

Vamos montar um mini piano. A parte mais difícil será montar o circuito.

### CIRCUITO

Você vai precisar de:

- Um piezo passivo;
- 7 botões do tipo *pushbutton* (Figura 1);
- 16 fios MM (Figura 2)

Usaremos INPUT\_PULLUP do Arduino que fará com que a porta conectada ao botão seja conectada ao 5 V quando o botão estiver liberado. Quando pressionado, o botão conecta a porta de entrada ao GND, portanto, para o Arduino, quando o botão está liberado, a entrada será HIGH e, quando pressionado, será LOW.

Atenção: observe a figura abaixo e note que, pela sua descrição, você deverá por exemplo conectar o GND no pino inferior e a porta digital no pino da direita (ou no que não está sendo exibido na figura).



Figura 1: Pushbutton. Os pinos abaixo e a esquerda da figura estão conectados; o pino abaixo e a direita só se conectam quando apertamos o botão.

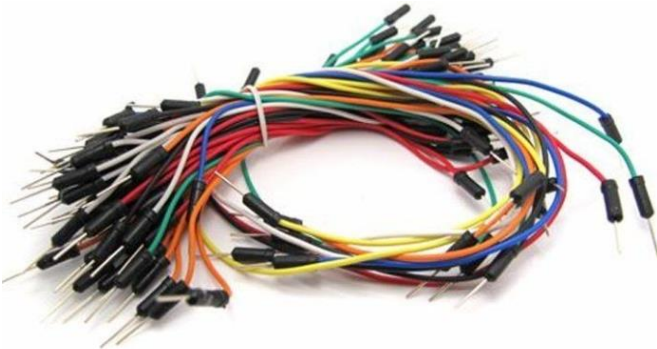


Figura 2: fios jumper MM.

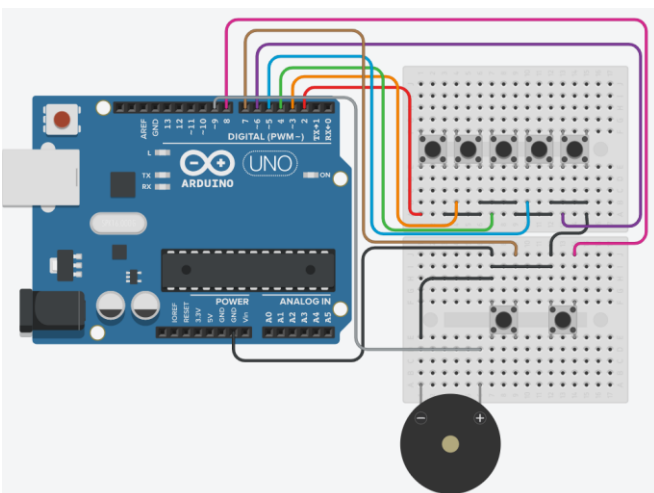


Figura 3: Circuito do mini piano.

### CÓDIGO

```
//portas
int portaDo = 2, portaRe = 3, portaMi = 4,
    portaFa = 5, portaSol = 6,
    portaLa = 7, portaSi = 8;

//notas (frequências)
int notaDo = 264, notaRe = 297, notaMi = 330,
    notaFa = 352, notaSol = 396,
    notaLa = 440, notaSi = 495;

//piezo
int portaPiezo = 9;

//duração da nota
int duracao = 200;

void setup()
{
    pinMode(portaDo, INPUT_PULLUP);
    pinMode(portaRe, INPUT_PULLUP);
    pinMode(portaMi, INPUT_PULLUP);
    pinMode(portaFa, INPUT_PULLUP);
    pinMode(portaSol, INPUT_PULLUP);
    pinMode(portaLa, INPUT_PULLUP);
    pinMode(portaSi, INPUT_PULLUP);
    pinMode(portaPiezo, OUTPUT);
}

void loop()
{
    // detecta botão pressionado e chama
    // a função correspondente
    if (digitalRead(portaDo) == LOW) {
        tone(portaPiezo, notaDo, duracao);
    }

    if (digitalRead(portaRe) == LOW) {
        tone(portaPiezo, notaRe, duracao);
    }

    if (digitalRead(portaMi) == LOW) {
        tone(portaPiezo, notaMi, duracao);
    }

    if (digitalRead(portaFa) == LOW) {
        tone(portaPiezo, notaFa, duracao);
    }

    if (digitalRead(portaSol) == LOW) {
        tone(portaPiezo, notaSol, duracao);
    }

    if (digitalRead(portaLa) == LOW) {
        tone(portaPiezo, notaLa, duracao);
    }

    if (digitalRead(portaSi) == LOW) {
        tone(portaPiezo, notaSi, duracao);
    }
}
```

