

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

**IMPLANTAÇÃO DE REDE INDUSTRIAL WIRELESS
PARA O MONITORAMENTO E CONTROLE
AUTOMÁTICO DE MOVIMENTAÇÃO DE
AERONAVES EM AEROPORTOS**

TRABALHO DE CONCLUSÃO DE CURSO

Anderson Pereira Colvero

Santa Maria, RS, Brasil

2012

STRC/UFSM.RS

COLVERO. Anderson Pereira

Tecnólogo em Redes de Computadores

2012

IMPLANTAÇÃO DE REDE INDUSTRIAL WIRELESS PARA O MONITORAMENTO E CONTROLE AUTOMÁTICO DE MOVIMENTAÇÃO DE AERONAVES EM AEROPORTOS

Anderson Pereira Colvero

Trabalho de Conclusão de Curso (TCC) apresentado ao Curso Superior de
Tecnologia em Redes de Computadores do Colégio Técnico Industrial de Santa
Maria, da Universidade Federal de Santa Maria (UFSM,RS), como requisito
parcial para obtenção de grau de
Tecnólogo em Redes de Computadores

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

**IMPLANTAÇÃO DE REDE INDUSTRIAL WIRELESS PARA O
MONITORAMENTO E CONTROLE AUTOMÁTICO DE
MOVIMENTAÇÃO DE AERONAVES EM AEROPORTOS**

elaborado por
Anderson Pereira Colvero

Como requisito parcial para a obtenção de grau de
Tecnólogo em Redes de Computadores

COMISSÃO EXAMINADORA:

Claiton Pereira Colvero, Dr.
(Orientador)

Eugênio de Oliveira Simonetto, Dr. (UFSM)

Murilo Cervi, Dr. (UFSM)

Santa Maria. 06 de julho de 2012

DEDICATÓRIA

Dedico, em primeiro lugar, a Deus, por me proporcionar saúde, apoio e condições adequadas para seguir essa jornada tão importante, nesta fase da vida;

A minha família, Amauri José Colvero (pai) e Edi Teresinha Pereira Colvero (mãe) e irmãos Claiton P. Colvero e Fabricio P. Colvero, por todo e exemplo de criação e convivência, os quais repassaram a verdadeira educação que só um verdadeiro lar pode oferecer, bem como o incentivo para que completasse a graduação, ao Claiton o agradecimento especial também pelo privilégio de tê-lo como professor e orientador, seu apoio foi fundamental;

A minha esposa Sunta Marta Felipetto Colvero, sempre companheira e incentivadora, que sempre me acompanhou nas horas mais difíceis, fazendo o que podia para ajudar, se estou concluindo este curso agora, grande parte do mérito é dela;

Aos meus avós (*in memorian*), Plínio e Angelina Pereira, Antônio e Lúcia Colvero, exemplos de vida, que continuam zelando pelo nosso sucesso;

A minha sogra, Marlene Bandinelli Felipetto, que sempre me apoiou e fez o possível para, dentro das limitações, garantir que houvesse um espaço adequado para estudar;

A meus colegas de aula, pela ajuda no melhor entendimento de conceitos, pelo companheirismo e convivência diária, tornando possível superar barreiras;

A meus colegas de trabalho, que sacrificaram muitas horas das suas vidas, para que eu pudesse assistir às aulas sem prejuízos de trabalho, em especial à Carmem Elisete Gabbi, que idealizou e dedicou-se muito desde a criação do curso, até que fosse concretizado, também pelo incentivo para que eu o realizasse;

Aos meus amigos, parentes e muitos nomes que não foram citados, mas que no coração sempre torceram por mim, todos são igualmente importantes;

Aos professores, que mesmo dentro das limitações físicas e pessoais existentes, fizeram o possível, dentro da sua habilidade e experiência, para que fosse repassado o ensinamento necessário para a carreira e para a vida;

A I-Dutto, que permitiu a confiança de fornecer e treinar o uso em equipamentos de alta tecnologia e valor agregado, em especial ao Sr. Vinícius Carneiro (Gerente) e Sr. Henrique Bonadio (Programador), com os quais mantivemos contato mais direto.

RESUMO

Trabalho de Conclusão de Curso (TCC)
Colégio Técnico Industrial De Santa Maria
Curso Superior de Tecnologia em Redes de Computadores
Universidade Federal de Santa Maria

IMPLANTAÇÃO DE REDE INDUSTRIAL WIRELESS PARA O MONITORAMENTO E CONTROLE AUTOMÁTICO DE MOVIMENTAÇÃO DE AERONAVES EM AEROPORTOS

AUTOR: ANDERSON PEREIRA COLVERO

ORIENTADOR: CLAITON PEREIRA COLVERO

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

Este trabalho apresenta o desenvolvimento de um projeto baseado em comunicação por meio de redes sem fio, utilizando a tecnologia *ZigBee*, a ser aplicado no Aeroporto Santos Dumont – RJ, com a finalidade de permitir o controle do tráfego de veículos próximo à cabeceira da pista, local onde ocorreram acidentes com vítimas. Este trabalho foi elaborado e executado em colaboração com a empresa I-Dutto - Soluções em Localização e Identificação Eletrônica Ltda, sediada na cidade do Rio de Janeiro, a qual forneceu a maior parte dos recursos financeiros e efetuou a aquisição dos equipamentos necessários, bem como o detalhamento das exigências do solicitante. Dentro do cronograma originalmente proposto, foi executada a revisão bibliográfica, foram orçados e adquiridos os equipamentos necessários, como módulos sensores, módulos *XBee* e *gateway* concentrador de comunicação. Na execução do projeto, foi promovida a comunicação *wireless* entre os sensores e o sistema concentrador através da montagem física dos mesmos, e na camada lógica, foi utilizada a linguagem de programação *Python*, bem como o fluxograma do funcionamento de sensoriamento, que serviu para orientar a implementação do programa de gerenciamento de pista através dos procedimentos internos do aeroporto devidamente mapeados. Os resultados obtidos demonstram a viabilidade do projeto aliado à execução dos trabalhos, que contribuíram de forma positiva para o desenvolvimento de uma visão mais ampla do autor em relação ao funcionamento de novas tecnologias, a experiência de lidar com situações reais e realizar a pesquisa com foco profissional e não somente acadêmico.

Palavras-chave: Redes sem fio. *ZigBee*. Sensoriamento.

ABSTRACT

Completion Of Course Work
Colégio Técnico Industrial de Santa Maria
Universidade Federal de Santa Maria

DEPLOYMENT OF WIRELESS NETWORK FOR INDUSTRIAL MONITORING AND AUTOMATIC CONTROL OF AIRCRAFT IN AIRPORT DRIVE

AUTHOR: ANDERSON PEREIRA COLVERO

SUPERVISOR: CLAITON PEREIRA COLVERO

Date and Place of Defense: Santa Maria, June 25, 2012.

This paper presents the development of a project based on communication through wireless networks, using ZigBee technology, to be applied in the Santos Dumont Airport - Rio de Janeiro, in order to allow control of the vehicle traffic near the end of the lane, where accidents had occurred with victims. This work was prepared and implemented in collaboration with the company I-Dutto – Solutions Based on Electronic Tracking and Identification Devices, Inc., headquartered in the city of Rio de Janeiro, which provided the most financial support and bought the necessary equipment, as well as detailing the requirements of the requester. Within the schedule originally proposed, the literature review was performed, the necessary equipment were budgeted and purchased, such as sensor modules, XBee modules and *gateway* hub of communication. On the implementation of the project, was promoted wireless communication between sensors and the system hub through the same physical and logical assembly, we used the Python programming language as well as the flowchart of the operation of sensing, which helped us on the implementation of the track management program, through the internal procedures of the airport properly mapped. The results demonstrate the feasibility of the project together with the execution of the work, which contributed positively to the development of an author broader view regarding the operation of new technologies, the experience of dealing on real situations and conduct a research on professional focus rather than only academic.

Keywords: Wireless network. ZigBee. Sensing.

LISTA DE ILUSTRAÇÕES

Figura 01 - Acidente fatal no Aeroporto S. Dumont. 2002.	16
Figura 02 - Aeroporto Santos Dumont – imagem de satélite editada.....	18
Figura 03 - Cancela de controle de veículos.....	18
Figura 04 - Causas de disparos falsos.....	19
Figura 05 - Funcionamento do RFID	23
Figura 06 - Sensoriamento Infravermelho.....	25
Figura 07 - Gráfico do sensor infravermelho com e sem ATC	25
Figura 08 - Frequências utilizadas pelo padrão <i>ZigBee</i>	29
Figura 09 - Pilha de Protocolo <i>ZigBee</i>	30
Figura 10 - Pilha de Protocolo <i>ZigBee</i>	30
Figura 11 - Frequências utilizadas pela camada física <i>ZigBee</i>	32
Figura 12 - Características de cada padrão adotado	32
Figura 13 - Formato do quadro PDU.....	32
Figura 14 - Formato geral do quadro da camada de enlace.....	33
Figura 15 - Estrutura <i>Superframe</i> – atuação com <i>beacons</i>	34
Figura 16 - Quadro <i>Beacon</i>	35
Figura 17 - Quadro de dados	35
Figura 18 - Quadro de confirmação.....	36
Figura 19 - Quadro de MAC de comando	36
Figura 20 - Topologias de trabalho do <i>ZigBee</i>	38
Figura 21 - Padrões <i>Wireless</i> e sua área de atuação	39
Figura 22 - Comunicação da UART com a interface do computador	41
Figura 23 - Transmissão do pacote de dados da UART através do transmissor	41
Figura 24 - Estrutura do frame API.....	42
Figura 25 - Exemplo de frame API	43
Figura 26 - Módulo de <i>ZigBee XBee Pro S2</i>	44
Figura 27 - Interface RS-232 (serial) para configuração do módulo <i>XBee</i>	46
Figura 28 - Interface CON-USBBEE	47
Figura 29 - Interface CON-USBBEE montada com o módulo	47
Figura 30 - Estrutura do comando AT.....	48
Figura 31 - Programa X-CTU.....	49
Figura 32 - Programa X-CTU.....	50
Figura 33 - Programa X-CTU.....	51
Figura 34 - Programa X-CTU.....	52
Figura 35 - Subdivisão da camada física 802.11	53
Figura 36 - <i>Gateway Connectport X4</i>	54
Figura 37 - <i>Gateway X4</i> – visão interna	55
Figura 38 - Tela inicial do <i>ConnectPort X4</i>	56
Figura 39 - Configuração da rede do <i>ConnectPort X4</i>	57
Figura 40 - Configuração DHCP do <i>ConnectPort X4</i>	57
Figura 41 - Configuração de serviços do <i>ConnectPort X4</i>	58
Figura 42 - Configuração <i>XBee</i> no <i>ConnectPort X4</i>	58
Figura 43 - Armazenamento de arquivos no <i>ConnectPort X4</i>	59
Figura 44 - Início automático de programas no <i>ConnectPort X4</i>	60
Figura 45 - Log de atividade do <i>ConnectPort X4</i>	60
Figura 46 - Tela inicial do sistema de desenvolvimento em “nuvem” da <i>Digi.Inc</i>	62
Figura 47 - Configuração do servidor do ambiente <i>iDigi</i>	63

Figura 48 - Tela do <i>ConnectPort X4</i> disponível na nuvem <i>iDigi</i>	63
Figura 49 - Tela de acesso aos dispositivos <i>XBee</i> conectados na rede.....	64
Figura 50 - Plataforma multifuncional <i>Digi Esp</i>	66
Figura 51 - Interpretador <i>Python</i>	67
Figura 52 - Interpretador <i>Python</i>	68
Figura 53 - Sensor de presença externo Bosch OD 850	69
Figura 54 - Módulos internos dos sensores de presença	70
Figura 55 - Módulo Fonte de Alimentação	71
Figura 56 - Módulo de Radiofrequência (comunicação de dados).....	71
Figura 57 - Módulo de Sensor de Presença	72
Figura 58 - Módulo de RF dos sensores e fonte de alimentação	73
Figura 59 - Módulo de RF dos sensores e unidades prontas para uso	73
Figura 60 - Topologia de rede do prevista para o aeroporto	74
Figura 61 - Testes em ambiente <i>outdoor</i>	75
Figura 62 - Testes em ambiente <i>outdoor</i>	75
Figura 63 - Tela de aquisição dos sensores	77
Figura 64 - Resumo do fluxo geral x Telnet entre cliente e servidor	79
Figura 65 - Filtro de separação do fluxo de <i>upload</i> para o <i>Gateway</i>	80
Figura 66 - Estatística de comprimento dos pacotes de dados	81
Figura 67 - Momento de chamada do programa no <i>Gateway X4</i>	82
Figura 68 - Momento de chamada do programa no <i>Gateway X4</i>	82

SUMÁRIO

1. INTRODUÇÃO	10
2. OBJETIVOS	12
3. METODOLOGIA.....	13
4. ANÁLISE DO AMBIENTE OPERACIONAL.....	14
4.1. Aeroporto Internacional Santos Dumont – um ambiente complexo	15
4.2. Exigências do solicitante	17
5. PROPOSTA DE SOLUÇÃO PARA O PROBLEMA.....	18
6. ANÁLISE DOS PROVÁVEIS DISPOSITIVOS A SEREM UTILIZADOS	22
7. DETERMINAÇÃO DO MEIO DE TRANSMISSÃO ENTRE OS SENSORES E O GATEWAY	27
7.1. Meio guiado:.....	27
7.2. Meio não guiado:	27
8. A TECNOLOGIA <i>ZIGBEE</i>	28
8.1. Padrão de frequências adotado no mundo:.....	29
8.2. Distribuição das camadas do protocolo:.....	30
8.2.1. Norma 802.15.4: Definição das camadas	31
8.2.1.1. Camada Física:	31
8.2.1.2. Camada de Enlace:	33
8.2.1.2.1. Modos de operação: Beaconing e Non-Beaconing	33
8.2.1.2.1.1. Modo Beaconing:	33
8.2.1.2.1.2. Modo Non-Beaconing:	35
8.2.1.2.2. Quadros de comunicação padrão da camada de enlace:	35
8.2.1.2.3. Endereçamento:	36
8.2.1.2.4. Segurança:	37
8.2.1.3. Camada de Rede:	37
8.2.1.3.1. Topologias de rede:	38
8.2.1.4. Camadas de Suporte a Aplicação e Camada de Aplicação:	38
8.3. Distribuição do padrão <i>Zigbee</i> dentro das redes por área de atuação:	39
8.4. Classificação dos dispositivos quanto à função na rede:.....	40
8.5. Classificação dos dispositivos quanto ao tipo:	40
8.6. Modos de Operação AT e API:	41
8.7. Formas de difusão das mensagens:	43
9. <i>ZIGBEE DIGI - XBEE PRO S2B</i>.....	44
9.1. Configuração do dispositivo:	46
9.2. Configurando o programa X-CTU:	49
10. GATEWAY DIGI CONNECTPORT X4 ZB HSPA	54
10.1. Configuração do <i>Gateway</i>:	55
11. KIT DE DESENVOLVIMENTO IDIGI PARA O <i>GATEWAY CONNECTPORT X461</i>	61
11.1. Ambiente de desenvolvimento <i>iDigi</i>	61
11.2. Acesso ao ambiente <i>Digi Developer Cloud</i>	61
11.3. IDE de desenvolvimento de programas em <i>Python</i>	65
12. SENSORES DE PRESENÇA BOSCH	69
12.1. Montagem dos sensores de presença:	69

12.2. Detalhes da montagem elétrica/eletrônica dos sensores:	70
12.3. Detalhes da montagem mecânica dos sensores:	72
13. TOPOLOGIA FINAL DA REDE	74
13.1. Teste prático em área livre	75
13.2. Aquisição de dados e análise dos resultados	76
14. CONCLUSÕES.....	84
14.1. Sugestões para pesquisas futuras:.....	86
15. REFERÊNCIAS	87

1. INTRODUÇÃO

O sistema de transporte aéreo, que se tornou popular nas últimas décadas, permitiu a abertura de novos horizontes para a interação entre os povos, a nível mundial. A possibilidade de deslocar pessoas e cargas em um tempo muito reduzido, em relação ao transporte terrestre ou marítimo convencional, permitiu estabelecer novos mercados e relações de negócios.

De forma análoga à maioria das novas modalidades de transporte coletivo, essa também necessitou adaptar-se a um mundo novo e cheio de oportunidades, porém em pleno funcionamento, de maneira a estabelecer o seu espaço no mercado. Em contrapartida, o mundo viu-se obrigado a fornecer condições de executar um serviço de forma organizada e com qualidade, dando origem aos primeiros aeroportos. Assim como ocorre em toda implantação de novas estruturas, a adequação da operação através de adaptações em processos e procedimentos sempre traz consigo diversos problemas associados, alguns imediatos e outros posteriores. Com a aviação não foi diferente: acidentes foram frequentes no início das operações, especialmente pela falta de uma estrutura organizada e amadurecida para tratar dos diversos procedimentos que envolvem o sistema de tráfego aéreo.

A partir da evolução dos aeroportos e da experiência dos administradores, o quesito segurança passou a ser a prioridade, de maneira que foram sendo estabelecidos Regulamentos internos rígidos, que visam padronizar os processos e procedimentos, de forma a evitar falhas. Com esta regulamentação em funcionamento, foi possível reduzir ao máximo o número de acidentes, sendo que o maior desafio atual ainda é o fator humano e as causas naturais, como tempestades, calor, terremotos, entre outros.

As forças da natureza são incontrolláveis e geralmente pontuais, mas hoje elas podem ser previstas com relativa precisão e sempre que possível podem ser contornadas sem que haja algum tipo de perigo à operação normal dos aeroportos. A tecnologia notadamente vem ajudando a minimizar o risco de falhas em relação à limitação do ser humano, tornando os sistemas cada vez mais automáticos e independentes, delegando ao operador especialmente a decisão em casos adversos, ao invés de condicioná-lo a uma rotina de procedimentos repetitivos e manuais que podem levar a falhas ao longo do tempo. Dentro dessa perspectiva, o sensoriamento dos aeroportos pode fornecer recursos valiosos para a manutenção dos serviços, controle e prevenção de acidentes, assim como a obtenção de dados estatísticos que permitam avaliar o funcionamento do sistema.

No Aeroporto Internacional Santos Dumont, no Rio de Janeiro, o sistema de controle e monitoramento de aeronaves em solo em procedimentos de decolagem, com o objetivo de acionar a cancela de obstrução de tráfego do aeroporto, objeto do estudo, hoje é realizado de forma totalmente manual por um operador, conhecido pelo termo “Cochileiro”, de forma a permitir a passagem segura de veículos e pedestres na cabeceira da pista, onde já ocorreram acidentes com vítimas fatais, justamente devido ao erro humano na operação destas tarefas repetitivas. Dessa forma, um sistema automatizado operando em uma rede industrial wireless e com sensores remotos na pista, poderá auxiliar o procedimento de decolagem de aeronaves, ajudando a manter uma operação mais eficaz da barreira sem a intervenção de operador, evitando que vidas se percam por falhas de percepção humana.

2. OBJETIVOS

Dentro da perspectiva de prevenção de erros, alguns objetivos se fazem necessários ao delineamento correto dos rumos a serem tomados na solução do problema:

- Promover a comunicação dos sensores com a central de processamento utilizando rede industrial sem fio, possibilitando o controle das aeronaves em solo;
- Descobrir os recursos mínimos necessários para o bom funcionamento da rede;
- Desenvolver um fluxo de trabalho do sistema que atenda aos requisitos de segurança internacionais e procedimentos internos do aeroporto;
- Analisar o sistema em operação em um ambiente externo com condições ambientais agressivas e com regulamentação interna rígida.

Este trabalho, da forma como foi especificado pelo solicitante, necessitou o embasamento direto nos padrões mais confiáveis de comunicação de redes industriais, devido à peculiaridade do local de trabalho, onde os elementos de controle estão diretamente em contato com ambiente agressivo, predisposto a intempéries típicas de cidades litorâneas, como a elevada salinidade e exposição solar constante muito e intensa. Estes devem ainda ser resistentes ao vento e produtos químicos emanados pelas aeronaves como óleos e combustíveis.

O projeto deve atender ainda a restrições no que se refere à radiação eletromagnética, devido ao funcionamento dos equipamentos de controle de voo, de forma a não causar interferências destrutivas nos sinais constantemente utilizados nos procedimentos.

3. METODOLOGIA

A metodologia do trabalho foi desenvolvida nas diferentes etapas:

- Pesquisa de mercado por uma solução que atendesse aos requisitos.

Após pesquisa de mercado, não foi encontrada uma solução comercial que atendesse completamente ao propósito. Desta forma foi estruturado um conjunto de elementos que pudessem suprir as necessidades de cada parte e então foi realizada a montagem e calibração em separado dos subconjuntos.

- Cotação dos dispositivos de cada subconjunto, de acordo com as características da função específica, como sensores, módulos de transmissão e outros.

- Envio da relação de dispositivos para a I-Dutto – Soluções em Localização e Identificação Eletrônica Ltda., sediada na cidade do Rio de Janeiro – RJ. A I-Dutto é a empresa responsável pelo financiamento deste projeto, fornecendo os recursos materiais para a elaboração e testes iniciais, fazendo a aquisição e envio do material para Santa Maria – RS.

- Recebimento e montagem dos módulos de teste. Foram recebidos quatro sensores de presença e também os dispositivos relativos aos transmissores, como os módulos *ZigBee*. Foi recebido também o *Gateway*, todos da marca Digi. Acompanha o *Gateway* um *kit* de desenvolvimento no site do fabricante.

- Foi estabelecido o funcionamento dos sensores em laboratório com a montagem provisória e testes de detecção de presença em primeira instância. A partir deste estágio foi executada a montagem dos módulos sensores de maneira definitiva e realizados os testes no Parque de Exposições da UFSM, onde foi feita a calibração final e ensaios com ambientes externo mais próximo do objetivo real.

Paralelamente, foi necessária a adaptação de programa em linguagem *Python* para a detecção dos sensores no aplicativo fornecido pela Digi, com resultados binários iniciais em nível baixo e alto (zero e um). Também foi desenvolvido a pedido da I-Dutto um fluxograma funcional do programa que irá controlar o acionamento da cancela do aeroporto de acordo com os procedimentos internos do mesmo.

- O referencial teórico foi essencialmente baseado em pesquisas no site da Digi, monografias, dissertações, teses e sites que tratam de documentação associada ao conteúdo do trabalho, detalhados no capítulo Referências Bibliográficas deste relatório.

- O estágio final do projeto consistiu em enviar o conjunto funcional para a I-Dutto para que seja testado na pista do aeroporto, onde deverá passar por criteriosa avaliação de operação.

4. ANÁLISE DO AMBIENTE OPERACIONAL

Os sistemas de controle do mundo moderno, assim como se previa já no século passado, vêm realmente aos poucos substituindo o trabalho humano, cada vez com mais propriedade, se utilizando de poderosos recursos eletrônicos e da inteligência artificial. As redes de computadores foram e estão sendo determinantes para o sucesso desta evolução, em todos os campos em que possamos imaginar.

De acordo com Correia (2004, p. 9), “As redes de computadores têm crescido assustadoramente nos últimos anos, hoje empresas de qualquer porte necessitam de uma rede de computadores para seu bom funcionamento, maximização de margens e de produtividade”. Esta afirmação reforça a importância da evolução desta área para o mundo moderno, porém, existe uma solução universal?

Segundo Petterson e Davie (2007, p. 2), provavelmente a mais importante característica das redes de computadores é a generalidade, por isso normalmente elas são idealizadas para o funcionamento transparente. Na prática, com a especialização dos processos, cada sistema busca a adequação ao ambiente e suas necessidades, mas sempre devemos procurar e/ou manter a conectividade geral, prevendo que futuras aplicações possam ser implementadas sem alteração significativa da configuração ou arquitetura da rede.

Não há como negar a forte tendência do mercado em relação à eliminação de conexões cabeadas. Empresas e usuários buscam cada vez mais desvencilharem-se das tramas de cabos metálicos ou de fibras ópticas, que encarecem e dificultam a instalação e a mobilidade.

Estamos assistindo ao surgimento de pessoas totalmente viciadas em informações: pessoas que precisam estar permanentemente on-line. Para esses usuários móveis o par trançado, o cabo coaxial e a fibra óptica não têm a menor utilidade. Eles precisam transferir dados para seus computadores laptop, notebook, palmtop de bolso ou de pulso sem depender da infraestrutura de comunicação terrestre. A resposta para esses usuários está na comunicação sem fios. (TANNENBAUM, 2003, p.89).

Embora essa tendência pareça apenas um modismo, em diversos casos a solução do uso de redes sem fio pode ser o único meio possível ou desejável, como ele próprio complementa:

No entanto existem algumas outras circunstâncias em que a comunicação sem fios apresenta vantagens até mesmo para dispositivos fixos. Por exemplo, quando há dificuldades para instalar cabos de fibra óptica em um prédio devido a acidentes geográficos (montanhas, florestas, pântanos, etc.) deve-se recorrer à tecnologia da transmissão sem fios. (TANNENBAUM, 2003, p.89).

A partir destas observações e afirmações dos autores acima citados, torna-se muito mais claro o entendimento de como deve funcionar qualquer rede disposta em um ambiente restrito, como de um aeroporto, que é exatamente o foco deste projeto: os dispositivos devem

se comunicar por uma rede padronizada, a qual tenha os parâmetros já definidos em normas internacionais, atuando de maneira organizada e preferencialmente sem o uso de cabeamento por se tratar de um ambiente de alto risco. Dentro destas premissas de trabalho, o estudo está contemplando o uso de dispositivos de comunicação sem fio que operem em frequências homologadas pelos órgãos reguladores de telecomunicações dentro do espectro radioelétrico disponível e protocolos de comunicação certificados, de maneira a não interferir no funcionamento dos demais equipamentos do aeroporto e inclusive das aeronaves.

4.1. Aeroporto Internacional Santos Dumont – um ambiente complexo

Não podemos de forma alguma relevar a importância dos aeroportos em relação ao desenvolvimento do mundo moderno como um todo. Muito mais do que pontos de convergência de rotas de cargas e passageiros, servem como terminais aéreos de permanente troca e em grande maioria, com funcionamento ininterrupto, sendo hoje muitos localizados em áreas de alta densidade demográfica, para melhor entendimento, a seguir temos um breve histórico mesclado dos sites da Infraero e Pablaerobrasil.

O Rio de Janeiro, sendo a segunda capital do Brasil desde 1763, já operava nas décadas de 30 e 40 o seu comércio aéreo por meio de aviões hidroplanos, através do atracadouro da Ponta do Calabouço, porém o pouso e a decolagem terrestre eram feitos no Campo de Mangueiros por civis, e a Aeronáutica ocupava o Campo dos Afonsos e do Galeão. Com o constante crescimento da cidade, na condição de Distrito Federal, exigia um aeroporto que comportasse o volume de cargas condizente com a situação, assim, foi aceita a proposta de fazer uso do Aterro do Calabouço, a qual foi aceita. Assim, as obras iniciaram em 1934 com a ampliação do aterro em cerca de 370 mil metros quadrados. Em 1936, apesar de já estar operando desde o início com limitações, foi inaugurado o primeiro aeroporto civil do país.

Iniciando com 400 metros em 1934, estendida para 700 metros em 1936, 1050 metros em 1938 e finalmente 1350 metros em 1947, passou a comportar cada vez mais aeronaves, também maiores, manteve-se até a atualidade com um volume considerável de tráfego, mesmo com a mudança da capital federal para Brasília, estimando-se que transporte mais de oito milhões de passageiros/ano.

