

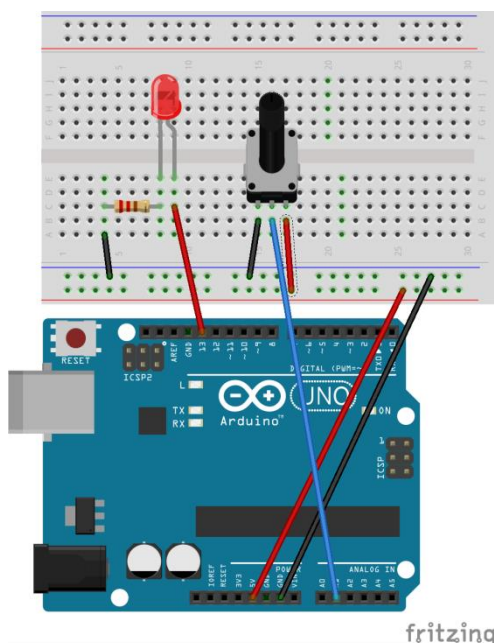
Projeto No. 2 – Potenciômetro

O objetivo deste projeto é controlar a frequência de acender e apagar (frequência de pisca-pisca) e a intensidade da luminosidade de um LED. Nesse workshop teremos dois experimentos para alcançar esses objetivos. Um potenciômetro é um resistor variável no formato de um botão giratório que fornece um valor analógico. Se girarmos o potenciômetro, alteramos a resistência em cada lado do contato elétrico que vai conectado ao terminal central do botão. Essa mudança implica em uma mudança no valor analógico de entrada. Quando o cursor for levado até o final da escala, teremos 0 volts e assim obtendo o valor 0 na entrada analógica. Quando giramos o cursor até o outro extremo da escala, teremos 5 volts e assim tendo o valor 1023 na entrada analógica. Outro conceito que podemos notar é a utilização dos pinos digitais com a marcação “~” (til) como, por exemplo, o pino digital “~9” usado no Programa 2.

Material necessário:

- 1 Arduino
- 1 Potenciômetro
- 1 Resistor de 220 ohms (vermelho, vermelho, marrom) ou 330 ohms (laranja, laranja, marrom) para o LED
- 1 LED de qualquer cor
- 1 Protoboard
- Jumper cable

Passo 1: Montagem do circuito



Conforme ilustra a figura ao lado:

- Conecte o pino 5v do Arduino à linha de alimentação positiva (vermelha) do protoboard;
- Conecte o pino GND do Arduino à linha de alimentação negativa (preta) do protoboard;
- Conecte um LED utilizando um resistor de 220 ohms (ou 330 ohms);
- Conecte o LED no pino digital 13;**
- Conecte o potenciômetro na protoboard conforme a figura ao lado (botão de girar virado para você);
- Conecte o pino da esquerda do potenciômetro na linha de alimentação GND;
- Conecte o pino da direita do potenciômetro na linha de alimentação positiva;
- Conecte o pino do centro do potenciômetro no pino analógico A1 do Arduino;



Passo 2: Programa 1 – Controlando a frequência do pisca-pisca

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

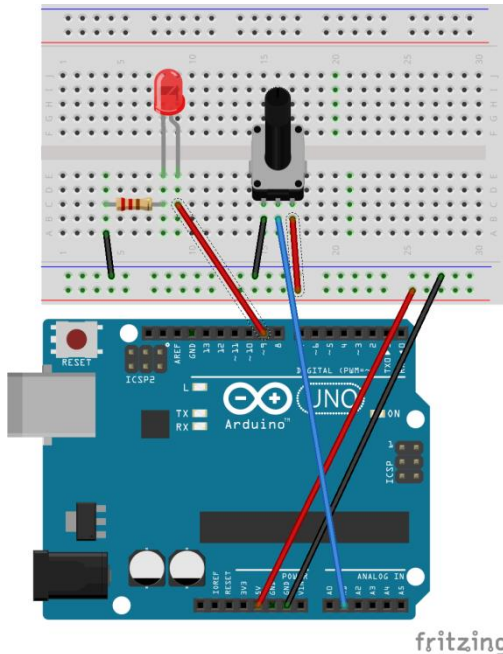
```
//Liga e desliga um LED na frequência determinada pelo potenciômetro.

int pot = A1;    // selecione o pino de entrada ao potenciômetro
int led = 13;    // selecione o pino ao LED
int valor = 0;   // variável p/ guardar o valor do potenciômetro

void setup() {
  pinMode(led, OUTPUT);
  pinMode(pot, INPUT);
}

void loop() {
  valor = analogRead(pot);
  digitalWrite(led, HIGH);
  delay(valor);
  digitalWrite(led, LOW);
  delay(valor);
}
```