PROFESSOR DANILO

**ANÁLISE DO CÓDIGO**

Declaramos abaixo quais as portas que devemos conectar os *pushbttons*. Note que de um lado de cada *pushbutton* temos o GND. Veja também que o código possui comentários que ajuda a entender o circuito.

```
//portas
int portaDo = 2, portaRe = 3, portaMi = 4,
    portaFa = 5, portaSol = 6,
    portaLa = 7, portaSi = 8;
```

Declaramos abaixo as variáveis que possuem o valor da frequência de cada nota que nosso piano terá.

```
//notas (frequências)
int notaDo = 264, notaRe = 297, notaMi = 330,
    notaFa = 352, notaSol = 396,
    notaLa = 440, notaSi = 495;
```

Abaixo, a porta que conectaremos o piezo.

```
//piezo
int portaPiezo = 9;
```

Cada nota terá duração de 200 milissegundos, e você pode mudar isso alterando o valor da variável *duracao*.

```
//duração da nota
int duracao = 200;
```

No setup (configuração), definimos cada porta de saída e de entrada. Note que temos que informar que as portas digitais usadas serão INPUT\_PULLUP, ou seja, cada porta fica em nível alto (1 ou HIGH) quando não conectamos nada a elas. Quando pressionarmos algum botão, levamos a porta conectada a ele (*pushbutton*) para o nível lógico baixo (0 ou LOW).

```
void setup()
{
    pinMode(portaDo, INPUT_PULLUP);
    pinMode(portaRe, INPUT_PULLUP);
    pinMode(portaMi, INPUT_PULLUP);
    pinMode(portaFa, INPUT_PULLUP);
    pinMode(portaSol, INPUT_PULLUP);
    pinMode(portaLa, INPUT_PULLUP);
    pinMode(portaSi, INPUT_PULLUP);
    pinMode(portaPiezo, OUTPUT);
}
```

Finalmente chegamos ao loop que irá tocar cada uma das notas associadas à cada botão.

```
void loop()
{
    // detecta botão pressionado e chama
    // a função correspondente
    if (digitalRead(portaDo) == LOW) {
        tone(portaPiezo, notaDo, duracao);
    }
}
```

Note que em cada *if*, testamos se a porta é LOW, isto é, se estamos conectando a porta ligada ao respectivo botão ao GND do Arduino. Lembre-se que ao fazer uma comparação, usamos dois iguais (==).

ROBÓTICA – 9º ANO 25/08/2022

```
if (digitalRead(portaRe) == LOW) {
    tone(portaPiezo, notaRe, duracao);
}
```

Vimos acima as notas Do e Re. Em cada uma delas usamos a função abaixo:

```
tone(portaPiezo, nota, duracao);
```

Esta função está recebendo três argumentos. Lembre-se da aula passada que usamos apenas duas e isso se deve por que o último argumento é facultativo e quando o omitimos dizemos que o Arduino deverá tocar por um tempo indefinido.

Quando omitimos o último argumento, devemos chamar a função

```
noTone();
```

para interromper o som.

Assim, a função *tone()* deve receber obrigatoriamente a porta e a nota, em Hz, respectivamente. A duração, terceiro argumento, deve ser informada em milissegundos.

O restante do código, copiado abaixo, não apresenta mais novidades.

```
if (digitalRead(portaMi) == LOW) {
    tone(portaPiezo, notaMi, duracao);
}
```

```
if (digitalRead(portaFa) == LOW) {
    tone(portaPiezo, notaFa, duracao);
}
```

```
if (digitalRead(portaSol) == LOW) {
    tone(portaPiezo, notaSol, duracao);
}
```

```
if (digitalRead(portaLa) == LOW) {
    tone(portaPiezo, notaLa, duracao);
}
```

```
if (digitalRead(portaSi) == LOW) {
    tone(portaPiezo, notaSi, duracao);
}
```

```
}
```

Veja o vídeo do circuito final obtido por seu professor no QR-code abaixo.



Figura 4: Vídeo do circuito montado pelo professor.