

Centro Estadual de educação tecnologia Paula Souza
Etec Júlio de Mesquita
Curso Técnico em Eletrônica

Kelly Cristina Alves da Silva
Nathan Magri do Santos
Lucas Matias Lira

Theremin

Santo André – SP

2022

Kelly Cristina Alves da Silva

Nathan Magri do Santos

Lucas Matias Lira

Theremin

Trabalho de Conclusão de Curso
apresentado ao Curso Técnico em
Eletrônica da Etec Júlio de Mesquita,
orientado pelo Prof. Egmar Accetto,
como requisito parcial para obtenção
do título de técnico em Eletrônica.

Santo André - SP

2022

Resumo

Um dos primeiros instrumentos musicais inteiramente eletrônico, o Theremin visa produzir efeitos sonoros e melodias. Os projetos originais eram instrumentos analógicos. O protótipo que foi montado é digital, e usa sensores ultrassônicos, alto falante, buzzer (em uma das versões), Arduino uno e carenagem de uma impressora 3D. Para tocá-lo não é necessário o contato físico com o instrumento. Basta aproximar e distanciar as mãos de seus sensores. É um protótipo de montagem simples e barata com intuito de aproximar esta montagem eletrônica de um instrumento musical. Além do Arduino Uno, o projeto usa dois sensores ultrassônico para medir a proximidade das mãos através de entradas analógicas do Arduino. Foram testadas diversas variações no circuito do Theremin e com os testes escolhemos a configuração atual. Os modelos analógicos antigos possuíam maior potência sonora. O modelo montado é mais lúdico e prático de ser usado e para melhorar a potência do som colocamos um amplificador de áudio na saída do circuito.

Palavras-Chaves: Theremin, Arduino, sensores ultrassônicos, carenagem 3D.

Abstract

One of the first entirely electronic musical instruments, the Theremin aims to produce sound effects and melodies. The original designs were analog instruments. The prototype that was assembled is digital, and uses ultrasonic sensors, speakers, buzzer, Arduino uno and fairing of a 3D printer. To play it, physical contact with the instrument is not necessary. Just move your hands closer and further away from your sensors. It is a simple and cheap assembly prototype with the aim of bringing this electronic assembly closer to a musical instrument. In addition to the Arduino Uno, the project uses two ultrasonic sensors to measure the proximity of the hands through analog inputs from the Arduino. Several variations on the Theremin circuit were tested and with the tests we chose the current configuration. The old analogue models had more sound power. The assembled model is more playful and practical to use and to improve the sound power we placed an audio amplifier at the output of the circuit.

Keywords: Theremin, Arduino, ultrasonic sensors, 3D fairing.

Agradecimentos

Agradecemos a todos os professores dessa escola técnica que em muito contribuíram para a realização desse projeto. Agradecemos os orientadores de TCC, o professor Egmar, Fred e Robson que nos guiaram com seus ensinamentos deste trabalho de conclusão de curso, sem o qual conhecimento nada disso seria possível.

Sumário

1. Introdução	6
1.1 Objetivos	7
2. Contexto Histórico.....	8
2.1 Biografia de Lev Termen.....	8
3. Fundamentação teórica	10
3.1 Arduino Uno	10
3.2 Sensor ultrassônico	11
3.3 Alto Falante de 8 ohms	11
3.4 Amplificador LM386	12
4. Metodologia da pesquisa.....	14
5. Desenvolvimento	15
5.1 Diagrama de blocos do circuito	15
5.2 Esquema no TinkerCard.....	15
5.3 Funcionamento do circuito	16
5.4 Programação.....	16
5.5 Lista de preços dos componentes utilizados	16
5.6 Testes.....	17
5.7 Projeto 3D	17
6. Conclusões	18
7. Considerações Finais	19
8. Referencias	20
9. Apêndice	21

1. Introdução

O theremin é um instrumento musical que tocado sem qualquer tipo de contato físico. De acordo com John Theo Sierpinski De Souza,

Na década de 1920, utilizando-se de toda a evolução da eletrônica até então, o russo Lev Termen desenvolve um aparelho capaz de variar a frequência de um som ao alterar-se a distância entre uma antena e a mão do operador. Nasce dessa forma o instrumento musical que posteriormente seria conhecido como theremin, capaz de produzir sons sem qualquer contato físico. Nele, existem duas antenas: uma para controle de frequência (quanto mais próxima a mão do instrumentista, maior a frequência), e outra para controle de volume/amplitude (quanto mais próxima a mão do músico, menor o volume). Originalmente, o theremin era composto por um circuito analógico e valvulado. Com o desenvolvimento da eletrônica, o circuito tornou-se diferente. Em ambos, o princípio de funcionamento é o mesmo.

