INSTITUTO FEDERAL DE SANTA CATARINA

RAFAEL TELES ESPINDOLA

Experimento de transmissão de dados através de ondas sonoras utilizando FSK

São José - SC

Julho/2023

EXPERIMENTO DE TRANSMISSÃO DE DADOS ATRAVÉS DE ONDAS SONORAS UTILIZANDO FSK

Monografia apresentada ao Curso de Engenharia de Telecomunicações do campus São José do Instituto Federal de Santa Catarina para a obtenção do diploma de Engenheiro de Telecomunicações.

Orientador: Prof. Roberto Wanderley da Nóbrega, Dr.

São José - SC

Julho/2023

Rafael Teles Espindola

Experimento de transmissão de dados através de ondas sonoras utilizando FSK

Este trabalho foi julgado adequado para obtenção do título de Engenheiro de Telecomunicações, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

São José - SC, 6 de julho de 2023

Prof. Roberto Wanderley da Nóbrega, Dr.
Orientador
Instituto Federal de Santa Catarina

Marcelo Maia Sobral, Dr.
Instituto Federal de Santa Catarina

Mario de Noronha Neto, Dr.
Instituto Federal de Santa Catarina

RESUMO

Como forma alternativa à comunicação através de ondas eletromagnéticas, este trabalho tem como objetivo realizar um experimento capaz de realizar uma comunicação ponto-a-ponto usando um canal sem-fio através do som e até mesmo um caráter educacional a estudantes de telecomunicações devido a sua forma simples de se obter esse resultado. A camada física utilizada possui caraterísticas como a baixa taxa de dados, baixo alcance, e ausência de detecção de acesso ao meio. A comunicação será digital, half-duplex e assíncrona. Não será utilizado nenhuma comunicação utilizando antenas, mas sim apenas um par de microfones e alto-falantes. Para a modulação será utilizado a técnica de modulação FSK que consiste em associar uma frequência a um símbolo que represente um ou uma sequência de bits. A aplicação final que permitirá a troca de mensagens entre os usuários não é o foco, mas sim uma exemplificação de como pode ser utilizado o protocolo.

Palavras-chave: Transferência de dados por ondas sonoras. Modulação FSK. Protocolo de comunicação.

ABSTRACT

As an alternative to communications through electromagnetic waves, this work aims to perform an experiment capable of establishing a peer-to-peer link using a wireless channel through sound, and even have an educational character for telecommunications students due to its simple way of achieving this result. The physical layer used has characteristics such as low data rate, low range, and absence of medium access detection. The communication will be digital, full-duplex, and asynchronous. No communication using antennas will be used, only a pair of microphones and speakers. The modulation technique used will be FSK modulation, which consists of associating a frequency with a symbol representing one or a sequence of bits. The final application that allows message exchange between users is not the focus, but rather an exemplification of how the protocol can be used.

Keywords: Data transfer by sound waves. FSK modulation. Communication protocol.

LISTA DE ILUSTRAÇÕES

Figura 1 – Mulher falando 28
Figura 2 – Efeito aliasing 29
Figura 3 – Modulação FSK 31
Figura 4 – S2C 32
Figura 5 – S2C 33
Figura 6 – Simples experimento half-duplex 35
Figura 7 – Fluxograma do transmissor 36
Figura 8 – Símbolo 39
Figura 9 – Símbolos em sequência sem suavização 39
Figura 10 – Máscara 40
Figura 11 – Símbolos suavizados em sequência 40
Figura 12 – Suavização do sinal de áudio do símbolo 40
Figura 13 – Símbolo do canal 1 com 430 Hz 41
Figura 14 – Símbolo do canal 2 com 1938 Hz 41
Figura 15 – Pacote em áudio 42
Figura 16 – Diagrama do receptor 43
Figura 17 – Fluxograma do receptor 48

LISTA DE QUADROS

LISTA DE TABELAS

Tabela 1 – Representação do quadro esquematizado na transmissão 37
Tabela 2 – Frequências de cada canal e seus respectivos símbolos 38
Tabela 3 – Tabela de testes 46
Tabela 4 – Tabela de testes 47

LISTA DE CÓDIGOS

Código C.1 – Transmissor 63
Código D.1 – Receptor 67

LISTA DE ABREVIATURAS E SIGLAS

BFSK Binary Frequency Shift Keying.

CPFSK Continuous Phase Frequency Shift Keying.

CRC Cyclic Redundancy Check.

CRC16 Cyclic Redundancy Check 16.

FFT Fast Fourier Transform.

FSK Frequency Shift Keying.

IF Frequência intermediária.

RF Radiofrequência.

LISTA DE SÍMBOLOS

Hz Hertz (unidade de medida de frequência).

MHz Mega Hertz (unidade de medida de frequência, equivalente a 1 milhão de Hertz).

SUMÁRIO

1 INTRODUÇÃO . . . 23
1.1 Objetivos . . . 24
1.1.1 Objetivo geral . . . 24
1.1.2 Objetivos específicos . . . 24
1.2 Organização do texto . . . 25

2 FUNDAMENTAÇÃO TEÓRICA . . . 27
2.1 Som . . . 27
2.2 Amostragem e o Teorema de Nyquist . . . 28
2.3 Modulação . . . 29
2.3.1 Modulação FSK . . . 30
2.4 CRC16 . . . 32
2.5 Modem acústico . . . 32

3 DESENVOLVIMENTO . . . 35
3.1 Sobre o experimento . . . 35
3.2 Transmissor . . . 36
3.3 Receptor . . . 39

4 EXPERIMENTOS E RESULTADOS . . . 45
4.1 Condições gerais para todos os testes . . . 45
4.2 Teste 1: Duração do símbolo . . . 45
4.3 Teste 2: Taxa de pacotes recebidos livre de erros . . . 47
4.4 Teste 3: Transmissão simultânea . . . 48

5 CONCLUSÕES . . . 51
5.1 Trabalhos futuros . . . 51

REFERÊNCIAS . . . 53

APÊNDICES 55

APÊNDICE A – TESTE 1 . . . 57

APÊNDICE B – TESTE 2 . . . 61

APÊNDICE C – CÓDIGO DO TRANSMISSOR 63

APÊNDICE D – CÓDIGO DO RECEPTOR 67

1 INTRODUÇÃO

A comunicação sem fio tem desempenhado um papel fundamental na sociedade moderna, permitindo a transferência de informação a distância de forma rápida, confiável e eficiente. As tecnologias atuais nos permitem transmitir dados sem fio através de ondas eletromagnéticas com altas taxas de transmissão, como nos padrões Wi-Fi e Bluetooth, e também a grandes distâncias, como no padrão LoRa ou na comunicação via satélites. O problema de pesquisa que este trabalho se propõe a abordar é um experimento de estabelecer uma comunicação sem fio entre dois dispositivos através de ondas sonoras de forma alternativa às ondas eletromagnéticas.

É sensato questionar por que utilizar ondas sonoras em vez das ondas eletromagnéticas convencionais. As ondas mecânicas, como o som, não são eficientes em atravessar objetos e se atenuam mais rapidamente no ar. À primeira vista, isso pode parecer desvantajoso, mas na realidade depende do objetivo da aplicação. O fato de não atravessar paredes pode ser vantajoso em uma situação em que se deseja que a comunicação seja restrita a uma pequena sala. Além disso, o fato de funcionar apenas em curtas distâncias permite que a mesma faixa de frequência seja reutilizada em locais próximos, evitando interferências em longas distâncias. Transmitir dados por meio do som também oferece uma alternativa viável em locais onde o uso de ondas eletromagnéticas é restrito devido a problemas de interferência. Além disso, ondas eletromagnéticas não são tão boas quanto o som para atravessar a água em grandes distâncias, o que torna o som uma opção melhor para a comunicação submarina. Por fim, há a possibilidade de reaproveitar o hardware simples e já existente em muitos equipamentos como em uma rede de sensores. Essa prática não é inovadora e já é utilizada em sonares de submarinos, assim como em exemplos encontrados online (RYPULA, 2015) e até mesmo em produtos de varejo, como o Chromecast (SEPPALA, 2014) e uma câmera da Intelbras.

Dentre as diversas técnicas utilizadas nesse contexto, destaca-se neste trabalho uma maneira não inovadora, porém autêntica e original em sua construção, de comunicação sem fio, que utiliza o som como meio físico para o transporte da informação. Este experimento permite a transmissão de dados entre dispositivos sem a necessidade de utilizar ondas eletromagnéticas como a maioria das tecnologias mais conhecidas e convencionais de transferência de dados.

A escolha desse tema está intrinsecamente relacionada ao enfoque do curso, que busca explorar as possibilidades de comunicação a distância. No âmbito do curso de Engenharia de Telecomunicações, explora-se esse experimento que envolve o uso de microfones e alto-falantes para realizar uma comunicação a distância por meio do som. Essa abordagem

representa uma interessante forma de realizar essa transmissão, uma vez que lida-se com um meio físico para a transmissão de informações não convencional e perceptível aos humanos.

Nesse contexto, nossos objetivos consistem em realizar uma troca de informação útil entre os dispositivos envolvidos utilizando-se a ténica de modulação Frequency Shift Keying (FSK) para modular os dados transmitidos, tendo um mínimo de confiabilidade na informação recebida utilizando a técnica de Cyclic Redundancy Check (CRC), e estabelecer uma comunicação full-duplex, possibilitando a transmissão e recepção simultâneas de dados. Busca-se alcançar esses objetivos mesmo diante das limitações inerentes ao uso do som como meio de transmissão, tais como baixas largura de banda e a alta susceptibilidade a ruídos.

Para atingir esses objetivos, a metodologia envolverá a implementação da parte transmissora e receptora de comunicação por som. A parte transmissora deverá transformar a informação da aplicação em um áudio na qual espera-se, sem garantia nenhuma, que a parte receptora receba o sinal em boas condições e o decodifique. Tais limitações podem afetar a qualidade e a confiabilidade da comunicação estabelecida.

Apesar de não apresentar novas contribuições para a área, este trabalho identificará direções para pesquisas futuras e poderá fornecer insights valiosos para a continuidade do experimento de comunicação sem fio por som além de ser uma demonstração para aguçar a curiosidade de entusiastas e estudantes da área já que de forma sucinta demonstra fundamentos importantes sobre as telecomunicações além de trazer um fator mais tátil que é o som em frequência audível se comparado às ondas eletromagnéticas nas quais precisamos utilizar instrumentos/equipamentos intermediadores para a sua detecção.

