



FACULDADE DE TECNOLOGIA DE PORTO ALEGRE– FAQI



Automação com Arduino na prática

Professor



- **Silvio Cesar Viegas**
- **Doutorando no Ensino de Ciências e Matemática**
- **Mestre no Ensino de Ciências e Matemática**
- **Bacharel em Informática**
- **Email: scviegas@gmail.com / silvio.viegas@qi.edu.br**
- **Curriculum Lattes: <http://lattes.cnpq.br/5020505141968701>**

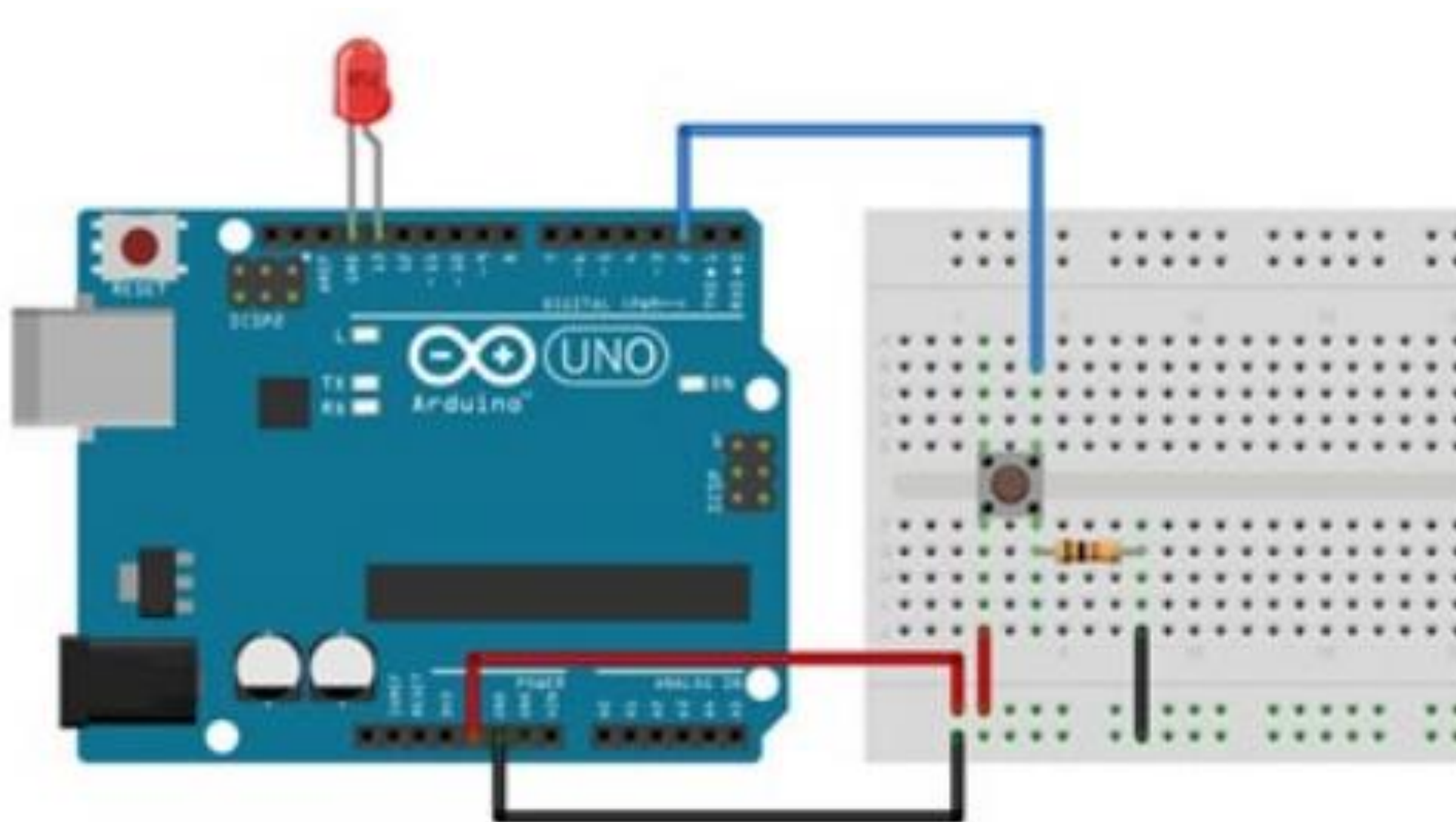


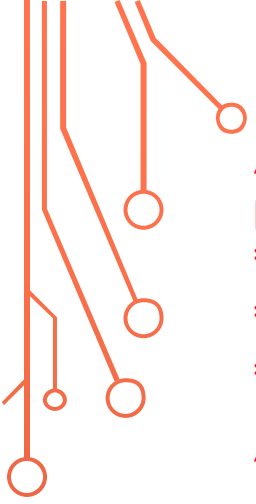


ARDUINO

Massimo Banzi says "You don't need anybody's permission to create something great." So what you waiting for?

Liga Led pressionando botão





/* Botao Liga e desliga um LED conectado ao pino digital 13 quando pressionado um botao conectado ao pino 2. O Circuito:

* LED conectado ao pino 13 e ao terra

* botao conectado ao pino 2 desde 5V

* resistor de 10K conectado ao pino 2 desde o terra */

// constantes nao sao alteradas.

// Sao usadas aqui para definir os numeros dos pinos:

const int buttonPin = 2;

// o numero do pino do botão

const int ledPin = 13;

// o numero do pino do LED

// variaveis que devem mudar:

int buttonState = 0;

// variavel para ler o estado do botao

void setup() {

// inicializa o pino do LED como saida:

pinMode(ledPin, OUTPUT);

// inicializa o pino do botao como entrada:

pinMode(buttonPin, INPUT); }

void loop(){

// faz a leitura do valor do botao:

buttonState = digitalRead(buttonPin);

// verifica se o botao esta pressionado.

// em caso positivo, buttonState e HIGH:

if (buttonState == HIGH) {

// liga o LED:

digitalWrite(ledPin, HIGH);

}

else { // desliga o LED:

digitalWrite(ledPin, LOW);

}

}



```
/* DigitalReadSerial Le a entrada digital no pino 2 e imprime o  
resultado no monitor serial. Este exemplo e de dominio publico. */
```

```
int pushButton = 2; // o pino 2 tem um botao ligado nele.
```

```
int ledPin = 13; // entrada do LED no pino 13.
```

```
void setup() {
```

```
// Inicializa a comunicacao serial a 9600 bits por segundo:
```

```
Serial.begin(9600);
```

```
pinMode(pushButton, INPUT);
```

```
// define o botao como uma entrada. pinMode(ledPin, OUTPUT);
```

```
//define o LED como uma saída.
```

```
}
```

```
void loop() {
```

```
// faz a leitura do pino de entrada:
```

```
int buttonState = digitalRead(pushButton);
```

```
if (buttonState == 1) {
```

```
digitalWrite(ledPin, HIGH); }
```

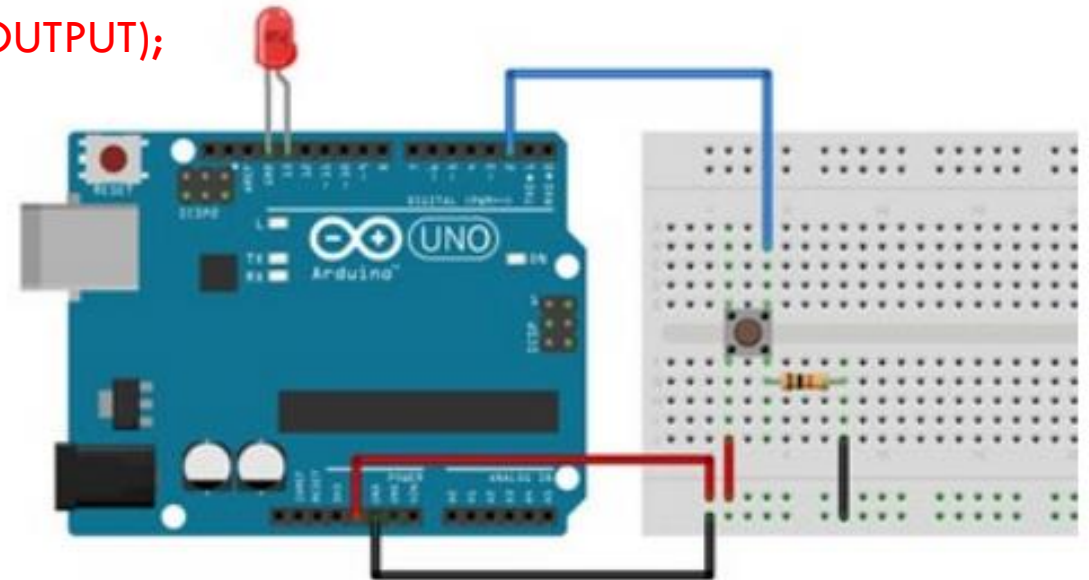
```
else { digitalWrite(ledPin, LOW); } // imprime o estado do botao:
```

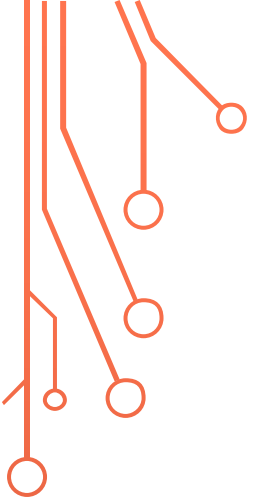
```
Serial.println(buttonState);
```

```
delay(1); // delay entre leituras (em milissegundos)
```

```
}
```

Exemplo serial





Contador de pulsos (edge detection) criado em 27/09/2005, modificado em 30/08/2011 por Tom Igoe Este exemplo e de dominio publico.

