

Rodrigo Stuqui Monzani

Estudo de Métodos de Localização Aplicados a Robôs Móveis Autônomos

Campo Grande, MS, Brasil

Novembro de 2023

Rodrigo Stuqui Monzani

Estudo de Métodos de Localização Aplicados a Robôs Móveis Autônomos

Trabalho de Conclusão de Curso (Relatório Técnico) apresentado ao Curso de Engenharia de Computação da Universidade Federal de Mato Grosso do Sul - UFMS, em cumprimento às exigências legais para a obtenção do título de bacharel em Engenharia de Computação, sob a orientação do Prof. Fábio Iaione.

Universidade Federal de Mato Grosso do Sul – UFMS

Faculdade de Computação

Campo Grande, MS, Brasil

Novembro de 2023

Resumo

Com a utilização cada vez mais presente de robôs móveis autônomos, a análise dos métodos de localização torna-se essencial para compreender o funcionamento e a interação desses sistemas. No caso em questão, examinaremos a implementação destes métodos para automatizar atividades no ambiente rural, incluindo o treinamento para competições de laço comprido. Uma análise minuciosa das tecnologias empregadas, tais como BLE e sensores, *encoder* e giroscópio, permitiu uma visão abrangente da capacidade de posicionamento desses robôs com um baixo custo.

Palavras-chaves: Localização. Robô móvel. Arduino. BLE. Sensores.

Lista de ilustrações

Figura 1 – Diagrama da estrutura de dados do iBeacon, ilustrando a distribuição e o tamanho dos diferentes campos de dados.	11
Figura 2 – Diagrama da estrutura de dados do Eddystone Beacon, ilustrando a distribuição e o tamanho dos diferentes campos de dados.	12
Figura 3 – Método de localização por triangulação.	13
Figura 4 – Estrutura simplificada de um <i>encoder</i> incremental.	14
Figura 5 – Sinais gerados por um <i>encoder</i>	14
Figura 6 – Exemplo de conexão I2C.	16
Figura 7 – <i>Pinout</i> do giroscópio MPU6050.	16
Figura 8 – Acelerômetro e giroscópio de 3 eixos do MPU6050 medindo aceleração e velocidade rotacional nos eixos X, Y e Z.	17
Figura 9 – Captura de dados do <i>beacon</i> transmissor a 1 m de distância.	19
Figura 10 – Robô móvel quadriciclo.	20
Figura 11 – Fixação do <i>encoder</i> ao lado do motor no robô móvel.	21
Figura 12 – Montagem experimental do circuito.	22
Figura 13 – <i>Shield</i> construído para montagem do circuito.	22
Figura 14 – Fluxograma do <i>firmware</i> do arduino com trajeto para treinamento de laço fechado.	23
Figura 15 – Dados de RSSI obtidos a 1 metro de distância.	25
Figura 16 – Localização do receptor utilizando BLE.	26
Figura 17 – Circuito com curvas localizado na UFMS.	27

Lista de tabelas

Tabela 1 – Definição da terminologia do barramento I2C.	15
Tabela 2 – Descrição da interface do giroscópio MPU6050.	17
Tabela 3 – Média de valores de n para cada distância medida.	25
Tabela 4 – Distâncias medidas com seus respectivos erros.	26
Tabela 5 – Distâncias medidas e seus respectivos erros.	27

Lista de abreviaturas e siglas

RTK	<i>Real Time Kinematic</i>
GNSS	<i>Global Navigation Satellite System</i>
BLE	<i>Bluetooth Low Energy</i>
RSSI	<i>Received Signal Strength Indication</i>
I2C	<i>Inter-Integrated Circuit</i>
SDA	<i>Serial Data</i>
SCL	<i>Serial Clock</i>
CSV	<i>Comma-Separated Values</i>
AoA	<i>Angle of Arrival</i>
UUID	<i>Universally Unique Identifier</i>
URL	<i>Uniform Resource Locator</i>
SSL	<i>Secure Sockets Layer</i>

Sumário

	Introdução	9
1	FUNDAMENTAÇÃO TEÓRICA	11
1.1	Beacon BLE	11
1.2	RSSI	12
1.3	Triangulação	13
1.4	<i>Encoder</i> incremental	14
1.5	Barramento I2C	15
1.6	Giroscópio MPU6050	16
2	METODOLOGIA	19
2.1	Localização por BLE	19
2.2	Movimentação com <i>encoder</i> e giroscópio	20
2.2.1	Acoplamento mecânico do <i>encoder</i>	20
2.2.2	Arquitetura do Hardware	21
2.3	<i>Firmware</i> Arduino	23
3	RESULTADOS	25
3.1	Localização por BLE	25
3.2	Movimentação com <i>encoder</i> e giroscópio	27
4	DISCUSSÃO E CONCLUSÃO	29
	REFERÊNCIAS	31

Introdução

Com o avanço contínuo da automação e robótica, a incorporação de robôs móveis autônomos tornou-se uma realidade em diversos setores. Esta evolução tecnológica não apenas automatiza tarefas cotidianas, mas também agiliza atividades repetitivas. Neste contexto, este trabalho visa aprofundar o entendimento sobre os métodos de localização que podem ser empregadas nesses robôs, destacando a implementação destas tecnologias para a automatização de atividades específicas, como as competições de laço comprido.

O laço comprido é uma competição em que o participante, montado em um cavalo, deve laçar um bovino solto em uma arena. Segurando o cavalo no brete até a soltura do bovino, a saída precoce resulta em penalidade. O laço de couro, com 18 a 20 metros, requer habilidade e sintonia entre o cavaleiro e o animal. Surgiu como método de retenção de gado, evoluindo para competições a partir dos anos 50 no Rio Grande do Sul. Atualmente, praticado principalmente no Rio Grande do Sul, Santa Catarina e Mato Grosso do Sul, o esporte não exige raça específica para os animais.

Esse trabalho abordará os métodos de localização *Bluetooth Low Energy* (BLE) e sensores como *encoder* e giroscópio para serem aplicados em um robô móvel utilizado para treino de laço comprido. Além disso, será explorada a viabilidade dessas tecnologias em termos de custo, proporcionando uma visão abrangente das soluções de baixo custo para a implementação desses sistemas. O GNSS foi descartado devido à intenção de desenvolver um sistema que seja aplicável tanto em ambientes externos quanto internos. Isso se deve à presença de pistas de laço com cobertura, tornando inviável a utilização do GNSS.

Dessa forma, ao compreender a integração de robôs autônomos em atividades específicas do ambiente rural, como as competições de laço comprido, podemos não apenas promover a eficiência operacional, mas também preservar e revitalizar práticas tradicionais por meio de abordagens tecnológicas modernas.

