

Trabalhando com Saídas e Entradas Analógicas

No capítulo anterior, aprendemos a usar as entradas e saídas digitais. Agora, neste capítulo, vamos nos aprofundar nas entradas e saídas analógicas, que permitem um controle mais preciso de sensores e atuadores.

- [Conceito de Sinais Analógicos](#)
- [Leitura de Entradas Analógicas com `analogRead\(\)`](#)
- [Função `analogWrite\(\)` e PWM \(Modulação por Largura de Pulso\)](#)
- [Sensores Analógicos](#)

Conceito de Sinais Analógicos

Conceito de Sinais Analógicos

Os **sinais digitais** possuem apenas dois estados: **HIGH** (5V) ou **LOW** (0V). Já os **sinais analógicos** variam continuamente em uma faixa de valores. O Arduino pode:

- **Ler** valores analógicos de sensores, como potenciômetros e sensores de luz.
- **Gerar** sinais que simulam saídas analógicas, usando a técnica de **modulação por largura de pulso (PWM)**.

Leitura de Entradas Analógicas com `analogRead()`

Leitura de Entradas Analógicas com `analogRead()`

Os pinos analógicos do Arduino permitem a leitura de valores que variam de 0 a 1023, o que representa uma faixa de 0V a 5V (em placas como o Arduino Uno).

Exemplo: Conectar um potenciômetro (ou outro sensor) a um pino analógico e ler seu valor.

- O potenciômetro é um resistor variável, que ajusta a tensão em seu terminal de saída.

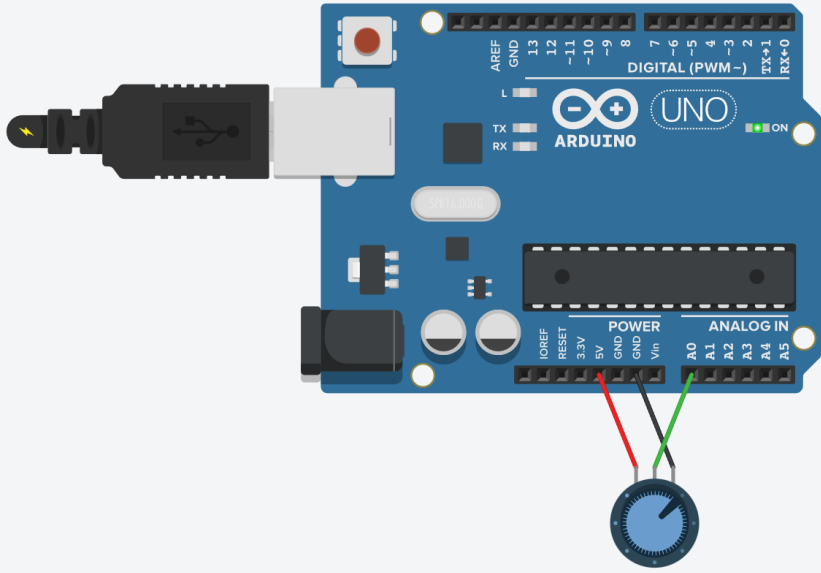
Código:

```
int potPin = A0; // Pino analógico A0
int sensorValue = 0;

void setup() {
  Serial.begin(9600); // Inicia a comunicação serial
}

void loop() {
  sensorValue = analogRead(potPin); // Lê o valor do potenciômetro
  Serial.println(sensorValue); // Exibe o valor lido no monitor serial
  delay(100); // Atraso de 100 ms
}
```

No código acima, o valor lido do potenciômetro será exibido no **Monitor Serial**, variando de 0 a 1023 conforme o ajuste do potenciômetro.



Função analogWrite() e PWM (Modulação por Largura de Pulso)

Função `analogWrite()` e PWM (Modulação por Largura de Pulso)

O Arduino não gera sinais analógicos verdadeiros, mas pode simular uma saída analógica usando **PWM**. Essa técnica gera um sinal digital que alterna rapidamente entre HIGH e LOW, variando a largura dos pulsos. O valor médio resultante pode ser percebido como um valor analógico pelos dispositivos controlados.

A função `analogWrite()` é usada para gerar uma saída PWM nos pinos marcados com o símbolo “~”. Os valores PWM variam de 0 a 255:

- **0**: Sinal equivalente a 0V.
- **255**: Sinal equivalente a 5V.