<http://arduino.cc/en/Tutorial/ButtonStateChange> */

// constantes nao sao alteradas:

const int buttonPin = 2;

// o numero do pino do botao

const int ledPin = 13;

// o numero do pino do LED

// variaveis que devem mudar:

int buttonPushCounter = 0;

// contador para o numero de impressoes do botao

int buttonState = 0;

// atual estado do botao

int lastButtonState = 0; // anterior estado do botao

void setup() {

pinMode(buttonPin, INPUT); // inicializa o pino do botao como entrada

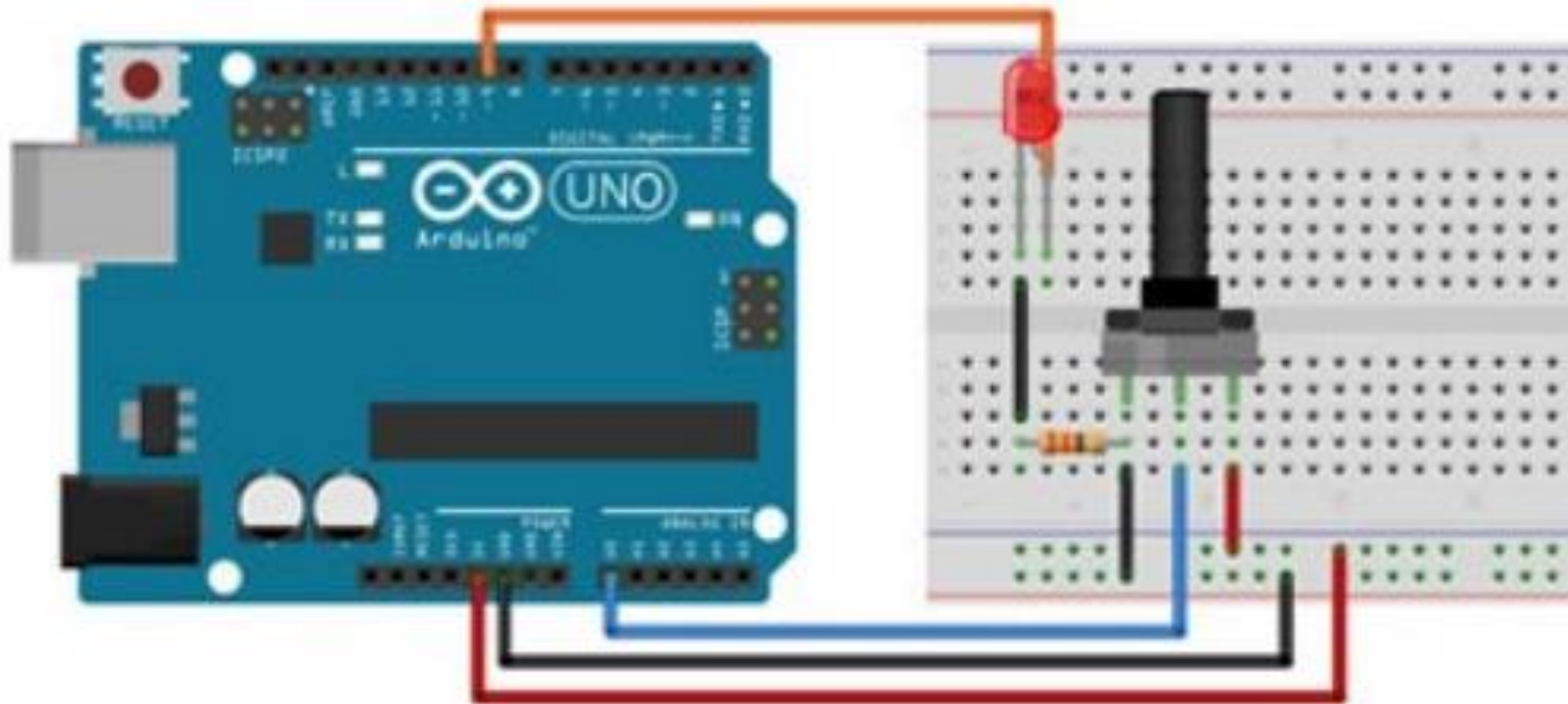
pinMode(ledPin, OUTPUT); // inicializa o pino digital como saida
Serial.begin(9600); // inicializa a comunicacao serial

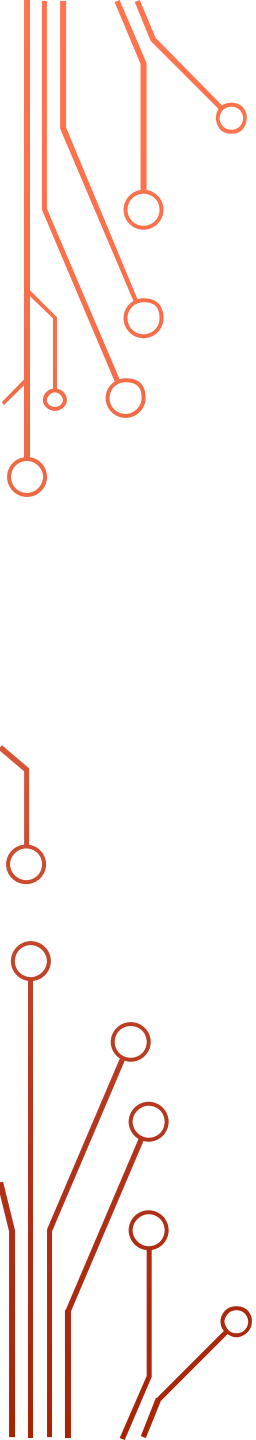
}



```
void loop() {  
  // faz a leitura do valor do botao:  
  buttonState = digitalRead(buttonPin);  
  // compara o estado atual do botao com seu estado anterior  
  if (buttonState != lastButtonState) {  
    // se o estado do botao foi alterado, incrementar o contador  
    if (buttonState == HIGH) {  
      buttonPushCounter++;  
      Serial.print("numero de pulsos: ");  
      Serial.println(buttonPushCounter);  
    }  
    // salva o estado atual do botao como ultimo estado para iniciar o  
    // proximo loop  
    lastButtonState = buttonState; // Liga o LED cada 4 pulsacoes checando o modulo de contador de  
    // botao  
    if (buttonPushCounter % 4 == 0) {  
      digitalWrite(ledPin, HIGH); }  
    else { digitalWrite(ledPin, LOW);  
    }  
  }  
}
```

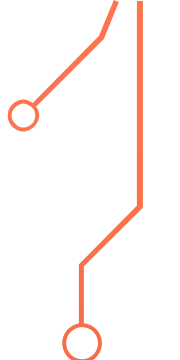
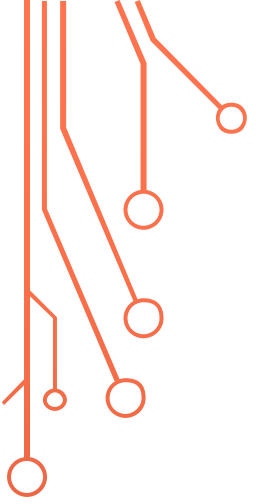

Entrada e saída analógica – saída serial



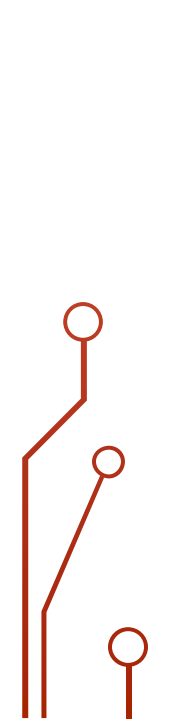
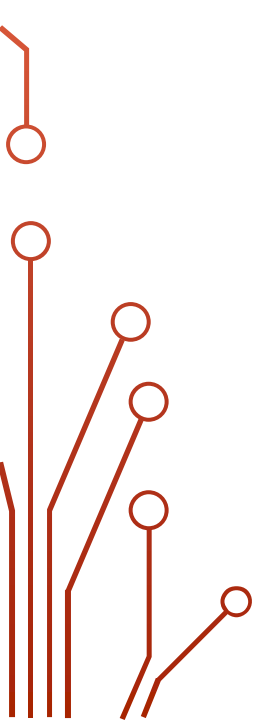
The slide features decorative circuit-like lines in blue and red. On the left, a blue line starts from the top, goes down, and branches into several paths ending in circles. On the right, a red line starts from the top, goes down, and branches into several paths ending in circles. At the bottom, there are more blue and red lines, some ending in circles, creating a symmetrical, circuit-like border around the central text.

