Relatório Final

Projeto Nível I e II em Eletrônica II

Vitória Beatriz Bianchin Gustavo Friol Bento Matheus Bateli Neumann

27 de março de 2025

1 Informações dos Alunos

O presente relatório foi feito em conjunto para as disciplinas EEL7863 e EEL7837 pelos alunos Vitória Beatriz Bianchin (Matrícula 19205016 — Disciplina EEL7863 — Turma 09202), Gustavo Friol Bento (Matrícula 20250477 — Disciplina EEL7837 — Turma 08202) e Matheus Bateli Neumann (Matrícula 19205851 — Disciplina EEL7837 — Turma 08202).

2 Revisão Bibliográfica

Em um estudo descritivo que usou dados de cursos de graduação em medicina que participaram do Censo do Ensino Superior coordenado pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep), em 2018, foi apresentado que as tecnologias assistivas para estudantes com deficiência auditiva e/ou visual que eram disponibilizadas nos cursos de medicina, considerando 323 cursos de medicina funcionando no Brasil, a maioria deles (90) confirmou a oferta de pelo menos um tipo de tecnologia assistiva (TA). A disciplina de Língua Brasileira de Sinais foi a TA mais frequentemente ofertada (80), e o material tátil foi a menos ofertada (32), tendo os cursos apresentado melhor completude de TA para apoiar estudantes com deficiência auditiva do que com deficiência visual.

O apoio a população com deficiência visual, por sua vez, tem um escopo de atenção pequeno nos cursos de medicina e no ensino superior de maneira geral, o que por sua vez reflete na ausência de medidas de apoio e inclusão para os deficientes visuais na sociedade.

A organização das cidades, que é um reflexo dos processos sociais em curso, deve ser concebida de maneira a garantir que as pessoas com deficiência tenham a capacidade de participar plenamente na vida urbana e realizar suas atividades diárias em igualdade de condições com os demais cidadãos. A acessibilidade não é apenas um direito em si, mas também uma condição essencial para a realização de todos os direitos fundamentais estabelecidos na Constituição Federal.

Nesse contexto, é fundamental entender os conceitos de mobilidade urbana e acessibilidade para as pessoas com deficiência visual.

2.1 Tecnologias de Assistência às Pessoas com Deficiências Visuais

Semelhante ao projeto proposto, existe uma tecnologia apresentada anteriormente, no Panamá, que recebeu o nome de MOVIDIS, que representa a sigla para "Mobilidade para Indivíduos com Deficiência Visual."Esta abordagem sistêmica aborda quatro módulos de comunicação por radiofrequência (RF), no qual é detalhado: o módulo para ônibus, direcionado aos motoristas (MOVI-BUS); o módulo do usuário, especialmente projetado para atender às necessidades das pessoas com deficiência visual (MOVI-ETA); o módulo de parada (MOVI-STOP), estrategicamente instalado nas paradas de ônibus, desempenhando um papel fundamental como elo de comunicação entre o MOVI-ETA e o MOVI-BUS; e, finalmente, o módulo MOVI-MASTER, implantado nas estações de ônibus para uma gestão abrangente do sistema.

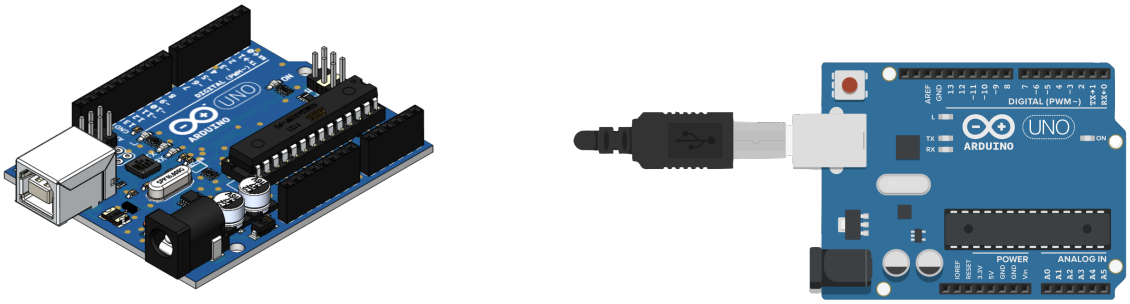
Quando uma pessoa com deficiência visual, que está usando um módulo MOVI-ETA, chega a uma parada de ônibus equipada com um módulo MOVI-STOP, ela informa, por meio do seu MOVI-ETA, qual é o seu destino final. O MOVI-STOP mantém uma comunicação constante para registrar a solicitação de destino do usuário, permitindo que qualquer ônibus cadastrado no sistema RF MOVIDIS

