

**ESCUELA POLITÉCNICA DEL EJÉRCITO**

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA EN  
AUTOMATIZACION Y CONTROL**

**PROYECTO DE GRADO PARA LA OBTENCIÓN DEL  
TÍTULO DE INGENIERÍA**

**DISEÑO E IMPLEMENTACIÓN DE UNA WSAN PARA  
UNA APLICACIÓN DOMÓTICA CON CAPACIDAD DE  
CONTROL REMOTO MEDIANTE MENSAJES SMS.**

**SORAYA NATHALIE RUIZ SALAZAR**

**SANGOLQUÍ – ECUADOR**

**2010**

## **CERTIFICACIÓN**

Certificamos que el presente proyecto de grado titulado: “Diseño e implementación de una WSAN para una aplicación domótica con capacidad de control remoto mediante mensajes SMS”, fue realizado en su totalidad por la señorita Soraya Nathalie Ruiz Salazar bajo nuestra dirección.

---

Ing. Román Lara  
DIRECTOR

---

Ing. Rodolfo Gordillo  
CODIRECTOR

## **RESUMEN**

El presente proyecto presenta el diseño e implementación de una WSN utilizando módulos de comunicación inalámbrica Xbee para una aplicación domótica con capacidad de control.

El proyecto fue dividido en tres etapas: la red WSN, la aplicación domótica y el control.

Para el desarrollo de la WSN se utilizó el kit de desarrollo de MAXSTREAM que contiene módulos de radio frecuencia Xbee.

Para la aplicación domótica se implementó un control de iluminación vía SMS a través del cual se puede seleccionar sus modos de activación y un sistema de detección de presencia mediante un sensor.

Complementariamente se implementó un control sobre la red, un prototipo de comunicación basado en la recepción de mensajes SMS, los mensajes fueron enviados desde un dispositivo móvil directamente por el usuario utilizando una interfaz de carácter demostrativa previamente instalada.

Finalmente se efectuaron las pruebas de funcionamiento del sistema implementado, uno de los resultados más relevantes fue la evolución del desempeño de este tipo de redes referente al alcance de lo cual se obtuvo que el alcance máximo en distancia permitido en interiores es de 15 metros dependiendo de la estructura de la edificación y 50 metros en exteriores con línea de vista.

## **DEDICATORIA**

A mis padres que fueron mi apoyo incondicional, quienes me hicieron creer en mí cuando pensaba que no podía, por ser mis amigos, por ser quienes siguieron mis pasos hasta formar una mujer de bien.

A mis hermanos que sirva como ejemplo que querer es poder que la vida nos pone obstáculos pero con la presencia del Dios en nuestra vida todo se hace más fácil.

## AGRADECIMIENTO

A Dios por ser mi motivación principal, por ser mi impulso a seguir y mantener mi fe por sobre todas las cosas, por guiarme cuando pensaba que todo estaba perdido, y ser quien escuchaba mis oraciones cuando más lo necesitaba.

A mis padres porque son mi mayor orgullo y mi ejemplo, por ser quienes estuvieron conmigo en todas las etapas de mi vida guiándome y fueron quienes me impulsaron a seguir siempre adelante, los que me enseñaron que la vida nos pone pruebas y hay que aprenderlas a superar con amor y dedicación.

A mis hermanos porque son mis confidentes, son los seres que llenan de alegría mi día a día.

Mi abuelito por apoyarme en todo lo que he necesitado, y ser mi mayor ejemplo de honestidad y trabajo.

A mis amigos y amigas porque son los hermanos que Dios olvidó darme, por ser quienes ponen el toque de alegría al transcurso de la mi vida, por ser los seres incondicionales que con tan solo escuchar solucionan todo.

A mis tutores, porque gracias a su valioso conocimiento supieron guiarme para finalizar con éxito este proyecto.

Y por último a una persona muy especial que supo ser mi apoyo incondicional al transcurso de una etapa en mi vida.

## PRÓLOGO

Los avances tecnológicos se han desarrollado a pasos agigantados en los últimos años. Estos avances no solo se han producido en los campos industriales, aeronáuticos, robóticos sino no también ha llegado al sector inmobiliario, concretamente con la domótica.

Considerando a la domótica como un término relativamente nuevo que se está posesionando de manera excepcional en la actualidad, su auge se debe principalmente a los beneficios que prestan estas en cuanto a bienestar, confort, seguridad y ahorro energético.

El desarrollo del proyecto se lo presentara dentro de cinco capítulos, distribuidos de la siguiente manera, el capítulo I consta de la presentación del proyecto, detallando claramente cuál es el esquema funcional al que se debe llegar al concluir el mismo.

El capítulo II contiene todo el fundamento teórico a utilizarse a lo largo de la realización.

A lo largo del capítulo III se especifica en detalle el diseño e implementación del prototipo receptor, la configuración de los módulos Xbee, diseño e implementación de la aplicación domótica e interfaz con el dispositivo móvil, por lo que se titulo materiales y métodos.

El capítulo IV contiene las pruebas y resultados respectivos y por último el capítulo V se detalla las conclusiones y recomendaciones del proyecto.

# CONTENIDO

CAPITULO I.....	16
PRESENTACIÓN DEL PROYECTO .....	16
1. 1 ANTECEDENTES. ....	16
1. 2 JUSTIFICACIÓN E IMPORTANCIA. ....	17
1. 3 ALCANCE DEL PROYECTO. ....	18
1. 4 OBJETIVO.....	19
1. 5 ZIGBEE EN DESARROLLO DEL PROYECTO.....	19
1. 6 Organización del Documento. ....	20
CAPÍTULO II.....	21
FUNDAMENTO TEÓRICO .....	21
2.1. SISTEMAS DOMÓTICOS .....	21
2.2. ZIGBEE .....	27
2.3. HISTORIA DE ZIGBEE.....	28
2.4. INTRODUCCION A LA COMUNICACIÓN INALAMBRICA .....	28
2.5. ZIGBEE COMO TECNOLOGIA INALÁMBRICA.....	29
2.1.1. Estándar IEEE 802.15.4 .....	30
2.1.2. Características de Zigbee.....	30
2.1.3. Ventajas .....	31
2.1.4. Desventajas .....	31
2.1.5. Tipos de dispositivos .....	32
2.1.5.1 Dispositivos Zigbee según su papel en la red .....	32
2.1.5.2 Dispositivos Zigbee según su funcionalidad.....	32
2.1.6. Tipos de Trafico de Datos .....	32
2.1.7. Arquitectura de Zigbee .....	33
2.1.8. Topología.....	35
2.1.9. Comparación de tecnologías inalámbricas .....	36
2.6. ZIGBEE EN APLICACIONES DOMÓTICAS .....	37
2.4.1 Servicios que ofrece Zigbee en el área domotica .....	38
2.7. ZIGBEE EN WSAN .....	39
2.7.1 WSAN .....	39
2.7.2 Características de WSAN .....	40
2.7.3 Arquitecturas de una WSAN .....	41

2.7.4	Aplicaciones de una WSN [5].....	41
2.8.	MÓDULOS DE MAXSTREAM.....	42
2.6.1	Módulos Xbee.....	42
2.6.2	Funcionamiento de los módulos Xbee .....	43
2.6.3	Conexión Básica .....	43
2.6.4	Configuración de Pines de módulo Xbee .....	44
2.6.5	Modos de operación. ....	47
2.9.	PROTOTIPO RECEPTOR .....	48
2.7.1	Microcontrolador .....	48
2.7.2	Microcontrolador ATMEL .....	48
2.7.3	Microcontrolador ATmega164 .....	49
2.7.4	Definición de GSM .....	52
2.7.4.1.1	Arquitectura de la red de telefonía móvil GSM .....	52
2.7.4.1.1.1	Subsistema de red y conmutación (NSS).....	53
2.7.4.1.1.2	Subsistema de Estación base (BSS) .....	54
2.7.4.1.1.3	Subsistema de Estación móvil (MS).....	54
2.7.5	SERVICIO SMS .....	55
2.7.5.1	Arquitectura de la red. ....	56
2.7.6	INTERFAZ CON MODEMS .....	58
2.7.7	INTERFAZ CON MÓDEMS GSM.....	59
2.7.7.1	Comandos AT+.....	60
2.7.8	DISEÑO DE LA INTERFAZ PARA EL DISPOSITIVO MÓVIL. ....	61
2.7.9	ENTORNO DE EJECUCIÓN .....	61
2.7.9.1	Máquina virtual.....	62
2.7.9.1.1	KVM.....	62
2.7.9.1.2	CVM.....	63
2.7.9.2	MIDLET .....	64
2.7.9.2.1	Descripción del MIDlet .....	65
2.7.9.2.2	Estados de un MIDlet .....	65
2.7.10	RECORD MANAGEMENT SYSTEM .....	66
2.7.11	WIRELESS MESSAGING API .....	67
2.7.12	COMPONENTES APLICACIONES J2ME .....	70
2.7.13	INTRODUCCIÓN A NETBEANS.....	71



2.7.13.1	Objetos.....	73
2.7.13.2	Displayables.....	73
2.7.13.3	Commands .....	74
2.7.13.4	Items .....	75
2.7.13.5	Flujo .....	76
CAPITULO III .....		77
MATERIALES Y MÉTODOS.....		77
3.1	PROTOTIPO RECEPTOR .....	77
3.1.1	Descripción del prototipo receptor.....	77
3.2	HARDWARE DEL PROTOTIPO RECEPTOR.....	78
3.2.1	Microcontrolador ATmega164 .....	78
3.2.2	Modem GSM .....	79
3.3	SOFTWARE DEL PROTOTIPO RECEPTOR.....	82
3.3.1	Programa del microcontrolador .....	82
3.4	DISEÑO DE LA WSAN.....	84
3.4.1	Programación de módulos Xbee .....	84
3.4.2	Instalación del programa X-CTU.....	84
3.4.3	Configuración del Puerto X-CTU .....	86
3.4.3.1	Módulos Xbee PRO Serie 2.....	86
3.4.3.2	Módulos Xbee series 2 .....	88
3.4.3.3	Programación para el módulo Xbee PRO series 2. ....	90
3.4.3.4	Para el módulo Xbee Series 2 .....	92
3.5	PROTOTIPO RECEPTOR CON INTERCONEXION DE TARJETAS XBEE .....	97
3.6	IMPLEMENTACIÓN DE LA APLICACIÓN DOMÓTICA.....	98
3.7	INTERFAZ GRÁFICA PARA EL DISPOSITIVO MÓVIL. ....	99
3.7.1	Descripción.....	99
3.7.2	Desarrollo .....	99
3.7.3	Código de configuración .....	100
CAPITULO IV .....		102
PRUEBAS Y RESULTADOS.....		102
4.1	PRUEBAS DEL PROTOTIPO. ....	102
4.1.1	Descripción y prueba del bloque 1.....	102
4.1.2	Descripción y prueba del bloque 2.....	103

4.1.2.1 Pruebas y resultados del microcontrolador ATmega164.....	103
4.1.2.2 Modem GSM .....	105
4.1.2.3 Descripción y prueba bloque 3 .....	108
4.1.2.4 Descripción y prueba bloque 4 .....	109
CAPITULO V.....	112
CONCLUSIONES Y RECOMENDACIONES .....	112
5.1 CONCLUSIONES.....	112
5.2 RECOMEDACIONES.....	113
REFERENCIAS BIBLIOGRÁFICAS.....	114
ANEXO 1 .....	115
PROGRAMA DE LOS MICRONTROLADORES ATmega16.....	115
ANEXO 2 .....	126
MANUAL DEL SOFTWARE X-CTU .....	126
ANEXO 3 .....	146
MANUAL DE LOS MÓDULOS XBEE/XBEE PRO.....	146

## INDICE DE TABLAS

<b>Tabla 2. 1. Tabla de comparación de tecnologías. ....</b>	<b>36</b>
<b>Tabla 2. 2. Pines del módulo Xbee.....</b>	<b>45</b>
<b>Tabla 2. 3. Clases en javax.wireless.messaging. ....</b>	<b>69</b>
<b>Tabla 4. 1 Resultados de funcionamiento del microcontrolador ATmega 164 .....</b>	<b>105</b>
<b>Tabla 4. 2 Resultados de funcionamiento del modem GSM.....</b>	<b>106</b>
<b>Tabla 4. 3 Resultados de funcionamiento del la comunicación inalámbrica. ....</b>	<b>109</b>
<b>Tabla 4. 3 Resultados de funcionamiento del prototipo final.....</b>	<b>111</b>

## INDICE DE FIGURAS

<b>Figura 2. 1. Sistemas Domótico.....</b>	<b>21</b>
<b>Figura 2. 2 Áreas de la domótica .....</b>	<b>22</b>
<b>Figura 2. 3. Arquitectura domótica descentralizada .....</b>	<b>25</b>
<b>Figura 2. 4. Arquitectura domótica centralizada.....</b>	<b>26</b>
<b>Figura 2. 5. Sistema cableado .....</b>	<b>26</b>
<b>Figura 2. 6. Sistema inalámbrico .....</b>	<b>27</b>
<b>Figura 2. 7. Áreas de integración Zigbee.....</b>	<b>27</b>
<b>Figura 2. 8. Tipos de Redes Inalámbricas. ....</b>	<b>29</b>
<b>Figura 2. 9. Zigbee spectrum .....</b>	<b>30</b>
<b>Figura 2. 10. Diferentes capas que conforman la pila de protocolos para ZigBee.....</b>	<b>33</b>
<b>Figura 2. 11. Aplicaciones de Zigbee .....</b>	<b>37</b>
<b>Figura 2. 12. Diseño funcional de sensor y actuador dentro de una red .....</b>	<b>40</b>
<b>Figura 2. 13. Zócalo demostrativo para módulo Xbee.....</b>	<b>43</b>
<b>Figura 2. 14. Conexiones mínimas requeridas para un modulo Xbee.....</b>	<b>43</b>
<b>Figura 2. 15. Diagrama de pines del modulo Xbee, vista superior .....</b>	<b>44</b>
<b>Figura 2. 16. Módulos Xbee .....</b>	<b>46</b>
<b>Figura 2. 17. Módulos Xbee PRO .....</b>	<b>46</b>
<b>Figura 2. 18. Distribución de pines ATmega164 .....</b>	<b>50</b>
<b>Figura 2. 19. Arquitectura de red GSM.....</b>	<b>53</b>
<b>Figura 2. 20. Servicio SMS.....</b>	<b>56</b>
<b>Figura 2. 21. Servicios básicos SM MT y SM MO. ....</b>	<b>57</b>
<b>Figura 2. 22. Niveles y servicios para el envío de mensajes cortos. ....</b>	<b>58</b>
<b>Figura 2. 23. Estados de un MIDlet.[10] .....</b>	<b>65</b>
<b>Figura 2. 24: Aplicación Modular Netbeans. ....</b>	<b>72</b>
<b>Figura 2. 25: Ejemplo de Aplicación Modular Netbeans.....</b>	<b>73</b>
<b>Figura 3. 1. Etapas del receptor.....</b>	<b>77</b>
<b>Figura 3. 2. Distribución de pines ATmega164 .....</b>	<b>78</b>
<b>Figura 3. 3. Nokia 3220 .....</b>	<b>79</b>
<b>Figura 3. 4. Cable DKU-5 .....</b>	<b>80</b>
<b>Figura 3. 5. Puerto del teléfono Nokia.....</b>	<b>80</b>
<b>Figura 3. 6. Conexión entre el módem GSM y el microcontrolador ATmega16. ....</b>	<b>82</b>
<b>Figura 3. 7. Pantalla de recolección de información.....</b>	<b>85</b>
<b>Figura 3. 8. Modulo Xbee PRO serie2.....</b>	<b>86</b>
<b>Figura 3. 9. Modulo Xbee Series 2.....</b>	<b>88</b>
<b>Figura 3. 10. Pantallas de inicialización de X-CTU .....</b>	<b>89</b>
<b>Figura 3. 11. Prueba de conectividad .....</b>	<b>90</b>
<b>Figura 3. 12. Configuración del modo Coordinador y el Router.....</b>	<b>91</b>
<b>Figura 3. 13. Direccionamiento de los módulos Xbee PRO series 2 .....</b>	<b>92</b>
<b>Figura 3. 14. Direccionamiento de los módulos Xbee series 2 .....</b>	<b>93</b>

<b>Figura 3. 15. Opción Write</b> .....	93
<b>Figura 3. 16. Modo terminal</b> .....	94
<b>Figura 3. 17. Falla de transmisión y recepción</b> .....	95
<b>Figura 3. 18. Transmisión de paquetes</b> .....	96
<b>Figura 3. 19. Recepción de paquetes</b> .....	96
<b>Figura 3. 20. Prototipo receptor coordinador Xbee</b> .....	97
<b>Figura 3. 21. Prototipo Router Xbee</b> .....	98
<b>Figura 3. 22. Red inalámbrica bidireccional</b> .....	98
<b>Figura 4. 1. Sistema de Control Domótico: Diseño funcional por bloques</b> .....	102
<b>Figura 4. 2. Circuito de prueba de funcionamiento del ATmega16.</b> .....	104
<b>Figura 4. 3. Simulación de activación del sensor</b> .....	104
<b>Figura 4. 4. Cable DKU-5</b> .....	105
<b>Figura 4. 5. Pantalla de la configuración de la conexión completa</b> .....	106
<b>Figura 4. 6. Coordinador Xbee</b> .....	107
<b>Figura 4. 7. Conexión del Nokia 3220 a la placa coordinador Xbee</b> .....	107
<b>Figura 4. 8. Comunicación Inalámbrica Xbee</b> .....	108
<b>Figura 4. 9. Router Xbee</b> .....	109
<b>Figura 4. 10. Diseño funcional por bloques del prototipo implementado</b> .....	110

## GLOSARIO

**ACK:** Acknowledgement

**AMS:** Application Management System

**API:** Application Programming Interface

**API:** Application Programming Interface

**BSC:** Base Station Controller

**BSS:** Base Station Subsystem

**BSS:** Subsistema de estación base.

**BTS:** Base transceiver station

**DL:** Destination Address.

**ETSI:** European Telecommunications Standard Institute

**GMSC:** Gateway Mobile Switching Centre.

**GSM:** Global System for Mobile communications

**HLR:** Home Location Register

**ISM:** Industrial, Scientific & Medical

**JVM:** Java Virtual Machine.

**MS:** Estación móvil.

**MSC:** Network Switching Subsystem

**MT Short Message Mobile:** Terminated Point-to-Point

**NSS:** Subsistema de red y comunicación

**RAM:** Random Access Memory

**RMI:** Remote Method Invocation

**RMS:** Record Management System

**ROM:** Read Only Memory

**SL:** Serial Number.

**SM MO:** ShortMessageMobile Originated Point-to-Point

**SM-AL:** Short Message Application Layer.

**SM-LL:** Short Message Lower Layers.

**SM-RL:** Short Message Relay Layer.

**SM-TL:** Short Message Transfer Layer.

**TPDU:** Transfer Protocol Data Units.

**USART:** Universal synchronous asynchronous receiver transmitter

**VLR:** Visitor Location Register.

**WSAN:** Wireless Sensor and Actor Networks

**ZC:** Zigbee Coordinator

**ZDO:** ZigBee Device Objects

**ZED:** Zigbee end device

**ZR:** Zigbee Router

# CAPITULO I

## PRESENTACIÓN DEL PROYECTO

### 1.1 ANTECEDENTES.

Los avances tecnológicos se han desarrollado a pasos agigantados en los últimos años. Estos avances no solo se han producido en los campos industriales, aeronáuticos, robóticos, sino también ha llegado al sector inmobiliario, concretamente con la domótica.

La domótica es un término relativamente nuevo que se está posesionando de manera excepcional en la actualidad, su auge se debe principalmente a los beneficios que prestan estas en cuanto a bienestar, confort, seguridad y ahorro energético

Las aplicaciones de la domótica no son nuevas, desde hace algún tiempo ya ha existido las denominadas “casas inteligentes”, las cuales brindan servicios de automatización de iluminación, aire acondicionado, persianas, control de accesos, etc. Pero las aplicaciones de la domótica se basan en la integración de todos estos sistemas.

Durante el desarrollo de la domótica se ha visto la necesidad de interactuar con dispositivos de control remoto en el hogar. Hace unos años, los mandos de TV por infrarrojo eran los únicos dispositivos de control remoto en nuestros hogares. En la actualidad podemos controlar remotamente en nuestra casa: la TV, el aire acondicionado, la iluminación, la simulación de presencia, el DVD, el Home Cinema, puertas del auto, el mando del garaje, la alarma, etc. Para interactuar remotamente con todos estos dispositivos, necesitamos trabajar con un solo estándar para tenerlos bajo una misma red, específicamente en nuestro hogar. Ahora bien, las tecnologías que la domótica han adoptados para realizar este control son WiFi, Bluetooth, WiMAX, USB inalámbrico, etc. Las cámaras *Wireless*, destacadas por el control remoto, son un ejemplo de cómo se pueden aplicar estas tecnologías para la domótica y el control de áreas. Pero el problema es que estas tecnologías no satisfacen los requerimientos de la domótica, porque su arquitectura no pensó en ello cuando fueron creadas.



Zigbee es un estándar de comunicaciones vía radio y bidireccional, que puede ser utilizado con dispositivos de domótica, automatización de edificios (Inmótica), control industrial, periféricos de PC, juguetería, sensores médicos. Además, Zigbee fue creado para cubrir la necesidad del mercado de un sistema a bajo costo, un estándar para redes *Wireless* de pequeños paquetes de información, bajo consumo, seguro y fiable.

## 1.2 JUSTIFICACIÓN E IMPORTANCIA.

El apareamiento de conceptos como la domótica en conjunto con el desarrollo de Redes Inalámbricas WSN (*Wireless Sensor and Actor Networks*), genera oportunidades propicias para el desarrollo tecnológico de nuevos proyectos considerándolo como un problema vigente de investigación.

Zigbee presenta las siguientes ventajas: tiene un menor consumo energético, un tiempo de respuesta instantáneo, permite la comunicación inalámbrica omnidireccional<sup>1</sup> fiable y de dos vías, agilidad de canales para una mejor coexistencia con otras tecnologías inalámbricas de 2,4 GHz, permite una instalación y configuración sencilla.

Por las ventajas que presenta Zigbee despertó el interés de la realización del proyecto, esta investigación aplicada se realizará con el fin de dar a conocer cómo se puede aplicar el protocolo inalámbrico Zigbee en una WSN dentro del campo de la domótica como tal, buscando mejorar el confort del usuario mediante el uso de esta nueva tecnología.

Para concebir la implementación del proyecto se dispone de un Kit de desarrollo "Digi" de MAXSTREAM que contiene módulos de radio frecuencia Xbee adquirido por la ESPE a través de los proyectos de investigación científica del DEEE. Recurso que permitirá realizar la comunicación inalámbrica del proyecto en cuestión, mientras que, para la codificación será factible el uso de un PIC con su respectiva programación.

Los resultados que se desprendan de la realización de este proyecto permitirán determinar los alcances, limitaciones y potencialización de Zigbee como protocolo inalámbrico dentro del área de la domótica, además de disponer de una plataforma base hacia la incursión en nuevos proyectos relacionados.

---

<sup>1</sup> antena omnidireccional puede orientar o utilizar en cualquier dirección o sentido

### 1.3 ALCANCE DEL PROYECTO.

Este proyecto pretende diseñar e implementar una WSN utilizando módulos de comunicación inalámbrica Xbee para una aplicación domótica con capacidad de control.

El desarrollo del proyecto constará tres etapas: la red WSN, la aplicación domótica y el control.

La red WSN es un grupo de sensores y actuadores vinculados por un medio inalámbrico para realizar detección distribuida y las tareas de actuación, esta red será la encargada de realizar la comunicación inalámbrica para la aplicación domótica.

Para la implementación de la WSN se dispone de un el Kit de desarrollo “Digi” de MAXSTREAM que contiene módulos de radio-frecuencia Xbee

Para la aplicación domótica se pretende implementar un control de iluminación vía SMS a través del cual se pueda seleccionar sus modos de activación.

Este dispositivo permitirá al usuario activar o desactivar un determinado actuador y mostrar la detección de cualquier cambio del estado de un sensor.

Complementariamente se implementará un control sobre la red, constará de un prototipo de comunicación basado en la recepción de mensajes SMS.

Los mensajes serán enviados desde un dispositivo móvil directamente por el usuario utilizando una interfaz de carácter demostrativa previamente instalada, esta solicitud será recibida a través de un dispositivo de enlace y se codificará mediante el uso de un microcontrolador.

La señal obtenida será enviada a la red WSN, activando o desactivando los actuadores que se encuentren conectados a la misma. Brindando al usuario con capacidad de tener control sobre el sistema domótico del hogar.

Para solventar el alcance del proyecto se presenta los siguientes objetivos.

## 1.4 OBJETIVO

### 1.4.1 General

Diseñar e implementar una WSN para validar su utilización en una aplicación domótica

### 1.4.2 Específicos

- Analizar el estado del arte de la utilización de Zigbee como protocolo de comunicaciones en aplicaciones de domótica.
- Diseñar y evaluar el funcionamiento de una WSN dentro de una aplicación domótica.
- Dotar de capacidad de control a la aplicación domótica a través de mensajes SMS utilizando la tecnología GSM.
- Determinar alcances y limitaciones de la aplicación y red implementadas.

## 1.5 ZIGBEE EN DESARROLLO DEL PROYECTO

El proyecto pretende diseñar e implementar una WSN utilizando módulos de comunicación inalámbrica Xbee para una aplicación domótica con capacidad de control.

Para una mejor concepción del proyecto se puede observar el diseño funcional por bloques que muestra la figura 1.1.

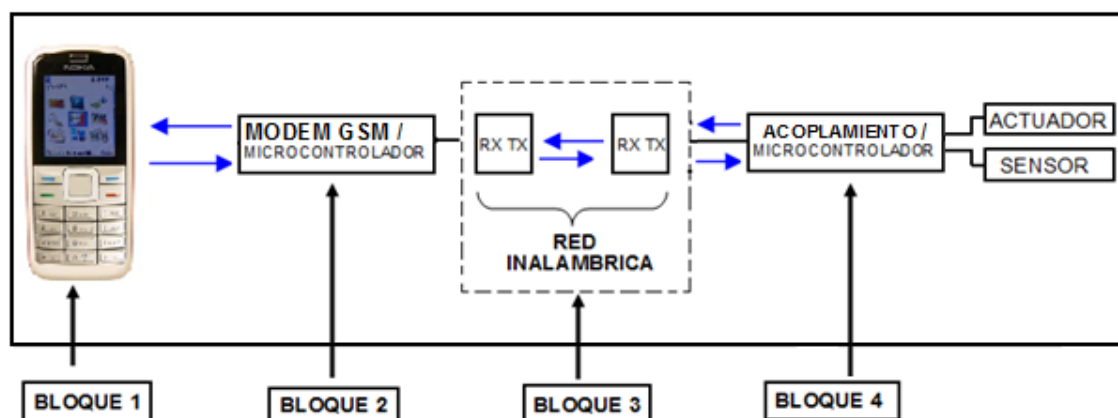


Figura 1. 1. Sistema de Control Domótico: Diseño funcional por bloques

- **Bloque 1.** Programación en J2ME para dispositivos móviles.  
Este bloque contiene un teléfono celular en donde se encuentra instalada la aplicación para realizar el control de la red.
- **Bloque 2.** Este bloque contiene dos partes.
  - El Modem GSM (Global System for Mobile communications)
  - Programación del Microcontrolador para transmisión y recepción de SMS.
- **Bloque 3.** Programación de los módulos Xbee para la comunicación inalámbrica.
- **Bloque 4.** Diseño e implementación del circuito de acoplamiento y programación del microcontrolador para atender peticiones de transmisión y recepción.

Finalmente se efectuarán las pruebas de funcionamiento del sistema implementado para determinar los alcances y limitaciones del mismo.

## 1.6 Organización del Documento.

El desarrollo del proyecto se lo presentará dentro de cinco capítulos, distribuidos de la siguiente manera:

CAPÍTULO I- Presentación de proyecto. Consta de la presentación del proyecto, detallando claramente cuál es el esquema funcional al que se debe llegar al concluir el mismo.

CAPÍTULO II - Fundamento teórico. Contiene todo el fundamento teórico a utilizarse a lo largo de la realización.

CAPÍTULO III -Materiales y Métodos. Se especifica en detalle el diseño e implementación del prototipo receptor, la configuración de los módulos Xbee, diseño e implementación de la aplicación domotica e interfaz con el dispositivo móvil.

CAPÍTULO IV –Pruebas y Resultados. Contiene las pruebas y resultados respectivos

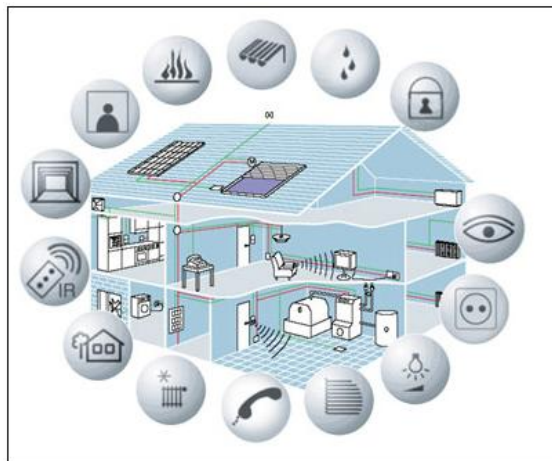
CAPÍTULO V- Conclusiones y Recomendaciones. Una vez finalizado el proyecto se detalla las conclusiones y recomendaciones del mismo.

## CAPÍTULO II

### FUNDAMENTO TEÓRICO

#### 2.1. SISTEMAS DOMÓTICOS

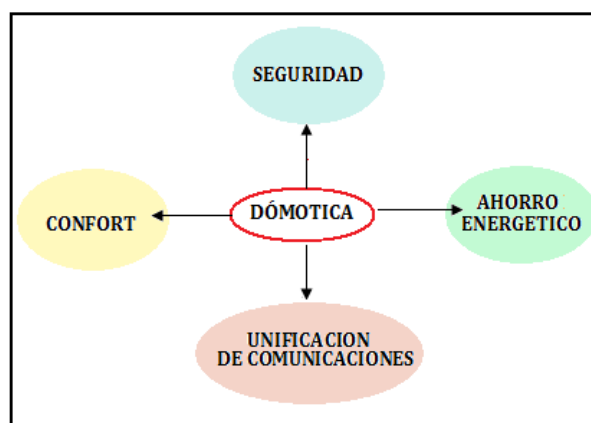
Los avances tecnológicos se han desarrollado a pasos agigantados en los últimos años. Estos avances no solo se han producido en los campos industriales, aeronáuticos, sino también ha llegado al sector inmobiliario, en el sector domótico como muestra la figura 2.1



**Figura 2. 1. Sistemas Domótico**

El término domótica proviene de la unión de las palabras *domus* (que significa casa en latín) y robótica (de *robota*, que significa esclavo). Se entiende por domótica al conjunto de sistemas capaces de automatizar una vivienda y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, con su respectivo control [1].

En la figura 2.2 muestra las cuatro áreas que se aplica la domótica:



**Figura 2. 2 Áreas de la domótica**

- **Confort**

Mejor calidad de vida, se busca a través de la automatización del control de luces, persianas, ventanas, cortinas y enchufes, la gestión automática de la climatización automática tanto por calefacción como por refrigeración, la automatización de tareas, riego automático la encendida de los aparatos en función de las tarificaciones horarias o de acontecimientos, con la ayuda de sondas de temperatura, luz y humedad situadas en el interior y el exterior de la vivienda, electroválvulas por agua y reguera, y de módulos de relés de potencia para el control de persianas, aire acondicionado o conectar los electrodomésticos como frigorífico, lavadora, lavaplatos, secadora, placa de cocción, horno, microondas, cafetera o tostadora.

- **Seguridad**

Aumento de la seguridad en el hogar, en el apartado de la seguridad incluye las alarmas, la gestión de servicios para emular la presencia de gente durante las ausencias prolongadas, el control de accesos, control biométrico, con la posibilidad de visualización remota de la vivienda y la ayuda de sensores de presencia y sensores magnéticos de apertura de puertas o ventanas, detectores de correspondencia al buzón, servicios de telealarma por emergencias médicas, control de temperatura, detección de escapes de gas o agua, o las alarmas técnicas por corte de suministros.

- **Ahorro energético**

- **Doble tarifa.** Las compañías eléctricas ofrecen dos tarifas, de la que la nocturna supone un ahorro de hasta el 55% en el kW/h.  
Así, se puede configurar el sistema domótico para que se adapte al horario nocturno y conecte determinados electrodomésticos (como la lavadora) durante la noche, o que acumule energía calorífica durante la noche para liberarla después en el transcurso del día.
- **Sistema de acumulación.** almacenan energía de la franja horaria más económica y la distribuyen durante el resto del día, normalmente en forma de calor ya que son fáciles de encontrar en calefacción o en calentamiento de agua. Este proceso se realiza de forma programada y utilizando temporizadores.
- **Racionalizadores de consumo.** tienen como finalidad desconectar una parte de los circuitos no primordiales para que la potencia consumida no supere a la contratada, lo que llevaría a una desconexión de todos los circuitos. Este sistema permite por tanto contratar una cantidad menor de potencia, reduciendo así los gastos de contratación.
- **Climatización.** se agrupan las diferentes estancias de la casa en dos tipos, dependiendo de cuando se frecuenten más, de noche o de día. Así, el sistema se encarga de modificar la temperatura para cada una de ellas, aumentando por ejemplo la temperatura en las habitaciones por la noche y disminuyendo la del salón.
- **Iluminación.** utilizando unos sensores, se puede graduar la intensidad de la luz, encenderse automáticamente o bien apagarse si resulta innecesaria, con el fin de consumir la menor cantidad de energía sin producir alguna molestia al usuario.
- **Teléfono.** el sistema puede prohibir llamadas según vayan destinadas al área urbana, metropolitana, nacional, internacional o a móviles con tal de evitar el abuso del aparato en caso de no estar el propietario presente. A su vez, puede limitar la

duración de la llamada, así como bloquear las llamadas salientes de modo que solo se puedan recibir llamadas.

- **Estadísticas.** el sistema recopila una serie de datos y almacena determinadas informaciones que posteriormente el usuario puede consultar, obteniendo así datos sobre el consumo de luz o de teléfono, dónde se ha producido un problema, cuál ha sido su causa.
- **Comunicaciones**
  - **Conexión remota al equipamiento.** Desde cualquier teléfono (fijo, móvil, cabina...) se puede conectar con el sistema domótico para activar o desactivar los servicios deseados, de la misma manera que el sistema llama en caso de accidente o peligro. También se puede abrir la puerta mediante un mando a distancia.
  - **Distribución de audio y vídeo.** Se puede hacer que la música de un reproductor situado en una habitación se escuche en toda la casa o ver en la televisión la señal del vídeo-portero.
  - **Redes telemáticas.** Estas redes permiten llevar a cabo una gran variedad de acciones sin moverse de casa, como la compra a distancia, realizar operaciones financieras, enviar mensajes a cualquier persona del mundo... opciones que día a día se van ampliando.

El sistema domótico está compuesto por tres elementos principales:

- **Sensores.** Son los encargados de captar cualquier tipo de cambio físico en el interior de la vivienda y transmitir la información a la unidad de control para que ésta actúe convenientemente.
- **Actuadores.** Son los aparatos que, mediante las órdenes que la unidad de control le da, actúa y transforma aquellos datos con acciones físicas (subir persianas, realizar una llamada)



- **Unidad de control.** Es el componente más importante del sistema, ya que es la parte encargada de gestionar la información y enviar los datos necesarios hacia el actuador para resolver los problemas. Suele tener los interfaces de usuario necesarios para presentar la información a este (pantalla, teclado, monitor, etc.).

Dependiendo de las características, el sistema domótico utilizado puede ser:

- **Descentralizado.** los sensores y los actuadores poseen unos pequeños sistemas de autocontrol adaptados que permiten la interacción directa entre unos y otros como muestra la figura 2.3.

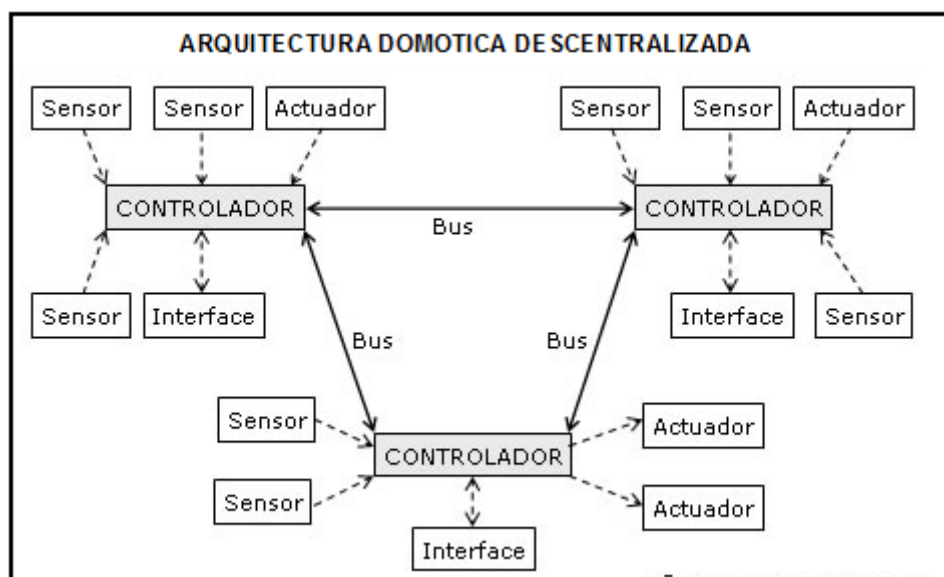


Figura 2. 3. Arquitectura domótica descentralizada

- **Centralizado.** todos los sensores están conectados a las entradas de la unidad de control y los actuadores a la de salida. Por tanto, toda la información es controlada por la unidad central: recibe los datos que las diferentes partes de la instalación captan, la gestiona y es la encargada de realizar todas las modificaciones. Para controlarlo se utiliza un módulo de control, que es programable y el encargado de transformar la información proveniente de las entradas dependiendo de la programación que se le haya hecho como se muestra en la figura 2.4

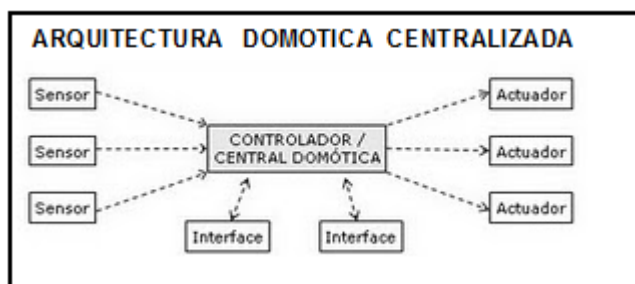


Figura 2. 4. Arquitectura domótica centralizada

También se pueden clasificar las centrales en otros dos tipos:

- **Centrales cableadas:** todos los sensores y actuadores, están conectados mediante un cableado a la central, controlador principal de todo el sistema. Normalmente cuenta con una batería de respaldo, permitiéndole así funcionar durante horas en caso de fallo del suministro eléctrico como se muestra en la figura 2.5.

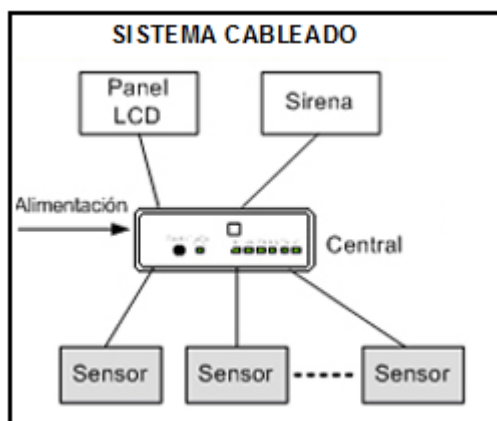


Figura 2. 5. Sistema cableado

- **Centrales inalámbricas:** las conexiones son mediante sensores inalámbricos alimentados por baterías o pilas y transmiten vía radio la información a la central, la cual está alimentada por red eléctrica y tiene sus baterías de respaldo como se muestra en la figura 2.6.

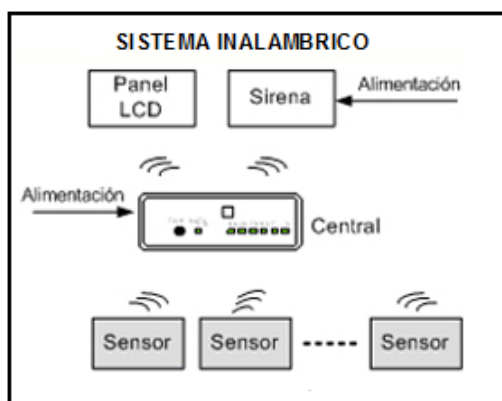


Figura 2. 6. Sistema inalámbrico

- **Centrales mixtos:** combinan el cableado con el inalámbrico

## 2.2. ZIGBEE

Zigbee es un estándar de comunicaciones, vía radio y bidireccional, para usarlo dentro de dispositivos de domótica, automatización de domótica e inmótica, control industrial, periféricos de PC, juguetería, sensores médicos. Zigbee fue creado para cubrir la necesidad del mercado de un sistema a bajo coste, un estándar para redes inalámbricas de pequeños paquetes de información, bajo consumo, seguro y fiable, tiene un campo muy amplio de integración como se muestra en la figura 2.7 [2].

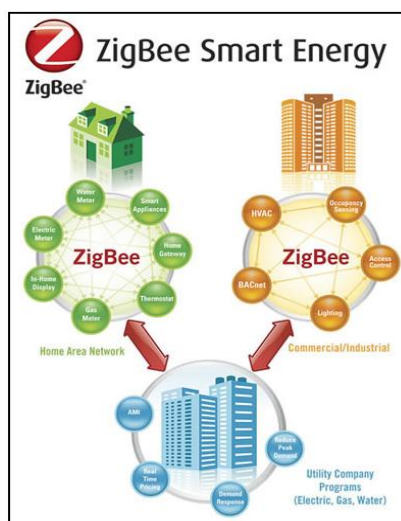


Figura 2. 7. Áreas de integración Zigbee

Además, Se define ZigBee como una pila de protocolos que permite la comunicación de forma sencilla entre múltiples dispositivos.

Se utilizara Zigbee como estándar de comunicación y un kit de desarrollo “Digi” que contienen módulos inalámbricos Xbee, para realizar la comunicación del proyecto en cuestión.

### **2.3. HISTORIA DE ZIGBEE**

Las redes ZigBee comenzaron a ser concebidas a partir del año de 1998 Particularmente con la una necesidad que vieron muchos ingenieros de utilizar de redes ad hoc inalámbricas.

El estándar de IEEE 802.15.4 fue terminado en mayo 2003.

En Octubre del 2004, ZigBee Alliance anunció una duplicación en su número de miembros en el último año a más de 100 compañías en 22 países. En abril de 2005 había más de 150 miembros corporativos, y más de 200 en diciembre del mismo año, y el 14 de diciembre del mismo año se aprueba la especificación Zigbee.

Al siguiente año, ZigBee 2004, se puso a disposición del público sin fines comerciales el 13 de junio en San Ramón, California.

Una vez ya en el mercado en el año 2006, “El precio de mercado de un transceptor compatible con ZigBee se acerca al dólar y el precio de un conjunto de radio, procesador y memoria ronda los tres dólares, en diciembre del mismo año se publicó la actual revisión de la especificación.

Y en Noviembre del 2007 se publicó el perfil *HOME AUTOMATION* de la especificación [3].

### **2.4. INTRODUCCION A LA COMUNICACIÓN INALAMBRICA**

Las comunicaciones inalámbricas son aquellas que propagan la información en condiciones de espacio libre, por medio de ondas electromagnéticas, por lo tanto este tipo de red carece de cables, existen varios tipos de redes inalámbricas como se muestra en la figura 2.8.

Las redes inalámbricas facilitan cualquier tipo de instalación. Permiten que los dispositivos remotos se conecten sin dificultad, sin necesidad de realizar cambios en la infraestructura del lugar donde se va a instalar. Esto ha hecho que el uso de esta tecnología se extienda con rapidez.

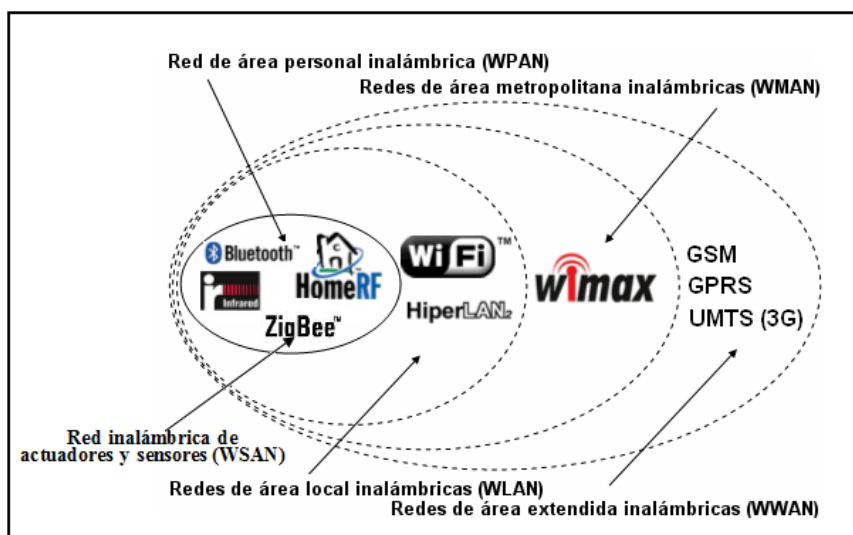


Figura 2. 8. Tipos de Redes Inalámbricas.

## 2.5. ZIGBEE COMO TECNOLOGIA INALÁMBRICA

El mayor interés global de mejorar la eficiencia energética puede desempeñar un papel fundamental en introducir a los consumidores el valor de las soluciones de automatización del hogar, para esto fue creado Zigbee.

ZigBee, también conocido como "*HomeRF Lite*", es una tecnología inalámbrica, basada en el estándar IEEE 802.15.4. Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de transmisión de datos y maximización de la vida útil de sus baterías.

ZigBee comunica una serie de dispositivos haciendo que trabajen más eficiente entre sí. Es especialmente útil para redes de sensores en entornos industriales, médicos y domóticos.

### 2.1.1. Estándar IEEE 802.15.4

IEEE 802.15.4 es un estándar que define el nivel físico y el control de acceso al medio de redes inalámbricas de área personal con tasas bajas de transmisión de datos (*low-rate wireless personal area network, LR-WPAN*). La actual revisión del estándar se aprobó en 2006. 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 [4]

### 2.1.2. Características de Zigbee

- ZigBee opera en las bandas libres ISM (Industrial, Scientific & Medical) de 2.4 GHz, 868 MHz (Europa) y 915 MHz (Estados Unidos).
- Tiene una velocidad de transmisión de 250 Kbps y un rango de cobertura de 10 a 75 metros.
- A pesar de coexistir en la misma frecuencia con otro tipo de redes como WiFi o Bluetooth su desempeño no se ve afectado, esto debido a su baja tasa de transmisión y, a características propias del estándar IEEE 802.15.4, como se puede observar en la figura 2.9 que muestra espectro de Zigbee frente a otras tecnologías.

