

# Curso de Arduino

---

## Sessão 1

# Sobre o Curso

4  
Sessões

Conceitos Introdutórios  
Sistema de leitor de “música”  
Sistema de cancelas I  
Sistema de cancelas II

Introdução

Abordagem *top-down*

Parte  
Prática

Guiões Práticos

Programação em C

Montagem de Circuitos

# Plano da Sessão

LED a piscar

Interação LEDs / botões

Voltímetro

Sistema de controlo de tensão

# Microcontroladores





# Microcontroladores

Versatilidade

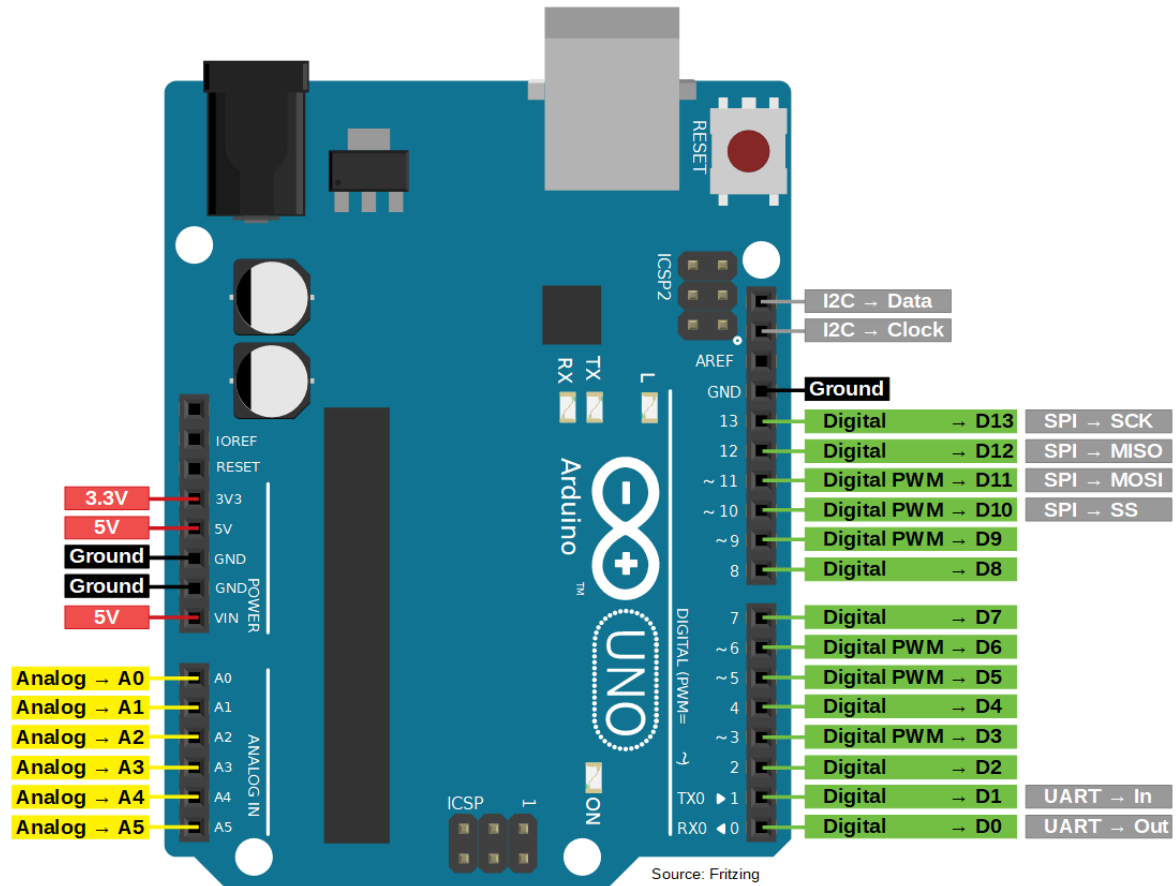
Flexibilidade



# Arduino

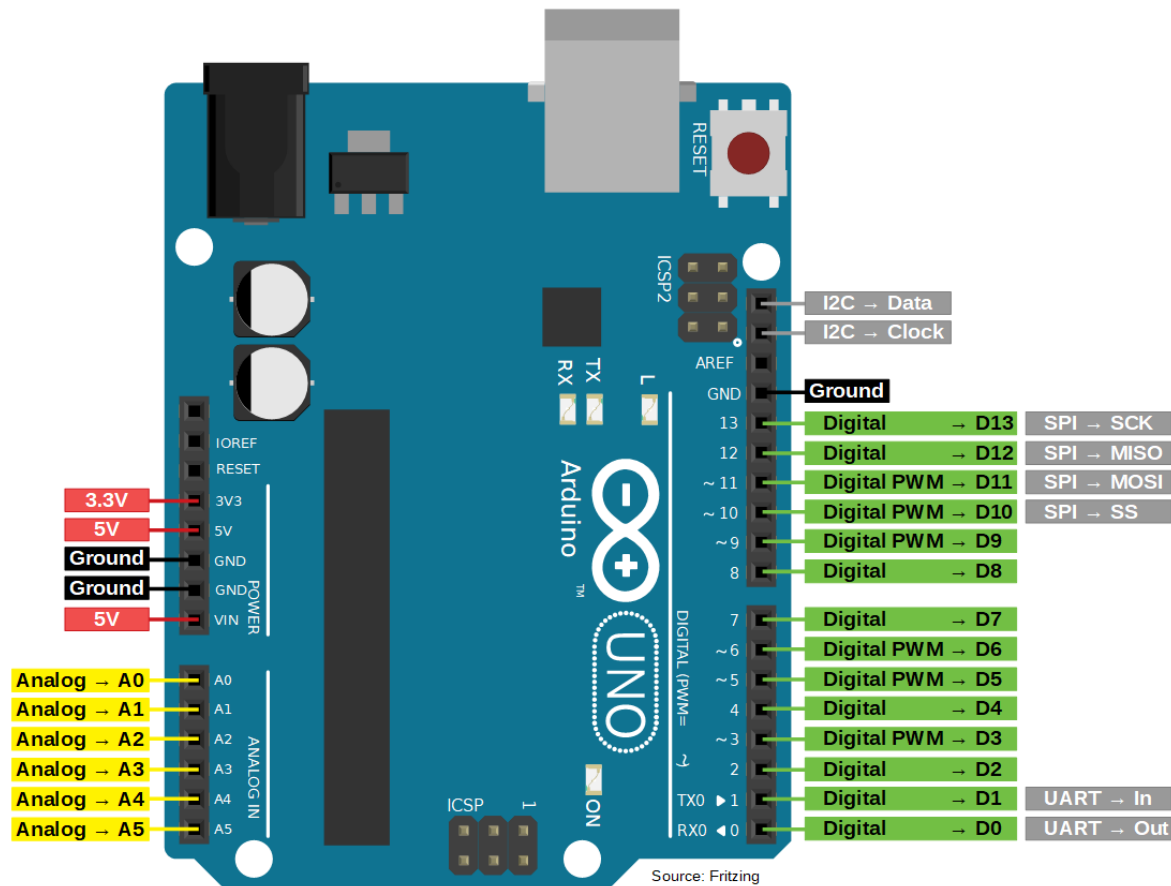


# Arduino





# Arduino



Como é que os portos “sabem” se são entradas ou saídas?

`pinMode()`

Como posso ler/escrever nos portos?

`digitalRead()`

`digitalWrite()`

`analogRead()`

`analogWrite()`





Digital

# LED a Piscar

```
void setup() {  
  
}  
  
void loop() {  
  
  
}
```

|       |  |                                      |