saiba que existe um usuário do MOVIDIS naquela parada específica. Quando um ônibus solicitado se aproxima da parada, o MOVI-STOP irá informar ao usuário sobre essa situação, enviando um sinal que acionará o alarme sonoro presente em seu MOVI-ETA.

3 Blocos Principais do Circuito

3.1 Arduino UNO

O Arduino Uno é uma placa de desenvolvimento com um microcontrolador ATMega328P. Ele oferece 14 pinos digitais de entrada/saída, 6 pinos analógicos, uma conexão USB para programação, e é alimentado por 5V. É amplamente utilizado devido à sua facilidade de programação e versatilidade em projetos de eletrônica, robótica e automação e por isso foi escolhido para ser o microcontrolador deste projeto.



(a) (b)

Figura 1: Visualizações do Arduino

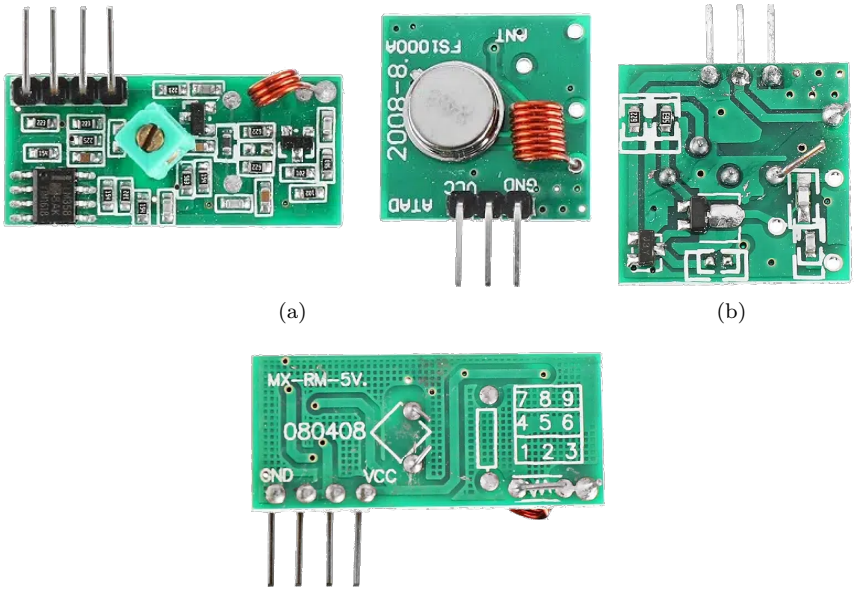
3.2 Módulo transmissor e receptor

Os módulos 433 MHz, frequentemente associados à tecnologia de modulação de amplitude ASK (Amplitude Shift Keying) modulam a amplitude do sinal de rádio para transmitir informações. Além disso, muitos módulos 433 MHz podem ser facilmente integrados a microcontroladores, como Arduino, através da biblioteca RadioHead, uma das bibliotecas utilizadas no projeto. Essa biblioteca oferece uma série de funções e recursos para facilitar a comunicação sem fio. Ela suporta uma variedade de modos de modulação, incluindo ASK, e permite a utilização de protocolos como SPI (Serial Peripheral Interface) para controle e coordenação dos dispositivos, como foi utilizado no projeto.

3.3 Alimentação

Para a alimentação foi usado duas baterias de 9V para alimentar o Arduino, do Arduino foi possível utilizar o pino de 5V para alimentar o receptor. Já no transmissor, como se deseja ter mais potência para transmitir sinais com maior alcance foi desencapado uma aprte do fio do *clip* ligado na bateria e soldado fios para conectar à alimentação do transmissor diretamente, sendo a única ligação dele com o Arduino o pino de dados. A Figura 3 mostra a bateria utilizada no projeto.

