

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**DENNIS GUILHERME DE MACEDO BRAGAGNOLO**

**RAFAEL RUVINSKI**

**CHAVEADOR *WIRELESS* PARA TECLADO E MOUSE**

**CURITIBA**

**2022**

**DENNIS GUILHERME DE MACEDO BRAGAGNOLO  
RAFAEL RUVINSKI**

**CHAVEADOR *WIRELESS* PARA TECLADO E MOUSE**

**Wireless Switch for Keyboard and Mouse**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção do  
título de Bacharel em Engenharia Eletrônica do  
Curso Superior de Engenharia Eletrônica da  
Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Bruno Sens Chang

**CURITIBA**

**2022**



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**DENNIS GUILHERME DE MACEDO BRAGAGNOLO  
RAFAEL RUVINSKI**

**CHAVEADOR *WIRELESS* PARA TECLADO E MOUSE**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção do  
título de Bacharel em Engenharia Eletrônica do  
Curso Superior de Engenharia Eletrônica da  
Universidade Tecnológica Federal do Paraná.

Data de aprovação: 13/junho/2022

---

Bruno Sens Chang  
Doutor  
Universidade Tecnológica Federal do Paraná

---

Luciano Scandelari  
Doutor  
Universidade Tecnológica Federal do Paraná

---

André Eugênio Lazzaretti  
Doutor  
Universidade Tecnológica Federal do Paraná

**CURITIBA  
2022**

*Àqueles que nos são caros.*

## **AGRADECIMENTOS**

É impossível prestar, neste poucos parágrafos, as devidas homenagens a todas as pessoas que tiveram parte nessa importante conquista de nossas vidas, apesar de nosso desejo de contemplá-las.

Os agradecimentos são direcionados inicialmente, como não poderia deixar de ser, à família, em especial às nossas companheiras de vida, Heloisa e Paola, por todo apoio e incentivo e, ainda, pela compreensão da necessidade de afastamento em razão das diversas horas expendidas no desenvolvimento deste trabalho de conclusão de curso.

Considerando que os amigos foram outros afetados pelas eventuais ausências motivadas pela realização do trabalho, fica ora registrada também a nossa gratidão.

Este trabalho é, ainda, resultado das contribuições de nossos colegas de sala e de trabalho, a quem não poderíamos deixar de lembrar.

Também não poderíamos deixar de prestar uma singela homenagem aos nossos professores, que tanto contribuíram para que chegássemos aqui, bem como como à própria instituição.

Assim, de forma breve, mas muito afetuosa, deixamos registrado o nosso muito obrigado a todos!

*Temos que ser capazes de transmitir as coisas de outros lugares em longas distâncias: imagens, a notícia, a energia. Agora, temos que libertar o pensamento, devemos nos livrar das limitações que o espaço e o tempo nos impõe. E ainda manter as suas principais características. Agora e em séculos futuros... A minha torre. Essa é a realidade, vocês verão...*  
(Nikola Tesla)

## RESUMO

O *home office* é uma tendência já sentida nos últimos anos e que ganhou uma escala monumental com a pandemia do novo coronavírus, em 2020, quando cerca de 11 por cento dos trabalhadores brasileiros chegou a atuar nesta modalidade. Juntamente com o trabalho remoto, também ganharam força as preocupações com a ergonomia e a saúde mental dos trabalhadores, uma vez que nem sempre o ambiente residencial possui a estrutura adequada ao desempenho do trabalho à distância e muitas vezes nem mesmo área disponível para uma reformulação, o que exige o melhor aproveitamento do espaço. Visando a propiciar essa melhor utilização espacial para aqueles trabalhadores que atuam com mais de um computador, geralmente um para uso pessoal e outro profissional, o projeto propõe reaproveitar um mesmo conjunto de teclado e mouse através de um dispositivo chaveador que dispense a troca física de conectores. À parte da redução do número de periféricos nas mesas e bancadas dos usuários, o chaveador proposto também tem o potencial de reduzir o número de cabos mediante o uso de uma das tecnologias sem fio disponíveis no mercado, o Bluetooth Low Energy. Visando a comportar o uso dos teclados e mouses de preferência do usuário, sem prendê-los aos poucos leiautes comercializados com a funcionalidade de multidispositivos, o chaveador proposto não está acoplado a um periférico específico, mas dispõe de uma entrada USB para conexão dos dispositivos. Desta forma, com o uso do chaveador sem fio, o usuário ganha em harmonia em seu ambiente residencial-laboral, contribuindo para seu bem-estar e produtividade a um custo de produção bastante razoável, de pouco mais de cem reais.

**Palavras-chave:** bluetooth low energy; usb; mouse; teclado; sem fio.

## ABSTRACT

Working from home has been a trend in recent years and has gained a monumental scale due to the coronavirus pandemic that started in 2020, when about 11 percent of Brazilians workers worked in this modality. Along with remote work, the concerns about ergonomics and workers' mental health also gained strength, especially considering that the residential environment does not always have the appropriate structure for the remote work and often not even area available for a redesign, which requires the best possible use of space. In order to provide this better use of space for those workers who use more than one computer, usually one personal and other from the company, the project proposes to use the same set of keyboards and mouse through a switching device that does not demand the physical exchange of connectors. Apart of reducing the number of peripherals on users' tables and benches, the proposed switch also has the potential to reduce the number of cables by using one of the wireless technologies available on the market, the Bluetooth Low Energy. Aiming to provide the utilization of the user's preferred keyboards and mice, without tying them to a few layouts, the proposed switch is not bound to a specific peripheral, but has an USB port for connecting devices. In this way, with the use of the wireless switch, the user gains in harmony in their residential-work environment, contributing to their well-being and productivity at a very reasonable production cost of just over a hundred Brazilian *reais*.

**Keywords:** bluetooth low energy; usb; mouse; keyboard; wireless.



## LISTA DE TABELAS

Tabela 1 – Velocidade de transmissão de dados nas gerações do USB . . . . .	20
Tabela 2 – Principais avanços na tecnologia Bluetooth . . . . .	25
Tabela 3 – Medição de intensidade do sinal Bluetooth do chaveador . . . . .	53
Tabela 4 – Resultados da medição de consumo energético pelo chaveador . . . . .	53
Tabela 5 – Custos para a fabricação unitária de um chaveador wireless . . . . .	55
Tabela 6 – Valor total para a produção do chaveador . . . . .	55

## LISTA DE ABREVIATURAS E SIGLAS

### Siglas

BLE	<i>Bluetooth Low Energy</i>
BT	Bluetooth
GAP	<i>Generic Access Profile</i>
GATT	<i>Generic Attribute Profile</i>
GPIO	<i>General Purpose Input/Output</i>
HID	<i>Human Interface Device</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	<i>Integrated Development Environment</i>
MAC	<i>Media Access Control</i>
MISO	<i>Master In. Slave Out</i>
MOSI	<i>Master Out. Slave In</i>
OEM	<i>Original Equipment Manufacturer</i>
PC	Computador Pessoal (do inglês Personal Computer)
PCB	<i>Printed Circuit Board</i>
PCI	Placa de Circuito Impresso
ROM	<i>Read-Only Memory</i>
SIE	<i>Serial Interface Engine</i>
SIG	<i>Special Interest Group</i>
SPI	<i>Serial Peripheral Interface</i>
SRAM	<i>Static Random Access Memory</i>
USB	<i>Universal Serial Bus</i>

## LISTA DE SÍMBOLOS

### LETRAS LATINAS

<i>A</i>	Ampére	
<i>B</i>	Byte	
<i>b</i>	Bit	
<i>C</i>	Capacitância	[F]
<i>I</i>	Corrente elétrica	[A]
<i>d</i>	Distância/Comprimento	[m]
<i>V<sub>in</sub></i>	Tensão de entrada	[V]
<i>dBm</i>	Decibel miliwatts	
<i>F</i>	Farad	
<i>f</i>	Frequência	[Hz]
<i>Hz</i>	Hertz	
<i>m</i>	Metro	
<i>P</i>	Potência	[W]
<i>R</i>	Resistência	[Ω]
<i>t</i>	Tempo	[s]
<i>V</i>	Tensão elétrica	[V]
<i>V</i>	Volt	
<i>W</i>	Watt	
<i>Wh</i>	Watt-hora	

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	O aumento do trabalho remoto	13
1.2	Objetivos e proposta	14
1.3	Estrutura do trabalho	17
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>18</b>
<b>2.1</b>	<b>USB</b>	<b>18</b>
2.1.1	História	18
2.1.2	Visão Geral	19
2.1.3	Topologia	20
2.1.4	Dispositivos de Interface Humana (HID)	22
<b>2.2</b>	<b>Comunicação sem fio</b>	<b>23</b>
<b>2.3</b>	<b>Bluetooth Low Energy</b>	<b>24</b>
2.3.1	História	24
2.3.2	Visão Geral	25
2.3.3	Topologia	27
2.3.4	Camadas, Protocolos e Perfis	29
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>31</b>
<b>3.1</b>	<b>Hardware</b>	<b>31</b>
3.1.1	ESP32	31
3.1.2	USB-Host	32
3.1.3	PCB	34
<b>3.2</b>	<b>Firmware</b>	<b>36</b>
3.2.1	Transmissão via BLE	38
3.2.2	Leitura USB	39
3.2.3	Compatibilização de leitura e transmissão	41
3.2.4	Transmissão múltipla	41
3.2.5	Chaveamento	42
3.2.6	Refinamento	43
<b>3.3</b>	<b>Embarcamento</b>	<b>45</b>
<b>4</b>	<b>TESTES E RESULTADOS</b>	<b>46</b>

4.1	Ambiente de testes . . . . .	46
4.2	Testagem . . . . .	47
4.3	Alcance e intensidade do sinal . . . . .	50
4.4	Consumo energético . . . . .	52
4.5	Custos de produção . . . . .	53
5	CONCLUSÃO . . . . .	55
5.1	Trabalhos futuros . . . . .	55
	REFERÊNCIAS . . . . .	57

# 1 INTRODUÇÃO

## 1.1 O aumento do trabalho remoto

A pandemia do Coronavírus, que tomou definitivamente conta do mundo todo a partir de meados de março de 2020, trouxe consigo, além do grande impacto na saúde e da voluptuosa modificação nas relações sociais, profunda alteração nas relações de trabalho. Foram diversas restrições impostas a toda a sociedade visando a conter o avanço da propagação da doença, as quais ultrapassaram em muito o reforço dos cuidados com a higiene – já vivenciado alguns anos antes nas crises sanitárias das gripes suína e aviária – e a obrigatoriedade do uso de máscaras, chegando ao fechamento de inúmeros estabelecimentos e à suspensão de quase toda atividade presencial que não fosse considerada essencial.

Isto levou muitas empresas a buscarem formas de minimizar seu prejuízo, adaptando ao regime de teletrabalho – com as novas facilidades tecnológicas de comunicação à distância – as atividades que assim permitiam<sup>1</sup>, em uma migração em massa que acelerou muito um processo que já vem se intensificando nos últimos anos.

Segundo dados do Instituto Brasileiro de Geografia e Estatística (IBGE), divulgados na Pesquisa Nacional por Amostra de Domicílios, o trabalho remoto alcançou, em 2020, o patamar de 11 por cento da população brasileira, ou seja, "foram 8,1 milhões de trabalhadores que tiveram de improvisar estações de trabalho e conciliar rotinas profissionais e pessoais enquanto seguiam suas jornadas a partir de suas casas"(BRIGATTI, 2021).

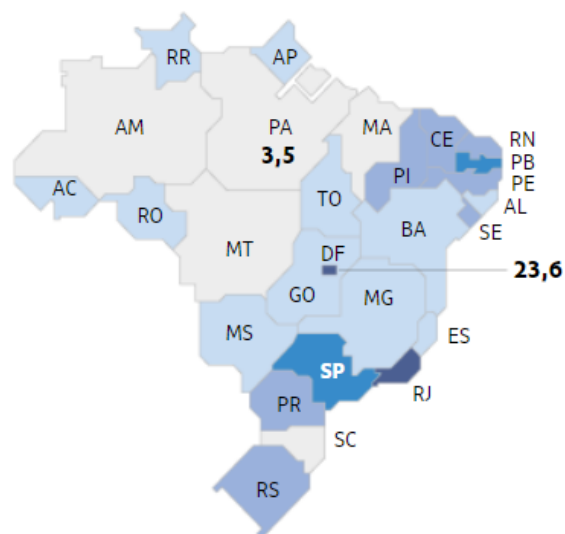
**Figura 1 – Trabalho remoto no Brasil em 2020.**

Trabalho remoto no Brasil em 2020

**Percentual de trabalho remoto na população ocupada**

- Mais de 16,1
- 12,1 a 16
- 9,1 a 12
- 6,1 a 9
- Até 6

**11** foi a média do país



**Fonte: Folha de São Paulo.**

<sup>1</sup> Estima-se que cerca de 73 por cento das empresas brasileiras adotaram o regime remoto durante a pandemia (ZAIDEM, 2021).

Assim, de uma hora para outra, grande parte dos trabalhadores – aqueles que não perderam seu sustento durante a pandemia – foi obrigada a improvisar postos de trabalho em suas residências para alocar os equipamentos de trabalho, principalmente computadores. Como é de conhecimento geral, não são todos os trabalhadores que possuem o luxo de manter, em suas casas, um ambiente exclusivo que lhes permita separar as atividades laborativas das cotidianas, especialmente se considerando a também progressiva redução do tamanho dos apartamentos nas últimas décadas<sup>2</sup>. Desta maneira, as mesas e bancadas antes destinadas a computadores de uso pessoal passaram a comportar também os computadores de trabalho, caracterizando um cenário bastante comum de abarrotamento de equipamentos eletrônicos, devido à duplicação da quantidade de notebooks, monitores, teclados e mouses.

O aumento do trabalho remoto já era uma tendência antes mesmo da pandemia e se intensificou nos últimos anos pelas razões sanitárias, existindo uma grande inclinação de diversas empresas em manter o formato nos próximos anos<sup>3</sup>.

A francesa Ipsos, uma das maiores empresas de pesquisas de todo o mundo, realizou recentemente pesquisa com mais de 12 mil trabalhadores distribuídos em 29 países que constatou esta tendência. Os números divulgados trazem que a média dos países de trabalhadores em trabalho ao menos parcialmente remoto aumentou de cerca de 24 por cento antes da pandemia para 39 por cento em 2021, sendo que uma média de 75 por cento gostaria de continuar trabalhando de suas casas ao menos um dia por semana após o final das restrições (IPSOS, 2021). Segundo este mesmo estudo, no Brasil, o percentual dos entrevistados em trabalho remoto chegou a 50 por cento e 80 por cento dos entrevistados gostaria de manter ao menos um dia de *home office*.

## 1.2 Objetivos e proposta

Diante desta nova realidade, além da própria produtividade, aspectos como ergonomia, conforto e saúde mental dos trabalhadores passam a ser uma grande preocupação. Uma forma de contribuir ao menos em um destes aspectos, permitindo a redução da quantidade de aparelhos e cabos sobre as mesas e bancadas pessoais dos trabalhadores residenciais, é a utilização de um mesmo conjunto de periféricos de mouse e teclado para mais de um computador. Neste sentido, mostra-se bastante apropriada a ideia de que teclas e cliques de um mesmo conjunto de teclado e mouse sejam transmitidas para diferentes aparelhos sem que haja a necessidade de desconectá-lo de um e reconectá-lo ao outro, bastando um simples pressionamento de botão para alterar o dispositivo de destino.

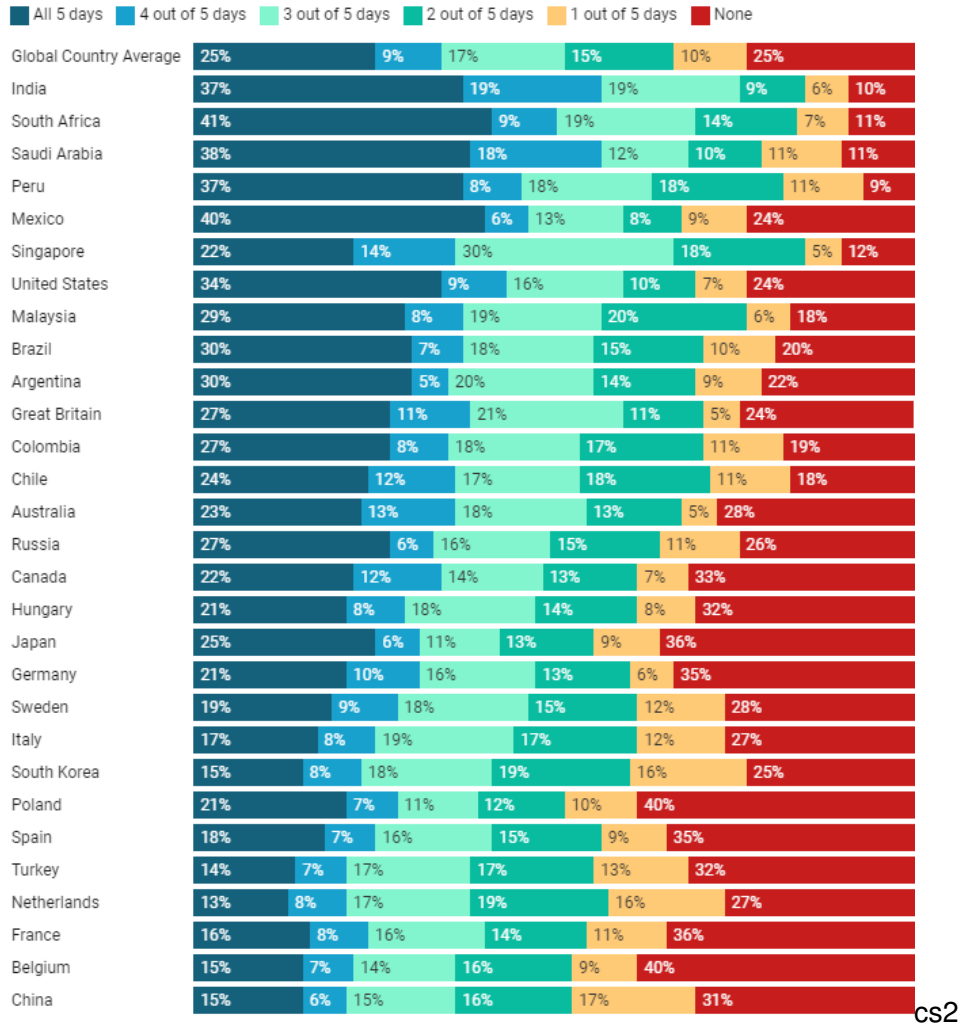
<sup>2</sup> Estudos demonstram que, por diversos motivos, dentre os quais se destaca o encarecimento do metro quadrado, as moradias urbanas têm tido seu tamanho reduzido nos últimos anos, com destaque para os apartamentos de um quarto, que registraram uma redução de 40 por cento no tamanho médio entre 2009 e 2019 (MARRA, 2019).

<sup>3</sup> Levantamento realizado pelo site brasileiro de vagas de emprego Empregos.com.br indica o aumento de aproximadamente 140 por cento nas ofertas de vagas em regime *home office* no ano de 2021 (MIRANDA, 2022).

**Figura 2 – Quantidade de dias em *home office* desejada pelos trabalhadores no retorno ao trabalho presencial.**

### Shift back to working outside of home

Q. When the pandemic is over and all pandemic restrictions have been lifted, if you were working on average 5 days a week, how many days per week would you prefer working from home?



Fonte: Ipsos.

cs2

Além da adequação ao *home office*, a quantidade de dispositivos eletrônicos também não para de crescer, tornando-se cada vez mais comum a pluralidade de aparelhos disponíveis a uma mesma pessoa. Neste sentido, dados de pesquisa divulgada pela Fundação Getúlio Vargas apontam que o brasileiro possui, em média, 2 dispositivos eletrônicos em uso, sendo cerca de 53 por cento composto por celulares e o restante por computadores (*notebooks*, *desktops* e *tablets*) (Poder 360, 2021). Ou seja, o dispositivo acima mencionado traria benefícios e encontraria mercado também entre aqueles que não dividem sua mesa entre os computadores pessoal e de trabalho mas possuem mais de um dispositivo, além, é claro, das pessoas que utilizam profissionalmente mais de um dispositivo ao mesmo tempo.

