INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DO RIO GRANDE DO NORTE

MATHEUS LUIZ NEGREIROS CÂMARA

MONITORAMENTO DE TEMPERATURA E UMIDADE DE UM AMBIENTE UTILIZANDO O PROTOCOLO ZIGBEE

NATAL–RN
2016

MATHEUS LUIZ NEGREIROS CÂMARA

MONITORAMENTO DE TEMPERATURA E UMIDADE DE UM AMBIENTE UTILIZANDO O PROTOCOLO ZIGBEE

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Redes de Computadores do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo em Redes de computadores.

Orientador: M.e Ivanilson França Vieira Júnior.
Coorientador: Prof. Nelson Ion de Oliveira

NATAL–RN
2016

C172m Câmara, Matheus Luiz Negreiros.

Monitoramento de temperatura e umidade de um ambiente utilizando o protocolo Zigbee / Matheus Luiz Negreiros Câmara. – 2016.

xx f. ; il.

Orientador: Prof. Me. Ivanilson França Vieira Júnior.

Coorientador: Prof. Tit. Nelson Ion de Oliveira.

Trabalho de Conclusão de Curso (Tecnologia em Rede de Computadores) — Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, 2016.

1. Protocolo Zigbee. 2. Tecnologia sem fio. 3. Drone. I. Vieira Júnior, Ivanilson França. II. Oliveira, Nelson Ion de. III. Título.

CDU 004.7

Ficha elaborada pelo Setor de Informação e Referência da Biblioteca Sebastião Fernandes do IFRN.

MATHEUS LUIZ NEGREIROS CÂMARA

MONITORAMENTO DE TEMPERATURA E UMIDADE DE UM AMBIENTE UTILIZANDO O PROTOCOLO ZIGBEE

Trabalho de Conclusão de Curso apresentado apresentada ao Curso Superior de Tecnologia em Redes de Computadores do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo em Redes de Computadores.

Trabalho de Conclusão de Curso apresentado e aprovado em ___/___/____, pela seguinte Banca Examinadora:

BANCA EXAMINADORA

Ivanilson França Vieira Júnior, M.e - Presidente

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

---

Nelson Ion de Oliveira, Prof. - Examinador

Universidade Federal do Rio Grande do Norte

Allyson Amilcar Angelus Freire Soares, ESp. - Examinador

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

AGRADECIMENTOS

Agradeço a Deus primeiramente, por ter me dado saúde, força e perseverança para mais essa conquista em minha vida.

Aos meus familiares e em especial para minha mãe Clodes Maria que sempre me deu apoio e nunca me deixou falta nada e a minha tia Noélia, que infelizmente não está mais entre nós, mas sempre torceu pelas minhas vitórias e minha felicidade.

A minha namorada Emanuelly que sempre me ajudou nos momentos difíceis e me deu forças para terminar o curso e elaborar esse Trabalho de Conclusão de Curso (TCC).

Ao meus amigos e colegas de curso pelo apoio que me deram durante todo o percurso, em especial, Michel, Isadora, Thiago, Manoel, Wallyson e Carlos Henrique, que me deram muita ajuda, inclusive nas adversidades encontradas durante esse trabalho.

Ao Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN) por ter me dado bases para meu aprimoramento profissional através da aprendizagem.

Finalmente agradeço aos professores da Diretoria Acadêmica de Gestão e Tecnologia da Informação (DIATINF), Carlos Rocha, Ronaldo Maia, Sales Filho, Rodrigo Siqueira, ao meu co-orientador Nelson, pois sem ele não teria sido possível concluir esse trabalho e principalmente ao meu orientador, Ivanilson França, que com sua pedagogia me tornou a pessoa que sou hoje.

RESUMO

A utilização das tecnologias *Zigbee* e *Arduino*, bem com o uso de *Drones* em atividades do cotidiano estão cada vez mais frequentes, o que leva a utilização desses dispositivos para inúmeras áreas. Esse trabalho tem como principal objetivo a coleta de dados de temperatura e umidade de um espaço ou ambiente, através do módulo para comunicação sem fio *Xbee* e do uso de um *Drone* para monitoramento, realizando o envio em tempo real desses dados. Para elaboração desse trabalho foram realizadas pesquisas e estudos sobre as tecnologias utilizadas, a configuração do módulo *Xbee* e a programação da placa de prototipagem *Arduino*, além da montagem e configuração de um *Drone*. Por fim foram realizados testes de envio dos dados coletados através dos módulos *Xbee*, seus resultados e as dificuldades encontradas durante todo o trabalho.
Palavras-chave: *Zigbee*. *Drone*. Placa de prototipagem *Arduino*. Módulo *Xbee.*

ABSTRACT

Use of the Zigbee technology and Arduino, as well as the use of Drones in daily activities are increasingly frequent, which leads to use of these areas for numerous devices. This work aims to collect temperature and humidity data from a space or environment through the module for wireless communication XBee and the use of a Drone for monitoring, performing sending in real time such data. For preparation of this work were carried out research and studies on the technology used, the configuration of the XBee module and programming the Arduino prototyping board, as well as installation and configuration of a Drone. Finally the data sending tests were performed collected via XBee modules, their results and the difficulties encountered during the work.
Keywords: Zigbee. Drone. Prototyping board Arduino. XBee module.

LISTA DE ILUSTRAÇÕES

Figura 1 - Componentes da placa Arduino 18
Figura 2 - Comparação entre o padrão IEEE 802.15.4 e o Modelo OSI 20
Quadro 1 - Resumo das bandas de frequência utilizadas pelo padrão IEEE 802.15.4 21
Figura 3 - Estrutura do superframe 22
Figura 4 - Camadas implementadas pelo Zigbee e pelo IEEE 802.15.4 23
Figura 5 - Tipos de quadro APS 25
Figura 6 - Camada ZDO como interface entre a camada APS e o Framework de Aplicação 26
Figura 7 - Perfil da Aplicação 27
Figura 8 - Aplicação do Código de Autenticação de Mensagens (MIC) na autenticação de dados 28
Figura 9 - Topologias Zigbee 29
Figura 10 - Módulos Xbee Series 2 Regular e PRO 30
Figura 11 - Tipos de antenas XBee 30
Figura 12 - Dimensões do Sensor DHT11 33
Quadro 2 - Especificações de temperatura do Sensor DHT11 32
Quadro 3 - Características elétricas do Sensor DHT11 33
Quadro 4 – Classificação e liberação de uso de Drones pela ANAC 34
Figura 13 - Drone montado para realização dos testes 41
Figura 14 - Placa controladora de voo KK Multicopter 43
Figura 15 - Placa USB Explorer para configuração do módulo Xbee 44
Figura 16 - Configuração do Módulo Emissor 45
Figura 17 - Configuração do Módulo Receptor 45
Figura 18 - Circuito Receptor e Circuito Emissor 48
Quadro 5 - Dados coletados pelo XBee e enviados para o receptor 50

LISTA DE SIGLAS

ANAC	Agência Nacional de Aviação Civil
AES	Advanced Encryption Standard
APL	Camada de Aplicação Zigbee
APS	Application Support
APSDE	APS Data Entity
APSME	APS Management Entity
BPSK	Binary Phase Shift Keying
CCM	Cipher Block Chaining Message Authentication Code
CMA	Certificado Médico Aeronáutico
CSMA-CA	Carrier Sense Multiple Access with Colision Avoidance
DSSS	Diret Sequence Spread Spectrum
EEPROM	Electrically-Erasable Programmable Read-Only Memory
EPO	Expanded Polyolefin foam
EPP	Expanded Polypropylene foam
EPS	Expanded Polystyrene foam
ESC	Electronic Speed Control
FPV	First Person View
HVAC	heating, ventilation, and air-conditioning
IEEE	Institute of Electrical and Electronics Engineers
IMD	Instituto Metrópole Digital
ISM	Industrial Scientific Medical
LRS	long range system
MIC	Message Integrity Code
NLDE	Network Layer Data Entity
NLME	Network Layer Management Entity
O-QSKY	Offset Quadrature Phase Shift Keying
PDU	Protocol Data Unit
RAB	Registro Aeronáutico Brasileiro
RF	Rádio Frequência
RPA	Remotely-Piloted Aircraft
RPAS	Sistemas de Aeronaves Remotamente Pilotada
SAP	Service Access Point

UAV	Unmanned Aerial Vehicle
VANT	Veículo Aereo não Tripulado
VARP	Veículo Aéreo Remotamente Pilotado
XPS	Extruded version of EPS
ZDO	Zigbee Device Objects
ZDP	Zigbee Device Profile

SUMÁRIO

1 INTRODUÇÃO 12

1.1 OBJETIVOS 13

1.1.1 Objetivos gerais 13

1.1.2 Objetivos específicos 13

1.2 JUSTIFICATIVA 13

1.3 ESTRUTURA DO TRABALHO 14

2 FUNDAMENTAÇÃO TEÓRICA 15

2.1 MICROCONTROLADOR 15

2.1.1 Evolução dos microcontroladores 15

2.1.2 Programação para microcontroladores 16

2.1.3 Aplicabilidade dos microcontroladores 16

2.2 A PLATAFORMA ARDUINO 17

2.2.1 Hardware e software 17

2.2.2 Características 18

2.3 O PADRÃO ZIGBEE 19

2.3.1 O IEEE 802.15.4 19

2.3.1.1 Camada física (PHY) IEEE 802.15.4 20

2.3.1.2 Camada de controle de acesso ao meio (MAC) IEE 802.15.4 21

2.3.2 As camadas implementadas pelo protocolo zigbee 22

2.3.2.1 Camada de rede zigbee 23

2.3.2.2 Camada de aplicação zigbee 24

2.3.3 Serviços de segurança 27

2.3.4 Topologias zigbee 28

2.3.5 Módulos zigbee 29

2.3.6 Aplicações zigbee 31

2.4 O SENSOR DE TEMPERATURA E UMIDADE DHT11 31
2.4.1 Características do sensor DHT11 32
2.4.2 Aplicações do sensor DHT11 33
2.5 O DRONE 33
2.5.1 Nomenclaturas utilizadas 34
2.5.2 Sistemas que compõem um drone 36
2.5.2.1 Sistema da estrutura 36
2.5.2.2 Sistema de rádio controle e piloto automático 36
2.5.2.3 Sistema de energia 37
2.5.2.4 Sistema de transmissão 38
2.5.3 A ANAC e a legislação dos drones 39
3 MONITORAMENTO DE TEMPERATURA E UMIDADE COM O USO DO PROTOCOLO ZIGBEE 41
3.1 MONTAGEM E CONFIGURAÇÃO DO DRONE 41
3.2 CONFIGURAÇÃO DO MÓDULO XBEE PRO S1 43
3.3 A PROGRAMAÇÃO NA PLACA ARDUINO 45
3.4 MONTAGEM DO CIRCUITO PARA OS TESTES 48
4 TESTES E RESULTADOS 49
4.1 TESTES REALIZADOS COM O DRONE SYMA X5SW 49
4.2 TESTES REALIZADOS COM O DRONE QUANUM NOVA 49
4.2.1 Teste em campo aberto 50
4.2.2 Teste sem campo de visada 50
5 CONSIDERAÇÕES FINAIS 52
5.1 CONCLUSÕES 52
5.2 DIFICULDADES ENCONTRADAS 53
REFERÊNCIAS 54
APÊNDICE A – Código inserido no Arduino com o módulo emissor 56

APÊNDICE B – Código inserido no Arduino com o módulo receptor 58

APÊNDICE C – Dados coletados pelo sensor e enviados através do módulo *XBee* série 1 PRO no dia 12 de março de 2016 60

ANEXO A – Foto do teste realizado com o *Drone* em campo de visada do módulo receptor no dia 12 de março de 2016 63

ANEXO B – Foto do teste realizado no dia 12 de março de 2016, em que o *Drone* perdeu a conexão com o módulo receptor 64

1 INTRODUÇÃO

Cada vez mais as novas tecnologias como o *Arduino*[1] e os VANTs (Veículo Aéreo não Tripulado) popularmente conhecidos como *Drones*, vêm conquistando bastante espaço no mercado com as mais diversas funcionalidades. No cenário atual os *Drones* estão sendo utilizados para realizar filmagens e fotografias aéreas, monitoramento de água, desmatamento, lixo e até entrega de produtos de pequeno porte, auxiliando assim no serviço de empresas. Eles exercem várias outras funções, as quais necessitam de modelos específicos das aeronaves, apesar de ainda existir diversas restrições pela ANAC (Agência Nacional de Aviação Civil). Analogamente o *Arduino* vem sendo usado em milhares de projetos e aplicações, graças à sua utilização simples e acessível. Professores utilizam-no para iniciar os alunos no estudo da programação e robótica, enquanto músicos e artistas utilizam-no para auxiliar na execução de suas apresentações.

Outra tecnologia que vem sendo bastante utilizada na área é o *Zigbee*, um "padrão sem fio global e aberto, que permite que objetos simples e inteligentes possam trabalhar em conjunto melhorando a eficiência na vida cotidiana" (ZIGBEE ALLIANCE, 2015). Esse padrão é definido pela "*ZigbeeAlliance*" – "uma associação sem fins lucrativos que promove a adoção do *Zigbee* como principal padrão de controle e detecção de rede sem fio em áreas industriais, comerciais e residenciais" (ZIGBEE ALLIANCE, 2015) - e trabalha junto ao padrão 802.15.4 do IEEE (*Institute of Electrical and Electronics Engineers*) o qual apresenta como principal característica o baixo consumo de energia, além de possuir uma fácil utilização e a capacidade de hospedar vários dispositivos em um rede.

Com a utilização em conjunto das tecnologias *Zigbee*, *Arduino* e *Drone* será possível realizar a coleta de dados de uma forma mais eficaz em lugares onde o acesso se tornaria inviável pelo ser humano ou até mesmo em situações onde se precisa de uma leitura rápida de um determinado ambiente, uma vez que o *Drone* poderá cobrir uma área em um tempo reduzido e as informações estarão sendo recebidas em tempo real. Nesse trabalho será apresentado o processo para se coletar dados de temperatura e umidade de um ambiente, no qual foi utilizado um *Drone*, o sensor de temperatura e umidade DHT11, a placa *Arduino*, onde foi realizada toda a lógica de programação e o módulo *XBee* PRO S1 para realizar a comunicação entre os dados coletados a partir do sensor e o computador que receberá esses dados.

[1] Plataforma de prototipagem eletrônica de hardware livre (LEMOS; Manoel, 2013)

1.1 OBJETIVOS

Este trabalho tem como objetivo gerar uma documentação fundamentada na atividade prática, bem como, fornecer maiores detalhes sobre as tecnologias utilizadas nesse processo.

1.1.1 Objetivos gerais

Realizar a coleta dos dados de temperatura e umidade através do sensor DHT11, utilizando para isso dois módulos *Xbee* PRO S1 para o envio dos dados coletados até a base, um *Drone*, utilizado na varredura do espaço a ser analisado e a placa *Arduino* que será programada para, em conjunto com os módulos *Xbee* e o sensor DHT, realizar a captura desses dados.

1.1.2 Objetivos específicos

a) montar um *Drone* para poder realizar os voos que irão servir para execução dos testes de coleta dos dados;
b) estudar e realizar a programação na placa *Arduino* necessária à coleta dos dados;
c) estudar e configurar o módulo *XBee* para realizar o envio dos dados capturados a partir do sensor;
d) armazenar os dados coletados de temperatura e umidade em uma planilha, através da utilização do *software* PLX-DAQ para futuras comparações e estudos.

1.2 JUSTIFICATIVA

Tendo como base a necessidade de uma ferramenta ágil e eficaz para monitoramento de fatores ambientais de uma determinada área, foi escolhido o *Drone*, que apresenta maior eficiência e alcance, bem como o sensor de temperatura e umidade, a placa *Arduino* e o módulo *XBee* que possui um menor custo. Esses equipamentos em conjunto, realizam a coleta e o envio de dados em tempo real, fazendo com que, o registro de temperatura e umidade de uma área de grandes proporções ou em locais de difícil acesso, sejam realizadas com maior rapidez e precisão, economizando horas de trabalho.