Módulo	Tensão	Origem
Receptor	5V	Pino 5V Arduino Rx
Transmissor	9V	Bateria 1
Arduino para Tx	9V	Bateria 1
Arduino para Rx	9V	Bateria 2



(a) (b)

(c)

Figura 2: Visualizações do Módulo 433MHz Rx/Tx

3.4 Motor de vibração

O motor de vibração foi retirado de um antigo controle de *PlayStation 2*, assim, conseguiu-se reutilizar o componente para a tensão desejada sem apresentar problemas pois num controle de *PS2* a tensão de alimentação é 3V. A Figura 4 mostra o motor utilizado.

3.5 Teclado Matricial

Foi utilizado um teclado matricial 4x4 de membrana com 8 pinos de conexão para ser utilizado como um dos periféricos. Uma modificação a ser adicionada é adesivos em Braile correspondendo aos números, tornando, então, acessível para deficientes visuais.

O Código utilizou da biblioteca *Keypad* para integrar as funcionalidades com o arduino. A Figura 5 mostra o componente utilizado.

3.6 Antena

Para a antena, tanto do receptor quanto do transmissor, foi utilizado um fio de cobre encapado de 17,3 cm. O comprimento foi escolhido com base na frequência de 433MHz da portadora do transmissor. Para que um sinal seja irradiado de forma eficiente, o ideal é que a antena meça $\lambda/2$ ou $\lambda/4$. Se a onda eletromagnética se propaga no espaço com velocidade $3x10^8 m/s$ e $\lambda = v/f$:

$$\lambda = 300.000.000/433.000.000 = 0,69m.$$

Portanto, o menor tamanho possível para um fio de cobre esticado funcionar de forma eficiente como antena, é 17,3 cm (69 cm / 4).

4 Descrição do Projeto

Segue abaixo alguma idealizações do projeto e seu fluxo de funcionamento.

4.1 Fluxo de entrada e saída

O projeto base consiste em o usuário digitar a linha de ônibus desejada e esperar até que o ônibus se aproxime e o aparelho vibre para alertar que o veículo está próximo e pode solicitar a parada.

Figura 3: Bateria 9V utilizada.

Figura 4: Motor de vibração.

Para isto o ônibus possui o módulo transmissor que periodicamente fica enviando o sinal da sua linha de referência enquanto junto com o usuário há o módulo receptor, a pessoa digita a linha desejada e dentro do aparelho há um código comparador rodando que relaciona a igualdade da linha recebida com a digitada e assim que essa igualdade é verdadeira, o usuário é alertado através de uma vibração no aparelho.

4.2 Funcionalidades associadas ao projeto

Dentre as funcionalidades do projeto, pode se citar a possibilidade de escolha da linha de ônibus através do *Keypad*, a vibração do dispositivo do usuário quando sua linha está próxima e também o acionamento de um sinal luminoso dentro do ônibus que avisa o motorista que tem um PCD em um ponto próximo esperando para embarcar.

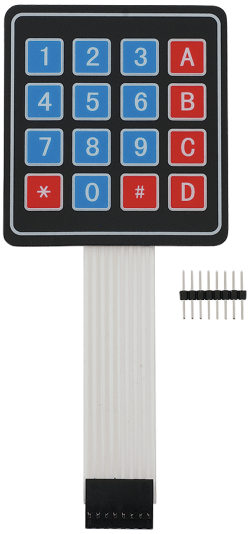


Figura 5: Teclado numérico de membrana.