1 Fundamentação teórica

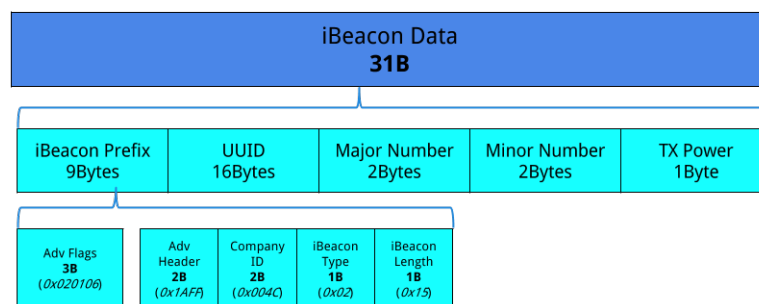
Esse capítulo abordará cinco tópicos normalmente relacionados a localização de robôs móveis: *Beacon* BLE, RSSI (*Received Signal Strength Indicator*), triangulação, *encoder* incremental e barramento I2C.

1.1 Beacon BLE

Os *beacons* são construídos com base no *Bluetooth* de Baixa Energia (BLE), um componente do padrão *Bluetooth* 4.0, composto por um *chipset*, uma bateria e uma antena. Este dispositivo de transmissão opera independentemente de uma conexão à internet, emitindo periodicamente pacotes *Bluetooth* contendo informações como a intensidade do sinal recebido (RSSI). Esses dados podem ser utilizados pelos receptores em intervalos regulares de tempo, sem a necessidade de conexão constante à rede (CAY et al., 2017).

É adicionado no *beacon* uma carga personalizada contendo o UUID exclusivo de 128 bits para identificar empresas ou entidades únicas, junto com dois valores de 16 bits ("*Major*" e "*Minor*") que possibilitam diferenciar estabelecimentos específicos (*Major*) e nós de *beacon* individuais (*Minor*).

Figura 1 – Diagrama da estrutura de dados do iBeacon, ilustrando a distribuição e o tamanho dos diferentes campos de dados.

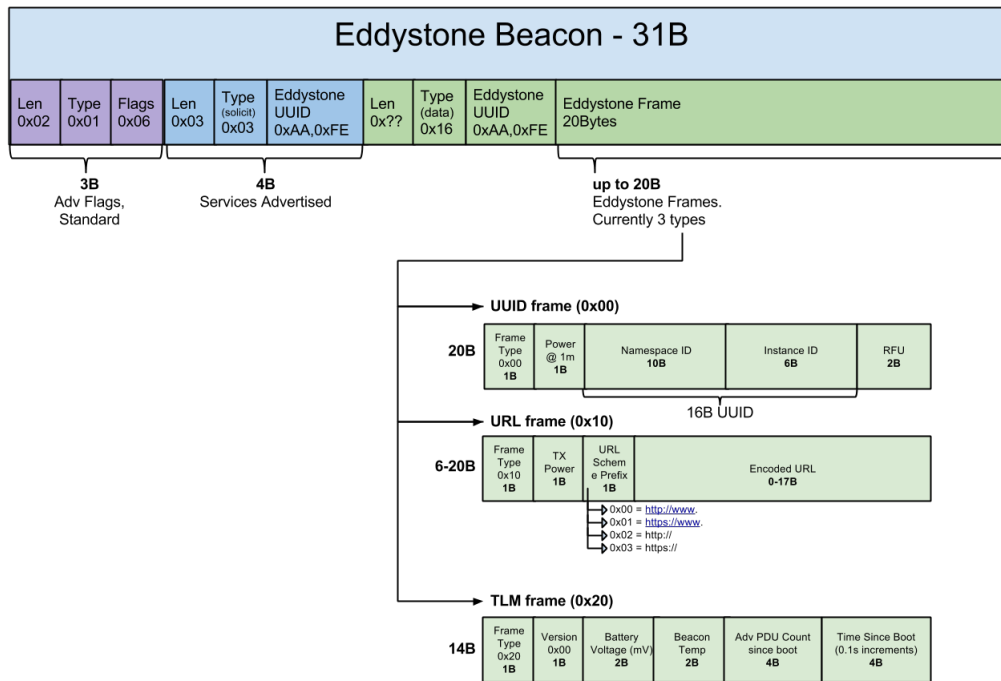


Fonte: Retirado de [Arm MBED](#)

A [Figura 1](#) exibe os campos de dados personalizados para o iBeacon, desenvolvido pela Apple. Além desse *beacon*, outro amplamente utilizado é o Eddystone Beacon, desenvolvido pela Google. Este último apresenta dados personalizados distintos em comparação com os demonstrados no iBeacon. Ao invés de transmitir apenas um pacote com informações, o Eddystone Beacon necessita transmitir três pacotes distintos: Eddystone-URL, Eddystone-UID e Eddystone-TML, conforme ilustrado na [Figura 2](#).

Cada protocolo de Eddystone é projetado para transmitir um tipo específico de informação. O Eddystone-URL, por exemplo, é utilizado para transmitir URLs de até 18 caracteres, redirecionando para sites protegidos por SSL. O Eddystone-UID transmite um código de identificação, permitindo que aplicativos recuperem informações dos servidores correspondentes. Esses códigos podem ser empregados em diversas aplicações, como localização interna, identificação de objetos físicos e interações personalizadas com aplicativos. O Eddystone-TML transmite dados sobre o farol, como nível da bateria, informações de sensores e outros dados relevantes para administradores de *beacons*.

Figura 2 – Diagrama da estrutura de dados do Eddystone Beacon, ilustrando a distribuição e o tamanho dos diferentes campos de dados.



Fonte: Retirado de [PARK et al. \(2017\)](#)

1.2 RSSI

O RSSI é um termo utilizado em comunicações sem fio para medir a potência do sinal recebido por um dispositivo. Normalmente é expresso em decibéis (dB) e fornece uma medida da potência do sinal em relação a um transmissor, sendo usado em tecnologias como Wi-Fi e bluetooth. Por meio dessa potência é possível calcular a distância entre os dispositivos conectados. A medição de distância usando o RSSI geralmente se baseia no modelo logarítmico da perda de percurso ([Equação 1.1](#)).

$$RSSI = -A(d_0) + 10n \log\left(\frac{d}{d_0}\right) \quad (1.1)$$

Na [Equação 1.1](#), d representa a distância entre o receptor e o transmissor, A é determinado pelo RSSI quando o transmissor está a uma distância d_0 , n é caracterizado pelo fator ambiental, sendo atribuído o valor de 2 para espaços abertos, com a possibilidade de aumento à presença crescente de obstáculos ([JUNG; LEE; HAN, 2011](#)). Para facilitar os cálculos, consideramos que d_0 é igual a 1 m, resultando na [Equação 1.2](#).