|-------|--|--------------------------------------|
| setup |  | É executada apenas <b>uma vez</b>    |
| loop  |  | É executada em <b>ciclo infinito</b> |

# LED a Piscar

```
void setup() {  
    pinMode(LED, OUTPUT);  
}  
  
void loop() {  
  
}
```

Utilização:

pinMode(PIN, MODE);

OUTPUT

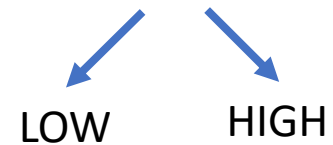
INPUT

# LED a Piscar

```
void setup() {  
    pinMode(LED, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(LED, HIGH);  
    delay(1000);  
    digitalWrite(LED, LOW);  
    delay(1000);  
}
```

Utilização:

digitalWrite(PIN, VALUE);



delay(1000);



Espera 1000 ms, até executar a próxima instrução

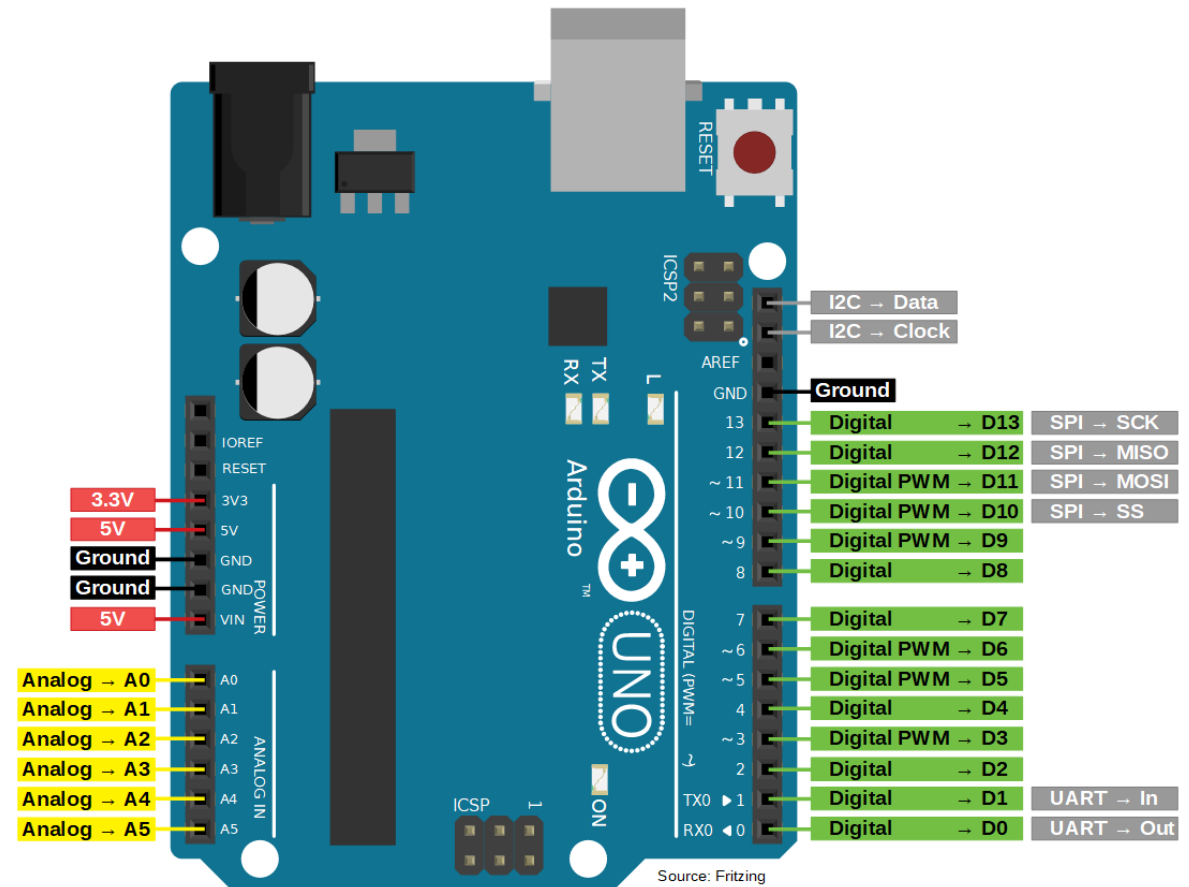


# LED a Piscar

```
#define LED 13

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

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



# LED a Piscar

