



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Facultad de Ingeniería Eléctrica y Computación**

CONTROL REMOTO DE ROBOTS USANDO MÓDULOS DE  
RADIOFRECUENCIA XBEE A 2.4GHZ CON CAPACIDAD DE  
COMUNICACIÓN SERIAL A DATALOGGER E INTERFAZ GRÁFICA

**TESINA DE SEMINARIO**

Previo a la obtención del Título de:

**INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

Presentado por:

Carlos Andrés Carrera Cadena  
Luis Miguel Guaiña Mejía

GUAYAQUIL – ECUADOR

2010

## **AGRADECIMIENTO**

A Dios, por ser la guía de nuestro camino.

A nuestros padres, por darnos su apoyo incondicional, desinteresado, y su amor imperecedero.

## **DEDICATORIA**

A nuestros padres, pilar fundamental de nuestras vidas, presentes durante todo este tiempo de aprendizaje y formación profesional.

**TRIBUNAL DE SUSTENTACIÓN**

---

ING. CARLOS VALDIVIESO A.  
**PROF. DEL SEMINARIO DE GRADUACIÓN**

---

ING. HUGO VILLAVICENCIO V.  
**DELEGADO DEL DECANO**

## **DECLARACIÓN EXPRESA**

“La responsabilidad del contenido de esta Tesina, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”.

(Reglamento de Graduación de la ESPOL)

---

Carlos A. Carrera Cadena

---

Luis M. Guaiña Mejía

## **RESUMEN**

El presente proyecto nos introduce al aprendizaje de la manipulación de los módulos de radiofrecuencia Xbee, que sirven para infinidad de aplicaciones, en especial para aquellas de control no permanente.

Representan un campo poco difundido y aplicado, al menos en el Ecuador, lo que invita a desarrollar proyectos con ellos en variados entornos.

Este trabajo pretende la comunicación inalámbrica mediante módulos Xbee para el control a distancia de robots, además de transferencia de datos; en nuestro caso, valores de temperatura, para su posterior almacenamiento mediante comunicación serial en un datalogger. Estos datos se presentarán usando una interfaz gráfica amigable.

# ÍNDICE GENERAL

RESUMEN

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

INTRODUCCIÓN

1.	DESCRIPCIÓN GENERAL DEL PROYECTO	
1.1	Antecedentes .....	1
1.2	Descripción del Proyecto .....	2
1.3	Módulos de comunicación similares.....	3
1.3.1	Transceiver nRF2401A.....	3
1.3.2	Transceiver nRF24L01-CAL-SP .....	4
1.3.3	Transceiver CC2500.....	5
1.4	Comparación entre transceivers similares con el Xbee .....	5
2.	FUNDAMENTACIÓN TEÓRICA	
2.1	Estándar IEEE 802.15.4/ZIGBEE .....	7
2.1.1	Arquitectura .....	9
2.1.2	Topologías .....	11
2.2	Zigbee .....	14
2.2.1	Tipos de dispositivos .....	15
2.2.2	Estrategias de conexión en una red Zigbee.....	17
2.2.3	Aplicaciones.....	19
2.3	Descripción Módulos Xbee .....	20
2.3.1	Datos Técnicos .....	21

	2.3.1.1 Especificaciones .....	22
	2.3.1.2 Descripción de los pines del módulo .....	23
2.3.2	Modos de Operación .....	23
	2.3.2.1 Modo Recibir/Transmitir .....	23
	2.3.2.2 Modo de Bajo Consumo .....	25
	2.3.2.3 Modo de Comando .....	25
	2.3.2.4 Modo Transparente .....	26
	2.3.2.5 Modo API .....	27
	2.3.2.6 Modo Idle .....	29
2.4	Breve descripción del sensor de temperatura LM35 .....	30
2.5	Herramientas de software .....	33
	2.5.1 MicroBasic Pro for Pic .....	33
	2.5.2 Proteus 7.5 .....	34
	2.5.3 Software X_CTU .....	35
3.	DISEÑO E IMPLEMENTACIÓN DEL SISTEMA	
	3.1 Diagrama de bloques .....	37
	3.2 Módulo Transmisor .....	38
	3.2.1 PIC 16F887 .....	38
	3.2.1.1 Código del programa .....	39
	3.2.2 LM 35 .....	49
	3.2.3 Módulo Xbee transmisor .....	49
	3.2.3.1 Consideraciones de alimentación y conexiones .....	50
	3.2.3.2 Configuración .....	53



3.2.4	Datalogger .....	56
3.2.5	Interfaz Gráfica .....	57
3.3	Módulo Receptor .....	57
3.3.1	PIC16F887 .....	57
3.3.1.1	Código del programa .....	57
3.3.2	Módulo Xbee receptor .....	66
3.3.2.1	Configuración .....	66
3.4	Tratamiento de errores e interferencia .....	67
4	PRUEBAS Y SIMULACIÓN	
4.1	Prueba de comunicación entre Xbee y PC .....	68
4.2	Prueba de transmisión- recepción entre XBee mediante PCs .....	69
4.3	Prueba de transmisión- recepción entre XBee mediante PICs .....	69
4.4	Prueba de recepción de paquetes .....	71
4.5	Simulación comunicación con datalogger .....	73

## **CONCLUSIONES**

## **RECOMENDACIONES**

## **ANEXOS**

## **BIBLIOGRAFÍA**

## ÍNDICE DE FIGURAS

Fig. 1.3.1:	Transceiver nRF2401A .....	5
Fig. 1.3.2:	Transceiver nRF24L01-CAL-SP .....	6
Fig. 1.3.3:	Transceiver CC2500 .....	7
Fig. 2.1.1.a:	Bandas de frecuencia de estándar 802.15.4.....	12
Fig. 2.1.1. b:	Gráfico comparativo de frecuencias y canales .....	13
Fig. 2.1.2:	Topologías.....	16
Fig. 2.2.a:	Arquitectura Zigbee .....	17
Fig. 2.2.b:	Red tipo malla(mesh) .....	18
Fig. 2.2.1:	Topologías soportadas por Zigbee .....	20
Fig. 2.2.3:	Comparación con otras tecnologías inalámbricas .....	23
Fig. 2.3:	Módulo Xbee Serie 1.....	24
Fig. 2.3.1.1:	Especificaciones Módulo Xbee Serie 1.....	25
Fig. 2.3.1.2:	Pines del Módulo Xbee Serie 1 .....	26
Fig. 2.3.2.2:	Consumo y configuración Modo Sleep.....	28
Fig. 2.3.2.3:	Formato Comando AT .....	29
Fig. 2.3.2.5:	Formato frame API.....	32
Fig. 2.4:	Características del sensor LM35 .....	34

Fig 2.4.a:	Distribución de pines del sensor LM35.....	35
Fig 2.4.b:	Configuración típica del sensor LM35.....	35
Fig 2.5.1:	Microbasic Pro.....	37
Fig 2.5.3:	X_CTU software.....	39
Fig 3.2.1:	Pines del PIC 16F887 .....	41
Fig 3.2.3.1.a:	Etapa de reducción de voltaje mediante LM317T.....	48
Fig 3.2.3.1.b:	Conexión PIC a módulo Xbee.....	51
Fig 3.2.3.2.a:	Configuración de direcciones con X-CTU.....	53
Fig 3.2.3.2.b:	Configuración de tasa de baudios con X-CTU.....	54
Fig 4.1:	Prueba con X-CTU.....	59
Fig 4.3:	Prueba Tx-Rx entre Xbee con PICs.....	61
Fig 4.4:	Pérdida de paquetes.....	63
Fig 4.5:	Comunicación con datalogger.....	64

# INTRODUCCIÓN

El presente Proyecto forma parte del seminario de graduación de “Microcontroladores Avanzados” y consiste en el diseño de un “Sistema de control remoto de robots”. El principal objetivo es desarrollar un proyecto basado en la comunicación entre módulos Xbee de radiofrecuencia, para el control de robots a distancia.

En el primer capítulo, se da una descripción del proyecto como tal, incluyendo los elementos a utilizarse, en especial los módulos Xbee, la base de nuestro trabajo; así como los alcances y limitaciones del sistema.

En el segundo capítulo, se detallará la base teórica sobre la cual se fundamenta el sistema, incluyendo el estándar IEEE 802.15.4, datos técnicos, eléctricos, y de configuración de los módulos de radiofrecuencia, sensor de temperatura y demás componentes del sistema.

En el tercer capítulo, se incluye cómo se implementó el sistema para su funcionamiento final.

En el cuarto y último capítulo, se recogerán las simulaciones y pruebas necesarias para garantizar el buen desempeño del sistema.

# **CAPÍTULO 1**

## **1 DESCRIPCIÓN GENERAL DEL PROYECTO**

### **1.1 Antecedentes**

La tecnología inalámbrica de transmisiones de datos para redes de área reducida, se desarrolla constantemente en busca de mejoras técnicas, económicas y de estandarización.

El mercado está ávido de acceder a dicha tecnología sin mayores dificultades, para desarrollar sus propios proyectos en variadas aplicaciones.

El presente proyecto se realizó, haciendo uso de módulos de comunicación inalámbrica, que justamente tratan de satisfacer las necesidades del mercado antes descritas, los denominados Xbee; cuyas ventajas y limitaciones se conocerán en este proyecto.

## 1.2 Descripción del Proyecto

El proyecto consiste en el desarrollo de un sistema que permita el manejo y control de robots de forma remota, basado en la comunicación entre módulos Xbee de radiofrecuencia a 2.4Ghz, los mismos que son de la Serie 1 y utilizan el estándar IEEE 802.15.4 para redes de área personal(WPAN).

Como parte controladora tendremos dos PICs 16F887 en ambos extremos de la comunicación, tanto en el transmisor como en el receptor, que se encargarán de indicar a los módulos Xbee respectivos, la tarea a realizar en determinado instante. Estos microcontroladores estarán programados mediante la herramienta de desarrollo, MikroBasic Pro, proporcionada por la empresa Mikroelectrónica.

Además, el sistema tendrá capacidad de comunicación serial a un datalogger e interfaz gráfica, para almacenamiento de datos y su correspondiente presentación, los cuales se emplearán de acuerdo a las necesidades y solicitudes del sistema durante su proceso.

Debido a que estos módulos son de la Serie 1, no nos permitirán implementar una red tipo malla. Además, este proyecto emplea el modo de

comunicación de todo lo que recibe serialmente transmite, sin hacer uso de comunicación con tramas.

### 1.3 Módulos de comunicación similares

#### 1.3.1 Transceiver nRF2401A

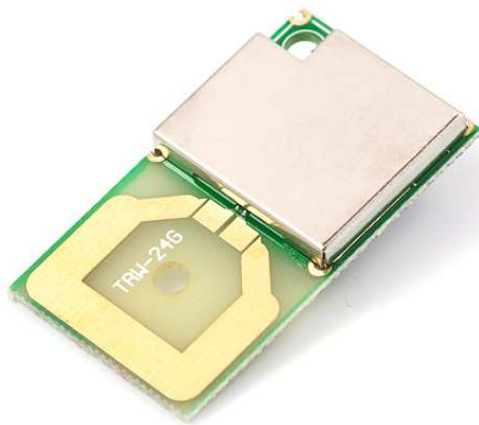


Fig. 1.3.1: Transceiver nRF2401A

Es un transceiver de la empresa Laipac de 24 pines, que consiste en un sintetizador de frecuencia, un amplificador de potencia, un oscilador con cristal, y un modulador, todo integrado.

Trabaja a 2.4 Ghz con una tasa de transferencia de hasta 1Mbps, puede utilizar hasta 125 canales. Referente a su alimentación, es de 1.9 a 3.6V, y su consumo de corriente oscila entre 10.5 y 18 mA.



### 1.3.2 Transceiver nRF24L01-CAL-SP

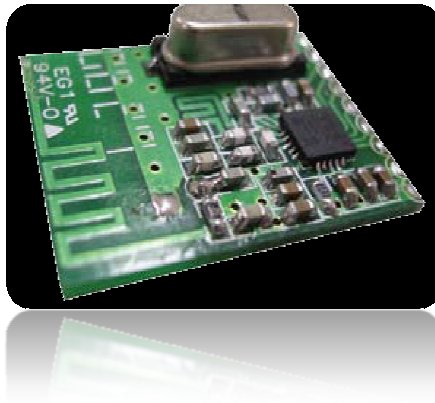


Fig. 1.3.2: Transceiver nRF24L01-C-SP

Es un módulo de RF a 2.4 Ghz de la empresa Nordic Semiconductor, de fácil manipulación, que soporta una tasa de transferencia de hasta 2Mbps.

Se alimenta con un voltaje de hasta 3.6V, y tiene un bajo consumo de potencia.

### 1.3.3 Transceiver CC2500

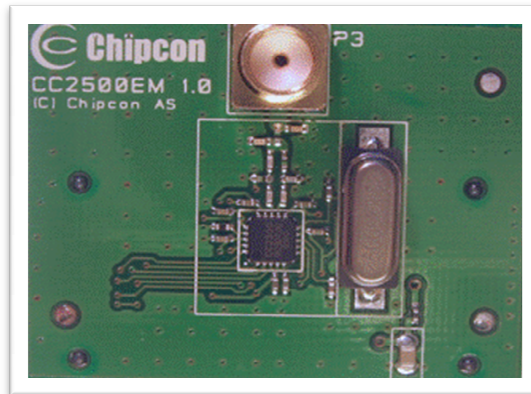


Fig. 1.3.3: Transceiver CC2500

Es un transceiver de bajo costo, diseñado para aplicaciones inalámbricas de baja potencia a 2.4Ghz. Tiene una tasa de transferencia configurable de 1.2 a 500 kbps, soporta varios tipos de modulación.

.Además posee chequeo de dirección, manejo de CRC, flexibilidad de longitud de paquete, y demás características embebidas en el chip.

### 1.4 Comparación entre transceivers similares con el Xbee.

El módulo Xbee se distingue de sus similares por lo siguiente:

Cada Xbee tiene un único número serial, lo que permite comunicarse con un módulo específico de forma exclusiva, ignorando señales de otros módulos.

