

Implementação de um Teclado Musical com Arduino

Pedro Henrique Carvalho Lima Cabral¹, Maria Klara Nunes Martins de Oliveira²,
Daniel dos Anjos Costa³

¹ Instituto Federal da Bahia (IFBA) – Euclides da Cunha – BA – Brasil

² Instituto Federal da Bahia (IFBA) – Euclides da Cunha – BA – Brasil

³ Colegiado de Informática – Instituto Federal da Bahia (IFBA) – Euclides da Cunha – BA –
Brasil

phclc2015@gmail.com, mariaklaranuness31@hotmail.com, daniel.anjos@ifba.edu.br

Abstract. *This project deals with the implementation of a musical keyboard at the logical and physical level. In the logical project developed we made a keyboard with 24 musical notes and full expansion possibility using encoder integrated circuits. In the physical implementation we developed a reduced version keyboard with only 16 musical notes, because of the complexity of its assembly. We used the arduino UNO to process the input signals coming from independent buttons and output a square wave electrical signal at the desired note frequency. To solve the limitation of amount of digital inputs of the arduino we implemented an encoding module.*

Resumo. *Este trabalho aborda a implementação de um teclado musical a nível lógico e físico. No projeto lógico desenvolvido confeccionamos um teclado com 24 notas musicais e plena possibilidade de expansão utilizando circuitos integrados codificadores. Na implementação física desenvolvemos uma versão reduzida teclado com 16 notas musicais, por causa da complexidade da sua montagem. Utilizamos o arduino UNO para processar os sinais de entrada vindo de botões independentes e emitir um sinal elétrico de onda quadrada na frequência da nota desejada. Para solucionar a limitação da quantidade de entradas digitais do arduino implementamos um módulo de codificação.*

1. INTRODUÇÃO

A música é uma manifestação artística onde se une diferentes ritmos e sons para formar composições sonoras agradáveis. Ela surgiu há milhares de anos, pois sempre foi do interesse humano a criação harmônica de sons em manifestações musicais. Desde os primórdios da humanidade instrumentos simples como tambores e flautas já eram utilizados em criações musicais. Com o passar dos anos a música, a teoria musical e os instrumentos musicais foram evoluindo.

Em relação aos instrumentos musicais podemos destacar nitidamente sua evolução, que saiu de instrumentos feitos com peles de animais até instrumentos eletrônicos de alta complexidade (AIDAR, 2020). Muitos acreditam que a eletrônica revolucionou a história musical, pois melhorou e acelerou a construção de novos instrumentos musicais, além de torná-los mais acessíveis ao público.

Um exemplo clássico de um instrumento modernizado com eletrônica é o teclado musical, que é muito confundido com o piano clássico acústico. Apesar da semelhança física entre os dois instrumentos, os seus mecanismos de funcionamento são totalmente diferentes. O piano acústico possui pequenos martelos no seu interior que

são responsáveis pela emissão do som após uma tecla ser pressionada. Com o acionamento de uma tecla, o seu respectivo martelo colide com uma corda e a vibração gerada pela corda é propagada por meio do ar até os nossos ouvidos. Em contrapartida, no teclado eletrônico quando uma tecla é pressionada, um sinal elétrico é transmitido para o microcontrolador do teclado, que o identifica e o associa a uma nota musical. O microcontrolador emite um sinal elétrico na frequência e timbre da nota desejada. Geralmente, esse sinal é amplificado e repassado a uma caixa de som que o converte em ondas sonoras (TEORIA MUSICAL, [S. l.]).

Visando contribuir um pouco com a história da música, desenvolvemos neste trabalho um teclado musical utilizando Arduino. Um Arduino é uma placa eletrônica a qual torna possível a criação de variados protótipos eletrônicos. Com ele pode-se receber e emitir sinais elétricos (analógicos ou digitais), armazenar um código executável em sua memória flash e processar esse código com um microprocessador.