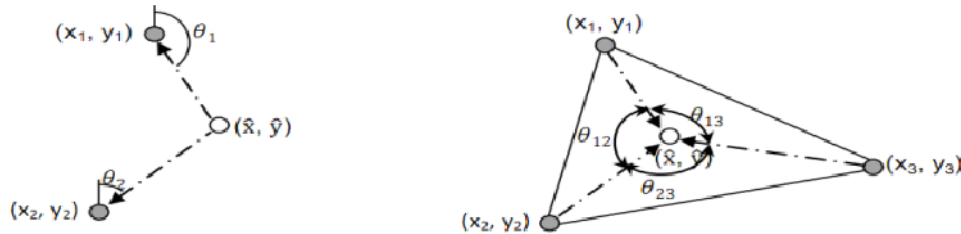
$$RSSI = -A + 10n\log(d) \quad (1.2)$$

1.3 Triangulação

Na área de localização, diversos métodos são empregados, como triangulação, trilateração, multilateração e *bounding box*. Essas técnicas utilizam informações como distâncias ou ângulos obtidos para determinar a posição de um objeto ou dispositivo.

A triangulação faz uso dos ângulos dos sensores em relação a uma referência, necessitando de no mínimo três sensores com localização conhecida. Alternativamente, é possível realizar a triangulação com apenas dois sensores, desde que a distância em relação à referência também seja conhecida, utilizando relações trigonométricas.

Figura 3 – Método de localização por triangulação.



Fonte: Retirado de [MEKELLECHE; HAFID \(2017\)](#)

A trilateração é um método utilizado quando as posições de três transmissores e as distâncias até o receptor são conhecidas. Nesse processo, são criados círculos com centros nas posições dos transmissores e raios correspondentes às distâncias até o receptor. A posição do receptor é determinada pela interseção desses círculos. A multilateração segue o mesmo princípio da trilateração, mas envolve mais de três transmissores, mantendo os princípios da trilateração.

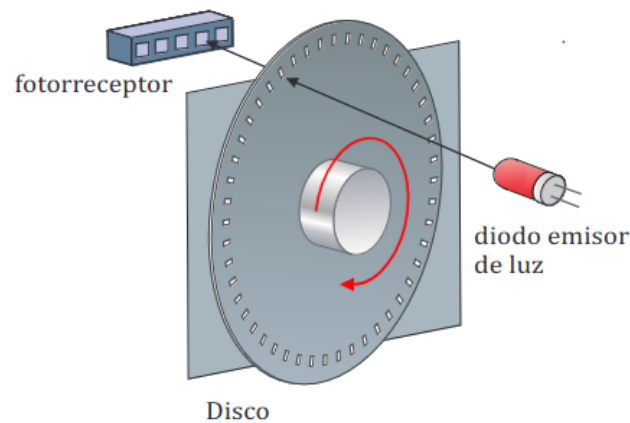
O *bounding box* também utiliza distâncias para calcular a posição, ao vez de criar círculos conforme feito na trilateração, são gerados quadrados. Isso reduz significativamente os cálculos necessários, embora haja uma perda de precisão no processo.

1.4 Encoder incremental

O *encoder* incremental é um dos transdutores mais utilizados quando se trata de posicionamento. O *encoder* fornece pulsos elétricos enquanto seu eixo gira, existindo uma grande variedade disponível no mercado, fabricados em diferentes tecnologias como eletromecânica, magnética e óptica.

Os *encoders* ópticos incrementais desempenham um papel fundamental na medição da velocidade e posição angular em motores e sistemas de robótica móvel. Em essência, eles atuam como os olhos desses dispositivos de navegação, utilizando uma configuração composta por uma fonte de luz, uma grade fixa, um disco do rotor com uma grade óptica e detectores ópticos. À medida que o rotor se movimenta, a luz atinge os detectores, criando pulsos quadrados (Figura 4).

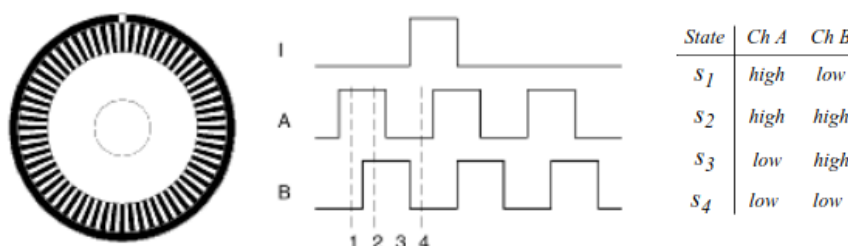
Figura 4 – Estrutura simplificada de um *encoder* incremental.



Fonte: Retirado de [ÁLVAREZ \(2019\)](#)

Uma abordagem comum é incorporar um segundo par emissor/detector, posicionados 90 graus deslocados em relação aos originais. Isso resulta em duas ondas quadradas distintas ([SIEGWART et al., 2004](#)). Com a implementação apropriada do *firmware*, é possível não apenas medir a posição e a velocidade, mas também determinar o sentido da rotação.

Figura 5 – Sinais gerados por um *encoder*.



Fonte: Retirado do [SIEGWART et al. \(2004\)](#)

1.5 Barramento I2C

Desenvolvido na década de 1980, o protocolo I2C apresentou uma inovação ao permitir uma comunicação eficiente entre diferentes componentes em uma placa de circuito, sendo utilizado em várias arquiteturas de controle, como o *System Management Bus* (SMBus), *Power Management Bus* (PMBus), *Intelligent Platform Management Interface* (IPMI), *Display Data Channel* (DDC) e *Advanced Telecom Computing Architecture* (ATCA).

Sua principal vantagem reside na habilidade de conectar múltiplos dispositivos utilizando apenas dois fios. O I2C facilita a interação entre dispositivos mestres e escravos, onde o mestre assume o papel de iniciar e controlar a comunicação, enquanto os dispositivos escravos respondem às solicitações do mestre. Cada dispositivo no barramento I2C possui um endereço único, o que permite ao mestre selecionar especificamente com qual dispositivo deseja interagir (NPX SEMICONDUCTORS, 2021).

No barramento I2C, dois fios, dados seriais (SDA) e relógio serial (SCL), transportam informações entre os dispositivos conectados ao barramento. Um dos canais desempenha a função de transmitir e receber dados, enquanto o outro canal, também por meio de um fio, é encarregado de fornecer o *clock*, controlando a velocidade e o sincronismo da comunicação. A Tabela 1 mostra alguns termos técnicos relacionados ao barramento I2C.