1.1 Objetivos

1.1.1 Objetivo geral

Desenvolver um experimento de transmissão de dados utilizando FSK para uma comunicação sem fio que utilize ondas sonoras como meio físico de transporte da informação.

1.1.2 Objetivos específicos

- Criar um transmissor FSK.
- Criar um receptor FSK.
- Criar um experimento simples e ponto-a-ponto para camada física.

- Utilizar uma técnica de detecção de erros para prover um mínimo de confiabilidade na comunicação.
- Transferir dados entre os dispositivos de forma full-duplex e assíncrona.

1.2 Organização do texto

No Capítulo 2 é apresentada a fundamentação teórica, no Capítulo 3 é esclarecido a forma como foi desenvolvida as partes receptora e transmissora do experimento e metodologia para este projeto, no Capítulo 4 é apresentado como foram realizados os experimentos práticos do projeto e seus resultados. Por fim, no Capítulo 5 são apresentadas as conclusões sobre este trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo apresenta-se informações e conceitos fundamentais para o bom entendimento do experimento realizado. Primeiro, é necessário compreender um pouco das características do som, por ser o meio físico escolhido para o transporte da informação. Em seguida, são abordados tópicos como a amostragem e o Teorema de Nyquist, fundamentais para entender como o som é digitalizado corretamente. Por fim, é apresentada uma revisão sobre a modulação e a técnica de detecção de erros escolhida, ressaltando sua relevância no processo.

2.1 Som

A definição mais geral de som é a de uma onda longitudinal que se propaga em um meio material. Pode se propagar no meio gasoso, líquido ou sólido. As ondas sonoras mais simples são as ondas senoidais, as quais possuem valores definidos para a amplitude, a frequência e o comprimento de onda. O ouvido humano é sensível aos sons com frequências compreendidas entre 20 e 20000 Hz, que delimitam o intervalo audível, mas usamos também a palavra som no caso de frequências maiores (ultrassom) ou frequências menores (infrassom) do que os limites do intervalo audível segundo Young e Freedman (2012).

Segundo Donoso (2006) o som é uma sensação auditiva que nossos ouvidos são capazes de detectar. Esta sensação é produzida pelo movimento organizado das moléculas que compõem o ar. Ao bater num diapasão provocamos uma perturbação que faz vibrar o ar e que se propaga até ser captada por nossos ouvidos, constituindo o que chamamos de som.

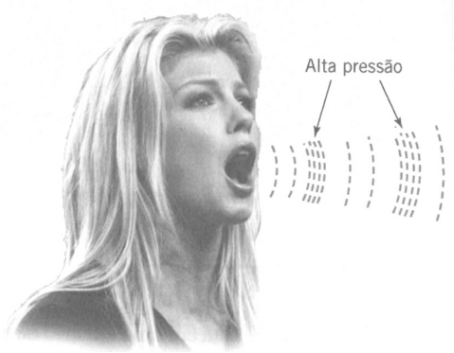
A Figura 1 mostra, de forma esquemática, o aspecto de uma onda sonora depois de deixar a boca de uma pessoa. Os traços representam moléculas de ar. Em algumas regiões, as moléculas estão mais concentradas; em outras estão mais rarefeitas. São estas regiões de compressão e rarefação que viajam pelo ar e constituem a onda sonora.

O som possui algumas características importantes para este experimento como frequência, amplitude e forma de propagação.

- **Frequência:** A onda sonora é caracterizada pela sua frequência (f), que corresponde ao número de vibrações por segundo (medida em hertz, Hz). Essa característica que traz a sensação de grave e agudo aos ouvidos.

- **Amplitude:** É a qualidade que permite ao ouvido diferenciar os sons fracos dos sons fortes. A unidade de medida do som é o decibel (dB). O decibel é uma

Figura 1 – Mulher falando



Fonte: Donoso (2006)

unidade logarítmica usada para expressar a relação entre duas quantidades, como a intensidade do som em relação a um valor de referência.

- **Forma de propagação:** As ondas mecânicas precisam de um meio para propagação. Elas não se propagam no vácuo. Nesse caso as ondas sonoras são propagadas pelo ar. As moléculas de ar vibram na direção em que o som se propaga e por isso são chamadas ondas longitudinais. E no ar possuem uma velocidade média de 320 m/s.

É importante ressaltar que um som para ser gerado e transmitido por um alto-falante, ou captado e recebido por um microfone, deve estar dentro das especificações do determinado hardware que o gera ou detecta. Os microfones e alto-faltantes possuem características intrínsecas à sua construção e materiais utilizados, na qual determinam quais frequências estão aptos a captarem e emitirem.

2.2 Amostragem e o Teorema de Nyquist

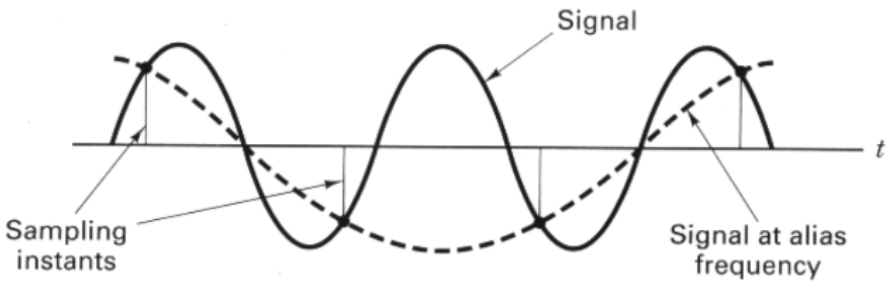
O sinal sonoro na natureza é contínuo e para conseguir ser processado deve ser transformado em um sinal discreto. Para transformar um sinal contínuo em discreto é necessário realizar a amostragem que é basicamente uma coleta periódica do sinal contínuo. Cada coleta é armazenada e esse conjunto de amostras formam uma cópia discreta do sinal original. Como a amostragem obtém apenas algumas amostras no tempo, isso significa que ela gera uma perda de informação em relação ao sinal original. Porém ainda sim é uma excelente aproximação que nos permite processá-la digitalmente e recuperar um cópia do sinal original muito fiel.

Segundo Sklar (2001) a amostragem é uma técnica crucial para a codificação de sinais contínuos em sinais discretos. A taxa de amostragem é a quantidade de amostras por unidade de tempo e é um parâmetro crítico que afeta a fidelidade do sinal codificado. De acordo com o Teorema de Nyquist, a taxa de amostragem deve ser pelo menos duas

vezes maior que a maior frequência presente no sinal original para garantir que não haja distorção.

De acordo com Sklar (2001) a frequência de amostragem é um parâmetro que está relacionada com a qualidade do sinal amostrado. Um sinal analógico pode ser derivado em senoides de diferentes frequências somadas. Para amostrar um sinal fielmente é necessário conseguir amostrar todas as senoides e isso inclui desde a menor até a sua de maior frequência. Se a senoide de maior frequência for amostrada com sucesso, a de menor também será. Para amostrar sua senoide de maior frequência é necessário no mínimo duas amostras por período, ou seja, deve-se seguir o teorema de Nyquist. Caso a frequência de amostragem não seja suficientemente alta, ocorrerá o efeito aliasing ilustrado na Figura 2.

Figura 2 – Efeito aliasing



Fonte: Sklar (2001)

2.3 Modulação

De acordo com Sklar (2001), a modulação digital é o processo pelo qual os símbolos são transformados em formas de onda compatíveis com as características do canal. No caso da modulação em banda base, essas formas de onda geralmente tomam a forma de pulsos moldados. Mas no caso da modulação em banda passante, os pulsos moldados modulam uma sinusoide chamada onda portadora, ou simplesmente uma portadora; para transmissão de rádio, a portadora é convertida em um campo eletromagnético para propagação para o destino desejado. Pode-se perguntar por que é necessário usar uma operadora para a transmissão de rádio de sinais de banda base. A resposta é a seguinte. A transmissão de campos eletromagnético através do espaço é realizada com o uso de antenas. O tamanho da antena depende do comprimento de onda e da aplicação. Para telefones celulares, as antenas são tipicamente de tamanho $\lambda/4$, onde o comprimento de onda é igual a c/f, e c, a velocidade da luz, é de 3×10^8 m/s. Considere enviar um sinal de banda base (digamos $f = 3000$ Hz) acoplando-o a uma antena diretamente sem uma onda portadora. Qual seria o tamanho da antena? Vamos dimensioná-lo usando o padrão da indústria telefônica de $\lambda/4$ como dimensão da antena. Para o sinal de banda base de 3000 Hz, $\lambda/4 = 25$ km

Transmitir um sinal de 3000 Hz através do espaço, sem modulação de ondas portadoras, seria necessária uma antena que se estende por 25 km. No entanto, se as informações da banda base forem primeiro moduladas em uma portadora de frequência mais alta, por exemplo, uma portadora de 900 MHz, o diâmetro equivalente da antena seria de cerca de 8 cm. Por esse motivo, a modulação de ondas portadoras ou passa-banda é um passo essencial para todos os sistemas que envolvem transmissão de rádio. No caso das ondas sonoras não é necessário uma antena, mas não é por isso que a modulação seria inútil. Uma vantagem dela, mas não exclusiva às ondas sonoras, é a possibilidade de realizar a multiplexação.

A modulação em banda passante pode fornecer outros benefícios importantes na transmissão de sinal. Se mais de um sinal utilizar um único canal, a modulação pode ser usada para separar os diferentes sinais. Tal técnica, conhecida como multiplexação por divisão de frequência. A modulação pode ser usada para minimizar os efeitos da interferência. A modulação também pode ser usada para colocar um sinal em uma faixa de frequência onde os requisitos de projeto, como filtragem e amplificação, podem ser facilmente atendidos. Este é o caso quando os sinais de Radiofrequência (RF) são convertidos para uma Frequência intermediária (IF) em um receptor.