Na literatura encontramos vários trabalhos com o objetivo de construir um teclado musical utilizando Arduino, e todos eles com estratégias de implementação e componentes eletrônicos diferentes. Neste trabalho apresentamos apenas dois. No primeiro foi desenvolvido um teclado de 8 notas musicais com emissão do som realizada por um buzzer piezoelétrico. O autor associa cada nota musical a uma das entradas digitais do Arduino UNO. Isso limita a quantidade de notas emitida pelo teclado à quantidade de portas digitais que o Arduino possui (TECLADO COM ARDUÍNO, 2022). O segundo projeto foi desenvolvido por Cabral (2022). Com um Arduino UNO, o autor desenvolveu um teclado com 23 notas musicais e com possibilidade de expansão da quantidade de notas. No trabalho é apresentado um módulo de codificação que combina 4 codificadores HCF4532B em cascata com 3 portas OU, proporcionando a possibilidade de receber 24 sinais diferentes e emitir uma combinação de 5 bits de saída diferentes, o que reduz consideravelmente a quantidade de entradas necessárias do Arduino UNO.

Em nosso projeto lógico desenvolvemos um teclado com 24 notas musicais com possibilidade de expansão. Implementamos um módulo de codificação que permitiu reduzir pela metade a quantidade de sinais digitais enviados para o Arduino, ou seja, utilizamos apenas 12 portas digitais do Arduino. Nossa implementação se diferencia do trabalho de Cabral (2022) por reduzir a complexidade implementando a mesma quantidade de notas com o mesmo modelo de Arduino.

Nosso trabalho traz consigo uma dupla motivação. Em primeiro lugar, busca-se impulsionar a inovação na área de instrumentos musicais eletrônicos, oferecendo soluções acessíveis e tecnologicamente avançadas. Além disso, o projeto se enquadra como uma ferramenta educativa, promovendo o aprendizado prático em eletrônica, programação e música, contribuindo assim para o desenvolvimento de habilidades fundamentais em estudantes e entusiastas.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, apresentamos a Fundamentação Teórica que serve de base para o desenvolvimento do sistema proposto neste trabalho.

2.1. Teoria Musical

Pode-se definir a música como a junção de notas musicais de maneira harmônica, em que cada nota deve ser executada em um determinado momento e com duração específica.

As notas musicais possuem frequência específica e conhecida. Dentro da teoria musical existem 7 notas musicais (dó, ré, mi, fá, sol, lá e si) e 5 notas acidentais formadas por bemóis e sustenidos, que são: dó# ou réb, ré# ou mi \flat , fá# ou solb, sol# ou láb e lá# ou si \flat . As notas musicais são divididas em oitavas, e cada oitava sempre é formada pelas mesmas notas musicais. O que as diferencia são as frequências das notas que a compõem. Na Tabela 01 apresentamos as frequências musicais de cada nota e suas respectivas oitavas.

Tabela 01 - Distribuição da frequência das notas musicais pelas oitavas.

NOTAS MUSICAIS	1° OITAVA	2° OITAVA	3° OITAVA	4° OITAVA	5° OITAVA	6° OITAVA	7° OITAVA	8° OITAVA	9° OITAVA
Dó	33	66	132	264	528	1056	2112	4224	8448
Dó #	34.947	69.894	139.79	279.6	559.15	1118.3	2236.6	4473.2	8946.4
Ré	37.026	74.052	148.1	296.2	592.42	1184.8	2369.7	4739.3	9478.7
Ré#	39.237	78.474	156.95	313.9	627.79	1255.6	2511.2	5022.3	10045
Mi	41.58	83.16	116.32	332.6	665.28	1330.6	2661.1	5322.2	10644
Fá	44.055	88.11	176.22	352.4	704.88	1409.8	2819.5	5639	11278
Fá#	46.662	93.324	186.65	373.3	746.59	1493.2	2986.4	5972.7	11945
Sol	49.434	98.868	197.74	395.5	790.94	1581.9	3163.8	6327.6	12655
Sol#	52.371	104.74	209.48	419	837.94	1675.9	3351.7	6703.5	13407
Lá	55.506	111.01	222.02	444	888.1	1776.2	3552.4	7104.8	14210
Lá#	58.806	117.61	235.22	470.4	940.9	1881.8	3763.6	7527.2	15054
Si	62.304	124.61	249.22	498.4	996.86	1993.7	3987.5	7974.9	15950
Dó	66	132	264	528	1056	2112	4224	8448	16896

Para descobrir matematicamente como se calcula a frequência de uma nota musical, devemos lembrar dos conceitos básicos de física ondulatória utilizando a fórmula $V = \lambda.f$ (velocidade = lambda multiplicado pela frequência). Isolando a frequência na fórmula tradicional, encontramos a frequência da onda, ficando $f = \lambda/V$.

