UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA – EST

MATEUS DE SOUSA PANTOJA

DESENVOLVIMENTO DE PROTÓTIPO DE SISTEMA PARA DETERMINAR O ÂNGULO DE CHEGADA DO SINAL *BLUETOOTH* APLICADO NA LOCALIZAÇÃO EM AMBIENTES INDOOR UTILIZANDO TECNOLOGIA *DIRECTION FINDING*.

Manaus
2022

MATEUS DE SOUSA PANTOJA

DESENVOLVIMENTO DE PROTÓTIPO DE SISTEMA PARA DETERMINAR O ÂNGULO DE CHEGADA DO SINAL *BLUETOOTH* APLICADO NA LOCALIZAÇÃO EM AMBIENTES *INDOOR* UTILIZANDO TECNOLOGIA *DIRECTION FINDING*.

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentado à banca avaliadora do Curso de Engenharia Eletrônica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro em Eletrônica.

Orientador: André Luiz Printes, Me.

Manaus

2022

Universidade do Estado do Amazonas – UEA
Escola Superior de Tecnologia - EST

Reitor:
André Luiz Nunes Zogahib
Vice-Reitor:
Kátia do Nascimento Couceiro
Diretor da Escola Superior de Tecnologia:
Ingrid Sammyne Gadelha Figueiredo
Coordenador do Curso de Engenharia Eletrônica:
Bruno da Gama Monteiro

Banca Avaliadora composta por: Data da defesa: <21/10/2022>.
***Prof. André Luiz Printes, Me** (Orientador)*
Prof. Fábio de Sousa Cardoso, Dr
Prof. Rubens de Andrade Fernandes, Me.

CIP – Catalogação na Publicação

Pantoja, Mateus de Sousa

Desenvolvimento de Protótipo de Sistema para Determinar o Ângulo de Chegada do Sinal Bluetooth aplicado na Localização em Ambientes Indoor Utilizando Tecnologia Direction Finding / Mateus de Sousa Pantoja; [orientado por] André Luiz Printes. – Manaus: 2022.
54 p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Eletrônica). Universidade do Estado do Amazonas, 2022.

1. Localização *Indoor*. 2. *Bluetooth Direction Finding*. 3. nRF52833. I. Printes, André Luiz.

MATEUS DE SOUSA PANTOJA

DESENVOLVIMENTO DE PROTÓTIPO DE SISTEMA PARA DETERMINAR O ÂNGULO DE CHEGADA DO SINAL *BLUETOOTH* APLICADO NA LOCALIZAÇÃO EM AMBIENTES *INDOOR* UTILIZANDO TECNOLOGIA *DIRECTION FINDING.*

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Eletrônica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro Eletrônico.

Nota obtida: _____ (_____)

Aprovada em _____/_____/_____.

Área de concentração: Sistemas Embarcados

BANCA EXAMINADORA

_____
Orientador: André Luiz Printes, Me.

_____
Avaliador: Fábio de Sousa Cardoso, Dr.

_____
Avaliador: Rubens de Andrade Fernandes, Me.

Manaus - 2022

AGRADECIMENTOS

A Deus, por sempre me proporcionar saúde e disposição para ir atrás de meus sonhos. Ao meu pai, que apesar de não está presente, sempre foi meu maior apoiador e incentivador, a minha mãe que sempre dedicou muito de sua vida a minha criação e educação. Aos meus amigos Ítalo Santos e Luana Mello pelos conselhos e apoio em momentos difíceis da minha vida. Aos amigos da faculdade Arlley, Matheus Assunção, Wesley Salvador, Leonardo Moraes e Laís Busnello que tornaram de alguma forma o curso mais leve. Agradeço aos meus amigos do Hub – Tecnologia e Inovação que contribuíram para elaboração desse projeto em especial Rubens, Manoel e Lennon. E por fim a todos os professores que de alguma forma acreditaram no meu potencial e auxiliaram-me nessa jornada.

“Seja forte e corajoso! Não se apavore, nem se desanime, pois, o Senhor, o seu Deus, estará com você por onde você andar".

Josué 1:9

RESUMO

Este trabalho apresenta o desenvolvimento de um protótipo de sistema para determinar o ângulo de chegada do sinal *bluetooth* aplicado na localização em ambientes *indoor* utilizando tecnologia *direction finding*. Com o auxílio da funcionalidade *bluetooth direction finding* presente no microcontrolador nRF 52833, os dados do sinal *Buetooth* são enviados para uma aplicação que calcula o ângulo de chegada e determina a direção do sinal *Bluetooth*. Em um primeiro momento será feita uma apresentação dos problemas que motivaram esse trabalho, em seguida uma fundamentação teórica sobre a diferença entre os sistemas de localização *Indoor* e *Outdoor*, o que é um sistema RTLS (Real Time Locating System) e quais as técnicas que são usadas para determinar a localização de um objeto ou pessoa, abordar o tema *Bluetooth Direction Finding* e como ele obtém o ângulo de chegada, assim como informações sobre o nRf52833. Posteriormente são mostradas quais etapas que orientaram o desenvolvimento do projeto, assim como quais componentes de *hardware*, *software* e equipamentos foram envolvidos no processo. Como resultado os ângulos calculados são mostrados pela aplicação desenvolvida em python o que por fim, demonstra que o protótipo não é capaz de fornecer com precisão os ângulos de chegadas, inviabilizando determinar a direção de emissão do sinal *Bluetooth*.

Palavras-chaves: Localização *Indoor, Bluetooth Direction Finding*, nRF52833.

ABSTRACT

This work presents the development of a prototype system to determine the angle of arrival of the bluetooth signal applied to the location in indoor environments using direction finding technology. With the help of the bluetooth direction finding functionality present in the nRF 52833 microcontroller, the data from the Buetooth signal is sent to an application that calculates the arrival angle and determines the direction of the Bluetooth signal. At first, a presentation of the problems that motivated this work will be made, then a theoretical foundation on the difference between Indoor and Outdoor location systems, what is an RTLS system (Real Time Locating System) and which techniques are used. used to determine the location of an object or person, discuss Bluetooth Direction Finding and how it obtains the angle of arrival, as well as information about the nRf52833. Subsequently, the steps that guided the development of the project are shown, as well as which hardware, software and equipment components were involved in the process. As a result, the calculated angles are shown by the application developed in python, which finally demonstrates that the prototype is not able to accurately provide the arrival angles, making it impossible to determine the emission direction of the Bluetooth signal.

Keywords: Indoor localization, Bluetooth Direction Finding, nRF52833.

SUMÁRIO

INTRODUÇÃO

O *Global Position System* (GPS) é o principal sistema de localização utilizado nos mais diversos dispositivos móveis como *tablets*, telefones celulares e veículos não tripulados. Essa grande difusão do GPS se dá pelo seu baixo custo e facilidade de uso, além de poder retornar à localização em ambientes externo com precisão na ordem de metros. Porém, o uso do GPS em ambientes internos não é adequado, pois o sinal dos satélites pode ser atenuado ou até extinguido antes de chegar ao receptor (MENDES *et al*, 2020).

A tecnologia *Bluetooth*, com a implementação do *Direction Finding*, na versão 5.1, possibilitou o desenvolvimento de dispositivos capazes de determinar a direção do sinal *Bluetooth*. Neste contexto, sistemas microprocessados como o *Nordic* nRF 52833, que oferece suporte a essa funcionalidade, permite o desenvolvimento aplicações para determinar o ângulo de direção do sinal *Bluetooth* (utilizando *Direction Finding*). A utilização dessa tecnologia permite a localização *indoor* de equipamentos, pessoas e objetos nos mais variados ambientes.

O Sistema de Posicionamento Global (*Global Position System* - GPS), possui maior precisão para determinar posição de equipamentos e pessoas em aplicações externas (*outdoor*). No entanto, a aplicação em ambientes internos (*indoor*) tem pouca efetividade e baixíssima precisão, o que dificulta a localização de ativos em tempo real.

Este trabalho tem como hipótese desenvolvimento de um protótipo de sistema para determinar o ângulo de chegada do sinal *Bluetooth* aplicado na localização em ambientes *indoor* utilizando tecnologia *Direction Finding*. Para realizar tal tarefa, serão desenvolvidos, utilizando o *Nordic* nrf 52833, dois dispositivos para compor o sistema, um receptor que determinará o ângulo de incidência do sinal *Bluetooth* (utilizando tecnologia *Direction Finding*) e o *beacon* que emitirá o sinal *Bluetooth*, assim como a aplicação que calculará o ângulo de chegada. O objetivo deste trabalho é desenvolver um protótipo de sistema que determine o ângulo de chegada utilizando a tecnologia *Bluetooth Direction Finding*.

A justificativa para a implementação do projeto baseia-se na ideia de que o protótipo permitirá o monitoramento de informações de direção do sinal *Bluetooth* para determinar o posicionamento em tempo real de equipamento, objetos e pessoas. Podendo ser aplicado em setores onde há a necessidade de se obter o posicionamento *indoor*, como por exemplo, em indústrias auxiliando na logística interna, hospitais em que se faz necessário a localização de equipamentos e pacientes, no setor comercial para localização de produtos e até mesmo para

determinar a presença de pessoas em ambientes. O projeto usará os conhecimentos adquiridos durante o curso de Engenharia Eletrônica, como das disciplinas: Construção Eletrônica, Microprocessadores e Microcontroladores, Projetos de Sistemas Microprocessados e Propagação e Antenas para Sistemas Embarcados.

Na revisão bibliográfica foram realizadas pesquisas sobre os principais assuntos envolvendo o projeto: a diferença entre os sistemas de localização *Indoor* e *Outdoor*, o que é um sistema RTLS e quais as técnicas que são usadas para se determinar a localização de um objeto ou pessoa, *Bluetooth Direction Finding* e como ele obtém o ângulo de chegada, assim como qual microcontrolador foi usado para desenvolvimento do projeto.

Este trabalho está dividido em quatro capítulos, os quais são: referencial teórico, materiais e métodos, implementação do protótipo e análise dos resultados obtidos.

No referencial teórico foi desenvolvida toda a base teórica dos assuntos relacionados ao projeto e que foram citados anteriormente.

No capítulo referente aos materiais e métodos, primeiro foi descrito como definiu-se a implementação do projeto, dividindo o desenvolvimento em: revisão bibliográfica, desenvolvimento da Matriz de Antenas, integração da Matriz de Antenas com o Nordic DK (*Development Kit*), implementação do *software* embarcado para obtenção de amostras IQ, implementação do *software* embarcado para emissão de sinal *Bluetooth*, desenvolvimento da aplicação de cálculo de ângulo de chegada, e por fim o teste e validação do sistema. Em seguida foram listados os principais componentes de *Hardware*, *Software* e equipamentos envolvidos no processo de desenvolvimento do projeto.

No capítulo de implementação do protótipo, são mostradas as principais etapas para o desenvolvimento do *hardware* e *software* do projeto, além de toda a preparação do ambiente de desenvolvimento. Neste capitulo foi definido a arquitetura do sistema, dividindo-o em três dispositivos de *hardware: o beacon*, receptor e computador. Assim como cada uma dessas entidades deveria funcionar.

No quarto capítulo, foram apresentados os resultados obtidos com a implementação do protótipo. Assim como, é mostrada a análise dos resultados baseadas no referencial teórico e no conhecimento adquirido durante o desenvolvimento do projeto.

Por fim, a conclusão mostra a comparação dos resultados obtidos com a hipótese e sugere novas implementações para obtenção de resultados melhores em trabalhos futuros.

1 REFERENCIAL TEÓRICO

1.1 SISTEMAS DE LOCALIZAÇÃO OU POSICIONAMENTO

Um sistema de localização ou posicionamento pode ser caracterizado com base em seu ambiente de operação. Essa classificação define os sistemas de localização ou posicionamento em dois tipos: Sistemas de localização *indoor* e sistemas de localização *outdoor* (BRÁS, 2009).

Os sistemas de localização *outdoor*, os mais comuns, realizam seu funcionamento ao "ar livre", são exemplos: O GPS (*Global Position System*) e Glonass (*Global Navigation Satellite System).* Esses são os únicos Sistemas que possuem constelação completa, o que possibilita alta precisão em seus dados (JEREZ; ALVES, 2018).

Os sistemas de localização *indoor* têm aplicabilidade em ambientes internos, ou seja, ambientes fechados e que inviabilizem, por algum motivo, o funcionamento de sistemas *outdoor*. Portanto "O posicionamento *indoor* pode ser definido como qualquer sistema que forneça um posicionamento preciso dentro de estruturas fechadas, como um shopping center, hospitais, aeroporto e metrô [...]" (FARID; NORDIN; ISMAIL, 2013, p.2).

1.2 SISTEMA DE LOCALIZAÇÃO EM TEMPO REAL (RTLS)

Segundo Cruz *et al* (2011), podemos definir um RTLS (*Real Time Locating System*) como um sistema capaz de fornecer a localização de ativos (pessoas ou objetos) em tempo real. Esses sistemas utilizam *hardware* e/ou *softwares* que viabilizam a obtenção de dados para determinar a localização.

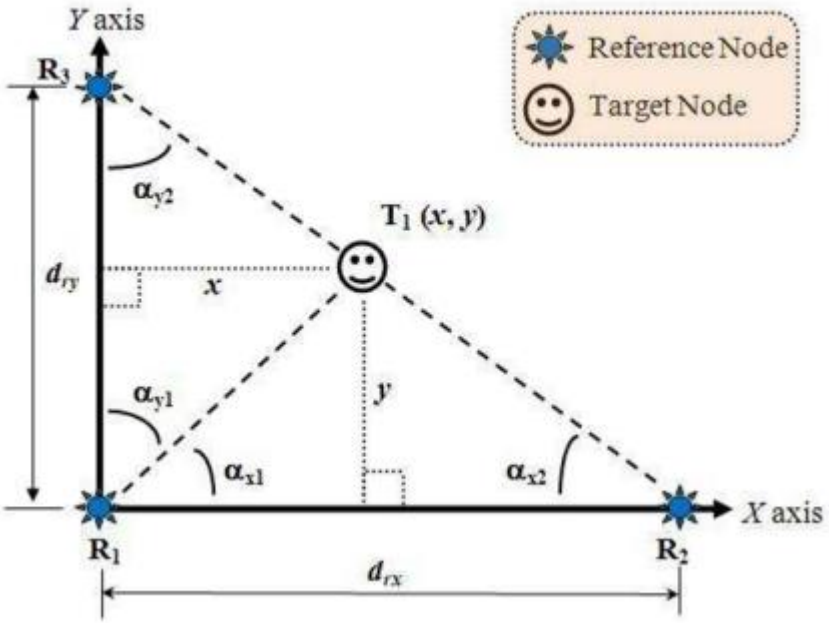
Um RTLS básico consistes em receptores ou leitores fixos que recepcionam sinais sem fios de *beacons* ou etiquetas anexadas em objetos de interesse em que se busca determinar a localização. Geralmente, as informações de posicionamento não contém detalhes de navegação como velocidade ou orientação espacial, se resumindo, muita das vezes, em parâmetros que possibilitem o cálculo de distância e ângulos em *softwares* fora do *hardware* do RTLS (BOULOS; BERRY, 2012).

1.2.1 Técnicas De Localização Implementadas Por Um RTLS.

Podemos citar 2 principais técnicas de Localização que um RTLS pode implementar. São elas: triangulação e trilateração.

A triangulação baseia-se em conceitos matemáticos geométricos dos triângulos. Utiliza distância e/ou diferença angular entre *beacon* e o receptor. Para obter distâncias o sistema pode utilizar técnicas como tempo de chegada (Time of Arriver - ToA) e RSSI (*Received Signal Strengh Indicator*). E para obter a diferença de fase utilizam-se técnicas como ângulo de chegada (*Angle of Arrival - AoA*).

