



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
CENTRO MULTIDISCIPLINAR DE PAU DOS FERROS
CURSO DE BACHARELADO EM CIÊNCIAS E TECNOLOGIA

JEFFERSON JOHN DO REGO LEITE

IMPLEMENTAÇÃO DE UM SISTEMA DE MONITORAMENTO DE TRANSPORTE
ESTUDANTIL UTILIZANDO ARDUINO E REDES ZIGBEE.

PAU DOS FERROS

2018

JEFFERSON JOHN DO REGO LEITE

IMPLEMENTAÇÃO DE UM SISTEMA DE MONITORAMENTO DE TRANSPORTE
ESTUDANTIL UTILIZANDO ARDUINO E REDES ZIGBEE.

Monografia apresentada a Universidade
Federal Rural do Semi-Árido como requisito
para obtenção do título de Bacharel em
Ciências e Tecnologia.

Orientador: Pedro Thiago Valério de Souza,
Prof. Me.

PAU DOS FERROS

2018

©Todos os direitos estão reservados à Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996, e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tornar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata, exceto as pesquisas que estejam vinculadas ao processo de patenteamento. Esta investigação será base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) seja devidamente citado e mencionado os seus créditos bibliográficos.

Ficha catalográfica elaborada pelo Sistema de Bibliotecas
da Universidade Federal Rural do Semi-Árido, com os dados fornecidos pelo(a) autor(a)

L533i Leite, Jefferson John do Rego. Implementação
de um sistema de monitoramento de transporte
estudantil utilizando Arduino e redes ZigBee. /
Jefferson John do Rego Leite. - 2018.
58 f. : il.

Orientador: Pedro Thiago Valério de Souza.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Ciência e
Tecnologia, 2018.

1. Ônibus. 2. Arduino. 3. GPS. 4. XBee. I.
Souza, Pedro Thiago Valério de, orient. II.
Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

JEFFERSON JOHN DO REGO LEITE

IMPLEMENTAÇÃO DE UM SISTEMA DE MONITORAMENTO DE TRANSPORTE
ESTUDANTIL UTILIZANDO ARDUINO E REDES ZIGBEE.

Monografia apresentada a Universidade
Federal Rural do Semi-Árido como requisito
para obtenção do título de Bacharel em
Ciência e Tecnologia.

Defendida em: 18 / 04 / 2 018.

BANCA EXAMINADORA

Pedro Thiago Valério de Souza

Pedro Thiago Valério de Souza, Prof. Me. (UFERSA)
Presidente

Francisco Carlos Gurgel da Silva Segundo

Francisco Carlos Gurgel da Silva Segundo, Prof. Me. (UFERSA)
Membro Examinador

Vinicius Samuel Valério de Souza

Vinicius Samuel Valério de Souza, Prof. Dr. (UERSA)
Membro Examinador

A minha mãe Jucileide, pelo amor,
educação e carinho dedico este trabalho.

AGRADECIMENTOS

A Deus, pelo dom da vida.

Aos meus pais, pela educação e carinho.

Ao meu orientador, Professor Pedro, pela paciência e apoio.

Ao amigo Francisco Segundo por viabilizar alguns dos instrumentos necessários para este trabalho.

Aos funcionários da Universidade Federal Rural do Semi-Árido Câmpus Pau dos Ferros, por ajudarem no trabalho.

A Universidade Federal Rural do Semi-Árido, por possibilitar a realização do trabalho.

Depois de algum tempo você aprende, aprende que nunca se deve dizer a uma criança que sonhos são bobagens... poucas coisas são tão humilhantes que seria uma tragédia se ela acreditasse nisso.

William Shakespeare

RESUMO

O transporte público é um meio de locomoção mais utilizado ao redor do mundo. Como em algumas universidades brasileiras este sistema é disponibilizado para a locomoção do corpo discente até a universidade, como é o caso da UFERSA – Câmpus Pau dos Ferros. Alguns problemas inerentes ao sistema, tais como atrasos, mudanças de horários e até mesmo mudança de rotas acabam trazendo prejuízos aos discentes que fazem uso deste meio de transporte. Visando minimizar estes problemas, este trabalho propõe um sistema de monitoramento do ônibus, fornecido pela UFERSA – Câmpus Pau dos Ferros, utilizando a placa de prototipagem juntamente com os sensores XBee e GPS. Com tal trabalho em execução, será possível, a verificação da posição do ônibus e a transmissão dessa informação aos usuários. Esta informação é compartilhada por meio de um mapa na qual é apresentado a última posição do ônibus e a sua trajetória.

Palavras-chave: Ônibus. Arduino. GPS. XBee.

ABSTRACT

Public transportation is one of the most widely used means of transportation around the world. As in some Brazilian universities this system is available for the locomotion of the student body to the university, as is the case of UFERSA - Campus Pau dos Ferros. Some problems inherent to the system, such as delays, changes in schedules and even change of routes end up causing damages to the students who make use of this means of transport. In order to minimize these problems, this work proposes a bus monitoring system, provided by UFERSA - Câmpus Pau dos Ferros, using the prototype board together with the XBee and GPS sensors. With such work in progress, it will be possible to verify the position of the bus and the transmission of this information to the users. This information is shared by means of a map showing the last position of the bus and its trajectory.

Keywords: Bus. Arduino. GPS. XBee.

LISTA DE FIGURAS

Figura 1 - Diagrama do Sistema Embarcado.....	21
Figura 2 - Módulo Transmissor.....	22
Figura 3 - Módulo Receptor.	22
Figura 4 - Linux Embarcado.....	23
Figura 5 - Funcionamento de uma rede sem fio.	24
Figura 6 - Módulo ZigBee.....	26
Figura 7 - Modelo OSI.	28
Figura 8 - Canais do IEEE 802.15.4.....	29
Figura 9 - Topologia ZigBee.	30
Figura 10 - Módulo XBee S2C ZigBee.....	31
Figura 11 - Exemplo de uma plataforma de desenvolvimento de sistema embarcado.....	32
Figura 12 - Placa Arduino.	33
Figura 13 - Microcontrolador Atmega328p.....	35
Figura 14 - Visual da IDE quando aberta.....	36
Figura 15 - Placa com Linux embarcado.....	37
Figura 16 - Arquitetura básica de um Linux embarcado.....	38
Figura 17 - Módulo GPS VK2828U7G5LF.	40
Figura 18 - Teste de campo.	46

LISTA DE GRÁFICOS

Gráfico 1 - Perda do sinal em função da distância em lugares internos.	42
Gráfico 2 - Perda do sinal em função da distância em lugares externos.	44

LISTA DE TABELAS

Tabela 1 - Características de comunicação ZigBee.....	27
Tabela 2 - Características Arduino UNO.	34
Tabela 3 - Características do ATMega 328p.....	35
Tabela 4 - Descrição dos pinos presente no conector do módulo VK2828U7G5LF.	40
Tabela 5 - Perda do sinal em função da distância em lugares internos.	53
Tabela 6 - Média e desvio padrão da perda de sinal para lugares internos.	54
Tabela 7 - Perda do sinal em função da distância em ambientes externos.....	55
Tabela 8 - Média e desvio padrão da perda de sinal para ambientes externos.....	57

LISTA DE ABREVIATURAS E SIGLAS

ADSL	<i>Assymetrical Digital Subscriber Line.</i>
AP	<i>Acess Point</i>
API	<i>Application Programming Interface.</i>
ARM	<i>Advanced RISC Machine.</i>
ASCII	<i>American Standard Code for Information Interchange.</i>
BSS	<i>Basic Service Set.</i>
CGI	<i>Common Gateway Interface.</i>
CPU	<i>Central Processing Unit.</i>
CWI	Centro para Matemática e Ciência.
dB	<i>Decibel miliwatt.</i>
E/S	Entrada e Saída
ESS	<i>Extended Service Set.</i>
FFD	<i>function devices.</i>
FWA	<i>Fixed Wireless Access.</i>
GHz	GigaHertz.
GPS	<i>Global Positioning System.</i>
Hz	Hertz.
I/O	<i>Input and Output</i>
I2C	<i>Inter-Integrated Circuit.</i>
IBSS	<i>Independent Service Set.</i>
IDE	<i>Integrated Development Environment.</i>
IEEE	<i>Institute of Electrical and Electronic Engineers.</i>
IN	<i>Input</i>
ISM	<i>Industrial Sientific and Medical.</i>
JSON	<i>JavaScript Object Notation.</i>
KB	<i>Kilobytes</i>
Kbps	<i>Kilobytes por segundo.</i>
Km	Quilometro.
LED	<i>Light Emitting Diode</i>
LTE	<i>Long Term Evolution.</i>
m	Metro.

mA	MiliAmpere
MAC	<i>Media Access Control.</i>
MCU	Sistema baseado em microcontrolador.
MHz	MegaHertz.
MPU	Sistema baseado em microprocessador.
MWA	<i>Mobile Wireless Access.</i>
NWA	<i>Nomadic Wireless Access.</i>
OSI	<i>Open Systems Interconnection.</i>
PHY	Camada física do modelo OSI.
PWM	<i>Pulse-Width Modulation.</i>
RAM	<i>Random Access Memory.</i>
RFD	<i>Reduced-function devices.</i>
SoC	<i>System on Chip</i>
SPI	<i>Serial Peripheral Interface</i>
TTL	<i>Transistor-Transistor Logic.</i>
UFERSA	Universidade Federal do Semi-Árido
USART	<i>Universal Synchronous Asynchronous Receiver Transmitter.</i>
USB	<i>Universal Serial Bus.</i>
V	Volts

SUMÁRIO

1. Introdução	15
1.1. Motivações	17
1.2. Objetivos.....	17
1.2.1. Objetivo Geral	17
1.2.2. Objetivos Específicos	17
1.3. Metodologia.....	17
2. Desenvolvimento do Sistema Embarcado	19
2.1. Redes de Comunicação Sem Fio	23
2.2. Caracterização do Canal da Comunicações sem Fio	25
2.3. Padrão de Comunicações ZigBee	25
2.3.1. Padrão IEEE 802.15.4	27
2.3.2. Rede ZigBee	29
2.3.3. Módulo XBee	30
2.4. Sistemas Embarcados	31
2.4.1. Arduino.....	32
2.4.2. Linux Embarcado	36
2.5. Google Maps API.....	38
2.6. Sistema De Posicionamento Global	39
2.6.1. Receptores GPS.....	39
3. Resultados e Discursões	41
3.1. Comunicação entre Transmissor e Receptor	41
3.1.1. Testes em Comunicação Indoor	41
3.1.2. Testes em Comunicação Outdoor.....	43
3.2. Comunicação entre Receptor e Linux embarcado.....	44
3.3. Realização do experimento em campo	45

4. Conclusão	48
REFERÊNCIAS	49
ANEXO A – TABELAS DE PERDA DE SINAL.....	53

1. Introdução

Nas transmissões sem fio a informação transmitida por meio de ondas eletromagnéticas que se propagam no ar até o seu destino. A primeira transmissão de dados sem fio aconteceu em 1896, em que o italiano Guglielmo Marconi transmitiu dados através de um telégrafo sem fio entre duas cidades, Wimereux e Dover, separadas por uma distância de 52 Km. Esse fato marcou o início de uma nova era de comunicação (NOBELPRIZE.ORG, 1967).

Os grandes avanços tecnológicos, possibilitaram viver em um mundo recheado de redes sem fios, proporcionando diversos serviços, por exemplo: o de telefonia móvel. Graças a sua comodidade, mobilidade e fácil instalação, as comunicações sem fio facilitam a comunicação entre pessoas e serviços, bem como a interligação com a Internet. Os dispositivos de comunicações móveis tornaram-se altamente presentes na vida moderna, e cada vez mais busca-se dispositivos com alto poder de processamento, portáteis, leves, de baixo consumo e baixo custo. Essas características podem ser encontradas em Sistemas Embarcados que já estão presentes em muitos produtos como televisores, *smartphones* eletrodomésticos e até mesmo em veículos.

De fato, o conceito de comunicação móvel é um dos pilares de sustentação dos Sistemas Embarcados. Diante disso, é necessário definir um padrão de comunicação para esses dispositivos embarcados, visando a interoperabilidade no sistema. Um dos padrões mais difundidos atualmente é o Zigbee. Este padrão visa facilitar a comunicação sem fio através de uma plataforma de baixo custo, permitindo que esta tecnologia seja amplamente utilizada em aplicações de monitoramento, bem como aplicações que não requerem uma alta taxa de transferência. Dispositivos que seguem esse padrão apresentam uma alta confiabilidade e segurança, aliado a uma alta relação de custo-benefício. É um padrão voltado principalmente a sistemas que utilizam bateria, com baixa taxa de tráfego de dados, baixo custo, baixa complexidade e baixas potências de transmissão.