O aeroporto já teve diversos contratempos, inclusive um grande incêndio em 1998. O prédio destruído foi recomposto e o passou a operar em menos de 180 dias. Apesar de todo o controle, percebe-se claramente que administrar um complexo ativo deste porte não é uma

tarefa trivial e por isso cada elemento funcional tem que ser cuidadosamente normatizado. Hoje o percentual de falhas é muito pequeno, mas ainda acontecem, o foco deste projeto se deve especialmente na ocorrência de dois acidentes ocorridos na Avenida Almirante Silvio de Noronha, uma estrada que passa atrás da cabeceira da pista principal, a qual dá acesso à Escola Naval, na ilha de Villegagnon.

Mais precisamente em 29 de janeiro de 2002, um taxi foi lançado contra as pedras da Baía de Guanabara pela força da turbina de um avião, em processo de decolagem, o motorista do veículo ficou gravemente ferido e acabou falecendo três dias depois, em virtude do acidente (ESTADÃO, 2002). Quase um mês depois, em 21 de fevereiro, duas escritoras também foram lançadas com seu carro da mesma maneira, porém desta vez apenas com ferimentos, e se recuperaram (Figura 01); (KIRSTEN, 2002).



Figura 01 - Acidente fatal no Aeroporto S. Dumont. 2002.

Fontes: Souza, Jr. (2007); Kirsten, R. I. (2008)

O tráfego local é ainda hoje limitado nos intervalos de decolagem por acionamento manual de duas cancelas por um vigilante denominado “Cochileiro”. Como o vigilante falhou mais de uma vez, por motivos diversos, em sua função repetitiva, logo algo precisava ser feito, afinal, não é possível que vidas se percam por estes erros humanos.

A partir das investigações, ficou evidente que se faz necessária a implementação de um sistema automatizado, que retire do ser humano a função de realizar um processo repetitivo e manual de acionamento, o qual pode ser feito através da moderna tecnologia existente com mais segurança e rapidez. Este sistema não eliminará a necessidade de uma supervisão de uma pessoa, porém a sua função será a de resolver as eventuais falhas de sistema como alarmes falsos e outros decorrentes de fatores adversos, bem como o controle pessoal das pessoas que insistem em trafegar a pé pela pista. Uma prova disso é a recente notícia de que em março deste ano: uma

família que transitava a pé foi arremessada longe pelas turbinas de uma aeronave, e sofreram escoriações leves. O fato foi divulgado pelos meios de comunicações (REDE RECORD, 2012).

Mais uma vez fica evidenciada a urgência de mais controle, e claramente provado que a presença humana é essencial, visto que os pedestres não respeitaram a sinalização. O sistema automático permitirá no futuro inclusive que seja comprovada origem do erro, permitindo assim que a estrutura seja sempre melhorada.

4.2. Exigências do solicitante

Obviamente um projeto implementado dentro de um aeroporto tem as suas limitações. Desta forma, fizeram-se obrigatórias algumas exigências, coerentes com o ambiente de atuação:

- O sistema de sensoriamento deve alterar o mínimo possível o ambiente local (não deve criar obstáculos na pista);
- O sistema de sensoriamento deve ser robusto, mas o suporte frangível;
- Os sensores devem ser resistentes a condições climáticas rigorosas, devem suportar o deslocamento de ar das aeronaves, ser a prova de chuva e resistentes à maresia;
- Caso opte-se por uma solução sem fios, deve-se trabalhar em uma faixa de frequência aberta ISM (sem a necessidade de licença) e não coincidente com os equipamentos de voo.

5. PROPOSTA DE SOLUÇÃO PARA O PROBLEMA

Mesmo antes de delinear os equipamentos, foi necessária uma avaliação do problema em questões como a distribuição física de espaço e localização dos pontos a serem monitorados. Através das informações repassadas pelo solicitante e especialmente com a ajuda do mapeamento via satélite, foi possível ter uma noção bem aproximada do campo de trabalho, como mostra a figura 02.

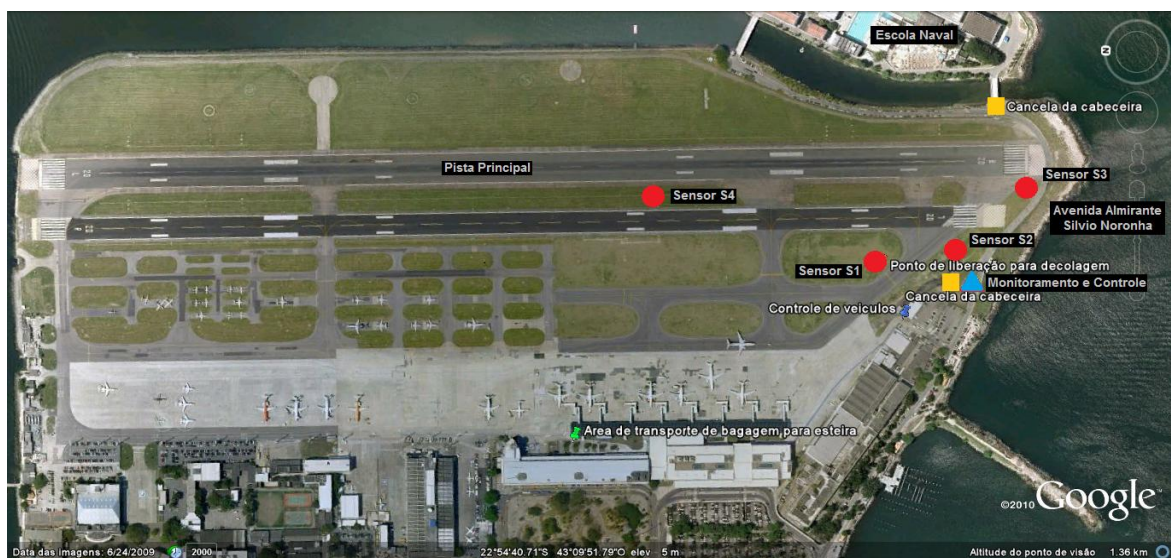


Figura 02 - Aeroporto Santos Dumont – imagem de satélite editada
Fonte: Google Maps (2011)

As cancelas de controle de tráfego da avenida, que operam em controle manual atualmente, podem ser melhor visualizadas na figura 03:



Figura 03 - Cancela de controle de veículos
Fonte: I-Dutto (2011)

Independentemente do tipo de sensoriamento a ser utilizado, foi definida a condição de trabalho em operação normal, e elaborada a distribuição nos pontos de relevância. O ponto de marcação fundamental é o de liberação para decolagem. Por este motivo foram previstos quatro sensores, sendo dois trabalhando em regime de redundância para se reduzir as chances de ocorrerem falhas de monitoramento. Quando a aeronave chega a este local para receber a liberação de voo, a cancela tem que ser fechada com prioridade máxima.

A condição essencial do projeto é prover o alerta para que seja feito o fechamento da cancela de controle de veículos da avenida quando a aeronave está em processo de decolagem, lançando a força das suas turbinas contra a via podendo causar acidentes, quando existe uma condição anormal como inversão da pista ou disparos eventuais dos sensores, ou ainda quando o fluxo é de aterrissagem e, portanto não promove o efeito de deslocamento de ar contra a avenida, não é necessária a interrupção do fluxo de veículos, evitando o congestionamento que poderia ocorrer pela alta rotatividade da Escola Naval. Para que um software de controle automático seja elaborado futuramente, foi de fundamental importância desenvolver um fluxograma com todas as condições possíveis de serem presenciadas para que a inteligência artificial atribuída ao sistema possa garantir a segurança operacional, desta forma foi elaborado o mapa de fluxo conforme solicitação e especificações do solicitante (Fluxograma 1).

A elaboração depende de condições estáticas, como um sensor acionado com a passagem do alvo, mas também necessita de um monitoramento de estado, ou seja, o programa necessita saber que o sensor x foi acionado, mas também vai usar o histórico dos outros sensores para determinar se é uma condição de decolagem, aterrissagem ou outros, como por exemplo, um disparo falso. Cabe salientar que disparos falsos nem sempre são oriundos de erros de sensores, pois o local por vezes pode sofrer influência de máquinas de manutenção, pessoas e animais, como pode ser visto na figura 04. As variáveis de estado podem ser chamadas de *flags* (bandeiras), que nada mais são do que pontos de marcação temporária de passagem pelo sensor.



Figura 04 - Causas de disparos falsos
Fonte: I-Dutto (2011)

Os sensores devem ter a precisão de detectar as mudanças de estado com a presença do objeto, mas também necessitam de uma calibragem tal que não disparem, por exemplo, com a chuva ou irradiação solar extrema. Para suprir a necessidade de todos estes estados, o fluxograma (Fluxograma 1) foi elaborado tomando como base as prioridades:

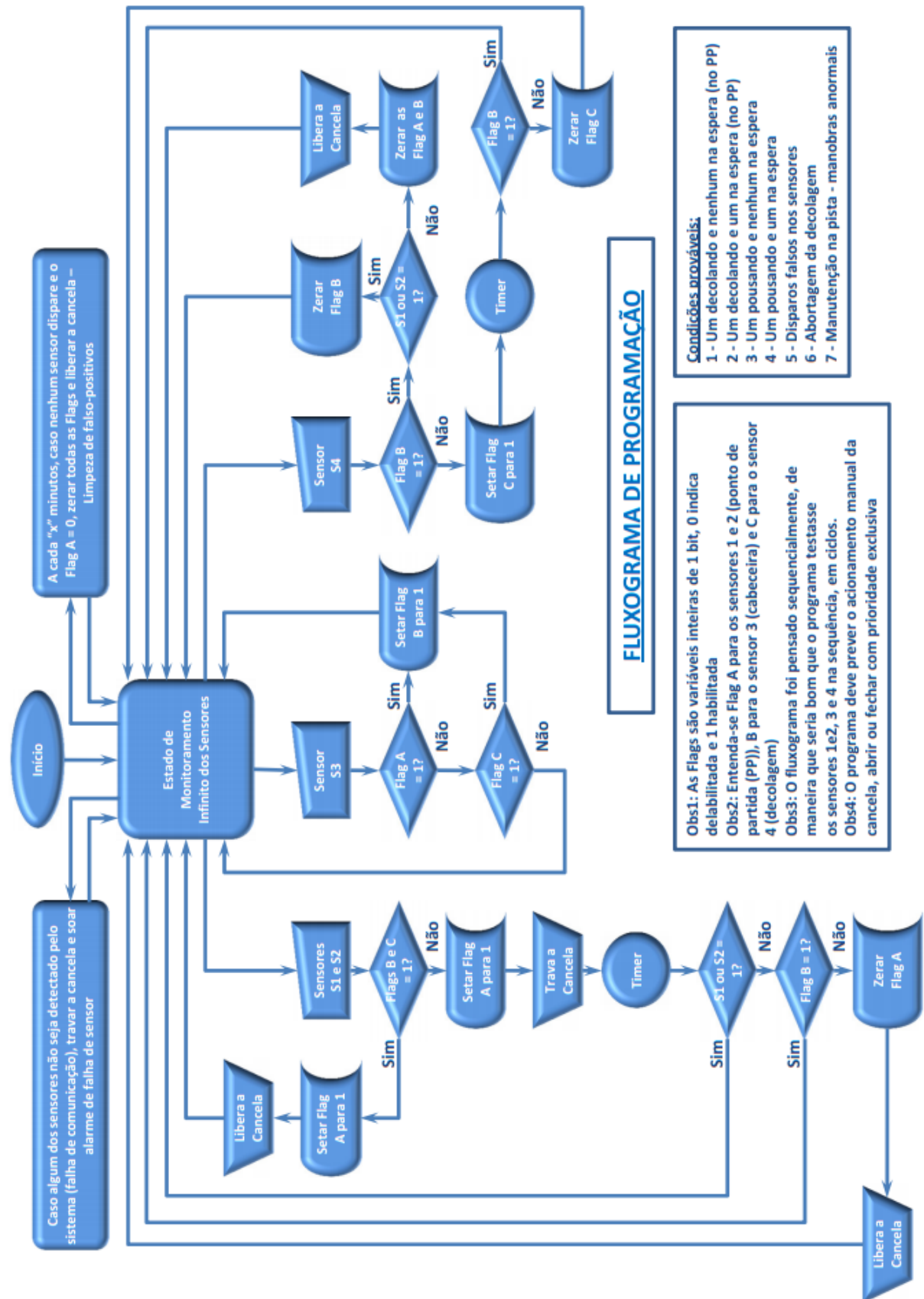
- Sensores 1 e 2: funcionam de forma redundante, garantindo mais segurança na detecção do objeto, no caso a aeronave, na preparação para a decolagem, na condição de presença constante neste local a cancela permanece fechada. A estes dois sensores está associada à *flag* de estado A.

- Sensor 3: faz a detecção da presença do objeto na cabeceira da pista em qualquer sentido e serve primordialmente para a constatação do sentido do trajeto percorrido pela aeronave no monitoramento. A este sensor está associada à *flag* de estado B.

- Sensor 4: percebe a presença do objeto na passagem para decolagem ou pouso, atuando com as demais *flags* e sensores. A este sensor está associada à *flag* C.

Como pode ser percebido, o sistema depende diretamente das bandeiras de estado e dos sensores, para determinar a função do acionador automático da barreira e todas as possibilidades pretendem serem previstas. Porém, como em todo sistema, podem ocorrer condições anormais que necessitarão da presença humana, como por exemplo, pessoas andando na avenida, conforme acidente ocorrido recentemente (Rede Record, 2012). Em testes de mesa, nos quais os percursos de fluxo foram simulados com os equipamentos em funcionamento e rede estabelecida, o sistema mostrou-se eficaz, porém cabe ao programador dar forma ao código, de maneira que possa ser implementado definitivamente. Esta pode ser uma proposta futura de pesquisa, a qual não pôde ser desenvolvida no curto período deste projeto e não era o objetivo final, lembrando que a proposta foi criar um sistema de monitoramento em rede *wireless* e uma sequência de fluxo de trabalho que permita ao solicitante implementar de forma definitiva o seu software de controle com base nas diretrizes criadas.

Fluxograma 1: Fluxo do programa de monitoramento da movimentação de aeronaves



Fonte: Autor

6. ANÁLISE DOS PROVÁVEIS DISPOSITIVOS A SEREM UTILIZADOS

Foram delineadas diversas alternativas, dentro das tecnologias existentes:

- **Controle por sensores GPS** (*Global Positioning System* ou Sistema de Posicionamento Global):

Em um sistema deste tipo, um receptor faz a detecção e identificação de alguns satélites ao seu alcance, e consegue por meio desta informação realizar a triangulação exata de sua localização terrestre, dentro da sua limitação de precisão. Segundo Marshall, “O Sistema de Posicionamento Global (GPS) é uma verdadeira constelação de 27 satélites em órbita ao redor da Terra (24 em operação e 3 extras caso haja falha nos outros)”.

Apesar de ser uma solução atual e muito precisa, existe uma grande barreira determinada pela necessidade de instalação e configuração individual de cada sensor para cada aeronave, de maneira que poderia não haver sensoramento caso algum avião não implementado desse entrada na pista, podendo causar acidentes. Sabemos que a maioria das aeronaves conta hoje com um ou mais dispositivos que utilizam esta tecnologia, mas ainda não é possível garantir a totalidade. Outro problema decorrente é o fato de determinar que 100% dos receptores de GPS a bordo estejam em funcionamento e sejam conectados com o sistema de controle, o que inicialmente é um problema a resolver a médio ou longo prazo. Desta forma este meio de sensoramento foi desconsiderado no projeto.

- **Controle por sensores RFID** (Identificação por Radiofrequência):

A tecnologia RFID é na atualidade um diferencial em identificação e tende a ser o padrão que irá substituir inclusive o tradicional código de barras. Segundo Santana (WirelessBR.org), “Esta nova tecnologia prevê uma mudança radical na operação do varejo mundial”.

Consiste basicamente no uso de percepção de campo elétrico (distante) ou magnético (próximo) para realizar a detecção de um dispositivo previamente sintonizado e conhecido, possibilitando o uso da informação para controle (figura 05).

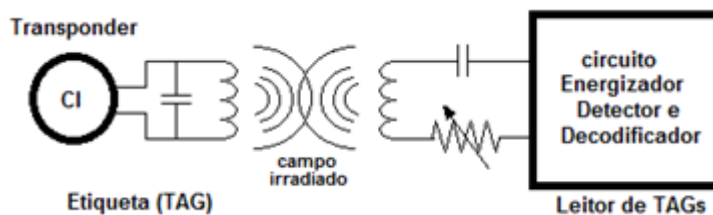


Figura 05 – Princípio básico do funcionamento RFID

Fonte: Autor

A etiqueta sintonizada pode ser passiva, e neste caso aproveita a energia do campo gerado pelo leitor de RFID e consegue, através de um circuito ressonante simples sintonizado na frequência certa, transmitir de volta a informação requerida, que pode ser apenas a indicação de presença até alguns dados. Quando se utiliza um sistema com etiquetas ativas, estas fazem o uso de alimentação por fonte própria para transmitir o sinal requerido pelo leitor, quando solicitado. Esta última seria o objeto de interesse para o projeto. Assim como o sistema por GPS, as *tags* teriam que ser instaladas em cada aeronave, o que, apesar de ter um custo reduzido, compromete a segurança, visto que nem sempre é possível garantir a presença das etiquetas funcionais em todos os aviões que passam pelo aeroporto, assim este tipo de tecnologia não foi escolhido.

- Controle por sensores de Ultra-Som:

Atuam pelo princípio do efeito *Doppler* (Lordello). Segundo *Silva*, da Equipe Brasil Escola, “O efeito *Doppler* é a alteração da frequência sonora percebida pelo observador em virtude do movimento relativo de aproximação ou afastamento entre a fonte e o observador.” O efeito foi descrito teoricamente pela primeira vez em 1842 por *Johann Christian Andreas Doppler*, recebendo posteriormente seu sobrenome, em homenagem.

Um exemplo clássico deste efeito é a sirene da ambulância, o observador percebe um toque mais agudo quando se aproxima e mais grave quando se afasta, isso porque a frequência relativa para o observador é a composição da diferença da frequência da fonte e a velocidade com que se desloca em relação a essa fonte, ou seja, o veículo, por exemplo, ao deslocar-se em direção ao observador ou afastando-se dele, cada onda sonora emitida estará mais próxima ou mais distante da antecessora em relação ao ponto de observação, alterando o seu comprimento de onda. Dado que o comprimento de onda é inversamente proporcional à frequência, a mesma é alterada proporcionalmente. Segundo *Silva*, podemos definir uma equação geral que define a frequência observada no efeito *Doppler*:

$$F_o = F_f \left(\frac{V \pm V_o}{V \pm V_f} \right)$$

Onde F_o é frequência percebida pelo observador

F_f é a frequência real da fonte

V é a velocidade da onda

V_o é a velocidade do observador, positiva se estiver se aproximando da fonte ou negativa se estiver se afastando.

V_f é a velocidade da fonte, positiva ao se afastar ou negativa ao se aproximar do observador.

Os sensores de Ultra-Som utilizam este efeito, emitindo sinais acústicos com frequência entre 22 kHz e 45 kHz, de forma que fazem a comparação do eco do sinal recebido com o transmitido e assim podem-se perceber mudanças no ambiente. Estes sensores são muito sensíveis a variações e não permitem um monitoramento confiável em ambientes externos, com variação de correntes de ar e também de umidade, especialmente. Por este motivo esse tipo de tecnologia não foi escolhido para este projeto.

- Controle por sensores por Micro-ondas:

Atuam também pelo princípio do efeito *Doppler*, porém utilizando-se de frequências relativamente maiores, na faixa de GHz. Funciona basicamente da mesma forma que o radar ou sensor de Ultra-Som, ou seja, emite um sinal em micro-ondas e “observa” o eco refletido, caso ocorra alguma alteração, é possível detectar com precisão e calcular o deslocamento ocorrido, velocidade, distância, etc. Os radares de Ultra-Som e Micro-ondas são dispositivos ativos no sistema, pois injetam um sinal (luz, micro-ondas ou som) para obter o retorno modificado pelo ambiente. Este tipo de sensor apresenta um elevado grau de precisão e mais robustez em relação a variações climáticas, por isso foi cotado como uma tecnologia a ser incluída no projeto.

- Controle por sensores Infravermelhos:

Estes dispositivos atuam pela detecção da energia infravermelha presente no ambiente, seja de pessoas, animais, plantas ou quaisquer objetos.

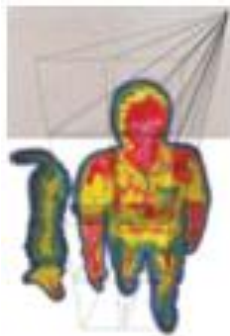


Figura 06 - Sensoriamento Infravermelho
Fonte: Carvalho, M. (2010)

A Samtek define que os sensores passivos são assim considerados por gerarem nem emitirem nenhum tipo de radiação eletromagnética, ou seja, apenas captam as radiações do ambiente e dos objetos no seu campo de visão. Operam em uma faixa de frequência situada entre as micro-ondas e a luz visível, mais especificamente acima da luz vermelha, conforme a tabela 01. Ainda segundo a Samtek, “A detecção do movimento ocorre quando uma fonte infravermelha com uma temperatura própria passa em frente de uma outra fonte com uma temperatura diferente, tal como uma parede”.

Este tipo de sensor sofre um forte problema quando a temperatura ambiente se aproxima da temperatura do objeto a ser detectado, por isso foi desenvolvido o sistema ATC (*Automatic Temperature Control*), que aumenta a sensibilidade do sensor à medida que se a diferença entre as temperaturas torna-se muito próxima, de acordo com o gráfico da figura 07.

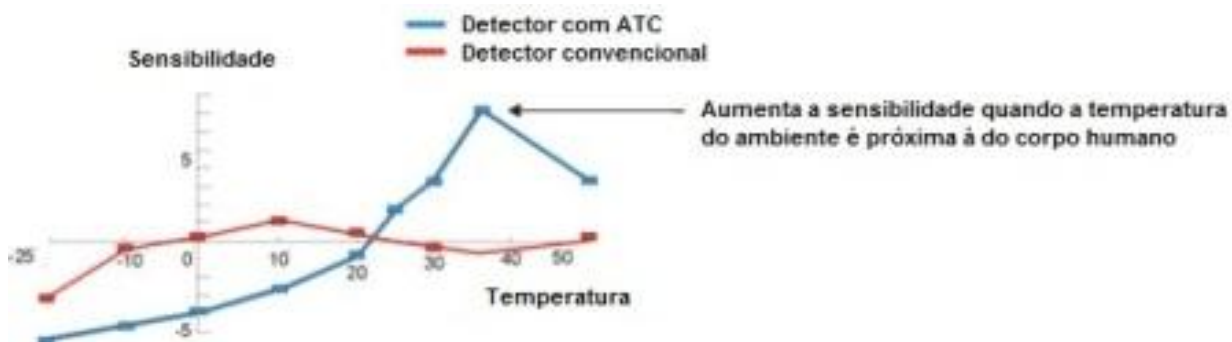


Figura 07 - Gráfico do sensor infravermelho com e sem ATC
Fonte: Samtek (2012)

Tabela 01 - Espectro Eletromagnético

Designação		Frequência (Hz)			Comprimento de onda		
Ondas de rádio		Inferior	a	3×10^{11}	Superior	a	0,3 m
Microondas		10^9 Hz	a	3×10^{11}	0,3 m	a	1 mm
Infravermelho		3×10^{11} Hz	a	$3,8 \times 10^{14}$	1 mm	a	789 nm
	Vermelho	$3,8 \times 10^{14}$ Hz	a	$4,8 \times 10^{14}$	789 nm	a	625 nm
	Laranja	$4,8 \times 10^{14}$ Hz	a	5×10^{14}	625 nm	a	600 nm
Visível	Amarelo	5×10^{14} Hz	a	$5,2 \times 10^{14}$	600 nm	a	577 nm
	Verde	$5,2 \times 10^{14}$ Hz	a	$6,1 \times 10^{14}$	577 nm	a	491 nm
	Azul	$6,1 \times 10^{14}$ Hz	a	$6,59 \times 10^{14}$	491 nm	a	455 nm
	Violeta	$6,59 \times 10^{14}$ Hz	a	8×10^{14}	455 nm	a	390 nm
Ultravioleta		8×10^{14} Hz	a	$2,4 \times 10^{16}$	390 nm	a	8,82 nm
Raios X		$2,4 \times 10^{16}$ Hz	a	5×10^{19}	8,82 nm	a	6 pm
Raios gama		Superior	a	5×10^{19}	Inferior	a	6 pm

Fonte: Lima, B.; Bakker, J. (2011)

Os sistemas de alarme modernos utilizam maciçamente este tipo de sensor com um grau de confiabilidade relativamente grande, por este motivo este tipo também foi cotado para ser possivelmente utilizado neste projeto.

7. DETERMINAÇÃO DO MEIO DE TRANSMISSÃO ENTRE OS SENSORES E O GATEWAY

Como foi descrito anteriormente, a informação básica do solicitante é de que o ambiente deveria ser alterado o mínimo possível, e desta forma, nenhum meio de transporte aparente poderia ser adicionado ao solo. De acordo com essa premissa, fez-se necessária a análise em duas formas distintas de transporte de dados:

7.1. Meio guiado:

Deve prover uma instalação de uma rede cabeada dentro de um ambiente como uma pista de pouso e decolagem como a do Aeroporto Santos Dumont não é uma tarefa simples, pois o fluxo de pista, como já foi dito, é muito intenso. Realizar a implantação de um sistema de cabeamento estruturado requer a instalação de dutos e caixas de passagem adequadas, com proteção extra contra as intempéries como alagamentos e salinidade, além de reforço mecânico devido ao peso exercido na superfície pelo deslocamento das aeronaves.

Somados a esses fatores, o tempo de instalação dos dutos requer um período relativamente grande para cortes e reparos na pista, aliados ao elevado custo, Estes fatores indicaram que a utilização de um meio de transmissão guiado é uma opção pouco viável para o projeto e por esse motivo não houve interesse na sua adoção.

7.2. Meio não guiado:

Como os sensores especificados para o projeto baseiam-se apenas no meio de detecção local, fez-se necessária a integração de um meio de transporte de dados até a base de recepção. Este meio, como foi definido, deve respeitar limites de licenciamento de canais e frequências, operar em conjunto com os sensores e ter uma velocidade compatível que permita o monitoramento completo sem perda de dados. Após análise foi escolhido o meio baseado na tecnologia *ZigBee*, como será descrito a seguir.

8. A TECNOLOGIA ZIGBEE

A expansão das redes sem fio tem trazido ao mundo uma percepção cada vez mais apurada da importância da utilização de dispositivos interconectados sem a necessidade de dispendiosas estruturas cabeadas, que possam facilitar o monitoramento dos mais diversos sensores e equipamentos, proporcionando um serviço de uso simplificado e modular. Estes dispositivos, por sua vez, devem ser suficientemente confiáveis e de custo reduzido, para que possam ser incorporados aos sistemas de forma a melhorar o rendimento sem comprometer o orçamento.

Seguindo a evolução da tecnologia, os grandes fabricantes de equipamentos de redes passaram a travar uma disputa acirrada por desempenho. A taxa de transmissão foi sendo ampliada gradualmente, com o auxílio do desenvolvimento acelerado de hardware e software, especialmente nas duas últimas décadas. A indústria, porém, especialmente no tratamento do meio fabril, viu-se em um paradigma: o desempenho já está além do necessário, mas é possível reduzir o custo? Essa reflexão levou à conclusão de que para boa parte dos equipamentos a serem monitorados, a taxa de transmissão era um fator irrelevante, mas o consumo não, especialmente com a utilização de dispositivos móveis e portáteis.

