

Microprocessadores e Sistemas Embebidos

LAB 4

Controlo de LEDs com Botões

Trabalho realizado por:

Guilherme Mesquita nº 1706041

João Pereira nº 1706083

Rui Sequeira nº1012122

1. Usando o circuito da Figura 1, como parte do circuito global, implemente um projeto (esquema, algoritmo, circuito e código) que permita ligar um LED quando o **switch button** é pressionado.

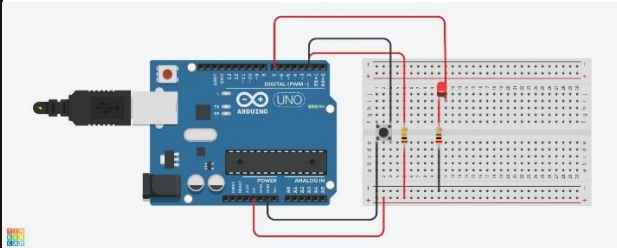
```

#define LED_PIN 7
#define BUTTON_PIN 2

void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT);
}

void loop() {
  if (digitalRead(BUTTON_PIN) == LOW) {
    digitalWrite(LED_PIN, HIGH);
  }
  else {
    digitalWrite(LED_PIN, LOW);
  }
}

```



2. Faça uma alteração ao projeto do Exercício 1 para que quando o **switch button for** pressionado o LED fique desligado.

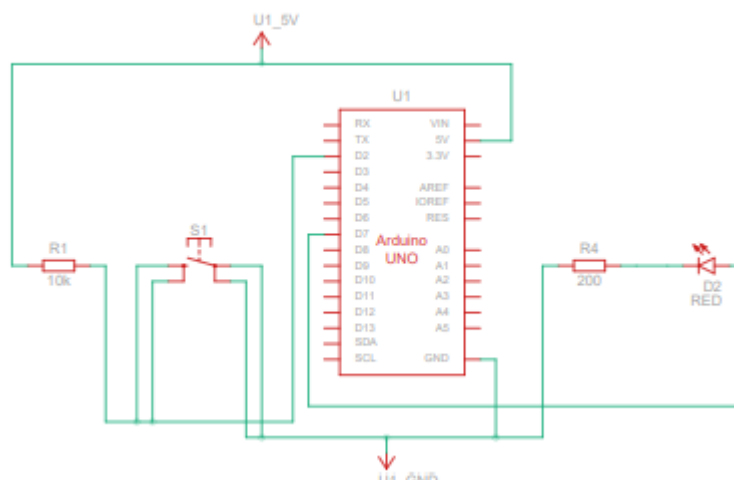
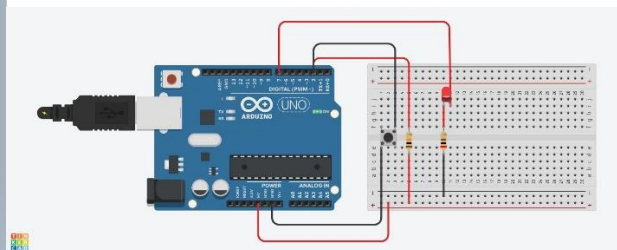
```

#define LED_PIN 7
#define BUTTON_PIN 2

void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT);
}

void loop() {
  if (digitalRead(BUTTON_PIN) == HIGH) {
    digitalWrite(LED_PIN, HIGH);
  }
  else {
    digitalWrite(LED_PIN, LOW);
  }
}

```



3. Implemente os Exercícios 1 e 2 apenas com 1 linha de código na função loop().

```
#define LED_PIN 7
#define BUTTON_PIN 2

void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT);
}

void loop() {
  digitalWrite(LED_PIN,digitalRead(BUTTON_PIN));
}
```

