

**OTTO JACOB LITJENS**

**AUTOMAÇÃO DE ESTUFAS  
AGRÍCOLAS UTILIZANDO  
SENSORIAMENTO REMOTO E O  
PROTOCOLO ZIGBEE**

Trabalho de Conclusão de Curso apresentado  
à Escola de Engenharia de São Carlos, da  
Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase em  
eletrônica

**ORIENTADOR:** Evandro Luís Linhari Rodrigues

São Carlos

2009



## RESUMO

A tecnologia de rede sem-fio ou *wireless*, como é conhecida mundialmente, vem passando por um processo contínuo de padronização com o intuito de facilitar a integração de dispositivos fabricados por empresas distintas. Em 2004, o protocolo de rede sem-fio ZigBee foi criado para fornecer uma opção nas soluções de automação e seus benefícios foram imediatamente notados em diversas áreas. Uma empresa de dispositivos de rede americana, *Digi*, notou as vantagens desta tecnologia e desenvolveu um módulo de baixíssimo consumo, o XBee, com uma série de funcionalidades incluídas e capaz de criar redes mesh com mais de 65.000 pontos. O estudo deste módulo levou à conclusão de que é possível desenvolver sensores portáteis que podem operar por mais de cinco anos com pilhas comuns sem nenhuma intervenção humana. Alguns sistemas como a automação de estufas agrícolas, uma área em que sensores são abundantes para monitorar condições climáticas, poderia utilizar estes sensores para trazer mobilidade ao sistema e uma expressiva redução dos cabos de informação. Com este objetivo, foi desenvolvido um projeto para modernizar a automação destes ambientes através de módulos sem-fio XBee sendo utilizados para integrar sensores, painéis e softwares em uma única rede. O projeto contou com o desenvolvimento de dois sensores móveis, uma placa de comando central e um software além de utilizar um *gateway* para o computador. Desta maneira, permitiu que os sensores fossem realocados livremente dentro da área coberta pela rede, facilitou a detecção de erros e a integração na rede de novos dispositivos imbuídos com o módulo XBee que provaram ser confiáveis e uma excelente opção na prática da automação.

**Palavras-Chave:** Sensores sem-fio, automação de estufas agrícolas, redes sem-fio, rede ZigBee, otimização de sistemas, módulos XBee.



## ABSTRACT

The wireless networking technology has been undergoing a continuous process of standardization in order to facilitate the integration of devices manufactured by different companies. In 2004, the wireless protocol ZigBee was created to deliver an option to the automation solutions and its benefits was immediately noted in several areas. An american network company, *Digi*, saw the advantages of this technology and developed an extremely low power module, the XBee, with a series of functionalities included and capable of creating mesh networks with more than 65.000 points. The study of this module has led to the conclusion that it is possible to develop portable sensors that can operate for more than five years on standard batteries with no human intervention. Some systems such as agricultural greenhouses automation, an area where sensors are abundant to monitor weather conditions, could use these sensors to bring mobility to the system and a significant decrease in the information cables. With this goal, a project was developed to modernize the automation of these environments through wireless XBee modules being used to integrate sensors, panels and software on a single network. The project involved the development of two mobile sensors, a central control board and a software as well as the use of a gateway to the computer. Thus, allow the sensors to be relocated freely within the area covered by the network, has facilitated the detection of errors and network integration of new devices imbued with the XBee module that proved to be reliable and an excellent choice in the practice of automation.

**Key-Words:** Wireless sensors, agriculture greenhouse automation, wireless networks, ZigBee network, system optimizer, XBee modules.



## LISTA DE FIGURAS

Figura 1.1 – Sistema de Automação de estufas Agrícolas .....	2
Figura 1.2 - Comparativo entre as Redes Sem-Fio.....	3
Figura 1.3 - Relação entre SNR e BER de Tecnologias de Comunicação Sem-fio.....	4
Figura 2.1 - Funcionamento de uma Rede Mesh.....	9
Figura 2.2 - Arranjos de Redes Sem-fio .....	9
Figura 2.3 - Etapas do PAN Scan.....	10
Figura 2.4 - Métodos de Transmissão.....	11
Figura 2.5 - Dimensões do módulo XBee.....	12
Figura 2.6 - Tipos de Antenas Encontradas em Módulos XBee .....	14
Figura 2.7 - Espectro de Radiação das Antenas dos Módulos XBee.....	15
Figura 2.8 - Estrutura de um pacote API .....	17
Figura 2.9 - Pacote API 0x8A: Modem Status .....	19
Figura 2.10 - Pacote API 0x08: AT Command .....	20
Figura 2.11 - Pacote API 0x88: AT Command Response.....	20
Figura 2.12 - Pacote API 0x17: Remote Command Request .....	20
Figura 2.13 - Pacote API 0x97: Remote Command Response.....	21
Figura 2.14 - Pacote API 0x10: ZigBee Transmit Request .....	21
Figura 2.15 - Pacote API 0x8B: ZigBee Transmit Status .....	22
Figura 2.16 - Pacote API 0x90: ZigBee Receive Packet .....	22
Figura 2.17 - Pacote API 0x92: ZigBee IO Data Sample Rx Indicator .....	23
Figura 2.18 - Pacote API 0x95: Node Identification Indicator.....	24
Figura 2.19 - Configuração Típica para Diagnóstico da Rede .....	27
Tabela 2.6 - Funções Disponíveis para o Commissioning Pushbutton.....	27
Figura 3.1 - Arquitetura do Sistema de Automação Simulado neste Projeto .....	30
Figura 3.2 - Desenho 3D da placa de Circuito Impresso dos Sensores Remotos .....	30
Figura 3.3 - Vista Inferior e Circuito de Aplicação Típico para o LM35 .....	32
Figura 3.4 - Divisor de Tensão para o Sensor de Luz (LDR) .....	33
Figura 3.5 - Desenho 3D da Placa de Circuito Impresso do Controle Central.....	34
Figura 3.6 - Circuito Regulador de Tensão para o Comando Central.....	35
Figura 3.7 - Pinagem do PIC16F877A .....	37
Figura 3.8 - Circuito de Saída para Conexão de Equipamentos ao Controle Central.....	37
Figura 3.9 - Circuito de Comunicação Serial .....	38
Figura 3.10 - Fluxograma do Caminho Percorrido pelas Informações de Comando .....	38
Figura 3.11 - CON-USBEE da RogerCom .....	40
Figura 3.12 - Descrição dos LEDs do dispositivo CON-USBEE.....	40

Figura 3.13 - Diagrama de Blocos da Interpretação de Dados do Software .....	41
Figura 3.14 - Configuração de SetPoint Disponível no Software .....	42
Figura 3.15 - Painel do Sistema.....	42
Figura 3.16 - Painel Geral do software Monitor XBee .....	43
Figura 3.17 - Programa de Monitoração e Automação da rede ZNet .....	44
Figura 4.1 - Módulos XBee conectados a PAN do coordenador.....	45
Figura 4.2 - Interpretação dos pacotes API obtidas no software Monitor XBee.....	46

## LISTA DE TABELAS

Tabela 2.1 - Comparação entre Módulos XBee .....	13
Tabela 2.2 - Pinagem dos Módulos XBee Série 2 .....	14
Tabela 2.3 - Comparação entre Antenas para Módulos XBee .....	15
Tabela 2.4 - Tipos de Pacotes API .....	19
Tabela 2.5 - Estrutura de uma Amostra de Sinais das Portas do XBee .....	25
Tabela 3.1 – Parametrização do Módulo XBee para os Sensores Remotos .....	31
Tabela 3.2 - Parametrização do Módulo XBee do Painel de Comando.....	36
Tabela 3.3 - Comandos para Atuação dos Equipamentos.....	39
Tabela 4.1 - Configuração de SetPoints para o Teste do Sistema .....	45
Tabela 8.1 – Comandos AT: Especial .....	54
Tabela 8.2 - Comandos AT: Endereçamento .....	54
Tabela 8.3 - Comandos AT: Rede .....	55
Tabela 8.4 - Comandos AT: Segurança .....	57
Tabela 8.5 - Comandos AT: Interface RF.....	57
Tabela 8.6 - Comandos AT: Interface Serial.....	57
Tabela 8.7 - Comandos AT: Comandos I/O.....	58
Tabela 8.8 - Comandos AT: Diagnóstico.....	60
Tabela 8.9 - Comandos AT: Sleep Commands .....	60



## SUMÁRIO

<b>RESUMO .....</b>	<b>III</b>
<b>ABSTRACT .....</b>	<b>V</b>
<b>LISTA DE FIGURAS .....</b>	<b>VII</b>
<b>LISTA DE TABELAS.....</b>	<b>IX</b>
<b>1. INTRODUÇÃO.....</b>	<b>1</b>
1.1. JUSTIFICATIVA.....	1
1.2. OBJETIVO .....	5
1.3. ORGANIZAÇÃO DO TRABALHO .....	5
<b>2. ESTADO DA ARTE.....</b>	<b>7</b>
2.1. O PROTOCOLO ZIGBEE .....	7
2.1.1. Topologia ZigBee .....	7
2.1.2. A Formação da Rede ZigBee.....	9
2.1.3. Segurança .....	10
2.1.4. Endereçamento .....	11
2.1.5. Transmissão de Dados.....	11
2.2. O MÓDULO XBEE .....	12
2.2.1. Antenas.....	14
2.2.2. A Rede ZNet.....	15
2.2.3. Sleep Mode.....	16
2.2.4. Modos de operação .....	17
2.2.5. Pacotes API .....	17
2.2.6. Portas Analógicas e o Pacote 0x92 .....	25
2.2.7. Commissioning Pushbutton e Associate LED .....	26
<b>3. METODOLOGIA .....</b>	<b>29</b>
3.1. O SENSOR REMOTO .....	30
3.1.1. Fonte de Alimentação .....	30
3.1.2. Parametrização do XBee .....	31
3.1.3. Sensor de Temperatura.....	32
3.1.4. Sensor de Luminosidade .....	32
3.2. CONTROLE CENTRAL .....	34
3.2.1. Fonte de Alimentação .....	35
3.2.2. Parametrização do XBee .....	35
3.2.3. O Microcontrolador.....	36
3.3. O COORDENADOR .....	39
3.4. O MONITOR XBEE.....	40
3.4.1. Painel do Sistema .....	41
3.4.2. Painel Geral .....	43
<b>4. RESULTADOS .....</b>	<b>45</b>
<b>5. CONCLUSÃO.....</b>	<b>47</b>
<b>6. TRABALHOS FUTUROS.....</b>	<b>49</b>

<b>7. BIBLIOGRAFIA.....</b>	<b>51</b>
<b>8. ANEXOS .....</b>	<b>53</b>
8.1. LISTA DE MATERIAIS.....	53
8.1.1. Sensores Remotos .....	53
8.1.2. Controle Central.....	53
8.2. COMMANDOS AT DO MÓDULO XBEE/XBEE-PRO ZNET 2.5.....	54

# 1. INTRODUÇÃO

A idéia de cultivar plantas em ambientes controlados existe desde os tempos do império romano. Todos os dias, o imperador Tiberius comia ao menos um vegetal semelhante a um pepino. Seus jardineiros viram a necessidade de cultivar este vegetal de modo a torná-los disponíveis na mesa do imperador todos os dias do ano e utilizaram métodos artificiais, similares a uma estufa, para acelerar o crescimento e melhorar a qualidade da planta. Desde então, iniciou-se pesquisas para identificar os fatores aceleradores do crescimento das plantas e os primeiros conceitos de estufas começaram a aparecer.

A primeira estufa moderna surgiu na Itália durante o século XIII e o conceito se espalhou rapidamente para a Holanda e Inglaterra onde havia um forte interesse em proteger as plantas do inverno rigoroso. Pouco tempo depois, as estufas já eram encontradas nos quatro cantos do mundo e evoluíram junto com a tecnologia.

Hoje, as estufas agrícolas possuem equipamentos microprocessados de alta tecnologia e sensores de precisão para monitorar e controlar diversas variáveis ambientais e garantir um clima perfeito ao desenvolvimento dos vegetais produzidos.

O projeto apresentado visa aprimorar o controle de uma estufa por meio da tecnologia *wireless*, indicando as vantagens desta tecnologia e demonstrar possíveis evoluções que as estufas poderão sofrer num futuro próximo.

## 1.1.JUSTIFICATIVA

Embora a definição de uma estufa seja a de manter o calor em seu interior, uma estufa agrícola tem como objetivo, além do controle da temperatura, a criação de um micro-clima completo e favorável ao desenvolvimento do vegetal produzido. Este micro-clima é um conjunto de variáveis ambientais como temperatura, iluminação, umidade relativa, ventilação, nível de CO<sub>2</sub>, etc.

É sempre necessário identificar as condições climáticas ideais para um determinado tipo de vegetal. Algumas plantas não suportam temperaturas elevadas enquanto outras desenvolvem-se mais rapidamente nestas condições. Dessa maneira, a estufa é responsável por controlar o ambiente para produção de um determinado tipo de planta de modo a, independentemente das condições externas, acelerar o crescimento, garantir a qualidade e elevar os lucros do produtor.

A prevenção e tratamento de doenças e fungos também é um fator influenciado pelo clima. Um exemplo é o *Míldio dos Citrinos* [12], um fungo (*Phytophthora Hibernalis*) que se desenvolve em ambientes de temperaturas acima de 12°C e 70% de umidade relativa ou superior. Sem o devido tratamento, este fungo pode originar elevada desfolhação e queda de frutos causando prejuízos aos produtores despreparados.

Para impedir a propagação destas doenças e garantir o clima ideal ao desenvolvimento da planta, uma estufa moderna precisa ser equipada com sensores que monitoram variáveis ambientais e equipamentos que alteram condições climáticas. Alguns equipamentos encontrados no mercado para controlar o clima são:

- Aquecedores
- Caldeiras
- Ventiladores e exaustores
- Computadores de clima
- Controladores de CO2
- Sombreamento
- Sistemas de irrigação

Para integrar os dados colhidos pelos sensores e guiar o funcionamento dos equipamentos de climatização, costuma-se utilizar um painel de comando no interior da estufa, semelhante à caixa de disjuntores de uma casa, que centraliza os sinais de comando e as alimentações dos dispositivos da estufa.

Em instalações mais modernas o painel compartilha informações recebidas dos sensores e equipamentos, com um PC ou dispositivo semelhante, localizado no escritório do produtor que costuma ser próximo a estufa. Assim, programas de computador processam estas informações disponibilizando-as ao usuário em gráficos e mantendo o histórico em um banco de dados para futuras consultas.

A automação individual de cada equipamento também pode ser feita remotamente no escritório – ou, em alguns casos, pela internet - permitindo o acesso às informações da estufa em qualquer PC conectado à rede mundial de computadores desde que o usuário tenha permissão para tal.

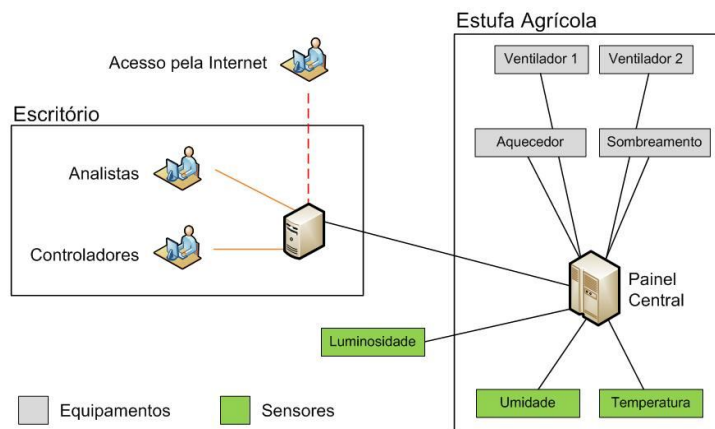


Figura 1.1 – Sistema de Automação de estufas Agrícolas

Um inconveniente neste sistema de controle é a presença de cabos elétricos como meio de propagação das informações. A substituição destes cabos por um sistema sem-fio pode reduzir o número de cabos de comunicação na estufa significativamente, facilitando tanto a instalação quanto a manutenção de equipamentos na estufa.

As opções de tecnologias sem-fio disponíveis no mercado que poderiam ser utilizadas na substituição dos cabos são principalmente o ZigBee, o Bluetooth e o Wi-Fi. Estas normas possuem características muito diferentes entre si, pois foram desenvolvidas com base em propósitos distintos como mostra a Figura 1.2 que também compara a taxa de transmissão e o alcance destas tecnologias.

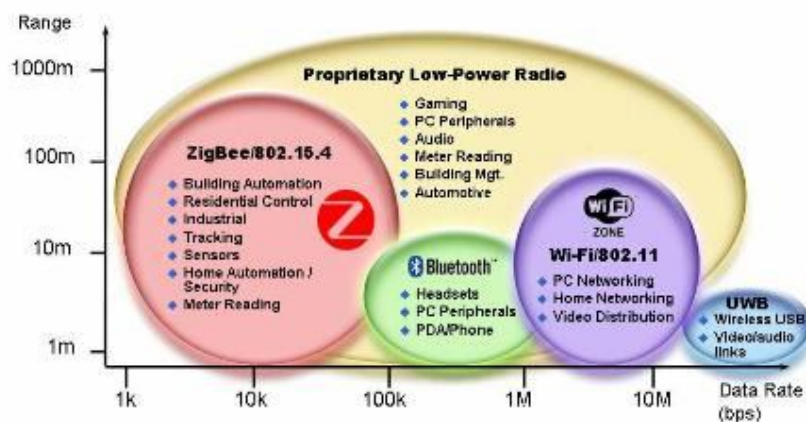


Figura 1.2 - Comparativo entre as Redes Sem-Fio

Fonte: [www.wirelessnetdesignline.com](http://www.wirelessnetdesignline.com)

Em um ambiente de automação, dificilmente encontra-se a necessidade de uma alta taxa de transmissão e o alcance pode ser um gargalo dependendo do local. Em estufas agrícolas, a distância média entre o centro da estufa (onde normalmente é localizado o painel central) e o canto mais distante fica entre 30 a 100 metros. Portanto, tanto o Bluetooth quanto o UWB não são adequados para o projeto.

Outro detalhe importante para a escolha da tecnologia wireless é a confiabilidade do sinal. Estufas agrícolas costumam ser ambientes ruidosos devido ao grande número de metais e equipamentos elétricos, portanto o sinal de RF (Rádio Freqüência) precisa possuir uma baixa taxa de erros em ambientes ruidosos. A Figura 1.3 foi disponibilizada por Jon Adams, presidente do grupo de qualificação da *ZigBee Alliance*, e ilustra a relação sinal-ruído e taxa de erros das tecnologias sem-fio.