Ao longo dos anos, várias plataformas de desenvolvimento de sistemas embarcados vêm sendo desenvolvidas. Essas plataformas de desenvolvimento visam facilitar o projeto do sistema embarcado. Uma destas plataformas, amplamente aceita pela comunidade acadêmica é o Arduino.

A plataforma de desenvolvimento livre Arduino é projetada com um microcontrolador Atmel AVR, possuindo suporte a padrões de comunicação embutidos e uma linguagem de programação derivada do C/C++. Com o objetivo de ser um dispositivo que seja barato, funcional e fácil de programar, o Arduino possui uma grande diversidade de módulos e sensores que se ligam a plataforma para desempenhar diferentes funções. Por possuir um Microcontrolador esta plataforma possui uma memória programável que pode ser escrita por meio de uma interface de desenvolvimento completa para programação.

Um dos módulos presentes para o Arduino é o GPS (*Global Positioning System* - Sistema de Posicionamento Global). O GPS é composto por satélites que formam um sistema, juntamente com bases estáticas instaladas na terra, possuindo o principal objetivo de calcular as coordenadas de um receptor GPS em qualquer lugar do planeta, coletando assim suas coordenadas de latitude e longitude. Este sistema é bastante preciso, se tornando adequado a muitas aplicações, tais como utilização na aviação geral e comercial e na navegação marítima. O receptor GPS também é considerado um Sistema Embarcado devido seu baixo consumo de energia e seu tamanho, facilitando assim sua portabilidade.

Atrasos em ônibus são coisas comuns no dia a dia de quem utiliza o transporte público, afinal o trânsito pode oferecer obstáculos que favorecem este atraso. Esse problema não é algo novo, sendo conhecido desde de algum tempo atrás.

Visando resolver a problemática do atraso dos ônibus na cidade de Palmas TO, que vinha sendo a principal reclamação dos usuários, foi implantado no mês de junho de 2014 um novo sistema de monitoramento a distância. Os coletivos receberam aparelhos de GPS que enviam informações sobre a rota e a pontualidade dos veículos para uma central da Secretaria de Transporte. Na capital, são 205 ônibus atendendo a cerca de 80 mil passageiros por dia (G1, 2014). Afim de resolver também as complicações de atraso de ônibus em sua cidade Rodrigues (2014), propõem como alternativa para amenizar o impacto desses problemas sobre os usuários, o desenvolvimento de um sistema de automação baseado na utilização de uma rede *wireless* adaptativa que promova as informações necessárias para que os usuários, e sejam informados através de uma tela, instalada na parada, quais unidades estão se aproximando da respectiva parada. Esta informação será uma estimativa de tempo de chegada de cada ônibus, podendo assim escolher qual o horário e linha é mais adequado, além de também oferecer informações de tempo de chegada nas paradas subsequentes.

1.1. MOTIVAÇÕES

Dado o problema de oscilação de horários em que o ônibus da UFERSA passa em cada parada, há a necessidade de desenvolvimento de um projeto que minimize esta desagradável situação. Desta forma, cumprindo-se os objetivos propostos, o sistema pode ajudar com os alunos, e todos que necessitam deste sistema de transporte. Este projeto é ambicioso para criação de tal funcionalidade, entretanto inicialmente haverá uma prototipação de forma a ajudar a entender o propósito de um projeto maior que pode ser desenvolvido. O objetivo do protótipo é propor melhorias, minimizar riscos e maximizar a funcionalidade.

1.2. OBJETIVOS

1.2.1. Objetivo Geral

- Implementar um sistema de monitoramento do ônibus escolar, fornecido pela UFERSA Campus Pau dos Ferros, utilizando o *kit* Arduino, juntamente com os sensores XBee e GPS.

1.2.2. Objetivos Específicos

- Entender o funcionamento básico do Arduino, bem como os princípios básicos de Sistemas Embarcados e Computação Perversiva.
- Estudar sobre os princípios de transmissão de dados digitais utilizando o padrão XBee;
- Ter conhecimento sobre as formas de comunicação do módulo XBee com o Arduino;
- Compreender os fundamentos sobre o módulo GPS e as formas de comunicação deste módulo com o Arduino;
- Implementar o sistema de monitoramento do ônibus escolar;
- Realizar testes de performance do sistema implementado, traçando figuras de mérito que possam avaliar o desempenho do sistema;

1.3. METODOLOGIA

Este trabalho foi dividido em cinco estágios para um melhor progresso e dinamismo dos resultados. Assim ele foi dividido entre:

1. Pesquisa bibliográfica;

2. Desenvolvimento do transmissor e receptor;
3. Construção de uma página web para observação dos resultados;
4. Teste de perda de sinal;
5. Experimentos práticos.

2. Desenvolvimento do Sistema Embarcado

Neste trabalho foram desenvolvidas as atividades relacionadas com a especificação do sistema, montagem, teste e verificação dos resultados, com a capacidade de fornecer a posição do ônibus dentro da área de alcance da rede *wireless*. Primeiramente foi realizado um estudo sobre as melhores tecnologias de comunicações comerciais disponíveis no mercado, de forma que suprissem satisfatoriamente as demandas deste projeto, fornecendo um protótipo funcional. Estes requisitos para os dispositivos são: o baixo consumo de energia, a possibilidade de comunicação direta entre os dispositivos, a alta confiabilidade na entrega dos pacotes, médio alcance e um custo de implementação viável ao trabalho.

Para o bom funcionamento deste sistema fez-se necessário inclusão de diferentes tipos de tecnologias e recurso com o objetivo de operacionalizar as informações na rede e viabilizar a implementação prática do projeto. O XBee por se tratar de um módulo que implementa o padrão ZigBee, que basicamente é composto por dispositivos de comunicação com um protocolo simples, possuindo baixo custo no desenvolvimento, menor perda por interferência, um tempo reduzido de conexão na rede. A escolha por essa tecnologia neste trabalho foi devido ao fato do baixo custo de aquisição e manutenção, e por ser uma tecnologia com características que facilitam os processos de automação. Já o Arduino é uma plataforma que de fácil programação, que para este trabalho viabiliza a comunicação entre o receptor e o Linux Embarcado. Para a realização da integração entre os componentes, foram desenvolvidos novos recursos de *hardware* e de *software* através das ferramentas, conforme demonstrado a seguir.

Afim de minimizar os custos de estrutura de rede para este trabalho a topologia escolhida foi a do tipo malha, que conta com apenas dois dispositivos e estes têm ligação direta um ao outro. Para melhor disponibilidade de recursos e melhor comunicação entre os módulos a forma de comunicação será feita somente do transmissor para o receptor e nunca ao contrário.

O sistema foi dividido em três módulos: transmissor, receptor e Linux embarcado. Sendo que cada um dos módulos desempenha uma função característica para que todo o sistema funcione corretamente. A Figura 1 ilustra os três módulos e suas devidas ligações. Inicialmente foi definido que a comunicação entre transmissor e o receptor seria feito por módulos XBee em configuração broadcast, para definir este tipo de configuração houve a

programação no modo API de cada um dos módulos, feita pelo envio de uma mensagem de dados que possui a identificação ID de origem e destino.

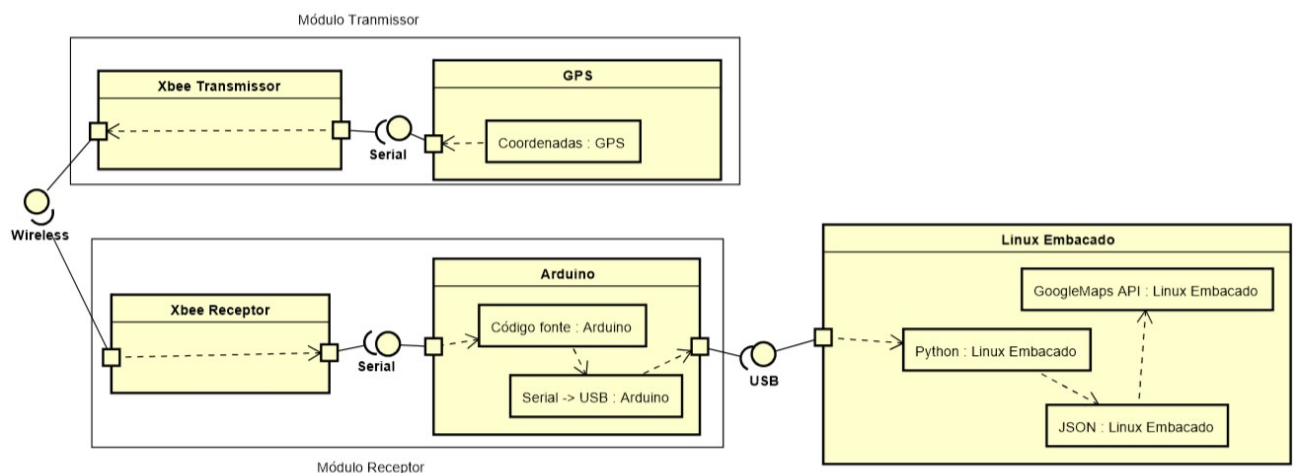
Iniciando-se pelo transmissor ilustrado na Figura 2, este módulo conta com dois *hardwares* básicos, o de GPS e o XBee. O GPS conta com uma interface serial que pode ser ligada diretamente a interface também serial presente no XBee. O GPS já vem pré-programado de fábrica para enviar informações em períodos constantes de 1 segundo referentes a localização por meio da sua interface serial. O XBee é configurado para que se comunique diretamente com outro XBee (o XBee receptor) que também foi programado da mesma forma. Logo todas as informações oriundas do GPS podem ser transmitidas graças a interface serial presente em ambos os *hardwares*. Este módulo necessita de energia para seu funcionamento, foi então acoplado uma bateria capaz de fornecer uma tensão de 5 volts afim de alimentar todo o módulo.

O receptor é um módulo composto por dois *hardwares* conforme a Figura 1 e 3 ilustra: o XBee e o Arduino. O XBee deste módulo foi configurado para receber todas as informações entregues ao XBee presente no transmissor, logo, ele irá receber todas as informações oriundas do GPS (também presente no transmissor). O XBee conta com uma interface serial, sendo assim, o Arduino também conta com essa mesma interface pela qual estes *hardwares* estão ligados. Havia a necessidade de um microcontrolador que realizasse algumas tarefas, já que as informações deveriam ser filtradas e convertidas antes de serem passadas ao Linux embarcado. O GPS através de sua porta serial envia informações, das quais grande parte é desnecessária para este trabalho como é o caso da elevação do terreno. A plataforma Arduino se mostrou pronta para atender esses requisitos, pois apresenta um micro controlador ATMEL, uma porta USB para comunicação com o Linux embarcado e uma porta serial para comunicação com o XBee receptor. Com a intenção de reduzir os gastos com este trabalho, utilizou-se apenas uma placa Arduino ligada diretamente ao módulo XBee receptor pela porta serial. Não havendo necessidade de outra placa no módulo transmissor, já que o módulo GPS conta com uma porta serial, para transferência das informações, que pode ser diretamente ligada ao módulo XBee transmissor. O Arduino foi programado por meio da sua IDE, que pode ser adquirida gratuitamente através do site oficial da plataforma, com um *sketch* que obtém apenas o número de satélites conectados e as coordenadas que são convertidas em um formato para que possam ser empregadas no mapa. Assim, como o transmissor, o receptor necessita de uma fonte de energia, preferencialmente de 5 *volts*. Mas este módulo é ligado

pela porta USB ao Linux embarcado e esta porta fornece a tensão e corrente necessárias para o bom funcionamento deste módulo.