Figura 1 - Triangulação

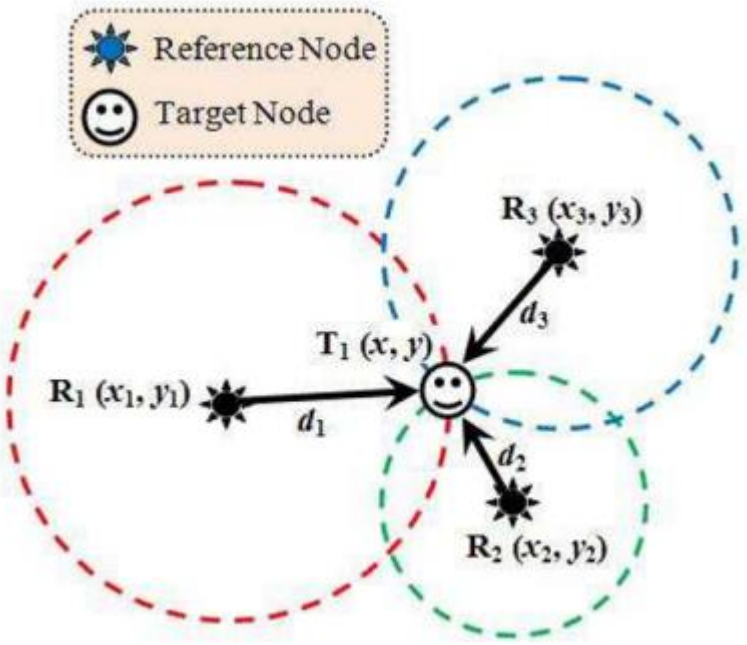


Fonte: (PU *et al*, 2011, p. 235)

O *Targe Node* (Figura 1) emite o sinal, os *References Nodes* recepcionam esse sinal e fazem os cálculos dos ângulos. Se as distâncias forem conhecidas é possível obter a posição do *Target Node* através de cálculos de trigonométricos. (PU *et al.*, 2011).

Segundo PU *et al.* (2011) a **trilateração,** mostrada na figura 2**,** consiste numa estimativa de posicionamento que usa a distância entre os objetos para determinar a coordenada de um alvo. O processo baseia-se em encontrar coordenas do alvo a partir de vários pontos de referências, a distâncias entre o alvo e os pontos de referências são considerados raios de círculos onde o centro são as coordenas dos pontos de referências, a localização do alvo é a interseção dessas circunferências.

Figura 2 - trilateração.



Fonte: (PU, 2011, p. 238).

1.3 BLUETOOTH

De acordo com Drechsler (2018) o *Bluetooth* é um padrão de comunicação sem fio também conhecido como IEEE 802.15.1, que opera na faixa de frequência conhecida como *Industrial, Scientific and Medical* (ISM) de 2,4GHz e possui alcances na ordem de 1 metro, 10 metros até 100 metros (dependendo da potência). O *Bluetooth* apresenta robustez, baixo consumo e custo acessível.

Já Cipriano (2018), define *Bluetooth* como padrão de rede sem fio por meios de onda de rádio que opera na frequência ISM e adota o sistema chamado *Adaptive Frequency Hopping* (AFH) que consiste em realizar saltos regulares de frequências em torno da faixa de trabalho (2.402GHz a 2.480GHz com espaçamento de 1MHz), assim o sistema garante uma redução de interferências na comunicação por meio deste protocolo.

1.3.1 Bluetooth Low Energy – Ble

O *Bluetooth Low Energy* (BLE) é a principal funcionalidade da versão 4.0 do *Bluetooth* e o mesmo permite a operação a longo prazo de dispositivos que transmitem baixos volumes de dados, não ultrapassam taxas de 1 Mb/s (ALECRIM, 2008). O padrão de

consumo de corrente para esses dispositivos tem picos de 15 mA, mas com uma média de 1 uA apenas (ARAUJO; VASCONCELLOS, 2012).

Neste contexto Cipriano (2018, p. 10) diz: “A inclusão desse padrão viabilizou a criação de pequenos sensores alimentados por pequenas baterias do tipo botão (*coin-cell*) com ciclo de vida de meses ou até anos, como é o caso dos *beacons* [...]”.

1.3.2 Beacons Ble

Figura 3 - Beacon Ble.

Fonte: (Cay *et al*, 2017, p. 4).

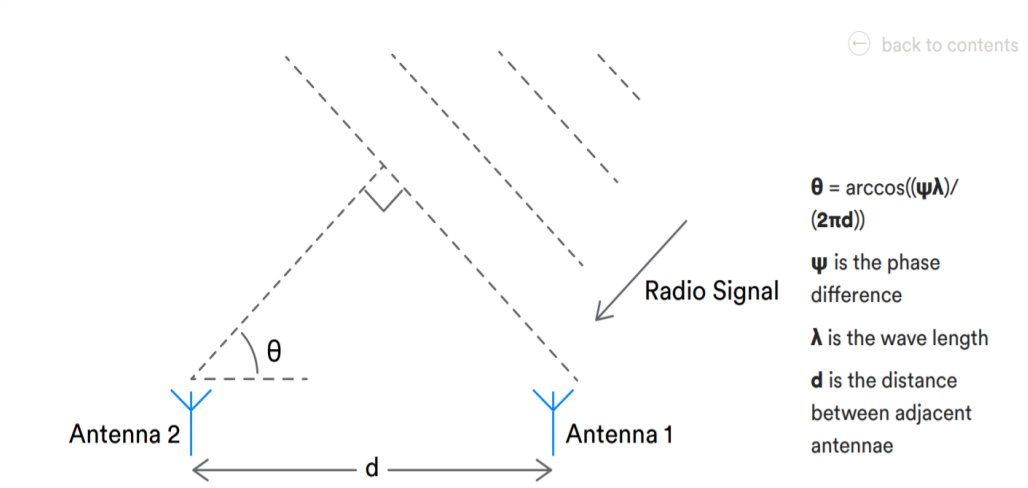
De acordo com Cay *et al*. (2017) um *Beacon BLE* pode ser entendido como um dispositivo constituído por um *chipset*, uma bateria e uma antena, que transmite pacotes *Bluetooth* com certa periodicidade contendo informações que podem ser utilizados pelos receptores, conforme mostrado na figura 3. Segundo Akpinar (2021), *Beacon* é um dispositivo com dimensões pequenas em torno de 3 cm x 5 cm x 2 cm que apenas transmitem dados, eles não recebem e não modificam nada nos receptores.

1.3.3 Bluetooth Direction Finding

De acordo com Woolley (2021) o *Bluetooth Direction Finding* é introduzido pela versão 5.1 do Bluetooth, e o mesmo possibilita encontrar com alta precisão (erros menores que 1 metro) a direção do sinal Bluetooth (o ângulo de direção). Para efetuar essa tarefa os dispositivos que suportam tal tecnologia têm de ser associados a uma matriz de antenas (conjunto de mais de umas antenas conectados entre si). A direção do sinal Bluetooth pode ser

determinada a partir de dois métodos conhecidos como ângulo de recepção (*Angle of Arrive - AoA*) e ângulo de partida (*Angle of Departure - AoD*). Esses ângulos podem ser determinados utilizando paramentos conhecidos como distâncias entre as antenas “d”, diferença de fases nas antenas Ψ (captadas pelo *chipset Bluetooth*) e comprimento de onda do sinal Bluetooth λ. O cálculo do ângulo de chegada é mostrado na figura 4, assim como o sistema trabalha utilizando um *array* de 2 antenas.

Figura 4 - Calculo de θ (AoA) utilizando parâmetros d, Ψ e λ .

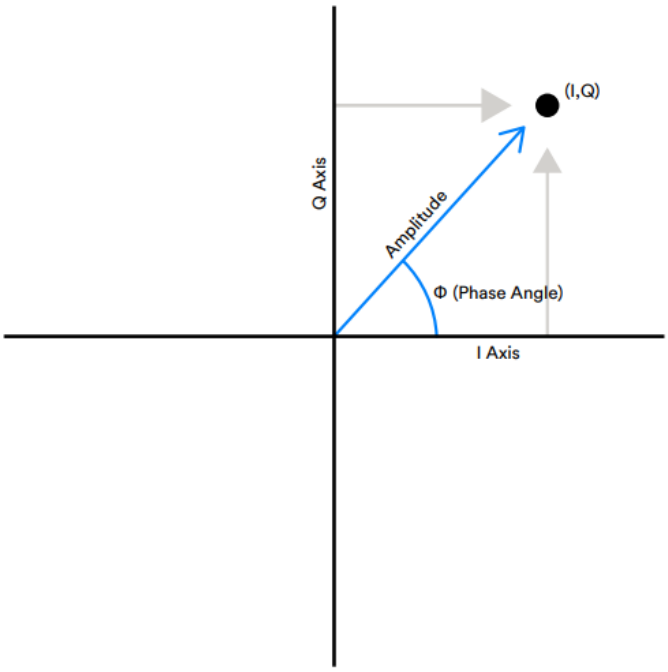


Fonte: (WOOLLEY, 2021, p. 14).

1.3.4 Amostras IQ

Para determinar a diferença de fase entre as antenas o *Bluetooth Direction Finding* faz o uso da técnica da amostragem IQ (*In-phase and Quadrature Sampling*). Uma amostra de IQ consiste na representação cartesiana da amplitude e da fase do sinal Bluetooth. O processo de amostragem é feito em um dispositivo quem contém uma matriz de antenas, onde cada amostra capturada deve ser atribuída a uma antena específica na matriz. Com os devidos cálculos, as amostras IQ cartesianas podem ser convertidas em coordenadas polares onde termos uma amplitude e um ângulo de fase, essa conversão pode ser ilustrada pela figura 5 (WOOLLEY, 2021).

Figura 5 - Angulo de fase e amplitude (I,Q).

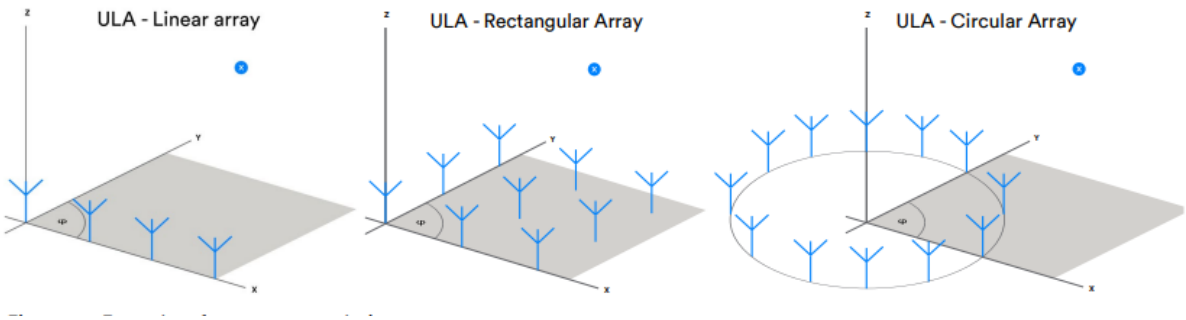


Fonte: (WOOLLEY, 2021, p. 15).

1.3.5 Matriz de Antenas e Switch RF.

Segundo Bevelacqua (2016), uma matriz de antenas pode ser definida como a construção de 2 ou mais antenas que atuam para combinar sinais ou processar (recepção), para se obter um melhor desempenho em relação a uma única antena. O uso dessa técnica pode ser útil em aplicações que: deseja-se aumentar o ganho geral; cancelar interferência de um determinado conjunto de direções; e também deseja-se determinar a direção de chegada de sinais de entrada.

Figura 6 - exemplos de Matrizes de Antenas.

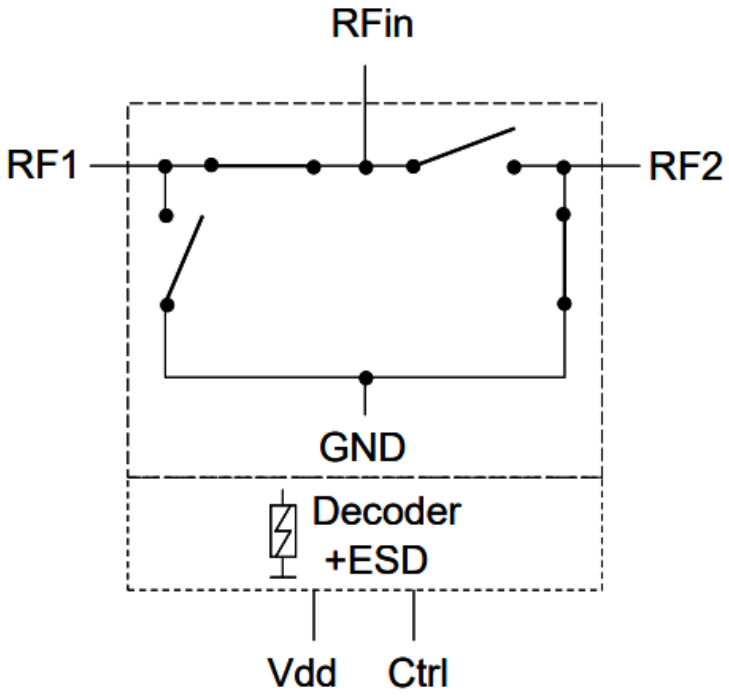


Fonte: (WOOLLEY, 2021, p. 15).

Uma matriz de antenas simples em formato linear permite que um ângulo seja calculado a partir de amostras IQ do sinal. Antenas mais complexas, permitem o cálculo de mais ângulos, nessas aplicações se faz necessário saber, por exemplo, a elevação e o azimute do sinal relativo ao plano de referência, exemplos de matrizes de antenas podem ser observadas na figura 6 (WOOLLEY, 2021).

O principal componente em uma matriz de antenas é o *switch RF*, ele é responsável por selecionar qual antena deve atuar na matriz. É importante levar em consideração alguns parâmetros para que se tenha um bom funcionamento do componente. A característica principal é faixa de frequência em que ele é destinado a atuar. O BGS12PL6E6327XTSA1 é um *switch RF* que atua entre 30 MHz e 4GHz, que é feito com tecnologia CMOS.

Figura 7 - Diagrama de Blocos do Switch RF.



Fonte: (INFINEON, 2015, p. 8).

Para controlar o *Switch RF* é necessário emitir níveis lógicos de tensão 0 e 1 na entrada controle (pino CTRL), se o circuito integrado (CI) estiver alimentado, o sinal de RF é chaveado para uma das entradas do CI, o diagrama de blocos da figura 7 mostra os principais pinos de controle do CI. Um resumo do funcionamento do CI é mostrado na tabela 1.

Tabela 1 - Resumo funcionamento RF.

Conexão do Switch	Nível Lógico
RFin – RF1	0
RFin – RF2	1

Fonte: Autoria Própria.

1.4 MICROCONTROLADOR

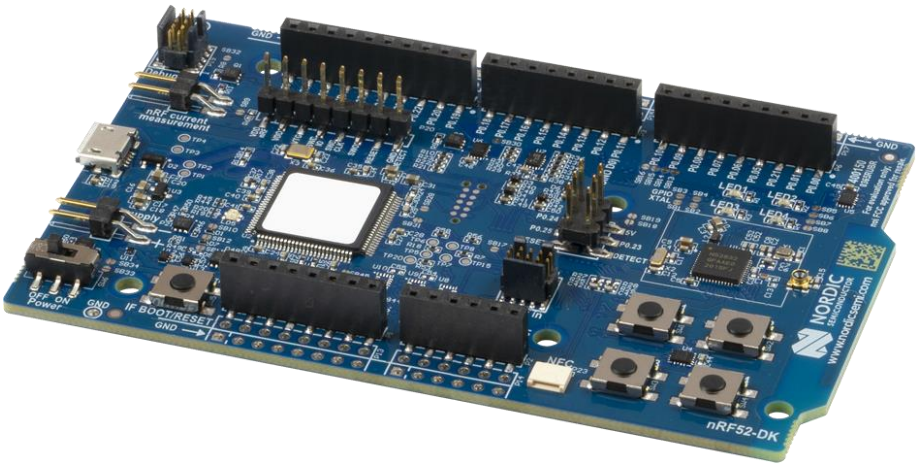
Para Penildo e Trindade (2013) um microcontrolador é um computador em um único chip que contém de modo geral um processador, memorias, periféricos de entrada e saída, temporizadores, dispositivos de comunicação serial entre outras funcionalidades.