2.2. Arduino

O Arduino é uma placa para a construção de protótipos eletrônicos com componentes de processamento, armazenamento, entrada e saída de sinais elétricos. Ele tem um microcontrolador para processar o código embarcado, pode receber sinais analógicos/digitais e emitir sinais digitais.

Nesta seção descreveremos o Arduino UNO, pois foi o que usamos no projeto. O Arduino UNO possui o Microcontrolador ATmega328P que é um componente de processamento responsável pela execução do código embarcado, interpretação de sinais recebidos e emissão de sinais atuadores (NUSSEY, 2019). Ele também possui 14 pinos

de Entrada/Saída Digital e 6 pinos de entrada analógica. Os pinos digitais podem receber tensões entre 0,0 V e 5,0V e os seus extremos representam os valores lógicos “LOW” e “HIGH”, ou “0” lógico e “1” lógico respectivamente (DOCUMENTAÇÃO DO ARDUÍNO, 2022).

2.3. Codificador

O codificador é um circuito integrado que possui um conjunto limitado de entradas diferentes e um valor de saída para cada uma delas. Por causa de sua característica esse componente é classificado por muitos como um tradutor, pois ele associa diretamente a cada uma das combinações de entradas lógicas uma única saída. Vale ressaltar que apenas uma entrada pode ser acionada por vez (GUIMARÃES, 2019). O valor de saída será um código binário que representa qual entrada está sendo acionada (SOUZA e FERREIRA, 2023).

O codificador utilizado neste projeto foi o 74HC148, sendo ele do tipo *8-to-3*, pois ele converte 8 sinais de entrada (0 a 7) para três pinos de saída de dados binários (A0 a A2) (DATASHEET DO 74HC148, 1992).

3. DESENVOLVIMENTO

Neste capítulo apresentaremos o detalhamento da implementação do teclado musical. O teclado desenvolvido no projeto lógico tem 24 teclas com possibilidade de expansão, entretanto na implementação física conseguimos colocar apenas 16 notas (Dó, Dó#, Ré, Ré#, Mi, Fá, Fá#, Sol, Sol#, Lá, Lá#, Si, Dó, Dó#, Ré, Ré#). Optamos em não implementar todas as notas por causa das limitações de componentes eletrônicos disponíveis e por causa do espaço na *proto-board*.

O projeto foi dividido em quatro partes, a saber: Módulos das Teclas, Módulo de Codificação, Arduino e Módulo de Emissão do Som. A Figura 01 exibe uma ilustração da organização dos módulos no projeto. O primeiro módulo foi desenvolvido com botões e resistores ligados a uma fonte de tensão de 5 volts e tem o objetivo de emitir 24 sinais elétricos diferentes (de 5V ou 0V simbolizando “1” ou “0” lógicos). Cada um dos botões será associado a uma nota musical do teclado.

O segundo módulo é o de codificação que tem a função de reduzir a quantidade de sinais de entrada, convertendo-os em sinais com codificação BCD (do inglês *Binary Coded Decimal*) diferente. O penúltimo módulo é o do Arduino que recebe os códigos binários do módulo de codificação, processa e emite o um sinal em formato de onda quadrada para o último módulo que é o de emissão do som, que lança as ondas sonoras no ar.