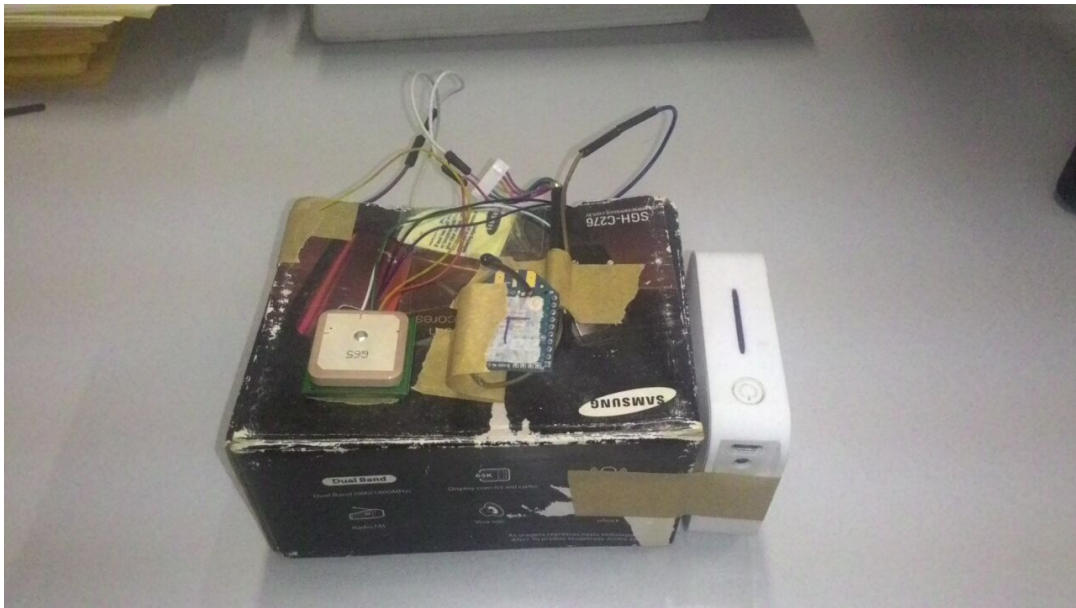
Um micro controlador sozinho não é capaz de trabalhar com uma página web com a agilidade que este trabalho demanda, onde se faz necessário que os pontos sejam carregados na API do Google Maps o mais rápido possível, evitando que o usuário receba informações com lentidão ou atrasos gerados pelo sistema. Para melhor aplicação deste trabalho utilizou-se um módulo com um Linux embarcado ilustrado na Figura 4, capaz de processar dados com uma rapidez bem mais elevada que um microcontrolador. Este sistema embarcado tem principal finalidade receber os dados processados pelo Arduino para então, torná-los disponíveis de uma forma mais amigável ao usuário. Através de um *script* codificado em linguagem Python, os dados são recebidos do micro controlado por meio da porta USB conforme a Figura 1, para então serem analisados antes de serem salvos em arquivos do tipo JSON, que neste trabalho armazenam as coordenadas, para então serem carregados pela página web que exibe num mapa, graças a API do Google Maps, os pontos em que o ônibus já passou bem como um ícone do veículo em sua última posição obtida. Assim como a maioria dos circuitos elétricos, este também necessita de uma excitação, só que diferente dos módulos anteriores, este consome uma potência de 10W (5 volts a 2 amperes). Para tal este circuito é alimentado por meio de uma fonte de tensão que é ligada a rede elétrica afim de fornecer tal energia para o seu bom funcionamento.

Figura 1 - Diagrama do Sistema Embarcado.



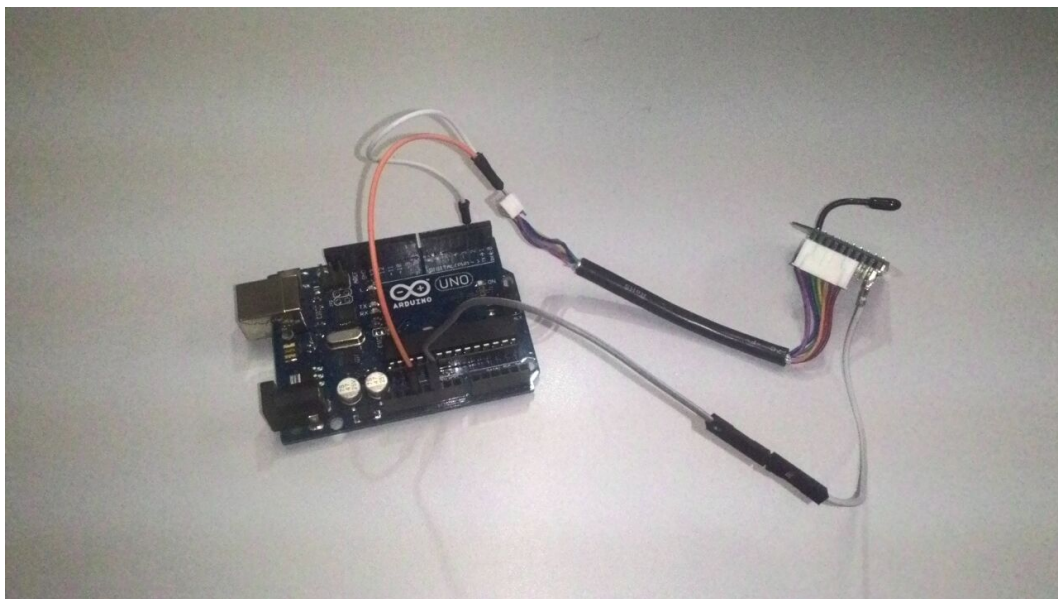
Fonte: Autor (2018).

Figura 2 - Módulo Transmissor.



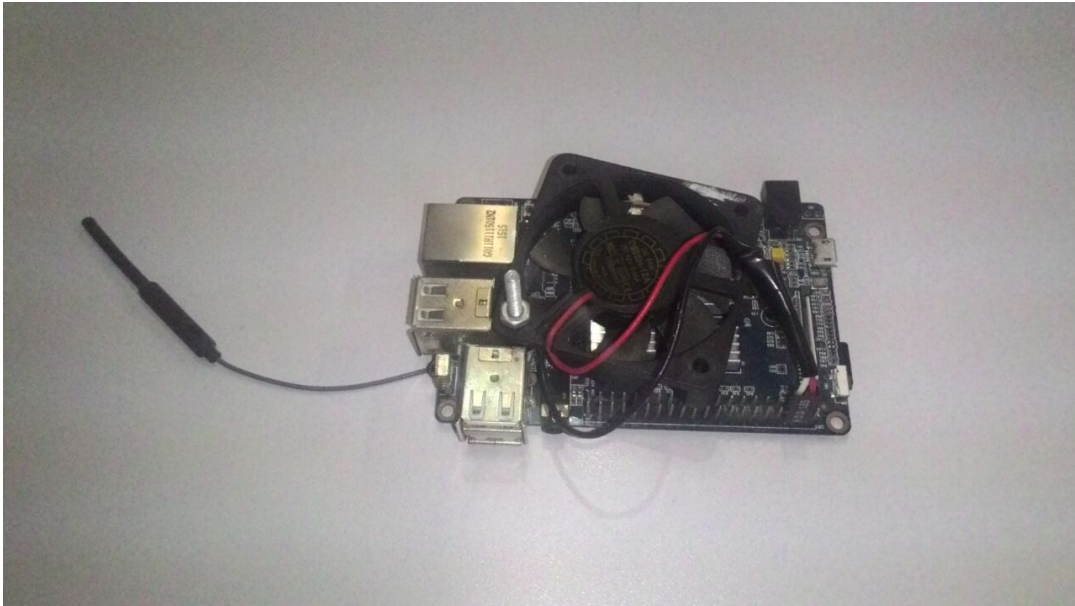
Fonte: Autor (2018).

Figura 3 - Módulo Receptor.



Fonte: Autor (2018).

Figura 4 - Linux Embarcado.



Fonte: Autor (2018).

2.1. REDES DE COMUNICAÇÃO SEM FIO

Com a evolução da tecnologia, é notável a variedade de padrões de redes sem fio (*wireless*) existentes para a comunicação de dados. As redes de comunicações sem fio estão presentes em nossas vidas quase que a todo momento conforme é ilustrado pela Figura 5, fato que se torna possível devido aos grandes avanços na área de telecomunicações, oferecendo grandes vantagens na utilização dessa rede.

Para Shannon (1948), um sistema de comunicações de dados digitais pode ser caracterizado por três elementos fundamentais: o transmissor, o receptor e o canal de comunicações. Neste contexto, o canal de comunicações tem um papel fundamental, dado ele é o elemento físico responsável por guiar as informações do transmissor ao receptor.

Os canais de comunicações podem ser dos mais variados tipos. De forma geral, os canais de comunicações podem ser divididos em duas classes: os canais com fio e os canais sem fio.

Nos canais de comunicações com fio, o elemento físico responsável por transportar a informação é um meio guiado, como um par metálico trançado, uma fibra ótica ou um cabo coaxial, por exemplo. Por sua vez, nos canais de comunicação sem fio, a transmissão da

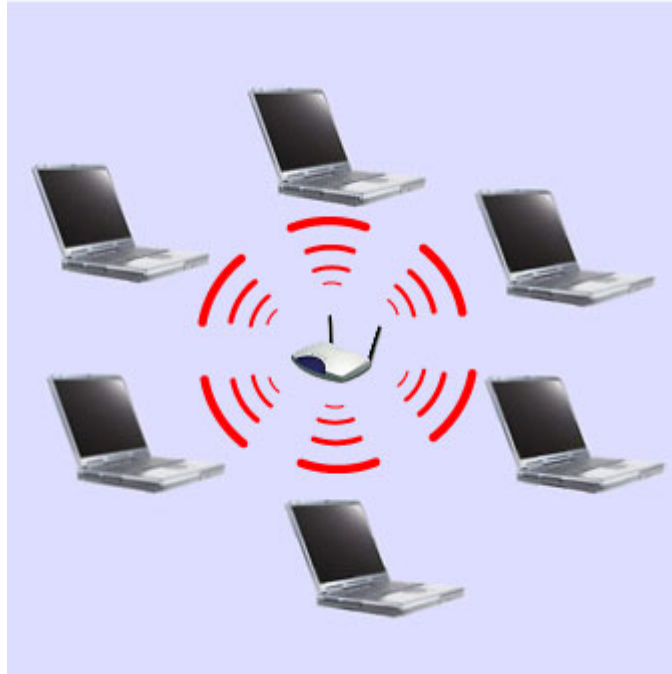
informação é realizada através de ondas eletromagnéticas, que se propagam no espaço livre entre o transmissor e o receptor.

Segundo Moraes (2009), as redes sem fio consistem de um sistema de transferência em que os dados trafegam sobre o espaço livre, não necessitando de canais de comunicações cabeados. Essas redes permitem a comunicação de dados com uma grande flexibilidade.

As redes sem fio podem ser utilizadas como ferramenta para expandir uma rede já existente, bem como uma alternativa às redes cabeadas. Segundo Chave (2010), as redes sem fio são sistemas complexos, que contém vários processos correlacionados. Um exemplo disso é o processamento de sinais que visa garantir a qualidade na transmissão dos dados.

Atualmente, cresce o interesse da comunidade científica pelo estudo dos padrões de comunicações sem fio. Isso se deve ao fato que esse tipo de rede possui características de mobilidade e independência de ligações por cabos, o que pode diminuir o valor necessário para sua instalação. Outra característica importante é pela rede ser escalável, permitindo que ela suporte o crescimento ou redução da demanda.

Figura 5 - Funcionamento de uma rede sem fio.



Fonte:

https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2011_2/rodrigo_paim/intro.html

2.2. CARACTERIZAÇÃO DO CANAL DA COMUNICAÇÕES SEM FIO

Tanebaum (2003) afirma que as ondas de rádio não apresentam dificuldade para serem geradas e que devido sua facilidade de atravessar paredes, são amplamente utilizadas tanto em ambientes fechado como também os abertos. Outra característica importante é que as ondas de rádio se propagam de uma antena isotrópica para o espaço de forma igualmente em todas as direções, isso faz com que o dispositivo final não necessite está em um alinhamento com o dispositivo de ligação.

Em uma transmissão sem fio, o sinal que se propaga pelo ambiente, da antena transmissora para a antena receptora na forma de ondas eletromagnéticas sobre um canal sem fio, que está sujeito a variações de atenuação e perdas de energia. Segundo Tude (2004) um ambiente de espaço livre apenas parte da energia transmitida através das ondas de rádio é captada pela antena receptora, esta energia é tanto menor quanto a maior a frequência e a distância, sendo esta perda no ambiente expressa em dB.

Tude (2004) ainda discute que, durante a transmissão num ambiente livre, as ondas que se propagam estão sujeitas a reflexões na atmosfera e no solo, e estas por sua vez sofrem deformações em sua amplitude e fase, gerando mudanças na potência de sinal recebido. Essa variação do sinal recebido ao longo do tempo é denominada de desvanecimento, que pode ser causada por esse e outros motivos, como a presença de obstáculos na trajetória da onda ou atenuação em ambientes de chuva. Para Cavalcante (2010) nos sistemas de transmissão de rádio, a previsão das perdas de propagação é de característica complexa com um alto nível de dificuldade, pois pode-se existir diferentes obstáculos em meio a uma transmissão como por exemplo: distintas estruturas arquitetônicas, tipos variados de relevo e eventos naturais.