```
#define LED 13

// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 13 (LED) as an output.
    pinMode(LED, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(LED, HIGH);    // turn the LED on (HIGH is the voltage level)
    delay(1000);                // wait for a second
    digitalWrite(LED, LOW);     // turn the LED off by making the voltage LOW
    delay(1000);                // wait for a second
}
```

# Interação botão/LED

```
#define LED 13  
#define BUTTON 6
```

```
int value;
```

Declaração de uma variável

```
void setup() {  
  pinMode(LED, OUTPUT);  
  pinMode(BUTTON, INPUT);  
}
```

```
void loop() {  
  value = digitalRead(BUTTON);
```

Utilização:

a = digitalRead(PIN);

LOW HIGH

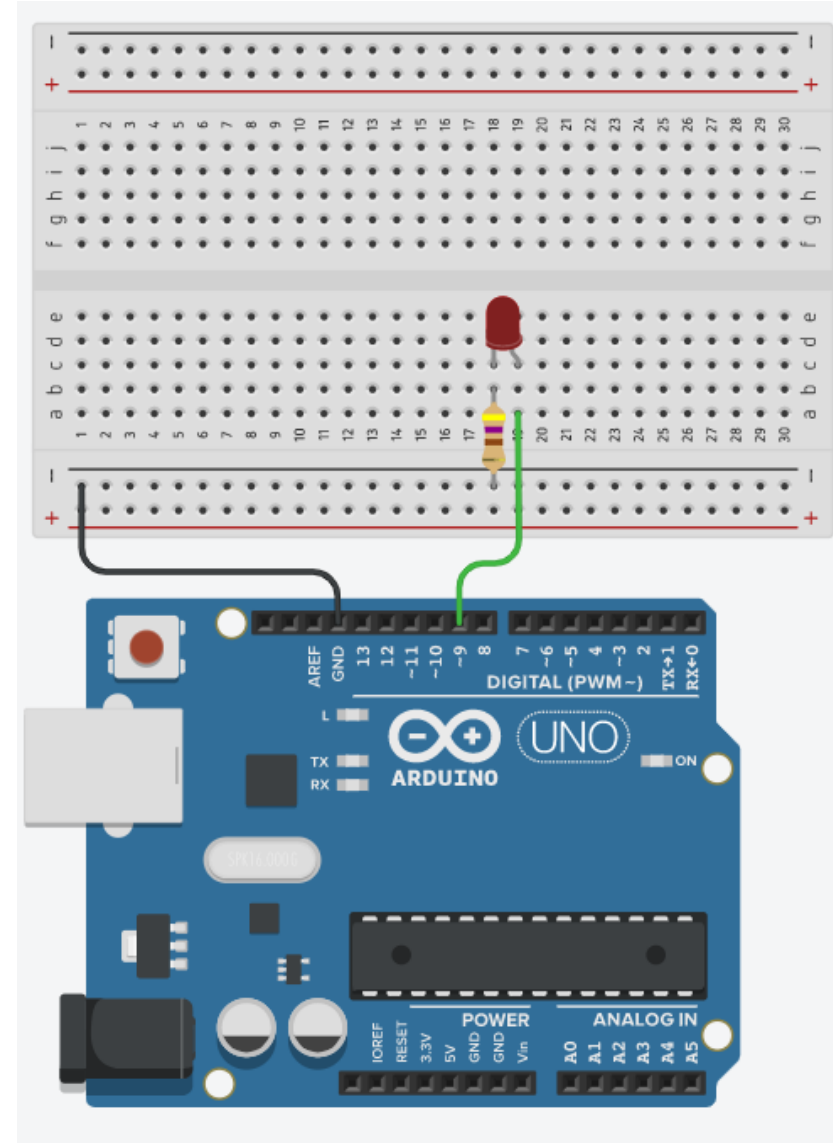
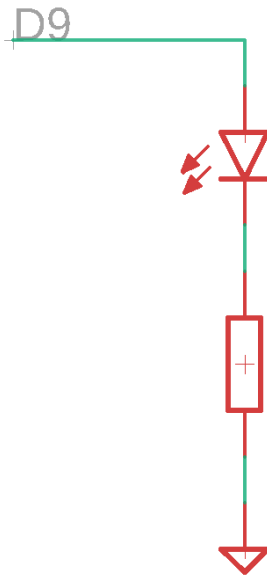
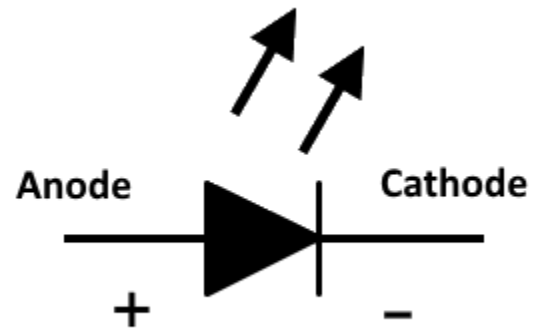
```
  if(value == HIGH) {  
    digitalWrite(LED, HIGH);  
  }  
  else {  
    digitalWrite(LED, LOW);  
  }  
}
```

