



## **Laboratório de Materiais e Equipamentos Elétricos**

### **Roteiro 6: Arduino - Saída Digital**

#### **1. Objetivos**

- Analisar e compreender as portas (pinos) de um Arduino.
- Definir função de entrada ou saída ao pino desejado.
- Acionar e controlar LEDs a partir de programação com Arduino.

#### **2. Material utilizado**

- Resistores de 150 Ohms
- Protoboard
- 2 LEDs
- Arduino UNO

##### **2.1. Saídas Digitais (Arduino)**

Um microcontrolador, como o Arduino, possui portas de saída digitais (que irão ter como saída 0 V ou 5 V). A Figura 1 mostra como as portas de um Arduino são distribuídas.

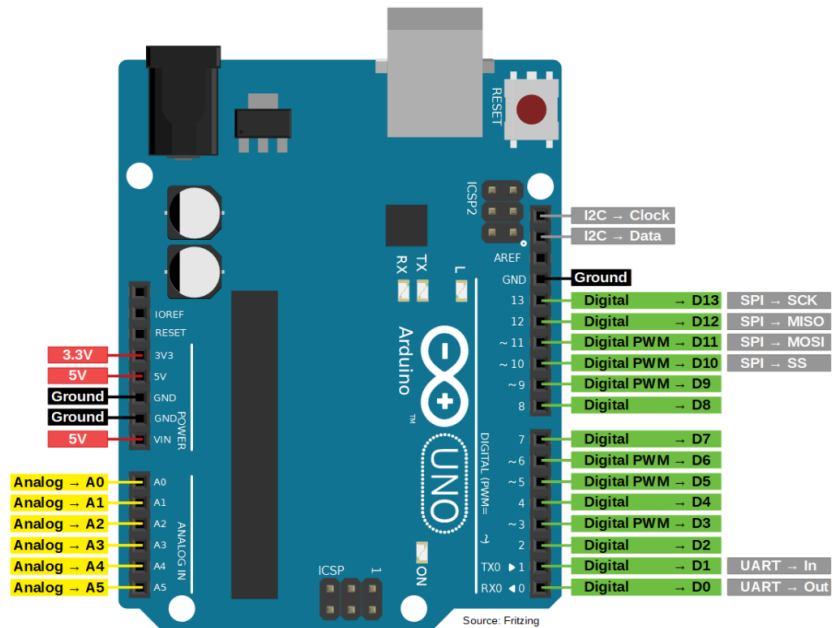


Figura 1: pinos disponíveis no Arduino UNO

### Exemplo de Configuração da porta de saída:

Considere o caso em que você deseje ligar e desligar um LED que esteja conectado na porta 13 como mostrado na Figura 2. No projeto queremos que o LED fique aceso por 1 segundo e depois apagado por 1 segundo em loop.

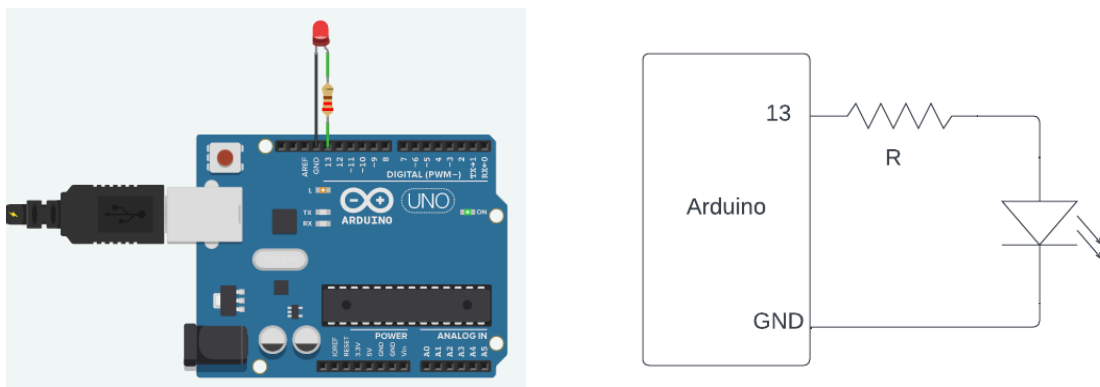


Figura 2: Montagem de um LED em série com resistor na porta 13 do Arduino.

Primeiramente, precisamos configurar que a porta 13 será de **SAÍDA**. Isso é feito dentro do loop setup pela função *pinMode*, onde ajustamos que o pino 13 será de saída (OUTPUT).

```
void setup()
{
  pinMode(13, OUTPUT);
}
```

Dentro do laço loop o código irá se repetir. É no loop que definimos o comportamento do LED. A função *digitalWrite* estabelece se a saída no pino será alta (5 V) ou baixa (0 V). A função *delay* gera um atraso em milissegundos. Portanto, para fazer com que o código espere por 1 segundo usamos *delay(1000)*.

```
void loop()
{
  // liga o LED
  digitalWrite(13, HIGH);
  delay(1000); // aguarda 1000 milissegundos

  // desliga o LED
  digitalWrite(13, LOW);
  delay(1000); // aguarda 1000 milissegundos
}
```