1.3 ESTRUTURA DO TRABALHO

O trabalho está dividido em seções para facilitar a sua leitura e entendimento. Na Seção 2 está descrita a fundamentação teórica, com uma explicação geral sobre todos os componentes envolvidos na realização do trabalho. Na seção 3 serão descritos a montagem do *Drone* e a programação da placa *Arduino*, bem como a configuração do módulo *Xbee*. Na seção 4 estarão os testes que foram realizados e na seção 5 serão descritas as conclusões a partir dos testes realizados e as dificuldades encontradas.

2 FUNDAMENTAÇÃO TEÓRICA

Essa seção descreverá os conceitos fundamentais das tecnologias utilizadas para o entendimento do trabalho realizado.

2.1 MICROCONTROLADOR

Um microcontrolador é um CI (circuito integrado) que pode ser programado para realizar funções específicas. “O microcontrolador é definido em SOUZA (2005 apud OLIVEIRA NETO; MONTEIRO, QUEIROGA, 2012, p. 2) como um pequeno componente eletrônico, dotado de uma inteligência programável, utilizado no controle de processos lógicos. ”.

Um microcontrolador integra diversos componentes a sua estrutura, como memória de leitura e escrita para armazenamento de dados, memória somente de leitura para armazenamento de programas, *timers*, contadores, EEPROM (Electrically-Erasable Programmable Read-Only Memory) para armazenamento permanente de dados, portas de entrada e saída digitais (I/O), comunicação serial, PWM, conversores analógico-digitais (ADC) e digitais-analógicos (DAC) entre outros.

Os microcontroladores operam a uma frequência e consumo muito baixos devido a tecnologia de fabricação mais comumente utilizada, o *chip* CMOS (Complementary Metal Oxide Semiconductor). Além disso podem ser realizadas diversas tarefas utilizando um único componente por ser possível programá-lo, o que o torna essencial no desenvolvimento de projetos e implementações de sistemas microcontrolados de pequeno e médio portes.

2.1.1 Evolução dos microcontroladores

Os microcontroladores tiveram seu início no desenvolvimento de circuitos integrados, onde foi possível armazenar todos os componentes necessários para um controlador em um único *chip*. Anteriormente, os sistemas de controle eram compostos por componentes lógicos com grandes dimensões físicas.

No processo de evolução do microcontroladores, primeiro surgiram os microcontroladores baseados em uma tecnologia de *hardware* relativamente primitiva, com grande limitação de memória e de processamento. Depois surgiram microcontroladores com interfaces mais amigáveis ao usuário fazendo uso de *display* de cristal líquido e utilizando a

comunicação com outros sistemas, através de interfaces seriais de comunicação. Atualmente os microcontroladores incorporam as características de *hardware* e *software* dos sistemas *desktops* atuais, processadores de 32 *bits*, alta capacidade de memória, interfaces de comunicação com redes heterogêneas e uma grande variedade de recursos.

2.1.2 Programação para microcontroladores

Os tipos de *softwares* desenvolvidos para sistemas microcontrolados não utilizam apenas os recursos do microntrolador, é comum se utilizar outros dispositivos, como mais memória, *display* gráfico, teclado e *mouse*, entre outros, para criação de um ambiente mais agradável. Logo, a aplicabilidade dos microcontroladores está ligada diretamente a programação, a qual, irá depender da linguagem utilizada. Com isso a linguagem de programação se torna um meio de interação entre o programador e o programa, fazendo com que um sistema possa compreender o que deve ser feito a partir dos comandos implementados. Após o programa ser criado, um compilador irá gerar o programa em linguagem binária, ou seja, realizará a tradução para linguagem de máquina.

Os microcontroladores sempre tiveram a sua programação associada à linguagem *Assembly*, por serem dispositivos com baixa quantidade de memória e estarem empregados em aplicações que envolvem o *hardware*. Apesar disso existem outras linguagens que podem ser utilizadas por eles, como *Python* e *JavaScript*, linguagens de alto nível em relação a linguagem *Assembly*, que estão sendo bastante utilizadas para programar microcontroladores e sistemas embarcados.

Atualmente a maioria dos microcontroladores disponíveis no mercado conta com compiladores de linguagem C para o desenvolvimento de software. (OLIVEIRA NETO; MONTEIRO, QUEIROGA, 2012).

2.1.3 Aplicabilidade dos microcontroladores

Os microcontroladores podem ser usados na automação industrial, automação comercial, automação predial, automação residencial, área automobilística, agrícola, produtos manufaturados, eletrodomésticos, telecomunicações, instrumentos biomédicos e de navegação via satélite, detectores de radar, equipamentos de áudio e vídeo, dentre outros.

São inúmeras aplicações, muitas vezes possuindo não um, mas vários microcontroladores, portanto, a introdução de um microcontrolador em um dispositivo

permite a inclusão de melhorias de segurança, funcionalidade, eficiência e redução de custos, facilitando e melhorando a vida dos usuários.

2.2 A PLATAFORMA ARDUINO

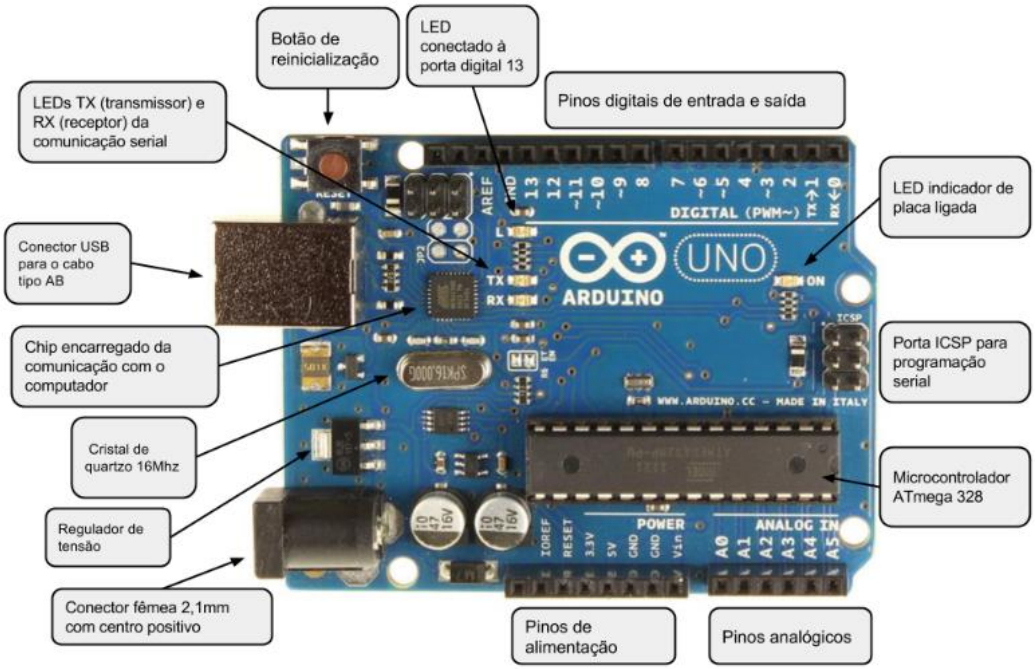
O *Arduino* nasceu no *Interaction Design Institute Ivrea*, na Itália para poder ser ensinada a disciplina de *Design* de Interação, que tinha como princípio fundamental a prototipação. Quando passou a ser difundido, a placa *Arduino* sofreu alterações no formato de sua placa simples de 8 *bits* para se adaptar as novas tendências, criando produtos para aplicações da *Internet* das coisas, tecnologias vestíveis (*wearable*), a impressão 3D e ambientes incorporados.

"O *Arduino* é uma plataforma de prototipagem de código aberto baseado em hardware e software de fácil utilização. " (ARDUINO, 2015). Tanto as placas quanto o *software* são *open-source* (código fonte aberto), o que torna seu uso de forma independente e seus projetos autônomos.

2.2.1 Hardware e software

O *hardware* é composto de um processador Atmel AVR, um cristal oscilador e um regulador linear de 5 volts e a placa é composta de pinos de entrada e saídas digitais e analógicas, onde podem ser conectados sensores e módulos externos através de *Shields*, placas que são plugadas em cima da placa *Arduino* estendendo suas capacidades, além de um conversor Serial/USB que pode ser utilizado tanto para inserir a programação no *Arduino*, quanto para alimentar a placa, sendo que a mesma também possui uma entrada para alimentação através de bateria. A figura 1 mostra uma placa *Arduino* UNO.

Figura 1 - Componentes da placa *Arduino*



Fonte: NATAL MAKERS (2015).

O *software* é a linguagem de programação desenvolvida para o microcontrolador, o qual é programado em *Arduino Programming Language*, baseado em *Wiring*, um Framework de programação de código aberto para microcontroladores que permite escrever um software multi-plataforma e criar todo tipo de objetos interativos. O ambiente de desenvolvimento integrado (IDE) do *Arduino* é um compilador gcc (C e C++) que utiliza uma interface gráfica construída em *Java* e baseado em *Processing*, “ um *scketchbook* de software flexível e uma linguagem para aprender como programar no contexto das artes visuais. ” (PROCESSING, 2016). Basicamente se resume a um programa IDE muito simples de se utilizar e que pode ser estendido a partir do uso de bibliotecas que podem ser facilmente encontradas. As funções da IDE do *Arduino* são, basicamente, compilar um software e enviá-lo à placa para que possa ser executado.

2.2.2 Características

Existem diversas outras plataformas de microcontroladores que apresentam funcionalidades semelhantes ao *Arduino* e todas estas plataformas transformam os detalhes complicados da programação de microcontroladores em um pacote mais simplificado, mas o *Arduino* se destaca devido as seguintes características:

a) Barato - placas *Arduino* são relativamente baratos em comparação com outras plataformas de microcontroladores.
b) Multi-plataforma - O Software *Arduino* (IDE) é executado em Windows, Macintosh OSX e sistemas operacionais Linux. A maioria dos sistemas de microcontroladores são limitados ao Windows.
c) Ambiente de programação simples - O *Software Arduino* (IDE) é de fácil utilização para os usuários novatos, mas suficientemente flexível para o aproveitamento de usuários avançados.
d) *Software* extensível e *Open source* - O software *Arduino* é uma ferramenta de código aberto, disponível para extensão por programadores experientes e a linguagem pode ser expandida através de bibliotecas C ++.
e) *Hardware* extensível e *Open source* - As placas *Arduino* são publicadas sob uma licença *Creative Commons*, para que os designers experientes de circuito possam fazer a sua própria versão do módulo, estendendo-o e melhorando-o. Mesmo usuários inexperientes podem construir uma versão do módulo da placa, a fim de entender como ela funciona e economizar dinheiro.

2.3 O PADRÃO ZIGBEE

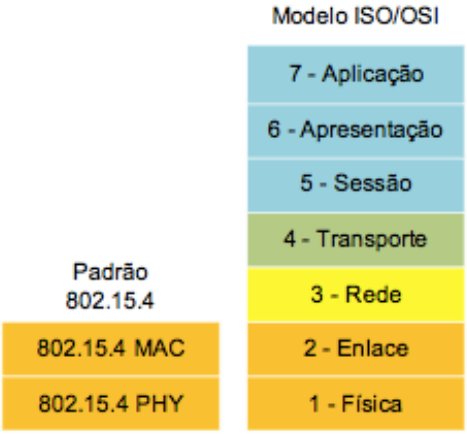
O *ZigBee* é um padrão definido por uma aliança de empresas de diferentes segmentos do mercado, chamada *ZigBee Alliance*, projetado para permitir comunicação sem fio confiável, com baixo consumo de energia e baixo custo, sendo operado pela especificação do padrão IEEE 802.15.4.

"*ZigBee* é uma nova norma técnica industrial, com abrangência global para comunicações sem fio de baixa potência e curta distância. " (STOLL, 2008).

2.3.1 O IEEE 802.15.4

O padrão IEE 802.15.4 opera em três faixas de frequências não-licenciadas de acordo com a localização geográfica e utiliza o CSMA-CA (*Carrier Sense Multiple Access with Colision Avoidance*) para o acesso ao meio físico. Esse padrão ficou responsável pela criação das duas camadas mais baixas da tecnologia *ZigBee*, as quais correspondem aos níveis Físico (PHY) e de Enlace (MAC), camadas semelhantes as primeiras camadas do modelo OSI. A figura 2 faz uma comparação entre as camadas.

Figura 2 - Comparação entre o padrão IEEE 802.15.4 e o Modelo OSI



Fonte: Santos (2013).

Nas próximas seções serão abordadas as camadas Física e de Enlace do padrão IEEE 802.15.4.

2.3.1.1 Camada física (PHY) IEEE 802.15.4

A camada PHY é a camada mais próxima do *hardware* e realiza o controle e comunicação diretamente com o rádio *transceiver*. Portanto a camada PHY fornece a interface com o meio físico e é responsável pela ativação e desativação do rádio, detecção de energia, qualidade da conexão (*link Quality*), pela transmissão e recebimento de pacotes de mensagens e por reportar canais livres (*Clear Channel Assesment*). Além disso, ela é responsável pelo estabelecimento do *link* RF entre dois dispositivos e pela sincronização entre transmissor e receptor.

O padrão IEEE 802.15.4 especifica um total de 27 canais *half-duplex* para as três bandas de frequência (868 MHz, 915 MHz e 2,4 GHz).

A banda 868 MHz varia entre 868.0 e 868.6 e é usada na Europa, utiliza modulação do tipo BPSK (*Binary Phase Shift Keying*) com DSSS (*Diret Sequence Spread Spectrum*) como método de propagação, possui somente um canal disponível com taxa de dados de 20 kbit/s.

A banda 915 MHz varia entre 902 e 928 MHz e é usada no norte da América e no Pacífico, também utiliza modulação do tipo BPSK (*Binary Phase Shift Keying*) com DSSS (*Diret Sequence Spread Spectrum*) como método de propagação, mas possui 10 canais disponíveis com uma taxa de dados de 40 kbit/s.

A banda 2,4 GHz, conhecida como ISM (*Industrial Scientific Medical*) se estende de 2400 até 2483.5 MHz e é utilizada em todo o mundo, utiliza modulação do tipo O-QSKY (Offset Quadrature Phase Shift Keying) com o método de propagação 16-array orthogonal, possui 16 canais disponíveis com taxa de dados de 250 kbit/s. A tabela 1 mostra um resumo com os dados das três bandas de frequência.

Quadro 1 - Resumo das bandas de frequência utilizadas pelo padrão IEEE 802.15.4

Banda	Frequência (MHz)	Modulação	Largura de Banda	Método de Propagação	Quantidade de canais	Local Disponível
868 MHz	868 - 868,6	BPSK	20 Kbps	DSSS Binário	1 canal	Europa
915 MHz	902 - 928	BPSK	40 Kbps	DSSS Binário	10 canais	EUA
2,4 GHz	2400 - 2483,5	O-QPSK	250 Kbps	16-array orthogonal	16 canais	Mundial

Fonte: Elaborado pelo autor deste trabalho (2015)

A camada PHY também é responsável pelo consumo de energia. O padrão permite que dispositivos, tanto transmissores quanto receptores, permaneçam inativos por 99% do tempo, ou seja, dispositivos compatíveis irão ser ativados somente no momento em que estiverem se comunicando.