Em 2002, algumas empresas, (*Honeywell, Invensys, Philips e a Mitsubishi Eletric*), segundo Campos (2006, p. 2) uniram seus esforços para criar um consórcio, a *ZigBee Alliance*, com o intuito de desenvolver um padrão que atendesse a diversos requisitos, entre eles:

- Confiabilidade na entrega dos dados;
- Baixo consumo de energia;
- Baixo custo de produção;
- Baixa irradiação de espúrios no meio;
- Padrão global aberto.

Atualmente o consórcio conta com mais de quarenta e cinco empresas, dada a relevância e sucesso obtido no projeto, com uma larga faixa de utilização dos equipamentos. O resultado da compilação de vários padrões já existentes na evolução dos meios de comunicação sem fio acabou por tornar-se uma versão atualizada da tecnologia *HomeRF Lite*, de acordo com Gouveia (2009, p. 12), sendo batizada de “*ZigBee*”. O nome derivou da alusão ao voo das abelhas em zig-zag, estratégia que permite a elas informar aos demais membros da equipe a localização aproximada da fonte de alimento.

Em 2003, a IEEE (*Institute of Electrical and Electronics Engineers*) estabeleceu o novo padrão 802.15.4, o qual serve como base para as camadas física (PHY) e de enlace

(MAC). A tecnologia *ZigBee* é dotada ainda das camadas de rede e de aplicação (subdividida), como será tratado a seguir, o que faz dela uma categoria similar, mas com propósitos distintos de WPAN (*Wireless Personal Area Network*). Para Vasques, as redes padrão 802.11 e 802.15 têm objetivos semelhantes entre si, como a comunicação sem fio usando frequências ISM (livre de licença) e com curto alcance, porém o padrão 802.15.4 além destes objetivos diferencia-se por proporcionar a comunicação com consumo reduzido (Santos, 2008, p. 10) utilizando baixa potência de transmissão, não levando em conta a necessidade de taxas elevadas de transporte de dados.

8.1. Padrão de frequências adotado no mundo:

Via de regra, em virtude das características intrínsecas dos sistemas de comunicação que utilizam o espectro de radiofrequências, os serviços de comunicação devem ser regulados, de forma a garantir o uso desse recurso escasso de maneira apropriada por todos os interessados. Entretanto, com o intuito de evitar sobrecarga de solicitações de licença nos órgãos reguladores, bem como para simplificar a utilização de RF por aplicações específicas, com baixas potências, criou-se a forma de uso não licenciado do espectro (TELECO, 2006).

A figura 08 apresenta um dado interessante, os padrões de frequência, apesar de buscarem a globalização, apresentam diferenças na Europa e América (mais precisamente nos Estados Unidos). Isso se deve ao fato de operarem nas faixas de frequência ISM (*Industry, Scientific and Medic*) que são diferentes nestas regiões. Os demais países operam a cerca de 2,4 GHz, igualmente sem a necessidade de licenças, o que torna mais fácil a aplicação em qualquer local.

Banda de Frequência	Licença requerida	Região Geográfica	Taxa de Dados	Número dos Canais
868,3 MHz	Não	Europa	20 kbps	0
902-928 MHz	Não	Américas	40 kbps	1-10
2405-2480 MHz	Não	Global	250 kbps	11-26

Figura 08 - Frequências utilizadas pelo padrão *ZigBee*

Fonte: Vasques, B.L.R.P; Coutinho, I.B.A; Lima, M.F.; Carneval, V.P.O.

8.2. Distribuição das camadas do protocolo:

O padrão estabelecido final é uma mescla da definição IEEE 802.15.4 e dos padrões definidos pela *ZigBee Alliance* como pode ser visto no quadro da figura 09.

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

Figura 09 - Pilha de Protocolo *ZigBee*

Fonte: Vasques, B.L.R.P; Coutinho, I.B.A; Lima, M.F.; Carneval, V.P.O.

A partir da visão destas camadas, baseando-nos no Modelo de Referência OSI, podemos entender que as camadas superiores são as principais responsáveis pela entrega dos dados no destino, deixando para as inferiores a função do acesso ao meio e transporte dos dados no ambiente físico.

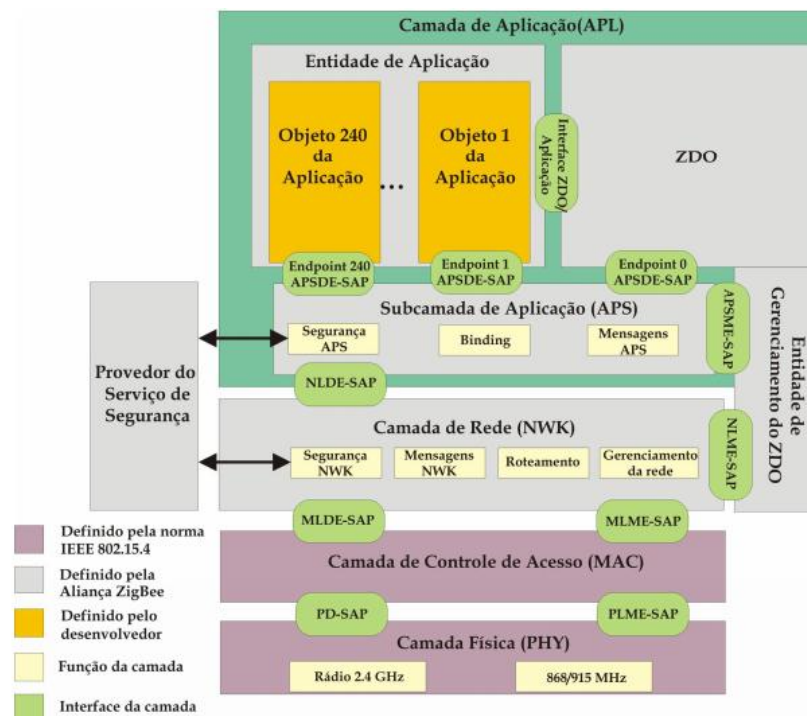


Figura 10 - Pilha de Protocolo *ZigBee*

Fonte: Vasques, B.L.R.P; Coutinho, I.B.A; Lima, M.F.; Carneval, V.P.O.

A figura 10 detalha bem a distribuição da pilha de protocolo *ZigBee*, onde as camadas são interligadas por interfaces específicas entre cada uma. As camadas física e de enlace, regulamentadas pelo padrão IEEE 802.15.4, encarregam-se pela codificação, suporte (controle, envio, recepção), acesso ao meio e alguma função de segurança, como será visto mais adiante. Já a camada de rede trata do gerenciamento, roteamento e segurança da rede, através de troca de mensagens entre os dispositivos. A camada de suporte a aplicação é onde o usuário recebe a base para atuar com a programação dos objetos de aplicação; a *ZigBee Alliance* estruturou um modelo mais complexo em relação a estas duas camadas e a camada de aplicação: não está bem definida a linha divisória entre o suporte e a aplicação, de forma o usuário modela seus aplicativos utilizando objetos do tipo ZDO (*ZigBee Device Object*). Segundo Garcia (2006), os objetos são determinados para aderirem a perfis público ou privado, de maneira que os perfis, ou *profiles*, determinam o ambiente da aplicação, o tipo de dispositivo e o *cluster* usado para se comunicar, sendo que os de perfil público garantem a operabilidade entre diferentes fabricantes. Um *cluster* é definido pelo Informeteczb “por um identificador de *cluster* (*Cluster ID*), este *cluster* se associa ao dispositivo que produz os fluxos de dados. Os indicadores de clusters são únicos dentro de um mesmo perfil”. As camadas de rede e de suporte a aplicação são munidas de um módulo de segurança denominado Provedor de Serviço de Segurança, o qual permite gerenciar acessos a conteúdos encriptados, desde que a função esteja habilitada. Este mecanismo é acionado através do ZDO e não constitui a principal segurança, sendo esta uma atribuição nativa dos perfis definidos em cada rede.

8.2.1. Norma 802.15.4: Definição das camadas

8.2.1.1. Camada Física:

Tem por finalidade a transmissão dos PDUs (*Protocol Data Units*) através de onda de rádio segundo Vasques (2010), além desta função, ainda é responsável por indicar a qualidade de transmissão, detectar a potência dos canais e reportar canais livres. Utilizando a modulação DSSS (*Direct Sequence Spread Spectrum*), cada bit é dotado de um padrão de redundância e colocado ao longo da banda do canal, de forma que permite uma aquisição mais confiável com detecção de erros, tornando o sistema mais robusto. A figura 11 ilustra a distribuição de frequências da camada física nas três regiões existentes:

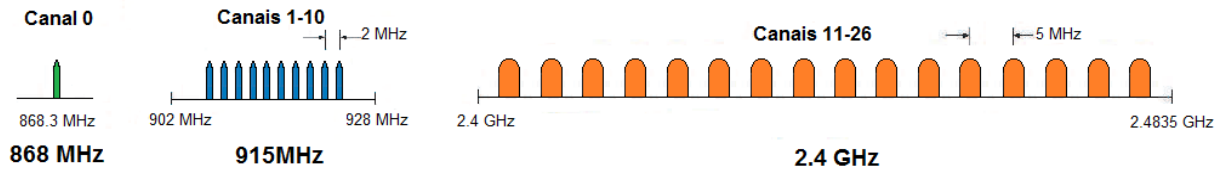


Figura 11 - Frequências utilizadas pela camada física *ZigBee*

Fonte: Adaptada de Vasques, B.L.R.P; Coutinho, I.B.A; Lima, M.F.; Carneval, V.P.O.

O espectro de frequências, assim como foi descrito, obedece a limitações por regiões e também associa diferentes características, sendo a mais utilizada banda relacionada a 2,4 GHz, que conseguiu a maior taxa de transferência, chegando a 250 kbps através da modulação O-QPSK (*Offset quadrature phase-shift keying*), conforme o quadro da figura 12.

PHY (MHz)	Banda de frequência (MHz)	Parâmetros de espalhamento		Parâmetros dos dados		
		Taxa de espalhamento (kchip/s)	Modulação	Taxa de bit (kb/s)	Taxa de símbolo	Símbolos
868/915	868-868.6	300	BPSK	20	20	Binário
	902-928	600	BPSK	40	40	Binário
2450	2400-2483.5	2000	O-QPSK	250	62.5	16 símbolos

Figura 12 - Características de cada padrão adotado

Fonte: Vasques, B.L.R.P; Coutinho, I.B.A; Lima, M.F.; Carneval, V.P.O.

Em relação aos quadros enviados, a figura 13 mostra a distribuição interna dos mesmos, destacando que basicamente um PDU é formado de um bloco de sincronismo SHR e um bloco de informação PHR, acrescido de um *payload* que representa a PDU vinda da camada de enlace.

Bytes	4	1	1	n
Camada Física	Sequência de Preâmbulo	Delimitador de início de quadro	Comprimento do quadro	MAC PDU
	SHR		PHR	
	6+n			
	PHY PDU			

Figura 13 - Formato do quadro PDU

Fonte: Vasques, B.L.R.P; Coutinho, I.B.A; Lima, M.F.; Carneval, V.P.O.

8.2.1.2. Camada de Enlace:

A camada de enlace desempenha um papel fundamental na arquitetura do *Zigbee*, realizando a função de empacotar os dados vindos das camadas superiores de forma a serem transmitidos pela camada física. É neste ponto que o desenvolvimento obteve uma das maiores vantagens: o baixo consumo, segundo Florido (2008, p. 20). Esta funcionalidade pode ser melhor entendida através do conhecimento dos modos de operação. A figura 14 expressa de forma geral o formato de um quadro da camada de enlace. Teixeira (2006, p. 47) descreve as vantagens do uso desta tecnologia utilizando os modos de operação *beaconing* e *non-beaconing*.

Octetos:2	1	0/2	0/2/8	0/2	0/2/8	variable	2
Frame de controle	Número sequência	Identificador de destino PAN	Endereço de destino	Identificador de origem PAN	Endereço de origem	Quadro de payload	FCS
		Campos de endereçamento					
MHR						MAC payload	MFR

Figura 14 - Formato geral do quadro da camada de enlace

Fonte: Vasques, B.L.R.P; Coutinho, I.B.A; Lima, M.F.; Carneval, V.P.O.

8.2.1.2.1. Modos de operação: Beaconing e Non-Beaconing

8.2.1.2.1.1. Modo Beaconing:

Nesta configuração, especialmente interessante em dispositivos finais, o trabalho com *ZigBee*'s mostra-se mais vantajoso. Através do modo *Beaconing*, o roteador, (ou roteadores caso exista), transmite regularmente um sinal na rede para, entre outras funções, avisar os dispositivos a ele agregados que está presente. Os receptores por sua vez, só precisam ajustar o seu ciclo e manterem-se ativos apenas no intervalo que o roteador transmite seus *beacons frames*, que são sinais que tem por finalidade manter o sincronismo entre os dispositivos, no restante do tempo permanecem em modo *sleep*, ou seja, “dormindo”, sendo que o único que não pode participar deste modo é o coordenador, mas é dele o controle. Essa redução de tempo de atividade é o diferencial que permite manter um *ZigBee* em operação por um longo período, sob a alimentação de baterias. Para trabalhar sob esta condição, é utilizada uma estrutura do tipo *Superframe*, de

acordo com Colvero (2011), definida pelo padrão IEEE 802 na subdivisão LR-WPAN (*Low Rate Wireless PAN*) no Grupo *TG4*, definida pelo coordenador apenas e dotada de um tempo que pode variar entre 15 ms e 252 ms, mas sempre dividido igualmente em 16 *slots*, conforme a figura 15.

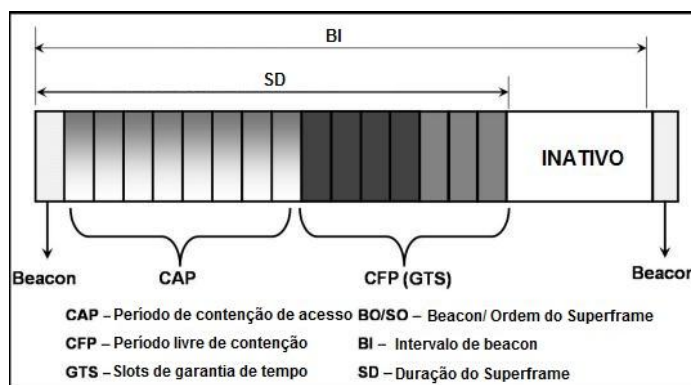


Figura 15 - Estrutura *Superframe* – atuação com *beacons*

Fonte: Vasques, B.L.R.P; Coutinho, I.B.A; Lima, M.F.; Carneval, V.P.O.

Percebe-se na Superestrutura que os *beacons* frames são muito importantes para manter o sincronismo dos dispositivos agregados, identificar a PAN e descrever a estrutura do *superframe*, de forma a perceber o tempo correto de leitura dos dados. Ainda de acordo com Vasques (2010), o meio de acesso se faz através de mecanismo CSMA-CA ALOHA (*Carrier Sense Multiple Access with Collision Avoidance*) onde os dispositivos disputam slots de tempo entre si com prevenção de colisão de dados, com um tempo de retorno aleatório exponencial decrescente. Para Kinney (2003), são definidos intervalos de tempo de contenção entre os beacons, sendo inicialmente o CAP (*Contention Access Period*), onde ocorre a disputa pelo uso do canal, e CFP (*Contention Free Period*) ou GTS (*Guarantee Time Slots*), onde é garantido um tempo para o acesso de cada dispositivo, no primeiro caso livre e no segundo caso quando o dispositivo necessita utilizar-se do recurso. Somente o coordenador pode determinar quando um GTS irá iniciar ou terminar dependendo a informação por ele recebida, sendo que podem ser alocados no máximo sete GTS's em cada PAN, resguardando o tempo CAP para outras tentativas de acesso. Após a liberação, o equipamento volta ao estado de repouso, guardando energia.

Após o frame ser recebido, o equipamento envia um frame curto de confirmação, para avisar a origem de que a mensagem foi recebida, conforme a figura 18:

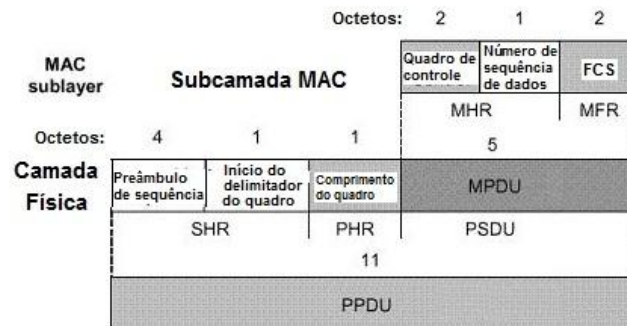


Figura 18 - Quadro de confirmação

Fonte: Vasques, B.L.R.P; Coutinho, I.B.A; Lima, M.F.; Carneval, V.P.O.

Existe também o quadro de comando MAC, que se destina a trabalhar com os endereços MAC dos *peers* de controle de tráfego, permitindo assim que o coordenador possa configurar os nós clientes, independente do tamanho da rede, vide figura 19.

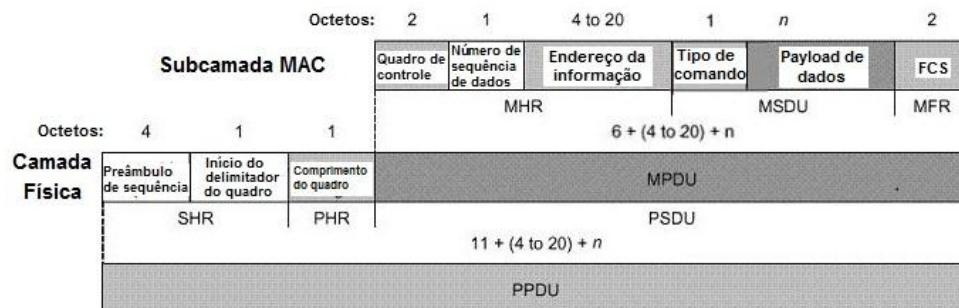


Figura 19 - Quadro de MAC de comando

Fonte: Vasques, B.L.R.P; Coutinho, I.B.A; Lima, M.F.; Carneval, V.P.O.

8.2.1.2.3. Endereçamento:

O sistema adota o padrão EUI-64, utilizando 64 bits para endereçamento, mas que pode ser reduzido para apenas 16 bits, de forma que a rede pode utilizar de seus recursos para configurar até aproximadamente 64 mil nós. Caso este número ainda não seja suficiente, um nó pode ser designado como *Gateway*, de forma a dar possibilidade de expansão da rede.

8.2.1.2.4. Segurança:

A camada de enlace permite implementar segurança por criptografia utilizando algoritmo AES (*Advanced Encryption Standard*), através da mudança de um bit no cabeçalho.

Caso seja necessário utilizar segurança, um bit do cabeçalho MAC será setado. Com isso, é anexado ao frame o Cabeçalho Auxiliar de Segurança que determina o tipo de proteção utilizado (*Security Control*), o Contador de Frames (*Frame Counter*) que garante a sequência e autenticação dos dados e guarda referência da chave (*Key Identifier*) de 128 bits a ser utilizada para determinado nó. (VASQUES, 2010).

De acordo com Kinney (2003), o receptor terá uma variedade de possíveis pacotes, de acordo com a necessidade, por exemplo, se for definida a necessidade de integridade na chegada, terá que decodificar a chave que é enviada, sendo esta uma composição do cabeçalho e o *payload* de dados que formam o MIC (*Message Integrity Code*), pode ser ainda definido que irá transportar dados confidenciais, acrescentando mais complexidade às chaves. Um detalhe a destacar é que a camada de acesso MAC só permite este recurso em um salto, de maneira que para saltos maiores deverá ser implementada a segurança nas camadas superiores.

8.2.1.3. Camada de Rede:

Andrighetto (2008, p. 43) afirma que “O coordenador da camada NWK é responsável por iniciar uma nova rede sempre que apropriado, e assinalar endereços para os novos dispositivos associados”. Esta camada atua de forma similar aos modelos OSI e TCP/IP, permitindo o endereçamento e roteamento das redes, com algumas características inerentes:

- Permite iniciar ou estabelecer uma rede;
- Permite associar ou desassociar uma determinada rede;
- Configurar um novo dispositivo;
- Sincronizar dispositivos dentro da rede;
- Prover segurança.

Além destas funcionalidades, permite adicionar sobre o quadro de enlace funções que permitam estender a rede, associar ou dividir a mesma. Na camada de rede podem ser formadas três topologias: árvore, estrela ou malha.

8.2.1.3.1. Topologias de rede:

O padrão *ZigBee*, assim como foi citado, é extremamente robusto e flexível. Os elementos envolvidos podem trabalhar sob diversas configurações de forma a permitir uma montagem praticamente livre. Conforme a Informeteczb, através da associação de dispositivos, podemos ter basicamente três principais topologias:

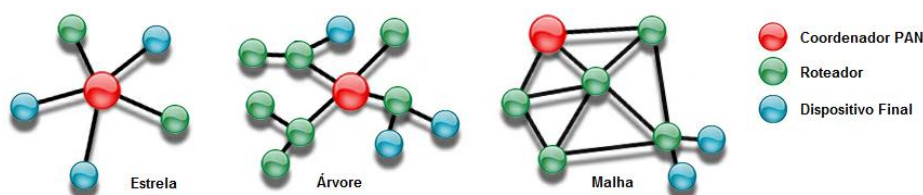


Figura 20 - Topologias de trabalho do *ZigBee*
Fonte: Adaptado de ICPDAS (2012)

- **Topologia em Estrela:** Consiste basicamente na distribuição de dispositivos finais, interligados por um nó central coordenado. É o modo mais simples de ser implementado, mas viável apenas quando existe a visibilidade de todos os dispositivos finais com o coordenador ou roteador, não necessitando de desvios de rota por obstruções no sinal.

- **Topologia em Árvore:** Assume uma distribuição semelhante à topologia em malha, onde o coordenador assume o papel de roteador mestre entre os outros roteadores e os dispositivos finais.

- **Topologia em Malha (Mesh):** É a topologia mais robusta e versátil de todas, onde a rede, além de poder inicializar automaticamente, possui a capacidade de gerenciar as melhores rotas entre os pontos de interesse no transporte de dados, podendo criar desvios em caso de quedas de percurso. Apesar de exigir uma configuração mais complexa, é a disposição que melhor explora a flexibilidade e robustez das redes *ZigBee*.

8.2.1.4. Camadas de Suporte a Aplicação e Camada de Aplicação:

Estas camadas estão intrinsecamente interligadas, coexistindo em compartilhamento de recursos. A camada de suporte a aplicação fornece uma interface entre a camada de rede e a de

aplicação, com serviços como o *Binding* e *Discovery*. O *Binding* faz a união de dois ou mais dispositivos de acordo com a necessidade, o *Discovery* faz a descoberta de pontos ativos ao alcance do dispositivo. Para Barrichello (2009), a camada de aplicação é composta de suporte a aplicação, ZDO's e funções específicas da empresa que desenvolveu o dispositivo. No ZDO define-se o papel deste, se ele atuará como coordenador, roteador ou dispositivo final. Também estão interligadas a esta camada definições de segurança e *binding* do suporte. A cada rádio podem estar associados até 240 objetos, sendo o endereço 0 usado pelo ZDO e o 255 para transmitir dados a todos os objetos de aplicação.

8.3. Distribuição do padrão *Zigbee* dentro das redes por área de atuação:

A figura 21 ilustra a distribuição das tecnologias de rede wireless, cada qual com sua especificidade.

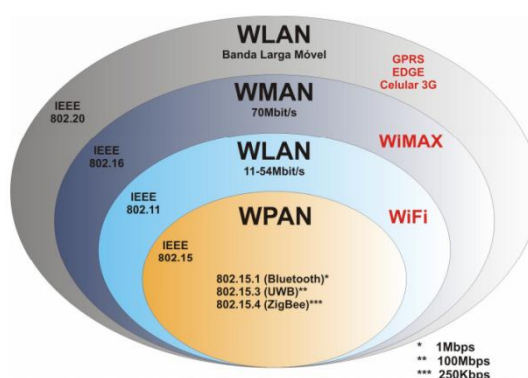


Figura 2 - Padrões Wireless.

Figura 21 - Padrões *Wireless* e sua área de atuação

Fonte: Vasques, B.L.R.P; Coutinho, I.B.A; Lima, M.F.; Carneval, V.P.O.

Então por que utilizar esse padrão se os outros já cumpriam a sua função? A resposta se resume em duas palavras: de acordo com Colvero (2011), robustez e baixo consumo. Os dispositivos que utilizam essa tecnologia são extremamente versáteis e econômicos. Uma rede bem configurada pode comportar em teoria até 65535 nós, rotas perdidas podem ser desviadas através das malhas e o conteúdo entregue no seu destino. De acordo com Monsignore (2007, p. 3), os dispositivos que não realizam roteamento podem ser configurados para operar por até anos com bateria sem necessitar de manutenção, fato que se se compararmos ao Bluetooth,

por exemplo, não é possível devido ao seu consumo mais elevado, lembrando novamente que a taxa de transmissão não é relevante nesse caso.

Uma curiosidade adicional é que a tecnologia continua evoluindo de tal forma que, o módulo utilizado neste trabalho já alcança cerca de uma milha (cerca de 1600 metros), deixando o ZigBee no alcance de uma WLAN (*Wireless Local Area Network*), porém ainda é considerada uma rede WPAN (*Wireless Personal Area Network*), ou seja, de acesso pessoal.

8.4. Classificação dos dispositivos quanto à função na rede:

- **Coordenador** (*Coordinator*): Responsável por inicializar, manter o controle dos nós das redes, distribuir os endereços dos dispositivos e manter o controle da rede entre outras funções. Obviamente só pode ser implementado a partir de um dispositivo FFD.

- **Roteador** (*Router*): Tem a função de gerenciar um nó normal da rede, mas também pode assumir o papel de intermediário entre duas redes (mesmo sem a gerência do roteador), dando possibilidade de expansão da rede. Também requer um dispositivo FFD.

- **Dispositivo Final** (*End Device*): Tem como principal objetivo o controle de sensores (monitoramento), utiliza a sua principal característica de baixíssimo consumo como diferencial e pode ser implementado com dispositivos RFD (não obrigatório). Definido assim por Zucato (2009, p. 34).

8.5. Classificação dos dispositivos quanto ao tipo:

- **FFD** (*Full Function Device*): Segundo Rogecom, são dispositivos de funções completas, este permite desempenhar qualquer função dentro da rede, de coordenador a dispositivo final. Eles podem se comunicar com qualquer dispositivo ao alcance na rede, mas para isso necessitam de um hardware mais completo com no mínimo 32 kB para memória de programa, bem como memória RAM para o armazenamento de tabelas de rotas e configurações. Esses requisitos demandam mais energia, o que torna esse tipo de equipamento pouco adequado para a operação somente com bateria.

- **RFD** (*Reduced Function Device*): Dispositivo de funções limitadas, este permite desempenhar apenas a função de dispositivo final dentro da rede. Esses se comunicam somente

com roteadores ou coordenadores, por sua vez requerem um hardware mais modesto, algo em torno de 6 kB para programa e um controlador de 8 bits, desta forma consomem pouca energia e são ideais para sensoriamento isolado que necessita do uso de baterias, podendo manter-se em atividade por anos com a carga destas. Com o natural desenvolvimento da tecnologia e redução das dimensões dos componentes na microeletrônica, os módulos RFD tem incorporado funções adicionais e, é bem possível que em pouco tempo sejam substituídos por FFDs.