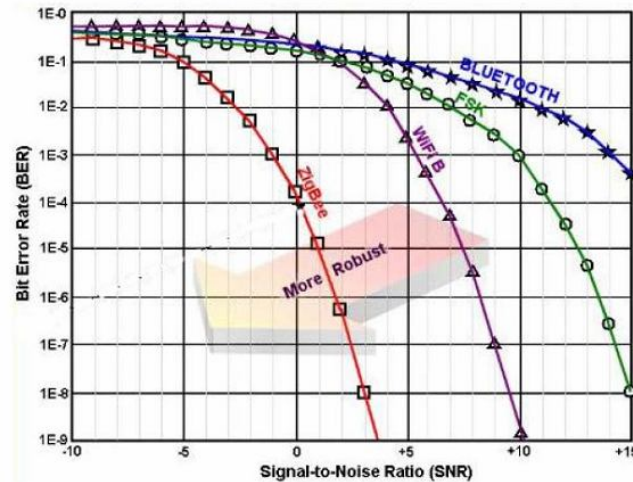


Figura 1.3 - Relação entre SNR e BER de Tecnologias de Comunicação Sem-fio

Fonte: ZigBee Alliance

O padrão ZigBee se destaca neste ponto apresentando a menor taxa de erros em ambientes ruidosos e além disso, possui outras vantagens que o diferem das demais tecnologias. A principal diferença está no consumo de energia.

O ZigBee foi desenvolvido para ser implementado em sistemas de automação onde sensores e atuadores são frequentemente utilizados. Como estes dispositivos não costumam ser utilizados 100% do tempo, a norma ZigBee apresenta um recurso chamado *Sleep Mode* em que os dispositivos podem dormir e poupar energia. Isto leva à criação de sensores alimentados por baterias e a eliminação de qualquer fio ou cabo nestes aparelhos, tornando-os portáteis e facilmente realocados.

Para implementar esta tecnologia nos sensores e no projeto de automação de estufas agrícolas, é necessário utilizar um dispositivo capaz de trabalhar dentro do padrão ZigBee. No mercado, existem CIs (circuitos impressos) como o CC243X da *Chipcon* [14] que utilizam microcontroladores da família 8051 e são capazes de transmitir e receber dados na norma ZigBee. Porém, estes CIs precisam de componentes externos e uma programação detalhada para funcionarem conforme o desejado.

A alternativa é utilizar módulos que incluem todos os componentes necessários para possibilitar a transmissão em RF. Estes módulos possuem antenas incorporadas, circuitos de proteção, chipset, osciladores e demais componentes permitindo serem utilizados com praticidade. Dois exemplos destes módulos são o XBee da *Digi internacional* e o EasyBee da *Flexipanel*.

O EasyBee utiliza o chipset Chipcon CC2420 mas o fato de ser pouco conhecido no Brasil dificulta a obtenção do módulo para realização de testes. Contudo, o XBee não compartilha o mesmo problema. A *Digi Internacional* possui distribuidores oficiais no Brasil e a crescente popularização do módulo está aumentando a oferta e tornando os preços mais competitivos.

O módulo XBee destaca-se dos demais por apresentar uma série de funcionalidades que permitem serem utilizados em diversas aplicações. No capítulo 2.2 (O Módulo XBee) estão descritas as principais características do módulo escolhido para o projeto.

Assim, a utilização do XBee em um sistema de automação de estufas agrícolas permitiria a prática de sensoriamento remoto e automação sem-fio garantindo vantagens aos produtores como:

- Redução dos cabos centralizados no painel, uma vez que somente a alimentação dos equipamentos utilizaria cabos.
- Fácil mobilidade dos sensores uma vez que podem ser alimentados por pilhas comuns.
- Fácil integração na rede de novos equipamentos com módulos XBee.
- Manutenção facilitada devido à redução de cabos.
- Utilização de uma tecnologia recente com boas perspectivas no mercado futuro – O ZigBee.

## 1.2. OBJETIVO

Utilizar os módulos XBee para criar uma rede de automação de estufas agrícolas com sensores móveis, um dispositivo central e um software para monitoração e comando. Assim, espera-se diminuir os cabos de comunicação em uma estufa, facilitando a instalação, manutenção e suporte aos equipamentos presentes nela, além de propiciar um sistema automatizado capaz de gerar um micro-clima ideal para o desenvolvimento das plantas produzidas.

## 1.3. ORGANIZAÇÃO DO TRABALHO

O trabalho foi organizado da seguinte maneira:

- O capítulo 2 contém informações sobre o funcionamento do protocolo ZigBee e como o módulo XBee se relaciona com este protocolo, introduzindo suas principais funcionalidades e características.
- No capítulo 3 inicia-se a discussão sobre o projeto propriamente dito incluindo as etapas do projeto, softwares utilizados, desenvolvimento dos circuitos, etc.
- No capítulo 4 é descrito o teste do sistema de automação desenvolvido e os resultados obtidos.
- O capítulo 5 conclui o projeto com informações sobre a viabilidade e aplicação do sistema *wireless* criado em estufas agrícolas.
- O capítulo 6 cita futuras continuações propostas para o projeto.



## 2. ESTADO DA ARTE

### 2.1.O PROTOCOLO ZIGBEE

Desenvolvido pela *ZigBee Alliance* [15], o protocolo de comunicação ZigBee 1.0 foi ratificado no dia 14 de dezembro de 2004 para aplicações que requerem uma baixa transmissão de dados, vida longa para baterias e uma rede segura, atuando na faixa de frequência de 2.4GHz, baseado na norma IEEE 802.15.4.

Rapidamente o protocolo ganhou espaço na automação, tanto industrial quanto residencial, devido ao baixo consumo (excelente para equipamentos portáteis de comunicação) e a possibilidade de construir redes mesh, ou seja, redes em que a informação pode ser reenviada por cada dispositivo até alcançar o destino final.

O alcance de transmissão de um módulo numa rede *mesh* pode ser estendido com a adição de outros módulos que atuam como repetidores de sinal para a rede e assim, diferenciam o ZigBee de outros protocolos do gênero como o Wi-Fi e Bluetooth. Mais detalhes sobre a rede mesh serão abordados no capítulo seguinte – Topologia ZigBee.

A maioria das informações disponibilizadas neste capítulo foi obtida no manual do módulo XBee [2].

#### 2.1.1. TOPOLOGIA ZIGBEE

Cada dispositivo de comunicação RF que utiliza a tecnologia ZigBee pode ser de dois tipos: Full Function Device (FFD) ou Reduced Function Device (RFD) [1].

**FFD** – *Full Function Device (Dispositivos de Funções Completas)* - Os dispositivos FFD são mais complexos, consomem mais energia, e podem se comunicar com quaisquer membros da rede. Como o próprio nome já diz, possuem todas as funções que o protocolo ZigBee permite e por isso podem ser configurados como coordenador para gerenciar a rede.

**RFD** – *Reduced Function Device (Dispositivos de Funções Reduzidas)* - Já os dispositivos RFD somente podem se comunicar com dispositivos FFD e sempre são utilizados como terminais em uma topologia de rede ZigBee. São mais simples por não apresentarem todas as funções do protocolo e consomem menos energia.

Os dispositivos FFD podem ser configurados para operar como coordenador, roteador ou *end device*, enquanto os dispositivos RFD somente podem atuar como *end device*.

**Coordenador** - Somente configurado em um dispositivo FFD, é responsável pela formação da rede *mesh* e por isso faz o controle de associação e dissociação de dispositivos na rede. Conseqüentemente, necessita de mais processamento e memória para operar e é recomendável utilizar uma fonte de alimentação conectada à rede elétrica. Também pode fazer a ponte entre outras redes ZigBee.

**Roteador** - Assim como o coordenador, o roteador somente pode ser configurado em um dispositivo FFD. Possui a característica de formação de nós na rede e pode rotear pacotes de informações sem a necessidade de passar pelo coordenador. Atua principalmente como extensor da rede.

**End Device (Terminal)** - Pode ser configurado tanto em dispositivos FFD quanto RFD. Costuma ser utilizado em sensores e atuadores, pois consome menos energia que as outras configurações principalmente por poder ser parametrizado para dormir (*Sleep*).

Uma rede ZigBee pode possuir três tipos de arranjos: rede *star*, rede *tree* ou, como mencionando anteriormente, uma rede *mesh*.

**Rede Star (Estrela)** - O coordenador é o responsável por inicializar a rede e manter os dispositivos conectados na mesma. Todos os outros dispositivos comunicam-se apenas com o coordenador.

Se um dispositivo precisa enviar dados para outro dispositivo, ele envia o dado para o coordenador então este encaminha o dado para o destinatário.

**Rede Tree (Malha)** - O coordenador é o responsável por iniciar a rede e escolher alguns parâmetros chave da mesma, mas a rede não é concentrada no coordenador e ela pode ser estendida com roteadores.

Os terminais podem se conectar na rede através do coordenador ou através dos roteadores.

**Rede Mesh (Cluster Árvore)** - É similar a configuração da rede *Tree*, exceto que os FFD's podem enviar mensagens diretamente a outros FFD's. A vantagem desta rede é a robustez e a confiabilidade caso algum nó falhe. A Figura 2.1 ilustra como é feito o roteamento em uma rede mesh em caso de falha de algum nó.

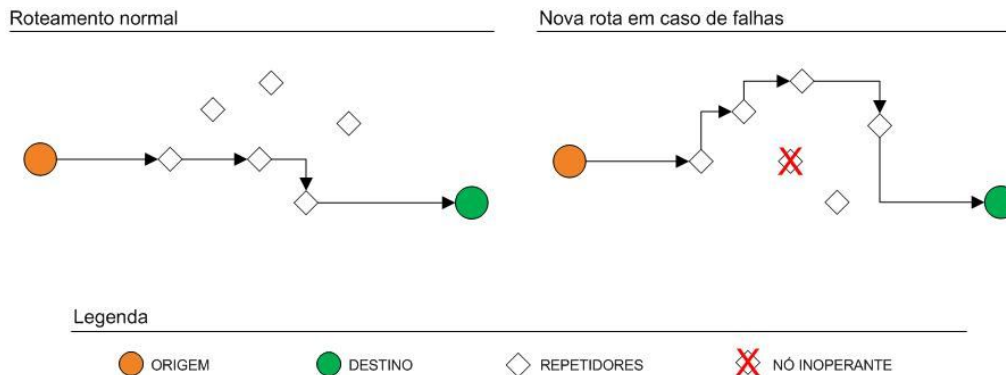


Figura 2.1 - Funcionamento de uma rede mesh

A Figura 2.2 obtida no site da *National Instruments*, ilustra os arranjos de uma rede ZigBee descritos anteriormente.

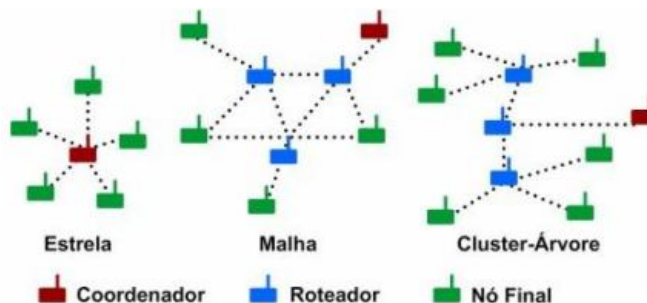


Figura 2.2 - Arranjos de Redes Sem-fio

Fonte: *National Instruments*

Outro detalhe importante da topologia de rede ZigBee é a definição de dispositivos **pais** e dispositivos **filhos**.

Um dispositivo **pai** (*Parent*) pode ser um roteador ou coordenador. Ele é responsável pela conexão do dispositivo em questão na rede. Todo o dispositivo, exceto o coordenador possui um módulo pai.

Os dispositivos **filhos** (*Children*) podem ser tanto roteadores como terminais. Cada dispositivo pode permitir a entrada de outros módulos (máximo 8) à rede. Desta maneira, se o módulo A é pai do módulo B, o módulo B será filho do módulo A.

### 2.1.2. A FORMAÇÃO DA REDE ZIGBEE

Redes ZigBee são redes de área pessoal conhecidas como PAN (*Personal Area Network*). Em cada PAN, é associado um número de 16 bits chamado PAN ID para identificar a rede e um número de 8 bits (1 byte) para o canal (*channel*) que determina a frequência de operação da rede. O

coordenador é responsável pelo início da rede ZigBee e por definir um PAN ID e canal para a rede. Para isto, o coordenador realiza o *Energy Scan* e em seguida o *PAN Scan*.

**Energy Scan** - Faz uma série de exames para descobrir o nível de atividade de RF em cada canal (frequência). Em seguida, os canais com menos atividade são armazenados em uma lista de possíveis canais.

**PAN Scan** - Testa a comunicação com outros coordenadores e roteadores próximos para descobrir os canais e PAN IDs utilizados por estes. A Figura 2.3 ilustra este processo.

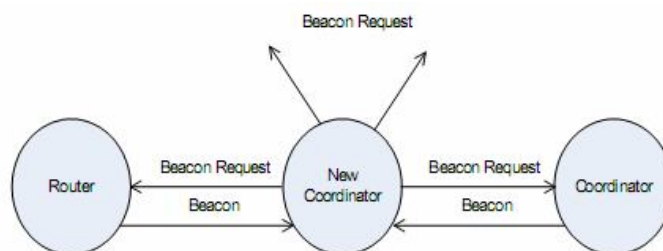


Figura 2.3 - Etapas do PAN Scan

Fonte: Manual XBee ZNet 2.5/XBee-PRO ZNet 2.5 OEM RF Modules, 2008

Assim que o *Energy Scan* e *PAN Scan* forem finalizados, o coordenador cria a rede a partir de um canal e PAN ID que não estão sendo utilizados pelos dispositivos próximos a ele.

Para se juntar a rede, roteadores e terminais precisam descobrir uma PAN ativa. Para isto, eles realizam um *PAN Scan* para listar todas as PANs em que podem participar.

Assim que um dispositivo (roteador ou terminal) encontrar uma PAN ID válida, que permite novas inclusões, ele envia um pedido de associação para aquele dispositivo da rede encontrada. Estes dispositivos podem ser configurados para participar de qualquer rede ZigBee disponível ou somente aquela que possui um PAN ID específico.

Todos os coordenadores e roteadores podem permitir novas inclusões na rede desde que dois itens sejam cumpridos:

1. O dispositivo está configurado para aceitar novas inclusões através do atributo NJ.
2. O dispositivo possui menos de 8 filhos.

### 2.1.3. SEGURANÇA

A rede ZigBee também possui a opção de habilitar uma chave de encriptação de 128 bits. Caso estiver ativada, somente dispositivos com a mesma chave de segurança podem se comunicar na

PAN. Roteadores e terminais que entrarão na rede precisam obter a chave de segurança correta que pode ser obtida de duas maneiras:

- Pré-instalação.
- Chave recebida por mensagem de RF (não codificada) durante a inclusão.

#### 2.1.4. ENDEREÇAMENTO

Cada dispositivo da PAN possui dois endereços. O endereço de rede que possui 16 bits e o endereço permanente, que é embutido em cada módulo durante a fabricação, e possui 64 bits.

O endereço de 16 bits é atribuído ao módulo quando este entra na rede. Ele é único para cada módulo da rede, porém pode ser alterado se o módulo sair e entrar novamente na rede.

#### 2.1.5. TRANSMISSÃO DE DADOS

Toda mensagem pode ser enviada para todos os dispositivos da rede por meio de uma *broadcast transmission* ou enviado a um destino específico através de uma *unicast transmission*.

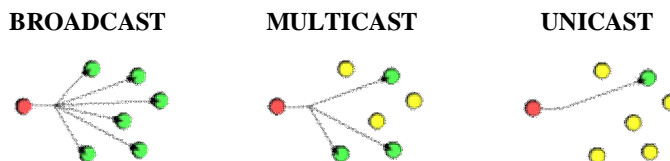


Figura 2.4 - Métodos de Transmissão

Fonte: <http://www.trainingsignaltraining.com/cisco-ip-multicasting/2008-04-28/>

*Broadcast transmissions* no protocolo ZigBee possuem a intenção de propagar por toda a rede de modo que todos os dispositivos recebam a transmissão. Para isto, todos os dispositivos que receberem uma *broadcast transmission* irão retransmitir esta mensagem 3 vezes.

Para impedir que esta mensagem circule na rede eternamente, cada dispositivo criará uma nota numa tabela para acompanhar qual mensagem broadcast já foi recebida anteriormente. A tabela possui 8 notas e cada nota permanecerá na tabela por 8 segundos.

Para cada *broadcast transmission* os módulos precisam reservar espaço na memória (buffer) para salvar uma cópia da mensagem e retransmiti-la quando necessário. Assim, não é aconselhável que estas transmissões sejam feitas com frequência.

As *unicast transmissions* são sempre endereçadas ao endereço de 16 bits do módulo de destino. Porém, como somente o endereço de 64 bits é permanente, os módulos ZigBee realizam um

procedimento para descobrir o endereço de rede chamado *Network Address Discovery*. Assim que o endereço 16 bits do destino é descoberto, a transmissão *unicast* ocorre normalmente.

**Network Address Discovery** - O dispositivo iniciador da transmissão realiza uma *broadcast transmission* na rede com o endereço de 64 bits do módulo de destino. Cada dispositivo da rede compara este endereço de 64 bits com o seu e caso for o mesmo, envia uma mensagem de volta ao dispositivo de origem com seu endereço 16 bits para então permitir ao dispositivo inicial realizar a transmissão de dados com o módulo em questão.

## 2.2.O MÓDULO XBEE

“The XBee/XBee-PRO ZNet 2.5 OEM (formerly known as Series 2 and Series 2 PRO) 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.”

Digi Internacional, Inc

O módulo de comunicação XBee foi escolhido para realizar toda comunicação RF do projeto por se tratar de um dispositivo pequeno, de baixíssimo consumo e por apresentar diversas funcionalidades embarcadas. Suas dimensões estão descritas na Figura 2.5.

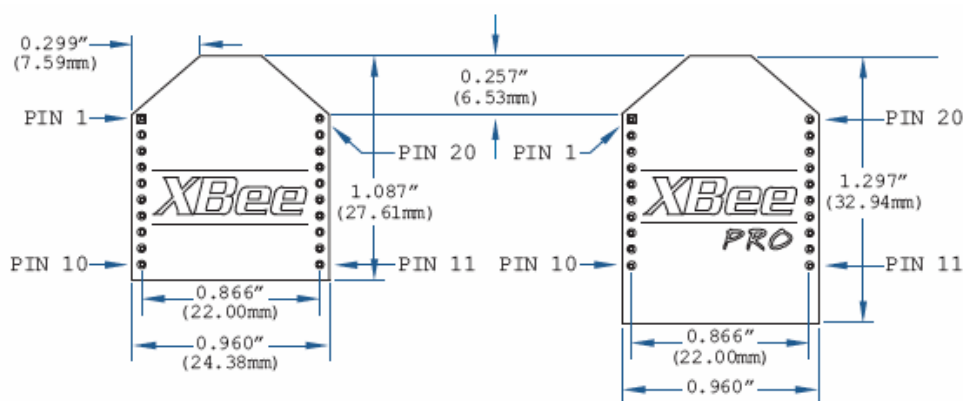


Figura 2.5 - Dimensões do módulo XBee

Fonte: Manual XBee ZNet 2.5/XBee-PRO ZNet 2.5 OEM RF Modules, 2008

Fabricado pela Digi Internacional, o módulo XBee possui diversos modelos que alteram a distância máxima de comunicação do dispositivo, a potência consumida, as opções de configuração, entre outros. Como exemplo, podemos citar o *XBee serie 2* que possui uma distância máxima de comunicação de 120 metros e o *XBee-PRO serie 2* que pode transmitir a até 1600 metros. Estes

módulos citados são intercambiáveis e por isso podem ser utilizados em uma mesma rede sem-fio que recebe o nome de rede ZNet.

