

**UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO TÉCNICO INDUSTRIAL DE SANTA MARIA
CURSO SUPERIOR DE TECNOLOGIA EM REDES DE
COMPUTADORES**

**REDE DE SENSORES DE VAZAMENTO DE GÁS
PARA TUBULAÇÕES SUBTERRÂNEAS COM
TECNOLOGIA DE COMUNICAÇÃO SEM FIO**

TRABALHO DE CONCLUSÃO DE CURSO

Fabio Lunardi Antunes

Santa Maria, RS, Brasil

2012

TCC/CSTRC/UFSM,RS

ANTUNES, Fabio Lunardi

Redes de Sensores de Vazamento de Gás com ZigBee 2012

**REDE DE SENSORES DE VAZAMENTO DE GÁS PARA
TUBULAÇÕES SUBTERRÂNEAS COM TECNOLOGIA DE
COMUNICAÇÃO SEM FIO**

Fabio Lunardi Antunes

Trabalho de Conclusão de Curso (TCC) do Curso Superior de Tecnologia em
Redes de Computadores, da Universidade Federal de Santa Maria (UFSM,RS).

Orientador: Prof. Dr. Claiton Pereira Colvero

Santa Maria, RS, Brasil

2012

**UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO TÉCNICO INDUSTRIAL DE SANTA MARIA
CURSO SUPERIOR DE TECNOLOGIA EM REDES DE
COMPUTADORES**

A Comissão Examinadora, abaixo assinada, aprova o Trabalho de Conclusão de
Curso

**REDE DE SENSORES DE VAZAMENTO DE GÁS PARA
TUBULAÇÕES SUBTERRÂNEAS COM TECNOLOGIA DE
COMUNICAÇÃO SEM FIO**

elaborado por
Fabio Lunardi Antunes

COMISSÃO EXAMINADORA

Claiton Pereira Colvero, Dr.
(Presidente/Orientador)

Eugênio de Oliveira Simonetto, Dr. (UFSM)

Walter Priesnitz Filho, Ms. (UFSM)

Santa Maria, 06 de julho de 2012.

DEDICATÓRIA

Dedico este trabalho a minha família. A minha esposa Rosirene Iracet dos Santos Antunes e a minha filha Camille Iracet Antunes, pela compreensão e carinho em todas as horas. Aos meus pais Jorge Antonio de Lima Antunes e Adeli Maria Lunardi Antunes pela minha vida e pela educação moral e escolar. A minha irmã Elisandra Lunardi Antunes pelo incentivo.

AGRADECIMENTOS

Universidade Federal de Santa Maria – pela qualidade de ensino público e gratuito;

Docentes do Curso Superior de Tecnologia em Redes de Computadores – pelo conhecimento passado e atenção prestada;

Claiton Pereira Colvero – pela orientação, apoio e confiança;

Colegas do Curso Superior de Tecnologia em Redes de Computadores – pela amizade cultivada e conhecimento repassado;

I-Dutto – pelo apoio financeiro;

A todas as pessoas que aqui não foram citadas, mas que de algum modo contribuíram para o desenvolvimento do trabalho;

A Deus – pela saúde e proteção e por ter me dado a oportunidade de aprender um pouco mais.

EPÍGRAFE

Não encontre um defeito, encontre uma solução.

(HENRY FORD)

RESUMO

**TRABALHO DE CONCLUSÃO DE CURSO
CURSO SUPERIOR DE TECNOLOGIA EM REDES DE COMPUTADORES
UNIVERSIDADE FEDERAL DE SANTA MARIA**

REDE DE SENSORES DE VAZAMENTO DE GÁS PARA TUBULAÇÕES SUBTERRÂNEAS COM TECNOLOGIA DE COMUNICAÇÃO SEM FIO

AUTOR: FABIO LUNARDI ANTUNES

ORIENTADOR: CLAITON PEREIRA COLVERO

Data e Local da Defesa: Santa Maria, 06 de julho de 2012.

Este trabalho apresenta o estudo, desenvolvimento e implantação de uma solução para detecção de vazamento de gás natural em tubulações subterrâneas com tecnologia de comunicação sem fio, através do padrão IEEE 802.15.4 (ZigBee), utilizando hardware e software dedicados. O trabalho contempla um sistema eficiente e autônomo de alarme em tempo real de eventuais vazamentos de gás natural em diferentes pontos da rede de distribuição subterrânea de gás, capaz de interligar todos os sensores da região metropolitana de uma grande cidade a uma central de controle e monitoramento, utilizando modernos recursos de rede mesh sem fio.

Palavras-Chave: GÁS NATURAL. IEEE 802.15.4. ZIGBEE.

ABSTRACT

COMPLETION OF COURSE WORK
SUPERIOR OF TECHNOLOGY COURSE IN COMPUTER NETWORKS
FEDERAL UNIVERSITY OF SANTA MARIA

GAS LEAK SENSORS NETWORK FOR UNDERGROUND PIPELINES USING WIRELESS NETWORK TECHNOLOGIES

AUTHOR: FABIO LUNARDI ANTUNES

ADVISER: CLAITON PEREIRA COLVERO

Place and date of the defense: Santa Maria, July 06, 2012.

This work presents the study, development and implementation of a solution to detect natural gases leak in underground pipelines using wireless network technologies through the IEEE 802.15.4 (ZigBee), using dedicated hardware and software. The work includes an autonomous and efficient real time system of casual natural gas leak on different points of underground distribution gas network, able to connect all sensors from a metropolitan city area to a monitoring and control center, using modern resources of wireless mesh networks.

Keywords: NATURAL GAS. IEEE 802.15.4. ZIGBEE.

LISTA DE ILUSTRAÇÕES

Figura 1 – Mapa dos gasodutos no Brasil.	17
Figura 2 – Mapa das explosões de bueiros no Rio de Janeiro.	18
Figura 3 – Bueiro após explosão.	19
Figura 4 – Comparação entre as tecnologias sem fio.	22
Figura 5 – Canais de Operação do ZigBee.	24
Figura 6 – Pilha do protocolo ZigBee.	25
Figura 7 – Pacote MAC com <i>bit</i> de segurança ativado.	29
Figura 8 – Topologias de redes ZigBee.	33
Figura 9 – Estrutura do <i>Superframe</i>	35
Figura 10 – Beacon <i>Frame</i>	39
Figura 11 – <i>Data Frame</i>	40
Figura 12 – <i>Acknowledgment Frame</i>	40
Figura 13 – MAC <i>Command Frame</i>	41
Figura 14 – Tipos de antenas dos módulos XBee.	43
Figura 15 – Interface do programa X-CTU.	45
Figura 16 – Formato dos comandos AT.	46
Figura 17 – <i>Frame</i> do modo API.	47
Figura 18 – Configuração do sensor de gás IDG-193.	48
Figura 19 – Circuito para teste com NTC.	50
Figura 20 – Circuito completo do sensor de gás.	52
Figura 21 – Vista externa da prova de conceito do sensor de gás.	53
Figura 22 – Vista interna da prova de conceito do sensor de gás.	53
Figura 23 – Vista externa do protótipo.	54
Figura 24 – Vista externa do sensor de gás.	54
Figura 25 – Refil interno do protótipo.	55
Figura 26 – Protótipo desmontado.	55
Figura 27 – Vista externa do <i>gateway</i>	57
Figura 28 – Vista interna do <i>gateway</i>	57
Figura 29 – Interface web do <i>gateway</i>	59
Figura 30 – Dados sem vazamento de gás.	63
Figura 31 – Dados sem vazamento de gás.	64

LISTA DE QUADROS

Quadro 1 – Comparativo das características das tecnologias sem fio.....	23
Quadro 2 – Funções dos dispositivos ZigBee.	31
Quadro 3 – Características dos Módulos XBee e XBee PRO.	42
Quadro 4 – Pinagem do módulo XBee.....	44
Quadro 5 – Temperatura e resistência do NTC.	49
Quadro 6 – Resultados dos testes com NTC.	50
Quadro 7 – Características do ConnectPort X4.....	58
Quadro 8 – Configurações dos módulos.	62
Quadro 9 – Valores de tensão no pino 19 do XBee.	63

LISTA DE ABREVIATURAS E SIGLAS

1xRTT	<i>Radio Transmission Technology</i>
2G	Telefonia móvel de segunda geração
3DES	<i>Triple Data Encryption Standard</i>
3G	Telefonia móvel de terceira geração
4G	Telefonia móvel de quarta geração
ABEGÁS	Associação Brasileira das Empresas Distribuidoras de Gás Canalizado
ACK	<i>Acknowledge</i>
A/D	Analógico/Digital
ADC	<i>Analog to Digital Converter</i>
ADDP	<i>Advanced Digi Discovery Protocol</i>
AES	<i>Advanced Encryption Standard</i>
AF	<i>Application Framework</i>
ANP	Agência Nacional do Petróleo, Gás Natural e Biocombustíveis
AODV	<i>Ad Hoc On-Demand Distance Vector</i>
API	<i>Application Programming Interface</i>
APS	<i>Application Support Sub-layer</i>
as built	Como construído
BD	<i>Baud Rate</i>
BH	<i>Broadcast Hope</i>
BPSK	<i>Binary Phase Shift Keying</i>
BTR	<i>Broadcast Transaction Record</i>
BTT	<i>Broadcast Transaction Table</i>
CAP	<i>Contention Access Period</i>
CEG	Companhia Estadual de Distribuição de Gás do Rio De Janeiro
CFP	<i>Contention Free Period</i>
°C	Graus Celsius
CCK	<i>Complementary Code Keying</i>
CDMA	<i>Code Division Multiple Access</i>
CSMA-CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
CTS	<i>Clear to Send</i>
DBPSK	<i>Differential Binary Phase Shift Keying</i>
DCD	<i>Data Carrier Detect</i>
DES	<i>Data Encryption Standard</i>
DH	<i>Destination Address High</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
DL	<i>Destination Address Low</i>
DOU	Diário Oficial da União
DQPSK	<i>Differential Quadrature Phase Shift Keying</i>
DSSS	<i>Direct Sequence Spread Spectrum</i>
DTR	<i>Data Terminal Ready</i>
EE	<i>Encryption Enable</i>
EV-DO	<i>Evolution Data Optimized</i>
EDGE	<i>Enhanced Data rates for GSM Evolution</i>
EUA	Estados Unidos da América
FCS	<i>Frame Check Sequence</i>
FFD	<i>Full Function Device</i>

GHz	Giga Hertz
GRE	<i>Generic Routing Protocol</i>
GSM	<i>Groupe Special Mobile</i>
GTS	<i>Guaranteed Time Slot</i>
HR-WPAN	<i>High Rate Wireless Personal Area Network</i>
HSPA	<i>High Speed Packet Access</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPS	<i>Hypertext Transfer Protocol Secure</i>
IEEE	Institute of Electrical and Electronic Engineers
INMETRO	Instituto Nacional de Metrologia, Normalização e Qualidade Industrial
IP	<i>Internet Protocol</i>
IPsec ESP	<i>IP Security Protocol Encapsulating Security Payload</i>
ISM	<i>Industrial, Scientific and Medical</i>
kB	Kilo Bytes
kbps	Kilo bits por segundo
km	Quilômetros
kΩ	Kilo Ohms
LDO	<i>Low Dropout</i>
LPD	<i>Line Printer Daemon Protocol</i>
Ltda	Limitada
LR-WPAN	<i>Low Rate Wireless Personal Area Network</i>
m	Metros
mA	Mili Ampère
μA	Micro Ampère
MAC	<i>Media Access Control</i>
Mbps	Mega bits por segundo
MHz	Mega Hertz
MIC	<i>Message Integrity Code</i>
MIMO	<i>Multiple Input and Multiple Output</i>
MR-WPAN	<i>Medium Rate Wireless Personal Area Network</i>
ms	Mili segundos
mW	Mili Watts
NI	<i>Node Identifier</i>
NJ	<i>Node Join Time</i>
NT	<i>Node Discover Time</i>
NMS	<i>Network Management System</i>
NTC	<i>Negative Temperature Coefficient</i>
NWR	<i>Network – Camada de Rede</i>
O-QPSK	<i>Offset Quadrature Phase Shift Keying</i>
OSI	<i>Open Systems Interconnection</i>
OUI	<i>Organizationally Unique Identifier</i>
PAN	<i>Personal Area Network</i>
PAN ID	<i>Personal Area Network Identifier</i>
PDU	<i>Protocol Data Unit</i>
PHY	Camada Física
PL	<i>Transmit Power Level</i>
PM	<i>Power Mode</i>
ppm	Partes por milhão
PWM	<i>Pulse Width Modulation</i>
QAM	<i>Quadrature Amplitude Modulation</i>

QPSK	<i>Quadrature Phase Shift Keying</i>
RF	<i>Radiofrequência</i>
RFD	<i>Reduced Function Device</i>
RTDT	<i>Reglamento Técnico de Dutos Terrestres</i>
RTS	<i>Request to Send</i>
RX	<i>Receptor</i>
SC	<i>Scan Channel</i>
SD	<i>Superframe Duration</i>
SM	<i>Sleep Mode</i>
SMS	<i>Short Message Service</i>
SNMP	<i>Simple Network Management Protocol</i>
SOFDMA	<i>Scalable Orthogonal Frequency Division Multiplexing Access</i>
SSH	<i>Secure Shell</i>
SSL	<i>Secure Sockets Layer</i>
TCP	<i>Transmission Control Protocol</i>
TLS	<i>Transport Layer Security</i>
TTL	<i>Time to Live</i>
UART	<i>Universal Asynchronous Receiver Transmitter</i>
UDP	<i>User Datagram Protocol</i>
UL/DL	<i>Upload/Download ratio, Uplink/Downlink ratio</i>
USB	<i>Universal Serial Bus</i>
V	<i>Volts</i>
Vcc	<i>Tensão Contínua</i>
VPN	<i>Virtual Private Network</i>
WEP	<i>Wired Equivalent Privacy</i>
WLAN	<i>Wireless Local Area Network</i>
WMAN	<i>Wireless Metropolitan Area Network</i>
WPA	<i>Wireless Protected Access</i>
WPA2	<i>Wireless Protected Access 2</i>
WPAN	<i>Wireless Personal Network</i>
WWAN	<i>Wireless Wide Area Network</i>
ZC	<i>ZigBee Coordinator</i>
ZDO	<i>ZigBee Device Objects</i>
ZED	<i>ZigBee End Device</i>
ZR	<i>ZigBee Router</i>

APÊNDICES

APÊNDICE A – PROGRAMA PYTHON	69
APÊNDICE B – RESULTADOS DO PROGRAMA COM 1 SENSOR.....	72
APÊNDICE C – RESULTADOS DO PROGRAMA COM 2 SENSORES.....	74

SUMÁRIO

1 INTRODUÇÃO	16
2 ZIGBEE	21
2.1 Histórico	21
2.2 Classificação e Comparação das WPANs	21
2.3 Características do ZigBee	23
2.4 Camadas ZigBee	24
2.4.1 Camada Física (PHY)	25
2.4.2 Camada de Controle de Acesso ao Meio (MAC)	25
2.4.3 Camada de Rede (NWK)	26
2.4.4 Camada de Aplicação	27
2.4.4.1 Subcamada de Suporte à Aplicação (APS)	27
2.4.4.2 <i>ZigBee Device Objects</i> (ZDO)	27
2.4.4.3 <i>Application Framework</i>	28
2.5 Segurança no ZigBee	28
2.6 Dispositivos Físicos ZigBee	29
2.6.1 Dispositivos de Funções Completas (FFD)	29
2.6.2 Dispositivo de Funções Reduzidas (RFD)	30
2.7 Dispositivos Lógicos ZigBee	30
2.7.1 Coordenador ZigBee(ZC)	30
2.7.2 Roteador ZigBee (ZR)	30
2.7.3 Dispositivo Final ZigBee (ZED)	31
2.8 Topologias das Redes ZigBee	32
2.8.1 Estrela (<i>Star</i>)	32
2.8.2 Árvore (<i>Cluster Tree</i>)	32
2.8.3 Malha (<i>Mesh</i>)	32
2.9 Formação da Rede ZigBee	33
2.10 Modos de Operação da Rede ZigBee	34
2.10.1 Modo <i>Non-Beaconing</i>	34
2.10.2 Modo <i>Beaconing</i>	34
2.11 Tipos de Mensagens na Rede ZigBee	35
2.11.1 Mensagens <i>Broadcast</i>	35
2.11.2 Mensagens <i>Multicast</i>	36
2.11.3 Mensagens <i>Unicast</i>	36
2.12 Endereçamento ZigBee	36
2.12.1 Endereço de 16 <i>Bits</i>	36
2.12.2 Endereço de 64 <i>Bits</i>	37
2.13 Roteamento ZigBee	37
2.13.1 Roteamento Hierárquico ou em Árvore	37
2.13.2 Roteamento Sob Demanda	38
2.14 Frames ZigBee	38
2.4.2.1 <i>Beacon Frame</i>	39
2.4.2.2 <i>Data Frame</i>	39
2.4.2.3 <i>Acknowledgment Frame</i>	40
2.4.2.4 <i>MAC Command Frame</i>	41
3 COMPONENTES DO SENSOR DE GÁS	42
3.1 XBee	42

3.1.1 Programa X-CTU	44
3.1.2 Modos de Comando do XBee.....	45
3.1.2.1 Modo de Comando AT	46
3.1.2.1.1 Formato dos Comandos AT.....	46
3.1.2.1.2 Modo de Comando API.....	47
3.2 Sensor de Gás	48
3.3 Sensor de Temperatura.....	49
3.4 Regulador <i>Low Dropout</i> (LDO).....	51
3.5 Circuito do Sensor de Gás	51
4 GATEWAY.....	56
5 DESENVOLVIMENTO E ANÁLISE DOS RESULTADOS.....	60
5.1 Programação em Python.....	60
5.2 Configuração dos Módulos	60
5.3 Análise dos Dados	62
6 CONSIDERAÇÕES FINAIS.....	65
REFERÊNCIAS BIBLIOGRÁFICAS	66

1 INTRODUÇÃO

Para que um país se desenvolva devem ser observados vários fatores, entre eles está a sua capacidade de prover meios de geração de energia. Esta energia permite que fábricas e indústrias sejam construídas ou ampliadas, acompanhando a revolução tecnológica. No Brasil não é diferente. Ele vive um período de crescimento econômico, com isto investimentos estão sendo realizados em diversas áreas.