O controle de frequência irá gerar uma onda base, de frequência fixa, e outra cuja frequência varia conforme a distância da mão à antena. Tais ondas são somadas (efeito Heteródino), e a resultante é direcionada a um amplificador de áudio. O controle de volume tem efeito similar, porém faz uso de elementos inversores em seu funcionamento (o volume é inversamente proporcional à distância da mão à antena). A resultante também é inserida no amplificador de áudio, que fará o processamento e fornecerá o som final, em uma saída de áudio. (SOUZA, MENDES, GASOTO, 2009)

Como visto acima, o Theremin originalmente era baseado em circuitos analógicos. Já o theremin desse projeto é digital baseado na plataforma Arduino. Existem dispositivos semelhantes ao montado nesse projeto. Esse projeto foi feito usando sensores ultrassônicos. Um dos testes utilizou sensores de campo elétrico (usa a capacitância criada pelo corpo humano próximo a placas condutoras). Essa ideia surgiu na busca de uma biblioteca de detecção capacitiva, o que torna o atual dispositivo original de certa forma.

1.1 Objetivos

Como objetivo geral deste projeto, tem-se o estudo das etapas envolvidas e a construção do theremin digital. Basicamente:

- Estudo histórico do theremin e seu criador;
- Busca pelos componentes necessários;
- Programar o Arduino;

- Uso de bibliotecas no Arduino;
- Aprender a programar sensores ultrassônicos;
- Desenvolver um projeto em uma impressora 3D;
- Montagem dos componentes na protobord;
- Testes práticos

Além da música, também serão discutidas outras aplicações do princípio de funcionamento do theremin.

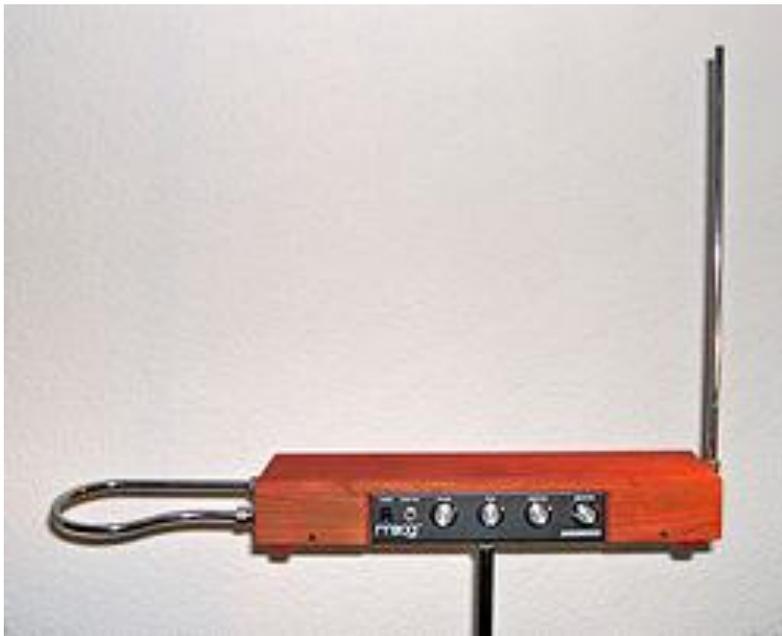


Figura 2: Theremin – (WIKIPÉDIA, 2021)

2. Contexto Histórico

2.1 Breve histórico de Lev Termen

De acordo com John Theo Sierpinski De Souza,

Lev Termen, mais tarde conhecido como Leon Theremin, nasceu em 1896 na cidade de São Petersburgo, da atual Rússia. Desde sua infância despertara interesse por ciência, em especial por eletricidade, área na qual aos 7 anos começou a dedicar atenção

Em 1920 theremin vai trabalhar com oscilações elétricas de alta frequência na nova divisão de Física e Tecnologia fundada por aquele em Leningrado, que produz uma série de outros experimentos como um oscilador de alta frequência para medir a constante de gases com alta precisão.