A Tabela 2.1 descreve as principais características dos módulos XBee disponíveis.

Tabela 2.1 - Comparação entre Módulos XBee

Característica	<u>XBee</u> <u>ZB</u>	<u>XBee PRO</u> <u>ZB</u>	<u>XBee</u> <u>802.15.4</u>	<u>XBee</u> <u>PRO</u> <u>802.15.4</u>	<u>XBee/</u> <u>XBeePRO</u> <u>ZNET 2.5</u>
<b>Potência de Transmissão</b>	1,25 mW	50 mW	1 mW	63 mW	1,25 mW/ 50 mW
<b>Sensibilidade de Recepção</b>	-96 dBm	-102 dBm	-92 dBm	-100 dBm	-97dBm/ -102 dBm
<b>Alcance interno</b>	40 m	90 m	30 m	90 m	40/ 120 m
<b>Alcance máximo</b>	120 m	1,6 Km	90 m	1,6 Km	120 m / 1,6 Km
<b>Alimentação (VDC)</b>	2,1 a 3,6 V	3,0 a 3,4 V	2,8 a 3,4 V	2,8 a 3,4 V	2,1 a 3,6 V / 3,0 a 3,4 V
<b>Consumo máximo</b>	40 mA	295 mA	45 mA	215 mA	40 mA / 295 mA
<b>Power Down Current</b>	< 1 uA	< 10 uA	< 10 uA	< 10 uA	< 1 uA / <10 uA
<b>Topologia</b>	P-to-P, P-to-Multipoint, ZigBee/Mesh	P-to-P, P-to-Multipoint, ZigBee/Mesh	P-to-P, P-to-Multipoint, Peer-toPeer	P-to-P, P-to-Multipoint, Peer-toPeer	P-to-P, P-to-Multipoint, Mesh
<b>Conversor A/D</b>	4* (10 bits)	4* (10 bits)	6* (10 bits)	6* (10 bits)	4* (10 bits)
<b>Serial</b>	3.3 V CMOS UART *	3.3 V CMOS UART *	3.3 V CMOS UART *	3.3 V CMOS UART *	3.3 V CMOS UART *
<b>Tipos disponíveis de Antenas e Conectores</b>	Antena Wire Whip, Antena Chip, Conector UFL e Conector RPSMA.	Antena Wire Whip, Antena Chip, Conector UFL e Conector RPSMA.	Antena Wire Whip, Antena Chip, Conector UFL e Conector RPSMA.	Antena Wire Whip, Antena Chip, Conector UFL e Conector RPSMA.	Antena Wire Whip, Antena Chip, Conector UFL e Conector RPSMA.

Fonte: <http://www.embeddedworld.com.br/produtos.asp?produto=5000>

O modelo escolhido para o projeto é o *XBee ZNet 2.5 Serie 2* por preencher as necessidades de alcance de transmissão, possibilidade de redes *mesh*, e possuir o menor consumo da família resultando em uma maior eficiência da pilha que alimenta os sensores embutidos com os módulos RF.

A Tabela 2.2 descreve a pinagem dos módulos XBee utilizados.

Tabela 2.2 - Pinagem dos Módulos XBee Série 2

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DIO12	Either	Digital I/O 12
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI / DIO10	Either	PWM Output 0 / RX Signal Strength Indicator / Digital IO
7	PWM / DIO11	Either	Digital I/O 11
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DIO8	Either	Pin Sleep Control Line or Digital IO 8
10	GND	-	Ground
11	DIO4	Either	Digital IO 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP / DIO9	Output	Module Status Indicator or Digital IO 9
14	[reserved]	-	Do not connect
15	Associate / DIO5	Either	Associated Indicator, Digital I/O 5
16	RTS / DIO6	Either	Request-to-Send Flow Control, 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 / Commissioning Button	Either	Analog Input 0, Digital IO 0, or Commissioning Button

Fonte: Manual XBee ZNet 2.5/XBee-PRO ZNet 2.5 OEM RF Modules, 2008

### 2.2.1. ANTENAS

Nos dados do fabricante nota-se que é possível utilizar diversos tipos de antenas acopladas ao módulo, porém, neste projeto somente foi utilizado antenas do tipo fio (*Whip*) por uma questão de ser mais facilidade encontrada no mercado.

A Figura 2.6 ilustra os principais tipos de antenas disponíveis para o módulo XBee.

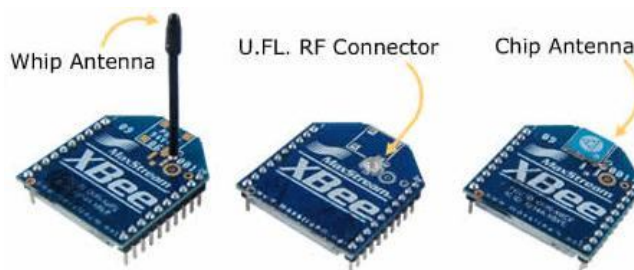


Figura 2.6 - Tipos de Antenas Encontradas em Módulos XBee

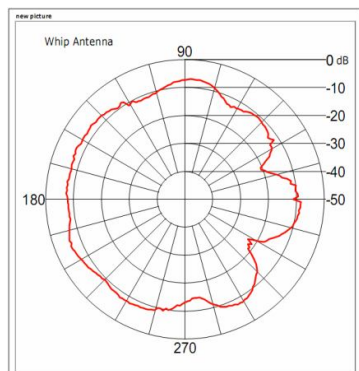
Fonte: Digi Internacional

As antenas apresentam diferentes ganhos e portanto é possível aumentar o alcance ou economizar energia optando por uma antena com um ganho maior. A *Digi* disponibiliza na internet um comparativo entre as antenas *Whip* e *Chip* (Tabela 2.3) e imagens do espectro de radiação das antenas de ambas as antenas conforme ilustrado na Figura 2.7.

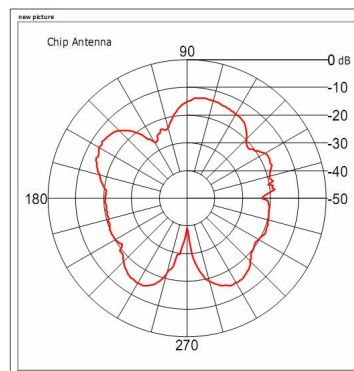
Tabela 2.3 - Comparação entre Antenas para Módulos XBee

Module	Antenna Type	Outdoor Distance (Visual Line-of-Sight)	Indoor Distance (Office Building)	Indoor Distance (Warehouse)
XBee	Chip	470 ft. (143 m)	80 ft. (24 m)	-
	Whip	845 ft. (258 m)	80 ft. (24 m)	84 ft. (26 m)
XBee-PRO	Chip	1690 ft. (515 m)	140 ft. (43 m)	-
	Whip	4382 ft. (1335 m)	140 ft. (43 m)	355 ft. (108 m)

Fonte: Digi Internacional



(a) Antena Whip



(b) Antena Chip

Figura 2.7 - Espectro de Radiação das Antenas dos Módulos XBee

Fonte: Digi Internacional

### 2.2.2. A REDE ZNET

Os módulos ZNet 2.5 utilizam um protocolo de comunicação muito parecido com o *ZigBee 2006 Standard*, mas incluem funções adicionais que são necessárias para manter uma rede mesh robusta.

Como numa rede *ZigBee*, o coordenador é responsável pela seleção do canal e do PAN ID da rede. Numa rede ZNet, estes fatores são influenciados pelos seguintes comandos:

**ID** – Seleciona uma PAN ID de preferência. Definindo o ID como 0xFFFF permitirá que qualquer PAN ID seja selecionada pelo coordenador.

**SC** – Seleciona uma lista de canais para serem examinadas no *Energy Scan* e *PAN Scan*.

**SD** – Determina a duração de tempo que cada um dos canais de SC será examinado pelo *Energy Scan* e *PAN Scan*.

Se o coordenador estiver operando no modo API (Capítulo 2.2.4), ele enviará o pacote API Modem Status “*coordinator started*” pela UART (*Universal Asynchronous Receiver/Transmitter*) do dispositivo assim que for inicializado. Em seguida permitirá que roteadores e terminais unam-se à rede.

Se o pino 15 do coordenador estiver configurado como *Associate LED*, a saída será nível lógico alto (1) enquanto a rede não for criada. Uma vez que o coordenador selecione o canal e o PAN ID, a saída do pino será alterada em um determinado tempo definido pelo comando LT. Se um LED for conectado a esse pino, o usuário poderá observar facilmente o momento em que o módulo entrou em uma PAN. Mais detalhes sobre o *Associate LED* poderão ser encontrados no capítulo 2.2.7.

### 2.2.3. SLEEP MODE

Os módulos XBee podem ser configurados para dormir em determinados momentos de modo que seu consumo seja reduzido significativamente, permitindo que seja alimentado por pilhas comuns durante um período prolongado.

Como descrito nas características do módulo XBee, seu consumo normal é próximo de 40mA, mas quando está dormindo seu consumo cai para menos de 1 $\mu$ A. Porém, neste estado o módulo é incapaz de realizar outras funções como transmitir dados RF e portanto somente terminais podem ser configurados para dormir.

Quando algum roteador recebe um pacote destinado a um terminal que esteja dormindo, ele armazena o pacote em seu buffer. Assim que o terminal acordar, ele envia uma mensagem ao seu pai informando que está disponível para transmissões RF. Em seguida o roteador envia o pacote armazenado em seu buffer para o terminal que está acordado e funcionando normalmente para realizar qualquer função suportada por ele. Após um determinado tempo, o terminal volta a dormir. Este tempo pode ser configurado pelo comando *ST (Time Before Sleep)*.

Existem duas maneiras para que um módulo acorde:

**Cyclic Sleep** – O dispositivo acorda em um intervalo de tempo definido pelo comando *SP (Sleep Period)* que pode variar de 320 milissegundos a 28 segundos.

**Pin Sleep** – O dispositivo acorda somente enquanto o pino 9 estiver em nível lógico 0.

Em ambas as maneiras de dormir, enquanto o módulo estiver acordado ele enviará a cada 100 ms, um pedido ao seu pai para verificar se existem informações destinadas a ele.

### 2.2.4. MODOS DE OPERAÇÃO

O módulo suporta duas interfaces de comunicação serial: Transparente ou API (Application Programming Interface).

**Modo Transparente:** Quando operando no modo transparente, o módulo atua como um substituto de uma linha de transmissão serial. Todos dados UART recebidos no pino de entrada (DIN) são empilhados para serem transmitidos via RF. Quando dados são recebidos via RF, estes são imediatamente enviados para o pino de saída serial (DOUT). Este modo é mais simples para se trabalhar, pois envolve menos dados de endereçamento e pouca preocupação com a interpretação de pacotes.

**Modo API:** O modo de operação API é uma alternativa ao modo Transparente. Neste, as informações são sempre transmitidas em pacotes que interagem com as funções de rede do módulo permitindo a este uma interação maior com a rede.

O modo API facilita muitas operações como os seguintes exemplos:

- Informações podem ser transmitidas para múltiplos destinos sem a necessidade de alterar parâmetros.
- Recebimento de sucesso/falha para cada pacote transmitido via RF.
- Identifica o endereço de origem de cada pacote recebido.

Embora este modo seja mais complexo para se trabalhar do que o modo Transparente, ele foi utilizado neste projeto para um melhor monitoramento da rede propriamente dita assim como a interface com o software.

### 2.2.5. PACOTES API

Um pacote API é constituído por 4 campos principais como ilustrado na Figura 2.8:

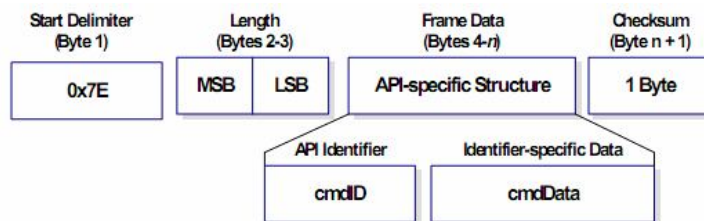


Figura 2.8 - Estrutura de um pacote API

Fonte: Manual XBee ZNet 2.5/XBee-PRO ZNet 2.5 OEM RF Modules, 2008

1. Start Delimiter (Delimitador Inicial)

Byte que indica o início de um pacote API

2. Lenght (Comprimento)

Campo constituído de 2 bytes que indicam o tamanho (em bytes) do campo de dados.

3. Frame Data (Pacote de Dados)

Possui as informações que se deseja transferir. É dividido em API Identifier (Identificador API) que indica qual mensagem API está contida no segundo campo e Identifier-specific Data (Identificador específico API), que possui os dados específicos para a mensagem.

4. Checksum

Campo utilizado para verificar a integridade da mensagem. É calculado através da subtração de 0xFF com os 8 bits menos significativos da soma de todos os bytes do campo Frame Data.

**Exemplo:**

<i>Start Delimiter</i>	<i>Lenght</i>	<i>Frame Data (5 bytes)</i>	<i>Checksum</i>
0x7E	0x00 0x05	0x08 0x01 0x4E 0x4A 0xFF	0x5F

0x7E → Start Delimiter.

0x0005 → Lenght (5 bytes do Frame Data).

0x08 → ID do pacote (API Identifier).

0x01 → Identificador do pacote (diferente de zero).

0x4E4A → Comando (“NJ” escrito em código ASCII).

0xFF → Valor do comando NJ

0x5F → Checksum (0xFF – (0x08+0x01+0x4E+0x4A+0xFF)).

Neste exemplo, o byte 0x08 indica que se trata de um pacote API “AT Command” que contém o comando NJ (*Node Join Time*) e possui 0xFF como parâmetro indicando, neste caso, que o módulo está apto a receber novos filhos independentemente do tempo após sua inclusão à rede.

A Tabela 2.4 descreve todas as opções de identificação de pacote API suportadas pelos módulos XBee.

Tabela 2.4 - Tipos de Pacotes API

Nome do pacote API	Valor
Modem Status	0x8A
AT Command	0x08
AT Command – Queue Parameter Value	0x09
AT Command Response	0x88
Remote Command Request	0x17
Remote Command Response	0x97
ZigBee Transmit Request	0x10
Explicit Addressing ZigBee Command Frame	0x11
ZigBee Transmit Status	0x8B
ZigBee Receive Packet (AO=0)	0x90
ZigBee Explicit Rx Indicator (AO=1)	0x91
ZigBee IO Data Sample	0x92
XBee Sensor Read Indicator (AO=0)	0x94
Node Identification Indicator	0x95

Fonte: Manual XBee ZNet 2.5/XBee-PRO ZNet 2.5 OEM RF Modules, 2008

Estes pacotes API possuem relações entre eles como, por exemplo, se enviar com sucesso um pacote 0x17 para um módulo remoto indicando pelo comando IR que envie as amostras das suas portas periodicamente, o módulo remoto responderá com um pacote 0x97 indicando sucesso na recepção do pacote 0x17 e em seguida enviará pacotes 0x92 periodicamente.

Cada pacote API possui sua própria estrutura de dados e deve ser analisado individualmente. Os tipos de pacotes utilizados neste projeto estão descritos a baixo. Todas as figuras utilizadas para visualizar a estrutura dos pacotes API foram obtidas no manual do módulo XBee [2]. Para demais informações sobre a estrutura dos tipos de pacote API consulte o manual do produto [2].

