Projeto No. 4 – Buzzer

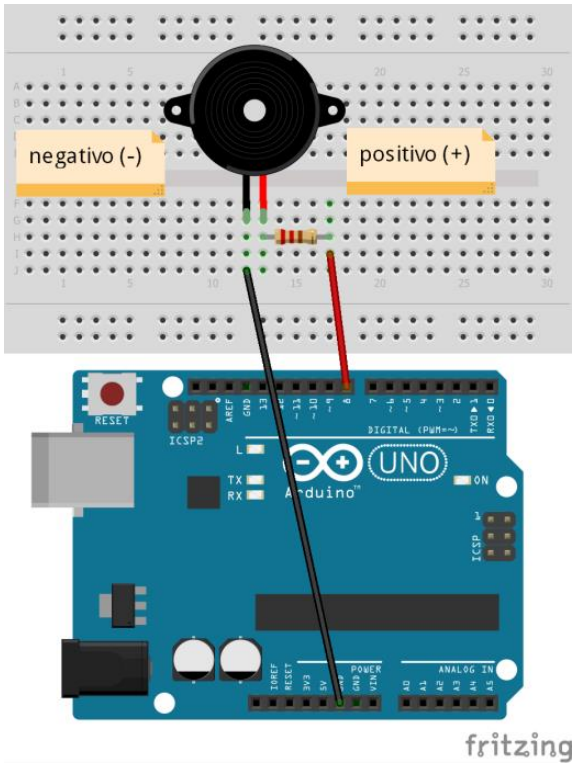
O objetivo deste projeto é acionar um buzzer utilizando as funções *tone()* e *noTone()*. Buzzer é um dispositivo para geração de sinais sonoros (*beeps*), como aqueles encontrados em computadores. Para a emissão do som, o buzzer vibra através de um oscilador. Essa oscilação é determinada por uma frequência, que por sua vez define um som específico. Nesse experimento será usado 2 modos de uso da função *tone()* e uma pequena variação de sons representando notas musicais. A função *tone()* possui 2 sintaxes: *tone(pino, frequência)* e *tone(pino, frequência, duração)*, onde **pino** referencia qual é o pino que irá gerar a frequência (ligado ao positivo do buzzer), a **frequência** é definida em hertz e a **duração** (opcional) é em milissegundos. Caso opte pela sintaxe sem duração é necessário usar a função *noTone(pino)* para parar a frequência enviada pelo pino definido.

Material necessário:

- 1 Arduino
- 1 Buzzer*
- 1 Resistor de 220 ohms (vermelho, vermelho, marrom) ou 330 ohms (laranja, laranja, marrom) para o buzzer*
- 1 Protoboard*
- Jumper cable

* *Podem ser substituídos pelo módulo P15-Buzzer da GBK Robotics.*

Passo 1: Montagem do circuito



Conforme ilustra a figura ao lado:

a. Coloque o buzzer na protoboard;

b. Conecte o pino GND do Arduino ao pino negativo do buzzer;

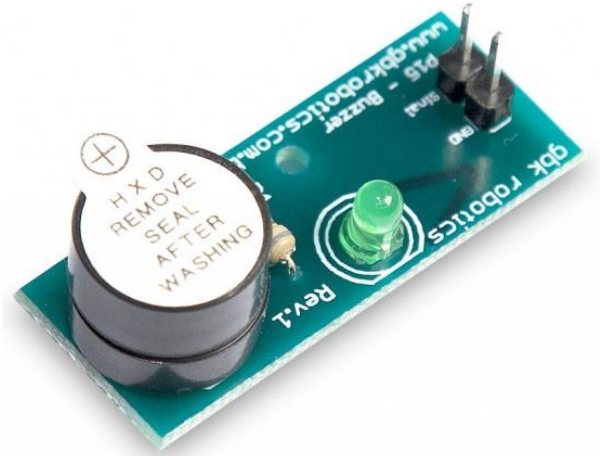
c. Coloque o resistor de 220 ohms (ou 330 ohms) ligado ao pino positivo do buzzer;

d. Nesse resistor, conecte um jumper até a porta digital 8 do Arduino.

FATECINO
Clube de Arduino
Fatec Jundiaí

Variação de Montagem

Módulo P15-Buzzer da GBK Robotics



Este projeto pode ser montado substituindo o buzzer, o resistor de 220 ohms (ou 330 ohms) e a protoboard pelo módulo P15-Buzzer da GBK Robotics, neste caso:

a. Conecte o pino GND do Arduino ao pino GND do módulo P15;
b. Conecte o pino 8 do Arduino ao pino Sinal do módulo P15.

IMPORTANTE: Não há alterações no sketch (programa).

Passo 2: Programa 1 – Uso das funções *tone(pino, frequência)* e *noTone(pino)*

Inicie o ambiente de desenvolvimento do Arduino e digite o sketch (programa) a seguir:

```
int buzzer = 8;// pino no qual o buzzer está conectado

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

void loop()
{
 tone(buzzer, 1200);//define pino e a frequência
 delay(500);
 noTone(buzzer);//desliga o buzzer
 delay(500);
}
```

Passo 3: Programa 2 – Uso a função *tone(pino, frequência, duração)*

Inicie o ambiente de desenvolvimento do Arduino e digite o sketch (programa) a seguir:

```
int buzzer = 8;// Pino no qual o buzzer está conectado

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

void loop()
{
  tone(buzzer, 1200, 500); // Define pino, a frequência e duração
  delay(1000);
}
```

Passo 4: Programa 4 – notas musicais com buzzer

Inicie o ambiente de desenvolvimento do Arduino e digite o sketch (programa) a seguir:

```
int buzzer = 8;// Pino no qual o buzzer está conectado

int numNotas = 10;
//vetor de inteiros com frequências diversas
int notas[] = {261, 277, 294, 311, 330, 349, 370, 392, 415, 440};
//notas          C  C#   D    D#   E    F    F#   G    G#   A
/*
C=Dó D=Ré E=Mi F=Fá G=Sol A=Lá B=Si
As notas sem o "#", são as notas naturais (fundamentais).
Aquelas com o "#", são chamadas "notas sustenidas" (por exemplo: C#=Dó Sustenido).
*/
void setup()
{
  pinMode(buzzer, OUTPUT);
}

void loop()
{
  for (int i = 0; i < numNotas; i++)
  {
    tone(buzzer, notas[i]);
    delay(500);
  }
  noTone(buzzer);
}
```