Já Cardoso (2020) afirma que o microcontrolador consiste em um único circuito integrado que reúne um processador, memorias dos tipos voláteis e não voláteis assim como diversos periférico de entrada e saída sendo capaz de realizar tarefas dedicadas de modo eficaz sob um tamanho compacto.

1.4.1 Nordic Nrf 52833

De acordo com a fabricante *Nordic Semiconductor* o nRF52833 é um sistema em um único *chip (system on chip - SoC)* multiprotocolo, que possui compatibilidade com *Bluetotth Low Energy* e *Direction Finding.* Ele possui um processador ARM Cortex – M4 de 64MHz associado a 128Kb de memória RAM e 512KB de memória flash para salvar os programas. Além da característica de multiprotocolo, essenciais para projetos de sistemas sem fios, o nRF52833 possui interfaces padrões para comunicação com outros dispositivos como UART, SPI, TWI, PDM, HS-SPI e I2S.

Figura 8 - Kit de Desenvolvimento da Nordic NRF 52833.



(NORDIC SEMICONDUCTOR, 2021, p. 1).

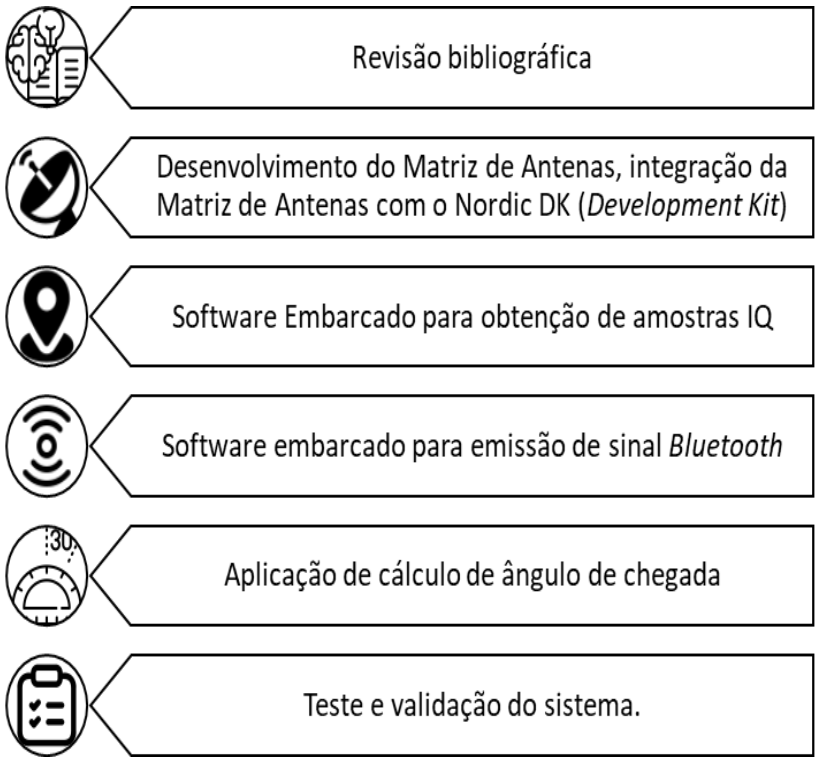
A fabricante do nRF52833 também disponibiliza o *kit* de desenvolvimento, mostrado na figura 8, baseado no mesmo SoC. O nRF52833 DK é um *kit* de desenvolvimento de placa única acessível para o desenvolvimento de aplicações *Bluetooth Low Energy*, *Bluetooth mesh*, NFC, *Thread* e *Zigbee* no SoC nRF52833 (NORDIC SEMICONDUCTOR, 2021).

2 MATERIAS E MÉTODOS.

2.1 MÉTODO PROPOSTO.

A pesquisa foi dividida nas seguintes etapas: Revisão bibliográfica, desenvolvimento da Matriz de Antenas, integração da Matriz de Antenas com o Nordic DK (*Development Kit*), implementação do *software* embarcado para obtenção de amostras IQ, implementação do *software* embarcado para emissão de sinal *Bluetooth*, desenvolvimento da aplicação de cálculo de ângulo de chegada, e por fim o teste e validação do sistema.

Figura 9 - Método Proposto.



Fonte: Autoria Própria.

Na revisão bibliográfica foram realizadas pesquisas sobre os principais assuntos envolvendo o projeto. Em um primeiro momento foi mostrado a principal diferença entre os sistemas de localização *Indoor* e *Outdoor*, com essa diferença bem definida foi mostrado o que é um sistema RTLS e quais as técnicas que são usadas para se determinar a localização de um objeto ou pessoa. Esses conceitos foram importantes para definir o *Bluetooth Direction Finding*, como técnica para obtenção da direção do sinal *Bluetooth*. Essa técnica, possibilita determinar o angulo de chegada (AoA). Para se utilizar o *Bluetooth Direction Finding* é

necessário um *hardware* que implemente essa funcionalidade, a partir disso definimos o Nrf 52833 como núcleo do projeto.

Com o núcleo do projeto definido e qual técnica seria usada, buscou-se implementar uma Matriz de Antenas, pois o *Bluetooth Direction Finding* obtém o ângulo a partir da diferença de fase entre as antenas da matriz. Para isso, foi elaborado o esquemático da matriz de antenas levando em consideração o *switch RF*. Após finalizar o esquemático, foi desenvolvido o *layout* da placa de circuito impresso. O *layout* possui duas antenas que foram utilizadas para recepção do sinal *Bluetooth*. A ferramenta usada nessa etapa foi o EasyEDA. Para a criação física da placa de circuito impresso, foi utilizada uma máquina CNC modelo LPKF ProtoMat M60 que fresa placas de fenolite conforme *layout* enviado para seu *software*. Por fim, foram soldados os componentes na matriz de antenas (SMD e PTH), assim como, realizados teste de manufatura para se detectar curtos e falhas nas trilhas.

Após a matriz ser finalizada, foi realizada a integração da mesma com o Nrf 5283 Dk. Para isso a matriz (possui formato de um *shield*) foi sobreposta e conectada ao DK. Após isso, foram executadas medições nos pinos de alimentação (GND e +3.3V), para verificar se o sistema estava alimentado como um todo.

A etapa de implementação do *software* embarcado para obtenção de amostras IQ é essencial para o desenvolvimento do protótipo, pois é a partir das amostras IQ que são feitos os cálculos de amplitude e fase do sinal Bluetooth em cada antena da matriz. Para a execução dessa etapa realizou-se a preparação do ambiente de desenvolvimento, onde executou-se a instalação de pacotes de software necessários para realizar a programação em C do nrf 52883 DK. Os pacotes instalados fazem parte de um RTOS (*Real time Operation System* – Sistema operacional de tempo real) chamado *zephyr projetc*. É nele que é implementada toda a *stack* (suporte de *software*) necessário para o funcionamento do *Bluetooth Direcion Finding*. Para instalação e programação do *kit* foi utilizado sistema operacional Linux, com Visual Studio Code como IDE (*Integrated Development Environment*).

Assim como na etapa anterior, para se implementar o *software* embarcado para emissão de sinal *Bluetooth* é necessário uso do *zephyr projetc.* Para essa etapa foi realizada a programação do nrf52833 com *software* disponibilizado pela própria *zephyr projetc* em seu repositório online. O objetivo dessa etapa é criar um *beacon* que emite de forma constante o sinal *Bluetooth* com todas as informações que o receptor deve adquirir e amostrar o sinal IQ.

Com o receptor e o *beacon* programados, o sistema está apto a enviar os dados das amostras IQ para uma aplicação que realize os cálculos de ângulos de chegada. Realizou-se o desenvolvimento de um *script python* que obtenha esses dados de amostra IQ e faça o cálculo

da diferença de fase e posteriormente do ângulo de chegada. A comunicação entre a aplicação e o receptor é feita traves de protocolo de comunicação serial UART.

Por fim, na última etapa foram realizados teste onde foram comparados os ângulos mostrados na aplicação e medidos com transferidores. Buscou-se variar a posição do *beacon* em relação ao receptor e comparando os dados mostrados na aplicação.

2.2 MATERIAS UTLIZADOS.

Para o desenvolvimento do projeto foram utilizados os seguintes materiais:

a) *Hardware* (Todos os componentes que fazem parte do protótipo final)
   - 2 - Nrf 52833 DK;
   - 2 - Cabos mini USB;
   - 2 - Baterias Coin – Cell CR2032;
   - 1 - Cabo Pigtail U.F;
   - 1 – Resistor de 10Kohms;
   - 1 – Capacitor de 0.1uF;
   - 1 - BGS12PL6E6327XTSA1(*Switch RF*);
   - 2 – Barras de conector header macho;
   - 1 – Placa de fenolite.

b) *Software* (Todos os componentes que auxiliaram de alguma forma o desenvolvimento do *software* embarcado, assim como nos testes e validações).

   - Computador com Sistema Linux;
   - Interpretador Python 3.8;
   - Compilador Cmake;
   - Compilador GCC;
   - Git;
   - SDK – zephyr OS;
   - Módulo python pyserial;
   - Módulo python pandas;
   - Módulo python Numpy;

- EasyEDA versão 6.4.25.
- Visual Studio Code;
- Google Colab;

c) Equipamentos (Todos os equipamentos que auxiliaram na montagem, teste e validações dos componentes e protótipo do projeto).

- Multímetro;
- Estação de solda Hikari;
- CNC modelo LPKF ProtoMat M60;
- Osciloscópio;
- Fonte DC 5V/2A;
- Transferidor de Ângulos.

3 IMPLEMENTAÇÃO DO PROTÓTIPO.

Este capítulo apresenta como foi desenvolvido o projeto como um todo. Como o ambiente de desenvolvimento foi preparado, e também, o desenvolvimento da matriz de antenas. Outro aspecto que será abordado neste capítulo é como o *Hardware* do protótipo se comunica com a aplicação e a mesma realiza o cálculo do ângulo de chegada.

São apresentados neste capítulo os seguintes tópicos:

a) Preparação do Ambiente de Desenvolvimento;

b) Desenvolvimento do Protótipo do Sistema;

c) Desenvolvimento da Aplicação para cálculo do ângulo de chegada.

3.1 PREPARAÇÃO DO AMBIENTE DE DESENVOLVIMENTO

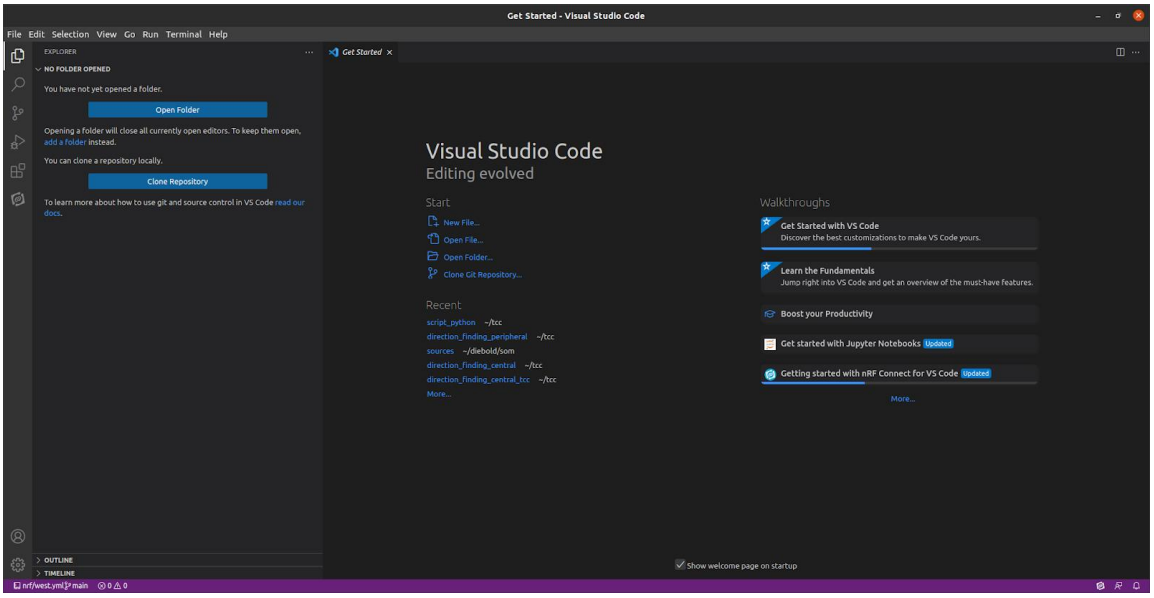
É importante salientar que o sistema operacional em que o *software* embarcado foi desenvolvido é o sistema Linux Ubuntu 20.04. A escolha desse sistema se dá pelo fato de que boa parte das ferramentas de *software* utilizadas para realizar a compilação de códigos é de origem nativa do sistema Linux, a figura 10 mostra as características do sistema que foi desenvolvido o *software* do sistema.

Figura 10 - Características do sistema operacional onde foi desenvolvido o software embarcado do protótipo.

Fonte: Autoria Própria.

Para a configuração do ambiente de desenvolvimento foi necessário a instalação do *Software* Visual Studio Code, mostrado na figura 11.

Figura 11 - Tela do Software VS Code.



Fonte: Autoria Própria.

Após isso, se fez necessário a instalação de pacotes que são pré-requisitos para a instalação do *zephyr project* e SDK da Nordic. Os pacotes serão apresentados na tabela 2, assim como qual comando para se obter o mesmo:

Tabela 2 - Lista de comandos para instalação de pré-requistos.

Nome do Pacote	Comando para Instalação
Git	sudo apt-get install git
Wget	sudo apt-get install wget
ncurses 5	sudo apt-get install libncurses5
Cmake	mkdir -p ${HOME}/cmake && cd ${HOME}/cmake wget https://github.com/Kitware/CMake/releases/download/v3.20.5/cmake-3.20.5-linux-x86_64.sh yes \| sh cmake-3.20.5-linux-x86_64.sh \| cat echo "export PATH=${PWD}/cmake-3.20.5-linux-x86_64/bin:${PATH}" | tee -a

	${HOME}/.zephyrrc ${HOME}/.bashrc cmake --version
Ninja	sudo apt-get install ninja-build
gperf	sudo apt-get install gperf
Ccache	sudo apt-get install ccache
dfu	sudo apt-get install dfu-util
DTC	[$(apt-cache show device-tree-compiler \| grep '^Version: .*$' | grep -Po '(\d.\d.\d+)' | sed 's/\.//g') -ge '146'] && sudo apt-get install device-tree-compiler || (wget http://mirrors.kernel.org/ubuntu/pool/main/d/device-tree-compiler/device-tree-compiler_1.4.7-1_amd64.deb && sudo dpkg -i device-tree-compiler_1.4.7-1_amd64.deb)
pip3	sudo apt-get install python3-pip
python3 setuptools	sudo apt-get install python3-setuptools
python3 wheel	sudo apt-get install python3-wheel
xz-utils	sudo apt-get install xz-utils
gcc	sudo apt-get install gcc-multilib
make	sudo apt-get install make