2.3. PADRÃO DE COMUNICAÇÕES ZIGBEE

A primeira etapa para realizar uma comunicação sem fio é definir os parâmetros dos transmissores, tal como taxa de transmissão, frequência de operação, algoritmo de codificação de canal, entre outros. Devido as grandes variedades de cenários de comunicações existentes, existe uma variedade igual de padrões de comunicações sem fio. Um destes padrões é o ZigBee que tem seu módulo ilustrado na Figura 6.

Para Evangelista (2010), o primeiro tipo de rede ZigBee foi construída em 1997, com o objetivo de sanar projetos em que as redes Wi-Fi e *Bluetooth* eram inefficientes, devido ao mal gerenciamento de energia e complexidade de protocolos para a transmissão de dados destas redes. Evangelista (2010) defende que a ZigBee Alliance, empresa que construiu essa

tecnologia de comunicação, tinha o objetivo de criar redes *wireless* altamente confiáveis, seguras e com baixo consumo, com um padrão voltado para aplicações que utilizassem bateria bem como uma baixa taxa de tráfego de dados. Outra característica inerente a comunicação ZigBee era o seu ótimo custo benefício.

Figura 6 - Módulo ZigBee.

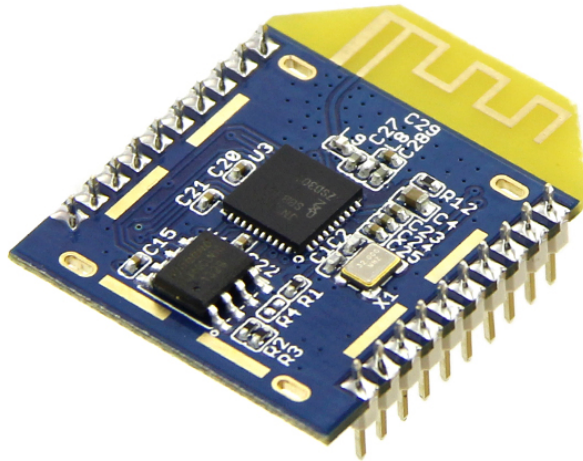


Tabela 1 - Características de comunicação ZigBee.

Frequências de comunicação	2,4 GHz (Global), 915 Mhz (EUA) e 868 Mhz (Europa).
Velocidades de transmissão	250 kbps (Global), 40 kbps (EUA) e 20 kbps (Europa).

Fonte Rodrigues (2010).

Segundo Evangelista (2010), a tecnologia ZigBee se utiliza de dispositivos *full-function devices* (FFD) e *reduced-function devices* (RFD) com o objetivo de que estes não acumulem funções às quais não são utilizadas, garantindo uma eficiência energética bem com a diminuição da complexidade do projeto. Os dispositivos FFD estão dispostos a executar todos as funções especificadas no padrão 802.15.4, podendo ser qualquer um dos tipos de dispositivos da rede. Os dispositivos RFD são mais simples e estes só possuem a função de dispositivos finais, com função de se comunicar com os dispositivos FFD.

2.3.1. Padrão IEEE 802.15.4

A grande maioria dos dispositivos de telecomunicações implementa a pilha de protocolos definida pelo modelo OSI, conforme a Figura 7. Embora exista uma padronização dos protocolos das camadas superiores, tal como a camada de aplicação e rede, apresentando padrões bem estabelecidos e utilizados na indústria, as camadas de enlace e física possuem vários padrões disponíveis. Esse fato decorre da natureza do próprio canal de comunicações no qual o dispositivo se presta.

Figura 7 - Modelo OSI.



Fonte: <https://sites.google.com/site/redesdecomunicacaonivel3/modulo-2---redes-de-computadores/4-modelo-geral-de-comunicacao/modelo-osi>

Desta maneira, o *Institute of Electrical and Electronics Engineers* (IEEE) determina padrões da camada de enlace para a transmissão de dados seguindo o modelo OSI. O objetivo dessa padronização consiste em unificar a comunicação de diferentes dispositivos, sendo que todos que seguem esse padrão consigam se comunicar de forma eficiente, mesmo com componentes diferentes.

O padrão 802.15.4, utilizado pelo ZigBee, só tem especificações para funções pertencentes ao protocolo *Physical Layer Protocol* (PHY), que visa especificar os algoritmos de codificação e decodificação de sincronismo de relógios e de quadros de dados, as interfaces com camadas de *Media Access Control* (MAC) e requisitos físicos do equipamento quanto a sensibilidade do receptor, como também a potência em que os dados estão sendo transmitidos (CUNHA, 2013).

Evangelista (2010) ainda diz que a camada física é uma camada que se encontra em um estado muito próximo ao *hardware*, afinal ela controla e se comunica com os dispositivos transmissores, sendo também associada ao processo designado a selecionar

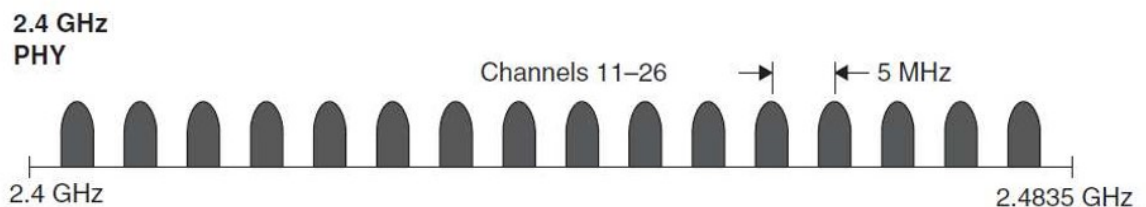
canais para a transmissão. Esta camada é responsável por ligar e desligar os transreceptores, transmitir e receber dados e selecionar o canal.

O protocolo 802.15.4 é dividido em 16 canais, sendo cada um uma faixa do espectro de rádio frequência. Estes por sua vez são enumerados do 11 ao 26 e todos atuam na faixa de frequência de 2.4 GHz, com uma diferença de um canal (banda de folga) ao outro de 5 MHz conforme a Figura 8.

Sobre estes canais é necessário mostrar que são do tipo *half-duplex*, ou seja, ou o dispositivo escuta o canal sendo um receptor, ou ele transmite os dados no canal, sendo incapaz de realizar as duas operações ao mesmo tempo.

Outra informação é que se o dispositivo está conectado a um canal ele é incapaz de receber ou transmitir qualquer de dados nos outros 15 canais (EVANGELISTA, 2010). Rodrigues (2010), fala que as redes do tipo ZigBee tem a rotina de trabalharem em canais estáticos, sendo estes capazes de diminuir a perda dos dados transmitidos mesmo que o canal esteja ocupado com outras transmissões em frequências próximas.

Figura 8 - Canais do IEEE 802.15.4.



Fonte: (GISLASON, 2008)

A camada MAC tem como principal objetivo permitir várias estruturas topológicas com baixa complexidade permitindo que um dispositivo RFD funcione corrente em uma rede sem a necessidade de grandes quantidades de memória, bem como gerir de forma eficiente muitos dispositivos sem que estes fiquem em espera, sendo um grande diferencial desta tecnologia quando confrontada com outras tecnologias sem fio.

2.3.2. Rede ZigBee

A rede ZigBee é composta por componentes que se organizam em diferentes topologias, dentre todos os principais componentes são: Coordenador ZigBee, *Router* ZigBee e *ZigBee End Device*.

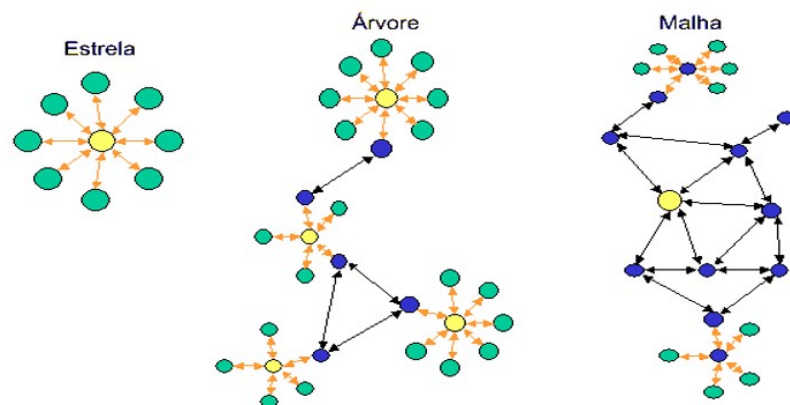
Estes dispositivos podem se organizar em diferentes topologias conforme é ilustrado pela Figura 9, sendo a mais indicada dependente do tipo da aplicação.

As topologias de rede do tipo estrela presente na Figura 9, são a mais fáceis de serem implantadas, entretanto possuem apenas um coordenado e são limitadas por uma distância do ZigBee *End Device* ao coordenador, sendo indicada para áreas com poucos obstáculos.

Outra topologia é a de árvore conforme a Figura 9, sendo formadas por sub-redes, apresentando um coordenador cada uma, que se ligam através dos roteadores ampliando assim o alcance da rede que na topologia de estrela era limitada.

Por fim a topologia de rede do tipo malha vista na Figura 9, visa construir uma rede autossuficiente, podendo potencializar a transmissão de dados de uma caracterização de ajuste automático da rede, se tornando a mais indicada para o presente trabalho devido a necessidade de uma comunicação direta entre os dois dispositivos ZigBee e um ajuste automático da rede afim de facilitar a comunicação entre os dispositivos desta rede.

Figura 9 - Topologia ZigBee.



Fonte: (MAES, 2014).

2.3.3. Módulo XBee

Utilizando-se do padrão de comunicação ZigBee, a empresa Digi International desenvolveu um módulo de comunicação chamado XBee indicado na Figura 10. Este módulo consiste em um ecossistema completo de módulos sem fio, *gateways*, adaptadores e *software*, todos projetados para acelerar o desenvolvimento sem fio para implementações globais.

Figura 10 - Módulo XBee S2C ZigBee.



Fonte: <https://www.filipeflop.com/produto/modulo-xbee-s2c-zigbee-wire-antenna/>

Já sobre o ponto de vista de Rhysider (2012) os XBees podem operar em um modo de dados transparente ou em um modo de interface de programação de aplicativos (API) baseado em pacotes. No modo transparente, os dados que entram no pino Data IN (DIN) são transmitidos diretamente pelo ar para os rádios receptores pretendidos, sem qualquer modificação. Os pacotes de entrada podem ser endereçados diretamente a um destino (ponto a ponto) ou transmitidos para vários destinos (estrela). Esse modo é usado principalmente em instâncias em que um protocolo existente não pode tolerar alterações no formato de dados. Os comandos AT são usados para controlar as configurações do rádio. No modo API, os dados são agrupados em uma estrutura de pacotes que permite endereçamento, configuração de parâmetros e *feedback* de entrega de pacotes, incluindo sensoriamento remoto e controle de E/S digital e pinos de entrada analógica.

2.4. SISTEMAS EMBARCADOS

Segundo Tanenbaum (2003) sistemas embarcados são sistemas computacionais completos e independentes, mais simples que um computador de propósito geral (*desktops*), encarregados de executar apenas uma função determinada tarefas pré-determinadas, com

requisitos específicos na qual executam geralmente repetidas vezes. Esses sistemas também são denominados de sistema embutido, seu computador é completamente encapsulado, totalmente dedicado ao dispositivo que controla.

Os sistemas embarcados exemplificados pela Figura 11, são compostos fundamentalmente pelos mesmos componentes de um computador pessoal, só que com tamanho e capacidade limitadas para o fim se destina. São muito utilizados no cotidiano, e seus usuários geralmente não os consideram com um computador, tornando a experiência na sua utilização perversiva. São exemplos de sistemas embarcados: aparelho de som, televisão, câmera digital, brinquedos, modem ADSL, entre muitos outros dispositivos. A evolução da microeletrônica e o barateamento das CPU's viabilizaram o emprego de sistemas embarcados nos diversos equipamentos.

Figura 11 - Exemplo de uma plataforma de desenvolvimento de sistema embarcado.



Fonte: https://pt.wikipedia.org/wiki/Programação_embarcada

Com o passar dos anos as informações passaram a ser tratadas por dispositivos embarcados, pois segundo Souza (2017) houve uma evolução no tratamento de informação - nos anos de 1980 esta função era desempenhada por grandes computadores denominados mainframes, já em 1990 que realizou este tratamento foram os computadores pessoais, e a partir do ano 2000 os sistemas embarcados começaram a exercer esta função.