Es de fácil integración, lo que permite una ventaja significativa respecto a reducción de tiempo de configuración e instalación.

Bajo consumo, en especial en modo sleep, lo cual implica un ahorro considerable de energía.

Permite la formación de redes de gran densidad, teóricamente hasta un máximo de 65536 dispositivos.

Permite la utilización de varios canales, un total de 16, a diferencia de otros transceivers que permiten mucho menos, hasta un máximo de 5 canales.

# **CAPÍTULO 2**

## **2 FUNDAMENTACIÓN TEÓRICA**

En el presente capítulo, se detalla el estándar en el que está basado el módulo Xbee, arquitectura y topología. Se incluye además una descripción de estos módulos, tanto en sus pines y modos de operación, así como también generalidades sobre el sensor de temperatura LM35DZ. Finalmente indicaremos las herramientas de software que utilizamos en este proyecto.

### **2.1 Estándar IEEE 802.15.4/ZIGBEE**

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

El grupo de trabajo IEEE 802.15 es el responsable de su desarrollo.

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

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

En su forma básica se concibe un área de comunicación de 10 metros con una tasa de transferencia de 250 kbps. Se definieron inicialmente tasas alternativas de 20 y 40 kbps; la versión actual añade una tasa adicional de 100 kbps. Se pueden lograr tasas aún menores con la consiguiente reducción de consumo de energía. Como se ha indicado, la característica fundamental de 802.15.4 entre las WPAN's es la obtención de costes de fabricación excepcionalmente bajos por medio de la sencillez tecnológica, sin perjuicio de la generalidad o la adaptabilidad. Un dispositivo que implementa el 802.15.4 puede transmitir en una de tres posibles bandas de frecuencia.

### **2.1.1 Arquitectura**

Los dispositivos se relacionan entre sí a través de una red inalámbrica sencilla. La definición de los niveles se basa en el modelo OSI.

El nivel físico (PHY) provee el servicio de transmisión de datos sobre el medio físico propiamente dicho, así como la interfaz con la entidad de gestión del nivel físico, por medio de la cual se puede acceder a todos los servicios de gestión del nivel y que mantiene una base de datos con información de redes de área personal relacionadas. De esta forma, PHY controla el transceptor de radiofrecuencia y realiza la selección de canales junto con el control de consumo y de la señal. Opera en una de tres posibles bandas de frecuencia de uso no regulado.

868-868,8 MHz: Europa, permite un canal de comunicación (versión de 2003), extendido a tres en la revisión de 2006.

902-928 MHz: Norte América, hasta diez canales (2003) extendidos a treinta (2006).

2400-2483,5 MHz: uso en todo el mundo, hasta dieciséis canales (2003, 2006).

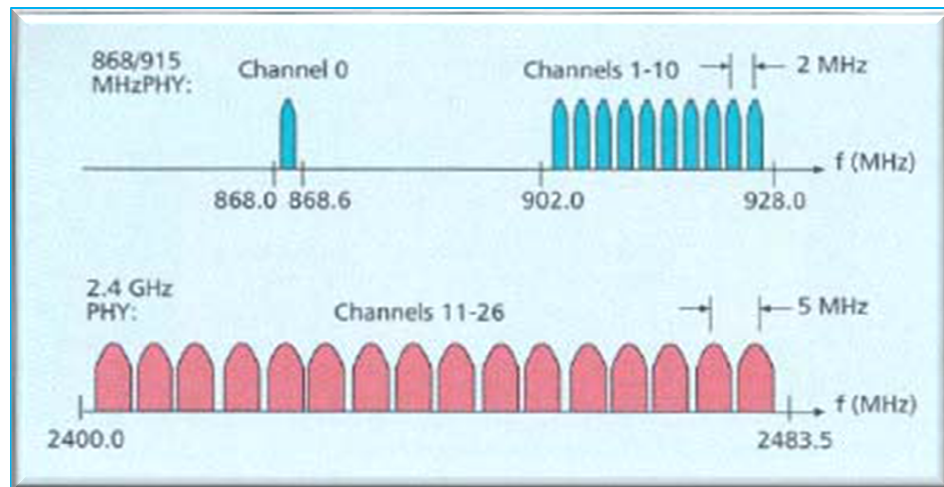


Fig. 2.1.1.a: Bandas de frecuencia de estándar 802.15.4

La versión original del estándar, de 2003, especifica dos niveles físicos basados en espectro ensanchado por secuencia directa (direct sequence spread spectrum, DSSS): uno en las bandas de 868/915 MHz con tasas de 20 y 40 kbps; y otra en la banda de 2450 MHz con hasta 250 kbps.

La revisión de 2006 incrementa las tasas de datos máximas de las bandas de 868/915 MHz, que permiten hasta 100 y 250 kbps.

El control de acceso al medio (MAC) transmite tramas MAC usando para ello el canal físico. Además del servicio de datos, ofrece un interfaz de control y regula el acceso al canal físico y al balizado de la red. También controla la validación de las tramas y las asociaciones entre nodos, y garantiza slots de tiempo. Por último, ofrece puntos de enganche para servicios seguros.

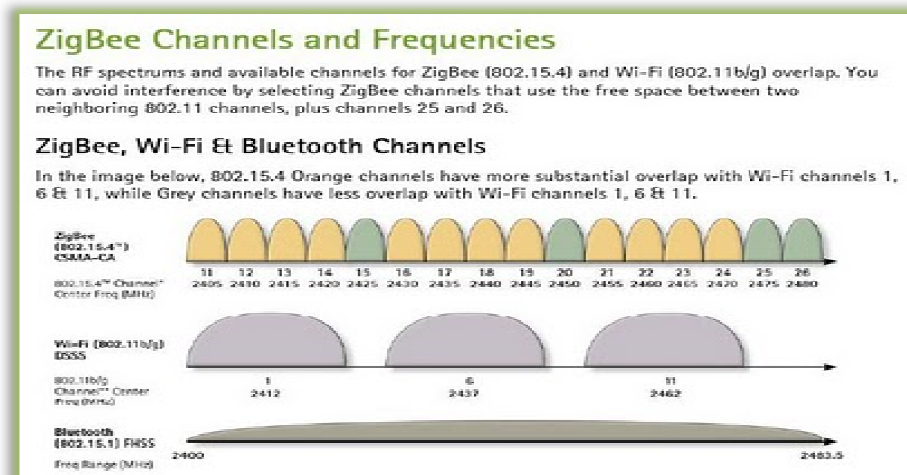


Fig. 2.1.1 b: Gráfico comparativo de frecuencias y canales

## 2.1.2 Topologías

El estándar no define niveles superiores ni subcapas de interoperabilidad. Existen extensiones, como la especificación de ZigBee, que complementan al estándar en la propuesta de soluciones completas.

El estándar define dos tipos de nodo en la red. El primero es el dispositivo de funcionalidad completa (full-function device, FFD). Puede funcionar como coordinador de una red de área personal (PAN) o como un nodo normal. Implementa un modelo general de comunicación que le permite establecer un intercambio con cualquier otro dispositivo. Puede, además, encaminar mensajes, en cuyo caso se le denomina coordinador (coordinador de la PAN si es el responsable de toda la red y no sólo de su entorno).



Contrapuestos a éstos están los dispositivos de funcionalidad reducida (reduced-function device, RFD). Se plantean como dispositivos muy sencillos con recursos y necesidades de comunicación muy limitadas. Por ello, sólo pueden comunicarse con FFD's y nunca pueden ser coordinadores.

Las redes de nodos pueden construirse como redes punto a punto o en estrella. En cualquier caso, toda red necesita al menos un FFD que actúe como su coordinador. Las redes están compuestas por grupos de dispositivos separados por distancias suficientemente reducidas; cada dispositivo posee un identificador único de 64 bits, aunque si se dan ciertas condiciones de entorno en éste pueden utilizarse identificadores cortos de 16 bits. Probablemente éstos se utilizarán dentro del dominio de cada PAN separada.

Las redes punto a punto pueden formar patrones arbitrarios de conexionado, y su extensión está limitada únicamente por la distancia existente entre cada par de nodos. Forman la base de redes ad hoc auto organizativas. El estándar no define un nivel de red, por lo que no se soportan funciones de ruteo de forma directa, aunque si dicho nivel se añade pueden realizarse comunicaciones en varios saltos.

Pueden imponerse otras restricciones topológicas; en concreto, el estándar menciona el árbol de clusters como una estructura que aprovecha que los RFD's sólo pueden conectarse con un FFD al tiempo para formar redes en las que los RFD's son siempre hojas del árbol, y donde la mayoría de los nodos son FFD's. Puede relajarse la estructura para formar redes en malla genéricas, cuyos nodos sean árboles de clusters con un coordinador local para cada cluster, junto con un coordinador global.

También pueden formarse redes en estrella, en las que el coordinador va a ser siempre el nodo central. Una red así se forma cuando un FFD decide crear su PAN y se nombra a sí mismo coordinador, tras elegir un identificador de PAN único. Tras ello, otros dispositivos pueden unirse a una red totalmente independiente del resto de redes en estrella.

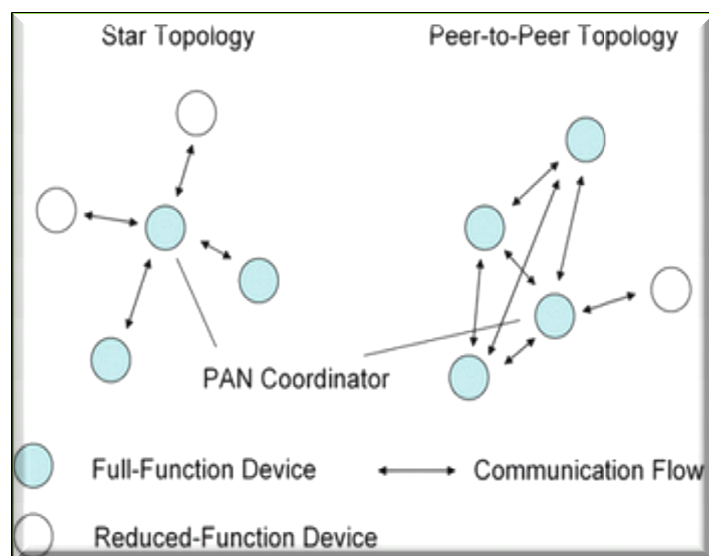


Fig. 2.1.2: Topologías

## 2.2 Zigbee

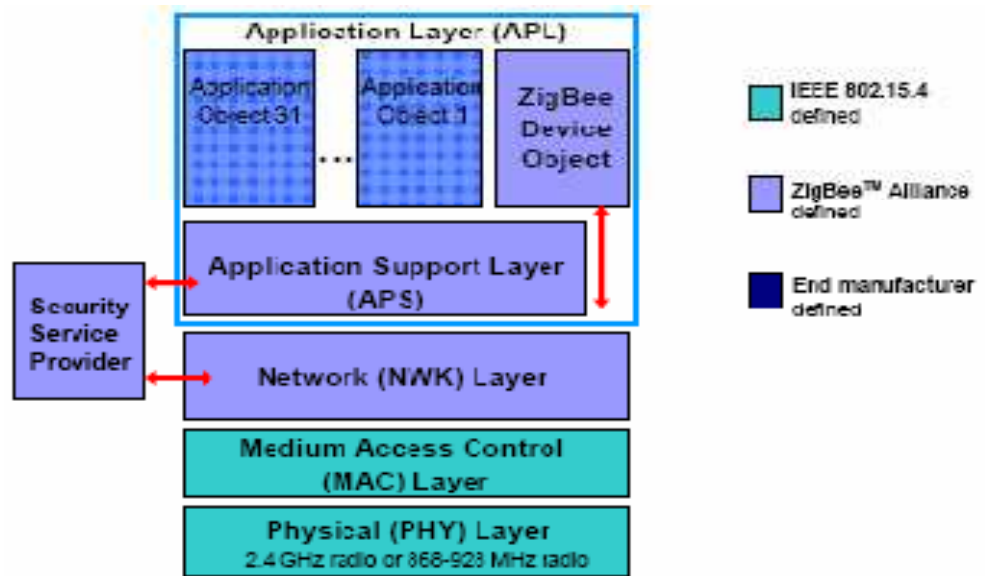


Fig. 2.2.a : Arquitectura Zigbee

Es un protocolo que usa el estándar 802.15.4 como base y añade funcionalidades de enrutamiento y de red. Fue desarrollado por la ZigBee Alliance. Esta alianza es un grupo de compañías que trabajaron en cooperación para desarrollar un protocolo de red que pueda ser usado en una variedad de aplicaciones tanto comerciales como industriales de baja tasa de transferencia de datos. Zigbee añade redes tipo mesh (malla) en donde el rango entre dos puntos puede estar más allá de los dos radios localizados en aquellos puntos, pero módulos intermedios están presentes para dirigir cualquier mensaje hacia o desde los radios deseados.



- Router ZigBee (ZigBee Router, ZR). Interconecta dispositivos separados en la topología de la red, además de ofrecer un nivel de aplicación para la ejecución de código de usuario.
- Dispositivo final (ZigBee End Device, ZED). Posee la funcionalidad necesaria para comunicarse con su nodo padre (el coordinador o un router), pero no puede transmitir información destinada a otros dispositivos. De esta forma, este tipo de nodo puede estar dormido la mayor parte del tiempo, aumentando la vida media de sus baterías. Un ZED tiene requerimientos mínimos de memoria y es por tanto significativamente más barato.

Las redes ZigBee han sido diseñadas para conservar la potencia en los nodos 'esclavos'. De esta forma se consigue el bajo consumo de potencia. La estrategia consiste en que, durante mucho tiempo, un dispositivo "esclavo" está en modo "dormido", de tal forma que solo se "despierta" por una fracción de segundo para confirmar que está "vivo" en la red de dispositivos de la que forma parte. Esta transición del modo "dormido" al modo "despierto" (modo en el que realmente transmite), dura unos 15ms, y la enumeración de "esclavos" dura alrededor de 30ms.

