

ESCOLA DE ENGENHARIA DA UFMG

PROJETO DE SISTEMAS EMBUTIDOS

---

## Laboratório 1 - Pisca LED

---

*Nome:* Giovanni Martins de Sá Júnior

*Matrícula:* 2017001850

*Semestre:* 2023/2

# Conteúdo

<b>1</b>	<b>Descrição do SDK</b>	<b>1</b>
<b>2</b>	<b>Interface GPIO de Entrada</b>	<b>3</b>
<b>3</b>	<b>Interface GPIO de Saída</b>	<b>4</b>
<b>4</b>	<b>Experimento</b>	<b>5</b>
<b>5</b>	<b>Conclusão</b>	<b>7</b>

# 1 Descrição do SDK

1. Liste o nome, modelo, microcontrolador, memórias e interfaces de E/S.
  - Nome: Arduino Uno R3.
  - Modelo: ATmega328P.
  - Memórias:
    - Flash (32 KB);
    - SRAM (2 KB);
    - EEPROM (1 KB).
  - Interfaces de E/S:
    - 14 pinos digitais de E/S (dos quais 6 podem ser usados como saídas PWM);
    - 6 entradas analógicas;
    - 1 porta USB;
    - 1 conector de alimentação;
    - 1 header ICSP;
    - 1 botão de Reset.
2. Apresente uma imagem com a localização de todos os sinais de interface disponíveis.

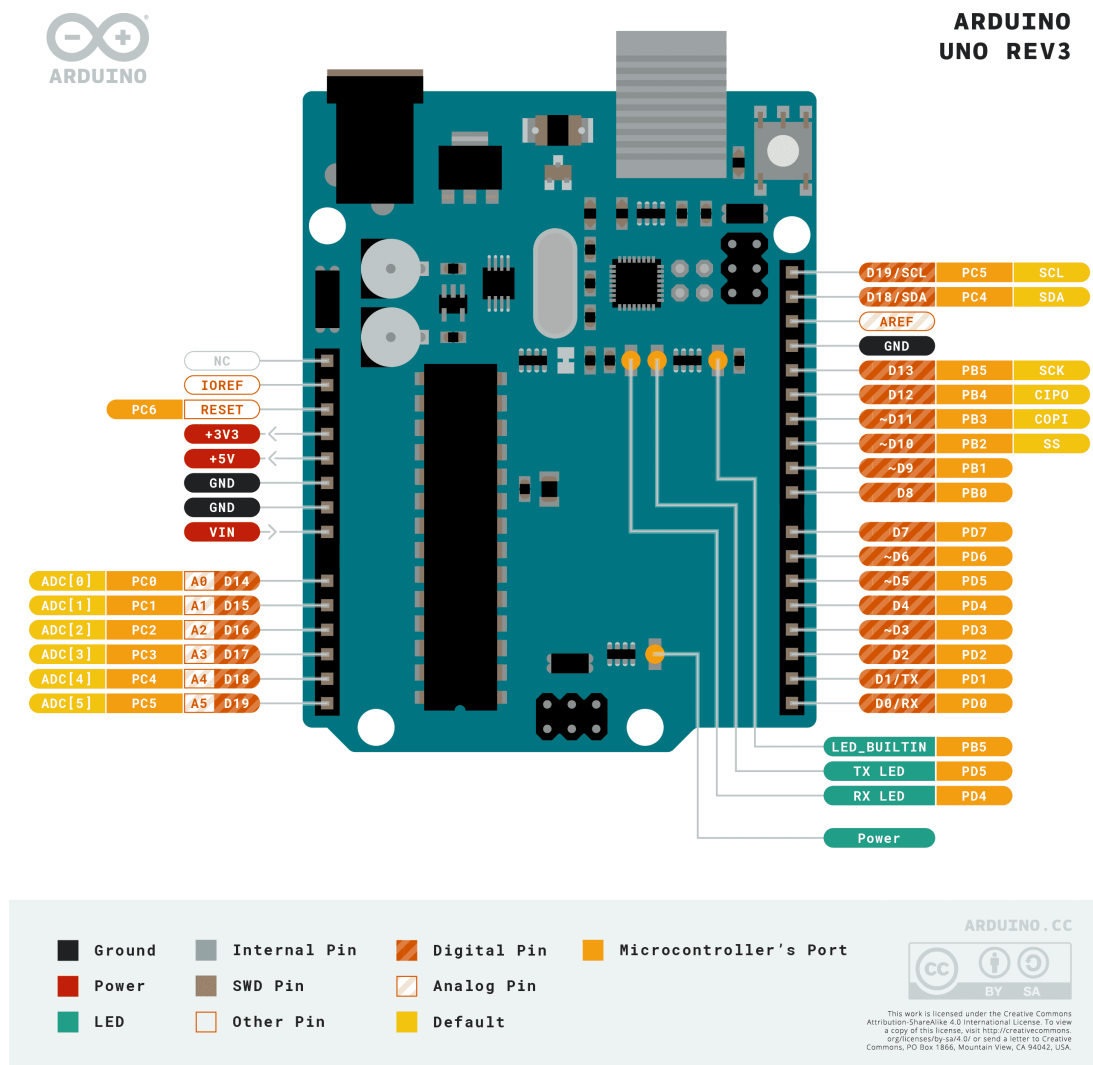


Figura 1: Fonte: <https://docs.arduino.cc/retired/boards/arduino-uno-rev3-with-long-pins>

3. Cite qual ambiente de desenvolvimento de SW será utilizado.

- Arduino IDE 2.2.1.