8.6. Modos de Operação AT e API:

Outra característica da tecnologia é proporcionar dois modos distintos de operação, sendo o modo AT ou modo Transparente o mais básico: os dados e comandos podem ser enviados diretamente via terminal de modo serial (enfileirados) através da UART do dispositivo. As figuras 22 e 23 a seguir ilustram claramente o modo AT:

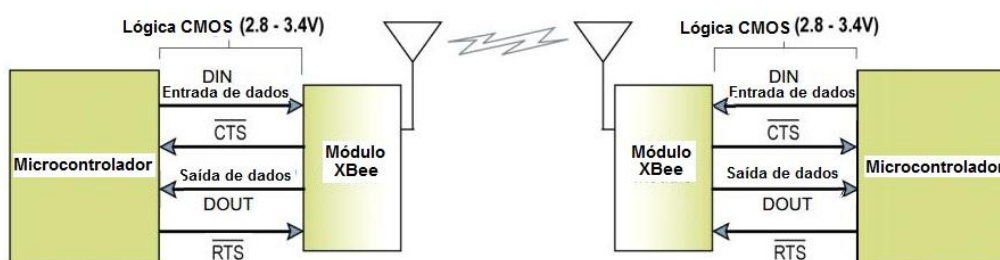


Figura 22 - Comunicação da UART com a interface do computador

Fonte: Digi (2012)

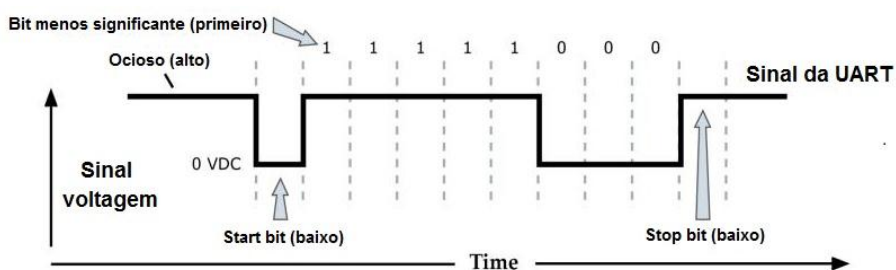


Figura 23 - Transmissão do pacote de dados da UART através do transmissor

Fonte: Digi (2012)

De acordo com a Digi, pode ser percebido o comportamento típico de uma transmissão serial na figura 22, com os controles de interface RTS (*Request To Send*) e CTS (*Clear To Send*), bem como o DIN (*Data In*) e o DOUT (*Data Out*), que são respectivamente os dois controles de transmissão e as duas vias de dados de entrada e saída, da mesma forma como operam os demais equipamentos seriais como mouses e modems mais antigos. A figura 23 ilustra o modo de transmissão serial com o controle de início e fim de transmissão baseado em um bit de *Start* e outro de *Stop*, no caso trata-se da transmissão do número decimal 31, percebe-se que o sentido da fila inicial pelo bit menos significativo.

No modo API (*Application Programming Interface*), um modo alternativo e mais complexo de realizar a transmissão, ao invés de enviar comandos diretamente através da interface serial, os comandos são colocados em uma interface estruturada, ou seja, é feita a comunicação de dados em uma ordenação dos frames pré-definida. Para que este modo de transmissão seja ativado, existe a necessidade de habilitar o comando AP. O modo API permite ao programador definir como comandos, respostas e status serão enviados e recebidos através da interface de frames da UART.

A figura 24 ilustra como é dividida a estrutura de frame de dados da UART, que ainda pode contar com caracteres de “escape” (por definição, caracteres de escape permitem mudar o significado dos frames subsequentes, caso este recurso seja necessário). Quando o comando AP for definido como 1 o frame é normal, e definido em 2 habilita os caracteres de escape.

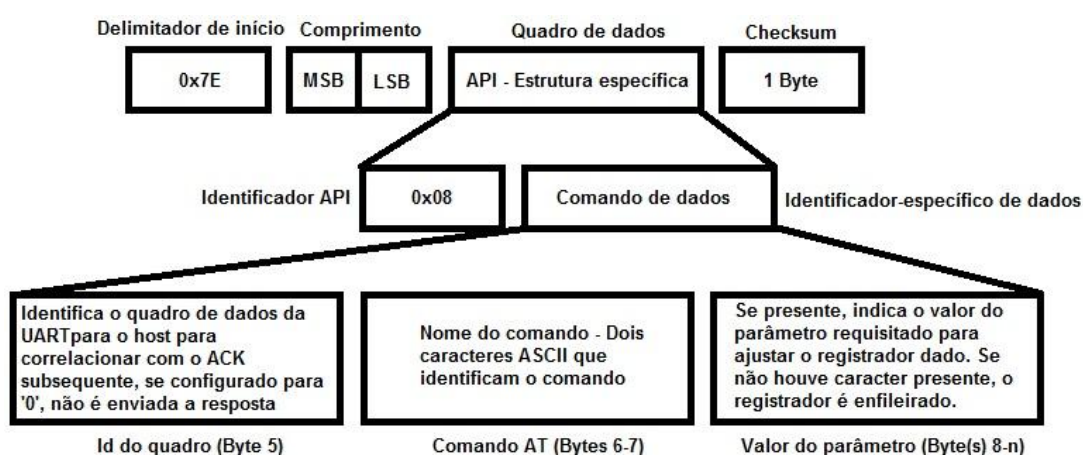


Figura 24 - Estrutura do frame API
Fonte: Laguna, R. (2009)

A programação em modo API exige do desenvolvedor um conhecimento mais apurados do funcionamento dos quadros, a seguir está um exemplo de um frame API destinado a endereçamento, apresentado por Ruben:

```

7E 00 1D 11 01 00 13 A2 00
40 3E 25 75 00 00 01 01 04
02 01 04 00 00 18 01 01 00
00 00 29 63 CE 9F

7E          : Start frame
00 1D       : Length (29 bytes)
11          : explicit addressing transmit
01          : Frame id
00 13 A2 00 40 3E 25 75 : 64 bit address
00 00       : 16 bit address
01          : source endpoint
01          : destination endpoint
04 02       : cluster id (temperature measurement cluster id/ Measurement
01 04       : profile id (home automation profile)
00          : broadcast radius
00          : options
18 01 01 00 00 00 29 63 CE : payload
9F          : checksum

```

Figura 25 - Exemplo de frame API

Fonte: Laguna, R. (2009)

8.7. Formas de difusão das mensagens:

O padrão *ZigBee*, da mesma forma que a maioria das redes, permite propiciar basicamente três formas de distribuir os pacotes de acordo com Rogercom:

- **Ponto a ponto ou *Unicast*:** É o modo mais simples, onde os pacotes são endereçados diretamente a um único destino. Para isso basta que o endereço de destino seja o correspondente ao receptor desejado. Essa opção, além da simplicidade, proporciona grande vantagem em relação à segurança contra invasões e também um melhor desempenho.

- **Multiponto ou *Multicast*:** Este modo utiliza o endereçamento de grupo para enviar os pacotes a um determinado grupo de interesse, para que isso ocorra é utilizado um bloco de 16 bits dedicado a essa finalidade.

- **Difusão ou *Broadcast*:** Os pacotes são endereçados a todos os membros da rede, para que essa opção seja implementada basta utilizar o endereço MAC 0xFFFF, desta forma todos os equipamentos entendem que devem processar a mensagem recebida.

9. ZIGBEE DIGI - XBEE PRO S2B

Este equipamento, base de comunicação principal entre os elementos ativos da rede, foi o primeiro a ser configurado, devido especialmente ao espaço limitado para testes e a necessidade de poucos recursos para tal. Paralelamente a este procedimento foi realizada a elaboração do sistema de suporte que iria sustentar todo o conjunto de sensoramento e transmissão.

O módulo utilizado escolhido foi o *XBee Pro S2B* fabricado pela Digi, um dispositivo que atende ao padrão 802.15.4 nas camadas inferiores e tem as camadas superiores definidas pela *ZigBee Alliance*, bem como a de aplicação, conforme descrito anteriormente. Segundo a Digi, "... é designado para operar sobre o protocolo *ZigBee* e suportar a necessidade de baixo custo, baixa potência em sensores de redes sem fios”.



Figura 26 - Módulo de *ZigBee XBee Pro S2*
Fonte: Digi (2012)

Este transmissor/receptor opera na frequência de 2,4 GHz, portanto na faixa de frequências ISM. Este equipamento tem como principais características:

- Alcance indoor ou urbano de 60 a 90 metros com variação.
- Alcance outdoor de 1500 a 3200 metros em linha de visada.
- Potência de transmissão de 10 a 68 mW ou 10 a 18 dBm.
- Taxa de transmissão em RF de até 250.000 bps.
- Sensibilidade de recepção de até -102 dBm.
- Tensão de trabalho de 2,7 a 3,6 Volts.
- Topologia de rede ponto-a-ponto, ponto-a-multiponto, *peer-to-peer* e *mesh*.
- 15 canais de RF. *DSSS (Direct Sequence Spread Spectrum)* é a sequência direta de espalhamento do espectro.

- Antena externa, ou seja, possui um conector RPSMA para o acoplamento da antena externa, visando um melhor ganho na transmissão ou recepção em relação aos modelos que utilizam antena interna, podendo ser mudado de acordo com a antena instalada.

O dispositivo é um FFD, ou completo, que pode atuar como qualquer elemento da rede *ZigBee*. A escolha deste modelo se deve especialmente pelo grande alcance em ambiente aberto, aliado ao ambiente poluição eletromagnética que é característico de um centro urbano e especialmente um aeroporto. Se considerarmos pelo manual do fabricante que a característica de sensibilidade na recepção do equipamento é de até -102 dBm ou $63,1 \cdot 10^{-12}$ mW e o *payload* é pequeno em relação ao controle do pacote recebido, teremos assim uma recepção garantida sob condições mais severas, por exemplo, ao aplicarmos a equação da propagação de sinal em espaço livre em condições ideais, teríamos, considerando o ganho unitário das antenas transmissora (G_t) e receptora (G_r), a uma distância estimada do sensor mais longe de 400 metros do *Gateway*, com uma perda unitária também L por fatores não associados à propagação, utilizando 10 mW que é a potência mínima de transmissão, teríamos:

Sendo a frequência $f = 2,4$ GHz, o comprimento de onda encontra-se com a relação entre a velocidade da luz $C \cong 3 \cdot 10^8$ m/s. Assim $\lambda = \frac{C}{f} = \frac{3 \cdot 10^8}{2,4 \cdot 10^9} = 0,125$ m

$$Pr = \frac{Pt \cdot G_t \cdot G_r \cdot \lambda^2}{(4\pi)^2 \cdot d^2 \cdot L} = \frac{10 \cdot 1 \cdot 1 \cdot 0,125^2}{(4\pi)^2 \cdot 400^2 \cdot 1} = 6,184 \cdot 10^{-9} \text{ mW}$$

Convertendo para escala logarítma, em decibéis:

$$Pr(db) = 10 \log Pr = 10 \log 6,184 \cdot 10^{-9} = -82,087 \text{ dBm}$$

Portanto, apesar de não ser apropriado o uso de potências muito baixas para não prejudicar a relação sinal/ruído, com 10 mW no transmissor, a potência recebida ficaria quase 20 dB acima do limiar do receptor, e ainda sobram 58 mW de ajuste no transmissor. Além dessa sobra de potência, temos que considerar que a rede permite o desvio de rotas adaptativo através dos seus elementos roteadores por rede em malha (*mesh*), o que permite a recuperação do sinal pelos nós.

9.1. Configuração do dispositivo:

O modo mais fácil de configurar módulos *ZigBee* é através de comandos *AT*, onde, apesar de recursos mais limitados, requer apenas que uma interface serial se interponha entre o computador e o módulo. Porém por economia de hardware o dispositivo *ZigBee* não possui a interface serial pronta para a comunicação e por isso é necessária a montagem de um circuito de interface como ilustra a figura 27 a seguir:

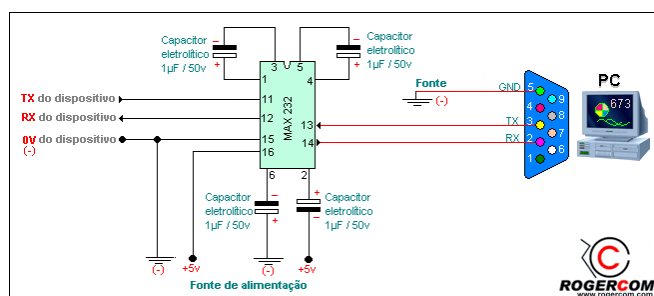


Figura 27 - Interface RS-232 (serial) para configuração do módulo *XBee*
Fonte: Rogercom (2012)

Existem alguns desenvolvedores nacionais que trabalham com este tipo de equipamento e soluções para eles, como o site www.rogercom.com.br, que pode ser considerado um referencial muito importante para pesquisas de acadêmicos e usuários iniciantes nesta tecnologia. Através do desenvolvedor podem-se obter programas gratuitos para configuração através da interface como a da figura 27, o programa Rcom Serial v.1.2 e outros, com a montagem e ligação entre os elementos (computador, interface e módulo *ZigBee*), a passagem de parâmetros no modo transparente é direta. Também pode-se utilizar um terminal do tipo *Hyperterminal* do Windows para o mesmo fim.

Existe também uma solução mais prática, especialmente para os novos computadores que não possuem mais interface serial, que é uma interface desenvolvida pelo Sr. Antônio Rogério Messias, proprietário da Rogercom, que faz a comunicação entre os equipamentos através da interface USB (*Universal Serial Bus*), a qual é a mais popular da atualidade para comunicação de dados (e alimentação inclusive) com periféricos compatíveis. A interface em questão é a CON-USBBEE como ilustra a figura 28.



Figura 28 - Interface CON-USBBEE
Fonte: Rogercom (2012)

Percebe-se claramente o quanto esta interface facilita o trabalho de configuração dos módulos, pois é possível conectá-los diretamente ao conector da interface sem risco de inversão de pinos e possível queima do equipamento. Outra grande vantagem é poder dispor da tecnologia *Hot-Plug* inerente à interface USB, a qual permite que o dispositivo seja conectado e desconectado sem que seja necessária a interrupção de energia, não presente na interface serial, a qual pode ser facilmente danificada com o mesmo processo.

A montagem física do *ZigBee* na interface é apenas por contato através de conectores, e no computador basta a instalação do *driver* apropriado, sendo que é fornecido em multi-plataformas (*Windows* 98, ME, 2000, XP, Vista, x64 e também para Linux e Mac).



Figura 29 - Interface CON-USBBEE montada com o módulo
Fonte: Rogercom (2012)

É possível notar pela figura 29 que existem alguns *leds* indicadores e um botão de *reset*, através destes é possível ter noção, sem precisar da tela de um computador, basicamente o nível de sinal (fraco, moderado ou forte), tráfego TX (transmissão) e RX (recepção) e associação do módulo (na rede) simultaneamente. Uma vez montados e instalados, os módulos *ZigBee* podem ser configurados de acordo com a necessidade, podendo inclusive ser feita a atualização do *firmware* do módulo e assim aplicar correções de problemas através de imagens de arquivo fornecidas pelo fabricante.

O *ZigBee* opera em modo AT basicamente da seguinte maneira:

A atuação inicial do equipamento é em modo em que está pronto para a transmissão ou recepção de dados, mesmo quando vem de fábrica em modo padrão ele tem por característica voltar sempre para este modo denominado *Idle* (*ocioso*). Para iniciar o modo de

configuração e passagem de dados é necessário digitar três vezes o símbolo de soma (+++). Se tudo estiver bem o módulo responderá com um “OK”, indicando que entrou em modo de configuração, a partir deste ponto pode-se começar a digitar os comandos AT no formato AT<comando><enter>, como mostra a figura 30.

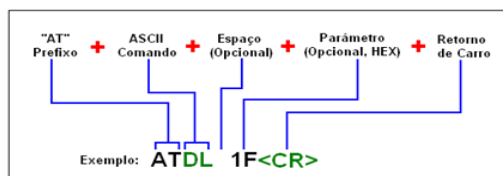


Figura 30 - Estrutura do comando AT

Fonte: Rogercom (2012)

A sequência de comandos será sempre seguida da resposta do módulo, com “OK” indicando o sucesso ou um código de erro. O único cuidado a ser tomado é que o tempo limite padrão de retorno para o modo *Idle* é de apenas dez segundos, e o módulo não recebe mais comandos, sendo necessário digitar a sequência (+++) novamente.

Até aqui foi descrito sequencialmente qual a maneira mais prática de configurar o *ZigBee* para testes, iniciando pela interface serial agora pela interface USB (*Universal Serial Bus*). Porém é necessário destacar aqui que todos os comandos executados em terminal são colocados em memória volátil e serão perdidas no primeiro desligamento de energia, então será necessária a gravação em memória permanente antes que todo o trabalho se perca. É possível atuar deste modo, mas não é nada prático, se lembrarmos de que uma rede *ZigBee* pode comportar até 65.535 dispositivos, é bem provável que muitos tenha a configuração muito similar entre si e digitar comando a comando seria uma tarefa extremamente improdutiva.

A grande vantagem de se trabalhar com tecnologias dominantes é que existe muito trabalho paralelo e também muito interesse das indústrias em disseminar ainda mais os seus produtos. Através da *Maxtream*, da qual a Digi Internacional é distribuidora, foi disponibilizado um programa chamado X-CTU, o qual permite exercer a configuração e gravação dos parâmetros dos módulos de forma extremamente mais simples, além de dispor das funcionalidades de terminal, blocos para testes com *strings* e *strings* padrão prontas e visualização simplificada da configuração atual do *ZigBee* gravada. Através deste programa é simples também realizar a importação e exportação de arquivos de configuração, de maneira que pode-se manter um *backup* permanente de configurações personalizadas e também fazer a

replicação por diversos módulos, alterando-se somente o necessário de modo individual. A figura 31 mostra a interface de trabalho do X-CTU.

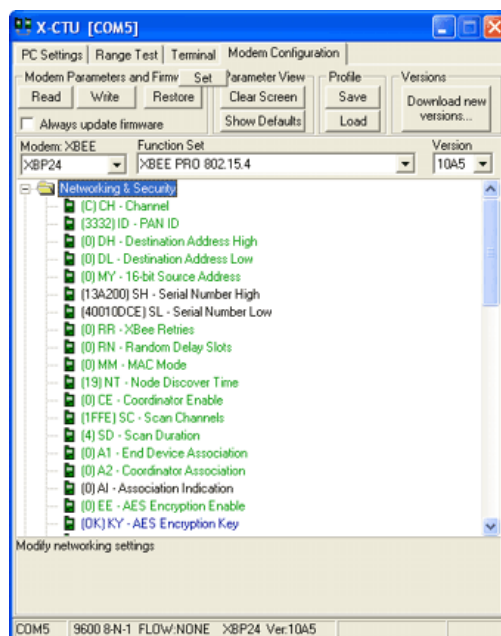


Figura 31 - Programa X-CTU

Fonte: Rogercom (2012)

Observa-se claramente na figura 31 que a configuração do dispositivo é mostrada na tela, os principais parâmetros podem ser modificados diretamente clicando sobre este. Esta tela só é possível ser vista quando o *driver* estiver corretamente instalado e o *ZigBee* conectado.

A instalação do programa não requer maiores explicações, basta executar o instalador e seguir as orientações. Este programa foi desenvolvido para trabalhar somente no sistema operacional *Windows* e posteriores, no caso específico deste trabalho operou sobre o *Windows Seven* sem problemas. Não foram realizados testes com máquinas virtuais em outros sistemas.

9.2. Configurando o programa X-CTU:

Assim como foi dito, o *ZigBee* precisa estar pronto para configurar, no experimento usamos o adaptador CON-USBBEE, que foi anteriormente instalado com o *driver* fornecido pelo fabricante, as informações a seguir são uma mescla do manual do X-CTU com a prática. A primeira tela de configuração, na aba *Pc Settings*, mostra os principais dispositivos de

comunicação serial, de maneira que basta selecionar a interface e configurar os parâmetros, sendo que o valor padrão serve bem à maioria dos casos:

- *Baud*: 9600 bps – taxa de transmissão (1.200 a 230.400);
- *Flow control*: *none* – controle de fluxo (*none/hardware* ou por *software* (xon-xoff));
- *Data bits*: 8 – quantidade de *bits* de dados (4 a 8);
- *Parity*: *none* – paridade (*none/odd/even/mark/space*);
- *Stop bits*: 1 – quantidade de *bits* de *stop* (1/1,5 ou 2).

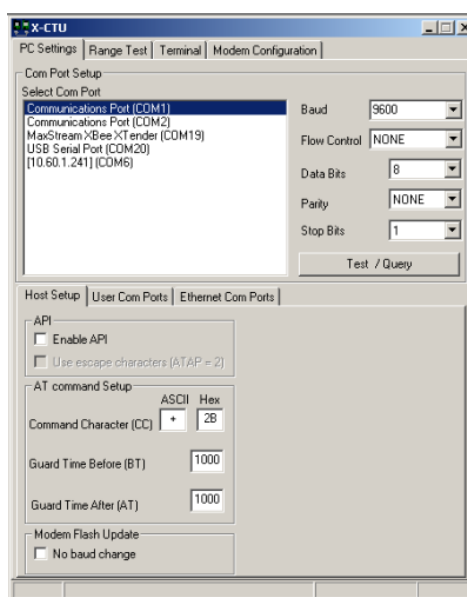


Figura 32 - Programa X-CTU
Fonte: Digi (2012)

Existe o botão de *Test/Query* presente, uma vez selecionada uma porta serial ou um modem (que também usa porta serial padrão RS-232), este permite que seja feito um teste de básico de comunicação na interface. Caso esta comunicação falhe não será possível acessar o *ZigBee* e, portanto este problema tem que ser resolvido antes de ir adiante no programa. Vale ressaltar que para o experimento não foi necessário alterar nenhum dado nesta aba.

Uma vez estabelecida a comunicação serial é possível iniciar a configuração do módulo, passando para a aba *Modem Configuration*. Neste ponto é importante frisar que sempre é interessante obter a versão mais recente do X-CTU, devido aos modelos *XBee* implementados no software, o qual acessamos pelo botão *Read* para que o dispositivo seja carregado com seus parâmetros pelo sistema. Caso não ocorra a descoberta correta, é possível selecionar manualmente pelo botão *Modem* com a versão mais próxima possível do *firmware* utilizado. A última etapa é

escolher no *Function Set* a finalidade a que se destinará o *ZigBee* na rede, como *end device*, *router* ou *coordinator* e o modo de operação, que neste caso, é no modo transparente *AT*.

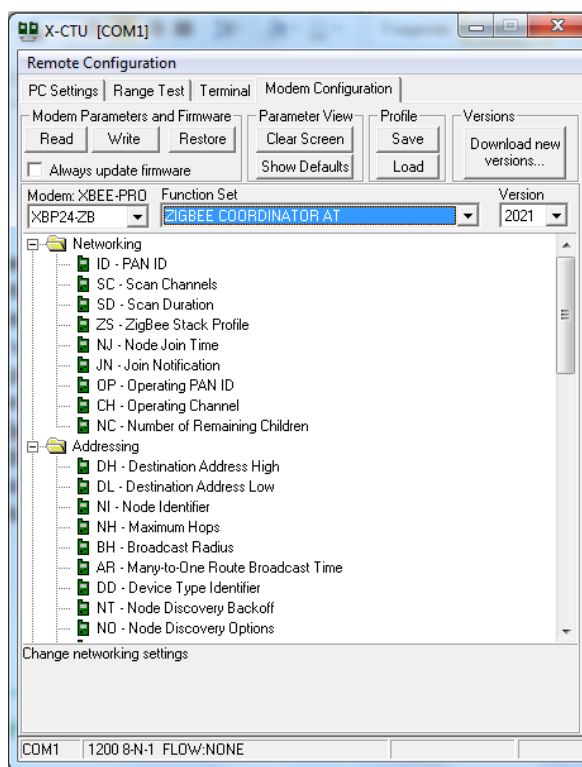


Figura 33 - Programa X-CTU
Fonte: Digi (2012)

Esta aba contém a maioria dos recursos disponíveis na configuração, de forma bem clara, clicando diretamente nos ícones e modificando os valores nas caixas de diálogo que aparecem. Paralelamente podem ser digitados os comandos AT na aba *Terminal*, da maneira descrita anteriormente. Vale lembrar novamente que estes parâmetros alterados são válidos enquanto estiver energia presente. Quando ligamos novamente a energia o módulo irá carregar os valores lidos na memória residente, caso queira tornar as alterações feitas para o modo permanente, basta utilizar o botão *Write* e o programa irá gravar no módulo.

Vale destacar ainda dois botões extremamente úteis: o *Save*, que permite gravar em arquivo a configuração (sem gravar no módulo) para *backup*, e o *Load* que carrega um arquivo pré-gravado para o programa. Esses botões são muito utilizados em testes, recuperação em acidentes e replicação de configurações. O botão *Restore* auxilia na recuperação de perdas acidentais da configuração, o programa ainda oferece a opção de atualização de *firmware* para versões mais recentes, se estiverem disponíveis no fabricante.

A aba *Terminal* não merece maior destaque, pois como foi explicado, a comunicação pode ser feita por qualquer terminal que faça acesso à UART por comandos AT.

A aba *Range Test* é muito importante, no sentido de que já provê um padrão pré-elaborado de dados em um bloco padrão de 32 bits, o que pode ser modificado de várias maneiras, tanto em conteúdo, tamanho do bloco e *delay* na transmissão dos blocos. Esta ferramenta é essencial para testes de *burn-in* e ajustes dos rádios, quando for necessária uma calibração adicional em locais de maior perturbação do sinal que possam gerar erros.

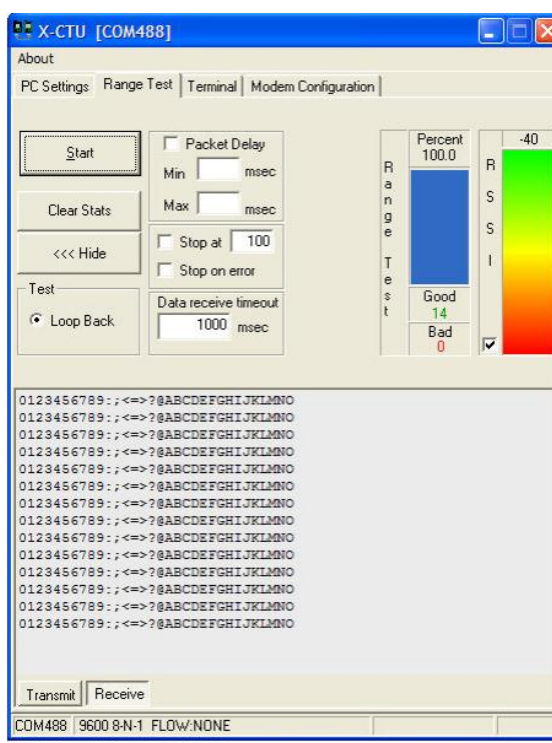


Figura 34 - Programa X-CTU

Fonte: Digi (2012)

No lado direito desta aba, também é mostrado a qualidade dos blocos recebidos em condições ou com defeito, bem como a intensidade de sinal *RSSI* (*Received Signal Strength Indication*), que é uma relação percentual de potência em comparação ao padrão em dBm (potência em decibéis em relação a um miliWatt) utilizado pela maioria dos fabricantes, alguns poucos optam por utilizar esta relação para a aferição de seus produtos. A empresa *WildPackets Inc.* fornece uma prática demonstração de como calcular e define que “O *RSSI* é um parâmetro opcional que tem um valor de 0 até *RSSI* máximo. Este parâmetro uma medida na subcamada física, da energia observada na antena, usada para receber o *PPDU* em uso. O *RSSI* deve ser medido entre início do *frame* delimitador (*SFD*) e o fim do cabeçalho de verificação de erro

PLCP (HEC)”. PLCP, para melhor entendimento pode ser explicado com uma visão em relação ao padrão mais utilizado, o 802.11. É definido pela CWAP (*Certified Wireless Analysis Professional*), uma divisão da CWNP Inc., uma indústria de TI especializada em certificação com foco no padrão 802.11, onde é considerado como sendo um Protocolo de Convergência da Camada Física e o PPDU é um Protocolo PLCP de Unidade de Dados, a subdivisão pode ser melhor visualizada na figura 35.