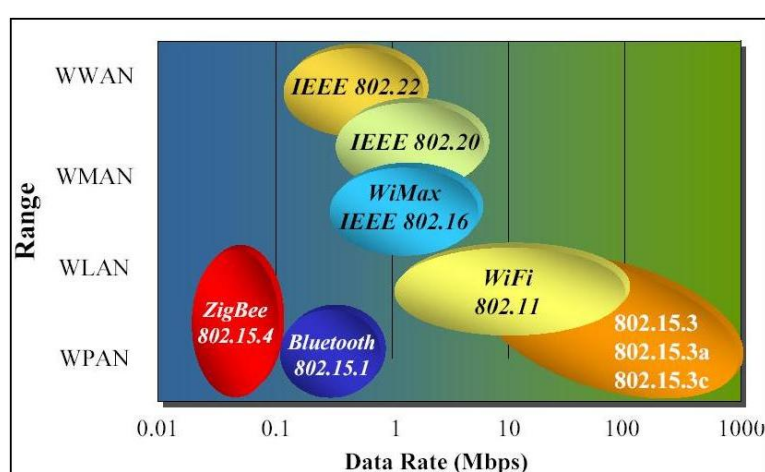


Figura 2. 9. Zigbee spectrum

- La fabricación de un transmisor ZigBee consta de menos circuitos analógicos de los que se necesitan habitualmente.
- Diferentes tipos de topologías como estrella, punto a punto, malla, árbol.
- Capacidad de operar en redes de gran densidad, esta característica ayuda a aumentar la confiabilidad de la comunicación, ya que entre más nodos existan dentro de una red, entonces, mayor número de rutas alternas existirán para garantizar que un paquete llegue a su destino.
- Escalabilidad de red: Un mejor soporte para las redes más grandes, ofreciendo más opciones de gestión, flexibilidad y desempeño.
- Cada red ZigBee tiene un identificador de red único, lo que permita que coexistan varias redes en un mismo canal de comunicación sin ningún problema.
- Puesta de servicio inalámbrico: El conjunto fue mejorado con capacidades seguras: para poner en marcha el servicio inalámbrico.

### **2.1.3. Ventajas**

- Ideal para conexiones punto a punto y punto a multipunto
- Diseñado para el direccionamiento de información y el refrescamiento de la red.
- Opera en la banda libre de ISM 2.4 Ghz para conexiones inalámbricas.
- Óptimo para redes de baja tasa de transferencia de datos.
- Reduce tiempos de espera en el envío y recepción de paquetes.
- Detección de Energía (ED).
- Baja ciclo de trabajo: Proporciona larga duración de la batería.
- Soporte para múltiples topologías de red: Estática, dinámica, estrella y malla.
- Hasta 65.000 nodos en una red.
- Provee conexiones seguras entre dispositivos.
- Son más baratos y de construcción más sencilla.

### **2.1.4. Desventajas**

- La tasa de transferencia es muy baja.
- Solo manipula textos pequeños comparados con otras tecnologías.
- Zigbee trabaja de manera que no puede ser compatible con bluetooth en todos sus aspectos porque no llegan a tener las mismas tasas de transferencia, ni la misma

capacidad de soporte para nodos.

- Tiene menor cobertura porque pertenece a redes inalámbricas de tipo WPAN.

### 2.1.5. Tipos de dispositivos

#### 2.1.5.1 Dispositivos Zigbee según su papel en la red

**Coordinador (ZigBee Coordinator, ZC).** Que es el encargado del control la red y los caminos que deben seguir los dispositivos para conectarse entre ellos.

**Router (ZigBee Router, ZR).** Que tiene la funcionalidad del enrutamiento de paquetes y ser origen o destino de información.

**Dispositivo final (ZigBee End Device, ZED).** Que serán los sensores y actuadores de la red.

#### 2.1.5.2 Dispositivos Zigbee según su funcionalidad.

**Dispositivo de funcionalidad completa (FFD).** Es capaz de recibir mensajes en formato del estándar 802.15.4. Gracias a la memoria adicional y a la capacidad de computar, puede funcionar como coordinador o router o puede ser usado en dispositivos de red que actúen de interface con los usuarios.

**Dispositivo de funcionalidad reducida (RFD).** Tiene capacidad y funcionalidad limitadas (especificada en el estándar) con el objetivo de conseguir un bajo coste y una gran simplicidad. Básicamente, son los sensores/actuadores de la red.

Para el desarrollo del proyecto se basara en los diferentes de dispositivos ZigBee según su papel en la red para implementar la red inalámbrica que se mencionara más adelante.

### 2.1.6. Tipos de Trafico de Datos

ZigBee/IEEE 802.15.4 dirige tres tipos de tráfico típicos:

**Cuando el dato es periódico.** La aplicación dicta la proporción, el sensor se activa, chequea los datos y luego desactiva.

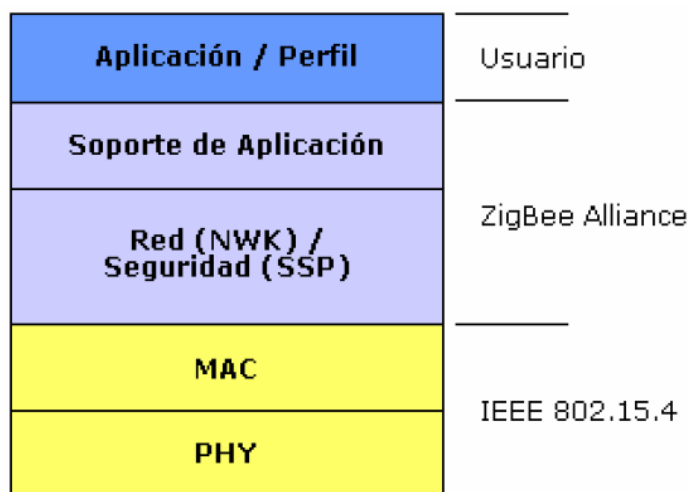


**Cuando el dato es intermitente.** La aplicación, u otro estímulo, determinan la proporción como en el caso de los detectores de humo. El dispositivo necesita sólo conectarse a la red cuando la comunicación se hace necesaria. Este tipo habilita el ahorro óptimo en la energía.

**Cuando el dato es repetitivo.** La proporción es a priori fija. Dependiendo de las hendeduras de tiempo repartidas, los dispositivos operan para las duraciones fijas.

### 2.1.7. Arquitectura de Zigbee

ZigBee es una pila de protocolos, que de manera similar al modelo OSI está constituido por diferentes capas, las cuales son independientes una de la otra. En la figura 2.3 se muestran las diferentes capas que conforman la pila de protocolos para ZigBee.



**Figura 2. 10. Diferentes capas que conforman la pila de protocolos para ZigBee**

La capa de más bajo nivel es la capa física (PHY), que en conjunto con la capa de acceso al medio (MAC), brindan los servicios de transmisión de datos por el aire, punto a punto. Estas dos capas esta descritas en el estándar IEEE 802.15.4.

El estándar trabaja sobre las bandas ISM<sup>2</sup> de uso no regulado, dónde se definen hasta 16 canales en el rango de 2.4 GHz, cada una de ellas con un ancho de banda de 5 MHz. Se utilizan radios con un espectro de dispersión de secuencia directa, lográndose tasas de

<sup>2</sup> **ISM (Industrial, Scientific and Medical)** son bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica. En la actualidad estas bandas han sido popularizadas por su uso en comunicaciones inalámbricas

transmisión en el aire de hasta 250 Kbps en rangos que oscilan entre los 10 y 75 m, los cuales dependen bastante del entorno.

La capa de red tiene como objetivo principal permitir el correcto uso del subnivel MAC y ofrecer una interfaz adecuada para su uso por parte de la capa de aplicación. En esta capa se brindan los métodos necesarios para: iniciar la red, unirse a la red, enrutar paquetes dirigidos a otros nodos en la red, proporcionar los medios para garantizar la entrega del paquete al destinatario final, filtrar paquetes recibidos, cifrarlos y autentificarlos. Se debe tener en cuenta que el algoritmo de enrutamiento que se usa es el de enrutamiento de malla, el cual se basa en el protocolo Ad Hoc On-Demand Vector Routing – AODV. Cuando esta capa se encuentra cumpliendo la función de unir o separar dispositivos a través del controlador de red, implementa seguridad, y encamina tramas a sus respectivos destinos; además, la capa de red del controlador de red es responsable de crear una nueva red y asignar direcciones a los dispositivos de la misma. Es en esta capa en donde se implementan las distintas topologías de red que ZigBee soporta (árbol, estrella y mesh network).

La siguiente capa es la de soporte a la aplicación que es el responsable de mantener el rol que el nodo juega en la red, filtrar paquetes a nivel de aplicación, mantener la relación de grupos y dispositivos con los que la aplicación interactúa y simplificar el envío de datos a los diferentes nodos de la red. La capa de Red y de soporte a la aplicación son definidas por la ZigBee Alliance.

En el nivel conceptual más alto se encuentra la capa de aplicación que no es otra cosa que la aplicación misma y de la que se encargan los fabricantes. Es en esta capa donde se encuentran los ZDO (ZigBee Device Objects) que se encargan de definir el papel del dispositivo en la red, si el actuará como coordinador, ruteador o dispositivo final; la subcapa APS y los objetos de aplicación definidos por cada uno de los fabricantes.

Cada capa se comunica con sus capas subyacentes a través de una interface de datos y otra de control, las capas superiores solicitan servicios a las capas inferiores, y éstas reportan sus resultados a las superiores. Además de las capas mencionadas, a la arquitectura se integran otro par de módulos: módulo de seguridad, que es quien provee los servicios para cifrar y autentificar los paquetes, y el módulo de administración del

dispositivo ZigBee, que es quien se encarga de administrar los recursos de red del dispositivo local, además de proporcionar a la aplicación funciones de administración remota de red.

### 2.1.8. Topología

ZigBee permite las siguientes topologías de red:

- **Topología en estrella.** el coordinador se sitúa en el centro.
- **Topología en árbol.** el coordinador será la raíz del árbol.
- **Topología de malla.** al menos uno de los nodos tendrá más de dos conexiones.
- **Topología punto a punto.** existe un solo FFD Coordinador. A diferencia con la topología estrella. Las aplicaciones orientadas para el monitoreo y control de procesos industriales, redes de sensores inalámbricos, entre otros, son ampliamente usados por estas redes. Proveen confiabilidad en el enrutamiento de datos (multipath routing).

### 2.1.9. Comparación de tecnologías inalámbricas

Tabla 2. 1. Tabla de comparación de tecnologías.

<b>Comparación de Tecnologías Inalámbricas</b>			
	<b>Wi-fi</b>	<b>Bluetooth</b>	<b>ZigBee</b>
Bandas de Frecuencias	2.4GHz	2.4GHz	2.4GHz, 868 / 915 MHz
Tamaño de Pila	~ 1Mb	~ 1Mb	~ 20kb
Tasa de Transferencia	11Mbps	1Mbps	250kbps (2.4GHz) 40kbps (915MHz) 20kbps (868MHz)
Números de Canales	11 - - 14	79	16 (2.4GHz) 10 (915MHz) 1 (868MHz)
Tipos de Datos	Digital	Digital, Audio	Digital ( <u>Texto</u> )
Rango de Nodos Internos	100m	10m - 100m	10m - 100m
Números de Dispositivos	32	8	255 / 65535
Requisitos de Alimentación	Media - Horas de Batería	Media - Días de Batería	Muy Baja - Años de Batería
Introducción al Mercado	Alta	Media	Baja
Arquitecturas	Estrella	Estrella	Estrella, Árbol, Punto a Punto y Malla
Mejores de Aplicaciones	Edificio con Internet Adentro	Computadoras y Teléfonos	Control de Bajo Costo y Monitoreo
Consumo de Potencia	400ma transmitiendo, 20ma en reposo	40ma transmitiendo, 0.2ma en reposo	30ma transmitiendo, 3ma en reposo
Precio	Costoso	Accesible	Bajo
Complejidad	Complejo	Complejo	Simple

## 2.6. ZIGBEE EN APLICACIONES DOMÓTICAS

Es el estándar mundial para el control de electrodomésticos, iluminación, el medio ambiente, gestión energética, y seguridad.

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 como se muestra en la figura 2.11, con características auto organizativas y bajo coste. Puede utilizarse para realizar control industrial, albergar sensores empotrados, recolectar datos médicos, ejercer labores de detección de humo y en el área de la domótica.

La red 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 cambio en su sistema de alimentación [3].



Figura 2. 11. Aplicaciones de Zigbee

Un sistema domótico ha de poder controlar diferentes configuraciones: *peer to peer* poder cubrir el área de una casa, y sobre todo la configuración MESH (rejilla) que nos permitirá no depender del rango.

La seguridad de las transmisiones y de los datos son puntos clave en la tecnología ZigBee.

ZigBee utiliza el modelo de seguridad de la subcapa MAC IEEE 802.15.4, la cual especifica 4 servicios de seguridad.

- Control de accesos. El dispositivo mantiene una lista de los dispositivos ‘comprobados’ en la red.
- Datos Encriptados. Los cuales usan una encriptación con un código de 128 bits.
- Integración de tramas para proteger los datos de ser modificados por otros.
- Secuencias de refresco, para comprobar que las tramas no han sido reemplazadas por otras.
- El controlador de red. Comprueba estas tramas de refresco y su valor, para ver si son las esperadas.
- Depende del dispositivo final que creemos será nuestra decisión el dotarlo de más o menos seguridad.

#### **2.4.1 Servicios que ofrece Zigbee en el área domotica**

##### **Ahorro energético.**

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

##### **Confort.**

- Iluminación:
  - Apagado general de todas las luces de la vivienda
  - Automatización del apagado/ encendido en cada punto de luz
  - Regulación de la iluminación según el nivel de luminosidad ambiente
- Automatización de todos los distintos sistemas/ instalaciones / equipos dotándolos de control eficiente y de fácil manejo

- Integración del portero al teléfono, o del videoportero al televisor
- Control vía Internet
- Gestión Multimedia y del ocio electrónicos
- Generación de macros y programas de forma sencilla para el usuario

### **Seguridad.**

- Simulación de presencia.
- Alarmas de Detección de incendio, fugas de gas, escapes de agua, concentración de monóxido en garajes.
- Alerta médica. Tele-asistencia.
- Cerramiento de persianas puntual y seguro.
- Acceso a Cámaras IP

### **Comunicaciones**

- Ubicuidad en el control tanto externo como interno, control remoto desde Internet, PC, mandos inalámbricos.
- Transmisión de alarmas.
- Intercomunicaciones.

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

## **2.7. ZIGBEE EN WSAN**

### **2.7.1 WSAN**

WSAN Se refieren a un grupo de sensores y actores como muestra la figura 2.12 vinculados por medio inalámbrico para realizar detección distribuida y las tareas de actuación.

En este tipo de red, los sensores son encargados de reunir información sobre el mundo físico, mientras que los actuadores toman decisiones y luego realizar acciones apropiadas sobre el medio en el que se encuentra.

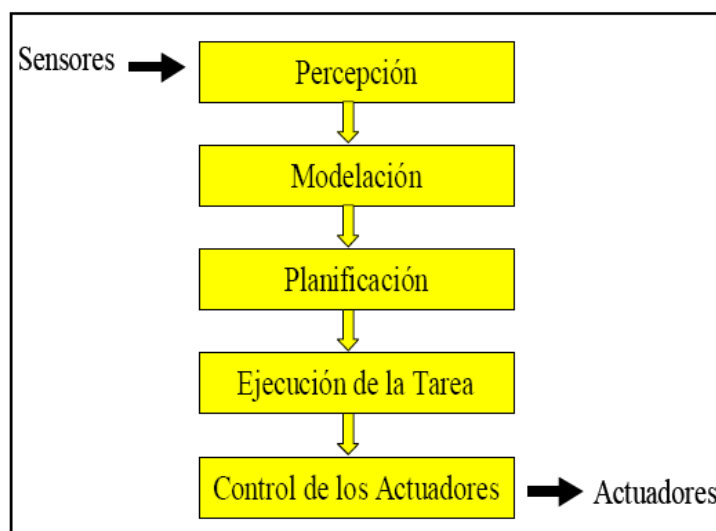


Figura 2. 12. Diseño funcional de sensor y actuador dentro de una red

### 2.7.2 Características de WSN

- Provee de servicios en tiempo real, con límites de retardos dados, acorde a los requerimientos de la aplicación.
- Asegura una comunicación eficiente entre sensores y actuadores considerando el consumo de energía.
- Asegura el orden entre diferentes eventos cuando estos son reportados a los actuadores.
- Provee de sincronización para diferentes sensores que reportan el mismo evento a múltiples o el mismo actuador con el fin de facilitar la respuesta de la región completa.
- Rastrea y reporta fenómenos sentidos a un conjunto diferente de actuadores, no necesariamente basados en la proximidad o limitaciones de energía, para el caso cuando el evento toma lugar en diferentes localidades [4].



### 2.7.3 Arquitecturas de una WSN

**Automatizada.** Debido a la no existencia de un controlador central. Los sensores detectan un fenómeno y transmiten sus lecturas al actuador, el cual procesa la información e inicia una acción apropiada.

- **Baja latencia.** Debido a que la comunicación es directamente del sensor al actuador, los cuales se encuentran muy cercanos unos de otros.
- **Mayor tiempo de vida de la red.** Los nodos a un salto de los actuadores pueden tener más carga que los demás nodos. Sin embargo es muy probable que cada evento tenga asignado diferentes actuadores por lo que la carga se distribuye más o menos entre todos los nodos, provocando mayor tiempo de vida de la red.
- **Ahorro de recursos.** Debido a que la información del evento es transmitida localmente a través de los nodos sensores alrededor del área del evento, los sensores que se encuentran alejados del área del evento no operan como retransmisores, lo cual resulta en un ahorro de recursos de la red.

**Semi-automatizada.** Debido a que el controlador central recolecta los datos y coordina las acciones a ejecutar, en este caso los datos de los sensores son ruteados al controlador el cual dirige los comandos de acción a los actuadores.

### 2.7.4 Aplicaciones de una WSN [5]

Dentro de las aplicaciones potenciales a desarrollar en el contexto de las WSN se tienen:

**Sector médico.** Sistemas económicos y portables, orientados al monitoreo, registro y análisis de variables fisiológicas, a partir de lo cual es posible indicar el estado de un paciente y detectar la presencia o el riesgo de desarrollar alguna enfermedad. Como también, el desarrollo de sistemas de detección y análisis de tendencias en el comportamiento diario de pacientes, que contribuyan a detectar oportunamente la presencia de un problema de salud, y ofrezcan una solución económicamente viable a la atención de pacientes en sociedades donde la población en edad avanzada es grande.

**Sector medioambiental.** Sistemas de monitorización continuo de especies en vía de extinción, sistemas de vigilancia en bosques para la detección de actividades ilícitas de la tala y caza, sistemas de detección y monitoreo de incendios forestales, sistemas de monitoreo continuo para la caracterización de cuencas hidrográficas, etc.

**Sector agrícola.** Sistemas de detección, monitorización y control de plagas en microclimas, para disminuir el uso de agroquímicos y realizar un control óptimo de plagas; sistemas de riego que realicen un uso óptimo del agua, etc.

**Sector industrial.** Sistemas económicos y de fácil instalación para la monitorización, diagnóstico y control de plantas y procesos industriales.

**Sector Domótico.** Seguridad, Alarmas, control de iluminación, control de aire acondicionado, control domótico.

Para establecer la comunicación inalámbrica de la red será necesario la utilización de los módulos Xbee de MAXSTREAM.

## **2.8. MÓDULOS DE MAXSTREAM.**

La característica fundamental que ha destacado a los productos MAXSTREAM es la sensibilidad de sus receptores. En vez de elevar la potencia del transmisor, con un aumento de consumo, para lograr mayor alcance, los productos MAXSTREAM incorporan receptores más sensibles, lo que les permite lograr un mayor alcance que otros productos similares, manteniendo un consumo reducido.

La corriente de operación de estos dispositivos ronda los 50mA, tanto para transmisión como para recepción, mientras que en el modo de bajo consumo se reduce a tan sólo 10uA.

### **2.6.1 Módulos Xbee**

Son módulos de radio frecuencia que trabajan en la banda de 2.4 GHz con protocolo de comunicación 802.15.4 fabricados por MAXSTREAM, viene en una especie de chip, están compuestos por un microcontrolador, un emisor y un receptor de RF, con una alimentación de 3.3v.

### 2.6.2 Funcionamiento de los módulos Xbee

Los módulos XBee fueron diseñados para ser montados en un zócalo, sin requerimiento de soldadura como se indica en la figura 2.13. Para ello disponen dos hileras de 10 pines separadas entre ellas por 22 mm. La separación entre pines es de 2mm.

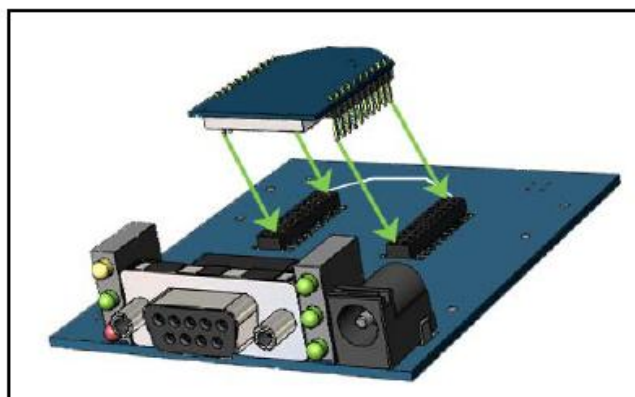


Figura 2. 13. Zócalo demostrativo para módulo Xbee

### 2.6.3 Conexión Básica

El módulo requiere una alimentación desde 2.8 a 3.4 V, la conexión a tierra y las líneas de transmisión de datos por medio del UART (Universal Asynchronous Receiver-Transmitter) (TXD y RXD) para comunicarse con un microcontrolador, o directamente a un puerto serial utilizando algún convertor adecuado para los niveles de voltaje [6], como muestra la figura 2.14

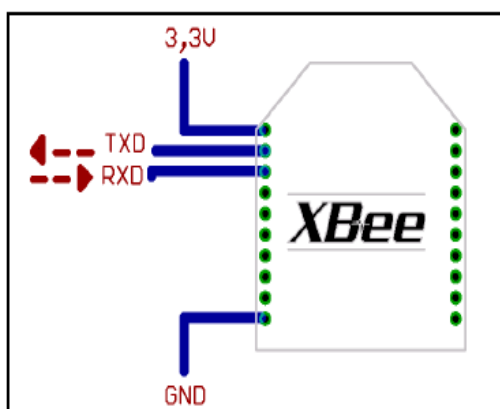


Figura 2. 14. Conexiones mínimas requeridas para un modulo Xbee

## 2.6.4 Configuración de Pines de módulo Xbee

En la figura 2.15 se muestra un Diagrama del modulo Xbee de Vista superior para facilitar la ubicación de los pines

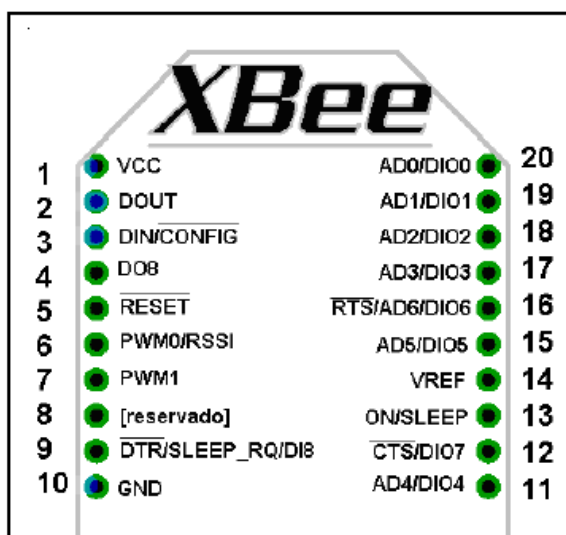


Figura 2. 15. Diagrama de pines del modulo Xbee, vista superior

Tabla 2. 2. Pines del módulo Xbee

Pin #	Nombre	Dirección	Descripción
1	VCC	-	Alimentación
2	DOUT	Salida	Salida UART
3	DIN/	Entrada	Entrada UART
4	DIO12	E/S	Digital E/S 12
5		Entrada	Modulo de reinicio
6	PWM0/RSSI/DIO10	E/S	PWM Salida 0/ Indicador de la intensidad de señal recibida / Digital E/S
7	DIO11	E/S	Digital E/S 11
8	Reservado	-	No conectar
9	/SLEEP_RQ/DIO8	E/S	Pin Sleep Control Line o Digital E/S 8
10	GND	-	Tierra
11	DIO4	E/S	Digital E/S 4
12	/DIO7	E/S	Control de flujo Clear-to-Send o Digital E/S 7.CTS, si se encuentra habilitado este es una salida.
13	ON/	Salida	Indicador del estado del modulo o Digital E/S 9
14	VREF	Entrada	No se utiliza en este módulo. Por compatibilidad con otros módulos XBee, se recomienda conectar este pin a un voltaje de referencia, si el muestreo analógico se desea. De lo contrario, conectarse a GND
15	Asociado/DIO5	E/S	Indicador asociado , Digital E/S 5
16	/DIO6	E/S	Control de flujo Request-to-Sent, Digital E/S 6.RTS, si se encuentra habilitado este es una entrada.
17	AD3/DIO3	E/S	Entrada analógica 3 o Digital E/S 3
Pin #	Nombre	Dirección	Descripción
18	AD2/DIO2	E/S	Entrada analógica 2 o Digital E/S 2
19	AD1/DIO1	E/S	Entrada analógica 1 o Digital E/S 1
20	AD0/DIO0/Botón/ Botón puesto en servicio	E/S	Entrada analógica 0, Digital E/S 0 o Botón de puesto en servicio.

## DIMENSIONES:

- ANCHO: 24.38 mm
- LARGO: 27.61 mm
- ALTO DE ANTENA: 25 mm

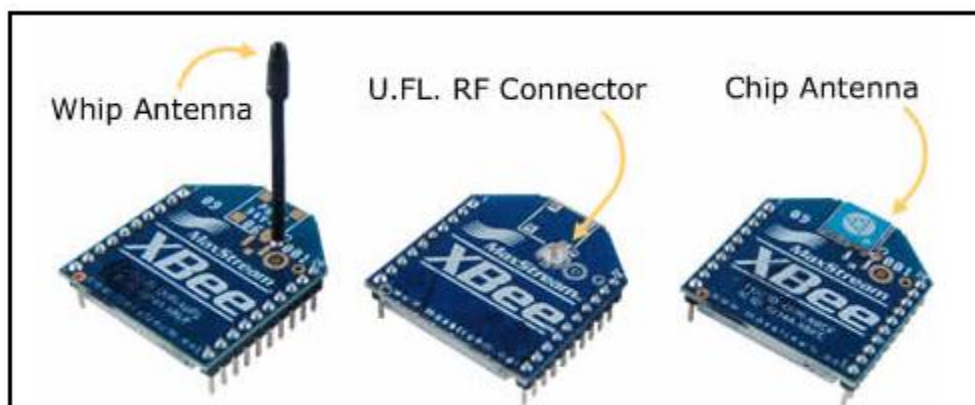


Figura 2. 16. Módulos Xbee

Los módulos Xbee que muestra la figura 2.16, utilizan el protocolo IEEE 802.15.4 mejor conocido como Zigbee. Este protocolo se creó pensando en implementar redes de sensores. El objetivo es crear redes tipo MESH que tengan las propiedades de auto-recuperación y bajo consumo de energía.

#### DIMENSIONES:

- ANCHO: 24.38 mm
- LARGO: 32.94mm
- ALTO DE ANTENA: 25 mm

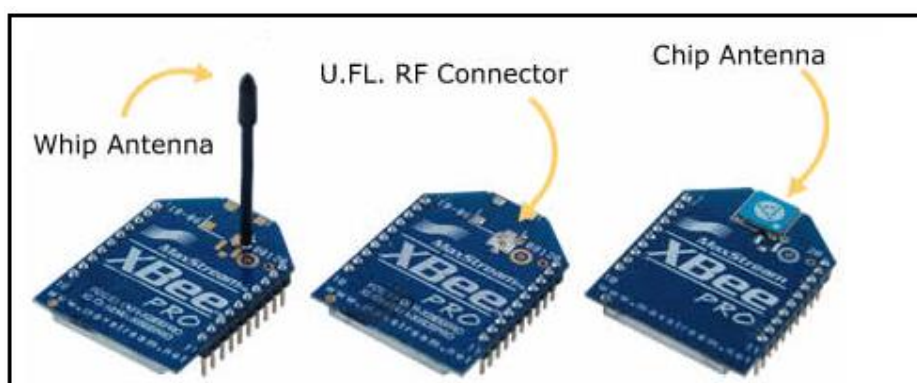


Figura 2. 17. Módulos Xbee PRO

Los Módulos Xbee PRO que se indican en la figura 2.17, permiten enlaces seriales de señales TTL en distancias de 30 metros en interiores, 100 metros en exteriores con línea de vista y hasta 1.5 km.

### 2.6.5 Modos de operación.

El funcionamiento estándar permite operar fundamentalmente en los siguientes modos:

- **Modo RECIBIR/TRANSMITIR.** Se encuentra en este modo cuando al módulo le llega algún paquete RF a través de la antena (modo *Receive*) o cuando se manda 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 mientras que en el modo Indirecto la información es retenida durante un período de tiempo y es enviada sólo cuando la dirección de destino la solicita.

Además es posible enviar información por *Unicast* y *Broadcast*. En el primer modo 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 (acknowledgement) (paquete llamado así, y que indica que recibió el paquete, el usuario no puede verlo, es interno de los módulos) a la dirección de origen. Quien envió el paquete, espera recibir un ACK, en caso de que no le llegue, reenviará el paquete hasta 3 veces o hasta que reciba el ACK. En el modo *Broadcast* la comunicación es entre un nodo y a todos los nodos de la red. En este modo, no hay confirmación por ACK.

- **Modo peer-to-peer.** Cada módulo habla con cualquier otro módulo, emitiendo broadcasts o direccionando un módulo remoto. Esto requiere que todos los módulos tengan su receptor constantemente encendido, dado que cualquiera puede recibir un mensaje en cualquier momento, pero permite mantener mensajes entre todos los módulos.
- **Modo con coordinador.** Uno de los módulos se configura para el rol de coordinador y está siempre alerta, pudiendo los remotos permanecer en modo bajo consumo por un tiempo determinado, el que se calcula para minimizar el consumo. Todas las

comunicaciones de los remotos son hacia el coordinador. Éste, puede almacenar hasta dos mensajes para un remoto, hasta que este último, al retornar a funcionamiento normal, interroga al coordinador si tiene algún mensaje para él.

La configuración de los módulos se realiza mediante comandos AT, pudiendo operar fundamentalmente en uno de dos modos:

- **Modo transparente.** La configuración en este modo se la realiza mediante comandos AT.
- **Modo API (*Application Programming Interface*).** En este modo no existe modo datos, ni modo comando, se utiliza un protocolo para establecer la comunicación.

## 2.9. PROTOTIPO RECEPTOR

Para el diseño del prototipo receptor se utilizará un microcontrolador y un modem GSM.

### 2.7.1 Microcontrolador

Un microcontrolador es un circuito integrado, capaz de llevar a cabo procesos lógicos, estos procesos o acciones son programados por el usuario en lenguaje ensamblador y son introducidos en este a través de un programador, este incorpora la mayor parte de los elementos que configuran un controlador.

Cumple con las siguientes características:

- CPU (*Central Processor Unit o Unidad de Procesamiento Central*)
- Memorias volátiles (RAM), para datos
- Memorias no volátiles (ROM, PROM, EPROM) para escribir el programa
- Líneas de entrada y salida para comunicarse con el mundo exterior.
- Algunos periféricos (comunicación serial, temporizador, convertidor A/D, etc.)

### 2.7.2 Microcontrolador ATMEL

Para el desarrollo del de proyecto se utilizo microntroladores ATMEL (ATmega164), cuyas características son:



- ATMEL son una familia de microcontroladores RISC<sup>3</sup>, y memoria flash reprogramable eléctricamente.
- El microcontrolador ATMEL tiene 32 registros de 8 bits.
- La concatenación de los 32 registros de entrada/salida y la memoria de datos conforman un espacio de direcciones unificado, al cual se accede a través de operaciones de carga/almacenamiento.
- El stack se ubica en el espacio de memoria unificado, y no esta limitado a un tamaño fijo.
- Los microcontroladores ATMEL tiene *pipeline* con dos etapas (cargar y ejecutar), que les permite ejecutar la mayoría de instrucciones en un ciclo de reloj, lo que los hace relativamente rápidos entre los microcontroladores de 8 bits.

### 2.7.3 Microcontrolador ATmega164

ATmega16 es un microcontrolador de alto rendimiento y baja potencia de 8k bytes de flash programable con capacidad de lectura y escritura, con 512 bytes de EEPROM, 1kbyte de *SRAM*, también cuenta con 32 I/O de propósito general, un *USART* Serial programable, interface SPI master/esclavo con generador del reloj, usualmente un cristal de cuarzo de frecuencias que genera una señal oscilatoria entre 1 y 40 MHz, o también resonadores o circuitos RC, en la figura 2.18 se muestra la descripción del pines del microcontrolador [7].

---

<sup>3</sup> RISC: (reduced instruction set computer ) Computadoras con un conjunto de instrucciones reducido

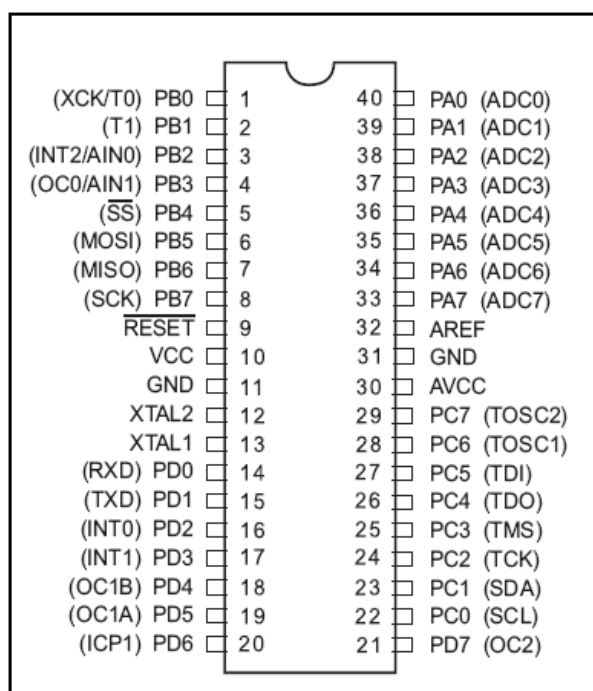


Figura 2. 18. Distribución de pines ATmega164

#### ▪ Arquitectura RISC avanzada

- 131 Instrucciones Potentes.
- La mayoría Ejecuta en solo ciclo de reloj.
- 32 x 8 Registros de propósito general.
- Rendimiento hasta 16 MIPS a 16 MHz.

#### ▪ Programa no volátil de datos y recuerdos

- De 16K Bytes En Sistema de auto-programable Flash.
- Resistencia: 10.000 Write/Erase Ciclos.
  - 512 Bytes EEPROM.
- Resistencia: 100.000 Write/Erase Ciclos.
  - 1K Byte SRAM Interior
- Programación de bloqueo de Software Security
  - • JTAG (IEEE Std. Compatible 1149,1) Interfaz.
- Capacidades de exploración de Fronteras Según la Norma JTAG.
- Amplia El chip de depuración de Apoyo.
- La programación de Flash, EEPROM, fusibles, y de bloqueo de bits a través de la

interfaz JTAG.

• **Características Periféricas**

- Dos de 8-bits temporizador / Contadores con Prescalers separado y comparar los modos.
- Un temporizador de 16-bit / Contador con Prescaler separado, comparar el modo de captura y Modo.
- Cuatro Canales PWM.
- 8-channel, 10-bits ADC
- Byte orientadas a dos cables de interfaz serial .
- Programable serie USART .
- Maestro / Esclavo de interfaz en serie SPI.
- Temporizador programable de vigilancia por separado con el chip oscilador.

• **Características especiales Microcontrolador.**

- Power-On Reset y programable de detección de Brown.
- Oscilador RC Calibrada Interna.
- Interrupción de Exteriores y del Interior Fuentes.
- Seis modos: Inactivo, ADC Reducción de ruido, de ahorro de energía, reducción de potencia, modo de espera y extensión de espera

• **I / O y encapsulado**

- 32 programable líneas de E / S
- 40-pines PDIP, 44-plomo TQFP, y 44-almojadilla FML

• **Los voltajes de operación**

- 2,7 - 5.5V para ATmega16L
- 4.5 - 5.5V para ATmega16

• **Grados de Velocidad**

- 0 - 8 MHz para ATmega16L
- 0 - 16 MHz para ATmega16

• **Consumo de energía a 1 MHz, 3V, y 25 ° C**

- Activa: 1,1 mA
- El modo de espera: 0,35 mA
- Power-Down Mode: <1  $\mu$ A

## 2.7.4 Definición de GSM

El Sistema Global para las Comunicaciones Móviles (GSM, proviene de "*Groupe Speciale Mobile*") es un sistema estándar, completamente definido, para la comunicación mediante teléfonos móviles que incorporan tecnología digital. Por ser digital cualquier cliente de GSM puede conectarse a través de su teléfono con su ordenador y puede hacer, enviar y recibir mensajes por e-mail, faxes, navegar por Internet, acceso seguro a la red informática de una compañía (LAN/Intranet), así como utilizar otras funciones digitales de transmisión de datos, incluyendo el Servicio de Mensajes Cortos (SMS) o mensajes de texto. [8]

GSM se considera, por su velocidad de transmisión y otras características, un estándar de segunda generación (2G), el ETSI (*European Telecommunications Standard Institute*) fue la organización que se encargó de su estandarización entre el 1982 y el 1992.

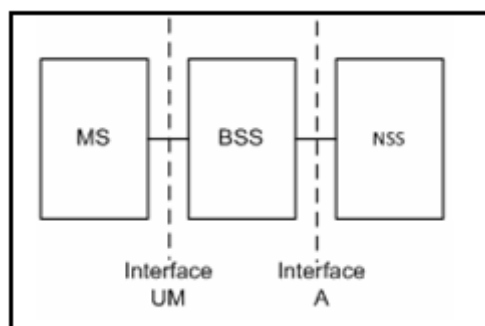
### 2.7.4.1.1 Arquitectura de la red de telefonía móvil GSM

La arquitectura del sistema GSM se compone de tres bloques o subsistemas que engloban el conjunto de entidades del sistema. Cada uno de estos subsistemas desempeña funciones específicas para poder ofrecer el servicio de telefonía móvil al usuario e interactuar con otras redes [9].

Los tres subsistemas son:

- NSS (Subsistema de red y comunicación).
- BSS (Subsistema de estación base).
- MS (Estación móvil).

En la figura 2.19 se puede observar los tres subsistemas del sistema GSM



**Figura 2. 19. Arquitectura de red GSM**

Estos subsistemas se intercomunican entre ellos a través de diferentes interfaces mediante protocolos de señalización específicos. La comunicación entre el conjunto de estaciones móviles y el subsistema de estación base se realiza mediante el “interface aire” o “interface Radio” y que se designa abreviadamente como interface Um. Por otro lado también encontramos la interface A que es la encargada de la comunicación entre el subsistema de estación base y el subsistema de red.

#### **2.7.4.1.1.1 Subsistema de red y conmutación (NSS)**

El subsistema NSS realiza las funciones de conmutación y encaminamiento de las llamadas en el sistema GSM, además de la gestión de las bases de datos con la información relativa a todos los abonados al servicio.

El NSS se encarga de establecer la comunicación entre usuarios móviles mediante la conmutación interna de red de un operador o entre usuarios del sistema GSM o usuarios de otras redes de telefonía, ya sea de telefonía fija o de telefonía móvil de otros operadores.

Dentro del subsistema NSS las funciones de conmutación las realizan las centrales de conmutación (MSC (*Network Switching Subsystem*) y GMSC (*Gateway Mobile Switching Centre*)).

La unidad MSC es el elemento de conmutación interno de la red GSM mientras que la unidad GMSC es el elemento de interconexión con otras redes.

La gestión de la base de datos la realizan el registro central de abonado identificado como HLR (*Home Location Register*) y el registro de posición del visitante identificado

como VLR (Visitor Location Register). Dentro del subsistema NSS existen otros elementos tales como el AUC y el EIR, entre otros, de los que no entraremos en detalle.

También existe el subsistema OSS (Subsistema de Operaciones y mantenimiento) que es el encargado de llevar a cabo las tareas de supervisión y mantenimiento de toda la infraestructura de la red GSM. Las acciones de operación y mantenimiento se llevan a cabo con el fin de conseguir el buen funcionamiento del sistema GSM en su conjunto, ya sea solucionando los problemas e incidencias o monitorizando y mejorando la configuración de los equipos para un mayor rendimiento.

#### **2.7.4.1.1.2 Subsistema de Estación base (BSS)**

El subsistema de estación base agrupa la infraestructura específica de los aspectos radio para el sistema GSM. Este subsistema se compone de estaciones base (*BTS, Base Transceiver Station*) conectadas a las controladoras de estaciones base que se denominan *BSC (Base Station Controller)*.

El BSS desempeña las siguientes funciones:

- Transmisión/Recepción radioeléctrica, a través de la interfaz Um.
- Localización de las MS para su registro o actualización.
- Establecimiento, supervisión y conclusión de las llamadas.
- Traspaso entre BTS controladas por el mismo BSC.
- Procesado de voz y adaptación de velocidad
- Control de equipos y funciones de reconfiguración.
- Control de mantenimiento rutinario.

#### **2.7.4.1.1.3 Subsistema de Estación móvil (MS)**

La estación móvil es lo que popularmente conocemos como teléfono Móvil. La infraestructura de GSM descompone la estación móvil en cuatro elementos. El terminal móvil (MT) que es el teléfono móvil. La tarjeta SIM (*Subscriber Identity Module*) que es la tarjeta de abonado que proporciona el operador al usuario cuando se contratan sus servicios. El elemento TA (*Terminal Adapter*) que es el elemento de adaptación para la interconexión del teléfono móvil con un equipo terminal de datos y el TE (*Terminal Equipment*) que es el adaptador para la transmisión de datos vía GSM.

### 2.7.5 SERVICIO SMS

El servicio de mensajes cortos o SMS (*Short Message Service*) es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos (también conocidos como mensajes de texto), entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano. SMS fue diseñado originariamente como parte del estándar de telefonía móvil digital GSM, pero en la actualidad está disponible en una amplia variedad de redes, incluyendo las redes 3G.

Un mensaje SMS es una cadena alfanumérica de hasta 160 caracteres de 7 bits, y cuyo encapsulado incluye una serie de parámetros. En principio, se emplean para enviar y recibir mensajes de texto normal, pero existen extensiones del protocolo básico que permiten incluir otros tipos de contenido, dar formato a los mensajes o encadenar varios mensajes de texto para permitir mayor longitud (formatos de SMS con imagen de Nokia, tonos IMY de Ericsson, estándar EMS para dar formato al texto e incluir imágenes y sonidos de pequeño tamaño).

El servicio SMS permite transferir un mensaje de texto entre una estación móvil (MS) y otra entidad (SME) a través de un centro de servicio (SC) como muestra la figura 2.20.

El servicio final ofrecido es una comunicación extremo-extremo entre la estación móvil (MS) y la entidad (SME). La entidad puede ser otra estación móvil o puede estar situado en una red fija. En el caso de envío de un mensaje entre dos móviles, ambas partes son estaciones móviles. Cuando se envía un mensaje para solicitar algún tipo de servicio (o realizar alguna votación, sobre todo en los concursos de la TV, que ahora están tan de moda), un extremo es una estación móvil y la otra es un servidor que atiende las peticiones (o anota los votos).

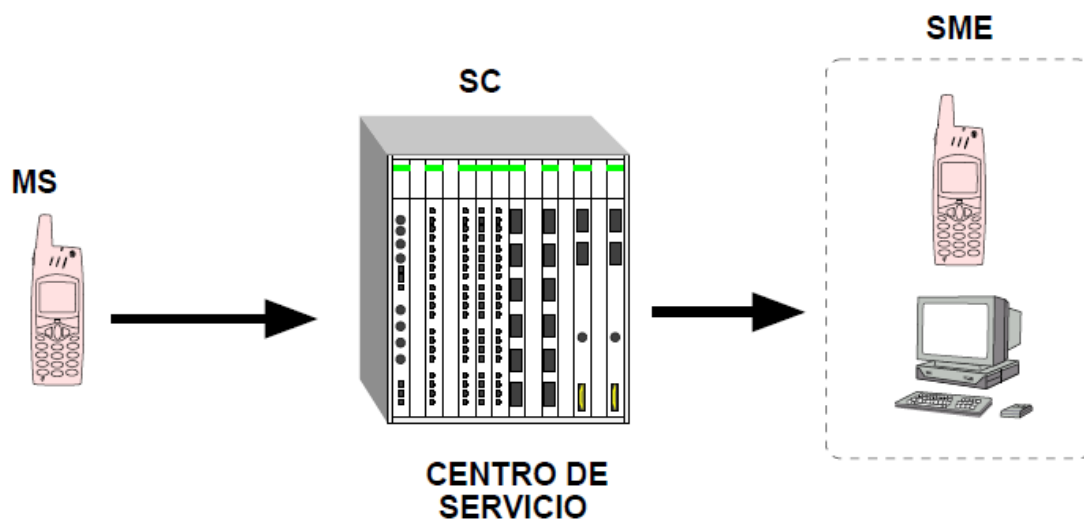


Figura 2. 20. Servicio SMS.