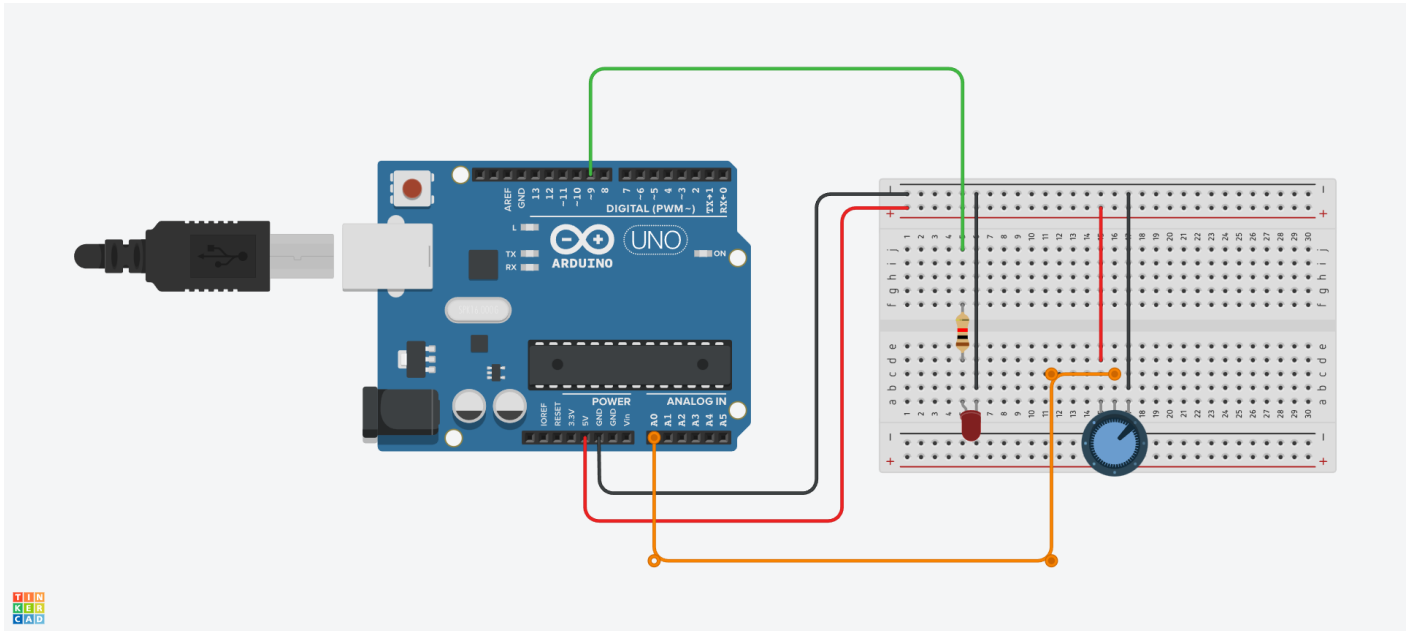
Exemplo: Controlar o brilho de um LED usando a função `analogWrite()` e um potenciômetro para ajustar a intensidade.

```
int potPin = A0; // Pino do potenciômetro
int ledPin = 9;  // Pino PWM do LED
int val = 0;     // Armazenar o valor lido

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

void loop() {
  val = analogRead(potPin); // Lê o valor do potenciômetro (0 a 1023)
  val = map(val, 0, 1023, 0, 255); // Converte o valor para a faixa de 0 a 255
  analogWrite(ledPin, val); // Ajusta o brilho do LED com base no valor do potenciômetro
}
```

Função `map()` : Converte valores de uma faixa para outra. No exemplo, converte o valor lido do potenciômetro (0-1023) para a faixa PWM (0-255).



Sensores Analógicos

Sensores Analógicos

Os sensores analógicos fornecem valores variáveis que podem ser lidos diretamente pelos pinos analógicos do Arduino. Alguns exemplos de sensores analógicos comuns incluem:

- **Sensor de temperatura LM35:** Fornece uma leitura analógica proporcional à temperatura.
- **Sensor de luz LDR:** Um resistor dependente de luz que altera sua resistência com base na intensidade luminosa.

Exemplo: Ler a temperatura com o sensor LM35

```
int tempPin = A0; // Pino do sensor LM35
float temperature = 0.0;

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

void loop() {
  int val = analogRead(tempPin); // Lê o valor analógico do LM35
  temperature = (val * (5.0 / 1023.0)) * 100; // Converte o valor para °C
  Serial.print("Temperatura: ");
  Serial.print(temperature);
  Serial.println(" C");
  delay(1000); // Aguarda 1 segundo
}
```

No exemplo acima, o valor lido do sensor LM35 é convertido para graus Celsius e exibido no Monitor Serial.

