

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.

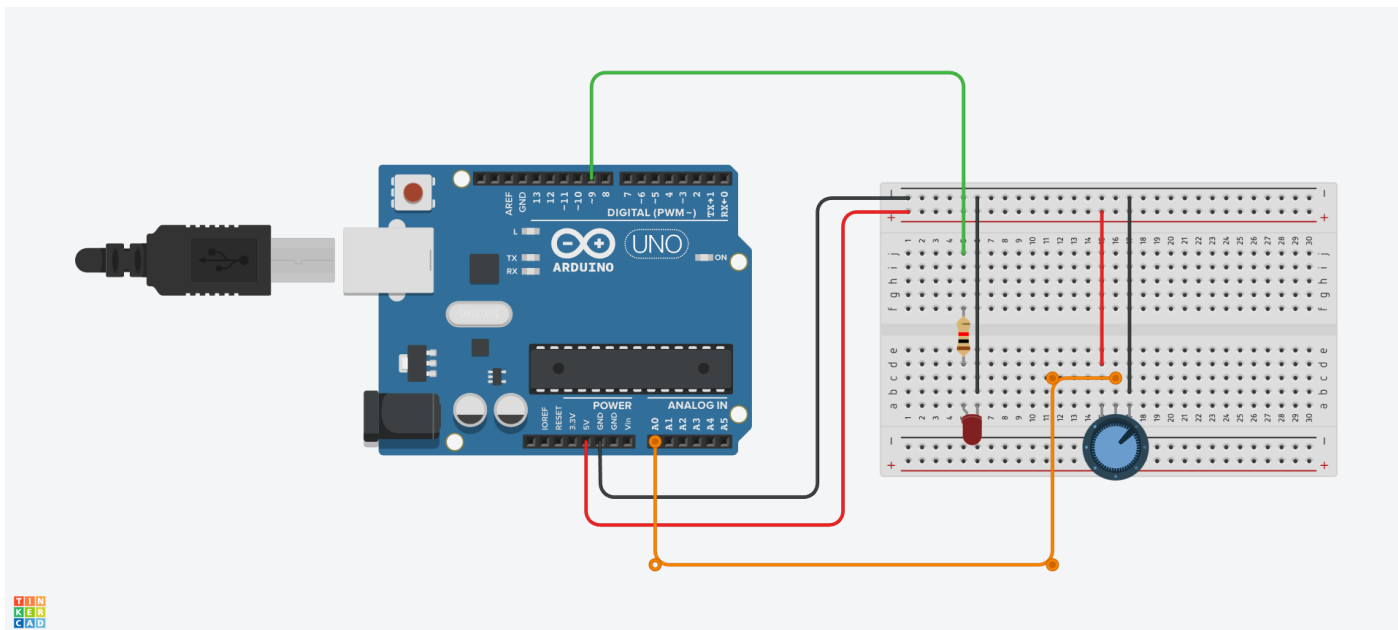
Exemplo: Controlar o brilho de um LED usando a função `analogWrite()` e um potenciômetro para ajustar a intensidade.

```
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).



Revision #3

Created 21 September 2024 14:30:09 by Ricardo Veiga

Updated 21 September 2024 16:17:59 by Ricardo Veiga