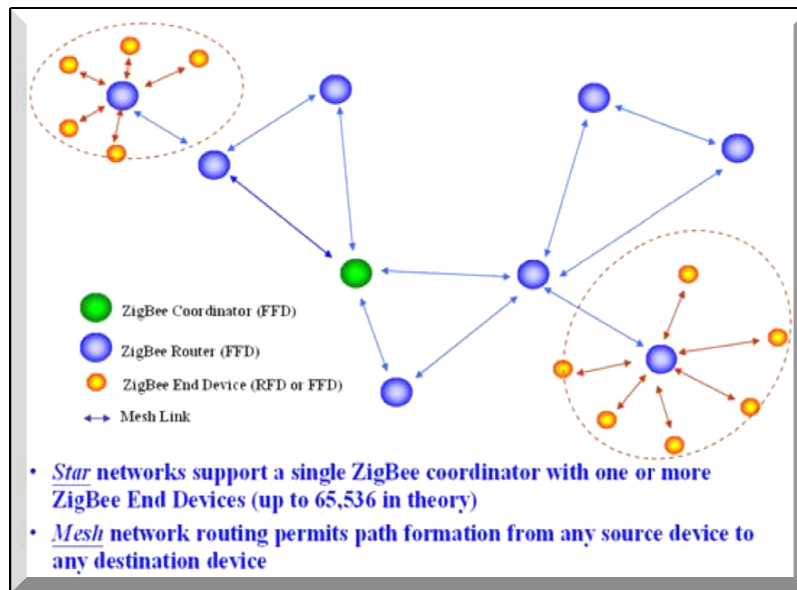


Fig. 2.2.1: Topologías soportadas por Zigbee

## 2.2.2 Estrategias de conexión en una red Zigbee

En las redes Zigbee, se pueden usar dos tipos de entornos o sistemas:

### 1. Con balizas

Es un mecanismo de control del consumo de potencia en la red. Permite a todos los dispositivos saber cuándo pueden transmitir. En este modelo, los dos caminos de la red tienen un distribuidor que se encarga de controlar el canal y dirigir las transmisiones. Las balizas que dan nombre a este tipo de entorno, se usan para poder sincronizar todos los dispositivos que conforman la red, identificando la red doméstica. Los intervalos de las balizas son asignados por el coordinador de red y pueden variar desde los 15ms hasta los 4 minutos.

Este modo es más recomendable cuando el coordinador de red trabaja con una batería. Los dispositivos que conforman la red, escuchan a dicho coordinador durante el "balizamiento" (envío de mensajes a todos los dispositivos -broadcast-, entre 0,015 y 252 segundos). Un dispositivo que quiera intervenir, lo primero que tendrá que hacer es registrarse para el coordinador, y es entonces cuando mira si hay mensajes para él. En el caso de que no haya mensajes, este dispositivo vuelve a "dormir", y se despierta de acuerdo a un horario que ha establecido previamente el coordinador. En cuanto el coordinador termina el "balizamiento", vuelve a "dormirse".

## 2. Sin balizas

Se usa el acceso múltiple al sistema Zigbee en una red punto a punto cercano. En este tipo, cada dispositivo es autónomo, pudiendo iniciar una conversación, en la cual los otros pueden interferir. A veces, puede ocurrir que el dispositivo destino puede no oír la petición, o que el canal esté ocupado.

Este sistema se usa típicamente en los sistemas de seguridad, en los cuales sus dispositivos (sensores, detectores de movimiento o de rotura de

cristales), duermen prácticamente todo el tiempo (el 99,999%). Para que se les tenga en cuenta, estos elementos se "despiertan" de forma regular para anunciar que siguen en la red. Cuando se produce un evento (en nuestro sistema será cuando se detecta algo), el sensor "despierta" instantáneamente y transmite la alarma correspondiente. Es en ese momento cuando el coordinador de red, recibe el mensaje enviado por el sensor, y activa la alarma correspondiente. En este caso, el coordinador de red se alimenta de la red principal durante todo el tiempo.

A lo que a nosotros respecta, usaremos entornos con balizas, donde el coordinador es alimentado mediante baterías.

### **2.2.3 Aplicaciones**

Los protocolos ZigBee están definidos para su uso en aplicaciones embebidas con requerimientos muy bajos de transmisión de datos y consumo energético. Se pretende su uso en aplicaciones de propósito general con características auto organizativas y bajo coste (redes en malla, en concreto). Puede utilizarse para realizar control industrial, albergar sensores empotrados, recolectar datos médicos, ejercer labores de detección de humo o intrusos o domótica. La red en su conjunto utilizará una cantidad muy pequeña de energía de forma que cada dispositivo individual pueda tener



una autonomía de hasta 5 años antes de necesitar un recambio en su sistema de alimentación.

	<b>ZigBee and 802.15.4</b>	<b>GSM/GPRS CDMA</b>	<b>802.11</b>	<b>Bluetooth</b>
<b>Focus Application</b>	Monitoring and Control	Wide Area Voice and Data	High-Speed Internet	Device Connectivity
<b>Battery Life</b>	Years	1 Week	1 Week	1 Week
<b>Bandwidth</b>	250 Kbps	Up to 2 Mbps	Up to 54 Mbps	720 Kbps
<b>Typical Range</b>	100+ Meters	Several Kilometers	50-100 Meters	10-100 Meters
<b>Advantages</b>	Low Power, Cost	Existing Infrastructure	Speed, Ubiquity	Convenience

Tabla 2.2.3 : Comparación con otras tecnologías inalámbricas

### 2.3 Descripción Módulos Xbee

Los módulos Xbee y Xbee Pro, fueron diseñados para soportar el estándar IEEE 802.15.4, operan en la banda ISM a 2.4Ghz y son compatibles pin a pin entre ellos.

Proveen una confiable entrega de datos entre los dispositivos, lo que ha permitido que sean aplicados para redes inalámbricas de sensores, de bajo costo y baja potencia.



Fig. 2.3: Módulo Xbee Serie 1

### 2.3.1 Datos Técnicos

## 2.3.1.1 Especificaciones

Specification	XBee	XBee-PRO
<b>Performance</b>		
Indoor/Urban Range	up to 100 ft. (30 m)	Up to 300' (100 m)
Outdoor RF line-of-sight Range	up to 300 ft. (100 m)	Up to 1 mile (1500 m)
Transmit Power Output (software selectable)	1mW (0 dBm)	60 mW (18 dBm) conducted, 100 mW (20 dBm) EIRP*
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 - 115200 bps (non-standard baud rates also supported)	1200 - 115200 bps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
<b>Power Requirements</b>		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	If PL=0 (10dBm): 137mA(@3.3V), 139mA(@3.0V) PL=1 (12dBm): 155mA (@3.3V), 153mA(@3.0V) PL=2 (14dBm): 170mA (@3.3V), 171mA(@3.0V) PL=3 (16dBm): 188mA (@3.3V), 195mA(@3.0V) PL=4 (18dBm): 215mA (@3.3V), 227mA(@3.0V)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 $\mu$ A	< 10 $\mu$ A
<b>General</b>		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector	Integrated Whip, Chip or U.FL Connector
<b>Networking &amp; Security</b>		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	
<b>Agency Approvals</b>		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max. 10 dBm transmit power output)*
Japan	n/a	005NYCA0378 (Max. 10 dBm transmit power output)**

Tabla 2.3.1.1: Especificaciones Módulo Xbee Serie 1

### 2.3.1.2 Descripción de los Pines del Módulo

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

Tabla 2.3.1.2: Pines del Módulo Xbee Serie 1

## 2.3.2 Modos de Operación

### 2.3.2.1 Modo Recibir/ Transmitir

Se encuentra en estos modos cuando el módulo recibe algún paquete RF a través de la antena(modos Receive) o cuando se envía información serial al buffer del pin 3 (UART Data in) que luego será transmitida (modo Transmit).

La información transmitida puede ser Directa o Indirecta. En el modo directo la información se envía inmediatamente a la dirección de destino. En el modo Indirecto la información es retenida por el módulo durante un período de tiempo y es enviada sólo cuando la dirección de destino la solicita.

Además es posible enviar información por dos formas diferentes. Unicast y Broadcast.

Por la primera, la comunicación es desde un punto a otro, y es el único modo que permite respuesta de quien recibe el paquete RF, es decir, quien recibe debe enviar un ACK (paquete llamado así, y que indica que recibió el paquete, el usuario no puede verlo, es interno de los módulos) a la dirección de origen. Quien envió el paquete, espera recibir un ACK, en caso de que no le llegue, reenviará el paquete hasta 3 veces o hasta que reciba el ACK. Después de los 3 intentos se incrementa el registro de fallas ACK en uno. Para ver el conteo se utiliza el comando EA (ATEA en el Modo de Comandos- ACK Failures), el cual se satura hasta 0xFFFF (65535decimal). Para resetear el registro se debe ingresar ATEA0. En el modo Broadcast la comunicación es entre un nodo y a todos los nodos de la red. En este modo, no hay confirmación por ACK.

### 2.3.2.2 Modo de Bajo Consumo

El modo de Bajo Consumo o modo Sleep, ocurre cuando el módulo entra en un estado de bajo consumo de energía, cuando no está en uso. Para que entre en este estado se debe dar lo siguiente:

- Sleep\_RQ (pin 9) debe estar en alto
- El módulo debe estar sin recibir ni transmitir por el tiempo definido en el parámetro ST (Time before sleep).

Modo	Consumo alimentación			Modo Sleep	Modo Wake-up
	2.8 - 3 V	3.2 V	3.4V		
SM=1	<3 uA	32 uA	255 uA	Sleep_RQ	Sleep_RQ
SM=2	<35 uA	48 uA	170 uA	Sleep_RQ	Sleep_RQ
SM=3	(reservado)			(reservado)	(reservado)
SM=4	<34 uA	49 uA	240 uA	Comando ST	Comando SP
SM=5	<34 uA	49 uA	240 uA	Comando ST	Sleep RQ

Tabla 2.3.2.2: Consumo y configuración en el Modo Sleep

### 2.3.2.3 Modo de Comando

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

cosas. Para poder ingresar los comandos AT es necesario utilizar el Hyperterminal de Windows, el programa X-CTU4 o algún microcontrolador que maneje UART y tenga los comandos guardados en memoria o los adquiera de alguna otra forma.

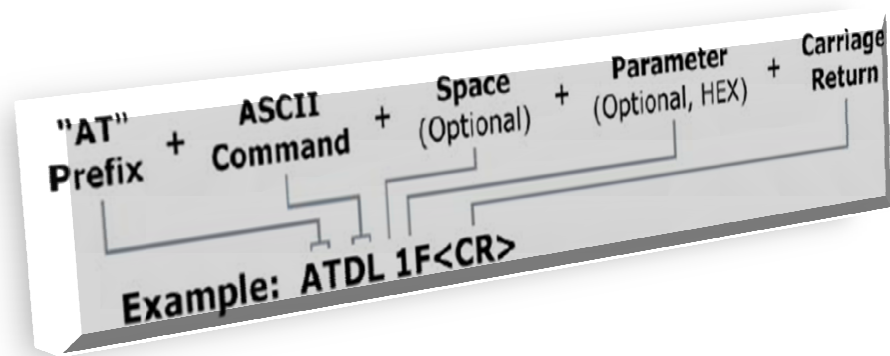


Fig. 2.3.2.3: Formato comando AT

#### 2.3.2.4 Modo Transparente

En este modo todo lo que ingresa por el pin 3 (Data in), es guardado en el buffer de entrada y luego transmitido y todo lo que ingresa como paquete RF, es guardado en el buffer de salida y luego enviado por el pin 2 (Data out). El modo Transparente viene por defecto en los módulos Xbee.

Este modo está destinado principalmente a la comunicación punto a punto, donde no es necesario ningún tipo de control. También se usa para

reemplazar alguna conexión serial por cable, ya que es la configuración más sencilla posible y no requiere una mayor configuración.

En este modo, la información es recibida por el pin 3 del módulo Xbee, y guardada en el buffer de entrada. Dependiendo de cómo se configure el comando RO, se puede transmitir la información apenas llegue un carácter (RO=0) o después de un tiempo dado sin recibir ningún carácter serial por el pin 3. En ese momento, se toma lo que se tenga en el buffer de entrada, se empaqueta, es decir, se integra a un paquete RF, y se transmite. Otra condición que puede cumplirse para la transmisión es cuando el buffer de entrada se llena, esto es, más de 100 bytes de información.

### **2.3.2.5 Modo API**

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



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

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

Mientras que un Frame de Recepción de Información incluye:

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

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

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

(potencia de la señal de recepción) e información de la carga útil de los paquetes recibidos.

Entre las opciones que permite la API, se tienen:

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

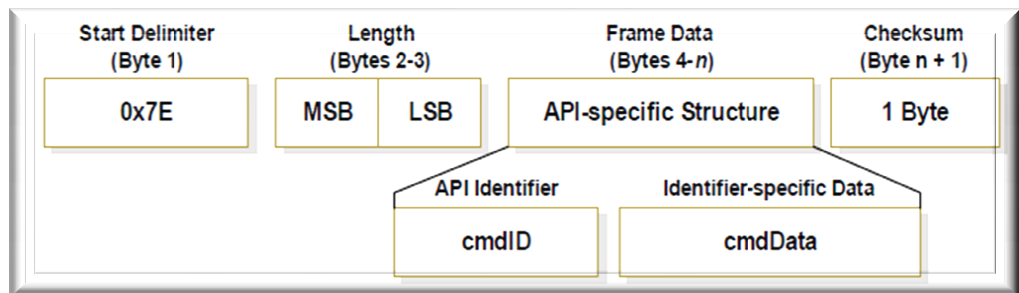


Fig. 2.3.2.5: Formato frame API

### 2.3.2.6 Modo Idle

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

## **2.4 Breve descripción del Sensor de temperatura LM35**

El circuito integrado LM35 es un sensor de temperatura de precisión, cuyo voltaje de salida es linealmente proporcional a la temperatura en grados Celsius o Centígrados.