2.3.1 Modulação FSK

A modulação Frequency Shift Keying (FSK) é uma técnica de modulação digital amplamente utilizada em sistemas de comunicação. Ela envolve a transmissão de dados digitais através de diferentes frequências de portadora, onde cada frequência representa um símbolo específico (SKLAR, 2001). Nesta seção, serão apresentadas as principais características, formulações e explicações sobre a técnica de modulação FSK

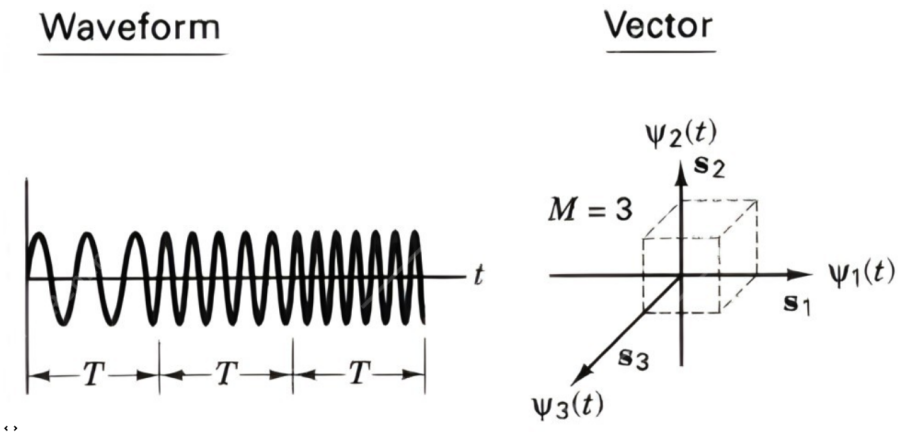
A modulação FSK é baseada na variação da frequência da portadora em resposta aos dados a serem transmitidos conforme ilustrada na Figura 3 mostra a típica mudança de frequência na transição de símbolos. Os símbolos de dados são mapeados para frequências de portadora distintas, onde cada frequência representa um símbolo específico.

Em sua forma mais simples, conhecida como Binary Frequency Shift Keying (BFSK), são utilizados apenas dois valores de frequência, chamados de frequência "espaço" (frequência baixa) e frequência "marca" (frequência alta), que representam os dois símbolos binários 0 e 1, respectivamente. Esse caso pode ser descrito matematicamente da seguinte forma:

$$\begin{aligned} s(t) &= A_c \cos(2\pi f_s t), \text{se o símbolo transmitido é 0} \\ s(t) &= A_c \cos(2\pi f_m t), \text{se o símbolo transmitido é 1} \end{aligned} \tag{2.1}$$

Onde:

Figura 3 – Modulação FSK



Fonte: Sklar (2001)

- A_c é a amplitude da portadora
- f_s é a frequência espaço
- f_m é a frequência marca
- t é o tempo

Além do BFSK também é possível realizar uma forma mais genérica de transmissão utilizando M frequências distintas. Este caso é conhecido como M-FSK. Se M for uma potência de 2, então possível transmitir $\log_2 M$ bits por símbolo. Por exemplo, se for desejado transmitir 4 bits por símbolo, então é necessário ter disponível $16 = 2^4$ frequências, sendo cada frequência um símbolo e cada símbolo um conjunto de 4 bits específicos.

A modulação FSK possui diversos pontos fortes e fracos em sua aplicação. Entre os pontos fortes, destacam-se sua simplicidade de implementação, tornando-a mais acessível e econômica em termos de hardware. Além disso, a FSK é menos suscetível a interferências de ruídos de amplitude, o que melhora a qualidade da transmissão em ambientes ruidosos. Sua tolerância a distorções também é uma vantagem, permitindo melhor desempenho em canais de comunicação com problemas de propagação e desvanecimento. A alta imunidade a interferências seletivas é outro aspecto positivo, garantindo uma transmissão mais robusta em ambientes com múltiplas fontes de transmissão próximas.

Por outro lado, a modulação FSK possui algumas limitações. Em termos de eficiência espectral, a FSK é menos eficiente do que técnicas mais avançadas, como a modulação por amplitude e fase (QAM), pois transmite menos bits por Hertz de largura de banda. Ou seja, em comparação com outras modulações digitais, a FSK pode requerer largura de banda mais ampla para transmitir a mesma quantidade de informações, o que pode ser uma desvantagem em sistemas com restrições de banda.

Quanto à utilização do som como meio de transmissão físico, a modulação FSK é frequentemente empregada em aplicações de telecomunicação, como a transmissão de dados por modems acústicos. Nesse contexto, os dados digitais são convertidos em sinais sonoros modulados por FSK, e esses sinais são transmitidos através de meios de comunicação que permitem a propagação de ondas sonoras, como a linha telefônica. Na recepção, o sinal acústico é demodulado e decodificado para recuperar os dados originais. Essa abordagem é comum em sistemas de comunicação entre computadores e dispositivos móveis que não possuem acesso direto à internet ou em situações em que a transmissão por cabos ou ondas de rádio não é viável. No entanto, é importante notar que, em comparação com outras formas de transmissão, o uso do som pode apresentar limitações em termos de velocidade de transmissão e alcance efetivo, tornando-a mais adequada para aplicações específicas e de alcance limitado.

2.4 CRC16

O Cyclic Redundancy Check 16 (CRC16) é um algoritmo amplamente utilizado em telecomunicações para detecção de erros em transmissões de dados. Ele calcula um valor de verificação de 16 bits que é anexado aos dados transmitidos. O receptor então recalcula o valor de verificação e o compara com o valor recebido. Se os valores forem diferentes, isso indica a presença de erros na transmissão (PETERSON; BROWN, 1975).

2.5 Modem acústico

Esse experimento tenta replicar uma técnica já conhecida e utilizada em mercado. Os modens acústicos são uma das maneiras de comunicação submarina. A empresa Evo Logics possui alguns modens dessa categoria, como por exemplo o modelo S2C da Figura 4.

Figura 4 – S2C

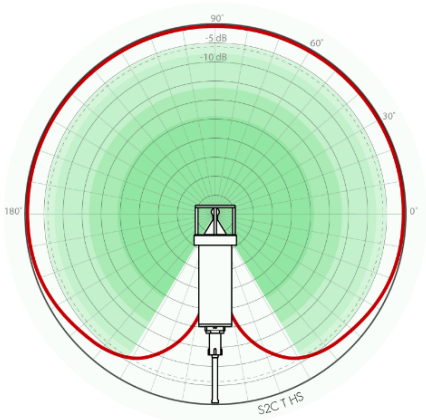
Fonte: Logics (2018)

O modelo S2C possui as seguintes características citadas abaixo:

- Alta frequência para transmissões mais rápidas
- Ultra leve e compacto para aplicações sensíveis a isso

- O padrão do feixe é ideal para transmissões em canais verticais e inclinados como a Figura 5 ilustra
- Alcance operacional: 300 m
- Banda utilizada: 120-180 kHz
- Padrão de propagação: Omnidirecional
- Transmissão: até 62.5 kbit/s
- BER: Menor que 10^{-10}

Figura 5 – S2C



Fonte: Logics (2018)

3 DESENVOLVIMENTO

Este capítulo está organizado da seguinte forma: Na seção 3.1 é apresentado uma introdução sobre o experimento onde ilustra da mais simples forma um esboço de como poderia ser realizado inicialmente. Na seção 3.2 é apresentada toda a parte transmissora que é a responsável por emitir as ondas sonoras. Na seção 3.3 é apresentada toda a parte receptora que a responsável pelo monitoramento do som e detecção das informações.

3.1 Sobre o experimento

Um primeiro esboço para se entender inicialmente o experimento é representado na Figura 6 onde ilustra um transmissor enviando um áudio ao receptor.

Figura 6 – Simples experimento half-duplex

Fonte: Desenvolvida pelo autor

A função principal do experimento é realizar um envio de informação do transmissor ao receptor. A informação em si pode ser qualquer coisa codificada em bytes e, em termos práticos, não importa seu tamanho pois será entregue por partes. O transmissor tem o objetivo de codificar os bytes em som e os transmitir enquanto o receptor tem o papel de analisar o som e decodificar a informação nela contida. Não há garantias de entrega nesse experimento. Cabe a quem for utiliza-lo garantir as seguintes condições físicas para uma experiência satisfatória:

- Um ambiente silencioso e sem reverberação
- Uma distância curta de aproximadamente 1 metro entre o receptor e transmissor
- Um alto-falante para o transmissor
- Um microfone para o receptor
- Uma amplitude sonora equivalente a aproximadamente 4/5 da capacidade total de um notebook de uso genérico para o transmissor

- Uma sensibilidade no microfone ajustada equivalente a aproximadamente 3/5 da total de um notebook de uso genérico para o receptor

Com todas essas condições físicas atendidas e com o software funcionando plenamente, é esperado que seja possível o envio de informações do transmissor ao receptor.

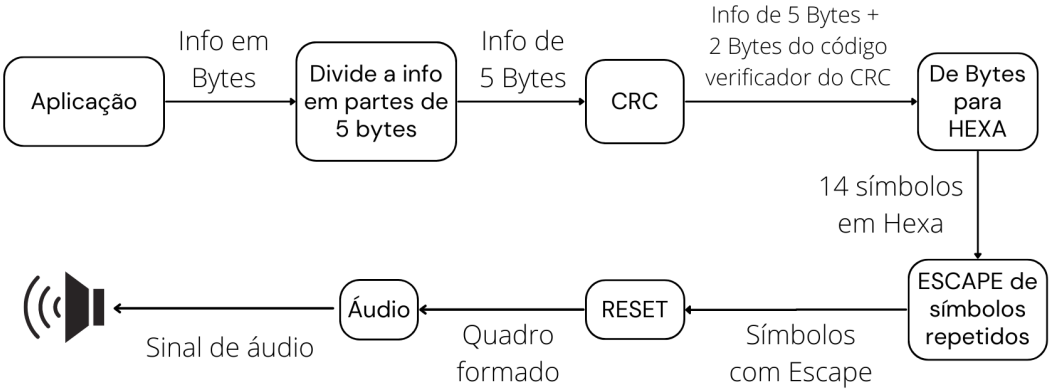
Esse transmissor e receptor ao longo desse experimento não se limitam a um notebook, assim como feito pelo autor, mas sim a qualquer hardware que consiga rodar o código e que tenha um alto-falante e um microfone que tenham em sua especificação as frequências utilizadas pelo software que inclusive podem ser alteradas para se adaptar ao hardware utilizado.