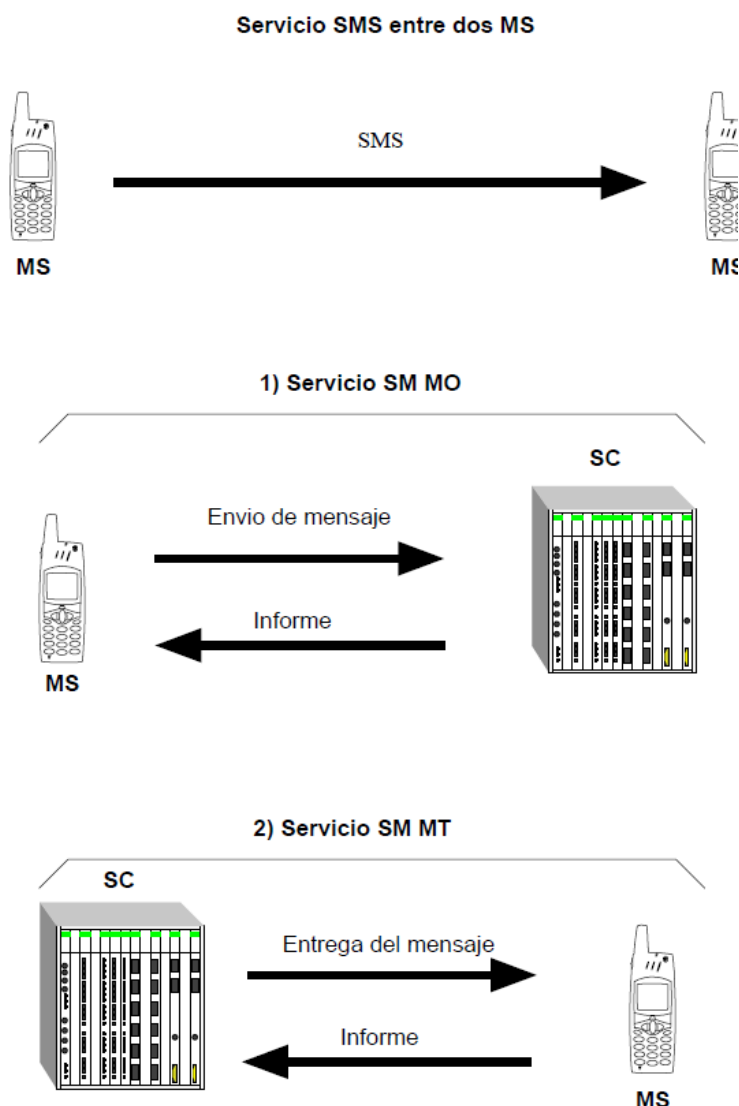
El servicio SMS se divide en dos servicios Básicos:

- SM MT (*Short Message Mobile Terminated Point-to-Point*). Servicio de entrega de un mensaje desde el SC hasta una MS, obteniéndose un informe sobre lo ocurrido.
- SM MO (*ShortMessageMobile Originated Point-to-Point*). Servicio de envío de un mensaje desde una MS hasta un SC, obteniéndose un informe sobre lo ocurrido

#### 2.7.5.1 Arquitectura de la red.

La estructura básica de la red para el servicio SMS se muestra en la siguiente Figura 2.21.





**Figura 2. 21. Servicios básicos SM MT y SM MO.**

- **MS.** Estación móvil
- **MSC.** Centro de conmutación
- **SMS-GMSC.** MSC pasarela para el servicio de mensajes cortos (Servicio SM MT)
- **SMS-IWMSC.** MSC de interconexión entre PLMN y el SC (Servicio SM MO)
- **SC.** Centro de Servicio
- **HLR, VLR.** Para la descripción detallada de la arquitectura, se utiliza un modelo de capas, en el que cada capa o nivel proporciona un servicio a la capa superior, y este servicio se implementa mediante el protocolo correspondiente. La arquitectura se divide en 4 capas.

- **SM-AL** (*Short Message Application Layer*). Nivel de aplicación.
- **SM-TL** (*Short Message Transfer Layer*). Nivel de transferencia. Servicio de transferencia de un mensaje corto entre una MS y un SC (en ambos sentidos) y obtención de los correspondientes informes sobre el resultado de la transmisión. Este servicio hace abstracción de los detalles internos de la red, permitiendo que el nivel de aplicación pueda intercambiar mensajes.
- **SM-RL** (*Short Message Relay Layer*). Nivel de repetición. Proporciona un servicio al nivel de transferencia que le permite enviar TPDU (Transfer Protocol Data Units) a su entidad gemela.
- **SM-LL** (*Short Message Lower Layers*). Niveles inferiores.

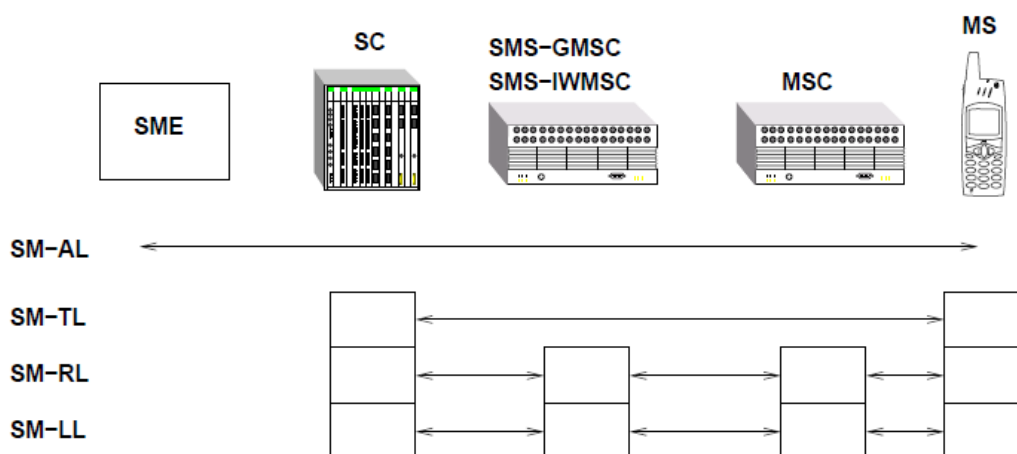


Figura 2. 22. Niveles y servicios para el envío de mensajes cortos.

## 2.7.6 INTERFAZ CON MODEMS

La comunicación con los Módems se realiza a través de una línea serie, y dependiendo del módem, se pueden usar los niveles definidos por la norma RS232 (*Módems para PC*) niveles TTL (*Módems para Circuito impreso*).

El estándar para controlar los módems se basa en los comandos *AT HAYES*, o más comúnmente conocidos como comandos AT. El módem, antes de realizar una conexión con otro módem, se encuentra en modo comando. En este modo podemos configurar y controlar el módem utilizando los comandos AT. Una vez establecida la conexión con un módem remoto, se pasa del modo comando al modo conexión, por lo que la información

que le llega al módem por la línea seria no es interpretada como comandos AT sino como información a transmitir. Una vez terminada la conexión el módem vuelve al modo comando.

Los comandos AT con cadenas ASCII que comienzan por los caracteres AT y terminan con un retorno de carro. Cada vez que el módem recibe un comando, lo procesa y devuelve un resultado, que normalmente es una cadena ASCII salvo que hayamos indicado lo contrario. Al estar la comunicación en ASCII, pondremos utilizar un terminal de comunicaciones desde un ordenador para acceder al módem, bien para configurarlo, bien para hacer pruebas o bien para establecer una comunicación con otro módem.

A continuación se listan algunos comandos AT básicos:

- ATA: responder a una llamada entrante.
- ATD: llamar a un número de teléfono.
- ATE: encendido (1) o apagado (0) de eco de comandos.
- ATF: seleccionar modo de conexión.
- ATH: colgar o descolgar.
- ATL: volumen del altavoz.
- ATM: control del altavoz.
- ATZ: reset del módem.

### **2.7.7 INTERFAZ CON MÓDEMS GSM**

Los módems GSM no sólo se comportan de forma muy parecida a un modem normal, permitiendo el intercambio de datos con otro módem y utilizándose los comandos AT originales, sino que incluyen muchas más características. Son como pequeños teléfonos móviles, que incluyen su propia tarjeta SIM para poder funcionar y por tanto permiten gestionar la base de datos de teléfonos, la lista de los mensajes SMS recibidos, enviar mensajes SMS, configurar diversos parámetros, etc.

Para tener acceso a todos esos servicios, y dado que los comandos AT estaban muy extendidos y muy estandarizados, se ha realizado una ampliación, añadiéndose nuevos comandos. Estos nuevos comandos comienzan por las letras AT+, y se denominan comandos AT+.

### 2.7.7.1 Comandos AT+

En este apartado se listan los comandos AT+ implementados en el módem GSM para el presente proyecto.

- AT+CMGF (*Message Format*). Este comando selecciona el formato de entrada y salida para los mensajes SMS, a ser utilizado por el teléfono. El formato para este comando es el siguiente:
- AT+CMGF=< modo >  
Donde < modo > puede ser 0 para el modo PDU o 1 para el modo texto.
- AT +CNMI (New Message Indications to DTE). Selecciona el procedimiento o la forma en que la se receptorá nuevos mensajes de la red al DTE.  
Sintaxis: AT + CNMI = [<modo> [, <mt> [, <bm> [, <ds> [, <bfr> ]]]]  
<modo>:

0 Todas las indicaciones del buffer en el adaptador de datos.

1 No hay indicaciones cuando el enlace DTE-DCE es reservado (on-line data mode).

2 Todas las indicaciones del buffer en el adaptador de datos cuando el enlace DTE-DCE es reservado.

<mt>:

0 No hay indicaciones de mensajes recibidos al DTE.

1 Las indicaciones de mensajes recibidos son ruteadas al DTE.

2 Los mensajes recibidos son ruteados directamente al DTE usando el código +CMT (Excepto los mensajes de clase 2 indicando +CMTI).

3 Los mensajes de clase 3 son recibidos directamente en el DTE indicando el código +CMT

Y los de otras clases se indican con el código +CMTI.

<bm>:

0 No se rutean mensajes de difusión de celular al DTE.

2 Nuevos mensajes de difusión de celular son ruteados directamente al DTE indicando el código +CMB.

<ds>:

0 Reportes de estado no son ruteados al DTE.

1 Reportes de estado son ruteados al DTE indicando con el código +CDS

<bfr>:

0 Las indicaciones del buffer del adaptador de datos son limpiadas en el DTE cuando

<modo> 1 o 2 es introducido.

1 Las indicaciones del buffer del adaptador de datos son borradas en el DTE cuando

<modo> 1 o 2 es introducido.

### **2.7.8 DISEÑO DE LA INTERFAZ PARA EL DISPOSITIVO MÓVIL.**

Para el diseño de la aplicación para el dispositivo móvil se utilizó para plataforma Java Micro Edition.

La plataforma *Java Micro Edition*, o anteriormente *Java 2 Micro Edition (J2ME)*, es una especificación de un subconjunto de la plataforma Java orientada a proveer una colección certificada de API de desarrollo de software para dispositivos con recursos restringidos. Está orientado a productos de consumo como PDAs, teléfonos móviles o electrodomésticos.

Esta edición tiene unos componentes básicos que la diferencian de las otras versiones, como el uso de una máquina virtual denominada KVM (*Kilo Virtual Machine*, debido a que requiere sólo unos pocos Kilobytes de memoria para funcionar) en vez del uso de la JVM (Java Virtual Machine) clásica.

J2ME contiene una mínima parte de las APIs de Java, esto es debido a que la edición estándar de APIs de Java ocupa 20 Mbytes, y los dispositivos pequeños disponen de una cantidad de memoria mucho más reducida.

### **2.7.9 ENTORNO DE EJECUCIÓN**

Un entorno de ejecución determinado de J2ME se compone entonces de una selección de:

- Máquina virtual (KVM *Kernel-based Virtual Machine*).
- Configuración (una configuración es el conjunto mínimo de APIs Java que permiten desarrollar aplicaciones para un grupo de dispositivos CLDC O CDC).

- Perfil (bibliotecas Java de clases específicas orientadas a implementar funcionalidades de más alto nivel para familias específicas de dispositivos).
- Paquetes Opcionales.

### 2.7.9.1 Máquina virtual

Una máquina virtual Java es un programa encargado de interpretar código intermedio (*bytecode*) de los programas Java pre-compilados, a código máquina ejecutable por la plataforma, efectuar las llamadas pertinentes al sistema operativo subyacente y observar las reglas de seguridad y corrección de código definidas para el lenguaje Java. J2ME define dos máquinas virtuales con diferentes características que son:

- KVM
- CVM

#### 2.7.9.1.1 KVM

Su nombre KVM proviene de Kilobyte (haciendo referencia a la baja ocupación de memoria, entre 40Kb y 80Kb). Se trata de una implementación de Máquina Virtual reducida y especialmente orientada a dispositivos con bajas capacidades computacionales y de memoria. La KVM está escrita en lenguaje C. Fue diseñada para ser:

- Pequeña, con una carga de memoria entre los 40Kb y los 80 Kb, dependiendo de la plataforma y las opciones de compilación.
- Alta portabilidad.
- Modulable.
- Lo más completa y rápida posible y sin sacrificar características para las que fue diseñada.

Sin embargo, esta baja ocupación de memoria hace que posea algunas limitaciones con respecto a la clásica Java Virtual Machine (JVM):

- No hay soporte para tipos en coma flotante ya que existen por tanto los tipos *double* ni *float*. Esta limitación está presente porque los dispositivos carecen del hardware necesario para estas operaciones.

- No existe soporte para JNI (*Java Native Interface*) debido a los recursos limitados de memoria.
- No existen cargadores de clases (*class loaders*) definidos por el usuario. Sólo existen los predefinidos.
- No se permiten los grupos de hilos o *hilos daemon*. Cuando se necesite utilizar grupos de hilos utilizaremos los objetos Colección para almacenar cada hilo en el ámbito de la aplicación.
- No existe la finalización de instancias de clases. No existe el método *Object.finalize()*.
- No hay referencias débiles.
- Limitada capacidad para el manejo de excepciones debido a que el manejo de éstas depende en gran parte de las APIs de cada dispositivo por lo que son éstos los que controlan la mayoría de las excepciones.

#### 2.7.9.1.2 CVM

La CVM (*Compact Virtual Machine*) ha sido tomada como Máquina Virtual Java de referencia para la configuración CDC y soporta las mismas características que la Máquina Virtual de J2SE. Está orientada a dispositivos electrónicos con procesadores de 32 bits de gama alta y en torno a 2Mbytes o más de memoria RAM. Las características que presenta esta Máquina Virtual son:

- Sistema de memoria avanzado.
- Tiempo de espera bajo para el recolector de basura.
- Separación completa de la VM del sistema de memoria.
- Recolector de basura modularizado.
- Portabilidad.
- Rápida sincronización.
- Ejecución de las clases Java fuera de la memoria de sólo lectura (ROM).
- Soporte nativo de hilos.
- Baja ocupación en memoria de las clases.
- Proporciona soporte e interfaces para servicios en Sistemas Operativos de Tiempo Real.

- Conversión de hilos Java a hilos nativos.
- Soporte para todas las características de Java2 v1.3 y librerías de seguridad, referencias débiles, Interfaz Nativa de Java (JNI), invocación remota de métodos (RMI), Interfaz de depuración de la Máquina Virtual (JVMDI).

### 2.7.9.2 MIDLET

Las aplicaciones J2ME desarrolladas bajo la especificación MIDP, se denominan MIDLets. Las clases de un MIDLet, son almacenadas en bytecodes java, dentro de un fichero .class. Estas clases, deben ser verificadas antes de su “puesta en marcha”, para garantizar que no realizan ninguna operación no permitida. Este preverificación, se debe hacer debido a las limitaciones de la máquina virtual usada en estos dispositivos. Esta máquina virtual se denomina KVM. Para mantener esta máquina virtual lo más sencilla y pequeña posible, se elimina esta verificación, y se realiza antes de la entrada en producción. La pre-verificación se realiza después de la compilación, y el resultado es una nueva clase, lista para ser puesta en producción.

Los MIDLets, son empaquetados en ficheros “.jar”. Se requiere alguna información extra, para la puesta en marcha de las aplicaciones. Esta información se almacena en el fichero de “manifiesto”, que va incluido en el fichero “.jar” y en un fichero descriptor, con extensión “.jad”. Un fichero “.jar” típico, por tanto, se compondrá de:

- Clases del MIDLet
- Clases de soporte
- Recursos (imágenes, sonidos...)
- Manifiesto (fichero “.mf”)
- Descriptor (fichero “.jad”)

Un fichero “.jar” puede contener varios MIDLets. Esta colección de MIDLets, se suele llamar “MIDLet Suite”. Esta unión de varios MIDLets en una distribución, permite compartir recursos (imágenes, sonidos...), y por tanto optimizar los recursos del dispositivo.



### 2.7.9.2.1 Descripción del MIDlet

Los MIDlets, deben heredar de la clase `ax.microedition.midlet.MIDlet`, contenida en el API MIDP estándar.

Esta clase define varios métodos, de los cuales destacaremos los siguientes:

- `startApp()` – Lanza el MIDlet
- `pauseApp()` – Para el MIDlet
- `destroyApp()` – Destruye el MIDlet

### 2.7.9.2.2 Estados de un MIDlet

Un MIDlet durante su ejecución pasa por 3 estados diferentes

- **Activo:** El MIDlet está actualmente en ejecución.
- **Pausa:** El MIDlet no está actualmente en ejecución, en este estado el MIDlet no debe usar ningún recurso compartido. Para volver a pasar a ejecución tiene que cambiar su estado a Activo.
- **Destruído:** El MIDlet no está en ejecución ni puede transitar a otro estado.

La Figura. 2.23 muestra el diagrama de estados de un MIDlet en ejecución:

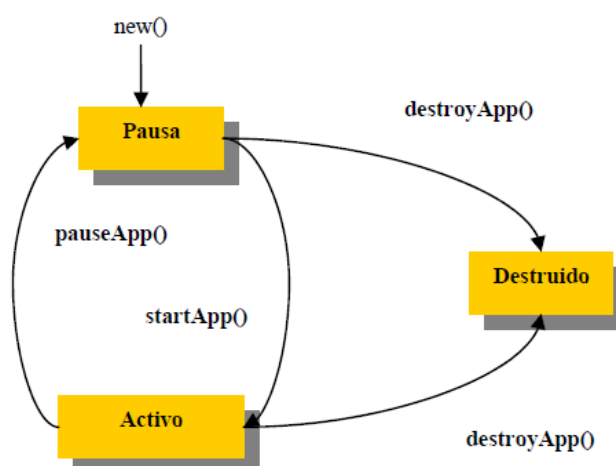


Figura 2. 23. Estados de un MIDlet.[10]

Como vemos en el diagrama, un MIDlet puede cambiar de estado mediante una llamada a los métodos `MIDlet.startApp()`, `MIDlet.pauseApp()` o `MIDlet.destroyApp()`. El gestor de aplicaciones cambia el estado de los MIDlets haciendo una llamada a cualquiera de los métodos anteriores. Un MIDlet también puede cambiar de estado por sí mismo.

Ahora vamos a ver por los estados que pasa un MIDlet durante una ejecución típica y cuáles son las acciones que realiza tanto el AMS como el MIDlet. En primer lugar, se realiza la llamada al constructor del MIDlet pasando éste al estado de “Pausa” durante un corto período de tiempo. El AMS por su parte crea una nueva instancia del MIDlet. Cuando el dispositivo está preparado para ejecutar el MIDlet, el AMS invoca al método `MIDlet.startApp()` para entrar en el estado de “Activo”. El MIDlet entonces, ocupa todos los recursos que necesita para su ejecución. Durante este estado, el MIDlet puede pasar al estado de “Pausa” por una acción del usuario, o bien, por el AMS que reduciría en todo lo posible el uso de los recursos del dispositivo por parte del MIDlet.

Tanto en el estado “Activo” como en el de “Pausa”, el MIDlet puede pasar al estado “Destruído” realizando una llamada al método `MIDlet.destroyApp()`. Esto puede ocurrir porque el MIDlet haya finalizado su ejecución o porque una aplicación prioritaria necesite ser ejecutada en memoria en lugar del MIDlet. Una vez destruido el MIDlet, éste libera todos los recursos ocupados. [11]

#### **2.7.10 RECORD MANAGEMENT SYSTEM**

Un dispositivo móvil (al menos por ahora) no dispone de disco duro donde almacenar información permanentemente. J2ME resuelve el problema mediante el RMS (*Record Management System*)

El RMS (*Record Management System*) es un pequeño sistema de bases de datos que permite añadir información en una memoria no volátil del celular o dispositivo móvil. En una base de datos RMS, el elemento básico es el registro (*record*).

Un registro es la unidad de información más pequeña que puede ser almacenada. Los registros son almacenados en un *recordStore* que puede visualizarse como una colección de registros.

Cuando se almacena un registro en el *recordStore*, a éste se le asigna un identificador que identifica unívocamente al registro. Para poder utilizar RMS se debe importar el paquete *javax.microedition.rms*. Algunas operacion

es básicas con registros son las siguientes:

- Abrir y cerrar un *recordStore*. Para abrir un *recordStore* se utiliza el método *openRecordStore()*, mientras que para cerrar el *recordStore* se utiliza el método *RecordStore.closeRecordStore()*
- Añadir registros. Una vez abierto el *recordStore* se pueden añadir registros con el método *addRecord()*.
- Leer registros. Para leer registros se utiliza el método *getRecord()*, el cual permite acceder al registro deseado, siempre que se conozca su identificador.
- Borrar registros. El borrado de registros se realiza con el método *deleteRecord()*.
- Recorrer registros. Se hace uso del objeto *RecordEnumeration* para recorrer todos los registros almacenados en la base de datos. Para crear una enumeración se utiliza el método *enumerateRecords()*.

Se puede obtener una descripción más detallada en [12].

### 2.7.11 WIRELESS MESSAGING API

WMA es una extensión de las especificaciones de CLDC y MIDP para el envío, la recepción y la gestión de SMS desde MIDlets. El API está compuesto de interfaces ubicadas bajo el paquete *javax.wireless.messaging*. Estas interfaces con sus respectivos métodos se muestran en la Tabla

Los *javax.wireless.messaging.MessageConnection* pueden funcionar de dos modos:

- En modo cliente.- Sólo sirve para enviar SMS a un destinatario. El modo cliente se lo especifica de la siguiente manera:
- `MessageConnection con=(MessageConnection)Connector.open("sms://+5938404066");`

- `MessageConnectioncon=(MessageConnection)Connector.open("sms://+5938404066:16500");`
- En ambos casos el SMS sería enviado al teléfono 08404066 de Ecuador (por el prefijo +593), pero en el primer caso, el SMS sería tratado por la aplicación que por defecto tiene instalada el teléfono, mientras que, en el segundo caso, el SMS sería tratado por la aplicación que esté escuchando en el puerto 16.500.
- En modo servidor.- Sirve para recibir y tratar los SMS que son dirigidos hacia él.
- En este modo también se pueden enviar SMS. El modo servidor se especifica de la siguiente manera:

Tabla 2. 3. Clases en javax.wireless.messaging.

<b>Interfaz</b>	<b>Descripción</b>	<b>Métodos</b>
Message	Interfaz base, de la que derivan TextMessage y BinaryMessage	getAddress() getTimestamp() setAddress()
BinaryMessage	Subinterfaz de Message que proporciona métodos para fijar y detectar el payload binario	getPayloadData() setPayloadData()
TextMessage	Subinterfaz de Message que proporciona métodos para fijar y detectar el payload de texto	getPayloadText() setPayloadText()
MessageConnection	Interfaz a través de la cual se realiza el envío y la recepción de mensajes	newMessage() receive() send() setMessageListener() numberOfSegments()
MessageListener	Define la interfaz listener para implementar la notificación asíncrona de objetos Message	notifyIncomingMessage()

## 2.7.12 COMPONENTES APLICACIONES J2ME

Una aplicación J2ME está formada por un archivo JAR que es el que contiene a la aplicación en sí y un archivo JAD (*Java Archive Descriptor*) que contiene diversa información sobre la aplicación.

El gestor de aplicaciones o AMS (*Application Management System*) es el software encargado de gestionar los MIDlets. El AMS realiza dos grandes funciones:

- Por un lado gestiona el ciclo de vida de los MIDlets.
- Es el encargado de controlar los estados por los que pasa el MIDlet mientras está en la memoria del dispositivo, es decir, en ejecución.

Cuándo un MIDlet comienza su ejecución, está en el estado “Activo” pero, ¿qué ocurre si durante su ejecución recibimos una llamada o un mensaje? El gestor de aplicaciones debe ser capaz de cambiar el estado de la aplicación en función de los eventos externos al ámbito de ejecución de la aplicación que se vayan produciendo. En este caso, el gestor de aplicaciones interrumpiría la ejecución del MIDlet sin que se viese afectada la ejecución de éste y lo pasaría al estado de “Pausa” para atender la llamada o leer el mensaje. Una vez que terminemos de trabajar con el MIDlet y salgamos de él, éste pasaría al estado de “Destruído” dónde sería eliminado de la memoria del dispositivo. Cuándo decimos que el MIDlet pasa al estado “Destruído” y es eliminado de memoria, nos referimos a la memoria volátil del dispositivo que es usada para la ejecución de aplicaciones. Una vez finalizada la ejecución del MIDlet podemos volver a invocarlo las veces que queramos ya que éste permanece en la zona de memoria persistente hasta el momento que deseemos desinstalarlo.

MIDlet puede cambiar de estado mediante una llamada a los métodos *MIDlet.startApp()*, *MIDlet.pauseApp()* o *MIDlet.destroyApp()*. El gestor de aplicaciones cambia el estado de los MIDlets haciendo una llamada a cualquiera de los métodos anteriores. Un MIDlet también puede cambiar de estado por sí mismo. [12]

### 2.7.13 INTRODUCCIÓN A NETBEANS

NetBeans es un entorno de desarrollo integrado (IDE por sus siglas en inglés). Esto quiere decir que integra todas las herramientas que necesitamos para poder desarrollar. Originalmente la programación en Java era algo complicada porque Java cuenta con una enorme cantidad de librerías y funciones que era preciso aprenderse de memoria, viendo esto muchas compañías construyeron diferentes entornos de programación para facilitar la tarea del programador. Entre los más populares surgió Eclipse que reinó como el único y más importante IDE de Java durante varios años. *Sun Microsystems* desarrollo su propio IDE, que tenía la ventaja de que fue creado por las mismas personas que crearon Java años antes, este IDE fue *NetBeans* y después de varios años de desarrollo ha llegado a ser tan útil y poderoso como Eclipse o quizás un poco más.

Por otro lado, *NetBeans Platform* es un *Framework* con una amplia variedad de APIs que resuelven gran cantidad de problemas con los que nos encontramos a la hora de construir una aplicación. Él es el corazón sobre el cual se construye, entre otras aplicaciones, *NetBeansIDE*.

*NetBeans Platform* hace fuerte hincapié sobre la construcción del software de forma modular como muestra la figura 2.24, módulo sobre módulo, y es ahí precisamente donde mayor provecho podremos sacar de esta plataforma, ya que nos ofrece implementados los mecanismos de descubrimiento de nuevos módulos (y de actualizaciones de los existentes) desde repositorios remotos, resolución de dependencias, activación/desactivación de módulos en caliente, comunicación entre los mismos, etc. permitiéndonos preocuparnos por la lógica y rápidamente desplegar nuestras aplicaciones, pudiendo ir extendiendo su funcionalidad a medida que pasa el tiempo.



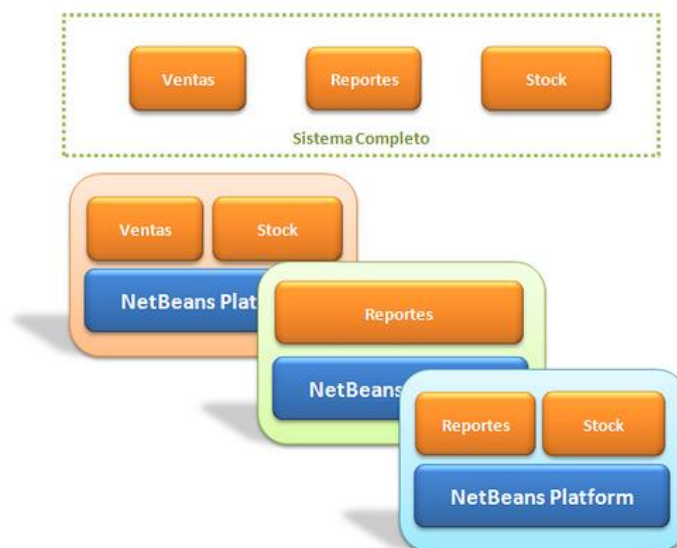
**Figura 2. 24: Aplicación Modular Netbeans.**

Una gran ventaja de la construcción modular es que podemos crear una aplicación conformada por X cantidad de módulos diferentes, cada uno responsable de llevar a cabo determinadas responsabilidades, y según el rol de la persona que la va a utilizarla solo se carga en la aplicación los módulos que permiten cumplir con su tarea, permitiéndonos tener un abanico de aplicaciones sin tener que programar una sola línea de código adicional. Por ejemplo se indica en la Figura 2.25.

Otras características que hacen interesante la elección de *NetBeans Platform* como plataforma para nuestra aplicación son las siguientes:

- Los proyectos desarrollados no dejan de ser multiplataforma, y poseen lanzadores (launchers) para cada plataforma.





**Figura 2. 25: Ejemplo de Aplicación Modular Netbeans**

- Sistema de ventanas práctico para desarrollar las interfaces de usuario
- Sistema de ficheros virtual en el cual se van montan los diferentes módulos con el cual se van adaptando automáticamente los menús, barra de herramientas, menús contextuales, etc. de la aplicación
- Su licencia nos permite construir tanto aplicaciones *open source* como comerciales
- Compatibilidad con *Java Web Start*
- No es obligatorio que una aplicación deba tener interfaz de usuario grafica (GUI), ya que la plataforma permite dejar de lado la misma y seguir disfrutando del resto de los beneficio, por ejemplo la actualización de módulos desde un repositorio remoto.
- Soporte completo para desarrollar desde *NetBeans IDE*, por lo que no necesitaremos otra herramienta adicional para el desarrollo

### 2.7.13.1 Objetos

Objetos generados por el *wizard* del *Netbeans*.

### 2.7.13.2 Displayables

- **Alert** ⚠

Una alerta es una pantalla que muestra los datos al usuario y espera durante un cierto período de tiempo antes de proceder a mostrar la siguiente. Una descripción puede contener una cadena de texto y una imagen.

El uso de Alerta es informar al usuario acerca de errores y otras condiciones excepcionales.

- **Text Box** 

La clase TextBox es una pantalla que permite al usuario introducir y editar texto.

- **Form** 

Un formulario es una pantalla que contiene una mezcla arbitraria de elementos: imágenes, campos de texto de sólo lectura, campos de texto editable, editar los campos de fecha, medidores, grupos de elección, y los temas personalizados. En general, cualquier subclase de la clase puede ser la partida que figura dentro de un formulario. La aplicación se encarga de diseño, transversal, y el desplazamiento.

- **Login Screen** 

La pantalla de inicio de sesión personalizado, constituye un valioso componente de interfaz de usuario estándar con elementos tales como nombre de usuario de campo, Contraseña de campo y botón de acceso. Puede utilizar este componente personalizado para crear la interfaz de usuario para acceder a funciones de protección tales como GSM bancario

- **Splash Screen** 

Splash Screen se utilizan para mejorar el aspecto de una aplicación. Normalmente, se utiliza una pantalla de bienvenida cuando se inicia el programa, o para mostrar un logotipo o marca de información. Se ofrece a los usuarios la primera impresión de su aplicación.

### 2.7.13.3 Commands

Todos los componentes de comandos tienen la misma funcionalidad y comportamiento. El Comando de la clase es un concepto que engloba la información semántica de una acción. El comportamiento que se activa el comando no es encapsulado en este objeto. Esto significa que sólo el componente contiene información acerca de un "comando", y no la acción que se produce cuando el comando está activado. La acción que

se produce se define en el CommandListener mostrar asociados con el. Comando objetos se presentan en la interfaz de usuario. Cómo se presentan pueden depender de la información semántica contenida en el comando.

Aquí tenemos algunos de los comandos:

 Back Command

 Exit Command

 Ok Command

#### 2.7.13.4 Items

- **Choice Group**

Un ChoiceGroup es seleccionar un grupo de elementos destinados a ser introducidos en un formulario. El grupo se puede crear con una modalidad que requiere de una única elección que se hizo o que permite múltiples opciones. La aplicación se encarga de proporcionar la representación gráfica de estos modos, y facilitará los gráficos visualmente diferentes para los distintos modos de transporte. Por ejemplo, podría usar "botones" para el único modo de elección y "casillas" para el modo de selección múltiple.

- **Text Field**

Un campo de texto editable es un componente de texto que se puede poner en un formulario. Se puede dar un trozo de texto que se utiliza como valor inicial

- **Table Item** 

El componente TableItem permite crear rápidamente tablas que constan de una o más columnas, cada uno con una cabecera y un área de datos que se repite para cada registro. El cuadro puede ser más grande que una pantalla en ambas direcciones, los usuarios pueden utilizar un cursor para moverse en todas direcciones.

- **String Item**

Un elemento que puede contener una cadena. Un StringItem es sólo visual el usuario no puede editar el contenido. Tanto la etiqueta y el contenido textual de un StringItem pueden ser modificados por la solicitud. La representación visual de la etiqueta puede diferir de la de los contenidos textuales

### 2.7.13.5 Flujo

Principalmente responsable de la aplicación lógica.

- **If** 

"If" dirige el flujo basado en una determinada condición.

- **Switch** 

"Switch" es similar a la componente "If" pero tiene que especificar los casos mediante la asignación de "Cambiar el asunto" componentes [13].

## CAPITULO III

### MATERIALES Y MÉTODOS

#### 3.1 PROTOTIPO RECEPTOR

El prototipo receptor planteado en el siguiente proyecto es un dispositivo que permite recibir mensajes de texto SMS que son enviados mediante una aplicación en el dispositivo móvil, actuando sobre la aplicación domótica.

##### 3.1.1 Descripción del prototipo receptor.

El prototipo receptor consta de tres etapas para su correcto funcionamiento lo cual se muestra en la figura 3.1

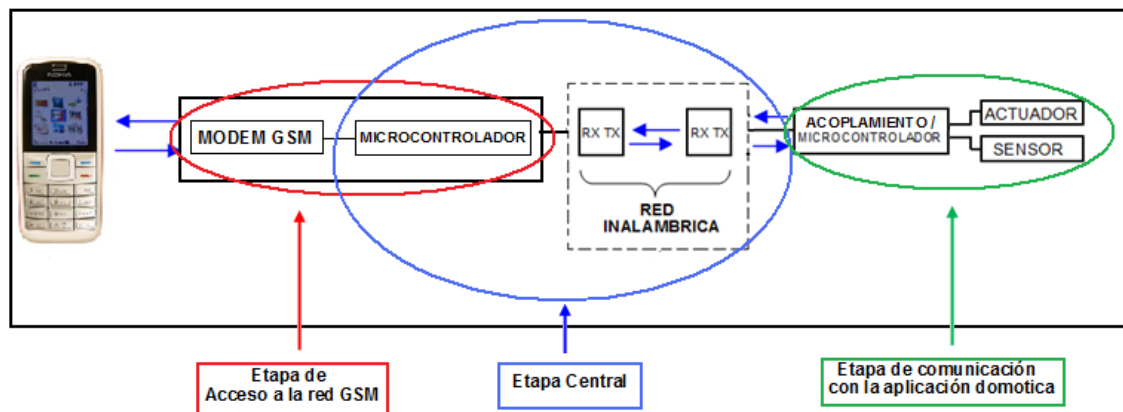


Figura 3. 1. Etapas del receptor

- **Etapa de Acceso a la red GSM.** Contiene un celular que es programado mediante comandos AT para que trabaje como modem GSM, tiene como objetivo comunicarse vía serial con un microcontrolador para manipular el servicio de mensajes de texto.
- **Etapa Central.** El prototipo receptor tiene un microcontrolador que es el encargado de recibir los mensajes del Modem GSM, procesarlos y enviarlos a la red inalámbrica Zigbee.

- **Etapa de comunicación con la aplicación domótica.** Está conformado de un microcontrolador el cual realiza las peticiones solicitadas vía SMS las cuales son recibidas a través de la red inalámbrica Zigbee y mediante comunicación serial activa los sensores y actuadores según corresponda.

### 3.2 HARDWARE DEL PROTOTIPO RECEPTOR

Para la implementación del hardware se va a utilizar un microcontrolador ATmega 164 y un modem GSM como se menciona anteriormente.

#### 3.2.1 Microcontrolador ATmega164

Cuenta con 32 I/O de propósito general, un USART Serial programable, interface SPI master/esclavo generador del reloj. Usualmente un cristal de cuarzo de frecuencias que genera una señal oscilatoria entre 1 y 40 MHz, o también resonadores o circuitos RC, a continuación se muestra la descripción del pines del microcontrolador en la figura 3.2.

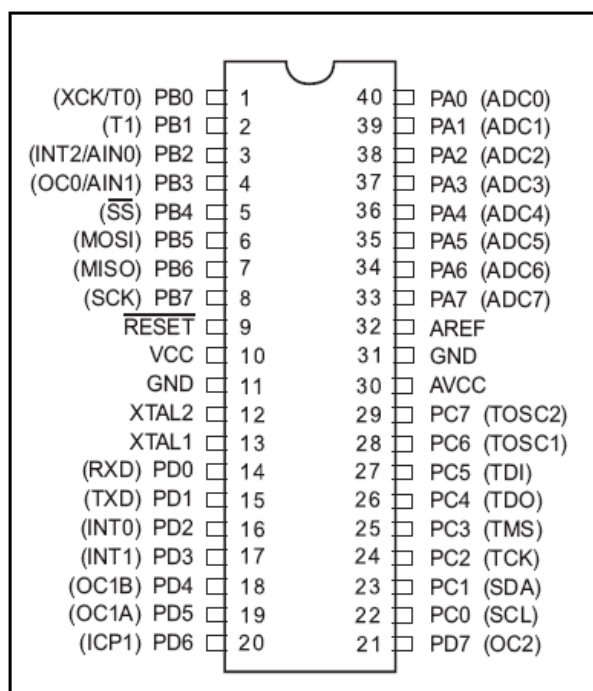


Figura 3. 2. Distribución de pines ATmega164

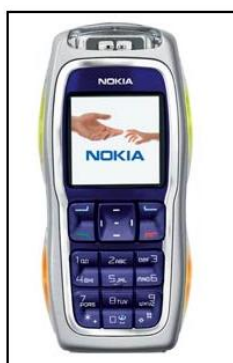
### 3.2.2 Modem GSM

Un módem GSM puede ser un dispositivo externo o una tarjeta PC Card / PCMCIA Card. Por lo general, un módem GSM externo está conectado a un computador mediante un cable serie o un cable USB.

Al igual que un teléfono móvil GSM, un módem GSM requiere una tarjeta SIM de un proveedor de servicios inalámbricos para poder funcionar.

El modem GSM se lo configura mediante líneas de comandos AT para el envío / recepción de mensajes SMS, como se define en la ETSI<sup>4</sup>.

Para el proyecto se utilizó el modem GSM embebido en un teléfono de la marca Nokia, modelo 3220 que se presenta en la Figura 3.3.



**Figura 3. 3. Nokia 3220**

Este teléfono móvil cumple con las siguientes características:

- Tecnología GSM
- Utiliza mensajes SMS
- Velocidades de transmisión de datos hasta 9600 bps.
- Bandas de operación GSM 900 / GSM 1800 / GSM 1900
- Soporta comandos AT. Admite modo texto y modo PDU.

---

<sup>4</sup> **European Telecommunications Standards Institute** (ETSI) o Instituto Europeo de Normas de Telecomunicaciones, es una organización de estandarización de la industria de las telecomunicaciones (fabricantes de equipos y operadores de redes) de Europa, con proyección mundial. El ETSI ha tenido gran éxito al estandarizar el sistema de telefonía móvil GSM.

Al modem embebido se puede acceder mediante un cable de conexión DKU5 como muestra la figura 3.4 el cual va conectado al puerto de teléfono celular como se puede observar en figura 3.5 que mediante los pines 6, 7, 8 se logra la comunicación.



**Figura 3. 4. Cable DKU-5**



**Figura 3. 5. Puerto del teléfono Nokia.**

A continuación se presenta una tabla con la descripción de los pines del puerto del teléfono Nokia.



Tabla 3. 1 Descripción de pines del puerto del teléfono Nokia.

# Pin	Nombre Pin	Descripción
1	Vin	Voltaje de Entrada
2	GND	Tierra
3	ACI	Accesorio interfaz de control (corta con el pin 2 para el reconocimiento de manos libres)
4	V Out	Conectado al pin 3 en DKU2 cable de datos
5	USB Vbus	También actúan como la detección de alimentación USB Debe estar conectado al USB en el pin 1 cable de datos USB
6	FBus Rx/USB D+	USB existe sólo en algunos modelos. Debe estar conectado al USB en el pin 3 del cable de datos USB.
7	FBus Tx/USB D-	USB existe sólo en algunos modelos *. Debe estar conectado a USB pin 2 del cable de datos USB
8	GND	Data GND
9	X Mic-	Entrada de audio - Ext. Entrada para micrófono (-)
10	X Mic+	Entrada de audio - Ext. Entrada para micrófono (+)
11	HS Ear L-	Salida de audio - Ext. Salida de audio - a la izquierda, (-)
12	HS Ear L+	Salida de audio - Ext. Salida de audio - a la izquierda, (+)
13	HS Ear R-	Salida de audio - Ext. salida de audio - a la derecha, (-)
14	HS Ear R+	Salida de audio - Ext. Salida de audio - positiva derecha. Pines 10-14 para utilizar las conexiones de antena
	GND	Tierra

En la Figura 3.6 se muestra la conexión entre el módem GSM y el microcontrolador ATmega16.

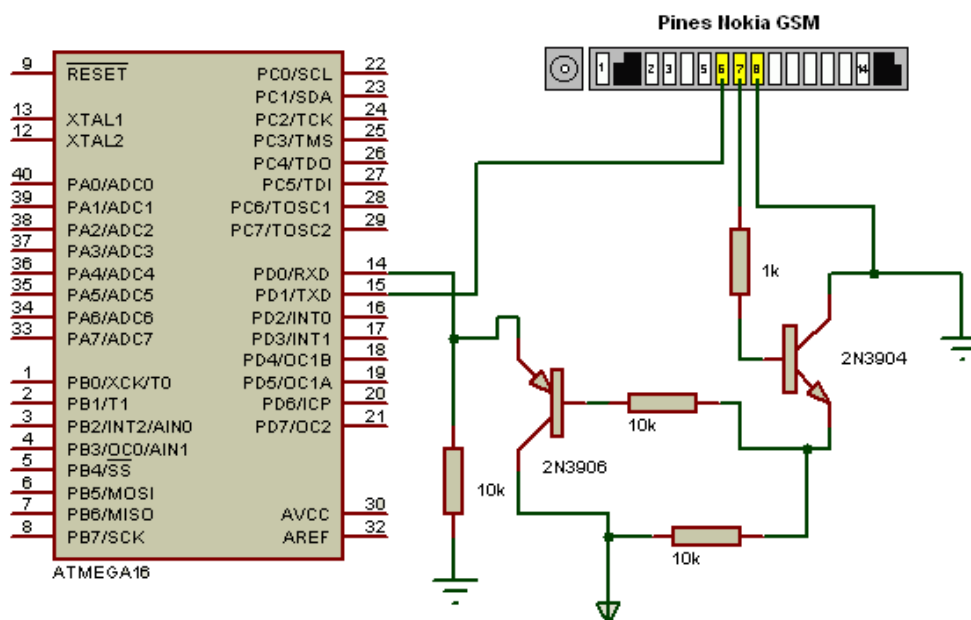


Figura 3. 6. Conexión entre el módem GSM y el microcontrolador ATmega16.

Es importante mencionar que los pines de transmisión y recepción del microcontrolador trabajan con niveles TTL de 0-0,8 V para el estado 0 lógico, y 2-5 V para el estado 1 lógico; mientras que el módem GSM del teléfono celular trabaja con niveles TTL de 0 V (estado 0 lógico) y 3,3 V (estado 1 lógico). Por esta razón, fue necesario añadir un circuito regulador para obtener el voltaje deseado en el pin RX evitando incompatibilidad de niveles de voltaje. En el otro sentido de la comunicación serial no se presenta este inconveniente de incompatibilidad.

### 3.3 SOFTWARE DEL PROTOTIPO RECEPTOR.

#### 3.3.1 Programa del microcontrolador

Para el desarrollo del programa se utilizó un compilador llamado BASCOM-AVR.

La parte principal del programa es establecer la comunicación con el MODEM GSM el cual se realiza mediante comandos AT la cual se describe a continuación

Print "AT" + Chr(13) + Chr(10) inicia la comunicación

Print "ATE0" + Chr(13) + Chr(10) ; Apaga el hecho

Configura el MODEM GSM como modo texto a los mensajes SMS

Print "AT+CMGF=1" + Chr(13) + Chr(10)

Habilita el paso directo del sms al microcontrolador sin que el MODEM GSM reciba el sms en la buzón de entrada

Print "AT+CNMI=1,2,0,0,0" + Chr(13) + Chr(10)

Se realiza un programa para las subfunciones que realizan lo que detalla a continuación:

Dim B As Byte

Guarda la respuesta del celular ("ok", "error"=)

Dim Sret As String \* 20

Guarda El Mensaje A Enviar( "SENSOR VENTANA1 ACTIVADO")

Dim Mensa As String \* 50

Guardamos el mensaje recibido ("ALARMA ON")

Asignamos el numero móvil del usuario

Numero = "+59384404066"

SUBFUNCION PARA CONFIGURACION INICIAL DEL CELULAR

Declare Sub Config\_inicial()

SUBFUNCION PARA OBTENER OK

Declare Sub Getok(s As String)

SUBFUNCION PARA LIMPIAR BUFFER

Declare Sub Limpiarbuffer()

SUBFUNCION PARA ENVIAR MENSAJE

Declare Sub Enviarmensaje(s As String , N As String)

SUBFUNCION PARA RECIBIR MENSAJE

Declare Sub Recibirmensaje()

SUBFUNCION PARA VALIDAR MENSAJE

Declare Sub Validarmensaje(s As String)

### **3.4 DISEÑO DE LA WSAN**

Para el desarrollo de las redes WSAN (Redes inalámbricas de Actuadores y Sensores) se empleará el kit de desarrollo de MAXSTREAM que contienen los módulos Xbee.

#### **3.4.1 Programación de módulos Xbee**

Existen dos tipos de interfaces, serial y USB, que pueden ser utilizadas para programar los módulos Xbee con un software propietario llamado X-CTU. Con este software se puede definir de una forma rápida todos los parámetros que se quiera modificar en los módulos.

#### **3.4.2 Instalación del programa X-CTU**

Esta instalación se puede realizar tanto en linux (a través del programa *wine*) como en Windows.

A continuación aparecerá la pantalla de bienvenida en la que pulsaremos el botón “Next”.

Seguido, aparecerá una pantalla de acuerdo de licencia, en la cual tendremos que seleccionar la opción “I Agree” y pulsar “Next” para continuar.