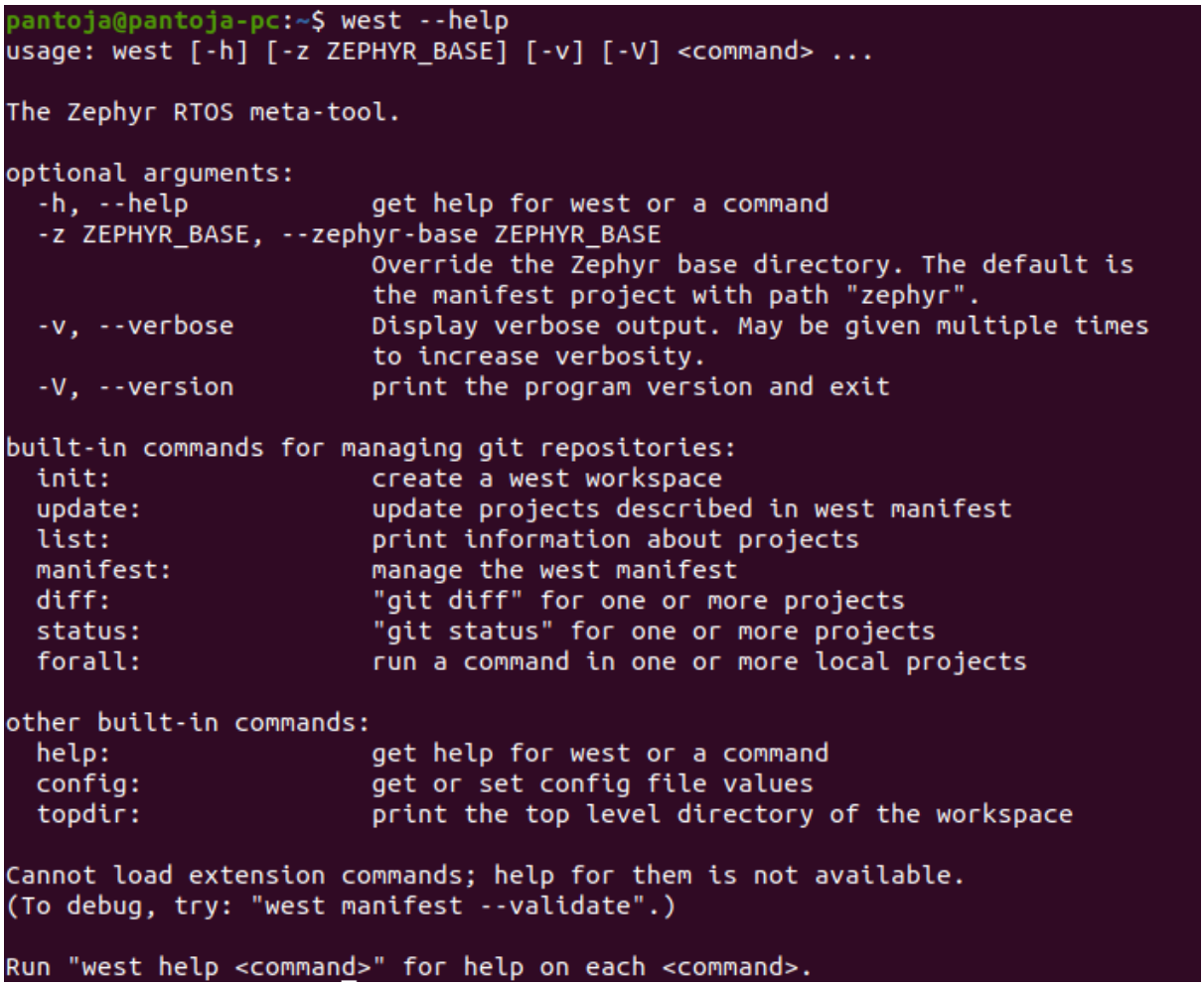
Fonte: Autoria Própria.

O *zephyr projetc* utiliza uma ferramenta chamada *west* para gerenciar as versões e repositórios de onde se obtém o SDK. Para obtê-la utilizamos o comando:

- pip3 install --user West

Com o *west* instalado podemos chamá-lo como um comando shell em um terminal, exemplificado na figura 12. É através dele que iremos manter o controle das versões da base de código do *zephyr projetc*.

Figura 12 - Chamando o comando West no shell Linux.

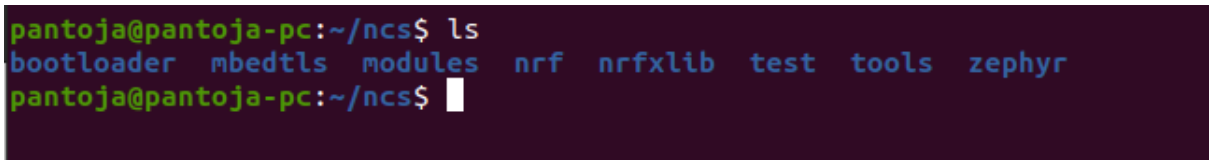


Fonte: Autoria Própria.

Após a instalação do West, foi obtida a base de código do zephyr, para tanto foi criado um diretório chamado “ncs” o qual iniciou-se um repositório west e executado o *download* dos dados, o diretório é mostrado na figura 13. A sequência de comandos que realiza essa etapa:

- mkdir ncs
- cd ncs
- west init -m https://github.com/nrfconnect/sdk-nrf
- west update
- west zephyr-export

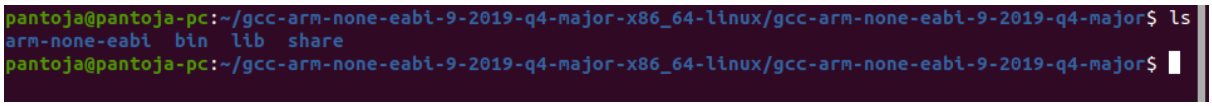
Figura 13 - Diretório ncs após obter base de código do zephyr project.



Fonte: Autoria Própria.

Por fim, o último pré-requisito é a *toolchain* que é usada para cross-compilar objetos para alvos de arquitetura ARM. Foi feito o *download* dela e a mesma foi extraída no diretório home do computador host, conforme mostrado na figura 14.

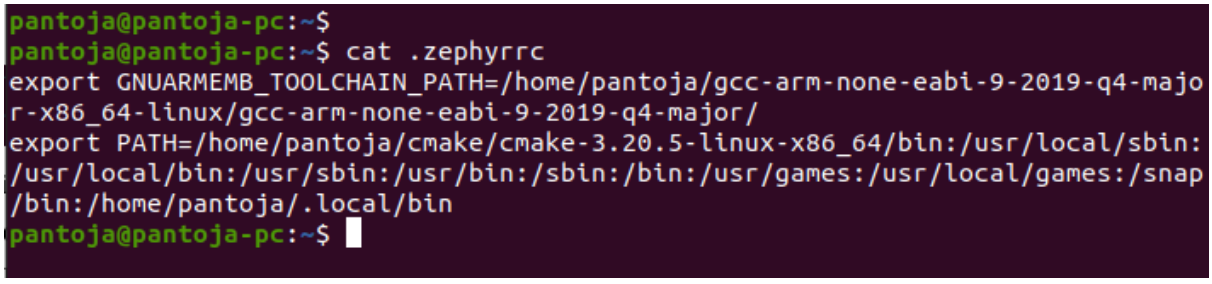
Figura 14 - Diretório da toolchain.



Fonte: Autoria Própria.

Para que a *toolchain* fique disponível para qualquer outro aplicativo do zephyr ou do nordic SDK é necessário adicionar a mesma no path do zephyr. Para isso, basta abrir e editar o arquivo .zephyrrc, que contém as variáveis de ambientes GNUARMEMB_TOOLCHAIN_PATH e PATH, conforme mostrado na figura 15.

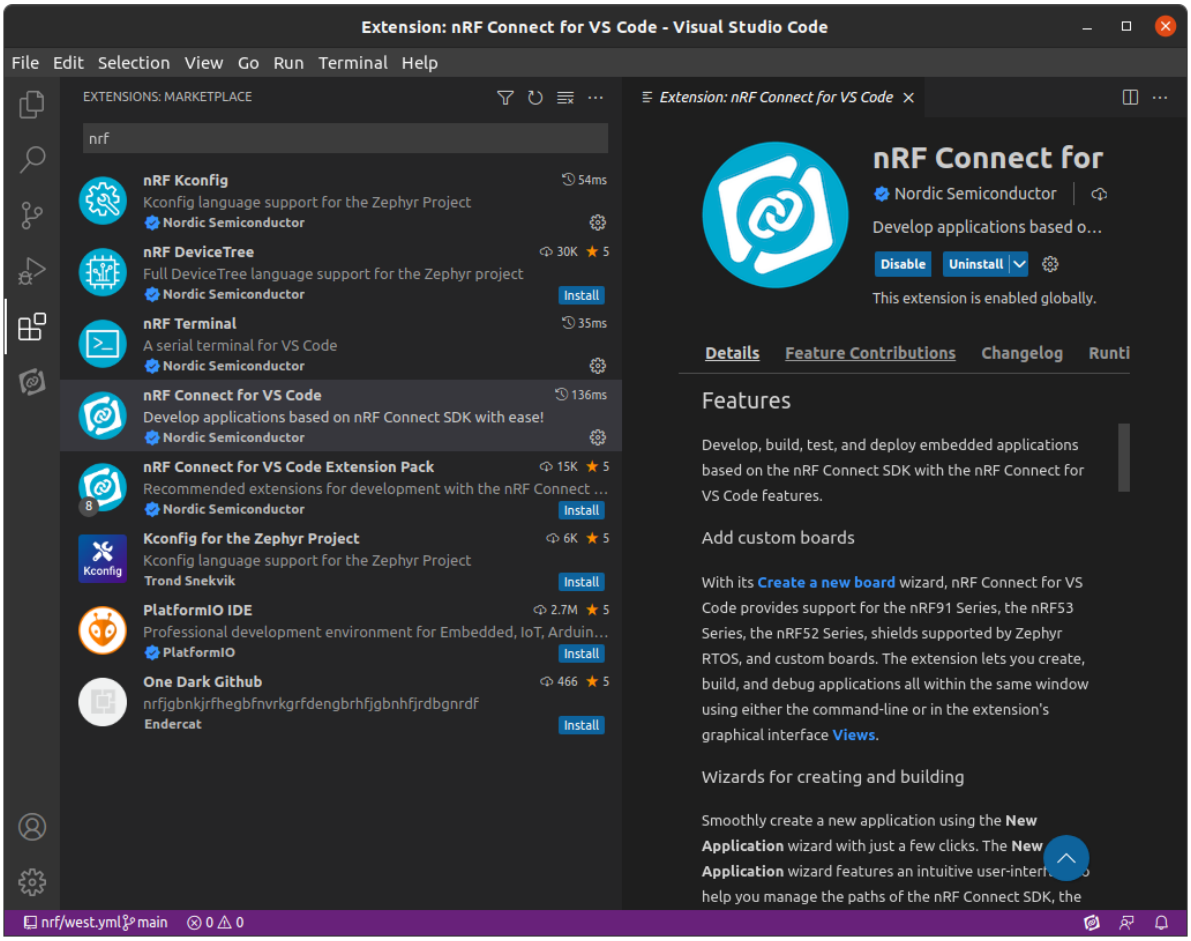
Figura 15 - Configurando as variáveis de ambiente para usar a toolchain.



Fonte: Autoria Própria.

Com todos os pré-requisitos instalados, a última etapa é configurar a extensão nRF Connect for VS Code. Essa extensão possibilita a integração de todas as funcionalidades disponíveis no *zephyr projetc* e Nordic SDK de maneira visual e simples. Para realizar a instalação é necessário buscá-la na ferramenta extensões dentro do VS Code, como é mostrado na figura 16.

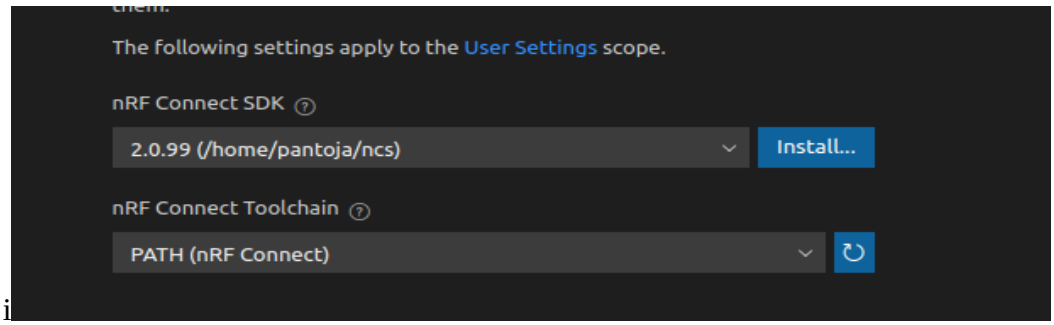
Figura 16 - Extensão nrf Connect for VS Code.



Fonte: Autoria Própria.

Depois de encontrada, a extensão nrf *Connect*, basta clicar em “Install” que a mesma será instalada. Dentro da página inicial da extensão na guia Quick Setup existem dois campos a serem preenchidos: um é o path do SDK (a pasta NCS) e outro é variável de ambiente PATH (contém todos os comandos shell utilizados pelo zephyr). Os campos foram configurados com as devidas configurações, conforme é mostrado na figura 17.

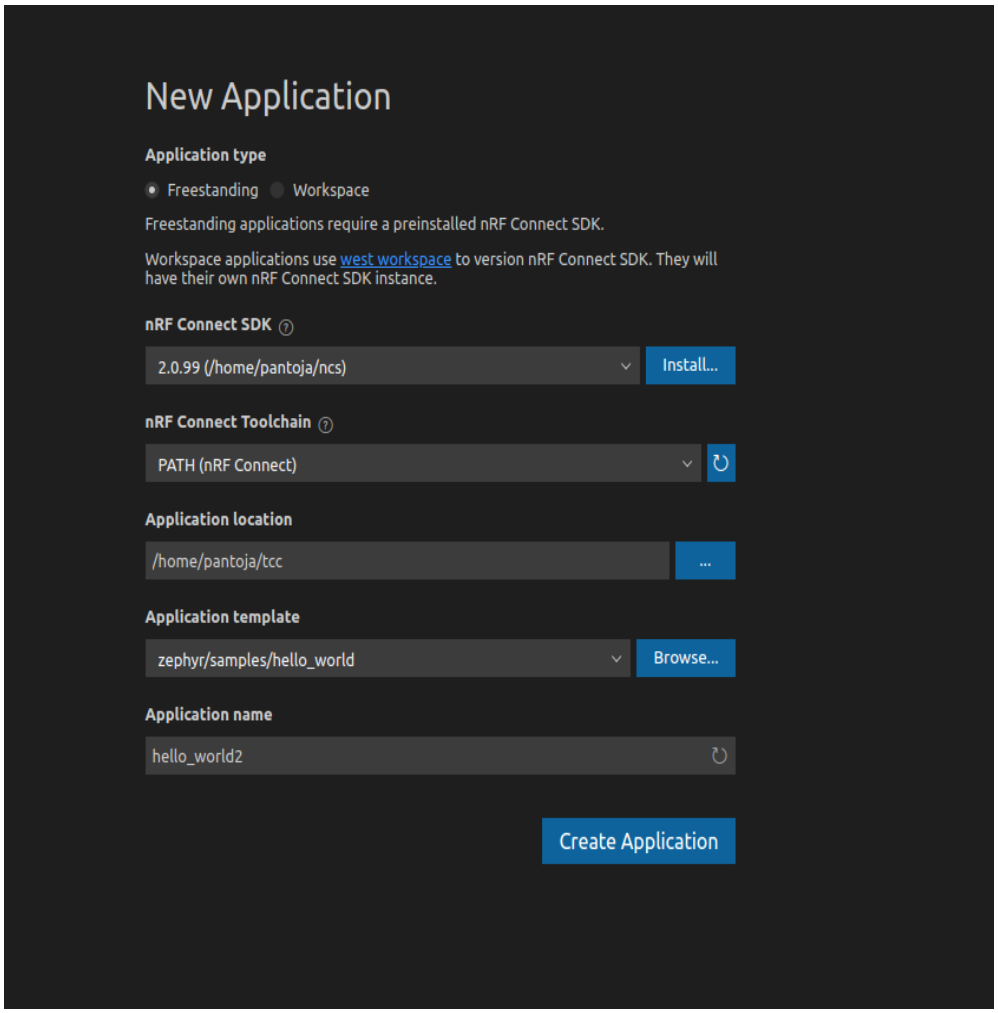
Figura 17 - Indicando a localização do SDK e a toolchain para ferramenta no VS Code.



Fonte: Autoria Própria.

Após todos os passos esse é o ambiente de desenvolvimento que está apto para programar possibilitando escolher exemplos para iniciar a codificação.