Esse experimento também tem o potencial de ser amplamente modificado para se adaptar a outras situações como por exemplo uma comunicação subaquática, para isso bastaria alterar o hardware para um microfone e alto-falante subaquáticos e com isso teria-se um sonar.

3.2 Transmissor

O transmissor é responsável por realizar a transformação dos dados de entrada em bytes em um formato de áudio capaz de ser decodificado pelo receptor, minimizando a chance de erros. Para alcançar este objetivo, o processo de transmissão consiste em algumas etapas e estratégias específicas ilustradas na Figura 7 e explicadas posteriormente.

Figura 7 – Fluxograma do transmissor



Fonte: Desenvolvida pelo autor

Primeiramente, a informação que se deseja transmitir é convertida em bytes na aplicação e entregue à parte transmissora. Em seguida a informação é dividida em partes de cinco bytes, e cada parte é sucedida por um trailer contendo seu respectivo CRC para formar um quadro. O quadro resultante, contendo informações e CRC, é transformado em símbolos, sendo que a cada quatro bits é necessário um símbolo. Os símbolos em sequência não podem ser repetidos, pois a recepção está preparada para detectar cada símbolo na

variação de frequência, e não na duração. Para evitar a repetição de símbolos, é utilizado um símbolo especial de escape para representar um símbolo repetido em sequência. Como os símbolos de informação são hexadecimais, variando de `'0'` a `'f'`, o símbolo de escape é definido como `'g'`. Por exemplo, uma sequência de símbolos como `3a3b3e2e3c900f` com escape ficaria assim: `3a3b3e2e3c90gf`. Repare que a sequência `"00"` foi alterada para `"0g"`. Após a aplicação da regra de escape, para delimitar o quadro, é adicionado um símbolo `'h'` no começo e no final, ficando assim, por exemplo: `h3a3b3e2e3c90gfh`. A Tabela 1 representa visualmente em ordem as partes, tamanho e símbolos de exemplo de um quadro.

Tabela 1 – Representação do quadro esquematizado na transmissão

Campo	Delimitador	Payload	CRC	Delimitador
Comprimento	1 Byte	De 1 a 5 Bytes	2 Bytes	1 Byte
Exemplo	`h`	`3a3b3e2e3c`	`90gf`	`h`

Um outro exemplo de mensagem para símbolo pode ser dado a seguir:

- **Mensagem:** `Olá pessoal`
- **Fragmentação:** `Olá p + essoa + al`
- **Exemplo com a primeira parte:** `Olá p`
  - **Codificação UTF-8:** `b'Ol\xc3\xa1 p'`
  - **Adição de CRC16:** `bytearray(b'Ol\xc3\xa1 p\xff\x05')`
  - **Tranformação para Hexadecimal:** `4f6cc3a12070ff05`
  - **Aplicação do Ecape:** `4f6cg3a12070fg05`
  - **Adição dos delimitadores:** `h4f6cg3a12070fg05h`

Resultado: `Olá p` = `h4f6cg3a12070fg05h`

Uma vez que a sequência de símbolos é definida, o próximo passo é gerar o áudio. Um dos parâmetros definidos para a criação do áudio é o canal escolhido pelo usuário. O canal é um conjunto de 18 frequências que serão utilizadas separadamente para representar os símbolos. Existem dois canais de comunicação implantados, um com frequências de 430 a 1161 Hz e outro de 1205 a 1938 Hz. A escolha de cada frequência foi realizada levando em conta o receptor, e cada frequência possui um intervalo de 40 a 46 Hz entre si que foram escolhidos empiricamente. Essa variação não é linear, pois foi percebida uma não linearidade na detecção do receptor, e foram necessários ajustes manuais para a detecção correta de cada frequência. Esses ajustes são necessários pois levam em conta o hardware específico da máquina utilizada no desenvolvimento, porém escolha ainda sim possui uma certa linearidade.

Cada frequência escolhida e seu símbolo de significado está descrito na Tabela 2.

Tabela 2 – Frequências de cada canal e seus respectivos símbolos

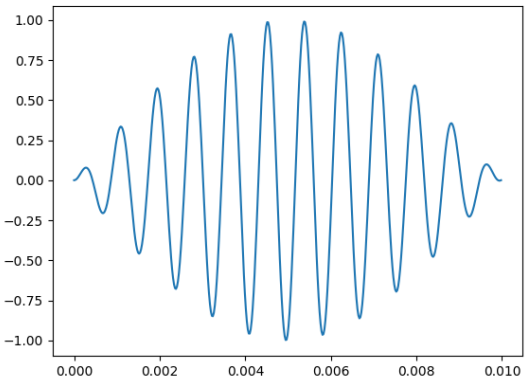
Frequência em Hz do Canal 1	**Frequência em Hz do Canal 2**	**Símbolo**
430.0	1205.0	`0`
474.0	1249.0	`1`
516.0	1292.0	`2`
562.0	1335.0	`3`
604.0	1378.0	`4`
646.0	1422.0	`5`
689.0	1465.0	`6`
733.0	1508.0	`7`
774.0	1550.0	`8`
819.0	1594.0	`9`
861.0	1637.0	`a`
905.0	1680.0	`b`
948.0	1723.0	`c`
991.0	1766.0	`d`
1033.0	1808.0	`e`
1076.0	1852.0	`f`
1120.0	1896.0	`g` (ESCAPE)
1161.0	1938.0	`h` (Delimitador)

Para cada símbolo é criado um áudio numa duração, intensidade e frequência. A taxa de amostragem é definida no momento da transmissão e ajustada para 44100 Hz. A duração mínima testada com sucesso para um símbolo ser detectável pela parte receptora é de 80 ms. Já a duração com 200 ms teve uma taxa de acertos melhor na recepção. Com um símbolo de 200 ms o transmissor possui uma taxa de transmissão de **5 símbolos por segundo**, lembrando que um quadro no máximo possui entre 1 e 5 bytes de informação útil (2 a 10 símbolos), mais 2 bytes de CRC (4 símbolos) e mais 2 símbolos delimitadores resultando em quadros que variam de 8 a 16 símbolos por conta da informação carregada.

A geração do áudio tem uma característica específica em seu formato. É necessário realizar um sinal que tenha um começo e fim suavizados em relação à amplitude como a Figura 8 demonstra porque os falantes geram ruídos caso o sinal tenha uma mudança súbita na fase do sinal, ou seja, quando há uma descontinuidade de fase como em destaque na Figura 9 que exibe dois símbolos em sequência, porém sem suavização. Para contornar esse problema foi realizado esse aumento e diminuição da amplitude de cada símbolo multiplicando todos por uma máscara exibida na Figura 10 para que os símbolos em sequência tenham um ponto em comum em relação à amplitude e não criem esse ruído indesejável, dessa forma dois símbolos em sequência ficam como na Figura 11

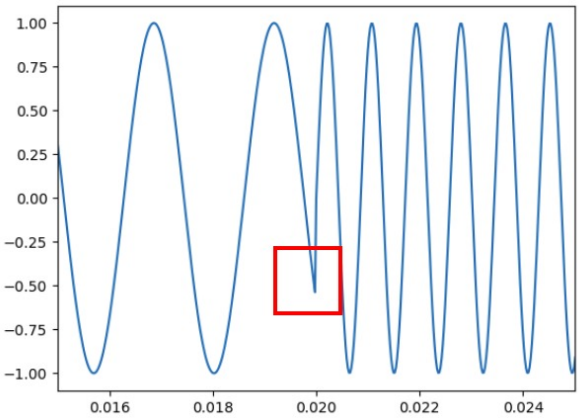
De forma resumida a Figura 12 ilustra o processo de suavização do sinal de um símbolo.

Figura 8 – Símbolo



Fonte: Desenvolvida pelo autor

Figura 9 – Símbolos em sequência sem suavização



Fonte: Desenvolvida pelo autor

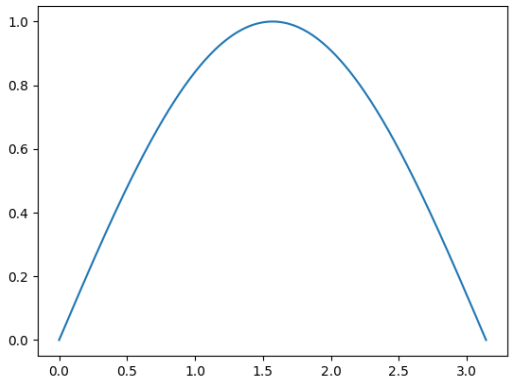
As figuras Figura 13 e Figura 14 demonstram respectivamente a menor e maior frequência transmitidas de um sinal. A menor frequência é do canal 1 com 430 Hz e a maior é do canal 2 com 1938 Hz.

Por fim, para exemplificar um áudio gerado a Figura 15 exibe um pacote completo com símbolo de duração 10 ms na qual carrega a informação útil `'a'`. Nele pode-se distinguir todos os 8 símbolos: `h61f782h`.