2.3.1.2 Camada de controle de acesso ao meio (MAC) IEE 802.15.4

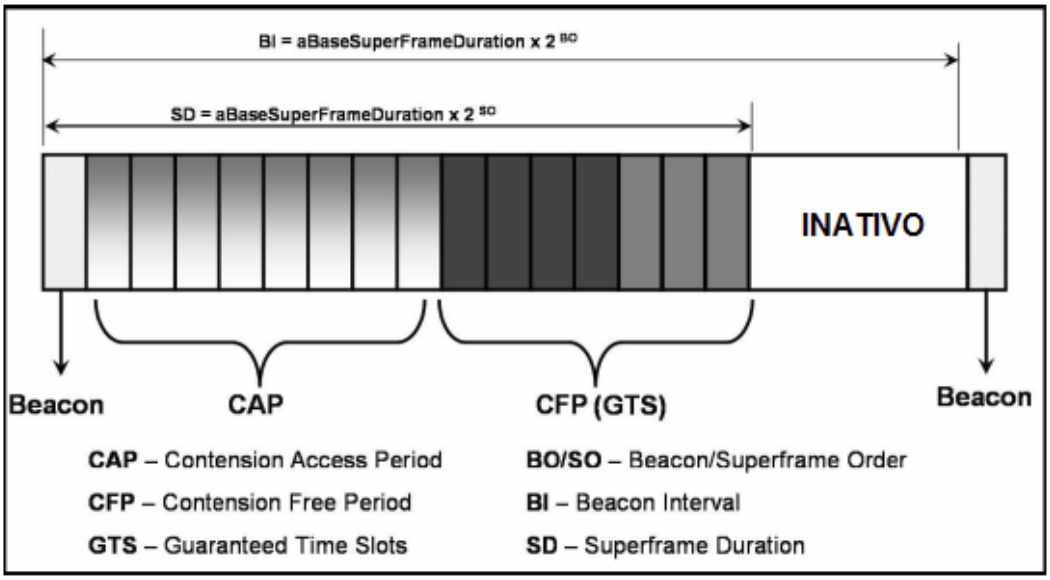
A camada MAC fornece a interface entre a camada física (PHY) e a camada de rede (NWK) e é responsável por gerar sinais de alerta e sincronizar esses sinais entre os dispositivos. A camada MAC também é responsável por fornecer controle de segurança e funções para redes na topologia estrela.

O controle de acesso ao meio define dois modos de operação, o modo *Beacon* e o modo *Non-Beacon*. No modelo *Beacon* os nós da rede só precisam estar ligados no momento da sinalização emitida pelo dispositivo com função de roteador, proporcionando assim, um consumo mínimo de energia. Para o seu funcionamento é usado a estrutura de *superframe*.

Um *superframe* é um formato de transmissão de doze quadros, delimitado por dois quadros *Beacon* e sua estrutura é opcional no padrão IEEE 802.15.4. Em um *superframe* pode existir até três tipos de períodos, o período de acesso de contenção (CAP), o período de livre

contenção (CFP) e o período inativo. A estrutura de um *superframe* (figura 3) é definida pelo coordenador e configurada pela camada de rede (NWK) que provê banda livre e baixa latência nas transmissões.

Figura 3 - Estrutura do *superframe*



Fonte: Carvalho, Anjos (Década provável [200-?]).

No modo *Non-Beacon* os nós da rede permanecem desligados por muito tempo, podendo então perder transmissões e atualizações de eventos importantes. Para solucionar o problema, uma possibilidade é fazer com que os dispositivos permaneçam a maior parte do tempo com seus receptores ligados, precisando com isso ser alimentado com baterias mais robustas e consumindo mais energia, o que compromete a principal característica da rede que é o baixo consumo de potência.

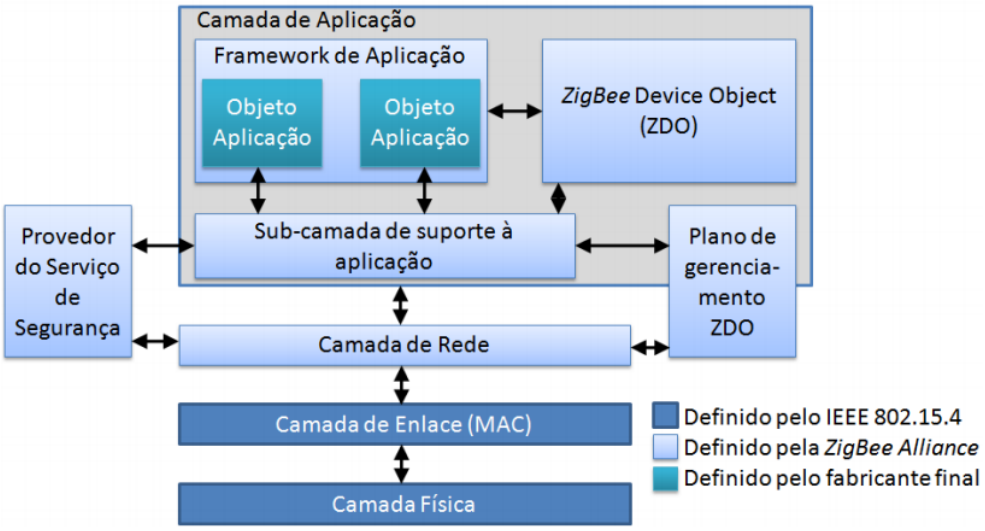
O mecanismo de acesso ao canal suportado pelo IEEE 802.15.4 é o *Carrier Sense Multiple Access with Colision Avoidance* (CSMA-CA). Existem dois tipos de CSMA-CA: *Slotted* e *Unslotted*. O *Slotted CSMA-CA* se refere ao desempenho CSMA-CA enquanto existir uma estrutura de *superframe*, já o algoritmo *Unslotted CSMA-CA* é utilizado quando não existe uma estrutura de superframe.

2.3.2 As camadas implementadas pelo protocolo zigbee

O *Zigbee* é um padrão que implementa as camadas de rede (NWK) e de aplicação (APL), logo, define uma série de protocolos de software de nível mais alto, fornecendo para a interface aplicações específicas. Como a versão proprietária do *Zigbee* foi criada para

trabalhar em conjunto com o padrão IEEE 802.15.4 existe uma confusão na apresentação dos padrões, onde muitas vezes são confundidos ou descritos como um único padrão, portanto, para um melhor entendimento das camadas que cada protocolo implementa a figura 4 pode ser visualizada.

Figura 4 - Camadas implementadas pelo *Zigbee* e pelo IEEE 802.15.4



Fonte: Silva (2011).

2.3.2.1 Camada de rede zigbee

A camada de rede é a responsável por garantir o correto funcionamento da subcamada MAC e proporcionar uma interface de serviço para a camada de aplicação. “A camada NWK fornece dois tipos de serviço: administração e dados. ” (FARAHANI, 2008, p. 80).

A *Network Layer Data Entity* (NLDE) é a responsável pela transmissão de dados entre dois ou mais dispositivos, e também, por gerar a PDU em nível de rede, enviar essa PDU para um dispositivo através de um roteamento específico e realizar a segurança, garantindo a autenticação e confidencialidade da transmissão.

A *Network Layer Management Entity* (NLME) realiza a administração dos serviços, permitindo ao dispositivo coordenador configurar novos dispositivos, estabelecer uma nova rede, entrar, reentrar e sair de uma rede, descobrir rotas e realizar roteamento, atribuir um endereço de rede de 16 bits para cada dispositivo na rede e realizar a descoberta de vizinhos a um salto de um dispositivo.

Os mecanismos de comunicação podem ser dos tipos, *broadcast*, *multicast* e *unicast*. Uma mensagem *broadcast* é enviada para qualquer dispositivo que receba mensagens, já o mecanismo *multicast* envia mensagens para um grupo específico na rede, enquanto uma

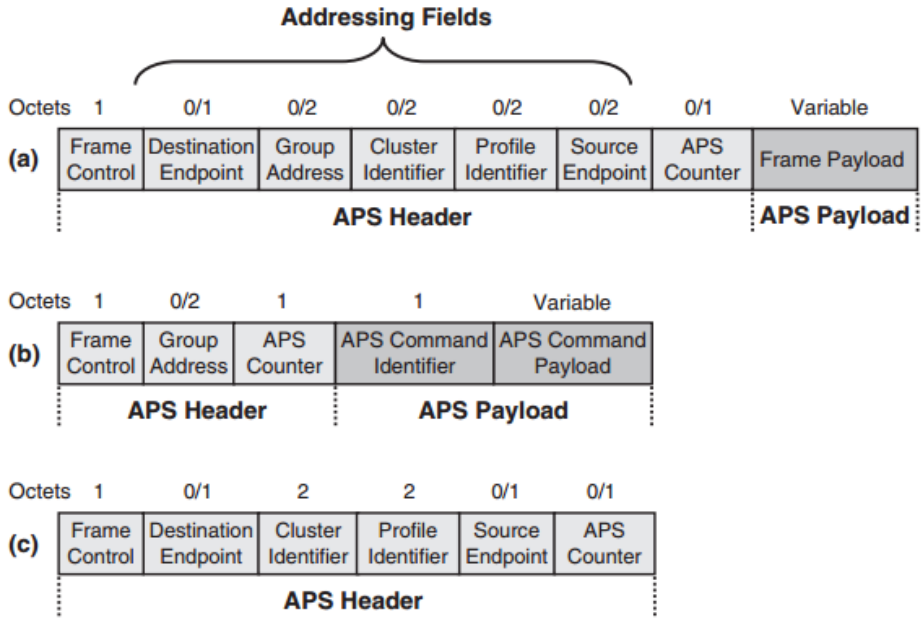
mensagem unicast é enviada para um único dispositivo, logo, é necessário se ter o endereço específico do dispositivo desejado. Se não for especificado nenhum padrão, o *unicast* será o modo de comunicação padrão.

2.3.2.2 Camada de aplicação zigbee

Na camada de aplicação *Zigbee* (APL) estão inseridas, a subcamada de suporte à aplicação (APS - *Application Support*), o *Zigbee Device Objects* (ZDO) e o *Framework* de Aplicação. A subcamada de suporte à aplicação (APS) proporciona uma interface entre a camada de rede (NWK) e a camada de aplicação (APL) e assim como todas as camadas inferiores, suporta os serviços de dados e gerenciamento. O serviço de dados é fornecido pela APS *Data Entity* (APSDE) e acessado pelo APSDE *Service Access Point* (SAP), já os recursos de gerenciamento são oferecidos pelo APS *Management Entity* (APSME) e acessados através do APSME-SAP. A subcamada APS fornece serviços através da APS *Data Entity* (APSDE), tanto para o *Framework* de Aplicação quanto para o ZDO. O serviço fornecido pelo APSDE é o de recebimento de dados a partir de qualquer ZDO ou *Framework* que tem que ser transmitido sobre a forma de uma PDU (*Protocol Data Unit*). Depois de recebe-los o APSDE adiciona cabeçalhos apropriados para a PDU e o transmite para a camada de rede.

Na subcamada APS existe uma outra maneira de se entregar a mensagem, que não utiliza as formas convencionais que já foram tratadas na seção 2.3.2.1 (*unicast, multcast e broadcast*). Essa forma de entrega é conhecida como encaminhamento indireto, onde um dispositivo que está vinculado com outros dispositivos em uma rede, pode enviar uma mensagem sem ter o endereço de destino desejado. Existem três tipos de quadros APS, são eles: quadro de dados, quadro de comandos e quadro de reconhecimento. O subcampo de entrega define as opções de transmissão. Se o modo de entrega for o de reconhecimento indireto, o subcampo de endereço deve ser omitido do quadro, se ele vier definido como um, significa que o destino é o coordenador *zigbee* e se ele estiver definido como zero, significa que ele utilizará o coordenador *zigbee* para enviar a mensagem. Os três tipos de quadros APS podem ser visualizados na figura 5.

Figura 5 - Tipos de quadro APS



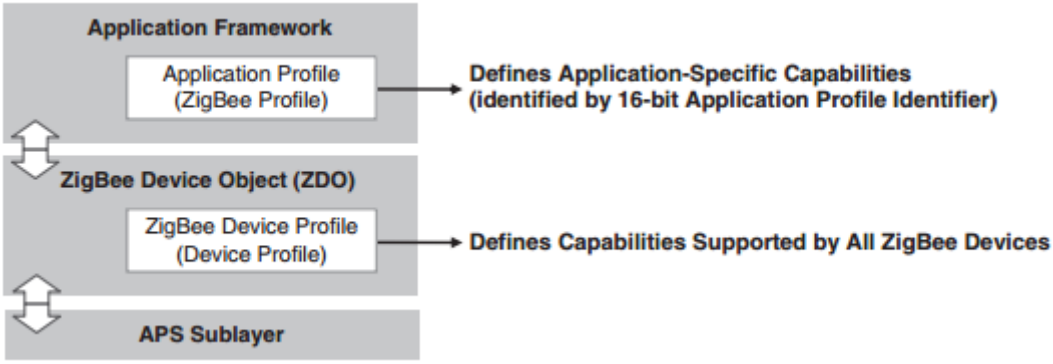
Fonte: Farahani (2008).

A *Zigbee Device Objects* (ZDO) fornece a interface entre a subcamada APS e o *Framework* de Aplicação. Ela possui as funcionalidades para configurar os dispositivos nos três tipos lógicos possíveis, o coordenador *Zigbee* - responsável por realizar a manutenção e a descoberta de roteadores na rede - o roteador *Zigbee* - que além de realizar a descoberta de roteadores, faz a associação com um coordenador permitindo que outros dispositivos se associem à rede - e o dispositivo final *Zigbee*. Seu acesso a subcamada APS é feito através da entidade gerenciadora APSME-SAP.

"A ZDO é a responsável por inicializar o APS, a camada de rede e o provedor de serviço de segurança (SSP). " (FARAHANI, 2008, p. 117). Ele possui um perfil semelhante aos perfis definidos no Framework de Aplicação, esse perfil é chamado de *Zigbee Device Profile* (ZDP) e contém descrições de dispositivos e clusters, fornecendo suporte para a descoberta de serviços e dispositivos. A descoberta de dispositivos tem a capacidade de descobrir a identidade de outro dispositivo na rede, enquanto a descoberta de serviços tem a capacidade de realizar um pedido com informações detalhadas sobre outro dispositivo. Além disso o dispositivo pode também, solicitar uma lista de cluster de entrada e saída.

A figura 7 mostra o ZDO como uma interface entras as camadas APS e Framework de Aplicação.

Figura 6 - Camada ZDO como interface entre a camada APS e o Framework de Aplicação



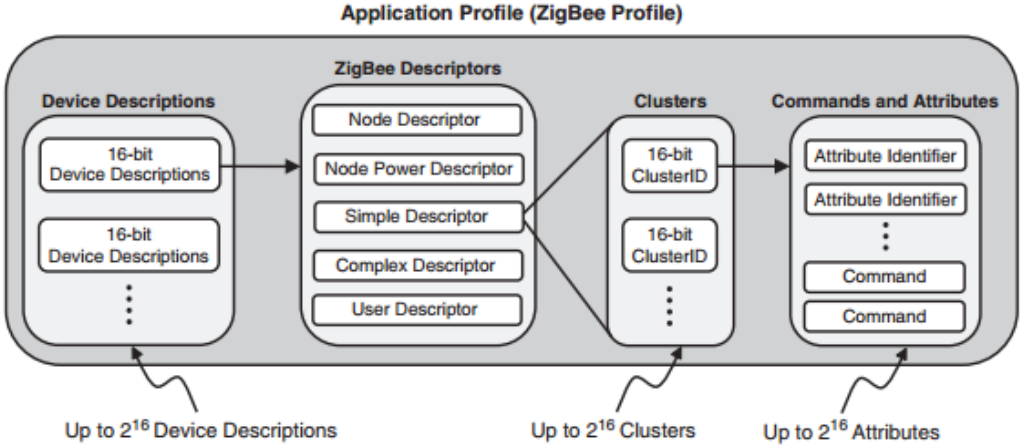
Fonte: Farahani (2008).

"O *Framework* de Aplicação é o ambiente no qual os objetos de aplicativos são hospedados para controlar e gerenciar as camadas de protocolo em um dispositivo *Zigbee*. " (FARAHANI, 2008, p. 111).

"A norma *Zigbee* oferece a opção de usar perfis de aplicação no desenvolvimento de um aplicativo. " (FARAHANI, 2008, p. 111). O uso de um perfil de aplicação permite a interoperabilidade entre produtos desenvolvidos por diferentes fornecedores. Cada perfil possui um identificador de 16 bits que só pode ser emitido pela *Zigbee allience*.

