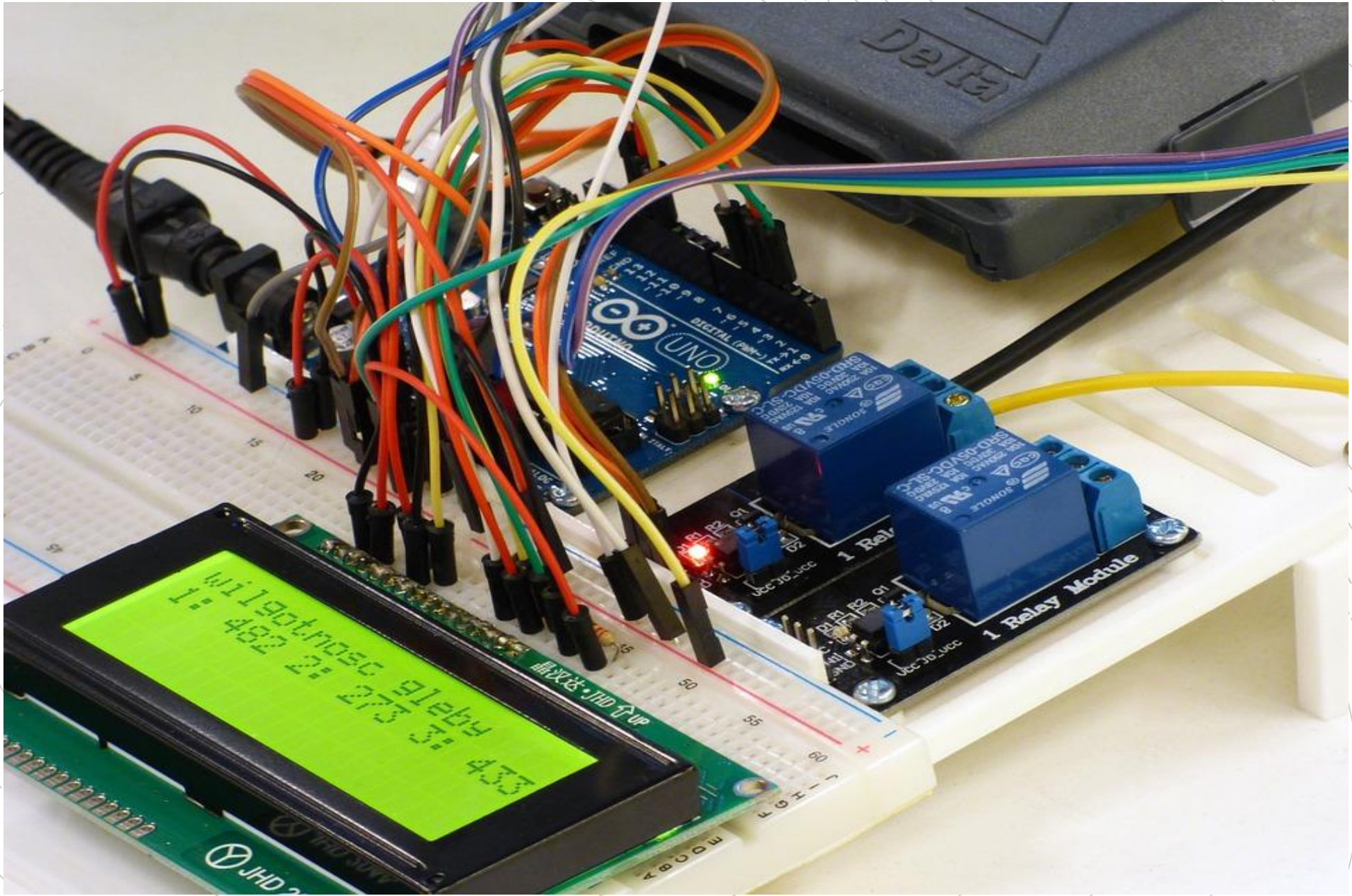

UNO
1 Relay Module
JHD

Potenciômetro

Potenciômetro

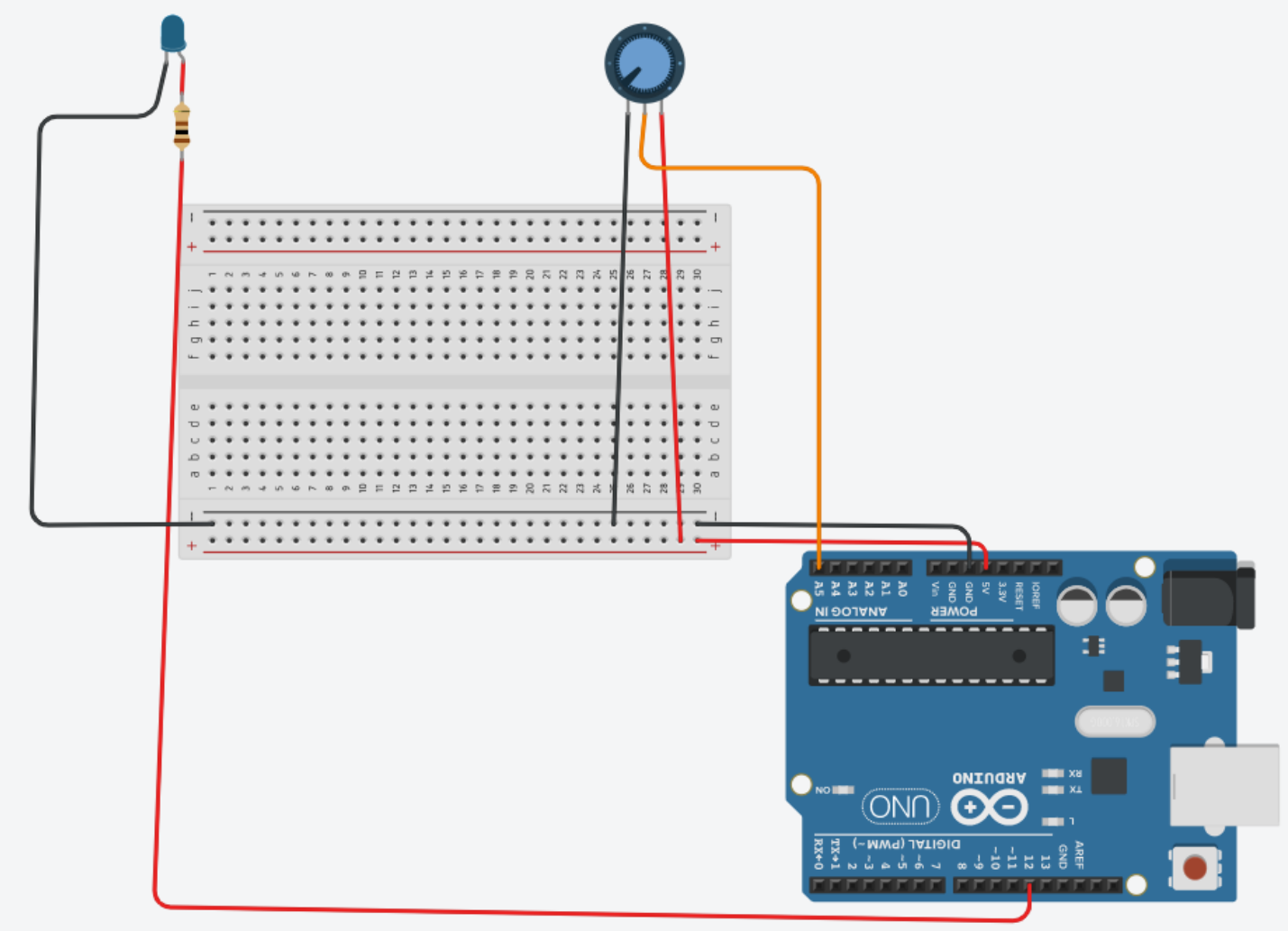
Um potenciômetro é um tipo de resistor variável que é comumente usado em eletrônica para ajustar a resistência em um circuito, controlando a quantidade de corrente que passa por ele.

Quando utilizado em projetos com Arduino, o potenciômetro pode ser empregado para diversas finalidades, como controlar a intensidade de LEDs, ajustar o volume de um sinal de áudio, ou fornecer entradas analógicas que podem ser lidas pelo microcontrolador.

Potenciômetro

Entrada Analógica no Arduino: Quando ligado a uma entrada analógica do Arduino, o potenciômetro pode fornecer um valor de tensão variável, que é lido pelo pino analógico como um valor entre 0 e 1023 (em um Arduino padrão com uma resolução de 10 bits).

