

Alarme residencial com Sensor de movimentos

Jean Tonin | jeanmftonin@gmail.com

Universidade Estadual de Londrina

Tutorial de Eletrônica

1 O sonho de Cladeusbaldo

Cladeusbaldo sempre gostou muito da área de tecnologia e adorava fazer projetos com seu Arduino. Um dia, ao voltar da escola, Cladeusbaldo encontrou sua casa aberta e com os móveis todos revirados, ele havia sido roubado!

Felizmente (e curiosamente) seu computador não havia sido levado, então Cladeusbaldo pensou num plano para evitar que essa situação ocorra novamente: um alarme! Procurou na internet o que poderia utilizar e encontrou um sistema simples que poderia implementar com os componentes que tinha em casa.

Agora é sua vez, incorpore o Cladeusbaldo e **construa um alarme** com os componentes que serão citados, você utilizará um **microcontrolador Arduino UNO**, para isso instale a IDE no site da Arduino, será nela que você fará a programação deste projeto.

2 Componentes

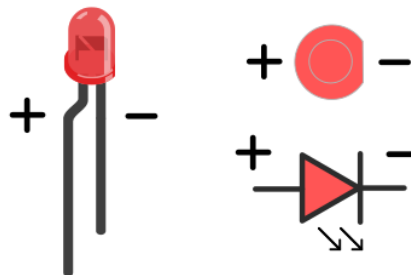
2.1 LED

Começaremos com o que é provavelmente o componente mais simples de se trabalhar no arduino: o LED (*Light Emitting Diode*). Ele é quase uma lâmpada, mas que precisa de bem menos energia para acender.

Precisamos tomar alguns cuidados ao lidar com o LED em um circuito:

- O LED possui polaridade, ou seja, se ele for colocado de modo invertido ele não funcionará, pelo contrário, será danificado. A "perna" positiva do LED deve estar sempre voltada para a tensão mais alta, e a negativa voltada para o Terra do circuito.

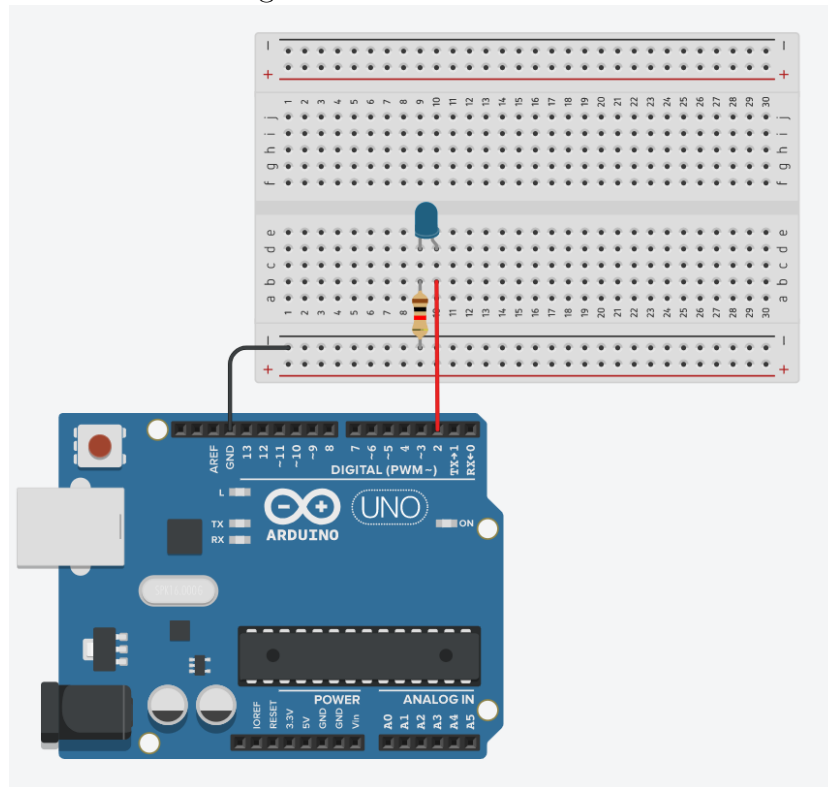
Figura 1: Polaridade do LED



- Além disso, o LED precisa de um **resistor** junto dele para limitar a corrente no circuito e, assim, evitar que o LED "queime". **Descubra qual o valor de resistência indicado para acompanhar um LED**

Veja um exemplo de montagem do LED com o arduino:

Figura 2: Circuito com LED



Agora vamos para a prática!

