Ministério da Educação
Universidade Tecnológica Federal do Paraná
Coordenação de Engenharia de Computação
Materiais e Equipamentos Elétricos
Campus Apucarana

Laboratório de Materiais e Equipamentos Elétricos

Roteiro 10: Arduino - Saída Analógica

1. Objetivos

- Analisar e compreender as portas (pinos) de um Arduino.
- Configurar a saída analógica do Arduino para PWM

2. Material utilizado

- 1 LED vermelho
- 1 LED RGB
- 1 Resistor de 150 Ohms
- 2 Resistores de 100 Ohms
- 1 Resistor de 10k Ohms
- Fotoresistor LDR
- Protoboard
- Arduino UNO
- Osciloscópio

2.1. Saídas Analógicas (Arduino)

No campo da eletrônica, a saída analógica é uma das funcionalidades essenciais para o controle preciso de dispositivos externos. O Arduino UNO oferece recursos de saída analógica que permitem gerar sinais contínuos em uma variedade

de aplicações. Neste roteiro experimental, vamos explorar como funciona a saída analógica do Arduino UNO, bem como entender o conceito de Modulação por Largura de Pulso (PWM) para controle de dispositivos analógicos.

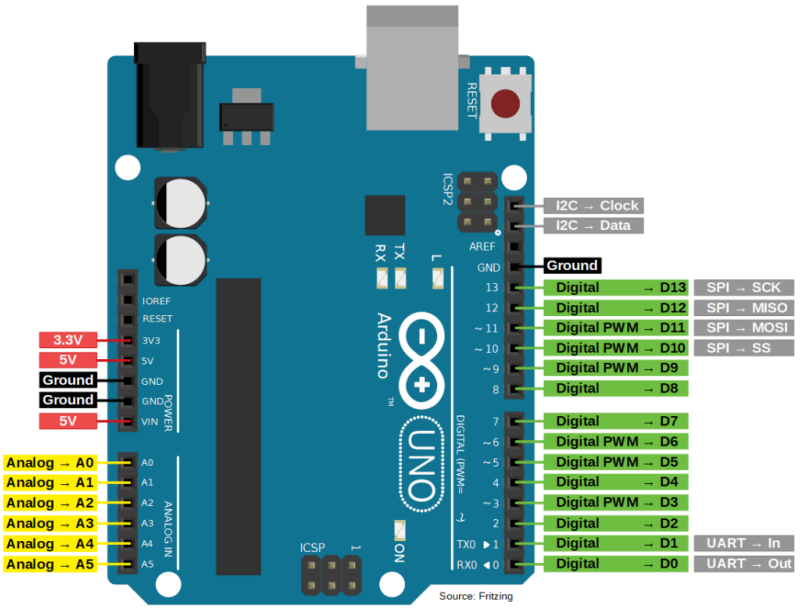

Figura 1: pinos disponíveis no Arduino UNO

No Arduino as saídas que podem ser configuradas como **saídas analógicas** são as mesmas das saídas digitais, porém com o **símbolo ~**, que na Figura estão sinalizadas pelo termo **PWM**.

2.2. PWM

Diferentemente da saída digital, que possui apenas dois estados (ligado/desligado), a saída analógica pode assumir uma variedade de valores intermediários. Isso é alcançado por meio de um processo chamado Modulação por Largura de Pulso (PWM). O PWM funciona emitindo pulsos elétricos de largura variável, onde a proporção entre o tempo em que o pulso está "ligado" e "desligado" determina o valor analógico resultante. Como a frequência de “liga” e “desliga” é alta, o que vemos como resultado é a potência média entregue que é proporcional ao tempo ligado em relação ao tempo desligado. Esse valor é chamado de **Duty Cicle**, ou **Ciclo de Trabalho** ou **Razão Cíclica**.