"O perfil do aplicativo consiste em dois componentes principais: clusters e descrições de dispositivos. " (FARAHANI, 2008, p. 112). Os *clusters* são um conjunto de atributos que se encontram agrupados e são utilizados para armazenamento de dados, já as descrições do dispositivo são as responsáveis pelas informações do próprio dispositivo. A descrição do dispositivo é formada por cinco seções, o nó descritor, o nó descritor de energia, e os descritores simples e complexo e o descritor de usuário. O nó descritor oferece informações como o tipo lógico (coordenador, roteador ou dispositivo final) e o código do fabricante. O nó descritor de energia irá determinar se o dispositivo está sendo alimentado por bateria e qual a duração da mesma, já o descritor simples irá fornecer o identificador de perfil e clusters. As informações como número de série e modelo do dispositivo irão estar contidos no descritor complexo, uma vez que este é opcional. As demais informações se encontrarão no descritor de usuário. A estrutura geral do perfil de aplicação pode ser visualizada na figura 7.

Figura 7 - Perfil da Aplicação



Fonte: Farahani (2008).

2.3.3 Serviços de segurança

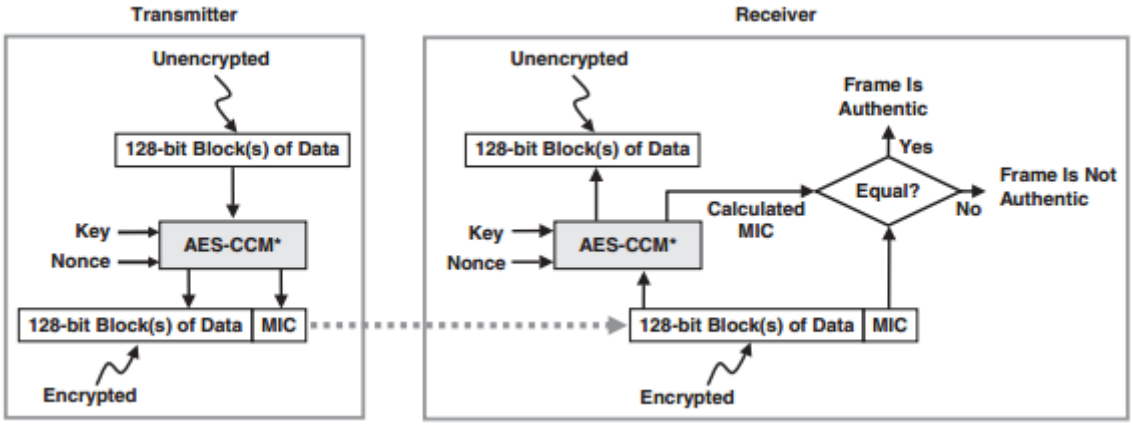
Em uma rede com aplicações simples, a segurança pode não ser um fator de grande preocupação. Como nas redes *Zigbee* a mensagem pode ser recebida por qualquer outro dispositivo *Zigbee* que se encontre nas proximidades será preciso ter uma atenção maior no quesito de segurança, pois um dispositivo poderá interceptar o envio da mensagem para ouvi-la ou até mesmo modificar e reenviar essa mensagem. Para evitar esses problemas, o padrão *Zigbee* utiliza os protocolos de criptografia e autenticação.

A criptografia é a prática de modificar uma mensagem, transformando-a em um código por meio da substituição das palavras, ou seja, é um conjunto de técnicas que permitem tornar incompreensível uma mensagem que foi escrita com clareza, para que somente o destinatário a consiga decifrá-la e compreender a informação. A norma *Zigbee* suporta o uso do padrão *Advanced Encryption Standard* (AES) - um algoritmo de criptografia de blocos adotado como padrão pelos Estados Unidos desde 1998 – e utiliza uma cifra de bloco de 128 bits.

A autenticação é o método que faz uso de senhas, certificados, números especiais e outros tipos de dados para efetuar um acesso a um dispositivo ou a uma rede. Na norma *Zigbee* existem dois tipos de autenticação: a de dispositivos e a de dados. Autenticação de dispositivo é o ato de confirmação em que um novo dispositivo se une a rede, sendo capaz de receber uma chave de rede e definir os atributos adequados dentro de um determinado período de tempo para se tornar autenticado, já na autenticação de dados, o receptor verifica se os dados foram alterados ou modificados. O objetivo da autenticação de dados, é saber se algum dado foi modificado durante o percurso. Para isso é preciso que o transmissor acompanhe o

quadro com um código especifico, o qual é conhecido como *Message Integrity Code* (MIC). A MIC será gerada por um método conhecido entre o transmissor e o receptor. No receptor, o procedimento será repetido e se o valor do MIC corresponder ao valor que foi enviado, será realizada a autenticação. No Zigbee o MIC é gerado pelo protocolo *Cipher Block Chaining Message Authentication Code* (CCM*), que é definido para ser utilizado em conjunto com os 128 bits do AES e também utiliza a mesma chave de segurança. A figura 8 mostra o papel da AES-CCM na autenticação de dados e confidencialidade.

Figura 8 - Aplicação do Código de Autenticação de Mensagens (MIC) na autenticação de dados



Fonte: Farahani (2008).

2.3.4 Topologias zigbee

As redes *Zigbee* suportam três tipos de topologias, que são as redes em estrela, arvore e malha. A rede estrela é a mais simplificada, sendo constituída por um coordenador central e vários dispositivos finais e não apresenta grande alcance, não sendo também recomendado usá-la em locais com muitos obstáculos, já as topologias em arvore e malha são mais adequadas para grandes redes, distribuídas por grandes áreas.

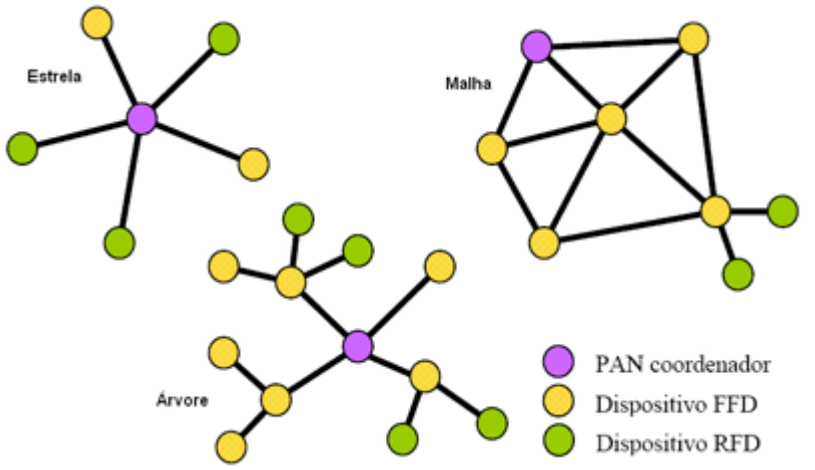
O *Zigbee* suporta três tipos de dispositivos, o coordenador e o roteador, os quais tiveram suas funções informadas anteriormente na seção 2.3.2.2 (Camada de Aplicação *Zigbee*), e o dispositivo final, que passa a maior parte do tempo em modo *sleep*, por não pode executar a descoberta de roteadores, nem permitir a associação de outros dispositivos a rede.

A topologia em arvore apresenta apenas um caminho para envio da mensagem entre os pares comunicantes, o que torna o roteamento muito simples e ela aparenta ser melhor que as demais topologias, mas por não ter caminhos redundantes, ela não é robusta em casos de falha.

A topologia em estrela é formada por nós que transmitem os dados para os nós de nível mais baixo até chegar no coordenador, que se encontra no nível zero. O procedimento de formação da topologia é iniciado pelo coordenador que irá transmitir pacotes *beacon* para nós próximos, quando um nó recebe o pacote, ele pode solicitar ao coordenador para juntar-se a rede e se o coordenador aceitar, será feito o envio periódico de pacotes *beacon*.

Na topologia em malha existem vários caminhos virtuais entre os pares de nós, o que requer o uso de protocolos eficientes de roteamento, tornando a rede mais flexível e robusta, para que, a informação não seja perdida caso algum nó esteja inoperante ou apresente alguma falha. Ela opera sobre a banda de 2,4 GHz do IEEE 802.15.4, já que as outras bandas não oferecem taxa de transmissão suficiente. Na topologia em malha não existe uma hierarquia como na topologia em arvore, mas os dispositivos podem ser conectados, diretamente ou pelo roteador *Zigbee*, com outros dispositivos na rede. Existe mais de um caminho para cada par de dispositivos na rede, para o caso de mudanças no ambiente, onde o dispositivo pode encontrar um caminho alternativo. O coordenador é o responsável por iniciar a rede e, junto com o roteador, pela descoberta e administração de rotas. Para realizar a descoberta de rotas o dispositivo de origem envia um pacote *Route Request* (RREQ), depois de recebido o pacote pelo dispositivo de destino, será feito uma replicação como uma mensagem unicast do pacote *Route Reply* (RREP) para o dispositivo de origem. As três topologias podem ser visualizadas na figura 9 abaixo.

Figura 9 - Topologias Zigbee



Fonte: Carvalho, Anjos (Década provável [200-?]).

2.3.5 Módulos zigbee

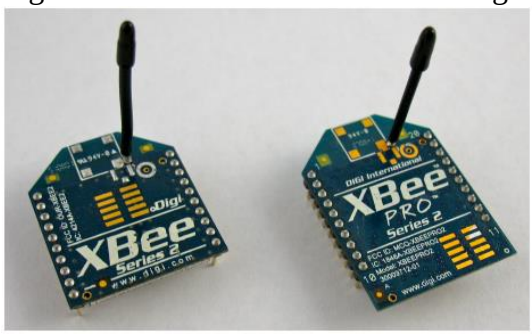
Os módulos são fabricados pela *Digi Internacional* e existem uma gama de rádios com diferentes componentes de *hardware*, protocolos de *firmware*, potência de transmissão e

opções de antenas. Os módulos podem ser alimentados por bateria e possuem pinos de entrada e saída.

Um módulo muito utilizado é o *XBee*. Existem dois tipos básicos desse módulo, o *XBee Series 1* e o *XBee Series 2*. O *XBee Series 1* utiliza um *microchip* fornecido pela *Freescale*, enquanto o *XBee Series 2* utiliza um *microchip* da *Ember Networks*, as duas séries possuem dois tipos de potência de transmissão, Regular e PRO, como pode ser visto na figura 10. A versão PRO, apresenta uma maior potência e consequentemente um maior custo.

Os módulos *XBee Series 1* são mais populares e de ótima implementação para sistemas de menor porte e substituição de simples cabos, enquanto o *XBee Series 2* suporta o protocolo *Zigbee* completo e é projetado para redes de sensores maiores.

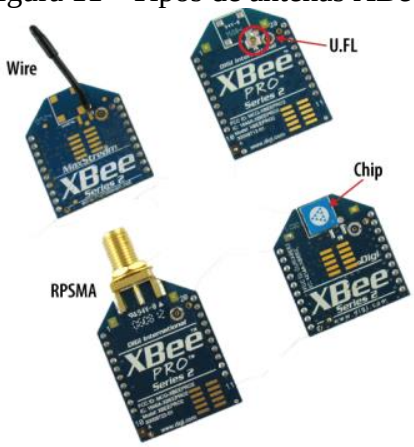
Figura 10 - Módulos *Xbee Series 2* Regular e PRO



Fonte: Faludi (2011).

Para a transmissão e recebimento de dados, os rádios precisam de uma antena, onde cada tipo possui vantagens e desvantagens. Os tipos de antenas disponibilizadas pela *Digi* são: *Whip* ou *wire antenna*, *Chip antenna*, *PCB antenna*, *U.FL connector* e *RPSMA conector*, as quais podem ser visualizadas na figura 11 abaixo.

Figura 11 - Tipos de antenas XBee



Fonte: Faludi (2011).

2.3.6 Aplicações zigbee

O *Zigbee* pode ser implementado em uma quantidade diversificada de aplicações, tais como, automação residencial, controle comercial e cuidado médico pessoal, além de aplicações que necessitam da longa duração das baterias, baixo custo e pequenos circuitos. O *Zigbee* pode ter uma significativa importância em situações em que ainda não existe o cabeamento no ambiente já existente, em uma rede experimental e quando o custo para o projeto for um fator determinante.

A automação residencial ou domótica como também é conhecida é uma das áreas de maior aplicação para redes *Zigbee*. Em um sistema de segurança por exemplo, o *Zigbee* pode ser utilizado, simplificando a instalação e modernizando o sistema. Um outro exemplo é um sistema automático de leitura de medidores, que oferece a oportunidade de ser monitorado remotamente o uso da água e energia elétrica de uma residência e poderá ser entregue além do uso mensal, informações detalhadas como vazamentos e problemas com equipamentos. Além dos exemplos anteriormente citados, pode-se ter ainda um sistema de irrigação, onde se teria um gerenciamento mais eficiente da água, um sistema de controle de luzes ou um sistema HVAC (*heating, ventilation, and air-conditioning*), para ajudar a economizar energia, controlando o fluxo de ar para cada quarto e evitando arrefecimento ou aquecimento de áreas desnecessárias.

Na automação industrial, o *Zigbee* pode ser utilizado no gerenciamento de energia, controle de luz, controle de processos e gerenciamento de ativos.

No setor de saúde uma das utilizações do padrão, seria o acompanhamento de pacientes remotamente, onde nesse caso pode ser utilizado a rede *Zigbee* para recolher dados de vários sensores conectados ao paciente.

Além de todas essas utilidades do *Zigbee* em diversas áreas, ele ainda pode ser utilizado no sistema para acesso a quartos, substituindo os tradicionais cartões-chaves e reduzindo o custo de instalação por não necessitar de fios, além do monitoramento de extintores de incêndio, indicando se está funcionando corretamente e se a manutenção é necessária.

2.4 O SENSOR DE TEMPERATURA E UMIDADE DHT11

O sensor de temperatura e umidade DHT11, é um sensor composto, que possui uma saída de sinal digital calibrada e faz uso de um componente de medição de umidade do tipo

resistivo e um componente de medição de temperatura NTC (*Negative Temperature Coeficient*), o qual se conecta a um microcontrolador de 8-bits de alto desempenho para verificar o ar ambiente, oferecendo uma resposta rápida e capacidade de anti-interferência.

Existe uma interface serial *single-wire* que realiza a integração do sistema com facilidade e rapidez. Como o sensor possui um tamanho reduzido, além do baixo consumo de energia e do sinal de transmissão - que pode chegar até 20 metros – ele acaba se tornando uma ótima opção para várias aplicações.

2.4.1 Características do sensor DHT11

O sensor DHT11 possui como principais características o baixo custo, estabilidade a longo prazo, excelente medição de temperatura e umidade relativa, saída de sinal digital, transmissão de sinal a longa distância, baixo consumo de energia e quatro pinos empacotados e totalmente permutáveis. Uma outra vantagem é que não será necessário o uso de componentes adicionais, a única desvantagem será a demora com que o sensor transmite os dados, sendo preciso esperar dois segundos para obter novos dados.

O sensor permite fazer leitura de temperaturas entre 0 e 50º Celsius com uma precisão de mais ou menos 2º Celsius e umidade entre 20 e 90% com precisão de mais ou menos 5%. Um resumo pode ser visto na tabela 2 abaixo.

Quadro 2 - Especificações de temperatura do Sensor DHT11

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 ℃	±5%RH	±2℃	1	4 Pin Single Row

Fonte: D-Robotics (2010).

A alimentação do sensor varia entre 3V e 5.5V e possui uma corrente que vai de 0.5 mA até 2.5 mA.. Um resumo das características elétricas pode ser visualizado na tabela 3 abaixo.