Nas pesquisas realizadas, foram encontradas algumas alternativas para a utilização de um único conjunto de periféricos para mais de um computador: o chamado KVM switch e alguns conjuntos específicos de teclado e mouse da fabricante Logitech. O KVM switch permite alternar



os sinais de teclado (keyboard), vídeo e mouse entre dispositivos, mas os modelos disponíveis no mercado pressupõem a ligação cabeada entre o switch e os dispositivos destino. Já os conjuntos da fabricante Logitech, como o MK850, permitem a comunicação via Bluetooth com os dispositivos e a alternância entre eles mediante o pressionamento de um botão, entretanto estão disponíveis apenas algumas opções de layout, todas com formato de teclado americano, e que possuem um custo bastante considerável no mercado brasileiro (cerca de 500 reais).

**Figura 3 – Chaveador KVM.**



Fonte: Wikipedia.

**Figura 4 – Conjunto de teclado e mouse multi-dispositivos MK850.**



Fonte: Logitech.

O presente trabalho tem como escopo o desenvolvimento de um dispositivo capaz de contribuir com o trabalho remoto nos termos acima e de mitigar os poréns apresentados, ou seja, um dispositivo que realize o chaveamento dos sinais dos periféricos pelo pressionamento de um botão, mas sem a utilização de cabos ligados aos computadores, sem limitar os layouts de teclado a apenas algumas poucas opções e com menor custo. Para isso, propõe-se a utilização da tecnologia Bluetooth e de um microcontrolador de baixo custo já disponível no mercado.

Em convergência com a proposta acima, alguns objetivos a serem resguardados durante o desenvolvimento foram definidos, iniciando-se pela redução da quantidade de periféricos necessários para o desempenho de atividades em múltiplos computadores, assim como a quantidade de cabos nas mesas e estações de trabalho. Além disso, a simplicidade, facilidade e rapidez no chaveamento também devem nortear a elaboração do projeto, bem como o baixo custo final.

### 1.3 Estrutura do trabalho

O presente trabalho foi organizado em cinco capítulos, com a introdução já incluída no primeiro deles, na qual se apresentam os fundamentos que levaram à escolha do presente projeto, bem como os objetivos a serem alcançados e alguns aspectos relativos à organização do documento. No referencial teórico, segundo capítulo, é apresentada a fundamentação teórica dos principais conceitos que orientam o estudo. É, portanto, capítulo essencial para a compreensão do trabalho e dos métodos e materiais utilizados, com exposição, em profundidade considerada suficiente, do histórico, da topologia e dos principais conceitos das tecnologias utilizadas no projeto.

No terceiro capítulo, detalha-se o desenvolvimento do projeto e a estrutura escolhida, elucidando-se os materiais e métodos utilizados e as principais escolhas para a conclusão deste estudo. O capítulo subsequente detalha os testes realizados no curso do trabalho e seus resultados, as dificuldades em que se esbarrou e as alternativas encontradas para suplantá-las, assim como os custos envolvidos na fabricação do protótipo desenvolvido.

Como desfecho, nas páginas derradeiras, são tecidas as considerações finais sobre o projeto, com a apresentação de pertinentes observações e as lições aprendidas, além de possíveis aprimoramentos ao produto desenvolvido para que seja alcançada uma melhor experiência dos usuários.

Em relação à terminologia utilizada no presente documento, são necessárias algumas ressalvas, iniciando-se pela utilização do termo "computador", que, exceto quando outro significado foi dado de maneira explícita, foi utilizado de forma ampla, representando todos os dispositivos modernos que se enquadram nesta categoria, tais como *desktops*, *notebooks*, *smartphones*, *smarTVs* e outros. Também, a fim de evitar uma possível confusão entre o termo genérico "dispositivo" para designar aparelhos eletrônicos em geral e a "função de dispositivo" (*device*) nas conexões USB, priorizou-se o uso do termo em inglês quando havia intenção de se referir de forma específica a esta última acepção.

## 2 REFERENCIAL TEÓRICO

Antes de adentrar o mérito do desenvolvimento do trabalho, faz-se necessário, para melhor compreensão, revisitar-se alguns dos conceitos envolvidos, bem como um breve mergulho nas tecnologias envolvidas e seu funcionamento. Esta jornada tem início – apenas por formalidade, tendo em vista que são universalmente conhecidos – em uma sintética definição dos dois componentes mais importantes do presente trabalho: mouse e teclado.

Reconhecido como o dispositivo primário de entrada de dados em um computador, um teclado é utilizado, além da operação no computador, tipicamente utilizado na inserção de texto, cujo padrão prevê a existência de ao menos 103 teclas (USB Implementers' Forum, 2022).

Também um dispositivo de entrada de dados, o mouse foi pensado de modo a facilitar a experiência dos usuários dos computadores que ganhou importância a partir da evolução da interface gráfica dos sistemas operacionais. Pode ser definido como um dispositivo manual que, quando rolado por uma superfície plana, direciona um indicador de forma correspondente na tela de um computador, permitindo, através do uso dos botões e do mapeamento do movimento em 2 eixos, que o operador realize seleções e manipulações gráficas ou de texto na utilização do computador (USB Implementers' Forum, 2022).

### 2.1 USB

Uma das mais utilizadas interfaces em todo o mundo, a Porta Serial Universal, ou *Universal Serial Bus* (USB), está presente de alguma forma em praticamente todos os dispositivos eletrônicos produzidos atualmente, como computadores, tablet e smartphones, além dos periféricos que são conectados a eles, como teclados, mouses, impressoras, pen drives e inúmeros outros.

As razões para o emprego quase universal desta tecnologia são diversas: o USB é considerado versátil, confiável, rápido, de baixos consumo energético e custo e compatível com os sistemas operacionais de grandes e pequenos computadores (AXELSON, 2015).

#### 2.1.1 História

Apesar de hoje o USB ser amplamente utilizado, nem sempre foi assim. Idealizada nos anos 90 e lançada em 1996 por um grupo de fabricantes americanas que incluía IBM, Intel e Microsoft, a tecnologia encontrou uma certa resistência para seu emprego: as fabricantes de computadores iam lentamente incluindo a nova interface em seus dispositivos e as fabricantes de periféricos também relutavam em adotá-la antes de se tornar um padrão, além da incompatibilidade dos sistemas operacionais. Foi somente em 1998, com a popularidade da primeira geração do iMac, lançada pela Apple e que contava apenas com portas USB, que o USB passou

a ser utilizado pelas demais fabricantes (BRITANNICA, 2021), além do lançamento do Windows 98, que corrigiu *bugs* e melhorou a compatibilidade da versão 1.0 (AXELSON, 2015).

Desde o lançamento da primeira versão do USB, a interface se encontra em constante aperfeiçoamento, com o lançamento de novas versões para acompanhar o avanço da tecnologia empregada nos dispositivos e o aumento crescente da quantidade de dados a ser transmitida, chegando, em 2019, à versão 4.0, capaz de transferir dados a uma velocidade de 40 Gbps.

**Tabela 1 – Velocidade de transmissão de dados nas gerações do USB**

Versão	Lançamento	Velocidade máxima de transmissão
1.0	1996	12 Mbit/s
1.1	1998	12 Mbit/s
2.0	2001	480 Mbit/s
3.0	2011	5 Gbit/s
3.1	2014	10 Gbit/s
3.2	2017	20 Gbit/s
4.0	2019	40 Gbit/s

Fonte: Adaptado de uk.rs-online.com.

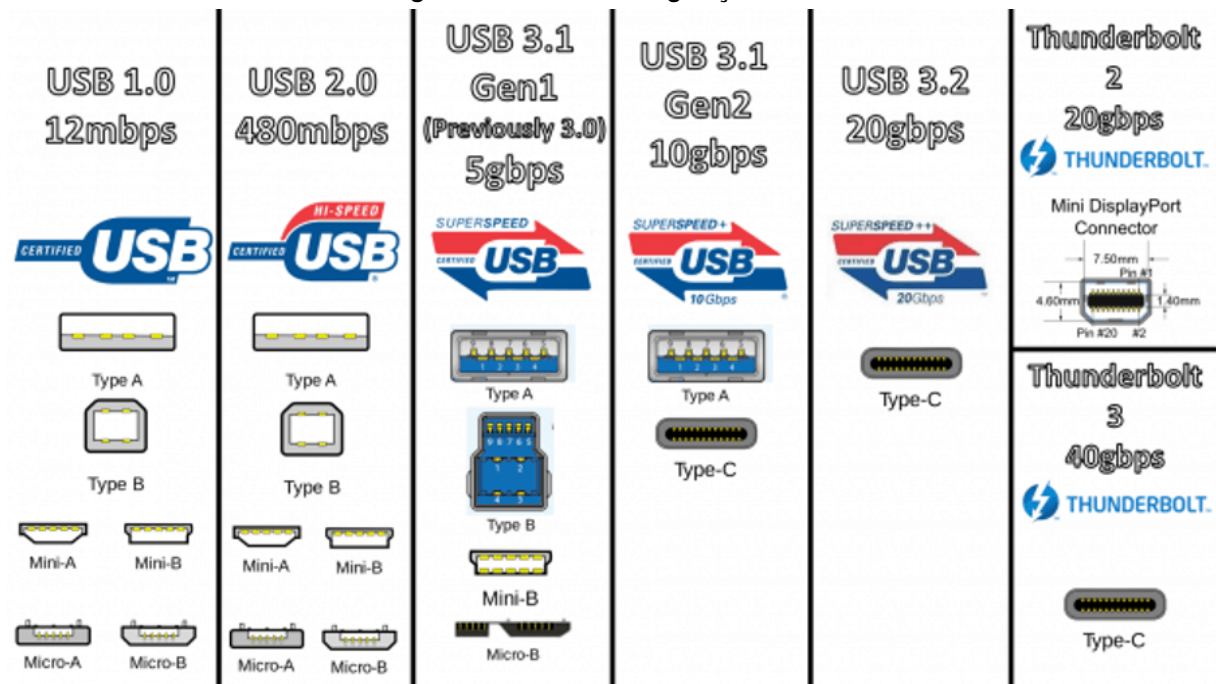
### 2.1.2 Visão Geral

O padrão USB foi inicialmente concebido para a transmissão de dados de forma serial em dois cabos (USB Implementers' Forum, 2001), com outros dois para fornecimento de alimentação, totalizando quatro pinos e cabos posicionados e coloridos de forma padrão: vermelho (5V), branco (D+), verde (D-) e preto (terra). Entretanto, devido aos avanços para introdução de novas velocidades e funcionalidades, houve o rearranjo dos pinos para acomodá-los na variedade de tipos de conectores USB que hoje conhecemos, conforme demonstrado na Figura 5.

Para efetuar a comunicação com os periféricos conectados, o computador deve possuir um controlador USB, chamado frequentemente de *Root Hub* e com o qual o software se comunica a fim de gerir o dispositivo conectado (LUNT, 2014).

Devido a estas características de desenvolvimento, tem-se como vantagem dos dispositivos USB em relação aos típicos seriais e paralelos a capacidade de serem conectados e operar sem a necessidade de desligar ou reiniciar o computador. Além disso, quando inicializada a conexão entre os dispositivos, o computador, por meio do software *host*, é capaz identificar o dispositivo mediante a requisição de pacotes, incluindo *drivers*, o que muitas vezes dispensa a instalação manual pelo usuário. Uma outra vantagem do USB é a capacidade de comportar até 127 dispositivos, incluindo *hubs* e periféricos, em um único barramento, graças à sua topologia em forma de estrela em camadas (*tiered star*)

Figura 5 – Conectores e gerações USB.



Fonte: Cambrionix.

### 2.1.3 Topologia

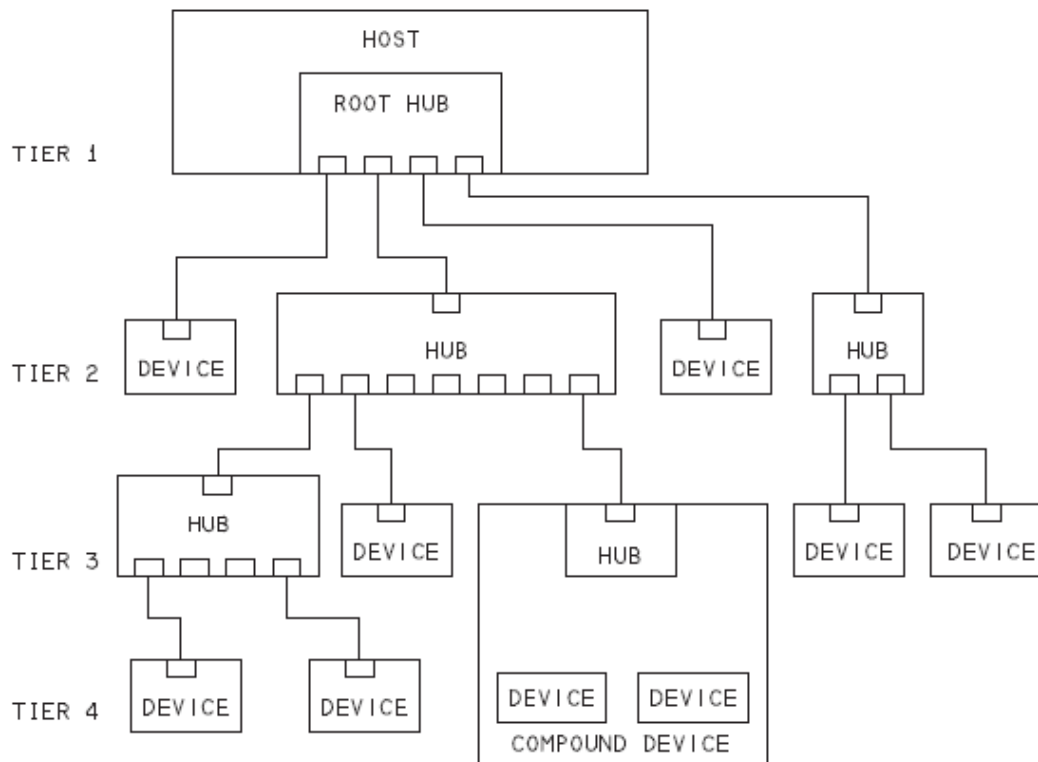
O barramento do USB foi idealizado de modo a disponibilizar a conexão de diversos dispositivos e camadas em um mesmo barramento, mediante a utilização de *hubs*, na topologia conhecida como *tiered star*, ou estrela em camadas. Desta maneira, é possível estabelecer comunicação de até 127 dispositivos em um mesmo *root hub*, que podem estar distribuídos em até 5 níveis ou camadas, como exemplificado na figura 6.

Para que se tenha sucesso na transmissão de dados via USB, a comunicação entre os dispositivos pressupõe a configuração de duas funções distintas a serem desempenhadas: a de *host* e a de *device*.

A função de *host* demanda a existência de hardware e software capazes: o hardware, como já tratado anteriormente, geralmente é composto de um controlador e um *root hub* que possui uma ou mais portas, enquanto o software é tipicamente um sistema operacional que permite a comunicação com os drivers dos dispositivos conectados ao hardware USB (AXELSON, 2015). Desta maneira, o *host* possui grande parte das responsabilidades da transferência de dados em uma comunicação USB, como a detecção de *devices*, o gerenciamento do fluxo de dados, a verificação de erros e o fornecimento e gerenciamento de energia para os *devices*.

Já um *device*, atuando como um espelho do *host*, também possui algumas funções específicas: detectar e responder às comunicações iniciadas pelo *host* e também realizar checagem de erros e gerenciamento da energia, além, é claro, das funções para as quais o dispositivo foi desenvolvido, como a detecção de movimentos e cliques no caso de um mouse.

Figura 6 – Topologia USB.



Fonte: (AXELSON, 2015).

A transmissão de dados via USB é realizada por meio de transferências iniciadas pelo *host*, que carregam dados, endereços, bits de detecção de erros e informações de status e controle. Os dados são carregados para ou a partir dos *endpoints* – espécie de *buffers* dos *devices* – por meio das transferências, que possuem formatos específicos de acordo com o seu tipo e direção, sendo formadas por uma ou mais transações, as quais, por sua vez, contêm pacotes de informação e são classificadas do ponto de vista do *host*, podendo ser de entrada, saída ou configuração (AXELSON, 2015).

De acordo com suas características, divide-se as transferências em quatro categorias: de controle, em massa, de interrupção e isocrônicas. As de controle servem para definir os parâmetros da comunicação, estabelecendo o tipo de dispositivo, seu endereço e outras configurações, devendo ser suportadas por todo *device*. Transferências em massa, por sua vez, são geralmente utilizadas em impressoras e scanners e são caracterizadas por carregar uma considerável quantidade de dados, mas que não dependem de uma resposta rápida ou tampouco alta velocidade de transmissão, podendo aguardar em caso de congestionamento do barramento. Mouses e teclados utilizam as interrupções, que são transferências caracterizadas por demandar atenção do *host* periodicamente a fim de minimizar o *delay* (ou latência). Por fim, as transferências isocrônicas são aquelas que necessitam de uma taxa de transferência de dados mínima, mas que permitem a ocorrência de erros ou perdas para garanti-la, como nas transmissões de áudio e vídeo, por exemplo.

É importante notar que um *device* pode (e deve) comportar diferentes tipos de transferências. Em primeiro lugar, todos devem ser suportar – além das transferências adequadas ao tipo de dispositivo, como as de interrupção no caso de um mouse – as transferências de controle, para que o *host* identifique e configure adequadamente o dispositivo. Também pode ocorrer de um mesmo *device* utilizar transferências distintas de acordo com o tipo de interação, como ocorre nas caixas de som, em que são utilizadas transferências isocrônicas para transmissão do áudio e transferências de interrupção para controle do volume, por exemplo. A fim de orientar o desenvolvimento de dispositivos com a devida compatibilidade, há sua divisão em classes que especificam o comportamento comum e os protocolos para dispositivos com função similar, como nos casos de dispositivos de áudio, de armazenamento ou de interface humana (HID).

#### 2.1.4 Dispositivos de Interface Humana (HID)

A classe de dispositivos de interface humana (em inglês *Human Interface Device* (HID)) traz a especificação para *devices* que é utilizado para transmitir informação para o *host* por meio de alguma forma de interação humana, sendo mouse e teclado os dispositivos mais conhecidos deste classe (LUNT, 2014).

Um HID deve necessariamente possuir um *endpoint* IN e pode possuir ou não um OUT. Nessa classe, a transmissão de dados com o *host* ocorre por meio de relatórios (*reports*), que são estruturas de tamanho fixo transportadas em transferências de controle ou interrupção. As transferências de interrupção são utilizadas para que ocorra a checagem periódica pelo *host*, pois, em razão da finalidade destes dispositivos, geralmente é impossível saber exatamente o instante em que a transmissão vai ocorrer e quando deve haver a leitura, ou seja, quando um usuário vai mover um mouse, por exemplo.

Os detalhes do HID e dos seus relatórios são identificados pelo *host* por meio da leitura dos descritores do HID, que, como o próprio nome indica, descrevem as características de sua operação, tais como o tipo de dispositivo, o tamanho e formato dos relatórios, intervalo de *polling* e o número de interfaces. Desta maneira, cada aspecto do dispositivo possui um descritor associado que é responsável por definir os parâmetros que serão utilizados, como as informações do próprio dispositivo, a configuração, as interfaces, os relatórios e os *endpoints*.

Realizada a leitura dos descritores pelo *host* na fase de enumeração, este passa a fazer requisições ao *device* de acordo com os parâmetros estabelecidos. A leitura das informações é realizada periodicamente mediante a recuperação pelo *host* dos dados armazenados no *buffer* do *device* de forma assíncrona, geralmente em uma frequência alta o suficiente para que a latência seja imperceptível ao usuário.

## 2.2 Comunicação sem fio

Para que ocorra qualquer tipo de comunicação, deve haver ao menos um transmissor, que, como o próprio nome já diz, é quem faz a transmissão da informação, e outro receptor, que, por sua vez, capta a informação transmitida. Portanto, a geração e codificação da informação normalmente ocorre do lado do transmissor, enquanto a decodificação do sinal recebido é papel do receptor.