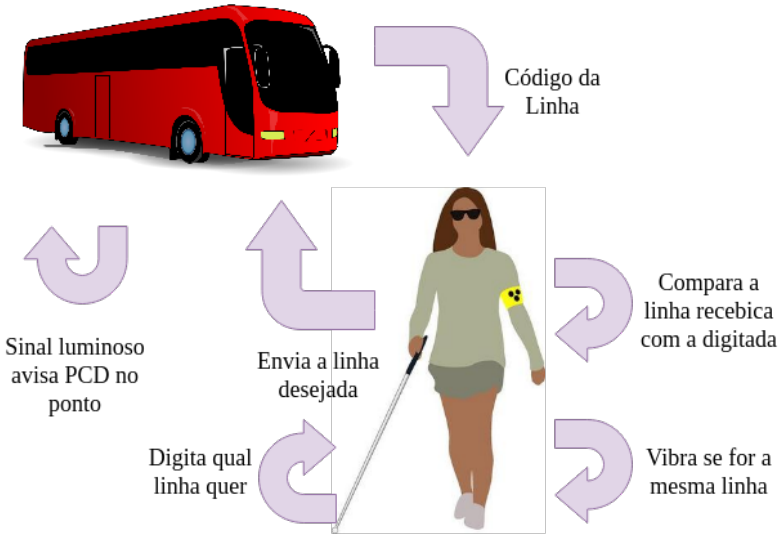


Figura 6: Fluxo de funcionamento do produto.

5 Design e Montagem do Projeto

5.1 Prototipagem

5.2 Inicial

Ambos os módulos transmissor e receptor contavam com pinos de alimentação (VCC, GND) e dados (DATA), assim, foi feito a ligação aos pinos do arduino e bateria através da protoboard. A antena foi soldada diretamente nos módulos e para as baterias foi utilizado um *clip* que conectava seus terminais e bifurcado e soldado o fio para alimentar diretamente o transmissor também com tal tensão para ter mais potência de transmissão.

O módulo de teclado foi conectado diretamente ao arduino através de jumpers, utilizou-se apenas 7 de seus 8 pinos pois a última coluna de letras não foi utilizada já que sua funcionalizade envolvia

apenas números e as teclas * e #.

O motor também foi conectado diretamente, normalmente se utiliza alguma chave, transistor ou até mesmo resistor para ser a conexão intermediária e não queimar um pino do microcontrolador, porém a tensão de utilização desse motor de vibração é muito baixa e não apresentou problemas em conexão direta.

5.2.1 Final

Na segunda parte do projeto foi adicionado mais um módulo 433MHz para que fosse possível o usuário enviar ao motorista que estava esperando por tal ônibus. Além disso, houve uma troca de Arduino do Uno para o Nano para que ficasse mais miniaturizado.

Ambos os códigos também foram implementados fazendo com que fosse possível a recepção e envio simultâneo dos sinais, assim, o motorista envia o sinal com número da linha e o receptor também envia a que ele deseja após ter selecionado através do teclado e quando ambas as linhas batem, o motorista é avisado através de um sinal luminoso representado pelo LED do pino 13 do Arduino.

A Figura 9 mostra a finalização da prototipagem.

5.3 Códigos Finais

5.3.1 Ônibus

```
#include <RHReliableDatagram.h>                          // biblioteca Radiohead
    reliableDatagram
#include <RH_ASK.h>                                      // biblioteca Radiohead ASK
#include <SPI.h>                                         // biblioteca SPI

#define TX_ADDRESS 1                                     // endere o do transmissor
#define RX_ADDRESS 2                                     // endere o do recept

RH_ASK driver;                                           // inst ncia RH ASK
RHReliableDatagram gerente(driver, TX_ADDRESS);          // configurando o gerenciador

const int pinoLED = 13;
uint8_t count = 1;                                       // contador
uint8_t data[] = "177";                                 // mensagem a ser enviada
uint8_t buf[RH_ASK_MAX_MESSAGE_LEN];                     // buffer da mensagem
uint8_t tamanho;                                         // tamanho da mensagem
uint8_t from;                                            // endere o de quem transmite

void setup()
{
  pinMode(pinoLED, OUTPUT);                        // Configura o pino como sa da
  Serial.begin(9600);                                    // inicializa console serial 9600
     bps
  if (!gerente.init())                                   // se a inicializa  o do
    gerenciador falhar
    Serial.println("Falha na inicializacao");            // print na console serial
}

void loop()
{

  if (gerente.available())                               // se gerenciador estiver ativo
  {
    tamanho = sizeof(buf);                               // determina o tamanho do buffer
    if (gerente.recvfromAck(buf, &tamanho, &from))       // se o gerenciador receber
    mensagem
    {
      Serial.print("Recebido de: 0x");                   // print na console serial
      Serial.print(from, HEX);                           // print do endere o do
    transmissor em Hexadecimal
      Serial.print(": ");                                // print na console serial
      Serial.println((char*)buf);                        // print da mensagem recebida

      if (strcmp((char*)buf, (char*)data) == 0) {
        // Ativa o led'
        digitalWrite(pinoLED, HIGH);
```

```
        delay(1000);                                    // Ativa por 1 segundo
        digitalWrite(pinoLED, LOW);
        }
      }

    }
    else {
      Serial.print("Transmitindo mensagem n. ");               // print na console
    serial
      Serial.println(count);                                   // print do contador
      if (!gerente.sendtoWait(data, sizeof(data), RX_ADDRESS))  // se gerenciador
    enviar mensagem
      {
        count++;                                               // incrementa contador
      }
      delay(1000);                                              // atraso 0,5 segundo

      }
}
```

