

Universidade do Minho
Escola de Engenharia

Grupo 5:

José Gomes a93083

Diogo Cerqueira a93108

Rui Cunha a93093

Francisco Martins a93079

Luís Oliveira a89380

Relatório Final

MIETI

Projeto Integrador em Telecomunicações e Informática
2022/2023

Índice

Lista de figuras	iii
Lista de tabelas	iv
1 Introdução	1
2 Arquitetura do sistema final	2
2.1 Protocolo de comunicação entre aplicação (PC) e a placa ESP32	2
2.2 Protocolo de comunicação entre placas ESP32	4
3 Comunicação ótica por infravermelhos entre placas ESP32	7
3.1 Implementação do <i>link</i> ótico	7
3.1.1 Emissor	7
3.1.2 Recetor	8
a) Filtro Passa-alto	9
b) Amplificador de transimpedância	10
c) Comparador de Tensão	11
d) Retificador de meia onda	12
3.2 Formas de onda dos circuitos emissor e recetor	14
4 Interface gráfica	15
5 Testes	17
5.1 Teste de distância	17
5.2 Teste de distância com obstáculos	19
6 Autoavaliação/Reflexão	20
7 Conclusão	23
Referências Bibliográficas	24

Lista de figuras

Arquitetura final do sistema.	2
Algoritmo do protocolo porta série (lado do emissor).	3
Algoritmo do protocolo porta série (lado do recetor).	4
Estrutura da trama.	5
Algoritmo da placa emissora.	5
Algoritmo da placa recetora.	6
Circuito do emissor.	7
Circuito do recetor.	8
Filtro passa-alto.	9
Amplificador de transimpedância.	10
Forma de onda dos circuitos emissor e recetor (transimpedância).	10
Comparador de Tensão.	11
Forma de onda dos circuitos emissor e recetor (comparador).	12
Retificador de meia onda.	12
Forma de onda dos circuitos emissor e recetor (retificador).	13
Formas de onda dos circuitos emissor e recetor.	14
Menu Principal.	15
Menu Emissor.	16
Menu Recetor.	16
Testes de distância entre os circuitos emissor e recetor.	17
Testes de distância entre os circuitos emissor e recetor (alcance máximo).	18
Forma de onda do circuito recetor.	19

Lista de tabelas

Tabela com distâncias dos testes efetuados.	18
Tabela com distâncias dos testes efetuados nos quais houve erros de transmissão.	19

Acrónimos

PC *Personal Computer*

LED *Light Emitting Diode*

USB *Universal Serial Bus*

AC *Alternating current*

GND *Ground*

1 Introdução

Serve o presente relatório como visão global e objetiva do sistema desenvolvido, do qual inclui a descrição das fases propostas e funcionalidades derivadas.

O propósito principal deste projeto é o desenvolvimento de um protótipo que possibilite a transmissão de informação entre dois PCs (*Personal Computers*) sobre um *link* óptico. O sistema desenvolvido contém duas entidades essenciais: um emissor, que transmite a informação com recurso a um LED (*Light Emitting Diode*), e um recetor, que é capaz de receber a informação e imprimi-la a um determinado utilizador.

A comunicação série entre cada PC e o correspondente microcontrolador da placa ESP32 é feita através de um cabo USB (*Universal Serial Bus*).

O sistema implementado pelo grupo proporciona as funcionalidades seguintes:

- Aplicação geradora da mensagem e aplicação para visualização da mensagem recebida.
- Protocolos de comunicação, entre cada PC e a respetiva ESP32, e para a comunicação entre placas ESP32.
- Mecanismos de codificação e descodificação dos *bits* no *link* físico, tendo em consideração as ligações por fios necessárias, os níveis lógicos e as respetivas tensões.
- Mecanismos de sincronização do fluxo de *bits* entre placas ESP32, a nível do *bit*, do *byte* e da trama.
- Circuitos eletrónicos do emissor e do recetor de infravermelhos.

2 Arquitetura do sistema final

A figura 2.1 ilustra a arquitetura final do sistema.

Através da visualização da figura referida, a comunicação é estabelecida com recurso a um *link* óptico (representado pela seta amarela) que será gerado pela lâmpada LED. Posteriormente, o *link* óptico será detetado pelo fotodetector, que é um componente intrínseco ao recetor e é responsável pela transformação da informação presente no *link* óptico num sinal elétrico. De seguida, do lado do recetor, o PC irá executar uma aplicação que receberá e apresentará a informação correspondente.

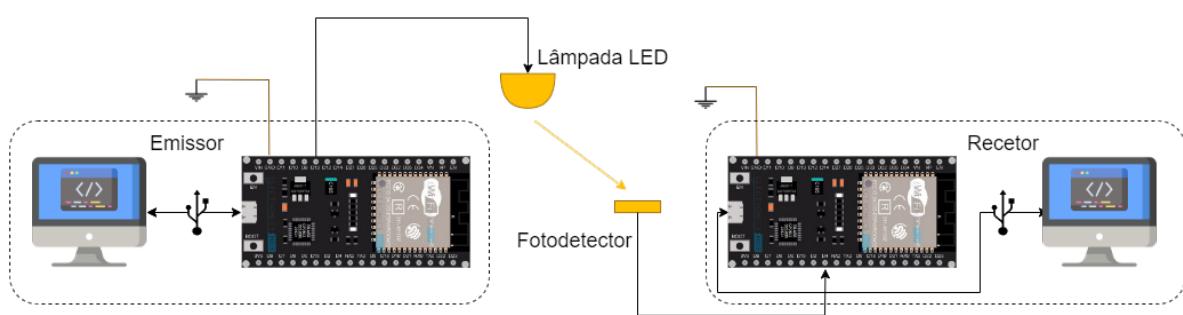


Figura 2.1: Arquitetura final do sistema.

2.1 Protocolo de comunicação entre aplicação (PC) e a placa ESP32

O protocolo de comunicação entre o PC e a placa ESP32 foi implementado com recurso à linguagem de programação Java [1]. Dentro do domínio da linguagem referida, recorreu-se à biblioteca “**jssc**” [2], que serviu de base à implementação do protocolo referido. A biblioteca citada fornece um conjunto de recursos, nomeadamente, classes e métodos que permitem a comunicação com a porta série. Os recursos dessa biblioteca enquadram-se nas categorias seguintes:

- Listagem das portas: Listam as portas série aptas para comunicar.
- Escrita da informação: Escrita da informação na porta série.
- Leitura da informação: Leitura da informação proveniente da porta série.

Em junção com as categorias descritas, a biblioteca referida também fornece uma classe do tipo *Event Listener*, que permanece de forma contínua à escuta de possíveis eventos na porta série, como é o caso da receção de tramas.

A figura 2.2 ilustra o algoritmo (executado pelo emissor) relativo ao funcionamento do protocolo de comunicação entre o PC e a placa ESP32.

