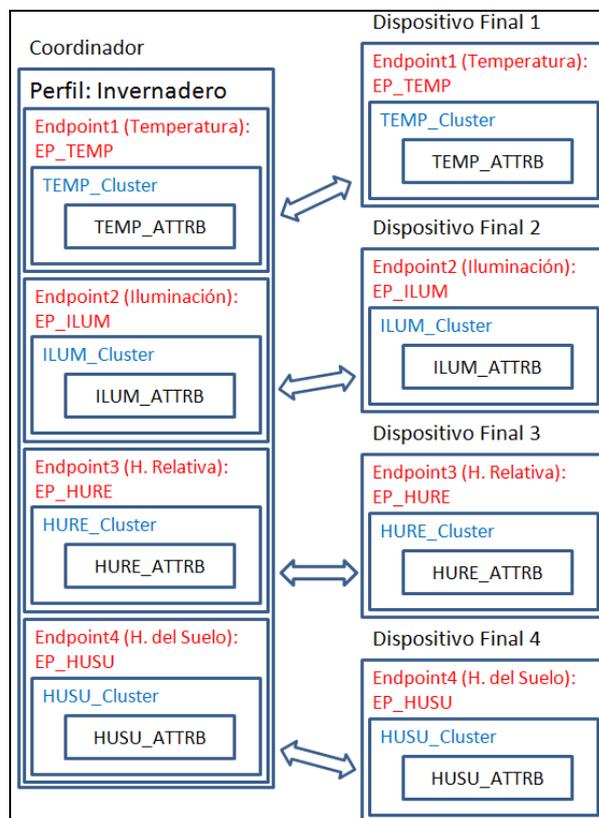
La red se conforma de varios elementos. Primero se consideran los dispositivos finales. Existirán sensores de diverso tipo: temperatura, iluminación, humedad relativa y humedad del suelo. Cada uno de estos sensores se conecta a un microcontrolador y éste, a su vez, se conecta al módulo ZigBee para realizar el envío de información de manera inalámbrica. El dispositivo coordinador no tiene

conexión con ningún sensor puesto que es el dispositivo al que llegarán los datos obtenidos de los sensores. El dispositivo coordinador puede comunicarse con una PC por medio del módulo USART del microcontrolador.

4.3 Perfil

Para poder realizar la comunicación entre dispositivos haciendo uso del firmware diseñado en el capítulo anterior hay que usar un perfil. Dado que el presente trabajo de grado es realizado con fines investigativos y no comerciales se utiliza un perfil no certificado. Para fines comerciales hay que ser miembro de la Alianza ZigBee, la cual certifica los perfiles para poder comercializarlos. Los perfiles definen los valores de los identificadores de cluster e identificadores de atributo y también el formato de cada atributo. Los perfiles, una vez creados y certificados por la Alianza ZigBee, pueden ser utilizados por la aplicación. La figura 4.3 muestra la estructura básica de un perfil diseñado para la presente aplicación.

Figura 4.3. Perfil no certificado Invernadero.



La figura 4.3 muestra la creación del perfil en el caso de que exista un solo sensor de cada tipo enviando datos al coordinador. Cabe recalcar que existen varios atributos asociados a cada cluster y varios clusters agrupados en lo que se llama Librería de Clusters ZigBee (ZCL por sus siglas en inglés). Por ejemplo, un cluster perteneciente a ZCL puede ser el cluster 'Basic' con un ID igual a 0x0000 (los ID Clusters son de 16 bits) y poseer atributos tales como la versión del ZCL, versión de la aplicación, versión del stack, etc. Otro cluster puede ser TEMP_Cluster con ID igual a 0x0100 y poseer los atributos TEMP_ATTRB1 (que puede ser el valor medido por el sensor de temperatura 1), tolerancia, valor mínimo y valor máximo. Los perfiles y ZCL pueden ser comprados a Microchip y editados para obtener la funcionalidad deseada.