Ao ver tal resultado, loffe requisitou que Leon buscase outras aplicações para o experimento (THEREMIN, 1970). Uma dessas aplicações foi em um sensor de presença que tinha uma área de atuação de cerca de 5 metros e sensível a variações de 1mm (THEREMIN, 1983). Outra foi adicionar um gerador de áudio no equipamento e perceber que, ao manipular sua mão nas proximidades, a frequência ouvida variava (THEREMIN, 1970). Em outubro do mesmo ano, com uma apresentação a loffe e outros professores, L. Theremin mostra as potencialidades de um novo "instrumento musical". Ao manipular as mãos próximas ao instrumento, busca notas musicais e tenta assim reproduzir "Swans"(Camille Saint-Saens) lembrando-se do tempo em que estudava cello, e percebendo o quão diferente era a interação entre manipulador e produtor de som, no caso instrumentista e instrumento. Nessa versão do instrumento o volume era controlado por um pedal acoplado. . (SOUZA, MENDES, GASOTO, 2009)



Figura 3: criador do Theremin (SOUZA, MENDES, GASOTO, 2009)

Em 1921, Theremin pede a patente pelo seu evento, a qual recebe em 1924. Desta época até 1927 foca-se também na transmissão televisiva ou também chamada de televisão sem o conseguindo mostrar

movimento em uma imagem usando 64 linhas intercaladas, usando a técnica por ele inventada: o Interlace, que é uma técnica de aperfeiçoamento de imagens de um sinal de vídeo. (SOUZA, MENDES, GASOTO, 2009)

3. Fundamentação teórica

A seguir são descritos detalhes dos principais componentes utilizados nesse projeto.

3.1. Arduino Uno

Arduino Uno é uma plataforma da família arduino das mais conhecidas.

Segundo o site vida de silício o arduino uno é composto pelos seguintes blocos:



Figura 4: Blocos do arduino Uno (MOTA, 2021)

Características do arduino uno:

- **Microcontrolador:** ATmega328
- **Tensão de Operação:** 5V
- **Tensão de Entrada:** 7-12V
- **Portas Digitais:** 14 (6 podem ser usadas como PWM)
- **Portas Analógicas:** 6
- **Corrente Pinos I/O:** 40mA
- **Corrente Pinos 3,3V:** 50mA
- **Memória Flash:** 32KB (0,5KB usado no bootloader)
- **SRAM:** 2KB
- **EEPROM:** 1KB
- **Velocidade do Clock:** 16MHz – (SILVEIRA, 2018)

3.2 Sensor ultrassônico

O sensor ultrassônico é muito usado em montagens para medida de distância, sensor de obstáculos e outros. Nesse projeto será usado para medir a distância das mãos através de duas entradas analógicas do Arduino. A ligação do sensor ultrassônico ao Arduino envolve 4 fios como mostrado abaixo:

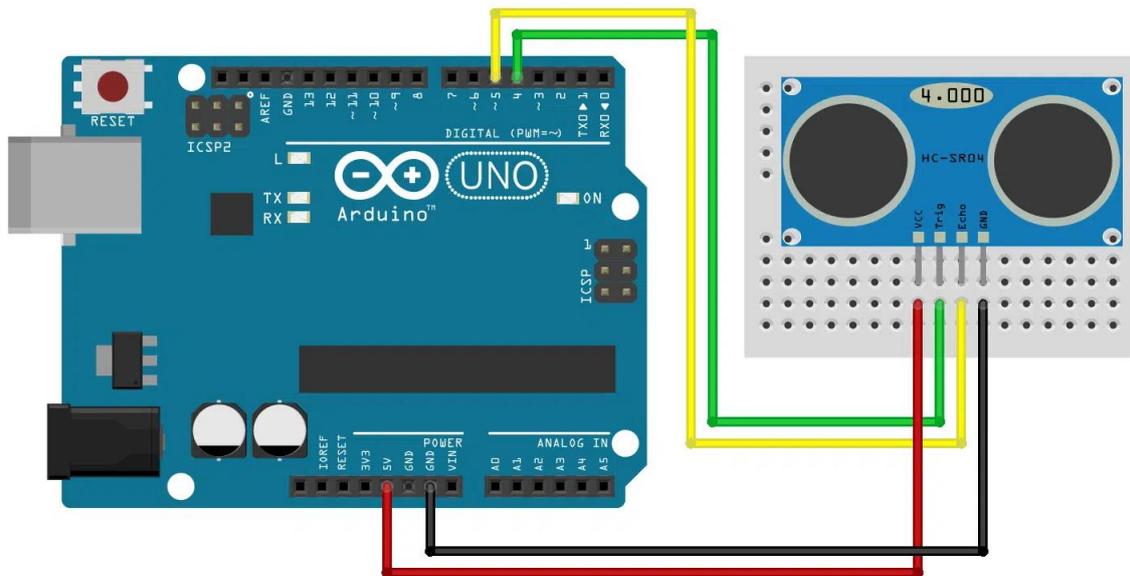


Figura 5: Ligação do sensor ultrassônico. (THOMSEN, 2011)

O sensor ultrassônico usa a seguinte biblioteca na programação:

```
#include <Ultrasonic.h>
```

3.3 Alto Falante de 8 ohms

“O alto falante faz a conversão de energia elétrica em energia mecânica. A energia mecânica comprime o ar e converte o movimento em energia sonora ou nível de pressão sonora.” (Mult comercial, 2021)

O alto-falante de 8 ohms usado no projeto é do tipo citado a seguir pelo site da Autocore Robótica.