El sensor LM35 tiene una ventaja sobre aquellos sensores calibrados en grados Kelvin, pues el usuario no necesita sustraer una constante de voltaje grande desde su salida para obtener una conveniente escala en grados Centígrados. No requiere de calibración externa o regulación para proveer medidas con variaciones de  $\pm 1.4$  °C en habitación y  $\pm 3.4$  °C en exteriores sobre una escala completa de -55 a 150 °C. Características como la baja impedancia de salida, salida lineal y una calibración interna precisa facilitan la interfaz de lectura y control.

El LM35DZ puede ser utilizado con una sola fuente de alimentación o más fuentes, pues consume sólo 60 uA de su fuente; tiene un calentamiento interno menor a 0.1°C en el aire. El sensor LM35 puede ser aplicado fácilmente en la misma forma que otros circuitos integrados. Puede ser pegado o fijado en una superficie y la temperatura registrada variará en 0.01 °C con respecto de la temperatura de la superficie.

Algunas características propias del sensor, se pueden detallar las mismas en la siguiente tabla:

<b>Parámetro</b>
Calibración directa en grados centígrados (°C)
Factor de escala lineal de +10.0mv/°C
Precisión garantizada de 0.5 a 25 °C (ambiente)
Rango completo de -55 a +150 °C
Adecuado para aplicaciones remotas
Bajo costo
Opera desde 4 a 20Vdc
Corriente de consumo menor a 60 ua
Baja impedancia de salida

Tabla 2.4: Características del sensor LM35

La distribución de pines es la siguiente:

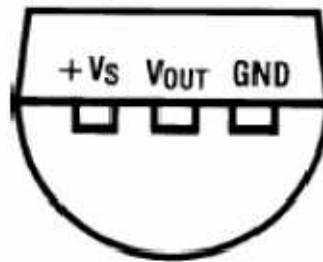


Fig 2.4.a: Distribución de pines del sensor LM35

Y la configuración típica que se utiliza en este proyecto, es la siguiente:

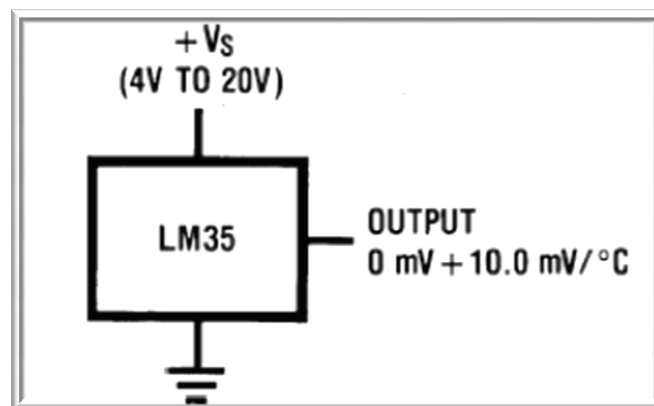


Fig 2.4.b: Configuración típica del sensor LM35

## 2.5 Herramientas de Software

### 2.5.1 MicroBasic Pro for PIC

Herramienta de desarrollo proporcionada por la empresa mikroelectrónica, es una versión nueva, y está en un nivel intermedio de programación entre C y ensamblador.

Permite al programador desarrollar código para microcontroladores de una manera más sencilla y comprensible, ya que incluye funciones y librerías muy útiles, que nos ahorran mucho trabajo y líneas de programa.

Además de permitir la depuración del código como la mayoría de herramientas de programación, incluye unas herramientas como GLCD Bitmap Editor, Usart Terminal, EEPROM Editor, etc.



### **2.5.3 Software X\_CTU**

Software gratuito, que se utiliza como interfaz de configuración, y prueba para varios módulos de radiofrecuencia de la empresa Maxstream(hoy Digi), entre ellos los Xbee.

Contiene 4 opciones, las cuales son las siguientes:

- Configuraciones del Pc : Sirve para configurar los puertos seriales que intervendrán en la comunicación con el módulo RF
- Opción de prueba de rango: Donde se puede realizar un test del rango del módulo RF, además presenta una pantalla con los paquetes recibidos y transmitidos.
- Opción Terminal: Usado con el fin de leer los datos recibidos y enviados hacia y desde otro dispositivo Xbee.
- Opción de Configuración del Módem: Para configurar o leer ciertos parámetros.



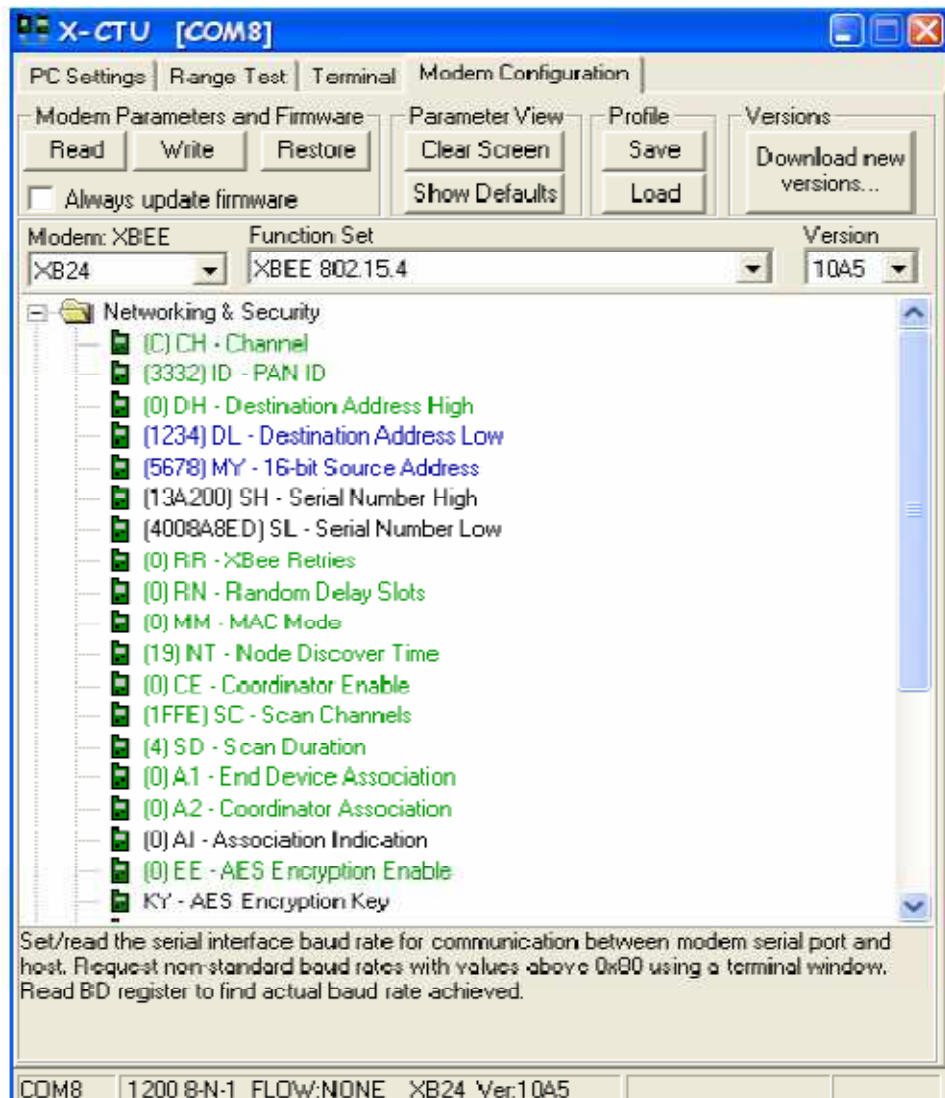


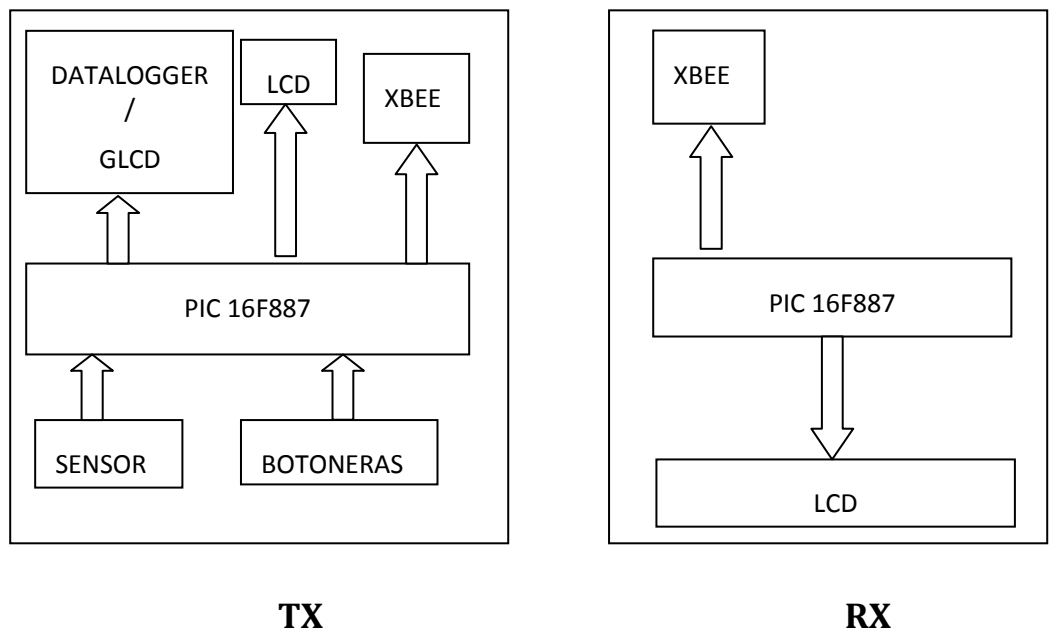
Fig 2.5.3: X-CTU software

# CAPÍTULO 3

## 3 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

Este capítulo nos introducirá al diseño de nuestro proyecto, para esto, inicialmente se presentará un diagrama de bloques general, el cual se irá desglosando a lo largo del capítulo, incluyendo los elementos en la parte transmisora y receptora, su configuración y programación.

### 3.1 Diagrama de Bloques



## 3.2 Módulo Transmisor

### 3.2.1 PIC 16F887

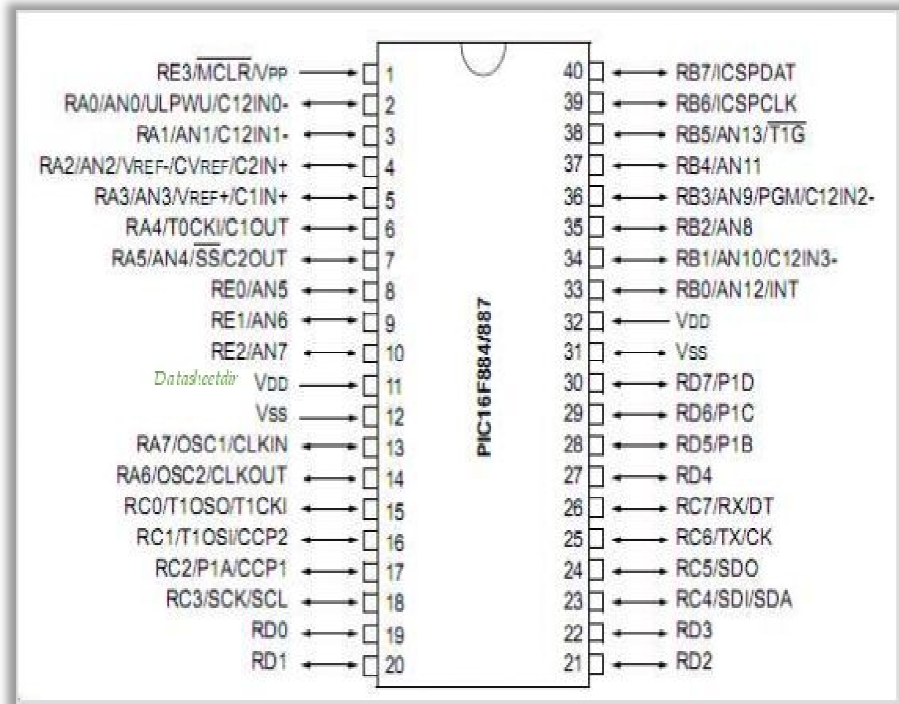


Fig 3.2.1: Pines del PIC 16F887

Será el encargado de controlar la parte de transmisión del sistema. Bajo su cargo tendrá el tomar los valores de voltaje de la salida del sensor de temperatura LM35 por el canal 2, estos datos serán transferidos al módulo Xbee mediante comunicación serial asincrónica, usando el módulo UART que contiene el encapsulado de este PIC, esto es RC6 para transmitir (TX) y RC7

para recibir (RX). Enseguida el Xbee enviará por radiofrecuencia los datos al receptor.

El puerto RB será utilizado para colocar un Led indicador, tanto de término de grabación en datalogger como de finalización de presentación en GLCD . El PIC trabajará a 20Mhz a una tasa de 9600 bps.

Los pines RC6 y RC7 serán configurados como salida y entrada respectivamente, y se conectarán tanto al Xbee transmisor como al datalogger que almacenará los datosy al GLCD que los mostrará.