A utilização do gás natural no Brasil teve início na década de 1940 com o objetivo de atender as indústrias do Recôncavo Baiano e atualmente o seu consumo tem aumentado consideravelmente em todo o país com a descoberta da Bacia de Campos, Pré-Sal e gasoduto Brasil-Bolívia. Após os apagões de 2001 e 2002, o Governo Federal decidiu buscar novas alternativas de geração de energia.

Uma das fontes de energia que está em expansão é o gás natural, que é utilizado na geração de energia, propulsão veicular, aquecimento de água, alimentos e etc. Segundo a Associação Brasileira das Empresas Distribuidoras de Gás Canalizado (ABEGÁS), o consumo de gás natural no Brasil encerrou o primeiro bimestre de 2012 com expressivo crescimento, tanto ao comparar com o mesmo período de 2011 como com o mês anterior. Um levantamento feito aponta que foi consumido diariamente, em média, 49,2 milhões de metros cúbicos de gás natural, a rede de distribuição soma 21.157,2 km de extensão e há 2.076.155 clientes em todos os segmentos de atuação das distribuidoras. A figura 1 mostra um mapa dos gasodutos no Brasil.

Este crescimento fez com que a preocupação com acidentes aumentasse, pois o gás natural apresenta riscos de asfixia, incêndios e explosões. Os incidentes e acidentes com gás natural são causados principalmente pela movimentação no terreno, intervenções no subsolo e problemas de instalação. Os sinistros envolvendo gás natural ocasionam vazamento nas redes de distribuição que culminam em acidentes contra a integridade física dos cidadãos, alguns são fatais, e transtornos no trânsito de veículos em espaços urbanos.

As tubulações enterradas de gás natural estão presentes no subsolo, que é um ambiente compartilhado e de concessão pública. Em grandes centros urbanos este ambiente encontra-se saturado por redes de água, esgoto, energia, telecomunicações e gás natural. Em um centro urbano, este ambiente sofre constantes intervenções e movimentações do solo, tanto para construção ou manutenção de redes como pelo trânsito de veículos de grande porte.



Figura 1 – Mapa dos gasodutos no Brasil.
Fonte: ABEGÁS.

Atualmente o mapa destas tubulações, que são conhecidos como *as built*¹, apresenta como agravante, que as companhias não conseguem manter os mapas das tubulações atualizados ou até não existem oficialmente catalogados. Funcionários de outras prestadoras de serviço que possuem sua rede de distribuição no subsolo não sabem onde se encontram as tubulações das outras distribuidoras, como por exemplo, a de gás natural. Com isto, estas intervenções podem vir a danificar as tubulações que são desconhecidas e gerar vazamento de fluídos no subsolo. Outro problema em relação aos vazamentos é a grande pressão em que o gás natural é distribuído.

Para que um vazamento de gás natural seja contido, deve-se bloquear a alimentação da tubulação com problema e realizar o reparo da mesma ou de seu componente defeituoso, porém estes vazamentos devem ser previamente identificados de forma segura e localizados precisamente para a tomada de ações corretivas antes de causar um acidente.

¹ Como construído.



Figura 3 – Bueiro após explosão.
Fonte: Último Segundo IG.

Devido aos constantes acidentes causados pelo vazamento de gás natural em tubulações subterrâneas, novos métodos de prevenção devem ser estudados e implementados a fim de aumentar a segurança da população e consequentemente diminuir os transtornos causados por estes vazamentos. De acordo com a Resolução da Agência Nacional do Petróleo, Gás Natural e Biocombustíveis (ANP) nº 6, de 3/2/2011, publicada no Diário Oficial da União (DOU) de 7/2/2011, e com o Regulamento Técnico de Dutos Terrestres (RTDT) da ANP nº 2/2011, os dutos de hidrocarbonetos devem ter Sistemas de Detecção de Vazamento instalados até fevereiro de 2013, assim este trabalho mostra um meio confiável e robusto de detecção de vazamento de gás natural em tubulações subterrâneas, utilizando redes sem fio, devido ao subsolo já estar saturado, utilizando o padrão IEEE 802.15.4 que oferece as características necessárias para a implementação do estudo.

O padrão IEEE 802.15.4 define as camadas *PHY*² e *MAC*³ dos dispositivos designados como ZigBee, que operam com baixo consumo de energia e custo, alta segurança e confiabilidade, permitindo comunicações robustas que trabalhem na faixa de frequência conhecida como *Industrial, Scientific and Medical (ISM)*, não necessitando de licença para funcionamento. Este padrão oferece uma excelente imunidade contra interferências e possui a

² Camada Física.

³ Camada de Controle de Acesso ao Meio.

capacidade de hospedar até 65.535 dispositivos em uma única rede, com taxas de transferências de dados variando de 20 kbps a 250 kbps.

Este trabalho foi desenvolvido em três fases distintas, sendo dividido em projeto e simulação, prototipagem e ensaios em laboratório e em campo. Dentro da fase de projeto e simulação, no período inicial foi avaliado um problema real proposto pela empresa I-Dutto – Soluções em Localização e Identificação Eletrônica Ltda., sediada na cidade do Rio de Janeiro – RJ.

Na sequência do desenvolvimento, a próxima etapa realizada foi à especificação e posterior aquisição de materiais e equipamentos selecionados para uso no trabalho. Na fase de prototipagem, foi realizada a confecção e montagem dos primeiros sensores e a definição de fluxograma de processos de monitoramento e controle, para o desenvolvimento do software dedicado a ser utilizado na central de tomada de decisão do sistema. Como última fase, foram realizados os primeiros testes de laboratório, em ambiente *indoor*, e após executados os testes práticos finais, em ambiente externo, com a aquisição das informações relevantes obtidos durante os ensaios.

A empresa se propôs a financiar todo o material e o treinamento necessários para o desenvolvimento do projeto, em troca da implementação de um sistema que atenda à necessidade de uma demanda proposta pelo Inmetro, que é o colaborador do projeto. A partir do funcionamento do sistema, que será instalado e testado na cidade do Rio de Janeiro, na rede de distribuição da Companhia Estadual de Distribuição de Gás do Rio De Janeiro (CEG), a empresa, caso julgue necessário, fará a adaptação do trabalho para o uso comercial.

O trabalho está estruturado em 6 capítulos. O capítulo 2 descreve o protocolo ZigBee, suas características e funcionamento. O capítulo 3 lista os componentes utilizados para a montagem do circuito do sensor de gás. O capítulo 4 mostra as especificações e funcionalidades do *gateway* utilizado para comunicação com os sensores de gás. O programa preliminar em Python, as configurações dos módulos XBee para uso nos sensores e a análise dos dados obtidos em testes são demonstrados no capítulo 5 e no capítulo 6 temos as considerações finais do trabalho.

2 ZIGBEE

2.1 Histórico

As redes sem fio conhecidas como *Wireless Local Area Network* (WLAN), *Wireless Metropolitan Area Network* (WMAN) e *Wireless Wide Area Network* (WWAN) foram projetadas para usuários finais de pequenas, médias e grandes empresas para transferências de grandes volumes de dados e voz com altas taxas de transmissão.

As *Wireless Personal Network* (WPAN) foram planejadas para interligar dispositivos a uma pequena distância, como o Bluetooth, e usar baixas taxas de transmissão. Com o desenvolvimento de novos produtos e tecnologias, foi necessário desenvolver um padrão de WPAN que contemplasse a aplicação em redes industriais. Estas redes necessitam de características como a baixa latência, baixo consumo de energia, possibilidade de interligar um grande número de dispositivos e segurança. Assim em 2002, Philips, Motorola, Honeywell, Invensys, Samsung e Mitsubishi Electric se reuniram em um consórcio chamado ZigBee Alliance e juntamente com o *Institute of Electrical and Electronic Engineers* (IEEE), criaram o padrão ZigBee IEEE 802.15.4.

Até o lançamento do ZigBee não havia no mercado nenhum padrão globalmente aceito para sensores e dispositivos de controle usados em redes industriais. O objetivo da ZigBee Alliance no desenvolvimento do ZigBee foi desenvolver um padrão de baixo custo, baixo consumo de energia, seguro, robusto, confiável e com a plataforma normalizada aberta.

2.2 Classificação e Comparação das WPANs

As redes WPANs podem ser classificadas em HR-WPAN, MR-WPAN e LR-WPAN.

A *High Rate* WPAN caracteriza-se pela alta taxa de transmissão, temos como exemplo os equipamentos de transmissão de imagem.

A *Medium Rate* WPAN caracteriza-se pela média taxa de transmissão e temos como exemplo o Bluetooth.

A *Low Rate* WPAN caracteriza-se pela baixa taxa de transmissão e é onde se encaixa a tecnologia ZigBee.

Na figura 4 podemos visualizar um comparativo entre as tecnologias sem fio quanto as suas taxas de transmissão e alcance.

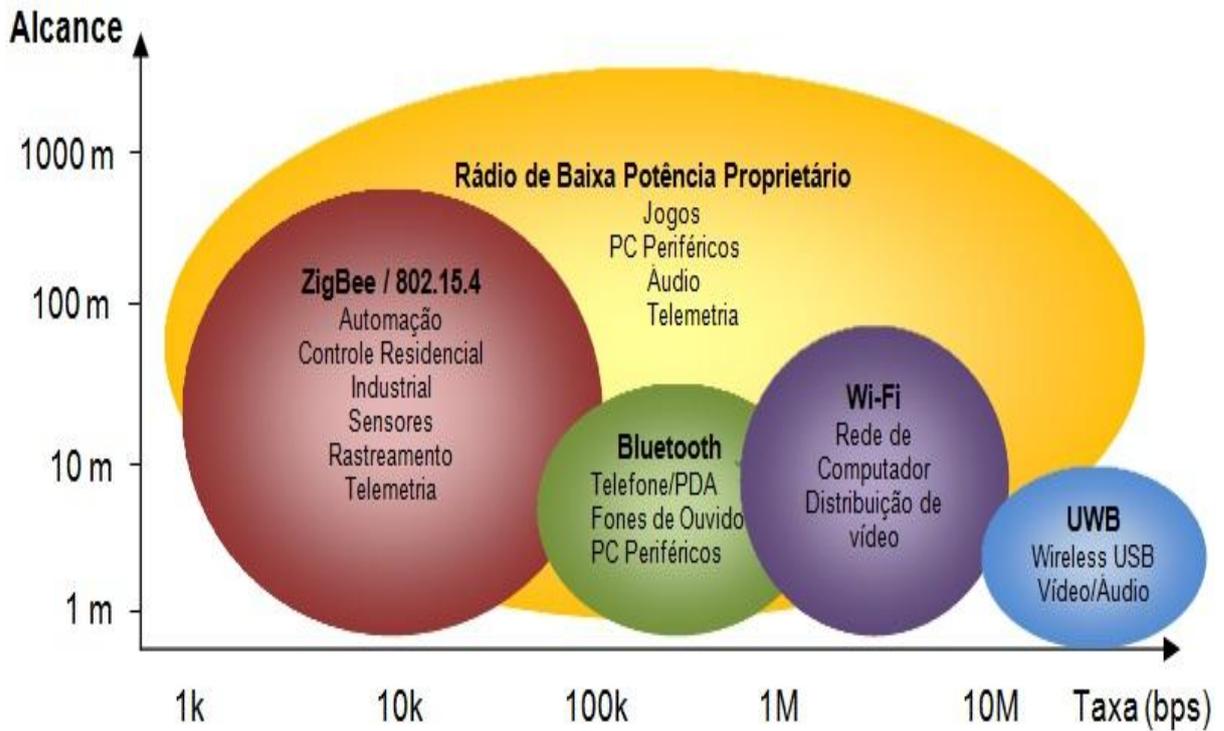


Figura 4 – Comparação entre as tecnologias sem fio.
Fonte: Adaptado de EE Times.

O quadro 1 mostra um comparativo das características das tecnologias sem fio.

(continua)

Característica	ZigBee	Bluetooth	UWB	Wi-Fi
Padrão IEEE	802.15.4	802.15.1	802.15.3	IEEE 802.11 a/b/g/n
Taxa de transferência (máxima)	250 kbps	1 Mbps	115 kbps até 4 Mbps	300 Mbps

(conclusão)

Característica	ZigBee	Bluetooth	UWB	Wi-Fi
Alcance	Até 1,6 km	10 m	10 m	100 m
Frequência	868/915 MHz 2,4GHz	2,4GHz	3,1 – 10,6 GHz	2,4GHz e 5 GHz
Encriptação	AES	EQ <i>Stream</i> <i>cipher</i>	AES	WEP, WPA/2, AES
Número de nós	> 65.000	8	8	2007

Quadro 1 – Comparativo das características das tecnologias sem fio.
Fonte: Adaptado de (ZUCATO, 2009, p. 27).

2.3 Características do ZigBee

O ZigBee é um padrão de protocolo desenvolvido pela ZigBee Alliance que se baseia na IEEE 802.15.4, projetado para permitir comunicação sem fio segura, com baixo consumo de energia e baixo custo.

Utiliza para a sua comunicação entre os dispositivos de uma rede a faixa de frequência ISM, que não precisa de licença para funcionamento. Nos EUA funciona na faixa de 915 MHz, com 10 canais e velocidade de 40 kbps. Na Europa em 868 MHz, 1 canal e velocidade de 20 kbps. No Brasil e no resto do mundo funciona em 2,4 GHz, com 16 canais e velocidade de 250 kbps. Tem a possibilidade de hospedar até 65.535 endereços por canal.

Com relação ao consumo de energia, o módulo ZigBee consome apenas 50 mA de corrente quando está ativo e menos de 10 μ A quando opera no modo *sleep*⁴.

Os dispositivos ZigBee podem ter uma topologia de rede ponto-a-ponto, ponto-a-multiponto, malha e *peer-to-peer*⁵. Um módulo coordenador pode controlar até 254 módulos escravos e a rede pode trabalhar com até 65.536 elementos.

Outras vantagens do protocolo ZigBee IEEE 802.15.4 são: possibilidade de configuração automática da rede, endereçamento dinâmico de módulos escravos, controle total por *handshaking*⁶ nas transferências de pacotes, garantindo maior integridade dos dados e possibilidade de utilização de criptografia Advanced Encryption Standard (AES) de 128 *bits*.
(BAUMANN, 2006 apud COUTO; FIGUEIREDO, 2008, p. 4).

⁴ O dispositivo fica num estado de baixo consumo de energia.

⁵ Par-a-par.

⁶ Troca de mensagens de controle.

Na banda de 2,4 GHz é utilizada a modulação *Offset Quadrature Phase Shift Keying* (O-QPSK) e para as bandas de 868 e 915 MHz é utilizado a modulação *Binary Phase Shift Keying* (BPSK)⁷.

A figura 5 mostra os canais de operação do ZigBee.

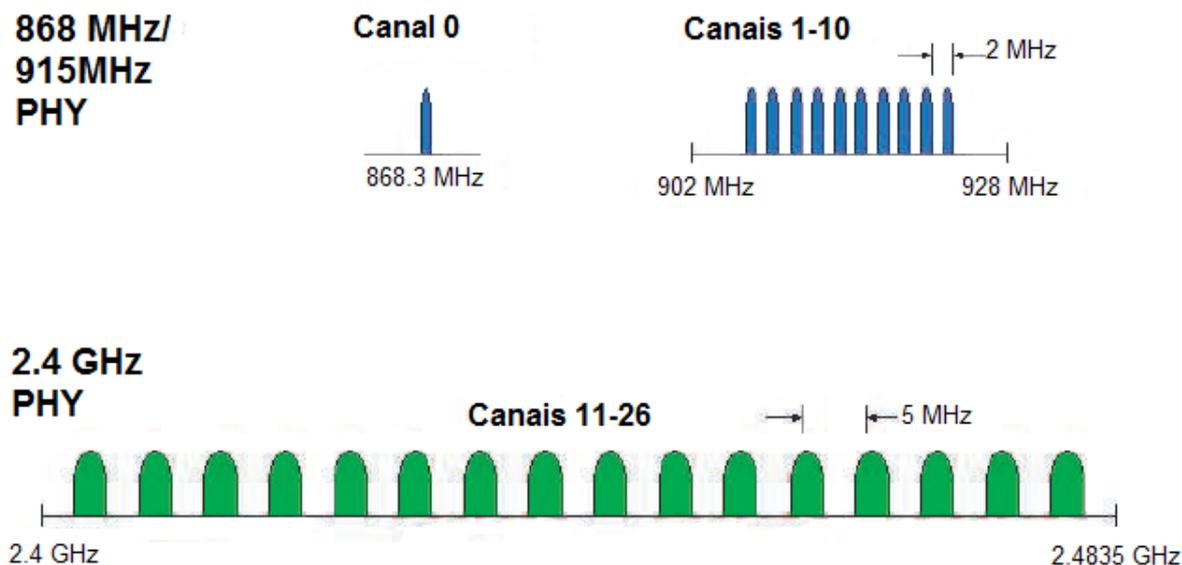


Figura 5 – Canais de Operação do ZigBee.
Fonte: GTA UFRJ, ZigBee.