```
if(condição) {  
  // código  
}  
else if(condição) {  
  // código  
}  
...  
else {  
  // código  
}
```

O que acontece se o *else* for retirado?

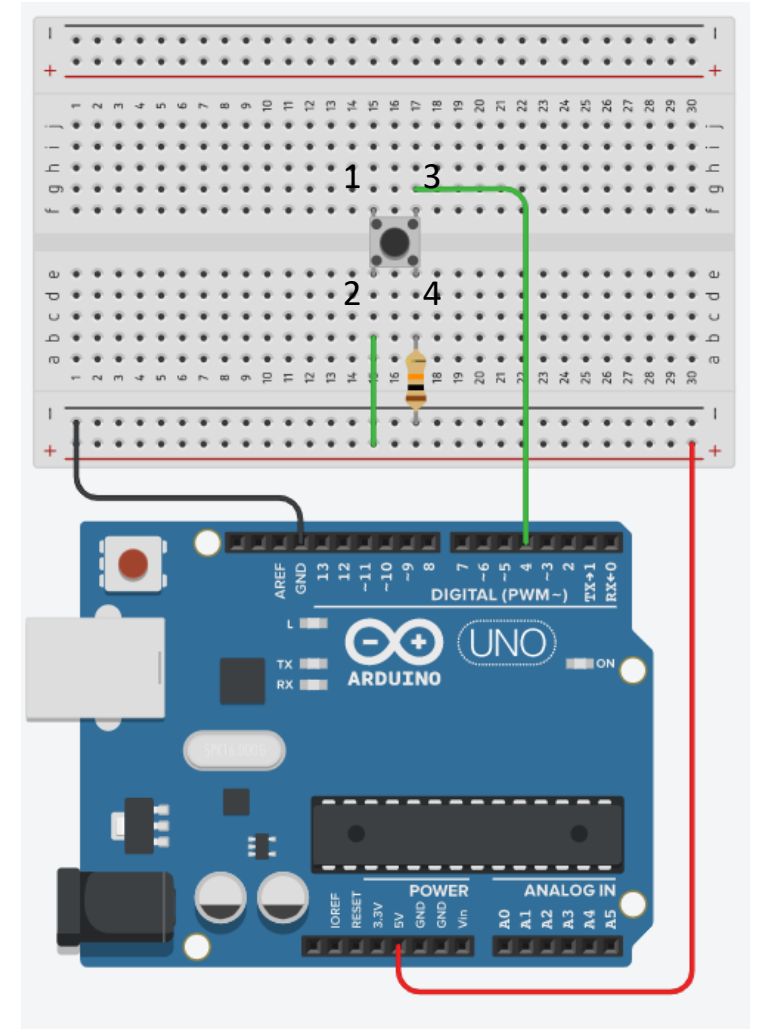
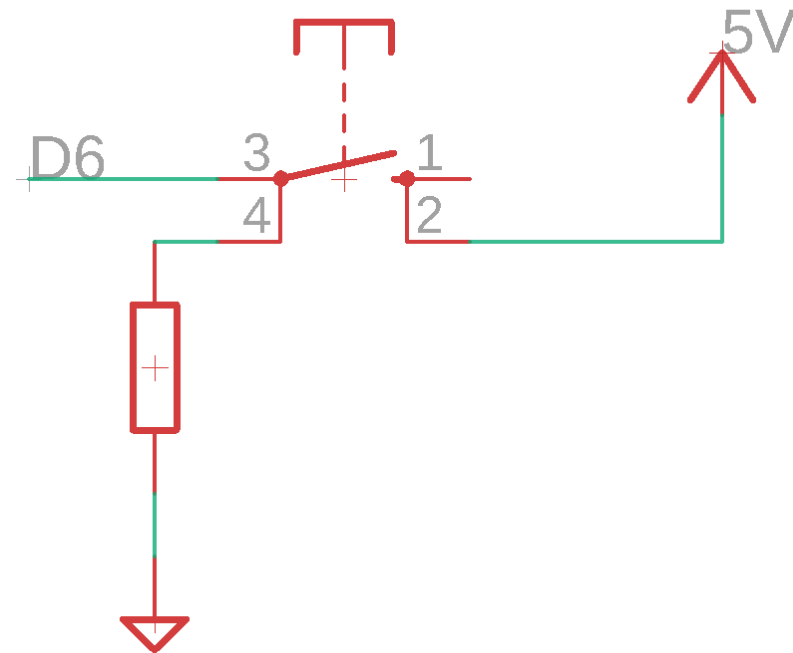
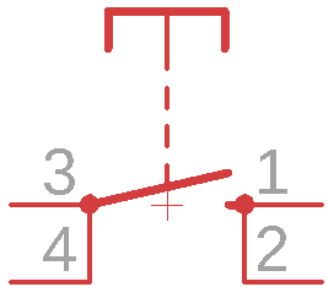
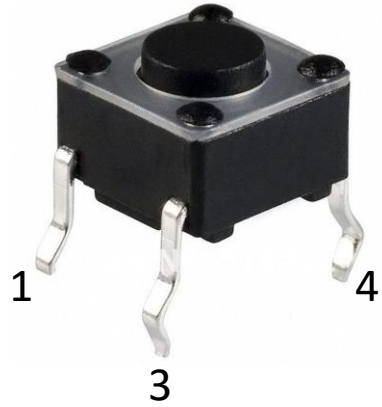
Os portos têm MEMÓRIA!

# LEDs





# Botões



# Funções Essenciais

✓ pinMode(**pin**, **value**);

✓ delay(**value**);

✓ digitalRead(**pin**);

✓ digitalWrite(**pin**, **value**);

retorna um **valor inteiro**  
0 (LOW) ou 1 (HIGH)

analogRead(**pin**);

analogWrite(**pin**, **value**);

retorna um **valor inteiro**  
[0, 1023]

The background of the image features a complex, light blue circuit board pattern on a darker blue background. The pattern consists of numerous parallel lines and small circular components, resembling a printed circuit board (PCB) layout. The lines are more prominent in the upper half and fade into a lighter blue background in the lower half.

# Analógico

# Leitura Analógica