A continuación se indica la carpeta de instalación, si queremos cambiar la carpeta en la que queremos que se instale el programa pulsamos en “Browser”, sino dejamos la carpeta por defecto y pulsamos “Next”

Aparecerá una pantalla de confirmación de instalación se debe escoger la opción “Next” dando paso a una pantalla de actualización a la que se pulsará el botón “Si”.

Una vez terminada la actualización, se presenta una pantalla con la que nos informa que la instalación ha sido completada, pulsando en “Close” para terminar la instalación del programa.

Cuando se haya finalizado la instalación podemos ejecutar el programa a través de un acceso directo ubicado en el escritorio, llamado X-CTU y se desplegará una pantalla como se muestra en la siguiente figura 3.7.

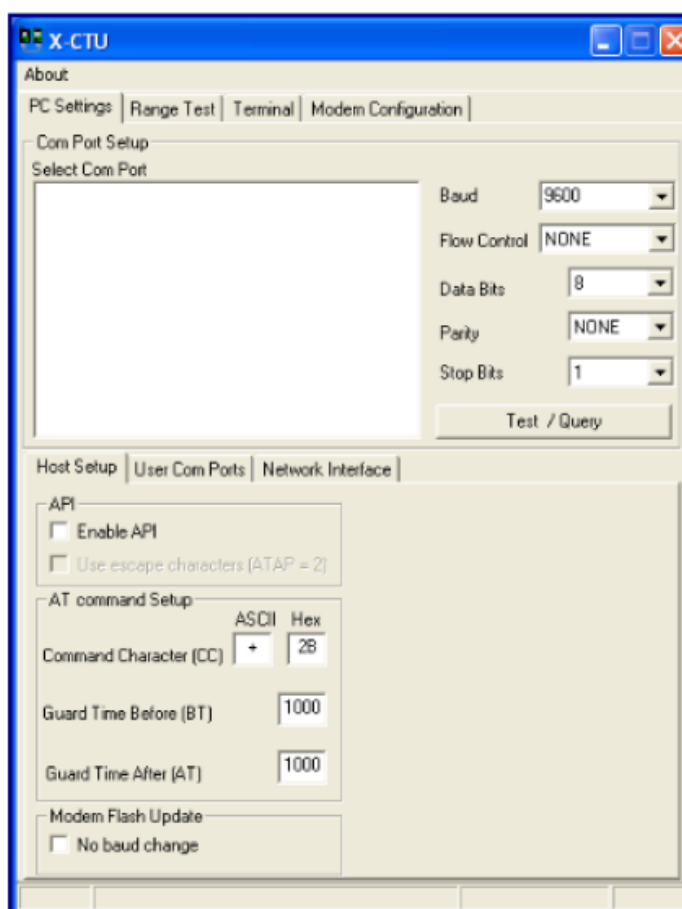


Figura 3. 7. Pantalla de recolección de información

Esta pantalla será la encargada de reunir la información correspondiente para la programación de los módulos Xbee cuando se encuentren conectados a la Pc.

### 3.4.3 Configuración del Puerto X-CTU

Para la configuración de los módulos Xbee se va a utilizar el programa X-CTU, se abre cuantas pantallas se desee de acuerdo al número de tarjetas conectadas a la PC, para este caso son dos ya que se va a programar la recepción y transmisión simultánea para la comunicación inalámbrica.

Dentro del Kit de desarrollo Digi existen dos tipos de zócalos, Serial y USB, los cuales se pueden utilizar indistintamente con los módulos Xbee.

Además contiene dos tipos de módulos:

- Módulos Xbee series 2
- Módulos Xbee PRO series 2

#### 3.4.3.1 Módulos Xbee PRO Serie 2

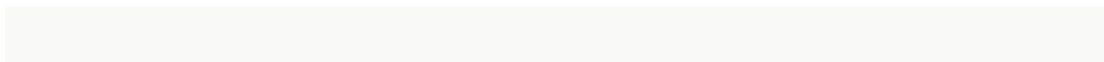


Figura 3. 8. Modulo Xbee PRO serie2

#### Características:

- Módulo para comunicación por RF
- Alcance máximo posible: 1600 metros en exteriores (línea de vista) y 300 metros en interiores
- Alimentación 3.3VDC
- Frecuencia de trabajo 2.4 [GHz]

- Antena desmontable
- Modos de trabajo AT y API



### 3.4.3.2 Módulos Xbee series 2



Figura 3. 9. Modulo Xbee Series 2

#### Características:

- Módulo para comunicación por RF
- Alcance máximo posible: 120 metros en exteriores (línea de vista) y 30 metros en interiores
- Alimentación 3.3VDC
- Frecuencia de trabajo 2.4 [GHz]
- Antena de alambre flexible de 1 pulgada
- Modos de trabajo AT y API

A continuación se mostrara la respectiva configuración de los diferentes módulos.

1. Modo Terminal
2. *Modem Configuration*

Si utilizamos la configuración a través de “Terminal” la configuración será mediante líneas de comandos.

Se debe escribir +++ antes de escribir cualquier comando.

Los comandos que va a utilizar para la configuración del modulo son:

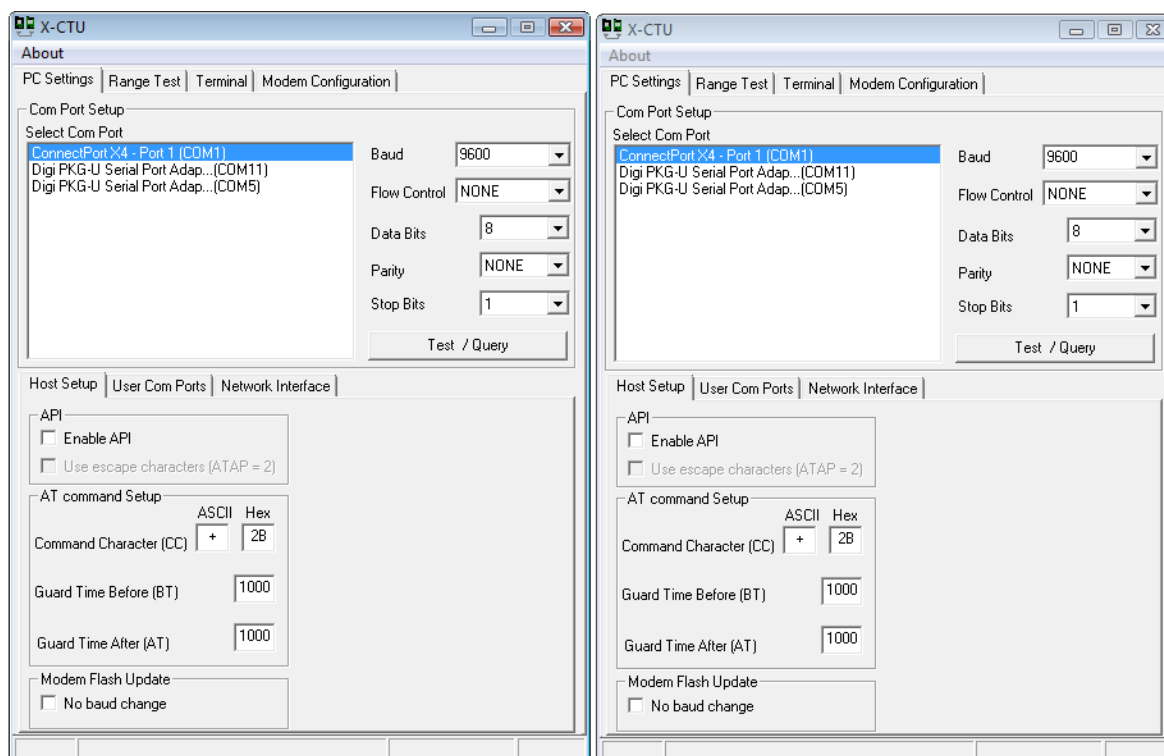
- **ATRE.** Restaura los valores predeterminados de fábrica antes de realizar cualquier modificación.
- **ATAP\*.** Configuración de la API de Xbee.



- **ATCE1.** Configuración del módulo Xbee en modo Coordinador.
- **ATMY\*.** Dirección del módulo Xbee en modo Coordinador.
- **ATID\*.** ID de la conexión que vamos a crear entre nuestros módulos Xbee.
- **ATCH\*.** Canal por el cual los módulos Xbee se van a conectar. El valor de \* (ATCH16C)
- **ATWR.** Escribe una nueva configuración en la memoria no volátil. Si no se escribiese este comando, las modificaciones realizadas solo duraría hasta que el módulo se quede sin batería.
- **ATFR.** Reinicia el módulo Xbee.

Si utilizamos la configuración “*Modem configuration*” se despliegan los parámetros para proceder a la programación mediante asignación de direcciones de los módulos.

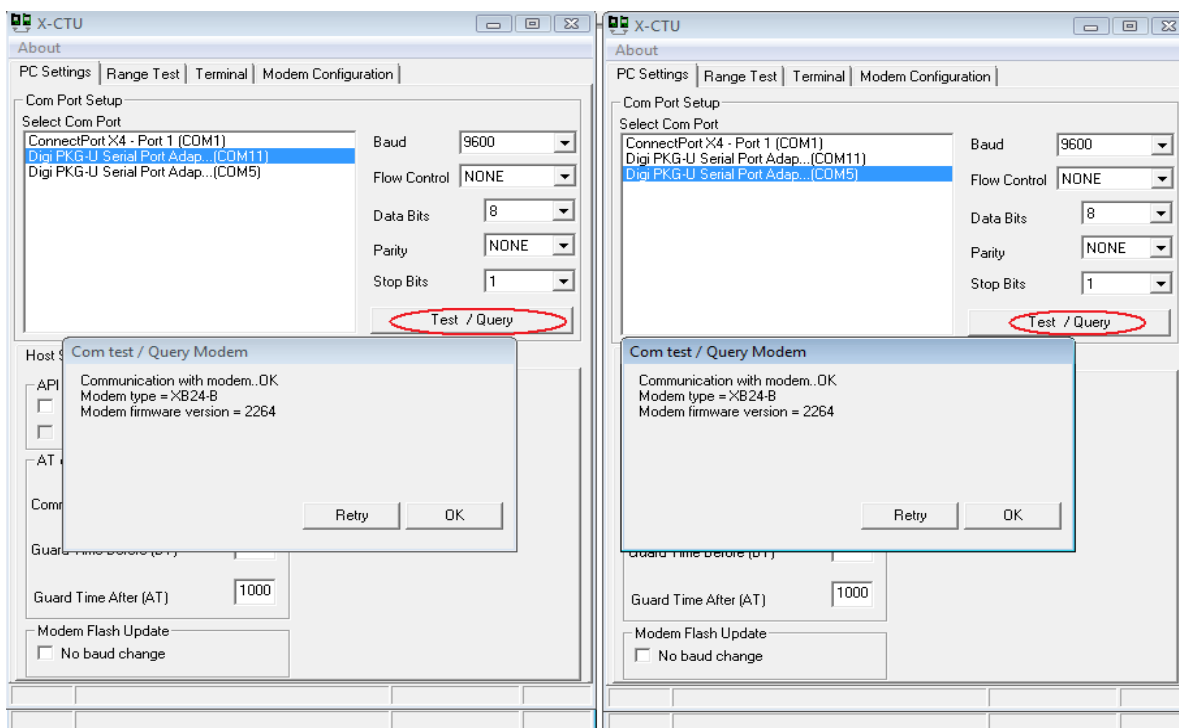
En la figura 3.10 se muestran las pantallas de inicialización del programa X-CTU, con los respectivos puertos conectados.



**Figura 3. 10. Pantallas de inicialización de X-CTU**

Una vez que la PC ha reconocido los puertos se procede a realiza un “*test/Query*” para comprobar la conectividad de las tarjetas, apareciendo como resultado un mensaje que

comprobación con el tipo de modem y la versión del firmware, como se muestra en la figura 3. 11.



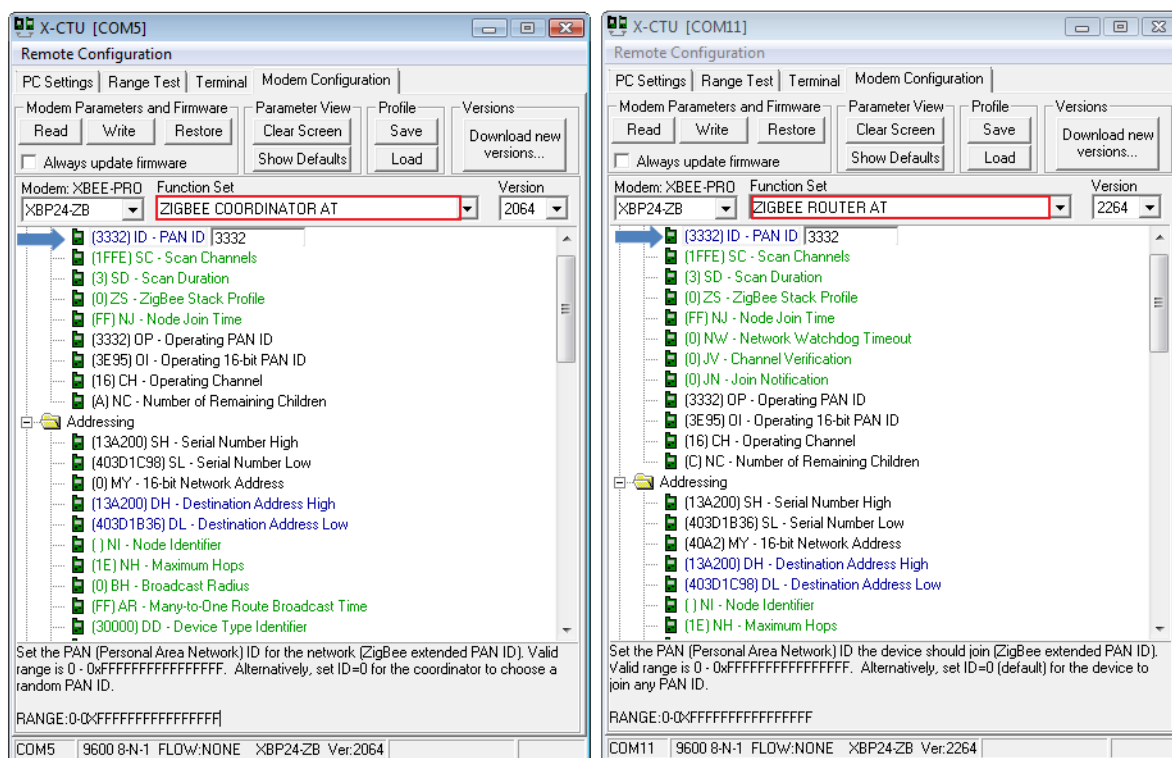
**Figura 3. 11. Prueba de conectividad**

Es en este paso de configuración es donde se diferencia la programación de los módulos Xbee PRO series 2 de los módulos Xbee series 2.

### 3.4.3.3 Programación para el módulo Xbee PRO series 2.

Si los módulos están pre-programados se puede utilizar la opción READ para ver la configuración, en este caso no se tiene programación alguna por lo cual se va a realizar la programación paso a paso.

El primer modem será COORDINATOR AT y en el segundo ROUTER AT como muestra la figura 3.12



**Figura 3. 12. Configuración del modo Coordinador y el Router**

Se debe configurar el direccionamiento, en ambos módulos se debe tener el mismo PAN ID, ahora en el COORDINATOR se configura DH y DL (*Destination Address*) con SH y SL (*Serial Number*) del ROUTER, y en el ROUTER se configura DH y DL con SH y SL del COORDINATOR como se indica en la figura 3.13.

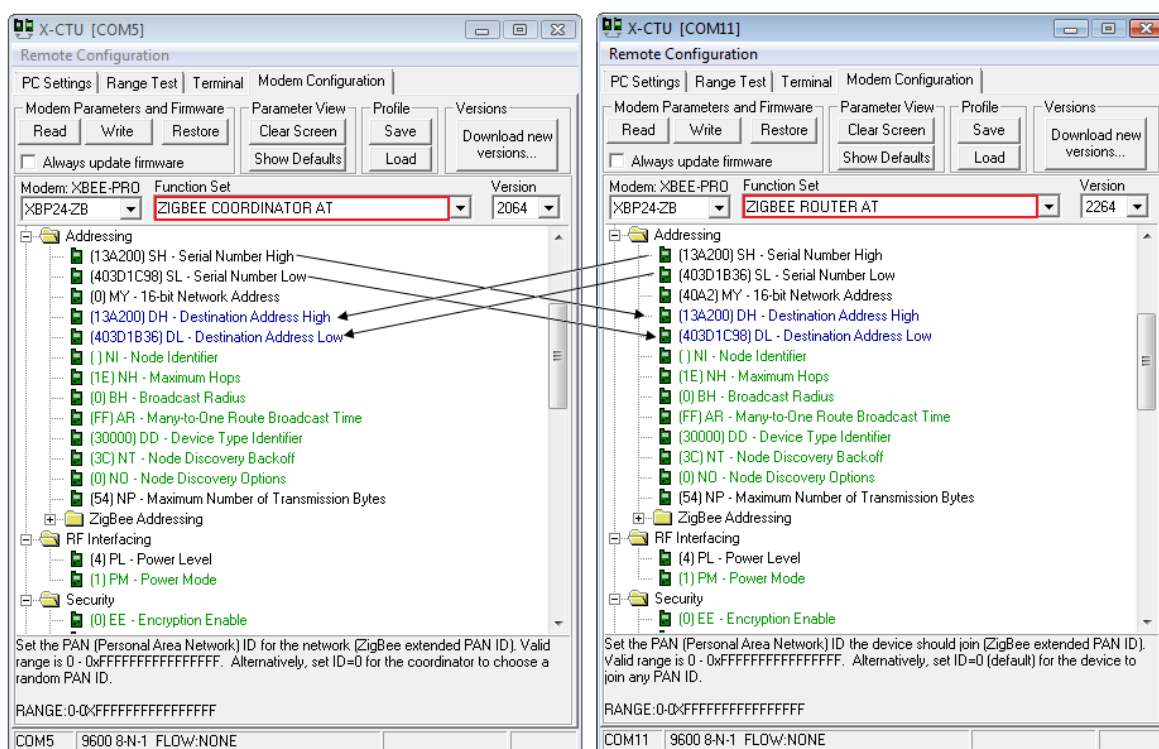


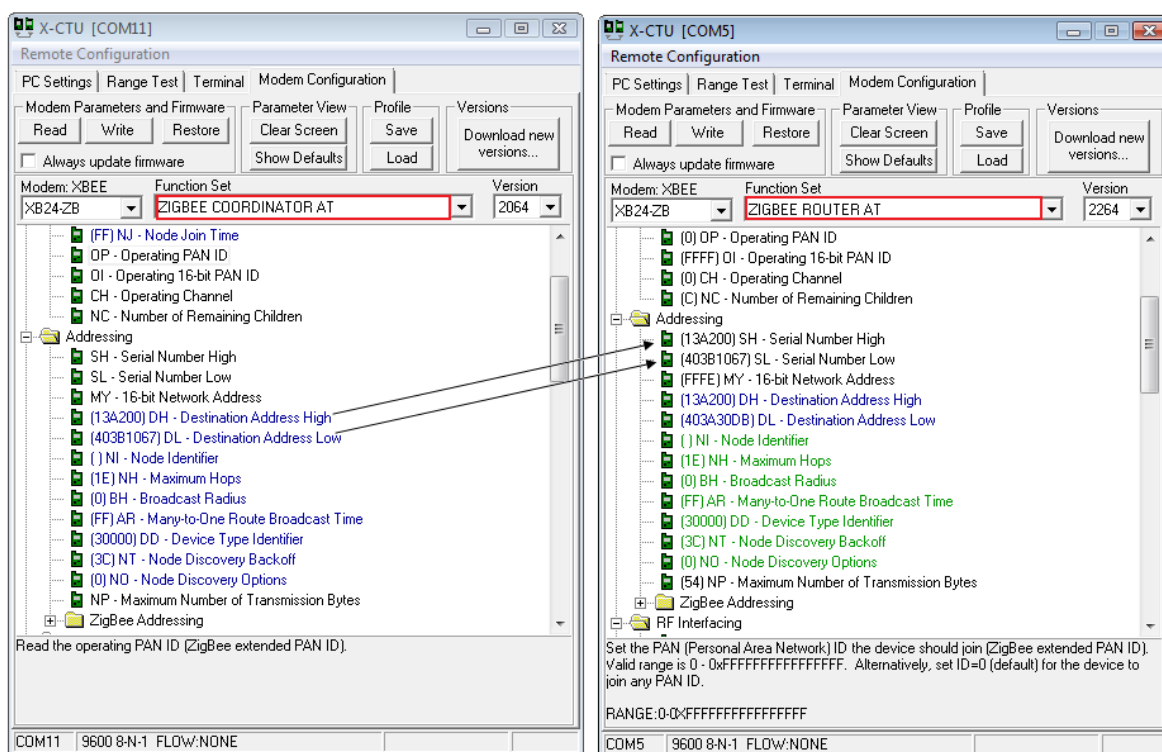
Figura 3. 13. Direccionamiento de los módulos Xbee PRO series 2

#### 3.4.3.4 Para el módulo Xbee Series 2

Si los módulos están pre-programados se puede utilizar la opción *READ* para ver la configuración, en este caso no se tiene programación alguna por lo cual se va a realizar la programación paso a paso.

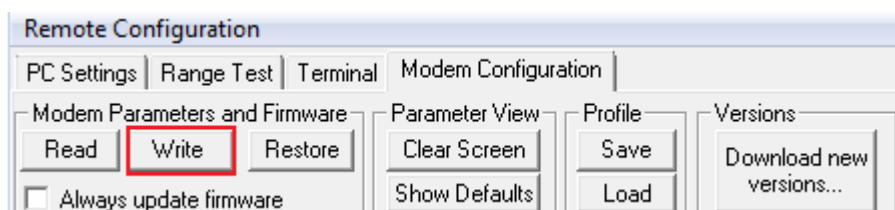
El primer modem será COORDINATOR AT y en el segundo ROUTER AT, para estos modulos se tiene diferente interfaz de configuración.

Para configurar el direccionamiento, en ambos módulos se debe tener el mismo PAN ID, ahora al Router configuramos DH y DL (*Destination Address*) con SH y SL (*Serial Number*) del COORDINADOR, como se puede observar en la figura 3.14.



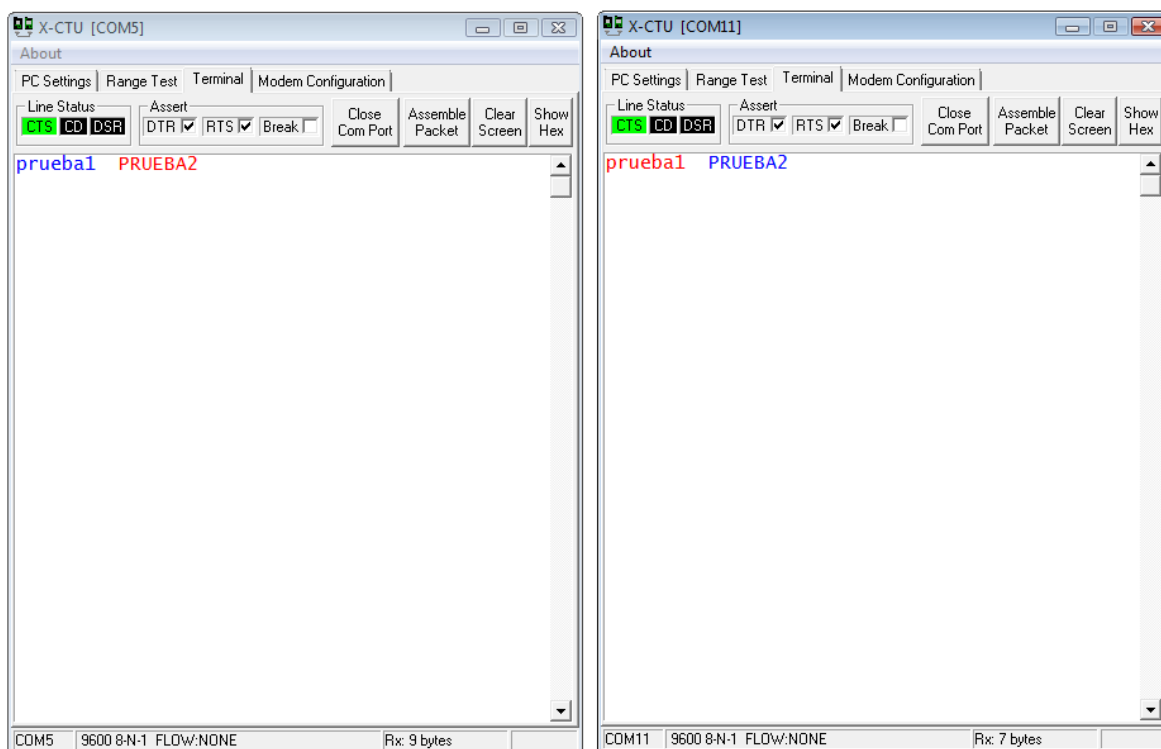
**Figura 3. 14. Direccionamiento de los módulos Xbee series 2**

Una vez realizada la configuración sea de los módulos Xbee PRO series 2 o Xbee series 2, se procede a escoger con un Clic en la opción **WRITE** como muestra la figura figura 3.15 para que suba la nueva configuración al módulo. Cuando termine de programar ya estarán listos para transmitir y recibir datos.



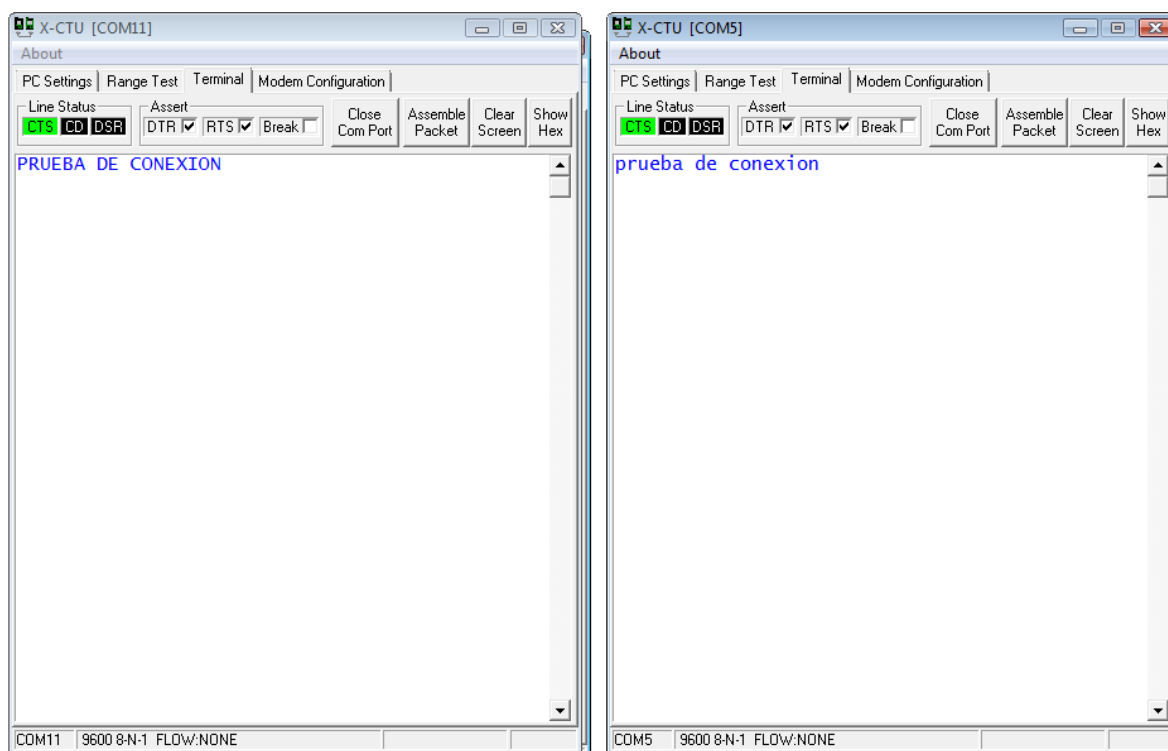
**Figura 3. 15. Opción Write**

Para probar la comunicación se utiliza el modo “terminal” para cualquier tipo de módulo, se mostrara en azul lo que se transmite y en rojo lo que recibe, como se muestra en la figura 3.16.



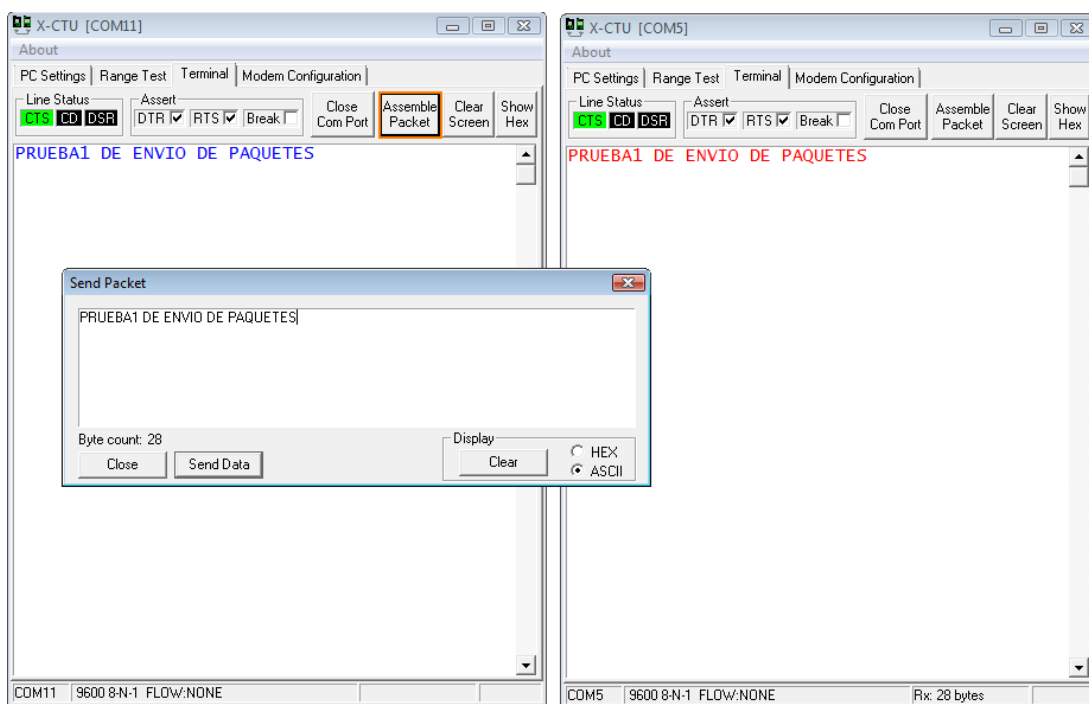
**Figura 3. 16. Modo terminal**

En caso de no recibir ni transmitir información en el modo terminal como muestra la figura 3.17, se recomienda revisar la configuración en la opción “*modem configuration*” que es donde se ingresa las direcciones destino de los módulos a interconectar.

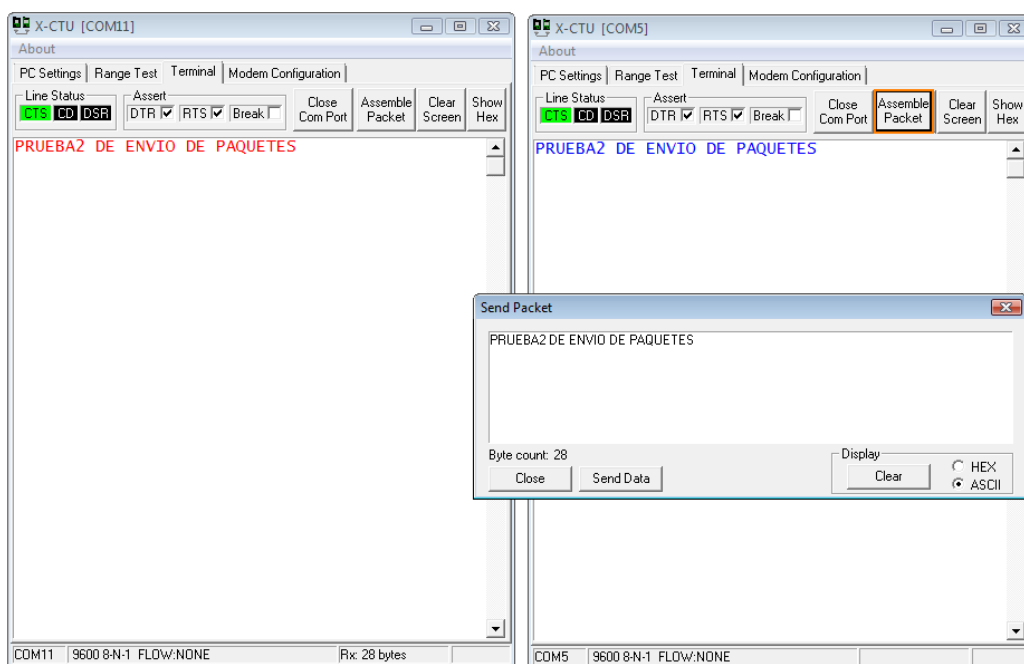


**Figura 3. 17. Falla de transmisión y recepción**

También se puede realizar la transmisión y recepción de paquetes con la respectiva opción como se puede observar en la figura 3.18 y 3.19.



**Figura 3. 18. Transmisión de paquetes**



**Figura 3. 19. Recepción de paquetes**

Una vez establecida la comunicación y con las pruebas respectivas se tienen los módulos listos e interconectados inalámbricamente para realizar la interconexión con el microcontrolador que se detallara a continuación.



### 3.5 PROTOTIPO RECEPTOR CON INTERCONEXION DE TARJETAS XBEE

Una vez realizado la configuración de las tarjetas Xbee y el programa del microcontrolador, se procede a realizar las placas respectivas con el coordinador Xbee como muestra la figura 3.20 y con el Router Xbee como muestra la figura 3.21 para realizar las pruebas respectivas.

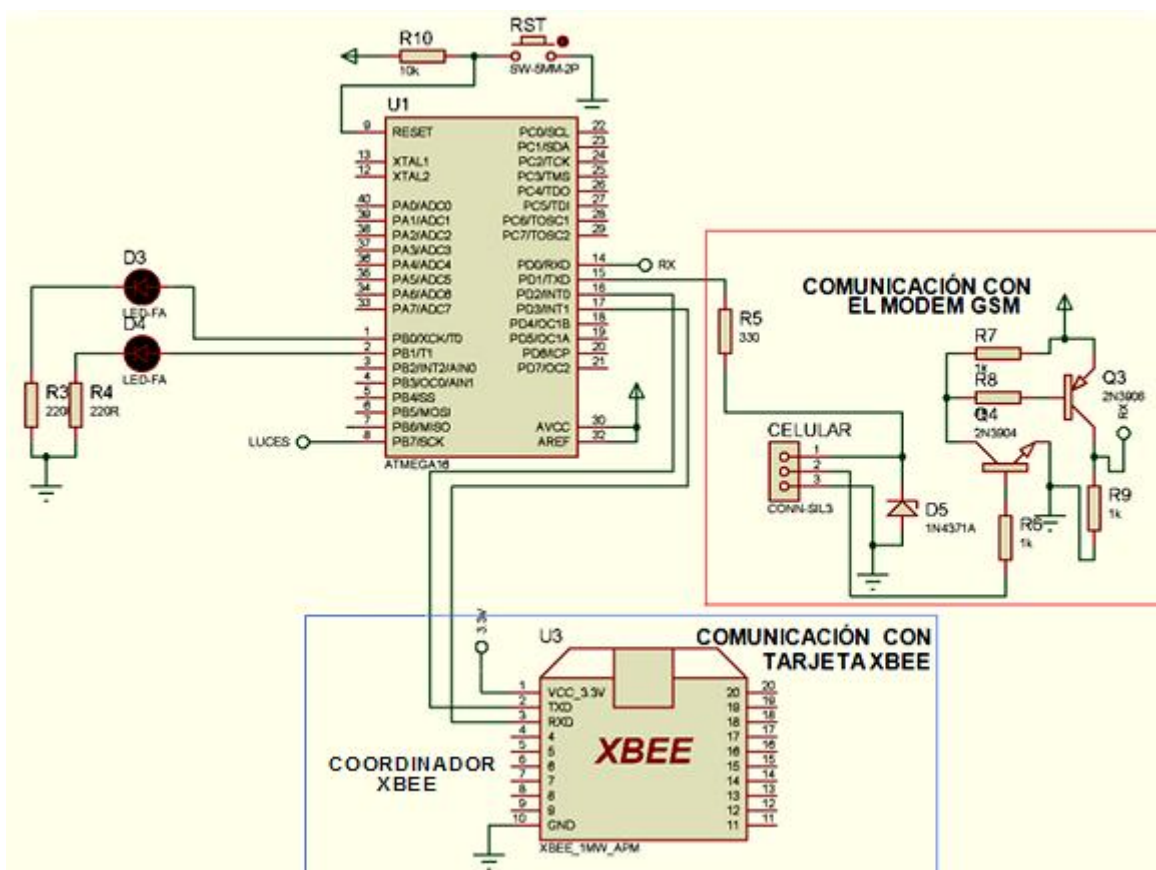


Figura 3. 20. Prototipo receptor coordinador Xbee

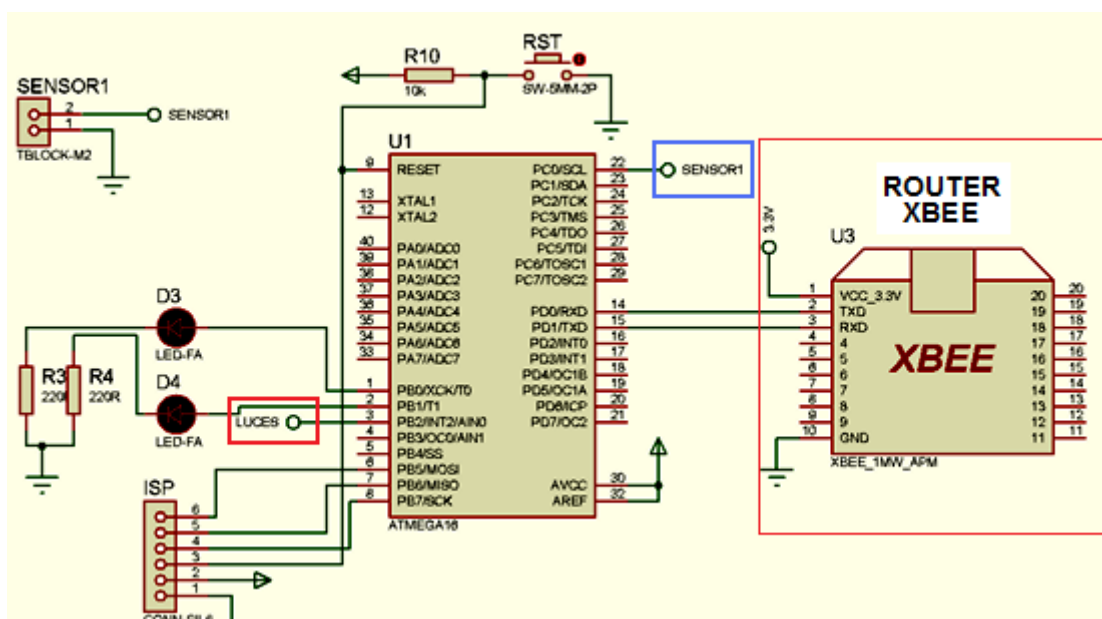


Figura 3. 21. Prototipo Router Xbee

### 3.6 IMPLEMENTACIÓN DE LA APLICACIÓN DOMÓTICA

Para la aplicación domótica se estableció que se va a implementar un control de iluminación y detección de un sensor de presencia, considerando que la red inalámbrica es bidireccional como muestra la figura 3.22.

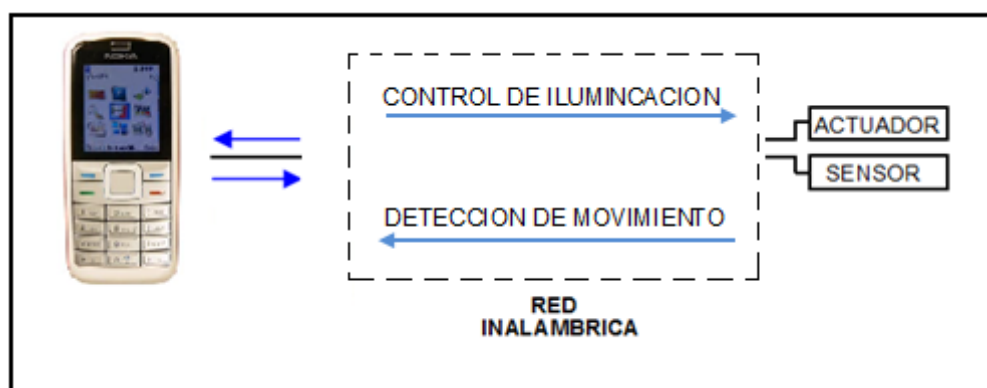


Figura 3. 22. Red inalámbrica bidireccional

### 3.7 INTERFAZ GRÁFICA PARA EL DISPOSITIVO MÓVIL.

La aplicación del dispositivo móvil es desarrollada para facilitar al usuario el manejo de control domótico.

#### 3.7.1 Descripción

La interfaz para el dispositivo móvil es una aplicación desarrollada en *Netbeans* sobre la plataforma *J2ME*.

La aplicación permite:

- Realizar la activación o desactivación del actuador, en este proyecto es una luminaria de 110V.
- Recibir un SMS en el buzón de entrada del dispositivo móvil, cuando el sensor de movimiento se activa o desactiva.

#### 3.7.2 Desarrollo

- **Pantalla de saludo**

Es la primera pantalla que aparece como bienvenida a la aplicación.

- **Pantalla de inicio de sesión**

Esta pantalla aparece al momento de ejecutar la aplicación, es la presentación del proyecto.

- **Pantalla de Menú de Operaciones de Usuario**

Esta pantalla se encarga de presentar las opciones que tiene la aplicación, en este caso presenta la opción de iluminación.

Al entrar a la opción de iluminación se despliega una cuarta pantalla con la opción de ON para encender el actuador y OFF para apagar el actuador.

### 3.7.3 Código de configuración

Se realiza un comparador para determinar la opción que escoge el usuario si desea encender o apagar el actuador como se observa en el código a continuación.

```
if (select_operacion1==1){  
  
    if (select_iluminacion==1){  
  
        com="LUZON";  
  
    }  
  
    if (select_iluminacion==2){  
  
        com="LUZOFF";  
  
    }  
}
```

Una vez escogida la opción se debe especificar el número de la tarjeta sim que se va a enviar el SMS.

```
public void enviar(String comando){  
  
    try{  
  
        sms_enviar=comando;  
  
        String num="sms://099067997";  
  
        MessageConnection conn=(MessageConnection)Connector.open(num);  
  
        TextMessage  
msg1=(TextMessage)conn.newMessage(MessageConnection.TEXT_MESSAGE);  
  
        msg1.setPayloadText(sms_enviar);  
  
        conn.send(msg1);  
}
```

```
conn.close();  
  
}catch(Exception e){  
  
    System.out.println("Error enviando mensaje:"+e.toString());  
  
}
```

## CAPITULO IV

### PRUEBAS Y RESULTADOS

Este capítulo describe las pruebas del funcionamiento de los equipos que conforman el proyecto y los resultados obtenidos en las pruebas realizadas.

#### 4.1 PRUEBAS DEL PROTOTIPO.

En la figura 4.1 se presenta el diagrama de proyecto previo a la descripción detallada del mismo.

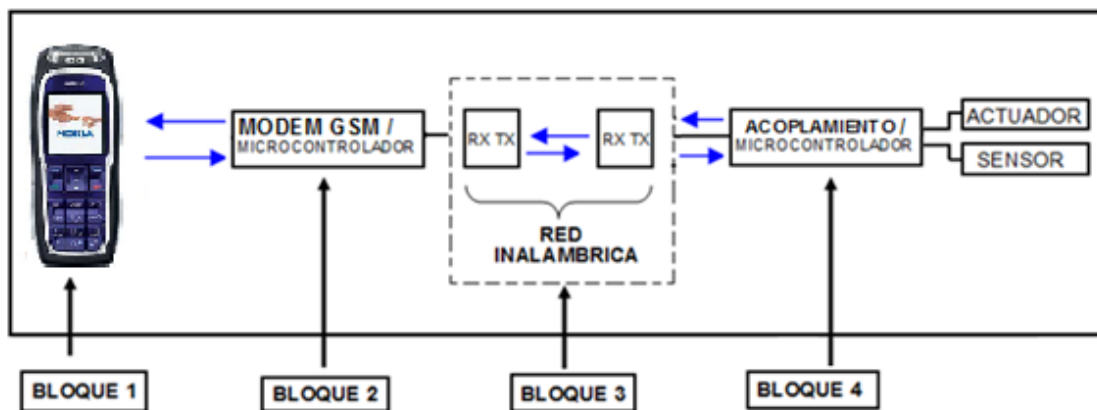


Figura 4. 1. Sistema de Control Domótico: Diseño funcional por bloques

##### 4.1.1 Descripción y prueba del bloque 1.

Este bloque 1 contiene un teléfono celular en donde se encuentra instalada la aplicación para realizar el control de la red.

La aplicación se la realizó en un programa llamado *NETBEANS*

Una vez realizada la aplicación se crean cuatro carpetas *built*, *dist*, *nbproject*, *scr*, dentro de la carpeta “*dist*” se encuentra el nombre del proyecto con extensión *.jar* (*domo.jar*) que va a ser el ejecutable de la aplicación y el que debe ser instalado en el teléfono móvil.

Para realizar las pruebas respectivas será necesario interconectar con los bloques que se describen a continuación.

#### **4.1.2 Descripción y prueba del bloque 2.**

Este bloque consta dos partes.

- Microcontrolador ATmega164 para transmisión y recepción de SMS
- El Modem GSM: (*Global System for Mobile communications*)

##### **4.1.2.1 Pruebas y resultados del microcontrolador ATmega164.**

Para analizar el funcionamiento del ATmega164 se realizó una aplicación utilizando un solo microcontrolador que simule la recepción de un sensor que se muestra en el anexo 4.1.

En caso de que exista movimiento, el sensor se encarga de enviar una señal a una luz indicadora LED y presentar en el LCD el estado del mismo, si se encuentra activado o desactivado.

En la figura 4.2 se puede observar los elementos para la simulación y la figura 4.3 muestra la activación del sensor.