2.4 Camadas ZigBee

O padrão IEEE 802.15.4 é uma parte da estrutura que está contida na tecnologia ZigBee, sendo ele responsável pela especificação básica dos serviços da camada física e de enlace⁸ do modelo *Open Systems Interconnection* (OSI).

Todos os módulos que trabalham com o padrão ZigBee possuem, necessariamente, um rádio transmissor/receptor IEEE 802.15.4, e um microcontrolador com a camada MAC embarcada em seu firmware. A definição da camada física e da camada de acesso ao meio é responsabilidade da norma IEEE 802.15.4, enquanto que o ZigBee define as outras camadas do modelo OSI. (COUTO; FIGUEIREDO, 2008, p. 5).

A pilha do protocolo é representada pela Figura 6.

⁷ Na O-QPSK e BPSK a modulação é feita na fase da portadora. O O-QPSK representa 2 *bits* e o BPSK representa 1 *bit* na transmissão, por isto a taxa de transmissão com o O-QPSK é maior.

⁸ Camadas 1 e 2 do modelo OSI.

Usuário	Aplicação
ZigBee Alliance	Suporte a Aplicação
	Rede (NWK) / Segurança (SSP)
IEEE 802.15.4	MAC
	PHY

Figura 6 – Pilha do protocolo ZigBee.
Fonte: GTA UFRJ, ZigBee.

2.4.1 Camada Física (PHY)

Foi projetada para definir a operação física do dispositivo, incluindo a potência de saída, sensibilidade de recepção, tipo de modulação, número de canais e velocidade de transmissão. Fornece um grau elevado de integração utilizando *Direct Sequence Spread Spectrum* (DSSS), permitindo a utilização de equipamentos mais simples e baratos.

Na técnica DSSS, a informação a ser transmitida é multiplicada por um sinal codificador, conhecido como código de Barker com 11 *bits* de tamanho. O código de Barker é gerado em uma frequência maior que a do sinal de informação a ser transmitido. Assim o sinal resultante da multiplicação apresenta uma banda mais larga que a do sinal de informação. Uma das vantagens do DSSS é que uma interferência em uma frequência específica não é tão prejudicial para o sinal, já que está espalhado na frequência da banda. (CARDOSO; SOARES, 2005, p.8).

2.4.2 Camada de Controle de Acesso ao Meio (MAC)

Foi projetada com a função de permitir múltiplas topologias com baixa complexidade. Permite o controle de um grande número de dispositivos sem a necessidade de colocá-los em espera, possibilita a utilização da criptografia AES, mas o seu controle é feito pelas camadas superiores. Admite que um *Reduced Function Device* (RFD) ⁹ opere na rede sem a necessidade do mesmo possuir memória grande.

⁹ Dispositivo de Funções Reduzidas. Será melhor definido mais adiante.

Tem com função controlar o acesso aos canais de Rádio Frequência (RF), utilizando *Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA)*¹⁰. Especifica o tipo de dispositivos permitidos na rede e a estrutura de pacotes. Em suma, a camada MAC é responsável por prover um mecanismo confiável de transmissão, pelo transporte de dados e sincronização.

Outras características da camada MAC são: sincronizar dispositivos por meio de *beacons*¹¹, proporcionar serviços de associação e desassociação na rede e encapsular os dados das camadas superiores.

A camada MAC possui quatro tipos de *frames*: *Beacon Frame*, *Data Frame*, *Acknowledgment Frame* e o *MAC Command Frame*, os quais serão descritos mais adiante.

2.4.3 Camada de Rede (NWK)

É a primeira camada definida pelo padrão ZigBee. Foi projetada para que a rede cresça sem a necessidade de utilizar equipamentos transmissores de alta potência e para manipular uma grande quantidade de nós com baixa latência. Com isto é possível adicionar dispositivos com função de roteadores para aumentar a rede sem a necessidade de aumento da potência de transmissão.

Entre outras funções da camada de rede temos:

- a) Inicialização da rede;
- b) Coordenar a entrada e saída de dispositivos na rede;
- c) Configurar novos dispositivos;
- d) Distribuição de endereços;
- e) Sincronização da rede;
- f) Roteamento de pacotes;
- g) Segurança dos dados.

A camada de rede implementa o parâmetro *Radius*. Este parâmetro é similar ao *Next Hop*¹² das redes TCP/IP. Pode-se configurar, por exemplo, o *Radius* para o valor 3, com isto,

¹⁰ Quando um nó deseja fazer uma transmissão na rede, ele envia um sinal para todos os componentes da rede, só então os dados são transmitidos. Caso durante uma transmissão um sinal de aviso for detectado o emissor interrompe o envio da mensagem, reiniciando a tentativa de transmissão após um determinado tempo.

¹¹ Pacotes de sinalização.

¹² Próximo salto.

o pacote não deve passar por mais de 3 equipamentos ZigBee. A cada salto o *Radius* é decrementado em uma unidade, chegando a zero, então o pacote é descartado, isto é equivalente ao *Time to Live* (TTL), com a diferença que o *Radius* serve para determinar até que nível de parentesco o pacote pode chegar.

2.4.4 Camada de Aplicação

Esta camada interage com as camadas inferiores com a finalidade de propor uma interação entre o dispositivo ZigBee e o usuário final. É nesta camada que os fabricantes desenvolvem aplicações para personalizar os seus dispositivos. Nesta camada é onde é definido se o dispositivo será um RFD ou um FFD.

Na camada de aplicação encontramos a Subcamada de Suporte à Aplicação (APS), o *ZigBee Device Objects* (ZDO) e o *Application Framework* (AF), explicados a seguir.

2.4.4.1 Subcamada de Suporte à Aplicação (APS)

Serve de interligação entre a camada de rede e a de aplicação. Oferece os serviços de *Discovery*¹³ e *Binding*¹⁴.

Fornecer uma interface entre a camada de rede e aplicação. É responsável por gerenciar a tabela de ligação, mantendo uma base de dados com os dispositivos conectados na rede. Também é sua função prover a transmissão de dados (PDUs) entre dois dispositivos na mesma rede. Para isso ele realiza a fragmentação e remontagem das PDUs, além dos serviços de segurança. (ESCHNER, 2011, p. 15).

2.4.4.2 *ZigBee Device Objects* (ZDO)

É onde está definido o papel que o dispositivo exerce na rede, isto é, se ele é um coordenador, um roteador ou um dispositivo final. Também é onde é definido o método de segurança a ser usado na rede e onde se dá o início das solicitações de *Binding*.

¹³ Descobrimto do nós ativos ao alcance da rede.

¹⁴ União de dispositivos de acordo com suas necessidades e serviços.

2.4.4.3 Application Framework

Contém o ambiente em que os aplicativos dos usuários interagem com o protocolo ZigBee, possibilitando configurar os dispositivos da aplicação.

Application Framework: é o ambiente onde são alocados os objetos de aplicação ZigBee (*Application Objects*). Esses objetos possuem funções definidas pelos fabricantes, incluindo primitivas de serviço de dados como *request*, *response*, *confirm*. Cada objeto representa um tipo (ou *profile*) de aplicação diferente, que é definido para um único dispositivo ZigBee, é possível armazenar até 240 objetos de aplicação distintos em um dispositivo. (ESCHNER, 2011, p. 15).

2.5 Segurança no ZigBee

A fim de garantir a segurança dos dados, o ZigBee utiliza como algoritmo de criptografia o AES de 128 *bits* para cifrar as mensagens. O AES utiliza uma *string* de *bits* como chave de segurança. A segurança é realizada para prover a confidencialidade, integridade e a autenticidade dos pacotes.

Para proteger pacotes com um único salto na rede é utilizada a segurança nos pacotes MAC. Para pacotes com mais de um salto na rede, a função de assegurar a segurança dos dados é deixada para as camadas superiores¹⁵.

Na camada MAC é utilizado o AES para criptografar e validar os dados a serem enviados, e o *Message Integrity Code* (MIC) garante a integridade e a validação dos dados. O MIC tem o seu funcionamento parecido com o *Checksum*, caso o receptor receber uma mensagem com o MIC diferente do calculado, o pacote será descartado. Com a utilização do MIC junto ao AES, evitamos que um invasor possa modificar e enviar uma mensagem que esteja cifrada. A camada MAC realiza o processamento de segurança, mas quem controla os processos são as camadas superiores.

A figura 7 mostra um pacote MAC, com o *bit* de segurança ativado.

Para garantir a confidencialidade dos pacotes, é utilizado um número, o *nonce*, que nada mais é que um contador de *frames* de 32 *bits*. Cada dispositivo da rede mantém este contador que é incrementado a cada transmissão, assim é possível verificar se o *frame* enviado é o que deveria vir na sequência. Este contador evita ataques por reenvio.

¹⁵ Camada de Rede.

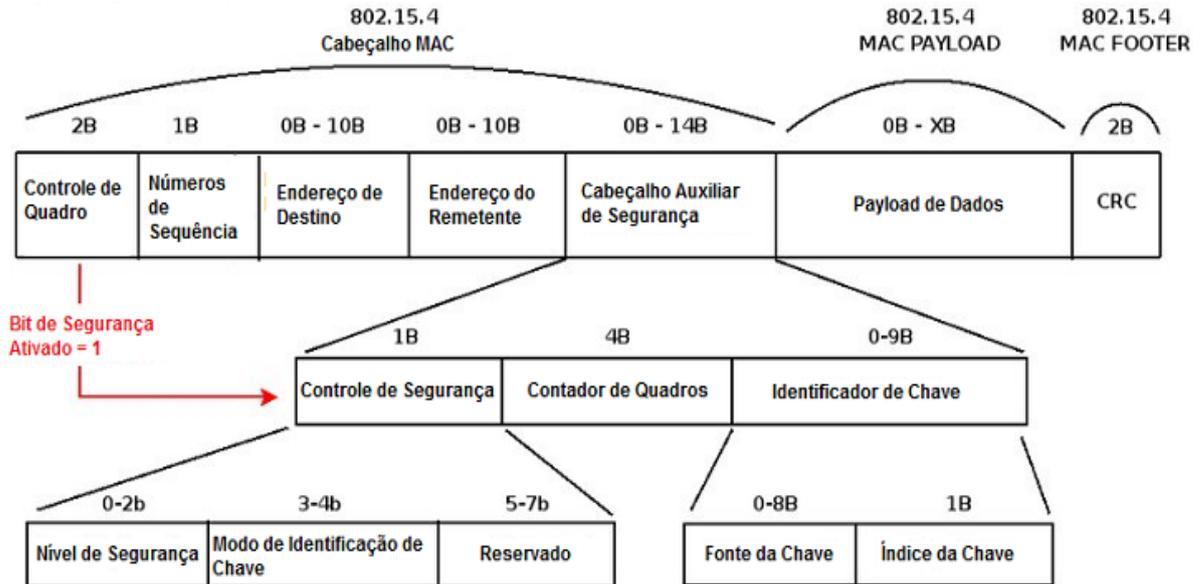


Figura 7 – Pacote MAC com *bit* de segurança ativado.
Fonte: GTA UFRJ, ZigBee

Conseqüentemente, as soluções de segurança anteriormente descritas para a camada MAC, também são utilizadas para a camada NWK.

O padrão ZigBee também pode definir um dispositivo, o coordenador, para funcionar como um distribuidor de chaves e como autenticador de adesão de dispositivos na rede, este dispositivo funciona como um *trust-center*. O *trust-center* funciona como uma central de armazenamento e distribuição de chaves, ele permite ou não que novos dispositivos entrem na rede, tendo um papel importante na segurança da mesma.

2.6 Dispositivos Físicos ZigBee

2.6.1 Dispositivos de Funções Completas (FFD)

Geralmente são alimentados pela energia elétrica, mas não é uma regra. Possuem uma construção mais complexa, conseqüentemente necessitam de um *hardware* mais potente para implementar a pilha do protocolo, assim consomem mais energia. Os FFD podem se comunicar com qualquer dispositivo da rede.

Tem também como característica possuir no mínimo 32kB de memória para administrar e construir tabelas de rotas e configurações de parâmetros.

2.6.2 Dispositivo de Funções Reduzidas (RFD)

Sua pilha de protocolo pode ser implementada utilizando uma quantidade mínima de memória, em média 6kB, conseqüentemente consomem menos energia. Só podem se comunicar com dispositivos FFD.

2.7 Dispositivos Lógicos ZigBee

2.7.1 Coordenador ZigBee(ZC)

Somente dispositivos FFD podem assumir a função de coordenadores. O coordenador é o dispositivo responsável pela manutenção da rede, distribuição de endereços e formação da mesma, ele também pode servir de elemento de interligação entre redes ZigBee.

Ficam alimentados por energia em tempo integral, sempre escutando o canal.

2.7.2 Roteador ZigBee (ZR)

Como o coordenador, somente dispositivos FFD podem assumir a função de roteador. Possui a função de servir de roteador entre os nós da rede, sem precisar de coordenador. Através dele podemos aumentar a capacidade de transmissão com a expansão da rede.

2.7.3 Dispositivo Final ZigBee (ZED)

Dispositivos FFD e RFD podem assumir a função de ZED. Na rede é onde os atuadores, sensores, etc., serão alocados. São os nós da rede que menos consomem energia, pois na maioria das vezes eles permanecem no modo *sleep*, consumindo o mínimo de corrente.

O quadro 2 mostra as funções que os dispositivos ZigBee podem realizar em uma rede.

ZigBee Coordinator (ZC)	ZigBee Router (ZR)	ZigBee End Device (ZED)	Função da camada de rede ZigBee
X			Estabelecer uma nova rede
X	X		Atribuir endereços lógicos da rede
X	X		Permitir a entrada e saída de outros dispositivos
X	X		Manter lista de vizinhos e rotas
X	X		Rotear pacotes da camada de rede
X	X	X	Transferir pacotes da camada de rede
X	X	X	Entrar e sair da rede ZigBee

Quadro 2 – Funções dos dispositivos ZigBee.
Fonte: GTA, UFRJ, ZigBee.

2.8 Topologias das Redes ZigBee

2.8.1 Estrela (*Star*)

É a topologia mais simples, composta de um coordenador e dispositivos finais. Deve ser usada em locais com poucos obstáculos.

A sua vantagem é a de ser uma rede simples de ser formada, mas a desvantagem é de contar somente com um coordenador, assim as mensagens terão um caminho único, já que todas elas deverão passar pelo coordenador, pois RFD só se comunicam com FFD.

2.8.2 Árvore (*Cluster Tree*)

A topologia em árvore pode ser entendida como uma expansão da rede estrela, de hierarquia maior e com a inserção de roteadores, aumentando assim a sua capacidade de transmissão (alcance). O coordenador assume o papel do nó central da rede.

A topologia Árvore é formada apenas modificando a topologia Estrela. Um ou mais dos RFDs conectados ao coordenador da Personal Area Network (PAN) é substituído por um FFD, e destes FFDs, mais RFDs ou FFDs podem ser conectados. Uma vantagem desta topologia é que ela pode ser usada para aumentar a cobertura geográfica da rede. (MONSIGNORE, 2007, p. 20).

2.8.3 Malha (*Mesh*)

Este tipo de rede fornece vários caminhos entre os dispositivos ZigBee, diminuindo os pontos de falha, pois caso um nó da rede esteja com problemas, os dados podem seguir por outro até chegar ao destino final.

Com a utilização desta rede temos uma maior flexibilidade e alcance, podendo ser usada em fábricas com vários setores ou em um amplo espaço geográfico. Esta foi a topologia escolhida para o projeto dos sensores de gás devido às características citadas anteriormente. Na figura 8 são mostradas as topologias das redes ZigBee.



Figura 8 – Topologias de redes ZigBee.
Fonte: GTA, UFRJ, ZigBee.

2.9 Formação da Rede ZigBee

Como visto anteriormente, somente o coordenador pode iniciar a formação de uma rede.

A formação de uma nova rede ZigBee é iniciada através de uma primitiva da camada de rede que é restrita ao coordenador ZigBee que não pertence a nenhuma rede. Para iniciar uma rede, um coordenador ZigBee procura por um conjunto de canais. Quando esta procura se completa, os canais são ordenados em ordem decrescente de níveis de energia (os canais de níveis mais baixos são descartados). O coordenador então procura em cada canal por dispositivos ou redes ZigBee. Baseado neste resultado, o coordenador escolhe o melhor canal para criar uma nova rede, dando preferência para canais nos quais não foram encontradas outras redes. Então o coordenador escolhe um identificador de rede lógico (um número) que será atribuído a todo dispositivo que ingressar na rede. Finalmente o coordenador permite outros dispositivos ingressarem na rede. (AZEVEDO, 2006, p. 2).

A varredura dos canais é feita enviando um *beacon* em broadcast em cada canal, caso um canal já esteja em uso, é recebido pelo coordenador uma resposta com o *Personal Area Network Identifier* (PAN ID) deste canal, identificando assim os canais e as PAN ID já em uso. Esta varredura é chamada de *PAN Scan*.

O tempo de entrada de um nó na rede é de aproximadamente 30 ms.

2.10 Modos de Operação da Rede ZigBee

2.10.1 Modo *Non-Beaconing*

Os dispositivos da rede ZigBee permanecem sempre ativos, o que leva eles a consumir mais energia, portanto eles devem ser alimentados por baterias mais duradouras ou por fontes de energia. O coordenador da rede deve operar neste modo.

2.10.2 Modo *Beaconing*

Os roteadores da rede transmitem de tempos em tempos sinalizações¹⁶, para confirmar a sua presença na rede. É possível realizar uma boa sincronia entre os dispositivos da rede, exceto o coordenador que permanece sempre ativo, para que entre uma sinalização e outra eles fiquem no modo *sleep*, consumindo assim o mínimo de energia.

Neste modo ao usar baterias similares as de computadores para que alimentem os módulos, estas têm a sua vida útil aumentada, podendo durar até quatro anos sem a necessidade de substituição. O tempo de transição de um dispositivo do modo *sleep* para o modo ativo é de aproximadamente 15 ms.