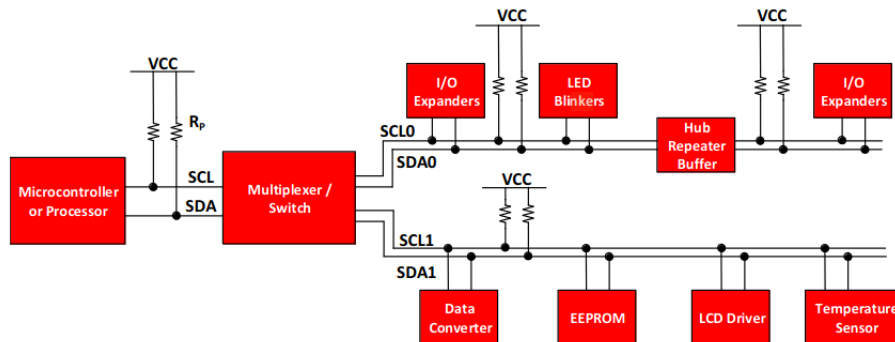
Tabela 1 – Definição da terminologia do barramento I2C.

Termo	Descrição
<i>Transmitter</i>	O dispositivo que envia dados para o barramento.
<i>Receiver</i>	O dispositivo que recebe dados do barramento.
<i>Controller</i>	O dispositivo que inicia uma transferência, gera sinais de <i>clock</i> e termina uma transferência.
<i>Target</i>	O dispositivo endereçado por um controlador.
<i>Multi-controller</i>	Mais de um controlador pode tentar controlar o barramento ao mesmo tempo sem corromper a mensagem.
<i>Arbitration</i>	Procedimento para garantir que, se mais de um controlador tentar controlar o barramento simultaneamente, apenas um é permitido fazer isso e a mensagem vencedora não é corrompida.
<i>Synchronization</i>	Procedimento para sincronizar os sinais de <i>clock</i> de dois ou mais dispositivos.

Fonte: Retirado de NPX SEMICONDUCTORS (2021)

A Figura 6 demonstra como o I2C trabalha em um sistema embarcado, controlando sensores, memórias, dispositivos de I/O, leds e outros.

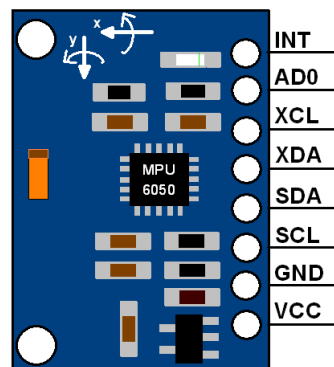
Figura 6 – Exemplo de conexão I2C.



Fonte: Retirado de [TEXAS INSTRUMENTS \(2015\)](#)

1.6 Giroscópio MPU6050

Esse dispositivo ([Figura 7](#)) integra giroscópio e acelerômetro em um único módulo, eliminando problemas de desalinhamento entre esses sensores. Na [Figura 7](#), o pino SCL desempenha o papel de *clock*, enquanto o pino SDA é responsável pela transmissão e recepção de dados ([TDK INVENSENSE, 2013](#)).

Figura 7 – *Pinout* do giroscópio MPU6050.

Fonte: Retirado de [Electronic Wings](#)

A [Tabela 2](#) mostra a função de cada sinal da interface do giroscópio.

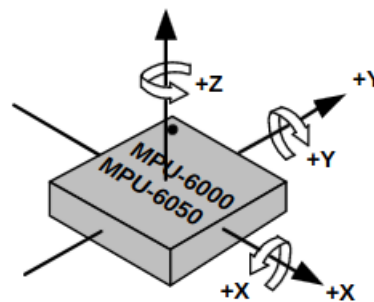
Tabela 2 – Descrição da interface do giroscópio MPU6050.

Pin Name	Description
Vcc	Fornece energia(+5V) para o módulo
Ground	Conecta ao ground do sistema
Serial Clock (SCL)	Prove o clock para a conexão I2C
Serial Data (SDA)	Prove a transferencia de dados para a conexão I2C
Auxiliary Serial Data (XDA)	Usado para conectar outras interfaces I2C(Opcional)
Auxiliary Serial Clock (XCL)	Usado para conectar outras interfaces I2C(Opcional)
AD0	Endereço do MPU6050
Interrupt (INT)	Indica que o dado esta pronto para ser lido

Fonte: Retirado de [Electronic Wings](#)

O giroscópio é um componente responsável por medir a taxa de rotação em torno dos três eixos espaciais: X, Y e Z. Ele funciona detectando alterações na orientação angular. Por outro lado, o acelerômetro é projetado para detectar acelerações lineares, incluindo a força da gravidade. Ele fornece informações sobre a inclinação e a orientação geral do dispositivo em relação à gravidade ou a qualquer aceleração linear aplicada. Ambos são mostrados na [Figura 8](#).

Figura 8 – Acelerômetro e giroscópio de 3 eixos do MPU6050 medindo aceleração e velocidade rotacional nos eixos X, Y e Z.



Fonte: Retirado de [TDK INVENSENSE \(2013\)](#)

2 Metodologia

A metodologia empregada no desenvolvimento desse trabalho possui duas partes principais. A primeira corresponde a implementação e testes da localização por meio do BLE (*Beacon*) e a segunda corresponde a implementação de testes de movimentação de um robô por meio do *encoder* e *giroscópio*.

2.1 Localização por BLE

A localização por meio do BLE foi implementada utilizando uma *Raspberry PI 4 2GB - Model B* como receptor e um *Ibeacon*, o *beacon* foi utilizado através de Iphones utilizando simulador chamado *Beacon Simulator* como transmissor.

A Raspberry Pi, inicialmente configurada para se comportar como um receptor de *beacon* (FERREIRA; ANTUNES, 2022), teve seus dados ajustados para aderir ao padrão mencionado na Figura 1. A modificação de dados foi realizada utilizando o BlueZ, um pacote disponível para o sistema operacional Raspbian. Esse ajuste foi necessário para garantir a correta recepção dos sinais transmitidos pelos iBeacons.

Após a configuração inicial, foi desenvolvido um programa em Python, utilizando bibliotecas específicas, para a leitura e processamento dos dados provenientes dos transmissores iBeacon. Estes dados, uma vez recebidos pela Raspberry Pi, foram armazenados de forma organizada em um arquivo CSV, facilitando a análise e manipulação posterior.