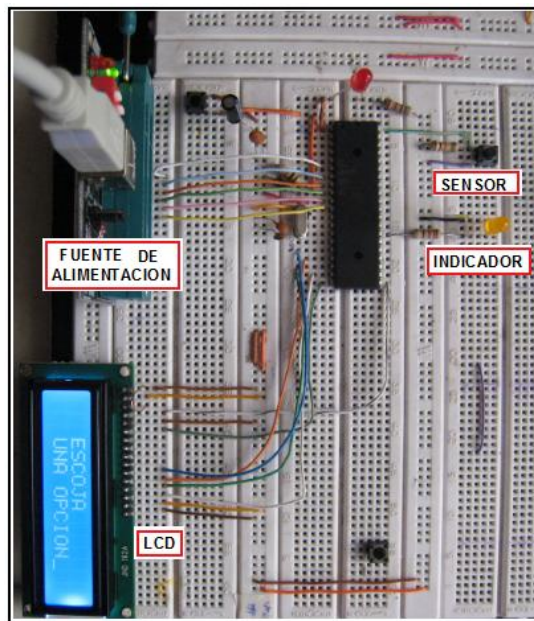


Figura 4. 2. Circuito de prueba de funcionamiento del ATmega16.

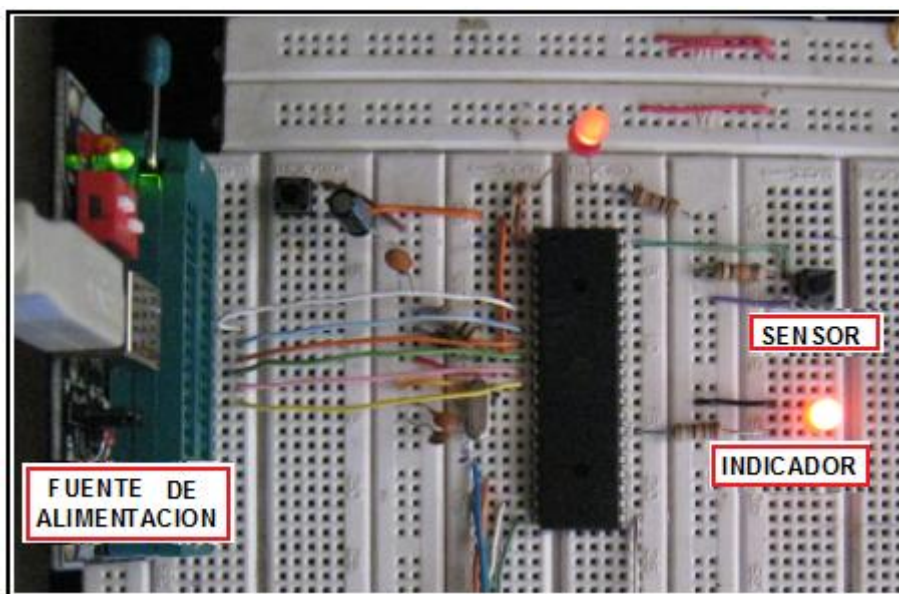


Figura 4. 3. Simulación de activación del sensor



En la tabla 4.1 se muestra los resultados del funcionamiento del microcontrolador.

**Tabla 4. 1 Resultados de funcionamiento del microcontrolador ATmega 164**

<b>ACTIVIDAD</b>	<b>SI</b>	<b>NO</b>
Funcionamiento de fuente de alimentación de 5V	X	
Muestra en LCD el estado del sensor	X	
Luz indicadora se enciende al activar el sensor	X	

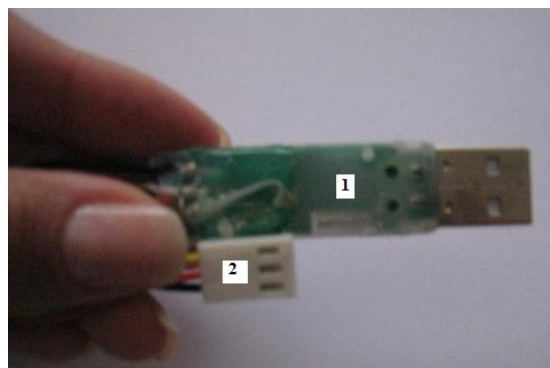
#### **4.1.2.2 Modem GSM**

Se utilizó el teléfono celular NOKIA 3220 considerando que tiene embebido un modem GSM para establecer la comunicación vía SMS con el microcontrolador ATmega164 además, buscando bajar costos en la implementación del proyecto.

Como ya se menciona el capítulo anterior se utilizará un cable DKU-5, como muestra la figura 4.4, para abrir los puertos del celular mediante un programa llamado Nokia PC Suite.

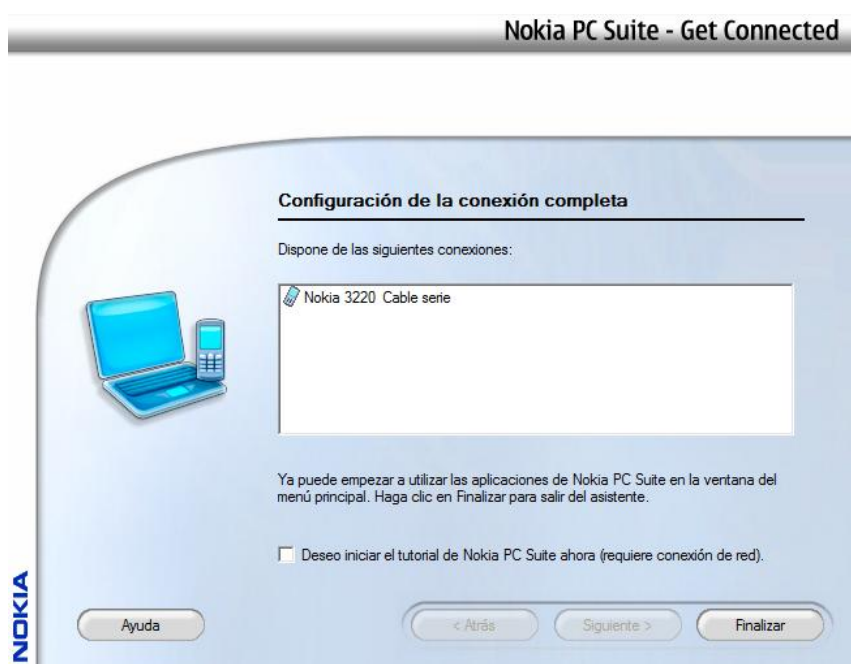
A continuación se realizar la apertura de los puertos del celular para la comunicación con el microcontrolador ATmega164.

Se conecta el celular a la PC mediante el USB marcado con el número uno “1” en la figura 4.2 del cable DKU-5.



**Figura 4. 4. Cable DKU-5**

En el programa PC Suite selecciona la opción “conexiones” para que reconozca automáticamente el teléfono celular como muestra la figura 4.5, una vez reconocido el dispositivo, mediante un hyperterminal utilizando un comando AT y recibiendo un OK, se comprueba que los puertos del celular se encuentran abiertos para transmitir y recibir información.



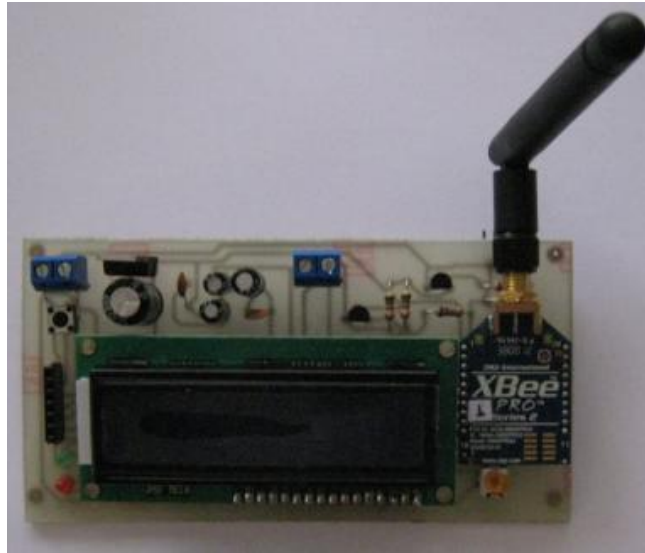
**Figura 4. 5. Pantalla de la configuración de la conexión completa**

En la tabla 4.2 se detalla el resultado de las pruebas del modem GSM.

**Tabla 4. 2 Resultados de funcionamiento del modem GSM.**

<b>ACTIVIDAD</b>	<b>SI</b>	<b>NO</b>
Teléfono en buen estado	X	
Batería cargada	X	
Cable DKU-5	X	
Reconocimiento del Nokia 3220 mediante PCSuite	X	
Verificación de puertos abiertos mediante hyperterminal	X	

Se diseñaron dos placas, una que contiene el coordinador Xbee con su respectivo microcontrolador ATmega 16 y un LCD que muestra un registro de funcionamiento como muestra la figura 4.6, mientras que la segunda placa contiene el Router Xbee, como muestra la figura 4.9.



**Figura 4. 6. Coordinador Xbee**

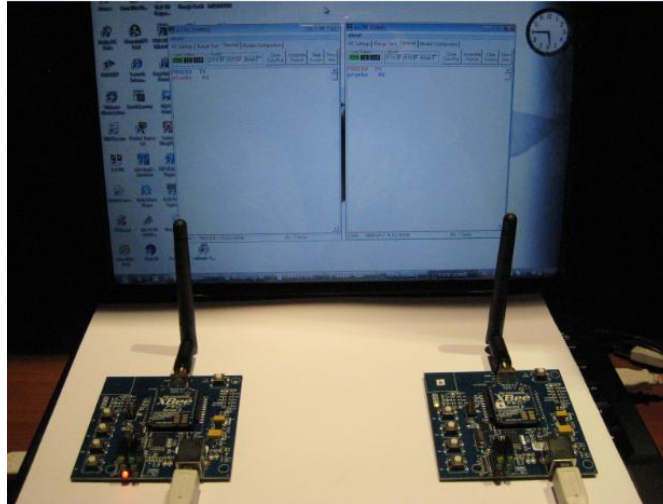
Para la placa que contiene el coordinador Xbee que se muestra en la figura 4.6 y el celular, se lo realiza mediante el conector dos “2” el cual es mostrado en la figura 4.4.



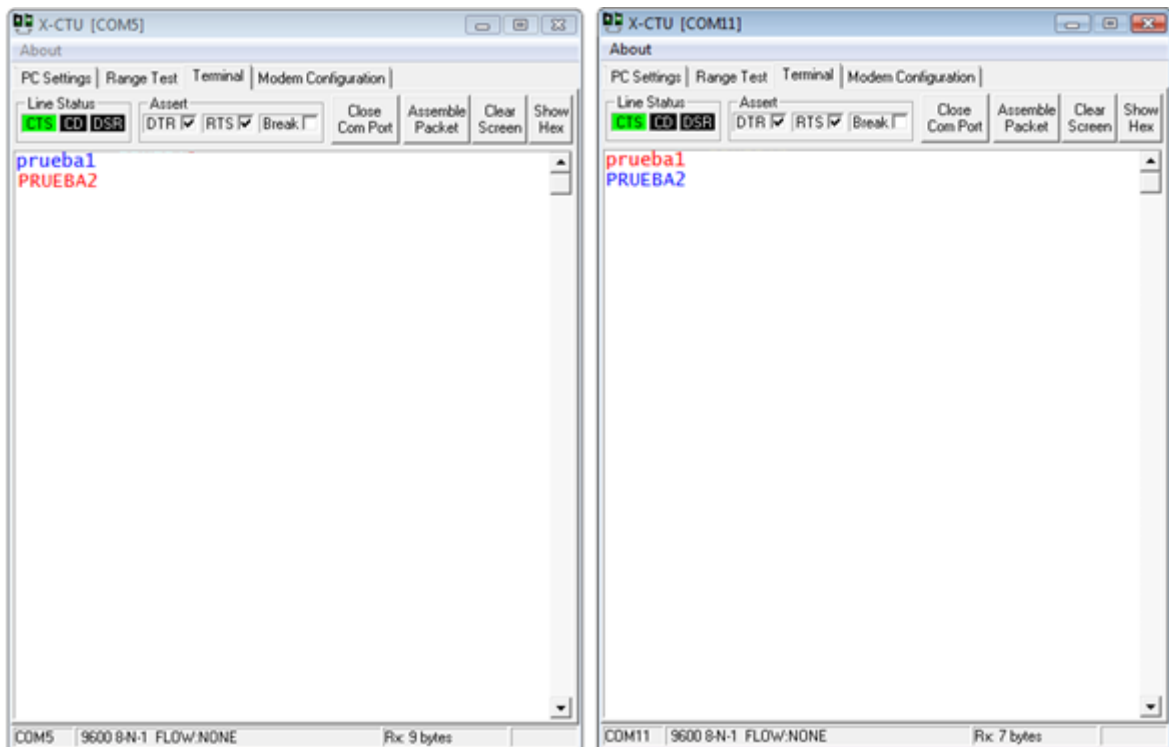
**Figura 4. 7. Conexión del Nokia 3220 a la placa coordinador Xbee**

### 4.1.2.3 Descripción y prueba bloque 3

En la figura 4.8 se muestra la comunicación inalámbrica establecida entre las tarjetas Xbee.



Sub-figura 4.8. 1. Comunicación inalámbrica con tarjetas Xbee.



Sub-figura 4.8. 2. Verificación de envío y recepción de la comunicación inalámbrica.

Figura 4. 8. Comunicación Inalámbrica Xbee

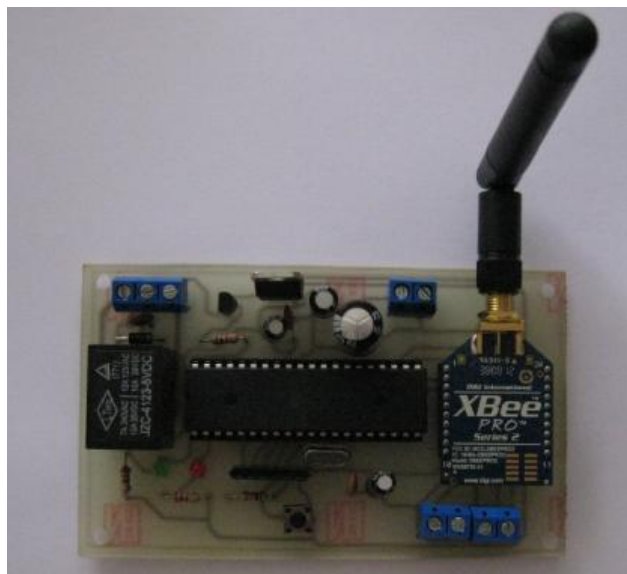
En la tabla 4.3 se detalla el resultado de las pruebas de la comunicación inalámbrica.

**Tabla 4. 3 Resultados de funcionamiento del la comunicación inalámbrica.**

<b>ACTIVIDAD</b>	<b>SI</b>	<b>NO</b>
PC instalada X-CTU.	X	
Zócalos Xbee para programación.	X	
Tarjetas Xbee Pro coordinador.	X	
Tarjetas Xbee Pro Router.	X	
Correcta transmisión mediante modo terminal.	X	
Correcta recepción mediante modo terminal.	X	

#### 4.1.2.4 Descripción y prueba bloque 4

Este bloque contiene el Router Xbee como se puede observar en la figura 4.9 con su respectivo microcontrolador ATmega16, el cual atiende las peticiones del coordinador según corresponda, activando o desactivando el actuador vía SMS o envía la acción del sensor mediante un mensaje de respuesta.

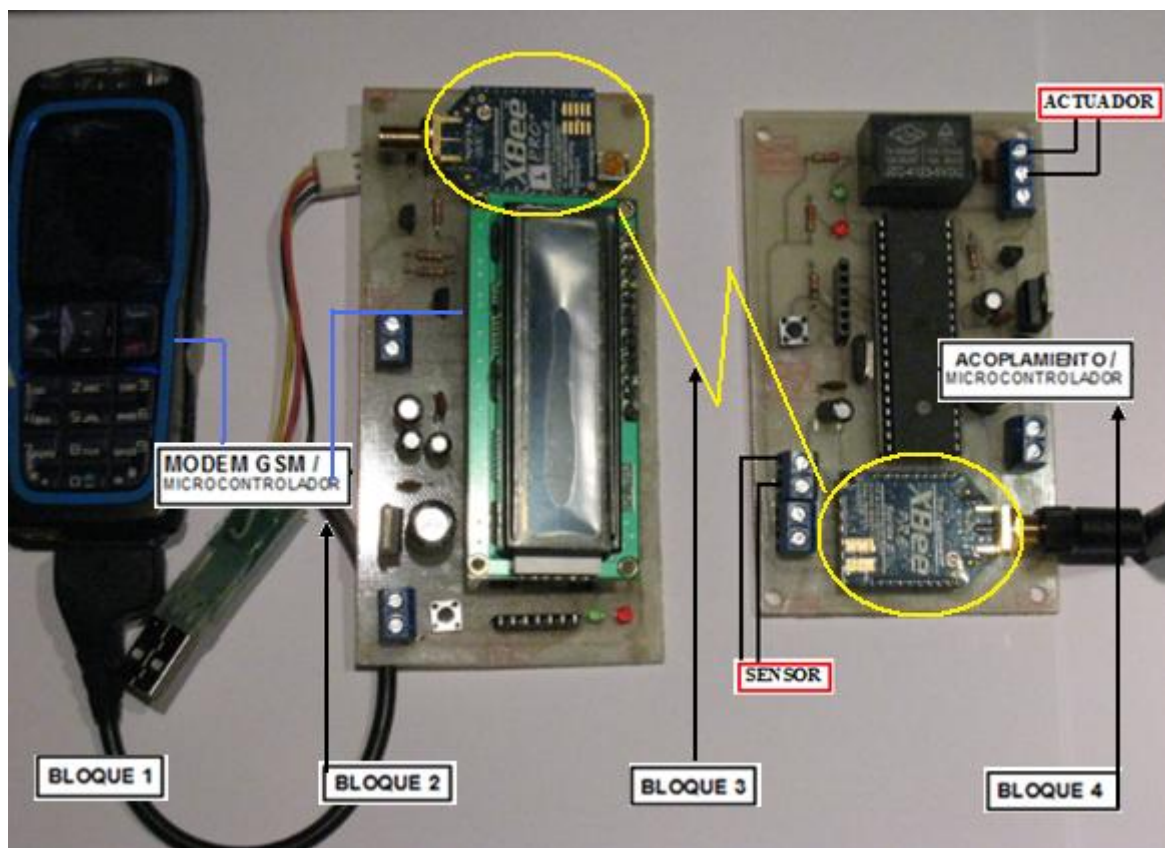


**Figura 4. 9. Router Xbee**

En la figura 4.11 se puede observar el diseño funcional por bloques del prototipo implementado.

Este es el prototipo, se encarga ya en conjunto y mediante el dispositivo móvil, del control de la aplicación domótica al enviar un SMS para activar o desactivar un actuador, que en este caso es una luminaria, indicando en el LCD la acción que se realizó.

Mientras que al activar o desactivar el sensor recibe un mensaje en el buzón de entrada del dispositivo móvil mostrando en el LCD el estado del mismo.



**Figura 4. 10. Diseño funcional por bloques del prototipo implementado**

Una vez realizadas las pruebas del prototipo se obtuvo que el esquema de diseño para la implementación de las placas es bastante robusto, el prototipo trabaja a 15 m de distancia en interiores dependiendo de la infraestructura de la edificación, en este caso es un ambiente con pisos de cemento y parquet, paredes de cemento, ventanas en cada habitación, y 50 m de distancia en exteriores con línea de vista.

En la tabla 4.3 se muestra un resumen detallado del funcionamiento del prototipo final.

**Tabla 4. 4 Resultados de funcionamiento del prototipo final.**

<b>INTERCONEXIÓN DEL PROTOTIPO</b>		
<b>ACTIVIDAD</b>	<b>SI</b>	<b>NO</b>
Dispositivo móvil con aplicación domotica	X	
Conexión entre modem GSM y placa con coordinador Xbee	X	
Comunicación inalámbrica	X	
<b>ACTIVACIÓN Y DESACTIVACIÓN DE LUMINARIA</b>		
<b>ACTIVIDAD</b>	<b>SI</b>	<b>NO</b>
Envío de mensaje ON	X	
Activación de la luminaria	X	
Envío de mensaje OFF	X	
Desactivación de luminaria	X	
<b>ACTIVACIÓN Y DESACTIVACIÓN DEL SENSOR</b>		
Señal de presencia en sensor	X	
Llega mensaje de activación del sensor al buzón de entrada	X	

## **CAPITULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1 CONCLUSIONES**

Se realizó el diseño e implementación de una WSN para controlar una aplicación domótica llegando a las siguientes conclusiones:

- Del estudio realizado se determinó que Zigbee es una tecnología competitiva ofreciendo fiabilidad de datos a bajos costos y presentando competencia con WiFi en la banda de 2,4 GHz y con otras aplicaciones de telecomunicaciones en las bandas ISM más bajas.
- De la investigación realizada se determina que la principal característica del estándar de IEEE 802.15.4 es su bajo consumo de energía garantizando baterías de larga duración por lo que es la mejor opción comparado con Bluetooth para aplicaciones domóticas.
- Se determinó que una WSN utiliza tecnología de radiofrecuencia que permite mayor portabilidad a los usuarios al minimizar las conexiones cableadas.
- Se determinó los alcances y limitaciones de la aplicación implementada llegando a los siguientes rangos de funcionamiento, la máxima distancia de trabajo a la que puede funcionar el prototipo en interiores es de 15 metros que depende de la estructura de la edificación, para este caso se realizaron las pruebas en una casa con paredes de cemento y 50 metros en exteriores con línea de vista.
- Se determinó que una de las limitaciones del proyecto es disponer de un equipo que contenga embebido el modem GSM fundamentalmente por el costo que representa.



- Se determino que Zigbee está más orientado hacia el control remoto y automatización, mientras que Bluetooth está orientado a la movilidad del usuario y la eliminación de cables entre dispositivos a corta distancia.
- El funcionamiento del sistema es satisfactorio y se cumplieron los objetivos del proyecto. Por lo tanto, se puede concluir, que SMS es una buena alternativa para este tipo de aplicaciones por su capacidad de funcionamiento en equipos móviles, de reducido tamaño y bajo consumo de potencia.

## 5.2 RECOMEDACIONES

- Se recomienda el uso del lenguaje Java, para la creación de aplicaciones para móviles, por la versatilidad y la gran cantidad de equipos que lo soportan.
- Se puede realizar la prueba de comunicación (interiores) en edificaciones de estructura diferente a la que presenta el proyecto. Las alternativas pueden ser: vidrios, madera, mixtas con la finalidad de comparar los resultados obtenidos con los presentados en el proyecto.
- Se recomienda realizar la prueba de radiofrecuencia, exponiendo a los módulos Xbee a interferencia de otros dispositivos que trabajen en la banda de 2,4 GHz como microondas, Router *Wireless*, equipos Bluetooth para analizar la comunicación inalámbrica y verificar que realice la elección del mejor canal para el funcionamiento de la red ZigBee.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Centric, Domotica, [http://www.centricvilanova.com/gen/domotica\\_seguretatar\\_llar.htm](http://www.centricvilanova.com/gen/domotica_seguretatar_llar.htm)
- [2] Zigbee org, <http://www.zigbee.org/>
- [3] Zigbee, <http://www.worldlingo.com/ma/enwiki/es/ZigBee>
- [4] IEEE802.15.4, [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lem/archundia\\_p\\_fm/capitulo4.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/archundia_p_fm/capitulo4.pdf)
- [5] Redes de Sensores y Actuadores Inalámbricas, <http://www.cea-ifac.es/actividades/jornadas/XXIX/pdf/256.pdf>
- [6] Francisco Blanes, José Simo, Alfons Crespo, Redes de Sensores y Actuadores Inalámbricas, <http://www.cea-ifac.es/actividades/jornadas/XXIX/pdf/256.pdf>
- [7] Guía de usuario Xbee, <http://www.scribd.com/doc/34935976/XBee-Guia-Usuario>
- [8] Características generales del Atmega 16, [http://www.sc.ehu.es/sbweb/webcentro/automatica/web\\_avr/archivos/Otros%20AVRs/ATmega/ATmega16.htm](http://www.sc.ehu.es/sbweb/webcentro/automatica/web_avr/archivos/Otros%20AVRs/ATmega/ATmega16.htm).
- [9] El sistema GSM, <http://www.edicionsupc.es/ftppublic/pdfmostra/TL02602M.pdf>
- [10] Sergio gálvez rojas lucas ortega d'az 2003 java a tope: j2me (java 2micro edition). Edición electrónica
- [11] netbeans.org, <http://VisualMobileDesignerPalatteReference>
- [12] J2ME, <http://pegasus.javeriana.edu.co/~mad/J2ME.pdf>
- [13] netbeans.org, <http://VisualMobileDesignerPalatteReference>

## **ANEXO 1**

### **PROGRAMA DE LOS MICRONTROLADORES ATmega16**

## Programa de microcontrolador 1 (Coordinador)

```
*****
**
**      DEFINICION MICROCONTROLADOR, CRISTAL Y VELOCIDAD SERIAL
**
*****
**
$regfile = "m164pdef.dat"          ' DEFINIMOS MICROCONTROLADOR
$crystal = 4000000                 ' VELOCIDAD DEL CRISTAL

*****
**
**      INCLUSION DE ARCHIVOS EXTERNOS - SUBFUNCIONES          *
**
*****
**
$include "SMS_INIT.BAS"

$baud = 2400
$baud1 = 9600
'Config Com2 = Dummy, Synchron = 0, Parity = None, stopbits command = 1, databits =
8, Clockpol = 0

Config Com1 = 2400 , Synchron = 0 , Parity = None , Stopbits = 1 , Databits = 8 ,
Clockpol = 0
Config Com2 = 9600 , Synchron = 0 , Parity = None , Stopbits = 1 , Databits = 8 ,
Clockpol = 0

Open "COM2:" For Binary As #1

*****
**
**      DEFINICION ENTRADAS Y SALIDAS          *
**
*****
**
Ddrc = 0                          ' PUERTO C COMO ENTRADAS
Portc = 255

Ddrb.0 = 1 : Portb.0 = 0           ' PORTB COMO SALIDAS
Ddrb.1 = 1 : Portb.1 = 0           ' PORTB COMO SALIDAS

Ddrb.4 = 1 : Portb.4 = 0           ' PORTB COMO SALIDAS
Ddrb.5 = 1 : Portb.5 = 0           ' PORTB COMO SALIDAS
Ddrb.6 = 1 : Portb.6 = 0
Ddrb.7 = 1 : Portb.7 = 0
```



```

Do
  Toggle Led1
  A = Ischarwaiting()
  If A = 1 Then
    Recibirmensaje
    Validarmensaje Mensaje
  End If

  B2 = Inkey(#1)
  If B2 > 0 Then
    Select Case B2
      Case 67:
        Print #1 , "APM MICRO " ; Str(b2)
        Mensa = "SENSOR ACTIVADO"
        Enviarmensaje Mensa , Numero
        Mensa = ""
    End Select
  End If

  If Sensor = 0 Then
    Set Led2

    For I = 1 To 5
      Print #1 , "SENSOR"
      Waitms 50
    Next

    Mensa = "SENSOR TEST"
    Enviarmensaje Mensa , Numero
    Reset Led2
  End If
  Waitms 25
Loop

End

Close #1

$include "SMS_END.BAS"

```

## Programa de microcontrolador 2 (Router)

```
*****
**
**      DEFINICION MICROCONTROLADOR, CRISTAL Y VELOCIDAD SERIAL
**
*****
**
$regfile = "m164pdef.dat"          ' DEFINIMOS MICROCONTROLADOR
$crystal = 4000000                 ' VELOCIDAD DEL CRISTAL

*****
**
**      INCLUSION DE ARCHIVOS EXTERNOS - SUBFUNCIONES          *
*****
**
$baud = 9600

Ddrc = 0                          ' PUERTO C COMO ENTRADAS
Portc = 255

Ddrb.0 = 1 : Portb.0 = 0           ' PORTB COMO SALIDAS
Ddrb.1 = 1 : Portb.1 = 0           ' PORTB COMO SALIDAS
Ddrb.2 = 1 : Portb.2 = 0           ' PORTB COMO SALIDAS

Ddrb.4 = 1 : Portb.4 = 0           ' PORTB COMO SALIDAS
Ddrb.5 = 1 : Portb.5 = 0           ' PORTB COMO SALIDAS
Ddrb.6 = 1 : Portb.6 = 0
Ddrb.7 = 1 : Portb.7 = 0

Sensor Alias Pinc.0
Sensor2 Alias Pinc.1

Led1 Alias Portb.0
Led2 Alias Portb.1
Rele Alias Portb.2

*****
**
**      DEFINICION VARIABLES Y ARREGLOS          *
*****
**
Dim I As Byte                      ' CONTADOR
Dim A As Byte                      ' RECIBIR EL PRIMER CARACTER DE
RESPUESTA DEL CELULAR
Dim B2 As Byte
Dim K As Byte
Dim X As Byte
```

Inicio:

```
Do
  Toggle Led1
  B2 = Inkey()
  If B2 > 0 Then
    Select Case B2
      Case 65:
        Set Led2
        Set Rele
      Case 66:
        Reset Led2
        Reset Rele
    End Select
  End If
  If Sensor = 0 Then
    Set Led2
    For I = 1 To 3
      Print "C"
      Waitms 50
    Next
    Reset Led2
  Do
    B2 = Inkey()
  Loop Until B2 = 0
  Waitms 500
End If
Waitms 25
Loop
```

## Declaración de subfunciones

```
*****
**
'*          DESARROLLO DE SUBFUNCIONES          *
*****
**
```

'++++SUBROUTINA DE CONFIGURACION INICIAL

Sub Config\_inicial()

```
  For I = 1 To 8
    Toggle Led1
    Waitms 50
  Next
```



Led1 = 0

Do

Print "AT" ; Chr(13);

' ENVIAMOS "AT" + ENTER

Getok Sret

Loop Until Sret = "OK"

Limpiarbuffer

For I = 1 To 8

Toggle Led1

Waitms 50

Next

Led1 = 0

Waitms 25

Do

Print "ATE0" ; Chr(13);

Getok Sret

Loop Until Sret = "OK"

Limpiarbuffer

For I = 1 To 8

Toggle Led1

Waitms 50

Next

Led1 = 0

Waitms 25

Do

Print "AT+CMGF=1" ; Chr(13);

Getok Sret

Loop Until Sret = "OK"

Limpiarbuffer

For I = 1 To 8

Toggle Led1

Waitms 50

Next

Led1 = 0

Waitms 25

Do

Print "AT+CNMI=1,2,0,0,0" ; Chr(13);

Getok Sret

Loop Until Sret = "OK"

Limpiarbuffer

For I = 1 To 8

Toggle Led1

Waitms 50

```

Next
Led1 = 0

Waitms 25
End Sub

' SUBROUTINA DE ESPERA RESPUESTA OK

' ESPERAMOS LA RESPUESTA DEL CELULAR -> "OK"

Sub Getok(s As String)
S = ""
Do
B = Inkey()
Select Case B
Case 0
Case 13
Case 10 : If S <> "" Then Exit Do
Case Else
S = S + Chr(b)
End Select
Loop
End Sub

' SUBROUTINA DE ESPERA RESPUESTA OK

' DEJAR LIMPIO LOS CANALES DE COMUNICACION ENTRE MICRO Y
CELULAR

Sub Limpiarbuffer()
Waitms 100
Do
B = Inkey()
Loop Until B = 0
End Sub
' AT+CMGS = "NUMERO" -> RESP: ">"
' ENVIAMOS MENSAJE + CTRL Z -> RESP: "OK"
Sub Enviarmensaje(s As String , N As String)
Do
Print "AT" ; Chr(13);
Getok Sret
Loop Until Sret = "OK"
Limpiarbuffer

Do
Print "AT+CMGS=" ; Chr(34) ; N ; Chr(34) ; Chr(13);
Sret = ""
Do

```

```

B = Inkey()
Select Case B

    Case 0
    Case 13
    Case 10 : If Sret <> "" Then Exit Do
    Case 62 : Goto Envio1
    Case Else
        Sret = Sret + Chr(b)
    End Select
Loop
Loop Until Sret = ">" Or Sret = " >"
Envio1:
Limpiarbuffer
Waitms 100
Limpiarbuffer
Print S ; Chr(26);
Do
    Getok Sret
Loop Until Sret = "OK"
Limpiarbuffer
Waitms 500
End Sub

```

```

Sub Recibirmensaje()

```

```

    Dim Flag_men As Bit
    Flag_men = 0
    Mensaje = ""
    Do
        B = Inkey()
        Select Case B
            Case 13:
            Case 10:
            Case 45:
                Exit Do
            Case Else
                Mensaje = Mensaje + Chr(b)
            End Select
        Loop
    Do
        B = Inkey()
    Loop Until B = 10
    Mensaje = ""
    Do
        B = Inkey()
        Select Case B
            Case 13:

```

```

Case 10:
  Exit Do
Case Else
  Mensaje = Mensaje + Chr(b)

  End Select
Loop
Do
  B = Inkey()
Loop Until B = 0
End Sub

Sub Validarmensaje(s As String)
  Select Case S
  Case "LUZON":
    Set Led2
    Print #1, "A"
    For I = 1 To 5
      Print #1, "A"
      Waitms 50
    Next

  Case "LUZOFF":
    Reset Led2
    Print #1, "B"

    For I = 1 To 5
      Print #1, "B"
      Waitms 50
    Next

  End Select
  S = ""
End Sub

'*****
**
'*          DECLARACIÓN DE SUBFUNCIONES          *
'*****
**
Dim B As Byte
Dim Sret As String * 20          ' GUARDA LA RESPUESTA DEL
CELULAR ("OK", "ERROR"=

```

```

Dim Mensa As String * 50          ' Guarda El Mensaje A Enviar(
"SENSOR VENTANA1 ACTIVADO)"
Dim Mensaje As String * 150      ' GUARDAMOS EL MENSAJE
RECIBIDO ("ALARMA ON")
Dim Numero As String * 13
Dim Numero2 As String * 13

Numero = "+59384404066"
Declare Sub Config_inicial()      ' SUBFUNCION PARA
CONFIGURACION INICIAL DEL CELULAR
Declare Sub Getok(s As String)   ' SUBFUNCION PARA OBTENER OK
Declare Sub Limpiarbuffer()      ' SUBFUNCION PARA LIMPIAR
BUFFER
Declare Sub Enviarmensaje(s As String , N As String) ' SUBFUNCION PARA
ENVIAR MENSAJE
Declare Sub Recibirmensaje()     ' SUBFUNCION PARA RECIBIR
MENSAJE
Declare Sub Validarmensaje(s As String) ' SUBFUNCION PARA VALIDAR
MENSAJE

```

## **ANEXO 2**

### **MANUAL DEL SOFTWARE X-CTU**

# X-CTU

## Configuration & Test Utility Software

---

### User's Guide

---

### Contents

---

<b>Introduction</b>	<b>2</b>
<b>PC Settings Tab</b>	<b>3</b>
COM port setup:	3
Host Setup:	4
User COM ports:	4
<b>Range Test Tab</b>	<b>4</b>
Packet Data and Size	4
RSSI:	6
API Function:	6
<b>The Terminal Tab</b>	<b>7</b>
The main terminal window	7
Assemble Packet	8
<b>Modem Configuration tab</b>	<b>9</b>
Reading the Radios firmware	9
Making changes to the radios firmware	9
Writing firmware to the radio	10
Downloading updated firmware files	11
Modem Profiles	12

---



Technical Support:

Online support: <http://www.digi.com/support/eservice/login.jsp>

Phone: (801) 765-9885

90001003\_A

2008.08.20

## Introduction

This User's Guide is intended to discuss the functions of Digi's X-CTU software utility. Each function will be discussed in detail allowing a better understanding of the program and how it can be used.

X-CTU is a Windows-based application provided by Digi. This program was designed to interact with the firmware files found on Digi's RF products and to provide a simple-to-use graphical user interface to them.

X-CTU is designed to function with all Windows-based computers running Microsoft Windows 98 SE and above. X-CTU can either be downloaded from Digi's Web site or an installation CD. When properly installed it can be launched by clicking on the icon on the PC desktop (see Figure 1) or selecting from the Start menu (see Figure 2).



Figure 1

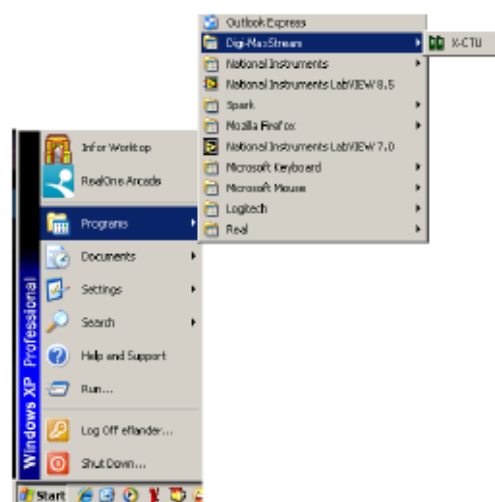


Figure 2

When launched, you will see four tabs across the top of the program (see Figure 3). Each of these tabs has a different function. The four tabs are:

**PC Settings:** Allows a customer to select the desired COM port and configure that port to fit the radios settings.

**Range Test:** Allows a customer to perform a range test between two radios.

**Terminal:** Allows access to the computers COM port with a terminal emulation program. This tab also allows the ability to access the radios' firmware using AT commands (for a complete listing of the radios' AT commands, please see the product manuals available online).

**Modem Configuration:** Allows the ability to program the radios' firmware settings via a graphical user interface. This tab also allows customers the ability to change firmware versions.





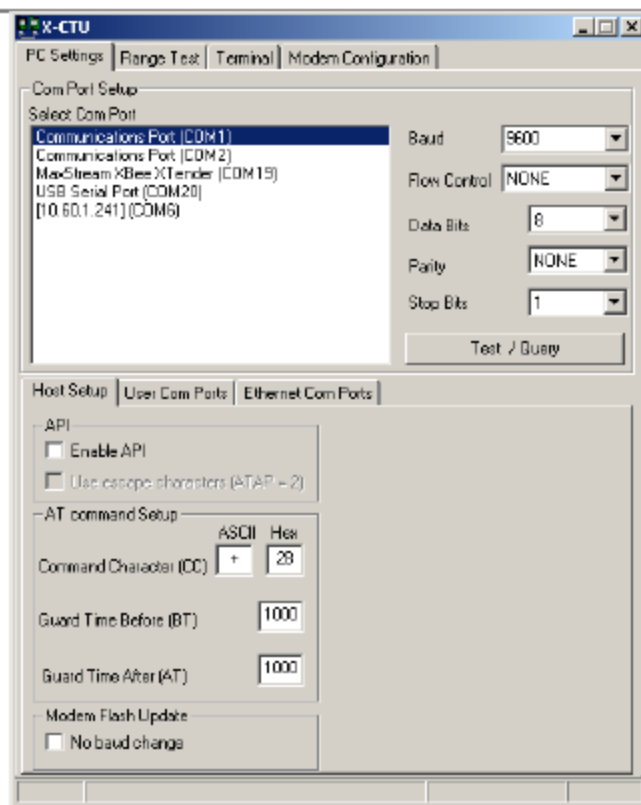


Figure 3

## PC Settings Tab

When the program is launched, the default tab selected is the "PC Settings" tab. The PC Settings tab is broken down into three basic areas: The COM port setup, the Host Setup, and the User Com ports.

### COM port setup:

The PC settings tab allows the user to select a COM port and configure the selected COM port settings when accessing the port. Some of these settings include:

Baud Rate:	Both standard and non-standard
Flow Control:	Hardware, Software (Xon/Xoff), None
Data bits:	4, 5, 6, 7, and 8 data bits
Parity:	None, Odd, Even, Mark and Space
Stop bit:	1, 1.5, and 2

To change any of the above settings, select the pull down menu on the left of the value and select the desired setting. To enter a non-standard baud rate, type the baud rate into the baud rate box to the left.

The **Test / Query** button is used to test the selected COM port and PC settings. If the settings and COM port are correct, you will receive a response similar to the one depicted in Figure 4 below.

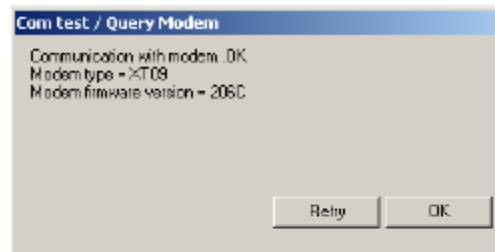


Figure 4

---

**Host Setup:**

The Host Setup tab allows the user to configure how the X-CTU program is to interface with a radio's firmware. This includes determining whether API or AT command mode will be used to access the module's firmware as well as the proper command mode character and sequence.

By default, the Host Settings are as follows:

API mode:	not enabled (Not checked)
Command mode Character:	+ (ASCII) 2B (Hex).
Before Guard Time:	1000 (1 Sec)
After Guard Time:	1000 (1 Sec)

This is the default value of our radios. If this is not the value of the AT, BT, or GT commands of the connected radio, enter the respective value here.

---

**User COM ports:**

The user COM port option allows the user to "Add" or "Delete" a user-created COM port. This is only for temporary use. Once the program has closed, the user-created COM port will disappear and is no longer accessible to the program.

## Range Test Tab

The range test tab is designed to verify the range of the radio link by sending a user-specified data packet and verifying the response packet is the same, within the time specified. For performing a standard range test, please follow the steps found in most Quick Start or Getting Started Guides that ship with the product.

### Packet Data and Size

By default, the size of the data packet sent is 32 bytes. This data packet specified can be adjusted in either size or the text sent.

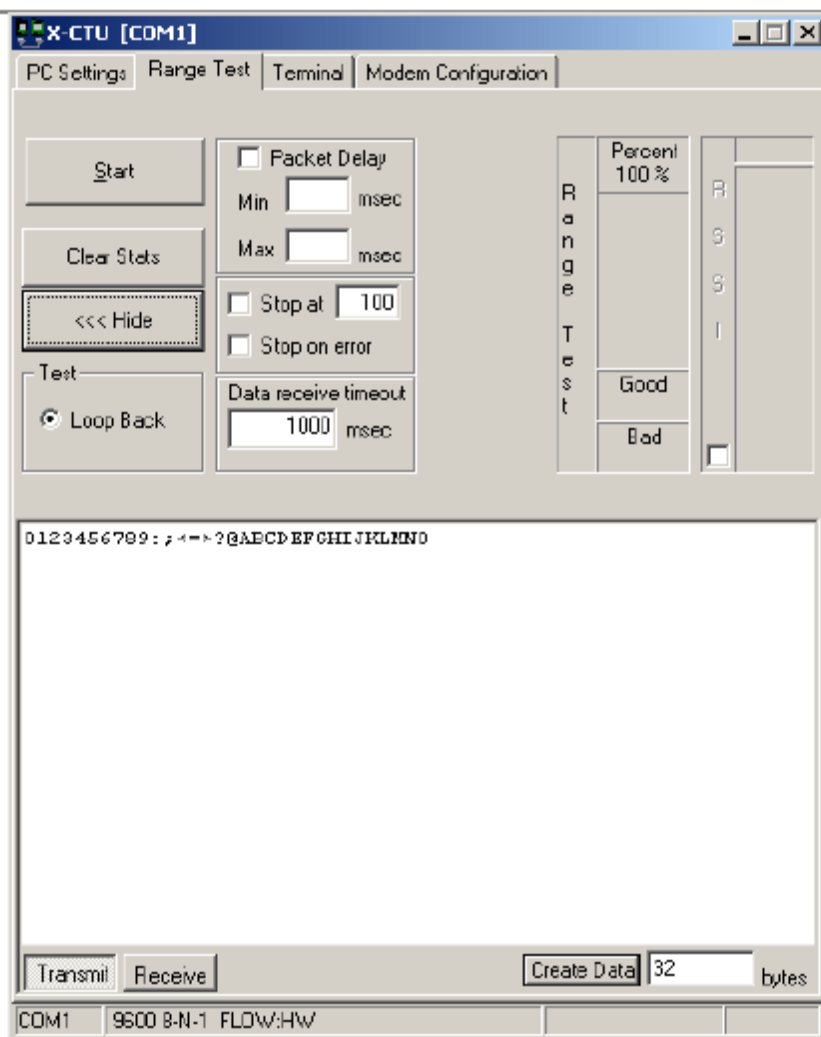


Figure 5

To modify the size of the packet sent, change the value next to the "Create Data" box and click on the "Create Data" button (see Figure 5). If you want to change the data sent, delete the text in the transmit window and place in your desired text.

By modifying the text, data packet size, packet delay and the data receive timeout; the user is able to simulate a wide range of scenarios.

---

**RSSI:**

The RSSI option of the X-CTU allows the user to see the RSSI (Received Signal Strength Indicator) of a received packet when performing a range test.

---

**API Function:**

The X-CTU also allows the user to test the API function of a radio during a range test.

To perform a range test with the API function of the radio, follow the steps outlined below:

- 1: Configure the Base with API enabled and a unique 16 bit or 64 bit source address.
- 2: Configure the remote radio with a unique source address and set the Destination address to equal the Base radio's source address.
- 3: Enable the API option of the X-CTU on the PC Settings tab and connect the base radio to the PC (See Figure 3).
- 4: Connect the red loopback adapter to the remote radio and place them a distance apart.
- 5: Enter either the 16 bit or 64 bit destination address of the remote radio into the Destination Address box on the Range Test tab (See figure 6).
- 6: Create a data packet of your choosing by typing in the data in the Transmit box
- 7: To start a Range test, click on Start.

You will notice the TX failures, Purge, CCA, and ACK messages will increment accordingly while the range test is performed.

To stop a range test, click on the Stop button.