No modo *Beaconing* é utilizado a estrutura do *Superframe*, conforme pode ser visto na figura 9.

Como pode ser notado o superquadro é composto de 2 períodos, denominados de ativo e inativo. O período ativo, que corresponde à duração do superquadro, denominado de *Superframe Duration (SD)*, é dividido igualmente em 16 compartimentos de tempo, os quais proporcionam a comunicação de dados. Cada período ativo pode ainda ser subdividido em dois subperíodos, sendo que o primeiro habilita o acesso ao meio por contenção e é denominado por *Contention Access Period (CAP)*, o qual utiliza a técnica CSMA-CA e o segundo, não possui contenção, ou seja, o acesso ao meio é garantido. Esse subperíodo é conhecido como *Contention Free Period (CFP)* e utiliza a técnica denominada de compartilhamento de tempo garantido (GTS). (CANSADO, 2011, p. 60).

¹⁶ *Beacon*.

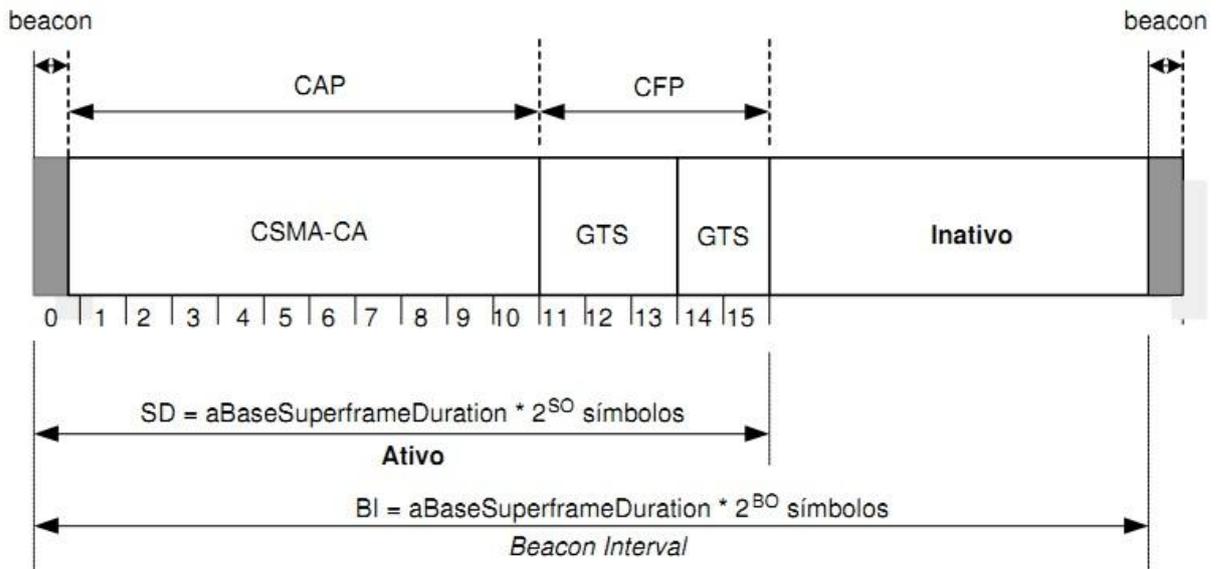


Figura 9 – Estrutura do *Superframe*.
 Fonte: (SANTOS, 2007, p. 15).

2.11 Tipos de Mensagens na Rede ZigBee

2.11.1 Mensagens *Broadcast*

Todos os dispositivos que estão no mesmo canal de frequência recebem a mensagem, independente da PAN em que se encontram.

Qualquer dispositivo dentro da rede pode iniciar uma transmissão broadcast para todos os outros dispositivos que fazem parte da mesma rede. A especificação ZigBee não suporta broadcast entre várias redes. Cada dispositivo mantém um registro de qualquer nova transação broadcast que se iniciou localmente, ou foi recebida de um dispositivo vizinho. Este registro é chamado de registro de transação broadcast (*Broadcast Transaction Record - BTR*) e deve conter o número de sequência e o endereço fonte do *frame* de broadcast. Os registros de transação broadcast são armazenados na tabela de transações broadcast (*Broadcast Transaction Table - BTT*). Quando um dispositivo recebe um *frame* de broadcast de um vizinho, ele deve comparar o número de sequência e o endereço fonte com seus registros na BTT. Se o dispositivo tem uma BTR para este *frame*, ele deve atualizar a BTR marcando o registro como já encaminhado e depois deve descartar o *frame*. Se nenhum registro é encontrado, o dispositivo cria um novo registro na BTT. Se o campo *radius* é maior que zero o dispositivo decrementa este campo e retransmite o *frame*, caso contrário o *frame* é descartado. Antes de retransmitir o dispositivo espera por um tempo aleatório. (AZEVEDO, 2006, p. 5).

Devido as mensagens em *Broadcast* serem reencaminhadas pelos nós da rede, estas devem ser usadas esporadicamente, pois aumentam o tráfego na mesma.

2.11.2 Mensagens *Multicast*

Este tipo de comunicação é como se fosse um *Broadcast*, com a diferença de que as mensagens são enviadas para um grupo específico de dispositivos, utilizando-se do endereço de 16 *bits* dos dispositivos. Deste modo não é necessário enviar várias mensagens ponto-a-ponto para cada dispositivo do grupo, diminuindo o tráfego na rede.

Os nós a quem não são destinadas as mensagens *Multicast* devem repassá-las e aos que se destinam, além de repassarem devem manter a mensagem.

2.11.3 Mensagens *Unicast*

As mensagens *Unicast* são enviadas diretamente ao destino, ganhando assim em agilidade, robustez e segurança dos pacotes. As transmissões *Unicast* sempre são endereçadas através do endereço de 16 *bits* do destino, mas para garantir a entrega dos dados, o endereço de 64 *bits* também é utilizado.

2.12 Endereçamento ZigBee

2.12.1 Endereço de 16 *Bits*

Este endereço é atribuído ao dispositivo quando ele adere à rede, distribuído pelo coordenador, por isso, também pode ser chamado de endereço na rede ou endereço curto.

As tabelas de roteamento e as transmissões na rede utilizam o endereço de 16 *bits*, mas como este endereço é dinâmico, nas transmissões também são incluídos os endereços de 64 *bits* de origem e destino, para garantirem a entrega do pacote.

2.12.2 Endereço de 64 Bits

O endereço de 64 *bits* é definido na fabricação do dispositivo, por isto ele é único para cada equipamento, é também chamado de endereço estendido. É formado por 24 *bits* referentes ao *Organizationally Unique Identifier (OUI)*¹⁷, atribuído pelo IEEE, e por 40 *bits* definidos pelo fabricante.

2.13 Roteamento ZigBee

2.13.1 Roteamento Hierárquico ou em Árvore

Quando um nó da rede envia um pacote, este é enviado ao coordenador/roteador o qual o nó está ligado. Caso o endereço de destino esteja ligado ao mesmo coordenador/roteador, este pacote é enviado de forma descendente ao seu destino. Se o destino não pertencer ao mesmo ramo do nó de origem, este pacote é enviado de forma ascendente ao coordenador/roteador a qual está ligado o nó de origem, chegando a ele de forma descendente, agora pelo ramo correto. Deste modo o roteador não precisa manter uma tabela completa dos endereços, e sim saber quem é o seu roteador pai e os seus filhos juntamente com os intervalos de roteamento.

¹⁷ Ao realizar o registro no IEEE, o fabricante recebe o OUI, que é o identificador exclusivo destinado ao mesmo.

2.13.2 Roteamento Sob Demanda

Este roteamento baseia-se no *AdHoc On-Demand Distance Vector* (AODV) e é indicado para trabalhar em redes com topologia em malha.

Todos os nós têm uma tabela de vizinhança com os endereços de seus nós vizinhos e sua relação familiar com os mesmos: pais ou irmãos. A escolha dos pais é feita por uma rotina para a descoberta de uma melhor rota, sendo a melhor rota aquela com o menor número de saltos (*hops*) até o coordenador, o chamado AODV (*Adhoc On Demand Distance Vector*). E isso é similar no roteamento. Se o nó fonte não tem qualquer informação de roteamento até o nó destino, ele envia uma mensagem em broadcast com o endereço do nó destino. Todos os nós ao alcance do nó fonte recebem essa mensagem. Caso tenham a informação do caminho até o destino em sua tabela de roteamento, respondem ao nó fonte com o custo até o mesmo (número de *hops*), e o nó fonte pode escolher aquela rota com o menor custo, caso não a tenham, dão um encaminhamento (*forward*) com uma mensagem em broadcast incrementando o campo de número de saltos. Também há a capacidade de atualizar a tabela de roteamento. Se a mensagem é marcada com Ack (*Acknowledgment*) e ela chega por uma rota diferente daquela salva em sua tabela com um número de *hops* menor do que o contido na mesma, pode ser feita uma atualização (*update*). (ZUCATO, 2009, p.37-38).

2.14 Frames ZigBee

O Controle de Quadro (*Frame Control*) define diversas informações que serão usadas no nó destino, como: se o nó que originou o quadro faz parte da mesma rede, o tipo de nó de que se trata (FFD ou RFD), o tipo de quadro, quadro de dados, quadro de *Beacon* ou quadro de comando. Essas são informações que farão o nó destino tomar as decisões corretas em seu software. O Número de Sequência (*Sequence Number*) é usado para controle de dados faltantes, muito comum na transmissão por pacotes da internet. O campo de endereços tem um tamanho bem flexível, pois, dependendo do tipo de requisição, é preciso enviar tanto o endereço do nó da rede, chamado *short address*, quanto o seu endereço de placa, o *MAC address*, além do endereço da rede em que faz parte, o *PAN address*. O campo de comando é a requisição ou uma resposta feita: se o nó fonte está requisitando uma associação a uma determinada rede ou respondendo a uma requisição que o nó destino já tinha feito. A carga do comando está relacionada à hierarquia de endereços. O FCS (*Frame Check Sequence*) refere-se aos caracteres *checksum* adicionados ao quadro para detecção e correção de erros. (ZUCATO, 2009, p. 33).

2.4.2.1 Beacon Frame

É o quadro de sinalização enviado pelo coordenador. Serve para delimitar o *Superframe*, sincronizar *frames* e dispositivos e para controle da entrada de dispositivos na rede. A figura 10 mostra o *Beacon Frame*.

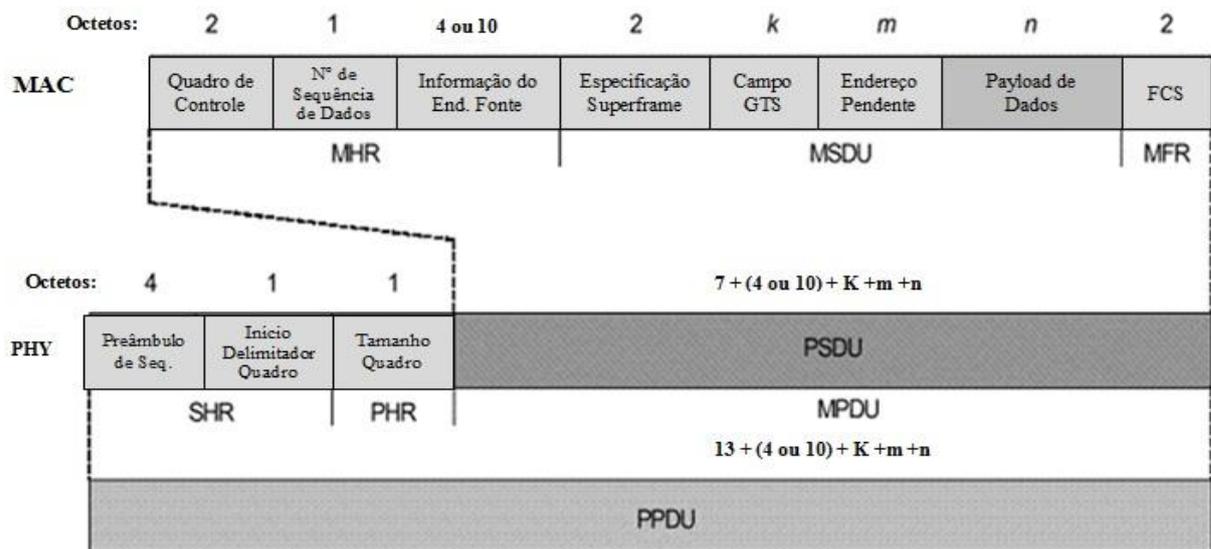


Figura 10 – Beacon Frame.
Fonte: Adaptado de GTA, UFRJ, ZigBee.

2.4.2.2 Data Frame

É o *frame* de dados. Utilizado por todos os dispositivos, independente da topologia da rede. Na figura 11 temos o formato do *Data Frame*.

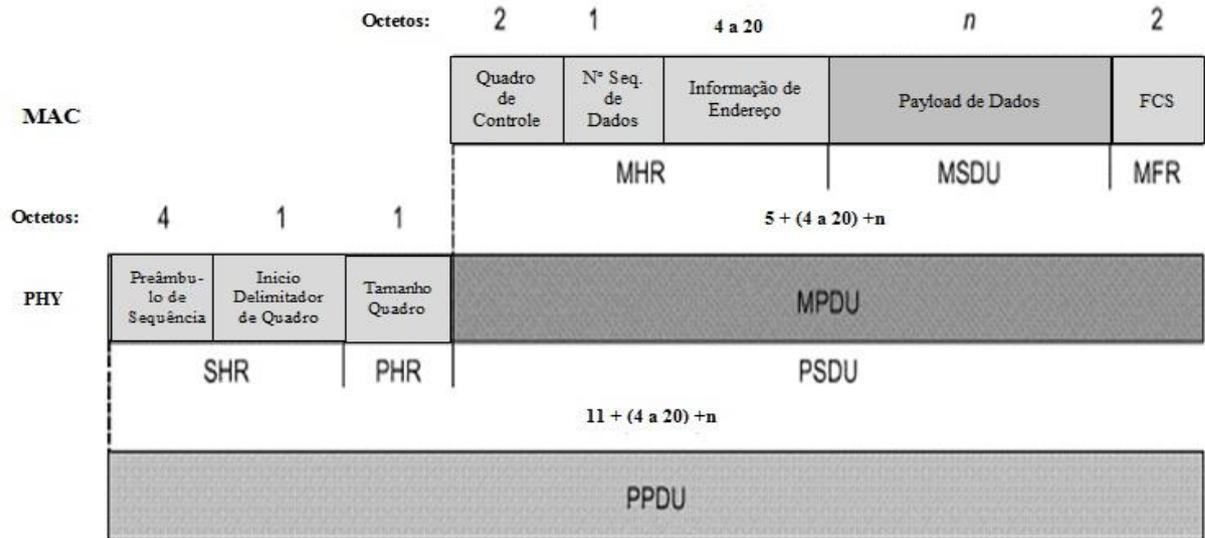


Figura 11 – *Data Frame*.
 Fonte: Adaptado de GTA, UFRJ, ZigBee.

2.4.2.3 Acknowledgment Frame

É usado pela camada MAC para confirmar a recepção de dados. Este *frame* só é enviado se o *frame* de origem contém a solicitação, caso contrário ele é descartado. O *Acknowledgment Frame* é mostrado na figura 12.

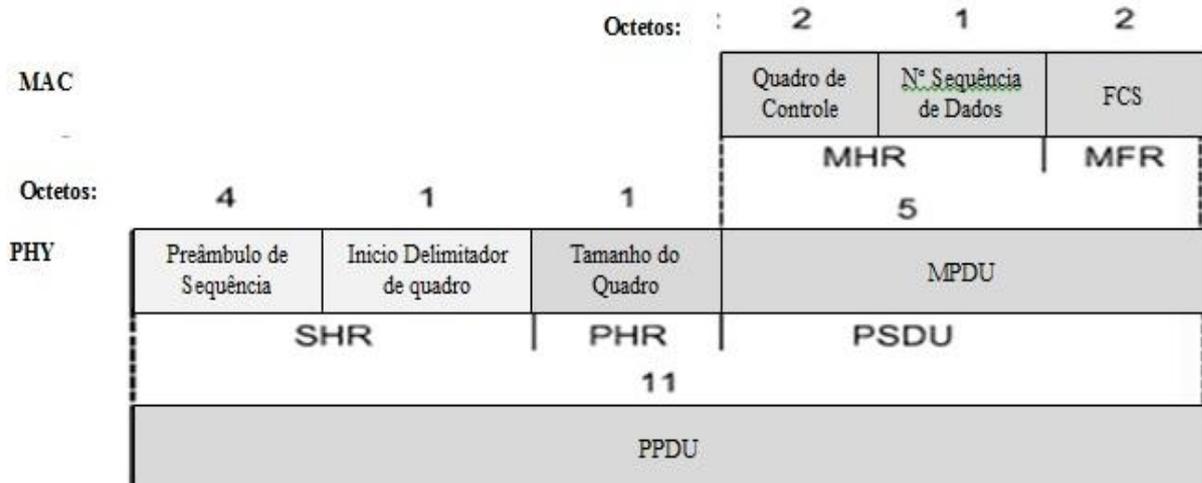


Figura 12 – *Acknowledgment Frame*.
 Fonte: Adaptado de GTA, UFRJ, ZigBee

2.4.2.4 MAC Command Frame

O *frame* de comando MAC permite que o coordenador configure os nós da rede. Na figura 13, temos a estrutura deste *frame*.