2.4.1. Arduino

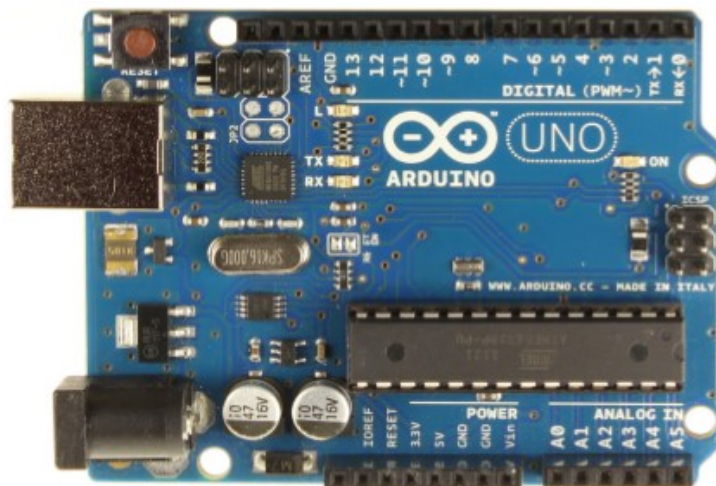
Considerado uma plataforma de fácil manipulação, desenvolvimento e prototipação, o Arduino ilustrado na Figura 12 é uma plataforma de desenvolvimento de sistemas embarcados bastante conhecida no mundo. Essa plataforma pode ser utilizada em aplicações bem

genéricas, já que é composto por um microcontrolador da família Atmega e um conversor USB serial.

Uma característica importante que fez o Arduino se popularizar é a que seu projeto é *Open Source* (Código aberto) e *Open Hardware* (*Hardware* aberto) permitindo que seja desenvolvido das mais diversas formas. Contudo, a fundação que desenvolveu o Arduino não permite que este nome seja usado, obrigando o desenvolvedor a criar outro nome para o seu projeto similar ao Arduino, com isso surge nomes como Feduino e Severino.

Quanto a programação esta plataforma conta com diversas bibliotecas, bem como o *software* depois de ser instalado já apresenta exemplos funcionais, permitindo o usuário desenvolver projetos assim que o *software* de programação for instalado. Mcroberts (2011), defende que a plataforma Arduino é um pequeno computador programável.

Figura 12 - Placa Arduino.



Fonte: <http://comphaus.com.br/home/wp-content/uploads/2014/03/Arduíno-Uno-R3-Front.jpg>

Existem várias placas Arduino, contudo a mais usada é a do tipo UNO, suas características são descritas na Tabela 2.

Tabela 2 - Características Arduino UNO.

Tamanho:	5,3cm x 6,8cm x 1,0cm
Microcontrolador:	ATmega328
Tensão de operação:	5 V
Tensão de entrada (recomendada):	7-12 V
Tensão de entrada (limites):	6-20 V
Pinos de entrada/saída (I/O) digitais:	14 (dos quais 6 podem ser saídas PWM)
Pinos de entrada analógicas:	6
Corrente DC por pino I/O:	40 mA
Corrente DC para pino de 3,3 V:	200 mA
Memória Flash:	32 KB (dos quais 0,5KB são usados pelo <i>bootloader</i>)
SRAM:	2 KB
EEPROM:	1 KB
Velocidade de Clock:	16 MHz
Temperatura de operação:	10 °C a 60 °C

Fonte: www.robocore.net/modules.php?name=GR_LojaVirtual&prod=530

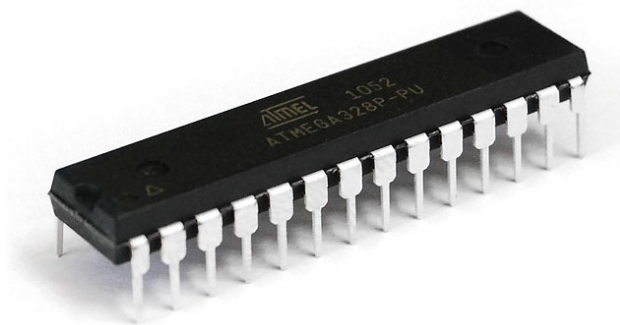
O ATmega 328p ilustrado na Figura 13, é o microcontrolador da placa Arduino UNO, sendo capaz de gerenciar a grande parte dos recursos da placa, suas características são descritas na Tabela 3.

Tabela 3 - Características do ATmega 328p.

Conjunto de Instruções	AVR Risc
Memória flash	32KB
EEPROM	1KB
SRAM	2KB
Linha de E/S (Entrada e Saída)	23
Registradores de propósito geral	32
Conversor A/D (Analogico Digital)	10 <i>bits</i>
Temporizadores	3
Interface de programação	USART e SPI
Clock interno	8 MHz
Tensão de operação	1,8-5,5 V

Fonte: Atmel Corporation (2016).

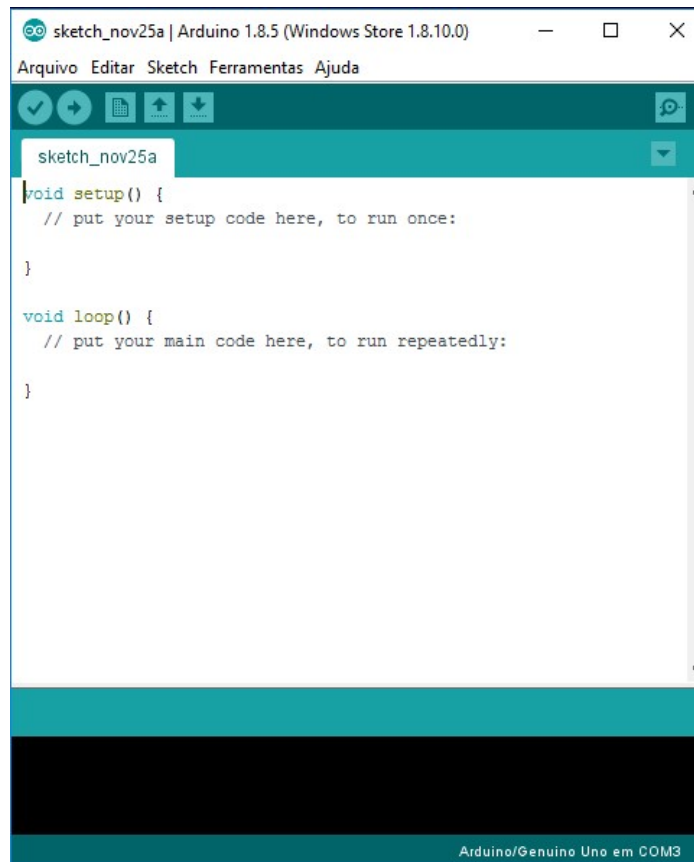
Figura 13 - Microcontrolador Atmega328p.



Fonte: https://image.ibb.co/cFt7HQ/atmega328p_pu.png

A placa Arduino se comunica com o computador por meio de uma IDE (*Integrated Development Environment*) ilustrada pela Figura 14, sendo este capaz de compilar o código programado pelo usuário e enviar à placa pela porta USB.

Figura 14 - Visual da IDE quando aberta.



Fonte: Autor (2018).

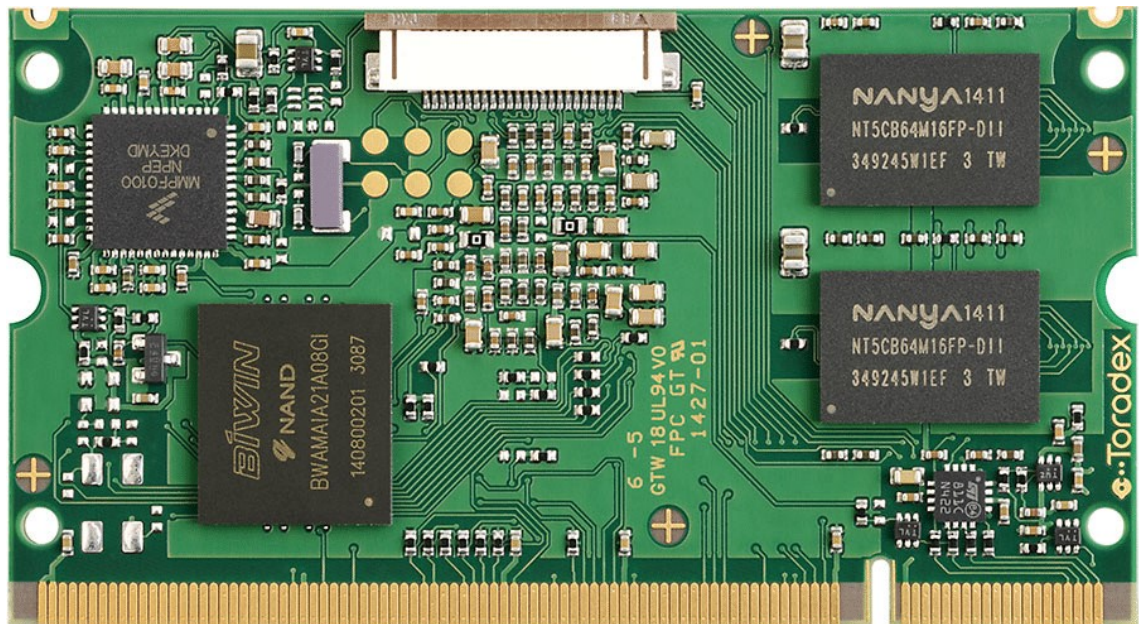
O microcontrolador é programado por uma linguagem denominada *Wiring*, sendo esta derivada do C/C++, por segundo Mcroberts (2011), essa linguagem torna-se ideal para a prototipação de projeto digitais mais rápida. Conforme a Wiring (2017), a linguagem é uma estrutura de programação de fonte aberta para microcontroladores, que permite a escrita de *software* multiplataforma para controlar dispositivos conectados a uma ampla gama de placas de microcontroladores para criar todos os tipos de codificação criativa, objetos interativos, espaços ou experiências físicas. A linguagem foi iniciada em 2003 por Hernando Barragán, anteriormente do *Interaction Design Institute Ivrea* e atualmente desenvolvido na Escola de Arquitetura e Design da Universidade de Los Andes, em Bogotá, Colômbia.

2.4.2. LINUX EMBARCADO

Para Sueiro (2013) um sistema Linux embarcado não se difere, no quesito sistêmico, de um sistema Linux *desktop*. A mesma estrutura e conceitos são aplicados em ambos os domínios. A principal diferença está nos requisitos de processamento, armazenamento, consumo de energia e confiabilidade.

Na maioria dos sistemas Linux embarcado exemplificado pela Figura 15, os recursos disponíveis são limitados e muitas vezes a interface com usuário é bastante limitada ou simplesmente não existe. Maciel (2014) também defende que o Linux Embarcado é a aplicação e uso do kernel Linux em uma placa eletrônica cujo principal elemento é o SoC (*System-on-a-chip*). SoC é um chip eletrônico que reúne todos os elementos básicos de um sistema computacional como CPU, memória principal (RAM), memória secundária (ROM/FLASH) e alguns periféricos. Atualmente um SoC contém internamente diversos periféricos dos mais variados tipos como I2C, SPI, UART, CAN, entre outros.

Figura 15 - Placa com Linux embarcado.



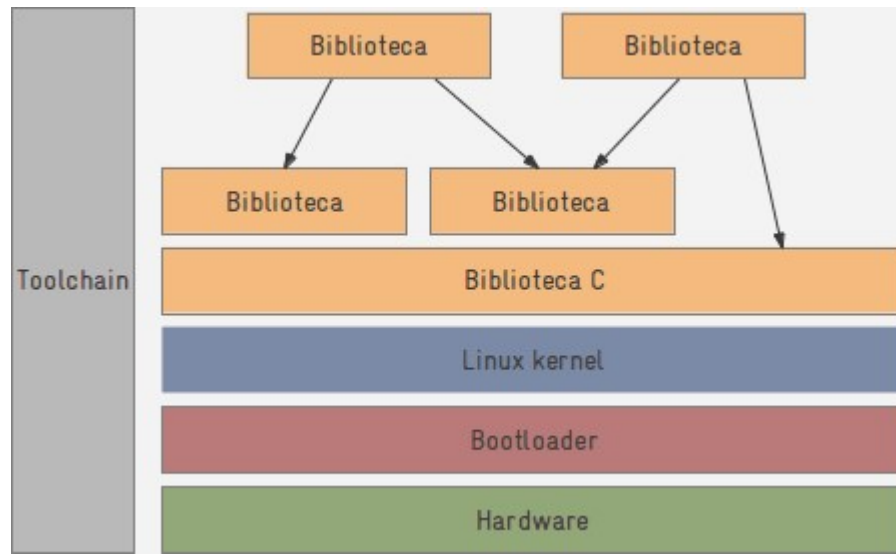
Fonte: <https://www.embarcados.com.br/pycharm-com-linux-embarcado/>

O Linux Embarcado apresenta basicamente em sua arquitetura características que são descritas abaixo e ilustradas pela Figura 16:

- *Hardware*, que é o produto;
- *Bootloader*, característicos de sistemas embarcados que é iniciado pelo hardware, responsável pela inicialização básica, carregamento e execução do kernel Linux;
- *Kernel Linux* que é o núcleo do sistema operacional. Gerencia CPU, memória e I/O, exportando serviços para as aplicações do usuário;

- *Roofs*, é o sistema de arquivos principal. Possui as bibliotecas do sistema para uso dos serviços exportados pelo *kernel*, além das bibliotecas e aplicações do usuário;
- *Toolchain*, conjunto de ferramentas para gerar os artefatos de *software* do sistema.