3.3 Receptor

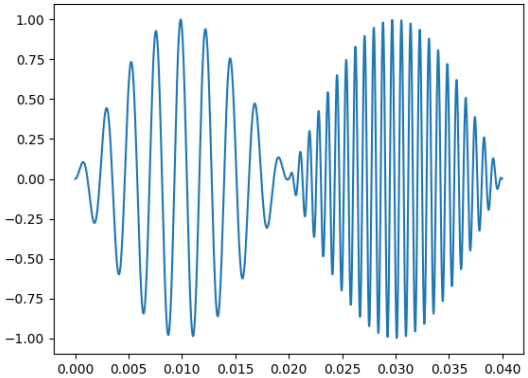
O receptor continuamente registra o áudio captado pelo microfone e o armazena em um buffer em uma taxa de amostragem de 44100 Hz. Esse buffer é consumido de 2048 em 2048 pontos para a análise do áudio. A cada análise é realizada a Fast Fourier Transform (FFT) nessa parte do áudio para detectar a frequência recebida. Mais de uma frequência pode ser detectada ao mesmo tempo e é função do receptor identificar se ela pertence

Figura 10 – Máscara



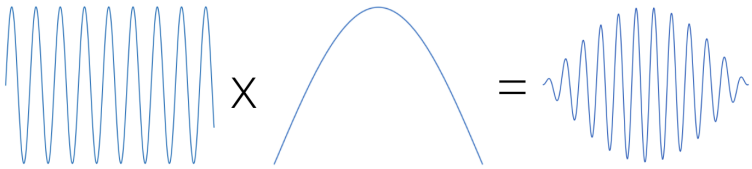
Fonte: Desenvolvida pelo autor

Figura 11 – Símbolos suavizados em sequência



Fonte: Desenvolvida pelo autor

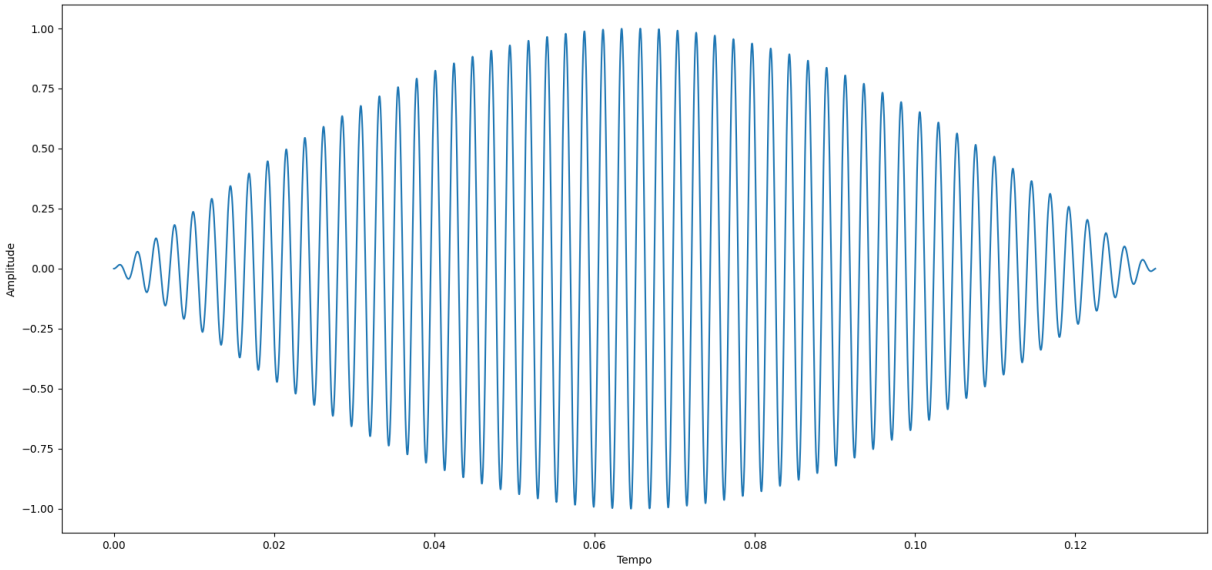
Figura 12 – Suavização do sinal de áudio do símbolo



Fonte: Desenvolvida pelo autor

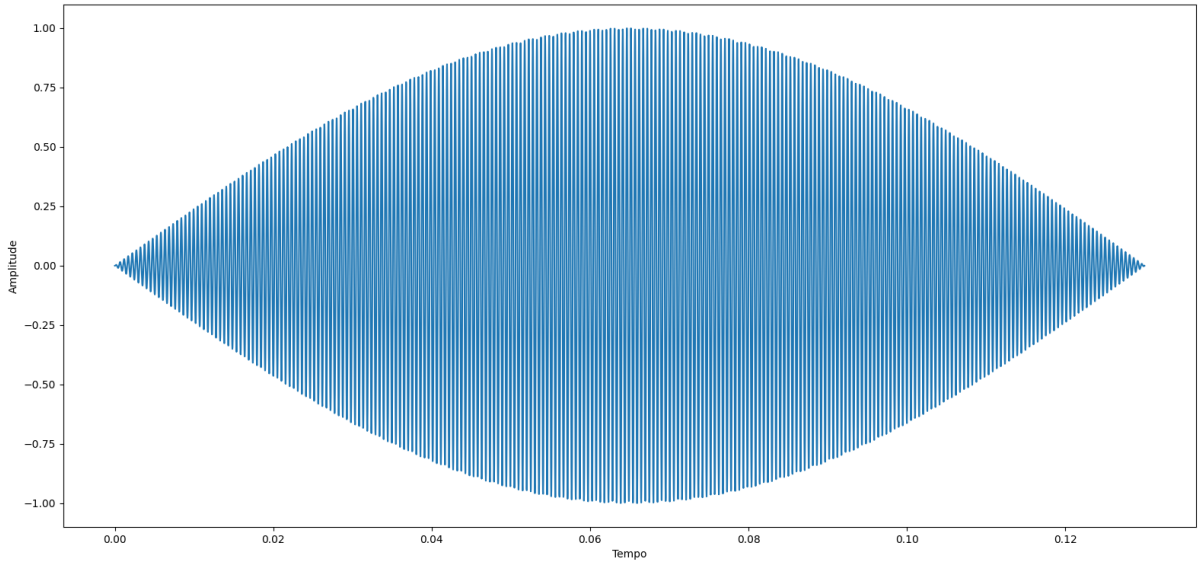
ao canal escolhido ou não para considerá-la como um símbolo. De todas as frequências detectadas as que serão consideradas como um símbolo precisam passar de um limiar de intensidade sonora na recepção, o qual foi escolhido empiricamente. Se mais de uma frequência for detectada e o canal de transmissão for o 1, a de frequência mais baixa será a escolhida como candidata a ser considerada como símbolo ou não, já se o canal for o 2, a segunda frequência mais grave será a escolhida como candidata. Já quando é detectado somente uma frequência então ela é sempre a candidata a ser ou não um símbolo.

Figura 13 – Símbolo do canal 1 com 430 Hz



Fonte: Desenvolvida pelo autor

Figura 14 – Símbolo do canal 2 com 1938 Hz



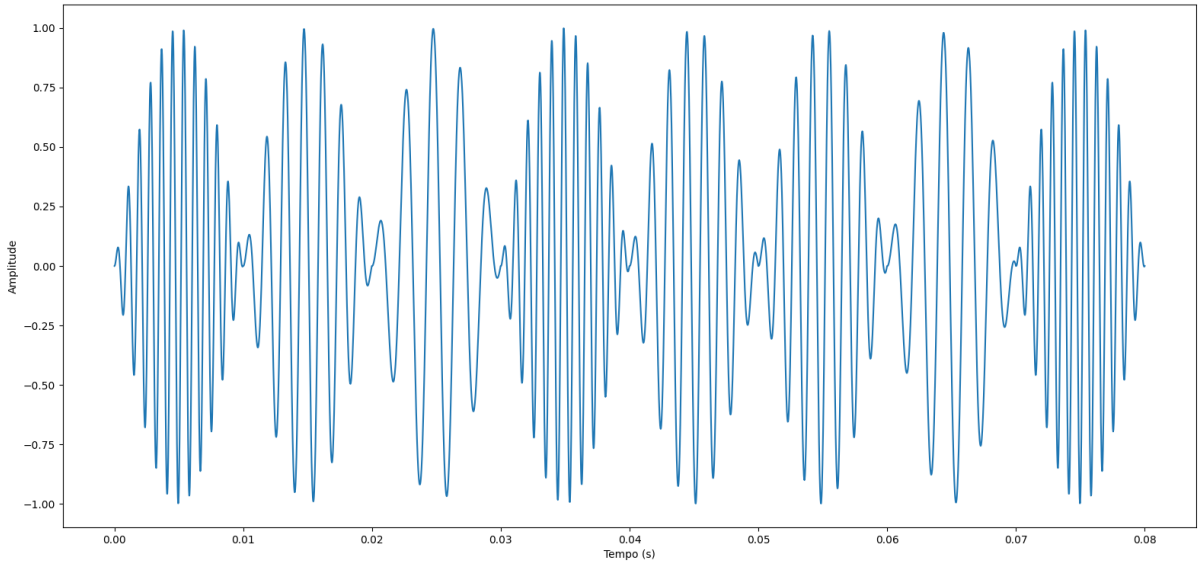
Fonte: Desenvolvida pelo autor

Uma frequência só prossegue na verificação se ela pertencer a um dos canais escolhidos para a comunicação, caso contrário ela é descartada.

Caso a frequência candidata esteja em um dos canais e seja diferente da última frequência detectada como símbolo, então estamos adiante de uma frequência que traz uma nova informação, um novo símbolo, já que a detecção dos símbolos se da na variação da frequência e não em uma duração específica.

Caso essa frequência carregue um símbolo de informação que vai de `'0'` a `'f'` em base haxadecimal, o símbolo é adicionado em um buffer, já que são necessários 2 símbolos

Figura 15 – Pacote em áudio



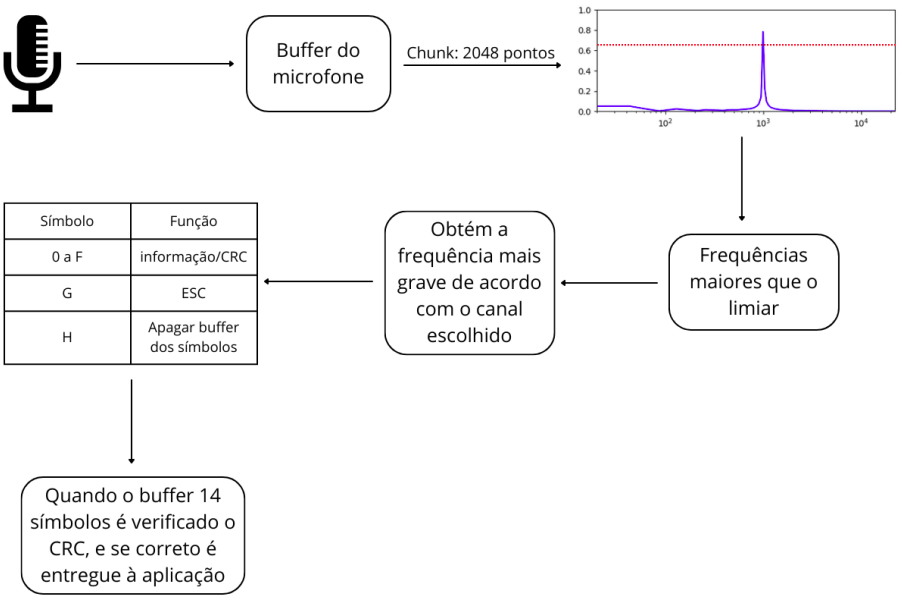
Fonte: Desenvolvida pelo autor

para formar um byte. Já se essa frequência for um símbolo de controle `'g'` que é o escape para símbolos repetidos em sequência, então caso o buffer tenha tamanho maior que 1, repete-se o último valor do buffer. Porém se a frequência recebida é o símbolo de controle `'h'`, apaga-se o buffer.