### **3.2.1.1 Código del programa**

El código del PIC transmisor se presenta a continuación:

```
program tx
```

```
dim uart_rd as byte
```

```
dim adc_ent, valor1, valor2,valor3,valor4 as integer
```

```
dim adc_rd ,muestra1,muestra2,muestra3,promedio as word
```

```
dim byte_read ,adc_rd_byte1,adc_rd_byte2 ,contador,cont as byte
```

```
dim conv as string[16]
```

```
dim LCD_RS as sbit at RB4_bit
```

LCD\_EN as sbit at RB5\_bit

LCD\_D4 as sbit at RB0\_bit

LCD\_D5 as sbit at RB1\_bit

LCD\_D6 as sbit at RB2\_bit

LCD\_D7 as sbit at RB3\_bit

LCD\_RS\_Direction as sbit at TRISB4\_bit

LCD\_EN\_Direction as sbit at TRISB5\_bit

LCD\_D4\_Direction as sbit at TRISB0\_bit

LCD\_D5\_Direction as sbit at TRISB1\_bit

LCD\_D6\_Direction as sbit at TRISB2\_bit

LCD\_D7\_Direction as sbit at TRISB3\_bit

sub function convertir() as word

    delay\_ms(300)

    muestra1= ADC\_Read(2)

    delay\_ms(300)

    muestra2 = ADC\_Read(2)

```
delay_ms(300)

promedio=(muestra1+muestra2)/2

result=promedio*0.488

end sub
```

```
sub procedure mostrar()

contador=0

while (contador<8)

do

    if (UART1_Data_Ready() = 1) then

        uart_rd = UART1_Read()

    end if

loop until (uart_rd = 0x17)

uart1_write(0x22)

delay_ms(1000)

adc_rd=ADC_Read(2)

valor2=adc_rd*0.488

inttostr(valor2, conv)

Lcd_out(2,4,conv)
```

```
delay_ms(1000)

uart1_write(valor2)

contador=contador+1

delay_ms(5000)

wend

portb.7=1

delay_ms(1000)

portb.7=0

end sub
```

```
sub procedure guardar_datalogger()

dim str_tx as word

dim txt , enter as string[20]

dim j as byte

enter[0] = 13

enter[1] = 0

UART1_Write_Text("INI")      ' sends back text
```

```
UART1_Write(13)

UART1_Read_Text(txt, "CF", 10)

UART1_Read_Text(txt, enter, 10)

UART1_Write_Text("CN")      ' sends back text

UART1_Write(13)

UART1_Read_Text(txt, "OK", 10)

UART1_Read_Text(txt, enter, 10)

UART1_Write_Text("XBEE")

UART1_Write(13)

UART1_Read_Text(txt, "OK", 10)

UART1_Read_Text(txt, enter, 10)

UART1_Write_Text("WR")

UART1_Write(13)

UART1_Read_Text(txt, "OK", 10)

UART1_Read_Text(txt, enter, 10)

delay_ms(100)

for j = 0 to 19

promedio =convertir()
```



```
WordToStr(promedio, txt)

UART1_Write_Text(txt)

UART1_Write(13)

UART1_Read_Text(txt, "OK", 10)

UART1_Read_Text(txt, enter, 10)

next j

UART1_Write_Text("END")

UART1_Write(13)

UART1_Read_Text(txt, "EF", 10)

UART1_Read_Text(txt, enter, 10)

portb.7=1

delay_ms(1000)

portb.7=0

end sub

main:

ansel=0x04

anselh=0x00

trisb.7=0
```

```
trisa.6=0

trisa.5=0

trisa.1=1

trisa.0=0

porta.7=0

porta.6=0

porta.5=0

trisc.7=1

trisc.6=0

delay_ms(1200)      'Retardo para que xbee se encienda y se resetee

oscon=0

contador=0

UART1_Init(9600)    ' Inicializar uart a 2400bps

Delay_ms(100)

Lcd_Init()          ' Initialize Lcd

Lcd_Cmd(_LCD_CLEAR)      ' Clear display

Lcd_Cmd(_LCD_CURSOR_OFF)  ' Cursor off

Lcd_Out(1,1,"Modulos XBEE")      ' Write text in first row

delay_ms(2000)
```

```

Lcd_Cmd(_LCD_CLEAR)      ' Clear display

Lcd_Out(1,1,"1-GUARDAR")    ' Write text in first row

Lcd_Out(1,12,"2-TX")      ' Write text in first row

Lcd_Out(2,1,"3-GLCD")     ' Write text in first row

delay_ms(2000)

while (true)

    if Button(PORTA, 0, 1, 0) then

        Lcd_Cmd(_LCD_CLEAR)      ' Clear display

        Lcd_Out(1,1,"Guardando datos..")    ' Write text in first row

        guardar_datalogger()

        Lcd_Cmd(_LCD_CLEAR)      ' Clear display

        Lcd_Out(1,1,"Datos guardados")      ' Write text in first row

        delay_ms(2000)

        Lcd_Cmd(_LCD_CLEAR)      ' Clear display

        Lcd_Out(1,1,"1-GUARDAR")    ' Write text in first row

        Lcd_Out(1,12,"2-TX")      ' Write text in first row

        Lcd_Out(2,1,"3-GLCD")     ' Write text in first row

        delay_ms(2000)

```

```
end if
```

```
if Button(PORTA, 1, 1, 0) then
```

```
cont=0
```

```
While(cont<5)
```

```
    Lcd_Cmd(_LCD_CLEAR)      ' Clear display
```

```
    Lcd_Out(1,1,"Transmitiendo")      ' Write text in first row
```

```
    valor1=convertir()
```

```
    uart1_write(valor1)
```

```
    while (UART1_Data_Ready() = 0) 'pregunta si llegó dato
```

```
    wend
```

```
    uart_rd=Uart1_Read()
```

```
    Lcd_Cmd(_LCD_CLEAR)      ' Clear display
```

```
    Lcd_Out(1,1,"Recibiendo")      ' Write text in first row
```

```
    Lcd_Out(2,4,"ACK")      ' Write text in first row
```

```
    delay_ms(3000)
```

```
    cont=cont+1
```

```

wend

Lcd_Cmd(_LCD_CLEAR)      ' Clear display

Lcd_Out(1,1,"1-GUARDAR")  ' Write text in first row

Lcd_Out(1,12,"2-TX")     ' Write text in first row

Lcd_Out(2,1,"3-GLCD")    ' Write text in first row

delay_ms(2000)

end if

if Button(PORTA, 3, 1, 0) then

Lcd_Cmd(_LCD_CLEAR)      ' Clear display

Lcd_Out(1,1,"Temperatura")  ' Write text in first row

mostrar()

Lcd_Cmd(_LCD_CLEAR)      ' Clear display

Lcd_Out(1,1,"1-GUARDAR")  ' Write text in first row

Lcd_Out(1,12,"2-TX")     ' Write text in first row

Lcd_Out(2,1,"3-GLCD")    ' Write text in first row

delay_ms(2000)

end if

wend

end.

```

### 3.2.2 LM35

En este proyecto, haremos uso de un sensor de temperatura ampliamente conocido, el LM35, cuya salida estará conectada al canal 2 del convertidor ADC del PIC, para digitalizar el valor de voltaje dado por este sensor.

Como vimos en el capítulo 2, el LM35 tiene una relación de salida de 10mv/°C, y el convertidor del PIC es de 10 bits; entonces para obtener el Valor en grados centígrados, hacemos el siguiente cálculo que se incluye en el código del PIC transmisor:

**Valor analógico**=Valor dado por el ADC\*Resolución del ADC\*  
Relación de salida del sensor

Donde:

Resolución = Voltaje de referencia/2<sup>#bits</sup>

Relación del sensor = 1°C/10mv

Entonces, tenemos que:

**Valor analógico**=Valor dado por el ADC\*(5V/2<sup>10</sup>)\*(1°C/10mv)

**Valor analógico**=Valor dado por el ADC\*0.488 [°C]

Se conectará un capacitor de 1uf en serie con un resistor de 100 ohmios en su salida, para reducir el ruido.

### 3.2.3 Módulo Xbee transmisor

El módulo será un Xbee de la Serie 1, con una antena tipo chip.

#### 3.2.3.1 Consideraciones de alimentación y conexiones

En primera instancia, es necesario conocer que estos módulos trabajan con un voltaje de alimentación de máximo 3.3V, y ya que nuestro PIC trabaja a 5V, fue necesario incluir una etapa de reducción usando un regulador de voltaje variable LM317T.

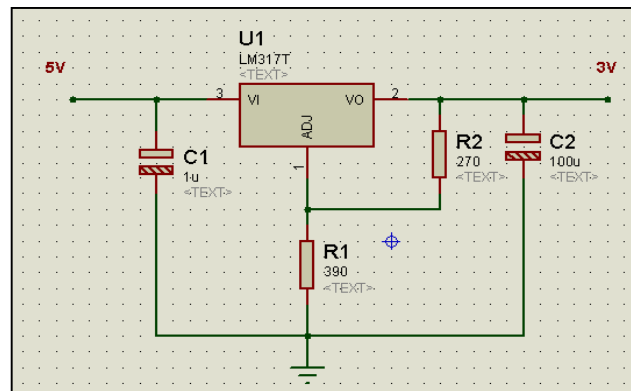


Fig 3.2.3.1.a: Etapa de reducción de voltaje mediante LM317T

Los valores de las resistencias vienen dados por la siguiente fórmula:

$$V_{out} = V_{ref} (1 + R_2/R_1) + I_{adj} R_2$$

Si sabemos que  $I_{adj}$  es muy baja y definimos  $R_1 = 240$  ohmios, entonces:

$$R_2 = R_1 [(V_{out}/V_{ref}) - 1]$$

$$R_2 = 240 [(3.3/1.25) - 1] = 393 \text{ ohmios}$$

De donde  $R_1 = 240$  ohmios y  $R_2 = 390$  ohmios.

Así mismo para el conexionado con el PIC, fue necesario un divisor de voltaje en el pin TX del microcontrolador, ya que este pone en 5V al pin en estado alto, lo que quemaría al módulo Xbee, ya que es muy sensible.

Las resistencias utilizadas fueron calculadas en base a la corriente que soporta el módulo y la corriente que es capaz de entregar el microcontrolador, considerando además si es una entrada para el módulo o una salida del módulo.

Para una salida: Tomando en cuenta que la corriente de salida del módulo es

$$I_{OUT} = 2\text{mA},$$

$$(5\text{V} - 3.3\text{V})/R < 2\text{mA}$$

Donde:

5V es el voltaje del PIC



3.3V es el voltaje del módulo

R la resistencia a calcular

De donde:  $R > 850$  ohmios.

Se escogió una resistencia de 12k entre la línea de recepción del PIC y transmisión el Xbee.

**Para una entrada:** Tomando en cuenta que la corriente de entrada del módulo es  $I_{IN} = 0.025\mu A$  y la corriente que entrega el pin del microcontrolador es 20mA se calculó el valor de las resistencias utilizadas. Teniendo un voltaje de 5 VDC (voltaje del microcontrolador) se puede calcular  $R_T$  para los dos casos:

$$R_T = 5VDC / 20mA = 250\Omega \quad \text{y} \quad R_T = 5VDC / 0.025\mu A = 200M\Omega$$

Donde  $R_T$  es la resistencia equivalente de las dos resistencias en serie que se utilizan en el módulo. Por lo tanto se debe cumplir:

$$250\Omega < R_T < 200M\Omega$$

Se escogió resistencias estándar  $R_5 = 22k\Omega$  con lo que se obtuvo  $R_7 = 15k\Omega$ .

De aquí tenemos que el voltaje en la entrada VDI del Xbee es:

$$V_{DI} = 5V \left( \frac{22k}{22k + 15k} \right) = 2.98 V$$

Que se encuentra dentro del rango de trabajo de 2.8 a 3.3 V.

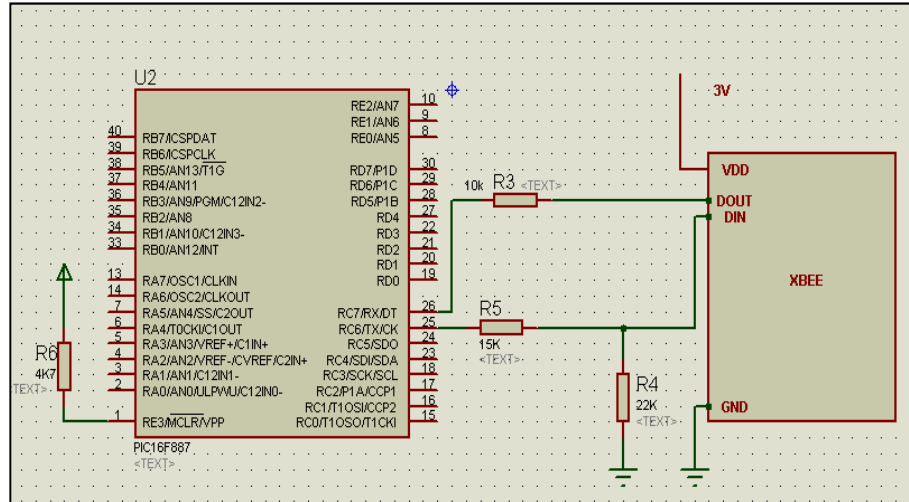


Fig 3.2.3.1.b: Conexión PIC a módulo Xbee

### 3.2.3.2 Configuración

Nuestros módulos, por defecto trabajan en modo transparente, que implica que cada mensaje recibido es puesto en el búfer de transmisión para su envío. Trabajaremos en este modo de comunicación por facilidad y porque no manejaremos mayor tráfico de datos, ya que la comunicación será punto a punto.

Los parámetros a configurar fueron:

PANID: Dirección de la red= 3332; ambos módulos deben tener el mismo PANID y el mismo canal para que estén en la misma red y puedan comunicarse entre sí.

Canal =C (12)

MY: Dirección de origen de 16-bit =1; es la dirección propia de cada módulo, que lo identifica en la red. Como la dirección es de 16 bits, una red podría contener  $2^{16} = 65536$  nodos, en teoría.

DL: Dirección de Destino =0; mientras DL sea  $< 0xFFFF$ , la transmisión será recibida por el módulo cuyo MY = DL.

Tasa de transferencia de datos seriales = 9600bps.