O Mini Alto-falante Speaker 8 Ohm é um dispositivo responsável pela emissão de sinais sonoros, sejam eles provenientes de

microcontroladores ou provenientes de dispositivos compatíveis com esse modelo de alto-falante.

O Mini Alto-falante Speaker 8 Ohm pode ser utilizado nos mais diversos projetos onde existe a necessidade de emissão de um sinal sonoro de alerta ou até mesmo uma música. O acionamento do Mini Alto-falante Speaker 8 Ohm é realizado através da variação de tensão DC no dispositivo, essa variação pode ser realizada através de um microcontrolador ou de um dispositivo compatível, amplificador de som ou outra saída de áudio. (Autocore ROBÓTICA, 2022)



Figura 6: Imagem ilustrativa do alto-falante. (Autocore ROBÓTICA, 2022)

Características do mini alto-falante de 8 ohms:

- Tensão de Trabalho: 5V;
- Tipo: Passivo;
- Potência: 0,5W;
- Impedância: 8 Ohms; . (Autocore ROBÓTICA, 2022)

3.4 Amplificador LM386

O Módulo Amplificador de Áudio LM386 é feito para você amplificar um sinal e ter uma maior potência. Com este módulo você pode pegar uma saída de áudio para um fone, por exemplo, e amplificá-la para pôr em uma caixa de som.

Na placa existem 4 pinos, que tornam bastante fácil o seu uso. Junto do módulo também vem um potenciômetro, para ajuste da amplificação, podendo ir até 20 vezes o sinal de entrada. (RECICOMP, 2022)



Figura 7: Imagem ilustrativa do amplificador. (RECICOMP, 2022)

Especificações:

- Chip: LM386 (datasheet)
- Tensão de operação: entre 5 V e 12 V
- Amplificação: até 20 vezes o sinal de entrada
- Potenciômetro de 10 k ohm, para ajuste de volume
- Dimensões: 41 mm x 13 mm
- Peso: 4 g (RECICOMP, 2022)

4. Metodologia do protótipo

A ideia inicial do projeto foi discutida e adotada durante as aulas de PTCC, com o professor Robson. Nessas aulas foram feitas também as pesquisas iniciais. Nas aulas de DTCC foi feita a simulação inicial do protótipo no software Tinkercad. A seguir pesquisou-se as peças necessárias ao básico do projeto e o preço das mesmas. Após isso montou-se e testou-se em partes o protótipo em protoboard. Foram sendo testadas várias versões do projeto. Inicialmente com sensores ultrassônicos Arduíno e buzzer. Numa versão posterior tentou-se usar placas de circuito impresso em lugar dos sensores ultrassônicos, o que não funcionou como o desejado. A seguir retornou-se aos sensores ultrassônicos. Por fim foi incluído o módulo amplificador de áudio para aumentar a potência de saída do som. Foi seguido esse método de tentativa e erro até se chegar ao protótipo final. Na montagem final foi usado um modelo moldado em impressora 3D (WELAUSEN, Fillipe; WELAUSEN, Priscila, 2020?)

5. Desenvolvimento

5.1. Diagrama de blocos do circuito

Aqui temos um diagrama de blocos explicando, o projeto:

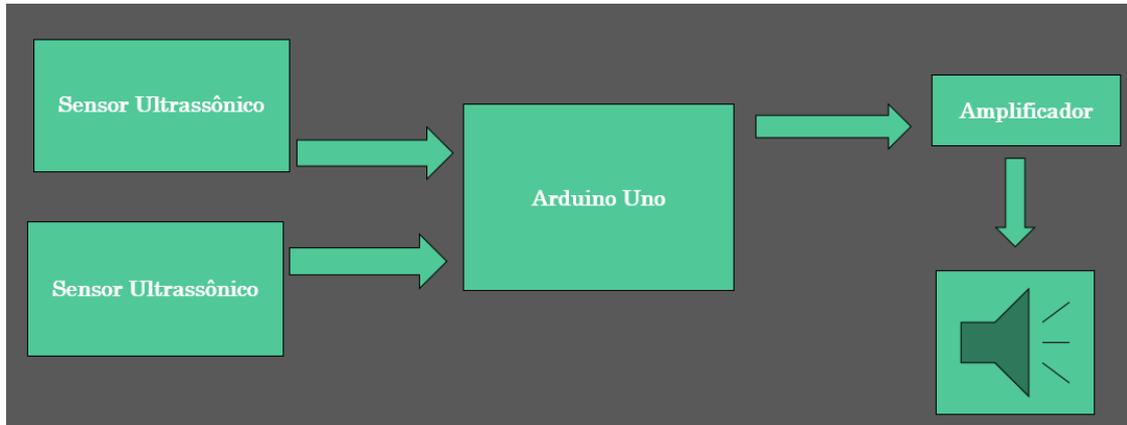


Figura 8: Imagem ilustrativa do amplificador. Dos autores

5.2. Esquema no TinkerCad.

A seguir está o projeto no montado no software TinkerCad:

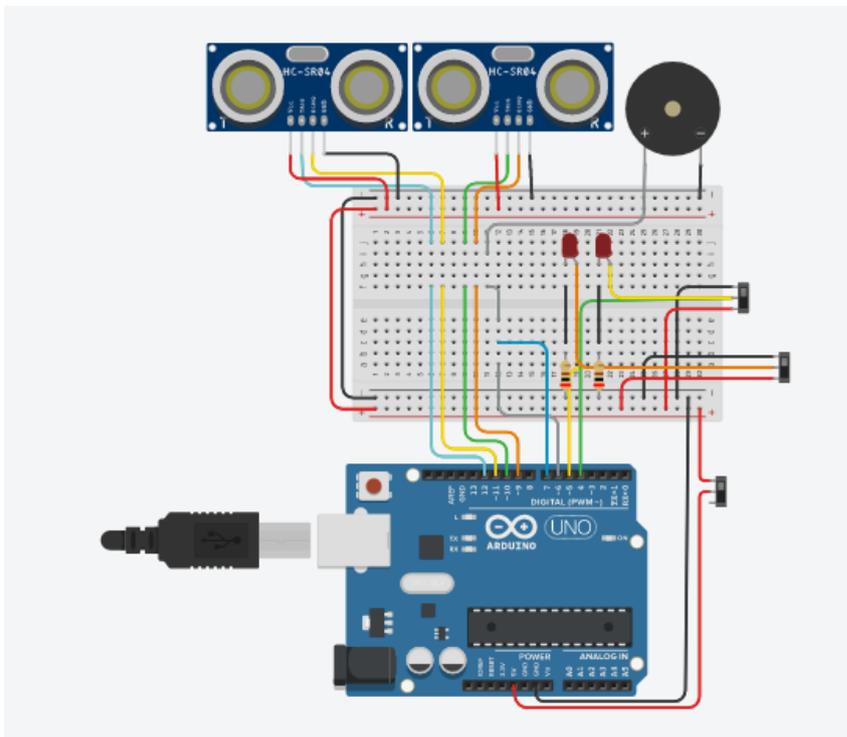


Figura 9: Imagem ilustrativa do amplificador. Dos autores

5.3 Funcionamento do circuito

Um Arduino uno recebe o sinal de dois sensores ultrassônicos através dos movimentos das mãos. Um dos sensores ultrassônicos está sendo usado para mudar a nota e o outro para amplitude. Na saída o mini amplificador aumenta o volume de som e envia para o auto falante.

5.4 Programação

Foi usada a biblioteca de sensores ultrassônicos. Essa biblioteca calcula a distância entre a mão da pessoa até o sensor e então vai em um dicionário pronto de definições no início do programa e pega a nota desejada, de acordo com a distância.

Na programação, o nosso maior aliado nas pesquisas foi o site arduino, que nos forneceu informações precisas. Dificuldade: como transformar os sinais que o Arduino manda em som. Vendo exemplos e comparando com nosso trabalho

Em pesquisas vimos que para gerar sons no arduino devemos utilizar uma função própria do Arduino, a qual chama-se tone(). Ela consegue gerar ondas em frequências específicas nos pinos de PWM e, com isso, gerar notas musicais. Essa função possui dois modos de utilização com 2 parâmetros - tone(pino, frequência) - ou três - tone(pino, frequência, duração).

No funcionamento inicial há uma parte musical na programação que foi extraída de um tocador de música com buzzer (COUTO, 2019)

5.5. Lista de preços dos componentes utilizados:

Placa Uno + Cabo USB + Pino	R\$: 63,21
2 Sensores ultrassônicos	R\$: 53,98
Alto-Falante de 8 ohms	R\$: 5,90
Botão	R\$: 2,00
Chave gangorra	R\$: 2,50
Módulo Amplificador LM386	R\$: 15,00
Total	R\$: 144,59

Note que não estão incluídos os valores gastos com outras peças que não estão no protótipo final, tais como o buzzer e nem o da carenagem que foi conseguida gratuitamente em outra escola onde um dos alunos estuda.