**0x8A – Modem Status** – Pacotes RF de status do dispositivo são enviados em determinadas condições.

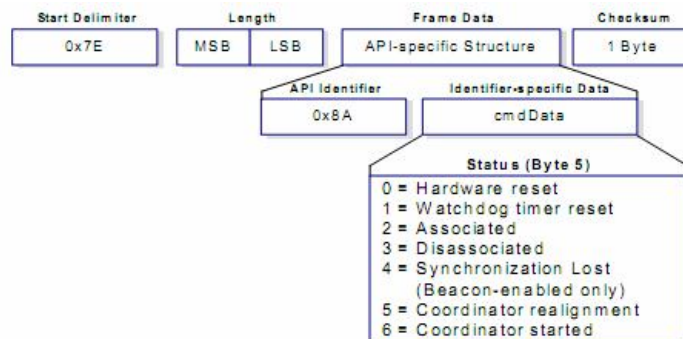


Figura 2.9 - Pacote API 0x8A: Modem Status

**0x08 – AT Command** – Permite que um parâmetro do módulo seja alterado ou enviado.

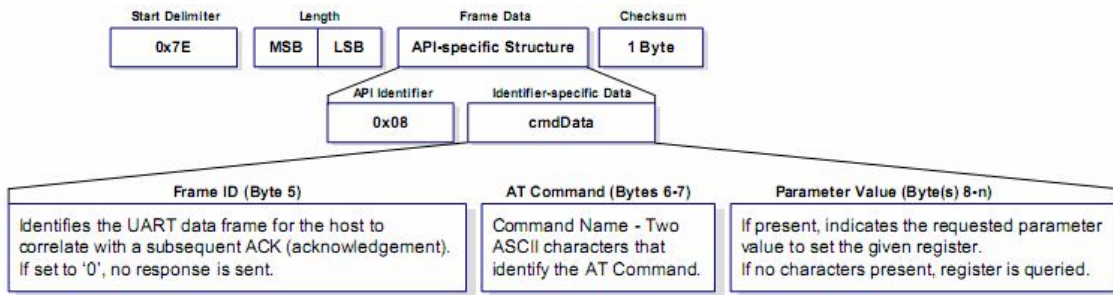


Figura 2.10 - Pacote API 0x08: AT Command

**0x88 – AT Command Response** – Resposta ao pacote anterior.

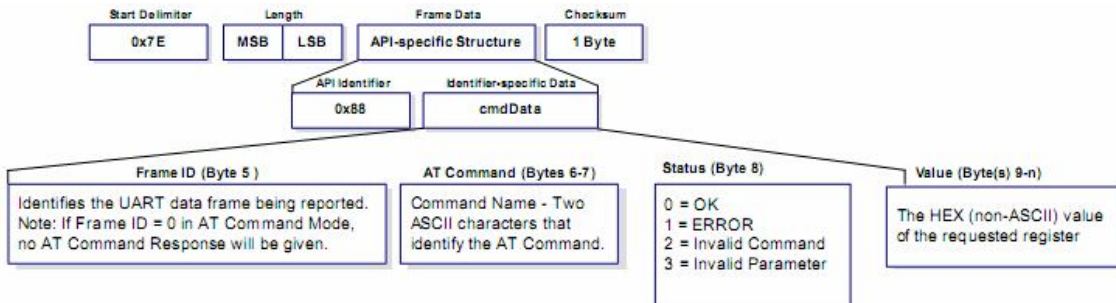


Figura 2.11 - Pacote API 0x88: AT Command Response

**0x17 – Remote Command Request** - Permite que um parâmetro de um módulo remoto seja alterado ou enviado por RF.

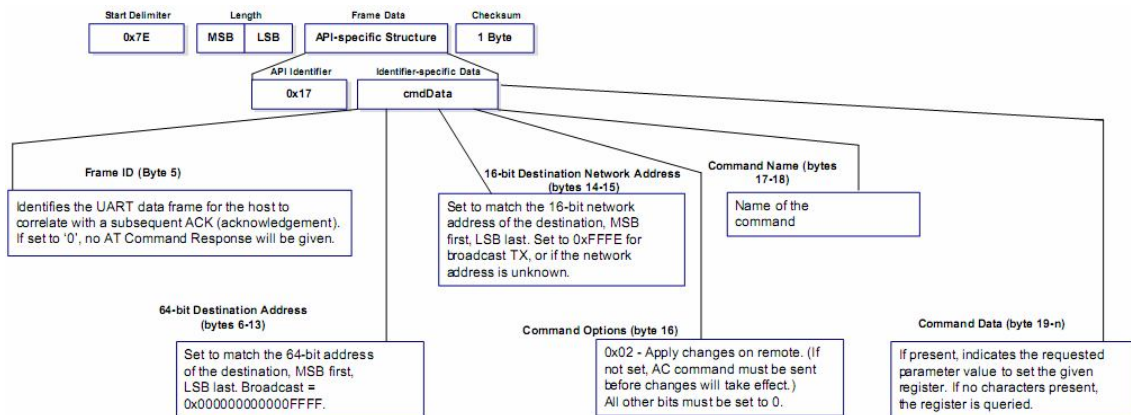


Figura 2.12 - Pacote API 0x17: Remote Command Request

**0x97 – Remote Command Response** – Resposta ao pacote anterior.

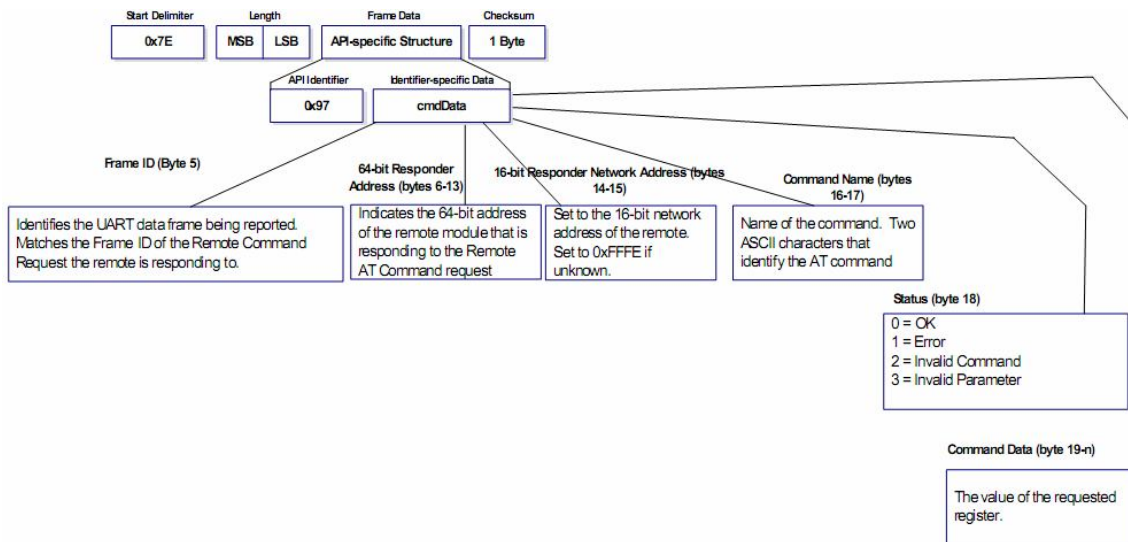


Figura 2.13 - Pacote API 0x97: Remote Command Response

**0x10 – ZigBee Transmit Request** – Pacote para enviar até 72 bytes de informação para ser transmitida serialmente no pino DOUT do módulo remoto.

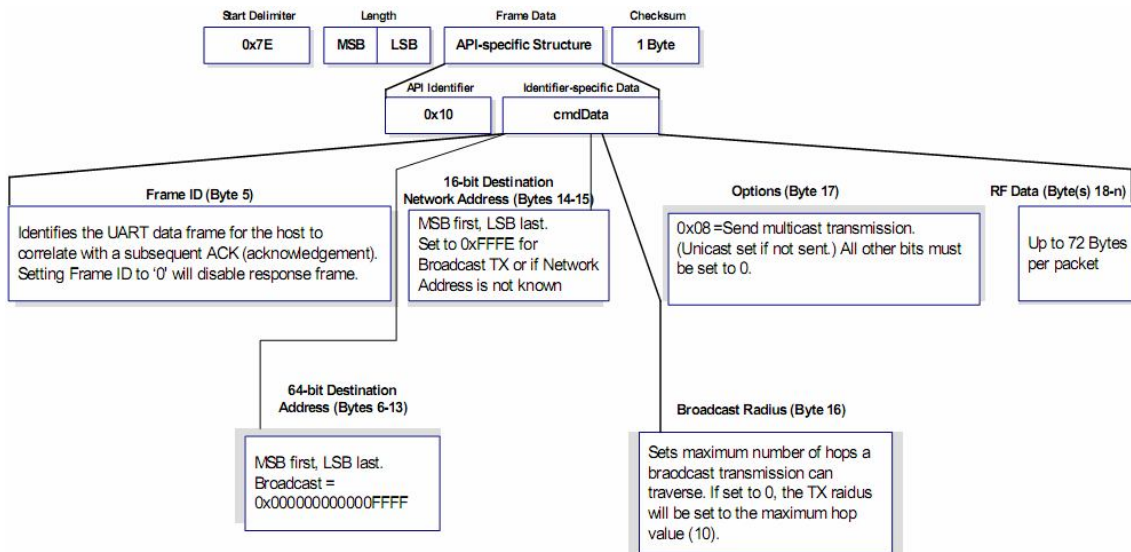


Figura 2.14 - Pacote API 0x10: ZigBee Transmit Request

**0x8B – ZigBee Transmit Status** - Resposta ao pacote anterior.

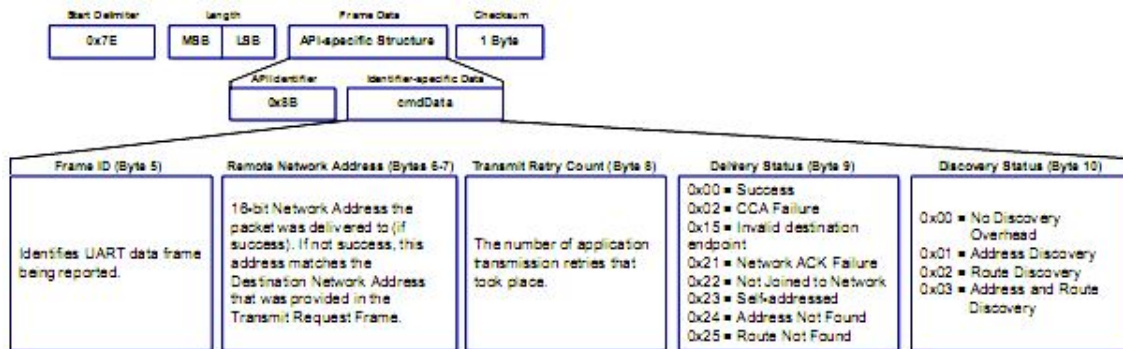


Figura 2.15 - Pacote API 0x8B: ZigBee Transmit Status

**0x90 – ZigBee Receive Packet** – Quando um módulo recebe um pacote 0x10 (ZigBee Transmit Request), ele envia um pacote 0x90 pela Uart com os dados do pacote recebido.

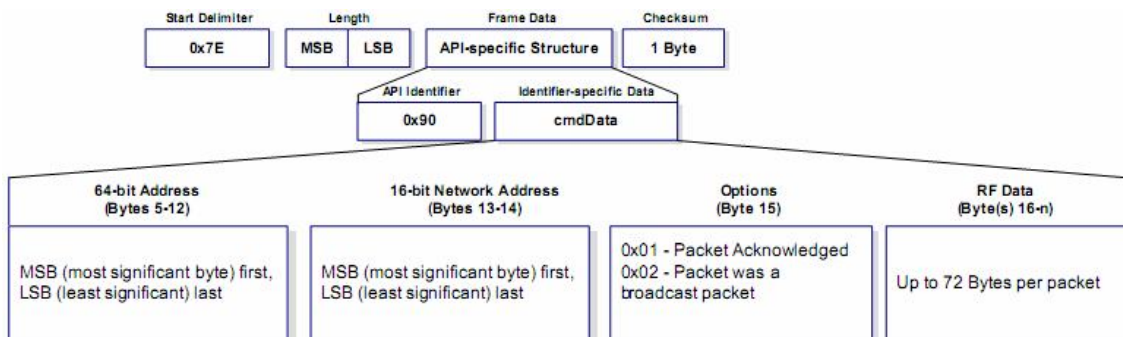


Figura 2.16 - Pacote API 0x90: ZigBee Receive Packet

**0x92 - ZigBee IO Data Sample Rx Indicator** – Quando o módulo recebe um pacote com amostras das portas IO, ele envia um pacote 0x92 pela UART com as informações do pacote recebido.

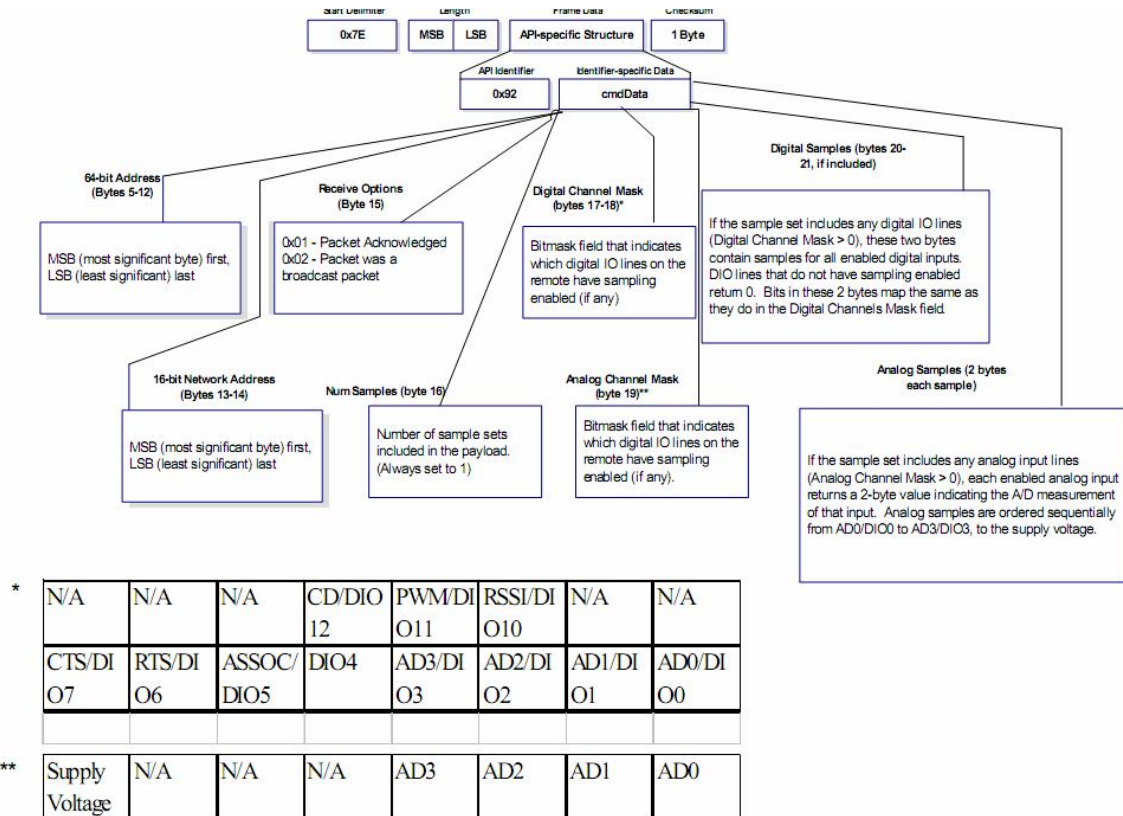


Figura 2.17 - Pacote API 0x92: ZigBee IO Data Sample Rx Indicator

**0x95 – Node Identification Indicator** – Este pacote é transmitido pela UART do coordenador quando um módulo emite uma mensagem para identificar seu nó ao coordenador.

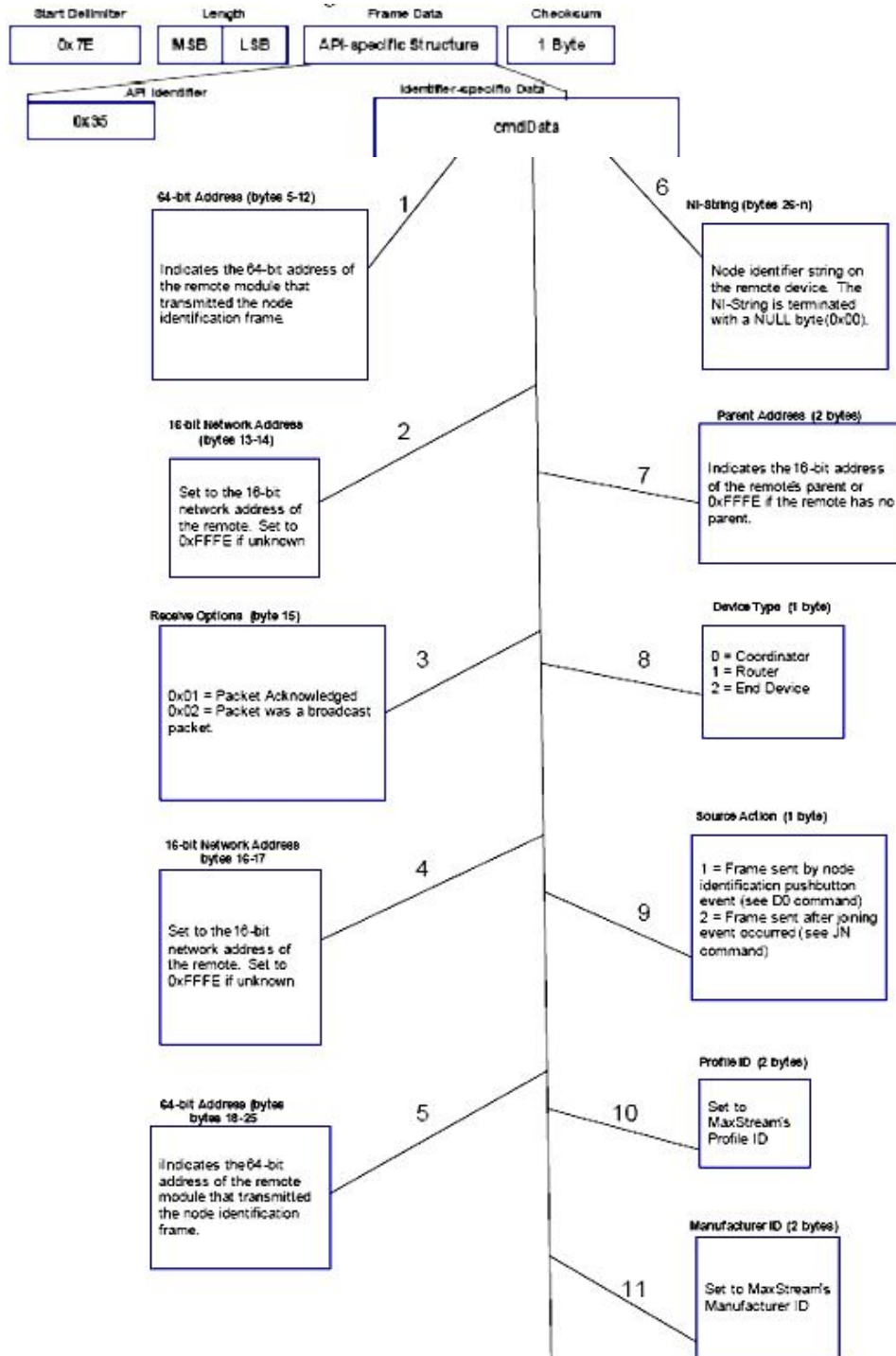


Figura 2.18 - Pacote API 0x95: Node Identification Indicator

### 2.2.6. PORTAS ANALÓGICAS E O PACOTE 0x92

O pacote API 0x92 é utilizado para informar o estado das portas digitais e analógicas de módulos remotos XBee, de acordo com a configuração de cada porta. Este pacote é essencial para o projeto, pois permite ao software interpretar a mensagem 0x92 para obter o valor da tensão presente nas portas analógicas do módulo e em seguida converter o valor da tensão em temperatura ou luminosidade de acordo com o dado recebido.

A Tabela 2.5 contém a tabela que descreve a estrutura do *frame data* para o pacote API 0x92.

Tabela 2.5 - Estrutura de uma Amostra de Sinais das Portas do XBee

Bytes	Name	Description
1	Sample Sets	Number of sample sets in the packet. (Always set to 1.)
2	Digital Channel Mask	Indicates which digital IO lines have sampling enabled. Each bit corresponds to one digital IO line on the module. <ul style="list-style-type: none"> <li>• bit 0 = AD0/DIO0</li> <li>• bit 1 = AD1/DIO1</li> <li>• bit 2 = AD2/DIO2</li> <li>• bit 3 = AD3/DIO3</li> <li>• bit 4 = DIO4</li> <li>• bit 5 = ASSOC/DIO5</li> <li>• bit 6 = RTS/DIO6</li> <li>• bit 7 = CTS/GPIO7</li> <li>• bit 8 = N/A</li> <li>• bit 9 = N/A</li> <li>• bit 10 = RSSI/DIO10</li> <li>• bit 11 = PWM/DIO11</li> <li>• bit 12 = CD/DIO12</li> </ul> For example, a digital channel mask of 0x002F means DIO0, 1, 2, 3, and 5 are enabled as digital IO.
1	Analog Channel Mask	Indicates which lines have analog inputs enabled for sampling. Each bit in the analog channel mask corresponds to one analog input channel. <ul style="list-style-type: none"> <li>• bit 0 = AD0/DIO0</li> <li>• bit 1 = AD1/DIO1</li> <li>• bit 2 = AD2/DIO2</li> <li>• bit 3 = AD3/DIO3</li> <li>• bit 7 = Supply Voltage</li> </ul>
Variable	Sampled Data Set	A sample set consisting of 1 sample for each enabled ADC and/or DIO channel. If any digital IO lines are enabled, the first two bytes of the data set indicate the state of all enabled digital IO. Only digital channels that are enabled in the Digital Channel Mask bytes have any meaning in the sample set. If no digital IO are enabled on the device, these 2 bytes will be omitted. Following the digital IO data (if any), each enabled analog channel will return 2 bytes. The data starts with AIN0 and continues sequentially for each enabled analog input channel up to AIN3, and the supply voltage (if enabled) at the end.