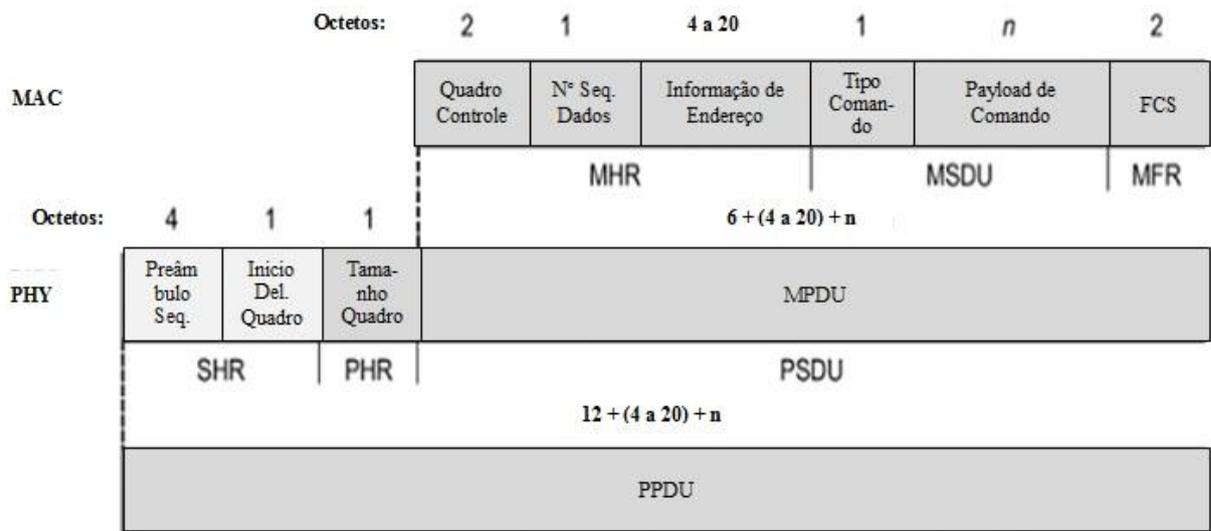


Figura 13 – MAC Command Frame.
Fonte: Adaptado de GTA, UFRJ, ZigBee.

3 COMPONENTES DO SENSOR DE GÁS

3.1 XBee

Para o desenvolvimento deste trabalho foi utilizado um módulo sem fio ZigBee, da MaxStream, o qual é fabricado pela empresa Digi, denominado XBee-PRO ZB.

O XBee é um módulo de rádio sem fio baseado no padrão IEEE 802.15.4 que implementa o protocolo ZigBee. Segundo a Digi, eles fabricam dois tipos de módulos: o XBee e o XBee PRO, que diferem somente na potência de transmissão, alcançando assim diferentes distâncias. O quadro 3 mostra as características dos módulos XBee e XBee PRO.

Característica	XBee	XBee PRO
Transmissão Indoor	Até 30 m	Até 100 m
Transmissão Outdoor	Até 100 m	Até 1600 m
Potência de Transmissão	1 mW (0 dBm)	60 mW (18 dBm)
Taxa de Transmissão	250 kbps	250 kbps
Tensão de Alimentação	2,8 - 3,4 V	2,8 - 3,4 V
Corrente de Transmissão	45 mA para 3,3 V	215 mA para 3,3V
Corrente de Recepção	50 mA para 3,3 V	55 mA para 3,3 V
Corrente em Sleep	< 10 μ A	< 10 μ A
Frequência de Operação	ISM de 2,4 GHz	ISM de 2,4 GHz
Número de Canais	16	12
Temperatura de Operação	-40 a 85°C	-40 a 85°C

Quadro 3 – Características dos Módulos XBee e XBee PRO.

Fonte: Manual do XBee.

Os módulos XBee possuem quatro tipos de antenas: *Whip*, *U-FL Connector*, *Chip* e *RPSMA Connector*, conforme mostrado na figura 14. Para utilização neste trabalho foram usados módulos com antenas *Whip* na prova de conceito e antenas *RPSMA Connector* no protótipo.

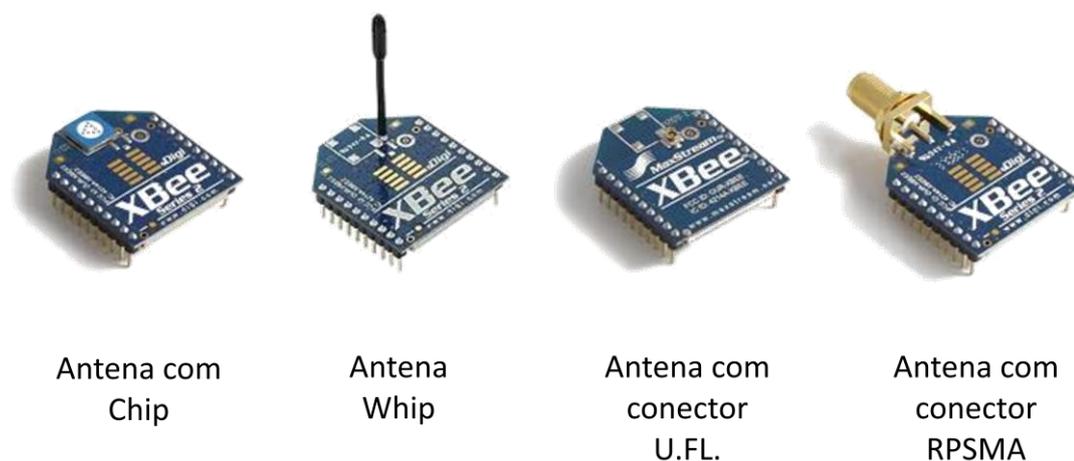


Figura 14 – Tipos de antenas dos módulos XBee.
 Fonte: Adaptado de Compututorials.

O XBee possui 20 pinos, entre eles estão os pinos de alimentação e os pinos de entrada/saída de dados, que podem ser usados tanto para dados digitais como analógicos. São através destes pinos que podemos conectar ao módulo os sensores utilizados neste trabalho. O quadro 4 mostra a função de cada pino XBee.

(continua)

Pino	Nome	Direção	Descrição
1	VCC	-	Alimentação 3,3V
2	DOUT	Saída	Saída de dados da UART
3	DIN / $\overline{\text{CONFIG}}$	Entrada	Entrada de dados da UART
4	DO8	Saída	Saída digital 8
5	$\overline{\text{RESET}}$	Entrada	Inicializa módulo (um pulso nível 0 de pelo menos 200ms)
6	PWM0 / RSSI	Saída	Saída do PWM 0 / Indicador de Força do sinal de RF (RX)
7	PWM1	Saída	Saída do PWM 1
8	(Reservado)	-	Ainda não tem uma função definida (futura implementação)
9	$\overline{\text{DTR}} / \text{SLEEP_IRQ} / \text{DI8}$	Entrada	Linha de Controle da Função Sleep ou Entrada digital 8

(conclusão)

Pino	Nome	Direção	Descrição
10	GND	-	Terra
11	AD4 / DIO4	Entrada/Saída	Só Entrada Analógica 4 ou Entrada/Saída Digital 4
12	$\overline{\text{CTS}}$ / DIO7	Entrada/Saída	Controle de Fluxo CTS ou Entrada/Saída Digital 7
13	ON / SLEEP	Saída	Indicador de Estado do Módulo
14	VREF	Entrada	Voltagem de Referência para as Entradas A/D
15	Associação / AD5 / DIO5	Entrada/Saída	Indicador de Associação, só Entrada Analógica 5 ou Entrada/Saída Digital 5
16	$\overline{\text{RTS}}$ / AD6 / DIO6	Entrada/Saída	Controle de Fluxo RTS, só Entrada Analógica 6 ou Entrada/Saída Digital 6
17	AD3 / DIO3	Entrada/Saída	Só Entrada Analógica 3 ou Entrada/Saída Digital 3
18	AD2 / DIO2	Entrada/Saída	Só Entrada Analógica 2 ou Entrada/Saída Digital 2
19	AD1 / DIO1	Entrada/Saída	Só Entrada Analógica 1 ou Entrada/Saída Digital 1
20	AD0 / DIO0	Entrada/Saída	Só Entrada Analógica 0 ou Entrada/Saída Digital 0

Quadro 4 – Pinagem do módulo XBee.
 Fonte: Manual do XBee.

3.1.1 Programa X-CTU

Este *software* foi desenvolvido pela Digi e serve para configurar, atualizar o *firmware* e realizar teste de comunicação dos módulos XBee. O software pode ser obtido de forma gratuita no site da Digi. Na figura 15 é possível visualizar a interface do X-CTU.

Através do programa é possível alterar as configurações de rede, endereço, segurança, entrada e saídas nos pinos do módulo. Estas configurações podem ser alteradas tanto através da aba *Modem Configuration* como através da aba *Terminal* mediante comandos AT.

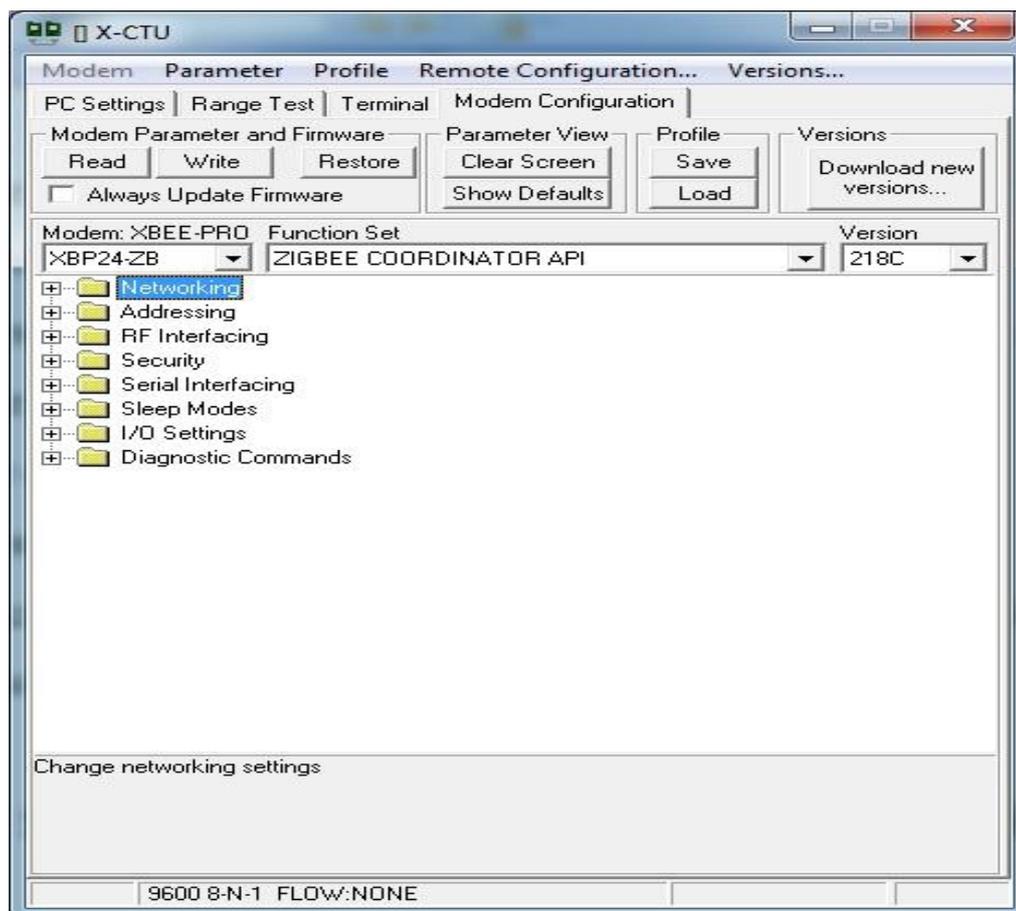


Figura 15 – Interface do programa X-CTU.

3.1.2 Modos de Comando do XBee

Através dos modos de comando do XBee, podemos alterar as configurações dos módulos e para isto o XBee possui dois modos: o Modo AT e o Modo API.

As configurações dos módulos podem ser alteradas através do *Gateway* ou através do programa X-CTU.

3.1.2.1 Modo de Comando AT

Este modo é também chamado de Modo de Comando Transparente, pois todos os dados recebidos pela porta serial são transmitidos pelo canal de RF e vice-versa, funcionando como um cabo serial.

Neste modo o equipamento é configurado através de comandos AT. No caso dos módulos serem configurados no programa X-CTU, para entrar no modo de comando AT, na aba *Terminal* do programa deve-se digitar +++ sem digitar *ENTER*, caso o módulo receba o comando corretamente ele enviará um *OK*, entrando assim no modo AT.

3.1.2.1.1 Formato dos Comandos AT

A figura 16 mostra o formato dos comandos AT. Caso o comando seja interpretado corretamente é retornado um *OK*, caso contrário é retornado *ERROR*.

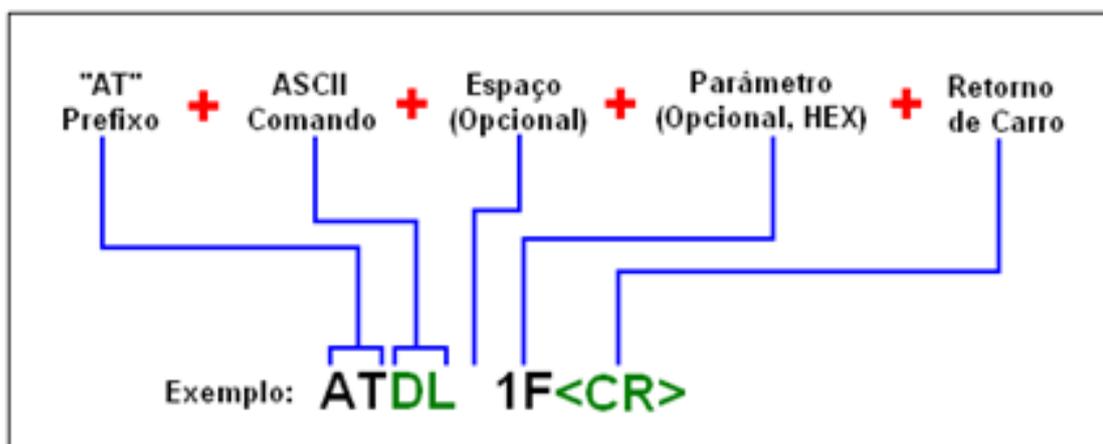


Figura 16 – Formato dos comandos AT.
Fonte: Adaptado do Manual do XBee.

3.1.2.1.2 Modo de Comando API

O modo API (*Application Programming Interface*) é uma alternativa para o modo transparente. A diferença é que neste modo os dados seriais são transmitidos através de *frames* especiais, que possuem dados adicionais para facilitar a configuração e o roteamento de pacotes. Assim, o modo API provê facilidades como envio de dados para múltiplos destinos sem necessidade de entrar no modo de comando, receber mensagens de sucesso ou falha após cada envio de dados e identificar o endereço do módulo que enviou a mensagem recebida. (COUTO; FIGUEIREDO, 2008, p. 36).

O formato do *frame* no modo API é mostrado na figura 17.

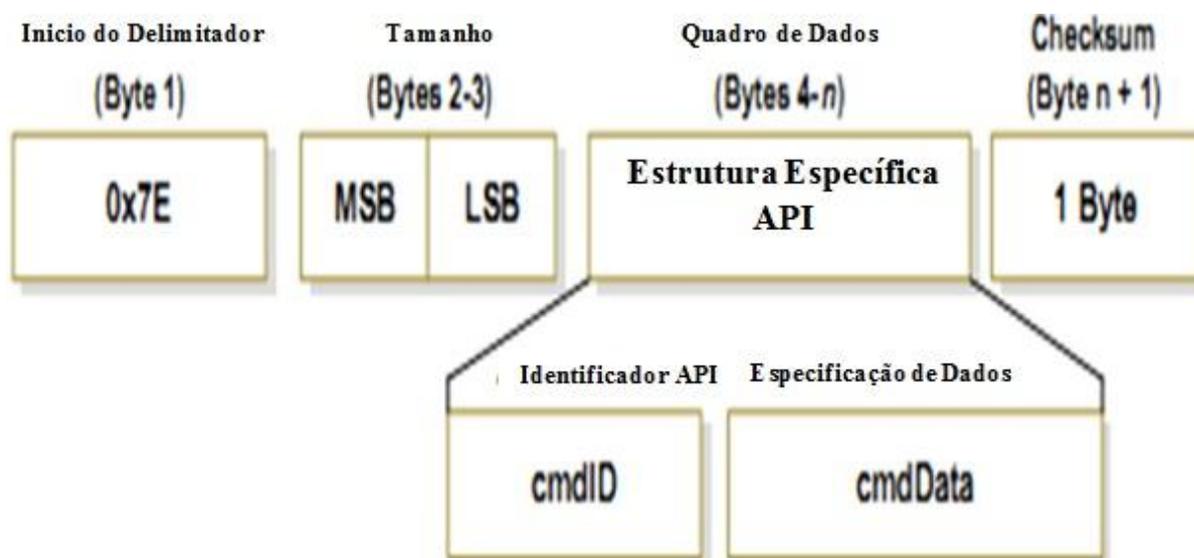


Figura 17 – *Frame* do modo API.
Fonte: Adaptado do Manual do XBee.

Segundo Couto e Figueiredo:

- Início do Delimitador – Byte que indica o início do quadro;
- Tamanho – Indica o tamanho do campo *Frame Data*;
- Quadro de Dados – Identificador e conteúdo da mensagem;
- Identificador API – Tipo de comando;
- Especificação de Dados – Conteúdo do *frame*;
- Checksum* – Byte que valida a integridade do quadro.

3.2 Sensor de Gás

Foi utilizado o Sensor de Gás Natural/Metano IDG-193. Conforme o manual do componente, ele é um sensor semicondutor de gás natural com sensibilidade de concentrações entre 200 e 10.000 ppm, com alta sensibilidade para gás metano, propano e butano, também podendo ser usado para detectar gases diferentes dos anteriormente citados.