5.3.2 Usuário

```
#include <RHReliableDatagram.h>                          // biblioteca Radiohead
    reliableDatagram
#include <RH_ASK.h>                                      // biblioteca Radiohead ASK
#include <SPI.h>                                         // biblioteca SPI
#include <Keypad.h>

#define TX_ADDRESS 1                                     // endere o do transmissor
#define RX_ADDRESS 2                                     // endere o do receptor

const int pinoVibracao = 9;
uint8_t count = 0;                                       // contador
uint8_t buf[RH_ASK_MAX_MESSAGE_LEN];                     // buffer da mensagem
uint8_t tamanho;                                         // tamanho da mensagem
uint8_t from;                                            // endere o de quem transmite

bool busline_conhecida = false;
int     busline_temp_pos = 0;
uint8_t busline_temp[4+1];          //vari vel de armazenamento intermedi rio
uint8_t busline[4+1];                                    //4    quantidade max de digitos
    da varial e 1 /0
RH_ASK driver;                                           // inst ncia RH ASK
RHReliableDatagram gerente(driver, RX_ADDRESS);          // configurando o gerenciador

const byte ROWS = 4;            //4 linhas
const byte COLS = 3;                                    //3 colunas
char keys[ROWS][COLS] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'*','0','#'}
};

byte rowPins[ROWS] = {10,8,7,6};                 //pinos das linhas
byte colPins[COLS] = {5,2,3};                    //pinos das colunas

char key;

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup()
{
  pinMode(pinoVibracao, OUTPUT);                         // Configura o pino como sa da
  Serial.begin(9600);                                    // inicializa console serial 9600
     bps
  if (!gerente.init())                                   // se a inicializa  o do
    gerenciador falhar
    Serial.println("Falha na inicializacao");            // print na console serial
  keypad.addEventListener(keypadEvent);                  // adiciona evento
}
```

```
46
47 void loop()
48 {
49   key = keypad.getKey();
50   if (gerente.available())                              // se gerenciador estiver ativo
51   {
52     tamanho = sizeof(buf);                             // determina o tamanho do buffer
53     if (gerente.recvfromAck(buf, &tamanho, &from))     // se o gerenciador receber
     mensagem
54     {
55       Serial.print("Recebido de: 0x");                 // print na console serial
56       Serial.print(from, HEX);                         // print do endere o do
     transmissor em Hexadecimal
57       Serial.print(": ");                              // print na console serial
58       Serial.println((char*)buf);                      // print da mensagem recebida
59
60       if (strcmp((char*)buf, (char*)busline) == 0) {
61         // Ativa o pino de vibra  o
62         digitalWrite(pinoVibracao, HIGH);
63         delay(1000);                                   // Ativa por 1 segundo
64         digitalWrite(pinoVibracao, LOW);
65         }
66       }
67
68     }
69     if (busline_conhecida == true){
70       Serial.print("envio periodico");
71       gerente.sendtoWait(busline, sizeof(busline), TX_ADDRESS);
72     delay(1000);
73     }
74   }
75
76  // cuidando dos eventos
77 void keypadEvent(KeypadEvent key){
78     if (keypad.getState() == PRESSED){
79         if (key == '#') {                              //Apagar
80     busline[0] = '0';
81     busline[1] = '0';
82     busline[2] = '0';
83     busline[3] = '0';
84     busline[4] = '\0';
85     busline_temp_pos = 0 ;
86     busline_conhecida = false;
87     Serial.print("Apagar");
88         }
89         else if (key == '*') {                         //Enviar e quando envia apaga
90             strcpy((char*)busline, (char*)busline_temp);
91     busline_temp[0] = '\0';
92     busline_temp_pos = 0 ;
93     Serial.print("Enviar");
94         Serial.println((char*)busline);
95 //        Serial.println((char*)busline_temp);
96         busline_conhecida =true;
97         }
98   else {
99     Serial.println(key);
100    if (busline_temp_pos > 4) {
101      digitalWrite(pinoVibracao, HIGH);
102      delay(1000);                          // Ativa por 1 segundo
103      digitalWrite(pinoVibracao, LOW);
104    }
105    else {
106      busline_temp[busline_temp_pos++] = key;
107      busline_temp[busline_temp_pos] = '\0';
108      }
109  }
110
111    }
112 }
```