Após cada etapa dessa e quando o buffer tem tamanho de 6 a 14 (mínimo 1 bytes de informação mais 2 de CRC totalizando 6 símbolos, ou no máximo 5 bytes de informação mais 2 de CRC totalizando 14 símbolos), ainda é verificado se o que se contém nele é dado como válido fazendo uma verificação de CRC. Caso seja verdadeiro, imprime-se a mensagem na tela. Caso seja falso, não se faz nada e passa-se para o próxima repetição.

A Figura 16 esclarece de forma sucinta as principais partes do receptor.

Figura 16 – Diagrama do receptor



Fonte: Desenvolvida pelo autor

4 EXPERIMENTOS E RESULTADOS

Esse capítulo tem o objetivo verificar em testes práticos quais foram a menor duração útil de um símbolo ainda capaz de realizar boas transmissões, obter uma taxa média de pacotes inteiros recebidos com a menor duração de símbolo e verificar se, como em teoria, será possível realizar ou não uma comunicação full-duplex

4.1 Condições gerais para todos os testes

Todos os testes foram realizados da seguinte maneira:

- Em uma sala silenciosa cuja reverberação era baixa.
- Com dois computadores distintos.
- Distância entre os computadores menos de 1 metro.
- Volume dos computadores em 80% da capacidade total.
- Sensibilidade dos microfones a aproximadamente 70% da capacidade total.

4.2 Teste 1: Duração do símbolo

O objetivo fundamental do teste 1 de duração do símbolo consiste em realizar uma análise minuciosa variando o tamanho do símbolo de 60 ms a 170 ms com incrementos de 10 em 10 ms para determinar na prática qual é o tamanho mínimo viável para um símbolo que seja capaz de transmitir informações sem sofrer muitas perdas numa transmissão devido à sua própria duração. Nesse contexto, busca-se identificar e estabelecer o limite inferior realista em termos de tamanho de símbolo, que ainda permita uma transmissão eficiente e sem comprometer a qualidade e integridade das informações transmitidas.

O teste envolve o computador 1 transmitir através do canal 1 enquanto o computador 2 recebe pelo mesmo canal. A informação transmitida envolve o seguinte conteúdo com 70 caracteres:

```
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
```

Resultados taxa de quadros recebidos enquanto os resultados detalhados podem ser encontrados no Apêndice A:

Tabela 3 – Tabela de testes

Duração	**Teste 1**	**Teste 2**	**Teste 3**	**Teste 4**	**Teste 5**	**Taxa de acerto**
60ms	2/14	0/14	0/14	0/14	0/14	5.71%
70ms	4/14	3/14	4/14	8/14	4/14	45.71%
80ms	4/14	8/14	5/14	5/14	4/14	50.00%
90ms	9/14	12/14	9/14	11/14	10/14	78.57%
100ms	10/14	13/14	13/14	11/14	12/14	85.71%
110ms	13/14	13/14	13/14	13/14	14/14	97.14%
120ms	12/14	11/14	13/14	12/14	13/14	91.43%
130ms	14/14	14/14	14/14	14/14	14/14	100.00%
140ms	13/14	13/14	14/14	13/14	14/14	97.14%
150ms	14/14	14/14	14/14	14/14	14/14	100.00%
160ms	14/14	14/14	12/14	14/14	14/14	97.14%
170ms	14/14	14/14	14/14	14/14	14/14	100.00%

Conclusão do teste 1: Os símbolos de duração a partir de 110 ms, nesse teste, apresentaram resultados satisfatórios. A partir de um valor já não faz mais diferença aumentar o seu tamanho em função dos resultados se estabilizarem.

Por escolha do autor, com base nos resultados, o melhor símbolo para ser utilizado no teste 2 foi o de 130 ms.

Com esta duração de símbolo podemos calcular algumas taxas:

- Taxa de transmissão de **símbolos por segundo**: Sendo que cada símbolo tem duração de 130 ms, logo, 1 / 0,13 = 7.69 símbolos por segundo

- Taxa de transmissão de **bytes úteis (payload) por segundo**: Como um pacote não é só formado por payload, nele há 2 símbolos delimitadores + 2 bytes de CRC. Lembrando que a carga útil pode variar de 1 a 5 bytes. Sendo assim, são existentes 5 tamanhos diferentes de pacotes na qual possuem uma transmissão de bytes úteis por segundo diferente entre si. As colunas 1 e 2 da Tabela 4 expõe como o payload influencia no tamanho do pacote.

  Para simplificar, explica-se apenas o caso com mais bytes de informação útil por pacote: 5 bytes. Dessa forma um pacote inteiro ficaria com 16 símbolos no total para ser transmitido. 16 símbolos possuem uma duração de 2.08 segundos. 5 bytes / 2.08 segundos = **2.40 bytes por segundo**. A coluna 3 da Tabela 4 exibe as taxas de transmissão calculadas.

Vale ressaltar que numa transmissão os pacotes sempre são enviados com o máximo de payload possível, ou seja, numa transmissão de por exemplo 1001 bytes, haverão 200 pacotes de 5 bytes e mais o último com apenas 1 byte útil.

A partir destes resultados também é possível calcular a **eficiência espectral (bps/Hz)**. Nesse teste o canal 1 foi utilizado e nele há 18 frequências de 430 Hs a 1161 HZ, ou seja, uma utilização de 731 Hz.

As colunas 4 e 5 da Tabela 4 exibe a Conversão de B/s para bps (bits por segundo) e cálculo de bps/Hz (eficiência espectral):

Tabela 4 – Tabela de testes

Tamanho do payload	**Tamanho do pacote**	**Taxa de bytes úteis**	**Taxa de bits úteis**	**Eficiência espectral**
5 bytes	16 símbolos	2.40 B/s	19.20 bps	0.2626 bps/Hz
4 bytes	14 símbolos	2.19 B/s	16.80 bps	0.0229 bps/Hz
3 bytes	12 símbolos	1.92 B/s	15.36 bps	0.0210 bps/Hz
2 bytes	10 símbolos	1.53 B/s	12.24 bps	0.0167 bps/Hz
1 byte	08 símbolos	0.96 B/s	07.68 bps	0.0105 bps/Hz

4.3 Teste 2: Taxa de pacotes recebidos livre de erros

O teste 2 tem como finalidade verificar a taxa de sucesso na recepção de pacotes livres de erros, utilizando um tamanho de símbolo de 130 ms por meio da transmissão de uma sequência de 1000 caracteres aleatórios consecutivos demonstrados a no Apêndice B cuja duração é de 6 minutos e 56 segundos de transmissão.

Para um teste melhor, foram realizadas 5 transmissões com as seguintes taxas de pacotes recebidos com sucesso:

- 83.5%
- 87.5%
- 69.5%
- 72.5%
- 98.5%

Conclusão do teste 2: Dos totais 5000 caracteres enviados durante os 5 testes, chegaram ao todo, sem erro, 4115 que é equivalente a uma **taxa de acerto média de 82.3%**. Apesar de que no teste 1 não houve falhas durante a transmissão, no teste 2 não se apresentou a mesma taxa de acerto apesar das mesmas condições iniciais e isso é normal visto que o ambiente não era isolado e pode ter havido alguns ruídos na qual atrapalharam a comunicação.

4.4 Teste 3: Transmissão simultânea

Foi realizado um experimento como a Figura 17 ilustra. Dois computadores distintos, separados por volta de um metro entre si. O computador 1 transmitindo pelo canal 2 e recebendo pelo canal 1 enquanto o computador com a configuração inversa transmitindo pelo canal 1 e recebendo pelo canal 2. Como cada canal possui frequências distintas, é esperado que não haja problemas na transmissão e recepção das informações. O objetivo desse teste é verificar se a comunicação simultânea é possível utilizando canais diferentes para transmissão e recepção dos dados.

Figura 17 – Fluxograma do receptor

Fonte: Desenvolvida pelo autor

Para começar, previamente foi testado cada transmissão e recepção separadamente para a verificação do pleno funcionamento. Após a checagem foi realizado o envio de uma pequena sequência de caracteres por cada computador ao mesmo tempo.

Em ambas recepções os dados não chegaram corretamente apesar de serem transmitidos da maneira certa. Ao realizar o teste full-duplex, não foi como o esperado. Quando ocorrem problemas ao tentar realizar comunicação full-duplex usando ondas sonoras simultaneamente, alguns fenômenos físicos podem estar causando interferência e dificultando a propagação adequada dos sinais. Aqui estão alguns fenômenos comuns que podem estar afetando a comunicação:

- Interferência de frequência: Se a transmissão está transmitindo em frequências próximas ou na mesma faixa, pode ocorrer interferência de frequência. Isso significa que as ondas sonoras podem se sobrepor e se misturar, tornando difícil a distinção entre os sinais transmitidos pelos dois computadores.
- Interferência de fase: Se as ondas sonoras estiverem fora de fase, pode ocorrer interferência de fase. Isso pode resultar em cancelamento parcial ou total do sinal em certas áreas, causando perda de dados e ruído indesejado.
- Limitações técnicas dos dispositivos: Os alto-falantes e microfones utilizados nos computadores podem ter limitações técnicas que afetam a capacidade de transmissão e recepção simultâneas de forma eficiente. Por exemplo, a resposta de frequência, a sensibilidade e a capacidade de cancelamento de eco dos dispositivos podem ser inadequadas para uma comunicação full-duplex eficaz.

Todos esses fenômenos podem ter ocorrido ao mesmo tempo ocasionando a não detecção correta na recepção.

Conclusão do teste 3: A comunicação através de ondas sonoras de maneira full-duplex nesse teste não foi realizada com sucesso, porém é possível realizar melhorias e mais testes minuciosos para a detecção e correção do problema.