5.6 Testes

No decorrer no projeto ocorreram diversos testes com sensores, amplificadores e placa de fenolite. Esses testes ajudaram a melhorar e desenvolver o projeto.

O primeiro teste feito foi com sensores ultrassônicos. Programamos os sensores usando um para controle de frequência e outro para amplitude.

O segundo teste foi usando placas de fenolite em lugar dos ultrassônicos. Primeiramente usamos um bombril para tirar a superfície de cobre da placa e deixa lá lisa e homogênea. Logo após soldamos um jumper macho para conectar no Arduino. A placa está sendo usada para gerar a capacitância do corpo, ela poderia ser usada como um sensor quando aproximar a mão a capacitância varia.

O terceiro teste foi feito com amplificador operacional, o objetivo era deixar o som do theremin mais alto e diferenciado.

5.7 Projeto 3D

A caixa que suporta o protótipo foi confeccionada na impressora 3D na escola E.E.João Galeão em Santo André. Ela ajuda o funcionamento correto do projeto e torna sua aparência mais bonita. O formato foi pensado para ter o encaixe da maioria dos componentes. O tamanho inicial foi pensado para usar um Arduino nano.

6. Conclusões

A partir deste projeto pôde-se contemplar os princípios por trás do funcionamento de sensores. No theremin, este sensor é utilizado para determinar a posição das mãos. Imagine um instrumento musical que é tocado sem que o músico encoste, sobre ou tenha contato direto com ele. Sendo possível determinar a distância da mão à antena no lado direito, quanto mais próximo o músico move a mão, mais silencioso o efeito sobre o som. Quanto mais afastar a mão, mais alto o som. Também utilizamos sensores ultrassônicos. Tudo por meio de um experimento de montagem simples e barata. Conforme esboçado no Relatório Parcial, analisamos os efeitos dos componentes eletrônicos individualmente no circuito, de forma a compreendermos os possíveis resultados ao variarmos os mesmos. Dessa forma, foi possível aperfeiçoar a proposta inicial, otimizando a qualidade do som e, portanto, desenvolvemos novas habilidades e conhecimento, concretizando assim, o nosso objetivo de aproximar esta montagem eletrônica de um instrumento musical. Foram feitos diversos testes em cima da ideia original alterando os tipos de sensores e as saídas de som. Ao final a melhor configuração testada usa um alto falante de 8ohms e um mini amplificador de áudio. Todas essas modificações causaram problemas que tiveram que de ser resolvidos durante o projeto. Tudo isso atrasou bastante o término do projeto.

7. Considerações Finais

No início deste projeto elencamos alguns objetivos, tais como melhorar a experiência sonora para produzir efeitos sonoros e melodias. O Theremin produz apenas uma nota por vez, cuja frequência varia conforme a mão é aproximada ou afastada do sensor. No projeto aperfeiçoamos nossos conhecimentos sobre Arduino uno, sua programação e suas bibliotecas. Montou-se um instrumento musical lúdico. Em uma versão futura do theremin a próxima equipe terá possibilidade de sugerir ideias tais como usar o arduino nano e um amplificador de maior potência e testar mais detalhadamente o uso dos dois sensores.

Referências

SOUZA, John Theo Sierpinski; MENDES, Luís Guilherme Bergamini; GASOTO, Renato Girardi. **DESENVOLVIMENTO DE THEREMIN FOCADO NO ESTUDO DE SENSORES**. 35 f. Monografia – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná, 2009 – Disponível em <<http://paginapessoal.utfpr.edu.br/msergio/portuguese/ensino-de-fisica/oficina-de-integracao-ii/oficina-de-integracao-ii/Monog-09-1-Theremin.pdf>> Acessado em 19 set. 2022

WELAUSEN, Fillipe; WELAUSEN, Priscila. **Arduino Ultrasonic Theremin**. Circuito Maker, 2020? Disponível em <<https://circuitomaker.com/arduino-ultrasonic-theremin/>> Acessado em 19 set. 2022.

WIKIPEDIA. **Theremin**. 2021. Disponível em <<https://pt.wikipedia.org/wiki/Theremin>> Acessado em 19 set. 2022.

MOTA, Allan. **O que é Arduino e como funciona?** , 2021. Disponível em <<https://portal.vidadesilicio.com.br/o-que-e-arduino-e-como-funciona/>> Acessado em 19 set. 2022.

SILVEIRA, André Luis Marques. **Arduino, Internet das coisas e Computação vestível** , 2018. Disponível em <<http://www.um.pro.br/arduino/index.php?c=ArduinoUno>> Acessado em 19 set. 2022.