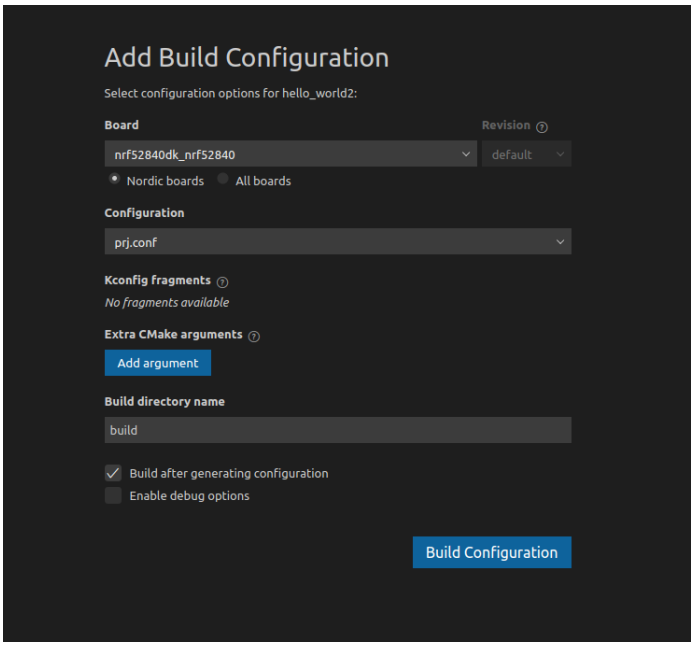
Figura 18 - Criando um novo projeto.



Fonte: Autoria Própria.

Por fim, para validar o processo de preparação do ambiente e atestar que o mesmo está apto para compilar e gravar o *firmware*, devemos criar uma nova aplicação, o processo é mostrado na figura 18. Ao tentar criar é solicitado que seja feito a configuração de build. Nesta etapa não são necessárias modificações, apenas confirmar as configurações e usar o padrão clicando em *Build Configuration*, essa etapa é mostrada na figura 19.

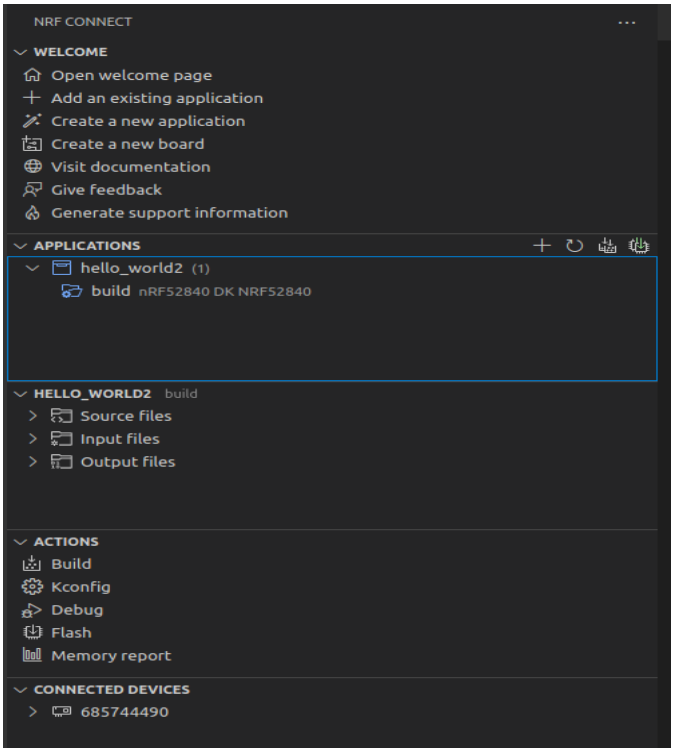
Figura 19 - Confirmando as configurações de build.



Fonte: Autoria Própria.

Ao clicar em *Build Configuration*, o processo de build inicia automaticamente, se não houver problemas ele irá mostrar uma tela de comando para o projeto conforme figura 20 abaixo:

Figura 20 - Tela gerada após a primeira compilação.

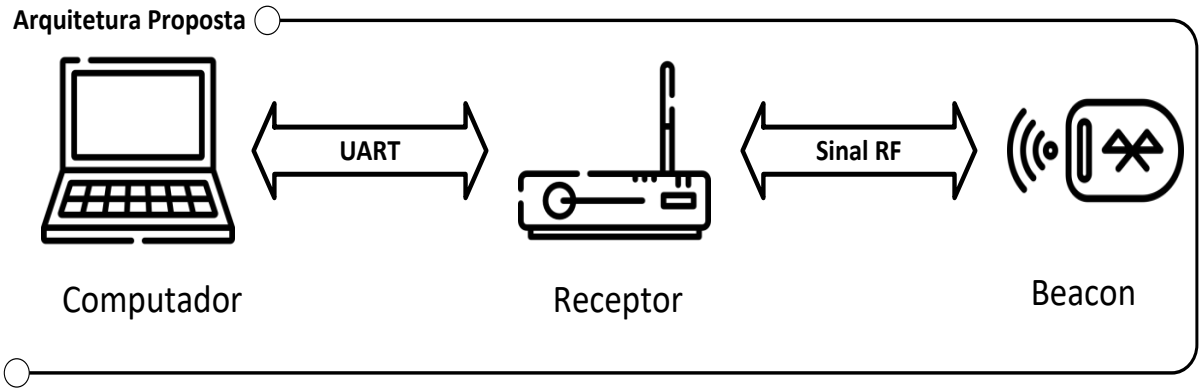


Fonte: Autoria Própria.

3.2 DESENVOLVIMENTO DO PROTÓTIPO DO SISTEMA

Para se desenvolver o Protótipo do sistema foi definida a arquitetura do mesmo, o objetivo dessa arquitetura é orientar o processo de desenvolvimento. O sistema é composto por três dispositivos de *hardware*: o *beacon*, o receptor e o computador (executa a aplicação), a arquitetura é mostrada na figura 21.

Figura 21 - Arquitetura do Projeto.



Fonte: Autoria Própria.

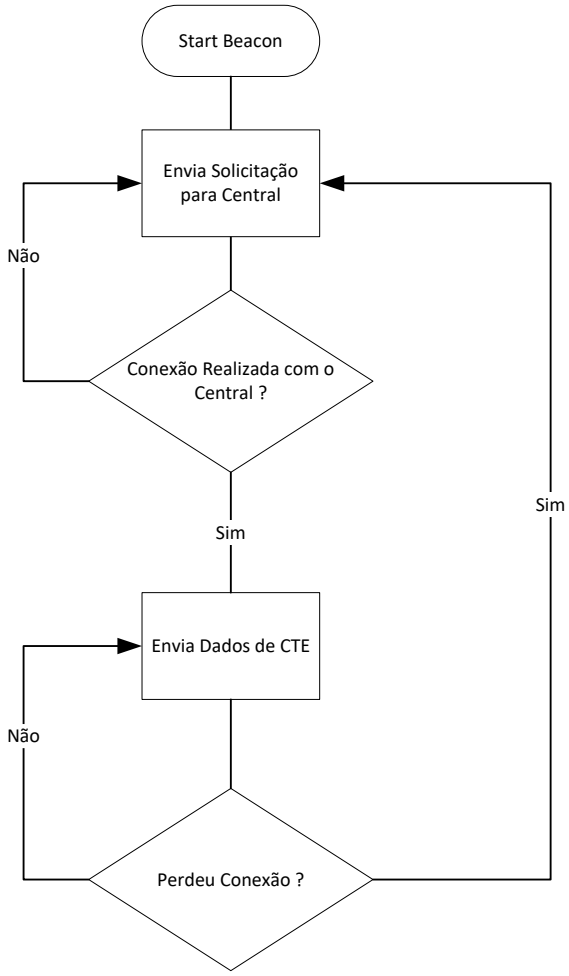
- O *beacon* - envia de forma constante sinal *Bluetooth* contendo informações de IQ, essencial para determinar o ângulo de chegada.
- O receptor - este contém a matriz de antenas, controla o chaveamento entre as mesmas e processa o sinal recebido e obtém as amostras IQ. Além de se comunicar com o computador via comunicação serial UART.
- Computador - conectado com o receptor via UART, recebe o sinal e obtém as amostras IQ, para determinar o ângulo de fase do sinal em cada antena e assim determinar a diferença de fase entre elas para se obter o ângulo de chegada.

O desenvolvimento do *beacon* consistiu em obter um Nrf 52833 DK e programá-lo com o devido *firmware* para gerar o sinal Bluetooth de forma constante. Como DK possui uma antena *on board*, o mesmo pode ser alimentado e gerar o sinal para obtenção das amostras IQ.

Para programar o *beacon* foi utilizado o exemplo disponível na base de código do SDK da nordic. O exemplo que foi utilizado foi o *Direction finding peripheral*. Por padrão as configurações do exemplo funcionam de acordo com a necessidade do projeto: transmitir de

forma constante o sinal Bluetooth e atuar no modo AoA, o funcionamento do *beacon* é mostrado de forma resumida no fluxograma da figura 22.

Figura 22 - Fluxograma de funcionamento do Beacon.



Fonte: Autoria Própria.

Para validar que o *software* foi gravado com sucesso na memória do nrf 52833, quando o mesmo faz uma conexão com o receptor ele retorna na saída serial de *debug* a mensagem de conexão, conforme mostrado na figura 23.

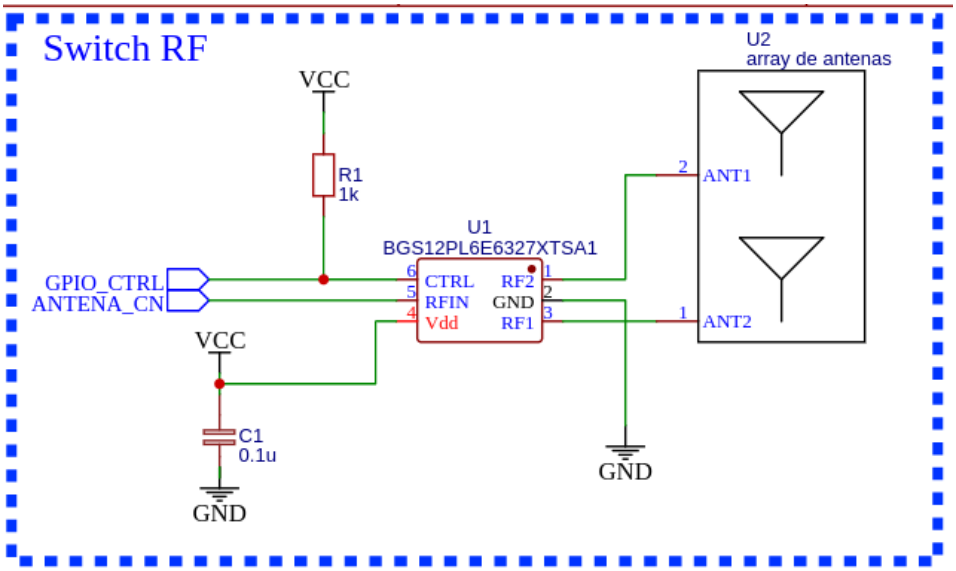
Figura 23 - Saída serial, mostrando que o beacon está emitido o sinal Bluetooth.

```
*** Booting Zephyr OS build v3.0.99-ncs1-4913-gf7b06162027d  ***
Bluetooth initialized
Advertising successfully started
[00:00:00.259,216] <inf> bt_hci_core: HW Platform: Nordic Semiconductor (0x0002)
[00:00:00.259,216] <inf> bt_hci_core: HW Variant: nRF52x (0x0002)
[00:00:00.259,246] <inf> bt_hci_core: Firmware: Standard Bluetooth controller (0x00) Version 3.1 Build 99
[00:00:00.260,101] <inf> bt_hci_core: Identity: CD:CD:49:EE:73:2E (random)
[00:00:00.260,101] <inf> bt_hci_core: HCI: version 5.3 (0x0c) revision 0x0000, manufacturer 0x05f1
[00:00:00.260,131] <inf> bt_hci_core: LMP: version 5.3 (0x0c) subver 0xffff
Connected
Set CTE transmission params...success.
Set CTE response enable...success.
```

Fonte: Autoria Própria.

O desenvolvimento do receptor consistiu nas seguintes etapas: elaboração do esquemático da matriz de antenas, desenvolvimento do *layout*, prototipação e montagem, validação e programação do mesmo. O esquema foi feito utilizando a ferramenta easyEAD.

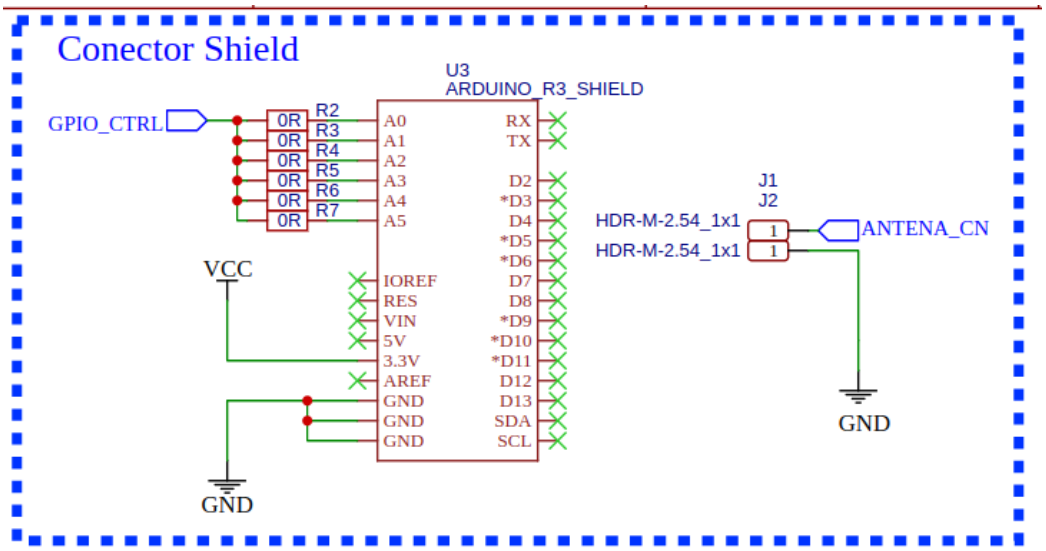
Figura 24 - Circuito de chaveamento das antenas.



Fonte: Autoria Própria.

O esquema foi dividido em duas partes: circuito de funcionamento do chaveamento das antenas, mostrado na figura 24, e conector *shield*, mostrado na figura 25, pois o Kit da nordic tem um fator de forma semelhante à de um Arduino Uno R3, o que facilita a conexão da matriz de antenas sobre o Kit.

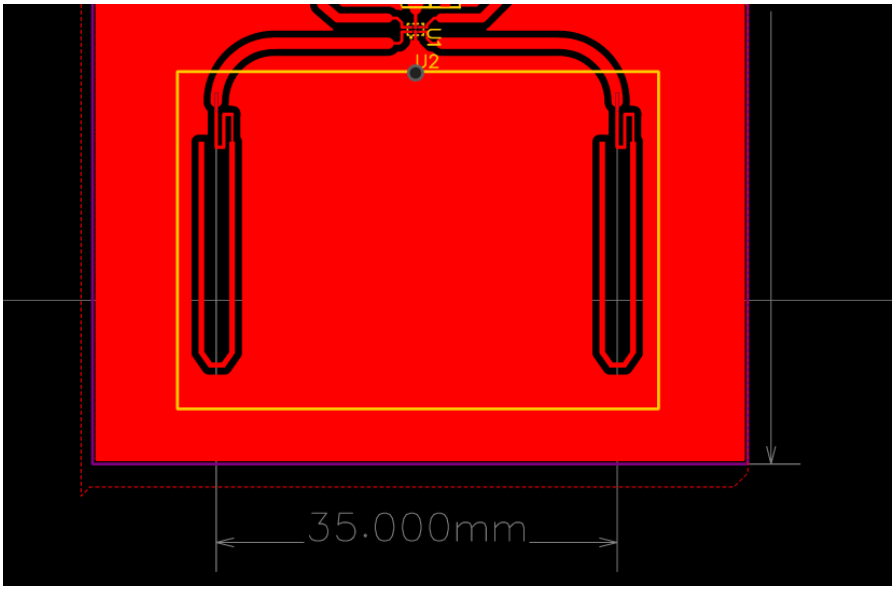
Figura 25 - Circuito conector Shield.



Fonte: Autoria Própria.