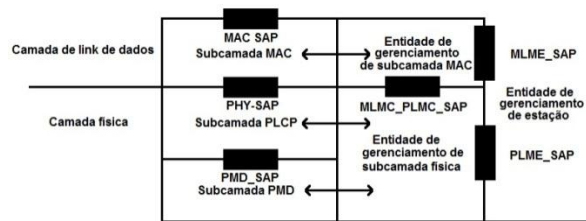


Figura 35 - Subdivisão da camada física 802.11
Fonte: Adaptado de CWAP (2012)

10. GATEWAY DIGI CONNECTPORT X4 ZB HSPA

Este equipamento é um dos mais completos modelos fabricados pela Digi, com a finalidade de integrar equipamentos através de um *middleware* especialmente produzido para cada caso e do software desenvolvido para cada aplicação. O modelo em questão permite fazer uso das interfaces padrões RS232, 802.3, 802.15.4, USB e 3G, algumas simultaneamente, para interligar diversos dispositivos em uma interface que pode ser adaptada conforme a situação desejada.



Figura 36 - Gateway Connectport X4
Fonte: Digi (2012)

Conforme o fabricante, o *Gateway X4*, possui as seguintes características:

- Interfaces: Porta *Ethernet* com interface padrão RJ-45, porta serial padrão RS232 com interface db9, porta padrão USB 2.0, interfaces RF padrão *ZigBee XBee Pro ZB 2,4 GHz*, *opcional GSM HSPA*, *GSM genérico HSPA*, *verizon EVDO*, *Sprint EVDO* e módulos *WiMAX*;
- Segurança/roteamento: Permite NAT, *Port Forwarding*, listas de controle de acesso (filtragem IP);
- Gerenciamento: HTTP/HTTPS, interface web, controle de acesso por senha, controle de serviço de porta IP, serviço opcional de gerenciamento seguro através da *iDigi Device Cloud* (gerenciamento por nuvem proprietária Digi);
- Segurança adicional: Tunelamento SSL, SSHv2, Fips 197 (porta serial);
- VPN: Ipsec com IKE/ISAKMP, suporte a múltiplo túnel, DES, 3DES e encriptação de até 256 bits AES, *VPN Pass-through*, *GRE forwarding*.

Este dispositivo serve para o projeto como meio integrador de recursos, é através dele que é feito o acesso de consulta aos sensores, através do ambiente que opera sob o programa de testes, desenvolvido em *Python*. O *Gateway* funcionou dentro de uma rede local através dos padrões 802.3 e 802.11 sem problemas e também na nuvem iDigi, onde o status pode ser acessado em qualquer parte do mundo que tenha acesso à internet. A figura 37 mostra o interior do equipamento com os módulos *ZigBee*, 3G e os *slots* GSM vazios:



Figura 37 - *Gateway X4* – visão interna
Fonte: Autor

10.1. Configuração do *Gateway*:

O equipamento é configurável através de um navegador web comum, bastando que seja feito o pelo IP do mesmo através da porta 80 padrão. De acordo com o manual do equipamento, ele já vem com o servidor DHCP habilitado por padrão, bastando que seja conectado um computador na mesma rede para iniciar a configuração. O acesso pode ser feito sem qualquer equipamento extra, bastando para isso que tenha um cabo *crossover* conectado entre ambos.

De acordo com o fabricante também, o endereço padrão para a linha *Connectport* X2/X3 ou X4 pode ser 0.0.0.0 ou 192.168.1.1 dependendo do *firmware* utilizado. Caso o *Gateway* tenha o seu IP original modificado, é possível obter o programa *Digi Device Discovery Utility* no site do fabricante, este software tem a finalidade de detectar o dispositivo na rede e mostrar o seu endereçamento IP e MAC, facilitando a busca.

A figura 38 mostra a tela inicial do equipamento quando configurado em modo local, nota-se claramente que a configuração é muito semelhante em modo a um roteador ou modem doméstico, porém com finalidade muito mais específica e complexa.

O *Gateway* possui, assim como a maioria dos equipamentos, uma memória pré-gravada (*firmware*) que pode (e deve) ser atualizado na medida do possível para a correção de problemas detectados pelo fabricante (segurança) e também adição de novas funcionalidades. Também existe uma memória de usuário não volátil que pode ser utilizada especialmente para a inserção de bibliotecas personalizadas, de maneira a tornar este aparelho um concentrador de recursos mais independente possível, ajustando para a configuração desejada.

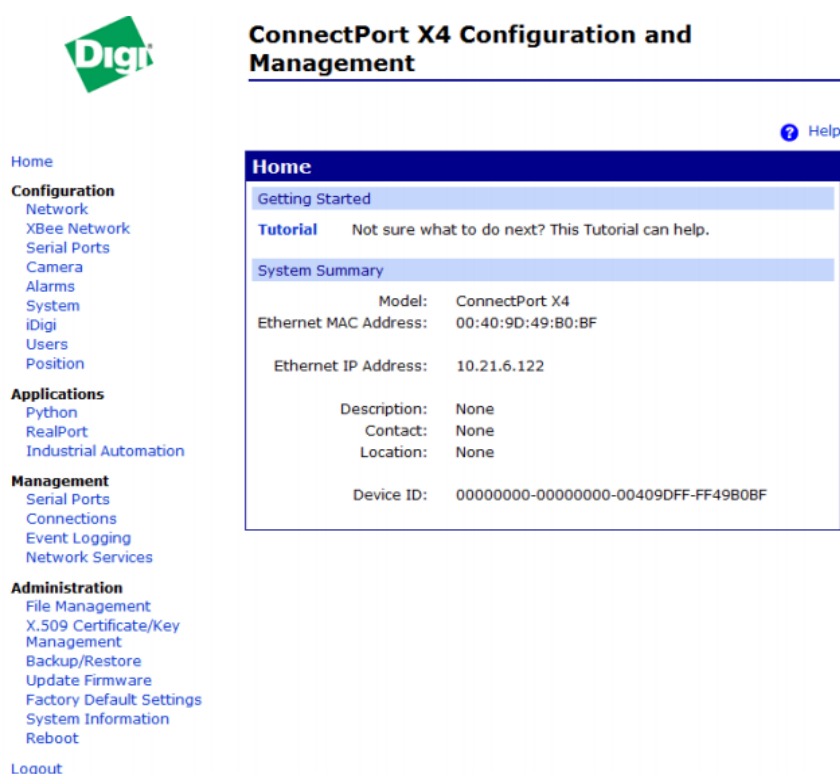


Figura 38 - Tela inicial do *ConnectPort X4*

Fonte: Autor

A configuração básica para o experimento não requer muitas alterações no sistema, sendo que inicialmente partimos para a configuração do endereçamento IP. De acordo com a figura 39, o programa do *X4* aceita que ele adquira o seu endereço automaticamente, caso seja fornecido por um roteador da rede, mas por se tratar de um elemento que atua com concentrador de recursos, é aconselhável que sejam configurados os endereços IP, máscara e o *gateway* padrão da rede, para que a referência seja sempre a mesma na rede, pois em uma configuração de IP dinâmico, uma queda de energia poderia habilitar outro host tomar posse do IP de trabalho e o *Gateway X4* passaria a ter outro endereço. Caso ele esteja mapeado por

outros computadores na rede seria perdida a conexão e consequentemente o acesso aos dados de controle.

Network Configuration

▼ Ethernet IP Settings

☐ Obtain an IP address automatically using DHCP *
☒ Use the following IP address:

* IP Address:
 * Subnet Mask:
 Default Gateway:

☒ Enable AutoIP address assignment

* Changes to DHCP, IP address, and Subnet Mask may affect your browser connection.

Figura 39 - Configuração da rede do *ConnectPort X4*
Fonte: Autor

Por padrão, como foi dito, o servidor DHCP interno habilitado para facilitar a configuração inicial do equipamento, desta forma basta determinar a faixa de endereços desejados a serem oferecidos aos hosts, ou pode-se desabilitar essa opção caso seja desejado que o *host* tenha endereço IP fixo.

Network Configuration

► Ethernet IP Settings

▼ DHCP Server Settings

Note: For the DHCP server to operate, the ConnectPort X4 must be config notes below.

☒ Enable Dynamic Host Configuration Protocol (DHCP) Server

Scope Name:

* IP Addresses: to

Lease Duration: days hrs mins

☒ Wait specified delay before sending DHCP offer reply

Delay: ms

☐ Check that an IP address is not in use before offering it

☒ Send the DHCP Server IP address as a DNS Proxy Server

Figura 40 - Configuração DHCP do *ConnectPort X4*
Fonte: Autor

A descoberta dos dispositivos anexos internos como a placa de rede padrão 3G, os cartões *GSM* e o *ZigBee* se dá de forma automática. A partir dos módulos encontrados pelo sistema, e possível configurá-los pelo próprio *Gateway*, inclusive remotamente, alterando os valores e salvando-os. Alguns serviços de rede estão pré-disponíveis na configuração padrão, como descrito na descoberta de dispositivos: *Realport*, protocolos como o *SNMP* (para gerenciamento dos

nodos da rede), e serviços de acesso como Telnet, SSH, HTTP e HTTPS, como mostra a figura 41, nesta tela pode tanto desabilitar um serviço como mudar a sua porta padrão. No caso do protocolo SNMP, o qual é muito importante na identificação dos dispositivos da rede, é deve-se navegar até a aba *System Configuration* e configurar as comunidades pública e privada. As comunidades *public* e *private* são configuradas como padrão no equipamento.

Network Configuration	
▶ Ethernet IP Settings	
▶ DHCP Server Settings	
▼ Network Services Settings	
<input checked="" type="checkbox"/> Enable Device Discovery (ADDP)	
<input checked="" type="checkbox"/> Enable Encrypted RealPort	TCP Port: 1027
<input checked="" type="checkbox"/> Enable RealPort	TCP Port: 771
<input checked="" type="checkbox"/> Enable Network Management Protocol (SNMP)	UDP Port: 161
<input checked="" type="checkbox"/> Enable Secure Shell Server (SSH)	TCP Port: 22
<input checked="" type="checkbox"/> Enable Telnet Server	TCP Port: 23
<input checked="" type="checkbox"/> Enable Web Server (HTTP)	TCP Port: 80
<input checked="" type="checkbox"/> Enable Secure Web Server (HTTPS)	TCP Port: 443
<input type="button" value="Apply"/>	

Figura 41 - Configuração de serviços do *ConnectPort X4*

Fonte: Autor

Para alterar a configuração do *ZigBee* interno ou outro da rede, é necessário ir até a aba *XBee Configuration* e depois em *XBee Devices*, será apresentada uma tela similar à figura 42:

XBee Configuration				
▼ XBee Devices				
Gateway Device Details				
PAN ID: 0x78e9 - 0xa6dc68775b2079f8				
Channel: 0x13 (2445 MHz)				
Gateway Address: 00:13:a2:00:40:70:9c:25				
Gateway Firmware: 0x2170				
Network View of the XBee Devices				
Select a device to configure:				
Node ID	Network Address	Extended Address	Node Type	Product Type
[1f3e]		00:13:a2:00:40:3b:8e:2c	end device	Unspecified
[0000]		00:13:a2:00:40:70:9c:25	coordinator	X4 Gateway
1 coordinator, 1 end device				
<input type="button" value="Discover XBee Devices"/>				
<input type="checkbox"/> Clear list before discovery				
▶ Gateway Firmware Update				
▶ OTA Firmware Update Setup				
▶ OTA Firmware Update Status				

Figura 42 - Configuração *XBee* no *ConnectPort X4*

Fonte: Autor

Pode-se perceber na figura 42 que os nodos da rede *ZibBee* podem ser identificados de maneira automática pela tecla *Discover XBee Devices*. Caso estejam operantes e endereçados corretamente, e ainda dentro do raio de alcance dos rádios, eles devem aparecer como mostra

a figura, com o seu endereço completo e a sua função na rede. Uma vez selecionado um dispositivo pode-se ver no topo da tela que aparece o endereço PAN ID da rede, o canal utilizado com a sua frequência e até a versão do *firmware*. Dentro desta mesma *XBee Configuration* existem ainda o *Basic Settings* e o *Advanced Settings*, que permitem configurar uma a uma os parâmetros de cada módulo *XBee* selecionado. A configuração utilizada no projeto será descrita no decorrer deste relatório.

Basicamente esta configuração permite que o projeto possa ser implementado de forma inicial, ainda faltando o programa de aquisição de dados. Quanto às demais funções do *ConnectPort X4*, não utilizadas neste projeto, permitem que se faça o uso e configuração por nuvem iDigi ou outra, DNS dinâmico, uso de VPNs, redirecionamento de portas de comunicação, uso de GPS, uso de dispositivos móveis GSM (celular), disparos de alarmes SNMP com envio via SMTP (email) ou SMS (via unidade móvel).

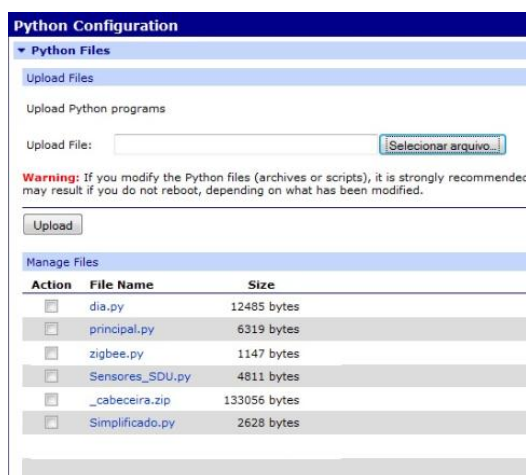


Figura 43 - Armazenamento de arquivos no *ConnectPort X4*

Fonte: Autor

Quanto aos programas, eles podem ser armazenados na memória do equipamento e acionados automaticamente ou apenas ficarem dispostos como na figura 43, o *Gateway* pode inclusive dispor de uma câmera de captura. Um recurso a destacar também é a geração de um *log* interno de eventos, que pode ser obtido na aba *Event Logging* conforme a figura 45. O acionamento automático dos programas dá-se pela aba *Python Configuration* e a seguir *Auto-Start Settings*, onde é possível colocar a linha de comando com o nome do arquivo e argumentos, conforme mostra a figura 44.

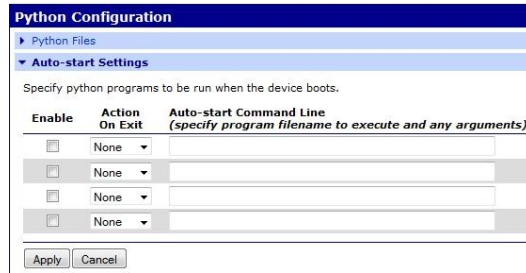


Figura 44 - Início automático de programas no *ConnectPort X4*
Fonte: Autor

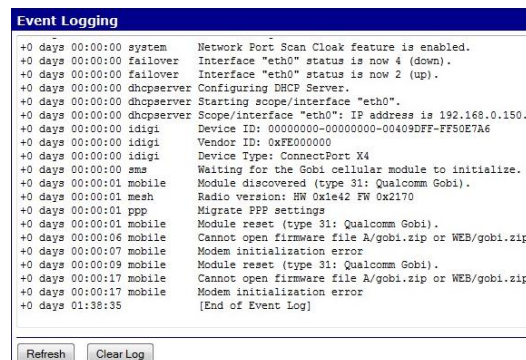


Figura 45 - Log de atividade do *ConnectPort X4*
Fonte: Autor

Como pode ser percebido, este equipamento apresenta um *log* de eventos que pode ser analisado, como forma de estabelecer a causa de problemas de funcionamento do sistema.

11. KIT DE DESENVOLVIMENTO IDIGI PARA O GATEWAY CONNECTPORT X4

11.1. Ambiente de desenvolvimento iDigi

A *Digi Internacional Inc.* desenvolveu, em conjunto com o projeto *Eclipse*, *Aptana* (adquirida recentemente pela *Appcelerator* e *RXTX*, uma suíte completa para desenvolvimento de aplicações em *Python*). Funcionando como uma IDE de desenvolvimento de projetos baseados no *hardware* dos produtos comercializados pela empresa. Embora seja uma solução proprietária, alguns módulos de outras marcas poderiam funcionar desde que adaptados ao software, exigindo desenvolvimento apropriado.

A Digi (www.digi.com) é uma empresa que faz pesquisa e implementação de soluções em comunicação sem fio ou cabeada, utilizando as mais populares tecnologias de comunicação para interligar dispositivos em redes locais ou remotas. A *Aptana* (www.apтана.com) é uma empresa que projeta IDEs de desenvolvimento web *open-source*, em *Python* que, juntamente com a *Appcelerator*, agora mantém uma nuvem de trabalho com grande capacidade de armazenamento e gerenciamento, especialmente para dispositivos móveis. A *RXTX* (www.rxtx.org) é uma organização que mantém projetos de comunicação para interfaces, como a serial ou paralela em aplicações *Java*. O projeto *Eclipse* é como os responsáveis definem “uma espécie de ferramenta universal – uma IDE aberta e extensível para qualquer coisa e nada em particular”. Ainda segundo está descrito no site do desenvolvedor, “o aspecto mais conhecido do projeto é a IDE para *Java*, mas ele não é apenas sobre isso. O que existe é uma grande infraestrutura para construção de diversos tipos de ferramentas e *frameworks* relacionados ao desenvolvimento de software. O *Eclipse* é um projeto *open-source*, livre de patentes, independente de fornecedor e multi-plataforma”.

11.2. Acesso ao ambiente *Digi Developer Cloud*

Esta suíte de programas, anteriormente descrita, é fornecida pela empresa para o desenvolvimento de aplicações utilizando o *Gateway* em questão e outros modelos,

basicamente possui todos os recursos disponíveis e já implementa bibliotecas específicas criadas para o ambiente de trabalho.

Fazendo acesso ao site <http://www.idigi.com/gatewaydevelopmentkit> é possível ter acesso ao programa de descoberta do dispositivo.

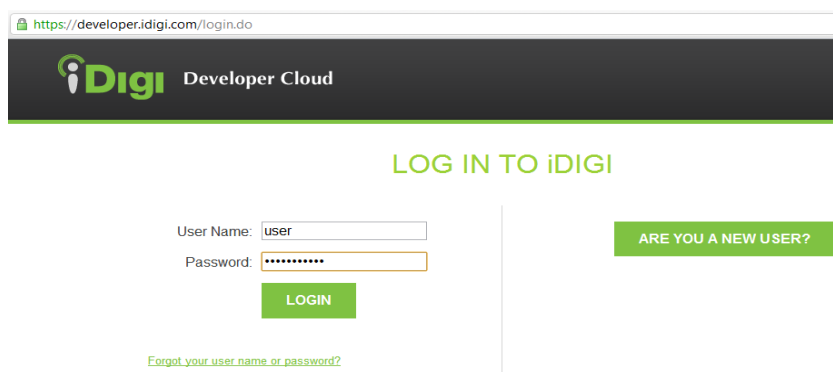


Figura 46 - Tela inicial do sistema de desenvolvimento em “nuvem” da *Digi.Inc*.

Fonte: Autor

De acordo com a figura 46, basta realizar um *login* pré-configurado, onde o fabricante solicita algumas informações do desenvolvedor, para ter acesso “em nuvem” ao sistema completo. Através deste sistema é possível monitorar e configurar o *Gateway X4* remoto e/ou seus dispositivos agregados teoricamente de qualquer lugar do mundo por meio da rede mundial através de um terminal.

A configuração do sistema em rede é simples, requer apenas uma pequena alteração no *Gateway* para que o acesso se dê automaticamente. Pelo manual do site iDigi, é necessário fazer acesso ao menu de configuração conforme descrito anteriormente. No menu *Configuration*, submenu *iDigi*, faz-se necessária a inclusão do site do servidor que irá fazer a conexão da “nuvem” do fabricante. Basta alterar o campo *iDigi Server Address* para *developer.idigi.com*, clicar no botão do rodapé da página e reiniciar o equipamento para que a nova configuração entre em uso.

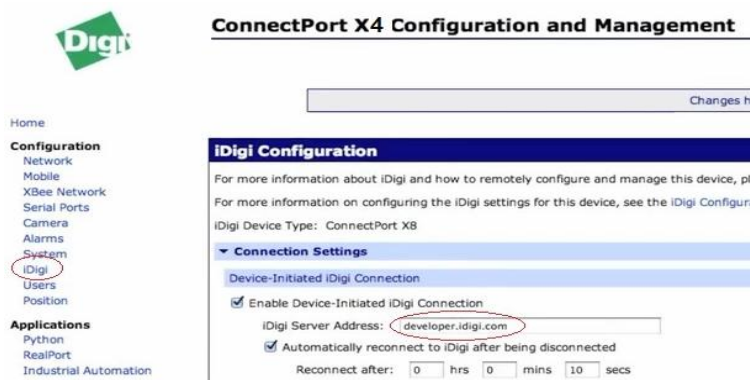


Figura 47 - Configuração do servidor do ambiente iDigi

Fonte: Autor

Percebe-se que é importante ativar a opção de reconexão automática conforme a figura 47 para que o serviço seja reiniciado sem a intervenção do operador, caso ocorra uma queda. A partir deste ponto, uma vez estando já autenticado no site da nuvem, é o momento de fazer a inserção do dispositivo no sistema *iDigi*, para que isto ocorra será necessário ir até a aba *iDigi Manager Pro - Devices* e inserir o dispositivo concentrador em uso. Para realizar este procedimento utiliza-se um botão de adição que levará para a tela de inserção, onde pode-se descobrir automaticamente através do botão *Discovery*, onde o *Gateway* deverá aparecer com seu nome, endereço MAC e endereço *IP* automaticamente, ou pode ser inserido manualmente através do botão *add manually* com o preenchimento manual do endereço MAC do equipamento, caso este não esteja ligado ou presente no ato da configuração. Caso todos os procedimentos tenham sido feitos corretamente, a rede esteja ativa e o equipamento em funcionamento, a tela deve ser similar à figura 48. Caso exista algum problema, deve aparecer da mesma forma, mas com o *status* de “*disconnected*”, e uma intervenção será necessária por parte do operador.

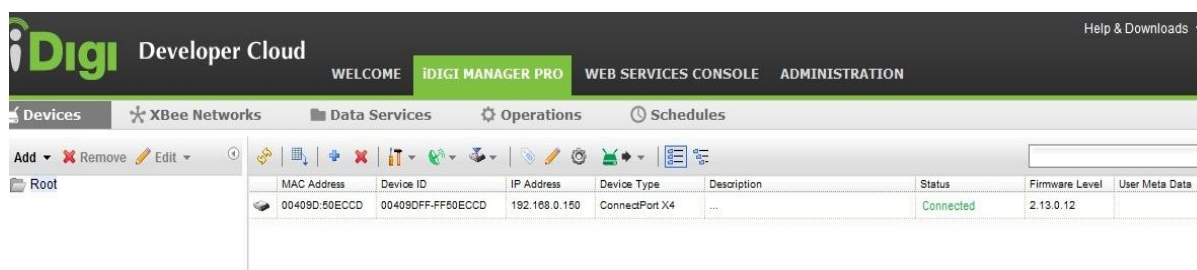
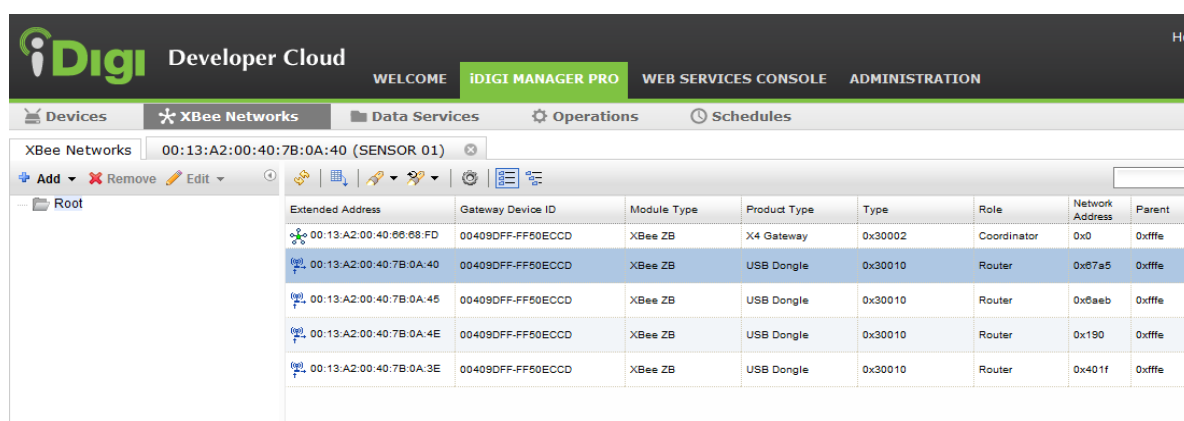


Figura 48 - Tela do *ConnectPort X4* disponível na nuvem *iDigi*

Fonte: Autor

Uma vez conectado na nuvem, teoricamente tem-se acesso ao concentrador *X4* e aos seus dispositivos interconectados, de maneira que o acesso ao sinal dos sensores pode ser visto na figura 48, percebe-se que a rede *XBee* está configurada para apenas um coordenador e os demais funcionam como roteadores, permitindo que se faça o encaminhamento dos pacotes por qualquer rota disponível de acordo com a funcionalidade *mesh* atuando na rede, para que esta condição seja disponível basta acessar a aba *iDigi Manager Pro – XBee Networks* conforme a figura 49:



Extended Address	Gateway Device ID	Module Type	Product Type	Type	Role	Network Address	Parent
00:13:A2:00:40:66:68:FD	00409DFF-FF50ECCD	XBee ZB	X4 Gateway	0x30002	Coordinator	0x0	0xffff
00:13:A2:00:40:7B:0A:40	00409DFF-FF50ECCD	XBee ZB	USB Dongle	0x30010	Router	0x67a5	0xffff
00:13:A2:00:40:7B:0A:45	00409DFF-FF50ECCD	XBee ZB	USB Dongle	0x30010	Router	0x6aeb	0xffff
00:13:A2:00:40:7B:0A:4E	00409DFF-FF50ECCD	XBee ZB	USB Dongle	0x30010	Router	0x190	0xffff
00:13:A2:00:40:7B:0A:3E	00409DFF-FF50ECCD	XBee ZB	USB Dongle	0x30010	Router	0x401f	0xffff

Figura 49 - Tela de acesso aos dispositivos *XBee* conectados na rede.