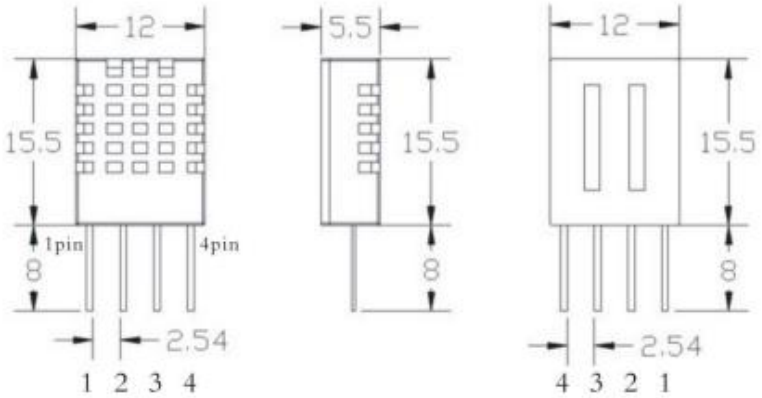
Quadro 3 - Características elétricas do Sensor DHT11

	Conditions	Minimum	Typical	Maximum
Power Supply	DC	3V	5V	5.5V
Current Supply	Measuring	0.5mA		2.5mA
	Average	0.2mA		1mA
	Standby	100uA		150uA
Sampling period	Second	1		

Fonte: D-Robotics (2010).

O sensor possui dimensões reduzidas de 15,5 X 12 mm e mais 8 mm para os terminais, como pode ser visto na figura 12 abaixo.

Figura 12 - Dimensões do Sensor DHT11



Fonte: Aosong Eletronics (2013).

2.4.2 Aplicações do sensor DHT11

O sensor DHT11 pode ser aplicado em diversas áreas, podendo ser utilizado em sistemas HVAC (*heating, ventilation, and air-conditioning*), coletores de dados, desumidificadores, eletrodomésticos, estações meteorológicas, reguladores de umidade, na área médica, ou qualquer outro aplicativo que realize o controle ou medição de umidade e temperatura.

2.5 O DRONE

Drone é um termo que está sendo utilizado com uma maior frequência para designar um veículo que possui autonomia de voo, ou seja, não possui um piloto interno. Essa

popularização da palavra gera muitas discussões, onde alguns profissionais defendem que somente os multirotores deveriam ser denominados *Drones*, devido ao som similar ao zumbido, produzido pelo equipamento que é constituído de hélices. No brasil são utilizados ainda outros termos como VARP (Veículo Aéreo Remotamente Pilotado) ou RPAS (Sistemas de Aeronaves Remotamente Pilotadas) e ainda UAV (Unmanned Aerial Vehicle).

Um *Drone* é um objeto voador, semelhante a um helicóptero com dimensões reduzidas e os modelos mais comuns possuem quatro hélices e são denominados quadricópteros, mas existem modelos com um número maior de hélices.

> *Drone* é um veículo aéreo não tripulado (VANT) dotado de uma controladora de voo que pode receber comandos através de rádio frequência, infra vermelho e/ou carregar missões previamente definidas por coordenadas GNSS (Global *Navigation Satellite System*) de uma constelação qualquer (ex.:GPS). (DOCTOR DRONE, 2015)

Quanto ao tipo de aplicação os *Drones* podem ser considerados como civis e militares e os *Drones* militares compreendem as aeronaves com mais de 150 quilos. A ANAC (Agência Nacional de Aviação Civil) tem uma proposta de regulamentação que será dada a partir do peso da aeronave e pelo modelo de voo, que podem ser dos tipos VLOS (aonde o piloto tem contato visual constante com o *Drone*) ou BLOS (voo além da linha de visada). A tabela 4 abaixo mostra a classificação e a liberação de uso proposta pela ANAC.

Quadro 4 – Classificação e liberação de uso de *Drones* pela ANAC

	Indoor		Área privada aberta		Área pública aberta		Áreas desabitadas	
	Área Privada	Área Pública	Até 400ft VLOS	> 400ft, BVLOS	Até 400ft VLOS	> 400ft, BVLOS	Até 400ft VLOS	> 400ft, BVLOS
Aeromodelo	Básica	X	Básica	X	1	X	Básica	X
RPA 25kg	Básica	1	Básica	1	1	X	Básica	1
RPA 25-150kg	X	X	2	2	X	X	2	2
RPA > 150kg	X	X	3	3	X	X	3	3
VANT autônomo	X	X	X	X	X	X	X	X

Legenda:
Básica: praticamente nenhuma exigência para operar;
1: nível de exigência baixo;
2: nível de exigência intermediário;
3: nível de exigência alta; e
X: operação proibida.

Fonte: Dr Drone (2015).

2.5.1 Nomenclaturas utilizadas

Existem várias nomenclaturas para o "*Drone*", sendo usado até para definir robôs táticos terrestres não tripulados. John M. Glover, autor do livro *Drone University*, traz as nomenclaturas mais usadas, que serão citadas abaixo.

RC *Airplane* – Apenas um avião simples de controle remoto.

FPV (*First Person View*) – Um avião de controle remoto, mas com a adição de uma transmissão de vídeo ao vivo.

Mini UAV – UM FPV acrescido de um piloto automático semiautônomo.

UAV – Termo genérico e antigo para veículos aéreos autônomos não tripulados, o qual é chamado agora de UAS.

UAS – Um termo não genérico que é utilizado para descrever todo o equipamento de solo, formação e documentação. O UAS geralmente é operado por mais de duas pessoas.

Cada um desses veículos é ligeiramente diferente, mas todos têm como base um avião RC básico ou helicóptero, o que faz com que um pequeno avião RC e um UAS militar tenham sistemas semelhantes, onde, em alguns casos, o mesmo equipamento esteja dimensionado apenas com servos e motores maiores.

No Brasil se tem as terminologias *Drone*, VANT, RPA e RPAS que são designadas pela Força Aérea Brasileira. O termo *Drone* é um termo genérico aplicado a qualquer objeto voador não tripulado, sendo utilizado para qualquer fim (profissional, recreativo, militar, comercial, etc.) e foi originado nos EUA (*Drone* = zangão, zumbido em português).

A terminologia oficial utilizada pelos órgãos reguladores brasileiros do transporte aéreo é VANT (Veículo Aéreo Não Tripulado). Apesar dos VANTs também serem veículos aéreos não tripulados, eles se diferenciam dos *Drones* pelo seu propósito. “No Brasil, segundo a legislação pertinente (Circular de Informações Aéreas AIC N 21/10), caracteriza-se como VANT toda aeronave projetada para operar sem piloto a bordo. Esta, porém, há de ser de caráter não recreativo e possuir carga útil embarcada. ” (DOCTOR DRONE, 2015).

Existem dois tipos diferentes de VANT, o RPA (Remotely-Piloted Aircraft / em português, Aeronave Remotamente Pilotada) e o RPAS (Remotely Piloted Aircraft Systems). O RPA tem o seu controle feito através de uma interface remota qualquer (computador, simulador, dispositivo digital, controle remoto, etc.), diferente da categoria denominada de “Aeronave Autônoma”, onde não é necessário o controle por meio da intervenção de um piloto, toda a programação da aeronave é feita antes de ser iniciado o voo. Esse tipo de aeronave não é permitido no Brasil. O termo RPAS deve ser usado quando se refere ao sistema de um RPA, ou seja, quando se refere a todos os recursos do sistema da aeronave: a estação de pilotagem remota, o link ou enlace de comando que possibilita o controle da aeronave, seus equipamentos de apoio, etc.

2.5.2 Sistemas que compõem um drone

Os *Drones* são compostos por várias partes que vão desde da sua estrutura (frame) até o sistema da placa principal. Alguns dos principais sistemas serão descritos nessa seção.

2.5.2.1 Sistema da estrutura

A estrutura de um *Drone*, nada mais é do que o seu chassi estrutural, que contém todos os seus componentes e fornece sustentação. As estruturas dos aviões são feitas com um tipo de espuma muito forte e durável. Os materiais que compõem a estrutura de aviões RC são feitos de espumas ou plástico corrugado, já os multirotores são constituídos de fibra de carbono, fibra de vidro ou plástico.

Os tipos de espumas podem ser: EPP, EPO, EPS e XPS. A espuma EPP (*Expanded Polypropylene foam*) é um tipo de espuma geralmente utilizada na embalagem de produtos eletrônicos, é durável e flexível, mas poucas aeronaves são fabricadas com esse tipo de espuma. A espuma EPO (*Expanded Polyolefin foam*) é uma espuma similar a EPP, sendo que este tipo de espuma é mais rígido. É o segundo melhor tipo de espuma e a maioria das aeronaves atuais são fabricadas com a espuma EPO. A espuma EPS (*Expanded Polystyrene foam*) é um tipo de espuma muito frágil e leve o que pode causar rupturas ou amassados no caso de impacto. A espuma XPS (*Extruded version of EPS*) é um tipo de espuma de painel isolante azul ou rosa que também é frágil leve e pode sofrer amassados ou rupturas com algum impacto. É também conhecida como Depron. O plástico corrugado (nome comercial *Coroplast*) é um produto de plástico utilizado para cobertura de baterias e em secções traseiras da aeronave. A fibra de carbono (CF) é um material extremamente leve e com bastante durabilidade, mas ocorre uma interferência com antenas e sinais de bloqueio. A fibra de vidro não apresenta o problema de interferência nas antenas e é também um material muito leve e semi-resistente.

2.5.2.2 Sistema de rádio controle e piloto automático

O sistema de rádio controle (RC) é efetuado através da ação humana, ou seja, existe uma pessoa que controla o *Drone* remotamente. Existem algumas frequências que podem ser utilizadas para o transmissor e o receptor RC. Para longo alcance é recomendado a utilização da faixa de frequência 443 MHz, enquanto para um curto alcance pode-se utilizar às faixas 2.4

e 5.8 GHz. A frequência 433MHz (UHF) LRS (*long range system*), se encontra em aparelhos de rádio locais onde o sinal passa ao redor de arvores e edifícios com muita facilidade. Com a utilização da frequência 2.4 GHz é possível atravessar alguns objetos e arvores, mas existem interferências ambientais, já com a frequência 5.8 GHz, o sinal é perdido completamente ao passar ao redor de arvores e edifícios.

O sistema de piloto automático não possui um piloto, mas é importante que a aeronave se encontre no campo de visão do piloto, ou que seja utilizado um link de vídeo em caso de algum problema na parte física. O sistema inclui os acelerômetros, IMU (bússola), GPS, sensor de velocidade, sensores de temperatura e barómetro, bem como outros sensores opcionais. Para acionar o sistema, os sinais de rádio do receptor são enviados para o piloto automático, onde serão processados e, em seguida, os dados são emitidos para os servos e controladores de velocidade. Os acelerômetros que o piloto automático possui, servem para estabilizar e eliminar movimentos indesejados, compensando a turbulência. O GPS é um módulo *plug and play*, onde é preciso apenas verificar se a conexão dos fios PWR, TX, RX e GND está correta e a taxa de transmissão definida, a qual deve estar configurada para a taxa de transmissão mais elevada.

2.5.2.3 Sistema de energia

O uso de aviões elétricos começou na década de 1980 e foi se aperfeiçoando. Hoje as aeronaves já conseguem atingir uma certa distância em voo devido a evolução das baterias. Algumas das baterias mais utilizadas serão citadas abaixo.

A LiPo é a bateria utilizada em aviões RC para todos os componentes. É o mesmo tipo de bateria encontrada em laptops e celulares, muito leve e com uma densidade de energia muito boa, mas se ocorrer uma sobrecarga pode acontecer uma explosão ou incêndio.

A Lithium-Ion é uma bateria com pouco peso e boa densidade de energia, pode explodir em caso de sobrecarga, curto-circuito ou submersão em agua. Está presente em telefones celulares mais antigos, laptops e carros híbridos.

A LiFePoF4e é uma bateria leve, segura e com dimensões físicas maiores. Não é muito popular devido à baixa densidade de energia.

A NiCad é um modelo de bateria pesado, seguro e confiável, muito utilizado em ferramentas elétricas e aviões comercias. Esse modelo de bateria não é indicado a ser utilizado por um pequeno *Drone* do tipo FPV, devido ao peso.

A Lead Acid é também um modelo muito pesado, portanto não é recomendado sua utilização em um *Drone* do tipo FPV. Esse tipo de bateria poderia ser utilizado para alimentação da estação base.

A NiMh é um modelo de bateria leve e seguro, pode ser comparada ao modelo NiCad, mas possui uma maior densidade de energia e é o tipo de bateria ideal para rádios e equipamentos auxiliares.

2.5.2.4 Sistema de transmissão

Para esse sistema é preciso saber quanto peso a aeronave irá carregar e quanto tempo de voo ela irá ter.

O tipo de motor utilizado é *Brushless* devido a sua baixa manutenção e o tamanho desse motor é indicado na referência do mesmo. Existem os motores *Outrunner* e os *Inrunner*.

No modelo *Outrunner* o motor principal gira junto com o eixo, mas possui uma extremidade fixa para conexão. Geralmente é instalado na parte externa do aeromodelo e possui mais entradas de ar para ventilação, maior torque e menos eficiência, por que gira mais lento, produzindo mais força e aproveitando menos a energia. Os *Outrunners* são menores e mais leves, permitindo sua utilização em aeromodelos de menor porte, são mais silenciosos, possuem um menor tempo de resposta e uma manutenção mais barata e simples.

No modelo *Inrunner* o motor principal é fixo e somente o eixo gira, sua instalação é mais comum na parte interna, mas existem alguns modelos que ficam localizados na parte externa do *Drone* e por ser mais fechado sua ventilação é menor. Ao contrário do *Outrunner*, possui menor torque e uma maior eficiência, girando mais rápido, produzindo menor força e aproveitando melhor a energia. São maiores e mais pesados, mas barulhentos, possuem um tempo de resposta maior e a manutenção é mais cara e difícil.

Os ESC’s (*Electronic Speed Control*) são os circuitos que controlam a velocidade do motor. Para realizar o dimensionamento de um ESC será preciso obter a amperagem gerada pela combinação do motor e da hélice e deixar um espaço livre de 20% para evitar problemas com superaquecimento. Ele possui um circuito eliminador de bateria que produz 5v e é usado pelo sistema de rádio e um acoplador óptico para suas linhas de dados, que impedi que a alta tensão volte para o rádio.

2.5.3 A ANAC e a legislação dos drones

Devido a popularização dos *Drones* surgiu a falta de uma regulamentação específica, a qual, está sendo produzida pela Agência Nacional de Aviação Civil (ANAC). Segundo o site *Doctor Drone*, no presente momento um pequeno *Phantom* (Tipo específico de *Drone*) será considerado tanto como um aeromodelo, quanto como um RPA, a partir da sua utilização. Caso seja utilizado para fins recreativos, será considerado um aeromodelo sendo apenas necessário respeitar as normas vigentes para o Aeromodelismo. Se for utilizado para fins comercias passará a ser classificado como um RPA, ou seja, uma Aeronave remotamente pilotada e deverá possui um Certificado de Autorização de Voo Experimental (CAVE), além de cumprir normas da aviação civil.

O regulamento proposto pela ANAC, tem como foco principal tonar viável as operações, preservando a segurança, minimizando a burocracia e permitindo os possíveis avanços do regulamento com o desenvolvimento do setor. As regras que entrarão em vigor serão observadas para operações civis de VANT não autônomos (RPA) e aeromodelos não autônomos. A proposta divide todos os tipos de aeronaves remotamente pilotadas em três classes, classificadas de acordo com o peso.