5 CONCLUSÕES

Este trabalho buscou explorar um experimento autêntico e original, porém não inovador, de comunicação sem fio que utiliza o som como meio físico para o transporte da informação para estabelecer uma comunicação a distância.

Com o uso da técnica de modulação por FSK, este estudo demonstrou a viabilidade de transmitir dados entre dispositivos por meio do som. Ao enfrentar as limitações de baixas taxas de transmissão e a susceptibilidade a ruídos, foram estabelecidos objetivos para garantir uma confiabilidade mínima dos dados recebidos e possibilitar ao menos a comunicação half-duplex.

Com base nos testes realizados concluí-se que há um grande potencial para melhorias para atingir melhores valores nos mesmos testes, conseguir uma comunicação full-duplex, melhores taxas de transmissão, maiores distâncias entres os hots e até mesmo diferentes meios de transmissão da informação como os subaquáticos.

O trabalho também pode ser utilizado para incentivo ao estudo das telecomunicações a estudantes e entusiastas visto que é um experimento interessante de se fazer além de simples se comparado às atuais tecnologias utilizadas diariamente.

Ao concluir este estudo, foi possível demonstrar que é viável estabelecer uma comunicação sem fio utilizando o som como meio físico, com o uso da técnica de modulação FSK. Apesar das limitações inerentes a esse meio de transmissão, este trabalho ressalta a importância de explorar novas possibilidades na área de comunicações sem fio, incentivando a investigação e o aprimoramento do experimento apresentado.

Espera-se que este estudo tenha contribuído para uma compreensão mais aprofundada das oportunidades e desafios relacionados a essa forma peculiar de comunicação sem fio por meio do som. A continuidade das pesquisas nesse campo poderá ampliar os horizontes da comunicação abrindo portas para aplicações inovadoras e soluções eficientes.

5.1 Trabalhos futuros

Como trabalho futuro ficam as seguintes sugestões:

- Utilizar outras técnicas de modulação para alcançar melhores taxas de transferência

- Melhorar a transmissão dos símbolos utilizando a técnica Continuous Phase Frequency Shift Keying (CPFSK). Atualmente é necessário aumentar e diminuir a amplitude dos símbolos para que não ocorra uma interferência de fase na transição de um símbolo para outro na transmissão, porém ao utilizar a técnica sugerida, espera-se não precisar mais desse artifício.

- Utilizar todo esse experimento apenas como camada física de algum protocolo maior e implementar camadas acima na qual tenham algum esforço em relação a "garantia de entrega".
- Melhorar a recepção levando em conta o nível de sinal que chega no receptor. Atualmente se o receptor receber duas frequências específicas que estão dentro do seu canal de recepção e tiver que decidir uma entre delas para ser a o sinal considerado como símbolo, ele considera a mais grave por uma regra empírica. Nesse caso o sinal mais forte que chega no receptor não é o considerado, mas sim o mais grave. Essa regra pode mudar visto que, em tese, na maioria das vezes, o sinal mais forte deve ser o sinal da transmissão.
- Levar em consideração o tempo que o TX está transmitindo o símbolo, visto que, na recepção atual, o RX detecta o símbolo até 5 vezes durante uma transmissão de símbolo porém após detecta-lo não considera mais o resto do mesmo símbolo, sendo assim um desperdício na qual poderia ser utilizado para dar mais robustez na recepção da informação.
- Substituir o CRC16 por CRC8
- Utilizar códigos corretores de erro para melhorar o desempenho
- Utilizar outras formas e tamanhos de quadro. Atualmente ele possui uma carga útil de 1 a 5 bytes, porém pode ser explorado mais possibilidades em relação a essa estrutura que foi definida empiricamente pelo autor.

REFERÊNCIAS

DONOSO, P. D. J. P. *Som e Acústica Primeira parte: ondas sonoras*. 2006. <https://www.ifsc.usp.br/~donoso/fisica_arquitetura/12_som_acustica_1.pdf>. 27, 28

LOGICS, E. *S2C t HS 55*. 2018. Acesso em: 19/07/2023. Disponível em: <https://evologics.de/product/s2c-t-hs-55>. 32, 33

PETERSON, W. W.; BROWN, D. T. Cyclic redundancy check (crc) computation: an implementation using bit-slice logic. *IEEE transactions on computers*, IEEE, C-24, n. 8, p. 782–793, 1975. 32

RYPULA, R. *Audio Network - send data over sound in JavaScript*. 2015. Disponível em: <https://audio-network.rypula.pl/>. 23

SEPPALA, T. J. *Chromecast's ultrasonic device pairing is much simpler than it sounds*. 2014. Disponível em: <https://www.engadget.com/2014-06-27-chromecast-ultrasonic-pairing.html>. 23

SKLAR, B. *Digital Communications: Fundamentals and Applications*. [S.l.]: Prentice Hall, 2001. v. 2. 28, 29, 30, 31

YOUNG, H. D.; FREEDMAN, R. A. *University Physics*. 12. ed. [S.l.]: Pearson Education, 2012. 27

Apêndices

APÊNDICE A – TESTE 1

60ms:

- YZ12345678
- Nenhum dado recebido
- Nenhum dado recebido
- Nenhum dado recebido
- Nenhum dado recebido

70ms:

- abcdezABCDEFGHI45678
- zABCDEFGHIJKLMN
- pqrstzABCDTUVWXYZ123
- fghijuvwxyzABCDEFGHIJKLMNTUVWXYZ12345678
- abcdeJKLMNYZ12345678

80ms:

- abcdezABCDEFGHI45678
- abcdeuvwxyzABCDEFGHITUVWX456789)(*&%$#@?
- abcdeEFGHIJKLMNYZ12345678
- abcdezABCDJKLMNYZ12345678
- abcdeuvwxyJKLMN9)(*&

90ms:

- abcdeklmnopqrstuvwxyzABCDEFGHIJKLMNTUVWX45678
- abcdepqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijpqrstzABCDEFGHIJKLMNTUVWXYZ12345678
- abcdepqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ12345678&%$#@?
- abcdepqrstzABCDEFGHIJKLMNOPQRSYZ123456789)(*&%$#@?

100ms:

- abcdefghijpqrstzABCDEFGHIOPQRSYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSYZ123456789)(*&%$#@?
- abcdefghijpqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdepqrstzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijpqrstzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?

110ms:

- abcdefghijklmnopqrstzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijpqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijpqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdeklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?

120ms:

- abcdefghijpqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWX456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHITUVWXYZ1239)(*&%$#@?
- abcdefghijpqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- klmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- fghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?

130ms:

- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?

140ms:

- fghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?

- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?

150ms:

- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?

160ms:

- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- fghijklmnopqrstuvwxyEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?

170ms:

- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?
- abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789)(*&%$#@?

APÊNDICE B – TESTE 2

Informação enviada:

1. Lorem ipsum elementum per eget sollicitudin sem scelerisque, ornare facilisis ullamcorper congue commodo eleifend, nibh molestie donec sagittis purus odio. malesuada pulvinar amet maecenas malesuada mattis cubilia purus in, volutpat tempus ultricies quis dolor venenatis hac eu duis, interdum in orci elit luctus lobortis praesent. imperdiet sapien feugiat sem massa mi aliquet amet leo ac nibh, neque felis est molestie semper praesent eros mi donec, molestie habitant curae ut lacinia mattis tempor lacus porttitor. aliquet semper molestie torquent egestas taciti at euismod, posuere sem netus bibendum morbi ut, adipiscing laoreet lectus enim vivamus non. Dictum primis a elit etiam amet senectus dapibus quisque etiam donec duis, lectus pharetra sed imperdiet aptent sit ipsum et dolor suscipit ornare habitant, congue porttitor ante dui iaculis praesent vehicula eu aliquam malesuada. nulla platea sem etiam cras purus lorem arcu, egestas dictumst aenean sodales consequat porttitor faucibus asr

Como resultado foram recebidas as seguintes sequências:

1. Lorem ipsum elementum per sollicitudin sem scsque, ornare facilisis ullamcorper congue commodo eleifend, nibh tie donec sagittis purus malesuada pulvinar amet maecenas mada mattis cubilia pin, vat teultriquis dolor venenatis hac eu duis, interdum in orci elit luctbortis praesent. impt sapien feugiat sem massa mi aliquo ac neque felis est molestie sempaesens mi donec, mol habitant curae ut lacinia mattis tempor lacus porttitor. aliquet sempe torquenttas tacitiuismod, posuere sem netus bibendum morbi ut, adipiscing laoreet l enim vivamus non. Dictum primis a elit senedapibisquem donis, lectus pharetra sed imperdiet aptent sit ipsum et dolor suscirnare habi congrttitte dui iaculis praesent vehicula eu aliquam malesuada. nulla platea sem etiam cras purus lorem arcu, egestas dictumst aenean sodales coat porttitor faucibus asr

2. Lorem ipsum elementum per eget sollicitudin sem scelerisque, ornare facilisis ullamcocongue commodo eleifend, nibh molestie donec sagittis podio. malesuada pulvinar maecenas malesuada mattis cubilia purus in, vat tempus ultriquis dolor venenatis hac eu duis, interdum in orci elit luctbortis praesent. impt sapien feugiat sem massa mi aliquet amet leo ac nibh, neque felt mol sempaesent eros mi donec, molestie habitant curae ut lacinia mattis tempor lacus porttitor. aliquet semplestie torquenttas taciti at euismod, posuere sem netus bibendum morbiadipiscing laoreet l enim vivamus non. Dictum primis a elit senectus dapibus quisquem donec duis, lectus pharsed imperdptent sit ipsumolor pit o habitant, congue porttitte dui iaculis praesent vehicula eu aliquam malesuada. nulla platea sem etiam cras purus lorem arcu, egestas dictumst aenean sodales corttitor faucibus asr

3. Lorem ipsum elementum per sollicitudin sem scsque, ornare facilisis ullamcocongue commodo eleifend, nibh tie donec sagittis podio. malesuada pulvinar maecenas mada mattis cubilia pin, vat teultriquis dolor venenatis hac is, ium in orci elit luctbortis pra. impt sapien feugiat sem massa mi aliquo ac e felt mol sempt eros mi donec, mol habitant acinia mattis tempor lacus porttitor. al sempe torquenttas tacitiuismod, po sem bibendum morbiadipiscingectus enim vivaon. Dictum primis a elit dapibisquem donis, l pharsed iptent sit ipsumolor pit ornare habi congue porttitte dui iaculis ent vehicula eu aliquam mada. nullaea seam cras puorem arcu,tas dictumst aenean sodalat porttitor faucibus asr