Fonte: Autor

Esta fase encerra a configuração da rede dos dispositivos, clicando em qualquer um dos equipamentos listados é possível ter acesso a eles e inclusive realizar alterações remotas através da nuvem *iDigi*, disponível em qualquer lugar a qualquer hora. Esta funcionalidade de desenvolvimento torna fácil o monitoramento e configuração dos dispositivos, acesso a relatórios de uso e diversos recursos fornecidos pelo fabricante, caso a empresa necessite de um recurso global de monitoramento permanente dos seus dispositivos. Obviamente esta condição torna o sistema dependente de um ambiente proprietário e sujeito a pagamento de uso dos serviços, de maneira que seria necessário o desenvolvimento de aplicativos específicos para obter uma utilização particular com este nível de interatividade.

O sistema do fabricante foi muito estável aos testes preliminares e não foi constatada nenhuma indisponibilidade de acesso. O escopo, porém, do projeto, é a interconexão do sistema de monitoramento de modo local e este recurso de acesso remoto serviu para demonstrar a potencialidade de utilização do sistema em uma rede global.

11.3. IDE de desenvolvimento de programas em *Python*

A ferramenta *Digi Esp™ for Python* é uma suíte de serviços criada pelo fabricante, de acordo com o projeto *Eclipse*, e também uma importante ferramenta para a elaboração dos programas de monitoramento dos sensores *XBee* que transmitem o sinal dos sensores instalados na pista do aeroporto. Como foi descrito anteriormente, o *kit* de desenvolvimento já incorpora API's próprias de comunicação com a maioria dos dispositivos do fabricante, sendo do desenvolvedor a tarefa de adequar o uso particular ao código em linguagem de alto nível *Python*.

Segundo Leno “*Python* é uma linguagem de programação de altíssimo nível interpretada e interativa. É completamente orientada a objetos e possui tipagem dinâmica.” Foi idealizada por Guido Van Rossum no Instituto de Pesquisa Nacional para Matemática e Ciência da Computação (CWI), localizado nos Países Baixos.

Por definição, existem basicamente as linguagens interpretadas e as compiladas, para *Bastos*, a diferença é um tanto controversa, mas se baseia no princípio de que, linguagens de programação como o C e C++ são compiladas, de forma que os códigos-fonte são convertidos estaticamente em linguagem de máquina que podem ser interpretados mais rapidamente e com menor custo no sentido de recursos de sistema, porém exigem mais custo operacional do programador, o qual deve ter mais conhecimento e trabalho pra programar em nível mais baixo. Já as linguagens mais modernas como *Java*, *C#* e *Python* são trabalhadas em uma interface mais interativa e amigável com o programador, tornando mais fácil o desenvolvimento de códigos complexos. O custo desta facilidade é maior, obviamente, os códigos gerados são uma linguagem de máquina intermediária, a qual necessita ter uma máquina virtual rodando que fará a interpretação desta linguagem para a de máquina de forma dinâmica. Apesar de exigir um poder de processamento muito maior em relação às linguagens interpretadas, o grande poder de processamento dos computadores atuais e o trabalho constante de reestruturação das máquinas virtuais compiladoras, no sentido de deixar o código mais enxuto, tem tornado o uso destas linguagens uma tendência cada vez mais forte.

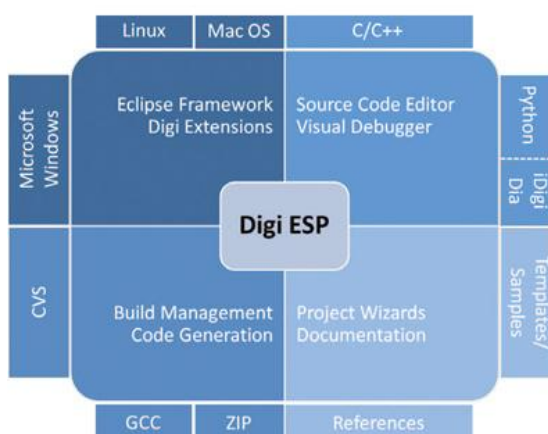


Figura 50 - Plataforma multifuncional *Digi Esp*
 Fonte: Digi (2012)

A *Digi Inc.* fornece soluções para *Windows*, *Mac OS* que é o *Digi Esp™ for Python*, *Linux*, o *Digi Esp™ for Digi Embedded Linux* e *NET+OS*, o *Digi Esp™ for NET+OS*. O primeiro em linguagem *Python* e os dois seguintes em linguagem *C* e *C++*, de forma que cabe ao desenvolvedor escolher a plataforma que deseja trabalhar, ou de acordo com o suporte a hardware disponível na versão disponibilizada do programa. Para o projeto foi escolhido o sistema *Windows* por ser a única plataforma com disponibilidade de suporte ao *Gateway ConnectPort X4* de acordo com o link de especificações:

<http://www.digi.com/products/wireless-wired-embedded-solutions/software-microprocessors-accessories/software/digiesp#specs>.

Conforme o fabricante, esta IDE (*Integrated Development Environment*) abrange a maioria dos ambientes de desenvolvimento como mostra a figura 50. O conjunto de funcionalidades abrange diversos componentes de trabalho, a destacar na versão *Windows*:

- Editor de código-fonte: Código fonte em *Python*, sintaxe codificada em cores por sintaxe, assistente inteligente de código, auto-identação e suporte a procura, assistência proativa e complementação do código, inserção de código modelo dinamicamente e API de ajuda de contexto sensível (ajuda completa);

- Editor visual e ágil para iDigi Dia: Editor de código fonte com codificação de sintaxe em cores, iluminado, verificação de erros de sintaxe e auto-identação, definição de elemento iDigi Dia (dispositivos, *loggers*, apresentações e serviços), configuração de elemento iDigi Dia, detecção de erros (padrões, faixas, etc.);

- Depuração: Depurador visual de código fonte, pontos de quebra condicionais, inspeção/modificação de dados (variáveis, expressões, etc.), depuração de multicaminhos e depuração remota (dispositivos).
- Interpretador *Python*: Suporte a *Python* 2.4 e 2.6;
- Gerenciamento de projeto e construção: projetos intuitivos *Python* e iDigi Dia, navegador de arquivo de projeto integrado, construção e reconstrução de projetos, suporte a sistema de controle de versão CVS (Sistema de Versões Concorrentes);
- Visualização em terminal: Serial, Telnet, SSH;
- Suporte a produtos Digi: *ConnectCore™* 3G, *ConnectCore™* Wi-9P 9215, *ConnectPort™* X2 / X3 / X4 / X5 / X8;
- Sistemas operacionais suportados: *Microsoft Windows XP*, *Microsoft Windows Vista*, *Microsoft Windows 7* e *Mac OS 10.6* (ou mais recente).

O *Digi Esp™ for Python* pode ser obtido diretamente no site do fabricante com o registro da empresa compradora do *kit*, para o experimento foi utilizada a versão 1.4.0. A instalação do programa dá-se de maneira intuitiva como a maioria dos programas desenvolvidos para *Windows*, após ter instalado o interpretador *Python* no computador (no projeto foi instalada a versão 2.6), basta iniciar o instalador executável da IDE para proceder a uma instalação padrão.

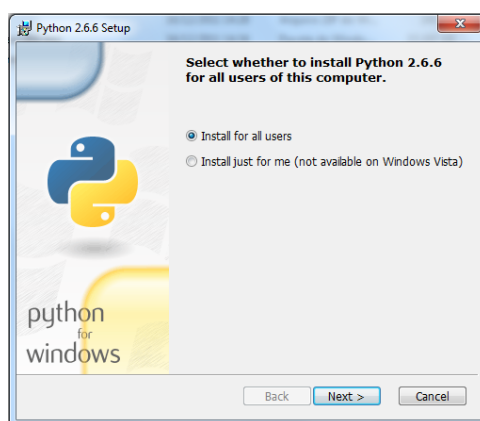


Figura 51 - Interpretador *Python*
Fonte: Autor



Figura 52 - Interpretador *Python*

Fonte: Autor

Uma vez instalada a IDE, parte do desenvolvedor buscar os meios para o seu ambiente originar a busca das informações dos equipamentos, o fabricante fornece diversos *links* com exemplos sobre aquisição de dados, como o *Smart Plug Interactive Demo*, de vital importância no projeto. O programa simplificado constante no apêndice A contém o sistema básico para a aquisição dos sensores *XBee* em uso, sendo interrogados em ciclos de intervalo reduzido. O software de aquisição foi elaborado também de acordo com exemplos de Malmsten (2010), e de documentação *Python* como o a função *struct*.

O programa faz a importação das bibliotecas, destacando a *zigbee*, que é específica para a manipulação com os pacotes que são recebidos dos sensores, a função principal rodará por *n* vezes especificadas na variável *runs*, com foi dito anteriormente este projeto tem por objetivo determinar o funcionamento dos sensores recebidos pela rede e elaborar o fluxograma de serviço que servirá de base para a implementação do programa de controle pelo solicitante. Desta forma, este programa detecta o estado de cada sensor configurado (até 65.535 sensores teoricamente) em tantos ciclos de programa. Ao executar o programa, abre-se um console que mostra o acesso via Telnet (porta 23) o acesso ao *X4*, se ocorrer alguma interrupção na comunicação o programa para e retorna o código de erro.

Após ser adquirido, o dado bruto é direcionado para uma função de *parse*, que irá classificar e separar os dados de interesse do sensor percebe-se que a função trata de tamanhos diferentes de arquivo sem perder a posição correta, a princípio este artifício trata de versões diferentes de equipamentos com a mesma lógica. Após o retorno é feita a impressão em tela do dado classificado. Com os dados classificados é possível ao operador visualizar a situação ou utilizar as variáveis para o programa de controle.

12. SENSORES DE PRESENÇA BOSCH

12.1. Montagem dos sensores de presença:

Esta talvez esta seja a peça mais delicada do projeto, pois como pôde ser visto anteriormente, cada tipo de sensor tem as suas especificidades e são eficazes para algumas situações, mas não para outras. O ambiente externo e agressivo do aeroporto, sujeito a chuva, maresia, sujeira, vento e demais fatores, dificultam muito a instalação de um dispositivo que atenda bem aos requisitos com tantas variáveis. Outro fator importante a destacar, é que talvez sejam necessários diversos ajustes no local até que seja disposto um patamar de confiabilidade, nos testes em laboratório o funcionamento foi perfeito, já nos testes *outdoor* foram necessários alguns ajustes até que fosse atingido o objetivo.

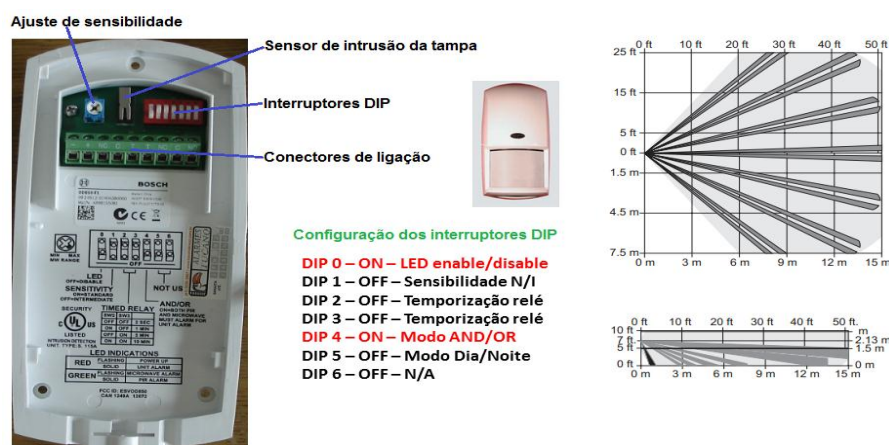


Figura 53 - Sensor de presença externo Bosch OD 850
Fonte: Montagem do catálogo de sensores do fabricante

O sensor selecionado foi o modelo OD 850 da Bosch, que atua por meio de Infravermelho e/ou micro-ondas, de maneira configurável, e possui as seguintes características:

- Atua com 42 Zonas de *Fresnel*, proporcionando maior versatilidade;
- Projetado para uso externo, portanto com capa protetora apropriada;
- Faixas de frequências de atuação em 10,525 GHz ou 10,588 GHz;
- Detector de intrusão na tampa;
- Seleção de modo dia/noite;
- Modo de trabalho dos sensores *AND/OR*;
- Indicador visual (*led*) de presença (configurável);

- Imune a insetos/corrente de ar;
- Auto teste por meio de micro-ondas;
- Compensação de temperatura;
- Tensão de operação de 10 a 15 V e corrente de 22 mA;
- Temperatura de operação de -40 a 55°C.

12.2. Detalhes da montagem elétrica/eletrônica dos sensores:

Basicamente, os sensores foram montados em três módulos distintos e isolados mecanicamente entre si: O módulo da fonte de alimentação geral, o módulo de Radiofrequência com o *XBee* interno e o módulo sensor de presença propriamente dito. A Figura 54 ilustra a interligação dos módulos:

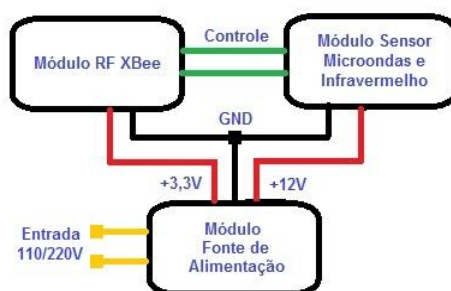


Figura 54 - Módulos internos dos sensores de presença

Fonte: Autor

- Módulo da fonte de alimentação: tem a finalidade de converter a tensão alternada de 110/220V da rede elétrica em tensões de saída de 3,3V e 12V contínuas, que estão alimentando os dispositivos elétricos e eletrônicos dos módulos sensor e de RF adequadamente. A Figura 55 ilustra o diagrama interno da fonte.

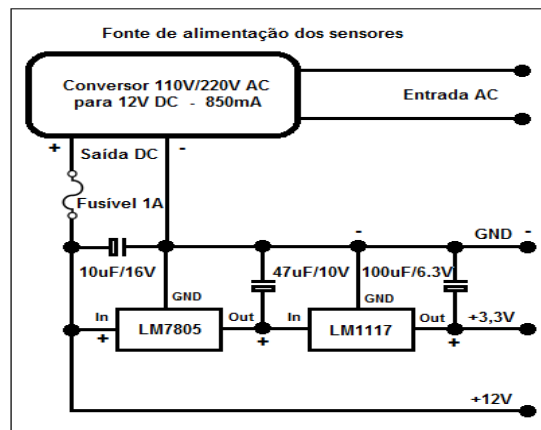


Figura 55 - Módulo Fonte de Alimentação
Fonte: Autor

- Módulo de Radiofrequência: Possui a finalidade de realizar a comunicação entre o sensor de presença e o *Gateway X4* através do módulo *XBee* configurado e ligado, fornecendo a presença ao concentrador e repassando o *status* em tempo real para que seja detectado remotamente. A ilustração do detalhamento interno do módulo pode ser visto na figura 56:

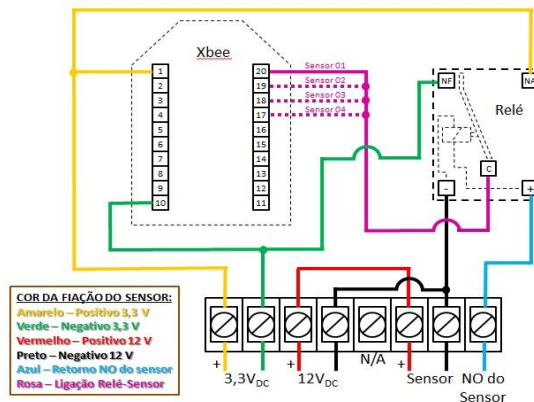


Figura 56 - Módulo de Radiofrequência (comunicação de dados)
Fonte: Autor

- Módulo de Sensor de Presença: É o elemento do sistema que realiza a detecção da presença da aeronave (ou objeto interferente) pela dupla tecnologia Micro-ondas/Infravermelho. A ilustração do detalhamento interno do módulo pode ser visto na figura 57:

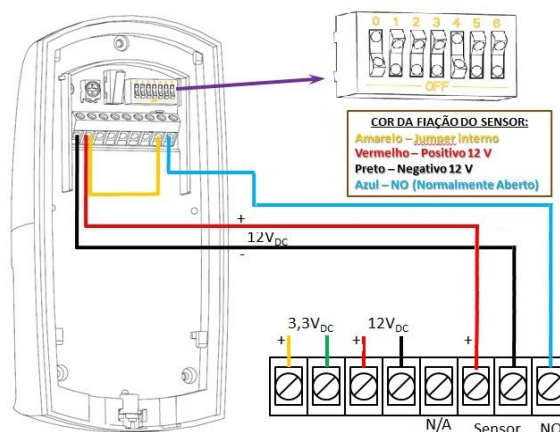


Figura 57 - Módulo de Sensor de Presença
 Fonte: Autor

12.3. Detalhes da montagem mecânica dos sensores:

Esta montagem exigiu mais habilidade manual, onde foi necessária a adaptação de algumas peças para que o conjunto fosse compacto, resistente, frangível e esteticamente apresentável. Para a proteção e suporte foi utilizada uma arandela de iluminação industrial para aplicação em ambientes hostis, todo em alumínio (resistente à corrosão) com uma grade de proteção robusta, da qual foi retirado o suporte da lâmpada e adaptado o fundo para fixar o suporte do sensor. No topo também foi inserida uma meia esfera protetora de aço inox reflexiva contra aproximação de aves e prováveis tempestades de granizo sobre o sensor, também reduzindo a ação da chuva. Aproveitando o suporte externo com sobra de espaço, foi também parafusada a antena de 3 dBi de ganho, que é ligada ao *ZigBee* interno. A figura 58 ilustra o suporte montado sem o sensor.

A montagem levou em conta o ambiente externo e assim toda a vedação possível foi providenciada, com anéis de vedação de borracha e cola de silicone onde houvesse a possibilidade de infiltração de umidade e maresia. Na parte interna, os sensores e a fonte de alimentação foram montados em unidades separadas para facilitar a manutenção e proporcionar ainda mais proteção aos componentes, onde pode-se observar na figura 58 que o sensor *XBee* foi soldado diretamente ao relé de chaveamento e ao conector de alimentação e este imerso em resina para que não ocorram problemas de mau contato entre as peças. A imagem mostra também que foram confeccionadas cinco unidades iguais, sendo quatro para o uso imediato e uma de reserva:

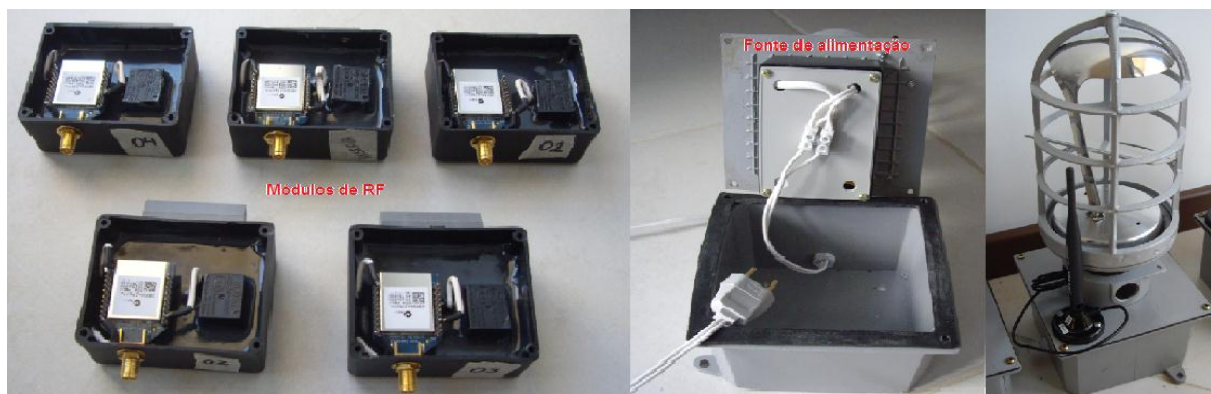


Figura 58 - Módulo de RF dos sensores e fonte de alimentação
Fonte: Autor

A última etapa de montagem e a versão final podem ser visualizadas na figura 59, com o sensor devidamente instalado e todos os módulos ligados:



Figura 59 - Módulo de RF dos sensores e unidades prontas para uso
Fonte: Autor

13. TOPOLOGIA FINAL DA REDE

No planejamento da distribuição dos elementos da rede, estes estão dispostos dentro de um raio de ação relativamente curto, estimado em não mais que 400 metros entre o *Gateway* e o sensor de decolagem, a figura 60 ilustra a topologia:

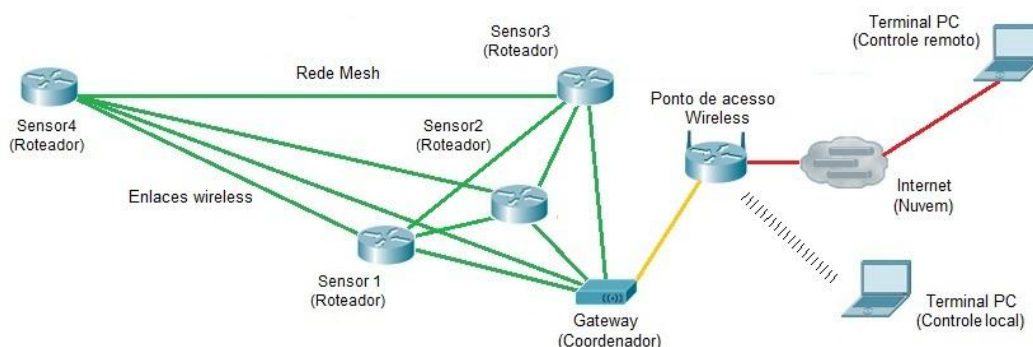


Figura 60 - Topologia de rede do prevista para o aeroporto
Fonte: Autor

Como pode ser visto, os módulos *XBee* estão configurados para serem roteadores. Para o projeto previsto poderiam ser programados como dispositivos finais, porém dada a relevância da segurança do local, a opção de roteamento permite entre outras, as vantagens:

- Roteamento alternativo por rede em malha (*mesh*), uma função importantíssima dos dispositivos *ZigBee*, permitindo que os pacotes sejam desviados por rotas alternativas caso a rota principal seja obstruída, possibilitando assim *backups* de percurso dinâmicos.

- Em malha, encurtam-se os caminhos dos enlaces, pela regeneração de sinal, assim não é necessário o uso de uma potência elevada nos transmissores, o que permite menor contribuição no aumento da poluição eletromagnética no ambiente do aeroporto, que tem por característica própria a interferência oriunda da comunicação de controle das aeronaves e a também dos equipamentos próximos como redes padrão 802.11 e outros que utilizam a mesma faixa de frequência e outros derivados da condição de localização urbana.

Além destas vantagens, o uso de sensores com comunicação *wireless* permite que os conjuntos sejam deslocados e ajustados de acordo com a disponibilidade local, apenas necessitando de um ponto de alimentação, o que pode proporcionar o reaproveitamento e expansão, caso ocorram mudanças no ambiente.

13.1. Teste prático em área livre

De posse de todo o *hardware* e *software* prontos e de testes preliminares positivos em bancada, foi necessário colocar à prova o trabalho desenvolvido. Como não existe localmente a possibilidade de testar com o objeto final (aeronaves), obtemos uma licença para realizar o teste preliminar no Parque de Exposições da UFSM, o qual não possui um uso específico fora dos eventos como feiras e possui uma pista de asfalto plana e com razoável espaço livre. Os sensores foram dispostos inicialmente lado a lado para detecção comparativa e depois de forma diagonal na tentativa de simular o circuito da aeronave. Foram testadas as alternativas do fluxograma e o sistema correspondeu ao esperado.

Comprovadamente, obstáculos interferem diretamente na detecção quando estão em movimento, por exemplo, pequenos ramos agitados pelo vento tendem a atrapalhar o sensor, o que não deverá ocorrer no aeroporto, onde não existe nada acima do nível da pista. As fotos a seguir ilustram o teste em campo:

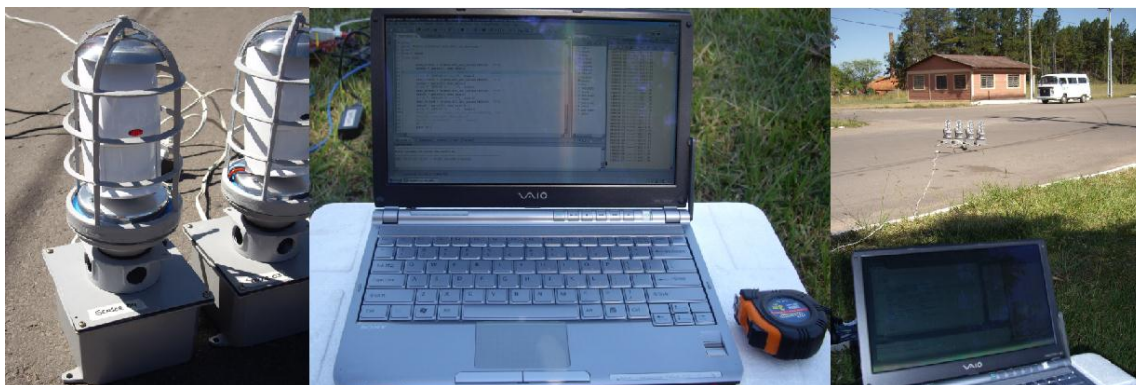


Figura 61 - Testes em ambiente *outdoor*

Fonte: Autor



Figura 62 - Testes em ambiente *outdoor*

Fonte: Autor

O equipamento foi montado em modo local com a disposição paralela inicial, para obter a aquisição de todos os sensores simultaneamente (pode-se observar na foto o *led* indicador do sensor aceso), nesta etapa é testada a passagem do objeto em distâncias e velocidades diferentes. O raio de atuação na pista foi calibrado para sete metros, que contempla até o outro lado da pista, logo após foram feitas cinco tomadas de passagem a velocidades diferentes, iniciando a 20 km/h até chegar a 100 km/h. Vale ressaltar que o ambiente foi liberado para tal teste e o tráfego é bloqueado durante este período. A temperatura ambiente esteve acima de 30 graus celsius, a uma umidade relativa de 40%, o que submeteu o equipamento ao teste de temperatura relativamente agressivo, o que colaborou para aumentar a dificuldade do teste.

13.2. Aquisição de dados e análise dos resultados

O experimento proporcionou a aquisição de dados em variáveis, as quais podem ser utilizadas no futuro, através de um programa que poderá permitir o controle automático efetivo da barreira que sinaliza e interrompe o trânsito próximo da cabeceira da pista. A figura 63 mostra a tela de aquisição do *Digi Esp for Python*.

Como pode ser visualizado na tela capturada, o programa faz a conexão e funciona utilizando acesso via Telnet na porta 23 do servidor (em particular, pelo endereço 192.168.1.150:23 neste caso). Para que isto ocorra, o programa precisa ser configurado na aba *Device Options / Device Manager*, na janela que se abre do *Device Manager*, na aba *Lan Connections* é colocado o endereço IP do *Gateway*.

A suíte de programas possui nativamente diversos recursos, a destacar a janela do programa de aquisição em *Python* propriamente dito, para a obtenção dos dados, e o console onde é visualizado o andamento do *software* em execução. Para iniciar a operação, é necessária realizar a chamada do programa no servidor pela aba *Run / Run As / Remote Digi Python Application* e aguardar até que todos os módulos sejam carregados, caso algum erro ocorra o programa irá retornar pela console o ponto de conflito e finalizar a execução. Vale salientar aqui que o *Python* é muito sensível à indentação, ou seja, quando algum comentário ou formato estiver fora do padrão, irá trancar o console, bem como falhas de identificação dos sensores também impedirão o funcionamento, exigindo a intervenção do operador.