A comunicação sem fio pode ser entendida como qualquer tipo de comunicação que não envolve a utilização de cabos, com a transmissão de informação entre dois ou mais pontos que não estão fisicamente conectados por cabos e que podem estar separados por alguns centímetros ou milhares de quilômetros (GUPTA, 2016). A comunicação sem fio é tão antiga quanto a própria comunicação: sinais de fumaça são um exemplo de comunicação sem fio, assim como a própria fala, em que a linguagem exerce a vez de codificação.

Em se tratando de dispositivos eletrônicos, ainda que a informação seja digital, a transmissão ainda ocorre de forma analógica, também dependendo de um meio físico para a propagação da informação, que pode ser o próprio ar, no caso das ondas de radiofrequência. Neste tipo de transmissão, o transmissor emite ondas que são captadas pelo receptor, geralmente mediante a utilização de antenas, mas, para que efetivamente ocorra a comunicação, é necessário que a onda chegue ao receptor com uma intensidade que lhe permita fazer a distinção entre os sinais.

Essa intensidade mínima para compreensão do sinal pode ser entendida como a sensibilidade do dispositivo, ou seja, é o mínimo de potência de sinal para que o receptor seja capaz de decodificar a mensagem após todas as perdas ocorridas e ruídos inseridos no percurso. Desta maneira, contribuem para que seja alcançada a sensibilidade, a potência de transmissão e os ganhos das antenas de transmissão e recepção, enquanto as perdas de percurso, relacionadas com a distância entre os dispositivos, atenuam o sinal.

Portanto, há uma relação entre o alcance e a potência do sinal, assim como entre a potência de transmissão e a energia consumida, sendo necessário avaliar a melhor proporção de acordo com a finalidade da transmissão, como no caso do *Bluetooth Low Energy* (BLE), em que se prioriza o baixo consumo energético em detrimento de outros aspectos, como o alcance, por exemplo.

No caso das comunicações sem fio, devido às dimensões muito baixas ou elevadas e para que os cálculos utilizem operações mais simples, de soma e subtração, costuma-se avaliar as potências, assim como os ganhos e perdas, não em W ou mW, mas em dBm, cuja relação se dá pela equação logarítmica a seguir:

$$P_{dBm} = 10 \cdot \log_{10}(P_{mW})$$



Obviamente, existem diversos outros conceitos bastante importantes relacionados à comunicação sem fio, como largura de banda, códigos corretores de erro, taxa de erro de bit, entre outros, mas tem-se que sua revisão em nada contribui para o trabalho em questão quanto aos objetivos fixados, de modo que não serão ora abordados.

## 2.3 Bluetooth Low Energy

Assim como ocorre com o USB, o Bluetooth é hoje conhecido e empregado de maneira quase universal, alcançando grande relevância desde a sua criação e sendo uma das tecnologias sem fio mais aplicadas nos computadores e seus periféricos desde a década de 2000.

### 2.3.1 História

Com a expansão dos dispositivos móveis na década de 90, as fabricantes passaram a buscar uma forma de substituir os cabos utilizados na comunicação de curta distância, até que a sueca Ericsson, em 1994, criou a tecnologia de transmissão via rádio hoje conhecida como Bluetooth<sup>1</sup>. Com o grande interesse e o sucesso inicial do desenvolvimento, outras companhias, como a Nokia, IBM e Intel, se uniram à Ericsson no chamado *Special Interest Group* (SIG) para desenvolvimento e aperfeiçoamento da tecnologia, que hoje é integrado por dezenas de milhares de companhias (Jaycon Systems, 2017).

Portanto, desde o seu surgimento, a tecnologia Bluetooth está em constante evolução, com o desenvolvimento de novas funcionalidades e aperfeiçoamento das existentes, como a implementação da multi transmissão na versão 2.0 (*multicast*) (ZHANG, 2011), o aumento da taxa de transmissão na versão 3.0 e a redução do consumo energético na versão 4.0, com a implementação do BLE.

**Tabela 2 – Principais avanços na tecnologia Bluetooth**

Versão	Ano	Principais Avanços
1.0	1999	-
1.2	2003	Salto adaptativos de frequência, inquiry-based RSSI
2.0	2004	Velocidade de transmissão de até 2.1 Mbps
2.1	2007	Velocidade de transmissão de até 3.0 Mbps
3.0	2009	Velocidade de transmissão de até 24 Mbps
4.0	2010	Menor consumo de energia, broadcasting, menor latência de conexão
4.1	2013	Melhorias no gerenciamento de energia
4.2	2014	Melhorias de segurança e no tamanho do pacote de dados
5.0	2016	Velocidade de transmissão de até 48 Mbps, aumento da capacidade das mensagens em broadcast e melhorias de confiabilidade da conexão e na eficiência energética

Fonte: Adaptado de (ZEADALLY; SIDDIQUI; BAIG, 2019).

<sup>1</sup> O nome Bluetooth foi dado em homenagem ao rei viking Harald Blåtand Gormsson (SAIRAM; GUNASEKARAN; REDD, 2002), conhecido por unificar os reinos vikings da Dinamarca e Noruega no século X, em razão da proposta da tecnologia, de unificar dispositivos, além, é claro, de ter sido criada por nórdicos

Devido aos aspectos técnicos, a partir da versão 4.0, divide-se a tecnologia Bluetooth em 2 tipos: o Bluetooth Clássico e o Bluetooth Low Energy (BLE)<sup>2</sup>. As versões possuem algumas similaridades entre si, como, por exemplo, a utilização da faixa de frequência dos 2.4GHz e a largura de banda de 80MHz, mas também trazem diversas diferenças, como a divisão da banda pelo BLE em 40 canais de 1MHz, ao invés dos 79 de 1 MHz da versão anterior, além de especificidades de modulação, velocidade, correção de erros e outras apresentadas de maneira sucinta na Figura 7.

**Figura 7 – Comparação entre BLE e Bluetooth (BT) clássico.**

	Bluetooth Low Energy (LE)	Bluetooth Classic
Frequency Band	2.4GHz ISM Band (2.402 – 2.480 GHz Utilized)	2.4GHz ISM Band (2.402 – 2.480 GHz Utilized)
Channels	40 channels with 2 MHz spacing (3 advertising channels/37 data channels)	79 channels with 1 MHz spacing
Channel Usage	Frequency-Hopping Spread Spectrum (FHSS)	Frequency-Hopping Spread Spectrum (FHSS)
Modulation	GFSK	GFSK, $\pi/4$ DQPSK, 8DPSK
Power Consumption	~0.01x to 0.5x of reference (depending on use case)	1 (reference value)
Data Rate	LE 2M PHY: 2 Mb/s LE 1M PHY: 1 Mb/s LE Coded PHY (S=2): 500 Kb/s LE Coded PHY (S=8): 125 Kb/s	EDR PHY (8DPSK): 3 Mb/s EDR PHY ( $\pi/4$ DQPSK): 2 Mb/s BR PHY (GFSK): 1 Mb/s
Max Tx Power*	Class 1: 100 mW (+20 dBm) Class 1.5: 10 mW (+10 dbm) Class 2: 2.5 mW (+4 dBm) Class 3: 1 mW (0 dBm)	Class 1: 100 mW (+20 dBm) Class 2: 2.5 mW (+4 dBm) Class 3: 1 mW (0 dBm)
Network Topologies	Point-to-Point (including piconet) Broadcast Mesh	Point-to-Point (including piconet)

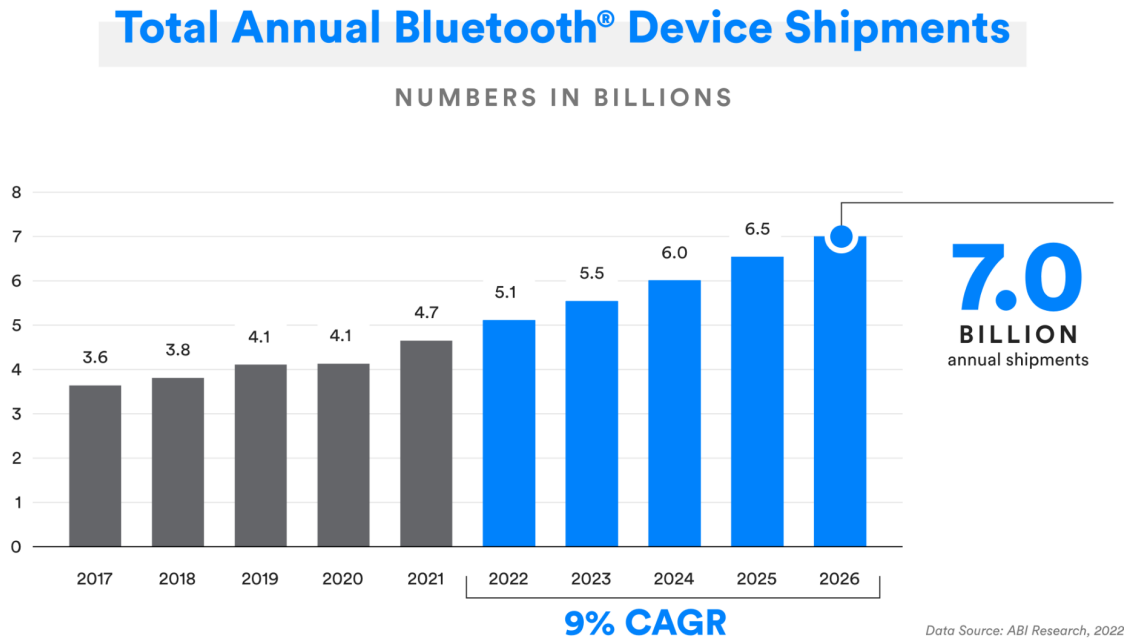
Fonte: (BLUETOOTH, 2021).

### 2.3.2 Visão Geral

A quantidade de dispositivos móveis não para de aumentar, assim como a de conexões sem fio, de modo que as tecnologias Wi-Fi e Bluetooth se tornaram praticamente onipresentes nas comunicações de longa e curta distâncias, respectivamente (HUNG *et al.*, 2017), estimando-se atingir o número de 7 bilhões de dispositivos embarcados em 2026 (Figura 8).

<sup>2</sup> Apesar de ser considerada uma evolução, o BLE (após a versão 4.0) e o Bluetooth Clássico na realidade são tecnologias diferentes, de modo que não são compatíveis entre si e somente estão disponíveis em conjunto nos dispositivos chamados *Dual-mode (Bluetooth Smart Ready)*, que implementam as duas tecnologias e são capazes de se comunicar com qualquer dispositivo Bluetooth.

Figura 8 – Número de dispositivos embarcados anualmente com Bluetooth.



Fonte: (BLUETOOTH, 2022).

Desenvolvido para curtas distâncias e com baixa potência de transmissão, de um modo geral, o Bluetooth, já em sua versão clássica, é considerado uma tecnologia que consome menos energia e que possui uma resistência relativamente alta à interferência na sua transmissão (KAJIKAWA *et al.*, 2016). E, como quase tudo na engenharia, o BLE também é caracterizado por *tradeoffs* em diversos aspectos, como a distância máxima, velocidade de transmissão e consumo de energia, este último sendo o priorizado na tecnologia em questão.

Como o foco do BLE é a economia de energia e uma das principais fontes de consumo energético em um dispositivo de transmissão sem fio é a justamente a transmissão, seu projeto ocorreu com a intenção de manter a transmissão desligada pelo maior tempo possível. Assim, são estabelecidos “intervalos de conexão”, seguidos por períodos de espera, ou ociosidade, (*idle*), transmitindo pequenas quantidades de dados a cada evento com o desligamento do rádio no período entre eles. Desta maneira, no BLE 4.0 entendeu-se que um intervalo que permitisse a economia de energia sem que ocorressem grandes prejuízos à transmissão seria de 7.5ms a 4s. (TOWNSEND *et al.*, 2014)

Portanto, tendo em vista que a transmissão não ocorre em tempo integral, há limitação da quantidade de dados trafegada e, por decorrência, da velocidade da transmissão.

Além da velocidade, o alcance é outro aspecto que foi limitado em função da economia de energia, o que é bastante intuitivo, pois, para se reduzir o consumo de energia, é também necessário reduzir a potência de transmissão, diretamente relacionada ao alcance. A potência de transmissão pode ser definida entre -30 e 0 dBm, sendo possível chegar a uma distância

de até 30 metros de transmissão confiável. Considerando-se que, quanto menor a potência de transmissão – e, portanto, o alcance –, menor é o consumo de energia, o alcance típico é de 2 a 5 metros. (TOWSEND *et al.*, 2014)

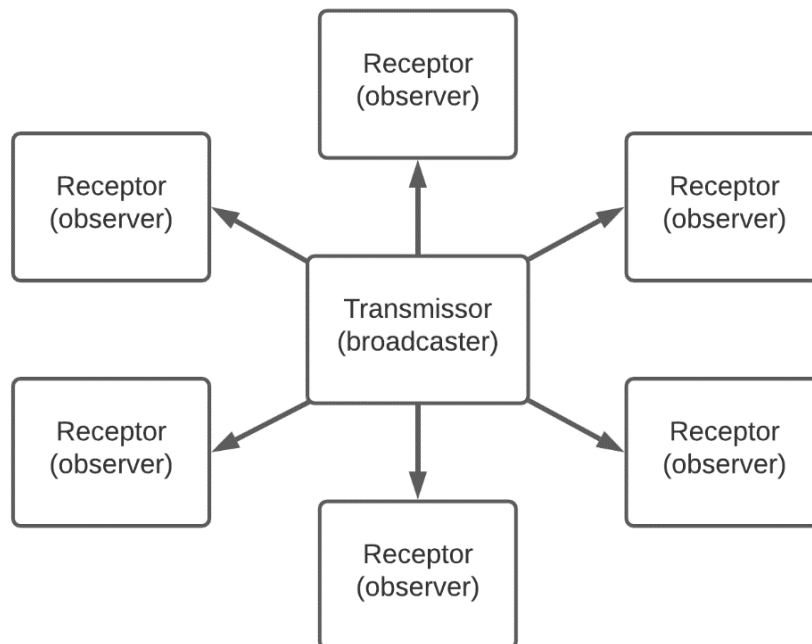
### 2.3.3 Topologia

Existem 2 formas de comunicação de um dispositivo BLE, via *broadcasting* ou conexão, cada qual com suas vantagens e desvantagens. Usando a transmissão sem conexão (*broadcasting*), os dados são enviados de maneira unilateral e podem ser recebidos por todos os dispositivos que estejam no alcance de recepção. Além disso, este tipo de transmissão se caracteriza pelo envio de dados pelo transmissor (*broadcaster*) sem que possua conhecimento do dispositivo receptor (observador) ou de sua recepção.

Portanto, os pacotes são enviados em “via única” e não há a possibilidade de controle da qualidade e tampouco de privacidade, já que, inexistindo conexão, não há ciência do recebimento dos pacotes (*acknowledge*) e tampouco sua restrição.

Este tipo de transmissão é caracterizado por pacotes com 31 bytes de dados (com a possibilidade de um segundo pacote adicional de mesmo tamanho), que incluem os dados de descrição do transmissor, e é indicado para transmissão de pequenas quantidades de dados para múltiplos receptores, mas não para dados sensíveis.

**Figura 9 – Topologia da transmissão via *broadcast*.**



**Fonte: A autoria própria (2022).**

Caso exista a necessidade de transmissão de dados nas duas direções ou de proteção dos dados transmitidos, é indispensável a existência de uma conexão entre os dispositivos, que se caracteriza por uma troca permanente, privada e periódica de dados.

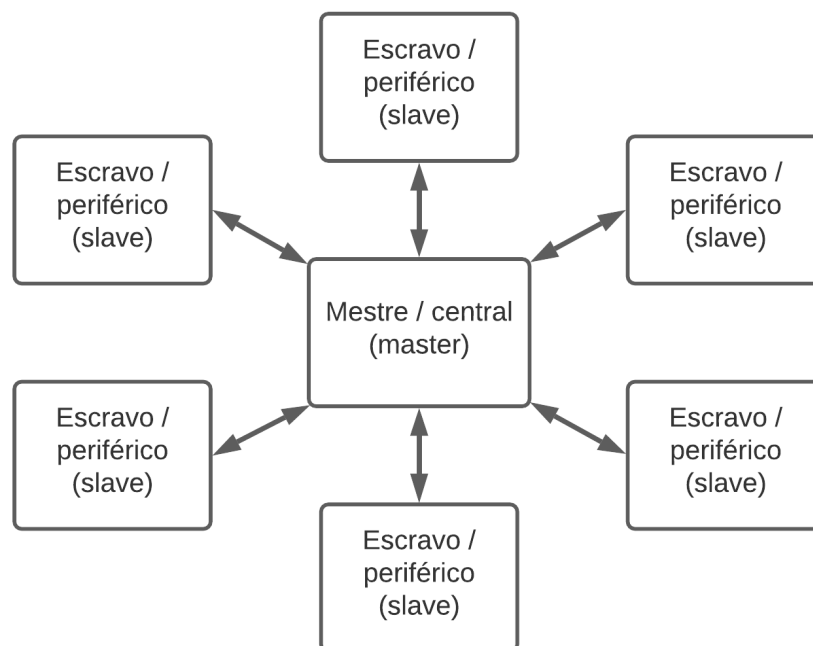
O Bluetooth Clássico também se diferencia do BLE na forma de conexão: enquanto no primeiro os dispositivos desempenham um único papel de cada vez ("descobrimo"ou "permitindo ser descoberto"), o BLE, através do mecanismo chamado "escaneamento passivo"(*passive scan*) possibilita que os dispositivos atuem simultaneamente na "propagação"e no "escaneamento"(KAJIKAWA *et al.*, 2016).

No BLE, a conexão sempre ocorre entre dois dispositivos com papéis distintos: o de mestre (central) e o de escravo (periférico) – nunca entre dois centrais ou entre dois periféricos.

O dispositivo periférico faz a publicidade de sua disponibilidade para os dispositivos próximos, enquanto o central assume o papel de "escaneamento"e busca por dispositivos disponíveis, enviando um pacote de requisição de conexão para tentar estabelecer a conexão (KAJIKAWA *et al.*, 2016). Então, o periférico aceita a conexão requisitada por um dispositivo mestre, seguindo, então as diretrizes de troca de dados deste dispositivo central. A conexão se dá de maneira exclusiva e, assim que é estabelecida, o periférico para de publicar sua disponibilidade, possibilitando o início da transmissão de dados.

Veja-se que, apesar de a conexão entre dois dispositivos ser exclusiva, ambos podem se conectar a outros dispositivos. Ou seja, um dispositivo pode ser central e periférico ao mesmo tempo (em conexões distintas), sendo que um periférico pode se conectar a múltiplas centrais e uma central pode se conectar a vários periféricos, como demonstra a Figura 10.