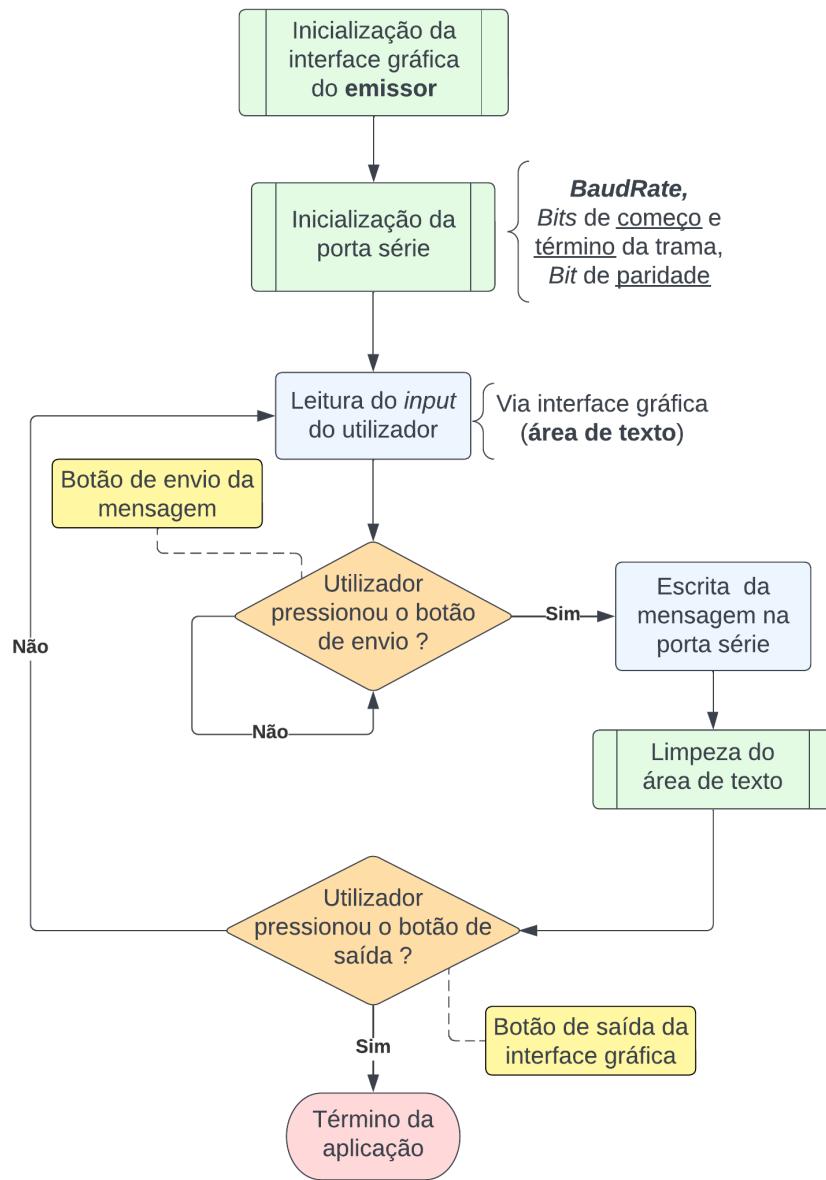


Figura 2.2: Algoritmo do protocolo porta série (lado do emissor).

A figura 2.3 ilustra o algoritmo (executado pelo receptor), inerente ao funcionamento do protocolo de comunicação entre o PC e a placa ESP32.

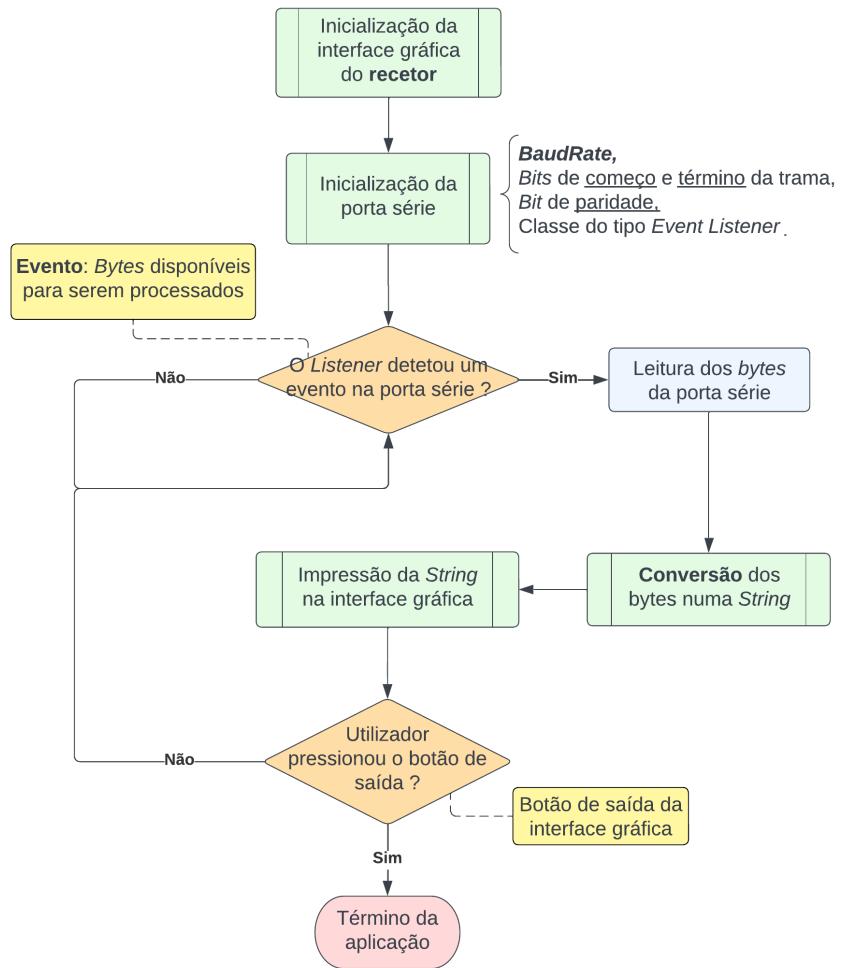


Figura 2.3: Algoritmo do protocolo porta série (lado do receptor).

2.2 Protocolo de comunicação entre placas ESP32

O protocolo de comunicação entre entre placas ESP32 foi implementado com recurso à linguagem de programação *Arduino* [3].

A figura 2.4 descreve a fisionomia de um datagrama, que se enquadra no comportamento do protocolo de comunicação entre as placas (ESP32), concebido pelo grupo. O protocolo concebido é adjacente ao protocolo de comunicação entre o PC e placa ESP32, visto que recebe como entrada os *bytes* provenientes da porta série, constrói um datagrama através desses *bytes* e codifica o mesmo, antes de ser encaminhado pelo pino de saída.

Tal como a figura sugere, o datagrama consiste na combinação de três campos essenciais: o campo de sincronização, o campo dados e o campo destinado à deteção e correção de erros.

- *Sync field*: Campo de sincronização, responsável pela sincronização dos pacotes.
- *Data*: Campo de dados, que armazena a mensagem transmitida.
- *Checksum*: Campo responsável pela deteção de erros.

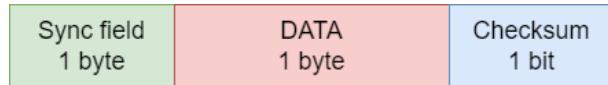


Figura 2.4: Estrutura da trama.

A figura 2.5 ilustra o algoritmo, que é executado pela placa emissora, enquadrado no protocolo de comunicação entre as placas.

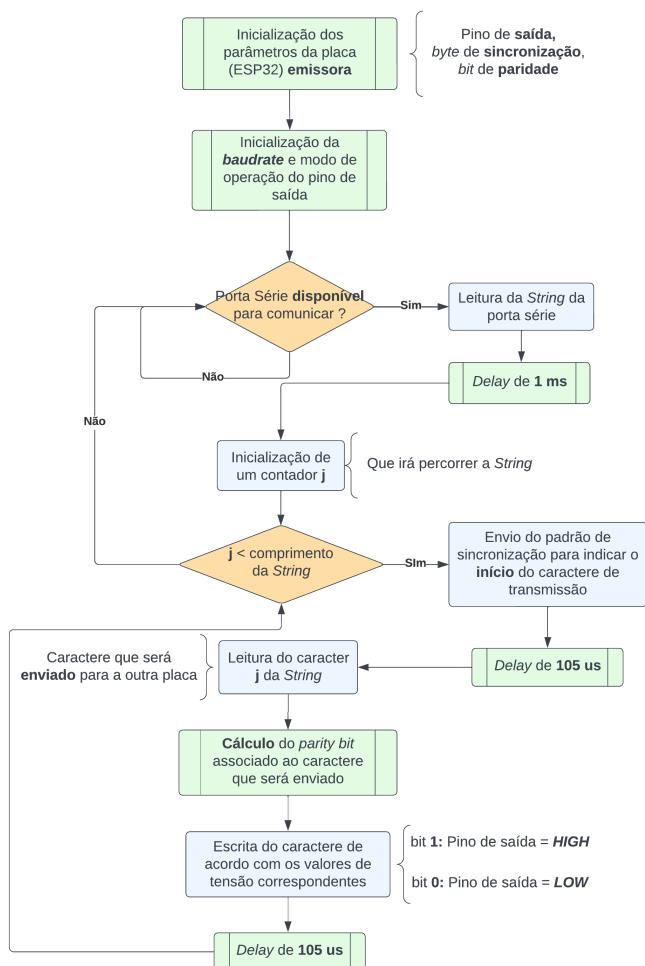


Figura 2.5: Algoritmo da placa emissora.