Com o esquema eletrônico finalizado foi desenvolvido o *layout* das antenas. Para o desenvolvimento do layout das antenas buscou-se posicionar as antenas a uma distância menor ou igual ½ λ, conforme mostrado na figura 26. Pois quando se é respeitada essa distância a diferença de fase sempre estará entre 0º e 180º (ou - 180º) o que possibilita saber qual antenas está mais próxima do transmissor já que a menor diferença de fase é sempre correta seja ela positiva ou negativa (PORTER; MYHR, 2019).

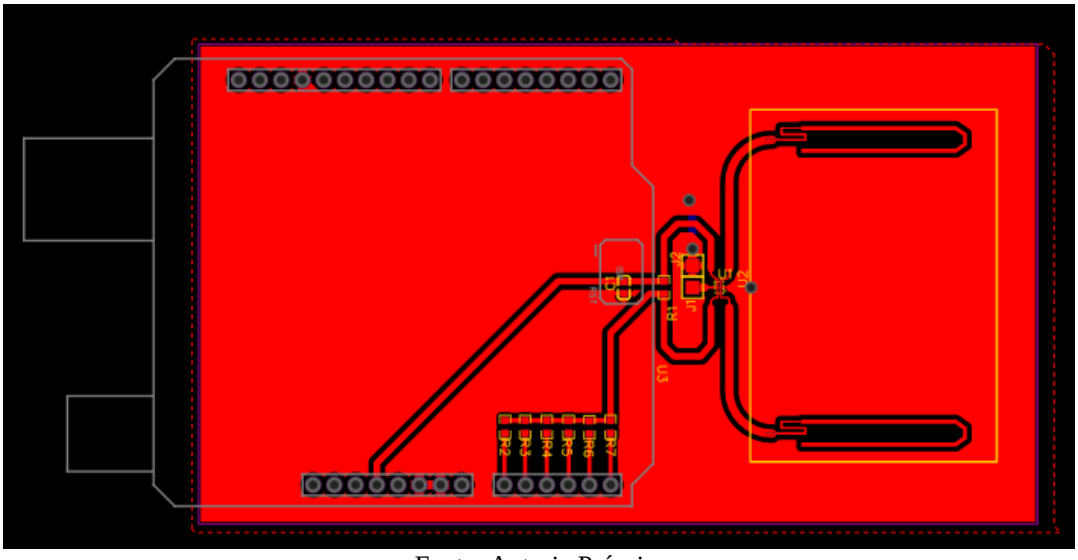
Figura 26 - Distância entre as duas antenas.



Fonte: Autoria Própria.

O formato da matriz de antenas consiste em um shield compatível com Arduino Uno R3, conforme mostrado na figura 27.

Figura 27 - Layout da Matriz de antenas em formato de Shield.



Fonte: Autoria Própria.

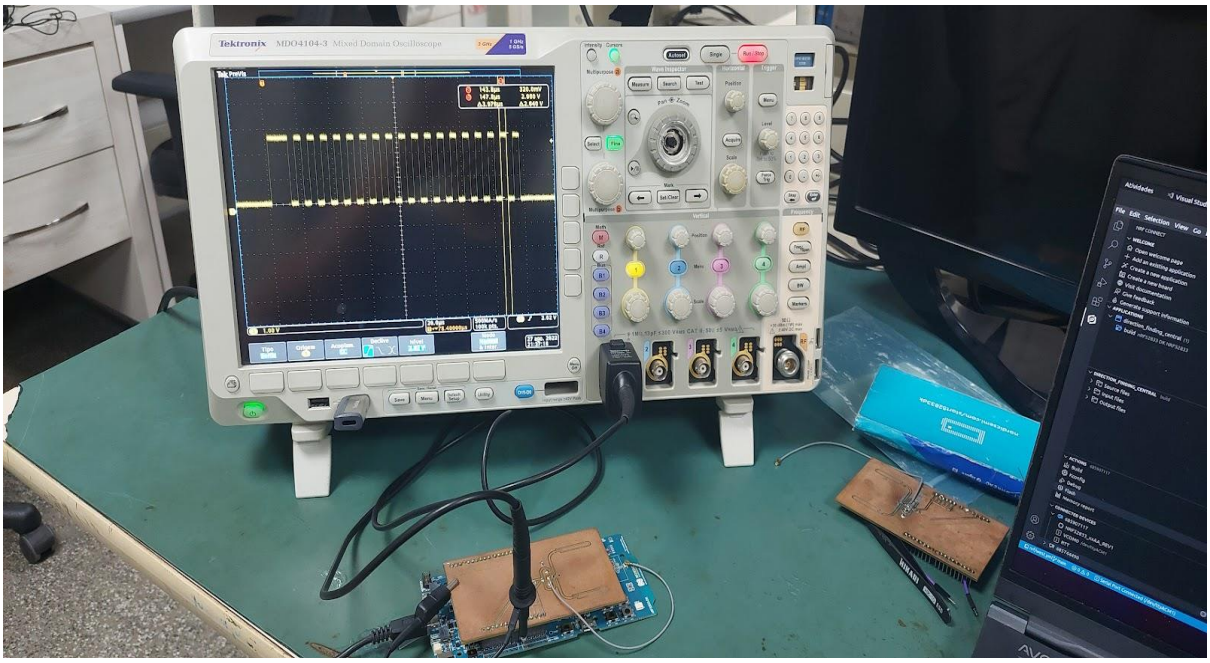
Com o *layout* definido, a próxima etapa foi realizar a prototipação. Essa etapa consistiu em passar o *layout* via gerber para o *software* da máquina CNC e a mesma realiza fresagem de uma placa de fenolite, esse processo é mostrado na figura 28.

Figura 28 - Máquina CNC fresando placa de fenolite para criação do Layout da matriz de Antenas.

Fonte: Autoria Própria.

Depois da prototipação da placa da matriz de antenas, os componentes foram soldados e a mesma foi validada através de medidas utilizando osciloscópio. A matriz foi acoplada no Kit da Nordic, onde foi criado um código que realiza a mudança do nível lógico do pino de controle do Switch RF, o chaveamento dessa etapa de validação foi medido com osciloscópio e mostrado na figura 29.

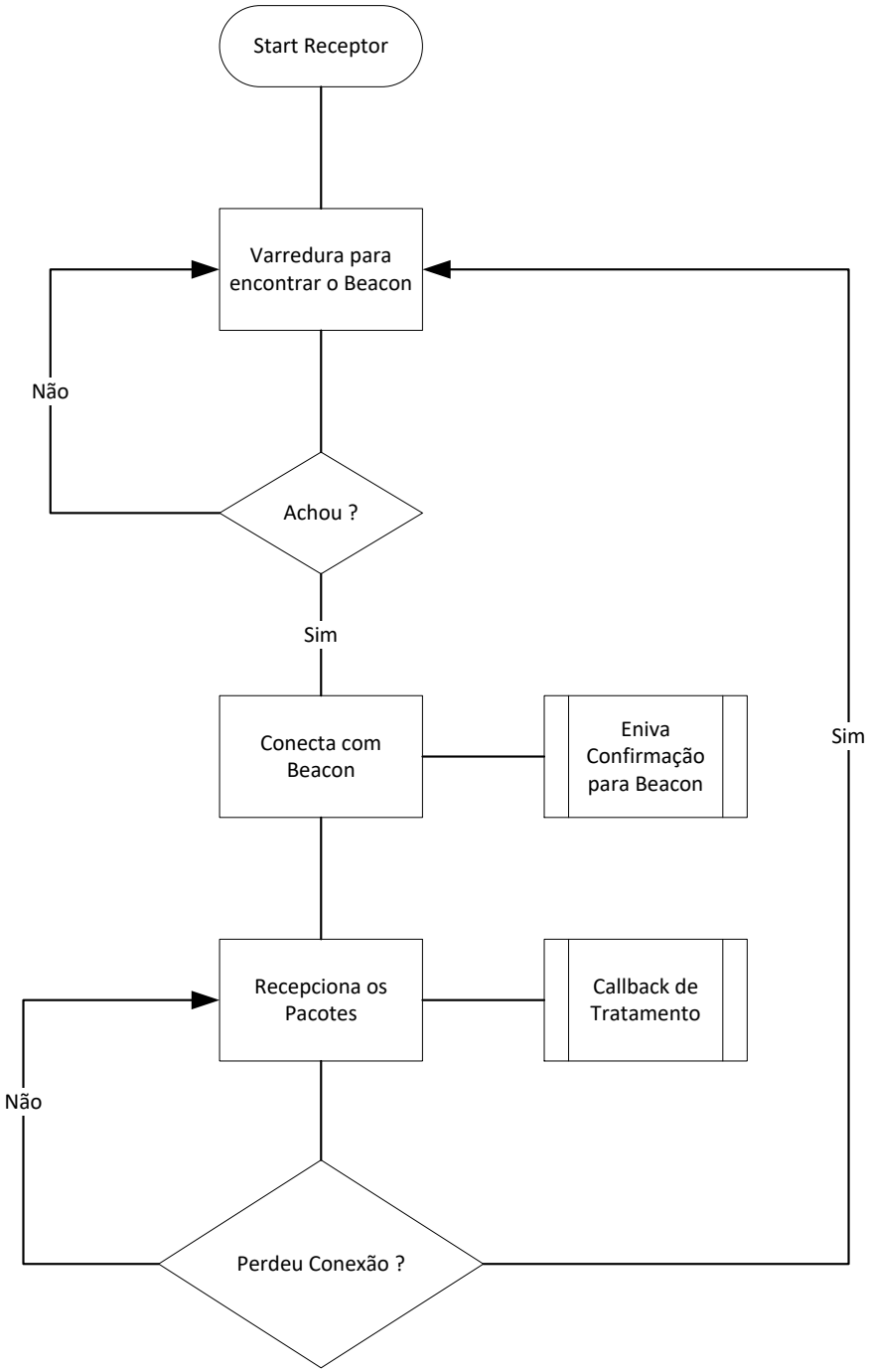
Figura 29 - Etapa de validação com Osciloscópio da Matriz de Antenas.



Fonte: Autoria Própria.

Com o *hardware* do receptor finalizado, a implementação do *firmware* do receptor iniciou-se com base no exemplo *Direction finding central* disponibilizado no SDK da nordic. O *firmware* do receptor realiza varredura para encontrar e realizar uma conexão com um beacon periférico. Após realizar a conexão, ele envia uma confirmação ao *beacon* que inicializa o envio de pacotes contendo informações de sinal IQ, o funcionamento do receptor é mostrado de forma resumida no fluxograma da figura 30.

Figura 30 - Fluxograma do funcionamento do Receptor



Fonte: Autoria Própria.

Para o correto funcionamento do *firmware* com a matriz de antenas desenvolvida neste projeto, é necessário realizar modificações na *stack* de código do mesmo, conforme mostrado na figura 31.

A primeira modificação é realizada no arquivo de overlay, nele é configurado a quantidade de antenas que existem na matriz de antenas (dfe-antenna-num = <2>); e para que

um único pino no Kit seja usado para o chaveamento do Switch RF através do pino CTRL (dfegpio0-gpios = <&gpio0 3 0>).

Figura 31 - Configurando a quantidade de antenas (presente na Matri) e qual pino será usado pelo Kit da Nordic.

```
&radio {
    status = "okay";
    /* This is the number of antennas that are available on the antenna matrix
     * designed by Nordic Semiconductor. For more information see README.rst.
     */
    dfe-antenna-num = <2>;
    /* This is a setting that enables antenna 0 (in antenna matrix designed
     * by Nordic Semiconductor) for Rx PDU. For more information see README.rst.
     */
    dfe-pdu-antenna = <0x0>;

    /* These are GPIO pin numbers that are provided to
     * Radio peripheral. The pins will be acquired by Radio to
     * drive antenna switching when AoA is enabled.
     * Pin numbers are selected to drive switches on antenna matrix
     * desinged by Nordic Semiconductor. For more information see README.rst.
     */
    dfegpio0-gpios = <&gpio0 3 0>;

    /* Os demais pinos são comentados, já que não serão usados nesse projeo */
    /*
    dfegpio1-gpios = <&gpio0 4 0>;
    dfegpio2-gpios = <&gpio0 28 0>;
    dfegpio3-gpios = <&gpio0 29 0>;
    */
};
```

Fonte: Autoria Própria.

A segunda modificação ocorre dentro do arquivo main.c, onde é feita a modificação no padrão de chaveamento das antenas, conforme figura 32.

Figura 32 - Modificação do padrão de antenas no código main.c.

```
// static const uint8_t ant_patterns[] = { 0x2, 0x0, 0x5, 0x6, 0x1, 0x4,
//                  0xC, 0x9, 0xE, 0xD, 0x8, 0xA };
static const uint8_t ant_patterns[] = {0x1, 0x0 };
```

Fonte: Autoria Própria.

Além disso, para se obter as amostras de IQ, foi modificada a função de callback de detecção de um sinal bluetooth. Essa função identifica quantas amostras foram adquiridas. Para cada amostra disponível na memória do receptor, foram obtidas as amostras IQ, o que é mostrado na figura 33. Por fim, essas amostras são enviadas para a serial de Debug do sistema, onde são capturadas pela aplicação de cálculo do ângulo de chegada.

Figura 33 - Função de callback onde se captura e envia para serial os valores de amostra IQ.

```c
static void cte_recv_cb(struct bt_conn *conn, struct bt_df_conn_iq_samples_report const *report)
{
    char addr[BT_ADDR_LE_STR_LEN];

    bt_addr_le_to_str(bt_conn_get_dst(conn), addr, sizeof(addr));

    if (report->err == BT_DF_IQ_REPORT_ERR_SUCCESS) {

        for(uint8_t idx = 0; idx < report->sample_count; idx ++)
        {
            printk("dt0%d,%d,%d,%d,%d\n", idx, report->rssi_ant_id, report->sample[idx].i, report->sample[idx].q, report->rssi);
        }

    } else {
        printk("CTE[%s]: request failed, err %u\n", addr, report->err);
    }
}
```

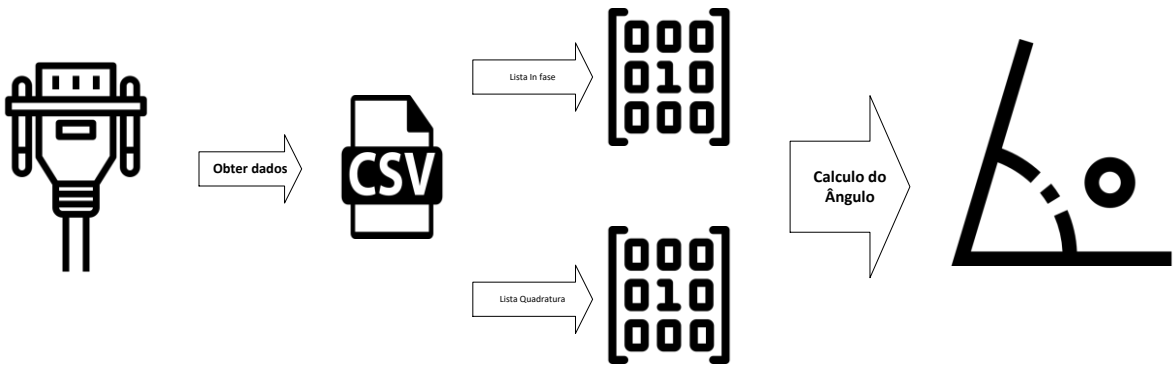
Fonte: Autoria Própria.

O último componente de *hardware* do sistema é o computador, ele executa a aplicação que será abordada com mais detalhes na próxima seção.

3.3 DESENVOLVIMENTO DA APLICAÇÃO PARA CÁLCULO DO ÂNGULO DE CHEGADA.

A aplicação que faz os cálculos do ângulo de chegada (AoA) foi desenvolvida em Python. A mesma possui três etapas: obtenção dos dados, organização dos dados e cálculo do ângulo, a figura 34 mostra de forma resumida essas etapas.

Figura 34 - Esquema de como a função de cálculo do angulo atua.



Fonte: Autoria Própria.

Para obter os dados, o script python faz uso do módulo pyserial, que implementa abstrações de gerenciamento de comunicação UART para facilitar a escrita e/ou leitura de dados nos barramentos físicos da UART. Os dados provenientes do receptor vêm em formato de CSV (Comma-separated values), o script obtém esses dados que são adicionados em uma variável do tipo lista, onde cada posição da lista é referente a um parâmetro (na ordem - index, ant_id, i, q, rssi).