O PWM é definido pelo comando “AnalogWrite” que recebe um valor inteiro

de 8 bits (**de 0 a 255**) que irá definir um Duty Cycle de **0 até 100%.**

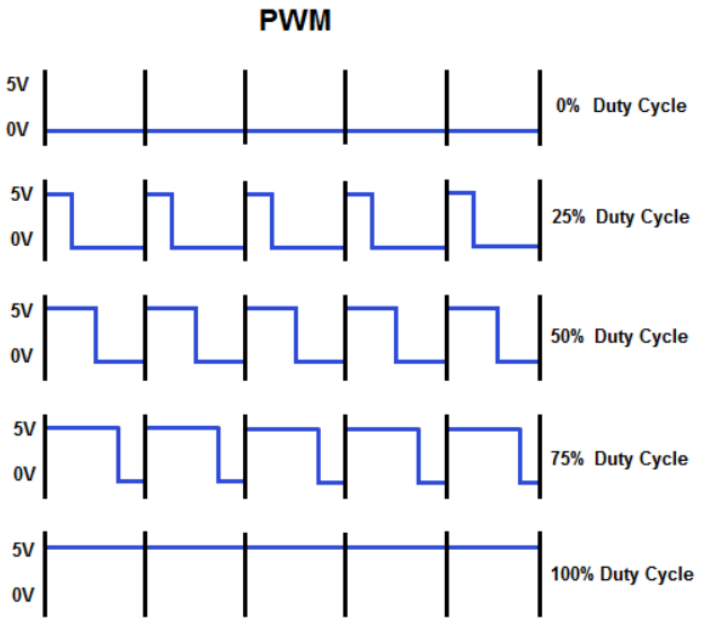

Figura 2: Modulação PWM

O código abaixo mostra um exemplo de como um LED conectado na porta 10 do Arduino pode ser sua intensidade luminosa modificada em 3 valores utilizando PWM.

```
const int ledPin = 10; // Pino do LED

void setup() {
  pinMode(ledPin, OUTPUT); // Define o pino do LED como saída
}

void loop() {
  // Variação do brilho em três valores diferentes

    analogWrite(ledPin, 100); // Define um valor de brilho intermediário
    delay(1000); // Aguarde 1 segundo

    analogWrite(ledPin, 200); // Define um valor de brilho mais alto
    delay(1000); // Aguarde 1 segundo

    analogWrite(ledPin, 50); // Define um valor de brilho mais baixo
    delay(1000); // Aguarde 1 segundo

}
```

2.3. Osciloscópio

Osciloscópio é um instrumento eletrônico amplamente utilizado em laboratórios de eletrônica para medir, visualizar e analisar formas de onda elétricas.

O funcionamento básico de um osciloscópio envolve a captura de um sinal elétrico através de uma sonda conectada ao circuito em análise. O sinal é então amplificado e convertido em uma representação visual na tela do osciloscópio. A tela geralmente é dividida em dois eixos: o eixo vertical, que representa a amplitude do sinal, e o eixo horizontal, que representa o tempo.

Para usar um osciloscópio, siga as etapas a seguir:

1. Conexão da ponteira: Conecte a ponta de prova do osciloscópio à fonte do sinal que deseja medir. Certifique-se de que as conexões estejam corretas e seguras com a garra no GND do circuito.
2. Configuração dos controles: Ajuste os controles do osciloscópio de acordo com as características do sinal que você deseja medir. Isso pode incluir configurações como a escala vertical (amplitude), escala horizontal (tempo), acoplamento (AC ou DC), entre outros.
3. Visualização do sinal: Você poderá ver o sinal na tela do osciloscópio. Ajuste os controles de posição vertical e horizontal, se necessário, para obter a melhor visualização do sinal.

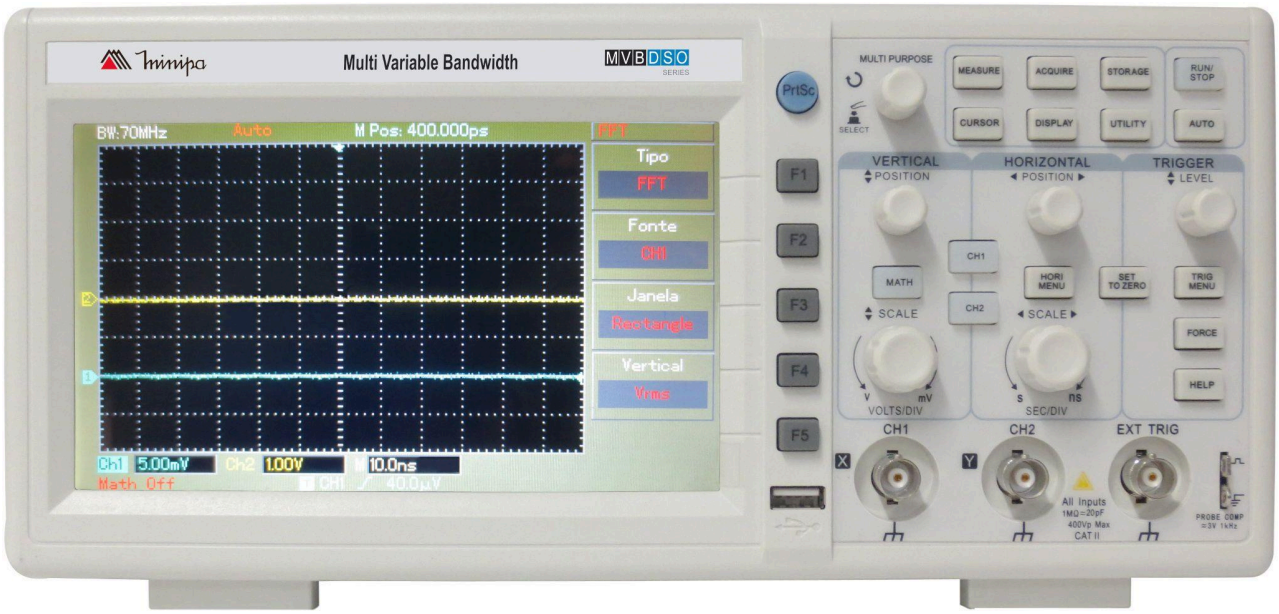

Figura 3: Osciloscópio

3. Parte prática

Considerando os circuitos da Figuras 4, elabore a prática como se pede nos itens abaixo.