Dada la complejidad de la creación de las librerías de clusters y el hecho de que no forma parte de los objetivos del presente trabajo de grado, tan sólo se considera la necesidad de la creación de los atributos y clusters indispensables para la operatividad de los sensores y se asume que se los crea a partir de la edición del perfil de medición de temperatura de Microchip. De tal manera, se considera, por ejemplo, la creación del atributo ILUM_ATTRB2 que es el valor medido por el sensor de iluminación 2, que pertenece al cluster ILUM_Cluster y se envía al dispositivo coordinador haciendo uso del endpoint EP_ILUM.

En el capítulo 3 se realizó el diseño del firmware para el control de un módulo ZigBee. En dicho diseño se determinó que ciertos parámetros deben ser establecidos en uno u otro valor. Tales parámetros son atributos. Un conjunto de atributos agrupados por ciertas características comunes se llama cluster. Los atributos y clusters son enviados y recibidos por medio de bloques de códigos llamados endpoints. En el presente capítulo se establece qué atributo necesario para la operatividad del sensor se crea, además el cluster al que pertenece y con qué endpoint se transmite.

4.4 Direccionamiento

Por otro lado, en el presente diseño se utiliza más de un sensor del mismo tipo enviando datos al coordinador. Cada dispositivo debe poseer una dirección MAC de 64 bits. Esta dirección se forma con 24 bits asignados por el Identificador Único Organizacional (OUI por sus siglas en inglés). Cada OUI se compra a IEEE para

garantizar la unicidad del dispositivo. Los 40 bits restantes son asignados por el desarrollador.

Dado que el presente trabajo de grado es realizado con fines investigativos se supone el uso de un OUI igual a 00-04-A3, el cual es el OUI de Microchip. Los dispositivos usarán la dirección MAC para comunicarse mientras están en el proceso de unirse a la red. Cuando la red ya está formada cada dispositivo será asignado con una dirección de red de 16 bits para comunicarse entre ellos. Las direcciones de red son asignadas por el coordinador por medio de los clusters pertenecientes a la capa de red (NWK) definidos en el stack ZigBee de Microchip.

4.5 Red ZigBee en el Invernadero

El presente trabajo de grado toma como referencia el invernadero de aproximadamente 1200m² ubicado en el sector de Solano. La disposición de los sensores (dispositivos finales) y dispositivo coordinador se muestra en el plano de la figura 4.4. Se observa que se plantea el uso de 3 sensores de temperatura, 2 de humedad relativa, 4 de iluminación y 10 de humedad del suelo. Se observa también que la distancia entre el dispositivo final más lejano y el coordinador es menor a 45m. Por tal motivo es más que suficiente el planteamiento de la topología de red en estrella. Las figuras 4.5 y 4.6 muestran la simbología utilizada en el plano de la figura 4.4.

Figura 4.4. Red ZigBee en el Invernadero de Solano.