A figura 2.6 ilustra o algoritmo, que é executado pela placa recetora, enquadrado no protocolo de comunicação entre as placas.

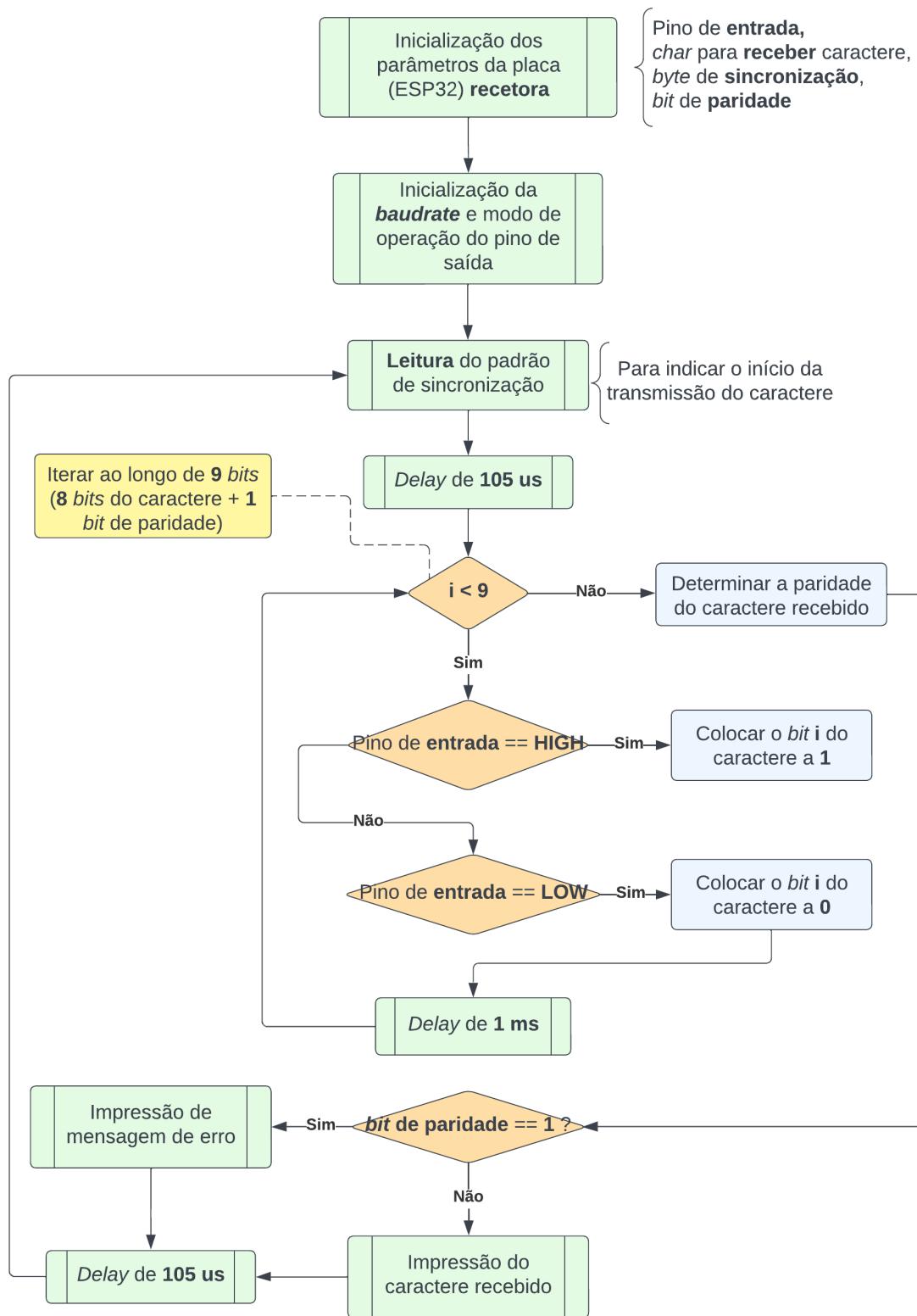


Figura 2.6: Algoritmo da placa recetora.

3 Comunicação ótica por infravermelhos entre placas ESP32

3.1 Implementação do link ótico

Visto que na fase inicial do projeto, a transmissão de dados entre as placas ESP32 foi realizada através de fios de ligação, na fase sucessora foi necessário substituir esses fios de ligação, pois um dos objetivos principais do projeto é a comunicação ótica por infravermelhos, entre placas ESP32.

A implementação do *link* ótico incluiu a execução das tarefas seguintes:

- Projeto do circuito **emissor** de infravermelhos para conversão do sinal elétrico extraído da porta série da estação emissora em sinal ótico.
- Projeto do circuito **recetor** para recuperação do sinal elétrico a partir do sinal ótico recebido e envio pela para porta série da estação recetora.

É de realçar que o grupo integrou alguns componentes eletrónicos nos circuitos do recetor para melhorias na performance da ligação ótica, como por exemplo, o uso de um filtro passa-alto para redução da interferência da luz ambiente sobre o sinal transmitido.

3.1.1 Emissor

A figura 3.7 ilustra o circuito emissor implementado pelo grupo. Para implementar o circuito emissor foi necessário relembrar certos conceitos do domínio da eletrónica, em especial, o modo de funcionamento do transístor bipolar. O circuito emissor, através de um emissor de infravermelhos (LED), transmite o sinal elétrico, oriundo da porta série da estação emissora.

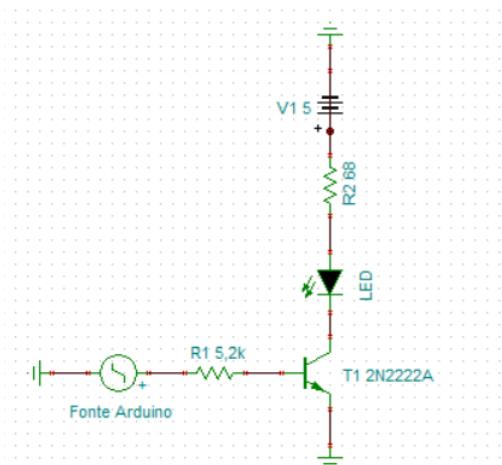


Figura 3.7: Circuito do emissor.

É de realçar, que antes do grupo ter posto em prática o circuito descrito, numa fase prévia, o circuito foi estudado, desenvolvido e testado com recurso ao software TINA [4].