1. Monte o circuito anterior e implemente o seguinte código na IDE do arduino:

```
void setup() {  
  //Define a porta do led como saída  
  pinMode(2, OUTPUT);  
}  
  
void loop() {  
  //Acende o led  
  digitalWrite(2, HIGH);  
  //Aguarda intervalo de tempo em milissegundos  
  delay(1000);  
  //Apaga o led  
  digitalWrite(2, LOW);  
  //Aguarda intervalo de tempo em milissegundos  
  delay(1000);  
}
```

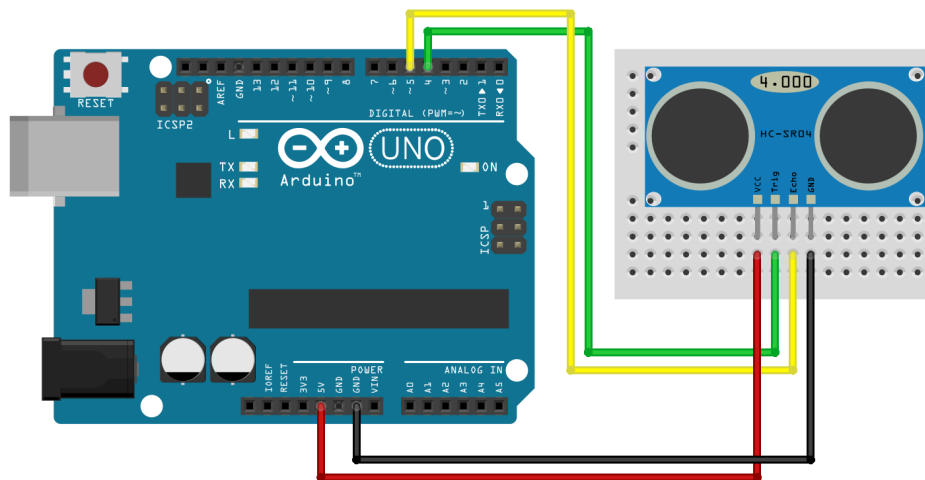
Agora, tente mudar o tempo que o LED fica aceso.

2.2 Sensor Ultrassônico

O sensor ultrassônico HC-SR04 é utilizado para medir distâncias entre 2 cm e 4 m, ele envia um sinal sonoro que colide com um obstáculo, e, de acordo com o tempo que leva para essa onda retornar, é calculada a distância.

A conexão com o Arduino UNO deve ser feita da seguinte maneira:

Figura 3: Esquema de conexão do sensor ultrassônico



Segue em anexo um programa de exemplo da utilização deste sensor. O código usa a biblioteca Ultrasonic, procure e faça o download. Descompacte a pasta Ultrasonic e coloque-a dentro da pasta LIBRARIES da IDE.

```
#include <Ultrasonic.h>

//Define os pinos para trigger e echo
#define pin_trigger 4
#define pin_echo 5

//Inicializa o sensor nos pinos definidos acima
Ultrasonic ultrasonic(pin_trigger, pin_echo);

void setup()
{
  Serial.begin(9600);
  Serial.println("Distância medida pelo sensor");
}

void loop()
{
  //Define a variável que vai receber a distância em cm
  float MedCM;
```

```
long microsec = ultrasonic.timing();  
MedCM = ultrasonic.convert(microsec, Ultrasonic::CM);  
//Exibe a distância no serial monitor  
Serial.print("Distancia em cm: ");  
Serial.print(MedCM);  
delay(1000);  
}
```