A tecnologia iBeacon permite a transmissão de pacotes contendo informações específicas como identificação única, potência do sinal e outros dados relevantes. O programa em Python foi projetado para capturar essas informações, diferenciando os pacotes com base nos dados transmitidos e o sinal RSSI de cada um dos pacotes recebidos, sendo possível assim calcular a distância (Figura 9).

Figura 9 – Captura de dados do *beacon* transmissor a 1 m de distância.

```
{'type': 'iBeacon', 'uuid': '66145bb8-08c5-4a35-b994-256cbf6318bc', 'major': 1, 'minor': 1, 'rssi': -57, 'macAddress': '57:32:8e:b5:4e:6a'}
{'type': 'iBeacon', 'uuid': '66145bb8-08c5-4a35-b994-256cbf6318bc', 'major': 1, 'minor': 1, 'rssi': -51, 'macAddress': '57:32:8e:b5:4e:6a'}
{'type': 'iBeacon', 'uuid': '66145bb8-08c5-4a35-b994-256cbf6318bc', 'major': 1, 'minor': 1, 'rssi': -51, 'macAddress': '57:32:8e:b5:4e:6a'}
{'type': 'iBeacon', 'uuid': '66145bb8-08c5-4a35-b994-256cbf6318bc', 'major': 1, 'minor': 1, 'rssi': -55, 'macAddress': '57:32:8e:b5:4e:6a'}
{'type': 'iBeacon', 'uuid': '66145bb8-08c5-4a35-b994-256cbf6318bc', 'major': 1, 'minor': 1, 'rssi': -55, 'macAddress': '57:32:8e:b5:4e:6a'}
{'type': 'iBeacon', 'uuid': '66145bb8-08c5-4a35-b994-256cbf6318bc', 'major': 1, 'minor': 1, 'rssi': -41, 'macAddress': '57:32:8e:b5:4e:6a'}
```

Fonte: Elaborado pelo Autor

Apos a implementação, foram realizados dezenas de testes de testes de localização com diferentes distâncias.

2.2 Movimentação com *encoder* e giroscópio

A movimentação com *encoder* e giroscópio foi implementada em um robô móvel (Figura 10) construído a partir de um quadriciclo, por uma empresa. O protótipo do robô móvel possui 0,9 m de comprimento, 0,53 m de largura e 0,74 m de altura, com massa de 44 kg. Possui um motor elétrico de 24V/350W, além de controles eletrônicos para controle da aceleração e direção.

Figura 10 – Robô móvel quadriciclo.



Fonte: Elaborado pelo Autor

2.2.1 Acoplamento mecânico do *encoder*

O *encoder* utilizado, modelo E38-S6-400-2-N-24, foi acoplado ao eixo do motor do robô por meio de duas polias e uma correia de borracha GT2 6mm. A fixação do *encoder* foi realizada por uma peça metálica em forma de "L" (Figura 11). O *encoder* gira 400 pulsos por rotação, possui sua saída um NPN (coletor aberto) e suporta uma rotação máxima de 5000 RPM.

Figura 11 – Fixação do *encoder* ao lado do motor no robô móvel.

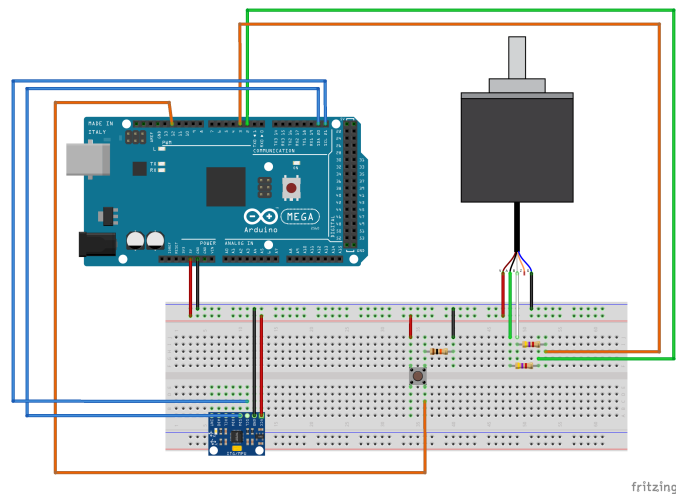
Fonte: Elaborada pelo autor

Para calcular o deslocamento linear do *encoder* por pulso, considerou-se o raio da roda do robô móvel, a relação entre a quantidade de dentes da engrenagem do eixo da roda com a quantidade de dentes da engrenagem do eixo do motor (5:1), e a relação entre a polia do eixo do motor com a polia do eixo do *encoder* (2:1). Com um raio de roda de 0,125 m, determinou-se que a cada rotação da roda, o robô móvel se desloca aproximadamente 0,785 m. Levando em conta as relações entre engrenagens e polias, concluiu-se que o *encoder* se move 0,0785 m por volta. Portanto, o deslocamento linear de cada pulso do *encoder* foi calculado como sendo de 0,196 mm por pulso.

2.2.2 Arquitetura do Hardware

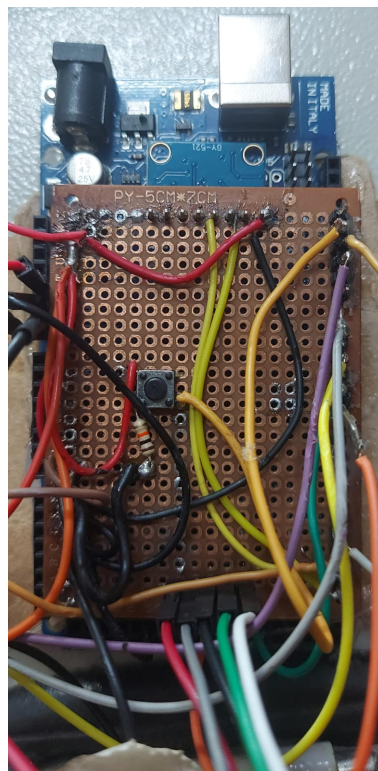
A Figura 12 mostra a montagem do circuito em uma matriz de contato. Esse circuito inclui um arduino mega 2560, o giroscópio MPU6050, o *encoder* e seus resistores *pull up*, e uma chave tátil que serve para ativar o modo automático, enquanto a chave tátil não é pressionada, o robô móvel mantém o modo manual.

Figura 12 – Montagem experimental do circuito.



Fonte: Elaborado pelo Autor

A utilização da matriz de contato em contextos práticos não é eficiente devido à possibilidade de conexões instáveis e a dificuldade de manter um circuito organizado. Como alternativa, foi construído um *shield* específico para o arduino (Figura 13), no qual todo o conjunto necessário foi montado. Este circuito é alimentado pela tensão de 5V fornecida pelo próprio arduino (ARDUINO DOCS, 2023a). O arduino por sua vez é alimentado com 12V, fornecido pela bateria do robô móvel.