The screenshot shows the DigiPython IDE interface. The main editor displays a Python script named `Simplificado.py` with the following code:

```

74     numerador += 1
75     print repr(dados1), "  SENSOR 01 ----->", dados1
76     print " "
77     print "Estado do SENSOR 01 ----->", dados1
78     print " "
79
80     dado_bruto2 = zigbee.ddo_get_param(SENSOR2, 'IS')
81     dados2 = parseIS( dado_bruto2 )
82     print "Estado do SENSOR ----->", dados2
83
84     dado_bruto3 = zigbee.ddo_get_param(SENSOR3, 'IS')
85     dados3 = parseIS( dado_bruto3 )
86     print "Estado do SENSOR 03 ----->", dados3

```

The left sidebar shows a project tree with folders like `icado`, `tings`, `ld`, and `ources`, and files like `Simplificado_mod.`, `Simplificado.py`, `ject`, and `devproject`.

The bottom console window shows the output of the script, indicating a connection to a Telnet server at 192.168.1.150:23. The output includes sensor data and status messages:

```

419
Dados brutos =
Sets = 1
Datamask = 0
Analogmask = 1
Data[4:] =
Estado do SENSOR 01 ----->{'AI0': 0}
Dados brutos =
Sets = 1
Datamask = 0
Analogmask = 1
Data[4:] =
Estado do SENSOR 03 ----->{'AI0': 528}
420
Dados brutos = y
Sets = 1
Datamask = 0
Analogmask = 1
Data[4:] = y
Estado do SENSOR 01 ----->{'AI0': 1023}
Dados brutos =

```

Figura 63 - Tela de aquisição dos sensores

Fonte: Autor

Em relação aos dados, a figura 63 mostra um estado provocado intencionalmente durante os testes: um erro por falha de contato. A intenção inicial do projeto era configurar as portas de sensoriamento para modo digital, mas foram alteradas para modo analógico, pois mesmo sendo capazes de cumprir a função principal, que é a de detectar o acionamento dos sensores remotos, o modo digital fornece menos informação com o mesmo custo de *hardware*. Como a precisão de 10 bits do conversor ADC (Analógico/Digital) proporciona um fator de tolerância que pode ser configurado, pode ser interessante adicionar uma segurança extra no programa, um controle de status de erro de sensor por valor de escala, assim exemplificado:

O sensor 01 da figura 63 tem, neste intervalo, uma variação de 0_d para 1023_d , ou para o valor de referência digital como desligado e ligado (0 ou 1 em binário – desprezando o tamanho da palavra). Assim, de maneira premeditada, foi desconectado o fio do sensor 03 do contato AI0 (*Analogic Input 0*) e o valor retornado foi 528_d . Esta variação se deve ao estado intermediário de tensão no pino 20 do *XBee*, por ruído induzido na entrada aberta. Circuitos digitais necessitam de limiares de estado bem definidos, isto significa que um valor similar a

esse, por exemplo, originado com o mau contato no pino 20 por fatores diversos (oxidação nos contatos, condutor quebrado, solda fria, etc.), quando configurado para porta digital, poderia tender a indicar um estado de desligado ou ligado, de forma aleatória, podendo causar problemas até que fosse detectada a falha.

A interface do *ZigBee* utilizado é alimentada por +3,3V, o que é uma tensão baixa, logo a variação por passos é de $3,3/1024$ volts por unidade de escala, resulta em aproximadamente em 0,003223V por passo. Pode-se assim estimar que temos uma tensão de 1,701563V, se multiplicarmos os 528 passos pela tensão unitária. Este erro pode ser detectado imediatamente, por exemplo, se programarmos o limiar de 0V a 0,5V (0_d a 155_d) como estado desligado e 2,8V a 3,3V (869_d a 1023_d) como estado ligado, sendo que qualquer valor entre estes limites, poderia acusar um erro na console e proporcionar a rápida detecção e correção. Caso fosse utilizando uma porta de entrada digital no módulo *XBee*, mesmo que o funcionamento normal fosse o esperado, uma condição de falha como esta dificilmente seria percebida até que um problema operacional fosse visível.

No apêndice B estão dispostos os dados coletados em uma situação de simulação de decolagem de uma aeronave, realizada com a utilização de um automóvel que serviu de objeto a ser detectado, com mais dificuldade, visto que tem uma massa razoavelmente menor que a de uma aeronave. Nas sequências destacadas, o acionamento dos sensores 1 e 2 acontece simultaneamente, conforme o regime de redundância previsto no projeto. Percebe-se que o sensor tem um tempo curto onde mantém o estado de ligado, por esse motivo não temos apenas um ciclo de atividade alta, pois alguns ciclos do programa captam o valor 1023_d ; ao desligar o contato o estado volta para 0_d , esse comportamento se deve ao tempo de permanência do sensor Bosch em estado alto antes de voltar ao estado original. O programa final deverá contar com temporizadores para tratar do atraso necessário, para que não ocorra erro de interpretação por repetição de dados. Nesse estágio a aeronave estaria na condição de espera para entrar na pista.

Na sequência ocorrem os estágios do sensor 3 e do sensor 4 de forma ordenada, simulando a presença na cabeceira da pista e após passando pelo sensor em direção à decolagem.

O gráfico 01, realizado no software de monitoramento e aquisição *Wireshark*, mostra o comportamento da comunicação entre o *Gateway* e o cliente, em relação ao tráfego total capturado pela interface do computador de aquisição, nota-se nitidamente a característica: a comunicação inicia com poucos dados durante o estabelecimento da conexão Telnet, a partir deste ponto o fluxo de dados mantém-se constante até o final da conexão, quando a porta do servidor e a porta do cliente são liberadas. Este gráfico é apenas para melhor entendimento visual e a escala deve ser desconsiderada, sendo melhor explicado a seguir.

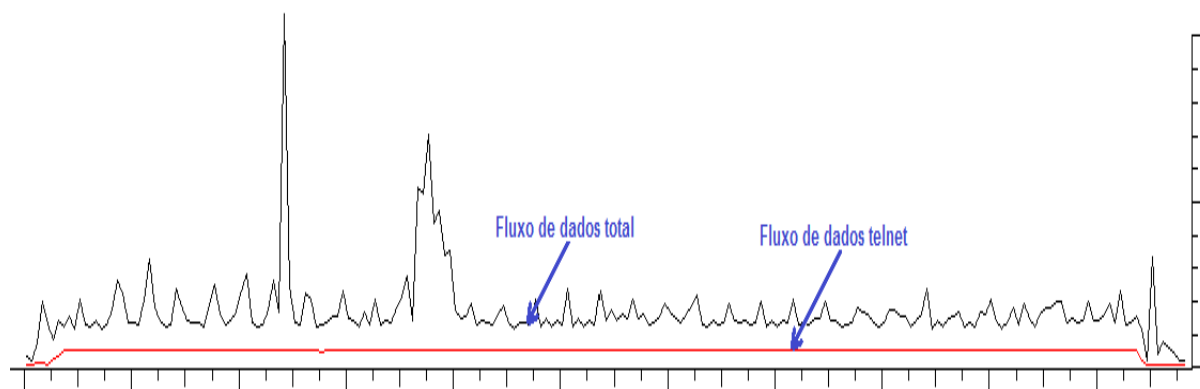


Gráfico 01 - Gráfico do fluxo geral x Telnet entre cliente e servidor

Fonte: Autor

Através de uma análise dos pacotes é possível ver que a porta alta do cliente de maior tráfego foi a 56.200, por onde a conexão manteve-se na maioria do tempo do teste, outras portas foram utilizadas, mas por curtos períodos de tempo até que se estabelecesse o diálogo entre as partes de fato ou na finalização, conforme a figura 64.

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A-B	Bytes A-B	Packets A-B	Bytes A-B	Rel Start	Duration	bps A-B
192.168.1.100	56171	192.168.1.150	telnet	10	595	5	290	5	305	0,000000000	0,1010	22978,03
192.168.1.100	56175	192.168.1.150	telnet	10	600	5	295	5	305	7,132779000	0,1018	23171,56
192.168.1.100	56199	192.168.1.150	telnet	9	530	4	230	5	300	29,926334000	0,0580	31749,87
192.168.1.100	56080	192.168.1.150	telnet	1	59	1	59	0	0	30,011365000	0,0000	N/A
192.168.1.100	56200	192.168.1.150	telnet	9 929	5 264 378	4	255	9 925	5 264 123	30,139430000	2028,6917	1,01

Figura 64 - Resumo do fluxo geral x Telnet entre cliente e servidor

Fonte: Autor

O tráfego entre cliente e servidor apresenta uma característica bem linear, e o gráfico 02 é uma prova evidente deste formato. Cabe aqui ressaltar que este comportamento reflete apenas a comunicação na interface do computador e não a atividade dos transceptores, a qual não é visível na interface do usuário. A seguir foi realizada uma busca mais refinada do comportamento, de modo que a figura 64 ilustra o gráfico de uma amostragem realizada obtendo o tráfego geral da interface, tendo logo abaixo o fluxo total de *download* com filtro Telnet e abaixo desta o fluxo de dados de *upload*.

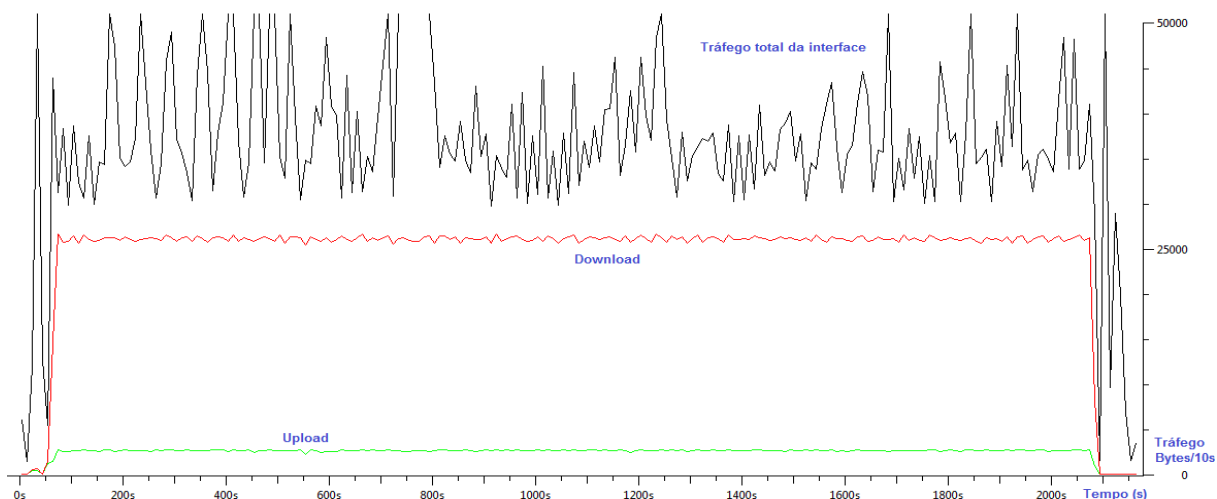


Gráfico 02 - Gráfico do fluxo geral x *download* x *upload* entre cliente e servidor

Fonte: Autor

A análise do gráfico 02 demonstra o comportamento linear do fluxo de dados, tanto de envio como recebimento, porém a figura 65, que demonstra a tela de filtragem dos pacotes de *upload* apresenta um detalhe interessante: apesar de aparecer o protocolo como *TCP* nos pacotes enviados para o *Gateway*, a própria informação a seguir já revela que estes fazem parte da conexão Telnet estabelecida. Este comportamento sugere realmente que os pacotes enviados são de confirmação *ACK* que a conexão *TCP* utiliza, sendo esta a principal comunicação de retorno durante toda a conexão.

No.	Time	Source	Destination	Protocol	Length	Data	Info
29318	2012-05-26 11:53:59.209624	192.168.1.150	192.168.1.100	TELNET	525	Dados brutos = \001,Sets	=Telnet Data ...
29319	2012-05-26 11:53:59.408890	192.168.1.100	192.168.1.150	TCP	54		56200 > telnet [ACK]
29320	2012-05-26 11:53:59.409624	192.168.1.150	192.168.1.100	TELNET	529	Dados brutos = \001,Sets	=Telnet Data ...
29323	2012-05-26 11:53:59.608912	192.168.1.100	192.168.1.150	TCP	54		56200 > telnet [ACK]
29324	2012-05-26 11:53:59.610869	192.168.1.150	192.168.1.100	TELNET	525	Dados brutos = \001,Sets	=Telnet Data ...
29325	2012-05-26 11:53:59.810922	192.168.1.100	192.168.1.150	TCP	54		56200 > telnet [ACK]
29326	2012-05-26 11:53:59.811694	192.168.1.150	192.168.1.100	TELNET	525	Dados brutos = \001,Sets	=Telnet Data ...
29327	2012-05-26 11:54:00.010947	192.168.1.100	192.168.1.150	TCP	54		56200 > telnet [ACK]
29328	2012-05-26 11:54:00.012887	192.168.1.150	192.168.1.100	TELNET	525	Dados brutos = \001,Sets	=Telnet Data ...
29330	2012-05-26 11:54:00.212955	192.168.1.100	192.168.1.150	TCP	54		56200 > telnet [ACK]
29331	2012-05-26 11:54:00.213986	192.168.1.150	192.168.1.100	TELNET	527	Dados brutos = \001,Sets	=Telnet Data ...
29332	2012-05-26 11:54:00.413972	192.168.1.100	192.168.1.150	TCP	54		56200 > telnet [ACK]
29333	2012-05-26 11:54:00.415916	192.168.1.150	192.168.1.100	TELNET	525	Dados brutos = \001,Sets	=Telnet Data ...
29334	2012-05-26 11:54:00.713969	192.168.1.100	192.168.1.150	TCP	54		56200 > telnet [ACK]
29335	2012-05-26 11:54:00.715925	192.168.1.150	192.168.1.100	TELNET	871	Dados brutos = \001,Sets	=Telnet Data ...

Figura 65 - Filtro de separação do fluxo de *upload* para o *Gateway*

Fonte: Autor

Também é possível perceber que estatisticamente o comprimento dos pacotes manteve-se em torno 320 e 639, chegando a 96,43% da amostra (figura 65), de modo que é possível manter uma previsão do consumo de banda na comunicação. É importante lembrar que o tráfego foi avaliado sob condições isoladas, e o sistema pode aceitar mais conexões pela rede aumentando assim a carga geral, porém a baixa velocidade de transferência dos dados

pela interface, configuradas para 9600 kbps, não deve proporcionar uma alteração tão significativa de modo geral no volume de pacotes transferidos.

Topic / Item	Count	Rate (ms)	Percent
[-] Packet Lengths	9959	0,004837	
0-19	0	0,000000	0,00%
20-39	0	0,000000	0,00%
40-79	42	0,000020	0,42%
80-159	1	0,000000	0,01%
160-319	1	0,000000	0,01%
320-639	9603	0,004664	96,43%
640-1279	312	0,000152	3,13%

Figura 66 - Estatística de comprimento dos pacotes de dados
Fonte: Autor

As figuras 67 e 68, que são originadas de uma amostra capturada com o programa *Wireshark* em dado momento, e fornecem uma vasta gama de dados para que se possa entender analiticamente o cenário em que se passa na comunicação entre os equipamentos. Como foi dito, não temos recurso neste experimento para atuar diretamente no transporte de dados entre os enlaces de rádio com os equipamentos do projeto, mas mesmo assim é possível obter através do *Gateway* a informação necessária, especialmente pelo formato aberto dos dados transportados via Telnet.

As imagens mostram no momento capturado, exatamente quando o cliente solicita a abertura do programa em *Python* que está gravado na memória secundária do *Gateway X4*, fazendo que este seja executado de forma remota. O *Wireshark* apresenta claramente a organização dos dados em cada camada, iniciando pela Física, onde os frames são destacados especialmente quanto ao seu tamanho, percebe-se que ocorre a divisão em blocos de 8 bits e neste caso, onde a conexão está iniciando, o tamanho é reduzido, ao contrário da maioria dos blocos como demonstra a figura 68.

Frame 462: 72 bytes on wire (576 bits), 72 bytes captured (576 bits)
 Arrival Time: May 20, 2012 11:26:02.522460000 Hora oficial do Brasil
 Frame Number: 462
 Frame Length: 72 bytes (576 bits) Palavra de 8 bits
 Capture Length: 72 bytes (576 bits)
 [Protocols in frame: eth:ip:tcp:telnet]
 [Coloring Rule Name: TCP]
 [Coloring Rule String: tcp]

Ethernet II, Src: RealtekS_68:02:eb (00:e0:4c:68:02:eb), Dst: Digiboar_50:e7:a6 (00:40:9d:50:e7:a6)
 Destination: Digiboar_50:e7:a6 (00:40:9d:50:e7:a6)
 Source: RealtekS_68:02:eb (00:e0:4c:68:02:eb)

Internet Protocol Version 4, Src: 192.168.1.101 (192.168.1.101), Dst: 192.168.1.150 (192.168.1.150)
 Version: 4
 Header length: 20 bytes
 Flags: 0x02 (Don't Fragment)
 Time to live: 128
 Protocol: TCP (6)
 Header checksum: 0x4ac3 [correct]
 Source: 192.168.1.101 (192.168.1.101)
 Destination: 192.168.1.150 (192.168.1.150)

Transmission Control Protocol, Src Port: 49852 (49852), Dst Port: telnet (23), Seq: 22, Ack: 27, Len: 18
 Source port: 49852 (49852)
 Destination port: telnet (23)
 [Stream index: 64]
 Sequence number: 22 (relative sequence number)
 [Next sequence number: 40 (relative sequence number)]
 Acknowledgement number: 27 (relative ack number)
 Header length: 20 bytes
 Flags: 0x018 (PSH, ACK)
 window size value: 64214
 [Calculated window size: 64214]
 [window size scaling factor: -2 (no window scaling used)]
 Checksum: 0xcea7 [validation disabled]
 [SEQ/ACK analysis]

Telnet
 Data: python dpdsrv.py \n

0000 00 40 9d 50 e7 a6 00 e0 4c 68 02 eb 08 00 45 00 .@.P....Lh....E.
 0010 00 3a 2b af 40 00 80 06 4a c3 c0 a8 01 65 c0 a8 .:+.@...J....e..
 0020 01 96 c2 bc 00 17 30 cd 6e ae e9 dd b6 85 50 180..n.....P..
 0030 fa d6 ce a7 00 00 70 79 74 68 6f 6e 20 64 70 64python dpd
 0040 73 72 76 2e 70 79 20 0asrv.py .

Figura 67 - Momento de chamada do programa no *Gateway X4*

Fonte: Autor

No.	Time	Source	Destination	Protocol	Length	Data	Info
432	55.493852	192.168.1.150	192.168.1.101	TELNET	60	\n,#>	Telnet Data
433	55.552119	192.168.1.101	192.168.1.150	TELNET	59	exit\r	Telnet Data
435	55.552582	192.168.1.150	192.168.1.101	TELNET	60		Telnet Data
438	55.578627	192.168.1.101	192.168.1.150	TELNET	59	exit\r	Telnet Data
445	55.655345	192.168.1.150	192.168.1.101	TELNET	60		Telnet Data
446	55.655936	192.168.1.101	192.168.1.150	TELNET	57		Telnet Data
447	55.656387	192.168.1.150	192.168.1.101	TELNET	60		Telnet Data
448	55.656450	192.168.1.101	192.168.1.150	TELNET	57		Telnet Data
449	55.660321	192.168.1.150	192.168.1.101	TELNET	60	\r	Telnet Data
450	55.660379	192.168.1.101	192.168.1.150	TELNET	69		Telnet Data
451	55.660938	192.168.1.150	192.168.1.101	TELNET	60	\n,#>	Telnet Data
453	55.865092	192.168.1.150	192.168.1.101	TELNET	66		Telnet Data
462	57.737138	192.168.1.101	192.168.1.150	TELNET	72	python dpdsrv.py \n	Telnet Data
463	57.757563	192.168.1.150	192.168.1.101	TELNET	60	p	Telnet Data
465	57.971139	192.168.1.150	192.168.1.101	TELNET	74	ython dpdsrv.py \r\n,\r\n	Telnet Data

Figura 68 - Momento de chamada do programa no *Gateway X4*

Fonte: Autor

Passando para a camada de Enlace, destaca-se a identificação dos endereços MAC das interfaces de rede envolvidas na conexão, onde é mostrado o modelo *Digiboar*, por exemplo, que é a interface do *X4*. Na camada seguinte, de Rede, são fornecidos os dados mais utilizados na análise humana das rotas, especialmente os endereços IP de origem e destino, bem como a versão 4 do protocolo em uso. Passamos para a camada de Transporte, onde são fornecidas as

portas do dispositivo de origem e também de destino da comunicação, a destacar que a porta de origem é a 49852, diferente da que transporta a maioria dos dados, a 56200 neste experimento, isto acontece porque o computador não tem uma porta de saída específica para saída e faz uso aleatório das portas altas para estabelecer conexão, já o servidor só aceita a entrada pela porta 23 desde que esta esteja configurada com o padrão do protocolo. A última camada, de aplicação, representada no relatório pela presença do protocolo Telnet, apresenta os dados do usuário na entrega; como foi comentado, na imagem pode ser visto que o terminal faz a solicitação do programa *dpdsrv.py* através do programa de interface desenvolvido em *Python* (Apêndice A), iniciando assim o acionamento do servidor de dados. Na sequência, caso o servidor seja iniciado e todos os parâmetros sejam atendidos, será estabelecida a conexão definitiva entre as partes e será possível a transferência dos dados.

Vale destacar aqui que, apesar do sistema apresentar a facilidade de comunicação interna entre clientes e servidor, o Telnet é um protocolo mais antigo e considerado de segurança fraca, pois todos os dados passam em texto aberto, inclusive senhas, podendo ser capturados por terceiros, motivo pelo qual tem sido substituído gradualmente por protocolos mais elaborados como o SSH (*Secure Shell*) onde os dados são criptografados antes de serem enviados. A solução da Digi leva em conta apenas a simplicidade e garantia da comunicação, porém o usuário deve garantir a segurança física, assim como implementar criptografia nos seus roteamentos, por exemplo, com o uso de tunelamento por VPN's (*Virtual Private Network*) ou fazer ainda o uso da nuvem que também implementa uma segurança mais robusta.

14. CONCLUSÕES

Promover o desenvolvimento de um projeto baseado em comunicação de redes de computadores, que atenda a todos os desejos das pessoas e organizações, sempre foi um desafio. Se no passado não muito distante, a exigência era muito menor, eram também complexos os meios chegar a razoáveis resultados que, de certa forma, eram muito bons se considerarmos as condições técnicas existentes.

A evolução tecnológica, especialmente nas últimas décadas, tem impulsionado o mundo para novos e diferentes caminhos, criando novos padrões de comportamento e uma necessidade crescente da troca de informações de forma rápida e segura. Para os novos desenvolvedores, o desafio também se faz presente, mas não poderia ser comparado aos primeiros trabalhos nesta área, pois os valores são específicos de cada época, a cada ano novos padrões são criados, produtos e serviços somam-se à vida e à rotina das pessoas, formando novas necessidades emergentes e somando demandas de comunicação cada vez maiores às redes.

O presente projeto foi idealizado e desenvolvido com a pretensão de tentar preencher uma pequena lacuna, desta a vasta área da comunicação de dados atual, aliando os conhecimentos adquiridos no curso, ao grande potencial tecnológico disponível e a grande biblioteca mundial relacionada existente. O maior privilégio, porém, foi a possibilidade de buscar uma solução que possa proporcionar maior segurança às pessoas, podendo preservar vidas, quando corretamente aplicada.

Dentro do escopo previsto no projeto, os objetivos foram atingidos, apesar do intervalo de tempo extremamente reduzido para as atividades. O trabalho conseguiu a determinação dos recursos mínimos necessários para a comunicação e monitoramento da movimentação de aeronaves nos procedimentos de decolagem e aterrissagem, bem como a implementação de fato desta comunicação em ambiente simulado de acordo com fluxograma elaborado durante este período (Fluxograma 1), assim sendo foram realizados testes *indoor* e *outdoor* e os resultados ficaram dentro do esperado pelo projeto. O aprendizado permitiu o contato direto com profissionais que atuam diretamente com modernas técnicas de trabalho, e a experiência permitiu uma visão mais ampla do mercado desta área de tecnologia. O projeto ainda mostrou ser viável a interação a nível mundial nas medições e controle, utilizando uma ampla gama de recursos existentes, desde a detecção dos eventos por sensores infravermelhos e de micro-ondas, passando por redes do padrão *ZigBee* e suas camadas, sendo decodificadas e

transportadas por redes IP locais e também ligadas ao um sistema de nuvem de dados, a qual permite a interligação global.

O objetivo de prover uma rede baseada em comunicação sem fio para atuar em um ambiente restrito foi contemplado, assim como o fluxograma desenvolvido permite que a rotina de funcionamento promova uma operação segura, desde que os padrões sejam obedecidos. Em relação ao software final de controle, apesar de não ter sido requisitado pela empresa I-Dutto, já é possível utilizar de forma manual o programa simplificado existente. Caso houvesse mais tempo disponível, possivelmente poderia ser desenvolvido, necessitando da aquisição de conhecimento adicional sobre o *Python*, com conceitos mais elaborados em relação à linguagem.

É importante destacar que a função de atuar como um profissional da área de redes é especialmente planejar e promover a comunicação entre os elementos destas. Esse objetivo foi cumprido, bem como permanecem preservados os papéis importantes das pessoas que atuarão, por exemplo, desenvolvendo o software de controle do sistema, na área do programador, a especificação de cabeamento, motorização e elementos mecânicos e elétricos de movimentação das barreiras, do engenheiro, e inclusive do “cochileiro” ou vigia da cancela, que manterá a nobre função de resguardar o seu posto, com mais segurança. O trabalho mostrou que o aluno do curso tem ainda carências que possivelmente possam ser supridas com disciplinas auxiliares, que permitam trabalhar com dados brutos e suas estruturas, bem como mais tempo dedicado à programação, inclusive com orientação a objetos, caso deseje ampliar a gama de pesquisas e desenvolvimento de software dedicado. Em relação à montagem dos sensores, talvez não fosse possível ser realizada sem um prévio conhecimento e experiência na área, dependendo de habilidade e algum curso extra em eletrônica aplicada, mesmo assim valem lembrar que essa função pode ser delegada a um profissional sem prejuízo final do trabalho.

A montagem dos equipamentos foi determinada e executada de forma a atender aos objetivos de recursos mínimos necessários para a implementação do sistema, por meio não guiado, atuando dentro de faixas de frequências ISM, que atendem ao requisito de mínima poluição eletromagnética do ambiente, com protocolos de rede padronizados. Ainda foi tomada toda a precaução em relação à proteção física dos dispositivos contra ações externas destrutivas, típicas do ambiente agressivo descrito. A intenção inicial do projeto seria realizar o teste definitivo no ambiente real do aeroporto, os sensores foram enviados e encontram-se no local, mas o tempo limitado de liberação da pista para testes não contemplou no calendário

uma data viável para que fosse feita a implantação do sistema, restando desta forma, a análise dos ensaios em Santa Maria.