Figura 16 - Arquitetura básica de um Linux embarcado.



Fonte: <https://www.embarcados.com.br/anatomia-de-um-sistema-linux-embarcado/>

2.5. GOOGLE MAPS API

O Google é uma empresa gigante de tecnologia nos dias atuais, oferecendo muitas soluções para o mercado corporativos. Uma dessas soluções é o Google Maps, sendo um serviço de pesquisa e visualização de mapas e imagens de satélite da Terra gratuito na web. Segundo Lorenzoni (2015) atualmente, o serviço disponibiliza mapas e rotas para qualquer ponto nos Estados Unidos, Canadá, na União Europeia, Austrália e Brasil, entre outros. Disponibiliza também imagens de satélite do mundo todo, com possibilidade de um zoom nas grandes cidades, como Nova Iorque, Paris, São Paulo, Rio de Janeiro, Brasília, entre outras. Nos Estados Unidos o Google fez uma parceria com órgãos públicos, que incluirão as linhas de trem americanas e seus cruzamentos com rodovias ao Google Maps.

Afim de facilitar o uso dos recursos presente no Google Maps a fornecedora deste serviço criou uma "Interface de Programação de Aplicativos", que pode ser chamada de API, permitindo a criação de mapas com locais definidos, controle de zoom, tipos de mapa, geração de rotas, pesquisa por estabelecimentos, e muitas coisas mais (FERNANDES, 2009). A fabricante ainda diz que A Google Maps API foi projetada para carregar rapidamente e

funcionar bem em dispositivos móveis. Especificamente, nós nos concentramos no desenvolvimento para dispositivos móveis avançados, como celulares Android e IOS. Os dispositivos móveis têm telas menores que os navegadores normais no desktop (GOOGLE DEVELOPERS, 2018).

2.6. SISTEMA DE POSICIONAMENTO GLOBAL

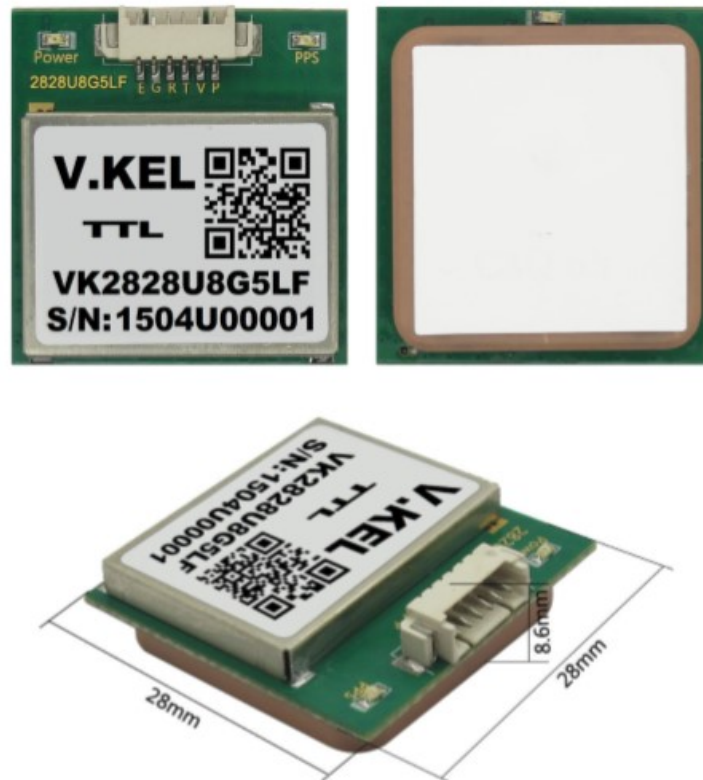
Desenvolvida no século XX a tecnologia do Sistema de Posicionamento Global foi desenvolvida, mas é conhecida popularmente por sua sigla GPS (acrônimo em inglês de *Global Positioning System*) que devido sua facilidade é bastante utilizado nos dias de hoje. De acordo com U.S. COAST GUARD (2017), o GPS é um sistema de navegação que utiliza como base os satélites em órbita da terra sendo desenvolvido e operado pelo Departamento de Defesa do Governo dos Estados Unidos da América que permite em qualquer lugar da terra que usuários determinem sua posição tridimensional, sua velocidade e o horário, 24 horas por dia, em todo tempo, em qualquer lugar do mundo, com uma precisão e exatidão superiores às de quaisquer outros sistemas de radionavegação atualmente disponíveis.

2.6.1. RECEPTORES GPS

No mercado atual o receptor GPS ilustrado pela Figura 17 é facilmente encontrado, cada um com suas devidas especificações variando de fabricante para fabricante. Em sua maioria eles apresentam características em comum como:

- Tensão de alimentação de 5 V;
- Interface de comunicação serial;
- Documento de dados relativo ao produto.

Figura 17 - Módulo GPS VK2828U7G5LF.



Fonte: V.kel Communications Equipment (2015).

Conta com um conector de 6 pinos que permite a sua usabilidade: PPS, VCC, TX, RX GND EN. A Tabela 4 descreve a função de cada pino.

Tabela 4 - Descrição dos pinos presente no conector do módulo VK2828U7G5LF.

Pino	Descrição
PPS	Pulsos de tempo de saída (pulsos por segundo).
VCC	Pino de alimentação, que pode ser de 3,3 volts até 5 volts.
TX	Pino de transmissão da interface UART/TTL serial.
RX	Pino de recepção da interface UART/TTL serial.
GND	Pino de aterramento.
EN	Pino de ativação, o nível lógico define ligado/desligado.

Fonte: O Autor (2018).

3. Resultados e Discursões

O sistema descrito nos capítulos anteriores foi construído, resultando em um protótipo funcional. Nesse capítulo detalha-se um conjunto de testes que foram realizados com o protótipo visando a sua validação. Foram realizados os seguintes testes:

- Comunicação entre transmissor e receptor;
 - Teste de comunicação *Indoor*;
 - Teste de Comunicação *Outdoor*;
- Comunicação entre receptor e o Linux embarcado;
- Experimento de Campo.

3.1. COMUNICAÇÃO ENTRE TRANSMISSOR E RECEPTOR

Após a construção de um protótipo funcional iniciou-se a primeira tentativa de comunicação entre o transmissor e o receptor. Observou-se que para pequenas distâncias o protótipo se demonstrou satisfatório, onde todos os dados enviados pelo transmissor chegaram ao receptor. Notou-se que em distâncias acima de 70 metros em lugares com obstáculos presentes, algumas das transmissões não chegaram de forma satisfatória. Essa perda de pacotes, contudo, já era esperada, sendo justificada, pois segundo a Digi International (2018) o alcance máximo dos módulos XBee é de apenas 90 metros em ambiente externos. Rodrigues (2014) também encontrou problemas ao utilizar módulos Zigbee em distância maiores que 30 metros, pois segundo ele os dispositivos funcionam com baixa potência impedindo um alcance maior.

Nesta etapa o Arduino ainda se faz necessário, pois necessita-se da programação no modo API do XBee receptor. O Arduino envia uma mensagem ao XBee que deve estar no modo API, esta mensagem conta com um comando que retorna à intensidade do sinal da última mensagem recebida do transmissor.

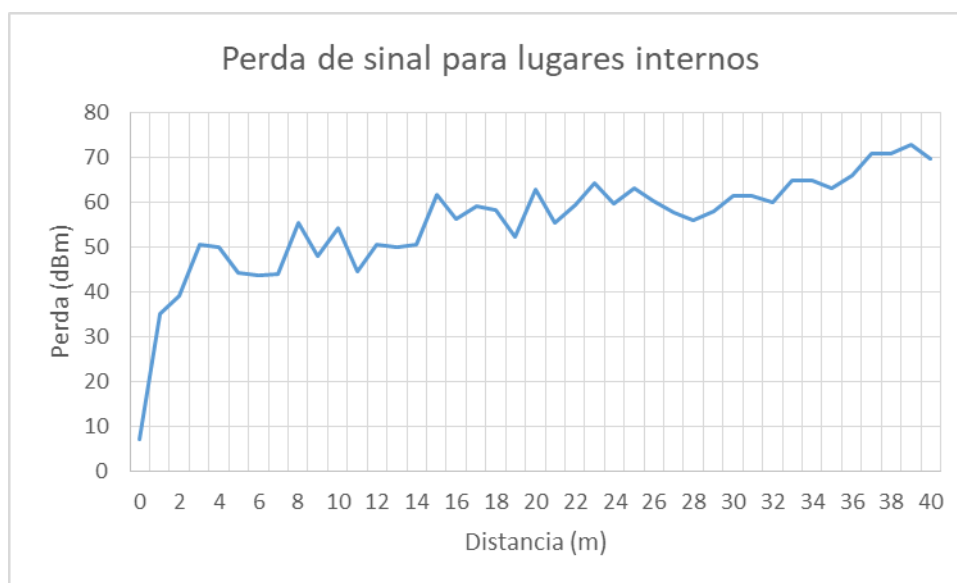
3.1.1. Testes em Comunicação Indoor

O experimento *indoor* ocorreu em um corredor presente no bloco de laboratórios da Universidade Federal do Semi-Árido - Câmpus Pau dos Ferros, localizada na BR-226 sem

número na cidade de Pau dos Ferros – RN, com coordenadas de latitude: $-6,106267^\circ$ e longitude: $-38,183940^\circ$. No corredor foi medido uma distância de 40 metros e dividida em distâncias de 1 metro. Iniciando do ponto 0 até o ponto de distância de 40 metros foram determinados 41 pontos, e em cada ponto foi medido o valor de perda do sinal. Este processo foi realizado 4 vezes de forma repetida, sendo que a vez que apresentou maior distância numérica entre as outras foi descartada.

Após os testes de perda de sinal em um local interno (comunicação *indoor*) obteve-se os resultados presentes na Tabela 5 na seção de Anexos, de perda de sinal expresso em dB em função da distância. Na Tabela 5 é possível observar claramente que a perda de sinal aumenta com a distância. Realizando análise estatísticas podemos obter valores de média e desvio padrão, já que foram realizadas 3 medições para cada distância. Após a análise estatística obteve-se os seguintes resultados para média e desvio padrão, que são apresentados na Tabela 6 também na seção de Anexos, e através destes os resultados podem ser visualizados no Gráfico 1.

Gráfico 1 - Perda do sinal em função da distância em lugares internos.



Fonte: Autor (2018).

Os valores de desvio padrão oscilam bastante dependendo da distância, sendo que em alguns pontos o desvio foi zero e em outro ele atingiu o pico de 14,43 dB. Este pico ocorreu num ponto distante 23 metros, quase que na metade da distância total. Afim de verificar melhor a variação dos dados de perda, foi feita a média dos valores de desvio padrão que para estes valores obtidos em testes feitos em um lugar interno, aferiu-se um número de 3,22 dB.

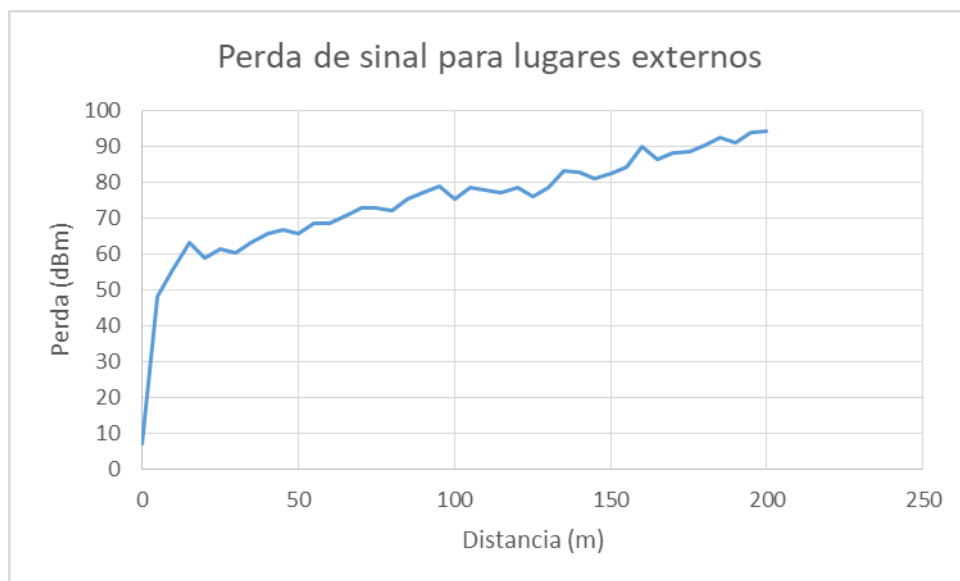
Com isso podemos afirmar que cada sinal varia em média, 3,22 dB do valor médio da perda de sinal naquela determinada distância. Este valor é pequeno, quando comparado a perda na distância zero (7 dB) que é maior que duas vezes a média de desvio padrão, mostrando que os testes foram bem realizado bem como que não há muita variação do sinal em um determinado ponto.