Entrada Analogica, Saida Analogica, Saida serial Le o pino de entrada analogica, mapeia o resultado para um intervalo entre 0 e 255 e usa o resultado para estabelecer o pulso PWM do pino de saida. Tambem e possivel acompanhar o resultado atraves do Monitor Serial. O circuito: - O pino central do Potenciometro conectado ao pino analogico 0. Os pinos laterais do potenciometro conectados no terra e 5V. - LED conectado no pino digital 9 e no terra. Criado em 29/12/2008, Modificado em 09/04/2012 por Tom Igoe Este exemplo e de dominio publico. */

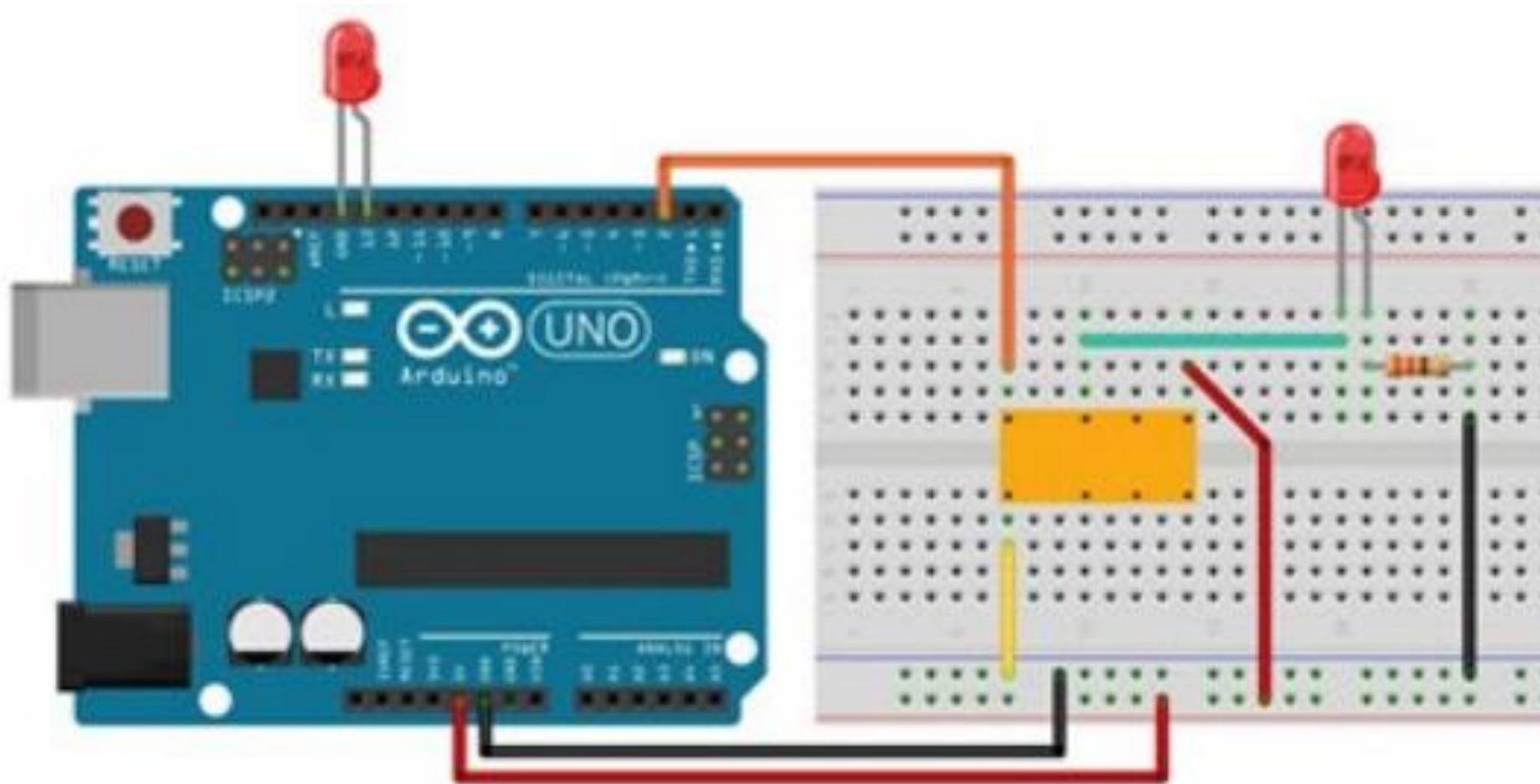
```
// constantes nao sao alteradas:
const int analogInPin = A0;
// Entrada analogica do potenciometro
const int analogOutPin = 9;
// Saida analogica onde o LED esta conectado
int sensorValue = 0;
// leitura do potenciometro
int outputValue = 0;
// leitura da saida PWM (analogica)
void setup() {
// inicializa a comunicacao serial:
Serial.begin(9600);
}
```



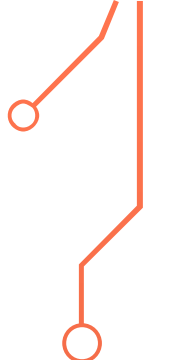
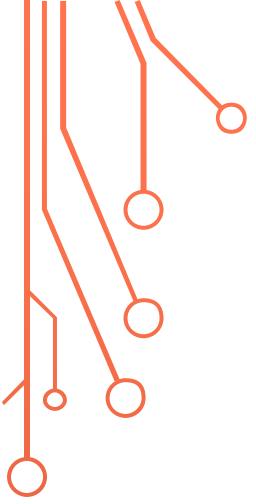
```
void loop() {  
  // faz a leitura da entrada analogica:  
  sensorValue = analogRead(analogInPin);  
  // mapeia o resultado da entrada analogica dentro do intervalo de 0 a 255:  
  outputValue = map(sensorValue, 0, 1023, 0, 255);  
  // muda o valor da saida analogica: analogWrite(analogOutPin, outputValue);  
  // imprime o resultado no monitor serial:  
  Serial.print("sensor = " );  
  Serial.print(sensorValue);  
  Serial.print("\t output = ");  
  Serial.println(outputValue);  
  // Aguarda 2 milissegundos antes do proximo loop:  
  delay(2);  
}
```



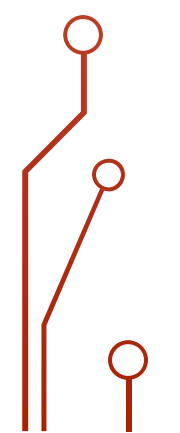
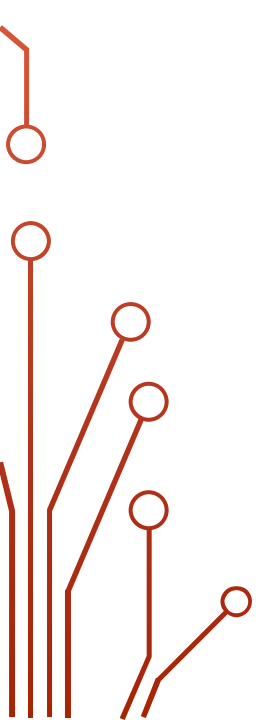
Acionando relé



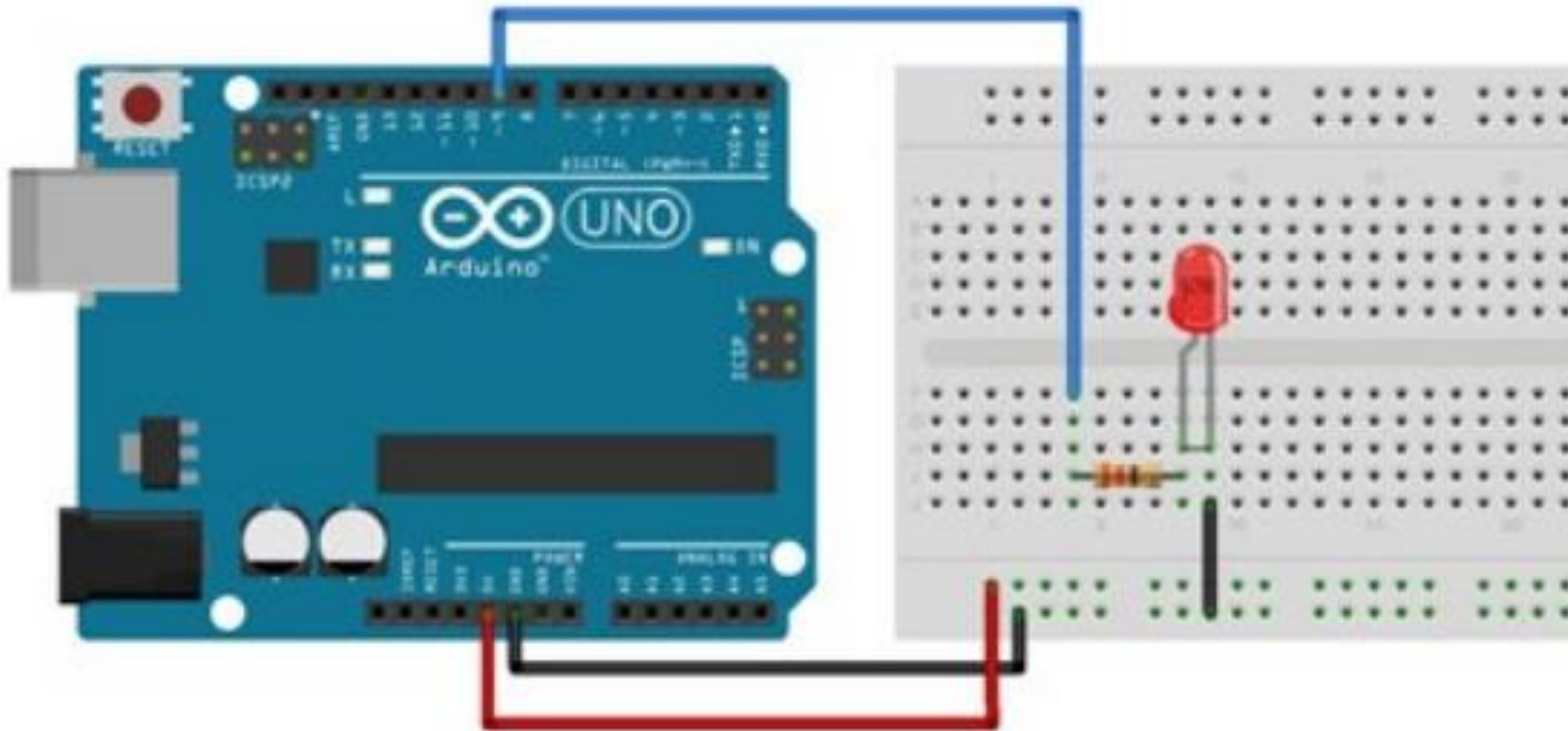
```
//*****  
//*Codigo para teste de Arduino acionando rele do kit Multilogica,  
//* ligado na saida digital 2 e GND, monitorado pelo Led 13  
//* este codigo tem dominio publico  
//*****  
//inicializa uma variavel do tipo char que utiliza 1 byte para armazenar  
//1 caracter  
char input= 0;  
int rele=2;  
int led=13;  
boolean y=true; // inicializa uma variavel do tipo booleano  
void setup() {  
  pinMode(rele,OUTPUT);  
  pinMode(led,OUTPUT);  
  Serial.begin(9600);  
  Serial.println();  
  Serial.print("**Codigo para acionar rele conectado ao pino 2 do Arduino ");  
  Serial.println("atraves do monitor serial**");  
  Serial.println("");  
  Serial.println("Pressione 1 e depois ENTER para inverter o estado do rele novamente");  
  Serial.println("Aguardando comando :");  
}
```