3.1. Módulo de Codificação

Um teclado convencional de 5 oitavas tem 61 teclas e um piano acústico ou eletrônico possui 88 teclas. A implementação de um protótipo de teclado com 5 oitavas (61 teclas) necessitaria de um Arduino com quantidade igual ou maior de entradas digitais, o que

tornaria mais caro os custos de implementação. Para solucionar este problema, propomos neste trabalho o módulo de codificação que reduz pela metade os sinais de entrada emitidos pelos botões e os converte para codificação binária BCD que é repassada para o Arduino.

No trabalho de Cabral (2022), foi proposto um módulo semelhante ao nosso com uso de codificadores em cascata. A abordagem de Cabral (2022) consegue reduzir os sinais de entrada vindo dos botões em uma quantidade menor de sinais de saída no formato BCD, entretanto a sua implementação tem alta complexidade e necessita de circuitos integrados com a porta OU.

Pensando na implementação física do projeto que é limitada pelos recursos eletrônicos disponíveis, desenvolvemos um módulo de codificação que possui apenas codificadores, conforme ilustrado na Figura 01. Observando a região de destaque do Módulo de Codificação percebemos que existem três codificadores 74HC148 nomeados com as siglas: CI_01, CI_02 e CI_03. Cada codificador recebe oito sinais digitais elétricos, o que dá um total de 24 sinais diferentes. As entradas do codificador CI_01 foram nomeadas com as siglas E1, E2, E3, E4, E5, E6, E7 e E8; as entradas do codificador CI_02 foram nomeadas com as siglas E9, E10, E11, E12, E13, E14, E15 e E16; e por fim as entradas do codificador CI_03 foram nomeadas com E17, E18, E19, E20, E21, E22, E23 e E24.

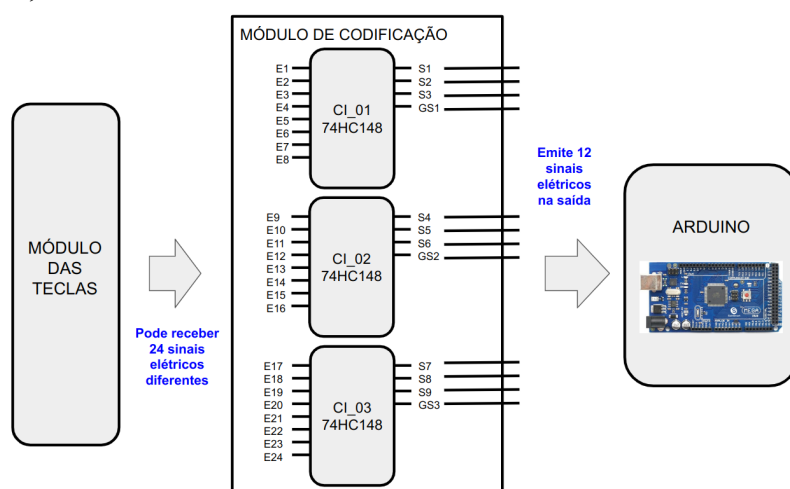


Figura 1 - Ilustração do Módulo de Codificação.

As saídas do codificador CI_01 foram nomeadas com S1, S2, S3 e GS1; as saídas de CI_02 foram nomeadas com S4, S5, S6 e GS2; e as saídas de CI_03 foram nomeadas com S7, S8, S9 e GS3. Observe que na Figura 01 as saídas dos codificadores são repassadas diretamente para o Arduino.

Na Tabela 01 apresentamos as combinações de entrada e saída realizadas pelo módulo de codificação. Na coluna sinais binários foi colocada sequências de 24 bits, que representam os sinais vindo dos botões do módulo anterior. Pode ser observado que em cada linha dessa coluna dentre os 24 bits, 23 tem valor lógico “1” e apenas um tem valor lógico “0”. Isso significa que apenas uma das teclas do teclado está sendo

pressionada, pois conforme explicado na seção anterior, quando a tecla é pressionada ela emite sinal 0v (“0” lógico) e quando a tecla não está pressionada ela emite sinal 5v (“1” lógico).