Os cálculos das resistências (R_1 e R_2) e da corrente de base do transístor efetuados pelo grupo, encontram-se listados nas equações abaixo:

$$5 - R_2 * I - 1.3 - 0.6 = 0 \Leftrightarrow 3.7 - R_2 * 0.05 = 0 \Leftrightarrow 3.7 = R_2 * 0.05 \Leftrightarrow \frac{3.7}{0.05} = R_2 \Leftrightarrow R_2 \approx 68 \Omega \quad (3.1)$$

$$I_c = \beta * I_b \Leftrightarrow I_b = \frac{0.05}{100} = 0.5 mA \quad (3.2)$$

$$3.3 - R_1 * I_b - 0.7 = 0 \Leftrightarrow 3.3 - R_1 * 0.5mA - 0.7 = 0 \Leftrightarrow R_1 = 5.2 k\Omega \quad (3.3)$$

3.1.2 Recetor

A figura 3.8 ilustra o circuito recetor implementado pelo grupo. Para implementar o circuito recetor também foi necessário relembrar certos conceitos do domínio da eletrónica, em especial, o modo de funcionamento dos AMPOPs.

O circuito recetor recupera o sinal elétrico através do sinal ótico e envia o mesmo pela porta série da estação recetora. Este circuito pode ser dividido em 4 partes: filtro passa-alto; amplificador de transimpedância; comparador de tensão e retificador de meia onda.

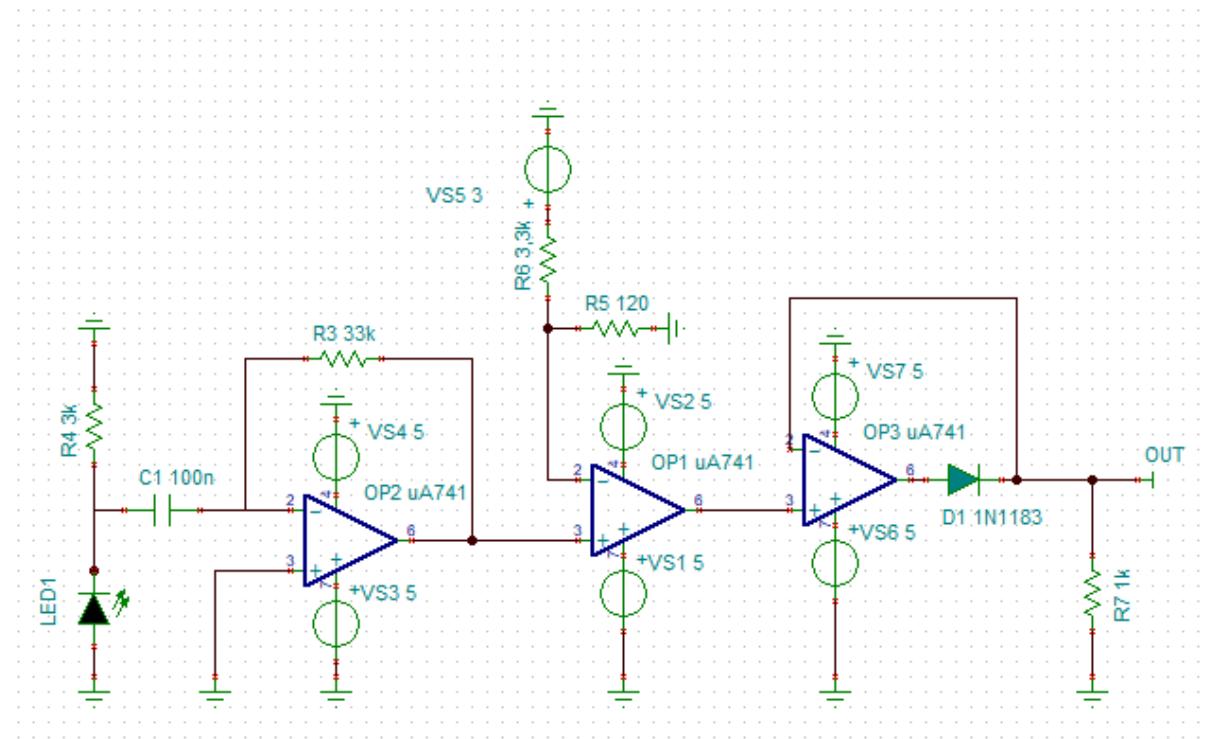


Figura 3.8: Circuito do recetor.

a) Filtro Passa-alto

Na figura 3.9 é possível observar o circuito do filtro passa-alto implementado:

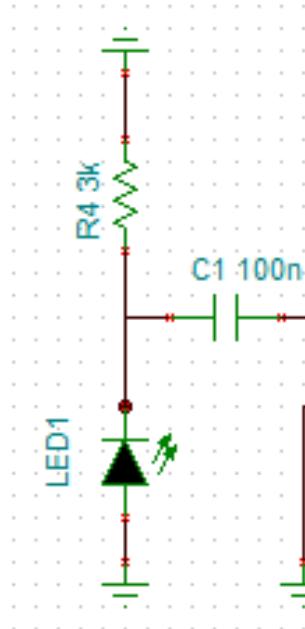


Figura 3.9: Filtro passa-alto.

O filtro passa-alto permite a filtragem da luz ambiente. A frequência de corte foi definida para o valor de **531 Hz**. Caso a frequência seja menor que a frequência de corte, esta é atenuada. Caso a frequência seja maior que a frequência de corte, o filtro permite a passagem dessa frequência. O estabelecimento da frequência de corte permitiu melhorar o sistema, de forma a que o recetor consiga receber o sinal sem interferência e ruído.

O cálculo da frequência de corte, associada ao filtro referido, encontra-se listada na equação abaixo:

$$f_{corte} = \frac{1}{2\pi * R * C} = \frac{1}{2\pi * 3 k\Omega * 100nF} = 531Hz \quad (3.4)$$

b) Amplificador de transimpedância

Na figura 3.10, está demonstrado o esquema do circuito do amplificador de transimpedância.

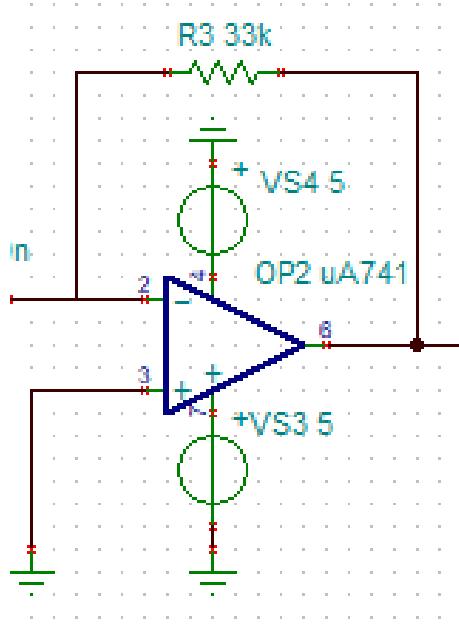


Figura 3.10: Amplificador de transimpedância.

O amplificador de transimpedância converte a corrente de entrada num sinal de tensão proporcional. O sinal de corrente de entrada é aplicado ao terminal negativo do amplificador, enquanto o terminal positivo foi conectado ao GND (*Ground*). O amplificador amplifica o sinal de corrente e converte-o em um sinal de tensão na saída. A magnitude da tensão de saída é diretamente proporcional à corrente de entrada. Este permitiu converter uma corrente muito baixa num sinal de tensão mensurável e amplificado.

Na figura 3.11, é possível observar a onda na saída deste amplificador.

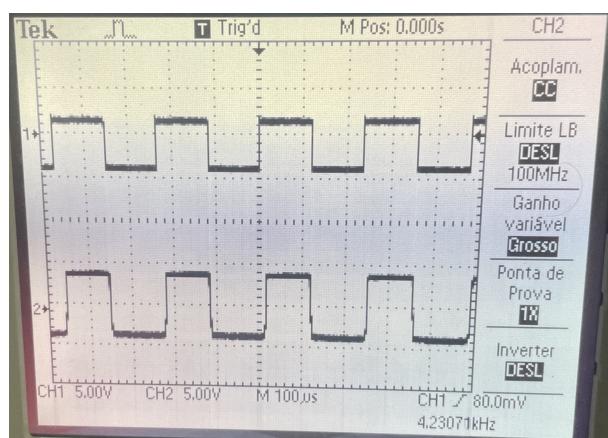


Figura 3.11: Forma de onda dos circuitos emissor e receptor (transimpedância).