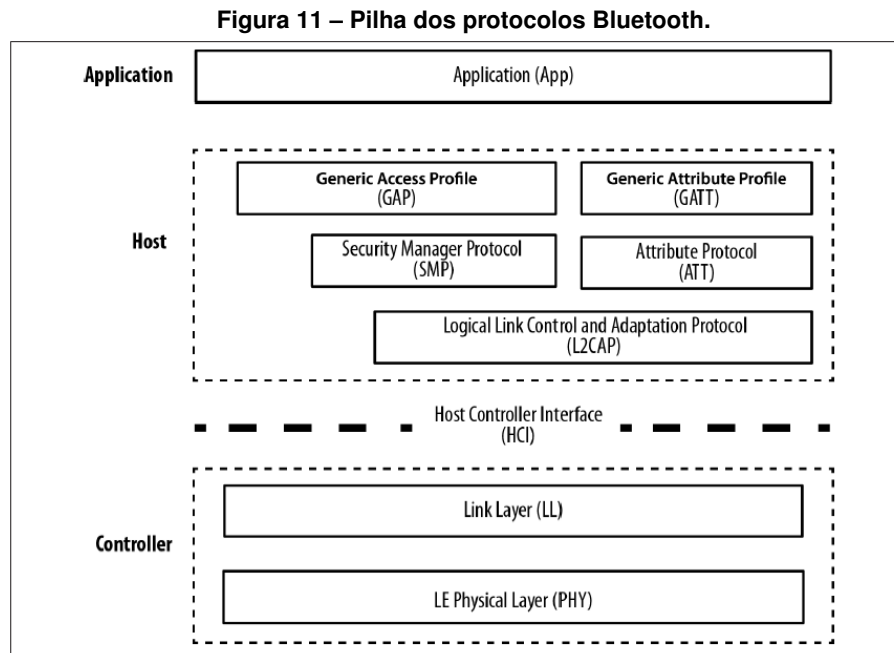
**Figura 10 – Topologia da transmissão via conexão.**



**Fonte: Autoria própria (2022).**

### 2.3.4 Camadas, Protocolos e Perfis

Um dispositivo BLE é dividido em três camadas básicas: controlador, *host* e aplicação – com a qual os usuários geralmente interagem –, cada qual dividida nas camadas que provêm as funcionalidades necessárias à operação, conforme demonstra a Figura 11. Alguns aspectos



Fonte: (TOWNSEND *et al.*, 2014).

bastante importantes relativos ao funcionamento dos dispositivos Bluetooth são o endereço e os pacotes de transmissão. Assim como ocorre com os endereços Ethernet *Media Access Control* (MAC), os endereços Bluetooth são únicos e possuem 48 bits de comprimento. Já os pacotes de dados, antes limitados em 27 bytes de *payload* até a versão 4.0 do Bluetooth, podem chegar até a 251 bytes na versão 5.0 (Silicon Labs, 2022).

Em relação à transmissão de dados, existem quatro formas de um cliente acessar um servidor: leitura, escrita, notificação e indicação. A primeira consiste na leitura da característica do servidor, enquanto a segunda é a modificação desta característica. Já as operações de notificação e indicação são realizadas pelo servidor para "avisar" o cliente da alteração de algum valor de característica, com a última se diferenciando pela necessidade de um reconhecimento (*acknowledgment*) pelo cliente (DIAN; YOUSEFI; LIM, 2018).

Existem, ainda, outros dois conceitos fundamentais a serem compreendidos no BLE, o de protocolo e o de perfil.

Protocolos são as camadas que implementam os formatos de pacote, rotas, multiplexação e codificação de modo a adequar os dispositivos às especificações do Bluetooth, enquanto os perfis são as funcionalidades que determinados modos de operação possuem, podendo ser genéricos ou com finalidades específicas e definindo como os protocolos devem ser utilizados. (TOWNSEND *et al.*, 2014)

Os perfis genéricos são obrigatórios e estabelecem parâmetros fundamentais de controle, no caso do Perfil de Acesso Genérico, em inglês *Generic Access Profile* (GAP), e de dados, no caso do Perfil de Atributo Genérico, em inglês *Generic Attribute Profile* (GATT). O GAP é quem determina aspectos relativos à operação de baixo nível dos dispositivos, como descobrimento, conexão e segurança. Já o GATT pode ser considerado o *backbone* da transmissão de dados via BLE, determinando como os dados são organizados e transferidos entre aplicações (TOWSEND *et al.*, 2014).

Já os perfis de uso específico se diferenciam de acordo com sua finalidade, podendo ser definidos pelos próprios fabricantes de dispositivos, como o *iBeacon*<sup>3</sup>, ou pelo SIG, como é o caso do Dispositivo de Interface Humana (HID, em inglês *Human Interface Device*), utilizado na interação entre dispositivos de uso por humanos, como teclados e mouses.

---

<sup>3</sup> Tecnologia disponibilizada pela fabricante de dispositivos eletrônicos Apple que possibilita aplicações com reconhecimento espacial utilizando *Bluetooth Low Energy*. (APPLE, 2014)

### 3 DESENVOLVIMENTO

Para o desenvolvimento e testes do dispositivo desenvolvido no presente trabalho, utilizou-se um conjunto de teclado e mouse Bluetooth, com *dongle* próprio conectado via USB no ESP32, com a leitura dos dados via USB e sua retransmissão via BLE – através da antena integrada – para dispositivos com Bluetooth integrado, notadamente, notebooks com sistema operacional Windows e *smartphones* com Android.

A escolha das tecnologias empregadas, USB e BLE, deu-se por diversos motivos, como (i) a adequação ao uso pretendido, qual seja a transmissão de dados de teclado e mouse, (ii) a utilização massiva, com ampla disponibilidade de material *open source* e de consulta, (iii) o baixo custo, convergindo com um dos objetivos do projeto, e (iv) a disponibilidade em microcontroladores ordinários e compatibilidade com os computadores, *smartphones* e seus sistemas operacionais.

A idealização do projeto e o seu desenvolvimento foram possíveis graças à existência do protocolo para dispositivos de interface humana, HID, tanto na tecnologia USB quando na Bluetooth, que é a adaptação do anterior (BLUETOOTH, 2015). Desta maneira, utilizando as diretrizes e especificações estabelecidas é possível utilizar o microcontrolador escolhido, com as devidas adaptações, em busca de se atingir o objetivo definido para o presente trabalho.

#### 3.1 Hardware

##### 3.1.1 ESP32

O ESP32 é um microcontrolador desenvolvido pela companhia chinesa, baseada em Shanghai, Espressif Systems, cuja produção em massa foi iniciada a partir de 2016, passando a ser embarcado em placas de desenvolvimento pelas *Original Equipment Manufacturer* (OEM) que podem ser adquiridas por valores bastante acessíveis<sup>1</sup> (KOLBAN, 2018) e que possui diversas versões, cada qual com recursos específicos.

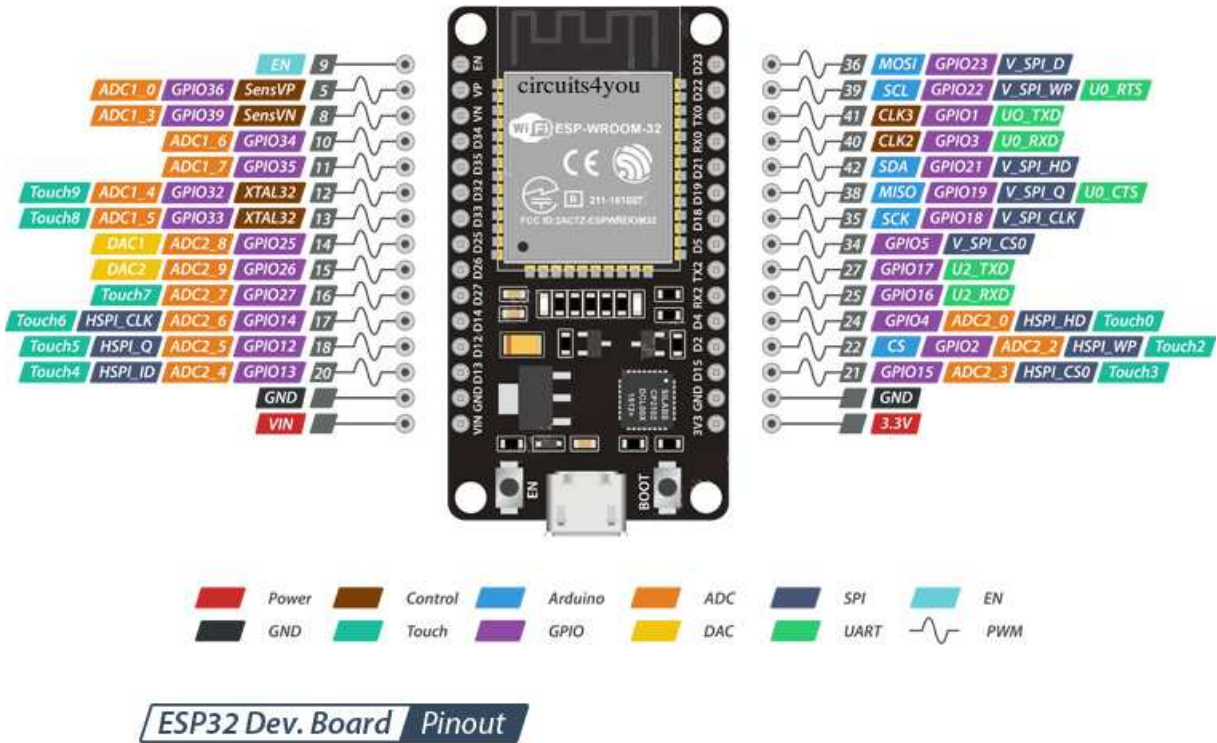
Para o presente trabalho, optou-se pelo kit de desenvolvimento ESP-WROOM-32 DevKit v2, utilizando-se durante a elaboração do trabalho, até o desenvolvimento da Placa de Circuito Impresso (PCI), o referido kit, que traz a disponibilidade de programação via porta micro USB, por meio do chip CP2102, e cujo esquema de pinos é apresentado na figura 12.

O kit utilizado embarca o chip ESP32-D0WDQ6, que conta com 2 microprocessadores Xtensa® 32-bit LX6, memória interna de 448kB de *Read-Only Memory* (ROM) para *boot* e funções internas e 520kB de *Static Random Access Memory* (SRAM) para dados e instruções, além de trazer integrada a antena utilizada na comunicação via Wi-Fi e Bluetooth. O referido microcontrolador foi escolhido por possuir nativamente o Bluetooth 4.2 (BLE), além de ser largamente

<sup>1</sup> Atualmente, podem ser encontradas facilmente online em sites brasileiros por valores que vão de 20 a 100 reais, dependendo da versão.



Figura 12 – Pinagem do kit de desenvolvimento ESP32.



Fonte: Circuits4you.

utilizado, ampliando a disponibilidade de material existente, e possuir baixo custo, contribuindo com os objetivos do trabalho.

Apesar das vantagens suso apresentadas, a versão escolhida do ESP32 não tem a capacidade de desempenhar a função de *host* USB, atuando tipicamente apenas como *device*, o que levou à necessidade de utilizar o módulo adicional apresentado na seção 3.1.2.

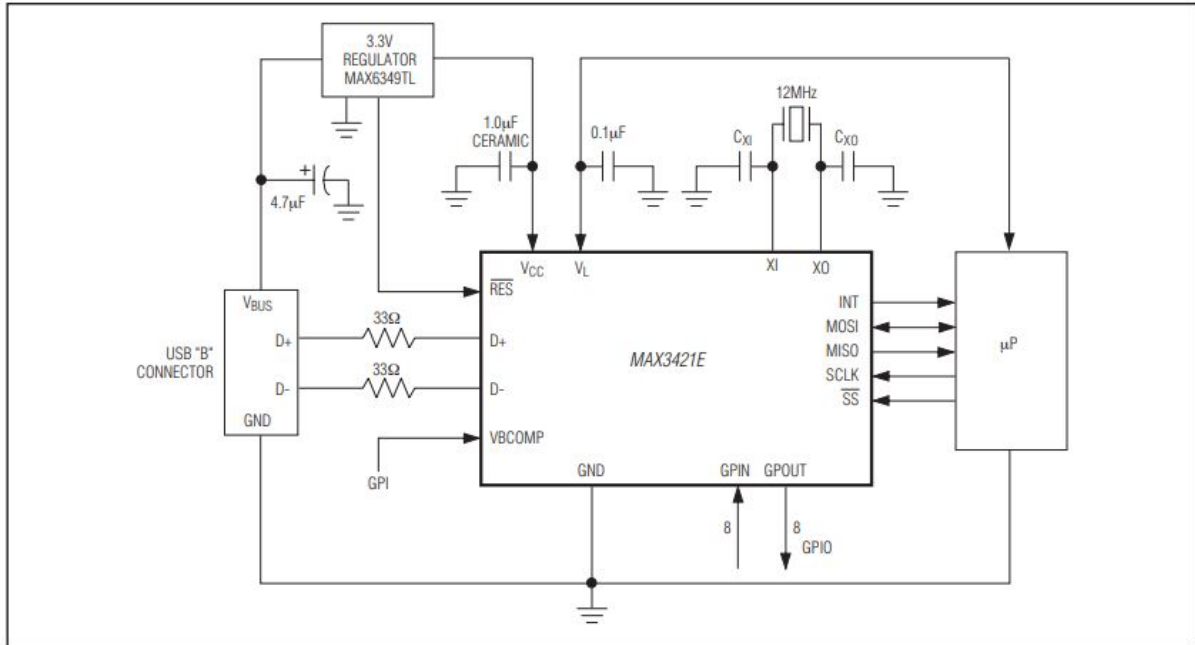
### 3.1.2 USB-Host

O módulo USB SHIELD HOST V2.0 utilizado nesse trabalho permite a comunicação de qualquer dispositivo USB, periféricos como mouse, teclados, câmeras e outros, com um microcontrolador.

O módulo se comunica com USB *Client* (*dongle*) através do protocolo USB e para o USB *Server* (microcontrolador) com o protocolo *Serial Peripheral Interface* (SPI) simples com 4 vias. O controlador do módulo, MAX3421E, opera até 26MHZ (MAXIM INTEGRATED, 2013) compatíveis com as especificações do USB 2.0 (Periférico de 12 Mbps de velocidade total, host de 12 Mbps/1,5 Mbps de velocidade total/baixa), possibilitando que o desenvolvedor tenha a liberdade de escolher qual frequência deseja trabalhar. Através dele um mecanismo de interface serial interno, *Serial Interface Engine* (SIE), tratam-se os dados do protocolo USB de baixo nível: tentativas de barramento e verificação de erros. Também constam no *shield* um oscilador

de 12MHz, um conector USB fêmea e um regulador de tensão para 3,3V, cujo esquema é apresentado na Figura 13.

**Figura 13 – Esquemático USB SHIELD HOST.**



**Fonte: (MAXIM INTEGRATED, 2013).**

Para a conexão com o microcontrolador são necessárias 4 saídas do SPI (SCLK, MISO, MOSI e SS) e um canal de interrupção (INT), além de um reset (RST).

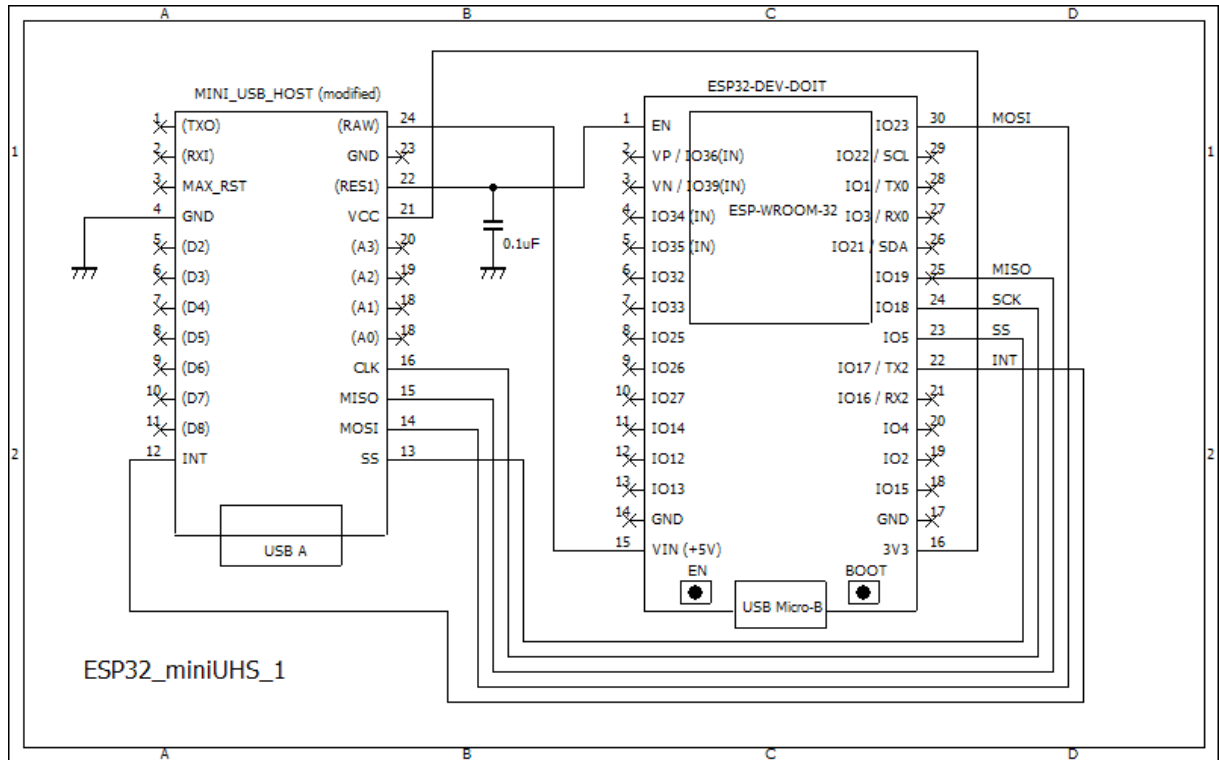
A SPI do módulo só estará ativa se o pino SS estiver em nível baixo. O terminal RST é o responsável pelo *reset* do módulo. Quando ele é conectado ao GND (terra), tendo a largura de pulso maior que 200ns, o chip MAX3421 define todos os seus registradores para o estado padrão, e sua fila é apagada, reiniciando o shield. O terminal INT se torna responsável pelos eventos de entrada de novos dispositivos no módulo, sinalizando o SPI, requerendo sua atenção. Este, quando em nível lógico alto, ativa os registradores de entrada para a comunicação com o protocolo USB. Para torna-lo nível baixo, é necessário um reset no terminal RST ou quando o *Dongle* é retirado da porta USB contida no módulo, ativando os sinalizadores de interrupção.

Os pinos MOSI (*master-out, slave-in*) e (MISO, *master-in, slave-out*) funcionam de maneiras síncrona dependendo da configuração do protocolo SPI. No Full duplex, utilizam uma arquitetura mestre-escravo e esses pinos são separados, porém o envio e recebimento ocorrem simultaneamente. O pino SCLK, *serial clock*, responsável pela sincronização dos pulsos de dados de entrada e saída para o módulo, fornecem o sinal cronometrado para a SPI do chip. Nas bordas de descida ocorrem a escrita os dados para a saída *Master In. Slave Out* (MISO), os dados do escravo são enviados para o mestre, e nas bordas de subida ocorre a leitura de entrada dos dados *Master Out. Slave In* (MOSI), dados do mestre são enviados para o escravo.

### 3.1.3 PCB

O desenvolvimento do dispositivo foi iniciado na protoboard, com os módulos sendo testados de forma unitária, afim de entender melhor seu funcionamento. Utilizando o Kit de desenvolvimento da Espressif com o ESP32, módulo *shield host* USB e botões, no esquemático abaixo mostra as ligações entre eles. Figura 14.

Figura 14 – Esquemático simplificado.