Os sinais de saída foram divididos em três grupos correspondentes aos codificadores associados. É importante destacar que existe uma ordem de prioridade entre os codificadores, ou seja, o CI_01 tem maior prioridade do que o CI_02 e CI_03, e o CI_02 tem maior prioridade que o CI_03. Essa lógica de prioridade foi implementada para evitar conflitos quando o usuário pressionar duas ou mais teclas ao mesmo tempo.

Quando o módulo de codificação recebe a sequência de sinais “11111111111111111110” (primeira linha da Tabela 02) o codificador CI_01 emite os seguintes valores: S1=1, S2=1, S3=1 e GS1=0. O sinal 0 emitido pelo GS1 indica que as outras saídas do codificador CI_01 são válidas. Com isso os valores de S1, S2 e S3 serão interpretados pelo Arduino na próxima etapa e é emitida a nota DÓ.

Tabela 02 - Combinações binárias de entrada e saída do módulo de codificação.

LINHA	ENTRADAS SINAIS BINÁRIOS	SAÍDAS												NOTA
		CI_01				CI_02				CI_03				
		S1	S2	S3	GS1	S4	S5	S6	GS2	S7	S8	S9	GS3	
1	11111111111111111110	1	1	1	0	X	X	X	X	X	X	X	X	DÓ
2	11111111111111111101	0	1	1	0	X	X	X	X	X	X	X	X	DÓ#
3	111111111111111111011	1	0	1	0	X	X	X	X	X	X	X	X	RE
4	1111111111111111110111	0	0	1	0	X	X	X	X	X	X	X	X	RE#
5	1111111111111111101111	1	1	0	0	X	X	X	X	X	X	X	X	MI
6	11111111111111111011111	0	1	0	0	X	X	X	X	X	X	X	X	FÁ
7	111111111111111110111111	1	0	0	0	X	X	X	X	X	X	X	X	FÁ#
8	1111111111111111101111111	0	0	0	0	X	X	X	X	X	X	X	X	SOL
9	11111111111111111011111111	X	X	X	X	1	1	1	0	X	X	X	X	SOL#
10	111111111111111110111111111	X	X	X	X	0	1	1	0	X	X	X	X	LÁ
11	1111111111111111101111111111	X	X	X	X	1	0	1	0	X	X	X	X	LÁ#
12	11111111111111111011111111111	X	X	X	X	0	0	1	0	X	X	X	X	SI
13	111111111111111110111111111111	X	X	X	X	1	1	0	0	X	X	X	X	DÓ
14	1111111111111111101111111111111	X	X	X	X	0	1	0	0	X	X	X	X	DÓ#
15	11111111111111111011111111111111	X	X	X	X	1	0	0	0	X	X	X	X	RE
16	111111111111111110111111111111111	X	X	X	X	0	0	0	0	X	X	X	X	RE#
17	1111111111111111101111111111111111	X	X	X	X	X	X	X	X	1	1	1	0	MI
18	11111111111111111011111111111111111	X	X	X	X	X	X	X	X	0	1	1	0	FÁ
19	111111111111111110111111111111111111	X	X	X	X	X	X	X	X	1	0	1	0	FÁ#
20	1111111111111111101111111111111111111	X	X	X	X	X	X	X	X	0	0	1	0	SOL
21	11111111111111111011111111111111111111	X	X	X	X	X	X	X	X	1	1	0	0	SOL#
22	111111111111111110111111111111111111111	X	X	X	X	X	X	X	X	0	1	0	0	LÁ
23	1111111111111111101111111111111111111111	X	X	X	X	X	X	X	X	1	0	0	0	LÁ#
24	0111111111111111111111111111111111111111	X	X	X	X	X	X	X	X	0	0	0	0	SI

Pode-se observar na tabela que as demais saídas dos outros codificadores tem valor X. Esse valor indica que não importa o valor lógico atribuído às demais saídas dos outros codificadores, pois apenas as saídas do codificador CI_01 serão interpretadas, porque ele tem maior prioridade que os demais.

Na segunda linha a sequência de entrada é “111111111111111111101”. Pode-se observar que a segunda nota está sendo pressionada. Quando o módulo de codificação recebe essa entrada ele emite os seguintes valores nas saídas do codificador

CI_01: S1=0, S2=1, S3=1 e GS1=0. Esses quatro sinais serão interpretados pelo Arduino e será emitido a nota DÓ#. O primeiro codificador atuará até a sequência “111111111111111101111111” que está na oitava linha.