Fonte: Manual XBee ZNet 2.5/XBee-PRO ZNet 2.5 OEM RF Modules, 2008

Os sinais analógicos são processados em valores de 10 bits. A leitura destes sinais é feita de modo que 0x0000 represente 0V, e 0x3FF represente 1200mV.

**Exemplo:**

*Sample Set*   *Digital C. Mask*   *Analog C. Mask*   *Digital Values*   *Analogic Values*

0x01	0x0C0C	0x03	0x0408	0x03D0	0x0124
------	--------	------	--------	--------	--------

*0x01* → *Sample Set*;

*0x0C0C* → Entradas Digitais: DIO 2, 3, 10 e 11;

*0x03* → Entradas Analógicas: AD 0 e 1;

*0x0408* → Estados das Entradas Digitais: Nível alto: 3 e 10; Nível baixo: 2 e 11;

*0x03D0* → Entrada Analógica AD 0: 976mV

*0x0124* → Entrada Analógica AD 1: 292mV

Caso a tensão de alimentação do módulo esteja abaixo do valor configurado pelo comando  $V+$ , o número hexadecimal referente à tensão de alimentação será incluído neste pacote e o bit 7 da *Analog Channel Mask* terá valor 1.

### 2.2.7. COMMISSIONING PUSHBUTTON E ASSOCIATE LED

Os módulos XBee são embutidos com um sistema para diagnóstico que é constituído por dois recursos que podem funcionar em conjunto.

Um destes recursos chama-se *commissioning pushbutton* e o segundo, *Associated LED*. Na Tabela 2.2 descrita anteriormente, observa-se estes nomes nos pinos 20 e 15, respectivamente.

O *commissioning pushbutton* disponibiliza ao usuário algumas funções de diagnóstico da rede dependendo do número de vezes que o botão for apertado. Em resposta, o *Associated LED* poderá alterar seu comportamento padrão para informar ao usuário o estado de operação do módulo. A montagem típica para utilizar estes recursos é mostrada na Figura 2.19 - Configuração Típica para Diagnóstico da Rede e as funções disponíveis para o *commissioning pushbutton* são apresentadas na Tabela 2.6.

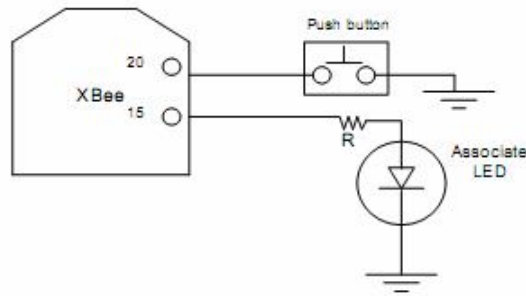


Figura 2.19 - Configuração Típica para Diagnóstico da Rede

Fonte: XBee ZNet 2.5/XBee-PRO ZNet 2.5 OEM RF Modules, 2008

Tabela 2.6 - Funções Disponíveis para o Commissioning Pushbutton

Button Presses	If module is joined to a network	If module is not joined to a network
1	<ul style="list-style-type: none"> <li>Wakes an end device for 30 seconds</li> <li>Sends a node identification broadcast transmission</li> </ul>	<ul style="list-style-type: none"> <li>Wakes an end device for 30 seconds</li> <li>Blinks a numeric error code on the Associate pin indicating the cause of join failure (see section 6.4.2).</li> </ul>
2	<ul style="list-style-type: none"> <li>Sends a broadcast transmission to enable joining on the coordinator and all devices in the network for 1 minute. (If joining is permanently enabled on a device (NJ = 0xFF), this action has no effect on that device.)</li> </ul>	<ul style="list-style-type: none"> <li>N/A</li> </ul>
4	<ul style="list-style-type: none"> <li>Causes the device to leave the PAN.</li> <li>Issues ATRE to restore module parameters to default values, including ID and SC.</li> <li>The device attempts to join a network based on its ID and SC settings.</li> </ul>	<ul style="list-style-type: none"> <li>Issues ATRE to restore module parameters to default values, including ID and SC.</li> <li>The device attempts to join a network based on its ID and SC settings.</li> </ul>

Fonte: XBee ZNet 2.5/XBee-PRO ZNet 2.5 OEM RF Modules, 2008

O *Associated LED* permite identificar se o módulo em questão está ou não conectado a alguma PAN. Quando estiver em nível lógico alto, o LED estará aceso indicando que o módulo não está conectado a nenhuma rede. Assim, que encontrar uma rede que permite a entrada do módulo, o *Associated LED* piscará indicando que o dispositivo está associado a uma ZNet.



### 3. METODOLOGIA

O projeto descrito visa criar um sistema de comunicação ZigBee para automatizar os equipamentos climáticos em estufas agrícolas. Para isso, o desenvolvimento do projeto foi dividido em 4 partes:

**Os Sensores Remotos** – Os sensores remotos são pequenos dispositivos que possuem os módulos XBee configurados como terminais pois são alimentados por pilhas e portanto precisam de hardwares de baixo consumo. São responsáveis pela coleta de informações climáticas do ambiente. Este projeto contará com dois destes sensores para coletar informações dentro e fora da estufa.

**O Controle Central** – Possui um módulo XBee configurado como roteador para fazer a ponte entre os sensores remotos e o coordenador. Este circuito também é responsável pelo acionamento dos equipamentos de climatização da estufa.

**O Coordenador** – Hardware que possui a função de gateway, ou seja, comunica-se serialmente com o computador através de uma porta USB (que simula uma porta RS232) e com outros módulos XBee existentes na rede ZNet via RF (ZigBee).

**O Monitor XBee** – Software responsável pela automação da estufa e monitoramento da rede ZNet. Comunica-se exclusivamente com o coordenador e faz a interface entre o usuário e o sistema de automação.

O sistema deste projeto simulará a presença de 3 equipamentos muito comuns em estufas. Todos eles são alimentados no painel central e acionados pelo controle central.

**Aquecedor** – Aquece o ambiente a partir da liberação de calor da queima de querosene ou diesel.

**Exaustor** – Elimina o ar quente no interior da estufa diminuindo ligeiramente a temperatura.

**Sombreamento** – Acionado para aumentar ou diminuir a luminosidade do ambiente.

A Figura 3.1 ilustra a arquitetura deste sistema.

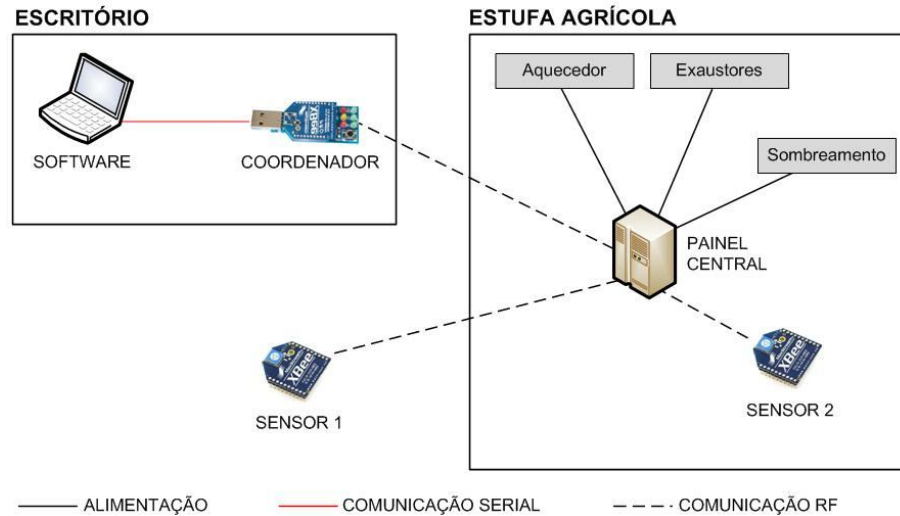


Figura 3.1 - Arquitetura do Sistema de Automação Simulado neste Projeto

### 3.1.0 SENSOR REMOTO

O sistema criado neste projeto conta com dois sensores móveis alimentados por pilhas comuns. Ambos os dispositivos são embutidos com um sensor de temperatura e um sensor de luz que deverão ser posicionados dentro e fora da estufa. Assim, pode-se monitorar tanto as condições internas da estufa na qual as plantas estão localizadas, quanto o ambiente externo para elevar as opções de automação.

A Figura 3.2 é uma representação em 3D da placa de circuito impresso do sensor remoto, feita no software Proteus ARES [5].

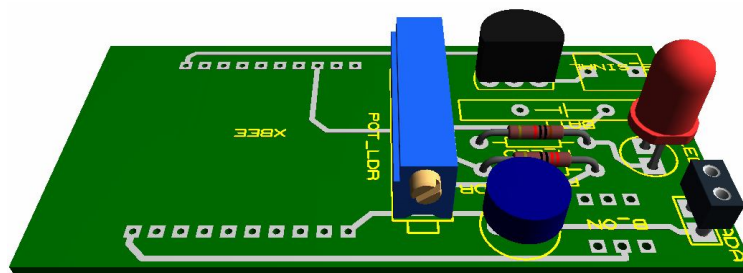


Figura 3.2 - Desenho 3D da placa de Circuito Impresso dos Sensores Remotos

#### 3.1.1. FONTE DE ALIMENTAÇÃO

As duas pilhas de 1.5V são utilizadas em série para gerar uma tensão de 3V que alimenta o módulo XBee. Porém, para alimentar o sensor de temperatura (LM35) é necessário elevar esta tensão

para, no mínimo, 4V. Para isto, utilizou-se uma pequena bateria de 3V que eleva a tensão gerada pelas pilhas para 6V permitindo que o sensor seja alimentado conforme sua especificação.

A bateria utilizada é a CR2032 da Panasonic [10]. Ela possui uma carga de 220mAh e o LM35 consome cerca de 50µA, portanto:

$$t = \frac{220000}{50} = 4400 \text{ horas} = 180 \text{ dias}$$

Assim, a cada 180 dias seria necessário substituir a bateria de 3V por uma nova. Este inconveniente pode ser resolvido substituindo o sensor LM35 pelo LM60 que pode operar com no mínimo 2.6V, permitindo que seja alimentado pelas duas pilhas de 1.5V. Porém o LM60 é difícil de ser encontrado no mercado e possui um preço mais elevado.

### 3.1.2. PARAMETRIZAÇÃO DO XBEE

Para realizar a transmissão de dados com o controle central, os sensores remotos utilizam o módulo XBee configurado para operar como *end device* (terminal). Desta maneira, é possível economizar energia e prolongar o tempo da pilha.

Os módulos dos sensores remotos foram parametrizados conforme a Tabela 3.1.

Tabela 3.1 – Parametrização do Módulo XBee para os Sensores Remotos

Comando AT	Nome	Valor	Descrição
<b>Sleep</b>			
SM	Sleep Mode	0x04	Operado para dormir periodicamente (ou ciclicamente).
SP	Sleep Period	0x03E8	Acorda a cada 10s (0x03E8 * 10ms)
SN	Number of Sleep Periods	0x03	Piscará seu Status LED a cada 30s (3*10s)
ST	Time Before Sleep	0x01F4	Quando acordar, voltará a dormir em 500ms (0x01F4 ms)
<b>Comandos I/O</b>			
IR	IO Sample Rate	0x03E8	Envia uma amostra das portas digitais, analógicas e tensão de alimentação do módulo a cada 1s (0x03E8 ms)
V+	Supply Voltage Monitoring	0xBFD	Monitora a alimentação do módulo pelo comando IR quando a tensão estiver abaixo de 3.6V (0x0BFD*0.8525mV)
D0	AD0/DIO0 Configuration	0x01	Pino 20 setado como <i>commissioning button</i>
D1	AD1/DIO1 Configuration	0x02	Pino 19 setado como entrada analógica
D2	AD2/DIO2 Configuration	0x02	Pino 18 setado como entrada analógica
<b>Interface RF</b>			
PM	Power Mode	0x01	Boost Mode habilitado

Entre o pino 13 (*ON/SLEEP*) do módulo e o potencial terra, foi ligado em série um resistor de  $680\Omega$  e um LED verde para indicar o status do módulo. Quando estiver em *sleep mode*, o pino 13 estará em nível baixo e o LED estará apagado. Assim que o módulo acordar, o pino irá alterar seu estado para nível lógico alto e o LED se acenderá indicando que o dispositivo pode realizar transmissões RF.

O sinal que alimentará os sensores de luz e temperatura também é originado no pino 13. Assim, os sensores somente consumirão energia quando o módulo estiver acordado e apto a interpretar os sinais de saída dos sensores em suas portas analógicas.

Para melhorar o controle do sistema, dois botões foram adicionados as placas dos sensores remotos. Um de posição fixa que permite desligar o dispositivo cortando a alimentação das pilhas. O segundo botão possui posição momentânea e foi conectado entre o pino 20 do módulo e o potencial terra. Este último, é utilizado como *commissioning button*, descrito no capítulo 2.2.7.

### 3.1.3. SENSOR DE TEMPERATURA

O circuito integrado LM35 [9] utilizado neste projeto é um sensor de temperatura de precisão que varia sua tensão de saída linearmente, proporcional à temperatura ambiente em graus Celsius. Também possui uma baixa impedância de saída e sem nenhuma calibração externa, ele é capaz de fornecer  $10\text{mV}/^\circ\text{C}$  em sua saída com uma precisão de  $\pm 1/4^\circ\text{C}$ .



Figura 3.3 - Vista Inferior e Circuito de Aplicação Típico para o LM35

Fonte: LM35 Datasheet

A alimentação do LM35, como descrita anteriormente, parte do pino 13 (*ON/SLEEP*) do módulo XBee e é elevada a 6V pela bateria. A saída do CI foi conectada diretamente no pino 19 (*Analog Input*) do módulo.

### 3.1.4. SENSOR DE LUMINOSIDADE

Para monitorar as condições de iluminação de um ambiente, este projeto utiliza um LDR. Estes componentes elétricos são capazes de variar sua resistência conforme a intensidade luminosa incidente sobre ele. A relação entre a resistividade e a intensidade de luz de um LDR típico é dada pela seguinte fórmula:

$$R_{LDR} = \frac{500}{lux} (k\Omega)$$

\*Testes realizados comprovaram que esta fórmula não pode ser utilizada em determinados níveis de luz. Um estudo mais aprofundado sobre o LDR poderia ser realizado em uma futura continuação do projeto para eliminar estes valores incorretos de medida.

A unidade lux é utilizada pelo sistema internacional de medidas para medir a intensidade luminosa. Uma noite de lua cheia apresenta cerca de 1 lux enquanto um dia com sol forte ultrapassa facilmente 100 klux no Brasil. Portanto o LDR pode variar sua resistência desde valores próximos a 50Ω até maiores que 500kΩ.

Em uma estufa agrícola, os principais equipamentos que alteram a intensidade luminosa são o sombreamento e as lâmpadas. Neste projeto será simulado um sombreamento para diminuir a intensidade luminosa da estufa. Portanto, o sensor de luz precisa identificar as altas intensidades de luz para fornecer informações úteis ao controle do sombreamento. Neste caso, as baixas intensidades de luz não precisam ser monitoradas com precisão e serão descartadas.

Sendo assim, este projeto utiliza o monitoramento de intensidades luminosas acima de 75klux, correspondendo a um dia nublado. Este valor induzirá o LDR a apresentar a seguinte resistência elétrica:

$$R_{LDR} = \frac{500}{75000} = 6.7k\Omega$$

Para obter um sinal de luz dentro da faixa estipulada, foi construído um divisor de tensão conforme a Figura 3.4, lembrando que o módulo XBee interpreta sinais de 0 a 1.2V em sua entrada analógica (10 bits).

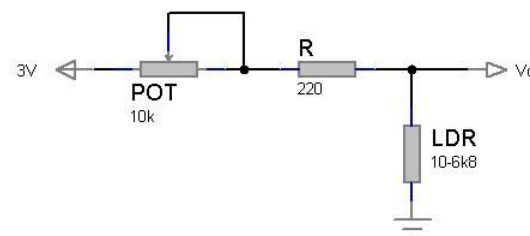


Figura 3.4 - Divisor de Tensão para o Sensor de Luz (LDR)

$$V_o = 3 * \frac{ldr}{ldr + pot + 220}$$

Para POT = 10kΩ:

$$V_o = 3 * \frac{l_{dr}}{l_{dr} + 10220}$$

Considerando uma intensidade luminosa de no mínimo 75 klux:

$$V_o = 3 * \frac{6700}{6700 + 10220} = 1.1879 V$$

Em intensidades maiores de luz, o LDR tende a diminuir sua resistência e conseqüentemente o valor final da tensão do divisor ( $V_o$ ) também diminuirá mantendo o sinal dentro da faixa de tensão suportada pelo módulo.

Intensidades de luz inferiores a 75klux farão com que  $V_o$  ultrapasse o valor máximo suportado pelo XBee e conseqüentemente o módulo interpretará estes valores como 1.2V. O software, quando receber estas informações, interpretará como baixa intensidade luminosa, porém sem valores de grandeza.

A presença do potenciômetro permite ao usuário ajustar a precisão do sensor de luminosidade à custa de uma elevação do nível mínimo de luz necessária para manter o sinal dentro da faixa da entrada analógica do módulo XBee.

### 3.2.CONTROLE CENTRAL

O controle central desempenha duas importantes funções para o sistema. A primeira função é realizada somente pelo módulo XBee da placa e consiste em rotear os pacotes envolvendo os sensores remotos e o coordenador. A segunda é gerenciar o funcionamento dos equipamentos da estufa e reportar informações ao coordenador. Mais detalhes sobre estas funções estão descritas no capítulo 3.2.2.

A Figura 3.5 foi desenvolvida no software Proteus ARES [5] e ilustra em 3 dimensões a placa de circuito impresso do controle central.

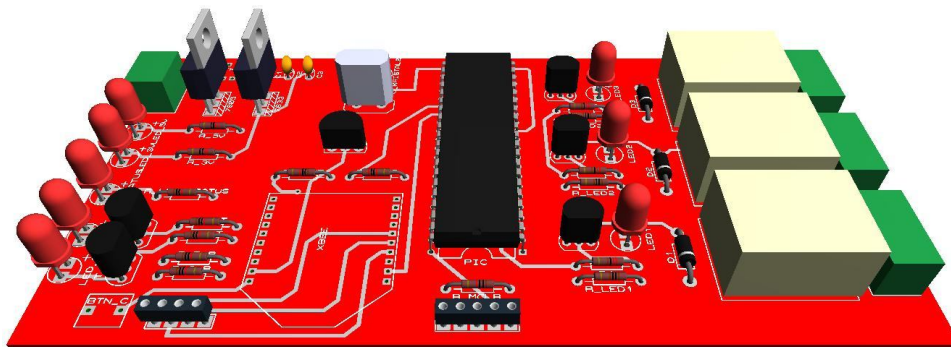


Figura 3.5 - Desenho 3D da Placa de Circuito Impresso do Controle Central

### 3.2.1. FONTE DE ALIMENTAÇÃO

O controle central é equipado com dois reguladores de tensão: LM7805 e UA78M33C. O primeiro é responsável pela tensão de 5V que alimenta o microcontrolador, o sensor de temperatura e o segundo regulador de tensão, UA78M33C, que por sua vez alimenta o módulo XBee com uma tensão de 3.3V. Para monitorar os CIs reguladores de tensão, LEDs foram conectados na saída de cada um deles para sinalizar se a tensão fornecida pelos reguladores está funcionando.

