PROFESSOR DANILO *ROBÓTICA – 9° ANO – 11/08/2022*

AULA 22 – PIEZO PARTE I

Vamos utilizar um piezo para emitir sons através do Arduino.

PIEZO

A piezeletricidade é uma propriedade de alguns materiais que consiste em produzir eletricidade quando comprimido ou esticado. O processo inverso também ocorre: ao aplicar uma diferença de potencial em um material piezelétrico, este material se comprime ou expande.

Tais materiais possuem extensas aplicações, indo desde seu uso como microfone ou alto-falante até sensores sísmicos ou clock. O Arduino possui um ou dois cristais de quartzo, com propriedade piezelétrica, para medir a passagem de tempo durante a execução das rotinas.

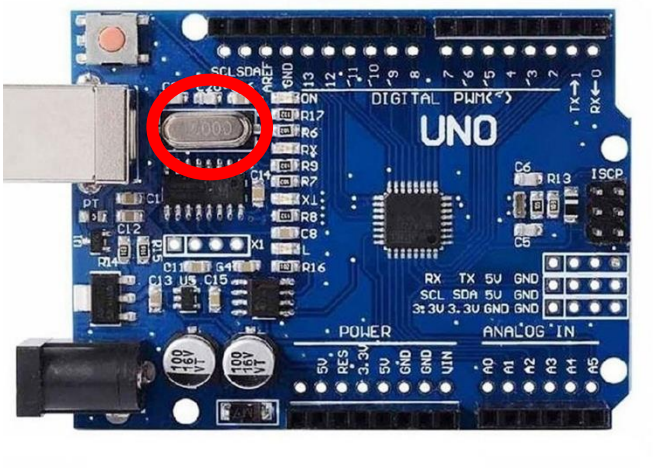

Figura 1: Arduino UNO com destaque para o clock

Para a nossa aplicação temos dois tipos comuns de piezos: o passivo e o ativo.

O piezo passivo é um circuito simples que, para ser usado como um mini alto-falante, deve ser submetido à uma tensão variável. Por exemplo, para tocar a nota lá, de 440 Hz, deve receber uma tensão variável de 440 Hz.

O piezo ativo, por outro lado, emite um som específico pré-determinado quando simplesmente alimentamos o circuito com uma tensão específica.

Na Figura 2 vemos a diferença entre dois piezos, em suas formas comerciais, utilizados em circuitos elétricos e comuns em projetos com Arduino. Para diferenciar ambos, o piezo passivo possui o circuito na base exposto, já o ativo possui uma resina tampando a base. Além disso, o piezo passivo um pino maior (VCC) que o outro (GND).

Figura 2: piezo passivo à esquerda e ativo à direita

Tabela 1: principais diferenças entre os piezos aplicados ao Arduino

Passivo	**Ativo**
Mais fácil de testar e usar, pois basta aplicar uma ddp que ele emite um som	Difícil de testar, pois é necessário aplicar uma tensão variável. Se ligarmos uma ddp, ele emite um perqueno estalo.
É mais apropriado para ser usado como alarme/aviso/ sinalização.	É mais apropriado para emitir melodias e é o que utilizaremos em nossa aplicação.
Por norma, a parte de traz do buzzer é lacrada e um dos pinos é maior (VCC) e o outro, menor (GND).	Por norma, a PCB (placa com circuito impresso) fica à mostra e ambos os pinos possuem o mesmo tamanho.
	
Chamado buzer ativo.	Chamado buzer passivo.

Fonte: https://www.arduinoportugal.pt/qual-diferenca-buzzer-ativo-vs-buzzer-passivo/

Vamos dividir nosso estudo em três partes:

- Parte I: montagem do circuito, emissão de som selecionando a frequência e estudo matemático da onda sonora;
- Parte II: aprenderemos um pouco sobre funções e criaremos um programa para criarmos as diversas notas musicais.
- Parte III: criaremos alguma melodia e montaremos um mini piano.

Para nosso primeiro circuito precisamos de:

- Arduino Uno;
- dois fios;
- um buzer passivo;
- um LDR;
- uma placa de ensaio para montarmos o circuito.

O CIRCUITO

Na Figura 3 vemos o circuito que teremos que montar.