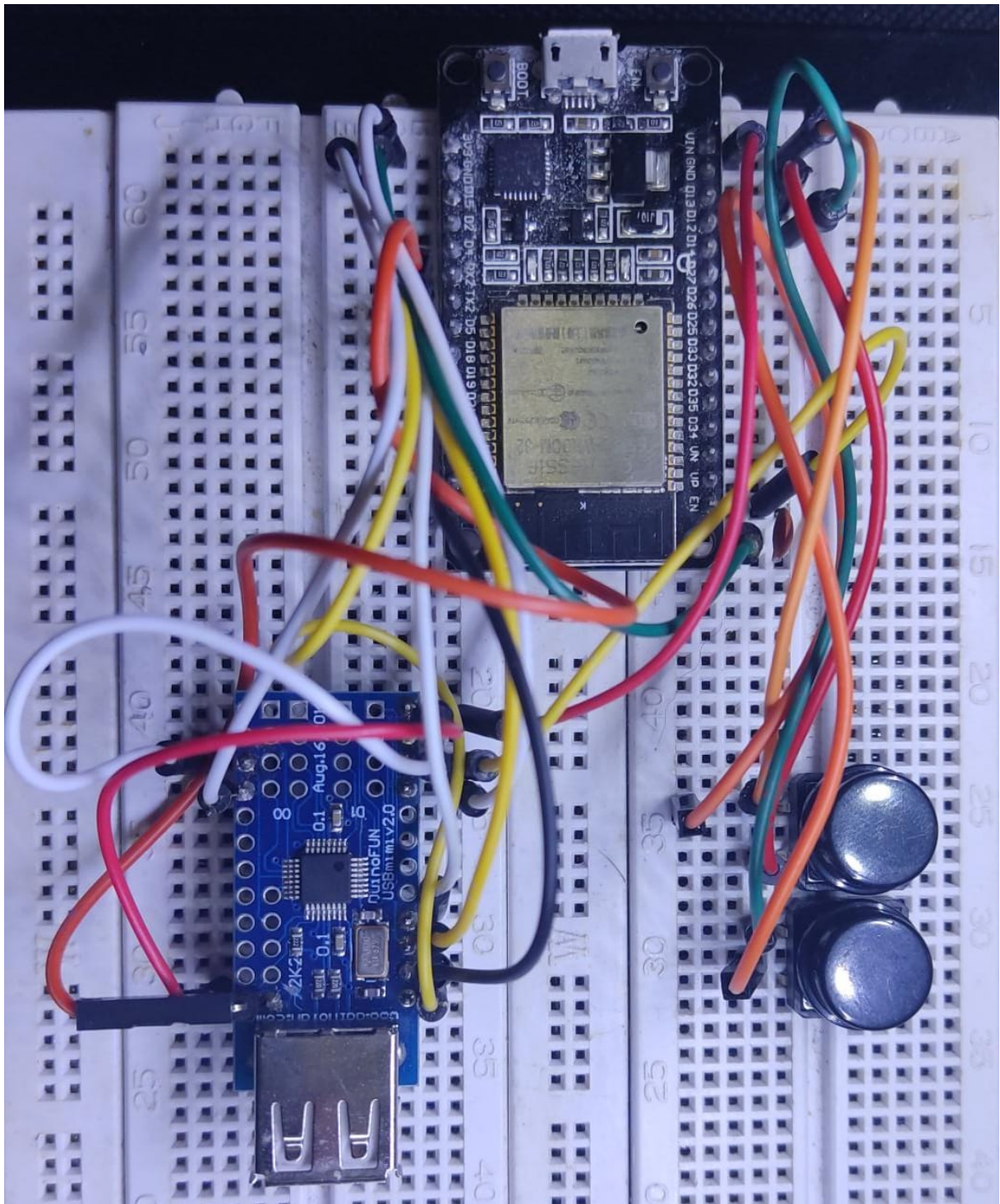
Fonte: (okiraku-camera, 2018).

A figura15 demonstra os módulos devidamente interligados através de *jumpers* na protoboard e os botões utilizados para a entrada de sinal nas portas *General Purpose Input/Output* (GPIO) do ESP-32 para o chaveamento entre *devices* e a reinicialização geral do protótipo.

Como o *shield* USB *host* adquirido apresenta um erro crônico de alimentação em seu layout, foi necessário a raspagem de uma das trilhas, retirando a alimentação de 5V, conforme demonstrado na Figura 16, para que a alimentação não exceda a  $V_{in}$  máx do chip MAX3421(3,3v). Outra modificação realizada foi interligar o RESET do *shield* com o MAX RESET, pino RST do chip, para que o mesmo também seja resetado quando solicitado.

Após os testes, foi desenvolvida a *Printed Circuit Board* (PCB). Com o propósito de alcançar um nível de precisão no desenvolvimento do dispositivo, através do site EasyEDA.com, o projeto com o esquemático em mãos e a prototipagem pela protoboard, foi desenvolvido o esquemático final para o chaveador e produzido pela empresa JLC PCB. Em sua versão final, a placa conta com o microcontrolador ESP-32 da fabricante Espressif, 3 botões para acionamento das portas utilizadas, um regulador de 5V para 3,3V, uma porta USB-B micro para alimentação

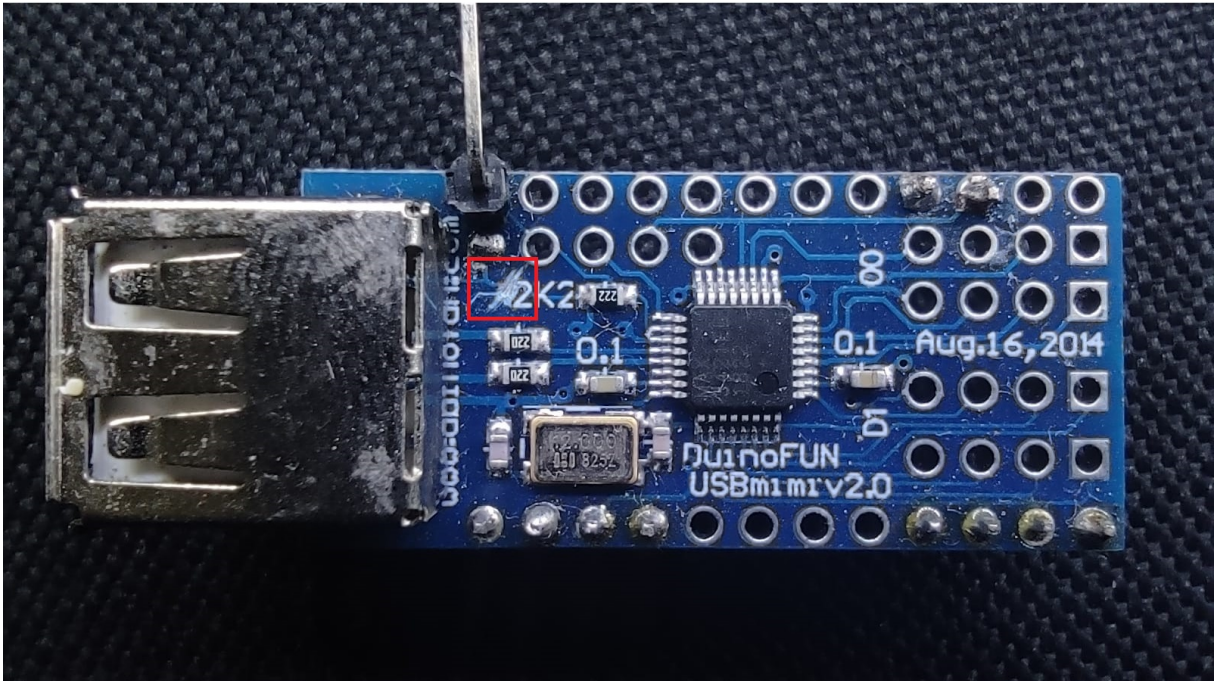
Figura 15 – Protótipo na protoboard.



Fonte: Autoria própria (2022).

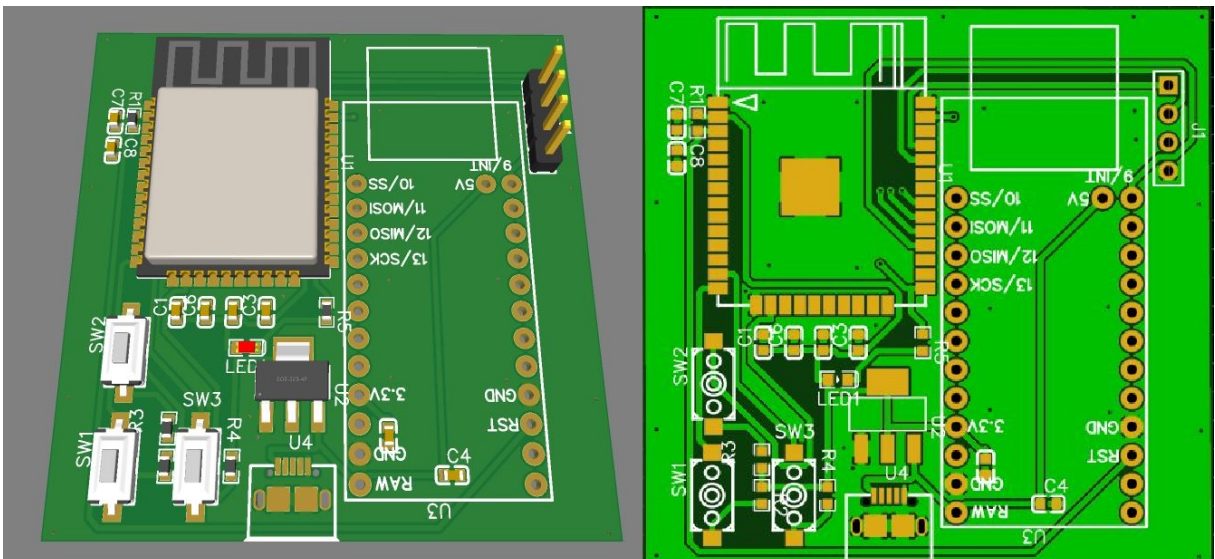
do circuito, barra de pinos para a programação do microcontrolador e um slot para o USB HOST SHIELD. A Figura 17 mostra o layout final da placa de circuito impresso e a simulação em 3D, enquanto a 18 traz a placa já montada.

Figura 16 – Trilha de Alimentação.



Fonte: Autoria própria (2022).

Figura 17 – Placa de circuito impresso - projeto.

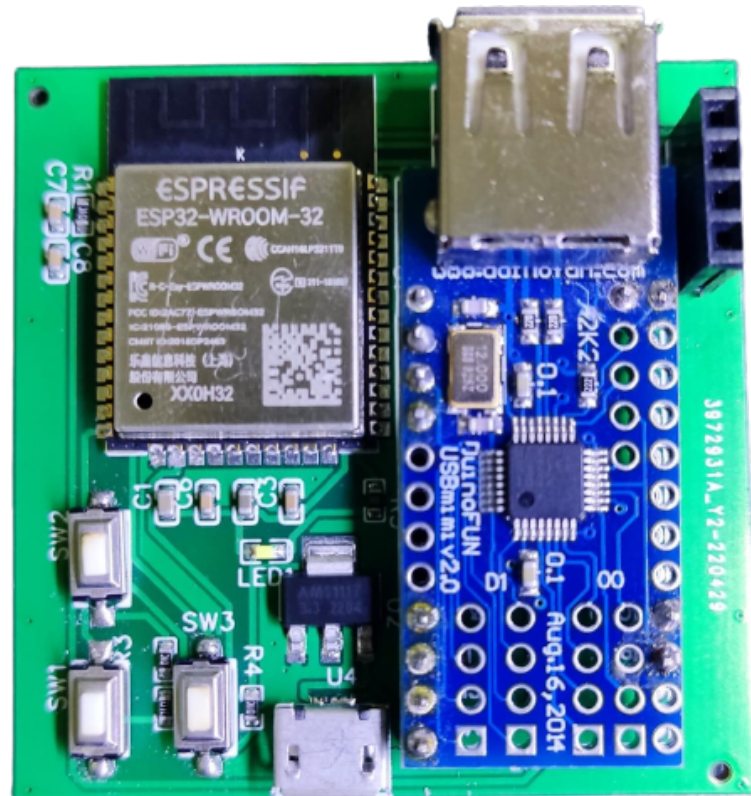


Fonte: Autoria própria (2022).

### 3.2 Firmware

O ESP32 é um microcontrolador bastante popular, que conta com uma pluralidade de ambientes disponíveis para sua programação, como o Espressif IDF, Micropython, LUA e a Arduino IDE, a escolhida para o presente trabalho devido principalmente à sua popularidade, com vasta disponibilidade de fontes de informação, e à facilidade de sua utilização, em especial pela

Figura 18 – Placa de circuito impresso montada.



Fonte: Autoria própria (2022).

simplicidade da interface. Além disso, para a manipulação das classes e bibliotecas empregadas no trabalho, foi utilizado o VS Code e todo o desenvolvimento e a maior parte dos testes se deu no sistema operacional Windows, nas versões 8.1 e 10.

Uma vez definido e configurado o ambiente, passou-se ao efetivo desenvolvimento do firmware, sendo que conforme exposto anteriormente, o presente trabalho foi desenvolvido com a utilização conjunta das tecnologias USB e BLE, o que traz a necessidade de implementar suas operações e realizar os ajustes necessários de modo a compatibilizá-las.

Na seção 2.1, apresentou-se o modo de funcionamento da comunicação USB, com ênfase para os dispositivos de interface humana (HID), que utilizam transferências de interrupção para a transmissão de dados em relatórios (*reports*). Por sua vez, no BLE, a transmissão dos dados de teclado e mouse é realizada por meio de operações de notificação, nas quais, até a versão 4.0 do Bluetooth, um pacote pode conter até 20 bytes de dados (DIAN; YOUSEFI; LIM, 2018), mostrando-se, portanto, adequado à finalidade pretendida.

Para estabelecimento da comunicação USB entre os periféricos e o ESP32, além das necessidades de hardware apresentadas na seção 3.1.2, há também a obrigatoriedade de implementação do software para que o microcontrolador possa desempenhar a função de *host* e, assim, realizar a leitura dos dados produzidos nos periféricos conectados.

Como a elaboração dos descritores, a formatação de pacotes e a definição dos parâmetros e funções dos dispositivos USB e BLE são bastante complexas, demandando um profundo

conhecimento não apenas de programação de microcontroladores, mas das próprias especificações e diretrizes das tecnologias, além de incontáveis horas de desenvolvimento, após uma extensa busca por fontes de informação, encontrou-se apoio em duas bibliotecas adicionais disponibilizadas livremente pelos desenvolvedores: a USB Host Shield Library 2.0<sup>2</sup>, desenvolvida por Oleg Mazurov e mantida por Kristian Sloth Lauszus, e a BLEcombo<sup>3</sup>, elaborada e distribuída por Dean Blackketter. Como os nomes já indicam, a primeira é utilizada para realizar a leitura USB dos periféricos, enquanto a segunda possibilita a emulação de teclado e mouse via BLE.

Visando a atingir o objetivo definido, foi realizada a divisão do projeto em 6 etapas principais, as quais são detalhadas a seguir: transmissão via BLE, leitura via USB, compatibilização entre leitura e transmissão, transmissão múltipla, chaveamento e refinamento.

### 3.2.1 Transmissão via BLE

Após a pesquisa e definição do hardware a ser utilizado para a transmissão sem fio de comandos de mouse e teclado, bem como sua aquisição, a primeira etapa consistiu em configurar esta transmissão de acordo com as premissas pré-estabelecidas.

Tendo-se optado pela transmissão via BLE, devido às suas vantagens já abordadas anteriormente, os desafios passaram a ser: estabelecer a conexão BLE entre ESP32 e computador, fazer o computador reconhecer o ESP32 como teclado e mouse e transmitir efetivamente entradas de teclado e mouse.

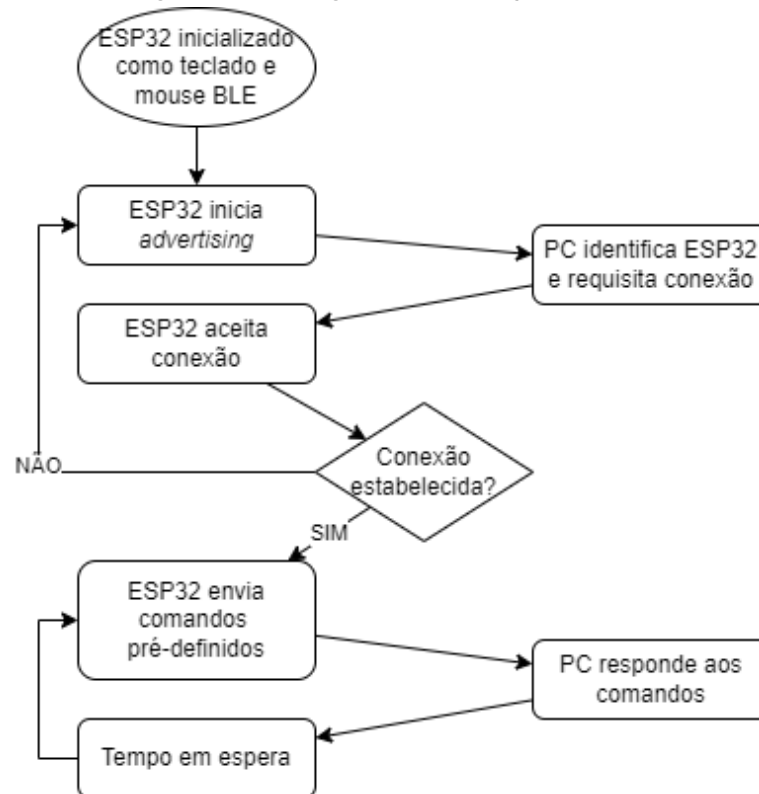
Como a versão do ESP32 escolhida possui nativamente a funcionalidade de BLE, incluindo a antena para transmissão, e as bibliotecas padrão já possibilitam o estabelecimento da conexão e trazem exemplos deste tipo de conexão, a primeira parte desta etapa não trouxe muitos obstáculos, tendo sido realizada de maneira relativamente fácil, mas ainda sem o reconhecimento como HID e tampouco a transmissão de dados. Neste procedimento, o primeiro passo é a inicialização do ESP32 na função de servidor, com a definição das características e descritores, seguido pelo início da propaganda de disponibilidade para conexão (*advertising*). Uma vez que o computador no qual se pretende conectar o ESP32 identifica sua disponibilidade, pode-se requisitar a conexão, que, neste momento inicial, uma vez estabelecida, encerra o *advertising* e possibilita o envio de dados na forma de notificações.

Estabelecida a conexão BLE, é necessário que o computador reconheça o ESP32 como HID e passe a realizar a sondagem periódica de dados disponíveis (*polling*). Neste passo, a biblioteca BLEcombo anteriormente citada vem ao auxílio, trazendo, além da definição dos parâmetros para que o computador reconheça o microcontrolador como teclado e mouse, também o passo seguinte, qual seja a possibilidade de enviar dados de entrada destes periféricos através das funções definidas. A própria biblioteca traz exemplos de emulação de comandos que funcionaram perfeitamente desde a sua implementação.

<sup>2</sup> Disponível em: <https://www.arduino.cc/reference/en/libraries/usb-host-shield-library-2.0/>.

<sup>3</sup> Disponível em: <https://github.com/blackketter/ESP32-BLE-Combo>.

Figura 19 – Fluxograma da emulação HID BLE.



Fonte: Autoria própria (2022).

Concluída esta primeira etapa, em razão do sucesso da transmissão BLE de comandos de teclado pré-definidos via código, pôde-se dar início ao desenvolvimento da leitura dos dados via USB.

### 3.2.2 Leitura USB

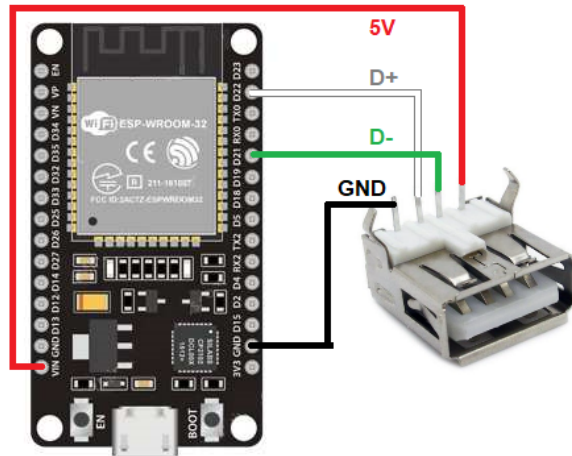
Considerando que o USB consiste na transferência serial de informação, para implementação da leitura dos dados de teclado e mouse, inicialmente se tentou utilizar um conector USB ligado a portas GPIO do microcontrolador, conforme mostrado na Figura 20, apoiando-se no trabalho prévio disponibilizado por Samsonov Dima <sup>4</sup>.

Oteve-se, então, relativo sucesso, uma vez que os dispositivos *Low-Speed* puderam ser lidos. Entretanto, a solução não era adequada aos dispositivos *Full-Speed*, o que limitaria o uso do dispositivo ora desenvolvido aos mouses e teclados mais simples ou antigos e não seria possível realizar a utilização dos conjuntos de teclado e mouse Bluetooth. Portanto, diante de tais restrições, abandonou-se essa primeira ideia, com a busca por uma alternativa mais adequada aos propósitos do trabalho, que passou pela necessidade de aquisição de módulo adicional para que o microcontrolador utilizado pudesse desempenhar a função de *host* USB – o *shield* descrito na seção 3.1.2, com a utilização do MAX3421E.