Ainda de acordo com o manual, o sensor funciona da seguinte maneira: é aplicada uma tensão de 5 Vcc no aquecedor (resistor), que garantirá a temperatura de trabalho do sensor (pinos A ou B). Esta mesma tensão é aplicada aos pinos A ou B que fará com que na sua saída apareça uma tensão analógica de acordo com a quantidade de gás presente no ambiente. O resistor RL serve para ajustar a sensibilidade do sensor, conforme mostra a figura 18.

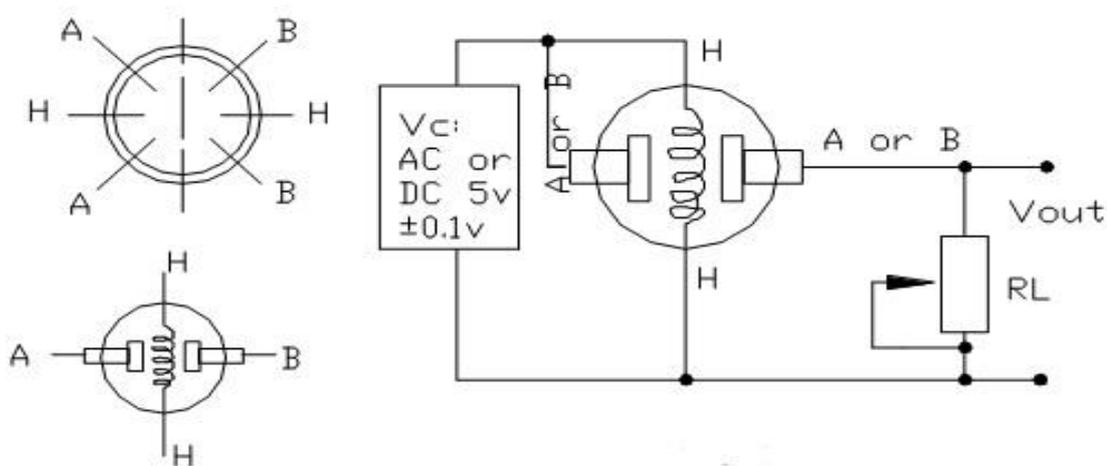


Figura 18 – Configuração do sensor de gás IDG-193.
Fonte: Manual do Sensor. I-Dutto IDG-193 – Uso restrito.

De uma maneira geral, o resistor do sensor é alimentado com uma tensão que faz com que o mesmo se aqueça. Quando não há presença de gás no ambiente, o resistor possui uma temperatura nominal de trabalho que é percebida pelo sensor e convertida em uma tensão analógica que é aplicada nos pinos de saída. Quando há presença de gás no ambiente, este entra em contato com o resistor que já está aquecido e como o gás é inflamável, as partículas deste gás explodem, elevando a temperatura do resistor, que por sua vez é percebido pelo sensor, alterando a tensão na saída do mesmo, indicando a presença de gás.

3.3 Sensor de Temperatura

Como o sensor de gás varia a tensão na sua saída de acordo com a temperatura que ele capta, foi observado durante os testes que se ocorresse uma variação da temperatura ambiente, o valor da tensão na saída do sensor de gás também iria variar, ocasionando um falso positivo. Por isto foi decidido colocar um sensor de temperatura no conjunto, para que fosse possível obter o valor da temperatura do ambiente.

O sensor de temperatura utilizado no projeto foi um termistor *Negative Temperature Coefficient* (NTC) de 10 k Ω a 25° C, ou seja, ele possui uma resistência de 10 k Ω quando a temperatura é de 25° C. O termistor é um sensor de temperatura fabricado com materiais semicondutores, ele altera a sua resistência de acordo com a temperatura. No caso do NTC, a sua resistência diminui com o aumento da temperatura.

O quadro 5 mostra o valor da resistência para cada temperatura no NTC.

Temperatura (°C)	Resistência (k Ω)
20	12,49
25	10,00
30	8,05
40	5,32
50	3,60
60	2,48
70	1,75
80	1,25
90	0,91
100	0,67

Quadro 5 – Temperatura e resistência do NTC.
Fonte: U.S.Sensor.

Utilizando-se dos valores de resistência do quadro 5, foram realizados testes com um resistor variável na função do termistor e verificado o valor de tensão que seria aplicado no

pino 19 de acordo com a temperatura. O quadro 6, mostra os valores de V_s^{18} , teórico e prático, obtidos e a figura 19 mostra o circuito utilizado para obter estes valores. O valor teórico de V_s foi obtido através da equação de divisor de tensão, que é dado por $V_s = \frac{V_e \times R_{NTC}}{R_{NTC} + R_i}$

Teste realizado com $R_i = 30 \text{ k}\Omega$ e $V_e = 3.3 \text{ Vcc}$			
$R_{ntc} \text{ (k}\Omega\text{)}$	Temperatura ($^{\circ}\text{C}$)	$V_s \text{ (V) Teórico}$	$V_s \text{ (V) Prático}$
12,49	20	0,97	0,96
10,00	25	0,82	0,84
8,05	30	0,69	0,70
5,32	40	0,49	0,48
3,60	50	0,35	0,36
2,48	60	0,25	0,25
1,75	70	0,18	0,19
1,25	80	0,13	0,14
0,91	90	0,09	0,09
0,67	100	0,07	0,07

Quadro 6 – Resultados dos testes com NTC.

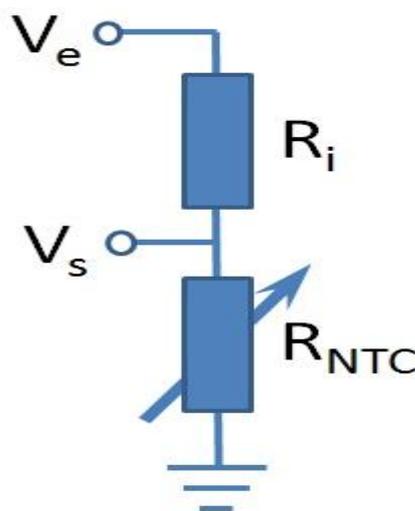


Figura 19 – Circuito para teste com NTC.

¹⁸ V_s = Tensão aplicada no pino 19 do módulo XBee.

3.4 Regulador *Low Dropout* (LDO)

Devido o sensor de gás ser alimentado com 5 Vcc e o módulo XBee com 3,3 Vcc, foi utilizado um Regulador LDO LM1117, para transformar a tensão de 5 Vcc em 3,3 Vcc.

Este regulador mantém a tensão de saída constante independente das variações da carga, temperatura e da tensão de entrada, por isto foi escolhido.

3.5 Circuito do Sensor de Gás

A figura 20 mostra o circuito completo do sensor de gás. O coordenador demonstrado na figura 20 encontra-se instalado no interior do *gateway*.

Na saída do sensor IDG-193 foi colocado um resistor fixo de 2,2 k Ω e um variável de 10 k Ω que servem para ajustar a sensibilidade do referido sensor. Optou-se por colocar um resistor variável, pois cada sensor IDG-193 tem uma resistência característica e isto influencia no cálculo do resistor de saída.

No pino 19 temos a informação de uma tensão analógica referente à temperatura do ambiente em que está o conjunto.

No pino 20 do XBee entra uma tensão analógica proveniente do sensor IDG-193, indicando se há ou não gás no ambiente.

As figuras 21 e 22 mostram a prova de conceito desenvolvida e nas figuras 23, 24 25 e 26 é mostrado o protótipo a ser aplicado na rede subterrânea de distribuição de gás.

O protótipo foi construído da maneira que caso ocorra a queima de algum componente do sensor, basta realizar a substituição do seu refil interno, assim o conjunto não ficará fora de operação até que seja realizado o conserto ou a troca do componente defeituoso, como pode ser observado na figura 25.

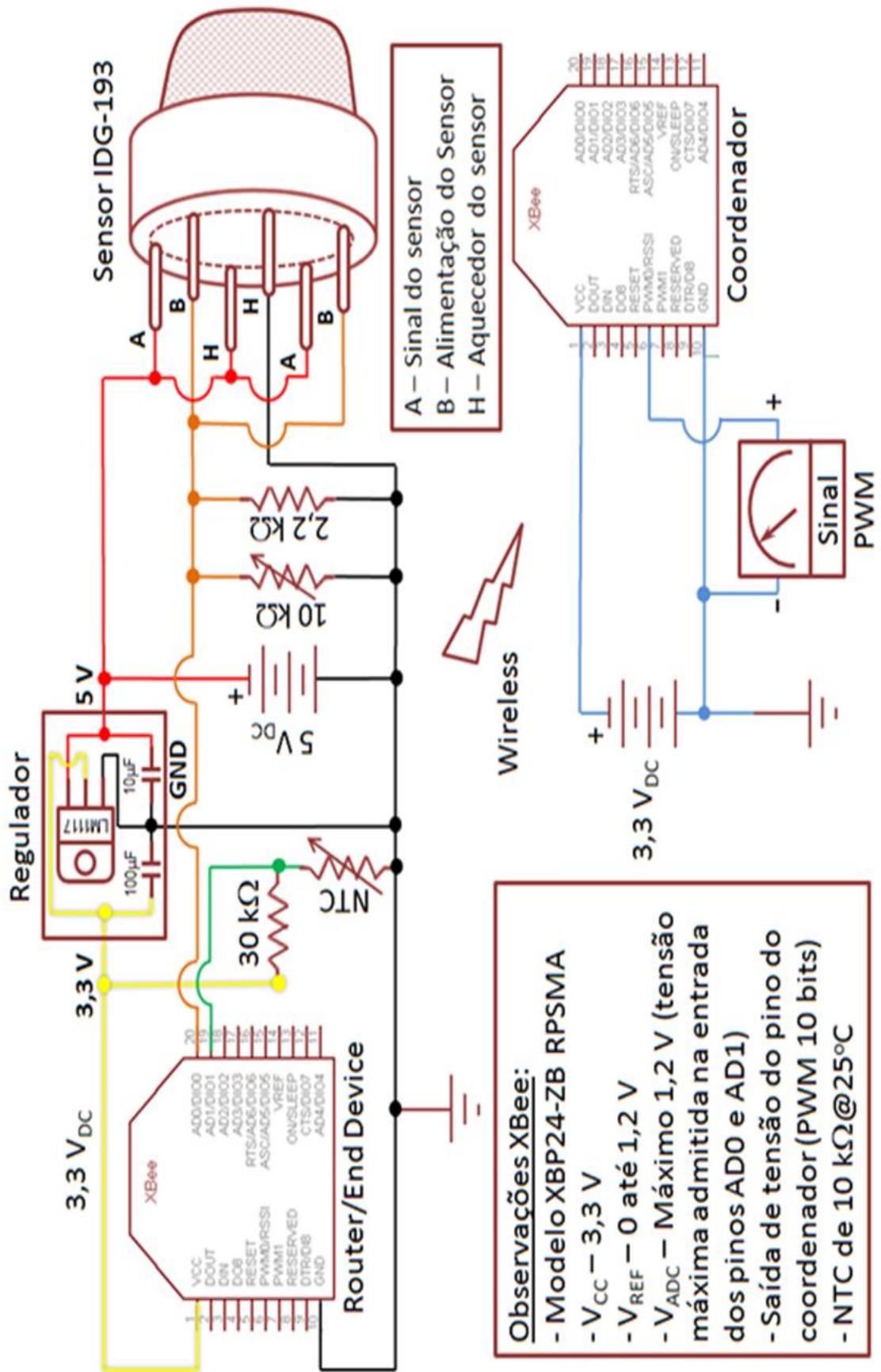


Figura 20 – Circuito completo do sensor de gás.

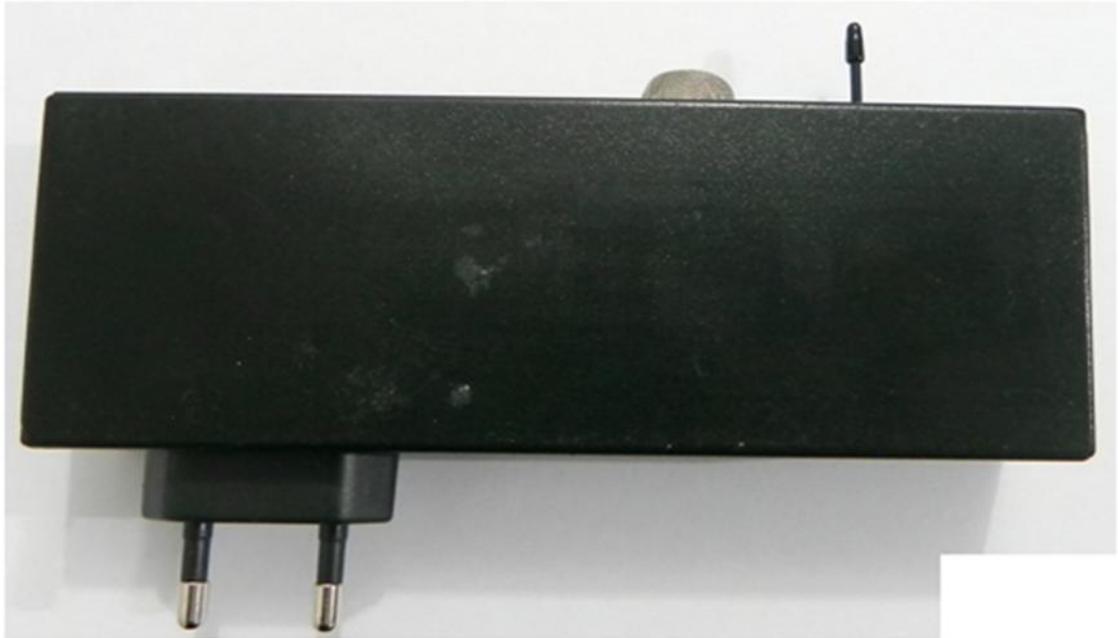


Figura 21 – Vista externa da prova de conceito do sensor de gás.

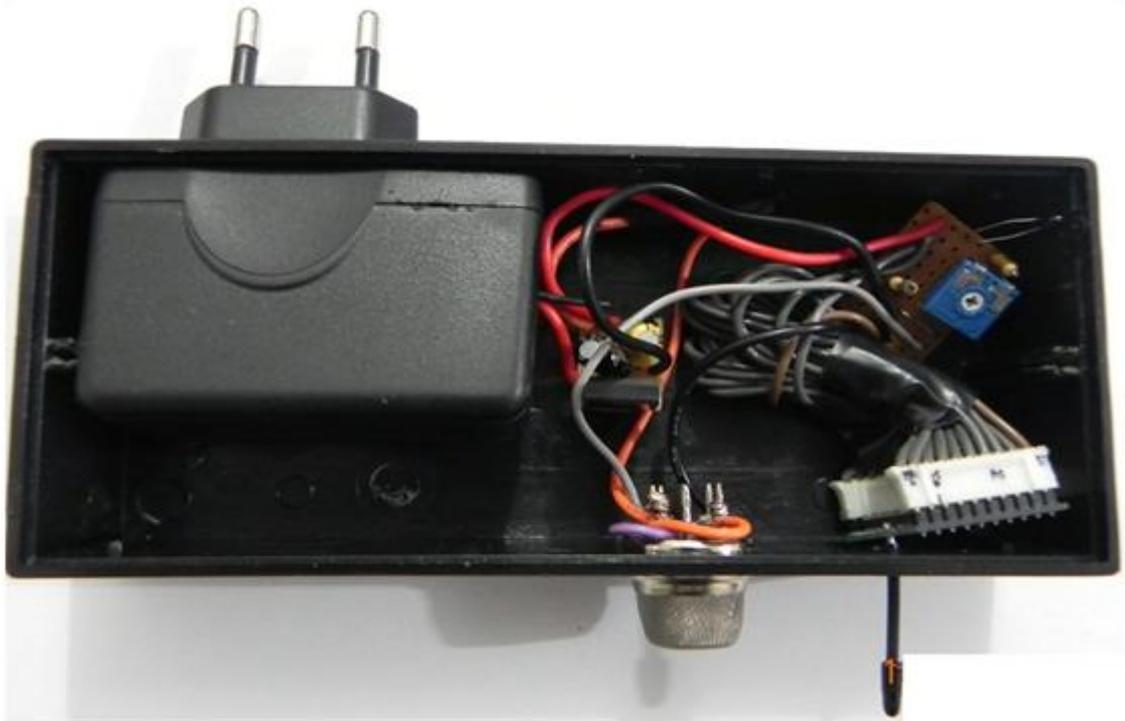


Figura 22 – Vista interna da prova de conceito do sensor de gás.



Figura 23 – Vista externa do protótipo.



Figura 24 – Vista externa do sensor de gás.



Figura 25 – Refil interno do protótipo.

