

# Rel06 – Introdução à Microcontrolador

O que deve ser entregue ao final desse relatório:

- Um arquivo .pdf contendo:
  - As práticas devem ser feitas usando o simulador Tinkercad.
    - Tire capturas de tela do resultado e coloque no relatório
    - Substitua o Esp32 do relatório pelo arduino do Tinkercad.

## Exemplos de uso de componentes:

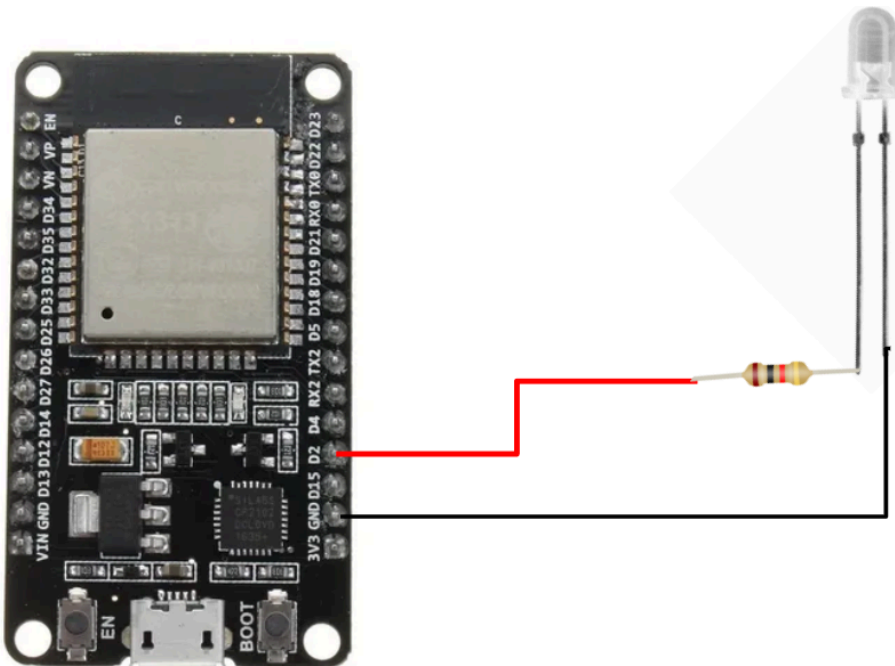
**Prática 1 - Pisca Pisca:** Na primeira prática, iremos controlar um led pelo Esp32 (Arduino no simulador Tinkercad).

Componentes que iremos usar:

- 1 Led.
- 1 Resistor (1K -> marrom, preto, vermelho).

Para isso, faça as seguintes ligações:

- A porta D2 do Esp32 (ou Arduino) deve ser ligada em uma das pernas do resistor (1K).
- A outra perna do resistor (1K) deve ser ligada no polo positivo do led.
- O polo negativo do led deve ser ligado na porta GND do Esp32 (ou Arduino).



O seguinte código deve ser escrito na IDE do arduino:

```
const int led1 = 2;

void setup() {
    pinMode(led1, OUTPUT);
}

void loop() {
    digitalWrite(led1, 1);
    delay(1000);
    digitalWrite(led1, 0);
    delay(1000);
}
```

Exercícios:

1. Teste o código acima e explique o que cada comando faz.
2. Altere para que o led pisque de 5 em 5 segundos.
3. Altere para ficar ligado por 10 segundos e desligado por 5 segundos.
4. Monte um novo circuito com 3 leds (1 verde, 1 amarelo e 1 vermelho). Simule o funcionamento de um semáforo, sendo que o led verde fica ligado por 4 segundos, amarelo por 1 segundo e vermelho por 5 segundos).