Quando o módulo de Codificação recebe a sequência “111111111111111101111111” (nona linha) é ativada as saídas do codificador CI_02 e desativada as saídas do demais. O CI_02 emite as seguintes saídas: S4=1, S5=1, S6=1 e GS2=0. Dessa forma, quando as saídas forem interpretadas pelo Arduino será emitida a nota SOL#. O segundo codificador atuará até a sequência “111111101111111111111111” que está na décima sexta linha. E o codificador CI_03 atuará entre as entradas “11111110111111111111111111” e “011111111111111111111111” (linhas 17 e 24 respectivamente).

Para ampliação da quantidade de notas do teclado é necessário ampliar a quantidade de codificadores. Nessa configuração, o módulo de codificação consegue reduzir pela metade a quantidade de sinais elétricos digitais que será transmitido ao Arduino.

3.2. Montagem e Programação do Arduino

Programamos o Arduino para reconhecer as combinações de sinais de entrada que correspondem às notas do teclado (exibido na Tabela 02) e emitir sua respectiva frequência em sinal elétrico, conforme exibido no pseudocódigo na Figura 02.

Na função “loop” programamos o Arduino em uma sequência de condicionais em cascata. A primeira camada de condicionais verifica qual é o codificador que está enviando código binário válido. Para isso, verifica-se qual das entradas SG1, SG2 e SG3 está recebendo valor lógico “0”. A segunda camada de condicionais verifica se o código BCD recebido dos sinais S1, S2, S3 se SG1=0; S4, S5, S6 se SG2=0; e S7, S8, S9 se SG3=0. Cada código BCD corresponde a uma respectiva nota que será gerada pelo Arduino.

A função que emite a nota musical gera um sinal de saída na sua respectiva frequência em formato de sinal elétrico de onda quadrada, conforme mostrado na ilustração da Figura 03. Na parte superior da figura, exibimos uma onda quadrada na frequência de 132Hz (frequência da nota Dó de 3ª oitava conforme Tabela 01) e na parte inferior da mesma figura é exibido uma onda quadrada na frequência de 148Hz (frequência da nota Ré de 3ª oitava).

Se uma nota musical tem frequência característica de 132 Hz, significa que ela tem 132 repetições ou oscilações do seu período em um segundo. No nosso caso, o período da onda que estamos formando corresponde a subida do sinal elétrico, a sua estabilização (por uma fração de X microssegundos), a descida do sinal e sua estabilização novamente por X microssegundos.

O controle do tempo de estabilização em microssegundos permite controlar a quantidade de variações que o sinal de saída terá em um segundo. Para calcular o

respectivo tempo de estabilização do sinal de saída de cada nota musical utilizamos a Equação 1 mostrada abaixo:

$$\text{Tempo de Estabilização} = 1.000.000 / (\text{Frequência da Nota} * 2) \quad (1)$$

```

2
3 void setup() {
4   // DEFINIÇÃO DOS PINOS DE ENTRADA E SAÍDA.
5 }
6
7 void loop() {
8
9   if(GS1 == 0) {
10    if (S1 == 1 && S2 == 1 && S3 == 1 ) { /* EMITIR NOTA DÓ */ }
11    else if(S1 == 0 && S2 == 1 && S3 == 1 ) { /* EMITIR NOTA DÓ# */ }
12    // (...)
13    else if(S1 == 0 && S2 == 0 && S3 == 0 ) { /* EMITIR NOTA SOL */ }
14  }
15  else if(GS2 == 0) {
16    if (S1 == 1 && S2 == 1 && S3 == 1 ) { /* EMITIR NOTA SOL# */ }
17    // (...)
18    else if(S1 == 0 && S2 == 0 && S3 == 0 ) { /* EMITIR NOTA RE# */ }
19  }
20  else if(GS3 == 0) {
21    if (S1 == 1 && S2 == 1 && S3 == 1 ) { /* EMITIR NOTA MI */ }
22    // (...)
23    else if(S1 == 0 && S2 == 0 && S3 == 0 ) { /* EMITIR NOTA SI */ }
24  }
25  else { /* NÃO EMITIR SOM */ }
26
27 }

```

Figura 2 - Pseudocódigo ilustrando a programação do Arduino.