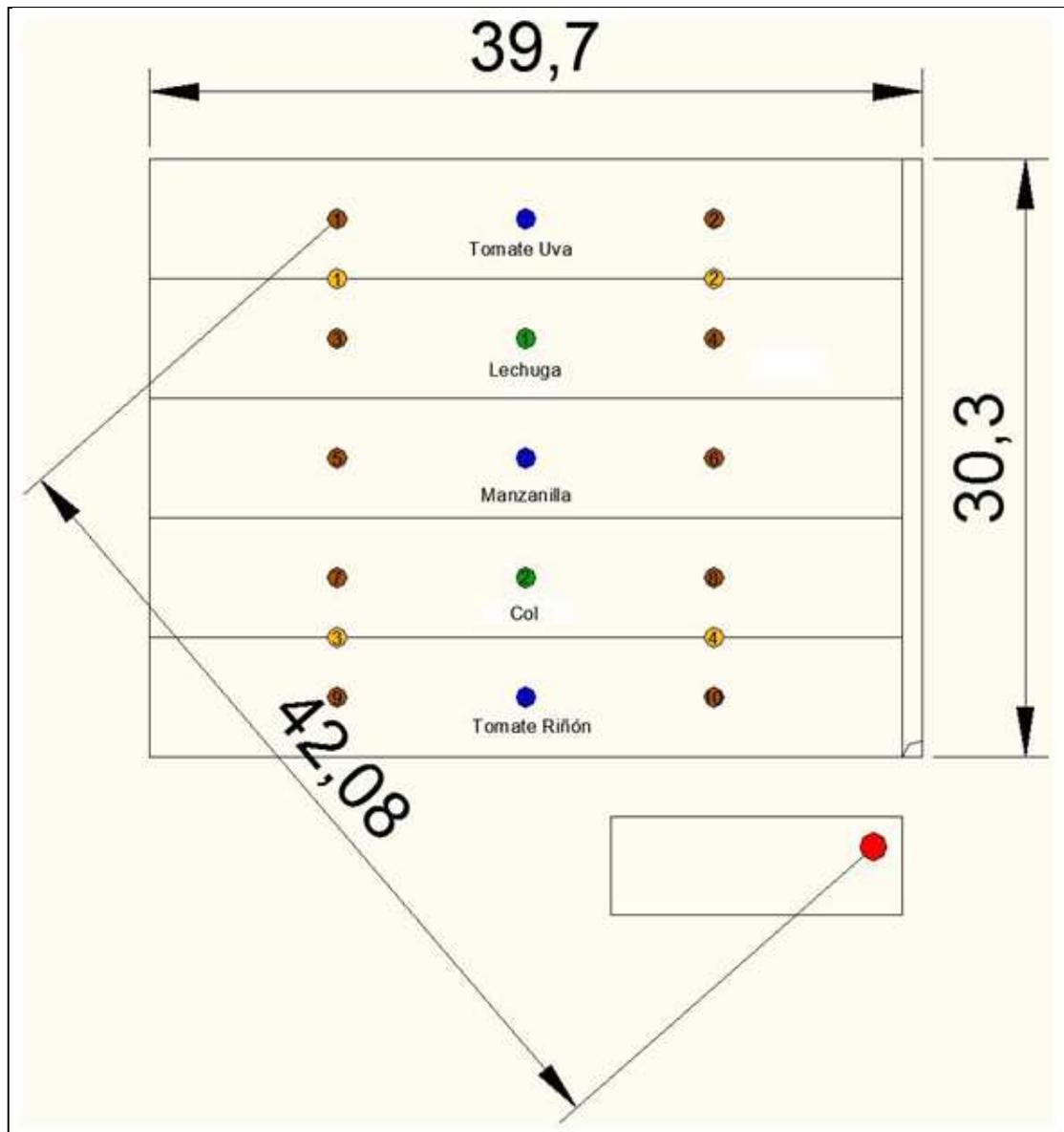


Figura 4.5. Simbología del Plano de la figura 4.4.

| Simbología Sensores | |
|---------------------------------------|-----------------|
| ● | S. Temperatura |
| ● | S. Humedad Rel. |
| ● | S. Iluminación |
| ● | S. Humedad Sue. |

Figura 4.6. Simbología del Plano de la figura 4.4.

| Simbología Dispositivos | |
|---|--------------------|
|  | D. Final RFD |
|  | D. Coordinador FFD |

La tabla 4.1 muestra las características más importantes de cada dispositivo en el diseño de la red ZigBee del invernadero. En ella se puede observar las direcciones MAC de cada dispositivo, el atributo que indica la medida de cada sensor, el cluster al que pertenece y el endpoint por el que se envían los datos. Los números de dispositivos indican los que se muestran en el plano de la figura 4.4.