3.1.2. Testes em Comunicação Outdoor

O experimento *outdoor* ocorreu em uma estrada as margens da BR-405 na cidade de Pau dos Ferros, com coordenadas de latitude: $-6,1338415^\circ$ e longitude: -38.2084388° . Nesta estrada foi medida uma distância de 200 metros e dividida em distâncias de 5 metros. Iniciando do ponto 0 até o ponto de distância de 200 metros foram determinados 41 pontos, e em cada ponto foi medido o valor de perda do sinal. Este processo foi realizado 4 vezes de forma repetida, sendo que a vez que apresentou maior distância numérica entre as outras foi descartada.

Após os testes de perda de sinal em um ambiente externo (comunicação *outdoor*) obteve-se os resultados presente na Tabela 7 da secção de Anexos, de perda de sinal expresso em dB em função da distância. Observou-se que os resultados da Tabela 7, em ambiente externos, foram totalmente análogos a perda de sinal em lugar interno, onde a perda aumenta conforme um aumento na distância. Para que haja o melhor entendimento dos resultados também foi feito análise estatísticas. Após a análise estatística obteve-se os seguintes resultados para média e desvio padrão, que são apresentados na Tabela 8 também na secção de Anexos, e através destes os resultados podem ser visualizados no Gráfico 2.

Gráfico 2 - Perda do sinal em função da distância em lugares externos.



Fonte: Autor (2018).

No ambiente externo o desvio padrão das perdas também oscilou bastante, porem o pico de 4,75 dB foi na distância de 45 metros, sendo um pico bem menor que na análise de lugar fechado. Com um intervalo de desvio padrão de 0 a 4,75 dB podemos afirmar que a variação dos sinais de perda em ambientes externo é bem menor do que em ambientes internos ou lugares fechados. Calculando-se a média dos valores de desvio padra para ambiente externo obtém-se o número de 2,00 dB, menor do que a média de desvio padrão de ambiente interno, confirmando assim que a variação em ambientes externo é menor no que em ambientes internos. Afim de comparação dos resultados obtidos neste trabalho, comparando-se com o trabalho de Guerra (2012), intitulado “Caracterização do Canal de Propagação para Redes de TV Digital de Frequência Única”, que também realiza teste de perda de sinal em um ambiente *outdoor*, ele obteve um valor de desvio padrão médio igual a 11,63 dB. Já Clemente (2010) com o seu trabalho “Caracterização de Ambiente Celular GSM” obteve um valor de médio de desvio padrão de 2,47 dB, sendo bem próximo e acima do valor médio obtido neste trabalho. Com estes resultados fica evidente uma melhor eficiência do sinal em ambiente *outdoor*.

3.2. COMUNICAÇÃO ENTRE RECEPTOR E LINUX EMBARCADO

Com o transmissor em funcionamento emitindo todas a informações originadas do GPS e o receptor ligado ao Arduino, que recebe as informações oriundas do transmissor, filtrando as informações e convertendo as coordenadas, passa-se estes dados para o Linux

embarcado através de uma mensagem contendo as coordenadas convertidas e o número de satélites a qual o módulo GPS está conectado. Um exemplo desta mensagem pode ser vista abaixo:

-6.1035921;-38.1846879;10

Com o *script* em Python analisando estas mensagens e carregando as informações em um arquivo do tipo JSON temos a criação dos arquivos:

- “trajetoria.json”, que contém todas as coordenadas recebidas, exceto a última.
- “onibus.json”, que contém a última coordenada recebida pelo receptor.

Estes arquivos armazenam informações como a latitude, longitude e um nome do ponto, conforme o exemplo abaixo:

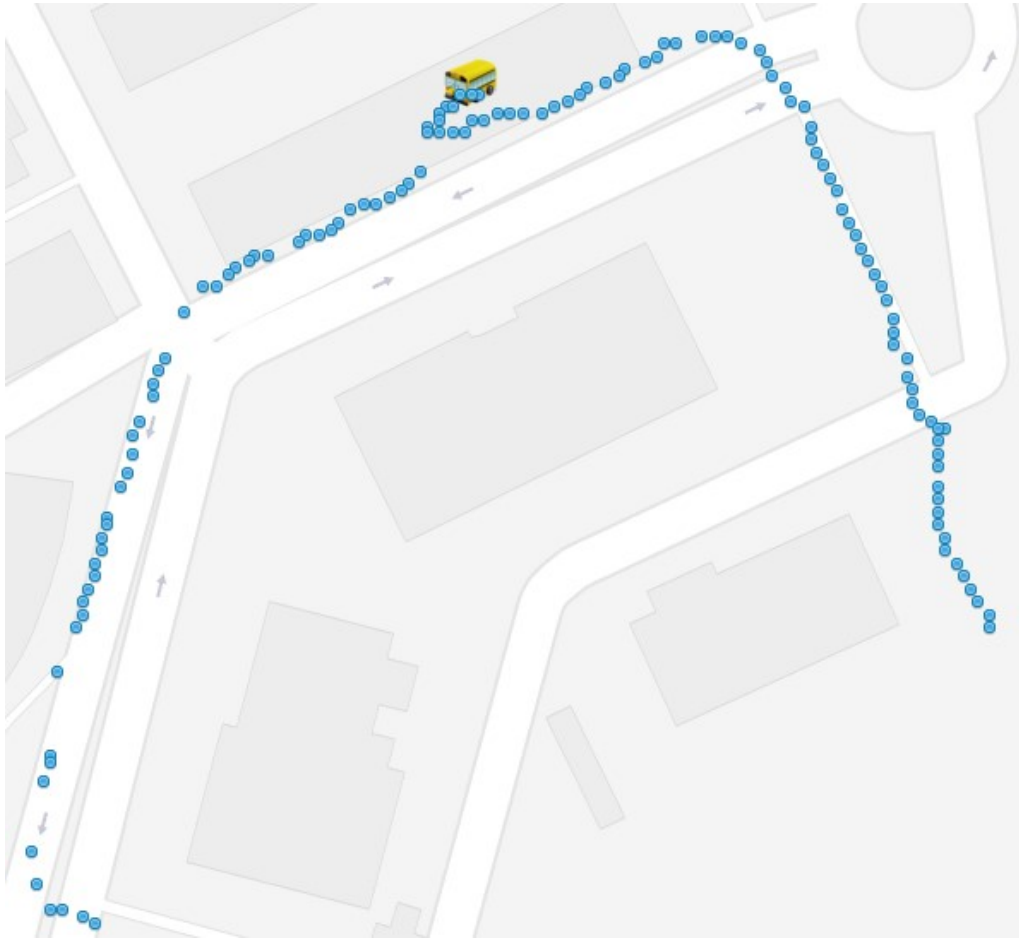
```
[{"lat": "-6.10454",
  "lon": "-38.18313",
  "nome": "Percurso"}]
```

Utilizando-se da API do Google Maps e de uma página web que faça essa sincronização entre os dados obtidos e a API podemos observar os pontos no mapa. Com a página web devidamente construída e pronta para exibir os dados presentes nos arquivos JSON.

3.3. REALIZAÇÃO DO EXPERIMENTO EM CAMPO

Para encerramento das atividades desenvolvidas, foi realizado um experimento em campo para os testes finais do protótipo implementado. O transmissor foi levado por meio de uma rota pré-definida dentro da UFERSA - Câmpus Pau dos Ferros. Em uma das salas presentes no bloco dos professores ficou instalado o Linux embarcado conectado ao receptor recebendo as informações obtidas do GPS. O Resultado é ilustrado na Figura 18:

Figura 18 - Teste de campo.



Fonte: Autor (2018).

Na Figura 18 verifica-se o trajeto percorrido pelo transmissor (pontos), no qual há pequenas variações na distância dos pontos. Cada ponto representa uma coordenada recebida, que posteriormente foi adicionada no mapa pelo Linux Embarcado. É visto que há variação na distância entre pontos, que pode ser explicada devido a uma possível variação na velocidade do módulo transmissor. Já que o período de envio das coordenadas é constante de 1 segundo, portanto se há uma variação na distância dos pontos, obviamente houve uma variação da velocidade do veículo, afinal se a velocidade fosse constante os pontos seriam igualmente espaçados (caso não houvesse falha de comunicação em nenhum momento). Observa-se também que com o aumento da distância entre o transmissor e receptor, a frequência de pontos vai caindo, isso já é notado no teste de perda de sinal, até chegar em um ponto que o sinal é perdido, retornando somente durante a aproximação do transmissor. A distância em linha reta que a transmissão foi interrompida é de 160 metros, enquanto a distância em que houve a volta da transmissão foi de 130 metros. Vemos que houve uma variação nas

distâncias de interrupção e retorno da transmissão, isso pode ser explicado devido aos prédios que se encontra em meio a comunicação entre o transmissor e o receptor, e estes prédios podem acentuar a perda do sinal sem fio, já que estes constituem de uma barreira física ao sinal. E um ambiente onde não há prédio entre a transmissão observa-se que a trajetória do veículo é marcada de forma constante mostrando assim o bom funcionamento do protótipo neste cenário.

O protótipo do sistema funcionou bem, havendo comunicação entres os módulos, o sistema é capaz de realizar as tarefas desejáveis. Dentro de uma área em que há comunicação entre transmissor e receptor, é possível obter todas as coordenadas enviadas pelo transmissor, já que por meio do receptor, o Linux embarcado recebe estas e mostra no mapa, sendo que a última coordenada recebida recebe o ícone do ônibus, e as passadas viram pontos. O mapa exibido para o usuário é atualizado de tempos em tempos fornecendo assim informações atualizadas quanto a posição do ônibus. Durante seu funcionamento não foram observadas falhas, já que mesmo com a perda de comunicação entre transmissor e receptor, o sistema continuava funcionando, mas sem novas informações da posição do ônibus, mostrando apenas a última coordenada recebida. Mas caso o receptor recebesse uma nova informação, o mapa seria atualizado com esta nova informação. Cada informação recebida é contabilizada posteriormente mostrada no mapa.

4. Conclusão

O trabalho desenvolvido comprova a possibilidade futura de expansão da rede de comunicação, fazendo com que o sistema de comunicação abranja toda a rota do ônibus, da UFERSA na cidade de Pau dos Ferros.

Afim de alcançar comunicação em todas as paradas feitas pelo ônibus dentro da cidade, o XBee utilizado neste trabalho poderia ser substituído por uma versão mais robusta, que tivesse uma cobertura bem maior do que o atual, ou até mesmo por outro que tenha a mesma funcionalidade, com uma área de cobertura bem maior que o atual. Com esse novo dispositivo o ônibus poderia ser mapeado por toda a cidade aumentando assim a utilização deste sistema e diminuindo as limitações encontradas neste trabalho.

Com o advento da telefonia móvel, que vem sendo cada vez mais presente no dia a dia das pessoas, visto que o celular é muito utilizado tanto para redes sociais quanto para busca de informações em aplicativos de transporte público. A criação de um aplicativo que mostrasse a posição do ônibus naquele instante seria bem atrativo para um trabalho futuro, facilitando assim a interação do usuário interessado com o sistema.