Depois de obtidos, os dados são organizados em duas listas, uma lista armazena dados de In Fase e outra armazena dados de Quadratura. É importante salientar que ao todo são obtidas 45 amostras IQ, no entanto, apenas a partir da 9ª amostra é que se deve levar em consideração como dados reais de IQ, já que as 8 primeiras amostras são usadas como referência pelo hardware do *direction finding*.

Por último, para se obter o ângulo de chegada, é realizado o cálculo do ângulo de cada amostra IQ. Por padrão sabe-se que cada amostra IQ é referente a uma antena na matriz, ou seja, cada ângulo obtido é a fase do sinal bluetooth naquela antena. A diferença de fase é a subtração dos ângulos entre antenas diferentes. A diferença de fase entre antenas é o principal valor para se obter o ângulo de chegada.

Os valores de λ e d são respectivamente 122 mm e 35 mm, e é determinado pelo *script python* a partir das amostras IQ, conforma mostra a figura 35.

Figura 35 - Função de Cálculo do AoA.

```python
def calc_angle(i_list,q_list):
    q_array = np.array(q_list)
    i_array = np.array(i_list)

    comprimento_onda = 122.0
    d = 35.0

    iq_signal = i_array + 1j*q_array

    iq_ang = np.angle(iq_signal)
    # print(iq_ang)

    for i in range(len(iq_ang)):
        if (i%2) == 0:
            if (i + 1 > len(iq_ang)-1):
                pass
            else:
                phase_dif = iq_ang[i+1]-iq_ang[i]
                num = comprimento_onda*phase_dif
                den = 2*np.pi*d
                x = num/den
                print(f"Dif Phase [{np.degrees(phase_dif)}] | X [{x}] | AOA [{np.degrees(np.arccos(x))}]")
```

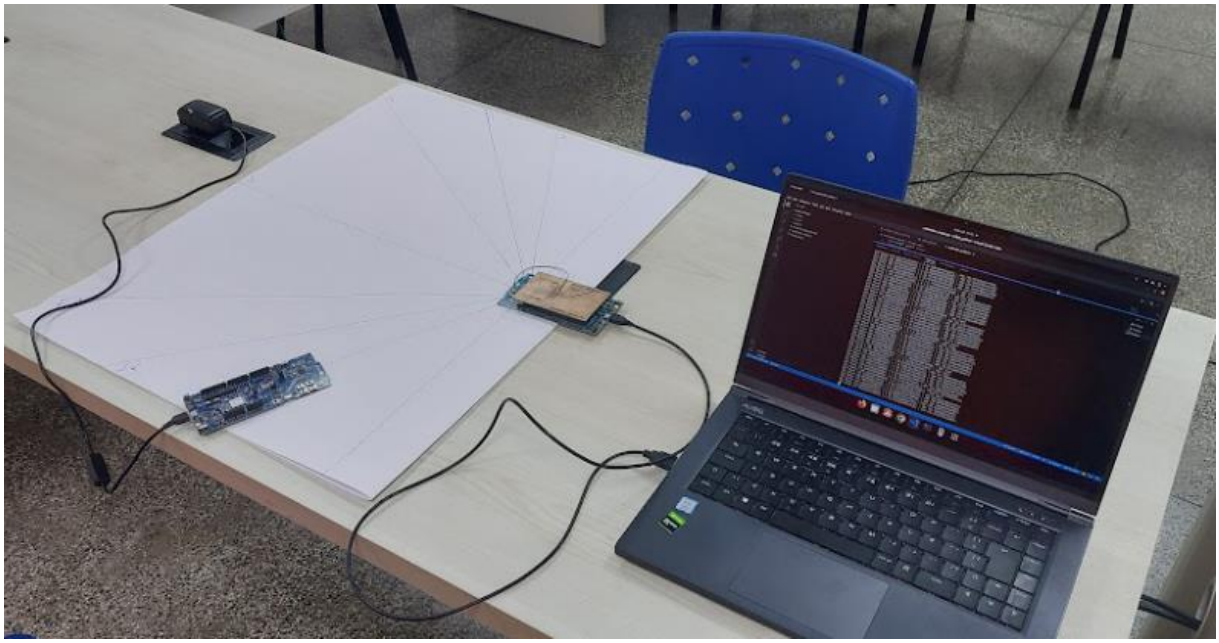
Fonte: Autoria Própria.

4 RESULTADOS OBTIDOS.

Este capítulo demonstra os resultados obtidos após todo o desenvolvimento da pesquisa. Os dados serão apresentados, assim como, o processo de validação e obtenção dos mesmos.

Em um primeiro momento foi necessário criar uma metodologia de validar as medições de AoA. Para isso, foi desenhado, utilizando transferidor de ângulo, linhas de projeção em uma folha de papel de tamanho A1. Essas linhas obedecem às medidas dos ângulos 0º, 30º, 45º, 60º, 90º, 120º, 135º, 150º e 180º. Com as linhas desenhadas, o sistema foi montado e processo de medição e validação dos dados foi executado, conforme mostrado na figura 36.

Figura 36 - Sistema montado sobre as linhas de validação dos ângulos.



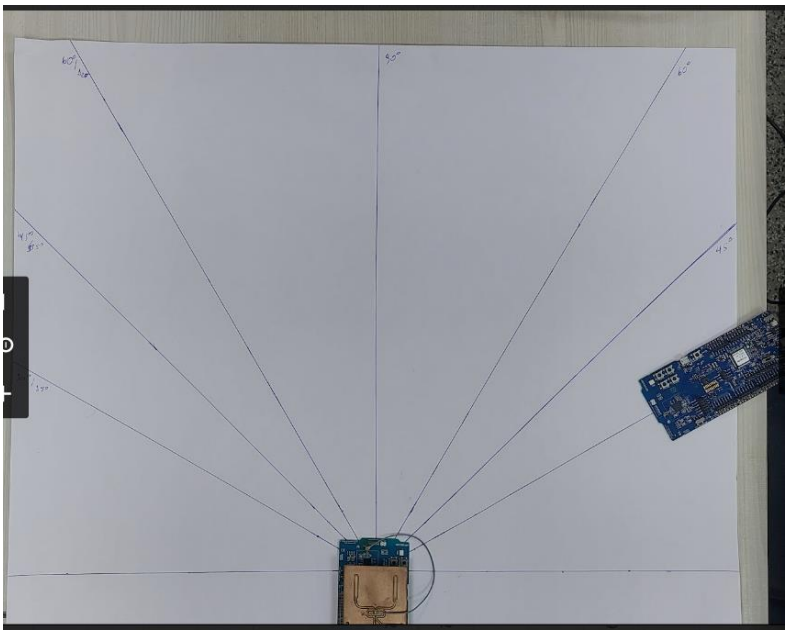
Fonte: Autoria Própria.

O processo de medida dos ângulos, consistiu em posicionar o beacon em cada linha correspondente ao ângulo que se deseja validar. O computador conectado receptor adquire os dados e os mostra no terminal do computador. Os dados foram capturados e documentados em uma planilha e um gráfico foi plotado para mostrar quais valores o AoA calculado assume, para a posição em questão.

4.1 RESULTADO PARA 30º

O primeiro ângulo, a ser validado foi o de 30º, o *beacon* foi posicionado sobre a linha do mesmo, conforme mostrado na figura 37.

Figura 37 - Posicionamento do beacon a 30º do receptor.



Fonte: Autoria Própria.

O gráfico do AoA calculado para 30º, figura 38, demonstra que os valores ficam próximo do valor de 80º (diferença de 50º a mais), assim como pico máximo de aproximadamente 140º e pico mínimo de 20º.

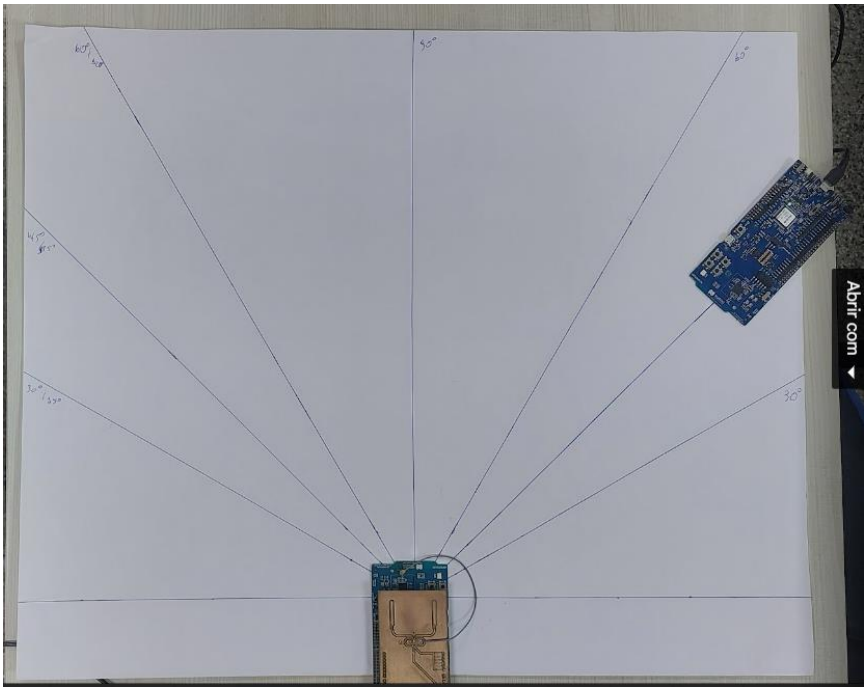
Figura 38 - Gráfico AoA, com beacon posicionado em 30º.

Fonte: Autoria Própria.

4.2 RESULTADO PARA 45º

O próximo ângulo a ser validado foi o de 45º, o *beacon* foi posicionado sobre a linha do mesmo, conforme mostrado na figura 39.

Figura 39 - Posicionamento do beacon a 45º do receptor.



Fonte: Autoria Própria.

O gráfico do AoA calculado para 45º, figura 40, demonstra que os valores variam de forma indefinida.

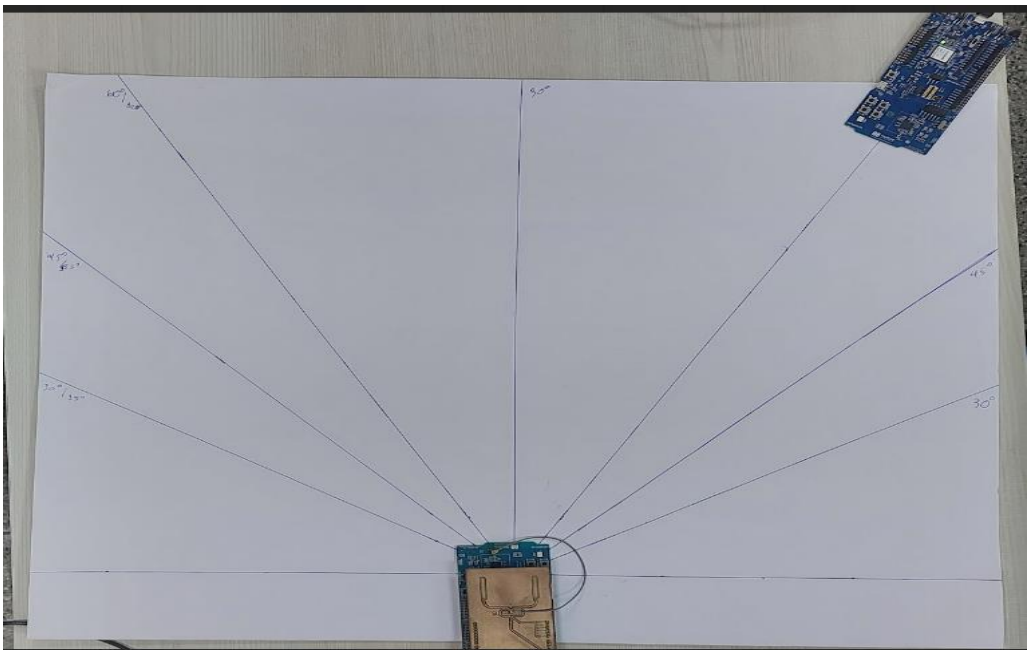
Figura 40 - Gráfico AoA, com beacon posicionado em 45º.

Fonte: Autoria Própria.

4.3 RESULTADO PARA 60º

O ângulo a ser validado nessa seção é de 60º, o *beacon* foi posicionado sobre a linha do mesmo, conforme mostrado na figura 41.

Figura 41 - Posicionamento do beacon a 60º do receptor.



Fonte: Autoria Própria.

O gráfico do AoA calculado para 60º, figura 42, demonstra que os valores variam próximo do ângulo de 60º.

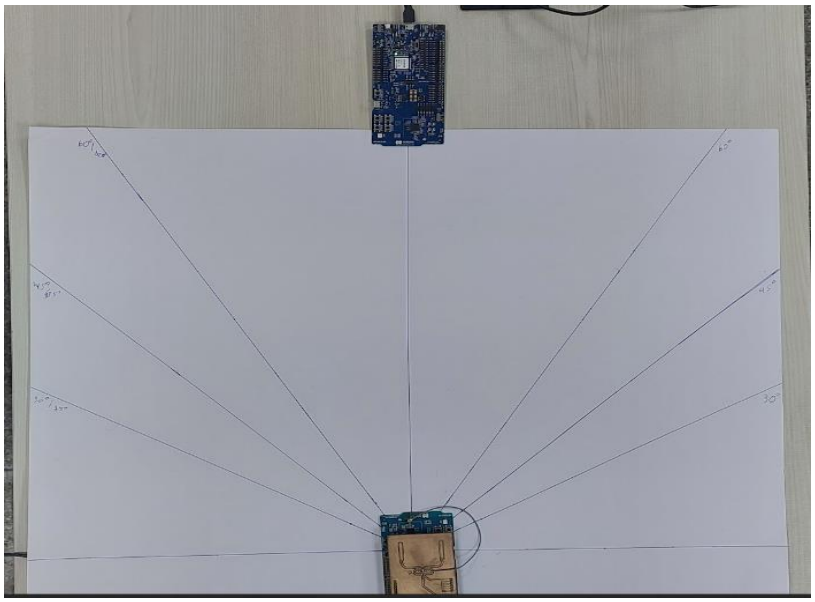
Figura 42 - Gráfico AoA, com beacon posicionado em 60º

Fonte: Autoria Própria.

4.4 RESULTADO PARA 90º

O ângulo a ser validado nessa seção é de 90º, o *beacon* foi posicionado sobre a linha do mesmo, conforme mostrado na figura 43.

Figura 43 - Posicionamento do beacon a 90º do receptor.



Fonte: Autoria Própria.

O gráfico do AoA calculado para 90º, figura 44, demonstra que os valores variam próximo do ângulo de 90º, assim como o ângulo anterior em que a variação ficou próxima do ângulo avaliado.

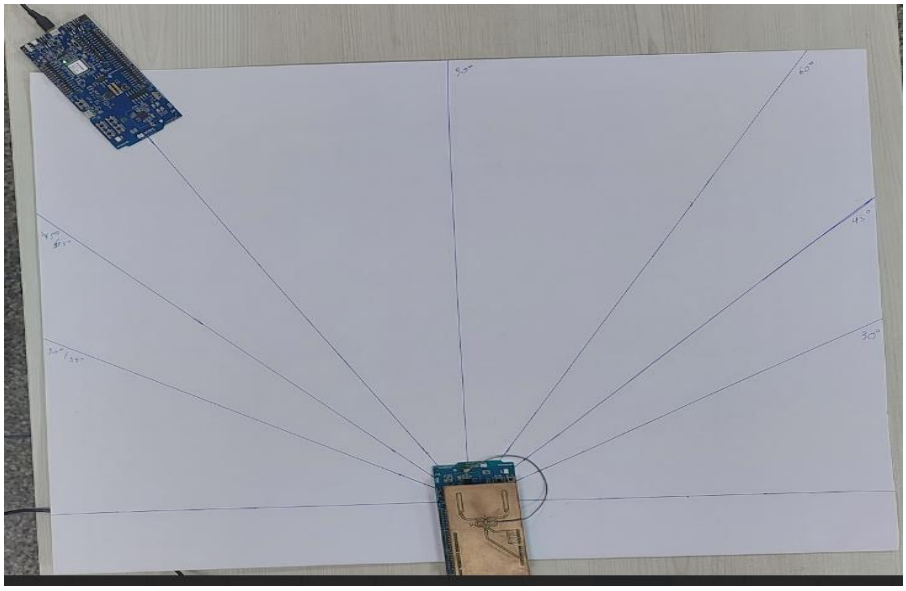
Figura 44 - Gráfico AoA, com beacon posicionado em 90º

Fonte: Autoria Própria.