| Dispositivo | No. Dispositivo | Tipo de Dispositivo | End Point | Cluster (Nombre) | Cluster ID (Hex) | Atributo | MAC Address (Hex) |
|---------------|-----------------|---------------------|----------------|------------------|------------------|----------------|-------------------------|
| Temperatura | 1 | RFD | EP_TEMP | TEMP_Cluster | 0x0100 | TEMP_ATTRB1 | 00.04.A3.01.01.01.01.01 |
| | 2 | RFD | EP_TEMP | TEMP_Cluster | 0x0100 | TEMP_ATTRB2 | 00.04.A3.01.01.01.01.02 |
| | 3 | RFD | EP_TEMP | TEMP_Cluster | 0x0100 | TEMP_ATTRB3 | 00.04.A3.01.01.01.01.03 |
| Hum. Relativa | 1 | RFD | EP_HURE | HURE_Cluster | 0x0200 | HURE_ATTRB1 | 00.04.A3.01.01.01.01.04 |
| | 2 | RFD | EP_HURE | HURE_Cluster | 0x0200 | HURE_ATTRB2 | 00.04.A3.01.01.01.01.05 |
| Iluminación | 1 | RFD | EP_ILUM | ILUM_Cluster | 0x0300 | ILUM_ATTRB1 | 00.04.A3.01.01.01.01.06 |
| | 2 | RFD | EP_ILUM | ILUM_Cluster | 0x0300 | ILUM_ATTRB2 | 00.04.A3.01.01.01.01.07 |
| | 3 | RFD | EP_ILUM | ILUM_Cluster | 0x0300 | ILUM_ATTRB3 | 00.04.A3.01.01.01.01.08 |
| | 4 | RFD | EP_ILUM | ILUM_Cluster | 0x0300 | ILUM_ATTRB4 | 00.04.A3.01.01.01.01.09 |
| Hum. Suelo | 1 | RFD | EP_HUSU | HUSU_Cluster | 0x0400 | HUSU_ATTRB1 | 00.04.A3.01.01.01.01.0A |
| | 2 | RFD | EP_HUSU | HUSU_Cluster | 0x0400 | HUSU_ATTRB2 | 00.04.A3.01.01.01.01.0B |
| | 3 | RFD | EP_HUSU | HUSU_Cluster | 0x0400 | HUSU_ATTRB3 | 00.04.A3.01.01.01.01.0C |
| | 4 | RFD | EP_HUSU | HUSU_Cluster | 0x0400 | HUSU_ATTRB4 | 00.04.A3.01.01.01.01.0D |
| | 5 | RFD | EP_HUSU | HUSU_Cluster | 0x0400 | HUSU_ATTRB5 | 00.04.A3.01.01.01.01.0E |
| | 6 | RFD | EP_HUSU | HUSU_Cluster | 0x0400 | HUSU_ATTRB6 | 00.04.A3.01.01.01.01.0F |
| | 7 | RFD | EP_HUSU | HUSU_Cluster | 0x0400 | HUSU_ATTRB7 | 00.04.A3.01.01.01.01.10 |
| | 8 | RFD | EP_HUSU | HUSU_Cluster | 0x0400 | HUSU_ATTRB8 | 00.04.A3.01.01.01.01.11 |
| | 9 | RFD | EP_HUSU | HUSU_Cluster | 0x0400 | HUSU_ATTRB9 | 00.04.A3.01.01.01.01.12 |
| | 10 | RFD | EP_HUSU | HUSU_Cluster | 0x0400 | HUSU_ATTRB10 | 00.04.A3.01.01.01.01.13 |
| Coordinador | | FFD | Solicita Todos | Solicita Todos | Solicita Todos | Solicita Todos | 00.04.A3.10.01.01.01.01 |

Tabla 4.1. Características de los dispositivos.

Los clusters ID se asumen como se muestran en la tabla 4.1 dado que en el perfil existirán muchos más clusters asociados con las diferentes capas del stack. Los atributos expuestos en la tabla 4.1 representan la lectura de cada uno de los sensores. Todos los dispositivos a los que se conectan sensores son dispositivos finales RFD. No se conecta ningún sensor al dispositivo coordinador. El dispositivo coordinador es FFD y será el encargado de pedir y recibir toda la información que le pueden proporcionar los dispositivos finales.

CONCLUSIONES

En el presente trabajo de grado se realizó el diseño de una red ZigBee aplicada al monitoreo de un invernadero proponiendo el uso de ciertas herramientas creadas por Microchip para tal propósito. Del desarrollo del presente trabajo de grado se pueden obtener las siguientes conclusiones:

- Un módulo transceptor ZigBee MRF24J40MA requiere de un microcontrolador para su control y funcionamiento.
- El módulo MRF24J40MA es compatible con los microcontroladores de la familia PIC18F fabricados por Microchip.
- El módulo MRF24J40MA y el microcontrolador se comunican entre sí por medio del interfaz SPI de cada uno.
- El microcontrolador PIC18F252 es apto para el control del módulo ya que es compatible con éste y posee un interfaz SPI. Además posee un conversor analógico digital de 10 bits de resolución, lo que es suficiente dada la aplicación.
- El stack ZigBee proporcionado por Microchip es escrito en el lenguaje de programación C y por tal motivo el firmware debe ser desarrollado en dicho lenguaje.
- Se realizan las tareas definidas en las funciones del stack de Microchip por medio de primitivas.
- Una aplicación ZigBee se desarrolla gracias a la creación de perfiles personalizados o a la edición de perfiles previamente definidos a fin de obtener la funcionalidad deseada.
- Los perfiles deben ser certificados por la Alianza ZigBee para poder realizar la comercialización de los productos desarrollados bajo el estándar.
- Los dispositivos se comunican por medio de bloques de códigos llamados endpoints.
- Los endpoints soportan uno o más clusters. Un cluster es un grupo de atributos.
- El invernadero del sector de Solano representa un ambiente ideal para tomar como referencia en el diseño de la red.
- La topología de red adecuada, dadas las características físicas del invernadero y las características propias requeridas por la red, es la topología de red en estrella.

- Dado que en el invernadero se requieren bajas tasas de transmisión y por las ventajas ofrecidas tales como costo, bajo consumo, etc., la tecnología de Microchip es adecuada para el diseño de red ZigBee del invernadero de Solano.
- Se puede realizar el monitoreo de los parámetros ambientales al interior del invernadero por medio del módulo USART del microcontrolador conectado al dispositivo coordinador.

RECOMENDACIONES

- Antes de comenzar con ZigBee usando tecnología de Microchip INC se recomienda revisar sus notas de aplicación.
- Adquirir destrezas en lenguaje C de programación.
- Adquirir conocimientos acerca de microcontroladores.
- Otras marcas, además de Microchip INC, son miembros de la Alianza ZigBee y también desarrollan tecnología para trabajar con el estándar. A fin de obtener más opciones se recomienda revisar tecnologías ofertadas por otras marcas también. MaxStream es una opción muy interesante.
- También se recomienda adquirir conocimientos acerca de estándares para comunicaciones inalámbricas desarrollados por IEEE.

BIBLIOGRAFÍA

REFERENCIAS BIBLIOGRÁFICAS

- FLOWERS David, OTTEN Kim y RAJBHARTI Nilesh, Microchip Technology Inc., 2006, Nota de Aplicación AN965. Microchip Stack for the ZigBee™ Protocol. Disponible en la Web:
<http://ww1.microchip.com/downloads/en/appnotes/00965b.pdf>
- IEEE Computer Society, 2003, IEEE Std. 802.15.4™-2003.
- IEEE Computer Society, 2006, IEEE Std. 802.15.4™-2006 (Revision of IEEE Std. 802.15.4-2003).
- LATTIBEAUDIERE Derrick P., Microchip Technology Inc., 2008, Nota de Aplicación AN1232. Microchip ZigBee-2006 Residential Stack Protocol. Disponible en la Web:
http://www.newark.com/pdfs/techarticles/microchip/AN1232A_ZigBee2006.pdf
- LATTIBEAUDIERE Derrick P., Microchip Technology Inc., 2009, Nota de Aplicación AN1255. Microchip ZigBee® PRO Feature Set Protocol Stack. Disponible en la Web:
<http://ww1.microchip.com/downloads/en/AppNotes/01255A.pdf>
- Microchip Technology Inc., 2002, PIC18Fxx2 Datasheet.
- Microchip Technology Inc., 2008, MRF24J40MA Datasheet.
- Microchip Technology Inc., 2010, MRF24J40 Datasheet.
- ZigBee Alliance Inc., 2008, ZigBee Specification Document 053474r17.
- ZigBee Alliance Inc., 2008, ZigBee-2007 Layer PICS and Stack Profiles Revision 03 Document 08006r03.

REFERENCIAS ELECTRÓNICAS

- Alianza ZigBee. Estándares ZigBee: <http://www.zigbee.org/>, 2011.
- Estándar ZigBee: <http://es.wikipedia.org/wiki/ZigBee>, 2011.
- ZigBee y Microcontroladores:
<http://decelectronics.com/html/XBEE/XBEE.htm>, 2011.
- Parámetros para cultivo adecuado de tomate uva:
<http://jardinplantas.com/cultivo-de-tomates-cherry/>, 2011.