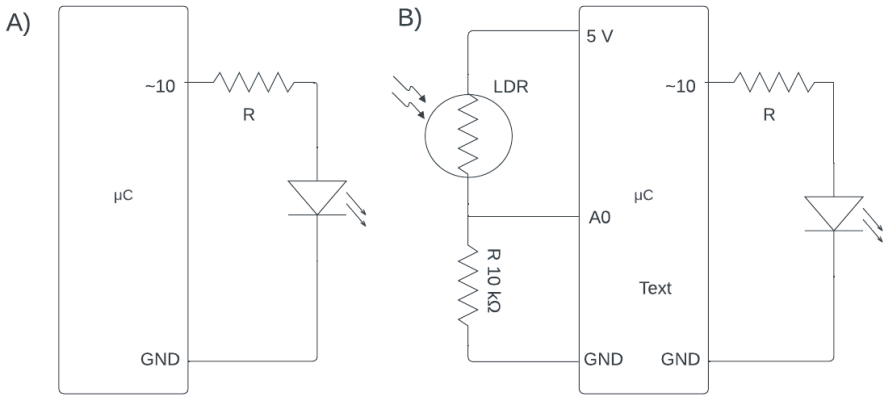

Figura 4: Circuito para a montagem prática (A,B, C)

A. **Fade-in/Fade-out**: Crie um código que irá fazer com o que o LED aumente o brilho gradualmente durante 1 segundo até chegar na intensidade máxima e depois diminua o brilho gradualmente durante 3 segundos até chegar na intensidade mínima. Visualize a curva da saída analógica (PWM) pelo osciloscópio (e tire uma imagem). Qual a frequência do sinal?
B. **LDR controlando LED**: Crie um código que faça com que os valores de intensidade luminosa lida por um LDR controle um LED de forma que quanto menor a intensidade detectada no LDR, maior será o brilho do LED. Visualize os dados do LDR pela serial e faça uma calibração da faixa lida pelo A/D usando a função map lendo valores mínimos e máximos de luminosidade.
C. **Desafio (LED RGB)**: Crie um código para fazer um LED RGB alternar entre 8 cores distintas a cada 1 segundo. Um LED RGB possui três LEDs de cores diferentes (vermelho, verde e azul) em um único encapsulamento e controla as cores resultantes atribuindo um valor de PWM diferente em cada pino.

   Dica 1: Você pode descrever as cores usando um vetor de 3 números, e definir a cor através de um mapa de cores de 8 bits:
   https://www.rapidtables.com/web/color/RGB_Color.html
   Dica 2: Após definir as 8 cores, você pode definir uma matriz de cores, e utilizar um laço para iterar entre as linhas da matriz para mudar as cores dos LEDs.
   Dica 3: Você pode criar uma função que recebe três parâmetros inteiros r,g,b de 0 a 255 e escreve o valor de PWM nos pinos.

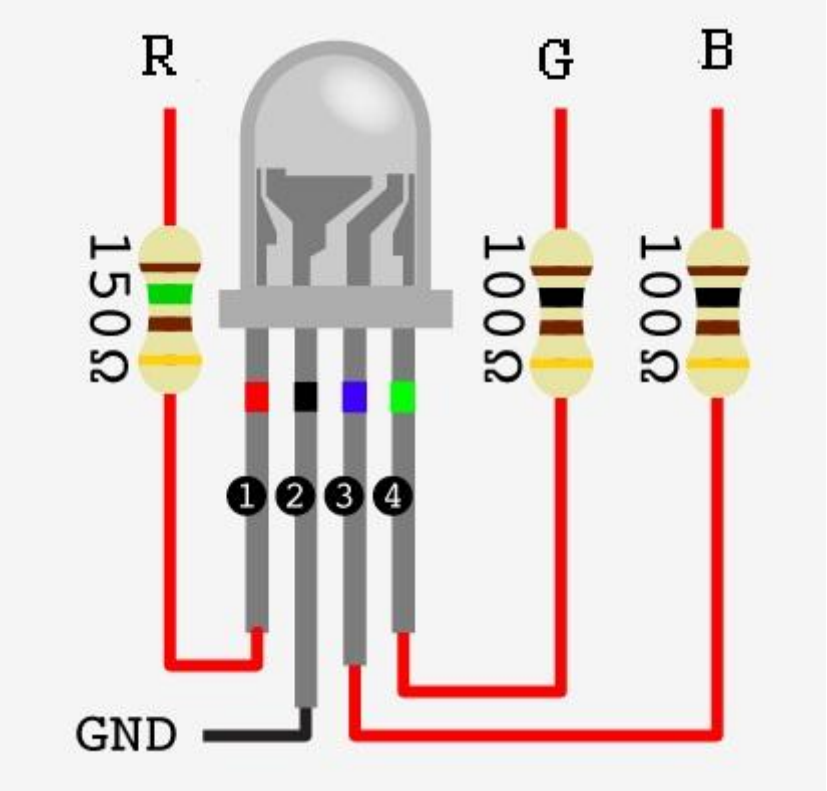

Figura 5: Ligação do LED RGB.