## 2 Interface GPIO de Entrada

1. Descreva como uma chave está, ou será, conectada a um pino de entrada.

- Para o desenvolvimento do trabalho, a chave se ao pino 2 do Arduino, com o auxílio de um resistor pull-up de 220 Ohms.

2. Inclua um diagrama elétrico. Seu uC tem resistor de pull-up interno?

- Sim, o Arduino Uno R3 apresenta um resistor de pull-up interno, porém ele é mantido desativado por configuração padrão. Contudo, para a realização do trabalho foi utilizado um resistor externo. Abaixo, é possível visualizar o diagrama elétrico do projeto, com o auxílio do software Tinkercad.

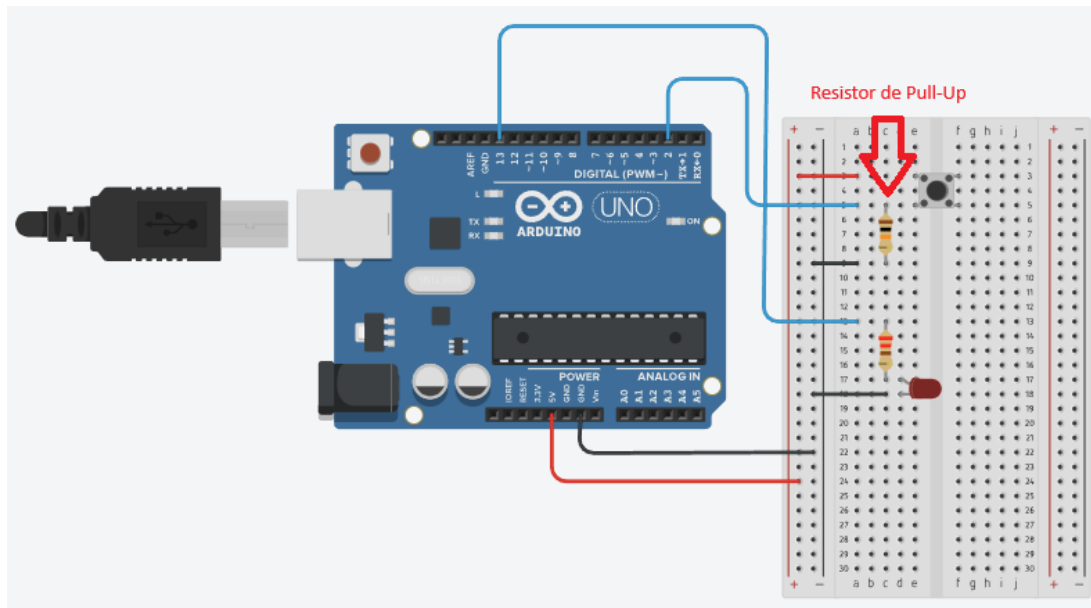


Figura 2: Diagrama Elétrico do Projeto. Desenvolvimento próprio

3. Descreva como esse pino será configurado e usado.
  - O pino 2 será utilizado como entrada com o resistor pull-up. Em seguida, esse pino será utilizado para ler o estado da chave, com o parâmetro *estado\_Botao* na função *loop*. Assim, a variável *estado\_Botao* armazenará o valor HIGH caso a chave estiver pressionada, e LOW caso estiver solta.

### 3 Interface GPIO de Saída

1. Descreva como um LED está, ou estará, ligado a um pino GPIO de saída
  - No problema descrito, o LED estará conectado ao pino 13 do Arduino, juntamente de um resistor e com uma fonte de 5V para o circuito.
2. Inclua um diagrama elétrico.
  - O mesmo, vide Figura 2.
3. Descreva como esse pino será configurado e usado.
  - O pino 13 será definido como saída. Em seguida, ele será utilizado exatamente para acender e apagar o LED. Para isso, será utilizado a função *digitalWrite()* dentro do *loop*, passando a variável *ledpino* e o valor HIGH ou LOW dependendo da situação.