<sup>4</sup> Disponível em: [https://github.com/sdima1357/esp32\\_usb\\_soft\\_host](https://github.com/sdima1357/esp32_usb_soft_host)



Figura 20 – Esquemático da ligação USB nas portas do ESP32.

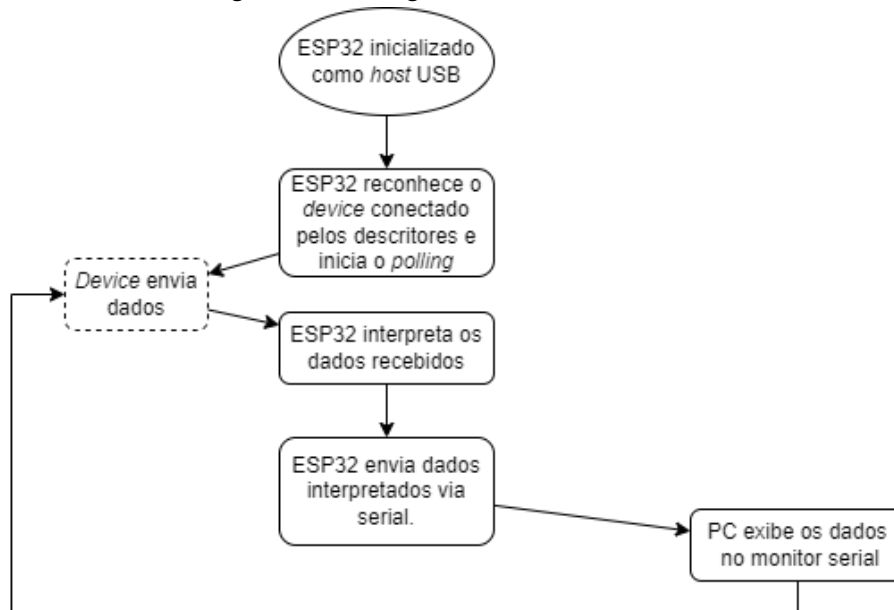


Fonte: Autoria própria (2022).

A partir da utilização do referido módulo, passou-se a utilizar a já mencionada biblioteca USB Host Shield Library 2.0, que já traz grande parte das definições e parâmetros utilizados na comunicação USB como *host*, inclusive daqueles necessários à comunicação com HID.

Para a leitura das entradas e teclado e mouse, além da especificação dos parâmetros de funcionamento, como a definição das funções para enumeração e para recuperar os relatórios, também é necessário definir classes para a interpretação dos pacotes enviados pelo mouse e teclado, bem como as funções de tratamento dos valores recebidos. Neste momento, os dados coletados via USB são interpretados e exibidos no monitor serial da IDE, sem qualquer ação ou resposta do Computador Pessoal (do inglês Personal Computer) (PC).

Figura 21 – Fluxograma da leitura HID USB.



Fonte: Autoria própria (2022).

Desta maneira, foi possível implementar a leitura de mouse e teclado via USB tanto para dispositivos *Low-Speed* como *Full Speed*, incluindo aqueles que realizam a transmissão via Bluetooth, utilizando um receptor *dongle* USB, completando-se mais uma etapa do desenvolvimento.

### 3.2.3 Compatibilização de leitura e transmissão

Após a conclusão com sucesso das etapas anteriores, tendo-se implementado as comunicações USB e BLE para HID, inicia-se a conjugação de ambos os desenvolvimentos, ou seja, estruturar os códigos elaborados e as bibliotecas utilizadas de maneira a harmonizá-los e permitir que se tenha o fluxo completo, com a leitura de dados de teclado e mouse via USB e sua transmissão BLE para o computador, com a efetiva ação que é esperada.

Em um primeiro momento, houve certa dificuldade para a combinação das duas ações, uma vez que ambas são contínuas e complexas e, do modo que estavam sendo arranjadas, interferiam uma na outra. Contudo, após uma série de ajustes, com a alteração de funções e correta ordenação dos processos, os conflitos foram sanados e a união dos dois desenvolvimentos anteriores foi possível, atingindo-se o objetivo pretendido para esta etapa.

A estratégia utilizada e o seu resultado podem ser compreendidos na Figura 22.

### 3.2.4 Transmissão múltipla

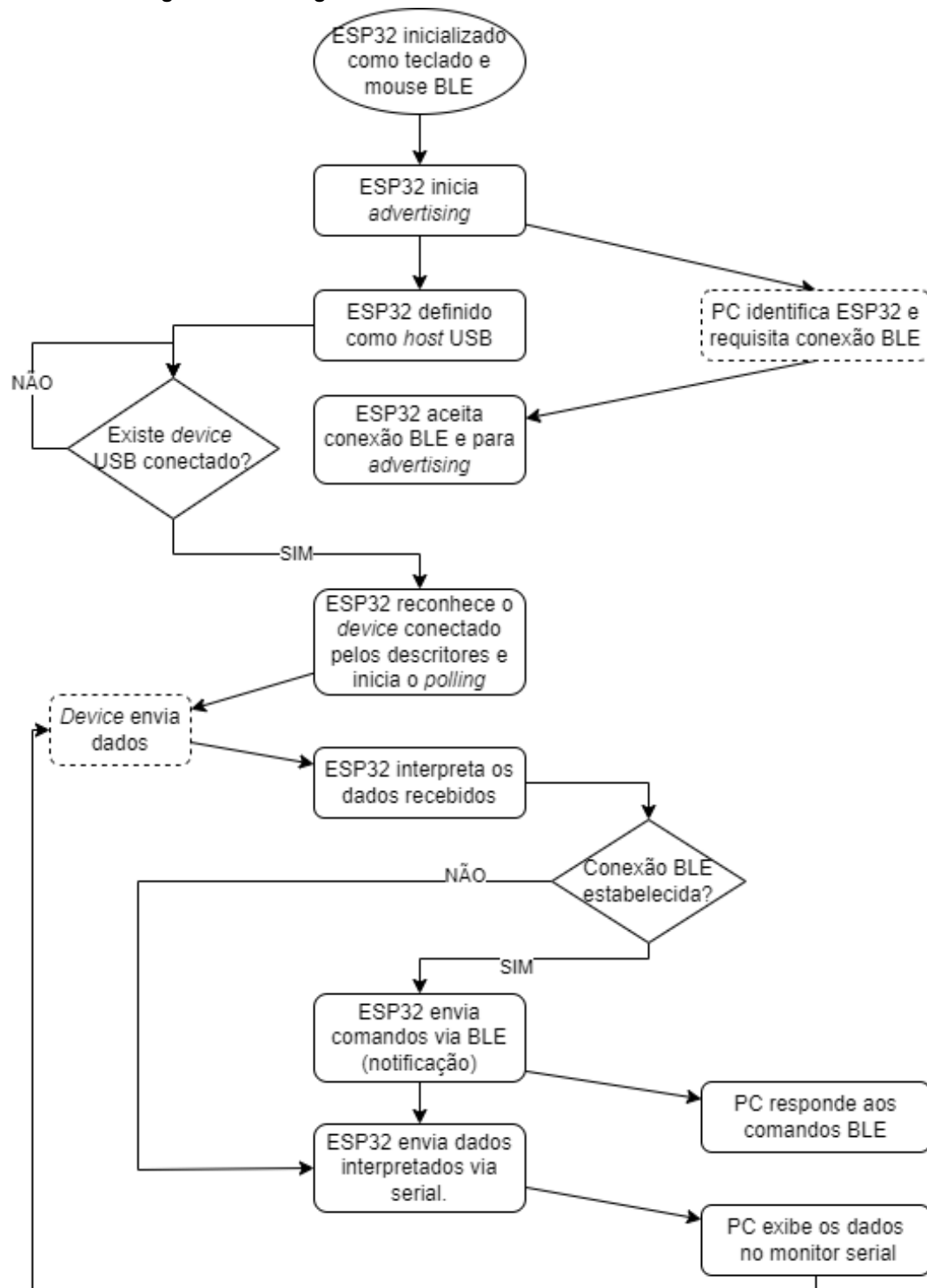
Com a conclusão da etapa anterior, é razoável dizer que foi construído um conversor USB-BLE para teclado e mouse indispensável ao desenvolvimento do dispositivo pretendido com este trabalho. Passa-se, então, à implementação da lógica para chaveamento do receptor, iniciando-se pela estruturação da transmissão dos dados a mais de um dispositivo.

Tendo em vista a capacidade do BLE de estabelecer múltiplas conexões simultâneas, entendeu-se que a conexão do ESP32 com mais de um computador de forma simultânea contribuiria para o objetivo pretendido e reduziria o *delay* no chaveamento do dispositivo receptor, pois não seria necessário realizar um novo procedimento de conexão a cada chaveamento.

As bibliotecas utilizadas para gerenciamento do BLE já são compatíveis com múltiplas conexões simultâneas, de modo que bastou inserir definição da quantidade limite de conexões simultâneas e adaptar o código para o ESP32 não cessar o *advertising* enquanto não fosse atingido esse limite, isto é, o ESP32 passou a reiniciar o *advertising* após cada conexão até a quantidade de dispositivos conectados igualar o máximo pré-estabelecido.

Neste momento, o dispositivo elaborado no presente trabalho é capaz de realizar a conversão USB-BLE das entradas de teclado e mouse com a transmissão simultânea a diversos computadores (ou equipamentos similares, como smartphones).

Figura 22 – Fluxograma da leitura HID USB e sua transmissão BLE.



Fonte: Autoria própria (2022).

### 3.2.5 Chaveamento

A partir da concretização do envio dos dados para mais de um computador simultaneamente, para atingir o objetivo estabelecido para o presente trabalho, faz-se necessário definir a forma de escolha do receptor, com a implementação da lógica correspondente.

Como já mencionado anteriormente, o processo de desconexão e reconexão Bluetooth resulta em um tempo muito alto para o chaveamento, o que minaria a agilidade pretendida com o dispositivo projetado – provavelmente seria mais rápido realizar a troca dos conectores físicos. Desta maneira, optou-se por um chaveamento via software, realizando-se o envio dos dados apenas para o endereço BLE correspondente ao dispositivo selecionado, mantendo-se a escolha física do receptor pelo usuário, por meio do acionamento do botão projetado para esta finalidade.

Para tornar possível a implementação da forma como definida e em conformidade com os objetivos do projeto, mantendo-se também a simplicidade no uso do chaveador, limitou-se a quantidade de receptores a dois, bastando o pressionamento de um botão para alternar rapidamente entre eles. Com isso, o usuário opta pela posição 1 ou 2 e informa a mudança ao microcontrolador através do botão.

De modo que somente o computador correspondente à posição selecionada pelo usuário receba a informação, faz-se a comparação da seleção com a posição na lista de dispositivos conectados via Bluetooth, e promove-se o envio da notificação apenas em caso de equivalência.

Visando a manter a simplicidade e a versatilidade do dispositivo projetado, definiu-se que as posições na lista de dispositivos conectados são ocupadas de acordo com a ordem de "chegada", ou seja, o primeiro dispositivo a conectar via Bluetooth ocupa a primeira posição. Isto permite que o chaveador seja utilizado em mais de dois dispositivos sem perder a facilidade de comutação, além de dispensar a necessidade de LEDs ou outra forma de indicador de seleção atual.

Além das já mencionadas acima, algumas outras escolhas de projeto foram necessárias para o correto funcionamento do chaveador, as quais são descritas a seguir.

Neste sentido, definiu-se que, em caso de desconexão de algum receptor, mantém-se a posição do outro até ocorrer também a sua desconexão ou reset do chaveador.

Para evitar conflitos ou desconexões involuntárias, o chaveador só inicia o *advertising* caso não exista nenhum dispositivo conectado na posição correspondente à seleção – por exemplo, caso exista um dispositivo conectado na posição 1 e a posição 2 esteja livre, o chaveador só realizará a propaganda para novas conexões Bluetooth se a seleção do botão corresponder à 2.

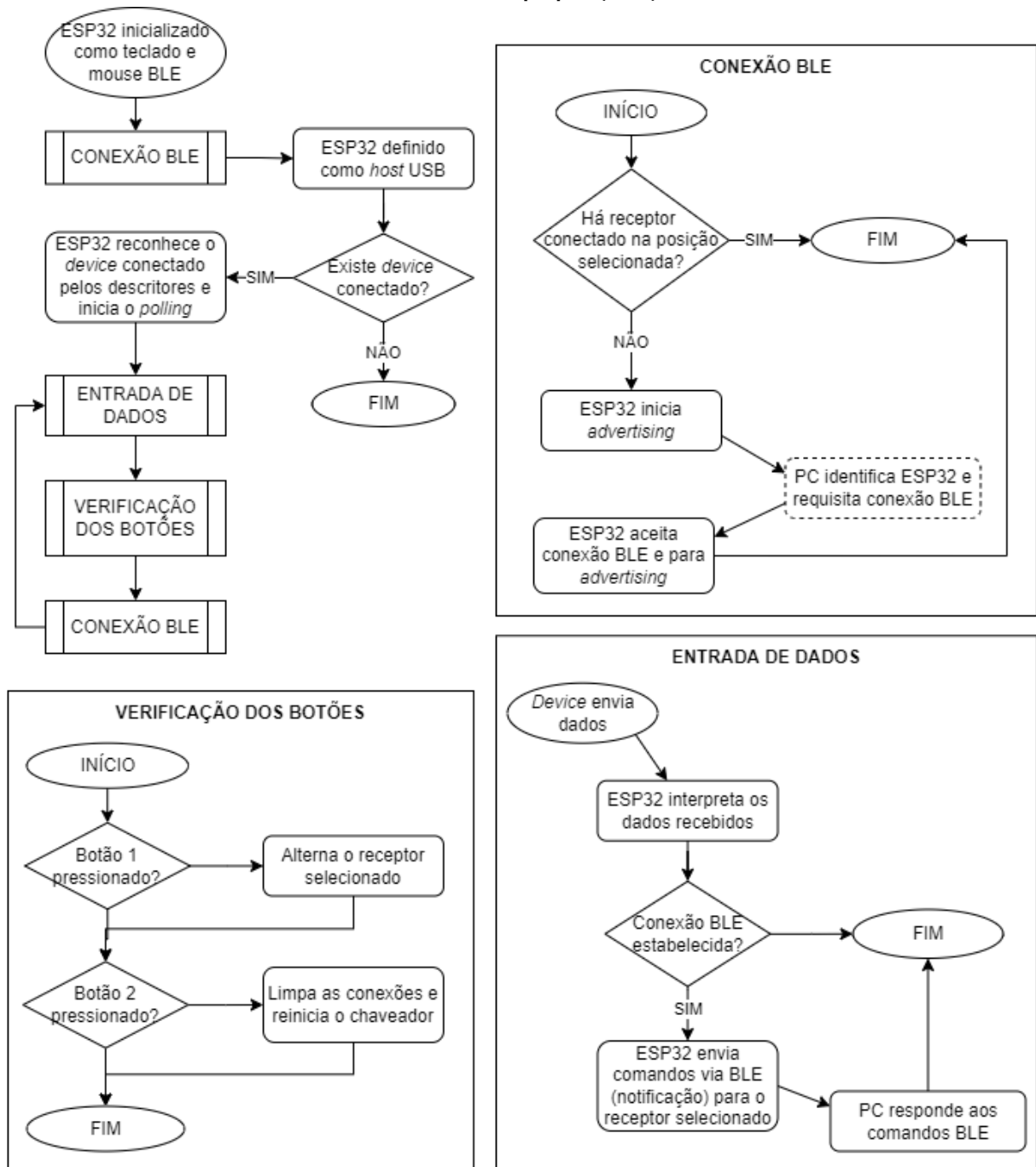
Por fim, foi também implementado um botão de reset, o qual, quando pressionado, promove a remoção dos endereços Bluetooth armazenados no ESP32 a fim de permitir novas conexões e também de limpar eventuais erros.

### 3.2.6 Refinamento

Após a finalização do chaveador, com o funcionamento ocorrendo de acordo com o projeto, foi realizada a revisão do código-fonte, promovendo-se a sua organização e limpeza, com a retirada de trechos desnecessários e adoção de boas práticas a fim de facilitar sua

Figura 23 – Fluxograma do chaveamento BLE.

Fonte: Autoria própria (2022).



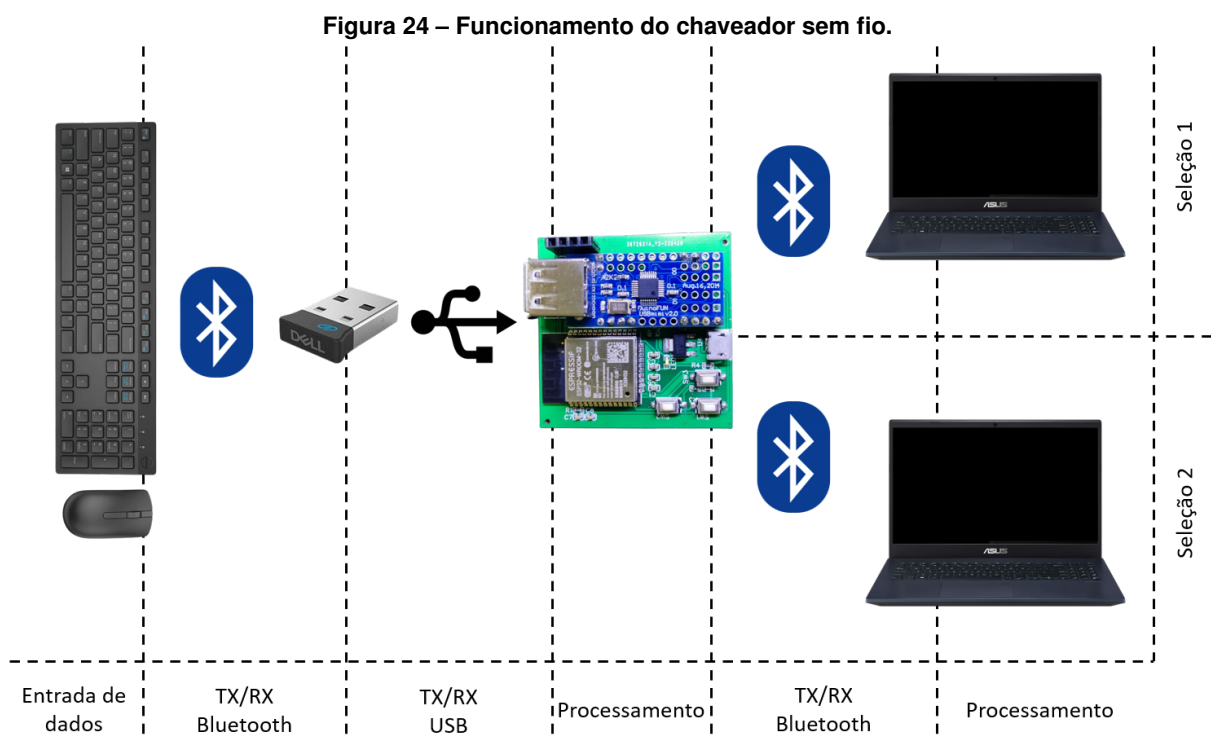
compreensão e manutenção. Também em razão dos constantes testes realizados, esta última etapa serviu para corrigir alguns *bugs* encontrados e realizar outros pequenos ajustes.

Nesta etapa, as alterações se deram de forma mais pontual, para correção interna dos itens já definidos do processo, de forma que não há alteração do fluxograma macro já apresentado na Figura 23.

### 3.3 Embarcamento

Com a montagem da PCB e seus componentes e a finalização do *firmware*, é possível realizar a programação via serial na placa desenvolvida, para que seja atingido o objetivo estabelecido quando da definição do escopo do trabalho: a montagem de um dispositivo de baixo custo para chaveamento sem fio entre dois computadores de periféricos de teclado e mouse.

Desta maneira, o chaveador projetado é capaz de realizar a leitura via USB dos dados inseridos em um conjunto de teclado e mouse, processá-los e enviá-los via Bluetooth para dois computadores distintos, um de cada vez, de acordo com a seleção realizada via botão, conforme ilustrado na Figura 24.



Fonte: Autoria própria (2022).