5.4 Simulação e Teste

5.4.1 Iniciais

Um dos primeiros passos feitos foi comprovar a comunicação entre transmissor e receptor, um modo de analisar o espectro e se utilizou da frequência esperada foi através da utilização de um Rádio Definido por Software e um software chamado *Gqrx*.

A Figura 7 mostra o teste feito. Nela é possível observar que o sinal eviado pelo transmissor é periódico como programado e está dentro da faixa de frequência esperada (433MHz). Assim, pudemos comprovar o funcionamento do dispositivo transmissor e que o projeto da antena estava correto.

Outro teste realizado foi o de distância mas ainda com alimentação 5V e não 9V para o transmissor, uma pessoa ficou no hall do CTC e a outra foi para perto da rua. Nesse ponto o SDR ainda assim registrou um sinal ainda que fraco vindo do transmissor. É esperado que com a adição já feita de 9V ao transmissor seja possível uma distância ainda maior e mais verossímil para o usuário e o ônibus.

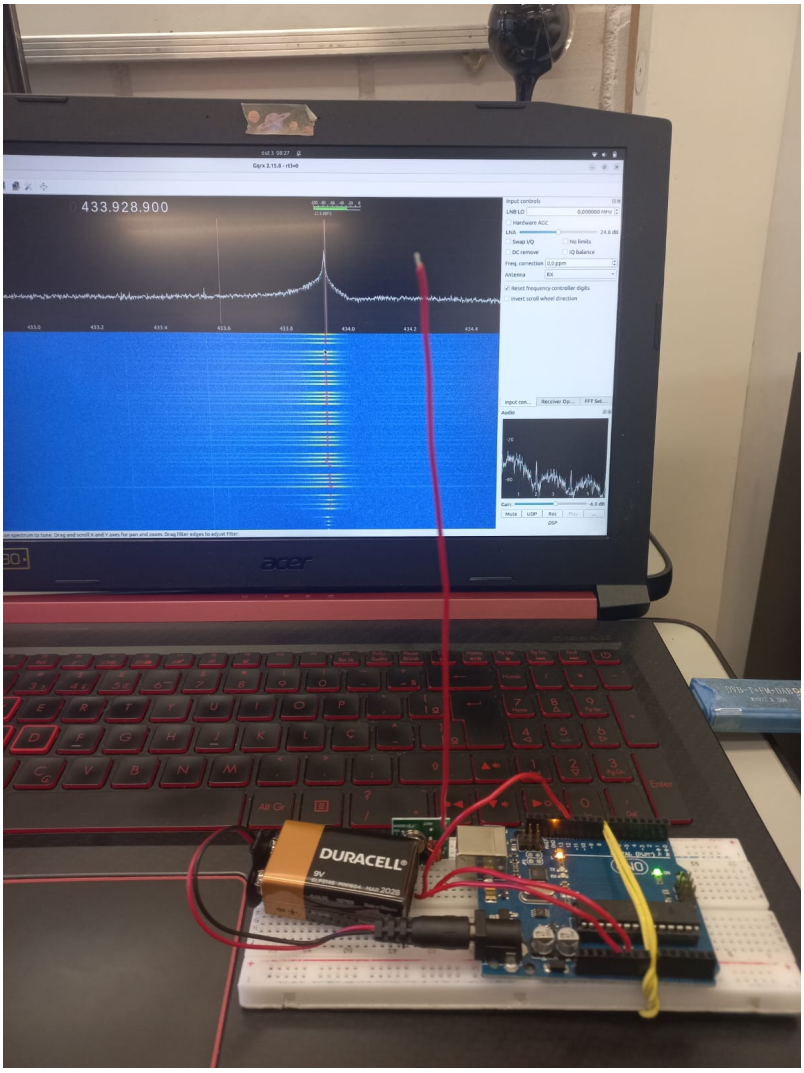


Figura 7: Espectro apresentado pelo Gqrx.

5.4.2 Finais

Após a alteração do código e adição de um módulo 433MHz para se obter recepção e transmissão de ambos os lados, foi necessário avaliar se haveria interferência nesses sinais e se eles se comportavam como o esperado.

Na Figura 8 é apresentado dois sinais, um deles é a já conhecida linha do ônibus, no qual o usuário irá receber, porém, o outro sinal apresentado vem do usuário e é recebido pelo ônibus, ele irá avisar que existe uma pessoa com deficiência visual que gostaria de entrar no veículo, assim, facilitaria a acessibilidade para o usuário.

Ao receber o aviso por parte do usuário, o motorista é avisado através de um sinal luminoso em seupainel, no caso do projeto tal sinal é representado pelo próprio LED, pino 13, do Arduino.

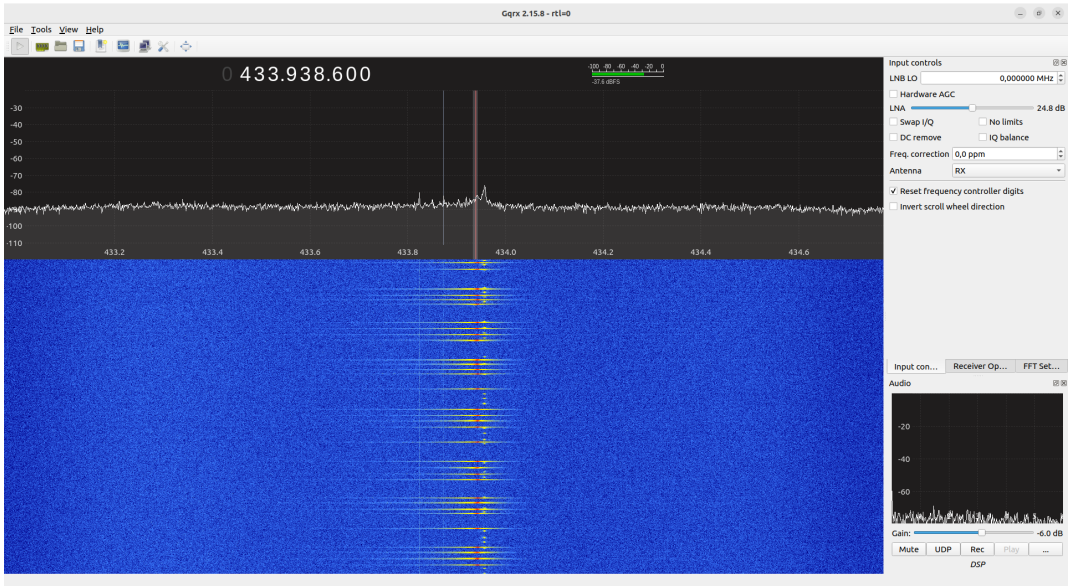


Figura 8: Recepção de ambos os sinais.