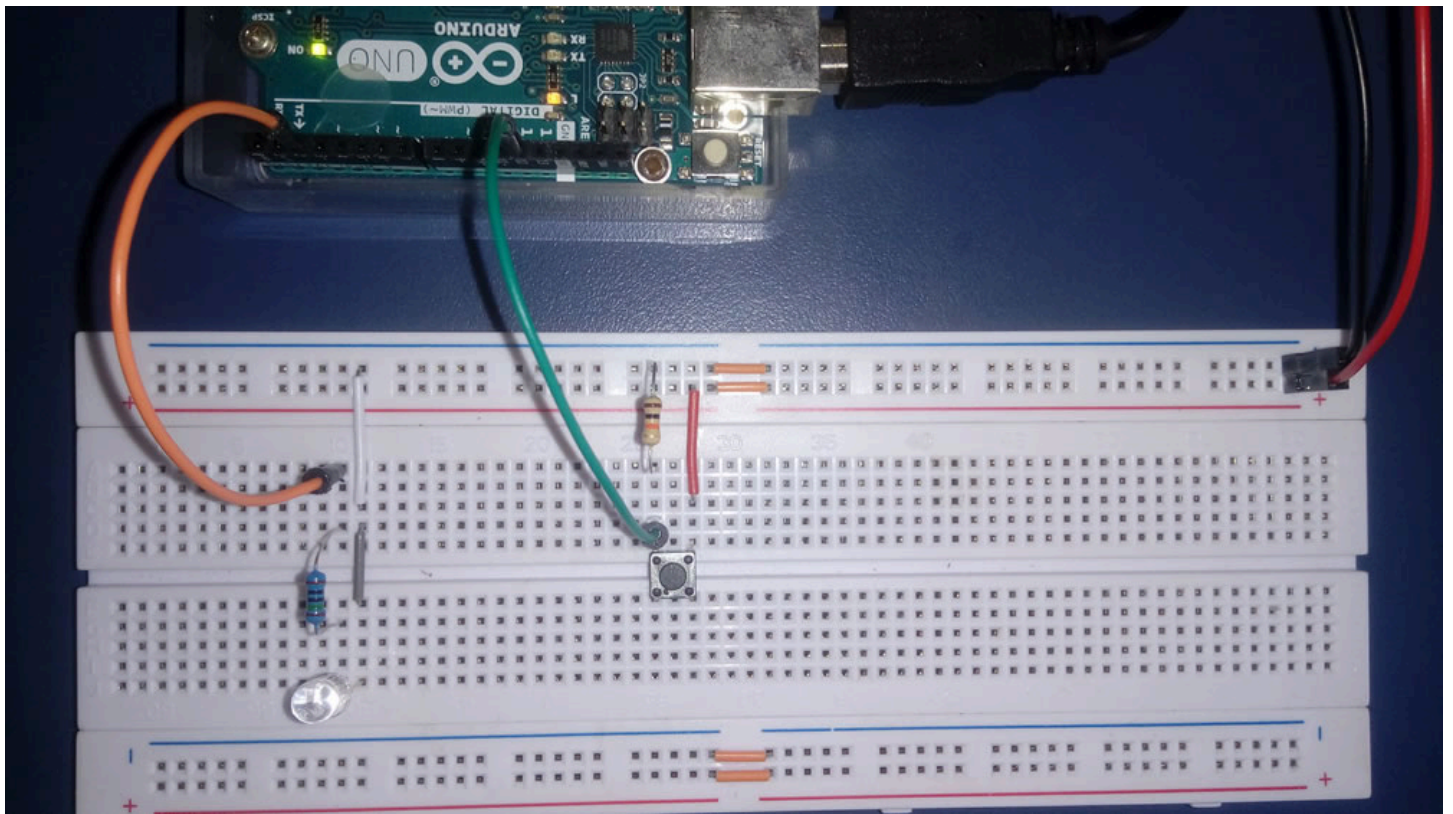
**Prática 2 - Pisca Pisca com Botão:** Na segunda prática, iremos controlar um led e um botão pelo Esp32 (ou Arduino).

Componentes que iremos usar:

- 1 Led.
- 2 Resistores (1K -> marrom, preto, vermelho).
- 1 Push Button.

Para isso, faça as seguintes ligações:

- A porta D2 do Esp32 (ou Arduino) deve ser ligada em uma das pernas do resistor (1K).
- A outra perna do resistor (1K) deve ser ligada no polo positivo do led.
- O polo negativo do led deve ser ligado na porta GND do Esp32 (ou Arduino).
- A porta D4 do Esp32 (ou Arduino) deve ser ligada a uma das extremidades do botão. Essa mesma extremidade também deve ser ligada a um resistor, em que a outra extremidade desse resistor deve ser ligado ao GND do Esp32 (ou Arduino).
- A extremidade ao lado da usada do botão, deve ser ligada ao 3V do Esp32 (ou Arduino).