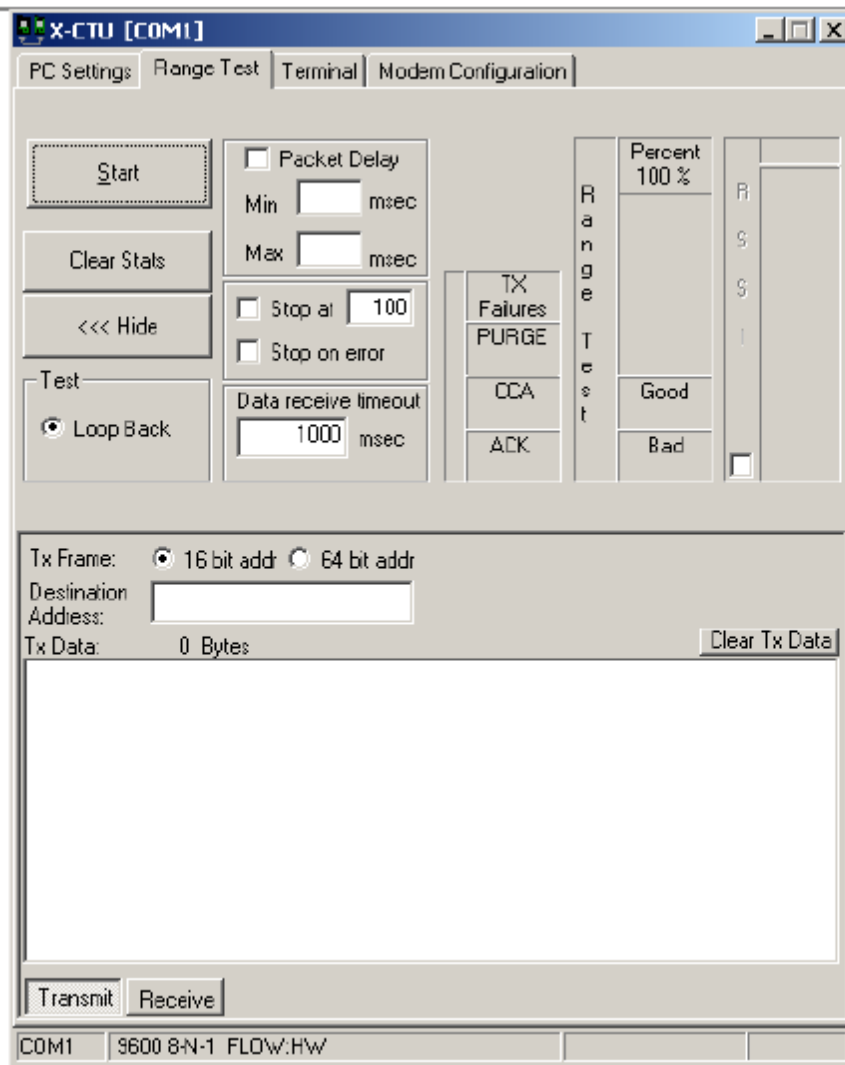


Figure 6

## The Terminal Tab

---

The Terminal tab has three basic functions:

- Terminal emulator
- Ability to send and receive predefined data packets (Assemble packet)
- Ability to send and receive data in Hex and ASCII formats (Show/Hide hex)

### The main terminal window

---

The main white portion of this tab is where most of the communications information will occur while using X-CTU as a terminal emulator. The text in **blue** is what has been typed in and directed out to the radio's serial port while the **red** text is the incoming data from the radio's serial port (see Figure 7).

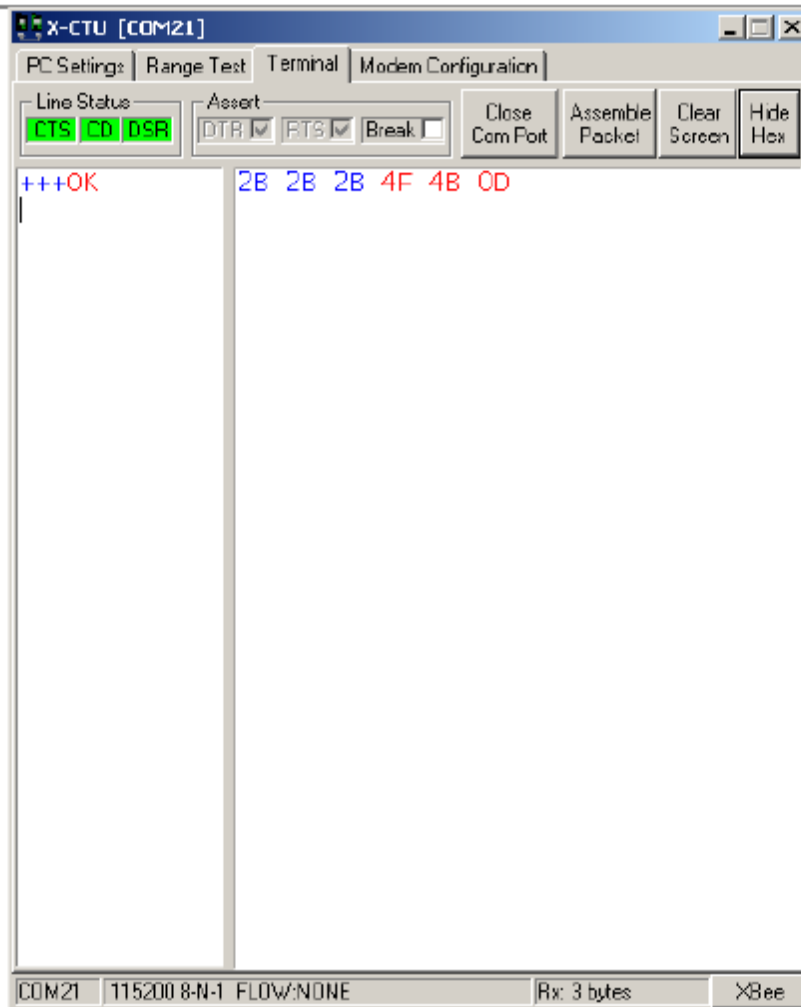


Figure 7





### Assemble Packet

---

The Assemble Packet option on the Terminal tab is designed to allow the user to assemble a data packet in either ASCII or Hex characters. This is accomplished by selecting the Assemble packet window and choosing either ASCII (default) or Hex. Once selected, the data packet is assembled by typing in the desired characters as depicted in Figure 8.

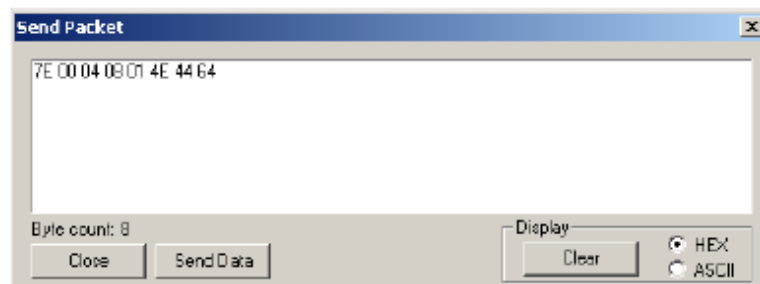


Figure 8

The **Line Status** indicators depicted in Figure 5 shows the status of the RS-232 hardware flow control lines. Green indicates the line is asserted while black indicates de-asserted.

The **Break** option is for engaging the serial line break. This can be accomplished by checking or asserting the Break option. Asserting the Break will place the DI line high and prevent data from being sent to the radio.

## Modem Configuration tab

---

The Modem configuration tab has four basic functions:

- 1: Provide a Graphical User Interface with a radio's firmware
- 2: Read and Write firmware to the radio's microcontroller
- 3: Download updated firmware files from either the web or from a compressed file
- 4: Saving or loading a modem profile

---

### Reading a radio's firmware

To read a radio's firmware, follow the steps outlined below:

- 1: Connect the radio module to the interface board and connect this assembly or a packaged radio (PKG) to the PC's corresponding port (IE: USB, RS232, Ethernet etc.).
- 2: Set the PC Settings tab (see Figure 3) to the radio's default settings.
- 3: On the Modem Configuration tab, select "Read" from the Modem Parameters and Firmware section (see Figure 9).

---

### Making changes to a radio's firmware

Once the radio's firmware has been read, the configuration settings are displayed in three colors (see Figure 10):

- Black - not settable or read-only
- Green - Default value
- Blue - User-specified

To modify any of the user-settable parameters, click on the associated command and type in the new value for that parameter. For ease of understanding a specific command, once the command is selected, a quick description along with its limits is provided at the bottom of the screen. Once all of the new values have been entered, the new values are ready to be saved to the radio's non-volatile memory.

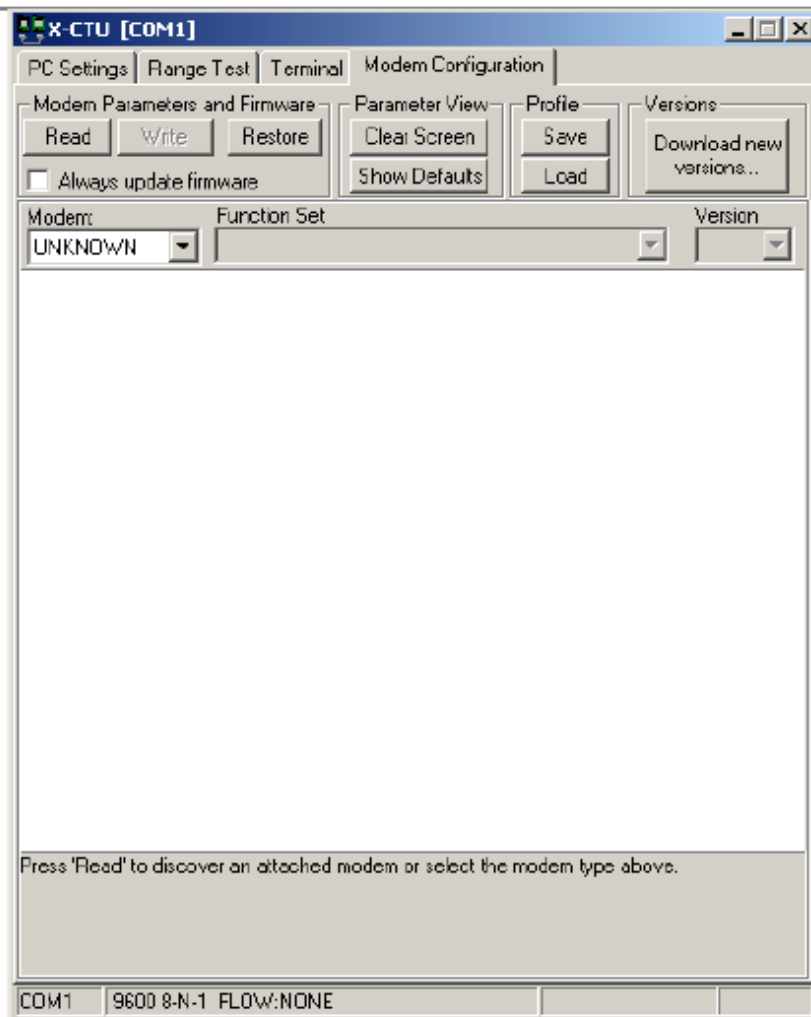


Figure 9

#### Writing firmware to the Radio

To write the parameter changes to the radio's non-volatile memory, click on the Write button located in the Modem Parameters and Firmware section (see Figure 10)

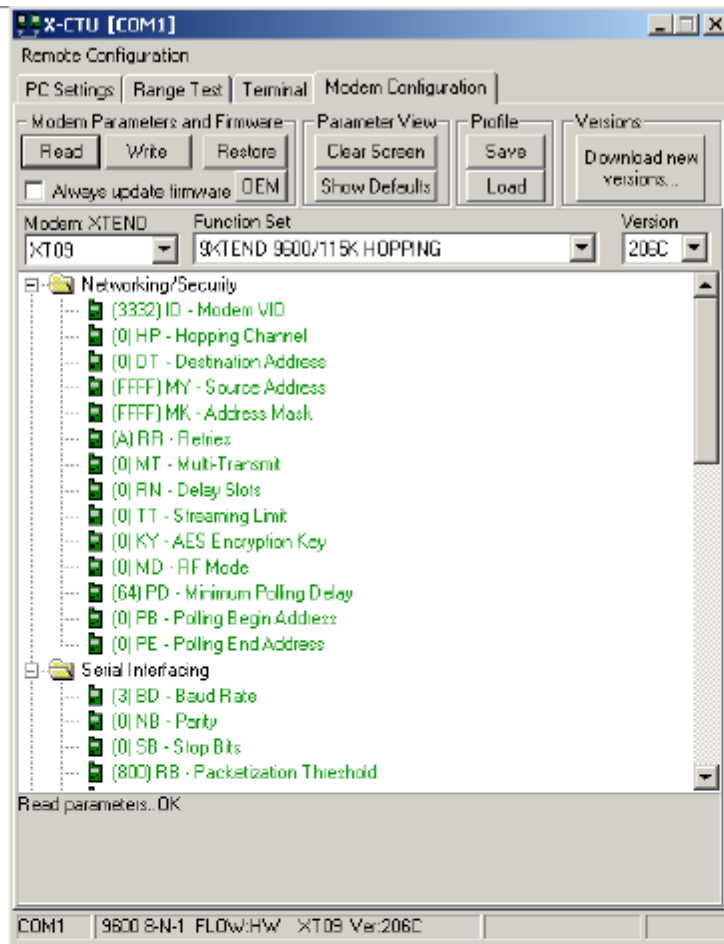


Figure 10

#### Downloading Updated Firmware Files

Another function of the Modem Configuration tab is allowing the user to download updated firmware files from either the web or install them from a disk or CD. This is accomplished by following the steps below:

- 1: Click on the Download New Versions... option under the Version section
- 2a: Click on Web for downloading new firmware files from the web
- 2b: Click on the File when installing compressed firmware files from a CD or saved file (see Figures 11 and 12)
  - 2bi: Browse to the location the file is saved at and click on Open (see Figure 13)
- 3: Click on OK and Done when prompted

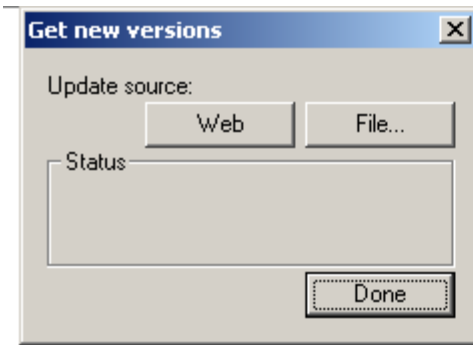


Figure 11

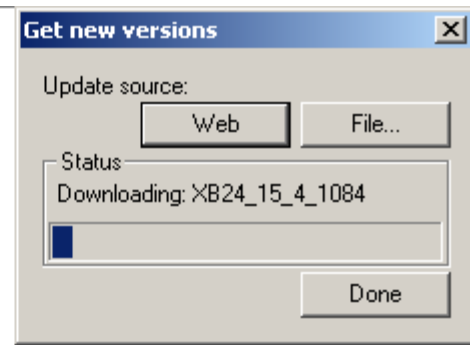


Figure 12

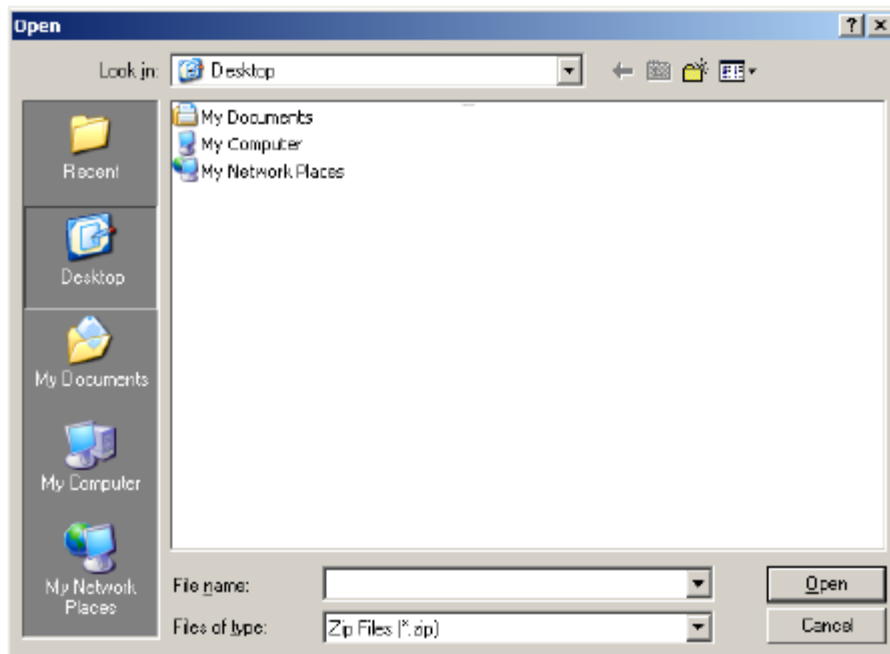


Figure 13

### Modem Profiles

The X-CTU has the ability to save and write saved modem profiles or configuration to the radio. This function is useful in a production environment when the same parameters need to be set on multiple radios.

How to save a profile:

- 1: Set the desired settings within the radio's firmware as described in the Making changes to the radios firmware section
- 2: Click Save in the Profile section
- 3: Type in the desired name of this profile in the File Name box (see Figure 14)
- 4: Browse to the location where you wish to save your profile
- 5: Click Save

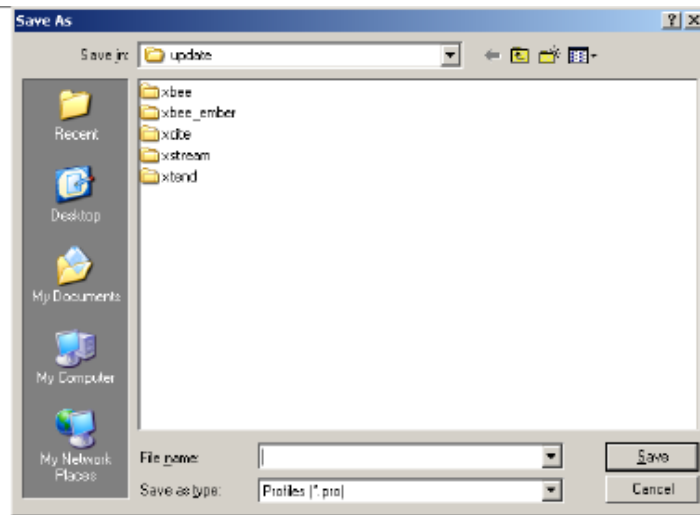


Figure 14

How to load a saved profile:

- 1: Click on Load from the profile section
- 2: Browse to the location of the file and click on the desired file (see Figure 15)
- 3: Click Open

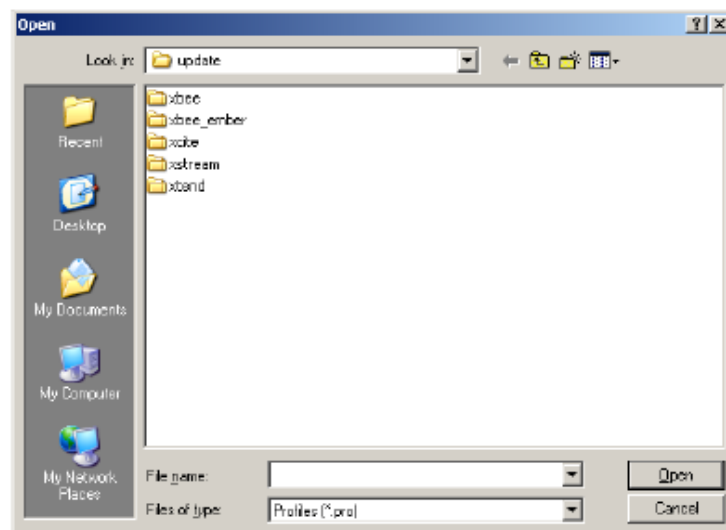


Figure 15

To save the loaded profile to the radio once you have loaded the file, follow the steps outlined in the [Writing firmware to the radio](#) section above.

To find out how to load the saved profiles in a production environment from a DOS prompt, please follow the steps outlined in Digi's online Knowledgebase at <http://www.maxstream.net/support/knowledgebase/article.php?kb=126>

---

### Remote Modem Management

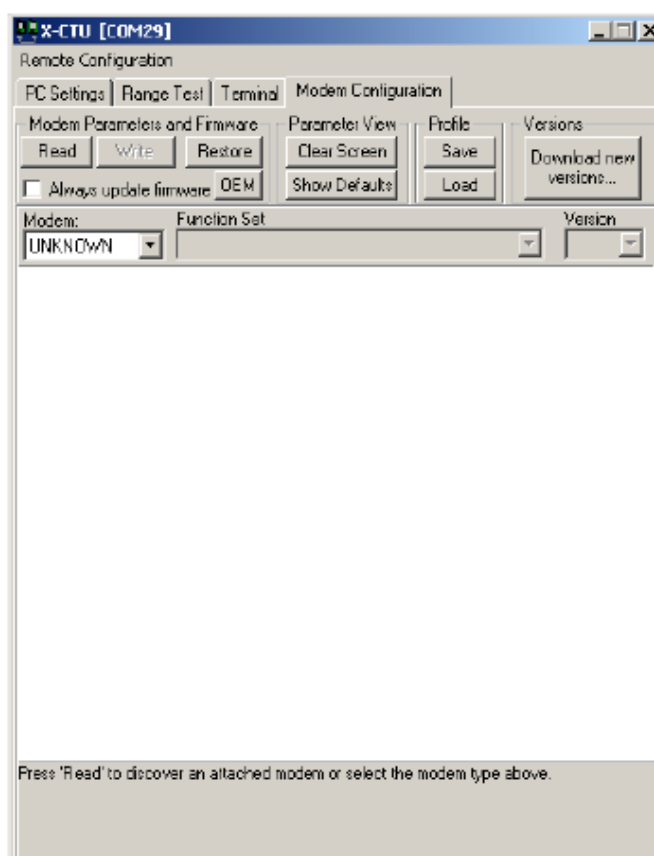
---

XBee 802.15.4 modules with firmware version 1xCx and above, XBee ZNet 2.5 modules, and XBee ZB modules offer the ability to be configured with over the air commands. With the addition of this new feature, the user is able to configure remote radio parameters with X-CTU or API packets. To use the remote configuration tool, the following is required:

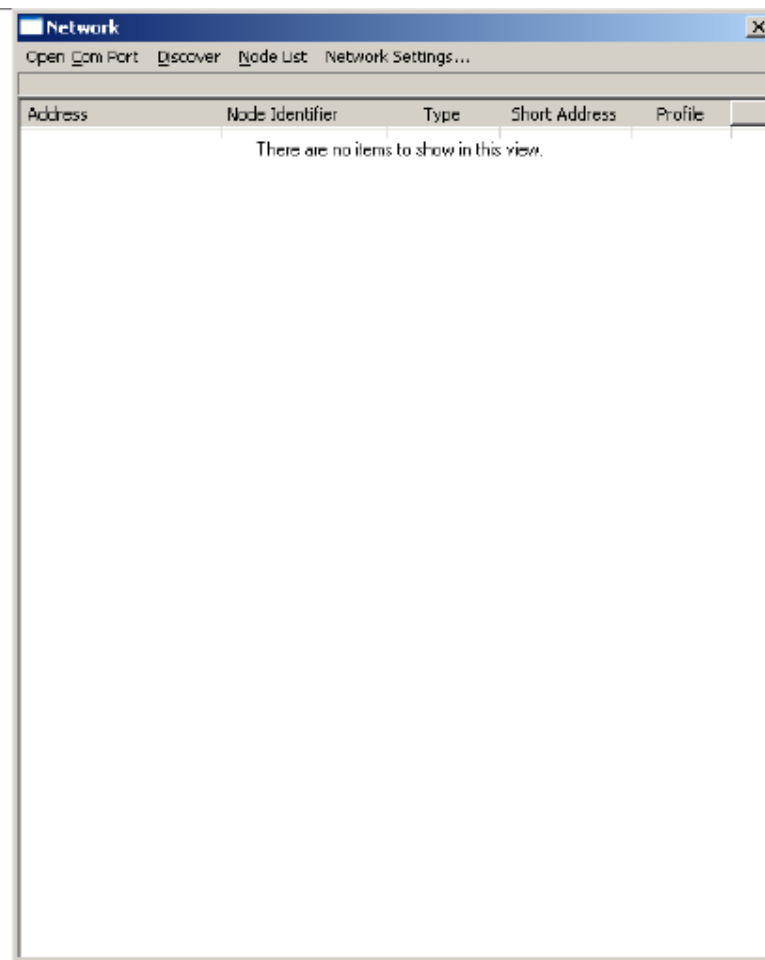
- The radio connected to the PC must be in API mode
- The remote radio must be associated or within range of the base radio

To access remote radios through X-CTU's Modem Configuration tab, perform the steps below:

- Enable API on the PC Settings tab
- Verify the COM port selection and settings
- On the Modem Configuration tab, select the Remote Configuration option on the top left corner of the program



- Select Open Com port
- Select Discover



- Select the desired modem from the discovered node list
- On the Modem configuration tab, select Read

The remote radio's configuration is now displayed on the Modem Configuration tab. At this point, the same options exist with respect to Read and Write parameter changes. Please note that the ability to change firmware versions is still limited to the radio's UART.

To clear the discovered node list, click on Node List and Clear.

The Node List option provides several additional options, including:

- Ability to print the discovered list
- Ability to remove a specific node from a list
- Ability to add additional nodes that have not been discovered
- Save the Node List
- Load a saved Node List
- Select/filter All, Routers, or End nodes

For specific questions related to the X-CTU configuration and test utility software, please contact our Support department, Mon - Fri, 8am - 5pm U.S. Mountain Time:



US and Canada Toll free:  
(866)765-9885

Local or International calls:  
(801) 765-9885

Online support: <http://www.digi.com/support/eservice/login.jsp>

## **ANEXO 3**

### **MANUAL DE LOS MÓDULOS XBEE/XBEE PRO**



**MaxStream**<sup>®</sup>

355 South 520 West, Suite 180

Lindon, UT 84042

Phone: (801) 765-9885

Fax: (801) 765-9895

rf-xperts@maxstream.net

www.MaxStream.net (live chat support)

M100344

2007.01.04

**© 2007 MaxStream, Inc. All rights reserved**

No part of the contents of this manual may be transmitted or reproduced in any form or by any means without the written permission of MaxStream, Inc.

ZigBee<sup>™</sup> is a registered trademark of the ZigBee Alliance.

XBee<sup>™</sup> and XBee-PRO<sup>™</sup> are trademarks of MaxStream, Inc.

**Technical Support:**

Phone: (801) 765-9885

Live Chat: [www.maxstream.net](http://www.maxstream.net)E-mail: [rf-xperts@maxstream.net](mailto:rf-xperts@maxstream.net)

# Contents

---

<b>1. XBee/XBee-PRO OEM RF Modules</b>	<b>4</b>	<b>4. RF Module Configuration</b>	<b>24</b>
1.1. Key Features	4	4.1. AT Command Programming Examples	24
1.1.1. Worldwide Acceptance	4	4.2. Command Reference Tables	25
1.2. Specifications	5	4.3. API Operation	29
1.3. Mechanical Drawings	6	4.3.1. API Frame Specifications	29
1.4. Mounting Considerations	6	4.3.2. API Types	30
1.5. Pin Signals	7	<b>Appendix A: Agency Certifications</b>	<b>34</b>
1.6. Electrical Characteristics	7	FCC Certification	34
1.7. Definitions	8	OEM Labeling Requirements	34
<b>2. ZigBee Networks</b>	<b>10</b>	FCC Notices	34
2.1. ZigBee Network Formation	10	FCC-Approved Antennas (2.4 GHz)	35
2.1.1. Node Types	10	European Certification	36
2.1.2. Network Limitations	10	OEM Labeling Requirements	36
2.2. ZigBee Network Communications	12	Restrictions	36
2.2.1. ZigBee Addressing	12	Declarations of Conformity	36
2.2.2. Mesh Routing	12	<b>Appendix B: Development Guide</b>	<b>37</b>
2.2.3. Broadcast Transmissions	13	Migrating from the 802.15.4 Protocol	37
<b>3. RF Module Operation</b>	<b>14</b>	ZigBee Command Set	37
3.1. Serial Communications	14	API / AT Firmware Versions	37
3.1.1. UART Data Flow	14	XBee/XBee-PRO Development Kits	38
3.1.2. Transparent Operation	15	Interfacing Options	38
3.1.3. API Operation	15	RS-232 Development Board	39
3.1.4. Flow Control	16	External Interface	39
3.2. XBee/XBee-PRO Networks	17	RS-232 Pin Signals	40
3.2.1. XBee/XBee-PRO Coordinator	17	Wiring Diagrams	41
3.2.2. XBee/XBee-PRO Router	17	Adapters	42
3.2.3. Network Reset	18	USB Development Board	43
3.2.4. Network Mapping	18	External Interface	43
3.3. XBee/XBee-PRO Addressing	19	USB Pin Signals	43
3.3.1. 64-bit Addressing	19	X-CTU Software	44
3.3.2. API Addressing	19	Install	44
3.3.3. NI-String Addressing	20	Serial Communications Software	44
3.3.4. Coordinator Addressing	20	<b>Appendix C: Additional Information</b>	<b>45</b>
3.3.5. Broadcast Addressing	20	1-Year Warranty	45
3.4. Modes of Operation	21	Ordering Information	45
3.4.1. Idle Mode	21	Contact MaxStream	46
3.4.2. Transmit Mode	21		
3.4.3. Receive Mode	21		
3.4.4. Command Mode	22		
3.5. Network Startup Examples	23		

# 1. XBee/XBee-PRO OEM RF Modules

The XBee and XBee-PRO OEM RF Modules were engineered to operate within the ZigBee protocol and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between remote devices.

Both modules operate within the ISM 2.4 GHz frequency band and are pin-for-pin compatible with each other.



## 1.1. Key Features

### High Performance, Low Cost

#### XBee

- Indoor/Urban: up to 100' (30 m)
- Outdoor line-of-sight: up to 300' (100 m)
- Transmit Power: 1 mW (0 dBm)
- Receiver Sensitivity: -92 dBm

#### XBee-PRO

- Indoor/Urban: up to 300' (100 m)
- Outdoor line-of-sight: up to 1 mile (1500 m)
- Transmit Power: 100 mW (20 dBm) EIRP
- Receiver Sensitivity: -100 dBm

RF Data Rate: 250,000 bps

### Advanced Networking & Security

Retries and Acknowledgements  
 DSSS (Direct Sequence Spread Spectrum)  
 Each direct sequence channel has over 65,000 unique network addresses available  
 Point-to-point, point-to-multipoint and peer-to-peer topologies supported  
 Self-routing, self-healing and fault-tolerant mesh networking

### Low Power

#### XBee

- TX Current: 45 mA (@3.3 V)
- RX Current: 50 mA (@3.3 V)
- Power-down Current: < 10  $\mu$ A

#### XBee-PRO

- TX Current: 215 mA (@3.3 V)
- RX Current: 55 mA (@3.3 V)
- Power-down Current: < 10  $\mu$ A

### Easy-to-Use

No configuration necessary for out-of box RF communications

AT and API Command Modes for configuring module parameters

Small form factor

Extensive command set

Free X-CTU Software  
 (Testing and configuration software)

**Free & Unlimited Technical Support**

### 1.1.1. Worldwide Acceptance

**FCC Approval (USA)** Refer to Appendix A [p34] for FCC Requirements.  
 Systems that contain XBee/XBee-PRO RF Modules inherit MaxStream Certifications.

ISM (Industrial, Scientific & Medical) **2.4 GHz frequency band**

Manufactured under **ISO 9001:2000** registered standards

XBee/XBee-PRO RF Modules are optimized for use in **US, Canada, Australia, Israel and Europe** (contact MaxStream for complete list of agency approvals).



## 1.2. Specifications

Table 1-01. Specifications of the XBee/XBee-PRO OEM RF Module (PRELIMINARY)

Specification	XBee	XBee-PRO
Performance		
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)
Power Requirements		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Operating Current (Transmit)	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)
Operating Current (Receive)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	not supported	not supported
General		
Operating Frequency Band	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
Networking & Security		
Supported Network Topologies	Point-to-point, Point-to-multipoint, Peer-to-peer & Mesh	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses
Agency Approvals		
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)*

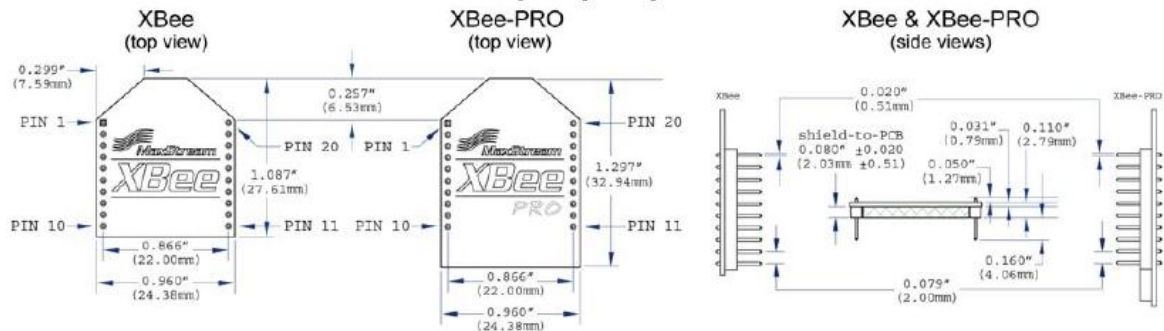
\* When operating in Europe: XBee-PRO RF Modules must be configured to operate at a maximum transmit power output level of 10 dBm. The power output level is set using the PL command. The PL parameter must equal "0" (10 dBm).

Additionally, European regulations stipulate an EIRP power maximum of 12.86 dBm (19 mW) for the XBee-PRO and 12.11 dBm for the XBee when integrating high-gain antennas.

Antenna Options: The ranges specified are typical when using the integrated Whip (1.5 dBi) and Dipole (2.1 dBi) antennas. The Chip antenna option provides advantages in its form factor; however, it typically yields shorter range than the Whip and Dipole antenna options when transmitting outdoors. For more information, refer to the "XBee Antenna" application note located on MaxStream's web site (<http://www.maxstream.net/support/knowledgebase/article.php?kb=153>).

### 1.3. Mechanical Drawings

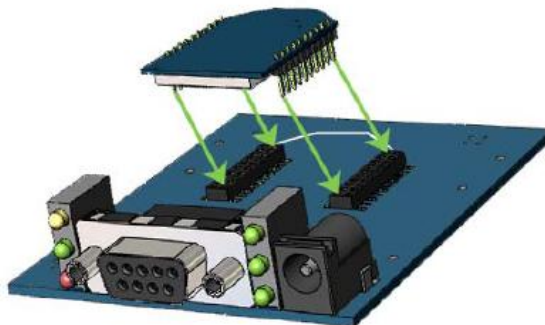
Figure 1-01. Mechanical drawings of the XBee/XBee-PRO OEM RF Modules (antenna options not shown)  
The XBee and XBee-PRO RF Modules are pin-for-pin compatible.



### 1.4. Mounting Considerations

The XBee/XBee-PRO RF Module was designed to mount into a receptacle (socket) and therefore does not require any soldering when mounting it to a board. The XBee Development Kits contain RS-232 and USB interface boards which use two 20-pin receptacles to receive modules.

Figure 1-02. XBee Module Mounting to an RS-232 Interface Board.



The receptacles used on MaxStream development boards are manufactured by Century Interconnect. Several other manufacturers provide comparable mounting solutions; however, MaxStream currently uses the following receptacles:

- Through-hole single-row receptacles - Samtec P/N: MMS-110-01-L-SV (or equivalent)
- Surface-mount double-row receptacles - Century Interconnect P/N: CPRMSL20-D-0-1 (or equivalent)
- Surface-mount single-row receptacles - Samtec P/N: SMM-110-02-SM-S

MaxStream also recommends printing an outline of the module on the board to indicate the orientation the module should be mounted.



## 1.5. Pin Signals

Figure 1-03. XBee/XBee-PRO RF Module Pin Number  
(top sides shown - shields on bottom)

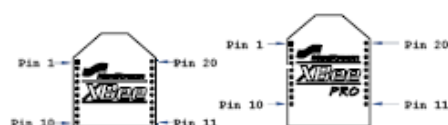


Table 1-02. Pin Assignments for the XBee and XBee-PRO Modules  
(Low-asserted signals are distinguished with a horizontal line above signal name.)

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	[reserved]	-	Do not connect
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

\* Functions not supported at the time of this release.

### Design Notes:

- Minimum connections: VCC, GND, DOUT & DIN
- Minimum connections for updating firmware: VCC, GND, DIN, DOUT, RTS & DTR
- Signal Direction is specified with respect to the module
- Module includes a 50k  $\Omega$  pull-up resistor attached to RESET
- Several of the input pull-ups can be configured using the PE command
- Unused pins should be left disconnected

## 1.6. Electrical Characteristics

Table 1-03. DC Characteristics of the XBee & XBee-PRO (VCC = 2.8 - 3.4 VDC)

Symbol	Parameter	Condition	Min	Typical	Max	Units
V <sub>IL</sub>	Input Low Voltage	All Digital Inputs	-	-	0.35 * VCC	V
V <sub>IH</sub>	Input High Voltage	All Digital Inputs	0.7 * VCC	-	-	V
V <sub>OL</sub>	Output Low Voltage	I <sub>OL</sub> = 2 mA, VCC ≥ 2.7 V	-	-	0.5	V
V <sub>OH</sub>	Output High Voltage	I <sub>OH</sub> = -2 mA, VCC ≥ 2.7 V	VCC - 0.5	-	-	V
I <sub>IN</sub>	Input Leakage Current	V <sub>IN</sub> = VCC or GND, all inputs, per pin	-	0.025	1	uA
I <sub>OZ</sub>	High Impedance Leakage Current	V <sub>IN</sub> = VCC or GND, all I/O High-Z, per pin	-	0.025	1	uA
TX	Transmit Current	VCC = 3.3 V	-	45 (XBee) 215 (PRO)	-	mA
RX	Receive Current	VCC = 3.3 V	-	50 (XBee) 55 (PRO)	-	mA
PWR-DWN	Power-down Current	SM parameter = 1	-	< 10	-	uA

## 1.7. Definitions

Table 1-04. Terms and Definitions

ZigBee Node Types	
Coordinator	<p>A node that has the unique function of forming a network. The Coordinator is responsible for establishing the operating channel and PAN ID for an entire network. Once established, the Coordinator can form a network by allowing Routers and End Devices to join to it. Once the network is formed, the Coordinator functions like a Router (it can participate in routing packets and be a source or destination for data packets).</p> <ul style="list-style-type: none"> <li>-- One Coordinator per PAN</li> <li>-- Establishes/Organizes PAN</li> <li>-- Can route data packets to/from other nodes</li> <li>-- Can be a data packet source and destination</li> <li>-- Mains-powered</li> </ul> <p>Refer to the XBee/XBee-PRO Coordinator section [p17] for more information.</p>
Router	<p>A node that creates/maintains network information and uses this information to determine the best route for a data packet. A router must join a network before it can allow other Routers and End Devices to join to it.</p> <p>A Router can participate in routing packets and is intended to be a mains-powered node.</p> <ul style="list-style-type: none"> <li>-- Several Routers can operate in one PAN</li> <li>-- Can route data packets to/from other nodes</li> <li>-- Can be a data packet source and destination</li> <li>-- Mains-powered</li> </ul> <p>Refer to the XBee/XBee-PRO Router section [p17] for more information.</p>
End Device	<p>End Devices have no routing capacity. End Devices must always interact with their parent node (Router or Coordinator) in order to transmit or receive data.</p> <p>An End Device can be a source or destination for data packets but cannot route packets. End Devices can be battery-powered and offer low-power operation.</p> <ul style="list-style-type: none"> <li>-- Several End Devices can operate in one PAN</li> <li>-- Can be a data packet source and destination</li> <li>-- All messages are relayed through a Coordinator or Router</li> <li>-- Low power</li> </ul> <p>End Devices are not supported in this release.</p>
ZigBee Protocol	
PAN	<p>Personal Area Network - A data communication network that includes a Coordinator and one or more Routers/End Devices. Network formation is governed by Network Maximum Depth, Maximum Child Routers and Maximum Children End Devices.</p>
Joining	<p>The process of a node becoming part of a ZigBee PAN. A node becomes part of a network by joining to a Coordinator or a Router (that has previously joined to the network). During the process of joining, the node that allowed joining (the parent) assigns a 16-bit address to the joining node (the child).</p>

Table 1-04. Terms and Definitions

Network Maximum Depth	The level of descendents from a Coordinator. In a MaxStream PAN, the Network Maximum Depth is 5.
Maximum Child Routers	The maximum number of Routers than can join to one node. The maximum number of Child Routers in a MaxStream PAN is 6.
Maximum Child End Devices	The maximum number of End Devices than can join to one node. The maximum number of Child End Devices in a MaxStream PAN is 14.
Network Address	The 16-bit address assigned to a node after it has joined to another node.
Operating Channel	The frequency selected for data communications between nodes. The operating channel is selected by the Coordinator on power-up.
Energy Scan	A scan of RF channels that detects the amount of energy present on the selected channels. The Coordinator uses the energy scan to determine the operating channel.
Route Request	Broadcast transmission sent by a Coordinator or Router throughout the network in attempt to establish a route to a destination node.
Route Reply	Unicast transmission sent back to the originator of the route request. It is initiated by a node when it receives a route request packet and its address matches the Destination Address in the route request packet.
Route Discovery	The process of establishing a route to a destination node when one does not exist in the Routing Table. It is based on the AODV (Ad-hoc On-demand Distance Vector routing) protocol.
ZigBee Stack	ZigBee is a published specification set of high-level communication protocols for use with small, low-power modules. The ZigBee stack provides a layer of network functionality on top of the 802.15.4 specification.  For example, the mesh and routing capabilities available to ZigBee solutions are absent in the 802.15.4 protocol.

## 2. ZigBee Networks

### 2.1. ZigBee Network Formation

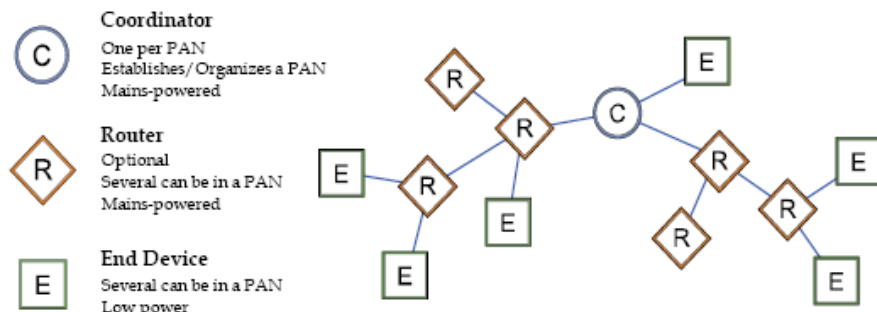
A ZigBee PAN is formed by nodes joining to a Coordinator or to a previously joined Router. Once the Coordinator defines the operating channel and PAN ID, it can allow Routers and End Devices to join to it. When a node joins a network, it receives a 16-bit Network Address.

Once a Router has joined the network, it can also allow other nodes to join to it. Joining establishes a parent/child relationship between two nodes. The node that allowed the join is the parent and the node that joined is the child. The parent/child relationship is not necessary for routing; however, it is necessary for network formation and Network Address assignment. If a Coordinator does not exist, a network cannot be formed. A node cannot transmit or receive data until it has joined a PAN.

#### 2.1.1. Node Types

A ZigBee PAN consists of one Coordinator and one or more Routers and/or End Devices. Refer to the Coordinator [p17] and Router [p17] sections of the "RF Module Operation" chapter for more information regarding each node type.

Figure 2-01. Node Types / Sample of a Basic ZigBee Network Topology



End Devices are not supported at the time of this release.

#### 2.1.2. Network Limitations

MaxStream ZigBee PANs are limited to the following boundaries:

- **Maximum Children** - The Coordinator and each joined Router can support up to 20 children, 6 of which can be Routers (Maximum Child Routers). Refer to Figure 2-02.
- **Network Maximum Depth** - The Maximum Network Depth is 5 - Maximum Network Depth refers to the level of descendants from the Coordinator. Refer to Figure 2-03.

Refer to the definitions table [p8] for more information.

Figure 2-02. Maximum Number of Child Devices per Parent

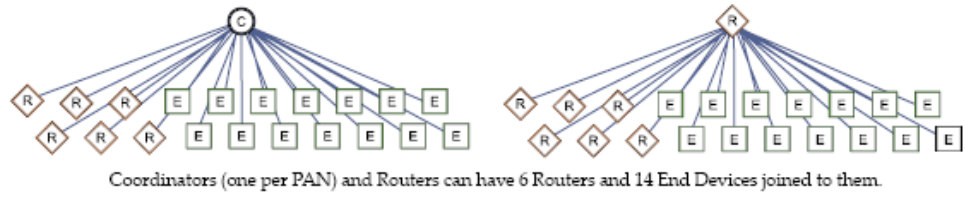
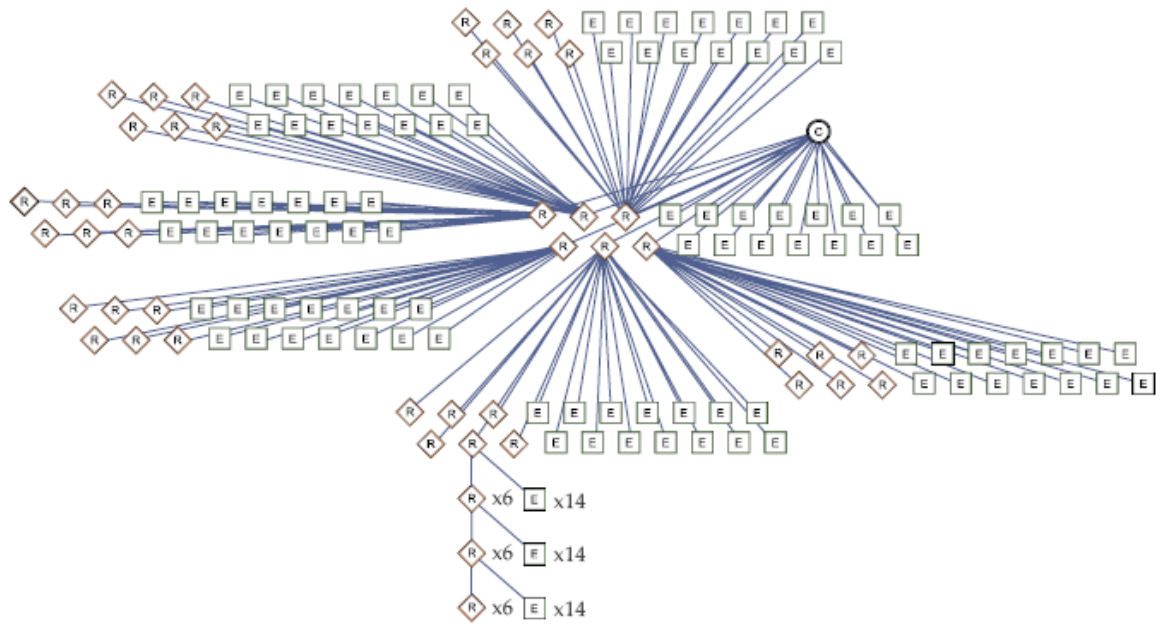


Figure 2-03. Maximum Network Depth  
Network depth can span 5 levels deep.



## 2.2. ZigBee Network Communications

---

### 2.2.1. ZigBee Addressing

---

The 802.15.4 protocol upon which the ZigBee protocol is built specifies two address types:

- 16-bit Network Addresses
- 64-bit Addresses

#### 16-bit Network Addresses

---

A 16-bit Network Address is assigned to a node when the node joins a network. The Network Address is unique to each node in the network. However, Network Addresses are not static - it can change.

The following two conditions will cause a node to receive a new Network Address:

1. An End Device cannot communicate with its parent.
2. A Router or End Device, when it power cycles, sends an 802.15.4 Orphan Notification command to locate its parent. If the parent node does not respond, the Router or End Device considers itself not-joined and repeats the process of joining the network. Once the node joins the network, it and all of its descendents will receive a new Network Address.