4. Lorem ipsum elementu sollicitudin sem scsque, ornare facilisis ullamcocongue commodo eleifend, nibh tie donec sagittis podio. malesuada pulvinar maecenas mada mattis cubilia pin, vat teultriquis dolor venenatis hac eu duis, ium in orci elit luctbortis pra. imperdiet sapien feugiat sem massa mi aliquo ac neque felt molestie semps mi donec, mol habitant acinia mattis tempor lacus porttitor. al semper moe torquenttas tacitiuismod, po sem bibendum morbiadipiscing laor enim vivaon. Dictum primis a elit etiamdapibisquem donis, l pharsed iptent sit ipsumolor pit ornare habi congue porttitte dui iaculis ent vehicula eu aliquam malesuada. nulla platea seam cras puorem arcu,tas dictumst aenean sodalnsequat poor faucibus asr

5. Lorem ipsum elementum per eget sollicitudin sem scelerisque, ornare facilisis ullamcorper congue commodo eleifend, nibh molestie donec sagittis purus odio. malesuada pulvinar amet nas malesuada mattis cubilia purus in, volutpat tempus ultricies quis dolor venenatis hac eu duis, interdum in orci elit luctus lobortis praesent. imperdiet sapien feugiat sem massa mi aliquet amet leo ac neque felis est mol semper praesent eros mi donec, molestie habitant curae ut lacinia mattis tempor lacus porttitor. aliquet semper molestie torquent egestas taciti at euismod, posuere sem netus bibendum morbi ut, adipiscing laoreet lectus enim vivamus non. Dictum primis a elit etiam amet senectus dapibus quisque etiam donec duis, lectus pharetra sed imperdiet aptent sit ipsum et dolor suscipit ornare habitant, congue porttitor ante dui iaculis praesent vehicula eu aliquam malesuada. nulla platea sem etiam cras purus lorem arcu, egestas dictumst aenean sodales consequat porttitor faucibus asr

APÊNDICE C – CÓDIGO DO TRANSMISSOR

Código C.1 – Transmissor

```
import numpy as np
import pyaudio
import crc_ifsc

channel = int(input('Canal? [1/2] '))

def envia(quadro):
        bytes_hex = quadro.data.hex()
        hex_esc = do_not_repeat_symbol(bytes_hex)
        esc_reset = insert_reset_symbol(hex_esc)
        from_hex_to_audio(esc_reset)

def do_not_repeat_symbol(bytes_hex):
    last_symbol = ''
    bytes_hex_esc = ''
    for b in bytes_hex:
        if b == last_symbol:
            bytes_hex_esc += 'g' # ESC
            last_symbol = 'g'
        else:
            bytes_hex_esc += b
            last_symbol = b
    last_symbol = ''
    print('esc: ' + bytes_hex_esc)

    return bytes_hex_esc

def from_hex_to_audio(bytes_hex_esc):
    global channel
    if channel == 1:
        #                 Explicação: representação_em_hexa[slot_freq]
        #                 0[20]  1[22]  2[24]  3[26]  4[28]  5[30]  6[32]  7[34]
        frequency_list = [430.0, 474.0, 516.0, 562.0, 604.0, 646.0, 689.0, 733.0,
        #                 8[36]  9[38]  a[40]  b[42]  c[44]  d[46]  e[48]  f[50]
                         774.0, 819.0, 861.0, 905.0, 948.0, 991.0, 1033.0,1076.0,
        #                 g(esc)[52] repete o último simbolo
                         1120.0,
        #                 h(reset)[54] reseta o buffer do rx
```

```
                        1161.0]
    elif channel == 2:
        #                 Explicação: representação_em_hexa[slot_freq]
        #                 0[56]  1[58]  2[60]  3[62]  4[64]  5[66]  6[68]  7[70]
        frequency_list = [1205.0,1249.0,1292.0,1335.0,1378.0,1422.0,1465.0,1508.0,
        #                 8[72]  9[74]  a[76]  b[78]  c[80]  d[82]  e[84]  f[86]

    1550.0,1594.0,1637.0,1680.0,1723.0,1766.0,1808.0,1852.0,
        #                 g(esc)[88] repete o último simbolo
                            1896.0,
        #                 h(reset)[90] reseta o buffer do rx
                            1938.0]

    duration = 0.2  # in seconds. Até 0.8 já funcionou bem.
    sampling_rate = 44100.0
    symbol_list = generate_symbols(frequency_list, duration, sampling_rate)
    audio = generate_audio(symbol_list, bytes_hex_esc)
    play(audio)


def play(audio):
    p = pyaudio.PyAudio()
    # for paFloat32 sample values must be in range [-1.0, 1.0]
    stream = p.open(format=pyaudio.paFloat32,
                    channels=1,
                    rate=44100,
                    output=True)

    stream.write(audio)
    stream.stop_stream()
    stream.close()
    p.terminate()

def generate_audio(symbol_list, sequence):
    audio = b''
    for s in sequence:
        audio += symbol_list[int(s, 18)].tobytes()
    return audio

def generate_symbols(frequency_list, duration, sampling_rate):
    symbol_list= []
    t = int(duration * sampling_rate)
    template = np.arange(t) / sampling_rate
    x = np.linspace(0, np.pi, t)
    mask = np.sin(x)
    for freq in frequency_list:
        sinal = np.sin(template * freq * 2 * np.pi)
```

```
        symbol = np.multiply(sinal, mask).astype(np.float32)
        symbol_list.append(symbol)

    return symbol_list

# Define o frame
def insert_reset_symbol(hex_esc):
    return 'h' + hex_esc + 'h'

# Insere o CRC
def msg_with_crc(msg):
    c = crc_ifsc.CRC16(msg)
    return c.gen_crc()


while 1:
    linha = input('>>> ')

    crc_chunk = 5

    while linha:
        l = linha[:crc_chunk]
        data_bytes = l.encode('UTF-8')
        msg_crc = msg_with_crc(data_bytes)
        bytes_hex = msg_crc.hex()
        hex_esc = do_not_repeat_symbol(bytes_hex)
        esc_reset = insert_reset_symbol(hex_esc)
        from_hex_to_audio(esc_reset)
        linha = linha[crc_chunk:]
```

APÊNDICE D – CÓDIGO DO RECEPTOR

Código D.1 – Receptor

```
import numpy as np
import pyaudio as pa
import struct
import crc_ifsc

global channel
channel = int(input('Canal? [1/2]'))

CHUNK = 1024 * 2
FORMAT = pa.paInt16
CHANNELS = 1 # não alterar
RATE = 44100

p = pa.PyAudio()

stream = p.open(
    format = FORMAT,
    channels = CHANNELS,
    rate = RATE,
    input=True,
    output=False,
    frames_per_buffer=CHUNK
)

buffer = ''
last_slot = ''

if channel == 1:
    freqs_meaning = {
        '20': '0',
        '22': '1',
        '24': '2',
        '26': '3',
        '28': '4',
        '30': '5',
        '32': '6',
        '34': '7',
        '36': '8',
        '38': '9',
        '40': 'a',
        '42': 'b',
```

```
42          '44': 'c',
43          '46': 'd',
44          '48': 'e',
45          '50': 'f',
46          '52': 'g', # esc
47          '54': 'h'  # reset rx buffer
48      }
49  elif channel == 2:
50      freqs_meaning = {
51          '56': '0',
52          '58': '1',
53          '60': '2',
54          '62': '3',
55          '64': '4',
56          '66': '5',
57          '68': '6',
58          '70': '7',
59          '72': '8',
60          '74': '9',
61          '76': 'a',
62          '78': 'b',
63          '80': 'c',
64          '82': 'd',
65          '84': 'e',
66          '86': 'f',
67          '88': 'g', # esc
68          '90': 'h'  # reset rx buffer
69      }
70
71  while 1:
72      data = stream.read(CHUNK)
73      data_int = struct.unpack(str(CHUNK) + 'h', data)
74      data_fft = np.abs(np.fft.fft(data_int))*2/(11000*CHUNK)
75      f_bins = data_fft[20:91] > 1
76
77      # Obtem o slot mais baixo e salva no buffer
78      if len(np.where(f_bins)[0]) > 0:
79
80          # Se achou mais de uma freq ao mesmo tempo, e tais no canal 2
81          #   Pegue a segunda freq
82          # Senão
83          #   Pegue a primeira mesmo
84          if len(np.where(f_bins)[0]) > 1 and channel == 2:
85              slot = int(np.where(f_bins)[0][1]) + 20
86          else:
87              slot = int(np.where(f_bins)[0][0]) + 20
88
```

```
89          if channel == 1 and slot not in
        [20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54]:
90              slot = -1 # não deixa reconhecer uma frequencia que não está no seu
        canal
91          elif channel == 2 and slot not in
        [56,58,60,62,64,66,68,70,72,74,76,78,80,82,84,86,88,90]:
92              slot = -1 # não deixa reconhecer uma frequencia que não está no seu
        canal
93
94          if str(slot) in freqs_meaning.keys() and last_slot != slot:
95              last_slot = slot
96              if freqs_meaning[str(slot)] not in 'gh':
97                  # Se for de 0 a f, adicione
98                  buffer += freqs_meaning[str(slot)]
99              elif freqs_meaning[str(slot)] == 'g':
100                 # Se for 'g', então é ESC.
101                 # Repita o último caractere
102                 if len(buffer) > 0:
103                     buffer += buffer[-1]
104             elif freqs_meaning[str(slot)] == 'h':
105                 # Se for 'h', então é pra limpar o buffer
106                 # Serve para manter sincronismo
107                 buffer = ''
108
109     if len(buffer) > 2:
110         try:
111             fcs = crc_ifsc.CRC16(bytes.fromhex(buffer))
112             if fcs.check_crc():
113                 mensagem = bytes.fromhex(buffer[:-4]).decode('UTF-8') # retira 2
        bytes (4 simbolos) do CRC
114                 print(mensagem, end = "",  flush=True)
115                 buffer = ''
116         except:
117             pass
```