Todos estos parámetros y muchos más se pueden configurar de una manera sencilla mediante el software X-CTU. Aquí se muestra lo dicho:

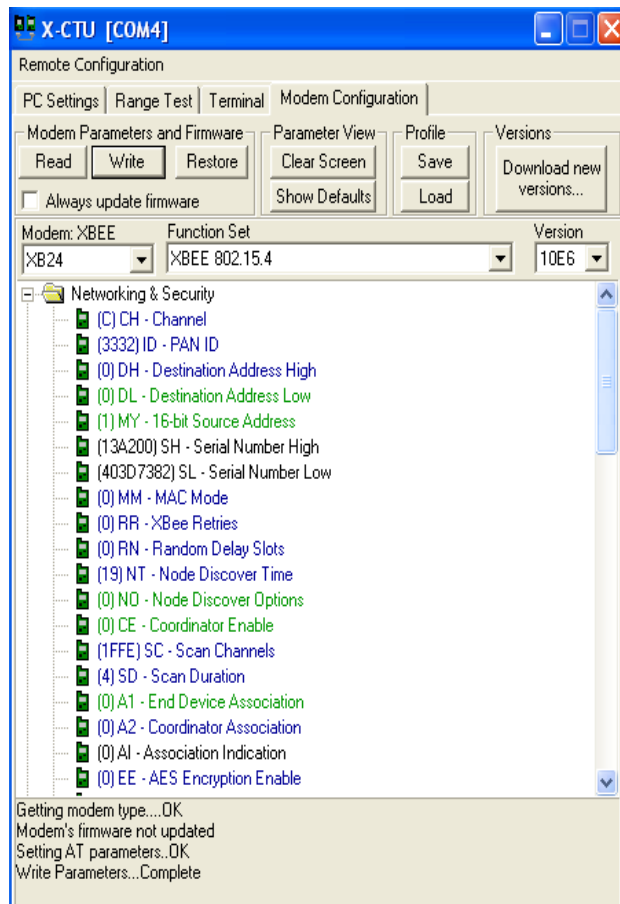


Fig 3.2.3.2.a: Configuración de direcciones con X-CTU

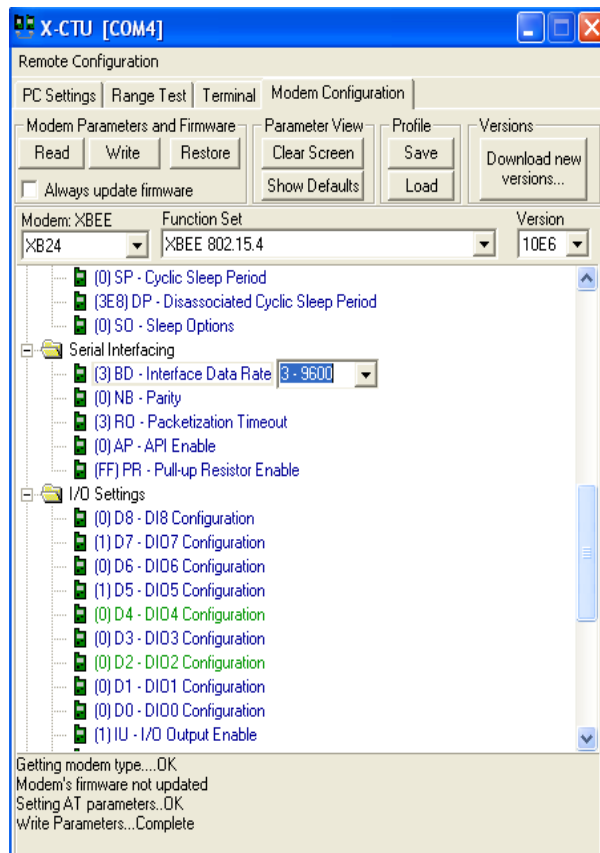


Fig 3.2.3.2.b: Configuración de tasa de baudios con X-CTU

### 3.2.4 Datalogger

Consiste en un módulo independiente de almacenamiento de datos en un pen drive. Accedemos a él tanto para escribir o leer de él, de acuerdo a ciertos comandos utilizados.

En nuestro caso, guardaremos en el datalogger, los datos de temperatura tomados del sensor LM35, con muestras a intervalos de 30s, esto se lo hará mediante comunicación serial usando el UART de nuestro PIC.

### **3.2.5 Interfaz Gráfica**

Para mostrar una gráfica de temperatura con datos en tiempo real, haremos uso de una pantalla GLCD.

## **3.3 Módulo Receptor**

### **3.3.1 PIC 16F887**

Será quien procese el paquete recibido por el módulo Xbee, mostrando mediante un Led si se recibió un dato. Además, enviará al módulo transmisor un mensaje de recibido, para indicarle que continúe con la comunicación.

#### **3.3.1.1 Código del programa**

```
program receptor
```

```
dim uart_rd as byte
```

```
dim adc_ent, valor1, valor2, valor3, valor4 as integer
```

```
dim adc_rd ,muestra1,muestra2,muestra3,promedio as word
```

```
dim error_flag, byte_read ,adc_rd_byte1,adc_rd_byte2 as byte
```

```
dim txt1,txt2,txt3,txt4,conv as string[16]
```

```
dim LCD_RS as sbit at RB4_bit
```

```
    LCD_EN as sbit at RB5_bit
```

```
    LCD_D4 as sbit at RB0_bit
```

```
    LCD_D5 as sbit at RB1_bit
```

```
    LCD_D6 as sbit at RB2_bit
```

```
    LCD_D7 as sbit at RB3_bit
```

```
    LCD_RS_Direction as sbit at TRISB4_bit
```

```
    LCD_EN_Direction as sbit at TRISB5_bit
```

```
    LCD_D4_Direction as sbit at TRISB0_bit
```

```
    LCD_D5_Direction as sbit at TRISB1_bit
```

```
    LCD_D6_Direction as sbit at TRISB2_bit
```

```
    LCD_D7_Direction as sbit at TRISB3_bit
```

```
sub function convertir() as word
```

```
    delay_ms(300)
```

```
muestra1= ADC_Read(2)

delay_ms(300)

muestra2 = ADC_Read(2)

delay_ms(300)

muestra3 = ADC_Read(2)

delay_ms(300)

promedio=(muestra1+muestra2+muestra3)/3

result=promedio*0.488

end sub
```

```
sub procedure mostrar()

dim contador as byte

contador=0

while (contador<8)

do

if (UART1_Data_Ready() = 1) then

uart_rd = UART1_Read()

end if

loop until (uart_rd = 0x17)
```



```
    uart1_write(0x22)

    delay_ms(1000)

    adc_rd=ADC_Read(2)

    valor2=adc_rd*0.488

    inttostr(valor2, conv)

    Lcd_out(2,4,conv)

    delay_ms(1000)

    uart1_write(valor2)

    contador=contador+1

    delay_ms(5000)

wend

end sub

sub procedure guardar_datalogger()

dim str_tx as word

dim txt , enter as string[20]
```

```
dim j as byte
```

```
'str_tx=0x1022
```

```
enter[0] = 13
```

```
enter[1] = 0
```

```
UART1_Write_Text("INI")      ' sends back text
```

```
UART1_Write(13)
```

```
UART1_Read_Text(txt, "CF", 10)
```

```
UART1_Read_Text(txt, enter, 10)
```

```
UART1_Write_Text("CN")      ' sends back text
```

```
UART1_Write(13)
```

```
UART1_Read_Text(txt, "OK", 10)
```

```
UART1_Read_Text(txt, enter, 10)
```

```
UART1_Write_Text("XBEE")
```

```
UART1_Write(13)
```

```
UART1_Read_Text(txt, "OK", 10)
```

```
UART1_Read_Text(txt, enter, 10)
```

```
UART1_Write_Text("WR")
```

```
UART1_Write(13)
```

```
UART1_Read_Text(txt, "OK", 10)
```

```
UART1_Read_Text(txt, enter, 10)
```

```
delay_ms(100)
```

```
for j = 0 to 19
```

```
    promedio=convertir()
```

```
    'INTToStr(valor1, txt)
```

```
    WordToStr(promedio, txt)
```

```
    UART1_Write_Text(txt)
```

```
    'UART1_Write_Text(str_tx)
```

```
    UART1_Write(13)
```

```
UART1_Read_Text(txt, "OK", 10)
```

```
    UART1_Read_Text(txt, enter, 10)
```

```
next j
```

```
UART1_Write_Text("END")
```

```
UART1_Write(13)
```

```
UART1_Read_Text(txt, "EF", 10)
```

```
UART1_Read_Text(txt, enter, 10)
```

```
portb.7=1
```

```
delay_ms(1000)
```

```
portb.7=0
```

```
end sub
```

```
main:
```

```
ansel=0x04
```

```
anselh=0x00
```

```
trisb.7=0
```

```
trisb.6=0
```

```
trisb.5=0
```

```
trisb.1=1
```

```
trisb.0=0
```

```
portb.7=0
```

```
portb.6=0
```

```
portb.5=0
```

```
trisc.7=1
```

```
trisc.6=0
```

```
delay_ms(1200)      'Retardo para que xbee se encienda y se resetee
```

```
oscon=0
```

```
txt1="Modulos XBEE"
```

```
txt2="1-GUARDAR"
```

```
txt3="2-TX"
```

```
txt4="3-GLCD"
```

```
UART1_Init(9600)      ' Inicializar uart a 2400bps
```

```
    Delay_ms(100)
```

```
Lcd_Init()            ' Initialize Lcd
```

```
Lcd_Cmd(_LCD_CLEAR)  ' Clear display
```

```
Lcd_Cmd(_LCD_CURSOR_OFF) ' Cursor off
```

```
Lcd_Out(1,1,txt1)     ' Write text in first row
```

```
delay_ms(2000)
```

```
Lcd_Cmd(_LCD_CURSOR_OFF) ' Cursor off
```

```
Lcd_Out(1,1,txt2)     ' Write text in first row
```

```
Lcd_Out(1,12,txt3)    ' Write text in first row
```

```
Lcd_Out(2,1,txt4)     ' Write text in first row
```

```
delay_ms(2000)
```

```
while (true)

    if Button(PORTA, 0, 1, 0) then

        guardar_datalogger()

    end if

    if Button(PORTA, 1, 1, 0) then

        While(true)

            valor1=convertir()

            uart1_write(valor1)

            while (UART1_Data_Ready() = 0) 'pregunta si llegó dato

            wend

            uart_rd=Uart1_Read()

        wend

    end if

    if Button(PORTA, 3, 1, 0) then

        mostrar()

    end if

wend

end.
```

### **3.3.2 Módulo Xbee receptor**

El módulo será un Xbee de la Serie 1, con una antena tipo látigo (whip).

#### **3.3.2.1 Configuración**

Los parámetros a configurar fueron:

PANID: Dirección de la red= 3332; ambos módulos deben tener el mismo PANID y el mismo canal para que estén en la misma red y puedan comunicarse entre sí.

Canal =C (12)

MY: Dirección de origen de 16-bit =0; que es la dirección DL del transmisor

DL: Dirección de Destino =1; que es la dirección MY del transmisor.

Tasa de transferencia de datos seriales = 9600bps.

### **3.4 Tratamiento de errores e interferencia**

El módulo en sí, presenta resistencia a la interferencia, debido a la técnica de transmisión que usa, que es DSSS (Direct Sequence Spread Spectrum), mediante la cual matemáticamente se amplía el ancho de banda transmitido, y cuya información requiere conocer un código, clave o secuencia para su decodificación. Además presenta una ventaja ante los escuchas no autorizados, ya que sin la clave, la señal spread spectrum aparece como ruido. El mensaje se lo considera invisible ya que no se lo puede interceptar.

Adicionalmente para la verificación de la integridad de los datos, este módulo usa checksum, en donde toma los bytes de datos, sin incluir la cabecera, los suma y posteriormente lo resta de 0xFF. Para comprobar pues, la suma de los bytes de datos y el checksum deben dar 0xFF.



# Capítulo 4

## **4 PRUEBAS Y SIMULACIÓN**

En este último capítulo de nuestro trabajo, mostramos las pruebas que realizamos con los módulos para verificar la comunicación, bien sea con el Pc, y entre PICs, además de la simulación de nuestro proyecto.

### **4.1 Prueba de comunicación entre Xbee y PC**

Inicialmente, quisimos comprobar la comunicación de los módulos con la PC, usamos el software de configuración X-CTU. Para esto disponíamos de una tarjeta adaptadora USB, que se conectaba a un puerto del mismo tipo en la PC.

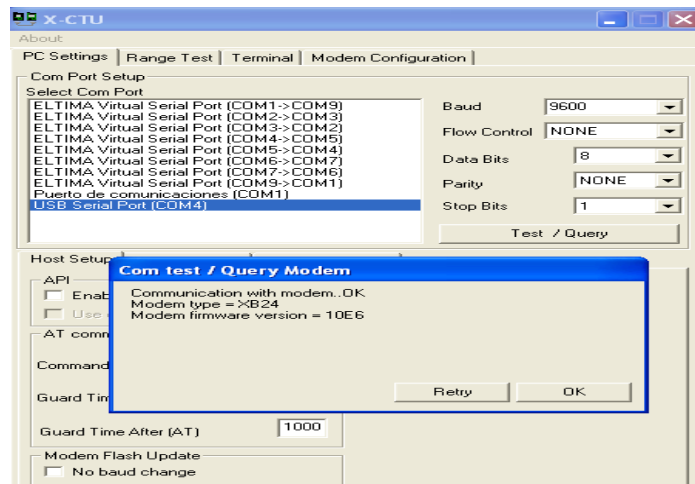


Fig 4.1: Prueba con X-CTU

La comunicación fue exitosa. En caso de falla se debía a que no se había instalado previamente un driver, que el puerto USB del PC estaba dañado, que el baud rate no era el correcto, o el módem estaba dañado.

## 4.2 Prueba de transmisión-recepción entre los Xbee mediante PCs

Se envió el texto “Prueba de comunicación exitosa”. Aquí se pudo observar que lo que se escribía en una PC en la pestaña Terminal del X-CTU en letras azules, se veía en la otra PC distante en letras rojas en la misma pestaña.