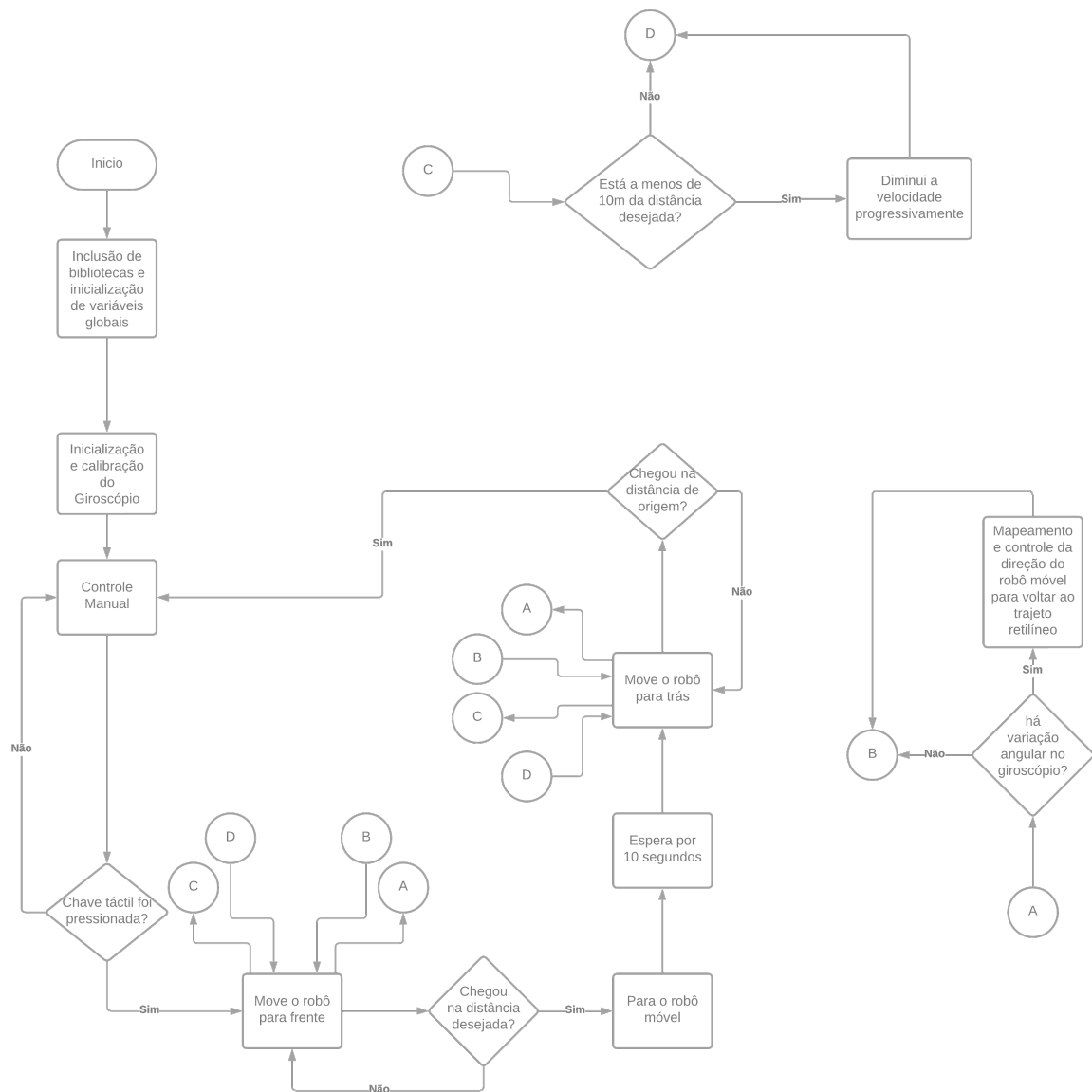
Figura 13 – *Shield* construído para montagem do circuito.

Fonte: Elaborada pelo autor

2.3 Firmware Arduino

O *firmware* foi desenvolvido utilizando-se a IDE Arduino. A Figura 14 mostra um fluxograma do *firmware*.

Figura 14 – Fluxograma do *firmware* do arduino com trajeto para treinamento de laço comprido.



Fonte: Elaborada pelo autor

Na calibração do giroscópio, é crucial que o dispositivo permaneça imóvel por alguns segundos para estabilização. Então são coletados aproximadamente 200 valores do giroscópio, e, em seguida, é calculada a média de suas variações. Essa média é utilizada para ajustar o zero, garantindo um desempenho mais preciso em suas leituras.

Sempre que o giroscópio detecta qualquer desvio em no trajeto linear do robô, o

firmware imediatamente identifica essa movimentação através do giroscópio e realiza um realinhamento com o trajeto designado. Isso é realizado através do mapeamento do ângulo de desvio detectado pelo giroscópio com o ângulo de movimentação da roda.

Para o *encoder*, os sinais A e B, conforme ilustrado na [Figura 5](#), são lidos utilizando interrupções geradas na borda de subida do sinal ([ARDUINO DOCS, 2023b](#)). Quando A está na borda de subida e B está em nível lógico baixo, realiza-se a adição correspondente à distância percorrida por pulso; da mesma forma, ocorre a subtração correspondente à distância percorrida quando B está na borda de subida e A está em nível lógico baixo.

Com todo o sistema configurado, ele espera que a chave tátil(botão) seja pressionada para começar a fazer o trajeto indicado, caso contrario, ele permanece no modo manual sendo controlado por um controle remoto.

Após a implementação, foram realizados dezenas de testes de movimentação, com diferentes distâncias de deslocamento.