A maioria das placas Arduino possuem um pino conectado a um LED on-board através de um resistor. Na maioria das placas, esse LED é conectado ao pino digital 13. A constante *LED\_BUILTIN* é o número do pino ao qual o LED on-board está conectado. O exemplo abaixo mostra como é possível usar o LED presente na placa.

```

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

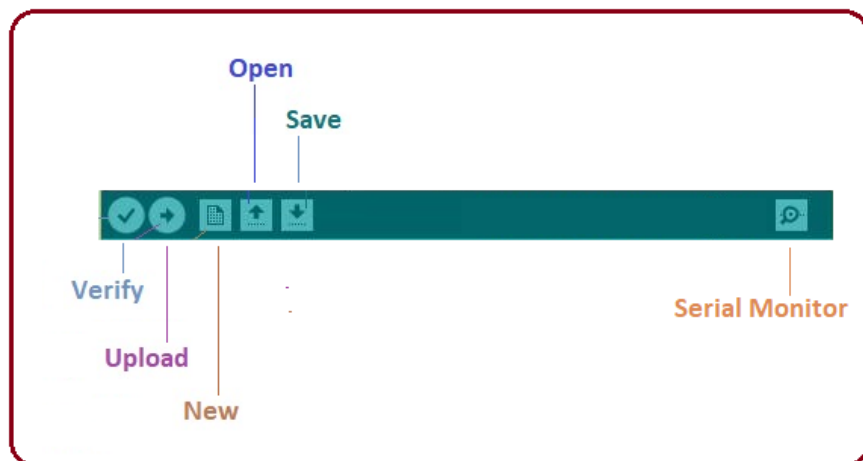
void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}

```

O código acima é chamado “Blink” e é frequentemente utilizado para testar o Arduino.

## 2.2. Arduino IDE

Para subirmos o código em C para o microcontrolador utilizamos uma interface (IDE). A interface mais comum para o Arduino é a Arduino IDE.

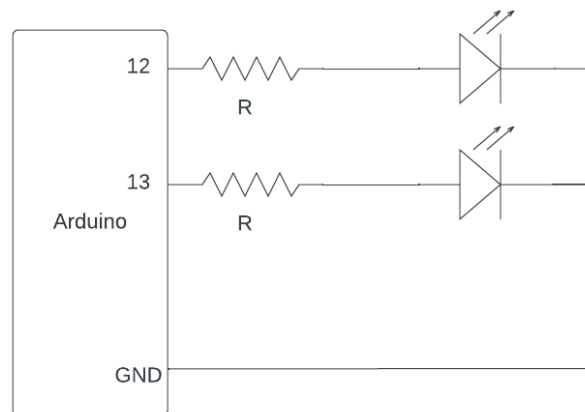


**Figura 3: Interface do Arduino IDE**

De forma resumida, o botão “**Verify**” irá checar se o código possui algum erro na compilação. Caso não apresente erros, o botão “**Upload**” carrega o código no microcontrolador conectado.

### 3. Parte prática


Considerando o circuito da Figura 4 com 2 LEDs nas saídas 12 e 13 do Arduino, realize os códigos com requisitos abaixo:



**Figura 4: Circuito para a montagem prática**

- A. Elabore um código para fazer com que o LED vermelho e o verde alternem entre si a cada 1 segundo (1 segundo com o vermelho ligado e depois 1 segundo o verde ligado). Enquanto um estiver ligado, o outro estará desligado.
- B. Elabore um código que irá acender os LEDs na seguinte ordem:
- LED vermelho liga por 100 ms
  - Todos LEDs desligam por 400 ms;
  - LED vermelho liga por 100 ms;
  - Todos LEDs desligam por 400 ms;
  - LED verde liga por 500 ms;
  - Todos LEDs desligam por 500 ms;
  - Repetir
- C. Elabore um código em que o LEDs irão piscar (ligar e desligar) na seguinte ordem:
- Piscar LED vermelho 1 vezes durante 1 segundo (500 ms em cada estado)
  - Piscar LED vermelho 2 vezes durante 1 segundo (250 ms em cada estado)
  - Piscar LED vermelho 4 vezes durante 1 segundos (125 ms em cada estado)
  - Piscar LED vermelho 8 vezes durante 1 segundos (62,52 ms em cada estado)
  - Ligar LED verde por 3 segundos
  - Desligar todos os LEDs por 1 segundo
  - Repetir

- D. **Desafio (opcional):** Elaborar um código que faz com que os LEDs verde e vermelho alternem (como no exercício A), porém o tempo de alternância irá mudar a cada ciclo. Considere que 1 ciclo significa o LED vermelho acender uma vez e depois o verde acender uma vez. Dessa forma o tempo alternância significa o tempo de 1 ciclo, por exemplo: tempo de alternância de 250 ms significa que o LED vermelho ficará aceso por 250 ms e depois o verde ficará aceso por 500 ms. O tempo seguirá o padrão “ $250 - k \cdot 25$ ”, ou seja, diminuir 25 ms a cada ciclo começando em 250 ms. Ao chegar em 25 ms, seguirá o padrão “ $25 + 25$ ”, ou seja, aumentará em passos de 25 ms começando em 25 ms até chegar em 250 ms novamente. Ou seja, o padrão abaixo será seguido do ciclo 1 ao 9, e depois irá retornar, do 9 ao 1

Ciclo	Tempo de Alternância	Sequência
1	250 ms	
2	225 ms	
3	200 ms	
4	175 ms	
5	150 ms	
6	125 ms	
7	100 ms	
8	75 ms	
9	50 ms	
10	25 ms	

**\*Dica:** os códigos C e D podem ser feito mais facilmente com laços de repetição.