O seguinte código deve ser escrito na IDE do arduino:

```
const int led = 2;
const int botao = 4;
int estadoBotao;

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

void loop() {
  estadoBotao = digitalRead(botao);
  if (estadoBotao == 1) {
    digitalWrite(led, 1);
  }
  if (estadoBotao == 0) {
    digitalWrite(led, 0);
  }
}
```

Exercícios:

5. Teste o código acima e explique o que cada comando faz.
6. Criar um programa que quando pressionarmos o botão, um led deverá piscar 5 vezes e de meio em meio segundo.

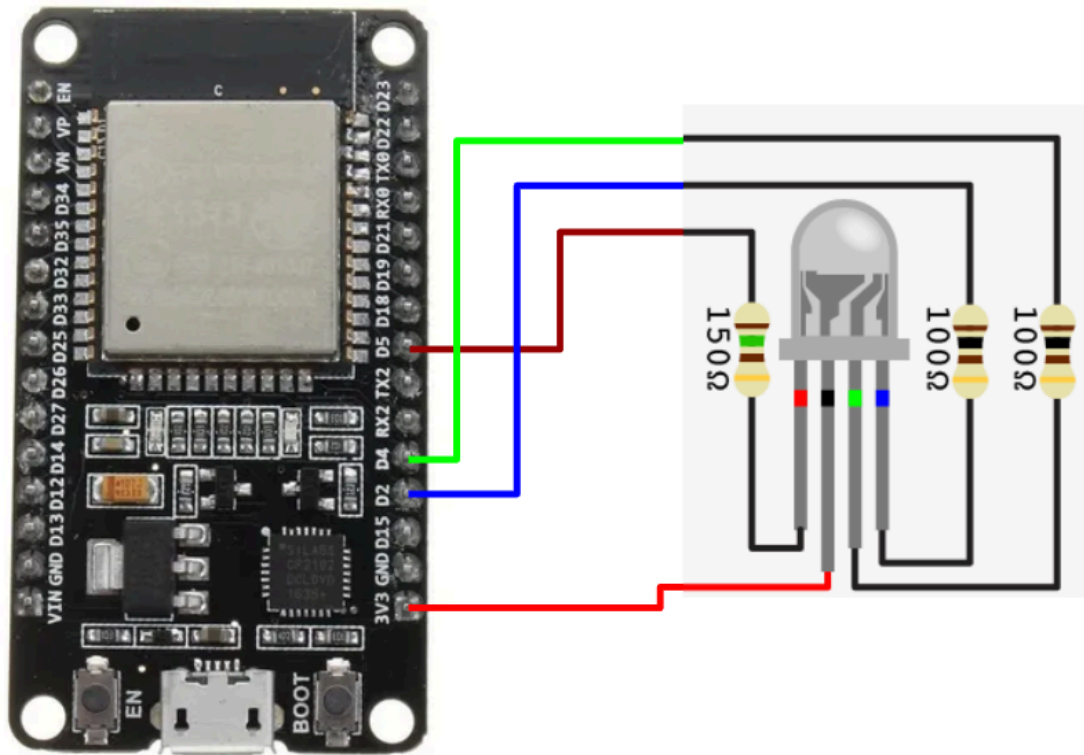
### Prática 3 - LED RGB: Na terceira prática, iremos controlar um led RGB pelo Esp32 (ou Arduino).