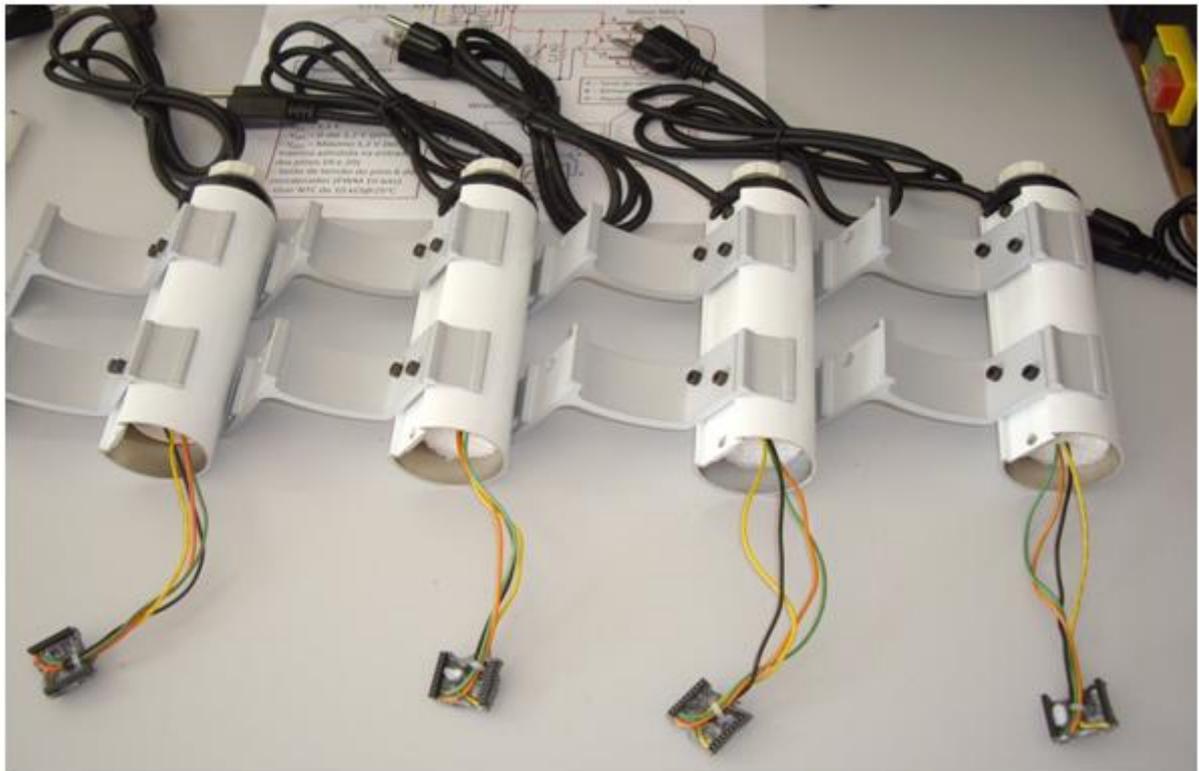


Figura 26 – Protótipo desmontado.

4 GATEWAY

As informações detalhadas neste capítulo foram baseadas no manual do equipamento. O *gateway* utilizado neste trabalho foi o ConnectPort X4 fabricado pela Digi. Segundo o manual do equipamento, o ConnectPort X4 fornece funções de *gateway* para várias tecnologias de rede, como *Ethernet*, celular 3G, Wi-Fi e XBee. Além de fornecer conectividade de rede IP entre celular, Wi-Fi e redes *Ethernet*, ele é projetado para conceber conectividade remota para redes XBee, assim como outros dispositivos conectados às portas locais como a USB e a serial.

Pode ser gerenciado através da nuvem iDigi, com opção de configuração remota de todos os nós da sua rede. Também pode ser configurado através de linhas de comando pela porta serial, telnet ou SSH, ou gerenciado através do protocolo *Simple Network Management Protocol* (SNMP) com o auxílio de um *Network Management System* (NMS).

O *gateway* fornece ainda um ambiente de desenvolvimento Python, permitindo personalizar aplicações dentro do *gateway* para as funcionalidades requeridas.

Possibilidade de ser configurado para funcionar como um servidor *Dynamic Host Configuration Protocol* (DHCP), atribuindo endereços IP automaticamente. O endereço IP padrão do *gateway* é 192.168.1.1. Através deste endereço pode ser acessada a sua interface *web*. Também é possível reservar endereços IP através do MAC dos dispositivos.

Permite configurações de envio de alarmes via *Short Message Service* (SMS), e-mail ou *trap* SNMP quando algum evento ocorra, por exemplo: caso o sinal caia abaixo de um nível especificado por um determinado tempo, ou quando uma grande quantidade de dados via celular é trocada.

Internamente, o *gateway* é composto por um módulo sem fio XBee Pro, que tem a função de coordenador da rede. As figuras 27 e 28 mostram a vista externa e interna do *gateway*.

Oferece os seguintes recursos de segurança e criptografia:

- a) Uma senha, um nível de permissão;
- b) Senhas emitidas para diferentes usuários;
- c) Serviços de rede como: ADDP, *RealPort*, *Encrypted RealPort*, HTTP/HTTPS, LPD, *Remote Login*, *Remote Shell*, SNMP, e Telnet;
- d) Controle de acesso às portas de entrada;

- e) Acesso para dispositivos específicos, endereços IP ou redes, através de filtragem de IP;
- f) SSL v3.0, TLS v1.0, baseados em criptografia: DES (64 *bits*), 3DES (192 *bits*), AES (128-/192-/256-*bits*), IPsec ESP (DES, 3DES, AES);
- g) Wi-Fi Protected Access (WEP/WPA/WPA2).



Figura 27 – Vista externa do gateway.



Figura 28 – Vista interna do gateway.

O quadro 7 mostra um resumo das características do ConnectPort X4, segundo o manual do equipamento.

GERAL	
Gestão	Interface web (HTTP/HTTPS), Controle de acesso por senha, Acesso por nuvem iDigi
Protocolos	UDP, TCP, DHCP, SNMP v1.
Segurança	Túnel SSL, SSHv2, FIPS 197 (serial)
VPN	IPsec, DES, 3DES, AES 256 bits, VPN <i>passthrough</i> , GRE <i>forwarding</i>
Antena	1 XBee, 2 Celular
INTERFACES	
USB	1 USB 2.0
Celular	2G/3G/4G / Gobi GSM, EDGE, HSPA / HSPA +, e CDMA 1xRTT-, EV-DO
WiMax	WiMAX (802.16e-2005); 2.3-2.7 GHz (WF3A <i>Profile</i>); Wave2; S-OFDMA MIMO; QPSK, 16 QAM, 64 QAM; 5/10 MHz <i>bandwidth</i> UL/DL
SERIAL	
Portas	1 porta serial RS-232 DB-9M
Velocidade	Até 230 kbps
Sinais Suportados	TXD, RXD, RTS, CTS, DTR, DSR e DCD
ETHERNET	
Portas	1 RJ-45
Taxa de Dados	10/100 Mbps
Wi-Fi (802.11b/g)	
Frequência	2.4 GHz
Taxa de Dados	Até 54 Mbps
Modulação	DBPSK (1 Mbps), DQPSK (2 Mbps), CCK (11, 5.5 Mbps), BPSK (6, 9 Mbps), QPSK (12, 18 Mbps), 16-QAM (24, 36 Mbps), 64-QAM (48, 54 Mbps)
Potência de Transmissão	15 dBm ± 2 dBm
ROTEAMENTO	
Tipos	NAT, <i>Port Forwarding</i> , <i>Access Control List</i>

Quadro 7 – Características do ConnectPort X4.
 Fonte: Manual do *Gateway*.

A figura 29 mostra a interface web do *gateway*.

The screenshot displays the web interface of a gateway device. On the left is a navigation menu with categories: Home, Configuration (Network, Mobile, XBee Network, Serial Ports, Camera, Alarms, System, iDigi, Users, Position), Applications (Python, RealPort, Industrial Automation), Management (Serial Ports, Connections, Event Logging, Network Services), Administration (File Management, X.509 Certificate/Key Management, Backup/Restore, Update Firmware, Factory Default Settings, System Information, Reboot), and Logout. A 'Help' icon is in the top right. The main content area, titled 'Home', includes a 'Getting Started' section, a 'Tutorial' section with the text 'Not sure what to do next? This Tutorial can help.', and a 'System Summary' section. The system summary lists: Model: ConnectPort X4, Ethernet MAC Address: 00:40:9D:50:E7:A6, Ethernet IP Address: 192.168.0.150, Mobile IP Address: Not Connected, Description: None, Contact: None, Location: None, and Device ID: 00000000-00000000-00409DFF-FF50E7A6.

System Summary	
Model:	ConnectPort X4
Ethernet MAC Address:	00:40:9D:50:E7:A6
Ethernet IP Address:	192.168.0.150
Mobile IP Address:	Not Connected
Description:	None
Contact:	None
Location:	None
Device ID:	00000000-00000000-00409DFF-FF50E7A6

Figura 29 – Interface web do *gateway*.

5 DESENVOLVIMENTO E ANÁLISE DOS RESULTADOS

5.1 Programação em Python

Como foi mencionado anteriormente, o *gateway* fornece um ambiente de desenvolvimento Python em que é possível criar aplicações para fins diversos. Através deste ambiente foi criado um programa para monitorar os nós da rede *mesh* em que os sensores estão inseridos.

O programa desenvolvido funciona de uma forma bem simples. Ele monitora os valores de tensão nos pinos 19 e 20 dos módulos XBee e esta informação é enviada via *gateway* a uma central de monitoramento onde estes dados serão tratados e verificado se algum sensor detectou a presença de gás.

Conforme a figura 20, os dados recebidos no pino 19 são referentes à temperatura no ambiente em que o sensor está e os dados recebidos no pino 20 referem-se aos dados que indicam se há presença de gás ou não.

Os valores de tensão nos pinos 19 e 20, fornecidos pelo circuito da figura 20 são analógicos, então o módulo XBee utiliza um ADC de 10 *bits* para converter estes valores em digitais para que sejam recebidos pelo programa.

O Apêndice A contém o programa implementado.

5.2 Configuração dos Módulos

Neste capítulo serão mostradas as configurações utilizadas nos módulos XBee presentes nos sensores de gás para comunicação com o *gateway*. O quadro 8 mostra estas configurações.

(continua)

PARÂMETRO	VALOR	DESCRIÇÃO
<i>Extended PAN ID</i> (ID)	0x0000000000000000	Permite o XBee aderir a qualquer PAN ID
<i>Node Identifier</i> (NI)	Sensor X	Nome dado ao módulo
<i>Discover Timeout</i> (NT)	60 (x 100 ms)	Tempo que o módulo espera para enviar a resposta quando for dado um comando de descobrimento de nós pelo coordenador.
<i>Scan Channels</i> (SC)	0x7fff	Define que o XBee irá procurar em todos os canais para achar uma rede válida.
<i>Scan Duration</i> (SD)	3	Tempo na qual o XBee ficará escutando <i>beacons</i> em cada canal. $tempo = 2^{parâmetro} \times 15,36 \text{ ms}$
<i>Transmit Power Level</i> (PL)	4	Nível de potência do sinal de RF, no caso 18 dBm
<i>Node Join Time</i> (NJ)	255	Permite que novos dispositivos entrem na rede.
<i>Broadcast Hops</i> (BH)	0	Define para máximo o número de saltos numa transmissão em <i>broadcast</i> .
<i>Baud Rate</i> (BD)	3	Taxa na qual os dados são enviados pela porta serial. (9600 bps)
<i>Flow Control</i> (D7)	<i>Enable CTS Flow Control</i>	Controle de Fluxo CTS habilitado.
<i>Destination Address</i> (DH/DL)	00:00:00:00:00:00:00:00!	Envio de dados em <i>broadcast</i> .
<i>AD0/DIO0 configuration</i>	2	Define que o pino 20 atuará como um ADC.
<i>AD1/DIO1 configuration</i> (D1)	2	Define que o pino 19 atuará como um ADC.
<i>Encryption Enable</i> (EE)	0	Criptografia AES desabilitada.

(conclusão)

PARÂMETRO	VALOR	DESCRIÇÃO
<i>Power Mode</i> (PM)	1	Habilita o PM, melhorando a sensibilidade da recepção em 1 dB e da transmissão em 2 dB
<i>Sleep Mode</i> (SM)	0	Modo <i>Sleep</i> desabilitado.

Quadro 8 – Configurações dos módulos.

5.3 Análise dos Dados

Para aquisição dos dados foram usados dois sensores, um como roteador e outro como dispositivo final. Os testes foram realizados tanto em ambiente *indoor* como *outdoor* e os dados obtidos ficaram dentro do que se esperava do projeto.

Realizados testes com obstáculos no caminho e todos apresentaram o mesmo resultado. Assim que o sensor detectava a presença do gás, no programa era possível observar esta mudança nos valores obtidos.

Os valores apresentados pelo programa eram em valores ADC de 10 *bits*, sendo assim, o valor mínimo possível é 0 (zero) e máximo são 1023 (um mil e vinte e três).

No caso do pino 20 do XBee, quando ele apresenta o valor de 1023 no programa, a tensão analógica no mesmo é de 1,2 Vcc. Assim o valor normal no programa sem vazamento de gás é em torno de 100, quando há vazamento, este valor vai para até 1023.

No pino 19, quando o programa lê o valor de 1023, a tensão no pino é de 1,2 Vcc.

O quadro 9 mostra os valores de tensão analógico e digital (conversor ADC) no pino 19, lido pelo programa e as respectivas temperaturas.

O Apêndice B mostra os resultados obtidos com o programa em execução com um sensor e o Apêndice C os resultados com dois sensores.

(continua)

Temperatura (°C)	Tensão Analógica (V)	Tensão Digital (V)
20	0,96	819
25	0,84	716

(conclusão)

Temperatura (°C)	Tensão Analógica (V)	Tensão Digital (V)
30	0,70	597
40	0,48	409
50	0,36	307
60	0,25	213
70	0,19	162
80	0,14	119
90	0,09	76
100	0,07	59

Quadro 9 – Valores de tensão no pino 19 do XBee.

Durante a execução do programa foi utilizado o Wireshark para acompanhar a troca dos dados entre os sensores, *gateway* e programa. Na figura 30 são mostrados os dados trocados entre o *gateway* e o programa, sem a ocorrência de vazamento de gás e a figura 31 mostra os dados obtidos com Wireshark, no momento em que ocorreu o vazamento de gás.

No.	Time	Source	Destination	Protocol	Length	Info
134	4.664517	192.168.1.150	192.168.1.232	TELNET	78	Telnet Data
136	4.883923	192.168.1.150	192.168.1.232	TELNET	232	Telnet Data
139	5.086234	192.168.1.150	192.168.1.232	TELNET	183	Telnet Data

```

Frame 136: 232 bytes on wire (1856 bits), 232 bytes captured (1856 bits)
Ethernet II, Src: Digiboar_50:e7:a6 (00:40:9d:50:e7:a6), Dst: Hewlett-_af:03:3f (2c:27:d7:af:03:3f)
Internet Protocol Version 4, Src: 192.168.1.150 (192.168.1.150), Dst: 192.168.1.232 (192.168.1.232)
Transmission Control Protocol, Src Port: telnet (23), Dst Port: 51340 (51340), Seq: 170, Ack: 40, Len: 178
Telnet
  Data: \r\n
  Data: \r\n
  Data: Dados recebidos modo ADC: ( n amostras)\r\n
  Data: \r\n
  Data: SENSOR 01 ---->{'AI1': 708, 'AI0': 327}\r\n
  Data: SENSOR 01 ---->{'AI1': 708, 'AI0': 327}\r\n
  Data: SENSOR 01 ---->{'AI1': 708, 'AI0': 327}\r\n
0000 2c 27 d7 af 03 3f 00 40 9d 50 e7 a6 08 00 45 00 ,'.?.@.P....E.
0010 00 da 00 0a 00 00 40 06 f5 45 c0 a8 01 96 c0 a8 .....@. .E.....
0020 01 e8 00 17 c8 8c ea ea 77 f1 d6 4e b2 28 50 18 ..... w.N.(P.
0030 22 11 04 81 00 00 0d 0a 20 0d 0a 44 61 64 6f 73 "..... ..Dados
0040 20 72 65 63 65 62 69 64 6f 73 20 6d 6f 64 6f 20   recib os modo
0050 41 44 43 3a 20 28 20 6e 20 61 6d 6f 73 74 72 61   ADC: ( n amostra
0060 73 29 0d 0a 20 0d 0a 20 20 53 45 4e 53 4f 52 20   s).. .. SENSOR
0070 30 31 20 2d 2d 2d 2d 3e 7b 27 41 49 31 27 3a 20   01 ----> {'AI1':
0080 37 30 38 2c 20 27 41 49 30 27 3a 20 33 32 37 7d   708, 'AI 0': 327}
0090 0d 0a 20 20 53 45 4e 53 4f 52 20 30 31 20 2d 2d   .. SENS OR 01 --
00a0 2d 2d 3e 7b 27 41 49 31 27 3a 20 37 30 38 2c 20   -->{'AI1 ': 708,
00b0 27 41 49 30 27 3a 20 33 32 37 7d 0d 0a 20 20 53   'AI0': 3 27}.. S
00c0 45 4e 53 4f 52 20 30 31 20 2d 2d 2d 2d 3e 7b 27   ENSOR 01 ---->{'
00d0 41 49 31 27 3a 20 37 30 38 2c 20 27 41 49 30 27   AI1': 70 8, 'AI0'
00e0 3a 20 33 32 37 7d 0d 0a   : 327}..

```

Figura 30 – Dados sem vazamento de gás.

No.	Time	Source	Destination	Protocol	Length	Info
179	7.644669	192.168.1.150	192.168.1.232	TELNET	183	Telnet Data ...
182	7.863384	192.168.1.150	192.168.1.232	TELNET	183	Telnet Data ...
184	8.066186	192.168.1.150	192.168.1.232	TELNET	185	Telnet Data ...