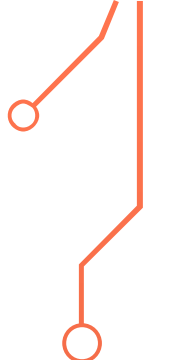
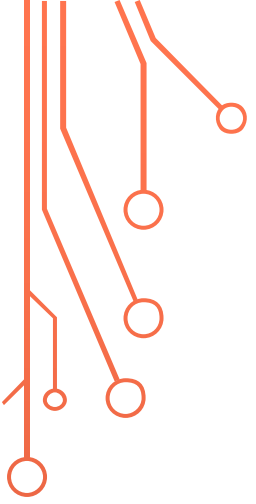


```
void loop() {  
  if (Serial.available() > 0) {  
    input= Serial.read();  
    if (input == '1'){  
      Serial.print("O rele agora esta ");  
      if(y){  
        digitalWrite(rele, HIGH);  
        digitalWrite(led, HIGH);  
        Serial.println("ligado");  
      }  
      else { digitalWrite(rele, LOW);  
        digitalWrite(led, LOW);  
        Serial.println("desligado");  
      } y=!y; // altera o valor de y, se le y e igual a nao y  
    }  
    else {  
      Serial.println("Comando invalido");  
    }  
  }  
}
```

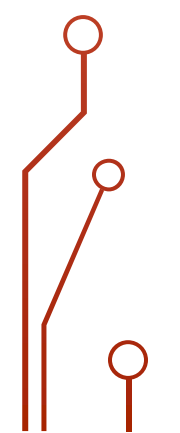
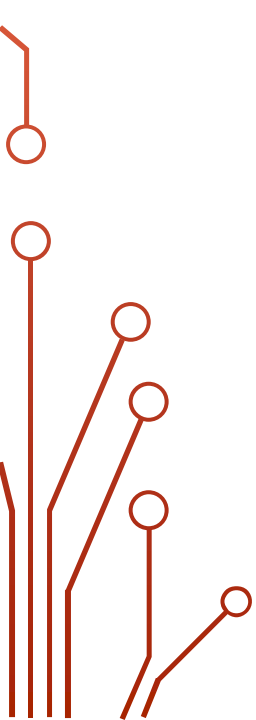


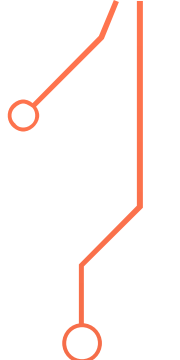
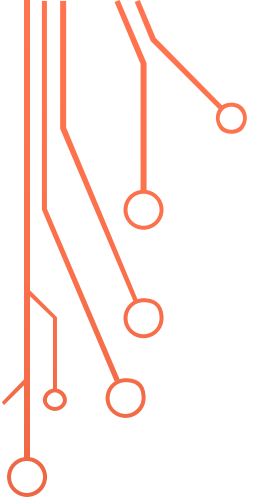
Este exemplo demonstra o uso da função `analogWrite()` para apagar um LED em fade (variação gradual). `AnalogWrite` usa um pulso PWM, alternando o pino digital on e off rapidamente, criando o efeito de fade.



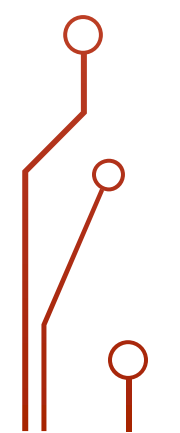
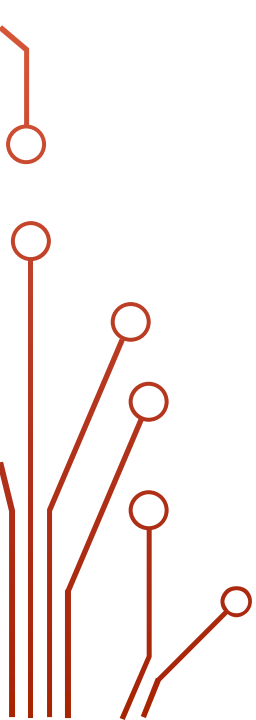


```
/* Fade Este exemplo mostra como executar um fade em um LED no pino
9 usando a funcao analogWrite(). Este exemplo e de dominio publico */
int led = 9;
// pino do LED
int brightness = 0;
// intensidade do brilho do LED
int fadeAmount = 5;
// em quantos pontos aplicar o fade no LED
void setup() {
// define o pino 9 como saida:
pinMode(led, OUTPUT);
}
```