Esta prueba también se realizó con hyperterminal, dando los mismos resultados exitosos. Cabe recalcar que para su éxito, se necesito configurar

los parámetros de tasa de transferencia, bits de parada y demás, de igual manera en las dos PCs.

### **4.3 Prueba de transmisión-recepción entre los Xbee mediante PICs**

Hicimos un programa mediante el cual se enviaba la letra “a”, el receptor al recibir el dato encendía un led, y hacíamos una comprobación si el dato recibido era el esperado y encendíamos otro led.

Luego, el receptor enviaba el caracter ASCII siguiente respecto del recibido, para el caso de “a”, era “b”.

Una vez que el transmisor recibía este dato se encendía un led en uno de sus puertos.

Para observar esta comunicación, utilizamos nuevamente el PC, con un Xbee conectado mediante la tarjeta adaptadora USB y el otro colocado en el transmisor junto al PIC, para que el PC haga las veces del receptor.

Luego alternamos para que haga las veces de transmisor.

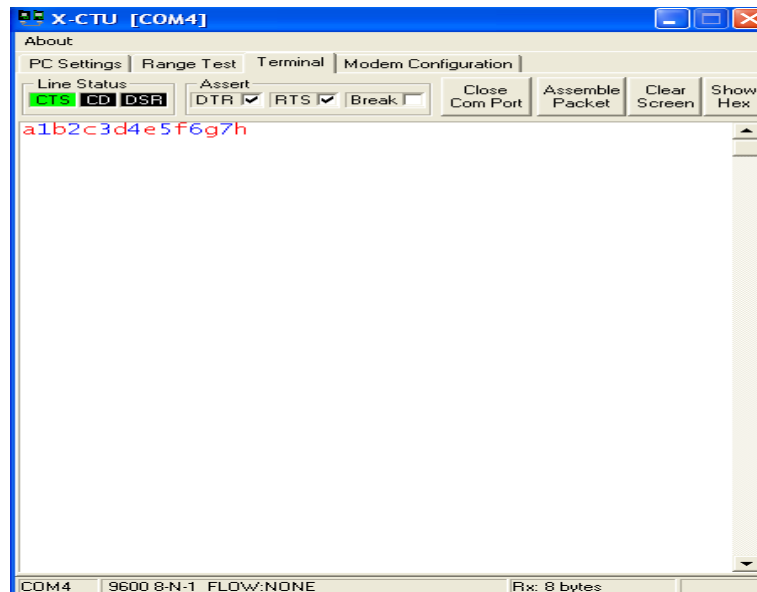


Fig 4.3: Prueba Tx-Rx entre Xbee con PICs

#### 4.4 Prueba de recepción de paquetes

Se realizó la prueba en el segundo piso del edificio de la Facultad de Ingeniería Eléctrica de la Espol, separando los dispositivos Xbee para verificar a qué distancia se pierde la comunicación.

Para la prueba se utilizó dos dispositivos Xbee, una laptop y una PC, conectados los xbee uno a cada una, mediante la tarjeta USB. En el software X-CTU se utilizó el modulo Range-test con su configuración por default. Esta configuración consiste en el envío de 32 bytes de datos de un dispositivo a otro el cual devuelve la trama de datos hacia el origen.

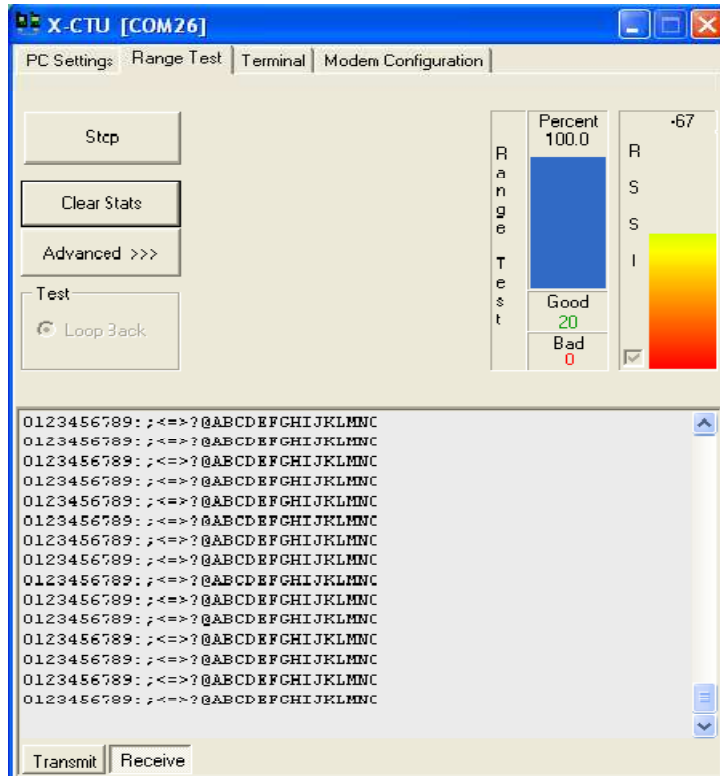


Fig 4.4.a: Paquetes recibidos correctamente

Vemos que un 100 % de los paquetes fueron recibidos, lo cual garantiza que a esa distancia tendremos una comunicación adecuada.

En cambio si aumentamos la distancia, los paquetes se comenzarán a perder.

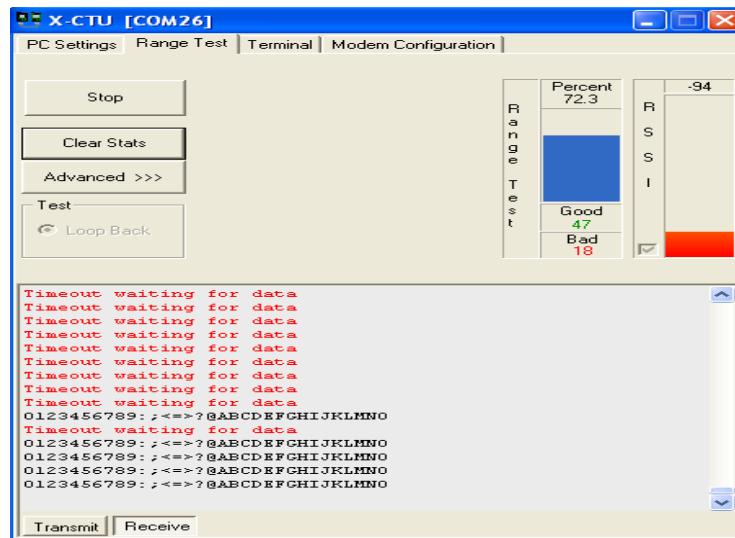


Fig 4.4.b: Pérdida de paquetes

## 4.5 Simulación Comunicación con datalogger

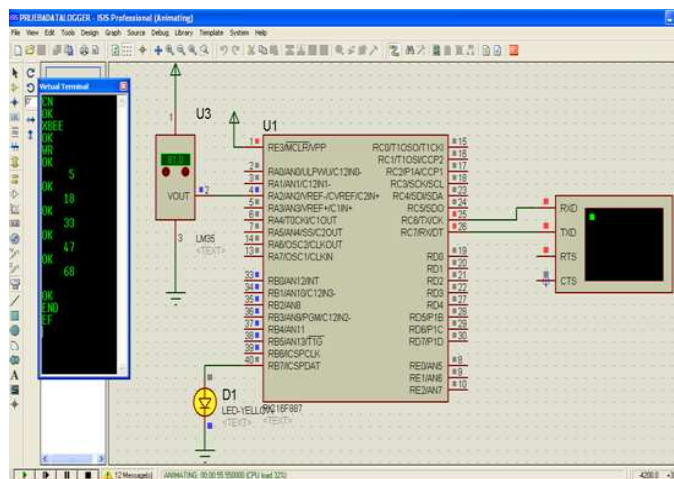


Fig 4.5: Comunicación con datalogger

# Conclusiones

1. Se pudo identificar que los módulos con tecnología Zigbee, son módulos inteligentes, que tienen una funcionalidad variada y amplia, a pesar del tamaño reducido que presentan, y permiten un ahorro significativo de energía. Los módulos Xbee de nuestro proyecto son de la serie 1, lo que implica lamentablemente que su capacidad es limitada en comparación con series más avanzadas, las cuales si permiten formar redes, el verdadero objetivo del protocolo Zigbee; en cambio en nuestro caso, solamente podemos realizar comunicación punto a punto o en estrella.
2. Durante la implementación del proyecto, pudimos notar que el módulo Xbee es muy sensible a fluctuaciones en la alimentación ya que los picos de voltaje de la fuente de alimentación afectaron al módulo a tal punto de dañarlos, y que además cuando el voltaje bajaba a niveles menores de 2.8V no funcionaba el módulo. Pudimos notar además que son de fácil configuración e implementación, ya que no requiere de mucha circuitería adicional, sólo adaptadores de voltaje a sus entradas; en nuestro caso con divisores de voltaje mediante resistores.

3. Los módulos Xbee son bastante inmunes al ruido, así lo pudimos comprobar en el laboratorio, cuando en cierto momento se conectaron varias fuentes de voltaje, unos 8 PCs, celulares con llamadas entrantes, además del WIFI, que podría interferir ya que trabaja en 2.4 Ghz, al momento de usar la red inalámbrica de la Espol, y los datos se transmitieron sin problema.
  
4. Se pudo comprobar que el alcance en interiores dado por la especificación del módulo, se aproxima a la medición real, donde la diferencia fue muy poca. Realizamos la medición en la Facultad de Electricidad de la Espol, y en esta se midió 24 m de alcance máximo, contra los 30 m especificados, para recibir todos los paquetes correctamente. Para exteriores se midieron 75 metros, frente a los 100 dados por la especificación.
  
5. Mediante las placas adaptadoras tanto para el protoboard, como para la conexión con un puerto USB de la PC, pudimos manipular de una manera más rápida y sencilla a los módulos Xbee, para cada una de las pruebas realizadas. La placa adaptadora del protoboard fue necesaria, ya que la separación entre los pines del módulo era más pequeña que aquella en el protoboard, además de permitirnos colocar y retirar los módulos conforme requeríamos. La placa adaptadora USB, en cambio nos facilitó, la comunicación con el PC para pruebas y configuración, sin necesidad de adaptar los niveles a RS-232 por nuestra cuenta.



# Recomendaciones

1. Al momento de alimentar el módulo, hacerlo mediante regulador y no directo desde una fuente de voltaje, debido a los picos presentados, que pueden afectar al equipo, procurando que sea siempre menor al máximo estipulado que es 3.3 V.
2. Es preferible utilizar un oscilador externo como un cristal, cuando se vaya a comunicar el PIC con el módulo, ya que son más exactos, además que preferible sea de frecuencia mayor conforme aumentemos la tasa de transferencia.
3. Cuando se lo quiera configurar mediante la aplicación X-CTU, verificar que se seleccionó el puerto correcto de la computadora, que el mismo no esté dañado, y que la aplicación tenga actualizada la lista de firmware de los módulos, para que sea compatible con el nuestro.
4. Para iniciar la comunicación serial, se debe dar un tiempo prudencial al módulo, al menos unos 500ms, para que se estabilice, antes de iniciar a transmitir. Verificar además que los parámetros de tasa de transferencia sean iguales tanto en ambos módulos Xbee, transmisor y receptor, como en el código de los PICs.

# ANEXO A

## COMANDOS AT

Comando	Rango	Descripción
<b>AT</b>		
<b>A1</b>	0 - 0x0F	Describe el modo de Asociación de un módulo. utilizado como Dispositivo Terminal (CE=0). Defecto=0.
<b>A2</b>	0 - 0x0F	Describe el modo de Asociación de un módulo utilizado como Coordinador (CE=1). Defecto=0.
<b>AC</b>	-	Aplica los cambios realizados explícitamente en l configuración.
<b>AP</b>	0 - 0x02	Habilita el modo de operación API. Defecto=0. 0 Modo API Deshabilitado. 1 Modo API habilitado. 2 Modo API habilitado con carácter de escape. Ajusta la tasa de transmisión entre el módulo y su cliente conectado a través de la interfaz serial. Para valores no-estándar revisar el manual. Defecto=3.
<b>BD</b>	0 - 0x07	0 1200 1 2400 2 4800 3 9600 4 19200 5 38400 6 57600 7 115200
<b>CC</b>	0 - 0xFF	Establece el carácter de secuencia a ser usado entre tiempos de esperas para entrar al modo de comandos. Defecto=0x2B ( carácter ASCII +)
<b>CH</b>	0x0B - 0x1A	Establece el canal por el cual se realiza la conexión RF entre módulos. Verificar Tabla 5-1 Frecuencia de Canales para configurar este parámetro. Defecto=0x0C.
<b>CE</b>	0 - 1	Indica el comportamiento del módulo. Defecto=0. 0 Dispositivo Terminal. 1 Coordinador.
<b>CN</b>	-	Sale del modo de Comando.
<b>D0-D4</b>	0 - 5	Ajusta la configuración de los pines I/O.