A alimentação da placa é ligada diretamente no LM7805 que permite tensões de entrada entre 5.7V e 35V, portanto é necessário utilizar uma fonte externa que forneça uma determinada tensão dentro desta faixa.

Capacitores foram adicionados no circuito de alimentação da placa para filtrar pequenas flutuações na tensão. A Figura 3.6 ilustra as posições e valores dos capacitores no circuito final do alimentador.

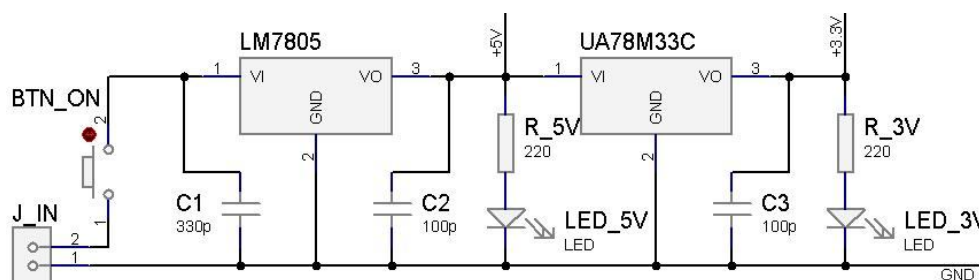


Figura 3.6 - Circuito Regulador de Tensão para o Comando Central

O botão BTN\_ON foi adicionado para permitir desligar a alimentação da placa quando não estiver em uso.

### 3.2.2. PARAMETRIZAÇÃO DO XBEE

O módulo transceptor de RF - XBee ZNet 2.5 - foi configurado para operar como roteador por possui duas funções importantes para o funcionamento do sistema de automação como descrito anteriormente. A primeira é fazer a ponte entre os sensores remotos e o coordenador, pois como os módulos *end device* (sensores remotos) ficam em *sleep mode* na maior parte do tempo, o roteador armazena as instruções destinadas a estes terminais até o momento em que acordarem, como descrito no capítulo 2.2.3.

A segunda função do roteador é transmitir e receber pacotes API contendo o status das portas I/O do microcontrolador, uma vez que este determina se um equipamento está ligado ou não. Assim, o XBee recebe pacotes 0x10 do coordenador para retransmiti-los ao microcontrolador que irá interpretar este pacote e alterar o funcionamento de algum equipamento se necessário. Também irá transmitir ao

coordenador pacotes API originados pelo microcontrolador contendo o status de todas as portas. Mais informações sobre a configuração do microcontrolador podem ser obtidas no capítulo seguinte.

O controle central também conta com um led ligado no pino 15 do módulo com a função de *associated indication LED*, um botão de posição momentânea como *commissioning button* no pino 20 e um sensor de temperatura LM35 com sua porta de saída conectada diretamente no pino 19 do XBee. Este sensor verifica a temperatura em que a placa do controle central está operando e não deve ser utilizado como medida de temperatura da estufa.

Para realizar as funções descritas anteriormente, o módulo XBee foi parametrizado conforme descrito na

Tabela 3.2.

Tabela 3.2 - Parametrização do Módulo XBee do Painel de Comando

Comando AT	Nome	Valor	Descrição
<b>Sleep</b>			
SM	Sleep Mode	0x00	Operar como roteador (não dormir)
<b>Comandos I/O</b>			
IR	IO Sample Rate	0xEA60	Envia uma amostra das portas digitais, analógicas e tensão de alimentação do módulo a cada 1 minuto (0xEA60 ms)
V+	Supply Voltage Monitoring	0xBFD	Monitora a alimentação do módulo pelos comando IR quando a tensão estiver abaixo de 3.6V (0xBFD*0.8525mV)
D0	AD0/DIO0 Configuration	0x01	Pino 20 configurado como <i>commissioning button</i>
D1	AD1/DIO1 Configuration	0x02	Pino 19 configurado como entrada analógica
D5	DIO5 Configuration	0x01	Pino 15 configurado como <i>associated indication LED</i>
<b>Interface RF</b>			
PM	Power Mode	0x01	Boost Mode habilitado

O parâmetro V+ foi utilizado para indicar se o regulador UA78M33C está fornecendo a tensão adequada ao módulo.

### 3.2.3. O MICROCONTROLADOR

Para monitorar e automatizar os equipamentos presentes na estufa, o microcontrolador PIC16F877 foi escolhido para o projeto por possuir até 32 portas I/O e suporte a comunicação UART, porém o projeto simula a atuação de somente 3 equipamentos e portanto este microcontrolador pode ser facilmente substituído sem alterar o funcionamento do sistema. A Figura 3.7 ilustra os pinos do PIC16F877A.

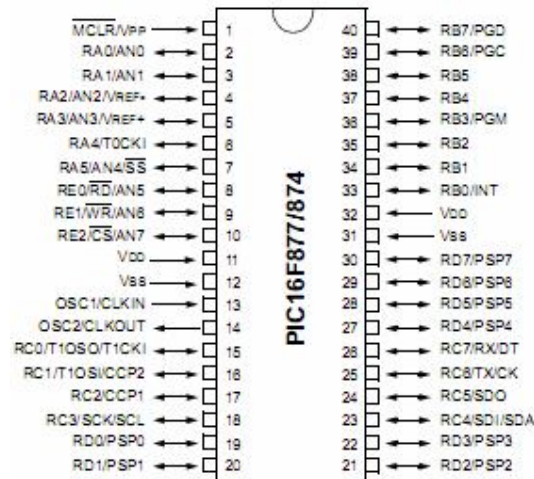


Figura 3.7 - Pinagem do PIC16F877A

Fonte: PIC16F877/874 Datasheet

A função do microcontrolador é monitorar o funcionamento dos equipamentos presentes na estufa agrícola. A interface entre a alimentação dos equipamentos e o PIC é realizada por um circuito ilustrado na Figura 3.8.

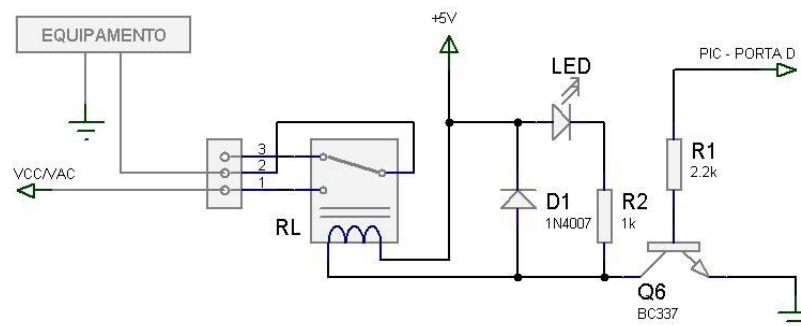


Figura 3.8 - Circuito de Saída para Conexão de Equipamentos ao Controle Central

Para transmitir e receber informações sobre o status das portas I/O, os pinos RX e TX do microcontrolador são conectados respectivamente nas portas DOUT e DIN do módulo XBee. Porém, como o módulo opera em 3.3V e o microcontrolador em 5V, é necessário diminuir a tensão do pino TX do PIC. Para isso, utilizou-se um divisor de tensão como demonstra o circuito da Figura 3.9. A comunicação entre os pinos DOUT do XBee e o RX do PIC ocorre normalmente sem a necessidade de ajustar os níveis de tensão.

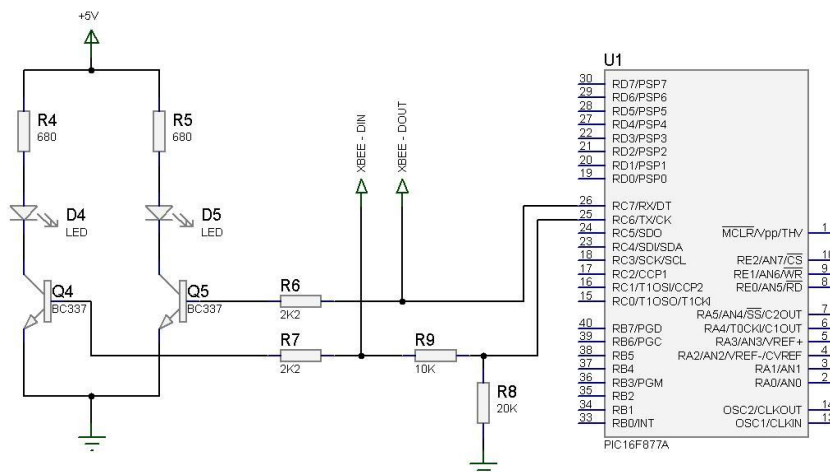


Figura 3.9 - Circuito de Comunicação Serial

Os LEDs e os transistores BC337 indicados na Figura 3.9 foram adicionados para visualizar se alguma comunicação entre o módulo XBee e o microcontrolador está sendo realizada.

Toda a comunicação necessária para automatizar os equipamentos pelo microcontrolador percorre o caminho ilustrado na Figura 3.10.

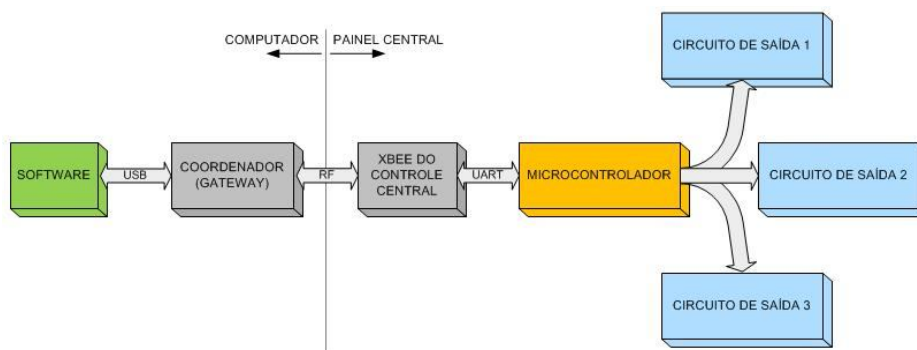


Figura 3.10 - Fluxograma do Caminho Percorrido pelas Informações de Comando

As decisões relacionadas ao funcionamento dos equipamentos da estufa agrícola é feita pelo software que envia um pacote API 0x10 pela porta USB do computador (que simula uma conexão serial) para o módulo XBee do coordenador. Este irá enviar o pacote por RF (no protocolo ZigBee) para o módulo roteador descrito no capítulo anterior. Em seguida, o roteador enviará dois pacotes API: Um 0x8B para o coordenador informando que recebeu o pacote 0x10 com sucesso, e outro 0x90 que será enviado pela UART para o microcontrolador contendo os dados do pacote 0x10 recebido anteriormente.

Este último pacote será recebido e interpretado pelo microcontrolador para alterar, se necessário, o sinal da porta D configurada como *output*. Os sinais interpretados pelo microcontrolador

possuem dois campos. O primeiro possui a função de acionar o reconhecimento de comandos e é identificado pelo número hexadecimal 0xCC. O segundo é responsável pelo comando em si. A Tabela 3.3 descreve os comandos e as funções utilizadas nesta etapa do projeto.

*Tabela 3.3 - Comandos para Atuação dos Equipamentos*

Comando	Função
<b>D0</b>	Altera o status do pino 19 (D0) para 0.
<b>D1</b>	Altera o status do pino 19 (D0) para 1.
<b>E0</b>	Altera o status do pino 20 (D1) para 0.
<b>E1</b>	Altera o status do pino 20 (D1) para 1.
<b>F0</b>	Altera o status do pino 21 (D2) para 0.
<b>F1</b>	Altera o status do pino 21 (D2) para 1.
<b>AA</b>	Envia um pacote API pelo pino TX indicando o status da porta D.

Independentemente do comando enviado, o microcontrolador responderá pelo pino TX com um pacote API 0x10 destinado ao coordenador, indicando o status da porta D. Dessa maneira, o software poderá interpretar o comando para identificar os sinais atuais da porta de saída do microcontrolador.

**Exemplo:** Com todos os equipamentos desligados, o software envia pelo pacote API 0x10 o comando CCD1 para o microcontrolador alterar o estado do pino 20. Assim que o PIC realizar esta alteração no pino, ele responderá para o coordenador - também com um pacote 0x10 - informando o nível lógico de todos os pinos da porta D, ou seja, enviará o byte 0x02 (0000010b) e o software identificará que o status do pino 20 foi efetivamente alterado.

### 3.3.O COORDENADOR

O coordenador é responsável em fazer o link para comunicação entre o PC e o controle central. Ele é composto por dois dispositivos:

- Um módulo XBee ZNet 2.5
- Uma placa CON-USBEE

O módulo XBee foi configurado para atuar como coordenador. Assim, é o responsável pela criação da rede do sistema de automação. Nenhuma parametrização foi realizada neste módulo, ou seja, todos os seus atributos foram deixados no padrão, exceto o atributo NI que foi configurado como “Coordenador” para dar um apelido ao módulo.

A placa CON-USBEE é fabricada pela RogerCom e está ilustrada na Figura 3.11. Sua função é permitir a comunicação serial entre um módulo XBee e um computador pelo conector USB. Para isto, ele utiliza um CI conversor USB/Serial e necessita instalar um driver no computador disponibilizado no site do fabricante. O driver simula uma porta serial no computador para transmitir os dados enviados serialmente à saída USB.



Figura 3.11 - CON-USBEE da RogerCom

Fonte: [www.rogercom.com.br](http://www.rogercom.com.br)

Além da principal função de gateway da placa CON-USBEE, o dispositivo também apresenta 3 LEDs RSSI que indicam a “força” da transmissão RF recebida. Assim, é possível identificar se o sinal está muito fraco e é necessário encurtar a distância entre os módulos ou se existe algum objeto interferindo na comunicação wireless.

LEDs conectados aos pinos DIN e DOUT também são encontrados no dispositivo para identificar se comunicações seriais estão sendo realizadas. O LED com a função de *associated indication LED* também é encontrado. A Figura 3.12, obtida no site da Rogercom, ilustra a função dos LEDs descritos.

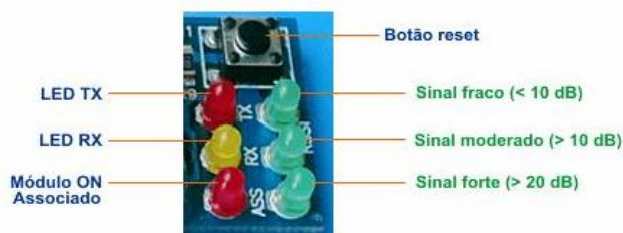


Figura 3.12 - Descrição dos LEDs do dispositivo CON-USBEE

Fonte: [www.rogercom.com.br](http://www.rogercom.com.br)

### 3.4.O MONITOR XBEE

O Monitor XBee é um software que foi desenvolvido em linguagem VB.Net através do compilador Visual Studio 2008 da Microsoft. É responsável pelo gerenciamento de informações na rede XBee e comunica-se exclusivamente com o coordenador conectado à porta USB.

Este software pode ser dividido em duas partes: uma responsável pelo gerenciamento do sistema criado neste projeto e é chamada de **painel do sistema**; outra parte, chamada de **painel geral**, é responsável pelo gerenciamento de todos os pacotes recebidos e transmitidos pela porta USB.

O fluxograma abaixo (Figura 3.13) demonstra como funciona a interpretação dos pacotes recebidos pelos painéis do software.

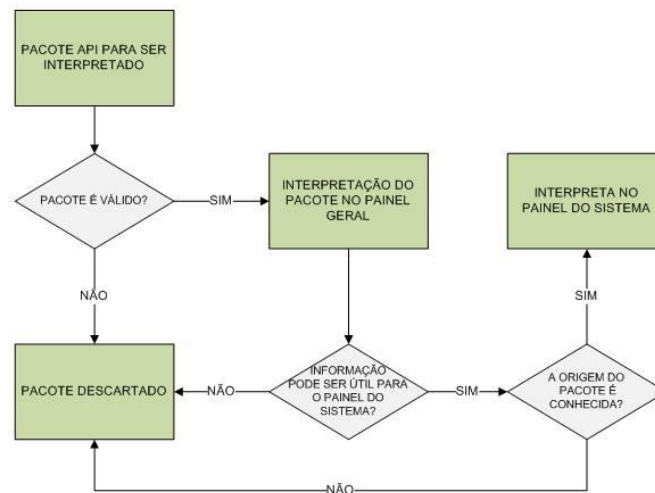


Figura 3.13 - Diagrama de Blocos da Interpretação de Dados do Software

### 3.4.1. PAINEL DO SISTEMA

Ela recebe informações sobre a tensão obtida nos sensores de temperatura e luz dos dois sensores remotos, faz o processamento desta informação e disponibiliza para o usuário o valor real da temperatura e intensidade de luz. Também disponibiliza uma interface de automação dos 3 equipamentos do sistema: o aquecedor, o ventilador e o sombreamento.

Para automatizar o funcionamento destes equipamentos, é possível definir valores limites chamados *SetPoints*, de modo que se a temperatura e/ou luminosidade ultrapassarem os valores definidos, o software enviará um pacote API 0x90 com o comando para alterar o estado do equipamento conforme descrito no capítulo 2.2.5. Existem diversas combinações de automação com *SetPoints* que podem ser configuradas. A Figura 3.14 ilustra como é feita a configuração desta automação enquanto a Figura 3.15 ilustra a interface do usuário com o sistema simulado neste projeto (Painel do Sistema).

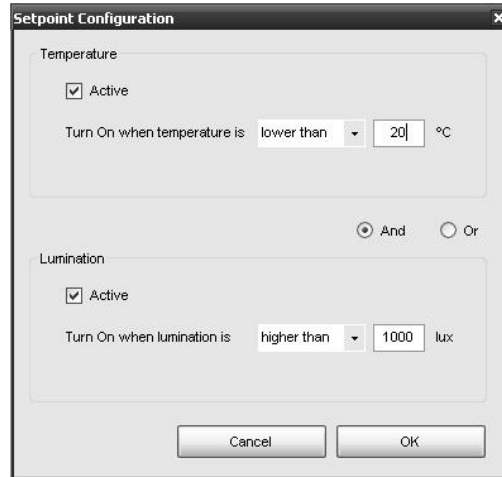


Figura 3.14 - Configuração de SetPoint Disponível no Software



Figura 3.15 - Painel do Sistema

### 3.4.2. PAINEL GERAL

A segunda parte do programa pode ser utilizada por qualquer rede envolvendo módulos XBee. A Figura 3.16 ilustra a interface do Painel Geral e suas 3 subdivisões.