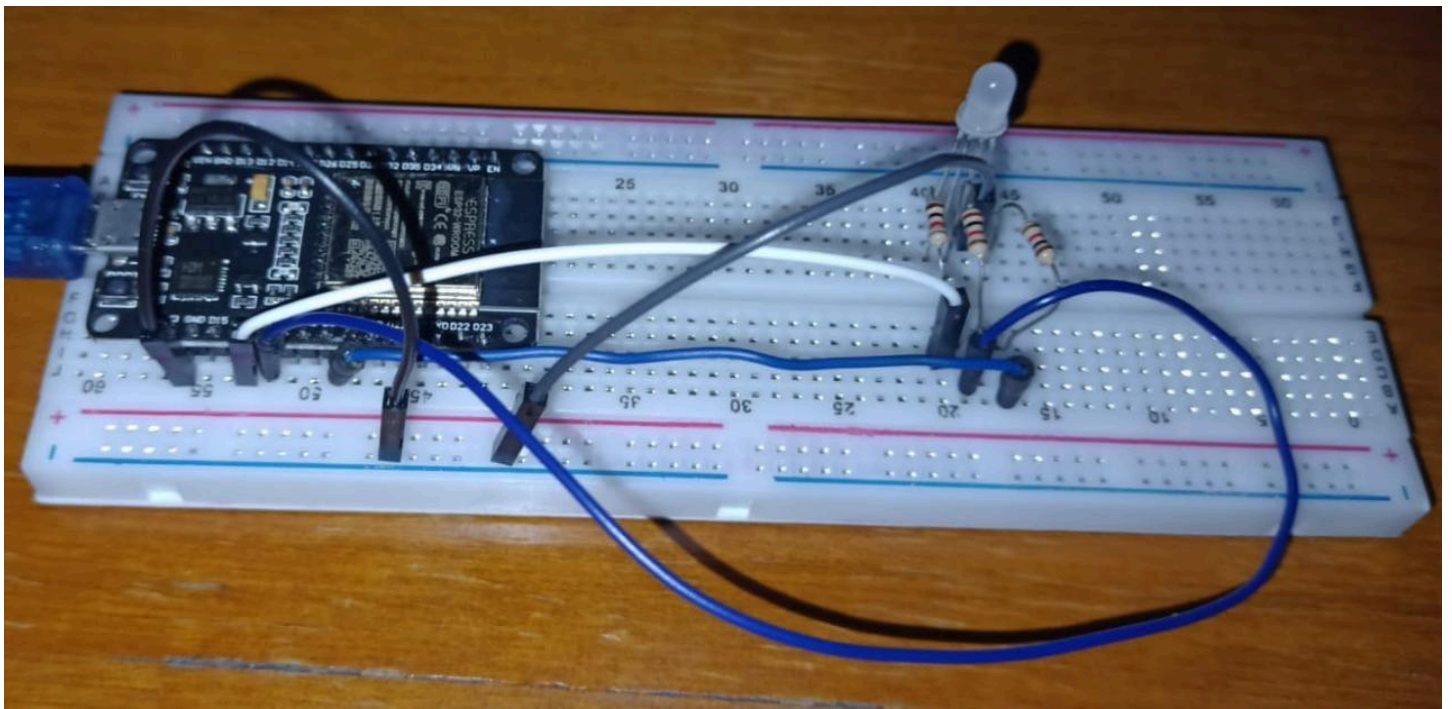
Componentes que iremos usar:

- 1 Led RGB.
- 2 Resistores (100  $\Omega$  -> marrom, preto, marrom).
- 1 Resistor (150  $\Omega$  -> marrom, verde, vermelho).

Para isso, faça as seguintes ligações:

- A porta D2 do Esp32 (ou Arduino) deve ser ligada em uma das pernas do resistor (1K).
- A outra perna do resistor (1K) deve ser ligada no polo positivo do led.
- O polo negativo do led deve ser ligado na porta GND do Esp32 (ou Arduino).





O seguinte código deve ser escrito na IDE do arduino:

```
const int azul = 2;
const int verde = 4;
const int vermelho = 5;

void setup() {
]
  pinMode(verde, OUTPUT);
  pinMode(azul, OUTPUT);
  pinMode(vermelho, OUTPUT);
}

void loop() {
  digitalWrite(vermelho, 0);
  digitalWrite(azul, 1);
  digitalWrite(verde, 1);
  delay(2000);

  digitalWrite(vermelho, 1);
  digitalWrite(azul, 0);
  digitalWrite(verde, 1);
  delay(2000);

  digitalWrite(vermelho, 1);
  digitalWrite(azul, 1);
  digitalWrite(verde, 0);
  delay(2000);

}
```

Exercícios:

- 7. Crie um programa que mostre cor a cor do led RGB (mostrando todas as cores) e altere cada cor a cada 1 segundo.

8. Utilize o led RGB para simular o funcionamento de um semáforo, sendo que o led verde fica ligado por 5 segundos, amarelo por 1 segundo e vermelho por 3 segundos).