A análise dos dados obtidos permitiu mostrar que o padrão *ZigBee* funciona dentro de uma hierarquia similar aos conceitos adquiridos em aula, no que se refere a topologias e estrutura dos protocolos conhecidos. Apesar de utilizar um padrão próprio, que mescla as diretrizes da *Zigbee Alliance* ao padrão IEEE 802.15.4, ficou claro que existe uma completa interação com a rede local e mundial, fazendo a comunicação primária, por exemplo, por Telnet, passando por redes 802.3, 802.11 e outros serviços através da internet. Assim sendo, é possível determinar com mais clareza que, uma vez estabelecida a comunicação de fato, não existem mais limites de distância para o sensoriamento remoto de qualquer dispositivo.

14.1. Sugestões para pesquisas futuras:

A comunicação de sensores, da maneira como foi proposta, permite uma variedade muito grande de projetos de utilização dos recursos. A faixa de variação disponível com palavra de 10 bits possibilita configurar até 1024 valores diferentes, atendendo bem a redes de monitoramento em geral e o controle de dispositivos, assim o presente projeto poderia ser estendido para diversas outras aplicações.

O serviço “em nuvem” também tem sido uma tendência muito forte e pode ser utilizado em várias aplicações, sem a necessidade de desenvolvimento local complexo, este recurso pode ser amplamente explorado para fins diversos. Uma forte possibilidade seria de estabelecer a comunicação em nuvem não proprietária, pois, apesar de funcionar muito bem, a solução iDigi limita o trabalho à disponibilidade de um serviço que, além do custo extra, poderia estar indisponível no futuro.

Uma sugestão de pesquisa importante é a de que o aluno desenvolvedor faça o processo inverso, ou seja, ao invés de apenas monitorar, promova, por exemplo, a troca de parâmetros à distância utilizando a mesma tecnologia ou similar com a finalidade de controlar algum equipamento ou fornecer informação a um operador distante.

15. REFERÊNCIAS

802.11 *PHY LAYERS. CWPAP*. Disponível em:

<http://media.techtarget.com/searchMobileComputing/downloads/CWPAP_ch8.pdf>. Acesso em: 09 mai. 2012.

ANDRIGHETTO, E. **Sistema de processamento de sinais biomédicos: rede wireless ZigBee com aplicação do padrão IEEE 802.15.4**. 2008. 163 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Santa Catarina, Florianópolis, 2008.

ARAÚJO, R. C. C. **Sistema telemétrico dinâmico sem fio aplicado a veículos rodoferroviários em malhas metroferroviárias**. 2009. 67 f. Tese (Doutorado em Engenharia Mecânica) – Universidade Federal da Paraíba, João Pessoa, 2009.

BARRICHELO, C. H. **Concepção de um nó sensor/atuator sem-fio para uma rede de gerenciamento de iluminação pública**. 2009. 121 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Santa Maria, Santa Maria, 2009.

BASTOS, H. Diferenças entre linguagem compilada e linguagem interpretada. **Henrique Bastos.Net**. Disponível em: <<http://henriquebastos.net/2008/09/06/diferencas-entre-linguagem-compilada-e-linguagem-interpretada/>>. Acesso em: 17 mai. 2012.

CAMPOS, F. P. S. C. **Estudo e especialização de um sistema de instrumentação para unidades de elevação de petróleo utilizando tecnologia sem fio**. 2006. 77 f. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Rio Grande do Norte, Natal, 2006.

COLVERO, C. P. Notas de aula e ilustrações apresentadas em aula na disciplina de Redes Industriais, Curso Superior de Tecnologia em redes de Computadores, Universidade Federal de Santa Maria, 2010.

CONNECTPORT® X4 GETTING START GUIDE. Digi Inc. 2011. Disponível em: <http://ftp1.digi.com/support/documentation/90001248_a.pdf>. Acesso em: 11 mai. 2012.

CONVERTING SIGNAL STRENGTH PERCENTAGE TO DBM VALUES. **WildPackets**. 2002. Disponível em: <http://www.wildpackets.com/elements/whitepapers/Converting_Signal_Strength.pdf>. Acesso em: 09 mai. 2012.

CARVALHO, M. **Segurança Patrimonial**. Blog pessoal. 2010. Disponível em: <http://segpatr.blogspot.com.br/2010_06_01_archive.html>. Acesso em: 23 jun. 2012.

CORREIA, M. F.: **Gerência de Redes**. 2004. Trabalho de Final de Curso. (Projeto de Conclusão de Curso de Bacharelado em Sistemas de Informação). União Educacional de Minas Gerais. 2004.

EFEITO DOPPLER. **SÓFísica**. Disponível em:

<<http://www.sofisica.com.br/conteudos/Ondulatoria/Acustica/doppler.php>>. Acesso em: 02 mai. 2012.

FERDINANDO, M. **Sensoriamento de ambiente utilizando o padrão ZigBee**. 2007. 92 f. Dissertação (Mestrado em Engenharia Elétrica) – Escola de Engenharia de São Carlos, da Universidade de São Paulo, São Carlos, 2007.

FLORIDO, I. R. **Redes de sensores sem fio em ambientes veiculares baseada no padrão ZigBee**. 2008. 136 f. Dissertação (Mestrado em Engenharia) – Escola Politécnica da Universidade de São Paulo, São Paulo, 2008.

WHAT IS ZIGBEE? ICPDAS Co. Disponível em:

<http://www.icpdas.com/products/GSM_GPRS/zigbee/zigbee_introduction.htm>. Acesso em: 12 mai. 2012.

IDIGI GATEWAY DEVELOPMENT KIT GETTING STARTED GUIDE – WIRELESS WAN VERSION. Digi Products. Disponível em:

<<http://www.digi.com/gatewaydevelopmentkit>>. Acesso em: 10 mar. 2012.

IEEE 802.15. **IEEE 802.15 WPAN™ Task Group 4 (TG4)**. *WPAN Home Page*. Disponível em: <<http://www.ieee802.org/15/pub/TG4.html>>. Acesso em: 14 jun. 2012.

INFORMETECZB. Repositório Institucional de La Universidad de Alicante. Alicante. Espanha. Disponível em:

<<http://rua.ua.es/dspace/bitstream/10045/1109/1/InformeTecZB.pdf>>. Acesso em: 15 jun. 2012.

INFRAERO. Aeroporto Santos Dumont. **Histórico**. Brasília – DF. Disponível em:

<<http://www.infraero.gov.br/index.php/br/aeroportos/rio-de-janeiro/aeroporto-santos-dumont/historico.html>>. Acesso em: 19 jun. 2012.

INFRAERO DÁ EXPLICAÇÕES SOBRE O ACIDENTE COM A FAMÍLIA NO AEROPORTO SANTOS DUMONT (RJ). **Rede Record**, Rio de Janeiro 14 mar 2012.

Disponível em: <<http://rederecord.r7.com/video/infraero-da-explicacoes-sobre-acidente-com-familia-no-aeroporto-santos-dumont-rj--4f60f37de4b0ee187a622b00/>>. Acesso em: 01 mai. 2012.

GARCIA, R. R. **Understanding the ZigBee Stack**. 2006. Artigo. Disponível em:

<http://www.eetasia.com/ARTICLES/2006JAN/PDF/EEOL_2006JAN02_RFD_NETD_TA_01.pdf?SOURCES=DOWNLOAD>. Acesso em: 09 jun. 2012.

GOUVEIA, B. A. T. **Dispositivos de monitorização e controlo automático de factores climáticos em Museus**. 2009. 141 f. Dissertação (Mestrado em Engenharia de Telecomunicações e Redes) – Universidade de Madeira, 2009.

KINNEY, P. **ZigBee Technology: Wireless Control that Simply Works**. Cadeira de grupo de estudos IEEE 802.15.4. Communications Design Conference. 2003. Disponível em:

<http://www.zigbee.org/imwp/idms/popups/pop_download.asp?contentID=5162>. Acesso em: 08 jun. 2012.

KIRSTEN, R. I. Nossos aeroportos são seguros? Arquivos de um repórter – Blog. 07 fev. 2008. Disponível em: <<http://arquivosreporter.blogspot.com.br/2008/02/nossos-aeroportos-so-seguros.html>>. Acesso em: 30 mai. 2012.

KUROSE, J. F.; ROSS, K. W.: **Redes de Computadores e a Internet: Uma abordagem top-down**. Trad. 3 ed., Addison Wesley, São Paulo, 2006.

LAZZETA, F. Áudio Digital. Disponível em: <http://www.eca.usp.br/prof/iazzetta/tutor/audio/a_digital/a_digital.html>. Acesso em: 03 mai. 2012.

LENO, M. [Curso de *Python*] O que é *Python*. **Under-Linux.Org**. Disponível em: <<http://under-linux.org/blogs/magnun/%5Bcurso-de-python%5D-o-que-e-python-1300/>>. Acesso em: 17 mai. 2012.

LIMA, B.; BAKKER, J. Espectroscopia no infravermelho próximo para monitorização da perfusão tecidual. ARTIGO DE REVISÃO. 11. 2011. **Anais eletrônicos**. University Medical Center Rotterdam, Netherlands. Disponível em: <<http://www.scielo.br/pdf/rbti/v23n3/v23n3a13.pdf>>. Acesso em: 05 mai. 2012.

LORDELLO, D. Sensores eletrônicos internos. **Tudo sobre segurança**. Disponível em: <http://tudosobreseguranca.com.br/portal/index.php?option=com_content&task=view&id=445&Itemid=149>. Acesso em: 02 mai. 2012.

MALMSTEN, P. *Python-xbee Documentation Release 2.0.0*. 2010. Disponível em <<http://code.google.com/p/python-xbee/>>. Acesso em 13 jun. 2012.

MARSHALL, B.; HARRIS, T. Como funcionam os receptores GPS. ComoTudoFunciona. Disponível em: <<http://informatica.hsw.uol.com.br/receptores-gps.htm>>. Acesso em: 01 mai. 2012.

MONSIGNORE, F. **Sensoriamento de Ambiente Utilizando o Padrão ZigBee**. 2007. 92 f. Dissertação (Mestrado em Engenharia Elétrica) – Escola de Engenharia de São Carlos. São Paulo. 2007.

MORRE taxista que teve carro tombado por avião. **O Estadão**, Rio de Janeiro 2 fev. 2002. Disponível em: <<http://www.estadao.com.br/arquivo/cidades/2002/not20020202p15172.htm>>. Acesso em: 25 abr. 2012.

PABLOAEROBRASIL. Santos Dumont. Textos. Disponível em: <http://www.pabloaerobrasil.net/textos/santos_dumont.htm>. Acesso em: 19 jun. 2012.

PETERSON, L. L.; DAVIE, B.S.: **Computer Networks – A System Approach**. 2007. 4ª Ed., Editora Elsevier, 2007.

ROGERCOM. *Wireless ZigBee*. Anadia – Alagoas. Disponível em: <<http://www.rogercom.com.br>>. Acesso em: 10 abr. 2012.

RUBENS'S BLOG. *Exemple of XBee API frames*. 12 mar. 2009. Disponível em: <<http://rubenlaguna.com/wp/2009/03/12/example-of-xbee-api-frames/index.html/>>. Acesso em: 23 jun 2012.

SANTANA S. RFID – Identificação por Rádio Frequência. **WirelessBR**. Disponível em: <http://www.wirelessbrasil.org/wirelessbr/colaboradores/sandra_santana/rfid_01.html>. Acesso em: 13 abr. 2012

SANTOS, S. A. **Sintetizador de frequências de 2.4 GHz em cmos, 0,35µm para aplicações em ZigBee**. 2008. 72 f. Dissertação (Mestrado em Engenharia Elétrica) – Escola Politécnica da Universidade de São Paulo, São Paulo, 2008.

SCHAUENBURG, F. F. **Plataforma modular de rede sem fio para monitoramento remoto**. 2008. 34 f. Projeto de Conclusão de Curso (Graduação em Engenharia Elétrica) – Pontifícia Universidade Católica do Paraná, 2008.

SEÇÃO: TUTORIAIS REGULAMENTAÇÃO. Regulação de espectro: Uso não licenciado. **TELECO – Inteligência em Telecomunicações**. Disponível em: <http://www.teleco.com.br/tutoriais/tutorialespecradio/pagina_2.asp>. Acesso em: 06 mai. 2012.

SENSOR DE INTRUSÃO. Guia de referência. **Bosch Security Systems**. 2006. Disponível em: <http://www.boschsecurity.com.br/servicios/soporte_tecnico/tablas/Intrusion/IntrusionDetect%20Reference_PT_0709_BR.pdf>. Acesso em: 18 mai. 2012.

SENSORES INFRAVERMELHOS PASSIVOS. **Samtek**. Suporte. Disponível em: <<http://www.samtek.com.br/suporte.php?osCsid=7m7a51b8bmmbdfnklq0c7js3s4>>. Acesso em: 15 mai. 2012.

SILVA, M. A. O efeito Doppler. Brasil Escola. Disponível em: <<http://www.brasilecola.com/fisica/o-efeito-doppler.htm>>. Acesso em: 02 mai. 2012.

SMART PLUG INTERACTIVE DEMO. Digi Demo Applications. Disponível em: <http://www.digi.com/wiki/developer/index.php/Smart_Plug_Interactive_Demo>. Acesso em: 13 jun. 2012.

SOUZA JR, J. R. Boing da Vasp faz taxi capotar no Aeroporto Santos Dumont. In **AEROPORTOSEMPERIGO**. Blog pessoal. 2007. Disponível em: <<http://juniornaweb.blogspot.com.br/2007/08/foto-arquivo-pessoal-do-autor-do-site.html>>. Acesso em: 05 mar. 2012.

STEVANOVIC, D. *ZigBee/IEEE 802.15.4 Standard*. Apresentação. Disponível em: <<http://www.cse.yorku.ca/~dusan/Zigbee-Standard-Talk.pdf>>. Acesso em: 08 jun. 2012.

STRUCT — INTERPRET STRINGS AS PACKED BINARY DATA. Python v2.7.3 documentation. Disponível em: <<http://docs.python.org/library/struct.html>>. Acesso em: 14 jun. 2012.

TANENBAUM, A. S. **Redes de Computadores**. 4ª Ed., Editora Campus (Elsevier), 2003.

TYSON, J. Como funciona a visão noturna. **Comotudofunciona**. Disponível em: <<http://eletronicos.hsw.uol.com.br/visao-noturna1.htm>>. Acesso em: 03 mai. 2012.

TEIXEIRA, L. M. **Desenvolvimento de uma aplicação com o protocolo ZigBee aplicado em implementação de ensaios em voo**. 2006. 163 f. Tese (Mestrado em Engenharia Eletrônica e Computação) – Instituto Tecnológico de Aeronáutica, São José dos Campos, 2006.

VASQUES, B.L.R.P; COUTINHO, I.B.A; LIMA, M.F.; CARNEVAL, V.P.O. **ZigBee**. Engenharia de Controle e de Computação. Redes de Computadores I. Universidade Federal do Rio de Janeiro. 2010. Disponível em: <http://www.gta.ufrj.br/grad/10_1/zigbee/>. Acesso em: 31 mai. 2012.

XBEE®/XBEE PRO® ZB RF MODULES. ZigBee RF Modules by Digi International. Disponível em: <<http://www.digi.com>>. Acesso em: 12 mar. 2012.

ZUCATO, F. L. **Rede ZigBee Gerenciada por Sistema de Monitoramento Remoto Utilizando TCP/IP e GPRS**. 2009. 138 f. Dissertação (Mestrado em Engenharia Elétrica – Telecomunicações) – Escola de Engenharia de São Carlos. São Paulo. 2009.

APÊNDICES

APÊNDICES

Apêndice A – Programa de aquisição dos sensores em *Python*

```
#####
# This is the main project module
# Created on: 16 April 2012
# Author: Anderson Colvero
# Description: Monitoramento de pista simplificado SDU
#####
import zigbee
import binascii
import struct
import xbee

# Endereca cada um dos sensores
GATEWAY="[00:13:a2:00:40:70:9c:25]!"
SENSOR1="[00:13:a2:00:40:3b:8e:36]!"
SENSOR2="[00:13:a2:00:40:3b:8e:34]!"
SENSOR3="[00:13:a2:00:40:2d:7b:30]!"
SENSOR4="[00:13:a2:00:40:3b:8e:3e]!"
RESERVA="[00:13:a2:00:40:7B:0A:36]!"

# inicia o processo de descoberta dos nodes da rede Zigbee
print "Monitorando a rede Xbee para Identificacao dos Dispositivos Ativos...\r\n"

# Obtem a lista com os Xbees nodes descobertos
node_list = xbee.getnodelist()

# Escreve uma tabela com a lista de dispositivos encontrados
print "%12s %12s %8s %28s" % \
      ("Nome (NI)", "Tipo", "Endereco", "End. Extendido")

print "%12s %12s %8s %28s" % \
      ("-"*12, "-"*12, "-"*8, "-"*28)

# Se existir algum Xbee achado, adiciona para a tabela
if node_list:
    for node in node_list:
        print "%12s %12s %8s %28s" % \
              (node.label, node.type, node.addr_short, node.addr_extended)

print " "
# Encerra o processo de descoberta dos nodes da rede Zigbee

# Inicia uma rotina para parseamento dos dados brutos
def parseIS(data):
    if len(data) % 2 == 0:
        sets, datamask, analogmask = struct.unpack("!BHB", data[:4])
        print "Dados brutos = ", data
        print "Sets = ", sets
        print "Datamask = ", datamask
        print "Analogmask = ", analogmask
        data = data[4:]
        print "Data[4:] = ", data

    else:
        sets, mask = struct.unpack("!BH", data[:3])
        print "Dados brutos = ", data
        print "Sets = ", sets
        print "Mask = ", mask
        print "Analogmask = ", analogmask
        print "Datamask = ", datamask
        data = data[3:]
        print "Data[3:] = ", data
        datamask = mask % 512 # Move os primeiros 9 bits para uma máscara separada
        analogmask = mask >> 9 # Move os últimos 7 bits para uma máscara separada

retidir = { }
```

```
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
        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
# Encerra a rotina de parseamento dos dados brutos
print " "
print "Dados recebidos modo ADC: (n amostras)"
print " "
numerador = 1
runs = 10000
while runs:

    dado_bruto1 = zigbee.ddo_get_param(SENSOR1, 'IS')
    print numerador
    dados1 = parseIS( dado_bruto1 )
    numerador += 1
    print " "
    print "Estado do SENSOR 01 ----->", dados1
    print " "
    dado_bruto2 = zigbee.ddo_get_param(SENSOR2, 'IS')
    dados2 = parseIS( dado_bruto2 )
    print "Estado do SENSOR 02 ----->", dados2
    dado_bruto3 = zigbee.ddo_get_param(SENSOR3, 'IS')
    dados3 = parseIS( dado_bruto3 )
    print "Estado do SENSOR 03 ----->", dados3
    dado_bruto4 = zigbee.ddo_get_param(SENSOR4, 'IS')
    dados4 = parseIS( dado_bruto4 )
    print "Estado do SENSOR 04 ----->", dados4
    print " "
    runs -= 1

```


Continua

[illegible]

Conclusão

[illegible]

Apêndice C – Configuração dos módulos XBee

PROGRAMAÇÃO DO COORDENADOR:	Extended PAN ID (ID): 0x0000000000000000 8 hex bytes;
Discover Timeout (NT): 60 x 100 msec (32-255);	Scan Channels (SC): 0x3fff hex (0x3fff=all channels);
Scan Duration (SD): 3 (0-7) ;	Allows Join Time (NJ): 255 seconds (0-255, 255=always) ;
Broadcast Hops (BH): 0 (0-32, 0=maximum);	RSSI PWM (P0): Yes (Enable RSSI PWM);
RSSI Timer (RP): 40 tenths of second (0-255);	Associate LED (D5): LED Blinks When Associated;
Baud Rate (BD): 9600	Parity (NB): None;
Flow Control (D7): Yes (Enable CTS Flow Control (DIO7))	Packetization Timeout (RO): 3 character times (0-255)
Aggregation route notification (AR): 255 x 10 sec (0-255);	Associate LED blink time (LT): 0 x 10 msec (0-255);
Broadcast radius (BH): 0 (0-32);	Command sequence character (CC): + (char);
Cluster identifier (CI): 0x11 (0x0-0xffff);	Command mode timeout (CT): 100 x 100 msec (2-655);
Destination address (DH/DL): 00:00:00:00:00:00:00:00! (address);	Destination endpoint (DE): 0xE8 (0x1-0xf0);
AD0/DIO0 configuration (D0): 2* ou 0 (0-5); *sensor1	AD1/DIO1 configuration (D1): 2* ou 0 (0-5); *sensor2
AD2/DIO2 configuration (D2): 2* ou 0 (0-5); *sensor3	AD3/DIO3 configuration (D3): 2* ou 0 (0-5); *sensor4
AD4/DIO4 configuration (D4): 0 (0-5);	DIO5/Assoc configuration (D5): 1 (0-5);
DIO6 configuration (D6): 0 (0-5);	DIO7 configuration (D7): 1 (0-7);
DIO10/PWM0 configuration (P0): 0 (0-5);	DIO11/PWM1 configuration (P1): 0 (0-5);
DIO12/CD configuration (P2): 0 (0-5);	DIO change detect (IC): 0x0 bitfield (0x0-0xffff);
Node discovery timeout (NT): 60 x 100 msec (32-255);	Encryption enable (EE): 0 (0-1);
Encryption options (EO): 0x0 bitfield (0x0-0xff);	Guard times (GT): 1000 msec (2-3300);
Node join time (NJ): 255 sec (0-255);	Join notification (JN): 0 (0-2);
Join verification (JV): 0 (0-1);	Link encryption key (KY): 0 (0-16 bytes);
Maximum hops (NH): 30 (0-255);	Network watchdog timeout (NW): 0 min (0-25855);
Node identifier (NI): SENSOR<Número> (0-20 chars);	Packetization timeout (RO): 3 chars (0-255);
Polling rate (PO): 0 x 10 msec (0-1000);	Power mode (PM): 1 (0-1);
Pull-up resistor enable (PR): 0x1fff bitfield (0x0-0xffff);	RSSI PWM timer (RP): 40 x 100 msec (0-255);
I/O sample rate (IR): 0 msec (0-65535);	Scan channels (SC): 0x3fff bitfield (0x1-0xffff);
Scan duration (SD): 3 exponent (0-7);	Serial interface parity (NB): 0 (0-4);
Serial interface data rate (BD): 3 (0-115200);	Peripheral sleep count (SN): 1 (1-65535);
Sleep mode (SM): 0 (0-6);	Sleep options (SO): 0x0 bitfield (0x0-0xff);
Cyclic sleep period (SP): 32 x 10 msec (32-2800);	Time before sleep (ST): 5000 msec (1-65535);
Source endpoint (SE): 0xE8 (0x1-0xff) ;	ZigBee stack profile (ZS): 0 (0-2);
Stop bits (SB): 0 (0-1);	

CONFIGURAÇÕES UTILIZADAS NOS SENSORES:
Modem Type: XBP24-ZB
Function Set: Zigbee Router AT - Version: 228C
SENSOR 1: End: 00:13:A2:00:40:7B:0A:40; Porta D0; Descrição: Sensor do ponto de parada da aeronave. Network address (MY): 0x67a5;
SENSOR 2: End: 00:13:A2:00:40:7B:0A:45; Porta D1; Descrição: Sensor redundante do ponto de parada. Network address (MY): 0x6aeb;
SENSOR 3: End: 00:13:A2:00:40:7B:0A:3E; Porta D2; Descrição: Sensor da cabeceira da pista. Network address (MY): 0x401f;
SENSOR 4: End: 00:13:A2:00:40:7B:0A:4E; Porta D3; Descrição: Sensor de detecção de decolagem (pista). Network address (MY): 0x190;
SENSOR RESERVA: End: 00:13:A2:00:40:7B:0A:36; Porta D0; Descrição: Apto para utilização em qualquer ponto.
PROGRAMAÇÃO DOS SENSORES:
XBee Node:
Node Type: router;
Profile ID: 0xc105;
Manufacturer ID: 0x101e.
PAN identifier (OI): 0x9384 ;
Extended PAN identifier (OP): 0xa6dc68775b2079f8 ;
Operating channel (CH): 0x13;
Parent address (MP): 0xffff;
Association indication (AI): 0x0 ;
Device type identifier (DD): 0x30010;
Number of remaining children (NC): 12;
Maximum RF payload (NP): 84 bytes;
Received signal strength (DB): 102 -dBm;
Transmit power at PL4 (PP): 17 dBm;
ACK failures (EA): 0;

Apêndice D – Cronograma das atividades

O presente projeto foi desenvolvido no decorrer de 4 (quatro) meses, estando as atividades assim programadas:

Cronograma de atividades	
Período	Atividades programadas
15 a 31 de março de 2012	<ul style="list-style-type: none">- Análise do problema proposto;- Fundamentação teórica e busca de soluções com equipamentos existentes no mercado;- Análise da viabilidade de execução do projeto em modo local;- Repasse das especificações obtidas visando à aquisição dos equipamentos.
01 a 30 de abril de 2012	<ul style="list-style-type: none">- Montagem dos equipamentos para o funcionamento individual e primeiros testes de desempenho;- Desenvolvimento de fluxogramas de trabalho e de software;- Ajustes e desenvolvimento de rotinas de testes;- Integração dos dispositivos e adequação do software.
01 a 31 de maio de 2012	<ul style="list-style-type: none">- Execução de ensaios iniciais, em ambiente indoor (laboratório);- Execução de ensaios em ambiente externo;- Aquisição das informações (gráficos e tabelas) obtidas;- Envio do equipamento para testes em ambiente real.
01 a 30 de junho de 2012	<ul style="list-style-type: none">- Conclusão do projeto desenvolvido;- Análise dos resultados obtidos;- Consolidação das informações obtidas para a confecção do relatório final;- Encaminhamento do relatório concluído.

ANEXOS

Anexo A – Pinagem do modulo XBee

Pino	Nome	Direção	Descrição
1	VCC	-	Alimentação 3,3V
2	DOUT	Saída	Saída de dados da UART
3	DIN	Entrada	Entrada de dados da UART
4	RESET	Entrada	Inicializa módulo (um pulso nível 0 de pelo menos 200ms)
5	DIO9	Entrada/Saída	Entrada/Saída Digital 9
6	PWM0 / RSSI	Saída	Saída do PWM 0 / Indicador de Nível RSSI
7	PWM1 / RSSI	Saída	Saída do PWM 1 / Indicador de Nível RSSI
8	CLOCKOUT	Saída	Saída de Clock de 4MHz
9	DIO8	Entrada/Saída	Entrada/Saída Digital 8
10	GND	-	Terra
11	AD3 / DIO3	Entrada/Saída	Entrada Analógica 3 ou Entrada/Saída Digital 3
12	DIO7	Entrada/Saída	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	AD4 / DIO4	Entrada/Saída	Entrada Analógica 4 ou Entrada/Saída Digital 4
16	AD5 / DIO5	Entrada/Saída	Entrada Analógica 5 ou Entrada/Saída Digital 5
17	DIO6	Entrada/Saída	Entrada Analógica 6 ou Entrada/Saída Digital 6
18	AD2 / DIO2	Entrada/Saída	Entrada Analógica 2 ou Entrada/Saída Digital 2
19	AD1 / DIO1	Entrada/Saída	Indicador de Associação, Entrada Analógica 1 ou Entrada/Saída Digital 1
20	AD0 /DIO0	Entrada/Saída	Indicador de Associação, Entrada Analógica 0 ou Entrada/Saída Digital 0

Fonte: www.digi.com