REFERÊNCIAS

- ANDRIOLLI, Marcelo. **Linux Embarcado**. 2018. Disponível em: <<http://e-stude.com/site/cursos/linux-embarcado/>>. Acesso em: 10 jan. 2018.
- ATMEL CORPORATION (Califórnia) (Org.). **ATmega328P**. 2016. Disponível em: <<http://www.atmel.com/pt/br/devices/atmega328p.aspx>>. Acesso em: 27 nov. 2017.
- CAVALCANTE, Gustavo Araújo. **Otimização de modelos de perdas de propagação aplicáveis a 3,5GHz usando algoritmos genéticos**. 2010. 75 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Federal do Rio Grande do Norte, Natal, 2010. Cap. 2.
- CLEMENTE, Douglas Henrique. **Caracterização de Ambiente Celular GSM**. 2010. Disponível em: <<http://www.teleco.com.br/tutoriais/tutorialalambcel2/default.asp>>. Acesso em: 12 jan. 2018.
- CHAVES, Fabiano de Sousa. **Controle de potência oportunista e equalização robusta em redes de comunicação sem fio: enfoques via controle automático e teoria dos jogos**. 2010. 267 f. Tese (Doutorado) - Curso de Engenharia Elétrica, Universidade Estadual de Campinas, Campinas, 2010. Disponível em: <<https://tel.archives-ouvertes.fr/file/index/docid/618079/filename/Chaves2010.pdf>>. Acesso em: 10 nov. 2017.
- CUNHA, Thiago, **Controle Centralizado de Equipamentos de Ar Condicionado via Rede Sem Fio ZigBee**, IFSC, São José-SC, 2013.
- DIGI INTERNATIONAL (Minnetonka). **XBee Product Manual v1**. Disponível em: <https://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwifoMOK_YLaAhVEiJAKHcMyCAMQFgg0MAA&url=https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf&usg=AOvVaw15918ksRTa3kvxWnRLDf-l>. Acesso em: 10 jan. 2018.
- ERICSSON (Sweden). **Why FWA makes sense today**. 2017. Disponível em: <<https://www.ericsson.com/en/networks/trending/hot-topics/fixed-wireless-access>>. Acesso em: 10 mar. 2018.
- EVANGELISTA, Davi Strill, **Integração de Redes de Sensores ZigBee para Automação Predial utilizando módulos MESHBEAN**. Universidade Federal de Brasília, 2010.

FERNANDES, Paulo. **Entendendo um pouco a API Google Maps**. 2009. Disponível em: <<https://imasters.com.br/artigo/13893/apis-google/entendendo-um-pouco-a-api-google-maps?trace=1519021197&source=single>>. Acesso em: 10 jan. 2018.

G1 (Palmas). **Saiba como funciona o sistema de monitoramento dos ônibus da capital**. 2014. Disponível em: <<http://g1.globo.com/to/tocantins/noticia/2014/07/saiba-como-funciona-o-sistema-de-monitoramento-dos-onibus-da-capital.html>>. Acesso em: 10 mar. 2018.

GISLASON, D. **Zigbee Wireless Networking**. Newnes, 2008.

GOOGLE DEVELOPERS. **Conceitos**: Desenvolvimento para dispositivos móveis. 2018. Disponível em: <<https://developers.google.com/maps/documentation/javascript/basics?hl=pt-br>>. Acesso em: 24 jan. 2018.

GUERRA, Maurício Vilela. **Caracterização do Canal de Propagação para Redes de TV Digital de Frequência Única**. 2012. 137 f. Tese (Doutorado) - Curso de Engenharia Elétrica, Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2012.

HAENGGI, Martin; SRINIVASA, Sunil. **Path loss exponent estimation in large wireless networks**. 2009. Disponível em: <ieeexplore.ieee.org/document/5044933>. Acesso em: 12 jan. 2018.

HAMILTON, Naomi. **The A-Z of Programming Languages: Python**. 2008. Disponível em: <https://www.computerworld.com.au/article/255835/a-z-programming_languages_python/?fp=4194304&fpid=1&pf=1>. Acesso em: 10 jan. 2018.

KUROSE, J.; Ross, W. K. **Redes de Computadores e a Internet – Uma Abordagem Top Down** 3ª Edição. Pearson, 2005.

LORENZONI, Leonardo. **Google Maps exibirá ferrovias nos Estados Unidos**. 2015. Disponível em: <<http://www.psaf.com/blog/google-maps-exibira-ferrovias-nos-estados-unidos/>>. Acesso em: 10 jan. 2018.

MACIEL, Francisco. **O que é Linux Embarcado**. 2014. Disponível em: <<http://softwarelivre.blog.br/author/fcomaciel/>>. Acesso em: 10 jan. 2018.

MAES, Nathalie, **ZigBee**, Technische Wetenschappen Geel Bachelor in de elektronica-ICT, 2014.

MCROBERTS, Michael. **Arduíno básico**. Novatec Editora, 2011.

MORAES, Alexandre Fernandes, **Redes de Computadores Fundamentos**, 6.ed.São Paulo: Érica Ltda, 2009.

NATIONAL INSTRUMENTS CORPORATION (Texas) (Org.). **Conceitos Gerais de Comunicação Serial**. 2015. Disponível em: <<http://digital.ni.com/public.nsf/allkb/32679C566F4B9700862576A20051FE8F>>. Acesso em: 27 nov. 2017.

NATIONAL MARINE ELECTRONICS ASSOCIATION (Severna Park). **The NMEA 0183 Protocol**. 2008. Disponível em: <<http://www.tronico.fi/OH6NT/docs/NMEA0183.pdf>>. Acesso em: 12 dez. 2017.

NOBELPRIZE.ORG. **Guglielmo Marconi**: Biographical. 1967. Disponível em: <<http://www2.ee.ufpe.br/codec/marconi.html>>. Acesso em: 23 mar. 2018.

REDMONK. **The RedMonk Programming Language Rankings**: January 2018. 2018. Disponível em: <<https://redmonk.com/sograzy/2018/03/07/language-rankings-1-18/>>. Acesso em: 7 mar. 2018.

RHYSIDER, Jack. **XBee S2 Quick Reference Guide**: Cheat Sheet and Video Tutorials to Getting Started. 2012. Disponível em: <<https://www.tunnelsup.com/xbec-guide/>>. Acesso em: 10 jan. 2012.

RODRIGUES, Glauco Oliveira. **REDE ADAPTATIVA WIRELESS PARA MONITORAMENTO DE ROTAS DO TRANSPORTE PÚBLICO**. 2014. 65 f. TCC (Graduação) - Curso de Curso Superior de Tecnologia em Redes de Computadores, Colégio Técnico Industrial de Santa Maria, Universidade Federal de Santa Maria, Santa Maria, 2014.

SILVA, R. **Características de Propagação ponto-área na faixa de 2 a 15GHz com Aplicação em Comunicações Móveis**. Dissertação de Mestrado. Instituto Militar de Engenharia. Rio de Janeiro, 2004.

SOUZA, Pedro Thiago Valério de. **Introdução aos Sistemas Embarcados**. Pau dos Ferros: Vídeo, 2017. 63 slides, color.

SUEIRO, Diego. **Anatomia de um Sistema Linux embarcado**. 2013. Disponível em: <<https://www.embarcados.com.br/anatomia-de-um-sistema-linux-embarcado/>>. Acesso em: 10 jan. 2018.

TANENBAUM, Andrew S, **Redes de Computadores: Tradução da Quarta Edição**, 4th ed. Rio de Janeiro: Elsevier, 2003.

TANENBAUM, Andrew. **Sistemas Operacionais Modernos**. 2a ed. Pearson - Prentice Hall. 2003. ISBN 85-87918-57-5.

TUDE, Eduardo. **Enlace Rádio Digital Ponto a Ponto**. 2004. Disponível em: <<http://www.teleco.com.br/pdfs/tutorialrdig.pdf>>. Acesso em: 10 nov. 2017.

U.S. COAST GUARD. **GENERAL INFORMATION ON GPS**. 2017. Disponível em: <<https://www.navcen.uscg.gov/?pageName=GPSmain>>. Acesso em: 5 dez. 2017.

V.KEL COMMUNICATIONS EQUIPMENT (Hong Kong). **Data Sheet: VK2828U7G5LF**. 2015. Disponível em: <[https://github.com/CainZ/V.KEL-GPS/blob/master/VK2828U7G5LF Data Sheet 20150902.pdf](https://github.com/CainZ/V.KEL-GPS/blob/master/VK2828U7G5LF%20Data%20Sheet%2020150902.pdf)>. Acesso em: 12 dez. 2017.

WIRING. **Wiring: About**. 2017. ISSN 2011-8376. Disponível em: <<http://wiring.org.co/about.html>>. Acesso em: 25 nov. 2017.

ZANOTTA, Daniel Capella; CAPPELLETTO, Eliane; MATSUOKA, Marcelo Tomio. O GPS: unindo ciência e tecnologia em aulas de física. **Revista Brasileira de Ensino de Física**, São Paulo, v. 2, n. 23, p.1-6, 12 jul. 2011. Disponível em: <<http://www.scielo.br/rbef/v33n2/a14v33n2.pdf>>. Acesso em: 5 dez. 2017.

ANEXO A – TABELAS DE PERDA DE SINAL

Tabela 5 - Perda do sinal em função da distância em lugares internos.

Distancia (m)	Perda (dB)	Perda (dB)	Perda (dB)
0	7	7	7
1	35	35	35
2	40	38	39
3	54	52	46
4	53	46	51
5	44	45	44
6	43	45	43
7	46	43	43
8	56	56	54
9	51	47	46
10	53	56	54
11	45	44	45
12	48	55	49
13	51	50	49
14	51	51	50
15	62	56	67
16	60	53	56
17	58	59	60
18	58	61	56
19	50	55	52
20	63	62	63
21	52	63	51
22	58	62	58
23	56	81	56
24	54	69	56
25	58	67	64
26	63	61	57
27	56	61	56
28	53	61	54

29	55	65	54
30	58	68	58
31	59	65	60
32	57	66	57
33	64	64	66
34	62	63	69
35	63	62	64
36	66	68	64
37	73	68	71
38	67	77	68
39	76	73	69
40	77	68	64

Fonte: Autor (2018).

Tabela 6 - Média e desvio padrão da perda de sinal para lugares internos.

Distancia (m)	Média (dB)	Desvio Padrão (dB)
0	7	0,00
1	35	0,00
2	39	1,00
3	51	4,16
4	50	3,61
5	44	0,58
6	44	1,15
7	44	1,73
8	55	1,15
9	48	2,65
10	54	1,53
11	45	0,58
12	51	3,79
13	50	1,00
14	51	0,58
15	62	5,51
16	56	3,51

17	59	1,00
18	58	2,52
19	52	2,52
20	63	0,58
21	55	6,66
22	59	2,31
23	64	14,43
24	60	8,14
25	63	4,58
26	60	3,06
27	58	2,89
28	56	4,36
29	58	6,08
30	61	5,77
31	61	3,21
32	60	5,20
33	65	1,15
34	65	3,79
35	63	1,00
36	66	2,00
37	71	2,52
38	71	5,51
39	73	3,51
40	70	6,66

Fonte: Autor (2018).

Tabela 7 - Perda do sinal em função da distância em ambientes externos.

Distancia (m)	Perda (-dB)	Perda (-dB)	Perda (-dB)
0	7	7	7
5	52	46	46
10	55	56	56
15	65	64	61
20	58	58	61

25	59	61	64
30	59	61	61
35	64	64	62
40	68	66	63
45	72	63	65
50	69	64	64
55	68	68	70
60	68	70	68
65	72	70	70
70	72	71	75
75	73	73	72
80	74	70	72
85	80	71	75
90	81	73	77
95	77	78	82
100	78	72	76
105	81	78	77
110	80	76	77
115	79	76	76
120	78	77	81
125	76	75	77
130	81	76	79
135	87	79	83
140	87	80	81
145	82	80	81
150	81	83	83
155	86	82	85
160	89	90	91
165	88	87	84
170	86	91	87
175	88	91	87
180	91	90	90
185	92	94	91

190	91	93	89
195	93	94	95
200	95	94	94

Fonte: Autor (2018).

Tabela 8 - Média e desvio padrão da perda de sinal para ambientes externos.

Distancia (m)	Média (dB)	Desvio Padrão (dB)
0	7	0,00
5	48	3,46
10	56	0,58
15	63	2,08
20	59	1,73
25	61	2,52
30	60	1,15
35	63	1,15
40	66	2,52
45	67	4,73
50	66	2,89
55	69	1,15
60	69	1,15
65	71	1,15
70	73	2,08
75	73	0,58
80	72	2,00
85	75	4,51
90	77	4,00
95	79	2,65
100	75	3,06
105	79	2,08
110	78	2,08
115	77	1,73
120	79	2,08
125	76	1,00

130	79	2,52
135	83	4,00
140	83	3,79
145	81	1,00
150	82	1,15
155	84	2,08
160	90	1,00
165	86	2,08
170	88	2,65
175	89	2,08
180	90	0,58
185	92	1,53
190	91	2,00
195	94	1,00
200	94	0,58

Fonte: Autor (2018).