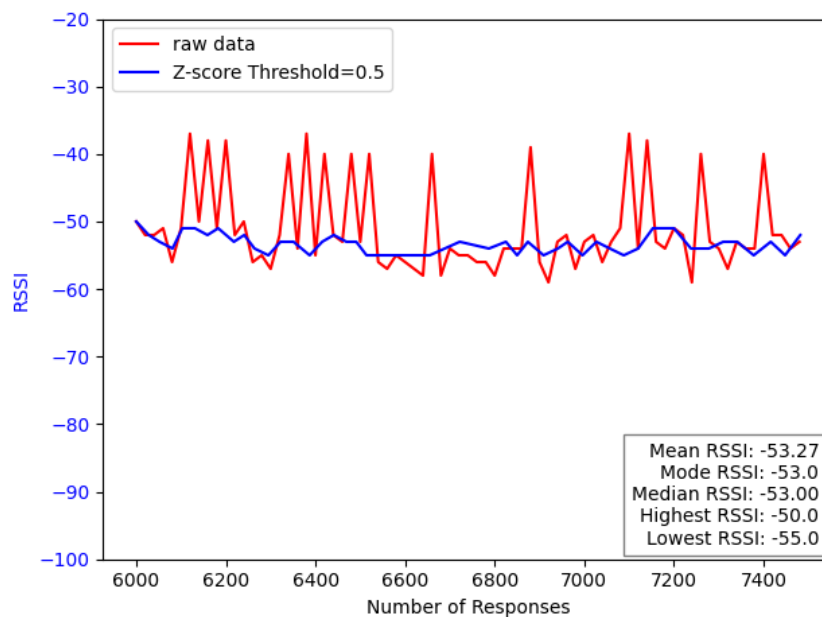
3 Resultados

Esse capítulo sintetiza os resultados obtidos na localização por BLE e na movimentação com *encoder* e giroscópio.

3.1 Localização por BLE

Nos testes iniciais, foi necessário determinar o valor de A conforme apresentado na [Equação 1.2](#). Para isso, procedemos à coleta de dados a uma distância de 1 metro. Como os dados apresentam uma dispersão elevada, foi aplicada uma técnica de remoção de *outliers* (Z -score com *threshold* de 0,5), como demonstrado na [Figura 15](#). O valor de A é dado pela média dos valores coletados após a remoção de *outliers*.

Figura 15 – Dados de RSSI obtidos a 1 metro de distância.



Fonte: Elaborado pelo Autor

Com base nos dados coletados para distâncias de 2, 5, 10, 15, 20, 25, 30, 40, 50 e 65 m, é necessário determinar o valor de n na [Equação 1.2](#). A [Tabela 3](#) foi construída utilizando esses dados para facilitar a determinação do valor do n médio.

Tabela 3 – Média de valores de n para cada distância medida.

Distancia(m)	2	5	10	15	20	25	30	40	50	65	n
Media de n	1.55	2.78	2.09	1.71	1.97	2.02	1.66	2.23	2.01	2.1	2,012

Fonte: Elaborado pelo Autor

Na [Tabela 4](#) pode-se observar o aumento progressivo do erro conforme a distância também aumenta. Isso pode ser atribuído a interferências eletromagnéticas externas, diminuindo drasticamente sua precisão.

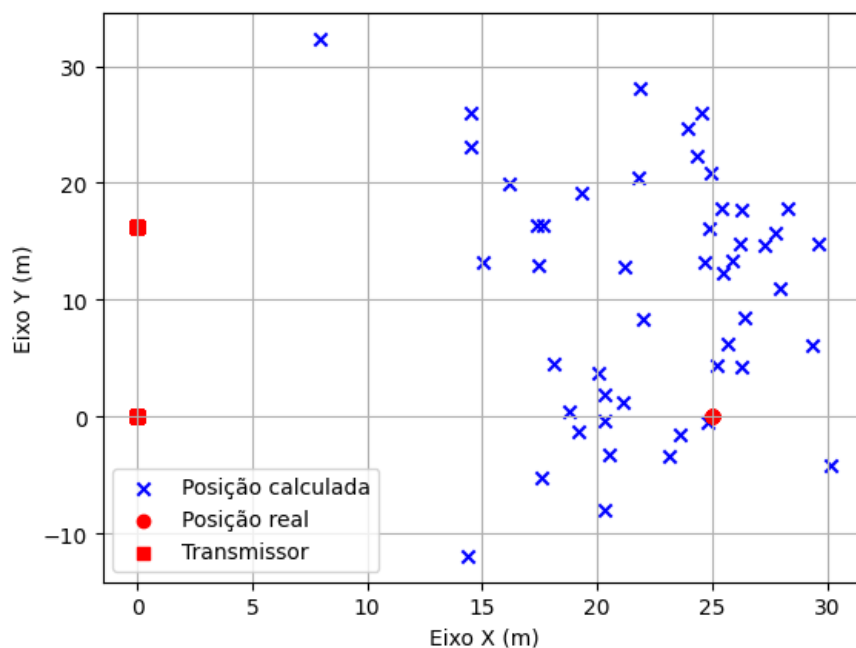
Tabela 4 – Distâncias medidas com seus respectivos erros.

Distância real(m)	Distância Medida(m)	Erro absoluto(m)
1	1.09	0.09
2	2.18	0.18
5	10.60	5.60
10	13.19	3.19
15	11.28	-3.72
20	22.43	2.43
25	29.89	4.89
30	20.19	-9.81
40	69.87	29.87
50	57.70	7.70
65	93.75	28,75

Fonte: Elaborado pelo autor

Para o posicionamento do receptor, foram utilizados dois transmissores a 25 m e 21,85 m de distância do receptor, enquanto a distância entre os transmissores era de 16,2 metros, utilizando técnicas de triangulação, foram obtidos resultados da [Figura 16](#).

Figura 16 – Localização do receptor utilizando BLE.



Fonte: Elaborado pelo Autor

3.2 Movimentação com *encoder* e giroscópio

Como citado anteriormente, esse teste foi realizado com robô móvel para prova de laço comprido. Esse teste objetivou verificar a habilidade do robô em percorrer caminhos pré-determinados, em condições de baixa complexidade com poucos obstáculos.

Durante os testes iniciais, constatou-se que, em velocidades elevadas, o arduino não conseguia captar os pulsos enviados pelo *encoder*. Posteriormente, identificou-se que a frequência gerada pelo *encoder* era maior do que a frequência de leitura dos sinais do arduino. Para solucionar esse problema, foi necessário aumentar a frequência padrão de leitura dos pinos do *chipset* ATmega2560 ([ATMEL, 2014](#)).

O robô deslocou-se na grama a velocidade aproximada de $10,8 \text{ km h}^{-1}$, percorrendo distâncias de 2, 5, 10, 25 e 75 m. Essas variadas distâncias foram escolhidas para avaliar o desempenho do robô em diferentes trajetos lineares, proporcionando uma análise abrangente de sua capacidade de navegação.

Tabela 5 – Distâncias medidas e seus respectivos erros.

Distância desejada(m)	Distância obtida(m)	Erro Absoluto(m)
2	2,08 (Ida)	0,06 (Ida)
	2,19 (Volta)	0,11 (Volta)
5	5,348 (Ida)	0,348 (Ida)
	5,637 (Volta)	0,289 (Volta)
10	10,74 (Ida)	0,74 (Ida)
	11,3 (Volta)	0,56 (Volta)
25	26,60 (Ida)	1,60 (Ida)
	28,5 (Volta)	1,90 (Volta)
75	82,2 (Ida)	7,2 (Ida)
	84,8 (Volta)	2,6 (Volta)

Fonte: Elaborado Pelo Autor

Também foi feito um segundo teste em uma pista com curvas, com a ideia de seguir todo o trajeto da pista mostrado na [Figura 17](#).