4.5 RESULTADO PARA 120º

O ângulo a ser validado nessa seção é de 120º, o *beacon* foi posicionado sobre a linha do mesmo, conforme mostrado na figura 45.

Figura 45 - Posicionamento do beacon a 120º do receptor.



Fonte: Autoria Própria.

O gráfico do AoA calculado para 120º, figura 46, demonstra que os valores variam próximo do ângulo de 80º.

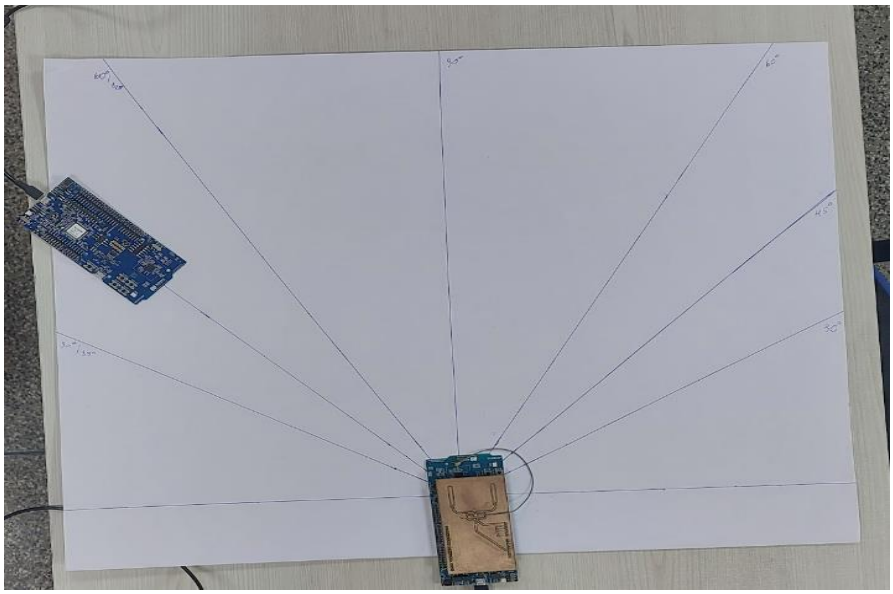
Figura 46 - Gráfico AoA, com beacon posicionado em 120º.

Fonte: Autoria Própria.

4.6 RESULTADO PARA 135º

O ângulo a ser validado nessa seção é de 135º, o *beacon* foi posicionado sobre a linha do mesmo, conforme mostrado na figura 47.

Figura 47 - Posicionamento do beacon a 135º do receptor.



Fonte: Autoria Própria.

O gráfico do AoA calculado para 135º, figura 48, demonstra que os valores variam de forma indefinida.

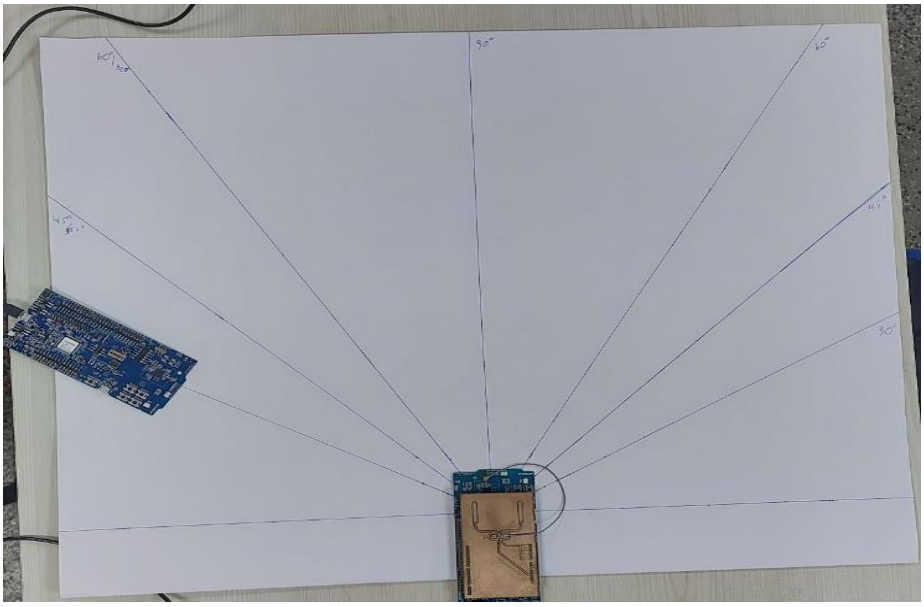
Figura 48 - Gráfico AoA, com beacon posicionado em 135º.

Fonte: Autoria Própria.

4.7 RESULTADO PARA 150º

O ângulo a ser validado nessa seção é de 150º, o *beacon* foi posicionado sobre a linha do mesmo, conforme mostrado na figura 49.

Figura 49 - Posicionamento do beacon a 150º do receptor.



Fonte: Autoria Própria.

O gráfico do AoA calculado para 150º, figura 50, demonstra que os valores variam próximo do ângulo de 80º. O mesmo possui bastante picos com valores próximos de 100º.

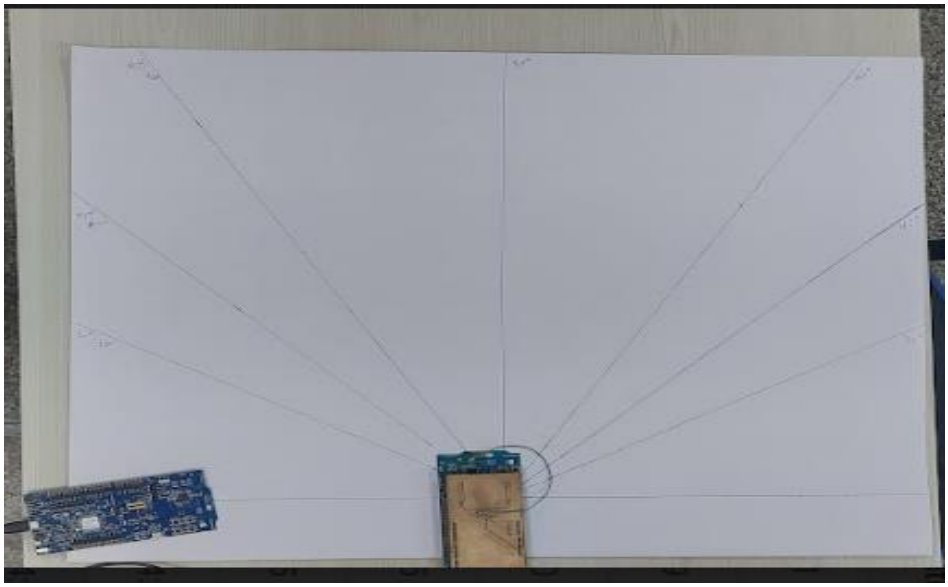
Figura 50 - Gráfico AoA, com beacon posicionado em 150º.

Fonte: Autoria Própria.

4.8 RESULTADO PARA 180º

O ângulo a ser validado nessa seção é de 180º, o *beacon* foi posicionado sobre a linha do mesmo, conforme mostrado na figura 51.

Figura 51 - Posicionamento do beacon a 180º do receptor.



Fonte: Autoria Própria.

O gráfico do AoA calculado para 180º, figura 52, demonstra que os valores variam de forma indefinida, com picos de valores de 160º.

Figura 52 - Gráfico AoA, com beacon posicionado em 180º.

Fonte: Autoria Própria.

CONCLUSÃO

A pesquisa foi desenvolvida com base em informações obtidas em estudos sobre os principais assuntos relacionados a localização *indoor* e determinação do ângulo de chegada usando *Bluetooth Direcion Finding*. Além disso, foi importante determinar quais componentes de *hardware* atendiam a necessidade do projeto, tais como o circuito integrado que foi usado como *Switch RF* e o microcontrolador que possibilitou o uso da técnica do *Bluetooth Direction Findg*, nRF 52833.

Após o desenvolvimento do projeto foram mostradas de forma detalhada as etapas de validação, valendo-se de técnicas para medição de ângulo com transferidor.

Conclui-se então que o protótipo de sistema, da maneira como foi projetado, não está apto a determinar com precisão a direção de emissão do sinal *bluetooth*, pois o ângulo de chegada calculado possui muita variação em seu cálculo. O sistema, composto por um *beacon*, receptor e computador, não trabalhou forma a se obter o AoA com confiança.

Recomenda-se, para implementação de trabalhos futuro, a criação da matriz de antenas com mais número de elementos de captação (no projeto foram duas antenas), além disso, é aconselhado que seja realizado um estudo sobre os parâmetros que uma antena pode ter para melhorar sua eficiência. Outro ponto importante, é a manufatura da placa, recomenda-se a solicitação de uma prototipação mais profissional, valendo-se da criação de mais camadas na placa de circuito impresso, com objetivo de evitar problemas na recepção do sinal. O microcontrolador pode ser substituído por um que possua mais trabalho desenvolvido e uma melhor documentação para avaliação da funcionalidade *Bluetooth Direction Finding.*

REFERÊNCIAS

AKPINAR, Ege. **Bluetooth beacons: Everything you need to know.** [s.l.], 2021. Disponível em: <https://www.pointr.tech/blog/beacons-everything-you-need-to-know>. Acesso em: 10 out. 2021.

ALECRIM, Emerson. **Tecnologia Bluetooth: o que é e como funciona?** [s.l.], 2008. Disponível em: < https://www.infowester.com/bluetooth.php >. Acesso em: 10 out. 2021.
ARAUJO, André Silveira de; VASCONCELLOS, Pedro de. **Bluetooth Low Energy.** Rio de Janeiro, 2012. Disponível em: <https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2012_2/bluetooth/ ble.htm >. Acesso em: 10 out. 2021.

ARAUJO, André Silveira de; VASCONCELLOS, Pedro de. **Bluetooth Low Energy.** Rio de Janeiro, 2012. Disponível em: <https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2012_2/bluetooth/ ble.htm >. Acesso em: 10 out. 2021.

BEVELACQUA, Peter. **Array Antennas,** [s.l.], 2016. Disponível em: < https://www.antenna-theory.com/arrays/main.php#phased >. Acesso em: 03 out. 2022.

BOULOS, Magned N Kamel; BERRY, Geoff. Real-time locating systems (RTLS) in healthcare: a condensed primer. **International Journal of Health Geographics**, Califórnia, V. 11, n. 25, jun. 2012. Disponível em: < https://ij-healthgeographics.biomedcentral.com/articles/10.1186/1476-072X-11-25>. Acesso em: 01 out. 2022.

BRÁS, Luiz Pedro Marques. **Desenvolvimento De Sistema De Localização Indoor De Baixo Consumo.** Dissertação (Engenharia Eletrônica e Telecomunicações) – Universidade de Aveiro, Aveiro, 2009. Disponível em: < https://ria.ua.pt/bitstream/10773/7462/1/5594.pdf>. Acesso em: 02 out. 2022.

CARDOSO, Matheus. **O Que É Um Microcontrolador?** [s.l.], 2020. Disponível em: < https://edu.ieee.org/br-ufcgras/o-que-e-um-microcontrolador/>. Acesso em: 10 out. 2021.

CAY, Elif. *et al*., **Beacons for indoor positioning**. *In: Int. Conf. on Engineering and Technology (ICET)*, 2017, Antalya. Anais eletrônicos... Antalya: IEEE *Xplore*, 2018, p. 1–5. Disponível em: < https://doi.org/10.1109/ICET42298.2017> . Acesso em: 10 out. 2021.

CIPRIANO, Wagner. **Localização Indoor baseada em tecnologia Bluetooth de baixa energia.** Dissertação (Mestrado em Modelagem Matemática e Computacional) – Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, 2018. Disponível em: < https://sig.cefetmg.br/sigaa/verArquivo?idArquivo=2203988&key=275b23e6bc0120b16548418c13905623>. Acesso em: 10 out. 2021.

CRUZ, Bitter A. G *et al*. Um Sistema para Localização em Tempo Real de Terminais Wi-Fi em Ambientes Indoor**. Cadernos do IME: Série Informática,** Rio de Janiero, V. 31, p. 21-42, jun. 2011. Disponível em: < https://doi.org/10.12957/cadinf.2011.6499>. Acesso em: 01 out. 2022.

DRECHSLER, Larissa. **Sistema De Posicionamento Indoor Utilizando Uma Rede De Sensores Sem Fios Baseada No Protocolo De Comunicação Zigbee.** Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Universidade Federal de Santa Maria, Santa Maria, 2018. Disponível em: < http://repositorio.ufsm.br/handle/1/15422>. Acesso em: 10 out. 2021.

FARID, Zahid; NORDIN, Rosdiadee; ISMAIL, Mahamod. Recent Advances in Wireless Indoor Localization Techniques and System, **Journal of Computer Networks and Communications**, v. 2013, Article ID 185138, p. 12, 2013. Disponível em: < https://www.hindawi.com/journals/jcnc/2013/185138/>. Acesso em: 12 out. 2021.

INFINEON TECHNOLOGIES AG. **BGS12PL6.** [s.l.], 2014. Disponível em: < https://www.infineon.com/dgdl/Infineon-BGS12PL6-DS-v02_05-EN.pdf?fileId=db3a30433f1b26e8013f2db8ea4c385f >. Acesso em: 04 jul. 2022.

JEREZ, Gabriel Oliveira; ALVES, Daniele Barroca Marra. GLONASS: Revisão teórica e estado da arte**. Revista Brasileira de Geomática,** Curitiba, V. 6, n. 2, p. 155-173, abr/jun. 2018. Disponível em: < https://periodicos.utfpr.edu.br/rbgeo>. Acesso em: 02 out. 2022.

MENDES, Raul de Queiroz *et al*. **Tecnologias, métodos e teorias na engenharia de computação:** Sistema De Localização Híbrido Baseado Em Nuvem Para Ambientes Internos E Externos**.** Ponta Grossa: Atena, 2020. Disponível em: < https://sistema.atenaeditora.com.br/index.php/admin/api/artigoPDF/41967 >. Acesso em: 11 out. 2021.

NORDIC SEMICONDUCTOR. **nRF52833 Product Specification.** [s.l.], 2021. Disponível em: < https://infocenter.nordicsemi.com/pdf/nRF52833_PS_v1.4.pdf>. Acesso em: 11 out. 2021.

PENILDO, Édilus de Carvalho Castro; TRINDADE, Ronaldo Silva. **Microcontroladores.** Santa Maria: Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria; Rede eTec Brasil, 2013. Disponível em: < https://www.ifmg.edu.br/ceadop3/apostilas/microcontroladores/@@download/file/microcontroladores.pdf>. Acesso em: 11 out. 2021.

PORTER, Donovan; MYHR, Reidar. **Bluetooth® Angle of Arrival (AoA) Antenna Design.** [s.l.], 2019. Disponível em: < https://www.ti.com/lit/an/tida029/tida029.pdf>. Acesso em: 04 jul. 2022.

PU, Chuan-Chin *et al*. Indoor Location Tracking Using Received Signal Strength Indicator, **Emerging Communications for Wireless Sensor Network**, Malaysia, South Korea, V. 1, n. 11, p. 230 – 256, feb. 2011. Disponível em: < https://www.intechopen.com/chapters/13525 >. Acesso em: 01 out. 2022.

WOOLLEY, Martin. **Bluetooth Direction Finding:** A Technical Overview. [s.l.], 2021. Disponível em: <https://www.bluetooth.com/wp-content/uploads/Files/developer/1903_RDF_Technical_ Overview_FINAL.pdf>. Acesso em: 11 out. 2021.