O cálculo da resistência R₃ , encontra-se listado na equação abaixo:

$$R_3 = \frac{V_{out}}{0.1 \text{ mA}} = \frac{3.3V}{0.1 \text{ mA}} = 33 \text{ k}\Omega \quad (3.5)$$

c) Comparador de Tensão

Na figura 3.12, é possível ver o circuito do comparador de tensão.

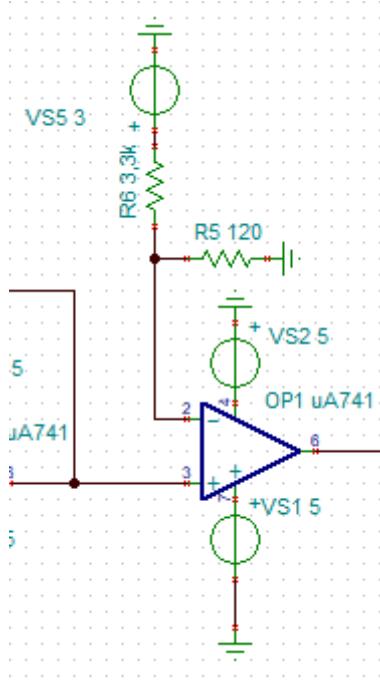


Figura 3.12: Comparador de Tensão.

O comparador de tensão compara dois sinais de entrada e produz um sinal de saída com base na relação de tensão entre eles.

Na entrada negativa foi aplicada o valor de 0.105 V, e cujo, será o valor de referência deste comparador. Este valor surgiu através dos cálculos seguintes:

$$V_{cc} = 3V; R_6 = 3.3 \text{ k}\Omega; R_5 = 120 \Omega.$$

$$V_{ref} = V_{cc} * \frac{R_5}{R_6 + R_5} = 3 * \frac{120 \Omega}{3.3 \text{ k}\Omega + 120 \Omega} \approx 0.105V \quad (3.6)$$

O funcionamento deste comparador de tensão pode ser observado pelo sistema de equações seguinte:

$$V_{saída} = \begin{cases} 3V, & V_{entrada} > 0.105V \\ 0V, & V_{entrada} < 0.105V \end{cases}$$

Na figura 3.13, é possível a observar a forma de onda à saída deste comparador.

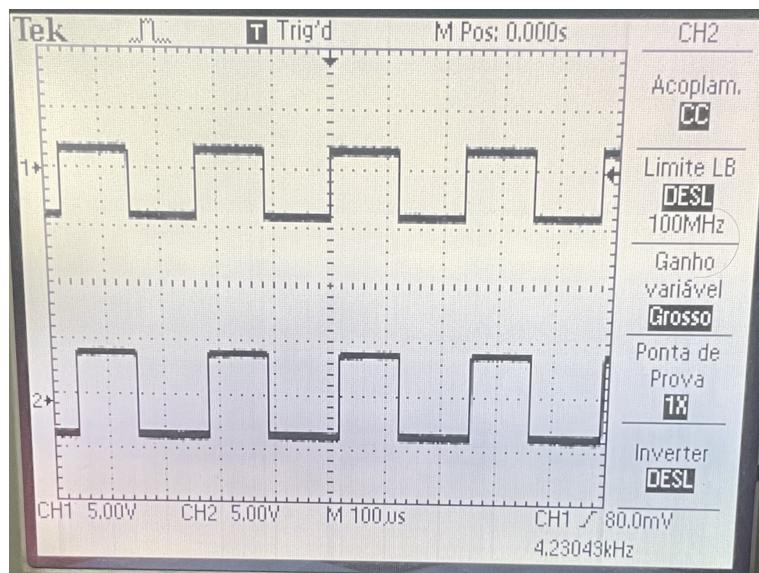


Figura 3.13: Forma de onda dos circuitos emissor e recetor (comparador).

d) Retificador de meia onda

Na figura 3.14, é demonstrado o circuito do retificador de meia onda.

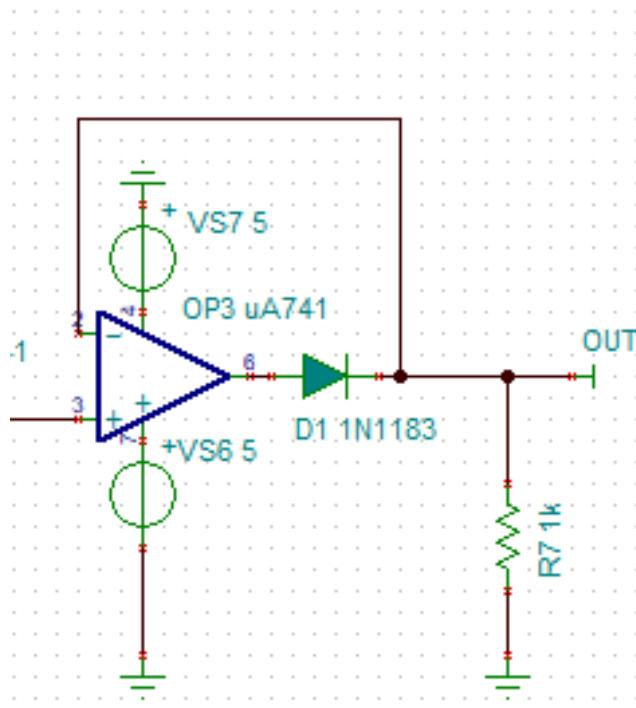


Figura 3.14: Retificador de meia onda.

O retificador de meia onda foi implementado de forma a permitir a passagem de apenas metade do ciclo da forma de onda de entrada.

Quando a tensão de entrada é positiva, o diodo fica diretamente polarizado e permite a passagem da corrente. No entanto, quando a tensão de entrada se torna negativa, o diodo fica inversamente polarizado e bloqueia a passagem da corrente. Isto permite eliminar a parte negativa da forma de onda, como é possível observar na figura 3.15, que mostra a forma de onda na saída do retificador.