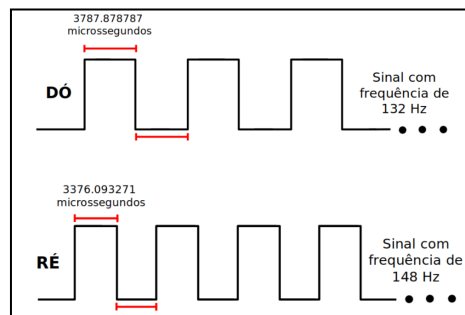


Figura 3 - Exemplificação de como o sinal de saída do Arduino foi gerado com uma respectiva frequência.

O valor 1.000.000 corresponde a quantidade de microssegundos existentes em um segundo e é dividido pelo valor da frequência da nota multiplicado por dois. Com o valor do tempo de estabilização de cada nota em mãos, usamos uma das saídas digitais do Arduino para alternar entre sinal alto (high ou “1” lógico) e sinal baixo (low ou “0” lógico) na frequência de cada nota.

Observe na Figura 03 parte superior, que corresponde a nota Dó, o tempo de estabilização do sinal é de 3787.878787 microssegundos. Esse valor foi obtido dividindo 1000000 por 132 vezes 2. E na mesma figura na parte inferior, que corresponde a nota Ré o tempo de estabilização é de 3376.093271 microssegundos, obtido dividindo 1000000 por 148 vezes 2. Os valores do Tempo de Estabilização foram calculados para cada uma das notas do teclado.

3.3. Módulo de Emissão de Som

No módulo de Emissão do Som implementamos duas abordagens. A primeira utilizando o componente eletrônico chamado Piezo. Escolhemos esse componente eletrônico por ele ser barato, fácil de utilizar e se adequar ao escopo do projeto. A primeira perna do Piezo ligamos ao pino de saída do Arduino com um resistor de $1k\Omega$, e a segunda perna conectamos ao potencial zero ou terra do circuito, conforme apresentado na Figura 04a.

A segunda abordagem ligamos em paralelo ao circuito do Piezo um terminal fêmea P2 para ligarmos o teclado em uma caixa de som amplificada. A Figura 04b ilustra o terminal fêmea P2 e o terminal macho P2. Escolhemos esse terminal para incrementar o projeto porque o terminal P2 macho é de fácil acesso e geralmente acompanha caixas amplificadas de computador.

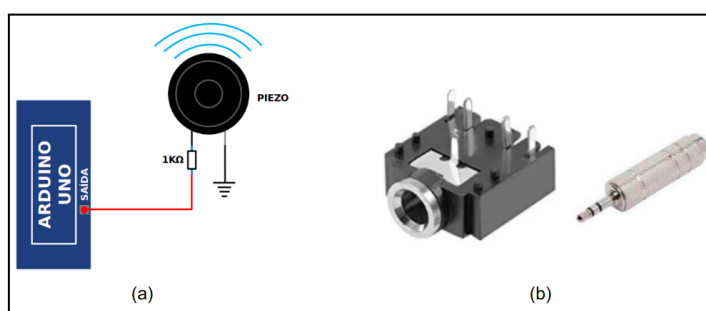


Figura 4 - (a) Ilustração mostrando o esquema de ligação do Piezo. (b) Terminal P2 fêmea e macho.

Na Figura 05 exibimos a montagem do protótipo em uma placa de ensaio. Na mesma figura podemos observar que o Arduino ainda está ligado ao computador via USB para transmissão do código executável e para alimentação energética.