Passo 3: Montagem do circuito



Conforme ilustra a figura ao lado:

- Conecte o pino 5v do Arduino à linha de alimentação positiva (vermelha) do protoboard;
- Conecte o pino GND do Arduino à linha de alimentação negativa (preta) do protoboard;
- Conecte um LED utilizando um resistor de 220 ohms (ou 330 ohms);
- Conecte o LED no pino digital 9;**
- Conecte o potenciômetro na protoboard conforme a figura ao lado (botão de girar virado para você);
- Conecte o pino da esquerda do potenciômetro na linha de alimentação GND;
- Conecte o pino da direita do potenciômetro na linha de alimentação positiva;
- Conecte o pino do centro do potenciômetro no pino analógico A1 do Arduino;

Passo 4: Programa 2 – Controle da intensidade da luminosidade

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

```
//Controla a intensidade da luminosidade de um LED pela frequência determinada pelo
potenciômetro.

int pot = 1;    // selecione o pino de entrada ao potenciômetro
int led = 9;    // selecione o pino ao LED
int valor = 0;  // variável p/ guardar o valor do potenciômetro

void setup() {
  Serial.begin(9600);
  pinMode(led, OUTPUT);
}

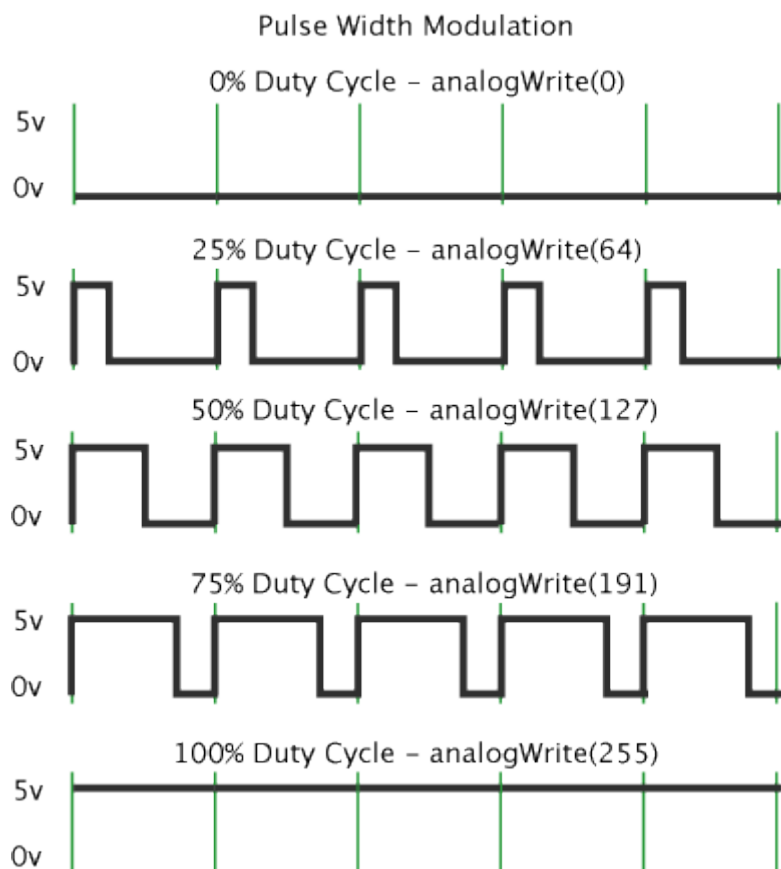
void loop() {
  valor = analogRead(pot);
  if(valor > 0){
    analogWrite(led, (valor/4)); //acende o led com intensidade proporcional ao
valor obtido
    Serial.println(valor); //mostra no Serial Monitor o valor obtido do
potenciômetro
  }
}
```

Dicas:

Como funciona o PWM?

A Modulação por Largura de Pulso (*Pulse Width Modulation* - **PWM**) é uma técnica que consiste em fornecer um sinal analógico através de meios digitais. A forma de onda do sinal digital consiste em uma onda quadrada que alterna seu estado em nível lógico alto e um nível lógico baixo (pode ser representado por ligado/desligado ou pelo sistema binário 1 e 0).

A razão entre o período de pico e o período total da onda é chamada de *Duty Cycle*. Podemos, então, entender que para termos uma onda quadrada real (que possui picos e vales iguais) é necessário que o *Duty Cycle* seja de 50%, ou seja, 50% de pico e 50% de vale.



No Arduino UNO, as portas digitais que permitem PWM são as portas 3, 5, 6, 9, 10 e 11. Essas portas são facilmente identificadas pelo símbolo "~" abaixo de cada porta.

Fonte: Tradução e imagem de "PWM" em <http://arduino.cc/en/Tutorial/PWM>