Figura 17 – Circuito com curvas localizado na UFMS.



Fonte: Retirado de [Google Earth](#)

No segundo teste, o *encoder* foi testado para percorrer uma pista com curvas com dimensões de 80 metros de comprimento por 15 metros de largura. Durante o trecho linear de aproximadamente 70 metros, o robô demonstrou um deslocamento de cerca de 76 metros antes de iniciar a curva, momentaneamente saindo da pista. Em seguida, a curva foi iniciada com um ângulo na roda de direção escolhido de maneira arbitrária para curvatura em relação ao percurso, fazendo o robô retornar à pista após concluir a curva. Este teste foi conduzido para avaliar a capacidade do robô móvel em realizar curvas de maneira estável.

4 Discussão e conclusão

O objetivo deste trabalho foi avaliar métodos de baixo custo que possibilitam a localização em ambientes internos e externos aplicados a robôs moveis. Realizou-se um estudo teórico sobre o assunto e para a montagem de um sistema de baixo custo, foram escolhidas as tecnologias com BLE e *encoder*/giroscópio.

A implementação do BLE não se mostrou eficaz na resolução do problema em questão devido à considerável variação nos dados obtidos e à baixa precisão alcançada. A variação nos dados, que pode ser provocada por interferências eletromagnéticas externas, obstáculos físicos ou condições ambientais desfavoráveis, resulta em medidas imprecisas e inconsistentes. Como sugestões de melhoria, o uso de inteligência artificial para prever a localização (MADURANGA; ABEYSEKERA, 2022) ou a utilização das versões de bluetooth mais recentes que possuem a tecnologia de AoA (COMINELLI; PATRAS; GRINGOLI, 2019).

A utilização conjunta de um *encoder* e um giroscópio mostrou-se eficaz no deslocamento longas distâncias com uma baixa perda de precisão, sendo uma solução eficaz para o problema dado. Cabe lembrar que cada pulso no *encoder* corresponde a um deslocamento de 0,196 mm. No entanto, essa abordagem apresenta a desvantagem de ser mais adequada para operação em ambientes com poucos obstáculos, a fim de evitar alterações drásticas na rota do robô móvel.

Como trabalhos futuros, a implementação de visão computacional para que o robô retorne automaticamente para a posição zero, além de sensores para proteção contra colisão.

Ambos os métodos se destacam pelo baixo custo para construir o sistema. O BLE possui módulos com um custo aproximado de R\$20,00, enquanto a combinação do *encoder* (aproximadamente R\$200,00) e do giroscópio (aproximadamente de R\$20,00) também se mostra economicamente vantajosa.

Por fim, acredita-se que este trabalho fez uma importante contribuição para o desenvolvimento de um robô móvel autônomo para ser utilizado no treinamento de laço comprido.

Referências

ARDUINO DOCS. *Arduino MEGA 2560 Rev3 Product Reference Manual*. [S.l.], 2023. Disponível em: <<https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf>>. Acesso em: 19.11.2023. Nenhuma citação no texto.

ARDUINO DOCS. *attachInterrupt()* - Documentação de referencia arduino. [S.l.], 2023. Disponível em: <<https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>>. Acesso em: 19.11.2023. Nenhuma citação no texto.

ATMEL. *8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash*. [S.l.], 2014. Disponível em: <https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf>. Acesso em: 19.11.2023. Nenhuma citação no texto.

CAY, E. et al. Beacons for indoor positioning. In: *2017 International Conference on Engineering and Technology (ICET)*. [S.l.: s.n.], 2017. p. 1–5. Nenhuma citação no texto.

COMINELLI, M.; PATRAS, P.; GRINGOLI, F. Dead on arrival: An empirical study of the bluetooth 5.1 positioning system. In: *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*. New York, NY, USA: Association for Computing Machinery, 2019. (WiNTECH '19), p. 13–20. ISBN 9781450369312. Disponível em: <<https://doi.org/10.1145/3349623.3355475>>. Nenhuma citação no texto.

FERREIRA, B. M. B.; ANTUNES, W. P. C. Estudo e desenvolvimento de gateways e beacons ble para rastreamento de pessoas em ambientes fechados. *Revista Ibero-Americana de Humanidades, Ciências e Educação*, 04 2022. Nenhuma citação no texto.

JUNG, S.; LEE, C. oh; HAN, D. Wi-fi fingerprint-based approaches following log-distance path loss model for indoor positioning. In: *2011 IEEE MTT-S International Microwave Workshop Series on Intelligent Radio for Future Personal Terminals*. [S.l.: s.n.], 2011. p. 1–2. Nenhuma citação no texto.

MADURANGA, M.; ABEYSEKERA, R. Bluetooth low energy (ble) and feed forward neural network (ffnn) based indoor positioning for location-based iot applications. *International Journal of Wireless and Microwave Technologies*, 2022. Disponível em: <<https://api.semanticscholar.org/CorpusID:248616076>>. Nenhuma citação no texto.

MEKELLECHE, F.; HAFID, H. Classification and comparison of range-based localization techniques in wireless sensor networks. *Journal of Communications*, v. 12, p. 221–227, 04 2017. Nenhuma citação no texto.

NPX SEMICONDUCTORS. *I2C-bus specification and user manual*. [S.l.], 2021. Rev. 7.1. Nenhuma citação no texto.

PARK, Y.-S. et al. Temperature detection and monitoring system of livestock through ear-tag based on iot. *Journal of Korea Multimedia Society*, v. 20, p. 474–481, 03 2017. Nenhuma citação no texto.

SIEGWART, R. Y. et al. Introduction to autonomous mobile robots. In: . [s.n.], 2004. Disponível em: <<https://api.semanticscholar.org/CorpusID:107033282>>. Nenhuma citação no texto.

TDK INVENSENSE. *MPU-6000 and MPU-6050 Register Map and Descriptions*. [S.l.], 2013. Rev. 4.2. Nenhuma citação no texto.

TEXAS INSTRUMENTS. *Understanding the I2C Bus*. [S.l.], 2015. Nenhuma citação no texto.

ÁLVAREZ, J. A. C. *Diseño de un Robot Autobalanceado*. Tese (Doutorado) — Universidad Tecnológica de la Habana, 06 2019. Nenhuma citação no texto.