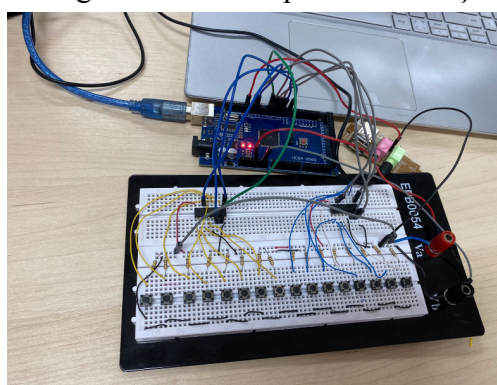


Figura 5 - Foto da implementação física do projeto.

4. CONCLUSÃO

Conforme apresentado nas seções anteriores, esse trabalho desenvolveu um projeto de um teclado musical de 24 notas (projeto lógico) e 16 notas (implementação física) utilizando um Arduino UNO. Utilizou-se um módulo de codificação intermediando a

troca de sinais entre os botões e do Arduino, e com isso reduzimos pela metade a quantidade de portas digitais utilizadas. O módulo de codificação implementado foi desenvolvido para reduzir a quantidade de sinais elétricos enviados para o Arduino.

Enfim, consideramos que o desenvolvimento do projeto foi um sucesso, pois cumprimos as premissas iniciais e ainda ampliamos a abrangência do mesmo. Esse projeto tem um grande potencial de aperfeiçoamento e podemos listar como continuidade os seguintes objetivos: 1) implementar um teclado com 1 notas e cinco oitavas; 2) melhorar do timbre emitido pelo Arduino; 3) criação de um circuito de amplificação do sinal de saída do Arduino para que o som seja emitido em uma caixa de som convencional; e 4) montar circuitos acessórios para permitir que o usuário possa mudar o timbre do teclado manualmente.

Em síntese, o desenvolvimento deste teclado musical com base no Arduino não apenas representa uma realização tecnológica, mas também abre portas para uma vasta gama de possibilidades. Desde sua aplicação como uma ferramenta educacional que capacita estudantes a explorarem os campos da eletrônica, programação e música, até seu potencial como um instrumento inovador para músicos e criadores.

REFERÊNCIAS BIBLIOGRÁFICAS

- AIDAR, Laura . Toda matéria, [S. l.], 2020?. Disponível em : <https://www.todamateria.com.br/historia-da-musica/> > Acesso em: 09 fev. 2023.
- CABRAL, Pedro H. C. **PROJETO DE UM TECLADO MUSICAL UTILIZANDO ARDUÍNO**. 2022. 35 f. TCC (Técnico em Informática) - IFBA, Euclides da Cunha-BA, 2022.
- DATASHEET DO 74HC148, 1992. Disponível em: <<https://pdf1.alldatasheet.com/datasheet-pdf/view/23041/STMICROELECTRONIS/74HC148.html> > Acesso em: 27 jan. 2023.
- DOCUMENTAÇÃO DO ARDUINO, 2023. Disponível em: <<https://reference.arduino.cc/reference/pt/>>. Acesso em: 13 de fev. 2023.
- GUIMARÃES, Fábio. Codificador e decodificador, 2019. Disponível em: <<https://mundoprojetado.com.br/codificador-e-decodificador-aula-7-ed/> >. Acesso em: 27 jan. 2023.
- NUSSEY, John. Arduino para leigos. 2. ed. São Paulo: Alta Books, 2019.
- SOUZA, Joedson. e FERREIRA. Magnum. **DESENVOLVIMENTO DE UM BRAÇO ROBÓTICO COM CONTROLE MULTIMODAL USADO ESP8266 NODEMCU**. 2023. 68 f. TCC (Técnico em Informática) - IFBA, Euclides da Cunha-BA, 2023.
- TECLADO COM ARDUINO. Amazonas: Universidade do Estado do Amazonas, 2022. Disponível em: <<https://stem.uea.edu.br/portfolio/aula12/>>. Acesso em: 13 fev. 2023.
- TEORIA MUSICAL. Descomplicando a música. [S. l.] Disponível em: <<https://www.descomplicandoamusica.com/curso-de-teclado-descomplicando-a-musica/> > Acesso:14 de fev.2023.