## 4 TESTES E RESULTADOS

### 4.1 Ambiente de testes

O desenvolvimento do trabalho ocorreu de forma simultânea à prototipagem, tendo sido, portanto, acompanhado por constantes testes, de modo que cada etapa fora exaustivamente testada antes de ser dada como concluída a fim de garantir a solidez do resultado final e permitir as adaptações necessárias durante o curso. Tendo o projeto sido voltado à simplificação de atividades ordinárias de grande parte das pessoas, especialmente durante a pandemia do Covid-19 e incluindo os próprios desenvolvedores, o cenário para o qual o dispositivo foi concebido pôde ser facilmente emulado, bastando a utilização dos aparatos já disponíveis nas bancadas dos desenvolvedores: conjuntos de teclado e mouse sem fio e computadores, além de softwares comuns, como editores de texto.

Durante a elaboração do projeto e para a realização dos testes de funcionamento, foram utilizados como periféricos os conjuntos de teclado e mouse sem fio Dell KM636 e Microsoft-Wireless Desktop 3050, mouse genérico com fio e mouse Logitech sem fio M720.

**Figura 25 – Periféricos utilizados nos testes e desenvolvimento.**



**Fonte: Autoria própria (2022).**

Os dados dos periféricos citados foram enviados, através do uso do chaveador projetado, a notebooks com Bluetooth genérico embarcado e sistemas operacionais Windows 10 e 8.1 e também a smartphones Samsung Galaxy S10 e Xiaomi Poco X3 Pro, ambos com SO Android 12.

**Figura 26 – Informações dos computadores utilizados nos testes e desenvolvimento.**

The image shows two screenshots. The top one is from an Android phone, displaying 'Informações do software' with details like 'Versão do One UI 4.1', 'Versão Android 12', and 'Atualização do sistema Google Play 1 de abril de 2022'. The bottom screenshot is from a Windows PC, showing 'Sobre' with device specifications (DNS-ASUS, Intel Core i5-8250U, 8GB RAM) and Windows OS details (Windows 10 Home Single Language, 21H2, installed 18/4/21).

Nº de telefone	Desconhecido
Nome do modelo	Galaxy S10
Número do modelo	SM-G973F
Número de série	[Redacted]
IMEI (slot 1)	[Redacted]
IMEI (slot 2)	[Redacted]

**Sobre**

### Especificações do dispositivo

Nome do dispositivo	DNS-ASUS
Processador	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
RAM instalada	8,00 GB (utilizável: 7,89 GB)
ID do dispositivo	[Redacted]
ID do Produto	[Redacted]
Tipo de sistema	Sistema operacional de 64 bits, processador baseado em x64
Caneta e toque	Suporte para caneta

[Copiar](#)

[Renomear este computador](#)

### Especificações do Windows

Edição	Windows 10 Home Single Language
Versão	21H2
Instalado em	18/4/21
Compilação do SO	19044.1645
Experiência	Windows Feature Experience Pack 120.2212.4170.0

**Propriedades de Intel(R) Wireless Bluetooth(R)**

Intel(R) Wireless Bluetooth(R)

Provedor do driver:	Intel Corporation
Data do driver:	3/10/19
Versão do driver:	20.100.6.4
Signatário digital:	Microsoft Windows Hardware Compatibility Publisher

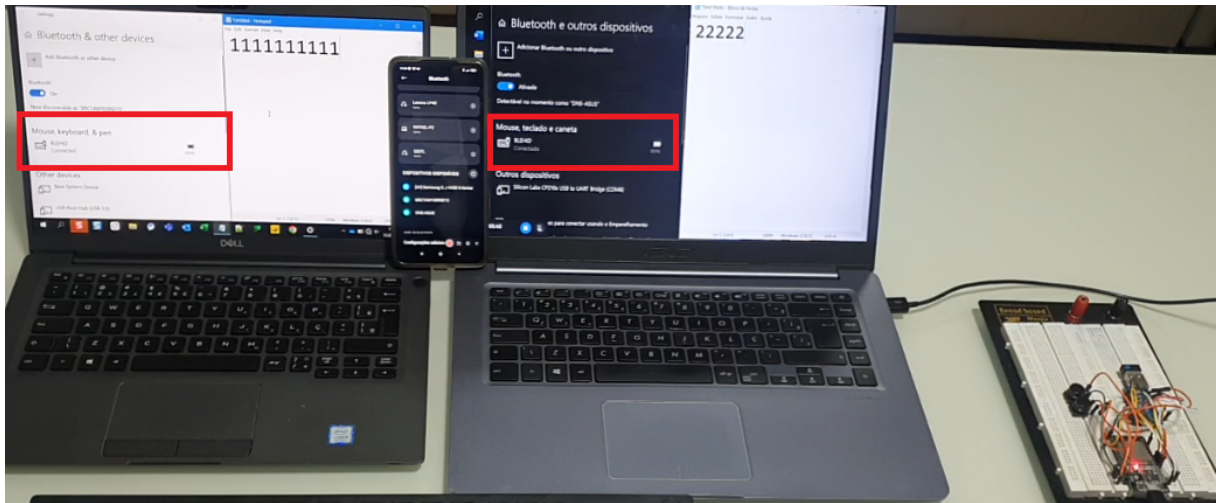
Fonte: A autoria própria (2022).

## 4.2 Testagem

Tendo sido realizados diversos testes e adaptações até que se fosse possível chegar ao dispositivo final, o primeiro desenvolvimento, conforme já mencionado, se deu em relação à conectividade Bluetooth. Os testes de conexão Bluetooth unitária transcorreram bem desde o início, encontrando-se dificuldades apenas após a implementação da leitura USB, especialmente quando chegadas as etapas de conexão com múltiplos receptores e de chaveamento entre eles, mas que, após os ajustes detalhados na seção 3.2.5, também foram bem-sucedidos.



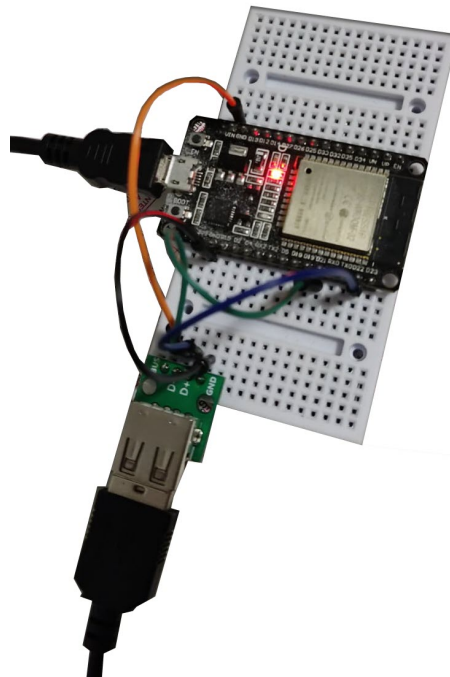
**Figura 27 – Conexão Bluetooth.**



**Fonte: Autoria própria (2022).**

A simples conexão entre os pinos do conector USB às portas GPIO, que funcionou conforme esperado apenas para o mouse com fio genérico mas foi mal-sucedida nos testes com periféricos mais modernos e complexos, como os conjuntos de teclado e mouse sem fio. Portanto, a escolha pela utilização do módulo *shield* para operar como *host* USB, foi um ajuste necessário que só foi possível em razão dos testes realizados.

**Figura 28 – Conector USB ligado às portas GPIO.**



**Fonte: Autoria própria (2022).**

Concluída a implementação do chaveador para realizar a leitura USB, transmissão BLE e chaveamento de receptor, passou-se a testar a resposta dos receptores às entradas realizadas nos periféricos, o que se deu com a utilização do monitor serial da *Integrated Development*

Figura 29 – Resultados dos testes da conexão USB às portas GPIO.

### Mouse "low-speed"

```

USB0: Ack = 6 Nack = 0 01 pcurrent->cb_Cmd = 14 state = 104 epCount = 1
in: 0x00 0x07 0xff 0x00 MOUSE X POSITION: 42 [+7]
MOUSE Y POSITION: -7 [-1]

in: 0x00 0x10 0xfb 0x00 MOUSE X POSITION: 58 [+16]
MOUSE Y POSITION: -12 [-5]

in: 0x00 0x0f 0xfc 0x00 MOUSE X POSITION: 73 [+15]
MOUSE Y POSITION: -16 [-4]

```

### Mouse "full-speed"

```

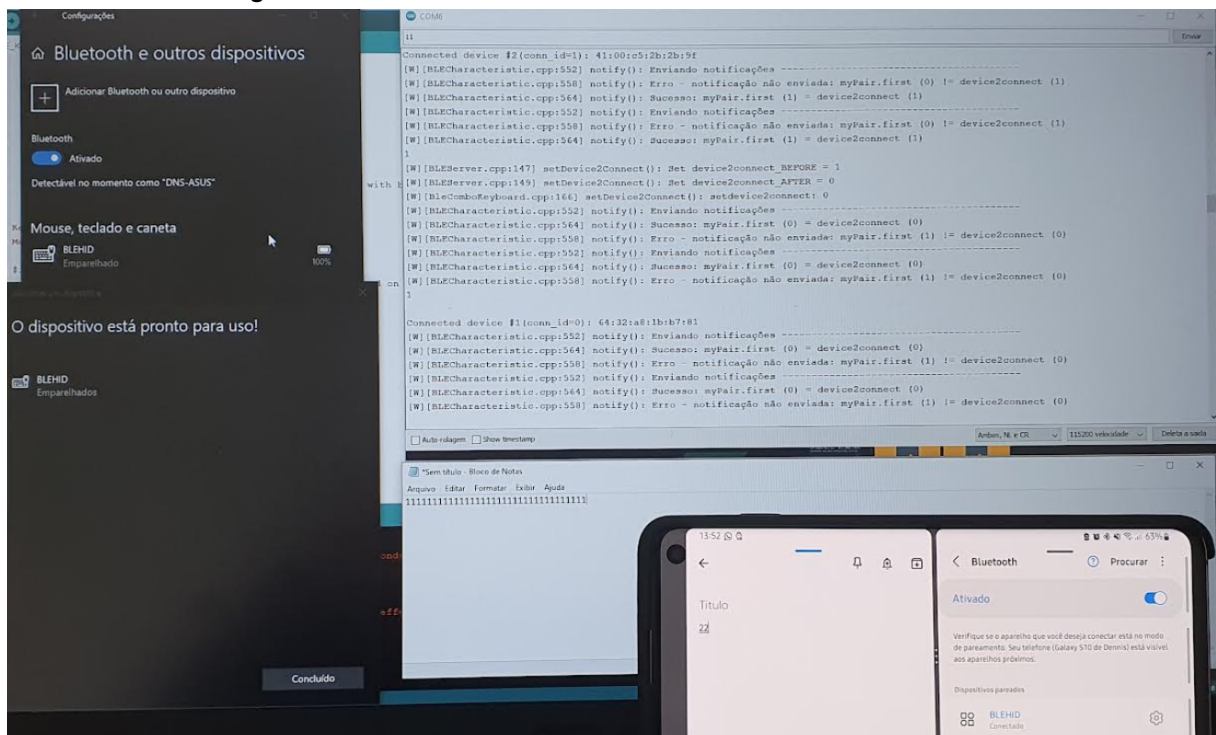
USB 1.1 FULL SPEED (12Mbit/s) DEVICE DETECTED... not supported.

```

Fonte: Autoria própria (2022).

*Environment* (IDE) Arduino e do bloco de notas padrão do Windows. Os testes desta etapa demonstraram o funcionamento adequado do dispositivo projetado.

Figura 30 – Resultados dos testes finais no kit de desenvolvimento.



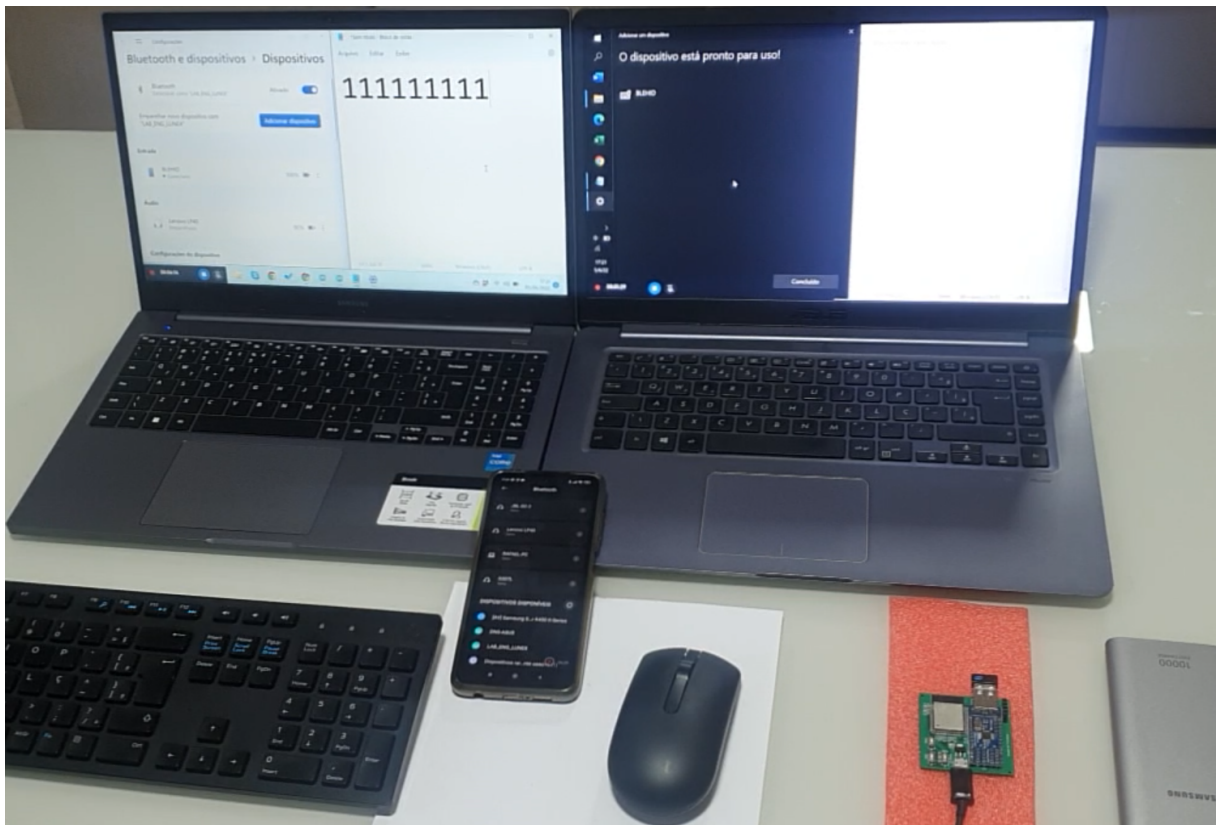
Fonte: Autoria própria (2022).

Finalizados os testes com o kit de desenvolvimento e confeccionada a PCB, também foram realizados testes para averiguar o correto funcionamento do dispositivo após a solda dos componentes e programação na placa.

Não sendo mais possível realizar o monitoramento via serial da operação, pela ausência do chip CP2102, os testes desta etapa foram realizados integralmente de forma prática, com a realização do chaveamento e verificação das respostas nos próprios computadores.

Em geral, o dispositivo se comportou bem desde o início, de modo que a única alteração necessária se deu em relação às portas utilizadas para comunicação dos botões. Ao contrário do que ocorria com o kit de desenvolvimento, as portas dos botões precisaram ser inicializadas como entrada ao invés de saída, mas, uma vez procedida com esta alteração, o chaveador passou a funcionar como já vinha ocorrendo anteriormente.

**Figura 31 – Resultados dos testes finais na PCB.**



Fonte: Autoria própria (2022).

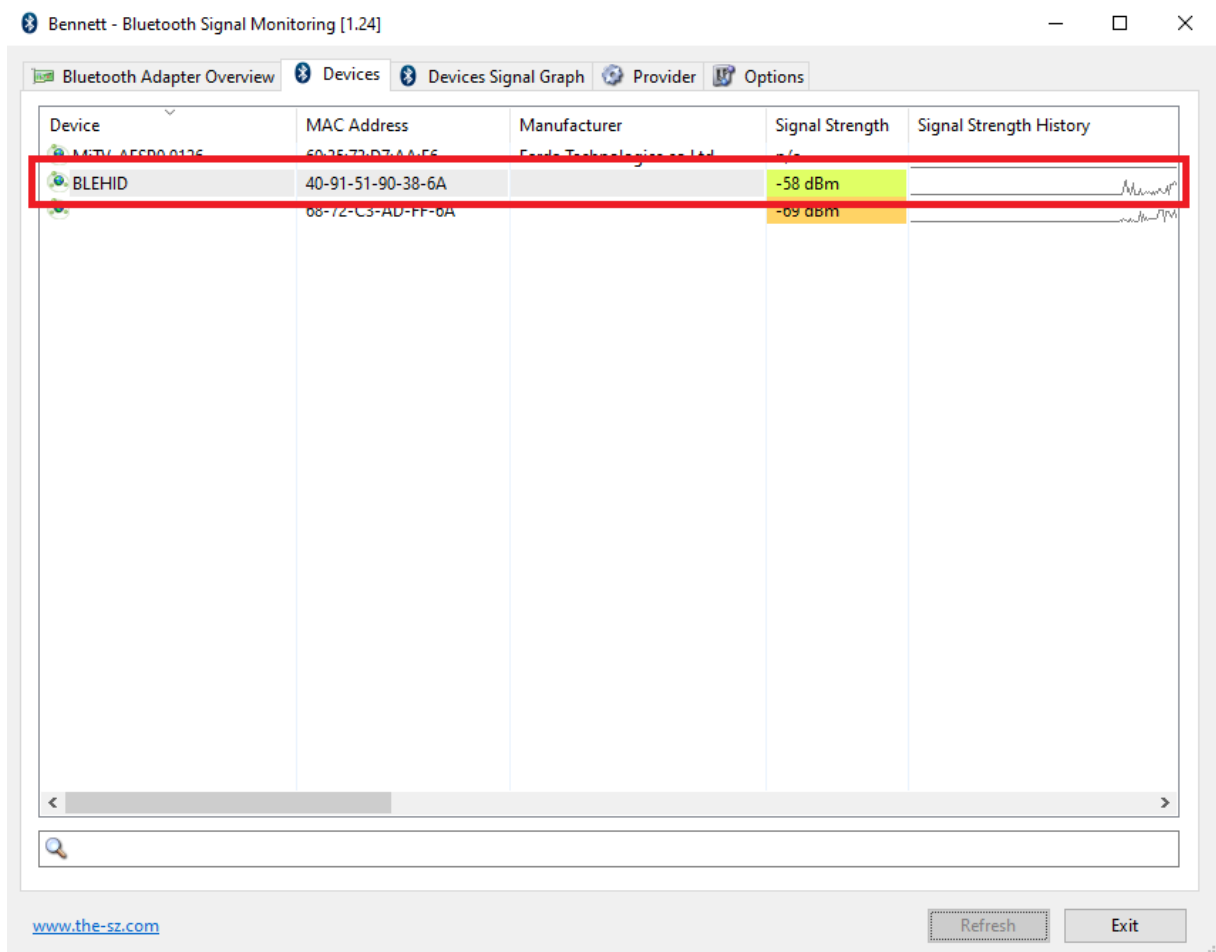
### 4.3 Alcance e intensidade do sinal

Para a verificação do alcance do sinal Bluetooth emitido pelo chaveador, utilizou-se o software *Bennet - Bluetooth Signal Monitoring 1.24*, em 7 diferentes cenários, sempre com chaveador e notebook à altura de 75cm do solo:

- Cenário 1 - computador e chaveador em extremidades opostas de um corredor de 8,0 m em visada direta;

- Cenário 2 - computador e chaveador em extremidades opostas de um corredor de 8,0 m com uma porta fechada entre eles;
- Cenário 3 - computador e chaveador em um corredor, separados pela distância de 4,1 m em visada direta;
- Cenário 4 - computador e chaveador em um mesmo cômodo, separados lateralmente pela distância de 1,5 m;
- Cenário 5 - computador e chaveador em um mesmo cômodo, separados pela distância de 1,5 m, com o chaveador atrás do notebook;
- Cenário 6 - computador e chaveador lado a lado, a 25 cm um do outro, com o chaveador na posição vertical;
- Cenário 7 - computador e chaveador lado a lado, a 25 cm um do outro, com o chaveador na posição vertical;

**Figura 32 – Medição de intensidade de sinal Bluetooth.**



Fonte: Autoria própria (2022).