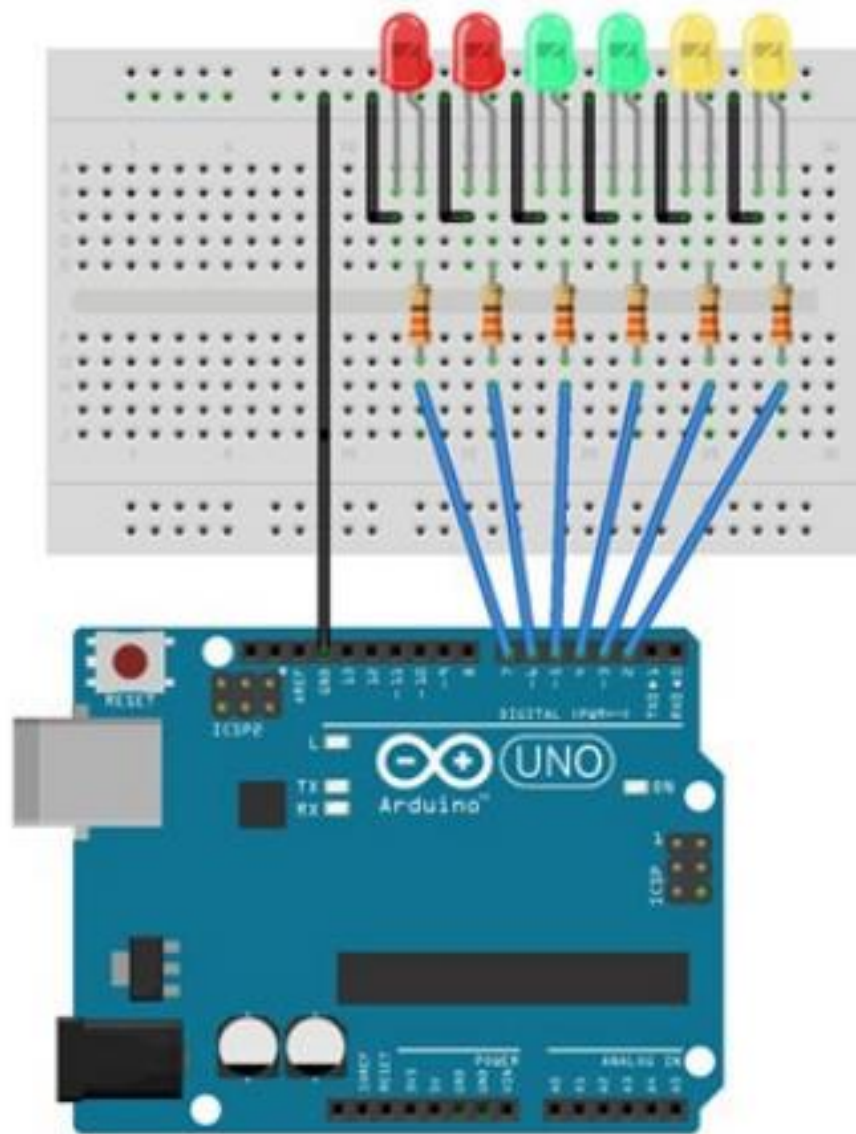


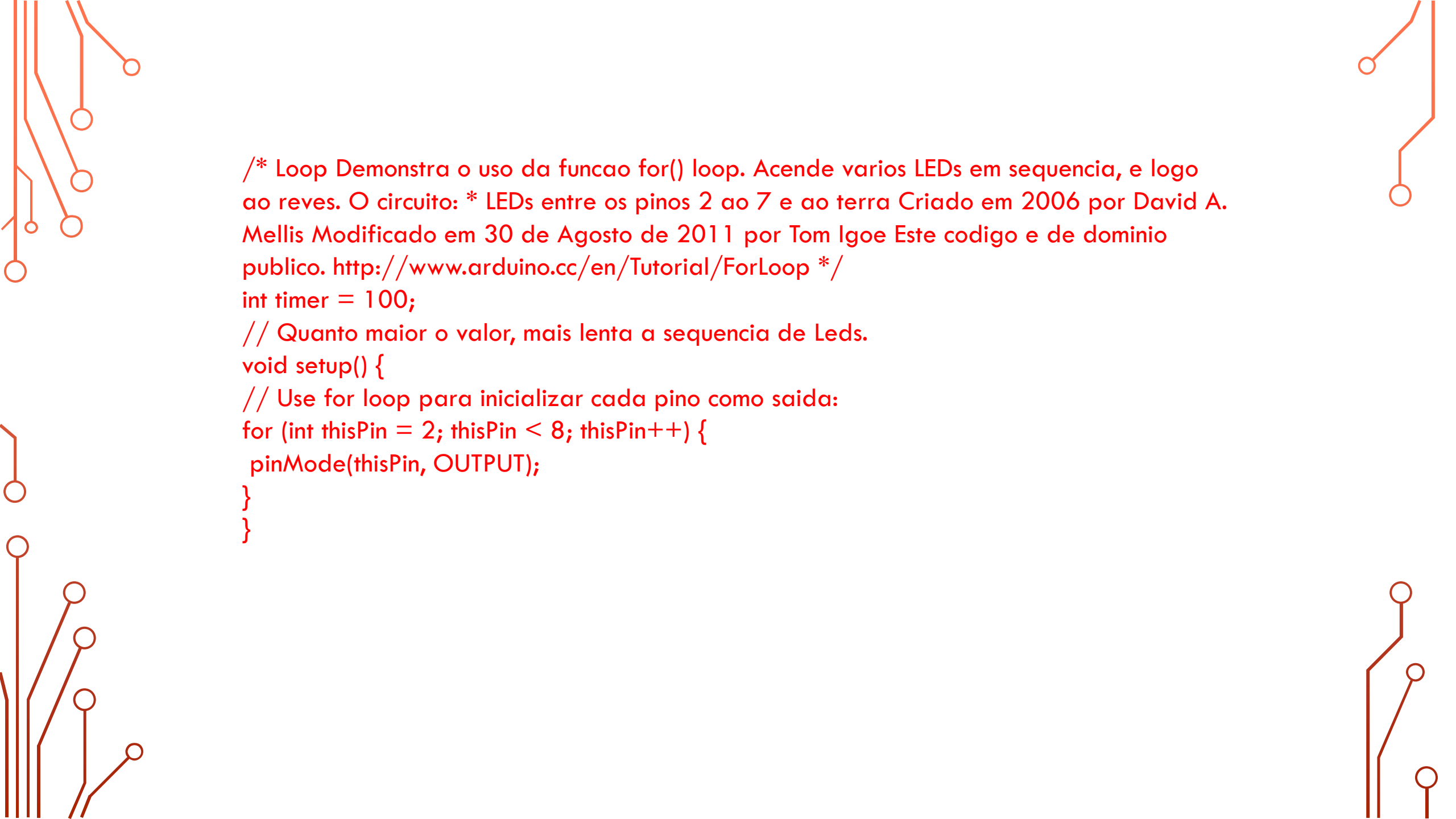


```
// o loop roda em sequencia continuamente:  
void loop() {  
  // define o brilho do pino 9:  
  analogWrite(led, brightness);  
  // muda o brilho para o proximo loop:  
  brightness = brightness + fadeAmount;  
  // inverte a direcao do fade ao final do mesmo:  
  if (brightness == 0 || brightness == 255) {  
    fadeAmount = -fadeAmount ;  
  }  
  // aguarda 30 milissegundos para ver o efeito dimer:  
  delay(30);  
}
```

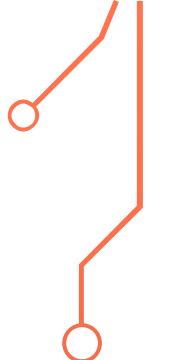
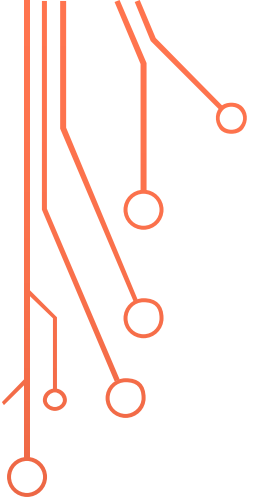


Acionar leds em sequencia

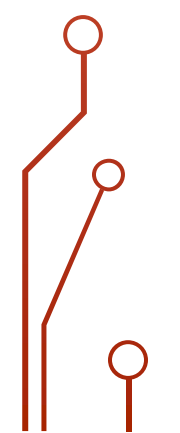
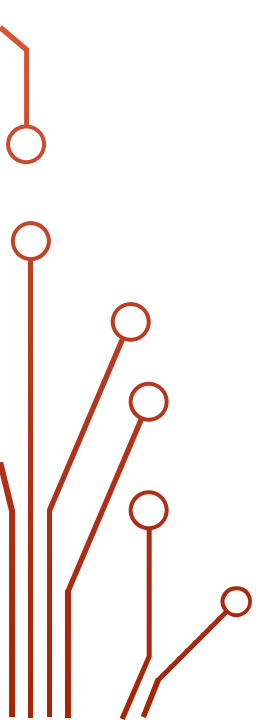


The image features decorative circuit board patterns in the corners, rendered in a light blue color. These patterns consist of lines and circles, resembling a stylized PCB layout. The patterns are located in the top-left, top-right, bottom-left, and bottom-right corners of the image.

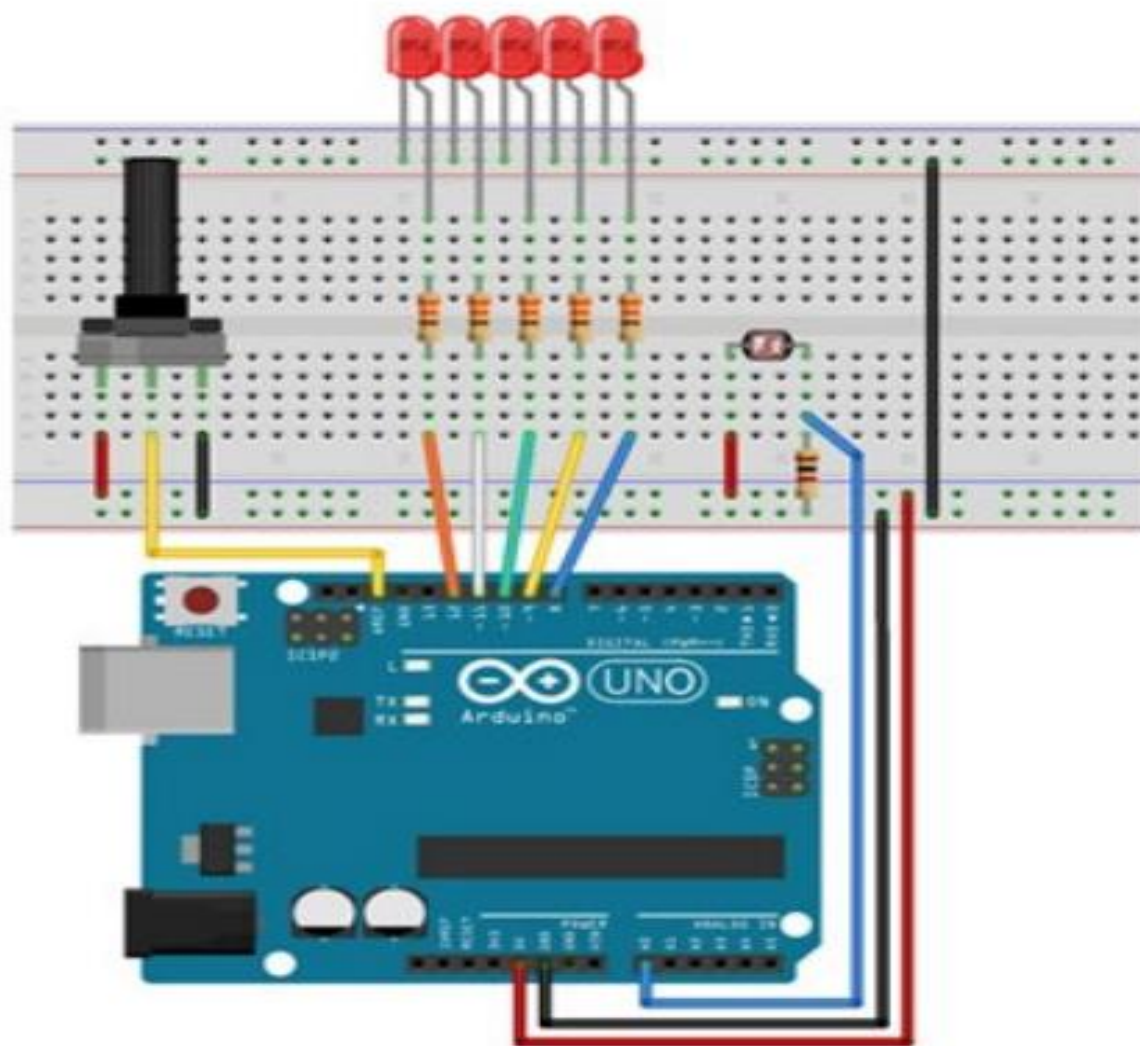
```
/* Loop Demonstra o uso da funcao for() loop. Acende varios LEDs em sequencia, e logo  
ao reves. O circuito: * LEDs entre os pinos 2 ao 7 e ao terra Criado em 2006 por David A.  
Mellis Modificado em 30 de Agosto de 2011 por Tom Igoe Este codigo e de dominio  
publico. http://www.arduino.cc/en/Tutorial/ForLoop */  
int timer = 100;  
// Quanto maior o valor, mais lenta a sequencia de Leds.  
void setup() {  
  // Use for loop para inicializar cada pino como saida:  
  for (int thisPin = 2; thisPin < 8; thisPin++) {  
    pinMode(thisPin, OUTPUT);  
  }  
}
```