THOMSEN, Adilson. **Como conectar o sensor ultrassônico HC-SR04 ao Arduino**, 2011. Disponível em <<https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>> Acessado em 19 set. 2022.

COMERCIAL, Mult. **Como funciona um alto-falante? Descubra agora!** , 2021. Disponível em <<https://blog.multcomercial.com.br/como-funciona-um-alto-falante-descubra-agora/#:~:text=Um%20alto%20falante%20%C3%A9%20um,movimento%20para%20energia%20ac%C3%BAstica%2Fsonora>> Acessado em 30 set. 2022.

ROBÓTICA, Autocare. **Mini Alto-falante speaker 8 Ohm 40 mm**, 2022. Disponível em <<https://www.autocorerobotica.com.br/mini-alto-falante-speaker-8-ohm-40mm>> Acessado em 02 out. 2022.

RECICOMP. **Módulo amplificador de áudio LM386**, 2022. Disponível em <<https://www.recicomp.com.br/produtos/modulo-amplificador-de-audio-lm386/>> Acessado em 12 out. 2022.

COUTO, Robson. **Arduino music player usando piezo buzzer**. 2021. Disponível em <<https://techatronic.com/arduino-music-player-using-piezo-buzzer>> Acessado 30 set. 2022.

Apêndice

Vai a seguir programação do projeto que inicia com música de Couto (2021)

```
#define NOTE_B0 31
#define NOTE_C1 33
#define NOTE_CS1 35
#define NOTE_D1 37
#define NOTE_DS1 39
#define NOTE_E1 41
#define NOTE_F1 44
#define NOTE_FS1 46
#define NOTE_G1 49
#define NOTE_GS1 52
#define NOTE_A1 55
#define NOTE_AS1 58
#define NOTE_B1 62
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 78
#define NOTE_E2 82
#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
```

```
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
#define NOTE_DS8 4978
#define REST 0 // (COUTO, 2021)
```

```
#define TRIG_PIN 10
```

```
#define ECHO_PIN 9
```

```
#define TRIG_PIN2 12
```

```
#define ECHO_PIN2 11
```

```
void setup(){
```

```
  initialize();
```

```
  pinMode(TRIG_PIN, OUTPUT);
```

```
  pinMode(ECHO_PIN, INPUT);
```

```
  pinMode(TRIG_PIN2, OUTPUT);
```

```
pinMode(ECHO_PIN2, INPUT);
pinMode(7, OUTPUT);
pinMode(6, OUTPUT);
pinMode(5, INPUT);
pinMode(4, INPUT);

digitalWrite(6, LOW);
digitalWrite(7, HIGH);
}

void loop(){

if(digitalRead(5)== HIGH){
  digitalWrite(7, LOW);
  digitalWrite(6, HIGH);
  noTone(7);
  Harry();
}else if(digitalRead(4)== HIGH){
  digitalWrite(7, LOW);
  digitalWrite(6, HIGH);
  noTone(7);
  Tetris();
}else{
  digitalWrite(TRIG_PIN, HIGH);
  digitalWrite(TRIG_PIN2, HIGH);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, LOW);
  digitalWrite(TRIG_PIN2, LOW);

const unsigned long dur = pulseIn(ECHO_PIN, HIGH);
int dist = dur/29/2;
```

```

const unsigned long dura = pulseIn(ECHO_PIN2, HIGH);
int dista = dura/29/2;
int nota = map(dist,0,300,131,988);
int del = map(dist,0,300,200,1000);

noTone(7);
delay(del);
tone(7, nota);
}
}

void initialize(){

int tempo = 105;
int buzzer = 7;

int melody[] = {
// Pacman
NOTE_B4, 16, NOTE_B5, 16, NOTE_FS5, 16, NOTE_DS5, 16, //1
NOTE_B5, 32, NOTE_FS5, -16, NOTE_DS5, 8, NOTE_C5, 16,
NOTE_C6, 16, NOTE_G6, 16, NOTE_E6, 16, NOTE_C6, 32, NOTE_G6, -16,
NOTE_E6, 8,

NOTE_B4, 16, NOTE_B5, 16, NOTE_FS5, 16, NOTE_DS5, 16,
NOTE_B5, 32, //2
NOTE_FS5, -16, NOTE_DS5, 8, NOTE_DS5, 32, NOTE_E5, 32, NOTE_F5,
32,
NOTE_F5, 32, NOTE_FS5, 32, NOTE_G5, 32, NOTE_G5, 32,
NOTE_G5, 32, NOTE_A5, 16, NOTE_B5, 8
};

int notes = sizeof(melody) / sizeof(melody[0]) / 2;
int wholenote = (60000 * 4) / tempo;