Frame 184: 185 bytes on wire (1480 bits), 185 bytes captured (1480 bits)						
Ethernet II, Src: Digiboar_50:e7:a6 (00:40:9d:50:e7:a6), Dst: Hewlett-_af:03:3f (2c:27:d7:af:03:3f)						
Internet Protocol Version 4, Src: 192.168.1.150 (192.168.1.150), Dst: 192.168.1.232 (192.168.1.232)						
Transmission Control Protocol, Src Port: telnet (23), Dst Port: 51340 (51340), Seq: 2154, Ack: 40, Len: 131						
Telnet						
Data: SENSOR 01 ---->{'AI1': 709, 'AI0': 954}\r\n						
Data: SENSOR 01 ---->{'AI1': 709, 'AI0': 1023}\r\n						
Data: SENSOR 01 ---->{'AI1': 709, 'AI0': 1023}\r\n						


```

0000 2c 27 d7 af 03 3f 00 40 9d 50 e7 a6 08 00 45 00  ,...?.@ .P....E.
0010 00 ab 00 19 00 00 40 06 f5 65 c0 a8 01 96 c0 a8  .....@. .e.....
0020 01 e8 00 17 c8 8c ea ea 7f b1 d6 4e b2 28 50 18  ..... ..N.(P.
0030 22 11 c0 ad 00 00 20 20 53 45 4e 53 4f 52 20 30  "..... SENSOR 0
0040 31 20 2d 2d 2d 2d 3e 7b 27 41 49 31 27 3a 20 37  1 ---->{'AI1': 7
0050 30 39 2c 20 27 41 49 30 27 3a 20 39 35 34 7d 0d  09, 'AI0 ': 954}.
0060 0a 20 20 53 45 4e 53 4f 52 20 30 31 20 2d 2d 2d  . SENSO R 01 ---
0070 2d 3e 7b 27 41 49 31 27 3a 20 37 30 39 2c 20 27  ->{'AI1': 709, '
0080 41 49 30 27 3a 20 31 30 32 33 7d 0d 0a 20 20 53  AI0': 10 23}.. s
0090 45 4e 53 4f 52 20 30 31 20 2d 2d 2d 2d 3e 7b 27  ENSOR 01 ---->{'
00a0 41 49 31 27 3a 20 37 30 39 2c 20 27 41 49 30 27  AI1': 70 9, 'AI0'
00b0 3a 20 31 30 32 33 7d 0d 0a                          : 1023}.. .

```

Figura 31 – Dados sem vazamento de gás.

Os dados foram trocados através de Telnet na porta 23, tendo como endereço fonte o endereço 192.168.1.150, o qual foi configurado no *gateway*, e o endereço de destino 192.168.1.232, que era o computador que continha o programa em Python.

Os valores ADC lidos pelo programa no pino 19 (AI1) do módulo XBee, são referentes a temperatura do ambiente em que está o sensor. No caso das figuras 30 e 31, no momento do teste, a temperatura estava entre 25 e 30° C.

Analisando a figura 31 nota-se que quando existe a presença de gás no ambiente em que está instalado o sensor, o valor ADC no pino 20 (AI0) do XBee, começa a elevar até saturar em 1023, que é o valor limite do ADC de 10 *bits*.

6 CONSIDERAÇÕES FINAIS

Diante do que estabelece a Resolução ANP nº 6, de 3/2/2011 e o RTDT da ANP nº 2/2011, em relação a Sistemas de Detecção de Vazamento de Gás, este trabalho propôs um método de detecção em tempo real de vazamento de gás natural através de um padrão de comunicação sem fio robusto e com baixo custo.

Os sensores de um modo geral têm o seu funcionamento muito simples, com componentes facilmente encontrados no comércio específico.

O padrão ZigBee adaptou-se bem à solução proposta para o problema, devido as vantagens que o mesmo oferece em relação a outras tecnologias, como visto no capítulo 2. A topologia da rede em malha oferece vários caminhos entre os dispositivos ZigBee, diminuindo os pontos de falha, disponibilizando uma maior flexibilidade e alcance, podendo ser usada em um amplo espaço geográfico, como por exemplo, em grandes centros urbanos.

Os primeiros resultados encontrados, quando a prova de conceito estava sem o sensor de temperatura instalado, não foram satisfatórios, pois no momento que ocorria uma variação de temperatura, originava-se um falso alarme de vazamento de gás. Os resultados encontrados após a instalação do sensor de temperatura ficaram dentro do que se esperava, funcionando como um alarme em tempo real de vazamento de gás.

Para trabalhos futuros, poderá ser realizada a integração entre o sensor de gás e o software de monitoramento que será alocado em uma central de observação da rede de distribuição de gás. Também poderá ser realizado testes *in loco* na rede de distribuição, com aplicação real, para que seja confirmada a eficácia apresentada neste trabalho. Poderá também ser desenvolvido um estudo para que os sensores e *gateway* disponham de uma fonte de energia de emergência, para os momentos que a energia elétrica não esteja presente. É aconselhável a calibração dos sensores, tanto do sensor de gás utilizado como do sensor de temperatura, antes dos mesmos serem testados em uma aplicação real. Outro fato a ser considerado é a integração entre sensores, *gateway*, central de monitoramento e um software de localização onde os sensores estejam instalados, como o Google Maps, para que quando exista algum alarme de vazamento, este informe a localização exata do ocorrido.

REFERÊNCIAS BIBLIOGRÁFICAS

ABEGÁS. Demanda de gás natural aumenta em fevereiro. Disponível em: <<http://www.abegas.org.br/Site/?p=1406>>. Acesso em: 28 fev.. 2012.

ABEGÁS. Gasodutos. Disponível em: < http://www.abegas.org.br/Site/?page_id=842>. Acesso em: 28 fev. 2012.

BRASIL, Diário Oficial da União de 07 fev. 2011. Seção 1. p. 92. Disponível em: <<http://www.in.gov.br/imprensa/visualiza/index.jsp?jornal=1&pagina=92&data=07/02/2011>>. Acesso em: 28 fev. 2012.

BRASIL. Regulamento Técnico de Dutos Terrestres para Movimentação de Petróleo, Derivados e Gás Natural (RTDT). Disponível em: < www.anp.gov.br/?dw=42266> Acesso em: 28 fev. 2012.

BRASIL. Resolução ANP N° 6, de 3 de fevereiro de 2011 – DOU de 07 de fevereiro de 2011. Disponível em: <[http://nxt.anp.gov.br/nxt/gateway.dll/leg/resolucoes_anp/2011/fevereiro/ranp%20%20-%202011.xml?f=templates\\$fn=document-frame.htm\\$3.0\\$q=\\$x](http://nxt.anp.gov.br/nxt/gateway.dll/leg/resolucoes_anp/2011/fevereiro/ranp%20%20-%202011.xml?f=templates$fn=document-frame.htm$3.0$q=$x)>. Acesso em: 28 fev. 2012.

AZEVEDO, Tiago. **Roteamento ZigBee**. 2006. Disponível em: <<http://www.gta.ufrj.br/ensino/CPE825/2006/resumos/TrabalhoZigbee.pdf>>. Acesso em 07 abr. 2012.

CANSADO, Jacinto C. A. **Instrumentalização de Redes de Sensores Sem Fio**. 2011. 181 f. Tese (Doutorado em engenharia Elétrica) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2011. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/3/3141/tde-26032012-113031/pt-br.php>>. Acesso em 20 mai. 2012.

COMPUTUTORIALS. Módulos XBee Lo básico. Disponível em: <<http://compututorials.blogspot.com.br/2011/11/modulos-xbee-lo-basico.html>>. Acesso em: 10 abr. 2012.

COUTO, Fillipe L.; FIGUEREDO, Luis F. C. **Medição Móvel de Conforto Térmico para Rede de Automação Predial Wireless**. 2008. 85 f. Projeto de Graduação (Grau de Engenheiro de Controle e Automação) – Universidade de Brasília, Brasília, 2008. Disponível em: <<http://lara.unb.br/~adolfo/tg/TG08%20Fillipe%20L.%20Couto%20e%20Luis%20F.C.%20FIGUEREDO.pdf>>. Acesso em 20 mai. 2012.

CARDOSO, Fernando R. M.; SOARES, José C. T. M. **Método para Implementação de Redes Sem Fio**. 2005. 132 f. Projeto Final de Engenharia Elétrica (Estágio Supervisionado) – Universidade de Brasília, Brasília, 2005. Disponível em: <<http://bdm.bce.unb.br/handle/10483/875>>. Acesso em 25 mai. 2012.

EETIMES. The ISM Revolution: The Next Big Thing. 2009. Disponível em:
<<http://www.eetimes.com/design/microwave-rf-design/4019022/The-ISM-Revolution-The-Next-Big-Thing>>. Acesso em 05 abr. 2012.

ESCHNER, Rodrigo H. **Sistema de Automação Residencial Baseado em Sensores ZigBee**. 2011. 38 f. Trabalho de Diplomação (Graduação em Ciências da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2011. Disponível em:
<<http://www.lume.ufrgs.br/bitstream/handle/10183/37175/000819983.pdf?sequence=1>>. Acesso em: 20 mai. 2012.

GTA UFRJ, ZigBee. VASQUES, Bruna L. R. P.; COUTINHO, Igor B. A.; LIMA, Manuela F. ; CARNEVAL, Vitor P. O. ZigBee. Universidade Federal do Rio de Janeiro. Disponível em: <http://www.gta.ufrj.br/grad/10_1/zigbee/index.html>. Acesso em: 20 mai.2012.

GTA UFRJ . Introdução ZigBee. UFRJ. Disponível em:
<http://www.gta.ufrj.br/grad/07_1/zigbee/introducao.html>. Acesso em: 20 mai. 2012.

MANUAL do Gateway ConnectPort X Gateways. Disponível em:
<http://www.digi.com/pdf/ds_connectportx.pdf>. Acesso em: 05 abr. 2012.

MANUAL do XBee/XBee-Pro OEM RF Modules Disponível em:
<ftp://ftp1.digi.com/support/documentation/90000982_A.pdf >. Acesso em 28 fev. 2012.

MONSIGNORE, Ferdinando. **Sensoriamento de Ambiente Utilizando o Padrão ZigBee**. 2007. 92f. Dissertação (Mestre em Engenharia Elétrica) – Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos, 2007. Disponível em:
<<http://www.teses.usp.br/teses/disponiveis/18/18152/tde-27042007-102640/pt-br.php>>. Acesso em 05 abr. 2012.

SANTOS, Sergio T. **Redes de Sensores Sem Fio em Monitoramento e Controle**. 2007. 87f. Dissertação (Mestre em Ciências em Engenharia Elétrica) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2007. Disponível em:
<<http://www.gta.ufrj.br/ftp/gta/TechReports/Sergio07/Sergio07.pdf>>. Acesso em: 05 abr. 2012.

ÚLTIMO Segundo IG. Explosão em bueiro queima turistas no Rio. Disponível em:
<<http://ultimosegundo.ig.com.br/brasil/rj/explosao+em+bueiro+queima+turistas+no+rio/n1237686896478.html>>. Acesso em: 28 fev. 2012.

ÚLTIMO Segundo IG. Explosão de bueiros: causas permanecem desconhecidas. Disponível em:
<<http://ultimosegundo.ig.com.br/brasil/rj/explosoes+de+bueiros+causas+permanecem+desconhecidas/n1597077473565.html>>. Acesso em: 28 fev. 2012.

US.SENSOR NTC Resistance – Temperature Chart. Disponível em:
<<http://www.ussensor.com/rt%20charts/DC103J2K.htm>>. Acesso em: 07 mar. 2012.

ZIGBEE Alliance. Disponível em: <<http://www.zigbee.org>>. Acesso em 28 fev. 2012.

ZUCATO, Fábio L. **Rede ZigBee Gerenciada por Sistema de Monitoramento Remoto Utilizando TCP/IP e GPRS**. 2009. 138 f. Dissertação (Mestrado em Engenharia Elétrica) – Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos, 2009.
Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18155/tde-03022010-095135/en.php>> Acesso em: 20 mai.2012.

Apêndice A – Programa Python

```
#####
# This is the main project module
# Created on: 14 April 2012
# Author: fabio
# Description:
#####

#importando as bibliotecas
import zigbee
import binascii
import struct
import xbee

# Endereçando o gateway e os sensores
GATEWAY="[00:13:a2:00:40:70:9c:25]!"
SENSOR1="[00:13:a2:00:40:3b:8e:2c]!"
SENSOR2="[00:13:a2:00:40:98:10:04]!"
#Fim do endereçamento

# Início do parseamento dos dados
#definindo a função parseIS
#Retorna os valores de cada DIO
def parseIS(data):
#Caso o módulo seja da Série 2 entra no if
    if len(data) % 2 == 0:
#Descompacta os dados do pacote. data:4, os quatro primeiros bits
        sets, datamask, analogmask = struct.unpack("!BHB", data[:4])
#todos os bits menos os 4 primeiros
        data = data[4:]

#Caso o módulo seja da Série 1 entra aqui
    else:
#Descompacta os dados do pacote
        sets, mask = struct.unpack("!BH", data[:3])
        data = data[3:]
# Move os primeiros 9 bits para uma máscara separada
        datamask = mask % 512
#Move os últimos 7 bits para uma máscara separada
        analogmask = mask >> 9

    retdir = { }
```

```

if datamask:
    datavals = struct.unpack("!H", data[:2])[0]
    data = data[2:]

    currentDI = 0
    while datamask:
        if datamask & 1:
            retdir["DIO%d" % currentDI] = datavals & 1
#Deslocamento de bit para a direita
            datamask >>= 1
            datavals >>= 1
            currentDI += 1

    currentAI = 0

    while analogmask:
        if analogmask & 1:
            aval = struct.unpack("!H", data[:2])[0]
            data = data[2:]

            retdir["AI%d" % currentAI] = aval
            analogmask >>= 1
            currentAI += 1

    return retdir

# Fim do parseamento dos dados

#primeira leitura dos dados dos sensores
#Imprimindo dados do Sensor1
print " "
print "Dados Sensor 01:"
print " "

#Obtém os valores dos parâmetros de um objeto Digi
dado_bruto1 = zigbee.ddo_get_param(SENSOR1, 'IS')
dados1 = parseIS( dado_bruto1 )
print dados1
#Fim dos dados do Sensor1

#Imprimindo dados do Sensor2
#print " "
#print "Dados Sensor2:"
#print " "

dado_bruto2 = zigbee.ddo_get_param(SENSOR2, 'IS')
dados2 = parseIS( dado_bruto2 )
print dados2
#Fim dos dados Sensor2

```

```
print " "  
print "Dados recebidos no modo ADC:"  
print " "  
#Número de vezes que roda o programa. Laço para ler os dados.  
runs = 100  
while runs:  
  
    dado_bruto1 = zigbee.ddo_get_param(SENSOR1, 'IS')  
    dados1 = parseIS( dado_bruto1 )  
    dado_bruto2 = zigbee.ddo_get_param(SENSOR2, 'IS')  
    dados2 = parseIS( dado_bruto2 )  
    print " SENSOR 01 ---->", dados1  
    print " SENSOR 02 ---->", dados2  
    #decrementa o runs  
    runs -= 1  
  
#Fim do programa
```

Apêndice B – Resultados do Programa com 1 Sensor

```
#> python dpdsrv.py
```

```
Launching Python application 'sensor_gas_fabio.py' ...
\x00
```

Dados classificados Sensor 01:

```
{'AI1': 758, 'AI0': 323}
```

Dados recebidos modo ADC: (n amostras)

```
SENSOR 01 ---->{'AI1': 758, 'AI0': 323}
SENSOR 01 ---->{'AI1': 758, 'AI0': 323}
SENSOR 01 ---->{'AI1': 758, 'AI0': 323}
SENSOR 01 ---->{'AI1': 758, 'AI0': 324}
SENSOR 01 ---->{'AI1': 758, 'AI0': 324}
SENSOR 01 ---->{'AI1': 758, 'AI0': 324}
SENSOR 01 ---->{'AI1': 758, 'AI0': 325}
SENSOR 01 ---->{'AI1': 757, 'AI0': 325}
SENSOR 01 ---->{'AI1': 758, 'AI0': 325}
SENSOR 01 ---->{'AI1': 757, 'AI0': 325}
SENSOR 01 ---->{'AI1': 757, 'AI0': 340}
SENSOR 01 ---->{'AI1': 757, 'AI0': 364}
SENSOR 01 ---->{'AI1': 757, 'AI0': 429}# inicio do vazamento de gás
SENSOR 01 ---->{'AI1': 758, 'AI0': 608}
SENSOR 01 ---->{'AI1': 757, 'AI0': 865}
SENSOR 01 ---->{'AI1': 757, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 759, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 789, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 829, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 800, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 764, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 757, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 757, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 757, 'AI0': 1023}
```

SENSOR 01 ---->{'AI1': 757, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 757, 'AI0': 1009}
SENSOR 01 ---->{'AI1': 757, 'AI0': 978}
SENSOR 01 ---->{'AI1': 757, 'AI0': 950}
SENSOR 01 ---->{'AI1': 757, 'AI0': 923}

#>

Apêndice C – Resultados do Programa com 2 Sensores

```
#> python dpdsrv.py
```

```
Launching Python application 'sensor_gas_fabio.py' ...
\x00
```

```
{'AI1': 791, 'AI0': 261}
```

```
{'AI1': 802, 'AI0': 117}
```

Dados recebidos modo ADC: (n amostras)

```
SENSOR 01 ---->{'AI1': 792, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 791, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 792, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 792, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 792, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 792, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 790, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 792, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 791, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 792, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 792, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 792, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 792, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 792, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 792, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 792, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 792, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```

```
SENSOR 01 ---->{'AI1': 792, 'AI0': 261}
```

```
SENSOR 02 ---->{'AI1': 802, 'AI0': 117}
```



```
SENSOR 01 ---->{'AI1': 792, 'AI0': 1023}
SENSOR 02 ---->{'AI1': 802, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 792, 'AI0': 1023}
SENSOR 02 ---->{'AI1': 802, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 792, 'AI0': 1023}
SENSOR 02 ---->{'AI1': 802, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 792, 'AI0': 1023}
SENSOR 02 ---->{'AI1': 802, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 792, 'AI0': 1023}
SENSOR 02 ---->{'AI1': 802, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 792, 'AI0': 1023}
SENSOR 02 ---->{'AI1': 802, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 792, 'AI0': 1023}
SENSOR 02 ---->{'AI1': 802, 'AI0': 1023}
SENSOR 01 ---->{'AI1': 792, 'AI0': 1023}
SENSOR 02 ---->{'AI1': 802, 'AI0': 1023}
```

#>