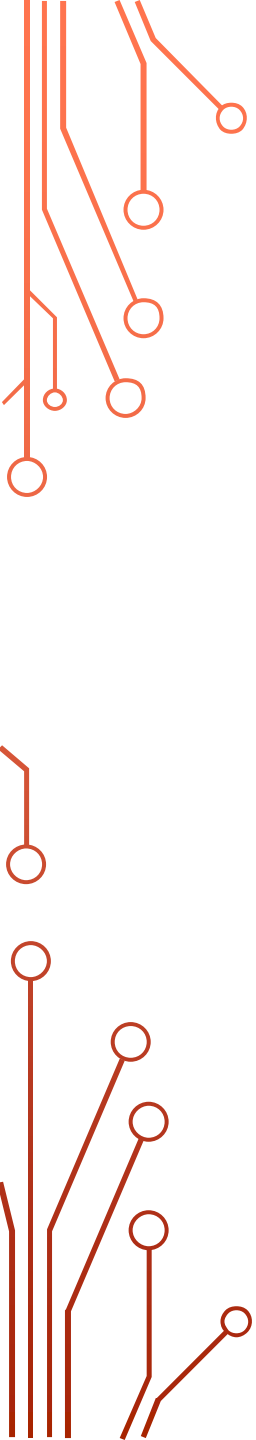


```
void loop() {  
  // loop desde o pino mais baixo ate o mais alto:  
  for (int thisPin = 2; thisPin < 8; thisPin++) {  
    // liga este pino:  
    digitalWrite(thisPin, HIGH);  
    delay(timer);  
    // desliga este pino:  
    digitalWrite(thisPin, LOW); }  
  // loop desde o pino mais alto ate o mais baixo:  
  for (int thisPin = 7; thisPin >= 2; thisPin--) {  
    // liga este pino:  
    digitalWrite(thisPin, HIGH);  
    delay(timer);  
    // desliga este pino:  
    digitalWrite(thisPin, LOW);  
  }  
}
```



Usando sensor LDR





```
/* Sensor LDR Conectar um LDR a uma entrada analogica para controlar cinco saidas em funcao da luz ambiente. Este codigo e de dominio publico. Criado em 27/11/2011 por Arduteka. Modificado em 13/01/2014 por Multilogica. */
```

```
//Armazenar os dados recolhidos pelo sensor LDR: int valorLDR = 0;
```

```
//Definir os pinos de entrada dos LEDs:
```

```
int pinLed1 = 12;
```

```
int pinLed2 = 11;
```

```
int pinLed3 = 10;
```

```
int pinLed4 = 9;
```

```
int pinLed5 = 8;
```

```
//Definir pino de entrada do sensor LDR
```

```
int pinLDR = 0;
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  //Definir os pinos de saida dos LEDs:
```

```
  pinMode(pinLed1, OUTPUT);
```

```
  pinMode(pinLed2, OUTPUT);
```

```
  pinMode(pinLed3, OUTPUT);
```

```
  pinMode(pinLed4, OUTPUT);
```

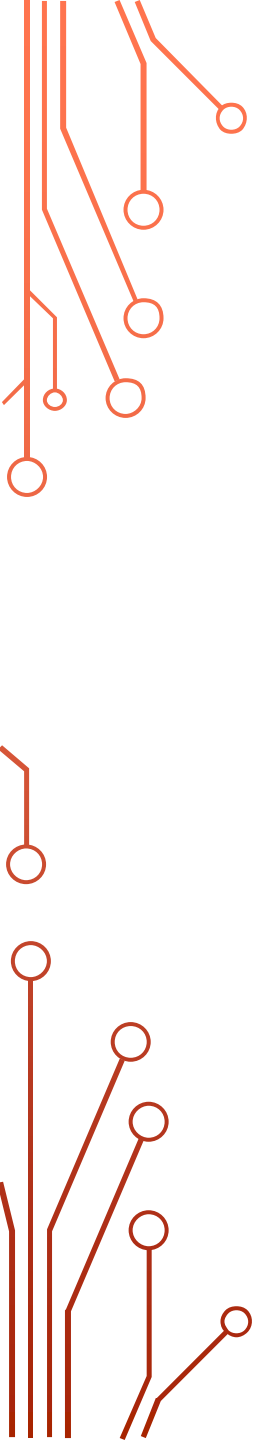
```
  pinMode(pinLed5, OUTPUT);
```

```
  //Definimos o uso de uma referencia externa:
```

```
  pinMode(EXTERNAL);
```

```
}
```



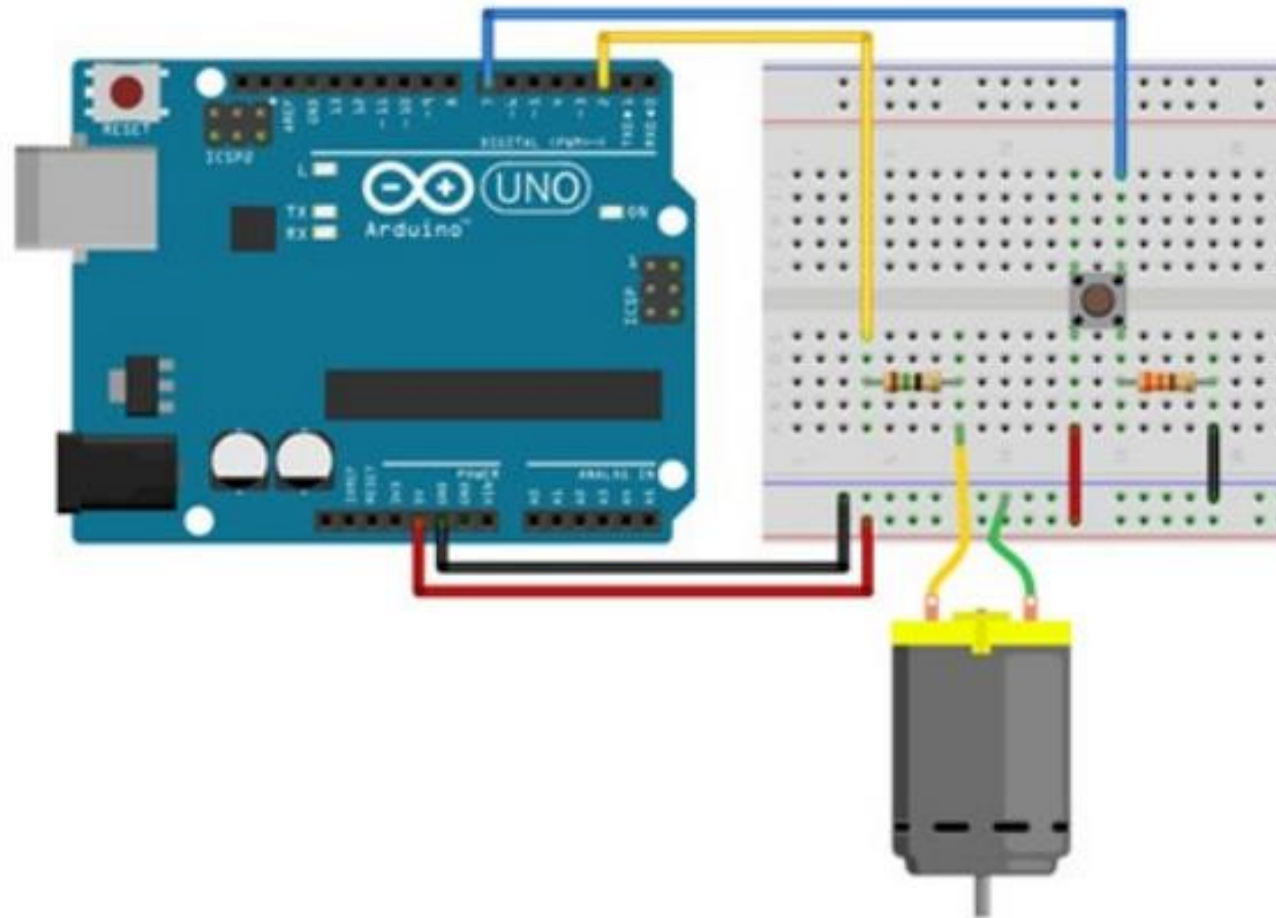


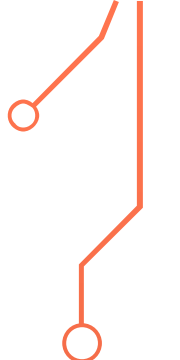
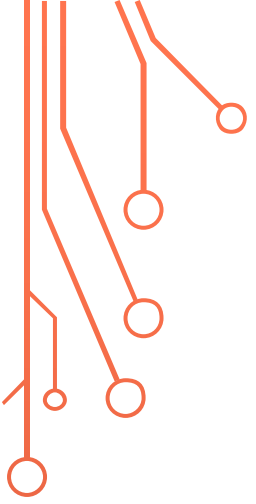
```
void loop() {  
  //Guardar o valor da leitura de uma variavel:  
  valorLDR = analogRead(pinLDR);  
  Serial.println(valorLDR);  
  //Definicao do padrao de controle dos LEDs:  
  if(valorLDR >= 1023) {  
    digitalWrite(pinLed1, LOW);  
    digitalWrite(pinLed2, LOW);  
    digitalWrite(pinLed3, LOW);  
    digitalWrite(pinLed4, LOW);  
    digitalWrite(pinLed5, LOW);  
  }  
  else if((valorLDR >= 823) & (valorLDR < 1023)) {  
    digitalWrite(pinLed1, HIGH);  
    digitalWrite(pinLed2, LOW);  
    digitalWrite(pinLed3, LOW);  
    digitalWrite(pinLed4, LOW);  
    digitalWrite(pinLed5, LOW);  
  }  
  else if((valorLDR >= 623) & (valorLDR < 823)) {  
    digitalWrite(pinLed1, HIGH);  
    digitalWrite(pinLed2, HIGH);  
    digitalWrite(pinLed3, LOW);  
    digitalWrite(pinLed4, LOW);  
    digitalWrite(pinLed5, LOW);  
  }  
}
```

```
    digitalWrite(pinLed5, LOW);  
  }  
  else if((valorLDR >= 423) & (valorLDR < 623)) {  
    digitalWrite(pinLed1, HIGH);  
    digitalWrite(pinLed2, HIGH);  
    digitalWrite(pinLed3, HIGH);  
    digitalWrite(pinLed4, LOW);  
    digitalWrite(pinLed5, LOW);  
  }  
  else if((valorLDR >= 223) & (valorLDR < 423)) {  
    digitalWrite(pinLed1, HIGH);  
    digitalWrite(pinLed2, HIGH);  
    digitalWrite(pinLed3, HIGH);  
    digitalWrite(pinLed4, HIGH);  
    digitalWrite(pinLed5, LOW);  
  }  
  else {  
    digitalWrite(pinLed1, HIGH);  
    digitalWrite(pinLed2, HIGH);  
    digitalWrite(pinLed3, HIGH);  
    digitalWrite(pinLed4, HIGH);  
    digitalWrite(pinLed5, HIGH);  
  }  
}
```