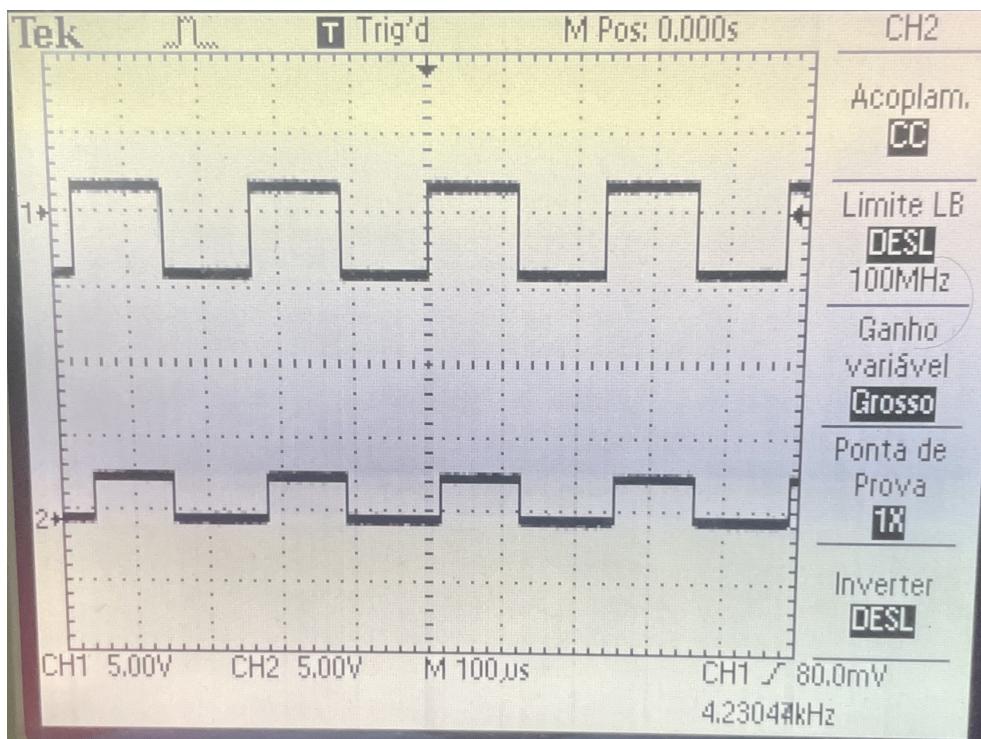


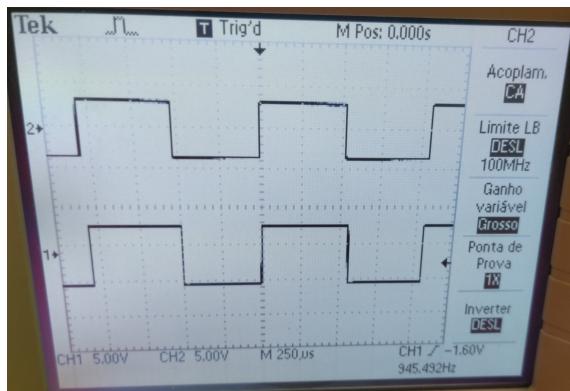
Figura 3.15: Forma de onda dos circuitos emissor e receptor (retificador).

3.2 Formas de onda dos circuitos emissor e recetor

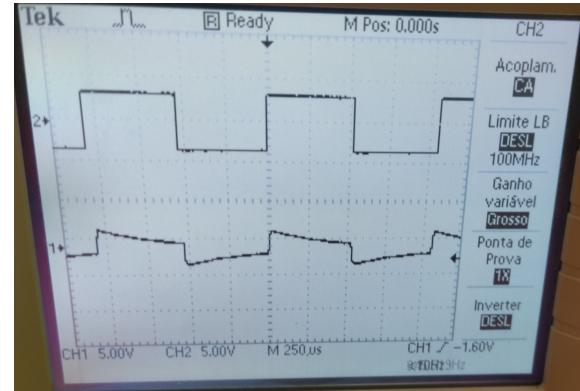
Após as montagens dos respetivos circuitos, o grupo procedeu à análise das formas de onda (sinais elétricos) nos pontos pertinentes dos circuitos emissor e recetor, das quais foram medidas e avaliadas com auxílio a um osciloscópio e, posteriormente, foram registadas fotograficamente.

A figura 3.16 ilustra as formas de onda do circuito emissor (onda acima do eixo horizontal) e do circuito recetor (onda abaixo do eixo horizontal) respetivamente, em diferentes modos de operação (não saturado e saturado).

Através da análise da figura, é possível verificar os parâmetros configurados para o canal dois do osciloscópio (CH2), tais como: acoplamento AC (*Alternating current*); limite de largura de banda a 100 MHz e ponta de prova colocada a "1X". Em contraste, a escala vertical está configurada para 5 V/Div e a escala horizontal a 250 us/Div.



(a) Formas de onda dos circuitos emissor e recetor (não saturado).



(b) Formas de onda dos circuitos emissor e recetor (saturado).

Figura 3.16: Formas de onda dos circuitos emissor e recetor.

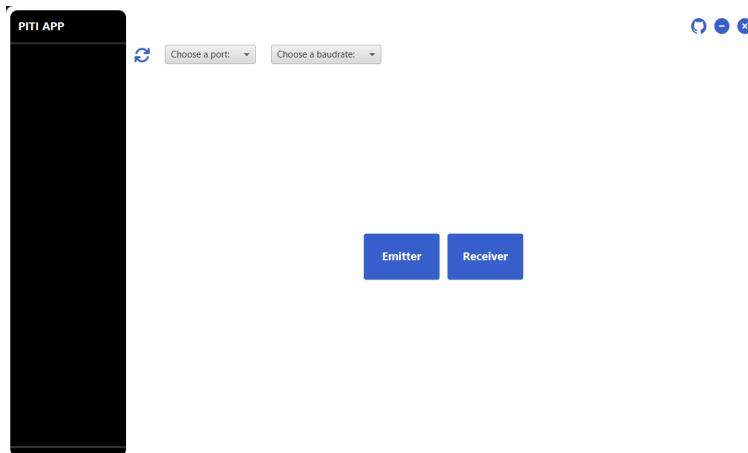
4 Interface gráfica da aplicação

A implementação das aplicações de interface gráfica com o utilizador nos PCs, inclui 3 menus:

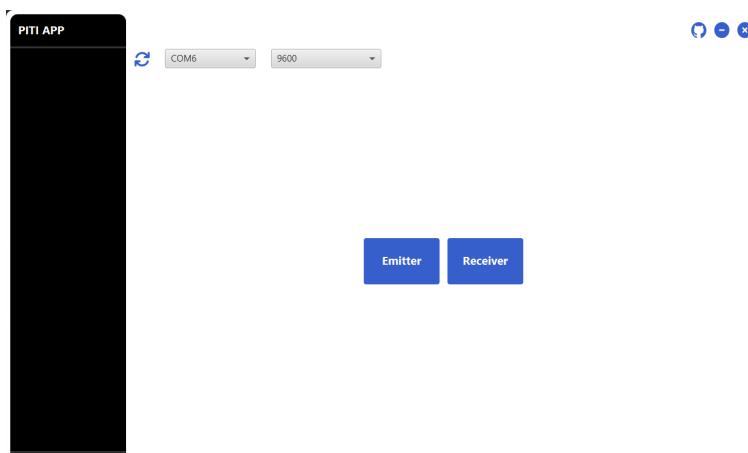
- Menu **Principal**: Para escolha entre os restantes menus, tanto como o *baudrate* e a porta a usar.
- Menu **Recetor**: Permite fazer ações relacionadas ao recetor.
- Menu **Emissor**: Permite fazer ações relacionadas com o emissor.

No menu principal, o utilizador tem de selecionar a porta COM por onde pretende comunicar, e a *baudrate* que pretende usar. Caso algum destes campos não tenha sido preenchido, o programa gera um alerta de erro, onde avisa o campo que está por preencher.

A figura 4.17 apresenta o menu principal em diferentes fases de execução.



(a) Menu Principal por preencher.



(b) Menu Principal preenchido.

Figura 4.17: Menu Principal.

No menu emissor, apresentado pela figura 4.18, o utilizador pode escrever mensagens, que serão depois enviadas. No fundo da página, consegue-se ver também o histórico de mensagens enviadas nesta sessão, permitindo assim ser possível ver e comparar a qualidade da transmissão e o tempo que demora a enviar a mensagem.

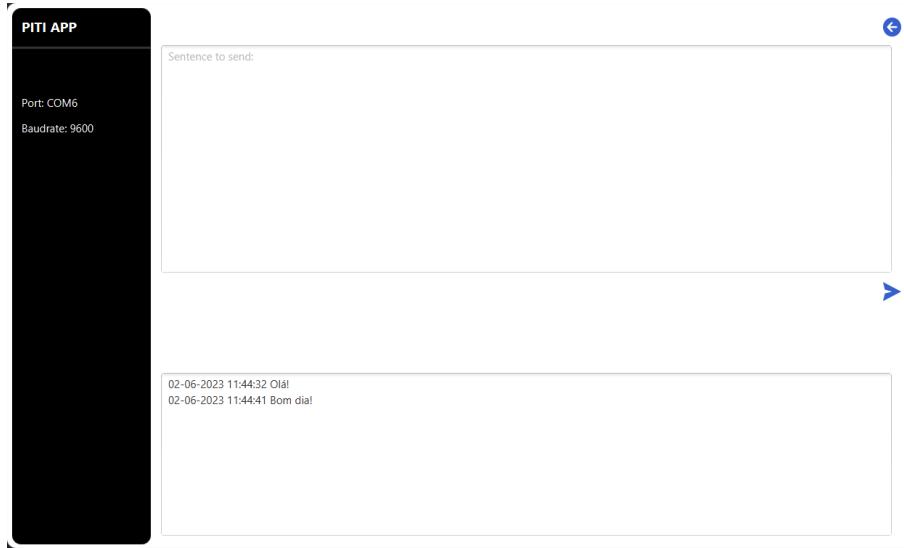


Figura 4.18: Menu Emissor.