As aeronaves que possuem peso maior que 150 kg, serão classificadas como classe 1 e deverão ser certificadas pela ANAC, ter registro no Registro Aeronáutico Brasileiro (RAB) e os pilotos deverão possuir Certificado Médico Aeronáutico (CMA), licença e habilitação. Além disso, todos os voos deverão ser registrados. As aeronaves que possuem peso menor ou igual a 150 kg e maior que 25 kg não precisam ser certificadas e se incluirão na classe 2, mas os fabricantes deverão observar os requisitos técnicos exigidos e também é necessário que o piloto tenha CMA, licença e habilitação, bem como a aeronave estar registrada no RBA. Nessa classe todos os voos também deverão ser registrados. As aeronaves que possuem peso menor ou igual a 25 kg estarão incluídas na classe 3 e não serão necessários CMA, nem registrar o voo, apenas o cadastro com os dados do operador e do equipamento, mas para isso será preciso pilotar a aeronave até 400 pés acima do nível do solo (aproximadamente 120 metros) em linha de visada visual e a uma distância de 30 metros de uma pessoa. “Em áreas urbanas e aglomerados rurais, as operações serão de no máximo 200 pés acima do nível do solo (aproximadamente 60 metros). ” (DOCTOR DRONE, 2015). Para aeromodelos, que são destinados a recreação, não haverá a necessidade de regulamentação junto a ANAC, deverá apenas ser observada a distância mínima de 30 metros de uma pessoa.

No dia 19 de novembro de 2015 o Departamento de Controle do Espaço Aéreo (DECEA), ligado ao Comando da Aeronáutica publicou a ICA 100-40, intitulada Sistemas De Aeronaves Remotamente Pilotadas e o Acesso ao Espaço Aéreo Brasileiro, que tem como objetivo regulamentar os procedimentos para o acesso seguro ao Espaço Aéreo Brasileiro por Sistemas de Aeronaves Remotamente Pilotadas (RPAS).

"As novas normas alteram apenas o comportamento no ar, mas não mexem no modo de solicitação junto às autoridades. " (DOCTOR DRONE, 2015). Para que os *Drones* sejam pilotados no Brasil é preciso se ter duas autorizações, uma da ANAC e outra da DECEA. Na ANAC é feito o registro do equipamento, enquanto na DECEA são realizados os pedidos para decolagem. Os novos parâmetros são aplicados apenas as atividades profissionais, institucionais e governamentais.

O documento fixa uma altura mínima de 30 metros de distância de prédios, casas, construções, veículos, animais e de concentração de pessoas que não estejam associadas à operação, além de uma velocidade máxima que varia entre 55 km/h e 110 km/h, dependendo do peso da aeronave. O documento pode ser acessado para maiores informações no link abaixo:

http://servicos.decea.gov.br/arquivos/publicacoes/dfe9dd11-67e7-4166-bb1b89e07af66313.pdf?CFID=8642aa76-55c5-4bfd-9ed2-3f34fe3cd19c&CFTOKEN=0

3 MONITORAMENTO DE TEMPERATURA E UMIDADE COM O USO DO PROTOCOLO ZIGBEE

Esta seção do trabalho descreverá o desenvolvimento realizado para se obter o resultado proposto, através de informações detalhadas da montagem do Drone e as configurações realizadas no módulo XBee e na placa Arduino.

3.1 MONTAGEM E CONFIGURAÇÃO DO DRONE

No primeiro momento foi montado um *Drone* para realização desse trabalho, um quadricóptero básico, constituído de um frame (estrutura), quatro motores, quatro controladores de velocidade (ESC), quatro hélices, uma placa de energia, onde ficam conectados os ESCs e o conector para a bateria, uma placa controladora e o módulo receptor do rádio controle, que recebe os sinais do rádio controle e repassa para os ESCs. Os controladores de velocidade são ligados aos motores e a placa controladora para realizar o controle da passagem de energia para os motores, fazendo com que estes se movimentem mais rápido ou mais devagar. Os ESCs são necessários em motores elétricos sem escovas (*brushless*). O Drone foi montado com esse modelo de motor e pode ser visualizado na figura 13.

Figura 13 - Drone montado para realização dos testes

Fonte: Elaborado pelo autor deste trabalho (2015).

A placa controladora é o principal componente dos *Drones* e possui como principal função o processamento dos sinais de entrada para gerar saídas adequadas para os componentes. A placa utilizada foi a KK *Multicoper*, uma placa para controle de voo de aeromodelos com até 6 rotores. A finalidade dessa placa é a estabilidade durante o voo, para isso é preciso ajustar o sinal dos três giroscópios na placa (roll, pitch e yaw). Essas informações serão salvas em um CI (circuito integrado) ATmega, que processa a informação de acordo com o software da KK *Multicoper* e envia um sinal de controle para os controladores de velocidade (ESC). A placa também transmite um sinal de controle do receptor de controle remoto (RX) e alimenta isto no CI através dos pinos *aileron*, *elevator*, *throttle* e *rudder*. Depois de processar essa informação o CI envia um sinal para os motores, para aumentar ou diminuir a aceleração e conseguir a estabilidade no voo, através dos comandos enviados pelo seu transmissor (TX).

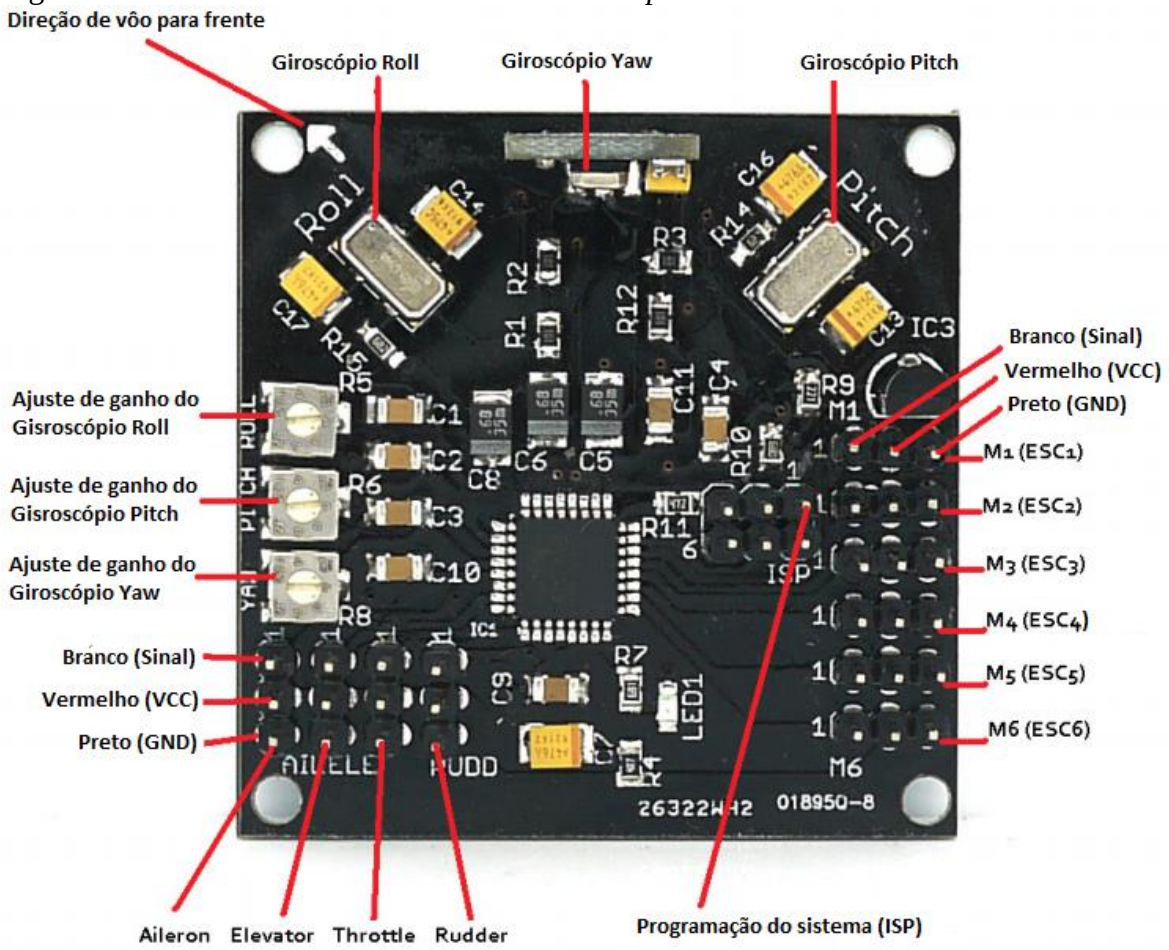
A placa possui um chip ATmega 168 que permite aos usuários fazer o *upload* de um *firmware* não padrão, possibilitando a configuração do seu próprio código. Para realizar a configuração do *firmware* é preciso ter um USB ASP Programador, que faz a conexão da placa controladora com o computador, e o programa *Flashtool*, que possui diversos firmwares para placas controladoras.

Para proteção da placa ao ser inserida no *Drone*, recomenda-se coloca-la sobre um material que amorteça as vibrações. Existe na placa uma seta na cor branca que indica a direção de voo para frente e também é recomendado que a placa seja posicionada nessa direção para saber qual a frente da aeronave.

A conexão do rádio é feita conectando os fios dos canais (CH1, CH2, CH3 e CH4), correspondentes aos pinos aileron, elevator, throttle e rudder, enquanto os controladores de velocidade ficam conectados nos pinos M1 a M6 na seguinte ordem dependendo da configuração do rotor de voo. A figura 14 mostra uma placa KK *Multicoper* e as respectivas ligações que devem ser realizadas.

Devido a problemas de instabilidade durante o vôo, não foi possível realizar os testes com o *Drone* que foi montado, sendo utilizado um Drone comercial.

Figura 14 - Placa controladora de voo KK *Multicopter*



Fonte: BAKKE, HERMANNSEN, BARTON (Década provável [200-?]).

3.2 CONFIGURAÇÃO DO MÓDULO XBEE PRO S1

Para realizar a comunicação utilizando módulos *XBee* é preciso optar por umas das séries (1 ou 2), porque elas não se comunicam entre si, devidos a algumas peculiaridades de cada série. O módulo utilizado para esse trabalho foi o *XBee* PRO série 1 que possui em suas especificações um alcance em locais abertos de 1,6 km e em ambientes fechados um alcance de 100 metros.

Para realizar a configuração do módulo é preciso ter a placa *XBee Explorer* e o software X-CTU. Existem três tipos de placas *Explorer*, todos projetados para criar um gateway de comunicação entre o computador e o *XBee*. As placas *Explorer* são as *USB Explorer*, *Explorer Dongle*, e *Serial Explorer*. O X-CTU é a interface de configuração do *Xbee*, sendo possível, a partir de sua utilização, configurar e gerenciar *XBees* e redes de teste *XBee*, realizar teste de comunicação e transmitir dados remotos entre o computador e o

módulo. Além disso é possível atualizar o firmware dos módulos, enviar comandos de um módulo *XBee* para o outro e até mesmo recuperar módulos com problema.

A placa utilizada no trabalho foi a USB *Explorer* que pode ser visualizada na figura 15.

Figura 15 - Placa USB *Explorer* para configuração do módulo *Xbee*

Fonte: Elaborado pelo autor deste trabalho (2015).

Após a instalação do software X-CTU e a conexão do módulo *XBee* na placa *Explorer* é possível realizar as configurações do transmissor e do receptor *XBee*. Primeiro é preciso abrir o programa e realizar uma busca por módulos no computador, depois será selecionada a porta serial que foi detectada durante o reconhecimento da placa *XBee Explorer*. Em seguida a esse procedimento o módulo será adicionado ao X-CTU e ao se clicar na imagem do módulo, serão apresentados os parâmetros a serem configurados os quais podem ser vistos nas figuras 16 e 17.

A configuração realizada para esse trabalho foi a ponto-a-ponto, portanto das configurações que são disponibilizadas no software, apenas cinco foram modificadas, o ID PAN ID, MY 16-bit *Source Address*, DH *Destination Address High*, DL *Destination Address Low* e NI *Node Identifier*.

ID PAN ID é o número de identificação da rede *XBee* ao qual o módulo irá se conectar, para se ter dois ou mais Xbees na mesma rede, basta configurá-los com o mesmo PAN ID.

MY 16-bit Source Address é um número único que identifica o módulo Xbee na rede.

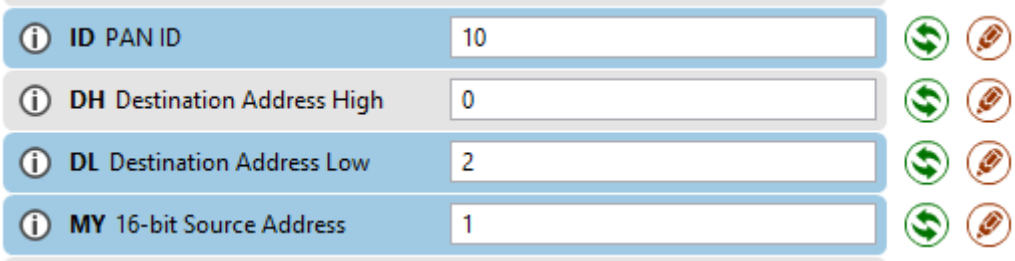
DH *Destination Address High* identifica os primeiros 32 bits do endereço dos 64 bits que podem ser atribuídos aos módulos *XBee*. Foi atribuído o valor 0, porque só foram utilizados dois módulos para realização dos testes.

DL *Destination Address Low* É a parte dos outros 32 bits restantes que será utilizada para colocar o endereço de destino. Como o parâmetro DH foi deixado em 0, foi utilizado o endereço de 16 bits especificado em MY 16-bit Source Address.

NI Node Identifier é o identificador do módulo, ou seja, o nome que aparece na identificação do rádio.

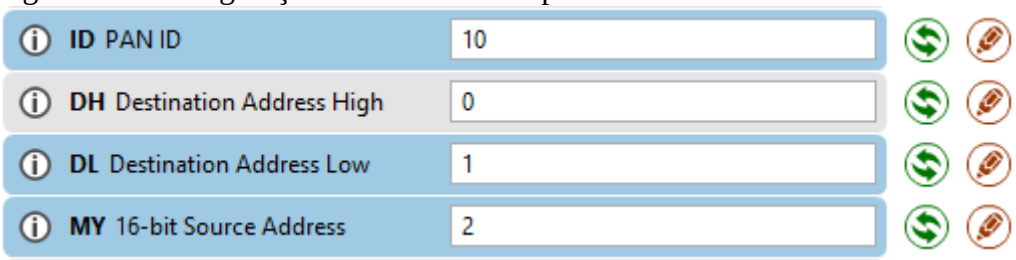
Após a realização dessas configurações é preciso escrever as alterações no módulo para concluir a configuração.

Figura 16 - Configuração do Módulo Emissor



Fonte: Elaborado pelo autor deste trabalho (2015).

Figura 17 - Configuração do Módulo Receptor



Fonte: Elaborado pelo autor deste trabalho (2015).

3.3 A PROGRAMAÇÃO NA PLACA ARDUINO

A programação realizada na placa *Arduino* foi executada para o envio dos dados coletados pelo sensor de temperatura DHT11 e enviá-los através do módulo *XBee* ao receptor que armazena os dados em uma tabela no Excel. Nessa seção será mostrada como foi feita toda a implementação.