Since all ZigBee communications use 16-bit addressing, a node's 16-bit address must be known before communications can take place.

#### 64-bit Addresses

---

Each node contains a unique 64-bit address. The 64-bit address uniquely identifies a node and is permanent.

### 2.2.2. Mesh Routing

---

Mesh routing allows data packets to traverse multiple nodes (hops) in a network to route data from a source to a destination. The route a packet can take in a mesh network is independent of the parent/child relationships established during joining. Before transmitting a data packet from source to destination nodes, a route must be established. Route discovery is based on the AODV (Ad-hoc On-demand Distance Vector routing) protocol.

#### AODV (Ad-hoc On-demand Distance Vector) Routing Algorithm

---

Routing under the AODV protocol is accomplished using tables in each node that store in the next hop (intermediary node between source and destination nodes) for a destination node. If a next hop is not known, route discovery must take place in order to find a path. Since only a limited number of routes can be stored on a Router, route discovery will take place more often on a large network with communication between many different nodes.

When a source node must discover a route to a destination node, it sends a broadcast route request command. The route request command contains the source Network Address, the destination Network Address and a Path Cost field (a metric for measuring route quality). As the route request command is propagated through the network (refer to the Broadcast Transmission section [p13]), each node that re-broadcasts the message updates the Path Cost field and creates a temporary entry in its route discovery table.

When the destination node receives a route request, it compares the 'path cost' field against previously received route request commands. If the path cost stored in the route request is better than any previously received, the destination node will transmit a route reply packet to the node that originated the route request. Intermediate nodes receive and forward the route reply packet to the Source Node (the node that originated route request).

---

Refer to the ZigBee specification for more details.

---

### 2.2.3. Broadcast Transmissions

---

Broadcast transmissions within the ZigBee protocol are intended to be propagated throughout the entire network such that all nodes receive the transmission. This requires each broadcast transmission be retransmitted by all Router nodes to ensure all nodes receive the transmission.

Broadcast transmissions use a passive acknowledgment scheme. This means that when a node transmits a broadcast transmission, it listens to see if all of its neighbors retransmit the message. If one or more neighbor nodes do not retransmit the data, the node will retransmit the broadcast message and listen again for the neighbor nodes to forward the broadcast transmission.

---

Refer to the ZigBee specification for more details.

---

## 3. RF Module Operation

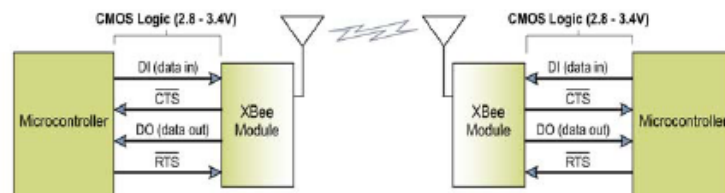
### 3.1. Serial Communications

The XBee/XBee-PRO OEM RF Modules interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device (For example: Through a Max-Stream proprietary RS-232 or USB interface board).

#### 3.1.1. UART Data Flow

Devices that have a UART interface can connect directly to the pins of the RF module as shown in the figure below.

Figure 3-01. System Data Flow Diagram in a UART-interfaced environment  
(Low-asserted signals distinguished with horizontal line over signal name.)



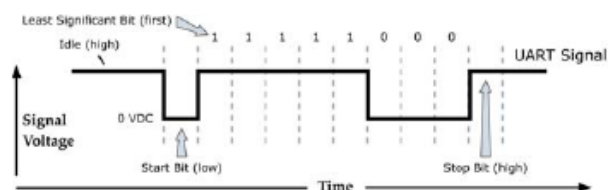
RTS flow control is not currently supported.

#### Serial Data

Data enters the module UART through the DI pin (pin 3) as an asynchronous serial signal. The signal should idle high when no data is being transmitted.

Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high). The following figure illustrates the serial bit pattern of data passing through the module.

Figure 3-02. UART data packet 0x1F (decimal number "31") as transmitted through the RF module  
Example Data Format is 8-N-1 (bits - parity - # of stop bits)



The module UART performs tasks, such as timing and parity checking, that are needed for data communications. Serial communications depend on the two UARTs to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits).



### 3.1.2. Transparent Operation

---

RF modules that contain the following firmware versions will support Transparent Mode: 8.0xx (Coordinator) and 8.2xx (Router).

When operating in Transparent Mode, modules are configured using AT Commands and API operation is not supported. The modules act as a serial line replacement - all UART data received through the DI pin is queued up for RF transmission. Data is sent to a module as defined by the DH (Destination Address High) and DL (Destination Address Low) parameters.

When RF data is received that is addressed to the module's 64-bit Address, the data is sent out the DO pin.

#### Serial-to-RF Packetization

---

Data is buffered in the DI buffer until one of the following causes the data to be packetized and transmitted:

1. No serial characters are received for the amount of time determined by the RO (Packetization Timeout) parameter. If RO = 0, packetization begins when a character is received.
2. Maximum number of characters that will fit (72) in an RF packet is received.
3. The Command Mode Sequence (GT + CC + GT) is received. Any character buffered in the DI buffer before the sequence is transmitted.

### 3.1.3. API Operation

---

API (Application Programming Interface) Operation is an alternative to the default Transparent Operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module.

When in API mode, all data entering and leaving the module is contained in frames that define operations or events within the module.

Transmit Data Frames (received through the DI pin (pin 3)) include:

- RF Transmit Data Frame
- Command Frame (equivalent to AT commands)

Receive Data Frames (sent out the DO pin (pin 2)) include:

- RF-received data frame
- Command response
- Event notifications such as reset, associate, disassociate, etc.

The API provides alternative means of configuring modules and routing data at the host application layer. A host application can send data frames to the module that contain address and payload information instead of using command mode to modify addresses. The module will send data frames to the application containing status packets; as well as source, RSSI and payload information from received data packets.

The API operation option facilitates many operations such as the examples cited below:

- > Transmitting data to multiple destinations without entering Command Mode
- > Receive success/failure status of each transmitted RF packet
- > Identify the source address of each received packet

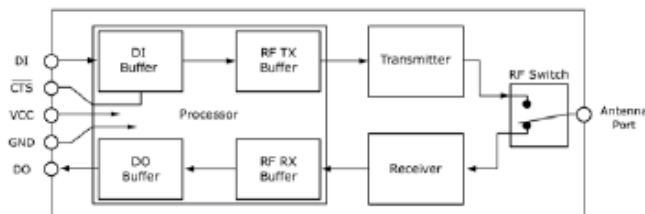
---

To implement API operations, refer to API sections [p29].

---

### 3.1.4. Flow Control

Figure 3-03. Internal Data Flow Diagram



#### DI (Data In) Buffer

When serial data enters the RF module through the DI pin (pin 3), the data is stored in the DI Buffer until it can be processed.

**Hardware Flow Control ( $\overline{CTS}$ ).** When the DI buffer is 17 bytes away from being full; by default, the module de-asserts  $\overline{CTS}$  (high) to signal to the host device to stop sending data [refer to D7 (DIO7 Configuration) parameter].  $\overline{CTS}$  is re-asserted after the DI Buffer has 34 bytes of memory available.

#### Cases in which the DI Buffer may become full and possibly overflow:

1. If the module is receiving a continuous stream of RF data, any serial data that arrives on the DI pin is placed in the DI Buffer. The data in the DI buffer will be transmitted over-the-air when the module is no longer receiving RF data in the network.
2. When data is ready to be transmitted, the module may need to discover a Network Address and/or a Route in order to reach the destination node. Discovery overhead may delay packet transmission.  
Refer to the ZigBee Networks --> Mesh Routing sections for more information.

Refer to the RO (Packetization Timeout) command description [p25] and the Mesh Routing section [p12] for more information.

#### DO (Data Out) Buffer

When RF data is received, the data enters the DO buffer and is sent out the serial port to a host device. Once the DO Buffer reaches capacity, any additional incoming RF data is lost.

**Hardware Flow Control ( $\overline{RTS}$ ).** If  $\overline{RTS}$  is enabled for flow control (D6 (DIO6 Configuration) Parameter = 1), data will not be sent out the DO Buffer as long as  $\overline{RTS}$  (pin 16) is de-asserted.

#### Cases in which the DO Buffer may become full and possibly overflow:

1. If the RF data rate is set higher than the interface data rate of the module, the module will receive data from the transmitting module faster than it can send the data to the host.
2. If the host does not allow the module to transmit data out from the DO buffer because of being held off by hardware or software flow control.

$\overline{RTS}$  flow control is not supported in this release.

## 3.2. XBee/XBee-PRO Networks

---

### 3.2.1. XBee/XBee-PRO Coordinator

---

#### Network Startup

In order to form a network, a Coordinator must select an unused operating channel and PAN ID on behalf of its network. To do this, the Coordinator first performs an energy scan on all channels as specified by its SC (Scan Channels) parameter. The scan time on each channel is determined by the SD (Scan Duration) parameter. Once the energy scan is completed, an Active Scan is issued. The Active Scan returns a list of discovered Coordinators and Routers (up to 5 results). The duration of the Active Scan on each channel is also determined by the SD parameter. An unoccupied operating channel is then chosen for PAN operations.

If the ID (PAN ID) parameter = 0xFFFF: The Coordinator will select a random PAN ID. Otherwise, the Coordinator will startup on its stored ID parameter.

After the Coordinator has started, it will allow nodes to join to it for a time based on its NJ (Node Join Time) parameter. If enabled, the Associate LED (D5 (DIO5 Configuration) command) will blink 1x per second after the Coordinator has started. At this point, the operating channel and PAN ID can be read using the CH (Operating Channel) and ID (PAN ID) commands. The 16-bit address of the Coordinator is always 0x0000.

If API is enable (AP parameter > 0):

The API Modem Status "Coordinator Started" is sent out the UART.

The AI (Association Indication) command can be used at any point during the Coordinator startup routine to determine the status of the startup operation.

### 3.2.2. XBee/XBee-PRO Router

---

#### Router Startup

A new Router must locate a Router that has already joined a PAN or a Coordinator to join to. To do this, it first issues an Active Scan on each of the SC channels. The scan duration on each of these channels is determined by the SD parameter. The Active Scan will return a list of discovered Coordinators and Routers (up to 5 results). The Router will then try to join to a parent (Router or Coordinator) that is allowing joining, based on the ID parameter. If ID = 0xFFFF, the Router will attempt to join a parent on any PAN ID. Otherwise, the Router will only attempt joining with a Router/Coordinator that operates on the PAN ID specified by the ID parameter. If a valid Router/Coordinator is found, the Router will attempt to join to that node. If the join succeeds, the Router has successfully started.

After the Router has started, it will allow nodes to join to it for a time based on the NJ (Node Join Time) parameter. If enabled, the Associate LED (D5 (DIO5 Configuration) command) will blink 2x per second after the Router has started. At this point, the operating channel and PAN ID can be read using the CH (Operating Channel) and ID (PAN ID) commands. The 16-bit Network Address of the Router can be read using the MY (16-bit Source Address) command.

If API is enable (AP parameter > 0):

The API Modem Status "Joined" is sent out the UART.

The AI (Association Indication) command can be used at any point during the Router startup routine to know the status of the startup operation.

#### Router Configuration

The SC (Scan Channel) and ID (PAN ID) parameter values affect Router startup by determining the channels the Router will scan [SC (Scan Channels) command] to find a parent to join and by determining the allowable PAN ID(s) the node will join (ID parameter).

Changing these parameters could be problematic if other nodes (children) have already joined the Router. These commands should not be changed once the Router has started and allowed nodes to join to it.

### 3.2.3. Network Reset

---

#### Resetting a Coordinator

---

**Upon reset** (Power-up, FR (Software Reset) or NR (Network Reset)):

1. If a Coordinator has formed a network, it will retain the PAN ID and operating channel information as well as a list of its child nodes. However, if either the SC (Scan Channels) or ID (PAN ID) parameters have changed and the Coordinator is reset, the Coordinator will startup using the new SC and/or ID values and will erase its list of child nodes.
2. If the Coordinator must change the operating channel of an established network, it can alert all nodes in the network to leave and reform the network by issuing the NR (Network Reset) command with a parameter of '1'. When this command is issued, the Coordinator sends a broadcast message across the entire network forcing all nodes to unjoin and rejoin to a new parent. The Coordinator will, after several seconds, restart and allow joining according to its saved NJ (Node Join Time) setting. Once the Coordinator has started, other nodes can join the PAN. The other nodes will attempt to rejoin by scanning all channels (as specified by the SC parameter) for a parent operating on the PAN ID (specified by its ID parameter). This will re-assign the 16-bit Network Addresses on all nodes. The NJ parameter value on the Coordinator must be non-zero if the NR command is issued to allow at least one Router to join to it.

#### Resetting a Router

---

**Upon reset** (Power-up, FR (Software Reset) or NR (Network Reset)):

1. If a Router has joined a network and a reset occurs (FR or power-up), the Router will send a broadcast transmission to find its parent in order to re-establish its operating channel, PAN ID and 16-bit Network Address. If the Router cannot find its parent, it will rejoin the network using its SC (Scan Channels) and ID (PAN ID) settings. This could change the Router's 16-bit Network Address.
2. Issuing an NR (Network Reset) command with a parameter of '0' will also cause a Router reset. The NR command will force the Router to rejoin the network and it may receive a different 16-bit Network Address. It will also erase the Router's list of child device addresses. This option should be used with caution.
3. Issuing an NR (Network Reset) command with a parameter of '1' will send a command to the Coordinator instructing it to reset the entire network. Resetting the entire network will reset all nodes and reform the PAN (Also refer to the "Resetting a Coordinator" section).

### 3.2.4. Network Mapping

---

The ND (Node Discover) command is useful for mapping out the network. When issued from the Coordinator or an End Device, the command sends a broadcast message across the network and returns a list of all nodes. Refer to the ND command for more information.

### 3.3. XBee/XBee-PRO Addressing

Each RF module has a unique 64-bit Address that is assigned at the factory that can be read using the SH (Serial Number High) and SL (Serial Number Low) commands. When a module starts or joins a network, it receives a 16-bit Network Address that is unique within the network; however, this address can change (refer to the 'ZigBee Addressing' section). In addition, each module can store a string-identifier using the NI (Node Identifier) command.

XBee/XBee-PRO RF modules can be addressed using their 64-bit Address, their NI-string or both the 64-bit Address and the 16-bit Network Address (API Mode). Under the ZigBee protocol, the 16-bit Network Address of a Coordinator is always "0". Nodes can address the Coordinator using its 16-bit Network Address.

#### 3.3.1. 64-bit Addressing

To send a packet to an RF module using its 64-bit Address (Transparent Mode)

Set the DH (Destination Address High) and DL (Destination Address Low) parameters of the source node to match the 64-bit Address (SH (Serial Number High) and SL (Serial Number Low) parameters) of the destination node.

To send a packet to an RF module using its 64-bit Address (API Mode)

Use the ZigBee Transmit Request API frame to set the DH (Destination Address High) and DL (Destination Address Low) parameters of the source node to match the 64-bit Address (SH (Serial Number High) and SL (Serial Number Low) parameters) of the destination node.  
If the 64-bit Address of the destination node is not known, set 16-bit Destination Network Address to 0xFFFFE (refer to the 'API Addressing section below).

Since the ZigBee protocol relies on the 16-bit Network Address for routing, the 64-bit Address must be converted into a 16-bit Network Address prior to transmitting data. If a module does not know the 16-bit Network Address for a given 64-bit Address, it will transmit a broadcast Network Address Discovery command. The module with a matching 64-bit Address will transmit its 16-bit Network Address back.

The modules maintain a table that can store up to seven 64-bit Addresses and their corresponding 16-bit Network Addresses.

#### 3.3.2. API Addressing

API Mode provides the ability to store and maintain 16-bit Network Address tables on an external processor. The 16-bit Network Address information is provided to the application through the following:

- The ZigBee Transmit Status Frame (contains the current 16-bit Network Address of the remote)
- The ND and DN commands (return 64-bit and 16-bit Network Addresses of remote nodes)

With this information, a table can be built in an application that maps a 64-bit Address to the corresponding 16-bit Network Address.

The ZigBee Transmit Request API frame specifies the 64-bit Address and the Network Address (if known) that the packet should be sent to. By supplying both addresses, the module will forego Network Address Discovery and immediately attempt to route the data packet to the remote. If the Network Address of a particular remote changes, Network Address and route discovery will take place to establish a new route to the correct node. Upon successful packet delivery, the TX Status Frame will indicate the correct Network Address of the remote.

Table 3-01. Sample table mapping 64-bit Addresses to 16-bit Network Addresses

Index	64-bit Address	16-bit Network Address
0	0013 4000 4000 0001	1234
1	0013 4000 4000 0002	5678
2	0013 4000 4000 01A0	A479
3	0013 4000 4000 0220	1F70

### 3.3.3. NI-String Addressing

---

To send a packet to an RF module using its NI-string (Transparent Mode)

Issue the DN (Destination Node) command using the NI (Node Identifier)-string of the destination node as the parameter.

To send a packet to an RF module using its NI-string (API Mode)

Issue the DN command as stated above using the AT Command API frame.

When the DN command is issued, a broadcast transmission is sent across the network to discover the module that has a matching NI (Node Identifier) parameter. If a module is discovered with a matching NI-string, the DH and DL parameters will be configured to address the destination node and the command will return both the 64-bit Address and the 16-bit Network Address of the discovered node. Data can be transmitted after the DN (Destination Node) command finishes.

### 3.3.4. Coordinator Addressing

---

A Coordinator can be addressed using its 64-bit address or NI string as described in the "NI-String Addressing" section. Alternatively, since the ZigBee Coordinator has a Network Address of "0", it can be addressed by its 16-bit Network Address.

To send a transmission to a Coordinator using its 16-bit Network Address:

Set the Destination Addresses of the transmitting module as shown below:  
 DL (Destination Low Address) = 0  
 DH (Destination High Address) = 0

### 3.3.5. Broadcast Addressing

---

Broadcast transmissions are sent using a 64-bit address of 0x0000FFFF. Any RF module in the PAN will accept a packet that contains a broadcast address. When configured to operate in Broadcast Mode, receiving modules do not send ACKs (Acknowledgements).

To send a broadcast packet to all modules

Set the Destination Addresses of the transmitting module as shown below:  
 DL (Destination Low Address) = 0x0000FFFF  
 DH (Destination High Address) = 0x00000000

NOTE: When programming the module, parameters are entered in hexadecimal notation (without the "0x" prefix). Leading zeros may be omitted.

Refer to the "Broadcast Transmissions" section [p13] for more information.

## 3.4. Modes of Operation

### 3.4.1. Idle Mode

When not receiving or transmitting data, the RF module is in Idle Mode. During Idle Mode, the RF module is also checking for valid RF data. The module shifts into the other modes of operation under the following conditions:

- Transmit Mode (Serial data in the DI Buffer is ready to be packetized)
- Receive Mode (Valid RF data is received through the antenna)
- Sleep Mode (End Devices only - not supported in this release)
- Command Mode (Command Mode Sequence is issued)

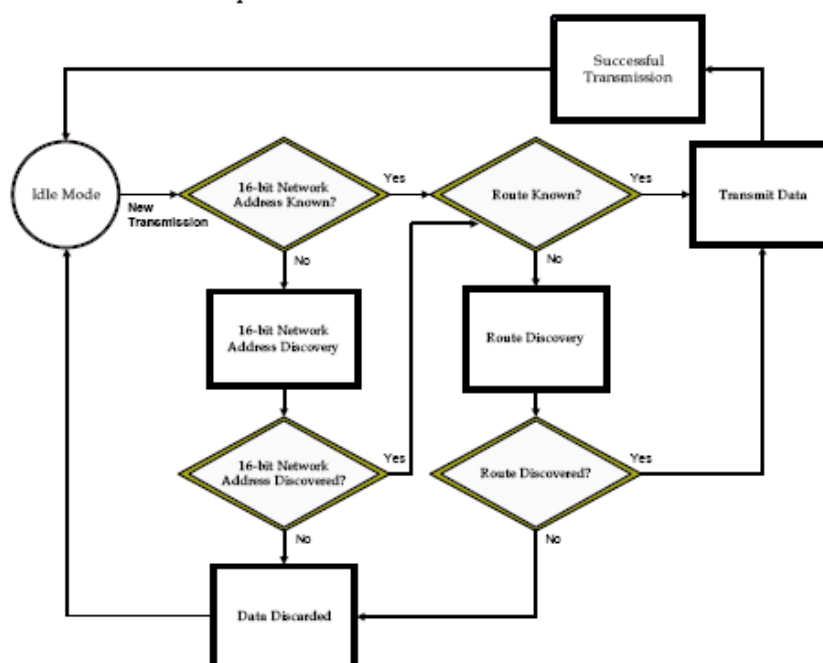
### 3.4.2. Transmit Mode

When serial data is received and is ready for packetization, the RF module will exit Idle Mode and attempt to transmit the data. The destination address determines which node(s) will receive the data.

Prior to transmitting the data, the module ensures that a 16-bit Network Address and route to the destination node have been established.

If the 16-bit Network Address is not known, Network Address Discovery will take place. If a route is not known, route discovery will take place for the purpose of establishing a route to the destination node. If a module with a matching Network Address is not discovered, the packet is discarded. The data will be transmitted once a route is established. If route discovery fails to establish a route, the packet will be discarded.

Figure 3-04. Transmit Mode Sequence



When data is transmitted from one node to another, a network-level acknowledgement is transmitted back across the established route to the source node. This acknowledgement packet indicates to the source node that the data packet was received by the destination node. If a network acknowledgement is not received, the source node will re-transmit the data.

### 3.4.3. Receive Mode

If a valid RF packet is received and its address matches the RF module's MY (16-bit Source Address) parameter, the data is transferred to the DO buffer.

### 3.4.4. Command Mode

To modify or read RF Module parameters, the module must first enter into Command Mode - a state in which incoming serial characters are interpreted as commands. Refer to the API Mode section [p29] for an alternate means of configuring modules.

#### AT Command Mode

##### To Enter AT Command Mode:

Send the 3-character command sequence "+++" and observe guard times before and after the command characters. [Refer to the "Default AT Command Mode Sequence" below.]

Default AT Command Mode Sequence (for transition to Command Mode):

- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]
- Input three plus characters ("+++") within one second [CC (Command Sequence Character) parameter = 0x2B.]
- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]

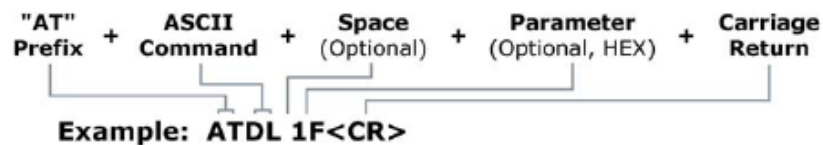
All of the parameter values in the sequence can be modified to reflect user preferences.

NOTE: Failure to enter AT Command Mode is most commonly due to baud rate mismatch. Ensure the 'Baud' setting on the "PC Settings" tab matches the interface data rate of the RF module. By default, the BD parameter = 3 (9600 bps).

##### To Send AT Commands:

Send AT commands and parameters using the syntax shown below.

Figure 3-05. Syntax for sending AT Commands



To read a parameter value stored in the RF module's register, omit the parameter field.

The preceding example would change the RF module Destination Address (Low) to "0x1F". To store the new value to non-volatile (long term) memory, subsequently send the WR (Write) command.

For modified parameter values to persist in the module's registry after a reset, changes must be saved to non-volatile memory using the WR (Write) Command. Otherwise, parameters are restored to previously saved values after the module is reset.

**System Response.** When a command is sent to the module, the module will parse and execute the command. Upon successful execution of a command, the module returns an "OK" message. If execution of a command results in an error, the module returns an "ERROR" message.

##### To Exit AT Command Mode:

1. Send the ATCN (Exit Command Mode) command (followed by a carriage return).  
[OR]
2. If no valid AT Commands are received within the time specified by CT (Command Mode Timeout) Command, the RF module automatically returns to Idle Mode.

For an example of programming the RF module using AT Commands and descriptions of each configurable parameter, refer to the "RF Module Configuration" chapter [p24].



### 3.5. Network Startup Examples

---

#### Start the Coordinator

1. Determine the operating channels list using the SC (Scan Channels) command and the PAN ID to operate using the ID (PAN ID) command. The default SD (Scan Duration) parameter value should suffice. If these values are changed from the defaults, they must be written to non-volatile memory using the WR (Write) command.
2. The Associate LED, if enabled using the D5 (DIO5 Configuration) parameter, will start blinking 1x per second once the Coordinator has started.  
If API is enabled (AP parameter > 0): The API Modem Status "Coordinator Started" is sent out the UART.
3. The AI (Association Indication) parameter will be 0 signifying a successful startup.
4. The MY (16-bit Source Address) attribute is 0 (the 16-bit Network Address of a ZigBee Coordinator).
5. After startup, the Coordinator will allow joining based on its NJ (Node Join Time) value.
6. It is recommended that the Coordinator be configured with an NI-String identifier. This NI-String identifier should be written to non-volatile memory using the WR (Write) command to be preserved through power-loss.

#### Adding a Child (Router)

1. Determine the operating channel list (SC) and the desired PAN ID to join (ID) (0xFFFF - join any Pan). The default SD parameter should suffice. If these values are changed from the defaults, they must be written to non-volatile memory using the WR (Write) command.
2. The Router, on power-up, will attempt to locate a parent to join according to its SC and ID parameters.
3. Once the Router has joined a parent, the Associate LED, if enabled (D5), will start blinking 2x per second. The ID and CH parameters will reflect the operating PAN ID and Channel. The MY parameter will reflect the 16-bit Network Address of the Router. The MP command returns the 16-bit Network Address of the Router's parent (node it joined to).  
If API is enabled (AP parameter > 0): The API Modem Status "Joined" is sent out the UART.
4. If the Router is not joining as expected, the AI (Association Indication) parameter can be read to determine the cause of failure.  
Verify the PAN contains a Coordinator or nearby joined Router that has matching Channel (SC, CH) and PAN ID (ID) settings and is allowing nodes to join to it (NJ parameter).
5. Once the Router has joined a PAN, the Router will allow joining based on the NJ parameter.
6. It is recommended that the Router be configured with a unique NI-String identifier. This NI-String identifier should be written to non-volatile memory using the WR (Write) command to be preserved through power-loss.

#### Transmit Data

1. Start a Coordinator (refer to instructions above).
2. Add one or more Child Router(s) to the Coordinator (refer to instructions above).
3. Once the Coordinator has started, all Routers and End Devices should join to a parent and their Associate LED should blink 2x per second.
4. If any nodes have not joined, read the AI command to determine why.
5. Issue the ATND command on the Coordinator to get a list of all nodes on the network.
6. Use the 'Terminal' tab of the X-CTU Software to send serial data between nodes. The data should be transmitted from the source to the destination node as specified by the DH & DL parameters.
7. (Optional) Change the Destination Address on any node to one of the 64-bit Addresses discovered using the ND command in step 5 (DH, DL Commands, or in the 'ZigBee Transmit Request' API Frame). Then repeat step 6 to transmit data.

## 4. RF Module Configuration

Two command mode protocols are supported by the XBee/XBee-PRO RF Modules: AT and API. The AT Command Mode protocol [p22] is printable and is intended for manual entry of commands and viewing parameter values. The API Command Mode protocol [p29] is a binary protocol and is intended for programmatic gets and sets of values.

Each command mode protocol requires its own firmware version. Refer to the VR (Firmware Version) command in the Diagnostics table to determine firmware versions.

### 4.1. AT Command Programming Examples

Refer to the 'X-CTU' section of the Development Guide [Appendix B] for more information regarding the X-CTU configuration software.

#### Setup

The programming examples in this section require the installation of MaxStream's X-CTU Software and a serial connection to a PC. (MaxStream stocks RS-232 and USB boards to facilitate interfacing with a PC.)

1. Install MaxStream's X-CTU Software to a PC by double-clicking the "setup\_X-CTU.exe" file. (The file is located on the MaxStream CD and under the 'Software' section of the following web page: [www.maxstream.net/support/downloads.php](http://www.maxstream.net/support/downloads.php))
2. Mount the RF module to an interface board, then connect the module assembly to a PC.
3. Launch the X-CTU Software and select the 'PC Settings' tab. Verify the baud and parity settings of the Com Port match those of the RF module.

NOTE: Failure to enter AT Command Mode is most commonly due to baud rate mismatch. Ensure the 'Baud' setting on the 'PC Settings' tab matches the interface data rate of the RF module. By default, the BD parameter = 3 (which corresponds to 9600 bps).

#### Sample Configuration: Modify RF Module Destination Address

Example: Utilize the X-CTU "Terminal" tab to change the RF module's DL (Destination Address Low) parameter and save the new address to non-volatile memory.

After establishing a serial connection between the RF module and a PC [refer to the 'Setup' section above], select the "Terminal" tab of the X-CTU Software and enter the following command lines ('CR' stands for carriage return):

Method 1 (One line per command)

Send AT Command	System Response
+++	OK <CR> (Enter into Command Mode)
ATDL <Enter>	{current value} <CR> (Read Destination Address Low)
ATDL1A0D <Enter>	OK <CR> (Modify Destination Address Low)
ATWR <Enter>	OK <CR> (Write to non-volatile memory)
ATCN <Enter>	OK <CR> (Exit Command Mode)

Method 2 (Multiple commands on one line)

Send AT Command	System Response
+++	OK <CR> (Enter into Command Mode)
ATDL <Enter>	{current value} <CR> (Read Destination Address Low)
ATDL1A0D,WR,CN <Enter>	OK<CR> OK<CR> OK<CR>

#### Sample Configuration: Restore RF Module Defaults

Example: Utilize the X-CTU "Modem Configuration" tab to restore default parameter values.

After establishing a connection between the module and a PC [refer to the 'Setup' section above], select the "Modem Configuration" tab of the X-CTU Software.

1. Select the 'Read' button.
2. Select the 'Restore' button.

## 4.2. Command Reference Tables

XBee/XBee-PRO RF Modules expect numerical values in hexadecimal. Hexadecimal values are designated by a "0x" prefix. Decimal equivalents are designated by a "d" suffix. Table rows are sorted by command category, then by logic of most commonly used.

### Special

Table 4-02. Special Commands

AT Command	Command Category	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
WR	Special	<b>Write.</b> Write parameter values to non-volatile memory so that parameter modifications persist through a subsequent resets. Note: Once WR is issued, no additional characters should be sent to the module until after the "OK!" response is received.	CRE	--	--
RE	Special	<b>Restore Defaults.</b> Restore module parameters to factory defaults. RE command does not reset the ID parameter.	CRE	--	--
FR	Special	<b>Software Reset.</b> Reset module. Responds immediately with an "OK" then performs a reset ~100ms later. Use of the FR command will cause a network layer restart on the node if SC or ID were modified since the last reset.	CRE	--	--
NR	Special	<b>Network Reset.</b> Reset network layer parameters on one or more modules within a PAN. Responds immediately with an "OK" then causes a network restart. All network configuration and routing information is consequently lost. If NR = 0: Resets network layer parameters on the node issuing the command. This option is only supported on Routers and End Devices and must be used with caution. Refer to the "Resetting Coordinator" section for more information. If NR = 1: Sends broadcast transmission to reset network layer parameters on all nodes in the PAN.	RE (when NR=0), CRE (when NR=1)	0 - 1	--

1. Node types that support the command: C = Coordinator, R = Router, E = End Device

### Networking & Security

Table 4-03. Networking Commands (Sub-categories designated within [brackets])

AT Command	Command Category	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
CH	Networking (Addressing)	<b>Operating Channel.</b> Read the channel number used for transmitting and receiving between RF modules. Uses 802.15.4 channel numbers.	CRE	0, 0x0B-0x1A (XBee) 0, 0x0C-0x17 (XBee-PRO) [read-only]	0
ID	Networking (Addressing)	<b>PAN ID.</b> Set/Get the PAN (Personal Area Network) ID. <i>Coordinator</i> - Set the preferred Pan ID. Set (ID = 0xFFFF) to auto-select. <i>Router / End Device</i> - Set the desired Pan ID. When the device searches for a Coordinator, it attempts to only join to a parent that has a matching Pan ID. Set (ID = 0xFFFF) to join a parent operating on any Pan ID. Changes to ID should be written to non-volatile memory using the WR command. ID changes are not used until the module is reset (FR, NR or power-up).	CRE	0 - 0x3FFF, 0xFFFF	0x0123 (291d)
DH <sup>2</sup>	Networking (Addressing)	<b>Destination Address High.</b> Set/Get the upper 32 bits of the 64-bit destination address. When combined with DL, it defines the destination address used for transmission. 0x0000000000000000 is the broadcast address for the PAN. DH is not supported in API Mode. 0x0000000000000000 is the Coordinator's 16-bit Network Address.	CRE	0 - 0xFFFFFFFF	0
DL <sup>2</sup>	Networking (Addressing)	<b>Destination Address Low.</b> Set/Get the lower 32 bits of the 64-bit destination address. When combined with DH, DL defines the destination address used for transmission. 0x0000000000000000 is the broadcast address for the PAN. DL is not supported in API Mode. 0x0000000000000000 is the Coordinator's 16-bit Network Address.	CRE	0 - 0xFFFFFFFF	0xFFFF (Coordinator) 0 (Router/End Device)
MY	Networking (Addressing)	<b>16-bit Network Address.</b> Get the 16-bit Network Address of the module.	CRE	0 - 0xFFFE [read-only]	0xFFFE
MP	Networking (Addressing)	<b>16-bit Parent Network Address.</b> Get the 16-bit parent Network Address of the module.	RE	0 - 0xFFFE [read-only]	0xFFFE
SH	Networking (Addressing)	<b>Serial Number High.</b> Read high 32 bits of the RF module's unique IEEE 64-bit address. 64-bit source address is always enabled.	CRE	0 - 0xFFFFFFFF [read-only]	factory-set
SL	Networking (Addressing)	<b>Serial Number Low.</b> Read low 32 bits of the RF module's unique IEEE 64-bit address. 64-bit source address is always enabled.	CRE	0 - 0xFFFFFFFF [read-only]	factory-set
RN	Networking (Addressing)	<b>Random Delay Slots.</b> Set/Read the minimum value of the back-off exponent in the CSMA-CA algorithm (used for collision avoidance). If RN = 0, collision avoidance is disabled during the first iteration of the algorithm (802.15.4 - macMinBE).	CRE	0 - 3 [exponent]	3
NI	Networking (Identification)	<b>Node Identifier.</b> Stores a string identifier. The register only accepts printable ASCII data. In AT Command Mode, a string can not start with a space. A carriage return ends the command. Command will automatically end when maximum bytes for the string have been entered. This string is returned as part of the ND (Node Discover) command. This identifier is also used with the DN (Destination Node) command.	CRE	20-Byte printable ASCII string	--

Table 4-03. Networking Commands (Sub-categories designated within (brackets))

AT Command	Command Category	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default																
BH	Networking (Identification)	<b>Broadcast Hops.</b> Set/Read the maximum number of hops for each broadcast data transmission. Setting this to 0 will use the maximum number of hops.	CRE	0 - 7																	
NT	Networking (Identification)	<b>Node Discover Timeout.</b> Set/Read the amount of time a node will spend discovering other nodes when ND or DN is issued.	CRE	0 - 0xFC [x 100 msec]	0x3C (60d)																
ND	Networking (Identification)	<p><b>Node Discover.</b> Discovers and reports all RF modules found. The following information is reported for each module discovered.</p> <p>MY&lt;CR&gt; SH&lt;CR&gt; SL&lt;CR&gt; NI&lt;CR&gt; (Variable length) PARENT_NETWORK_ADDRESS (2 Bytes)&lt;CR&gt; DEVICE_TYPE&lt;CR&gt; (1 Byte: 0=Coord, 1=Router, 2=End Device) STATUS&lt;CR&gt; (1 Byte: Reserved) PROFILE_ID&lt;CR&gt; (2 Bytes) MANUFACTURER_ID&lt;CR&gt; (2 Bytes) &lt;CR&gt;</p> <p>After (NT * 100) milliseconds, the command ends by returning a &lt;CR&gt;. ND also accepts a Node Identifier (NI) as a parameter (optional). In this case, only a module that matches the supplied identifier will respond.</p> <p>If ND is sent through the API, each response is returned as a separate AT_CMD_Response packet. The data consists of the above listed bytes without the carriage return delimiters. The NI string will end in a "0x00" null character. Once the ND command has finished executing, the API will return one more AT_CMD_Response packet (with no data).</p>	CE	optional 20-Byte NI or MY value	--																
DN	Networking (Identification)	<p><b>Destination Node.</b> Resolves an NI (Node Identifier) string to a physical address (case-sensitive). The following events occur after the destination node is discovered:</p> <p>&lt;AT Firmware&gt;</p> <ol style="list-style-type: none"> <li>DL &amp; DH are set to the extended (64-bit) address of the module with the matching NI (Node Identifier) string.</li> <li>OK (or ERROR) is returned.</li> <li>Command Mode is exited to allow immediate communication</li> </ol> <p>&lt;API Firmware&gt;</p> <ol style="list-style-type: none"> <li>The 16-bit network and 64-bit extended addresses are returned in an API Command Response frame.</li> </ol> <p>If there is no response from a module within (NT * 100) milliseconds or a parameter is not specified (left blank), the command is terminated and an "ERROR" message is returned. In the case of an ERROR, Command Mode is not exited.</p>	CRE	up to 20-Byte printable ASCII string	--																
SC	Networking (Joining)	<p><b>Scan Channels.</b> Set/Read the list of channels to scan.</p> <p><b>Coordinator</b> - Bit field list of channels to choose from prior to starting network.</p> <p><b>Router/End Device</b> - Bit field list of channels that will be scanned to find a Coordinator/Router to join.</p> <p>Changes to SC should be written using WR command. SC changes are not used until the module is reset (FR, NR or power-up).</p> <p>Bit (Channel):</p> <table border="0"> <tr> <td>0 (0x0B)</td> <td>4 (0x0F)</td> <td>8 (0x13)</td> <td>12 (0x17)</td> </tr> <tr> <td>1 (0x0C)</td> <td>5 (0x10)</td> <td>9 (0x14)</td> <td>13 (0x18)</td> </tr> <tr> <td>2 (0x0D)</td> <td>6 (0x11)</td> <td>10 (0x15)</td> <td>14 (0x19)</td> </tr> <tr> <td>3 (0x0E)</td> <td>7 (0x12)</td> <td>11 (0x16)</td> <td>15 (0x1A)</td> </tr> </table>	0 (0x0B)	4 (0x0F)	8 (0x13)	12 (0x17)	1 (0x0C)	5 (0x10)	9 (0x14)	13 (0x18)	2 (0x0D)	6 (0x11)	10 (0x15)	14 (0x19)	3 (0x0E)	7 (0x12)	11 (0x16)	15 (0x1A)	CRE	1 - 0xFFFF[bitfield] (bits 0, 13, 14, 15 not allowed for XBee-PRO)	0x1FFE (All XBee-PRO channels)
0 (0x0B)	4 (0x0F)	8 (0x13)	12 (0x17)																		
1 (0x0C)	5 (0x10)	9 (0x14)	13 (0x18)																		
2 (0x0D)	6 (0x11)	10 (0x15)	14 (0x19)																		
3 (0x0E)	7 (0x12)	11 (0x16)	15 (0x1A)																		
SD	Networking (Joining)	<p><b>Scan Duration.</b> Set/Read the scan duration exponent. Changes to SD should be written using WR command.</p> <p><b>Coordinator</b> - Duration of the Active and Energy Scans (on each channel) that are used to determine an acceptable channel and Pan ID for the Coordinator to startup on.</p> <p><b>Router/End Device</b> - Duration of Active Scan (on each channel) used to locate an available Coordinator / Router to join during Association.</p> <p>Scan Time is measured as: (# Channels to Scan) * (2 ^ SD) * 15.36ms - The number of channels to scan is determined by the SC parameter. The XBee can scan up to 16 channels (SC = 0xFFFF) and the XBee-PRO can scan up to 12 channels (0x1FFE).</p> <p>Sample Scan Duration times (13 channel scan):</p> <p>If SD = 0, time = 0.200 sec SD = 2, time = 0.799 sec SD = 4, time = 3.190 sec SD = 6, time = 12.780 sec</p>	CRE	0 - 7 [exponent]	3																
NJ	Networking (Joining)	<p><b>Node Join Time.</b> Set/Read the time that a Coordinator/Router allows nodes to join. This value can be changed at run time without requiring a Coordinator or Router to restart. The time starts once the Coordinator or Router has started. The timer is reset on power-cycle or when NJ changes.</p>	CR	0 - 0x40, 0xFF [x 1 sec]	0xFF (always allows joining)																

Table 4-03. Networking Commands (Sub-categories designated within (brackets))

AT Command	Command Category	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
AI	Networking (Joining)	Association Indication. Read information regarding last node join request. 0x00 - Successful completion - Coordinator started or Router/End Device found and joined with a parent. 0x21 - Scan found no PANs 0x22 - Scan found no valid PANs based on current SC and ID settings 0x23 - Valid Coordinator or Routers found, but they are not allowing joining (NJ expired) 0x24 - Router Full - Allow Join set, but cannot allow any more Routers to join 0x25 - Router Full - Allow Join set, but cannot allow any more End Devices to join 0x26 - Cannot join to a node because it was a child or descendent of this device 0x27 - Node Joining attempt failed 0x28 - Device is orphaned and is looking for its parent using Orphan Scans 0x29 - Router Start attempt failed <sup>2</sup> 0x2A - Coordinator Start attempt failed <sup>2</sup> 0xFF - Scanning for a Parent	CRE	0 - 0xFF [read-only]	-

1. Node types that support the command: C=Coordinator, R=Router, E=End Device
2. Command supported by modules using AT Command firmware only

### RF Interfacing

Table 4-04. RF Interfacing Commands

AT Command	Command Category	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
PL	RF Interfacing	Power Level. Select/Read the power level at which the RF module transmits conducted power.	CRE	0 - 4 (XBee / XBee-PRO) 0 = -10 / 10 dBm 1 = -6 / 12 dBm 2 = -4 / 14 dBm 3 = -2 / 16 dBm 4 = 0 / 18 dBm	4
CA	RF Interfacing	CCA Threshold. Set/Read the CCA (Clear Channel Assessment) threshold. Prior to transmitting a packet, a CCA is performed to detect energy on the channel. If the detected energy is above the CCA Threshold, the module will not transmit the packet.	CRE	0x24 - 0x50 [-dBm]	0x40 (-64dBm)

1. Node types that support the command: C = Coordinator, R = Router, E = End Device

### Serial Interfacing (I/O)

Table 4-05. Serial Interfacing Commands

AT Command	Command Category	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
AP <sup>2</sup>	Serial Interfacing	API Enable. Enable API Mode. The AP parameter is only applicable when using modules that contain the following firmware versions: 8.1xx (Coordinator), 8.3xx (Router), 8.5xx (End Device)	CRE	1 - 2 1 = API-enabled 2 = API-enabled (w/escaped control characters)	1
BD	Serial Interfacing	Interface Data Rate. Set/Read the serial interface data rate for communication between the module serial port and host. Any value above 0x07 will be interpreted as an actual baud rate. When a value above 0x07 is sent, the closest interface data rate represented by the number is stored in the BD register.	CRE	0 - 7 (standard baud rates) 0 = 1200 bps 1 = 2400 2 = 4800 3 = 9600 4 = 19200 5 = 38400 6 = 57600 7 = 115200 0x80 - 0x1C200 (non-standard rates)	3
RO	Serial Interfacing	Packetization Timeout. Set/Read number of character times of inter-character silence required before packetization. Set (RO=0) to transmit characters as they arrive instead of buffering them into one RF packet.	CRE	0 - 0xFF [x character times]	3
D7	Serial Interfacing	DIO7 Configuration. Select/Read options for the DIO7 line of the RF module.	CRE	0 - 1 0 = Disabled 1 = CTS Flow Control	1
D5	Serial Interfacing	DIO5 Configuration. Configure options for the DIO5 line of the RF module. Options include: Associated LED indicator (LED blinks 1x/sec when the module is powered and 2x/sec when module is associated to a Coordinator.).	CRE	0 - 1 0 = Disabled 1 = Associated indication LED	1

Table 4-05. Serial Interfacing Commands

AT Command	Command Category	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
P0	Serial Interfacing	PWM0 Configuration. Select/Read function for PWM0.	CRE	0 - 1 0 = Disabled 1 = RSSI PWM	1
RP	Diagnostics	RSSI PWM Timer. Time RSSI signal will be output after last transmission. When RP = 0xFF, output will always be on.	CRE	0 - 0xFF [x 100 ms]	0x28 (40d)

1. Node types that support the command: C = Coordinator, R = Router, E = End Device
2. Command supported by modules using API firmware only

### Diagnostics

Table 4-06. Diagnostics Commands

AT Command	Command Category	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
VR	Diagnostics	Firmware Version. Read firmware version of the module.	CRE	0 - 0xFFFF [read-only]	Factory-set
HV	Diagnostics	Hardware Version. Read hardware version of the module.	CRE	0 - 0xFFFF [read-only]	Factory-set

1. Node types that support the command: C = Coordinator, R = Router, E = End Device

### AT Command Options

Table 4-07. AT Command Options Commands

AT Command	Command Category	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
CT <sup>2</sup>	AT Command Mode Options	Command Mode Timeout. Set/Read the period of inactivity (no valid commands received) after which the RF module automatically exits AT Command Mode and returns to Idle Mode.	CRE	2 - 0x028F [x 100 ms]	0x64 (100d)
CN <sup>2</sup>	AT Command Mode Options	Exit Command Mode. Explicitly exit the module from AT Command Mode.	CRE	--	--
GT <sup>2</sup>	AT Command Mode Options	Guard Times. Set required period of silence before and after the Command Sequence Characters of the AT Command Mode Sequence (GT + CC + GT). The period of silence is used to prevent inadvertent entrance into AT Command Mode.	CRE	1 - 0x0CE4 [x 1 ms] (max of 3.3 decimal sec)	0x3E8 (1000d)
CC <sup>2</sup>	AT Command Mode Options	Command Sequence Character. Set/Read the ASCII character value to be used between Guard Times of the AT Command Mode Sequence (GT + CC + GT). The AT Command Mode Sequence enters the RF module into AT Command Mode. CC command is only applicable when using modules that contain the following "AT Command" firmware versions: 8.0xx (Coordinator), 8.2xx (Router), 8.4xx (End Device)	CRE	0 - 0xFF	0x2B ( '+' ASCII)

1. Node types that support the command: C = Coordinator, R = Router, E = End Device
2. Command supported by modules using AT Command firmware only

### 4.3. API Operation

Non-API XBee/XBee-PRO RF Modules act as a serial line replacement (Transparent Operation) - all UART data received through the DI pin is queued up for RF transmission. When the module receives an RF packet, the data is sent out the DO pin with no additional information.