## 4 Experimento

- Nesta seção, será tratado da implementação prática do trabalho. Neste sentido, declaramos logo abaixo, a implementação esperada a respeito do projeto, e em seguida, a implementação desenvolvida para o mesmo.
- Algoritmo de Funcionamento:
  1. Esperar a chave ser pressionada e liberada.
  2. Acender o LED por 1 segundo.
  3. Apagar o LED por 2 segundo.
  4. Finalmente, piscar rapidamente o LED por meio segundo indicando fim de ciclo, ou seja, acenda por 0,25 segundo e em seguida apague por 0,25 segundo.
  5. Volte ao passo 1.
- Algoritmo desenvolvido: Como pode ser visto abaixo, a implementação é dividida em três momentos: a primeira, composta pela declaração das variáveis, a segunda pela definição da função *setup()*, em que são definidas a entrada e a saída do circuito com o método *pinMode*, e no terceiro bloco, definido pela função *loop()*, em que se executa as instruções definidas pelo algoritmo.

```

1 // Declaracao de Variaveis
2 int botao_Pino = 13;
3 int led_Pino = 2;
4 int estado_Botao = LOW;
5
6 void setup() {
7     // Pino do Botao
8     pinMode(botao_Pino, INPUT);
9     // Pino do Led
10    pinMode(led_Pino, OUTPUT);
11 }
12
13 void loop() {
14     // Leitura do Estado da Chave
15     estado_Botao = digitalRead(botao_Pino);
16
17     // A chave foi pressionada
18     if(estado_Botao == HIGH) {
19         // Aguarda a liberacao da Chave
20         while(digitalRead(botao_Pino) == HIGH) {} // Passo 1
21
22         // Acende o Led
23         digitalWrite(led_Pino, HIGH); // Passo 2
24         delay(1000);
25         // Desliga o Led
26         digitalWrite(led_Pino, LOW); // Passo 3
27         delay(2000);
28
29         // Pisca o Led - Passo 4
30         digitalWrite(led_Pino, HIGH);
31         delay(250);
32         digitalWrite(led_Pino, LOW);
33         delay(250);
34     }
35     // Fim do Loop - Passo 5
36 }
37

```

Listing 1: Implementação do Código



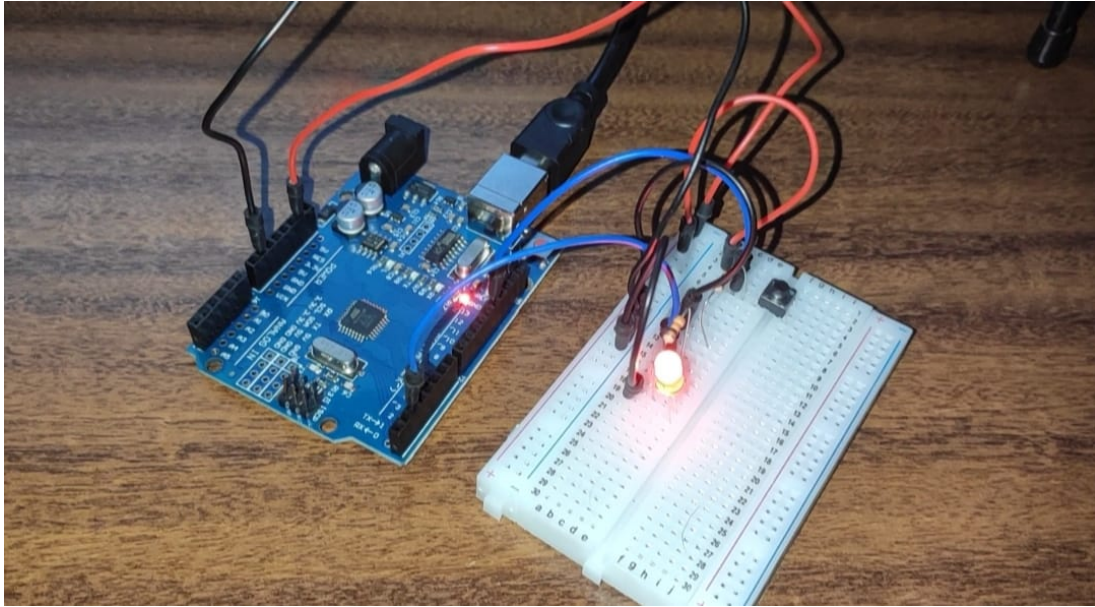


Figura 3: Montagem Final do Circuito

## 5 Conclusão

A realização do trabalho não trouxe muitas dificuldades para a sua resolução. A prototipação do projeto com o auxílio do tinkercad facilitou muito a montagem do circuito real, fazendo com que a danificação de componentes causada pela montagem incorreta pudesse ser evitada.

O único impedimento enfrentado, foi na escolha do resistor para interface de saída. Idealmente, para os cálculos de corrente e queda de tensão do resistor, seria melhor a escolha de um resistor de 220 Ohms. Contudo, o resistor de valor mais próximo disponível para o desenvolvimento do projeto, era de 330 Ohms, o que não prejudicou na execução da tarefa. Abaixo, é possível a montagem final do Pisca Led.