```
#define POT_PIN      A7

int analogValue;
double voltage;

void setup() {
    pinMode(POT_PIN, INPUT);
}

void loop() {
    analogValue = analogRead(POT_PIN);
    voltage = analogValue/204.6;

    delay(200);
}
```

Utilização:

a = analogRead(PIN);



[0, 1023]

$$\begin{array}{lcl} 5 & - & 1023 \\ x & - & 73 \end{array} \rightarrow x = 73 \times \frac{5}{1023}$$
$$x = 73/204.6$$
$$x = 0,36 \text{ V}$$

Como podemos confirmar o valor armazenado em **voltage**?

Interface Série



# Interface Série

- Utilizada para imprimir resultados no monitor do PC

```
Serial.begin(speed);
```

```
Serial.print(value);
```

```
Serial.println(value);
```

# Leitura Analógica

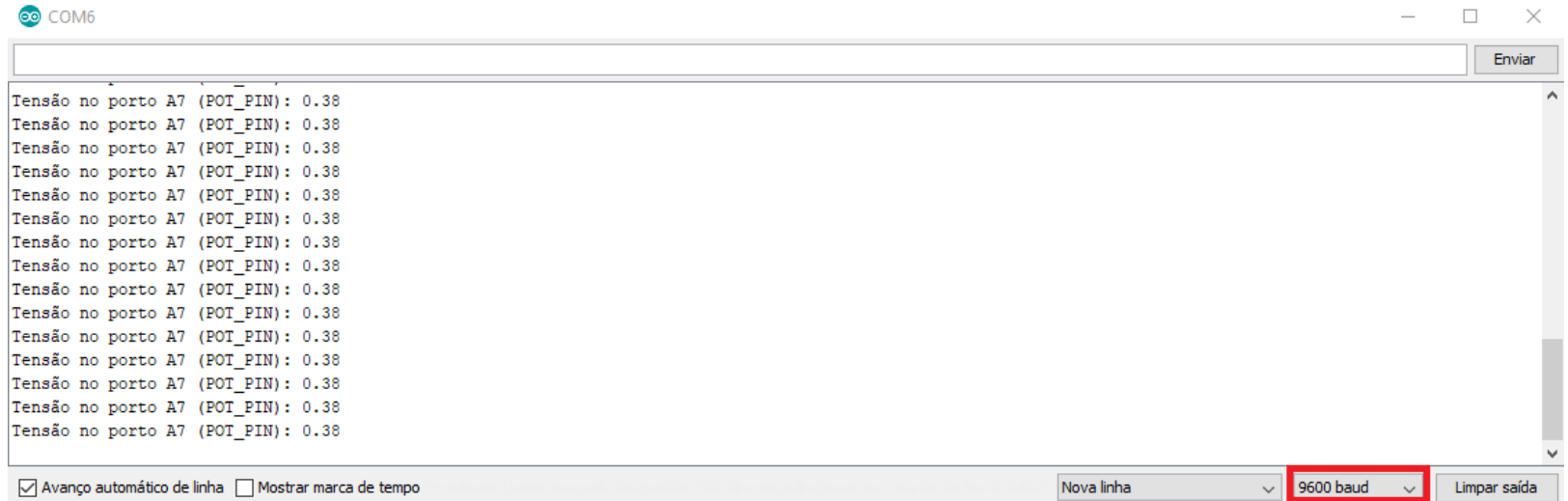
```
#define POT_PIN      A7
```

```
int analogValue;
```

```
double voltage;
```

```
void setup() {  
    pinMode(POT_PIN, INPUT);  
    Serial.begin(9600);  
}
```

```
void loop() {  
    analogValue = analogRead(POT_PIN);  
    voltage = analogValue/204.6;  
    Serial.print("Tensão no porto A7 (POT_PIN): ");  
    Serial.println(voltage);  
    delay(200);  
}
```



Não é uma boa prática!

Enviar demasiada informação  
através da interface série pode  
**tornar o código pouco eficiente!**

# Escrita Analógica

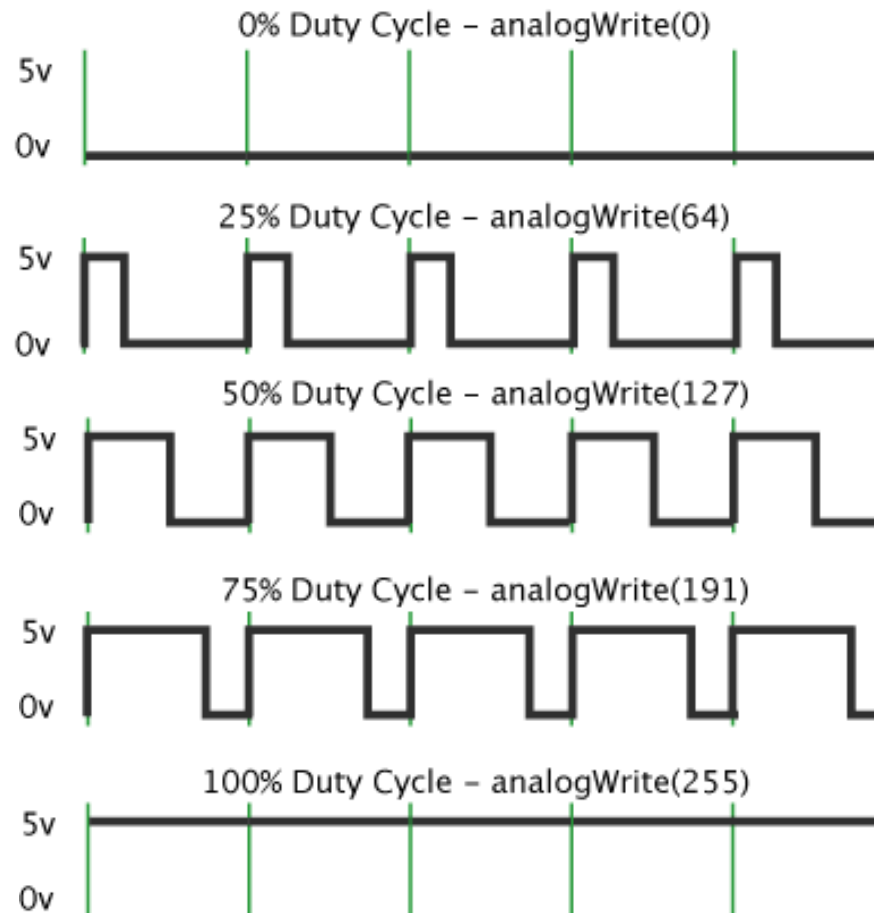
ADC ✓

DAC ✗