		0	Deshabilitado.
		1	--
		2	ADC.
		3	Entrada Digital.
		4	Salida Digital LOW.
		5	Salida Digital HIGH.
<b>D5</b>	0 - 5		Mismas funciones que D0 - D4, exceptuando lo siguiente:
<b>D6</b>	0 - 5	1	Indicador de Asociación. Mismas funciones que D0 - D4, exceptuando lo siguiente:
		1	Control de Flujo RTS.
		2	No tiene conversor ADC.
<b>D7</b>	0 - 5		Mismas funciones que D0 - D4, exceptuando lo siguiente:
		1	Control de Flujo CTS.
		2	No tiene conversor ADC.
<b>D8</b>	Sólo 0 y 3.		Ajusta la configuración del pin DI-8 (pin 9).
		0	Deshabilitado.
		3	Entrada Digital.
<b>DB</b>	0x17 - 0x5C (x-1dBm)		Lee la potencia de la señal del módulo del cual provino el último paquete RF recibido.
<b>DL</b>	0 - 0xFFFFFFFF		Ajusta los 32 bits menos significativos para direccionamiento. Defecto = 0.
<b>DH</b>	0 - 0xFFFFFFFF		Ajusta los 32 bits más significativos para direccionamiento. Defecto = 0.
<b>GT</b>	2 - 0x0CE4 (x 1 ms)		Tiempo de espera antes y después de ingresar el carácter de secuencia para entrar al modo de comandos. Defecto = 0x3E8.
<b>IA</b>	0 - 0xFFFFFFFF		Utilizado para crear el Cable Virtual. Indica la dirección del módulo de origen de los datos. Defecto= 0xFFFFFFFF (no permite el recibo de ningún paquete para cambiar las salidas.)
<b>ID</b>	0 - 0xFFFF		Ajusta la dirección PAN del módulo. Defecto = 0x3332
<b>IR</b>	0 - 0xFFFF (x1 ms)		Ajusta la tasa de muestreo de los pines I/O. Defecto = 0.
<b>IS</b>	1 - 0xFF		Forza al módulo a leer todos sus pines I/O. Si AP=0, el resultado se retorna del siguiente modo:
			- Número de Muestras.
			- Máscara de Canal.
			- Datos DIO.
			- Datos conversores ADC (se repite por cada conversor habilitado).
<b>IT</b>	1 - 0xFF		Número de muestras DIO y ADC que se deben

esperar, antes de transmitir. Defecto = 1.

<b>IO</b>	8 bits	Ajusta los niveles de las salidas digitales. Cada bits representa el nivel de los pines I/O configurados como salida .
<b>M0 - M1</b>	0 - 0x03FF	Ajusta el ciclo de trabajo de la salida PWM0 y PWM1. Si Mn=0 (0% PWM), Mn=0x01FF (50% PWM) y si Mn=0x03FF (100% PWM). Defecto=0. Configura la dirección de 16 bits para el módulo. Si My=0xFFFF o 0xFFFE, se habilita el modo de direccionamiento de 64 bit. Defecto = 0.
<b>MY</b>	0 - 0xFFFF	Ajusta la Paridad para la comunicación serial UART del módulo. Defecto =0 .
<b>NB</b>	0 - 4	<ul style="list-style-type: none"> <li>0 8 bit sin paridad o 7 bit con cualquier paridad.</li> <li>1 8 bit even.</li> <li>2 8 bit odd.</li> </ul>
<b>ND</b>	-	<ul style="list-style-type: none"> <li>3 8 bit mark.</li> <li>4 8 bit space.</li> </ul> <p>Reporta todos los dispositivos que se encuentren en el mismo canal y en la misma PAN que el módulo. El formato de respuesta es el siguiente cuando se encuentra en el modo Transparente.</p> <ul style="list-style-type: none"> <li>- MY (dirección de 16 bit)</li> <li>- SH (Serial Number High)</li> <li>- SL (Serial Nmuber Low).</li> <li>- DB (Fuerza de la señal proveniente de este módulo)</li> <li>- NI (Identificador del Nodo)</li> </ul>
<b>NI</b>	String de 20 caracteres ASCII	Define con un String el nodo o módulo .
<b>P0 - P1</b>	0 - 2	Configura el pin PWM0 y PWM1. Defecto P0 =1, Defecto P1=0.
<b>RE</b>	-	<ul style="list-style-type: none"> <li>0 Deshabilitado.</li> <li>1 RSSI.</li> <li>2 PWM habilitado.</li> </ul> <p>Restaura los valores de los parámetros a los valores por defecto que vienen de fábrica.</p>
<b>SM</b>	0 - 6	Configura el modo de operación SLEEP. Defecto = 0.
<b>SL</b>	0 - 0xFFFFFFFF	<ul style="list-style-type: none"> <li>0 Deshabilitado.</li> <li>1 Pin de Hibernado.</li> <li>2 Pin Doze.</li> <li>3 (reservado)</li> <li>4 Remoto Cyclic SLEEP.</li> <li>5 Remoto Cyclic SLEEP (con pin Wake-up).</li> <li>6 SLEEP Cordinador.</li> </ul> <p>Entrega los 32 bit menos significativos del Número Serial del módulo.</p>

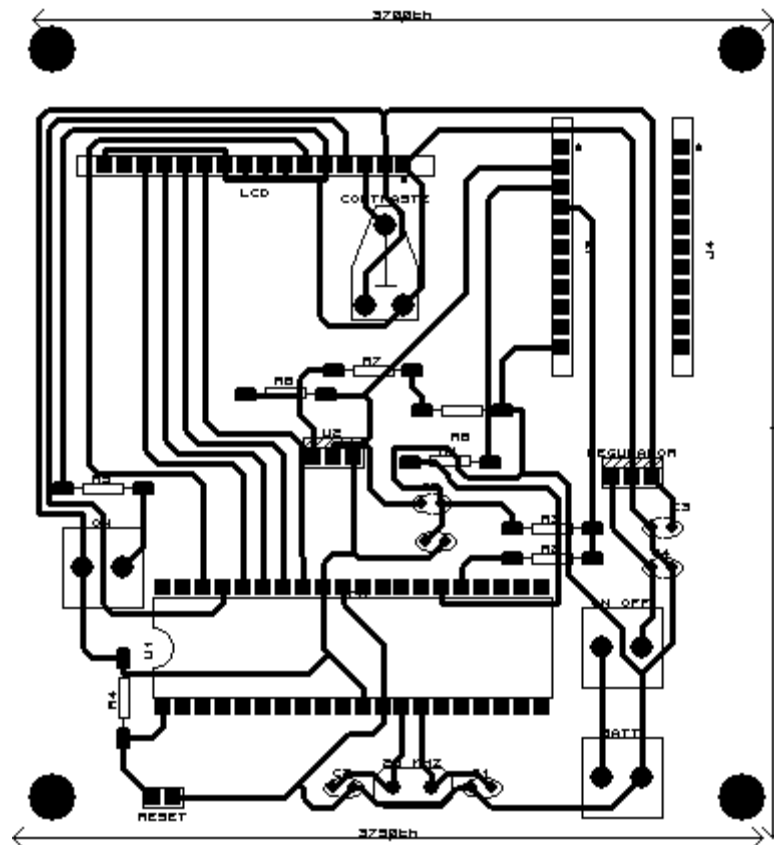
<b>SH</b>	0 - 0xFFFFFFFF	Entrega los 32 bit más significativos del Número Serial del módulo.
<b>SP</b>	1 - 0x68B0 (x10 ms)	Ajusta el tiempo de duración en que un módulo duerme o se mantiene en el modo SLEEP. Una vez terminado el período, busca por data entrante, si no hay nada vuelve a dormir y espera por un nuevo ciclo.
<b>ST</b>	1 - 0xFFFF (x1 ms)	Ajusta el tiempo de inactividad (datos ni recibidos ni enviados ya sea por RF o serial) antes de que el módulo ingrese al modo SLEEP. Defecto = 0x1388. Tiempo de espera de apagado para los Cables Virtuales. Si luego de este tiempo no hay cambios en alguna salida I/O, éstas cambian a su valor por defecto. Defecto = 0xFF.
<b>T0 - T7</b>	0 - 0xFF (x100 ms)	
<b>IU</b>	0 - 1	Habilita o no la salida I/O UART.
		0 Deshabilitado. Paquetes RF recibidos no serán enviados por UART.
		1 Habilitado. Paquetes RF recibidos serán enviados por UART.
<b>VL</b>	-	Entrega la versión del Firmware de forma Verbal. La respuesta entrega fecha de compilación de la aplicación, MAC, PHY y versión del bootloader y sus fechas de compilación.
<b>VR</b>	0 - 0xFFFF	Indica cual versión de firmware se encuentra actualmente en el módulo.
<b>WR</b>	-	Guarda en la memoria no-volátil del módulo, todos los valores de los parámetros.

Tabla 8-1 Tabla Resumen de Comandos.



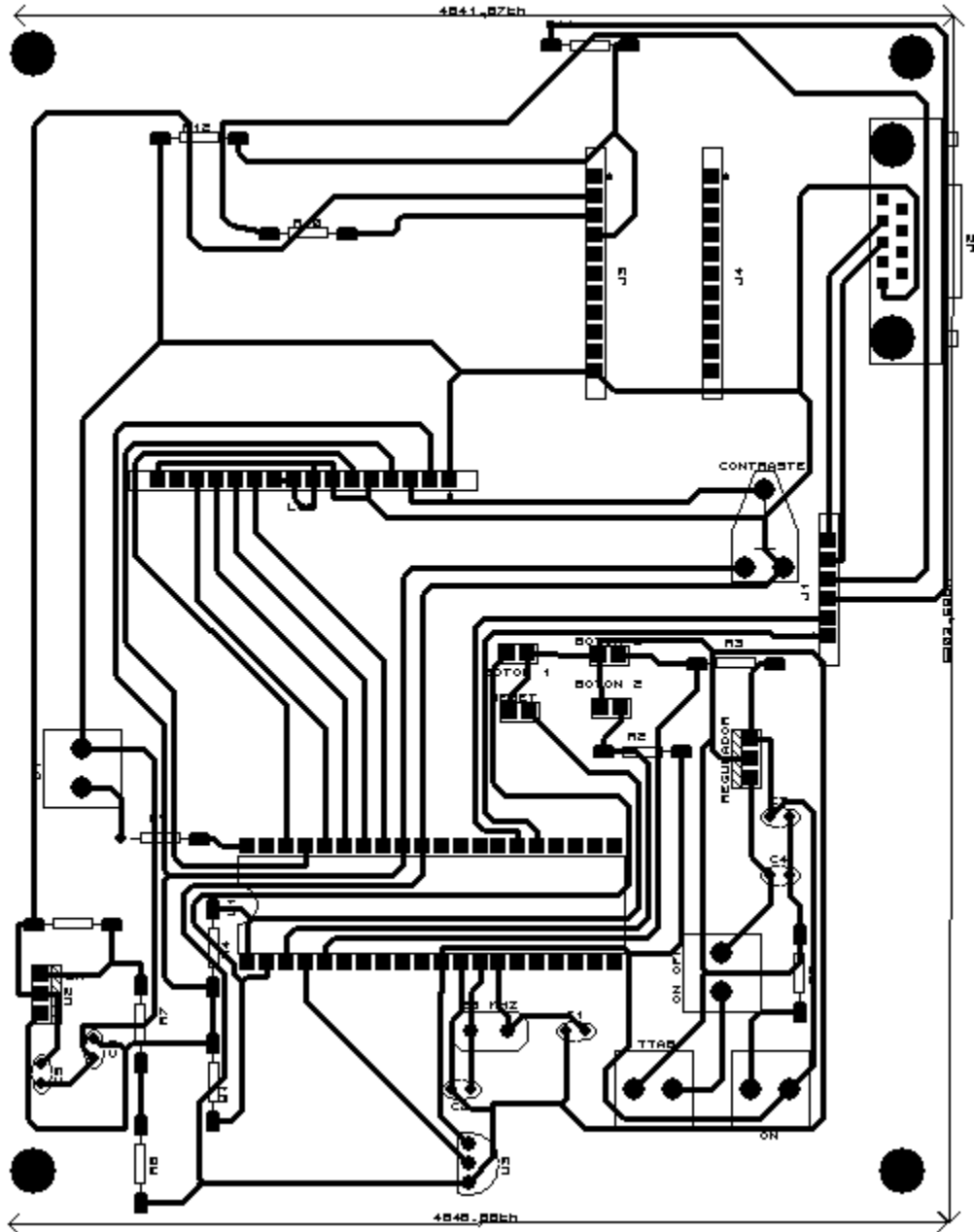


# PCB RX





# PCB TX



# BIBLIOGRAFÍA

- Protocolo Zigbee, Archivo HTML, <http://en.wikipedia.org/wiki/Zigbee>, 02 de marzo del 2010.
- Módulos Serie-1, Archivo HTML, <http://www.digi.com/products/wireless/point-multipoint/xbee-series1-module.jsp>, 24 de marzo del 2010.
- Manual del usuario Xbee, [http://www.makingthings.com/resources/datasheets/manual\\_xb\\_oem-rfmodules\\_802.15.4.pdf](http://www.makingthings.com/resources/datasheets/manual_xb_oem-rfmodules_802.15.4.pdf), 15 de abril del 2010.
- Iván Eleazar Cipriano, Proyecto ivki, Sistema para estadísticas e inventarios de una biblioteca utilizando escáneres de código de barras inalámbricos con Zigbee, <http://www.scribd.com/doc/17725240/Proyectoivki>, 22 de abril del 2010.

- Valverde J. , “ El estándar inalámbrico Zigbee”, Universidad Nacional de Trujillo, <http://seccperu.org/files/Zigbee.pdf>, 12 de abril del 2010.
- SINEM C. , “Zigbee 802.15.4, Summary” ,Ergen, 2004, [http://www.esi2.us.es/~fabio/apuntes\\_matlab.pdf](http://www.esi2.us.es/~fabio/apuntes_matlab.pdf), 1 de mayo del 2010.
- Jairo Santiago Paz, “Diseño y construcción de un modulo transmisor receptor inalámbrico para el manejo de sensores con el protocolo estándar IEEE 802.15.4 (Zigbee), Tesis de Grado, Universidad Tecnológica de la Mixteca, Octubre del 2008, Archivo PDF, 07 de mayo del 2010.
- Reinoso Pérez, Esteban Santiago,” Diseño e implementación de un prototipo para una red de domótica y seguridad para un hogar utilizando el estándar IEEE 802.15.4, EPN-Electrónica y Telecomunicaciones, Noviembre del 2008, Archivo PDF, 09 de mayo del 2010.
- Trabajo de Técnicas de Alta Frecuencia, Universidad Nacional de Ingenierías, Managua, Archivo PDF, <http://www.scribd.com/doc/4559979/Zigbee>, 11 de mayo del 2010.

- Página de DomoDesk: A fondo: ZigBee  
<http://www.domodesk.com/content.aspx?co=97&t=146&c>, 15  
de mayo del 2010.