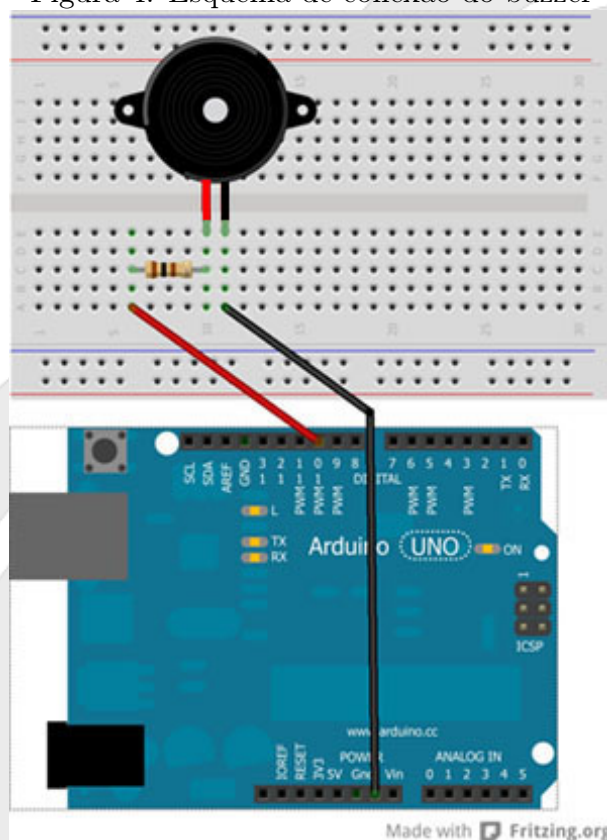
2 Implemente o programa acima. Depois tente converter para outras unidades de medida usando as funções da biblioteca Ultrasonic.

2.3 Buzzer

O *buzzer* é basicamente um pequeno alto-falante capaz de emitir sons em diversas frequências.

A conexão com o Arduino pode ser feita da seguinte maneira, utilizando um resistor de 100Ω . Observe que o *buzzer*, assim como o LED, tem polaridade, então cuidado para não conectá-lo erroneamente:

Figura 4: Esquema de conexão do buzzer



A seguir podemos ver um exemplo de programa para a utilização do *buzzer*:

```
//Pino em que o buzzer será ligado
const int pin_buzzer = 10;

void setup() {
  //Definindo pin_buzzer como saída
  pinMode(pin_buzzer,OUTPUT);
}
void loop() {
  //Ligando o buzzer com uma frequencia de 1500 hz.
  tone(pin_buzzer,1500);
  delay(500);

  //Desligando o buzzer.
  noTone(pin_buzzer);
  delay(500);
}
```

3 Implemente o programa acima. Se tirarmos o resistor, o que acontece?

3 Concluindo

Agora que você já sabe utilizar o LED, o sensor ultrassônico e o *buzzer*, você já está apto para fazer esse projeto. Só para recapitular, **faça um sensor que toque a "sirene" (*buzzer*) e pisque o LED quando detectar movimento**. Além disso, use uma **chave seletora** para dar a opção de ligar e desligar o aparelho, e também use um **botão** para poder desativar o alarme. Faça também uma **estrutura** para esse projeto, como uma caixa por exemplo, de preferência para que funcione sem nenhum fio a mostra.

Caso tenha qualquer dúvida não hesite em me perguntar!



Referências

- [1] Projeto 1 - pisca-pisca
<https://www.filipeflop.com/universidade/kit-maker-arduino/projeto-1-pisca-pisca/>
- [2] Como conectar o Sensor Ultrassônico HC-SR04 ao Arduino,
<https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>
- [3] Projeto Arduino, como emitir sons com o buzzer
<http://www.comofazerascoisas.com.br/projeto-arduino-como-emitir-sons-com-o-buzzer.html>