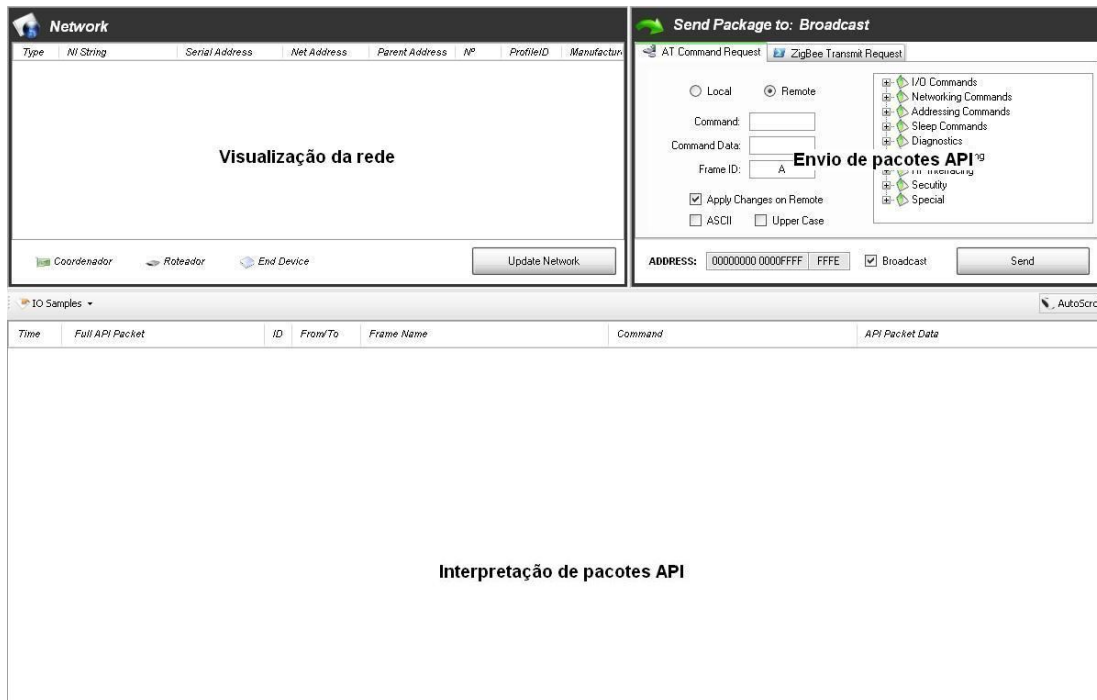


Figura 3.16 - Painel Geral do software Monitor XBee

**Visualização da Rede (Network)** – Através dos pacotes recebidos pelo programa após o envio do comando NI para o coordenador, disponibiliza informações sobre todos os módulos da rede como os apelidos de identificação (*NI String*), o número serial, endereço de rede, tipo do dispositivo e o endereço de rede do pai dos módulos XBee.

**Envio de Pacotes API (Send API Package)** – Aqui, é possível enviar todos os comandos AT suportados pelos módulos XBee para qualquer dispositivo da rede. Pacotes 0x08, 0x10 e 0x17 podem ser enviados em *unicast* ou *broadcast*.

**Campo de Interpretação dos Pacotes** – Esta parte do programa é responsável por interpretar todos os pacotes API que estão sendo enviados ou recebidos pela porta USB e disponibilizá-los ao usuário de maneira didática.

Na Figura 3.17 é possível visualizar a interface que o usuário possui com o software completo.

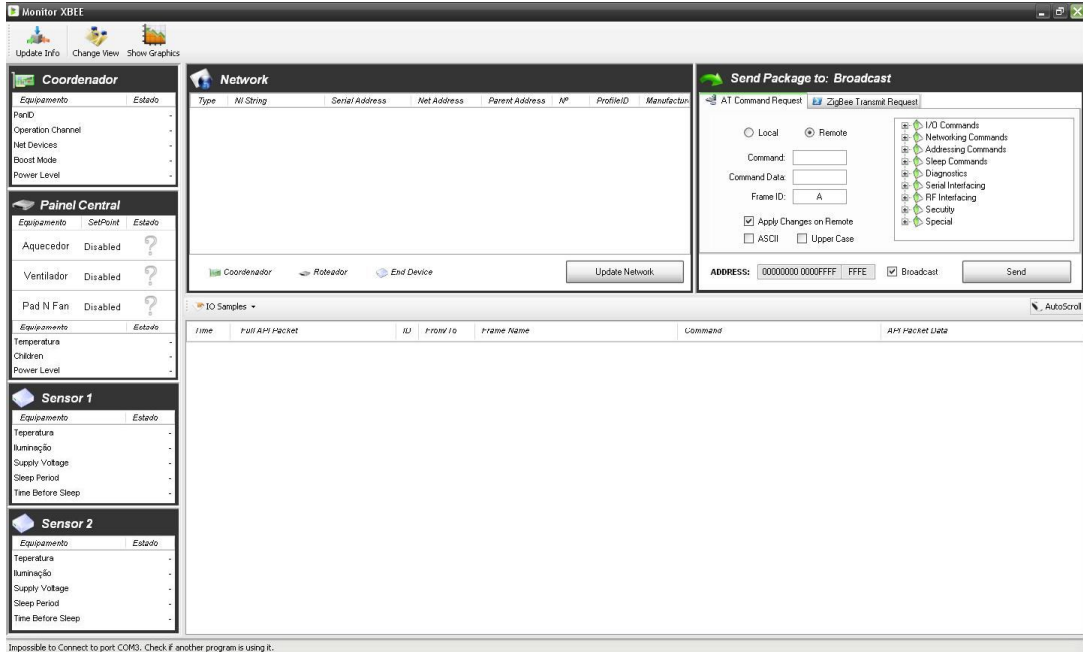


Figura 3.17 - Programa de Monitoração e Automação da rede ZNet

## 4. RESULTADOS

Os sensores remotos foram posicionados em uma distância de 40 metros do controle central, que por sua vez ficou a 40 metros do computador com o coordenador embutido em sua porta USB. Os sensores remotos 1 e 2 simularam as variações climáticas no exterior e interior da estufa, respectivamente, conforme ilustrado na Figura 3.1.

No painel do sistema do software Monitor XBee, os *SetPoints* dos três equipamentos foram configurados conforme a Tabela 4.1.

Tabela 4.1 - Configuração de *SetPoints* para o Teste do Sistema

Equipamento	Acionamento		
	Temperatura Interna	AND/OR	Iluminação Externa
Aquecedor	< 15°C	OR	-
Ventilação	> 25°C	OR	-
Sombreamento	-	OR	> 75 klux

Em seguida, todos os dispositivos foram ligados e o software foi inicializado no computador. No painel geral do software, foi possível visualizar os dispositivos conectados à rede ZNet do coordenador pelo painel “*Network*” conforme a Figura 4.1 - Módulos XBee conectados à PAN do coordenador.

Type	NI String	Serial Address	Net Address	Parent Address	Nº	ProfileID	Manufacturer
	Office	0013A200 40314E30	0000	FFFE	0	C105	101E
	Central	0013A200 40314E31	20B2	FFFE	0	C105	101E

Coordenador   
 Roteador   
 End Device   

Figura 4.1 - Módulos XBee conectados à PAN do coordenador

Pacotes API foram enviados para efeito de teste no painel geral do software e todos os pacotes foram interpretados pelo Monitor XBee no campo de interpretação de pacotes API conforme a Figura 4.2.

Time	Full API Packet	ID	From/To	Frame Name	Command	API Packet Data
30/10/2009 - 05:02:42	7E000F17570013A20040314E31FFFE024E445B...	W	Sensor1 (MY Missing)	0x17 - Remote AT Command Requ...	Apply Changes on remote.	Node Discovery Request
30/10/2009 - 05:02:43	7E002897570013A20040314E3000004E4400D...	W	0013A20040314E30 0000	0x97 - Remote Command Response	OK	Coordinator - Office (0013A200 40314E30 0000) Par...
30/10/2009 - 05:02:44	7E000408584E440D	X	Local Command.	0x08 - AT Command Request	Local Command.	Node Discovery Request
30/10/2009 - 05:02:48	7E002588584E440020B20013A20040314E314...	X	Local Command.	0x08 - AT Command Response	OK	Routerno one - Central_Panel (0013A200 40314E31 ...
30/10/2009 - 05:02:55	7E000F17410013A20040314E3120B2024E4997	A	Sensor1	0x17 - Remote AT Command Requ...	Apply Changes on remote.	Nothing Found! (NI - )
30/10/2009 - 05:02:55	7E001C97410013A20040314E3120B24E49004...	A	Sensor1	0x97 - Remote Command Response	OK	Nothing Found! (NI - 43656E7472616C5F50616E656C)
30/10/2009 - 05:03:08	7E001617410013A20040314E3120B2024E494...	A	Sensor1	0x17 - Remote AT Command Requ...	Apply Changes on remote.	Nothing Found! (NI - 43656E7472616C)
30/10/2009 - 05:03:08	7E000F97410013A20040314E3120B25752007	A	Sensor1	0x97 - Remote Command Response	OK	Nothing Found! (VWR - )
30/10/2009 - 05:03:12	7E000F17410013A20040314E3120B202575285	A	Sensor1	0x17 - Remote AT Command Requ...	Apply Changes on remote.	Nothing Found! (VWR - )
30/10/2009 - 05:03:12	7E000F97410013A20040314E3120B25752007	A	Sensor1	0x97 - Remote Command Response	OK	Nothing Found! (VWR - )
30/10/2009 - 05:03:14	7E000F17570013A20040314E31FFFE024E445B...	W	Sensor1 (MY Missing)	0x17 - Remote AT Command Requ...	Apply Changes on remote.	Node Discovery Request
30/10/2009 - 05:03:18	7E002897570013A20040314E3000004E4400D...	W	0013A20040314E30 0000	0x97 - Remote Command Response	OK	Coordinator - Office (0013A200 40314E30 0000) Par...
30/10/2009 - 05:03:18	7E000408584E440D	X	Local Command.	0x08 - AT Command Request	Local Command.	Node Discovery Request
30/10/2009 - 05:03:19	7E001F88584E440020B20013A20040314E314...	X	Local Command.	0x08 - AT Command Response	OK	Routerno one - Central (0013A200 40314E31 20B2) ...
30/10/2009 - 05:03:54	7E001010400013A20040314E3120B2000DCC...	M	Sensor1	0x10 - ZigBee Transmit Request	Maximum Number of hops: 10 - Un...	UART Data: CCE1
30/10/2009 - 05:03:55	7E0007984D20B200000055	M	20B2	0x08 - ZigBee Transmit Status	Number of transmission retries: 00	Success: No Discovery Overhead.
30/10/2009 - 05:03:55	7E0000900013A20040314E3120B20108F1		Sensor1	0x90 - ZigBee Receive Packet	Packet Acknowledged.	RF Packet: 06

Figura 4.2 - Interpretação dos pacotes API obtidas no software Monitor XBee

Os valores de temperatura e iluminação registrados pelo programa foram comparados com valores obtidos em um termômetro e um luxímetro.

As condições climáticas de luz e temperatura foram alteradas artificialmente no ambiente em que estavam localizados os sensores, enquanto os circuitos de saída do controle central foram monitorados para verificar a eficiência da automação com *SetPoints* definidos.

A corrente consumida pelos sensores também foi monitorada com o uso de um multímetro.

O sistema reagiu às variações climáticas conforme o previsto. A comparação entre a medida feita pelo termômetro e a medida de temperatura obtida pelo software registrou uma diferença de 0.75 °C enquanto que as diferenças das medidas de luz registradas ficaram próximas de 2.5 klux.

O consumo de energia medido nos sensores remotos não ultrapassou 18mA quando os dispositivos acordavam. Em *sleep mode*, o consumo ficou abaixo de 50µA.

O circuito de saída no painel central referente ao aquecedor foi acionado enquanto a temperatura registrada pelo sensor remoto 2 estava abaixo de 15 °C e assim que a temperatura superou 25°C o circuito de saída do ventilador foi acionado.

O sombreamento foi acionado assim que a iluminação externa medida pelo sensor remoto 1 ultrapassou 75 klux, conforme o previsto.

Em relação ao número de cabos, um painel normal - sem a tecnologia wireless - possuiria 7 cabos: 1 de alimentação do painel; 3 para alimentar os equipamentos; 2 para alimentar e receber dados dos sensores e 1 para levar as informações do painel da estufa para o computador no escritório. Neste sistema wireless foram utilizados somente 4 cabos: 1 de alimentação do painel e 3 para alimentar os equipamentos, diminuindo em 43% o número de cabos na estufa.

No quesito confiabilidade, não foi detectado nenhum erro de pacote durante a duração do teste. A distância de 40 metros provou ser suficiente para uma manter a confiabilidade do sistema.

## 5. CONCLUSÃO

A utilização de módulos XBee para o sistema de automação de estufas agrícolas trouxe vantagens que idealizam o futuro desta área.

Os sensores em especial foram os dispositivos mais beneficiados com este projeto. Tornaram-se totalmente móveis e independentes de cabos, sem a necessidade de nenhuma intervenção humana para operar durante vários anos.

No painel central da estufa, a aglomeração dos cabos - visão muito comum nestes ambientes - foi consideravelmente reduzida com a implantação da tecnologia wireless e desta maneira, a manutenção do sistema elétrico ficou mais simples e fácil para detectar erros.

O software apresentou uma interface simples de automação que abrange uma grande quantidade de requisitos dos usuários. Também foi uma ferramenta de monitoração da rede XBee muito útil para o projeto e pode ser utilizada em futuras pesquisas envolvendo o protocolo em questão.

A grande limitação do projeto para ser utilizado comercialmente se dá ao fato de não apresentar sensores de umidade uma vez que estes são utilizados com grande frequência em ambientes agrícolas. Porém, como o módulo XBee apresenta até 4 entradas analógicas de 10 bits de precisão, mais sensores podem ser implantados na placa permitindo que o projeto possa ser expandido e atender requisitos mais específicos.

Em relação ao alcance de transmissão, o limite de comunicação dos módulos RF é uma característica dependente do modelo do módulo utilizado e por isso, o sistema pode ser adaptado para diversos tipos de instalação desde que os sensores remotos utilizem o modelo deste projeto para obter o menor consumo possível.

Assim, os módulos XBee provaram sua utilidade em projetos de automação e tendem a permanecer no mercado por um longo período graças à norma em que é baseado, ZigBee, apresentar um futuro promissor em diversas áreas envolvendo redes de baixo consumo.



## 6. TRABALHOS FUTUROS

Este projeto permite ser ampliado ou aperfeiçoado da seguinte maneira:

### **Sensores Remotos:**

- Adicionar um sensor de umidade;
- Estudar outros sensores de temperatura disponíveis no mercado como, por exemplo, os sensores da família SHT da Sensirion [13] que permitem medir temperatura e umidade em um único dispositivo de baixo consumo;
- Estudar mecanismos para diminuir o consumo energético da placa e aperfeiçoar o aproveitamento das baterias utilizadas.

### **Controle Central:**

- Substituir o PIC16F877A por outro microcontrolador que tenha um custo/benefício superior para este projeto em particular.

### **Coordenador:**

- Desenvolver um driver que permite uma taxa de transferência de dados de 250 kbps com o módulo ZigBee presente no coordenador.

### **Software:**

- Adicionar gráficos ao programa para aprimorar o monitoramento da estufa.
- Permitir ao usuário acesso aos dados da estufa pela internet.

### **Sistema:**

- Estudar a substituição do módulo XBee por um CI como o CC2430 da Chipcon [14].
- Realizar testes e estudo de viabilidade com o módulo XBee-PRO.



## 7. BIBLIOGRAFIA

- [1]. **Antonio Rogério Messias. 2009.** *RogerCom*. [Online] 2009. <http://www.rogercom.com.br>.
- [2]. **Digi Internacional Inc. 2008.** *XBee ZNet 2.5/XBee-PRO ZNet 2.5 OEM RF Modules*. 2008.
- [3]. **Fairchild Semiconductor. 2001.** *3-Terminal 1A Positive Voltage Regulator*. 2001.
- [4]. **Fairchild Semiconductor. 2002.** *BC337/338 Datasheet*. 2002.
- [5]. **Labcenter Electronics. 2009.** *Labcenter Electronics - Professional PCB Design and Simulation Software*. [Online] 2009. <http://www.labcenter.co.uk>.
- [6]. **Libelium. 2009.** *Libelium Wireless Sensor Networks*. [Online] 2009. [www.libelium.com](http://www.libelium.com).
- [7]. **MaxStream. 2005.** *XBee & XBee-PRO OEM RF Module Antenna Considerations*. 2005.
- [8]. **Microchip. 2001.** *PIC16F87X Datasheet*. 2001.
- [9]. **National Semiconductor. 1995.** *Precision Centigrade Temperature Sensors*. 1995.
- [10]. **Panasonic. 2003.** *Manganese Dioxide Lithium Coin Batteries: Individual Specifications*. 2003.
- [11]. **Rogercom. 2008.** *Manual da placa CON-USBEE*. 2008.
- [12]. **SAPEC Agro. 2009.** *Mildio dos Citrinos. SAPEC*. [Online] 2009. [http://www.sapecagro.pt/internet/webteca/artigo.asp?id=197&url\\_txt=&link=](http://www.sapecagro.pt/internet/webteca/artigo.asp?id=197&url_txt=&link=).
- [13]. **Sensirion AG.** *SHT Sensors. The Sensor Company*. [Online] [http://www.sensirion.com/en/01\\_humidity\\_sensors/00\\_humidity\\_sensors.htm](http://www.sensirion.com/en/01_humidity_sensors/00_humidity_sensors.htm).
- [14]. **Texas Instruments. 2009.** *CC2430*. [Online] 2009. <http://focus.ti.com/docs/prod/folders/print/cc2430.html>.
- [15]. **ZigBee Alliance. 2009.** *ZigBee Alliance WebPage*. [Online] 2009. <http://www.zigbee.org>.



## 8. ANEXOS

### 8.1. LISTA DE MATERIAIS

#### 8.1.1. SENSORES REMOTOS

- 1 Módulo XBee ZNet 2.5;
- 2 Conectores de 10 pinos com 2mm de passo;
- 2 Pilhas comuns de 1.5V;
- 1 Bateria de 3V e 220mAh modelo CR2032;
- 1 Sensor de temperatura LM35;
- 1 Sensor de luz LDR;
- 1 Botão de posição fixa;
- 1 Botão de posição momentânea;
- 1 LED Verde;
- 1 Potenciômetro de 10k;
- 1 Soquete de 2 pinos;
- Resistores.

#### 8.1.2. CONTROLE CENTRAL

- 1 Módulo XBee ZNet 2.5;
- 1 Microcontrolador PIC16F877A;
- 1 Sensor de temperatura LM35;
- 5 Transistores BC377;
- 1 LM7805;
- 1 UA78M33C;
- 1 Cristal de 4MHz;
- 3 Diodos 1N4007;
- 3 Relés monoestáveis HJR-3FF-S-Z;
- 3 Conectores de 3 pinos;
- 8 LEDs;
- 1 Botão de posição fixa;
- 1 Botão de posição momentânea;
- 1 Conector de 5 pinos;
- 1 Conector de 4 pinos;
- Resistores;
- Capacitores.