No menu recetor, como mostra a figura 4.19, o utilizador consegue ver em tempo real as mensagens que estão a ser recebidas pelo circuito recetor e, de seguida, pela sua placa ESP32.

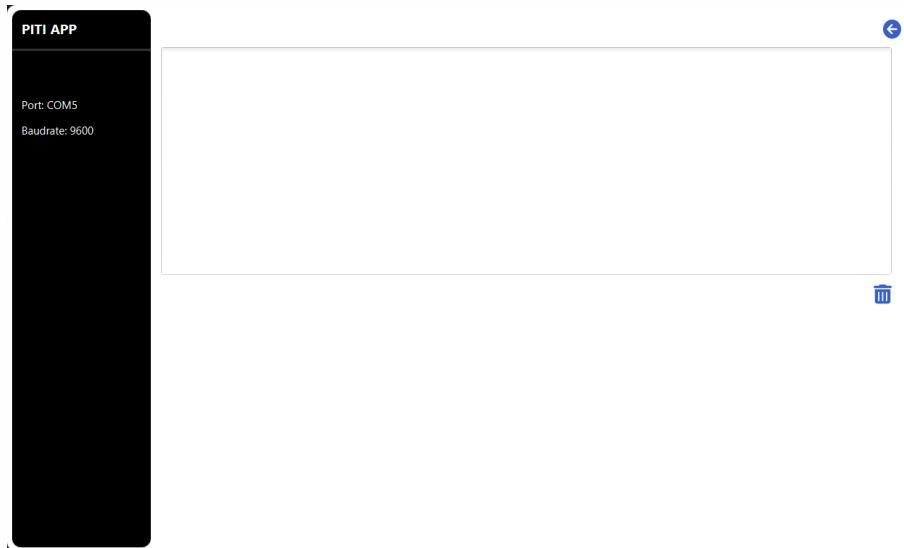


Figura 4.19: Menu Recetor.

Em termos de linguagens usadas, o interface gráfico é baseado na biblioteca JavaFX [5], na linguagem Java. Para o uso desta biblioteca, o uso de CSS e FXML foi necessário, ambos para obtermos o nível estético que pretendíamos no *design* da aplicação.

5 Testes efetuados

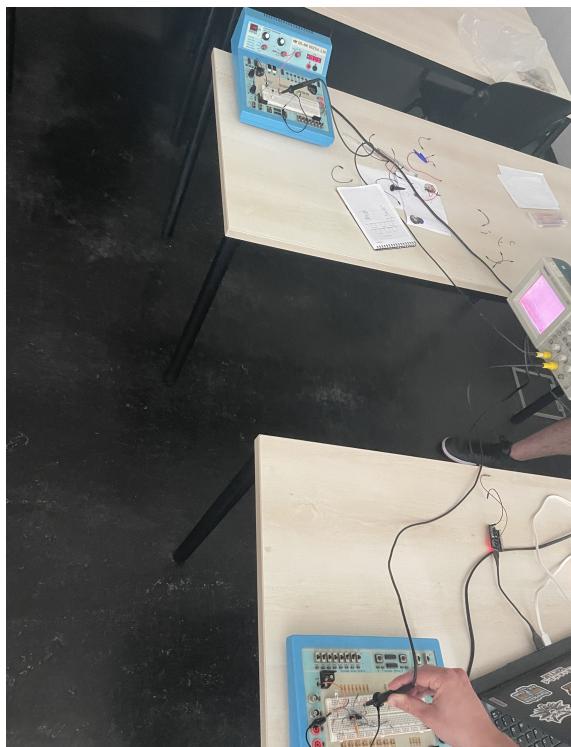
De forma a avaliar o sucesso e a eficácia da implementação dos circuitos, foram efetuados vários testes a fim de determinar se o sistema desenvolvido pelo grupo operava de forma expectável. Os testes desenvolvidos focaram-se no alcance do sinal ótico, emitido pelo circuito emissor.

Para determinar o alcance máximo do sinal ótico, o grupo aumentou de forma gradual, a distância entre os dois circuitos.

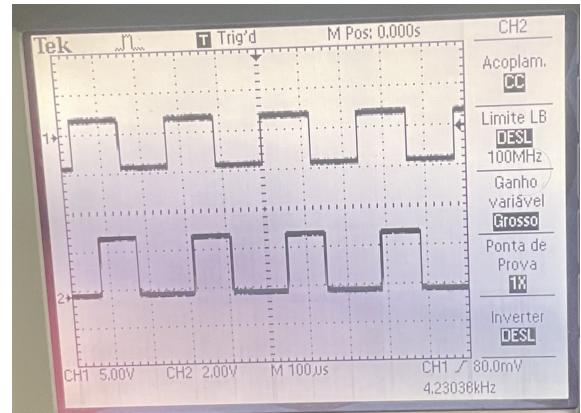
5.1 Teste de distância

Após a configuração do osciloscópio, o grupo afastou os circuitos emissor e recetor para verificar a forma de onda obtida.

A figura 5.20 ilustra a distância entre os circuitos emissor e recetor, em cerca de 175 cm e as respetivas formas de onda, sem as placas ESP32 estarem conectadas a ambos os circuitos.



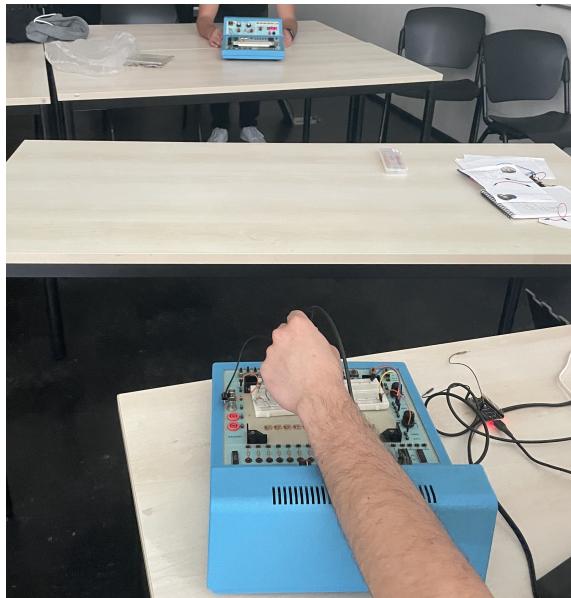
(a) Distância entre o recetor e o emissor (1,75m).



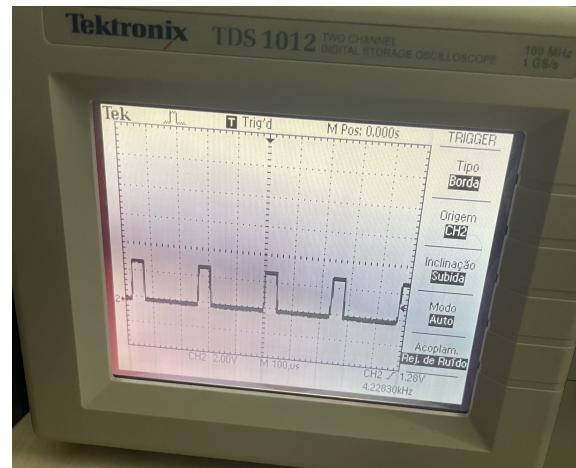
(b) Formas de onda dos circuitos emissor e recetor resultante.