```
analogWrite(pin, value);  
//value está entre [0, 255]
```

Como funciona o *analogWrite* ?

→ PWM



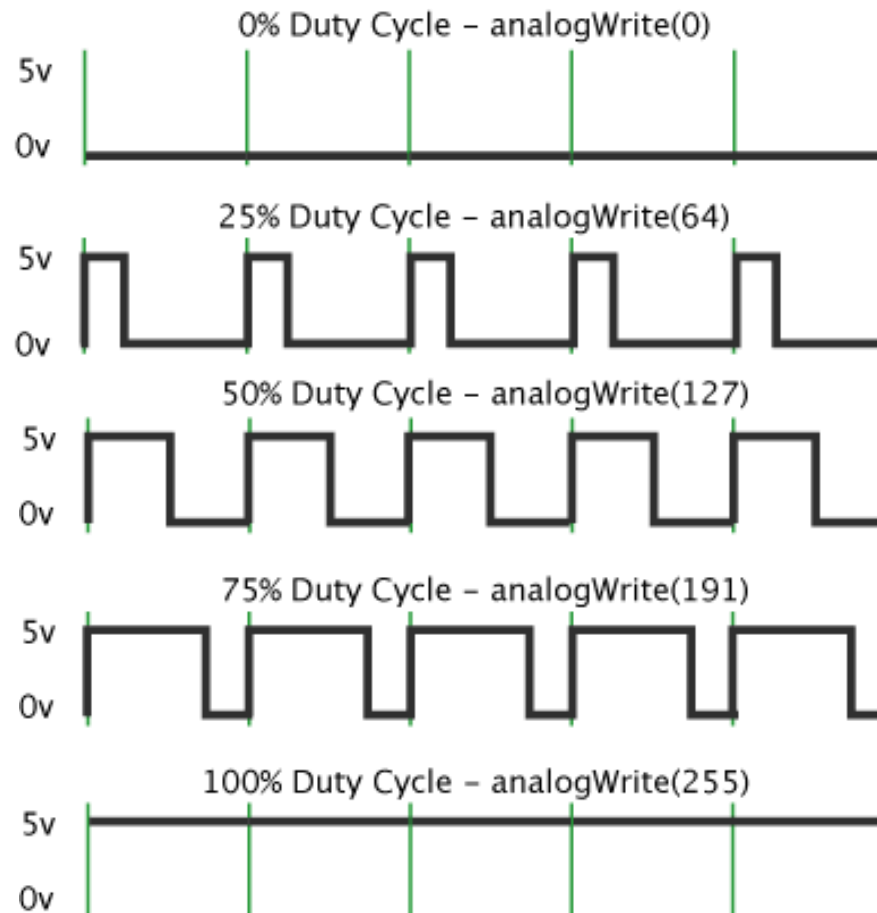
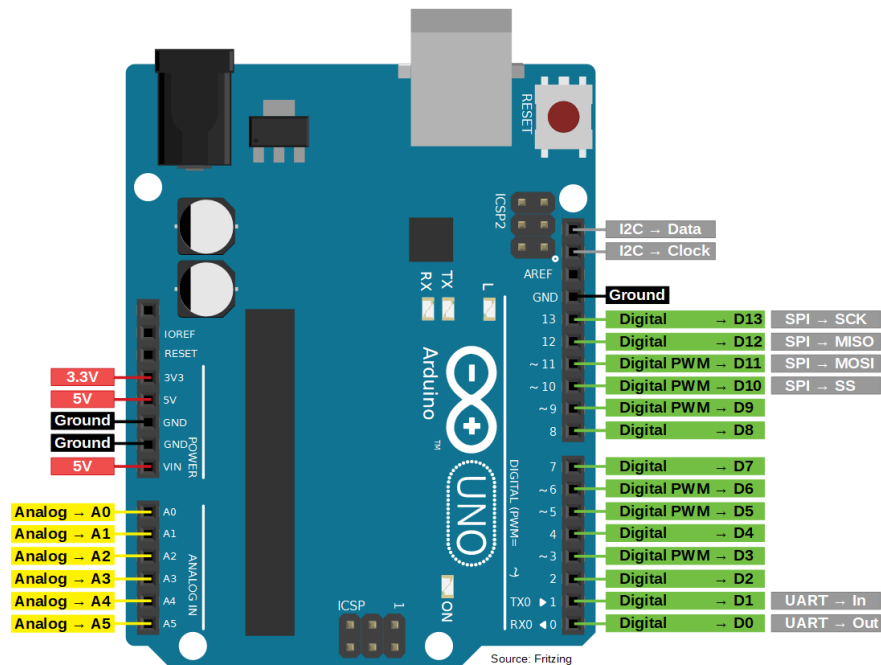
$$0,25 \times 256 = 64$$

Duty Cycle 25%:  
 $0,25 \times 5 + 0,75 \times 0 = 1,25V$

Duty Cycle 75%:  
 $0,75 \times 5 + 0,25 \times 0 = 3,75V$

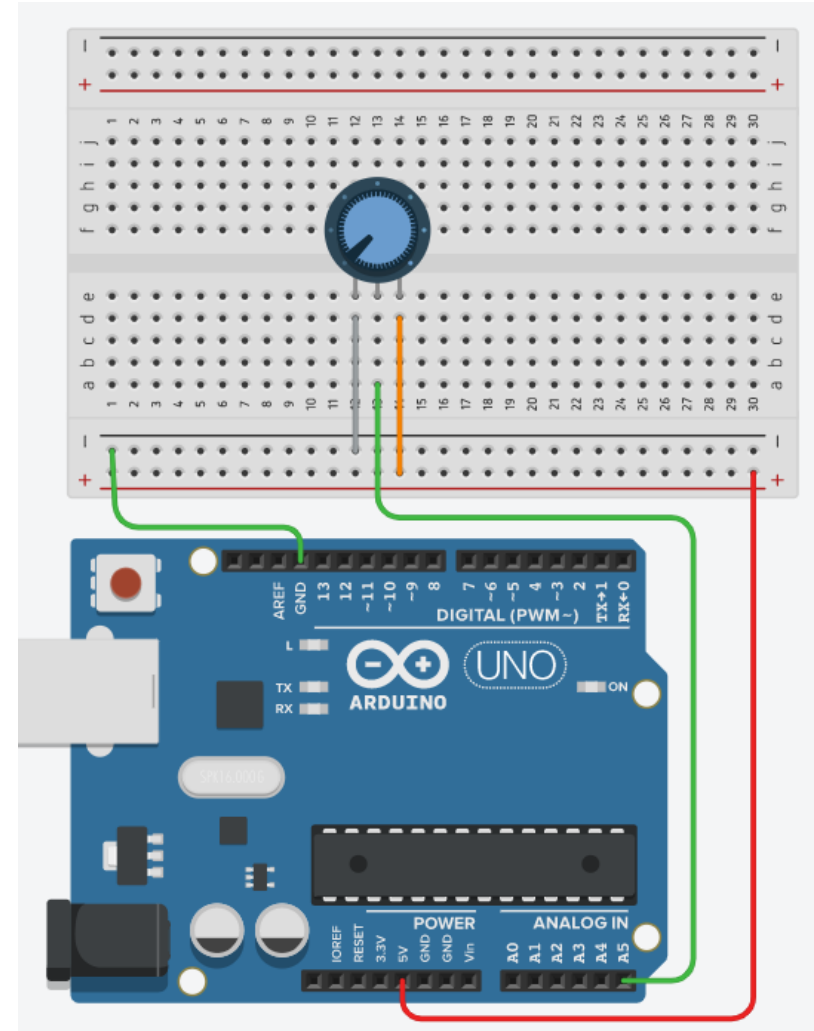