Ligação



Programação

```
int pot = A5;
int led = 13;
int valmed = 0;
void setup()
{
  pinMode(led, OUTPUT);
}

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

Função Map

A função map(), é uma função matemática que pode ser utilizada para transformar uma sequência de números em uma outra. Para utilizá-la é necessário inserir a primeira faixa numérica, que é a obtida a partir de uma medição utilizando as entradas analógicas, o número menor e o maior medido

Sintaxe:

```
map(variável, do valor menor , do valor maior , para menor valor , para o maior valor);
```

```
exemplo(A5, 0 , 1023, 255, 700);
```

Exemplo Função Map

```
int sensor = 0;
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  sensor = map(analogRead(A5),0,1023,0,100);
  //Serial.print para mostrar frases ou palavras
  Serial.print("Valor = ");
  Serial.println(sensor);
  delay(500);
}
```

map(variável, do valor menor , do valor maior , para menor valor , para o maior valor);

Exemplo Função Map

int sensor = 0;	Cria a variável sensor do tipo int e já atribui-se o valor 0 à ela.
void setup() {	Inicia-se o método setup.
Serial.begin(9600); }	Inicia-se a comunicação serial.
void loop() {	Inicia-se o método loop.
sensor = map (analogRead(A0),0,1023,0,100);	Atribui-se o valor lido em A0 já mapeado para a variável sensor.
Serial.print("Valor=");	
Serial.println(sensor);	Escreve na serial o valor da variável sensor e pula linha.

delay(1000);
 	
}
	Espera 1 segundo para nova leitura.

PWM

Um pino **PWM** (Pulse Width Modulation) em um microcontrolador, como o Arduino, é um pino digital que pode gerar um sinal de saída com diferentes larguras de pulso.

O PWM é uma técnica de modulação de largura de pulso usada para controlar a potência entregue a dispositivos eletrônicos, como LEDs, motores e outros atuadores.

COMO FUNCIONA

Sinal PWM alterna entre ligado (HIGH) e desligado (LOW) muito rapidamente.

A relação entre o tempo em que o sinal está **ligado** e o tempo total de um ciclo completo é chamada de **duty cycle** (ciclo de trabalho).

O ciclo de trabalho é expresso em porcentagem e determina a potência média entregue:

- ✓ **0%**: O sinal está sempre em LOW (desligado), ou seja, sem energia.
- ✓ **50%**: O sinal passa metade do tempo ligado e metade desligado, resultando em 50% da potência máxima.
- ✓ **100%**: O sinal está sempre em HIGH (ligado), fornecendo potência máxima.

EXEMPLO

Por exemplo, ao controlar o brilho de um LED:

Com um duty cycle de **0%**, o LED estará totalmente apagado.

Com um duty cycle de **50%**, o LED estará em meia-luz.

Com um duty cycle de **100%**, o LED estará em brilho máximo..

EXEMPLO

Por exemplo, ao controlar o brilho de um LED:

Com um duty cycle de **0%**, o LED estará totalmente apagado.

Com um duty cycle de **50%**, o LED estará em meia-luz.

Com um duty cycle de **100%**, o LED estará em brilho máximo..

USO NO ARDUINO

O Arduino tem pinos digitais específicos que suportam PWM. No Arduino Uno, por exemplo, os pinos que podem gerar sinais PWM são marcados com o símbolo ~ (pinos 3, 5, 6, 9, 10 e 11).

Quando você usa a função analogWrite(pino, valor) no Arduino:

- O valor deve estar entre **0** e **255**, onde:
  - **0** equivale a 0% (sempre desligado).
  - **255** equivale a 100% (sempre ligado).

CÓDIGO

```
// Definir pinos
const int potenciometroPin = A0;
const int ledPin = 9;

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

void loop() {
  // Ler o valor do potenciômetro (entre 0 e 1023)
  int valorPotenciometro = analogRead(potenciometroPin);

  // Mapear o valor do potenciômetro para o intervalo de PWM (0 a 255)
  int valorPWM = map(valorPotenciometro, 0, 1023, 0, 255);

  // Ajustar o brilho do LED
  analogWrite(ledPin, valorPWM);

  // Pequena pausa para estabilizar
  delay(10);
}
```

Exemplo Função Map

1) Monte um circuito com 3 LEDs, sendo verde, vermelho e amarelo conectados nas saídas 2,3,4 respectivamente.

Faça o mapeamento do potenciômetro.
Elabore programa que faça o LED vermelho acender quando o potenciômetro estiver na metade do seu cursor.
O LED verde conectado ao pino 2 deve acender com o quando o potenciômetro estiver em 185°. O LED amarelo conectado ao pino 4 acender quando o potenciômetro em 50° graus.

Obs. 2 ou mais led´s não deveram ficar ligados ao mesmo.

2) Monte um circuito com 2 LEDs, sendo vermelho e amarelo conectados nas saídas 5,6 respectivamente.

Faça o mapeamento e controle (liga e desliga) do LED vermelho pelo potenciômetro e o amarelo acende e apaga ao pressionar alguma tecla do teclado.

Enviar o código e o esquema do arduino através do arquivo do Word.