Antes de começar a programação propriamente dita, é preciso baixar a biblioteca do sensor DHT11 e colocá-la na pasta *Libraries* da IDE do *Arduino*, que se encontra na pasta arquivos de programas (x86), após isso é iniciado o código. No emissor foi realizado a inserção das bibliotecas DHT e *SoftwareSerial*. Sendo esta última foi desenvolvida para permitir comunicação serial através de outros pinos digitais do *Arduino*, uma vez que o *hardware* dá suporte apenas aos pinos 0 e 1 e o *XBee* utiliza os pinos 2 e 3 para comunicação. Em seguida a definição do pino de sinal e qual o tipo de sensor, uma vez que existem três

tipos, DHT11, DHT21 e DHT22 e na linha seguinte são informados os pinos utilizados pelo *XBee*, como pode ser visto no trecho de código abaixo

```
#include <DHT.h>
#include <SoftwareSerial.h>

#define DHTPIN 4
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

SoftwareSerial XBee(2, 3);
```

Em seguida é realizado a leitura dos dados de temperatura e umidade e é criado a *string* dados e utilizado o parâmetro *concat* para realizar a concatenação dos dados e enviá-los através do *XBee*, como pode ser visto na última linha do código.

```
float h = dht.readHumidity();
float t = dht.readTemperature();

String dados ="";
dados.concat(h);
dados.concat("#");
dados.concat(t);
XBee.println(dados);
```

No receptor foi realizado a programação para os dados serem salvos em um arquivo do Excel. Para que o ocorra a comunicação será preciso incluir novamente a biblioteca *SoftwareSerial* criados uma string que irá armazenar os dados enviados pelo *XBee* e as variáveis que serão utilizadas como rótulos dos valores que serão mostrados na tabela, como pode ser visto na primeira parte do código.

```
#include <SoftwareSerial.h>

SoftwareSerial XBee(2, 3);

String inString = "";

float Umidade = 0;
```

```
float Temperatura = 0;
int LABEL = 1;
```

Em seguida será inserida a linha utilizada para realizar a limpeza da tela quando o programa for desligado ou reiniciado e a linha seguinte utilizada para nomear as colunas.

```
Serial.println("CLEARDATA");
Serial.println("LABEL,Hora,Umidade,Temperatura");
```

No trecho do código abaixo será realizada a verificação dos valores recebidos e enquanto o valor for maior que zero o emissor realiza a leitura dos dados e então verifica se o valor recebido não é uma quebra de linha, se não for, o valor continuará a ser guardado na *string*, caso seja o final da *string* o código executa a linha *Serial.print("DATA,TIME,");* que inicia a impressão dos dados. Para que os dados fiquem separados por coluna no *Excel* é acrescentado a linha *Serial.print(",");* e para a divisão da *string* são utilizadas as linhas *Serial.print(inString.substring*(0,2)) e *Serial.println(inString.substring*(6,8)); as quais utilizam o parâmetro *substring* para selecionar apenas as partes da *string* que devem ser mostradas na tela.

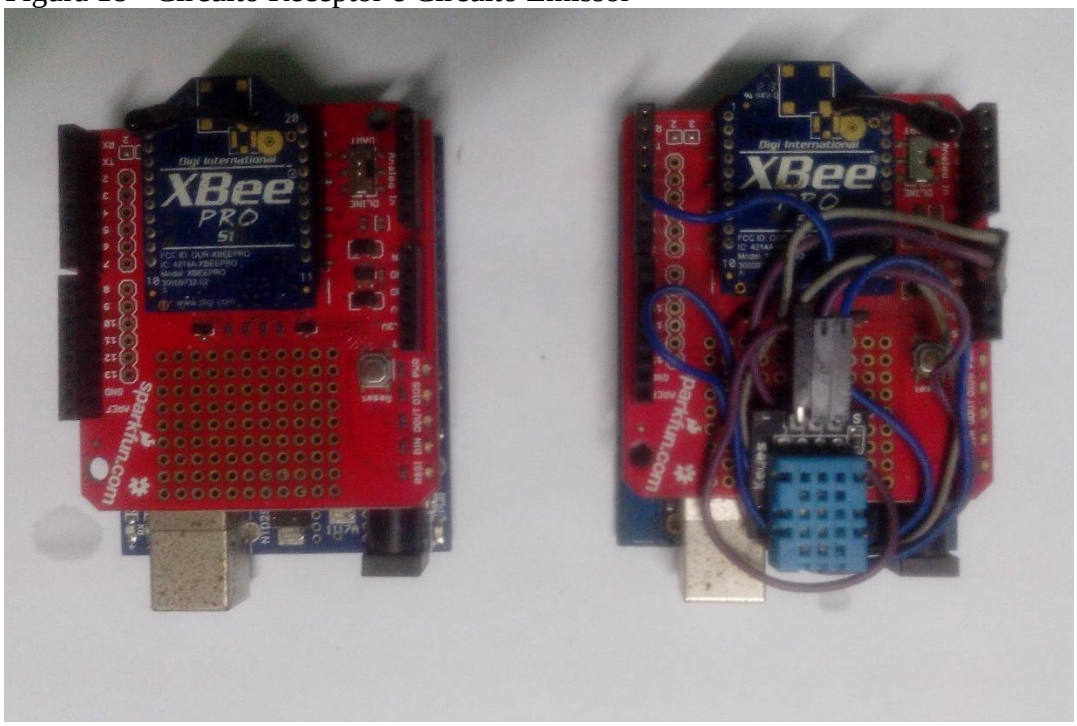
```
while (XBee.available() > 0) {
float inChar = XBee.read();

if (inChar != '\n') {

    inString += (char)inChar;
   }
   else {
    Serial.print("DATA,TIME,");
    Serial.print(inString.substring(0,2));

    Serial.print(",");

    Serial.println(inString.substring(6,8));

    inString = "";
   }
  }
}
```

Para fazer o gerenciamento entre a porta serial do *Arduino* e o *Excel* foi utilizado o programa PLX-DAQ. É preciso fazer a instalação no computador e para acessá-lo ir na pasta *Parallax Inc*, que se encontra na pasta arquivos de programas (x86). Para realizar o envio dos dados é preciso acrescentar no programa a porta serial do *Arduino* e verificar a velocidade de transmissão dos dados. Um último detalhe é que o programa só é compatível com a versão 32 *bits* do *Excel*, apresentando um erro antes de abrir o arquivo na versão 64 *bits*. O código completo pode ser visualizado no Apêndice A e B.

3.4 MONTAGEM DO CIRCUITO PARA OS TESTES

Após a realização da configuração do módulo *Xbee* e da inserção do código na plataforma *Arduino* foram encaixados os módulos *Xbee*, através das Shields no *Arduino*, bem como feito a conexão do sensor de temperatura ao *Arduino* que realiza o envio dos dados, ou seja, ao módulo Emissor. A figura 16 mostra o circuito do Emissor do Receptor.

Figura 18 - Circuito Receptor e Circuito Emissor



Fonte: Elaborado pelo autor deste trabalho (2016).

4 TESTES E RESULTADOS

Nessa seção serão descritos os testes realizados com o *Drone* para envio dos dados de temperatura e umidade através do módulo de comunicação *Xbee*.

Foram realizadas duas tentativas, na primeira foi utilizado o *Drone* comercial *Syma X5SW* que será descrito nas sessões seguintes, onde não foi possível realizar os testes por limitações do aeromodelo. Na segunda foi utilizado o *Drone* comercial *Quanum Nova*, onde foram realizados dois tipos de teste. O primeiro foi realizado em campo de visada do módulo receptor e o segundo foi feito com um obstáculo para ver qual o comportamento do envio dos dados coletados. Os testes foram realizados no estacionamento do IMD (Instituto Metrópole Digital) no dia 12 de março de 2016.

4.1 TESTES REALIZADOS COM O DRONE SYMA X5SW

O *Syma X5SW* possui um diâmetro de 31 cm e pode ser controlado em campo aberto até um rádio de 50 metros. Pesa 118g, dos quais 11 g são da câmera e possui quatro LEDS, dois laranjas e dois verdes que servem para indicar a orientação de voo, sendo que os verdes indicam a frente da Aeronave. Segundo Hugo Cura que realizou a análise do equipamento, o *Drone* possui autonomia de voo entre 7 e 8 minutos sem a utilização da câmera, ao colocar o dispositivo sua autonomia cai para um máximo que varia entre 6 e 7 minutos. A aeronave utiliza uma bateria de 500 mAh uma vez que fica embutida no *Drone* e o espaço é bastante reduzido.

Para realização desse teste foi acoplado ao *Drone* o circuito com uma bateria de 9 volts utilizada para alimentar a placa *Arduino.* Não foi possível levantar voo, uma vez que o peso do sistema era maior do que a potência do motor. Ao retirar a bateria foi possível realizar o voo, mas como não havia outra forma de alimentação para o sistema não foi possível realizar os testes de envio dos dados sendo necessário a utilização de um outro *Drone* comercial que será descrito na próxima sessão.

4.2 TESTES REALIZADOS COM O DRONE QUANUM NOVA

O *Quanum Nova* possui uma estrutura de plástico extremamente leve e possui luzes de LEDs frontais verdes e vermelhas na parte traseira para orientação do voo. Além disso ainda possui dois LEDs que indicação o status atual da bateria, onde passam a piscar e emitir sinal

sonoros quando a bateria estiver perto de descarregar. O *Drone* possui uma entrada mini USB para conexão ao computador e extração dos dados coletados, a partir da utilização do software compatível *MAVlink*, como por exemplo o *Mission Planner*. A aeronave possui uma autônima de voo aproximada de 15 minutos. Com esse *Drone* foi possível ser realizados os testes e a captura da altura em que o *Drone* conseguiu ainda comunicar-se, mesmo com perda de dados, com a base que se encontrava no chão. Esses dados podem ser visualizados nos anexos A e B.

4.2.1 Teste em campo aberto

Primeiramente foi realizado um teste onde o *Drone* estava no campo de visada do módulo receptor, chegando a uma altura de mais ou menos 13 metros e uma distância em linha reta de mais ou menos 42 metros. Nesse teste, o módulo *Xbee* PRO S1 começou a perder dados, uma vez que o modelo que está incluído na biblioteca do sensor DHT11 possui uma configuração para enviar os dados de temperatura e umidade a cada dois segundos, sendo adotada a mesma configuração na programação do *Arduino*, e em alguns momentos, quando o *Drone* se encontrava mais distante, o módulo emissor passou mais de dois segundos para transmitir os dados. A coleta foi feita diretamente para uma tabela no Excel com a utilização do programa PLX-DAQ. Os dados coletados com atraso podem ser vistos na tabela 5. A captura completa dos dados coletados e enviados pelo módulo podem ser visualizados no apêndice C.

Quadro 5 - Dados coletados pelo XBee e enviados para o receptor

Hora	Umidade	Temperatura
16:04:46	38	29
16:04:51	38	29
16:10:16	33	31
16:10:25	33	31

Fonte: Elaborado pelo autor deste trabalho (2016).

4.2.2 Teste sem campo de visada

No segundo momento, ainda durante a realização do primeiro teste, o Drone foi deslocado para a parte de trás do prédio do IMD, para analisar qual seria o comportamento do envio dos dados, nesse momento o módulo emissor perdeu totalmente a comunicação com a base, parando de transmitir os dados. Nesse teste a aeronave conseguiu uma altura de mais ou menos 20 metros e uma distância do módulo de mais ou menos 42 metros. Ao retornar o

Drone para o campo de visada do módulo receptor o envio de dados não pode ser reestabelecido, sendo necessário reiniciar o programa PLX-DAQ, que realiza o armazenamento dos dados no *Excel*, para poder ter novamente a conexão reestabelecida. Nesse momento a coleta de dados foi totalmente perdida, uma vez que reiniciado o programa, o mesmo realiza a limpeza da tabela, tendo que ser realizado um segundo teste em campo de visada do módulo receptor para coletar os dados novamente. Com esse teste pode ser observado que em um cenário onde tivéssemos a presença de vários obstáculos que impedissem a visada do módulo emissor até o módulo receptor haveria a necessidade de salvar os dados da coleta realizada, retornar o *Drone* até o local onde ele perdeu totalmente a comunicação e então reiniciar o programa para efetuar nova coleta.

5 CONSIDERAÇÕES FINAIS

Essa seção irá descrever às conclusões do relatório, que podem ser tiradas a partir da prática realizada, apresentando os conhecimentos adquiridos e as dificuldades encontradas para a produção desse trabalho.

5.1 CONCLUSÕES

O trabalho realizado se consistiu no envio de dados de temperatura e umidade através da utilização da comunicação sem fio *Zigbee*, a partir da utilização de um *Drone* para auxiliar em serviços demorados ou em locais com difícil acesso aos seres humanos.

Com os testes realizados pode-se verificar que o módulo *Xbee* PRO S1, o qual é o responsável pelo envio dos dados, apresentou perda de dados e perda total da comunicação quando submetido ao teste sem campo de visada com o módulo receptor, não retornando a enviar os dados até que o software fosse reiniciado, apresentando uma certa dificuldade em relação ao objetivo principal do trabalho, que seria a varredura de locais com acesso perigoso aos seres humanos e provavelmente com diversos pontos em que o módulo emissor não teria visada para o módulo receptor.

Portanto o módulo *Xbee* PRO S1 não se demonstrou eficaz para realizar o trabalho proposto, uma vez que o objetivo principal desse trabalho seria o de coletar dados de forma mais eficiente, de áreas de difícil acesso e enviá-los para uma base remota, gerando uma tabela para futuros estudos ou comparações de possíveis mudanças de temperatura e umidade do local da coleta. Assim podemos concluir que para esse tipo de coleta poderiam ser testados outros módulos *Xbee* que apresentam um maior alcance ou outra tecnologia como a dos módulos transmissores RF (Rádio Frequência), que possuem em suas especificações um alcance de até 200 metros sem obstáculos e operam a uma frequência de 443 MHz. Para alcançar essa distância é necessário a utilização de antenas, uma vez que pode ser possível agrega-las ao módulo. Uma outra opção para um alcance ainda maior seria a utilização do módulo Rádio *Wireless* Apc220, que permite um alcance de até 1.000 metros em campo aberto e trabalha a uma frequência de operação que varia entre 418 Mhz e 455 Mhz

5.2 DIFICULDADES ENCONTRADAS

A primeira dificuldade encontrada para a realização desse trabalho envolveu a programação da placa de prototipagem *Arduino*, uma vez que esse tipo de sistema não era conhecido e existiam várias limitações com o conteúdo a ser programado, gerando assim um segundo problema, também relacionado a parte da programação, o tratamento dos dados durante o envio. Após a correta inserção do código, o mesmo, apesar de não apresentar erros, não realizava o envio dos dados, não sendo demonstrados na IDE do *Arduino*, onde se encontrava o módulo receptor.

As demais dificuldades foram na utilização do *Drone* montado para os testes e consequentemente, obter um novo *Drone* para realização dos testes, uma vez que não se tinha conhecimento de ninguém que possuísse a aeronave. Existindo ainda, o problema do segundo *Drone* não levantar voo com o peso do circuito acoplado a ele.

REFERÊNCIAS

ALLIANCE, Zigbee. **WHAT IS ZIGBEE?** 2015. Disponível em: <http://www.zigbee.org/>. Acesso em: 06 out. 2015.

ARDUINO. **What is Arduino?** 2016. Disponível em: <http://www.arduino.cc/>. Acesso em: 05 out. 2015.

BAKKE, Rolf; HERMANNSEN, Jussi; BARTON, Mike. **KKmulticontroller v.5.5 "Blackboard"**: The Multicopter Flight Controller. Década provável [200-?]. Disponível em: <http://www.robotshop.com/media/files/pdf/user-manual-kk-control-package.pdf>. Acesso em: 22 jan. 2016.

BURATI, Chiara et al. **Sensor Networks with IEEE 802.15.4 Systems**: Distributed Processing, MAC, and Connectivity. 2011. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr;=&id=iuRmBlTmPnkC&oi=fnd&pg=PR5&dq=Sensor+Networks+with+IEEE+802.15.4:+Systems+Distributed+Processing,+MAC,+and+Connectivity&ots=h648Y3vDgT&sig=mmfKVPSPaCwvwzIaISt46zl14gM#v=onepage&q&f;=false>. Acesso em: 26 out. 2015.

CARVALHO, Thiago S. de; ANJOS, Fernando S. P. dos. **Zigbee**. Década provável [200-?]. Disponível em: <http://www.gta.ufrj.br/grad/07_1/zigbee/>. Acesso em: 09 set. 2016.

CURA, Hugo. **Análise**: Drone Syma X5SW, uma boa opção para iniciantes. 2015. Disponível em: <https://pplware.sapo.pt/analises-2/analise-drone-syma-x5sw-excelente-para-iniciantes/>. Acesso em: 02 set. 2016.

DRONE, Doctor. **Quase tudo sobre drones.** Copyright c2015. Disponível em: <http://doctordrone.com.br/>. Acesso em: 18 nov. 2015.

ELECTRONICS, Aosong(guangzhou). **Temperature and humidity module DHT11 Product Manual**. Década provável [200-?]. Disponível em: <http://akizukidenshi.com/download/ds/aosong/DHT11.pdf>. Acesso em: 09 nov. 2015.