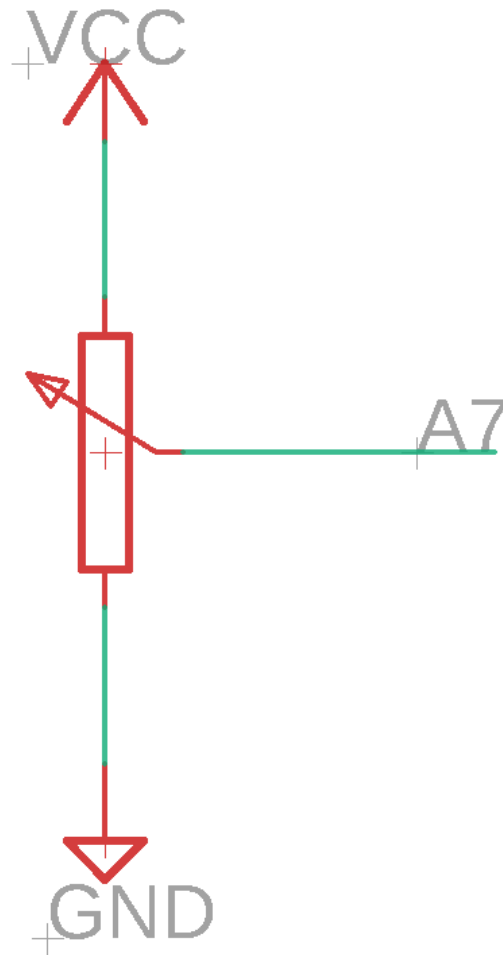
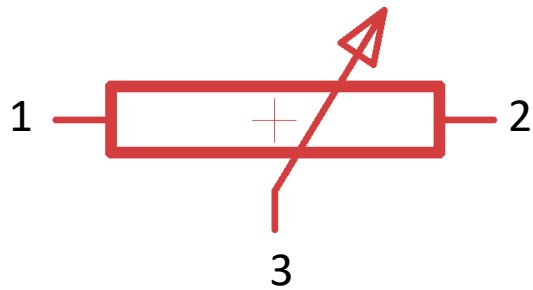
# Escrita Analógica

Como funciona o *analogWrite* ?

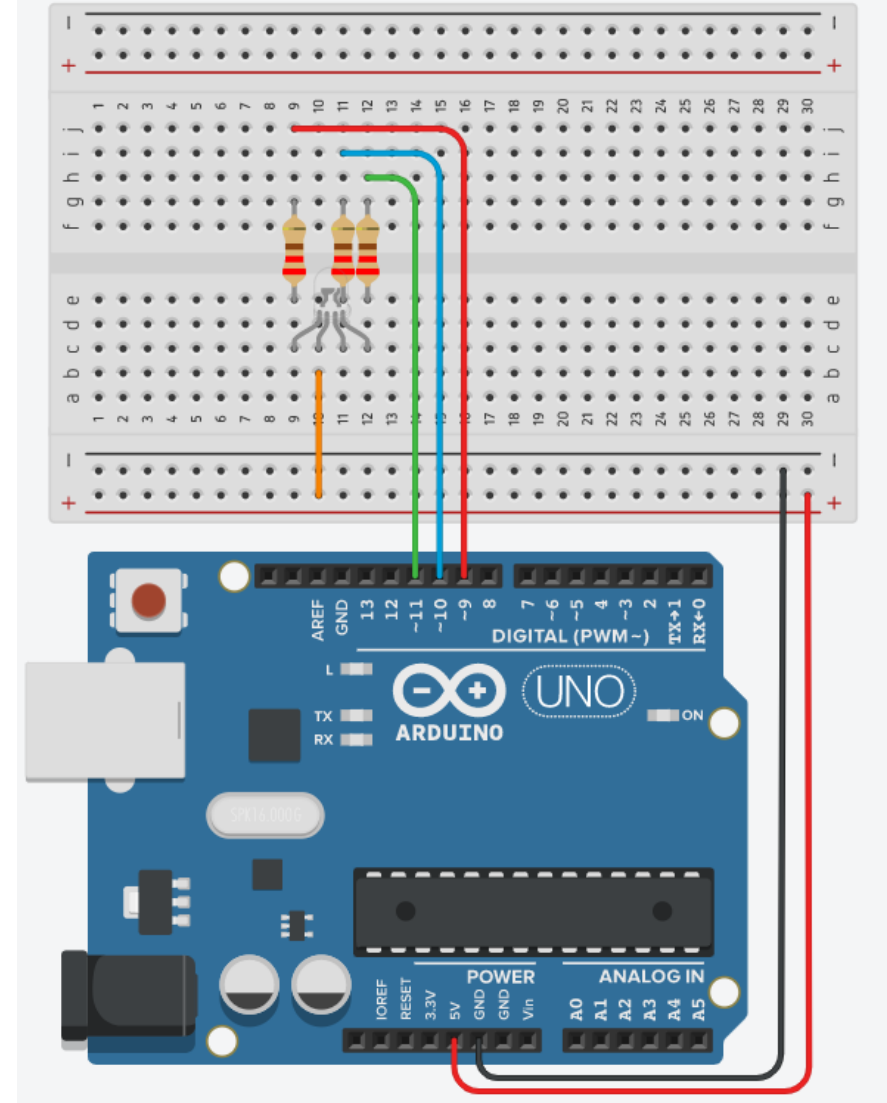
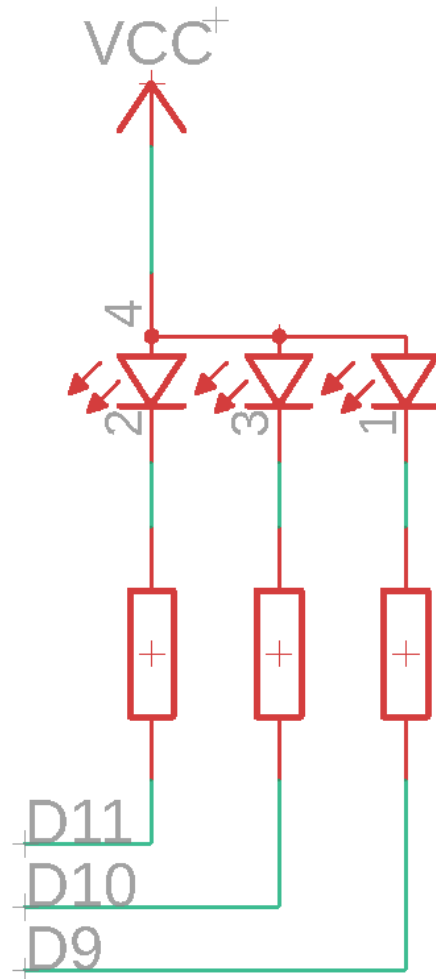
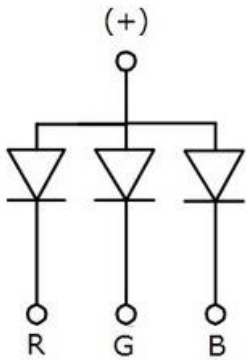
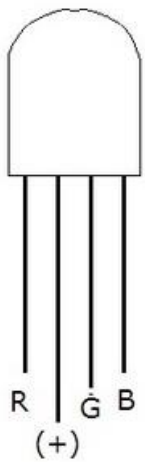




# Potenciómetro



# LED RGB



## ❑ Configuração:

```
pinMode(pin, value);
```

## ❑ Útil:

```
delay(value);  
Serial.begin(speed);  
Serial.print(value);  
Serial.println(value);
```

## ❑ Leitura/Escreita:

```
digitalRead(pin);  
digitalWrite(pin, value);  
analogRead(pin);  
analogWrite(pin, value);
```

```
if(condição) {  
    // código  
}  
else if(condição) {  
    // código  
}  
...  
else {  
    // código  
}
```

<https://www.arduino.cc/reference/en/>