Os resultados das medições são apresentados na Tabela 3 e, como se observa, os melhores resultados em termos de intensidade de sinal foram verificados no cenário 7, ou seja, com computador e chaveador posicionados lado a lado, separados pela distância de 25 cm e com o chaveador na posição vertical, tendo sido registrado o valor de -58 dBm em uma das medições e média de -59,4 dBm.

**Tabela 3 – Medição de intensidade do sinal Bluetooth do chaveador**

Medição	Cenário 1	Cenário 2	Cenário 3	Cenário 4	Cenário 5	Cenário 6	Cenário 7
1	-85 dBm	-83 dBm	-77 dBm	-74 dBm	-70 dBm	-62 dBm	-61 dBm
2	-82 dBm	-81 dBm	-85 dBm	-75 dBm	-68 dBm	-72 dBm	-58 dBm
3	-85 dBm	-83 dBm	-79 dBm	-81 dBm	-68 dBm	-66 dBm	-59 dBm
4	-82 dBm	-82 dBm	-81 dBm	-83 dBm	-66 dBm	-71 dBm	-61 dBm
5	-83 dBm	-83 dBm	-82 dBm	-82 dBm	-61 dBm	-67 dBm	-59 dBm
Média <sup>1</sup>	-83,2 dBm	-82,3 dBm	-80,0 dBm	-77,4 dBm	-65,3 dBm	-66,1 dBm	-59,4 dBm
Mediana	-83 dBm	-83 dBm	-81 dBm	-81 dBm	-68 dBm	-67 dBm	-59 dBm

Fonte: Autoria própria.

Apesar da significativa diferença na intensidade do sinal recebido, em relação à digitação não foi registrada variação perceptível entre os cenários de 3 a 7, de modo que os sinais de teclado e mouse foram recebidos sem falhas até a distância de 4,1 m em visada direta, com a ocorrência de eventos de desconexão esporádicos em distâncias superiores. Logo, podemos considerar o cenário 3 como o limiar de recepção em boa qualidade, ou seja, quando a intensidade do sinal recebido é de cerca de -80 dBm.

#### 4.4 Consumo energético

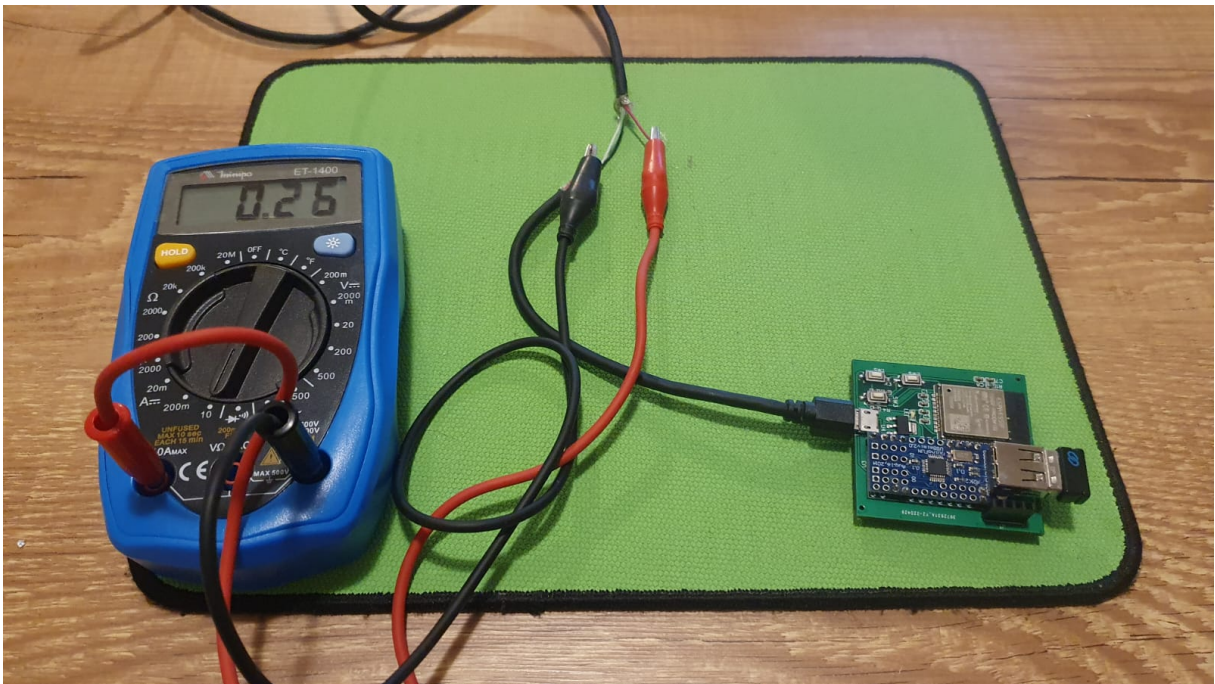
Para a obtenção dos valores de tensão e corrente, de modo a ser possível o cálculo do consumo energético do dispositivo, foi utilizado primeiramente um multímetro em paralelo com a alimentação USB provida por um *power bank* e, após a coleta do valor da queda de tensão, invariavelmente de 5 V, ligou-se o multímetro em série, conforme Figura 33, sendo verificados diferentes valores de corrente para diferentes cenários de uso do dispositivo, cujos resultados estão dispostos na Tabela 4.

**Tabela 4 – Resultados da medição de consumo energético pelo chaveador**

Cenário	Tensão	Corrente 2	Consumo
Chaveador <i>advertising</i> sem dispositivos conectados via BLE ou USB	5 V	0,22 A	1,10 W
Chaveador <i>advertising</i> sem dispositivo conectado via BLE com 1 dongle USB	5 V	0,23 A	1,15 W
Chaveador com 1 dispositivo conectado via BLE e 0 USB	5 V	0,23 A	1,15 W
Chaveador com 1 dispositivo conectado via BLE e 1 dongle USB	5 V	0,25 A	1,25 W
Chaveador <i>advertising</i> com 1 dispositivo conectado via BLE e 0 USB	5 V	0,24 A	1,20 W
Chaveador <i>advertising</i> com 1 dispositivo conectado via BLE e 1 dongle USB	5 V	0,26 A	1,30 W
Chaveador com 2 dispositivos conectados via BLE e 0 USB	5 V	0,24 A	1,20 W
Chaveador com 2 dispositivos conectados via BLE e 1 dongle USB	5 V	0,27 A	1,35 W

Fonte: Autoria própria.

Figura 33 – Medição do consumo energético.



Fonte: Autoria própria (2022).

Desta maneira, mediante a multiplicação dos valores de corrente e tensão, que correspondem a 0,27 A e 5 V no caso mais extremo, pode-se obter a potência e o consumo energético do dispositivo, de 1,35 W e 10,8 Wh por dia para um uso de 8 horas diárias.

#### 4.5 Custos de produção

Um dos objetivos iniciais do presente trabalho consistia em manter um baixo custo do dispositivo projetado, a fim de efetivamente agregar valor aos usuários e também de viabilizar a inserção no mercado.

Neste quesito, alguns obstáculos que não haviam sido mapeados inicialmente acabaram por elevar o custo total do chaveador, sendo que a necessidade de utilizar o módulo adicional para desempenho da função de *host* foi o que mais pesou, representando mais de 50 por cento do custo final do protótipo e quase dois terços do preço dos componentes, excluído o frete.

As tabelas a seguir apresentam o custo total para a fabricação do chaveador *wireless* por Bluetooth.

Dos valores acima apresentados, percebe-se que há uma redução significativa quando comparado com os valores dos kits de teclado e mouse para multi-dispositivos fornecidos por grandes fabricantes como a Logitech, ainda que se considere a necessidade de adquirir um conjunto separadamente.

**Tabela 5 – Custos para a fabricação unitária de um chaveador wireless**

Itens	Componentes	Quantidade	Preço unitário	Preço R\$
1	LED -SMD	1	R\$ 0,570	R\$ 0,57
2	RESISTOR 10k $\Omega$	4	R\$ 0,014	R\$ 0,06
3	ESP32-WROOM-32	1	R\$ 19,145	R\$ 19,15
4	CAPACITOR 22 $\mu$ F 6,3V	1	R\$ 0,241	R\$ 0,24
5	RESISTOR 120 $\Omega$	1	R\$ 0,069	R\$ 0,07
6	CAPACITOR 10 $\mu$ F 10V	2	R\$ 0,157	R\$ 0,31
7	CAPACITOR 100nF 10V	4	R\$ 0,032	R\$ 0,13
8	REGULADOR 3,3V	1	R\$ 0,743	R\$ 0,74
9	CAPACITOR 1 $\mu$ F 6,3V	1	R\$ 0,270	R\$ 0,27
10	MICRO USB FÊMEA V8	1	R\$ 2,000	R\$ 2,00
11	CHAVE DE TOQUE SMD	3	R\$ 0,900	R\$ 2,70
12	PCB	1	\$2,00	R\$ 9,94
13	STENCIL P&P	1	\$1,53	R\$ 7,60
15	USB HOST SHIELD	1	R\$ 68,89	R\$ 68,89
14	FRETE	1	\$4,14	R\$ 20,58

Fonte: Autoria própria.

**Tabela 6 – Valor total para a produção do chaveador**

Valor	Preço R\$
Valor sem Frete	R\$112,67
Valor com Frete	R\$133,25

Fonte: Autoria própria.

## 5 CONCLUSÃO

Uma vez encerrado o processo de desenvolvimento do dispositivo ora apresentado, restou evidenciada a grande importância que as criações das tecnologias USB e Bluetooth tiveram para o aperfeiçoamento dos aparatos eletrônicos e a sua difusão no mundo contemporâneo. Tais tecnologias revolucionaram a forma de transmissão de dados, possibilitando a concepção de funcionalidades antes tidas como futuristas ou até mesmo inimagináveis, como a utilização de um único conector para diversos periféricos e o pareamento de diversos dispositivos para reprodução de áudio em sincronia. A estrutura de colaboração e a sistematização que estão por trás de tais tecnologias também são de impressionar, permitindo uma padronização em grande escala e a aceleração do desenvolvimento de novas tecnologias.

Justamente em razão dessa revolução, foi possível projetar e desenvolver o chaveador sem fio descrito no presente documento em consonância com o objetivo primordial de facilitar o desempenho do trabalho de forma remota. Ou seja, o produto apresentado cumpre com a função maior de reduzir a quantidade de periféricos em uma mesma mesa ou bancada que contenha mais de um computador, além de atingir de maneira satisfatória os objetivos secundários, como simplicidade do dispositivo e facilidade de uso.

Desta maneira, tem-se que o aparelho apresentado pode trazer facilidades bastante significativas a quem efetivamente opera mais de um computador em seu dia-a-dia e necessita com frequência alternar entre eles, permitindo-lhes, além da comodidade, poupar espaço e maximizar o uso ou reaproveitar periféricos, portanto, tendo efeitos, também, na harmonia do ambiente de trabalho ou estudo. Apesar de a motivação principal consistir em prover benefícios às pessoas que foram surpreendidas diante de uma nova realidade e que não têm uma grande estrutura ou muito espaço à sua disposição, verifica-se que o presente projeto também pode trazer benefícios a locais de trabalho mais bem equipados e que também demandam a utilização de mais de um computador por parte de seus colaboradores.

Apesar do sucesso acima relatado, tem-se que ao menos um dos objetivos norteadores do presente trabalho foi atingido apenas de forma parcial: disponibilizar a solução a um baixo custo. Isto porque, apesar de haver a possibilidade de disponibilizar o chaveador projetado a um custo inferior ao de conjuntos de teclado e mouse vendidos por grandes empresas de tecnologia, conforme apresentado anteriormente, alguns contratemplos acabaram por elevar o custo inicialmente projetado, em especial a necessidade de acrescentar componentes adicionais para proporcionar a função de *host* para o microcontrolador escolhido.

### 5.1 Trabalhos futuros

Logo, o primeiro ponto de aperfeiçoamento do dispositivo projetado passa necessariamente pela redução dos custos de produção, o que se imagina que pode ser alcançado através de outras soluções para propiciar a função de *host* a este microcontrolador ou a substituição



por outro que já traga essa possibilidade de forma nativa, ainda que com preço ligeiramente superior. Também se prevê uma redução de custos pela produção em larga escala do dispositivo, o que certamente levaria a um menor preço unitário.

Em uma versão comercial do chaveador proposto, também deve ocorrer o refinamento do código-fonte utilizado, acreditando-se que isto pode impactar em um uso mais moderado do microcontrolador e demais componentes, em aspectos de memória e até na correção de eventuais bugs que passaram despercebidos. Este refinamento certamente resultaria em menos erros na utilização massiva do chaveador, além de provavelmente resultar em economia de energia, que foi um aspecto não controlado de forma significativa no presente trabalho, limitado, em relação a este ponto, à utilização do BLE.

Algumas outras melhorias que provavelmente ampliariam o mercado do dispositivo projetado incluem a disponibilização de uma porta USB adicional, a viabilização da personalização do número de dispositivos receptores e a implementação de algum tipo de *feedback* visual, considerado inicialmente desnecessário e inexistente no protótipo ora apresentado.

Por fim, o mapeamento de outros teclados, que não o padrão americano ou brasileiro, voltados para outras línguas e países, uma redução do tamanho do dispositivo, o desenvolvimento de um envoltório para proteção do equipamento e fornecer uma identidade visual também são fatores a serem considerados.

## REFERÊNCIAS

- APPLE. **Getting Started with iBeacon**. 2014. Acessado em Novembro de 2021. Disponível em: <https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf>.
- AXELSON, J. **USB Complete: The Developer's Guide**. 5. ed. [S.l.]: Lakeview Research LLC,, 2015.
- BLUETOOTH. **HUMAN INTERFACE DEVICE PROFILE 1.1**. 2015. Acessado em Março de 2021. Disponível em: <https://www.bluetooth.com/specifications/specs/human-interface-device-profile-1-1-1/>.
- BLUETOOTH. **Bluetooth Technology Overview**. 2021. Acessado em Maio de 2022. Disponível em: <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>.
- BLUETOOTH. **Bluetooth 2022 Market Update**. 2022. Acessado em Abril de 2022. Disponível em: [https://www.bluetooth.com/2022-market-update/?utm\\_campaign=bmu&utm\\_source=internal&utm\\_medium=blog&utm\\_content=new-trends-and-forecasts-for-the-next-5-years](https://www.bluetooth.com/2022-market-update/?utm_campaign=bmu&utm_source=internal&utm_medium=blog&utm_content=new-trends-and-forecasts-for-the-next-5-years)".
- BRIGATTI, F. **Reunião no escritório e buscar filhos na escola: trabalhadores querem flexibilidade no pós-pandemia**. Folha de São Paulo, 2021. Acessado em Maio de 2022. Disponível em: <https://www1.folha.uol.com.br/mercado/2021/09/reuniao-no-escritorio-e-buscar-filhos-na-escola-trabalhadores-querem-flexibilidade-no-pos-pandemia.shtml>.
- BRITANNICA, T. E. of E. **USB**. Encyclopedia Britannica, 2021. Acessado em Maio de 2022. Disponível em: <https://www.britannica.com/technology/USB>.
- DIAN, F. J.; YOUSEFI, A.; LIM, S. A practical study on bluetooth low energy (ble) throughput. In: **2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)**. [S.l.: s.n.], 2018. p. 768–771.
- GUPTA, N. **Inside Bluetooth Low Energy**. 2. ed. Boston, EUA: Artech House, 2016.
- HUNG, M.-H. *et al.* A reconfigurable dual-band wifi/bt combo transceiver with integrated 2g/bt sp3t, lna/pa achieving concurrent receiving and wide dynamic range transmitting in 40nm cmos. In: **2017 IEEE Asian Solid-State Circuits Conference (A-SSCC)**. [S.l.: s.n.], 2017. p. 177–180.
- IPSOS. **Workers want more flexibility from their employers after COVID**. 2021. Acessado em Maio de 2022. Disponível em: <https://www.ipsos.com/en/return-to-the-workplace-global-survey>.
- Jaycon Systems. **Bluetooth Technology: What Has Changed Over the Years**. 2017. Acessado em Março de 2021. Disponível em: <https://medium.com/jaycon-systems/bluetooth-technology-what-has-changed-over-the-years-385da7ec7154>.
- KAJIKAWA, N. *et al.* On availability and energy consumption of the fast connection establishment method by using bluetooth classic and bluetooth low energy. In: **Fourth International Symposium on Computing and Networking**. [S.l.: s.n.], 2016. p. 286–290.
- KOLBAN, N. **Kolban's book on ESP32**. [S.l.]: Leanpub, 2018.
- LUNT, B. D. **USB: The Universal Serial Bus**. 2. ed. [S.l.]: Forever Young Software, 2014.
- MARRA, R. **Imóveis encolhem para caber no bolso e no estilo de vida do morador**. Folha de São Paulo, 2019. Acessado em Abril de

2022. Disponível em: <https://www1.folha.uol.com.br/sobretudo/morar/2019/11/imoveis-encolhem-para-caber-no-bolso-e-no-estilo-de-vida-do-morador.shtml>.

MAXIM INTEGRATED. **MAX3421E USB Peripheral/Host Controller with SPI Interface**. [S.l.], 2013. Rev 4. Disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/132440/MAXIM/MAX3421E.html>.

MIRANDA, I. **Mercado de trabalho: vagas home office crescem 140% em 2022**. Contabeis.com.br, 2022. Acessado em Abril de 2022. Disponível em: <https://www.contabeis.com.br/noticias/50742/vagas-home-office-crescem-140-em-2022/>.

okiraku-camera. **Tente usar ESP32 e miniUHS para alterar o teclado USB para um teclado BLE HID**. 2018. Acessado em Maio de 2022. Disponível em: <https://okiraku-camera.tokyo/blog/?p=8333>.

Poder 360. **Brasil tem 2 dispositivos digitais por habitante, diz FGV**. 2021. Acessado em Maio de 2022. Disponível em: <https://www.poder360.com.br/tecnologia/brasil-tem-2-dispositivos-digitais-por-habitante-diz-fgv/>.

SAIRAM, K.; GUNASEKARAN, N.; REDD, S. Bluetooth in wireless communication. **IEEE Communications Magazine**, v. 40, n. 6, p. 90–96, 2002.

Silicon Labs. **Throughput with Bluetooth Low Energy Technology**. 2022. Acessado em Maio de 2022. Disponível em: <https://docs.silabs.com/bluetooth/latest/general/system-and-performance/throughput-with-bluetooth-low-energy-technology>.

TOWSEND, K. *et al.* **Getting Started with Bluetooth Low Energy**. [S.l.]: O'Reilly, 2014.

USB Implementers' Forum. **Device Class Definition for Human Interface Devices (HID) 1.11**. 2001. Acessado em Maio de 2022. Disponível em: <https://www.usb.org/document-library/device-class-definition-hid-111>.

USB Implementers' Forum. **HID Usage Tables FOR Universal Serial Bus (USB)**. 2022. Acessado em Maio de 2022. Disponível em: <https://usb.org/document-library/hid-usage-tables-13>.

ZAIDEM, M. **Afinal, o futuro será home office, híbrido ou presencial em 2022?** Exame, 2021. Acessado em Abril de 2022. Disponível em: <https://exame.com/bussola/afinal-o-futuro-sera-home-office-hibrido-ou-presencial-em-2022/>.

ZEADALLY, S.; SIDDIQUI, F.; BAIG, Z. 25 years of bluetooth technology. **Future Internet**, MDPI AG, v. 11, n. 9, p. 194, 2019. ISSN 1999-5903. Acessado em Abril de 2022. Disponível em: <http://dx.doi.org/10.3390/fi11090194>.

ZHANG, J. Design of embedded bluetooth information broadcast system. In: **2011 International Conference on Electrical and Control Engineering**. [S.l.: s.n.], 2011. p. 2142–2145.