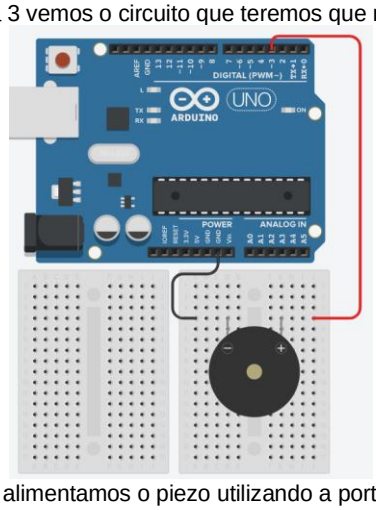

Figura 3: alimentamos o piezo utilizando a porta PWM 3.

PROFESSOR DANILO *ROBÓTICA – 9° ANO – 11/08/2022*

FUNÇÕES UTILIZADAS

Como novidade, vamos ver as funções `tone(porta, frequencia)`, `noTone(porta)` e `sin(anguloEmRadiano)`.

A primeira função é a que utilizaremos para informar em qual `porta` iremos ligar o positivo do piezo e qual a `frequencia` que o piezo irá emitir.

A segunda função, `noTone(porta)`, é para interromper o som que está sendo emitido.

A terceira função faz parte de um conjunto de funções matemáticas que o Arduino pode executar, calculando o seno de determinado ângulo. A função `sin(anguloEmRadiano)` recebe como argumento um ângulo em radianos e retorna o seno deste ângulo.

Precisaremos então convertermos o ângulo em radianos.

PRIMEIRO CÓDIGO

```
int frequencia =440, porta = 3;

void setup() {
  pinMode(porta, OUTPUT);
}

void loop() {
    tone(porta, frequencia);
    delay(2000);
    noTone(porta);
    delay(5000);
}
```

DISCUSSÃO DO PRIMEIRO CÓDIGO

Programa simples que emite um som com a frequência que você escolher. Começamos definindo as variáveis que determinam a frequência e a porta a ser utilizada (deve ser uma PWM).

```
int frequencia =440, porta = 3;
```

Na função setup, configuramos a porta escolhida como output.

```
void setup() {
  pinMode(porta, OUTPUT);
}
```

Por fim, usamos uma das funções vista acima para que o Arduino inicie a emissão de um som:

```
void loop() {
    tone(porta, frequencia);
```

E, após a espera de algum tempo (2000 milissegundos = 2 segundos), ele interrompe o som, espera mais algum tempo (5000 milissegundos = 5 segundos) e reinicia o loop.

```
    delay(2000);
    noTone(porta);
    delay(5000);
}
```

SEGUNDO CÓDIGO

Agora é com você: observe a Tabela 2 e perceba que para cada nota musical temos uma frequência específica.

Tabela 2: frequências de algumas notas musicais

Nota	Frequência (Hz)
dó	264
ré	297
mi	330
fá	352
sol	396
lá	440
si	495

Faça um programa que toque alguma música. Por exemplo, tente a seguinte sequência:

Dó, ré, mi, fá, fá, fá,
Dó, ré, do, ré, ré, ré,
Do, sol, fá, mi, mi, mi,
Do, ré, mi, fá, fá, fá (2x)

Experimente trocar os intervalos de tempos entre cada nota para ver como fica a melodia final.

O professor tentou montar um código para a primeira sequência de notas e o colou abaixo. Será que o seu professor é um bom compositor? Provavelmente você e seu grupo consegue fazer algo muito melhor.

```
int porta = 3,
  Do = 264,
  Re = 297,
  Mi = 330,
  Fa = 352,
  Sol = 396,
  La = 440,
  Si = 495;

void setup() {
  pinMode(porta, OUTPUT);
}

void loop() {
    tone(porta, Do);
    delay(200);
    noTone(porta);
    tone(porta, Re);
    delay(200);
    noTone(porta);
    tone(porta, Mi);
    delay(200);
    noTone(porta);
    tone(porta, Fa);
    delay(200);
    noTone(porta);
    tone(porta, Fa);
    delay(200);
    noTone(porta);
    tone(porta, Fa);
    delay(200);
    noTone(porta);
    delay(10000);
}
```