```

```

int divider = 0, noteDuration = 0;
for (int thisNote = 0; thisNote < notes * 2; thisNote = thisNote + 2) {

    // calculates the duration of each note
    divider = melody[thisNote + 1];
    if (divider > 0) {
        // regular note, just proceed
        noteDuration = (wholenote) / divider;
    } else if (divider < 0) {
        // dotted notes are represented with negative durations!!
        noteDuration = (wholenote) / abs(divider);
        noteDuration *= 1.5; // increases the duration in half for dotted
notes
    }
    tone(buzzer, melody[thisNote], noteDuration * 0.9);
// (COUTO, 2021)

    delay(noteDuration);
}
loop();
}

```

```

void Harry(){
int buzzer = 6;
int tempo = 144;

int melody[] = {
// Hedwig's theme from the Harry Potter Movies // (COUTO, 2021)

REST, 2, NOTE_D4, 4,
NOTE_G4, -4, NOTE_AS4, 8, NOTE_A4, 4,
NOTE_G4, 2, NOTE_D5, 4,
NOTE_C5, -2,
NOTE_A4, -2,
NOTE_G4, -4, NOTE_AS4, 8, NOTE_A4, 4,
NOTE_F4, 2, NOTE_GS4, 4,
NOTE_D4, -1,
NOTE_D4, 4,
NOTE_G4, -4, NOTE_AS4, 8, NOTE_A4, 4, //10
NOTE_G4, 2, NOTE_D5, 4,

```

```

NOTE_F5, 2, NOTE_E5, 4,
NOTE_DS5, 2, NOTE_B4, 4,
NOTE_DS5, -4, NOTE_D5, 8, NOTE_CS5, 4,
NOTE_CS4, 2, NOTE_B4, 4,
NOTE_G4, -1,
};

int notes = sizeof(melody) / sizeof(melody[0]) / 2;
int wholenote = (60000 * 4) / tempo;
int divider = 0, noteDuration = 0;

for (int thisNote = 0; thisNote < notes * 2; thisNote = thisNote + 2) {

    // calculates the duration of each note
    divider = melody[thisNote + 1];
    if (divider > 0) {
        // regular note, just proceed
        noteDuration = (wholenote) / divider;
    } else if (divider < 0) {
        // dotted notes are represented with negative durations!!
        noteDuration = (wholenote) / abs(divider);
        noteDuration *= 1.5; // increases the duration in half for dotted
notes
    }
    tone(buzzer, melody[thisNote], noteDuration*0.9); //(COUTO, 2021)

    delay(noteDuration);
    noTone(buzzer);
}

loop();
}

void Tetris(){
int tempo=144;
int buzzer = 6;

int melody[] = {
    NOTE_E5, 4, NOTE_B4,8, NOTE_C5,8, NOTE_D5,4, NOTE_C5,8,
    NOTE_B4,8,
    NOTE_A4, 4, NOTE_A4,8, NOTE_C5,8, NOTE_E5,4, NOTE_D5,8,
    NOTE_C5,8,
    NOTE_B4, -4, NOTE_C5,8, NOTE_D5,4, NOTE_E5,4,

```

```
NOTE_C5, 4, NOTE_A4,4, NOTE_A4,8, NOTE_A4,4, NOTE_B4,8,
NOTE_C5,8,
```

```
NOTE_D5, -4, NOTE_F5,8, NOTE_A5,4, NOTE_G5,8, NOTE_F5,8,
NOTE_E5, -4, NOTE_C5,8, NOTE_E5,4, NOTE_D5,8, NOTE_C5,8,
NOTE_B4, 4, NOTE_B4,8, NOTE_C5,8, NOTE_D5,4, NOTE_E5,4,
NOTE_C5, 4, NOTE_A4,4, NOTE_A4,4, REST, 4,
```

```
};
```

```
int notes=sizeof(melody)/sizeof(melody[0])/2;
int wholenote = (60000 * 4) / tempo;
int divider = 0, noteDuration = 0;
for (int thisNote = 0; thisNote < notes * 2; thisNote = thisNote + 2) {

    // calculates the duration of each note
    divider = melody[thisNote + 1];
    if (divider > 0) {
        // regular note, just proceed
        noteDuration = (wholenote) / divider;
    } else if (divider < 0) {
        // dotted notes are represented with negative durations!!
        noteDuration = (wholenote) / abs(divider);
        noteDuration *= 1.5; // increases the duration in half for dotted
notes
    }
    tone(buzzer, melody[thisNote], noteDuration*0.9); //(COUTO, 2021)
```

```
delay(noteDuration);
```

```
noTone(buzzer);
```

```
}
```

```
loop();
```

```
}
```