Inherent to Transparent Operation are the following behaviors:

- If module parameter registers are to be set or queried, a special operation is required for transitioning the module into Command Mode [refer to p22].
- In point-to-multipoint systems, the application must send extra information so that the receiving module(s) can distinguish between data coming from different remotes.

As an alternative to the default Transparent Operation, API (Application Programming Interface) Operations are available. API operation requires that communication with the module be done through a structured interface (data is communicated in frames in a defined order). The API specifies how commands, command responses and module status messages are sent and received from the module using a UART Data Frame.

#### 4.3.1. API Frame Specifications

Two API modes are supported and both can be enabled using the AP (API Enable) command. Use the following AP parameter values to configure the module to operate in a particular mode:

- AP = 1: API Operation
- AP = 2: API Operation (with escaped characters)

##### API Operation (AP parameter = 1)

When this API mode is enabled (AP = 1), the UART data frame structure is defined as follows:

Figure 4-06. UART Data Frame Structure:



MSB = Most Significant Byte, LSB = Least Significant Byte

Any data received prior to the start delimiter is silently discarded. If the frame is not received correctly or if the checksum fails, the module will reply with a module status frame indicating the nature of the failure.

##### API Operation - with Escape Characters (AP parameter = 2)

When this API mode is enabled (AP = 2), the UART data frame structure is defined as follows:

Figure 4-07. UART Data Frame Structure - with escape control characters:



MSB = Most Significant Byte, LSB = Least Significant Byte

**Escape characters.** When sending or receiving a UART data frame, specific data values must be escaped (flagged) so they do not interfere with the data frame sequencing. To escape an interfering data byte, insert 0x7D and follow it with the byte to be escaped XOR'd with 0x20.

**Data bytes that need to be escaped:**

- 0x7E – Frame Delimiter
- 0x7D – Escape
- 0x11 – XON
- 0x13 – XOFF

**Example** - Raw UART Data Frame (before escaping interfering bytes):  
 0x7E 0x00 0x02 0x23 0x11 0xCB  
 0x11 needs to be escaped which results in the following frame:  
 0x7E 0x00 0x02 0x23 0x7D 0x31 0xCB

Note: In the above example, the length of the raw data (excluding the checksum) is 0x0002 and the checksum of the non-escaped data (excluding frame delimiter and length) is calculated as:  
 $0xFF - (0x23 + 0x11) = (0xFF - 0x34) = 0xCB$ .

**Checksum**

To test data integrity, a checksum is calculated and verified on non-escaped data.

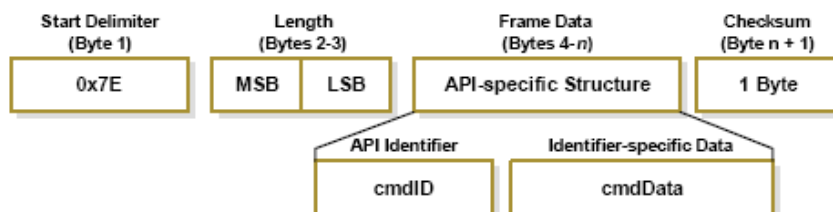
**To calculate:** Not including frame delimiters and length, add all bytes keeping only the lowest 8 bits of the result and subtract the result from 0xFF.

**To verify:** Add all bytes (include checksum, but not the delimiter and length). If the checksum is correct, the sum will equal 0xFF.

**4.3.2. API Types**

Frame data of the UART data frame forms an API-specific structure as follows:

Figure 4-08. UART Data Frame & API-specific Structure:



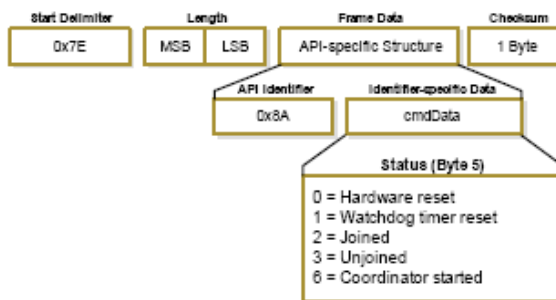
The cmdID frame (API-identifier) indicates which API messages will be contained in the cmdData frame (Identifier-specific data). Refer to the sections that follow for more information regarding the supported API types. Note that multi-byte values are sent big endian.

**Modem Status**

API Identifier: 0x8A

RF module status messages are sent from the module in response to specific conditions.

Figure 4-09. Modem Status Frames





## AT Command

API Identifier Value: 0x08

Allows for module parameter registers to be queried or set.

Figure 4-10. AT Command Frames

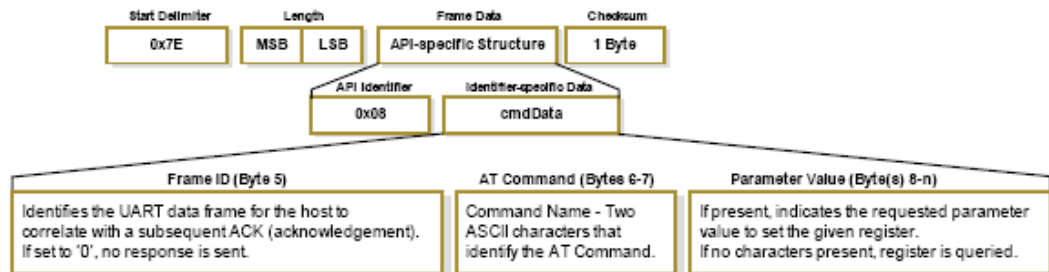


Figure 4-11. Example: API frames when reading the NJ parameter value of the module.



\* Length [Bytes] = API Identifier + Frame ID + AT Command

\*\* "R" value was arbitrarily selected.

Figure 4-12. Example: API frames when modifying the NJ parameter value of the module.



\* Length [Bytes] = API Identifier + Frame ID + AT Command + Parameter Value

\*\* "M" value was arbitrarily selected.

A string parameter used with the NI (Node Identifier), ND (Node Discover) and DH (Destination Address High) command is terminated with a 0x00 character.

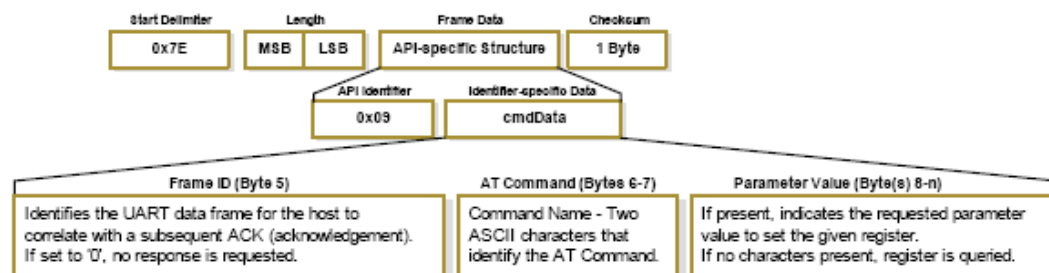
## AT Command - Queue Parameter Value

API Identifier Value: 0x09

This API type allows module parameters to be queried or set. In contrast to the "AT Command" API type, new parameter values are queued and not applied until either the "AT Command" (0x08) API type or the AC (Apply Changes) command is issued. Register queries (reading parameter values) are returned immediately.

Figure 4-13. AT Command Frames

(Note that frames are identical to the "AT Command" API type except for the API identifier.)

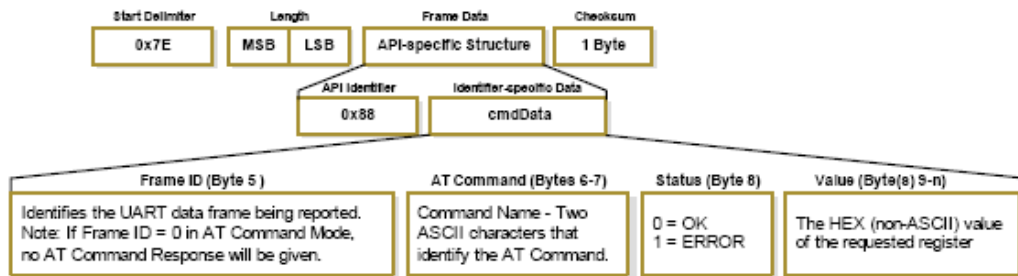


### AT Command Response

API Identifier Value: 0x88  
 Response to previous command.

In response to an AT Command message, the module will send an AT Command Response message. Some commands will send back multiple frames (for example, the ND (Node Discover) command). These commands will end by sending a frame with a status of ATCMD\_OK and no cmdData.

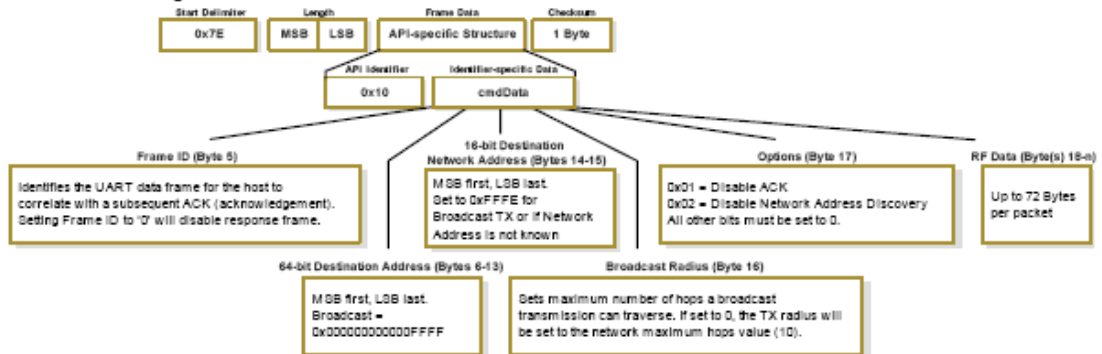
Figure 4-14. AT Command Response Frames.



### ZigBee Transmit Request

API Identifier Value: 0x10  
 A TX Request message will cause the module to send RF Data as an RF Packet.

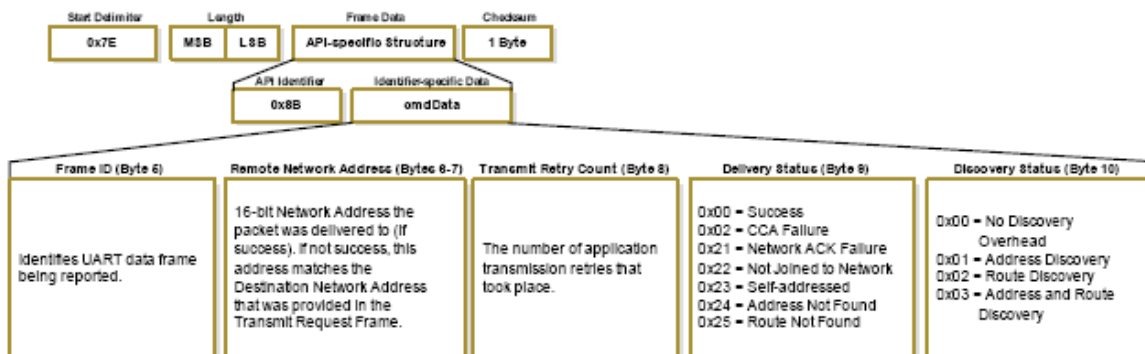
Figure 4-15. TX Packet Frames



### ZigBee Transmit Status

API Identifier Value: 0x8B  
 When a TX Request is completed, the module sends a TX Status message. This message will indicate if the packet was transmitted successfully or if there was a failure.

Figure 4-16. TX Status Frames

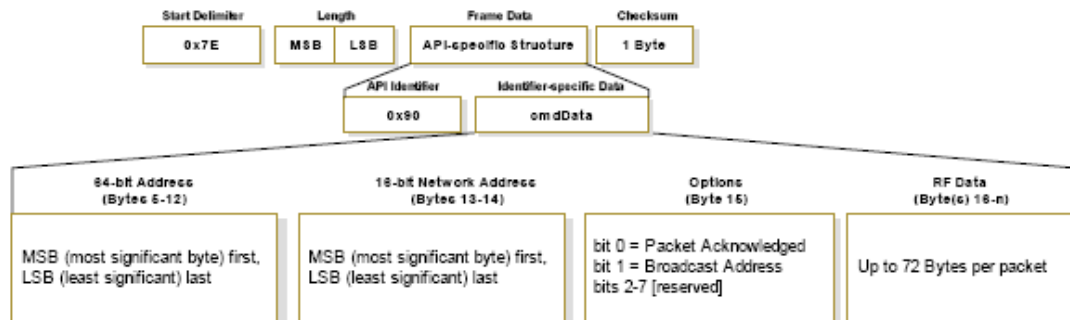


## ZigBee Receive Packet

API Identifier Value: 0x90

When the module receives an RF packet, it is sent out the UART using this message type.

Figure 4-17. RX Packet Frames



## Appendix A: Agency Certifications

### FCC Certification

XBee/XBee-PRO RF Modules comply with Part 15 of the FCC rules and regulations. Compliance with the labeling requirements, FCC notices and antenna usage guidelines is required.

To fulfill FCC Certification requirements, the OEM must comply with the following regulations:

1. The system integrator must ensure that the text on the external label provided with this device is placed on the outside of the final product [Figure A-01].
2. XBee/XBee-PRO RF Modules may only be used with antennas that have been tested and approved for use with this module [refer to the antenna tables in this section].

### OEM Labeling Requirements



**WARNING:** The Original Equipment Manufacturer (OEM) must ensure that FCC labeling requirements are met. This includes a clearly visible label on the outside of the final product enclosure that displays the contents shown in the figure below.

Figure A-01. Required FCC Label for OEM products containing the XBee/XBee-PRO RF Module

Contains FCC ID: OUR-XBEE/OUR-XBEEPRO\*

The enclosed device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (i.) this device may not cause harmful interference and (ii.) this device must accept any interference received, including interference that may cause undesired operation.

\* The FCC ID for the XBee is "OUR-XBEE". The FCC ID for the XBee-PRO is "OUR-XBEEPRO".

### FCC Notices

**IMPORTANT:** The XBee/XBee-PRO OEM RF Module has been certified by the FCC for use with other products without any further certification (as per FCC section 2.1091). Modifications not expressly approved by MaxStream could void the user's authority to operate the equipment.

**IMPORTANT:** OEMs must test final product to comply with unintentional radiators (FCC section 15.107 & 15.109) before declaring compliance of their final product to Part 15 of the FCC Rules.

**IMPORTANT:** The RF module has been certified for remote and base radio applications. If the module will be used for portable applications, the device must undergo SAR testing.

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures: Re-orient or relocate the receiving antenna, Increase the separation between the equipment and receiver, Connect equipment and receiver to outlets on different circuits, or Consult the dealer or an experienced radio/TV technician for help.

## FCC-Approved Antennas (2.4 GHz)

XBee/XBee-PRO RF Modules can be installed using antennas and cables constructed with standard connectors (Type-N, SMA, TNC, etc.) if the installation is performed professionally and according to FCC guidelines. For installations not performed by a professional, non-standard connectors (RPSMA, RPTNC, etc) must be used.

The modules are FCC-approved for fixed base station and mobile applications on channels 0x0B - 0x1A (XBee) and 0x0C - 0x17 (XBee-PRO). If the antenna is mounted at least 20cm (8 in.) from nearby persons, the application is considered a mobile application. Antennas not listed in the table must be tested to comply with FCC Section 15.203 (Unique Antenna Connectors) and Section 15.247 (Emissions).

**XBee OEM RF Modules (1 mW):** XBee Modules have been tested and approved for use with all of the antennas listed in the tables below (Cable-loss IS NOT required).

**XBee-PRO OEM RF Modules (60 mW):** XBee-PRO Modules have been tested and approved for use with the antennas listed in the tables below (Cable-loss IS required when using antennas listed in Table A-02).

Table A-01. Antennas approved for use with the XBee/XBee-PRO RF Modules (Cable-loss is not required.)

Part Number	Type (Description)	Gain	Application*	Min. Separation
A24-RSM-45U	Dipole (Half-wave articulated RPSMA - 4.5')	2.1 dBi	Fixed/Mobile	20 cm
A24-HABSMA	Dipole (Articulated RPSMA)	2.1 dBi	Fixed	20 cm
A24-HABUF-PSI	Dipole (Half-wave articulated bulkhead mount U.F.L. w/ 5' pigtail)	2.1 dBi	Fixed	20 cm
A24-QI	Monopole (Integrated whip)	1.5 dBi	Fixed	20 cm

Table A-02. Antennas approved for use with the XBee/XBee-PRO RF Modules (Cable-loss is required)

Part Number	Type (Description)	Gain	Application*	Min. Separation	Required Cable-loss
A24-C1	Surface Mount	-1.5 dBi	Fixed/Mobile	20 cm	-
A24-Y4NF	Yagi (4-element)	6.0 dBi	Fixed	2 m	8.1 dB
A24-Y6NF	Yagi (6-element)	8.8 dBi	Fixed	2 m	10.9 dB
A24-Y7NF	Yagi (7-element)	9.0 dBi	Fixed	2 m	11.1 dB
A24-Y9NF	Yagi (9-element)	10.0 dBi	Fixed	2 m	12.1 dB
A24-Y10NF	Yagi (10-element)	11.0 dBi	Fixed	2 m	13.1 dB
A24-Y12NF	Yagi (12-element)	12.0 dBi	Fixed	2 m	14.1 dB
A24-Y13NF	Yagi (13-element)	12.0 dBi	Fixed	2 m	14.1 dB
A24-Y15NF	Yagi (15-element)	12.5 dBi	Fixed	2 m	14.6 dB
A24-Y16NF	Yagi (16-element)	13.5 dBi	Fixed	2 m	15.6 dB
A24-Y16RM	Yagi (16-element, RPSMA connector)	13.5 dBi	Fixed	2 m	15.6 dB
A24-Y18NF	Yagi (18-element)	15.0 dBi	Fixed	2 m	17.1 dB
A24-F2NF	Omni-directional (Fiberglass base station)	2.1 dBi	Fixed/Mobile	20 cm	4.2 dB
A24-F3NF	Omni-directional (Fiberglass base station)	3.0 dBi	Fixed/Mobile	20 cm	5.1 dB
A24-F5NF	Omni-directional (Fiberglass base station)	5.0 dBi	Fixed/Mobile	20 cm	7.1 dB
A24-F8NF	Omni-directional (Fiberglass base station)	8.0 dBi	Fixed	2 m	10.1 dB
A24-F9NF	Omni-directional (Fiberglass base station)	9.5 dBi	Fixed	2 m	11.6 dB
A24-F10NF	Omni-directional (Fiberglass base station)	10.0 dBi	Fixed	2 m	12.1 dB
A24-F12NF	Omni-directional (Fiberglass base station)	12.0 dBi	Fixed	2 m	14.1 dB
A24-F15NF	Omni-directional (Fiberglass base station)	15.0 dBi	Fixed	2 m	17.1 dB
A24-W7NF	Omni-directional (Base station)	7.2 dBi	Fixed	2 m	9.3 dB
A24-M7NF	Omni-directional (Mag-mount base station)	7.2 dBi	Fixed	2 m	9.3 dB
A24-P8SF	Flat Panel	8.5 dBi	Fixed	2 m	8.6 dB
A24-P8NF	Flat Panel	8.5 dBi	Fixed	2 m	8.6 dB
A24-P13NF	Flat Panel	13.0 dBi	Fixed	2 m	13.1 dB
A24-P14NF	Flat Panel	14.0 dBi	Fixed	2 m	14.1 dB
A24-P15NF	Flat Panel	15.0 dBi	Fixed	2 m	15.1 dB
A24-P16NF	Flat Panel	16.0 dBi	Fixed	2 m	16.1 dB
A24-P19NF	Flat Panel	19.0 dBi	Fixed	2 m	19.1 dB

\* If using the RF module in a portable application (For example - If the module is used in a handheld device and the antenna is less than 20cm from the human body when the device is operation): The integrator is responsible for passing additional SAR (Specific Absorption Rate) testing based on FCC rules 2.1091 and FCC Guidelines for Human Exposure to Radio Frequency Electromagnetic Fields, OET Bulletin and Supplement C. The testing results will be submitted to the FCC for approval prior to selling the integrated unit. The required SAR testing measures emissions from the module and how they affect the person.

### RF Exposure



**WARNING:** To satisfy FCC RF exposure requirements for mobile transmitting devices, a separation distance of 20 cm or more should be maintained between the antenna of this device and persons during device operation. To ensure compliance, operations at closer than this distance is not recommended. The antenna used for this transmitter must not be co-located in conjunction with any other antenna or transmitter.

The preceding statement must be included as a CAUTION statement in OEM product manuals in order to alert users of FCC RF Exposure compliance.

## European Certification

---

The XBee/XBee-PRO RF Module has been certified for use in several European countries. For a complete list, refer to [www.maxstream.net](http://www.maxstream.net).

If the XBee/XBee-PRO RF Modules are incorporated into a product, the manufacturer must ensure compliance of the final product to the European harmonized EMC and low-voltage/safety standards. A Declaration of Conformity must be issued for each of these standards and kept on file as described in Annex II of the R&TTE Directive.

Furthermore, the manufacturer must maintain a copy of the XBee/XBee-PRO user manual documentation and ensure the final product does not exceed the specified power ratings, antenna specifications, and/or installation requirements as specified in the user manual. If any of these specifications are exceeded in the final product, a submission must be made to a notified body for compliance testing to all required standards.

### OEM Labeling Requirements

---

The 'CE' marking must be affixed to a visible location on the OEM product.

Figure A-02. CE Labeling Requirements



The CE mark shall consist of the initials "CE" taking the following form:

- If the CE marking is reduced or enlarged, the proportions given in the above graduated drawing must be respected.
- The CE marking must have a height of at least 5mm except where this is not possible on account of the nature of the apparatus.
- The CE marking must be affixed visibly, legibly, and indelibly.

### Restrictions

---

**Power Output:** The power output of the XBee-PRO RF Modules must not exceed 10 dBm. The power level is set using the PL command and the PL parameter must equal "0" (10 dBm).

**France:** France imposes restrictions on the 2.4 GHz band. Go to [www.art-telecom.fr](http://www.art-telecom.fr) or contact MaxStream for more information.

**Norway:** Norway prohibits operation near Ny-Alesund in Svalbard. More information can be found at the Norway Posts and Telecommunications site ([www.npt.no](http://www.npt.no)).

### Declarations of Conformity

---

MaxStream has issued Declarations of Conformity for the XBee/XBee-PRO RF Modules concerning emissions, EMC and safety. Files are located in the 'documentation' folder of the MaxStream CD.

#### Important Note

MaxStream does not list the entire set of standards that must be met for each country. MaxStream customers assume full responsibility for learning and meeting the required guidelines for each country in their distribution market. For more information relating to European compliance of an OEM product incorporating the XBee/XBee-PRO RF Module, contact MaxStream, or refer to the following web sites:

CEPT ERC 70-03E - Technical Requirements, European restrictions and general requirements: Available at [www.ero.dk/](http://www.ero.dk/).

R&TTE Directive - Equipment requirements, placement on market: Available at [www.ero.dk/](http://www.ero.dk/).

## Appendix B: Development Guide

---

### Migrating from the 802.15.4 Protocol

---

The following are some of the differences in the ZigBee firmware assuming familiarity with the 802.15.4 application:

- ZigBee Command Set
- Address Assignment
- API / AT Firmware Versions

Also, refer to the "Getting Started" section [p23] for more information.

#### ZigBee Command Set

---

##### Modified Commands

---

- CH - Read Only command that displays the operating channel that was selected from SC.
- MY - Read Only command that displays the assigned 16-bit Network Address of the device.
- AI - ZigBee definitions added to this command. See documentation.
- A1, A2 and CE commands are not supported.

##### New Commands

---

- NJ (Node Join Time) - This value determines how long a Coordinator or Router will allow other devices to join to it. This command is supported on Coordinators & Routers only.
- MP (16-bit Parent Network Address). This value represents the 16-bit parent Network Address of the module.
- BH (Broadcast Hops). This value sets the maximum number of hops for each broadcast data transmission. Setting this to 0 will use the maximum number of hops.

#### API / AT Firmware Versions

---

The 802.15.4 firmware supports the AP command for setting the module into No API (AP=0), API without escaping (AP=1), or API with escaping (AP=2) modes. The first digit in the 802.15.4 firmware versions is a '1'.

The ZigBee firmware comes in different versions to support the API interface (AP 1, 2 modes) or the AT command set (AP 0 mode). The first digit in the ZigBee firmware versions is an '8'.

The following is a list of firmware versions:

- 8.0xx - Coordinator, AT Command support (Transparent Mode)
- 8.1xx - Coordinator, API support (AP 1, 2)
- 8.2xx - Router, AT Command support (Transparent Mode)
- 8.3xx - Router, API support (AP 1, 2)
- 8.4xx - End Device, AT Command support (Transparent Mode)
- 8.5xx - End Device, API support (AP 1, 2)

## XBee/XBee-PRO Development Kits

The XBee Professional Development Kit includes the hardware and software needed to rapidly create long range wireless data links between nodes (XBee and XBee-PRO Starter Kits that contain fewer modules and accessories are also available).

**Table B-01. Items Included in the Development Kit**

Item	Qty.	Description	Part #
XBee-PRO Module	2	(1) OEM RF Module w/ U.FL antenna connector (1) OEM RF Module w/ attached wire antenna	XBP24-AUI-002 XBP24-AWI-002
XBee Module	3	(1) OEM RF Module w/ U.FL antenna connector (1) OEM RF Module w/ attached whip antenna (1) OEM RF Module w/ chip antenna	XB24-AUI-002 XB24-AWI-002 XB24-ACI-002
RS-232 Development Board	4	Board for interfacing between modules and RS-232 nodes (Converts signal levels, displays diagnostic info, & more)	XBIB-R
USB Development Board	1	Board for interfacing between modules & USB nodes (Converts signal levels, displays diagnostic info, & more)	XBIB-U
RS-232 Cable (6', straight-through)	1	Cable for connecting RS-232 interface board with DTE nodes (nodes that have a male serial DB-9 port - such as most PCs)	JD2D3-CDS-6F
USB Cable (6')	1	Cable for connecting USB interface board to USB nodes	JU1U2-CSB-6F
Serial Loopback Adapter	1	[Red] Adapter for configuring the module assembly (module + RS-232 interface board) to function as a repeater for range testing	JD2D3-CDL-A
NULL Modem Adapter (male-to-male)	1	[Black] Adapter for connecting the module assembly (module + RS-232 interface board) to other DCE (female DB-9) nodes	JD2D2-CDN-A
NULL Modem Adapter (female-to-female)	1	[Gray] Adapter for connecting serial nodes. It allows users to bypass the radios to verify serial cabling is functioning properly.	JD3D3-CDN-A
Power Adapter (9VDC, 1 A)	1	Adapter for powering the RS-232 development board	JP5P2-9V11-6F
Battery Clip (9V)	1	Clip for remotely powering the RS-232 board w/ a 9V battery	JP2P3-C2C-4I
RPSMA Antenna	2	RPSMA half-wave dipole antenna (2.4 GHz, 2.1 dB)	A24-HASM-450
RF Cable Assembly	2	Adapter for connecting RPSMA antenna to U.FL connector	JF1R6-CR3-4I
CD	1	Documentation and Software	MD0030
Quick Start Guide	1	Step-by-step instruction on how to create wireless links & test range capabilities of the modules	MD0026

### Interfacing Options

The development kit includes RS-232 and USB interface boards. The boards provide a connection to PC ports and therefore give access to the RF module registries. Parameters stored in the registry allow OEMs and integrators to customize the modules to suite the needs of their data radio systems.

The following sections illustrate how to use the interface boards for development purposes. The MaxStream Interface board provides means for connecting the module to any node that has an available RS-232 or USB connector. Since the module requires signals to enter at TTL voltages, one of the main functions of the interface board is to convert signals between TTL levels and RS-232 and USB levels.

---

Note: In the following sections, an OEM RF Module mounted to an interface board will be referred to as a "Module Assembly".

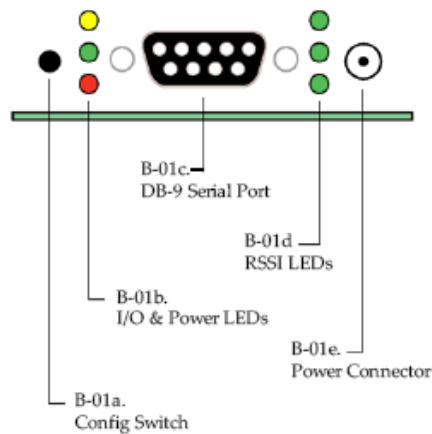
---



## RS-232 Development Board

### External Interface

Figure B-01. Front View



#### B-01a. Reset Switch

The Reset Switch is used to reset (re-boot) the RF module. This switch only applies when using the configuration tabs of MaxStream's X-CTU Software.

#### B-01b. I/O & Power LEDs

LEDs indicate RF module activity as follows:

Yellow (top LED) = Serial Data Out (to host)  
 Green (middle) = Serial Data In (from host)  
 Red (bottom) = Power/Association Indicator (Refer to the D5 (DIO5 Configuration) parameter)



#### B-01c. Serial Port

Standard female DB-9 (RS-232) connector.

#### B-01d. RSSI LEDs

RSSI LEDs indicate the amount of fade margin present in an active wireless link. Fade margin is defined as the difference between the incoming signal strength and the module's receiver sensitivity.

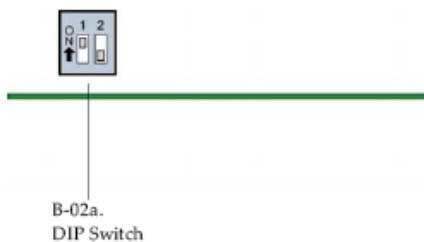
3 LEDs ON = Very Strong Signal (> 30 dB fade margin)  
 2 LEDs ON = Strong Signal (> 20 dB fade margin)  
 1 LED ON = Moderate Signal (> 10 dB fade margin)  
 0 LED ON = Weak Signal (< 10 dB fade margin)

#### B-01e. Power Connector

5-14 VDC power connector

#### B-02a. DIP Switch

Figure B-02. Back View



DIP Switch functions are not supported in this release. Future downloadable firmware versions will support DIP Switch configurations.

## RS-232 Pin Signals

Figure B-03. Pins used on the female RS-232 (DB-9) Serial Connector

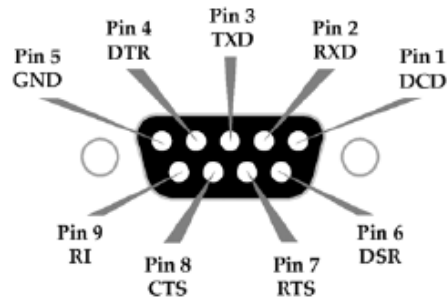


Table B-02. Pin Assignments and Implementations

DB-9 Pin	RS-232 Name	Description	Implementation*
1	DCD	Data-Carrier-Detect	Connected to DSR (pin6)
2	RXD	Receive Data	Serial data exiting the module assembly (to host)
3	TXD	Transmit Data	Serial data entering into the module assembly (from host)
4	DTR	Data-Terminal-Ready	Can enable Power-Down on the module assembly
5	GND	Ground Signal	Ground
6	DSR	Data-Set-Ready	Connected to DCD (pin1)
7	$\overline{\text{RTS}}$ / CMD	Request-to-Send / Command Mode	Provides $\overline{\text{RTS}}$ flow control or enables Command Mode
8	$\overline{\text{CTS}}$	Clear-to-Send	Provides $\overline{\text{CTS}}$ flow control
9	RI	Ring Indicator	Optional power input that is connected internally to the positive lead of the front power connector

\* Functions listed in the implementation column may not be available at the time of release.

**Wiring Diagrams**

Figure B-04. DTE node (RS-232, male DB-9 connector) wired to a DCE Module Assembly (female DB-9)

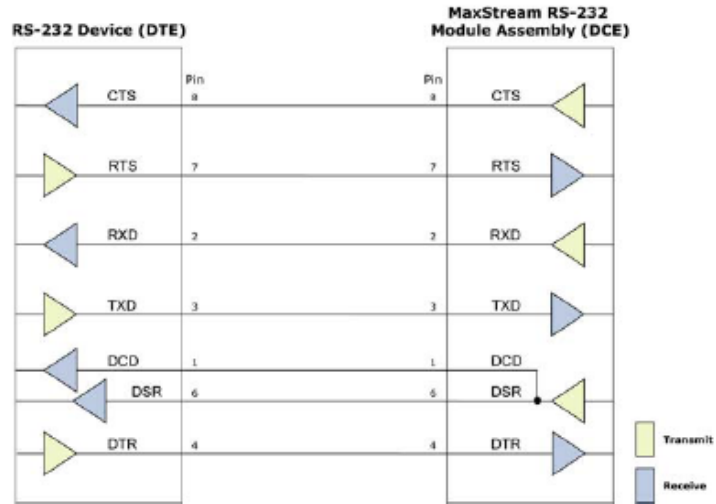
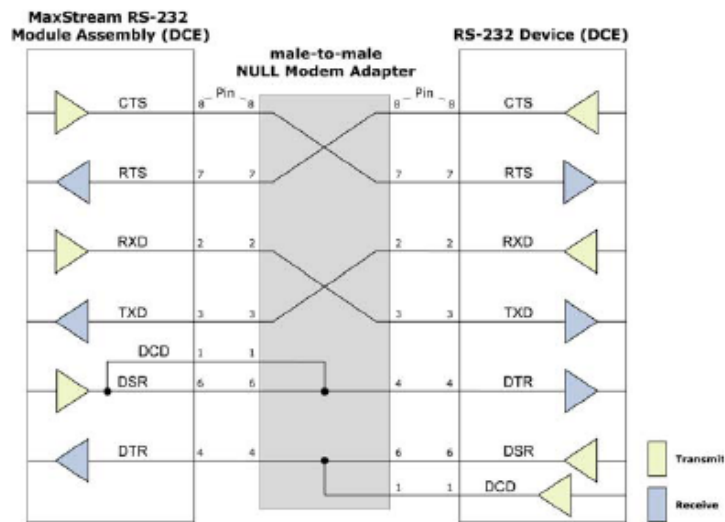
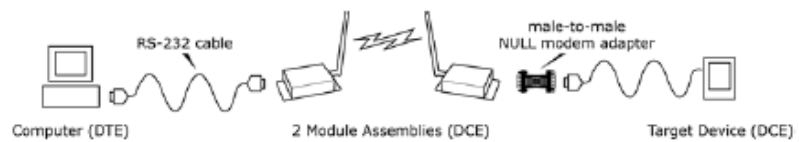


Figure B-05. DCE Module Assembly (female DB-9 connector) wired to a DCE node (RS-232, male DB-9)



**Sample Wireless Connection: DTE <--> DCE <--> DCE <--> DCE**

Figure B-06. Typical wireless link between DTE and DCE nodes



## Adapters

The development kit includes several adapters that support the following functions:

- Performing Range Tests
- Testing Cables
- Connecting to other RS-232 DCE and DTE nodes
- Connecting to terminal blocks or RJ-45 (for RS-485/422 nodes)

### NULL Modem Adapter (male-to-male)

**Part Number: JD2D2-CDN-A (Black, DB-9 M-M)** The male-to-male NULL modem adapter is used to connect two DCE nodes. A DCE node connects with a straight-through cable to the male serial port of a computer (DTE).

Figure B-07. Male NULL modem adapter and pinouts

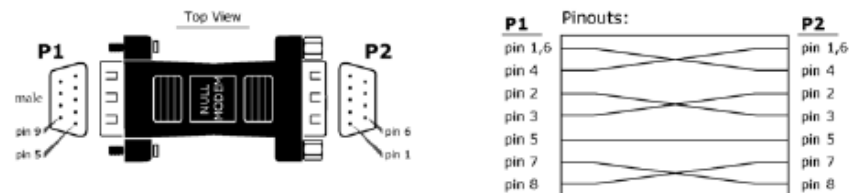
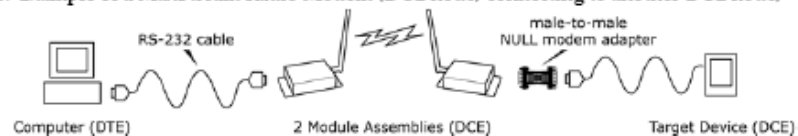


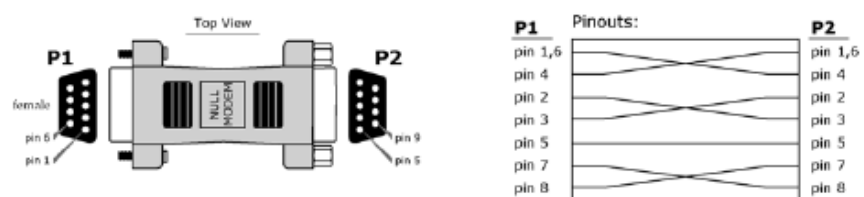
Figure B-08. Example of a MaxStream Radio Modem (DCE node) connecting to another DCE node



### NULL Modem Adapter (female-to-female)

**Part Number: JD3D3-CDN-A (Gray, DB-9 F-F)** The female-to-female NULL modem adapter is used to verify serial cabling is functioning properly. To test cables, insert the female-to-female NULL modem adapter in place of a pair of module assemblies (RS-232 interface board + XTend Module) and test the connection without the modules in the connection.

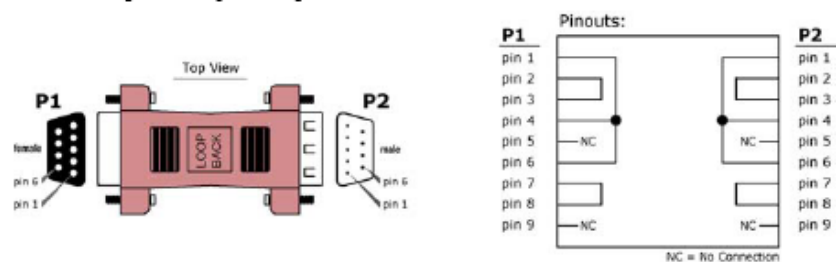
Figure B-09. Female NULL modem adapter and pinouts



### Serial Loopback Adapter

**Part Number: JD2D3-CDL-A (Red, DB-9 M-F)** The serial loopback adapter is used for range testing. During a range test, the serial loopback adapter configures the module to function as a repeater by looping serial data back into the radio for retransmission.

Figure B-10. Serial loopback adapter and pinouts

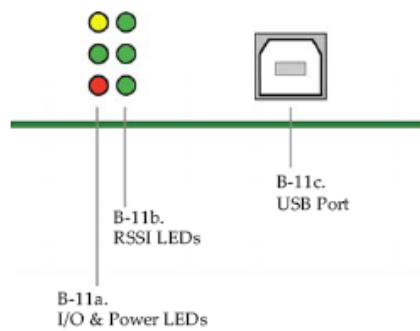


## USB Development Board

### External Interface

#### B-11a. I/O & Power LEDs

Figure B-11. Front View



LEDs indicate RF module activity as follows:

- Yellow (top LED) = Serial Data Out (to host)
- Green (middle) = Serial Data In (from host)
- Red (bottom) = Power/Association Indicator (Refer to the D5 (DIO5 Configuration) parameter)



#### B-11b. RSSI LEDs

RSSI LEDs indicate the amount of fade margin present in an active wireless link. Fade margin is defined as the difference between the incoming signal strength and the module's receiver sensitivity.

- 3 LEDs ON = Very Strong Signal (> 30 dB fade margin)
- 2 LEDs ON = Strong Signal (> 20 dB fade margin)
- 1 LED ON = Moderate Signal (> 10 dB fade margin)
- 0 LED ON = Weak Signal (< 10 dB fade margin)

#### B-11c. USB Port

Standard Type-B OEM connector is used to communicate with OEM host and power the RF module.

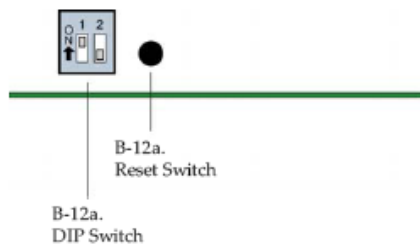
#### B-12a. DIP Switch

DIP Switch functions are not supported in this release. Future downloadable firmware versions will support the DIP Switch configurations.

#### B-12b. Reset Switch

The Reset Switch is used to reset (re-boot) the RF module.

Figure B-12. Back View



### USB Pin Signals

Table B-03. USB signals and their implantations on the XBee/XBee-PRO RF Module

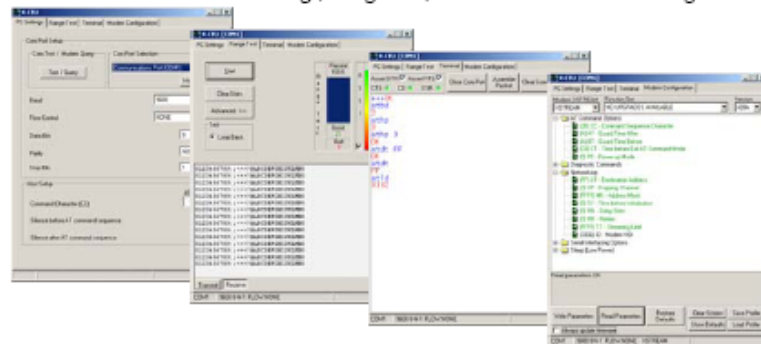
Pin	Name	Description	Implementation
1	VBUS	Power	Power the RF module
2	D-	Transmitted & Received Data	Transmit data to and from the RF module
3	D+	Transmitted & Received Data	Transmit data to and from the RF module
4	GND	Ground Signal	Ground

## X-CTU Software

X-CTU is a MaxStream-provided software program used to interface with and configure MaxStream RF Modules. The software application is organized into the following four tabs:

- PC Settings tab - Setup PC serial ports for interfacing with an RF module
- Range Test tab - Test the RF module's range and monitor packets sent and received
- Terminal tab - Set and read RF module parameters using AT Commands
- Modem Configuration tab - Set and read RF module parameters

Figure B-13. X-CTU User Interface (PC Settings, Range Test, Terminal and Modem Configuration tabs)



NOTE: PC Setting values are visible at the bottom of the Range Test, Terminal and Modem Configuration tabs. A shortcut for editing PC Setting values is available by clicking on any of the values.

## Install

Double-click the "setup\_X-CTU.exe" file and follow prompts of the installation screens. This file is located in the 'software' folder of the MaxStream CD and also under the 'Downloads' section of the following web page: [www.maxstream.net/support/downloads.php](http://www.maxstream.net/support/downloads.php)

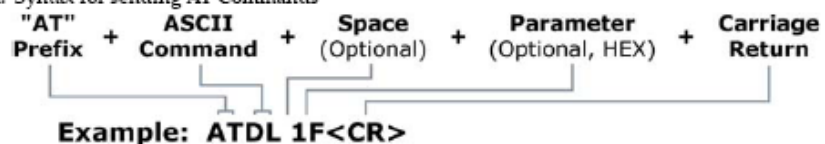
### Setup

To use the X-CTU software, a module assembly (An RF module mounted to an interface Board) must be connected to a serial port of a PC. The interface data rate and parity settings of the serial port ("PC Settings" tab) must match those of the module (BD (Baud Rate) and NB (Parity) parameters).

## Serial Communications Software

A terminal program is built into the X-CTU Software. Other terminal programs such as "HyperTerminal" can also be used. When issuing AT Commands through a terminal program interface, use the following syntax:

Figure B-14. Syntax for sending AT Commands



NOTE: To read a parameter value stored in a register, leave the parameter field blank.

The example above issues the DL (Destination Address Low) command to change destination address of the module to "0x1F". To save the new value to the module's non-volatile memory, issue WR (Write) command after modifying parameters.

## Appendix C: Additional Information

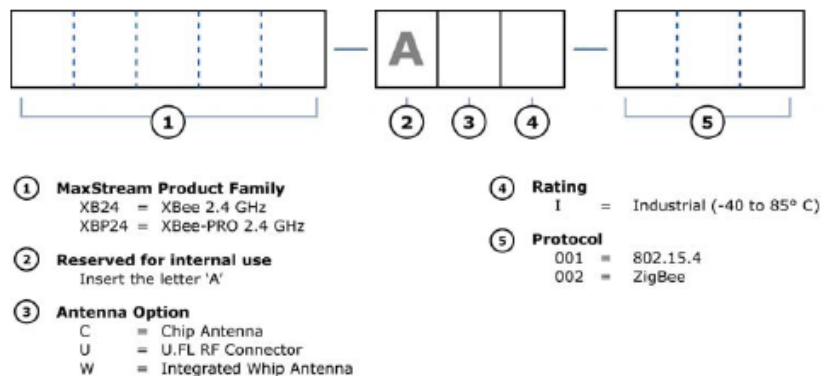
### 1-Year Warranty

XBee/XBee-PRO RF Modules from MaxStream, Inc. (the "Product") are warranted against defects in materials and workmanship under normal use, for a period of 1-year from the date of purchase. In the event of a product failure due to materials or workmanship, MaxStream will repair or replace the defective product. For warranty service, return the defective product to MaxStream, shipping prepaid, for prompt repair or replacement.

The foregoing sets forth the full extent of MaxStream's warranties regarding the Product. Repair or replacement at MaxStream's option is the exclusive remedy. THIS WARRANTY IS GIVEN IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, AND MAXSTREAM SPECIFICALLY DISCLAIMS ALL WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL MAXSTREAM, ITS SUPPLIERS OR LICENSORS BE LIABLE FOR DAMAGES IN EXCESS OF THE PURCHASE PRICE OF THE PRODUCT, FOR ANY LOSS OF USE, LOSS OF TIME, INCONVENIENCE, COMMERCIAL LOSS, LOST PROFITS OR SAVINGS, OR OTHER INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT, TO THE FULL EXTENT SUCH MAY BE DISCLAIMED BY LAW. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES. THEREFORE, THE FOREGOING EXCLUSIONS MAY NOT APPLY IN ALL CASES. This warranty provides specific legal rights. Other rights which vary from state to state may also apply.

### Ordering Information

Figure C-01. Divisions of the XBee/XBee-PRO RF Module Part Numbers



For example:

XB24-AWI-002 = XBee-PRO OEM RF Module, 2.4 GHz, integrated whip antenna, Industrial temperature rating, ZigBee protocol

## Contact MaxStream

---

Free and unlimited technical support is included with every MaxStream Radio Modem sold. For the best in wireless data solutions and support, please use the following resources:

Documentation:	<a href="http://www.maxstream.net/support/downloads.php">www.maxstream.net/support/downloads.php</a>	
Technical Support:	Phone.	(866) 765-9885 toll-free U.S.A. & Canada (801) 765-9885 Worldwide
	Live Chat.	<a href="http://www.maxstream.net">www.maxstream.net</a>
	E-Mail.	<a href="mailto:rf-xperts@maxstream.net">rf-xperts@maxstream.net</a>

MaxStream office hours are 8:00 am - 5:00 pm [U.S. Mountain Standard Time]