5.5 Materiais

- Duas baterias 9V;
- Clip para bateria;
- 2 Protoboard;
- 2 Arduinos UNO e seus cabos de conexão;
- Estanho para solda;
- Motor de vibração retirado de um controle de PS2;
- Keypad HX 543;
- Jumpers e fios diversos;
- Estilete;
- *Software Defined Radio* RTL;

5.6 Finalização e Funcionamento

Pensando na implementação real do protótipo, seria crucial a sua miniaturização como um todo, a criação de uma *case* para o dispositivo do usuário, a adição de um sistema de monitoramento de bateria e a possibilidade de recarregá-la via USB, por praticidade. Apesar disso, a idealização inicial do projeto foi concluída com sucesso e é apresentada na Figura 9.

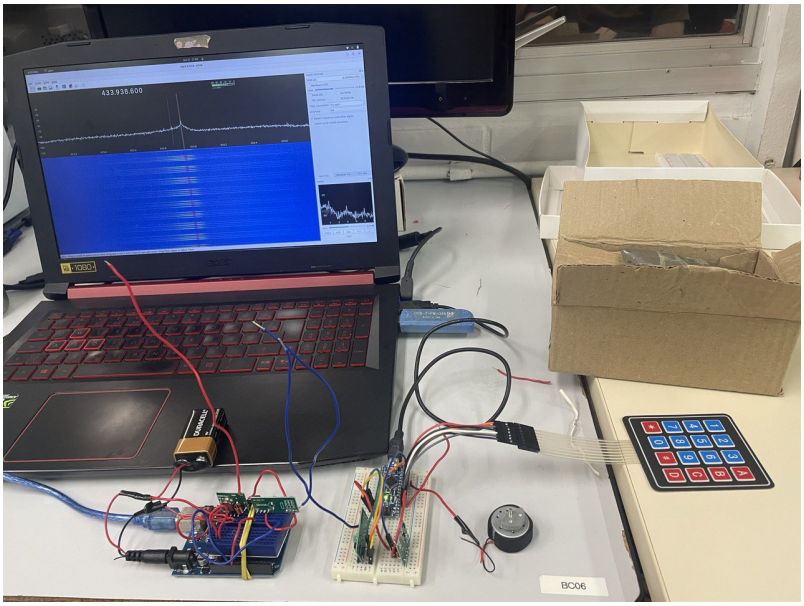


Figura 9: Finalização do projeto funcionando.

Referências

[1] NIDCD. Assistive Devices for People with Hearing, Voice, Speech, or Language Disorders. 2011. Disponível em: `https://www-nidcd-nih.ez46.periodicos.capes.gov.br/health/assistive-devices-people-hearing-voice-speech-or-language-disorders`

[2] NASCIMENTO, Maria Isabel Do; TORRES, Rhian Costa; RIBEIRO, Klynsman Grisotto Faria. Tecnologias assistivas para deficiência visual e auditiva ofertadas aos estudantes de medicina no Brasil. **Revista Brasileira de Educação Médica**, v.46, n.1, p.e037, 2022. Disponível em: `<http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-55022022000100223&tlng=pt>`. Acesso em: 5.set.2023.

[3] SOCIAL SCIENCES, Health. **LibGuides: Blind/Visual Impairment: Common Assistive Technologies.** Disponível em: `<https://guides.library.illinois.edu/c.php?g=526852&p=3602299>`. Acesso em: 5 set. 2023.

[4] https://www.sense.org.uk/information-and-advice/communication/assistive-technology/

[5] Sáez, Y., Muñoz, J., Canto, F., García, A., Montes, H. (2019). Assisting Visually Impaired People in the Public Transport System through RF-Communication and Embedded Systems. MDPI, 19(6), 1282. Disponível em: https://www.mdpi.com/1424-8220/19/6/1282

[6] Lourete, L. M. A., Santos, T. P. Título: ACESSIBILIDADE PARA PESSOAS COM DEFICIÊNCIA VISUAL: UM ESTUDO SOBRE A SINALIZAÇÃO SEMAFÓRICA DA AVENIDA JOÃO FELIPE CALMON NO MUNICÍPIO DE LINHARES/ES. URL: https://colatina.ifes.edu.br/images/tccs/AdmPub2018/TCC$_{A}dmPub_{2}018_{T}aisPereiraSantos.pdf$