## 8.2.COMMANDOS AT DO MÓDULO XBEE/XBEE-PRO ZNET 2.5

Todas as tabelas abaixo foram retiradas do manual do módulo XBee [2].

Tabela 8.1 – Comandos AT: Especial

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
WR	<b>Write.</b> Write parameter values to non-volatile memory so that parameter modifications persist through subsequent resets. Note: Once WR is issued, no additional characters should be sent to the module until after the "OK" response is received. The WR command should be used sparingly. The EM250 supports a limited number of write cycles."	CRE	--	--
WB	<b>Write Binding Table:</b> Writes the current binding table to non-volatile memory.	CRE	--	--
RE	<b>Restore Defaults.</b> Restore module parameters to factory defaults. RE command does not reset the ID parameter.	CRE	--	--
FR	<b>Software Reset.</b> Reset module. Responds immediately with an "OK" then performs a reset ~2 seconds 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	<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. if NR = 1: Sends broadcast transmission to reset network layer parameters on all nodes in the PAN.	CRE	0 - 1	--

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

Tabela 8.2 - Comandos AT: Endereçamento

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
DH <sup>2</sup>	<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. 0x0000000000000000FFFF 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>	<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. 0x0000000000000000FFFF 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	<b>16-bit Network Address.</b> Get the 16-bit network address of the module.	CRE	0 - 0xFFFE [read-only]	0xFFFE
MP	<b>16-bit Parent Network Address.</b> Get the 16-bit parent network address of the module.	E	0 - 0xFFFE [read-only]	0xFFFE
NC	<b>Number of Children.</b> Read the number of end device children that have joined to the device. This command returns the number of child table entries currently in use.	CR	0 - 8	read-only
SH	<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	<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
NI	<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	ASCII space character (0x20)
DD	<b>Device Type Identifier.</b> Stores a device type value. This value can be used to differentiate multiple XBee-based products.	CRE	0 - 0xFFFFFFFF [read-only]	0x20000
ZA <sup>2</sup>	<b>ZigBee Application Layer Addressing.</b> Set/read the Zigbee application layer addressing enabled attribute. If enabled, data packets will use the SE, DE, and CI commands to address Zigbee application layer source and destination endpoints, and the cluster ID fields in all data transmissions. ZA is only supported in the AT firmware.	CRE	0 - 1	0
SE <sup>2</sup>	<b>Source Endpoint.</b> Set/read the ZigBee application layer source endpoint value. If ZigBee application layer addressing is enabled (ZA command), this value will be used as the source endpoint for all data transmissions. SE is only supported in AT firmware. The default value 0xE8 (Data endpoint) is the Digi data endpoint	CRE	1 - 0xEF	0xE8

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
DE <sup>2</sup>	<b>Destination Endpoint.</b> Set/read Zigbee application layer destination ID value. If ZigBee application layer addressing is enabled (ZA command), this value will be used as the destination endpoint all data transmissions. DE is only supported in AT firmware. The default value (0xE8) is the Digi data endpoint.	CRE	0 - 0xEF	1 - 0xEF
CI <sup>2</sup>	<b>Cluster Identifier.</b> Set/read Zigbee application layer cluster ID value. If ZigBee application layer addressing is enabled (ZA command), this value will be used as the cluster ID for all data transmissions. CI is only supported in AT firmware. The default value 0x11 (Transparent data cluster ID).	CRE	0 - 0xFF	0x11
BI <sup>2</sup>	<b>Binding Table Index.</b> Set/read the binding table index value. If this value is set to a valid binding table index, the addressing information at that index in the binding table will be used for all data transmissions. BI is only supported in AT firmware	CRE	0 - 0xFF	0xFF

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

2. Command supported by modules using AT Command firmware only

Tabela 8.3 - Comandos AT: Rede

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
CH	<b>Operating Channel.</b> Read the channel number used for transmitting and receiving between RF modules. Uses 802.15.4 channel numbers. A value of 0 means the device has not joined a PAN and is not operating on any channel.	CRE	0, 0x0B-0x1A (XBee) 0, 0x0C – 0x18 (XBee-PRO)	[read-only]
ID	<b>PAN ID.</b> Set/Get the PAN (Personal Area Network) ID. <b>Coordinator</b> - Set the preferred Pan ID. Set ID = 0xFFFF to auto-select. <b>Router / End Device</b> - 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.	CRE	0 - 0x3FFF, 0xFFFF	0x0234 (291d)
BH	<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 - 0x20	0
OP	<b>Operating PAN ID.</b> Read the PAN (Personal Area Network) ID. The OP value reflects the operating PAN ID that the module is running on. If ID < 0xFFFF, OP will equal ID.	CRE	0 - 0x3FFF	[read-only]
NT	<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	0x20 - 0xFF [x 100 msec]	0x3C (60d)
NO	<b>Network Discovery options.</b> Set/Read the options value for the network discovery command. The options bitfield value can change the behavior of the ND (network discovery) command and/or change what optional values are returned in any received ND responses or API node identification frames. Options include: 0x01 = Append DD value (to ND responses or API node identification frames) 002 = Local device sends ND response frame when ND is issued.	CRE	0 - 0x03 [bitfield]	0
ND	<b>Node Discover.</b> Discovers and reports all RF modules found. The following information is reported for each module discovered. MY<CR> SH<CR> SL<CR> NI<CR> (Variable length) PARENT_NETWORK_ADDRESS (2 Bytes)<CR> DEVICE_TYPE<CR> (1 Byte: 0=Coord, 1=Router, 2=End Device) STATUS<CR> (1 Byte: Reserved) PROFILE_ID<CR> (2 Bytes) MANUFACTURER_ID<CR> (2 Bytes) <CR> After (NT * 100) milliseconds, the command ends by returning a <CR>. ND also accepts a Node Identifier (NI) as a parameter (optional). In this case, only a module that matches the supplied identifier will respond. 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. The radius of the ND command is set by the BH command.	CRE	optional 20-Byte NI or MY value	--

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default																
DN	<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. The radius of the DN command is set by the BH command.</p>	CRE	up to 20-Byte printable ASCII string	-																
SC	<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.</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> <p>Note: Setting SC to include more than 12 continuous channels could cause data to be received on incorrect frequencies due to crosstalk issues with the EM250 at certain power levels. See Appendix E for details.</p> <p>Changing SC may result in not being able to communicate with long-range '-PRO' modules from Digi</p>	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	<p><b>XBee</b></p> <p>1 - 0xFFFF [bitfield]</p> <p><b>XBee-PRO</b></p> <p>2 - 0x3FFE [bitfield]</p> <p>(bits 0, 14, 15 not allowed)</p>	0x1FFE
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	<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).</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	<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)																
JV	<p><b>Channel Verification.</b> Set/Read the channel verification parameter. If JV=1, and the network is an open network (NJ=0xFF), a router will verify the coordinator is on its operating channel when joining or coming up from a power cycle. If a coordinator is not detected, the router will leave its current channel and attempt to join a new PAN. If JV=0, the router will continue operating on its current channel even if a coordinator is not detected.</p>	R	0 - Channel verification disabled 1 - Channel verification enabled	0																
AR	<p><b>Aggregate Routing Notification.</b> Set/read time between consecutive aggregate route broadcast messages. If used, AR should be set on only one device to enable many-to-one routing to the device. Setting AR to 0 only sends one broadcast</p>	CR	0 - 0xFF	0xFF																
AI	<p><b>Association Indication.</b> Read information regarding last node join request:</p> <p>0x00 - Successful completion - Coordinator started or Router/End Device found and joined with a parent.</p> <p>0x21 - Scan found no PANs</p> <p>0x22 - Scan found no valid PANs based on current SC and ID settings</p> <p>0x23 - Valid Coordinator or Routers found, but they are not allowing joining (NJ expired)</p> <p>0x27 - Node Joining attempt failed (typically due to incompatible security settings)</p> <p>0x2A - Coordinator Start attempt failed</p> <p>0xFF - Scanning for a Parent</p> <p>0x2B - Checking for an existing coordinator</p>	CRE	0 - 0xFF [read-only]	-																

Tabela 8.4 - Comandos AT: Segurança

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
EE	Encryption Enable. Set/Read the encryption enable setting.	CRE	0 - Encryption disabled 1 - Encryption enabled	0
EO	Encryption Options. Configure options for encryption. Unused option bits should be set to 0. Options include: 0x01 - Send the security key unsecured over-the-air during joins 0x02 - Use trust center	CRE	0 - 0xFF	
KY	Encryption Key. Set the 128-bit AES encryption key. This command is read-only; KY cannot be read.	CRE	0 - 0xFFFFFFFFFFFFFFFF	0

Tabela 8.5 - Comandos AT: Interface RF

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
PL	<b>Power Level.</b> Select/Read the power level at which the RF module transmits conducted power.	CRE	<b>XBee</b> (boost mode disabled) 0 = -8 dBm 1 = -4 dBm 2 = -2 dBm 3 = 0 dBm 4 = +2 dBm  <b>XBee-PRO</b> 4 = 18 dBm <b>XBee-PRO (International Variant)</b> 4 = 10dBm	4
PM	<b>Power Mode.</b> Set/read the power mode of the device. Enabling boost mode will improve the receive sensitivity by 1dB and increase the transmit power by 2dB Note: Enabling boost mode on the XBee-PRO will not affect the output power. Boost mode imposes a slight increase in current draw. See section 1.2 for details.	CRE	0-1, 0 = Boost mode disabled, 1 = Boost mode enabled.	1
DB	<b>Received Signal Strength.</b> This command reports the received signal strength of the last received RF data packet. The DB command only indicates the signal strength of the last hop. It does not provide an accurate quality measurement for a multihop link. DB can be set to 0 to clear it.			

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

Tabela 8.6 - Comandos AT: Interface Serial

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
AP <sup>2</sup>	<b>API Enable.</b> Enable API Mode. The AP parameter is only applicable when using modules that contain the following firmware versions: 1.1xx (coordinator), 1.3xx (router/end device)	CRE	1 - 2 1 = API-enabled 2 = API-enabled (w/escaped control characters)	1
AO <sup>2</sup>	<b>API Options.</b> Configure options for API. Current options select the type of receive API frame to send out the Uart for received RF data packets.	CRE	0 - Default receive API indicators enabled 1 - Explicit Rx data indicator API frame enabled (0x91)	0
BD	<b>Interface Data Rate.</b> 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 - 0x38400 (non-standard rates)	3

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
NB	<b>Serial Parity.</b> Set/Read the serial parity setting on the module.	CRE	0 = No parity 1 = Even parity 2 = Odd parity 3 = Mark parity	0
RO	<b>Packetization Timeout.</b> 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	<b>DIO7 Configuration.</b> Select/Read options for the DIO7 line of the RF module.	CRE	0 = Disabled 1 = CTS Flow Control 3 = Digital input 4 = Digital output, low 5 = Digital output, high 6 = RS-485 transmit enable (low enable) 7 = RS-485 transmit enable (high enable)	1
D6	<b>DIO6 Configuration.</b> Configure options for the DIO6 line of the RF module.	CRE	0 - Disabled 1 - RTS Flow Control	0

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

Tabela 8.7 - Comandos AT: Comandos I/O

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
IS	<b>Force Sample</b> Forces a read of all enabled digital and analog input lines.	CRE	--	--
1S	<b>XBee Sensor Sample.</b> Forces a sample to be taken on an XBee Sensor device. This command can only be issued to an XBee sensor device using an API remote command.	RE	-	-
IR	<b>IO Sample Rate.</b> Set/Read the IO sample rate to enable periodic sampling. For periodic sampling to be enabled, IR must be set to a non-zero value, and at least one module pin must have analog or digital IO functionality enabled (see D0-D8, P0-P2 commands). The sample rate is measured in milliseconds.	CRE	0 - 0xFFFF (ms)	0
IC	<b>IO Digital Change Detection.</b> Set/Read the digital IO pins to monitor for changes in the IO state. IC works with the individual pin configuration commands (D0-D8, P0-P2). If a pin is enabled as a digital input/output, the IC command can be used to force an immediate IO sample transmission when the DIO state changes. IC is a bitmask that can be used to enable or disable edge detection on individual channels. Unused bits should be set to 0. Bit (IO pin): 0 (DIO0) 4 (DIO4) 8 (DIO8) 1 (DIO1) 5 (DIO5) 9 (DIO9) 2 (DIO2) 6 (DIO6) 10 (DIO10) 3 (DIO3) 7 (DIO7) 11 (DIO11)	CRE	: 0 - 0xFFFF	0
P0	<b>PWM0 Configuration.</b> Select/Read function for PWM0.	CRE	0 = Disabled 1 = RSSI PWM 3 - Digital input, monitored 4 - Digital output, default low 5 - Digital output, default high	1
P1	<b>DIO11 Configuration.</b> Configure options for the DIO11 line of the RF module.	CRE	0 - Unmonitored digital input 3- Digital input, monitored 4- Digital output, default low 5- Digital output, default high	0
P2	<b>DIO12 Configuration.</b> Configure options for the DIO12 line of the RF module.	CRE	0 - Unmonitored digital input 3- Digital input, monitored 4- Digital output, default low 5- Digital output, default high	0

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
P3	<b>DIO13 Configuration.</b> Set/Read function for DIO13. This command is not yet supported.	CRE	0, 3-5 0 – Disabled 3 – Digital input 4 – Digital output, low 5 – Digital output, high	
D0	<b>AD0/DIO0 Configuration.</b> Select/Read function for AD0/DIO0.	CRE	0-5 0 – Disabled 1 - Node identification button enabled 2 - Analog input, single ended 3 – Digital input 4 – Digital output, low 5 – Digital output, high	1
D1	<b>AD1/DIO1 Configuration.</b> Select/Read function for AD1/DIO1.	CRE	0, 2-5 0 – Disabled 2 - Analog input, single ended 3 – Digital input 4 – Digital output, low 5 – Digital output, high	0
D2	<b>AD2/DIO2 Configuration.</b> Select/Read function for AD2/DIO2.	CRE	0, 2-5 0 – Disabled 2 - Analog input, single ended 3 – Digital input 4 – Digital output, low 5 – Digital output, high	0
D3	<b>AD3/DIO3 Configuration.</b> Select/Read function for AD3/DIO3.	CRE	0, 2-5 0 – Disabled 2 - Analog input, single ended 3 – Digital input 4 – Digital output, low 5 – Digital output, high	0
D4	<b>DIO4 Configuration.</b> Select/Read function for DIO4.	CRE	0, 3-5 0 – Disabled 3 – Digital input 4 – Digital output, low 5 – Digital output, high	0
D5	<b>DIO5 Configuration.</b> Configure options for the DIO5 line of the RF module.	CRE	0 = Disabled 1 = Associated indication LED 3 = Digital input 4 = Digital output, default low 5 = Digital output, default high	1
LT	<b>Assoc LED Blink Time.</b> Set/Read the Associate LED blink time. If the Associate LED functionality is enabled (D5 command), this value determines the on and off blink times for the LED when the module has joined a network. If LT=0, the default blink rate will be used (500ms coordinator, 250ms router/end device). For all other LT values, LT is measured in 10ms.	CRE	0x14 - 0xFF (200 - 2550 ms)	0
D8	<b>DIO8 Configuration.</b> Set/Read function for DIO8. This command is not yet supported.	CRE	0, 3-5 0 – Disabled 3 – Digital input 4 – Digital output, low 5 – Digital output, high	

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
PR	Set/read the bit field that configures the internal pull-up resistor status for the I/O lines. "1" specifies the pull-up resistor is enabled. "0" specifies no pullup.(30k pull-up resistors) Bits: 0 - DIO4 (Pin 11) 1 - AD3 / DIO3 (Pin 17) 2 - AD2 / DIO2 (Pin 18) 3 - AD1 / DIO1 (Pin 19) 4 - AD0 / DIO0 (Pin 20) 5 - RTS / DIO6 (Pin 16) 6 - DTR / Sleep Request / DIO8 (Pin 9) 7 - DIN / Config (Pin 3) 8 - Associate / DIO5 (Pin 15) 9 - On/Sleep / DIO9 (Pin 13) 10 - DIO12 (Pin 4) 11 - PWM0 / RSSI / DIO10 (Pin 6) 12 - PWM1 / DIO11 (Pin 7)	CRE	0 - 0x1FFF	0 - 0x1FFF
RP	<b>RSSI PWM Timer.</b> 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)
CB	<b>Commissioning Pushbutton.</b> This command can be used to simulate commissioning button presses in software. The parameter value should be set to the number of button presses to be simulated. For example, sending the ATCB1 command will execute the action associated with 1 commissioning button press. (See D0 command).	CRE		

Tabela 8.8 - Comandos AT: Diagnóstico

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
VR	<b>Firmware Version.</b> Read firmware version of the module.	CRE	0 - 0xFFFF [read-only]	Factory-set
HV	<b>Hardware Version.</b> Read hardware version of the module.	CRE	0 - 0xFFFF [read-only]	Factory-set
%V	<b>Supply Voltage.</b> Reads the voltage on the Vccpin. To convert the reading to a mV reading, divide the read value by 1023 and multiply by 1200. A %V reading of 0x8FE (2302 decimal) represents 2700mV or 2.70V.	CRE	-	-

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

Tabela 8.9 - Comandos AT: Sleep Commands

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
SM	<b>Sleep Mode</b> Sets the sleep mode on the RF module	RE	0-Sleep disabled 1-Pin sleep enabled 4-Cyclic sleep enabled Note: When SM=0, the device operates as a router. When SM changes to a non-zero value, the router leaves the network and rejoins as an end device. <b>Only end devices can sleep</b>	0
SN	<b>Number of Sleep Periods.</b> Sets the number of sleep periods to not assert the On/Sleep pin on wakeup if no RF data is waiting for the end device. This command allows a host application to sleep for an extended time if no RF data is present	RE	1 - 0xFFFF	1
SP	<b>Sleep Period.</b> This value determines how long the end device will sleep at a time, up to 28 seconds. (The sleep time can effectively be extended past 28 seconds using the SN command.) On the parent, this value determines how long the parent will buffer a message for the sleeping end device. It should be set at least equal to the longest SP time of any child end device.	CRE	0x20 - 0xAF0 x 10ms (Quarter second resolution)	0x20
ST	<b>Time Before Sleep</b> Sets the time before sleep timer on an end device. The timer is reset each time serial or RF data is received. Once the timer expires, an end device may enter low power operation. Applicable for cyclic sleep end devices only.	RE	1 - 0xFFFE (x 1ms)	0x1388 (5 seconds)
SO Command	<b>Sleep Options.</b> Configure options for sleep. Unused option bits should be set to 0. Sleep options include: 0x02 - Always wake for ST time 0x04 - Sleep entire SN * SP time Sleep options should not be used for most applications. See Sleep Mode chapter for more information.	E	0 - 0xFF	0