FALUDI, Robert. **Building Wireless Sensor Networks**. United States Of America: O'reilly Media, Inc., 2011. 293 p.

FARAHANI, Shahin**. ZigBee Wireless Networks and Transceivers.** Oxford: Newnes, 2008. 329 p.

GLOVER, John M.. **Drone University**. United Kingdom: Droneuniversity, 2014. 135 p.

LEMOS, Manoel**. Arduino**: Conheça esta plataforma de hardware livre e suas aplicações. 2013. Disponível em: <http://blog.fazedores.com/arduino-conheca-esta-plataforma-de-hardware-livre-e-suas-aplicacoes/>. Acesso em: 27 jun. 2014.

LUGLI, Alexandre Baratella; GUILERME SOBRINHO, Darlan. **TECNOLOGIAS WIRELESS PARA AUTOMAÇÃO INDUSTRIAL**: WIRELESS_HART, BLUETOOTH, WISA, WI-FI, ZIGBEE E SP-100. 2012. Disponível em:

<https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjxrZD8nIrMAhUBUJAKHXHtAxAQFgg2MAA&url=http://www.inatel.br/biblioteca/index.php/modelos-de-trabalhos/doc_download/6088-tecnologias-wireless-para-automacao-industrial-wireless-hart-bluetooth-wisa-wi-fi-zigbee-e&usg=AFQjCNGKOOfNFzSjmSIFgqhQx1GDcTSaMQ&bvm=bv.119408272,d.Y2I>. Acesso em: 14 out. 2015.

MAKERS, Natal. **Dispositivo**: Conhecendo as partes do Arduino Uno. 2015. Disponível em: <http://natalmakers.blogspot.com.br/2015/08/dispositivo-conhecendo-as-partes-do.html>. Acesso em: 18 out. 2016.

OLIVEIRA NETO, Benjamin Batista de; MONTEIRO, Priscila de França; QUEIROGA, Sandro Lino Moreira de. **Aplicabilidade dos Microcontroladores em Inovações Tecnológicas**. 2012. Trabalho apresentado ao 7. CONNEPI, Tocantins, 2012.

PROCESSING. **Processing**. Copyright c2004. Disponível em: <https://processing.org/>. Acesso em: 26 ago. 2016.

SANTOS, Everton Luiz Ferret dos. A IEEE 802.15.4 COMO PLATAFORMA DE COMUNICAÇÃO DE DADOS. **Ilha Digital**, Florianópolis, v. 4, n. 0, p.97-105, out. 2013.

SILVA, Leandro Honorato de Souza. **Desenvolvimento de uma Rede de Sensores Sem Fio Utilizando ZigBee para Aplicações Diversas.** 2011. 80 f. TCC (Graduação) - Curso de Engenharia da Computação, Escola Politécnica de Pernambuco, Recife, 2011. Disponível em: <http://tcc.ecomp.poli.br/20111/monografia-Leandro_Honorato_VersaoFinal.pdf>. Acesso em: 15 out. 2015.

STOLL, George R.. **O QUE É ESTE TAL DO ZigBee?** 2008. Disponível em: <http://www.mecomp.com.br/rumo/o_que_e_este_tal_do_zigbee_3_.pdf>. Acesso em: 15 out. 2015.

TAVARES, Luís Antônio. Uma solução com Arduino para controlar e monitorar processos industriais. **Controle & Instrumentação**, Santa Rita do Sapucaí, v. 3, n. 2, p.77-79, out. 2013.

UK, D-robotics. **DHT11 Humidity & Temperature Sensor**. 2010. Disponível em: <http://www.micropik.com/PDF/dht11.pdf>. Acesso em: 09 nov. 2015.

WIRING. **Wiring**. Copyright c2004. Disponível em: <http://wiring.org.co/>. Acesso em: 26 ago. 2016.

APÊNDICE A – Código inserido no Arduino com o módulo emissor

```
#include <DHT.h>
#include <SoftwareSerial.h>

#define DHTPIN 4
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

SoftwareSerial XBee(2, 3); // Arduino RX, TX (XBee Dout, Din)

void setup() {
  XBee.begin(9600);
  dht.begin();
  Serial.begin(9600);
}

void loop() {
  // Wait a few seconds between measurements.
  delay(2000);

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius
  float t = dht.readTemperature();

  String dados ="";
  dados.concat(h);
  dados.concat("#");
```

```
dados.concat(t);

//XBee.print(digitalRead(h));
//XBee.println(digitalRead(t));
XBee.println(dados);
//XBee.println(t);
Serial.println(dados);
```

APÊNDICE B – Código inserido no Arduino com o módulo receptor

```
#include <SoftwareSerial.h>

SoftwareSerial XBee(2, 3); // Arduino RX, TX (XBee Dout, Din)

String inString = "";

void setup() {
  // put your setup code here, to run once:
  XBee.begin(9600);
  Serial.begin(9600);
  Serial.println("DHT11 test!");
}

void loop() {
  // put your main code here, to run repeatedly:
  while (XBee.available() > 0) {
 float inChar = XBee.read();

    if (inChar != '\n') {

      // As long as the incoming byte
      // is not a newline,
      // convert the incoming byte to a char
      // and add it to the string
      inString += (char)inChar;
    }
    // if you get a newline, print the string,
    // then the string's value as a float:
    else {
      Serial.print("Umidade: ");
```

```
      Serial.print(inString.substring(0,5));
      //Serial.print(inString.toFloat());

      Serial.print("  -  ");

      Serial.print("Temperatura: ");
      Serial.println(inString.substring(6,11));

      //Serial.println(" *C ");
      // clear the string for new input:
      inString = "";
    }
  }
}
```

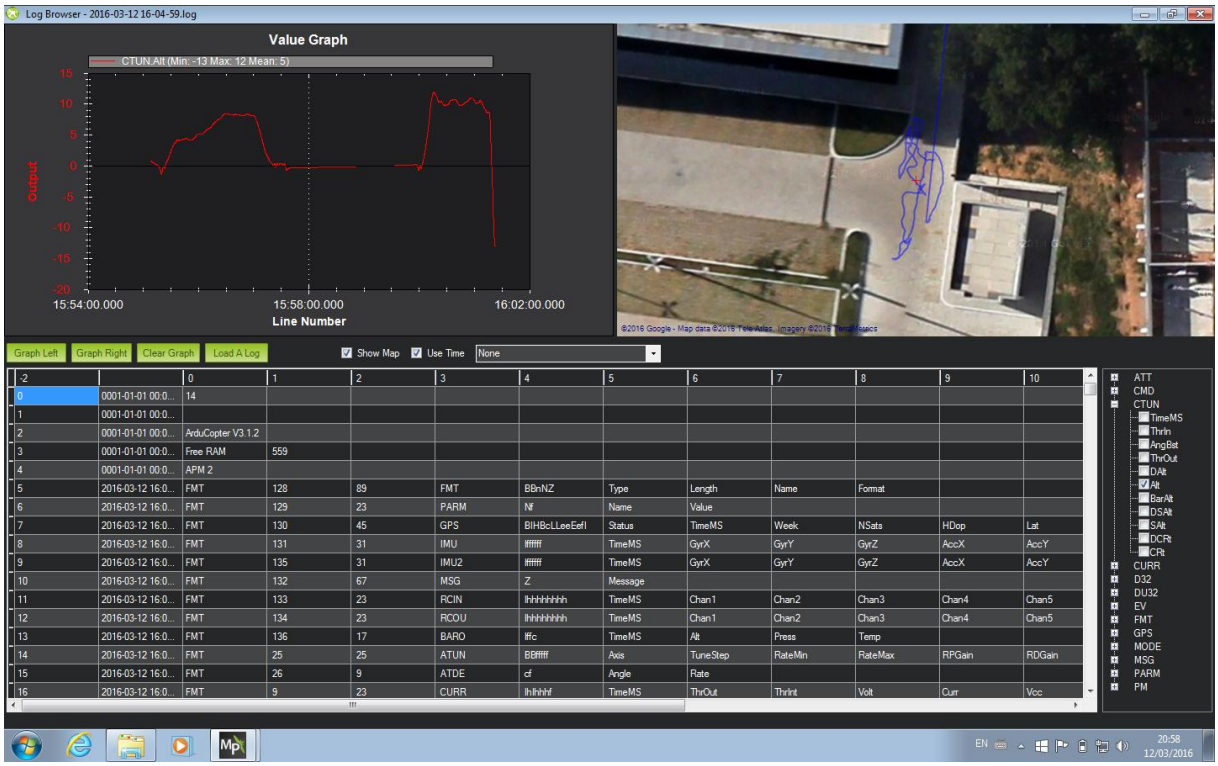
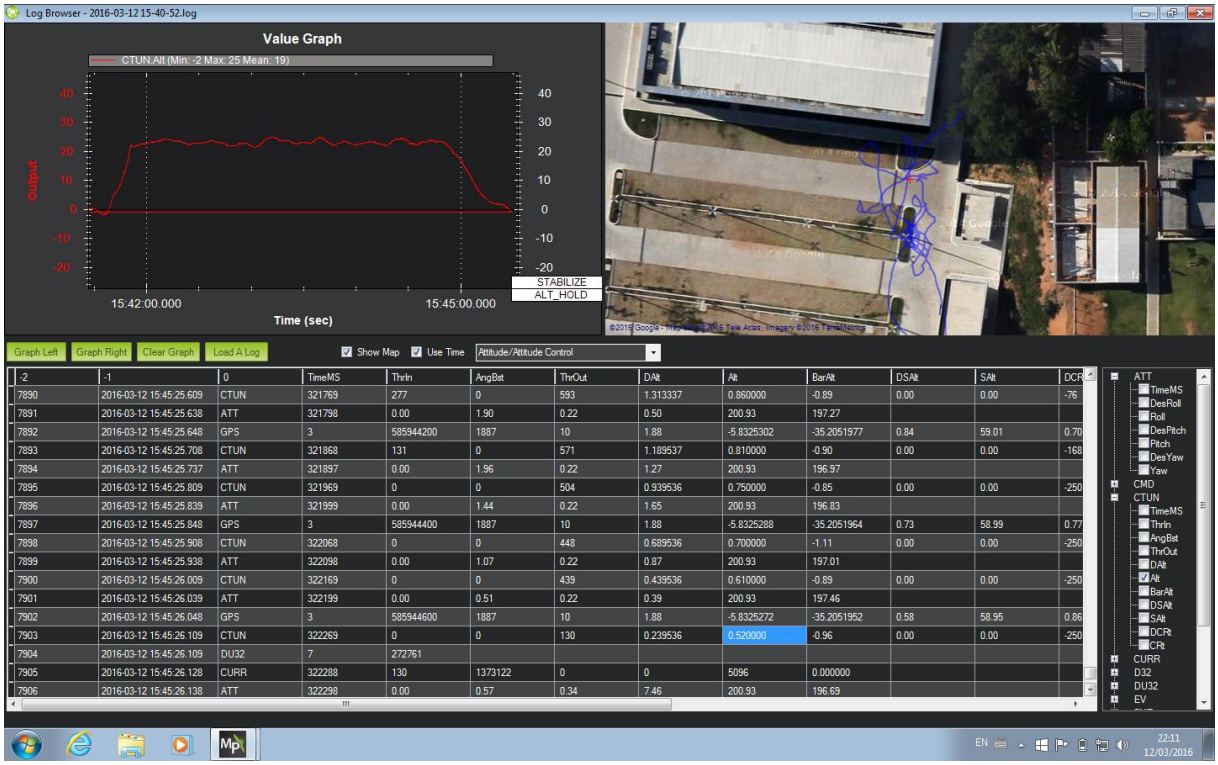
APÊNDICE C – Dados coletados pelo sensor e enviados através do módulo *XBee* série 1 PRO no dia 12 de março de 2016

Hora	Umidade	Temperatura
15:47:57	37	31
15:47:59	34	31
15:48:01	34	31
15:48:03	34	31
15:48:06	33	31
15:48:12	40	31
15:48:15	34	31
15:48:17	33	31
15:48:19	33	31
15:48:21	40	31
15:48:24	40	31
15:48:26	40	31
15:48:29	32	31
15:48:31	33	31
15:48:33	33	31
15:48:36	33	31
16:04:12	38	29
16:04:14	63	29
16:04:16	38	29
16:04:19	38	29
16:04:21	38	29
16:04:23	38	29
16:04:26	38	29
16:04:28	38	29
16:04:30	38	29
16:04:32	38	29
16:04:35	38	29
16:04:37	38	29
16:04:39	38	29
16:04:42	38	29
16:04:44	38	29
16:04:46	38	29
16:04:51	38	29
16:04:53	38	29
16:04:55	38	29
16:04:58	38	29
16:05:00	38	29
16:05:02	38	29
16:05:04	38	29
16:05:07	38	29
16:05:09	38	29
16:05:11	38	29
16:05:14	38	29
16:05:16	37	29
16:05:18	37	29
16:05:20	37	29
16:05:23	37	29
16:05:25	37	29
16:05:27	37	29
16:05:30	37	29
16:05:32	37	29

16:05:34	37	29
16:05:37	37	29
16:05:39	37	29
16:05:41	37	29
16:05:46	37	29
16:05:48	37	29
16:05:50	37	29
16:05:53	37	29
16:05:55	37	29
16:05:57	37	29
16:05:59	37	29
16:06:04	37	29
16:06:06	37	29
16:06:09	37	29
16:06:11	37	29
16:06:13	37	29
16:06:15	37	29
16:06:18	37	29
16:06:20	37	29
16:06:22	37	29
16:06:25	37	29
16:06:27	37	29
16:06:29	37	29
16:06:31	37	29
16:06:34	37	29
16:06:36	37	29
16:06:38	36	29
16:06:41	37	29
16:06:43	36	29
16:06:45	36	29
16:06:48	36	29
16:06:50	36	29
16:06:52	36	29
16:06:54	36	29
16:06:57	36	29
16:06:59	36	29
16:07:01	36	29
16:07:04	36	29
16:07:06	36	29
16:07:08	36	29
16:07:10	36	29
16:07:13	36	29
16:07:15	36	29
16:07:17	36	29
16:07:20	36	29
16:07:22	36	29
16:07:24	36	29
16:07:26	36	29
16:07:29	36	29
16:07:31	36	29
16:07:33	36	29
16:07:36	36	29
16:07:38	36	29
16:07:40	36	29
16:07:42	36	29
16:07:45	36	29

16:07:47	36	29
16:07:49	36	29
16:07:52	36	29
16:07:54	36	29
16:07:56	36	29
16:07:59	36	29
16:08:01	36	29
16:08:03	36	29
16:08:05	36	29
16:08:08	36	29
16:08:10	36	29
16:08:12	36	29
16:08:15	36	29
16:08:17	36	29
16:08:19	36	29
16:08:21	36	29
16:08:24	36	29
16:08:26	36	29
16:08:28	36	29
16:08:31	36	29
16:08:33	36	29
16:08:35	36	29
16:08:37	36	29
16:08:40	36	29
16:08:42	36	29
16:08:44	36	29
16:08:47	36	29
16:08:49	36	29
16:08:51	35	29
16:08:53	35	29
16:08:56	35	29
16:08:58	36	29
16:09:00	35	29
16:09:03	35	29
16:09:05	35	29
16:09:07	35	29
16:09:09	35	29
16:09:21	35	29
16:09:30	35	30
16:09:58	33	31
16:10:07	33	31
16:10:11	33	31
16:10:14	33	31
16:10:16	33	31
16:10:25	33	31
16:10:27	33	31
16:10:30	33	31
16:10:32	33	31
16:10:34	33	31
16:10:39	33	31
16:10:48	33	31
16:10:50	33	31
16:10:57	33	31
16:11:09	33	31
16:11:34	33	31

ANEXO A – Foto do teste realizado com o *Drone* em campo de visada do módulo receptor no dia 12 de março de 2016



ANEXO B – Foto do teste realizado no dia 12 de março de 2016, em que o *Drone* perdeu a conexão com o módulo receptor