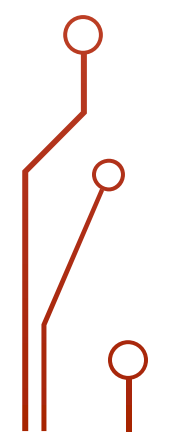
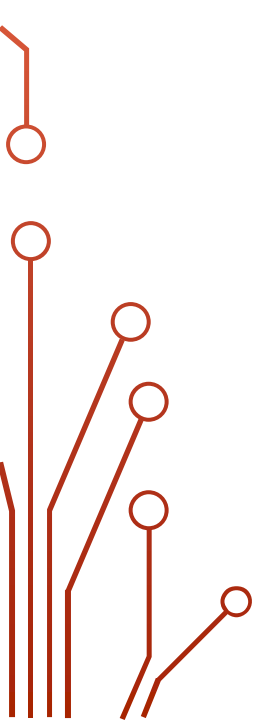


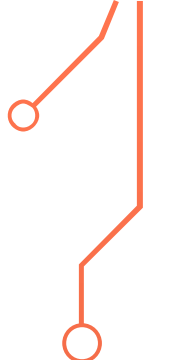
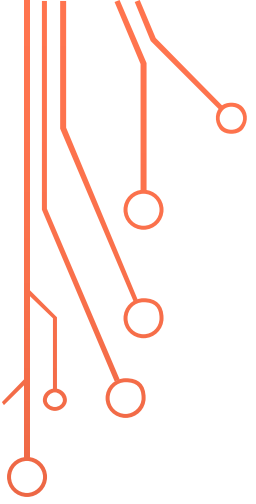
Aciona Motor



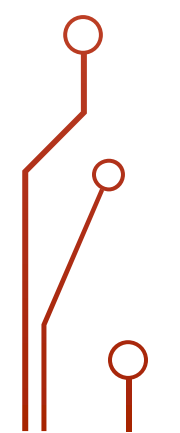
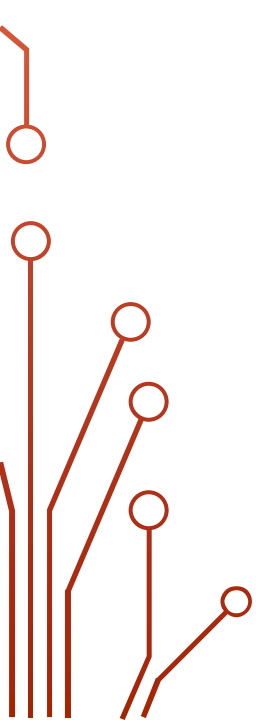


```
// Ligar motor no pino 2 em serie com um resistor de 15 ohms
// para limitar a corrente em 40mA para nao sobrecarregar o Arduino
//Este codigo e de dominio publico.
//Criado em 2014 por Multilogica.
const int motorPin = 2;
const int buttonPin = 7;
int buttonState = 0;
void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(motorPin, OUTPUT);
}
```

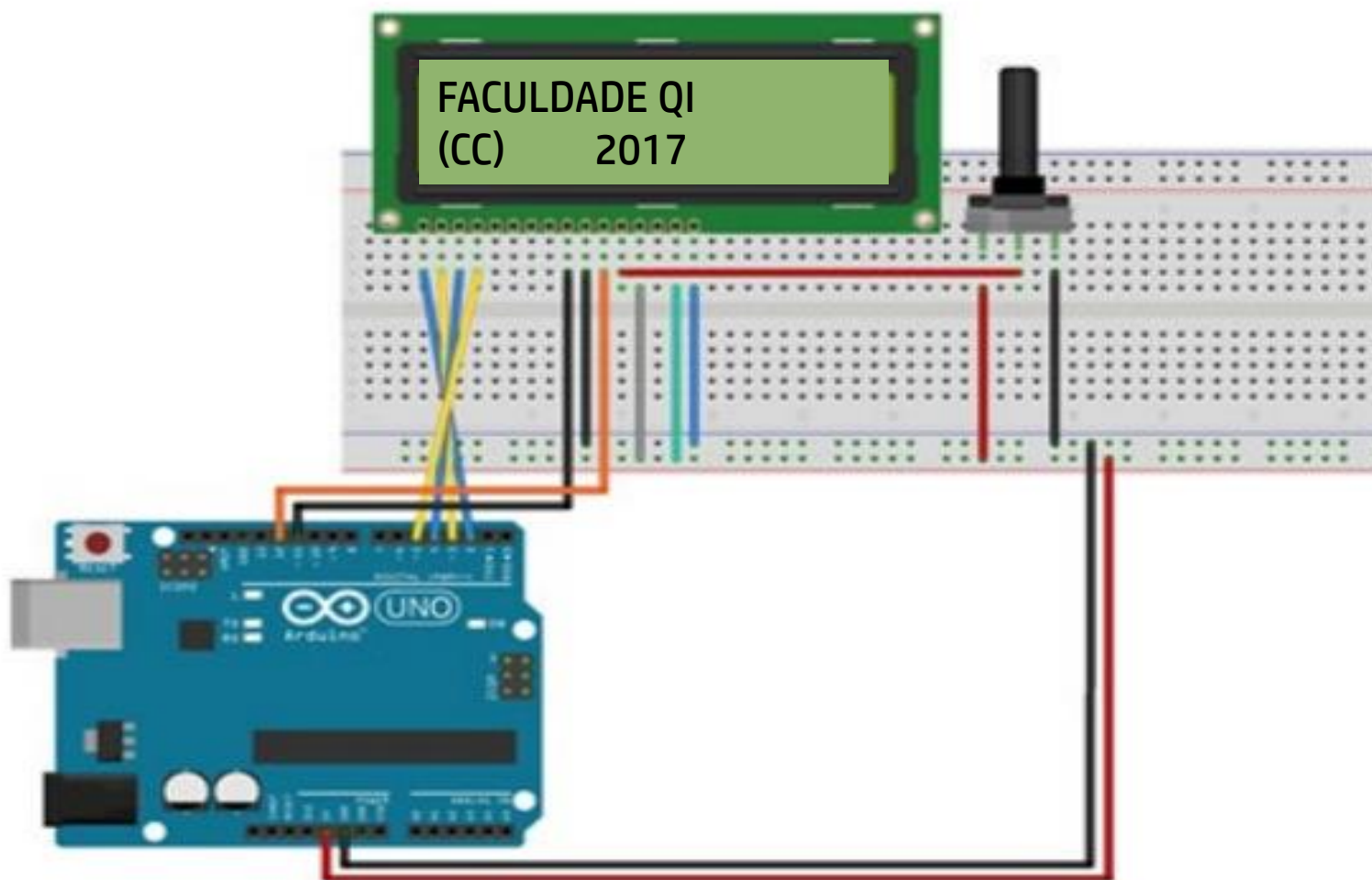




```
void loop() {  
  buttonState = digitalRead(buttonPin);  
  if (buttonState == HIGH) {  
    digitalWrite(motorPin, HIGH);  
  }  
  else {  
    digitalWrite(motorPin, LOW);  
  }  
}
```



Escrevendo em um LCD



/* Biblioteca LiquidCrystal Demonstra o uso do display de 16x2 caracteres Esta biblioteca funciona com todos displays compatíveis com o driver Hitachi HD44780. Este código escreve :
FACULDADE QI (cc) 2017 Circuito :

- * LCD pino RS no pino digital 12
- * LCD pino Enable no pino digital 11
- * LCD pino D4 pin no pino digital 5
- * LCD pino D5 pin no pino digital 4
- * LCD pino D6 pin no pino digital 3
- * LCD pino D7 pin no pino digital 2
- * LCD pino R/W no terra
- * Trimpot de 10K :
- * +5V no +5V
- * Terra no terra

* wiper to LCD VO pin (pin 3) Código de domínio público baseado no tutorial original :

<http://www.arduino.cc/en/Tutorial/LiquidCrystal> */

// Inclui o código da biblioteca:

#include < LiquidCrystal.h>

// Inicializa a biblioteca e define os pinos utilizados

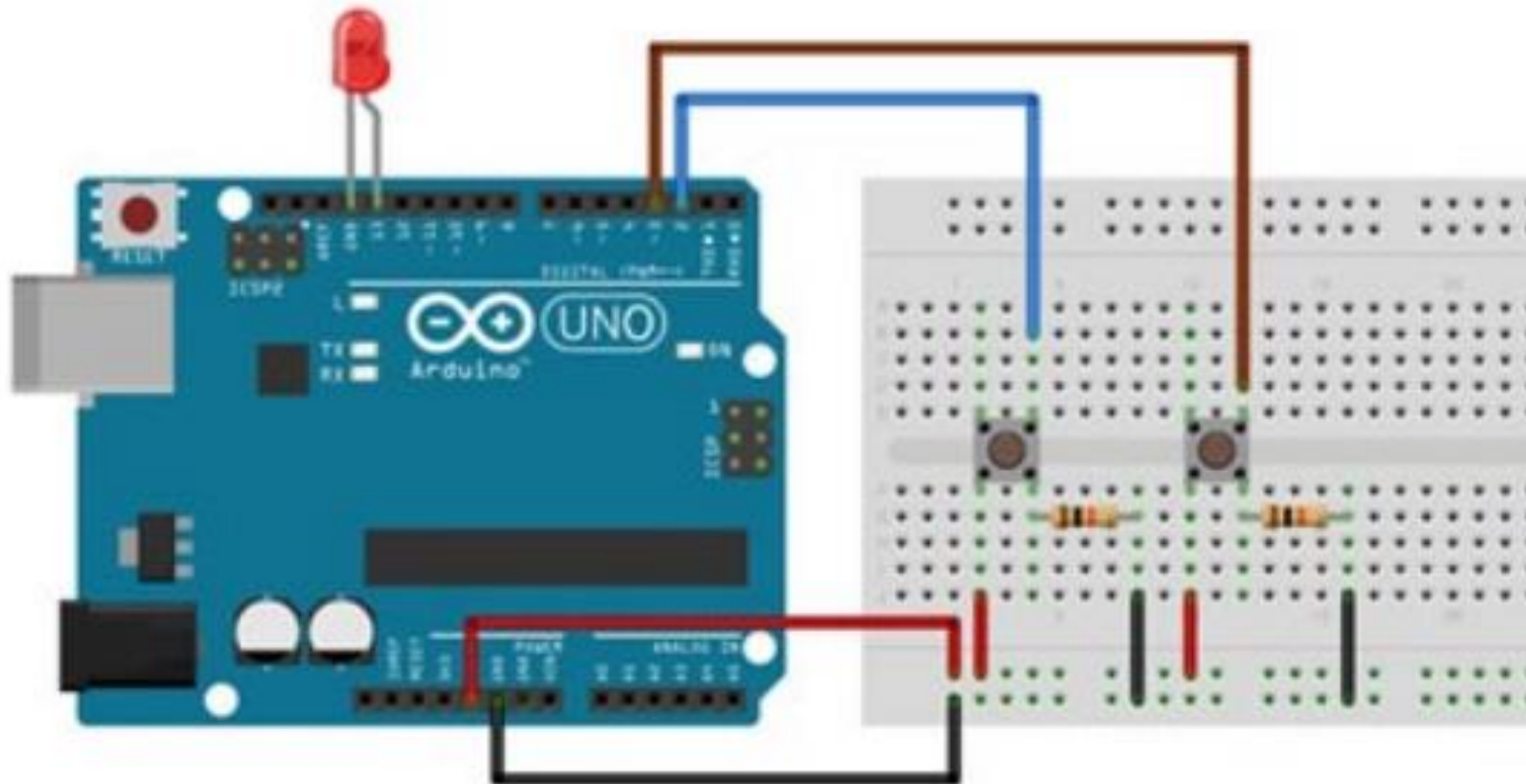
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

```
void setup() {  
  // define o numero de colunas e linhas do Display :  
  lcd.begin(16, 2);  
  // Envia a mensagem para o display.  
  lcd.print("FACULDADE QI");  
  lcd.setCursor(0, 1);  
  //Posiciona o cursor na primeira coluna(0) e na segunda linha(1) do  
  Display  
  lcd.print(" (cc) 2017 ");  
}
```

```
void loop() {  
  
}
```

Exercício 1

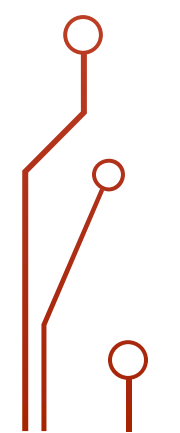
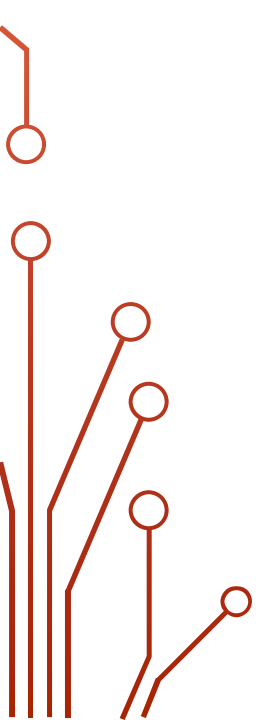
Para evitar acidentes no ambiente de trabalho, uma regra de segurança em vários equipamentos industriais é obrigar que um usuário aperte dois botões, um com cada mão, para acionar uma máquina. É o caso da máquina de corte usada em fábricas de papel. Com a seguinte montagem podemos simular esta situação. O LED somente acende se os dois botões do circuito estiverem pressionados:



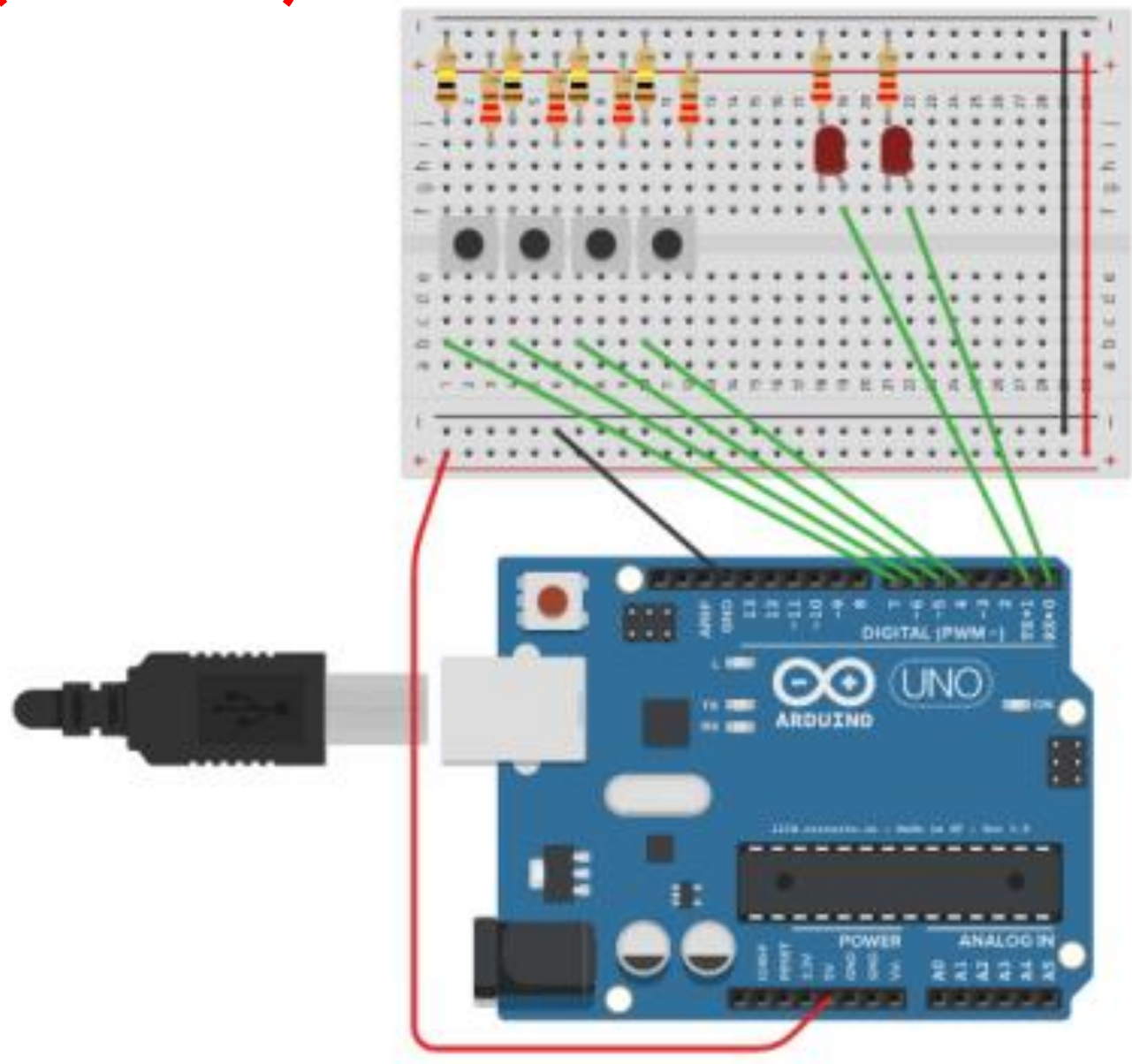


Exercício 2

Faça mais uma modificação no código fonte do exercício 1 para que você possa acender o LED do pino 13 pressionando ou o botão 1 ou o botão 2. Ao deixar de pressionar, o LED se apaga.



(Desafio)



(Desafio)

Escreva um programa para a plataforma Arduino Uno, tal que:

- Os pino de entrada/saída de números 4 a 7 sejam configurados como entradas;
- O pino de entrada/saída de números 0 e 1 sejam configurado como saídas;
- A partir do momento em que o pino 5 for para nível lógico alto, apenas o LED conectado à saída 1 deverá piscar;
- A partir do momento em que o pino 4 for para nível lógico alto, apenas o LED conectado à saída 0 deverá piscar;
- O LED que estiver piscando deverá fazê-lo:
 - em um intervalo de 200ms a partir do momento em que a entrada de número 7 for a nível lógico alto;
 - em um intervalo de 400ms a partir do momento em que a entrada de número 6 for a nível lógico alto;

The image features a minimalist design with the word "SHIELDS" in a large, bold, red sans-serif font centered on a white background. The corners of the image are decorated with thin, dark red lines that form abstract circuit board patterns, including straight lines, right-angle turns, and small circles representing solder pads or vias. These decorative elements are located in the top-left, top-right, bottom-left, and bottom-right corners.

SHIELDS

Um shield é uma placa que permite expandir as funcionalidades originais do Arduino. Alguns exemplos:

Arduino Ethernet Shield R3



Kit Motor Shield R3



Arduino WiFi Shield



Arduino XBee Shield



Arduino ProtoShield R3



Kit Joystick Shield



Shield celular com SM5100B



Shield GPS



Shield LCD Colorido



Shield MP3 Player



Shield WiFly



Wireless SD Shield