Figura 5.20: Testes de distância entre os circuitos emissor e recetor.

Após a realização de vários testes, o grupo conseguiu obter uma forma de onda razoável a uma distância de 240 cm, que corresponde ao alcance máximo do sinal.



(a) Distância máxima entre o receptor e o emissor.



(b) Formas de onda dos circuitos emissor e recetor resultante.

Figura 5.21: Testes de distância entre os circuitos emissor e recetor (alcance máximo).

A tabela 5.1 apresenta os valores dos parâmetros avaliados (colunas da tabela), que foram registados no decorrer dos testes laboratoriais. Para diferentes distâncias até 130 cm entre os circuitos emissor e recetor, a dimensão da trama enviada foi a mesma (4 bytes).

Dimensão da trama (bytes) enviada	Dimensão da trama (bytes) recebida	Distância (cm)
4	4	60
4	4	83
4	4	97
4	4	113
4	4	123
4	4	130

Tabela 5.1: Tabela com distâncias dos testes efetuados.

Através da tabela 5.2 é possível observar que, para distâncias superiores a 130 cm, começam a surgir erros na transmissão.

Dimensão da trama (bytes) <u>enviada</u>	Dimensão da trama (bytes) <u>recebida</u>	Distância (cm)
4	3	150
4	3	160
4	2	170
4	2	175
4	1	180
4	1	190

Tabela 5.2: Tabela com distâncias dos testes efetuados nos quais houve erros de transmissão.

5.2 Teste de distância com obstáculos

Após a realização dos testes de distância sem a presença de obstáculos, testou-se novamente ambos os circuitos com a presença de um obstáculo no percurso do *link* ótico, com o intuito de determinar se o circuito receptor seria capaz de detetar o respetivo *link*.

A figura 5.22 ilustra a forma de onda do circuito receptor, na presença de um obstáculo, como por exemplo, a colocação de uma folha branca a meio do percurso percorrido pelo sinal. Através da análise da forma de onda do circuito receptor, é possível visualizar que o receptor não é capaz de detetar o sinal ótico.

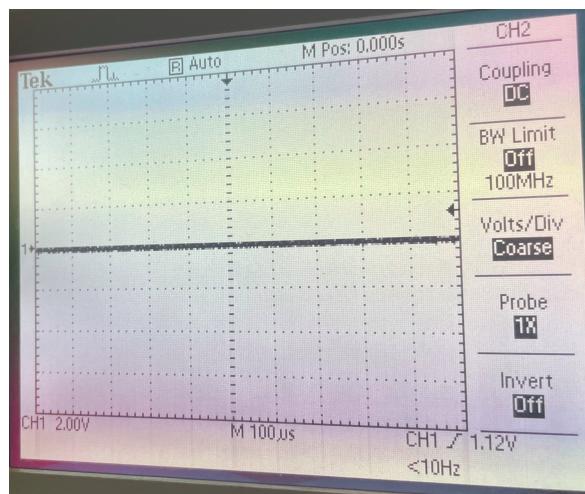


Figura 5.22: Forma de onda do circuito receptor.

6 Autoavaliação/Reflexão

Esta secção do relatório inclui uma descrição pormenorizada das contribuições efetivas de cada elemento do grupo no desenvolvimento do projeto. O conteúdo da lista é fundamentado nas tarefas do cronograma proposto. Para cada elemento do grupo é indicada a percentagem do seu contributo para cada tarefa.

- Rui Cunha:
 - a) Desenvolvimento da fase A
 - Escrita do Relatório de especificação Fase A (20%)
 - Elaboração do protocolo de comunicação (10%)
 - b) Desenvolvimento da fase B
 - Escrita do Relatório de especificação Fase B (20%)
 - Montagem do circuito (10%)
 - Testagem do circuito (10%)
 - c) Desenvolvimento da Fase C
 - Escrita do Relatório de especificação Final (20%)
 - Desenvolvimento do Ambiente Gráfico (10%)
 - Melhoramento de etapas anteriores (40%)
 - Testes Finais (20%)
- Francisco Martins:
 - a) Desenvolvimento da fase A
 - Escrita do Relatório de especificação Fase A(20%)
 - Elaboração do protocolo de comunicação (10%)
 - b) Desenvolvimento da fase B
 - Escrita do Relatório de especificação Fase B(20%)
 - Montagem do circuito (35%)
 - Testagem do circuito (35%)
 - c) Desenvolvimento da Fase C

- Escrita do Relatório de especificação Final(20%)
 - Desenvolvimento do Ambiente Gráfico (10%)
 - Melhoramento de etapas anteriores (15%)
 - Testes Finais (20%)
- Luís Oliveira:
 - a) Desenvolvimento da fase A
 - Escrita do Relatório de especificação Fase A (20%)
 - Elaboração do protocolo de comunicação (35%)
 - b) Desenvolvimento da fase B
 - Escrita do Relatório de especificação Fase B (20%)
 - Montagem do circuito (10%)
 - Testagem do circuito (10%)
 - c) Desenvolvimento da Fase C
 - Escrita do Relatório de especificação Final (20%)
 - Desenvolvimento do Ambiente Gráfico (60%)
 - Melhoramento de etapas anteriores (15%)
 - Testes Finais (20%)
 - José Gomes:
 - a) Desenvolvimento da fase A
 - Escrita do Relatório de especificação Fase A (20%)
 - Elaboração do protocolo de comunicação (35%)
 - b) Desenvolvimento da fase B
 - Escrita do Relatório de especificação Fase B (20%)
 - Montagem do circuito (10%)
 - Testagem do circuito (10%)
 - c) Desenvolvimento da Fase C
 - Escrita do Relatório de especificação Final (20%)

- Desenvolvimento do Ambiente Gráfico (10%)
 - Melhoramento de etapas anteriores (15%)
 - Testes Finais (20%)
- Diogo Cerqueira:
 - a) Desenvolvimento da fase A
 - Escrita do Relatório de especificação Fase A (20%)
 - Elaboração do protocolo de comunicação (10%)
 - b) Desenvolvimento da fase B
 - Escrita do Relatório de especificação Fase B (20%)
 - Montagem do circuito (35%)
 - Testagem do circuito (35%)
 - c) Desenvolvimento da Fase C
 - Escrita do Relatório de especificação Final (20%)
 - Desenvolvimento do Ambiente Gráfico (10%)
 - Melhoramento de etapas anteriores (15%)
 - Testes Finais (20%)

7 Conclusão

Concluído o projeto proposto, é inegável o carácter didático do mesmo.

A realização deste trabalho permitiu ao grupo consolidar o conhecimento, de forma prática, relativo aos protocolos de comunicação, a nível das placas ESP32 como entre o PC e a placa, e mecanismos de codificação e descodificação dos *bits* no *link* óptico.

Em algumas ocasiões, surgiram algumas dificuldades, nomeadamente, na implementação dos circuitos do emissor e do receptor, visto que foi necessário relembrar certos conceitos do espetro da eletrónica.

Em contrapartida, a resolução do projeto contribuiu para adquirir conhecimento e competências em várias áreas, incluindo: comunicações óticas em espaço livre; modulação e codificação; deteção e correcção de erros; protocolos de ligação e aplicações em rede.

Referências Bibliográficas

- [1] Oracle. What is java technology and why do i need it? URL https://www.java.com/en/download/help/whatis_java.html.
- [2] Java Native. Java library for talking to serial ports (with added build support for maven, cmake, msvc). URL <https://github.com/java-native/jssc>.
- [3] Arduino. Language reference. URL <https://www.arduino.cc/reference/en/>.
- [4] DesignSoft. Circuit simulator for analog, digital, mcu and rf circuits. URL <https://www.tina.com>.
- [5] OpenJFX. Java library for ui design. URL <https://github.com/openjdk/jfx>.