Figura 1- Quando se carrega no botão desliga

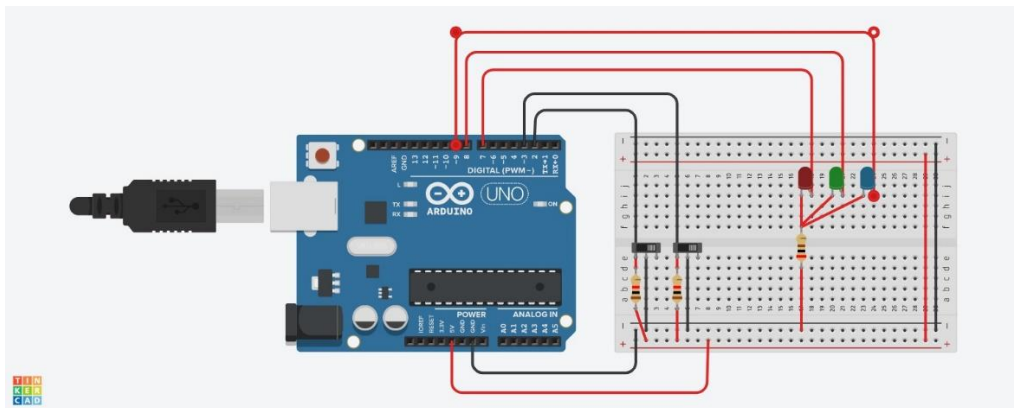
```
#define LED_PIN 7
#define BUTTON_PIN 2

void setup() {
  pinMode(LED_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT);
}

void loop() {
  digitalWrite(LED_PIN,!digitalRead(BUTTON_PIN));
}
```

Figura 2- Quando se carrega no botão liga

4. Usando **2 slider buttons** e **3 LEDs**, implemente, para cada um dos LEDs, as funções AND, OR e XOR fazendo uso dos **2 switch buttons**. O **LED 1** deve responder à função AND, o LED 2 deve responder à função OR, enquanto o **LED 3** deve responder à função XOR.



```
#define LED_PIN_AND 7
#define LED_PIN_OR 8
#define LED_PIN_XOR 9
#define BUTTON_PIN_1 2
#define BUTTON_PIN_2 3

int x = 0;
int y = 0;

void setup() {
    Serial.begin(9600);

    pinMode(LED_PIN_AND, OUTPUT);
    pinMode(LED_PIN_OR, OUTPUT);
    pinMode(LED_PIN_XOR, OUTPUT);
    pinMode(BUTTON_PIN_1, INPUT);
    pinMode(BUTTON_PIN_2, INPUT);
}

void loop() {
    x = digitalRead(BUTTON_PIN_1);
    y = digitalRead(BUTTON_PIN_2);

    digitalWrite(LED_PIN_AND, x&y);
    digitalWrite(LED_PIN_OR, x|y);
    digitalWrite(LED_PIN_XOR, x^y);

    Serial.println(x&y);
}
```

5. Pesquise na documentação da função `pinMode()` as opções de configuração. Implemente todos os exercícios anteriores sem usar a resistência R1 no circuito.

Em vez de usarmos uma resistência, utilizamos a própria resistência do Arduino, resistências Pull-Up.

```
#define LED_PIN_AND 7
#define LED_PIN_OR 8
#define LED_PIN_XOR 9
#define BUTTON_PIN_1 2
#define BUTTON_PIN_2 3

int x = 0;
int y = 0;

void setup() {
    Serial.begin(9600);

    pinMode(LED_PIN_AND, OUTPUT);
    pinMode(LED_PIN_OR, OUTPUT);
    pinMode(LED_PIN_XOR, OUTPUT);
    pinMode(BUTTON_PIN_1, INPUT_PULLUP);
    pinMode(BUTTON_PIN_2, INPUT_PULLUP);
}

void loop() {
    x = digitalRead(BUTTON_PIN_1);
    y = digitalRead(BUTTON_PIN_2);

    digitalWrite(LED_PIN_AND, x&y);
    digitalWrite(LED_PIN_OR, x|y);
    digitalWrite(LED_PIN_XOR, x^y);

    Serial.println(x&y);
}
```

