

Professor: Alexandre Moraes Tannus - 2018

Arduino: Entradas e Saídas Analógicas

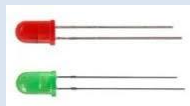
1. OBJETIVOS:

- Conhecer os fundamentos do uso de entradas e saídas analógicas
- Entender o funcionamento da modulação por largura de pulso (PWM – *Pulse Width Modulation*)
- Implementar entradas analógicas e PWM no Arduino.

2. MATERIAIS:



Uma placa Arduino Uno



LEDs



Resistores



Potenciômetro

3. PARTE TEÓRICA

Sinais analógicos podem assumir valores em um determinado intervalo. Dentre os dispositivos que podem ser utilizados, em conjunto com o Arduino, como entradas analógicas estão sensores de luminosidade LDR (*Light Dependent Resistor*), sensores de distância infravermelhos, sensores de movimento PIR (*Passive InfraRed*) e potenciômetros. As saídas analógicas do Arduino são obtidas a partir de variações na saída digital, utilizando modulação por largura de pulso (PWM – *Pulse Width Modulation*).

Para realizar a conversão analógico-digital (A/D) o Arduino utiliza um conversor de 10 bits para as entradas e um de 8 bits para as saídas. Tanto as entradas como as saídas possuem faixa de tensão de 0 a 5V (EVANS; NOBLE; HOCHENBAUM, 2013).

3.1. Pulse Width Modulation (PWM)

A modulação por largura de pulso (PWM – *Pulse Width Modulation*) é uma técnica que permite variar a intensidade da potência fornecida na saída digital. Para a utilização do PWM devem ser utilizadas os pinos do Arduino marcadas com o símbolo ~ ou indicadas com a sigla PWM. Estes pinos possuem conversores A/D de 8 bits, tendo uma faixa de variação entre 0 e 255. No Arduino Uno os pinos 3, 5, 6, 9, 10 e 11 possuem a funcionalidade de PWM. Para utilizá-los como saída analógica a função que deve ser utilizada é

analogWrite(pino, valor)

A Figura 1 mostra o funcionamento do PWM e como o ciclo de funcionamento (*duty cycle*) é alterado para cada valor utilizado. Quanto maior for a duração do pulso mais tempo o LED ficará ligado em um ciclo de *clock*. A alteração dos pulsos é tão rápida que é impossível perceber o piscamento do LED. Para o observador a impressão é que o LED tem mais ou menos brilho (MONK, 2013).

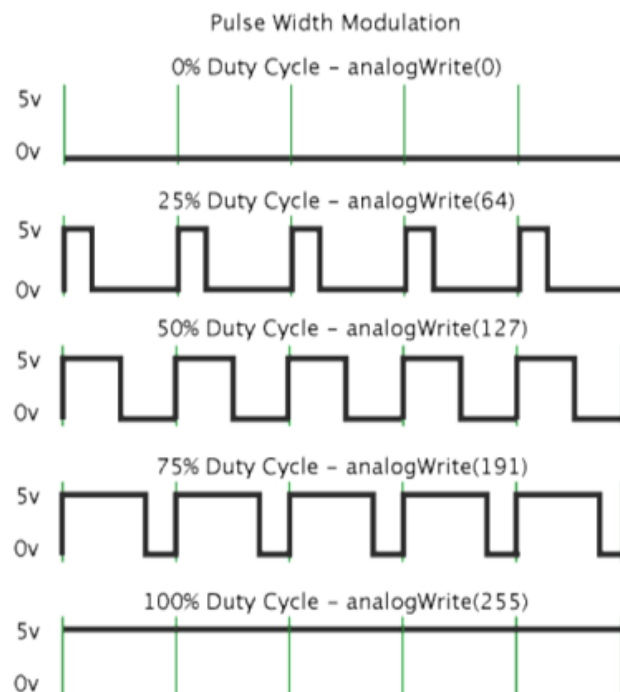


Figura 1 - PWM

3.2. Entradas analógicas

As entradas analógicas são capazes de fornecer um valor de tensão aplicado à elas (BANZI, 2011). Para isso um conversor A/D de 10 bits é utilizado, permitindo que a faixa de tensão de 0 a 5V seja traduzida para valores entre 0 e 1023. Um dos componentes mais simples para a explicação do funcionamento das entradas analógicas é o potenciômetro (Figura 2)



Figura 2 – Potenciômetro

O potenciômetro possui três terminais. Conectando apenas dois deles o dispositivo funcionará como um resistor variável. Caso sejam conectados os três pinos e uma tensão seja aplicada o potenciômetro se torna um divisor de tensão. Nesta aula será utilizada a conexão com três pinos, sendo um conectado ao pino terra (GND), outro à alimentação positiva (5V) e o terminal central ligado a um pino analógico do microcontrolador.

A placa Arduino Uno possui 6 entradas analógicas, nomeadas A0 até A5. Para utilizá-las é necessário utilizar a função `analogRead(pino)`. Os pinos analógicos não necessitam ser configurados como entrada (`INPUT`) ou saída (`OUTPUT`).

4. PARTE PRÁTICA

4.1. Prática 01 - Saídas analógicas (PWM – *Pulse Width Modulation*)

Nesta prática um entrada digital (pino 3) será utilizada como entrada analógica, controlando a luminosidade de um LED via PWM. Para realizar este experimento monte o circuito mostrado na Figura 3 e escreva o programa contido no Código 1

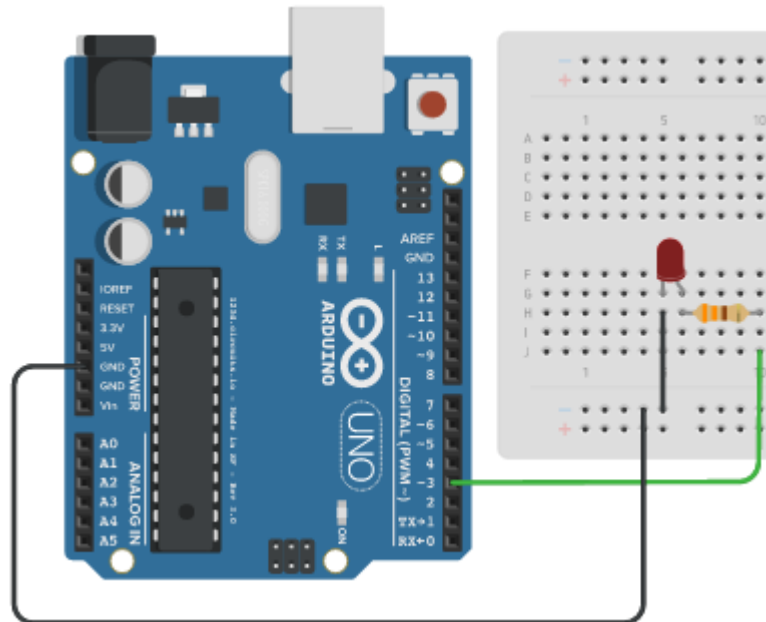


Figura 3 - Circuito da Prática 01

```
#define LED 3

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

void loop() {
  analogWrite(LED,0);
  delay(1000);
  analogWrite(LED,64);
  delay(1000);
  analogWrite(LED,127);
  delay(1000);
  analogWrite(LED,192);
  delay(1000);
  analogWrite(LED,255);
  delay(1000);
}
```

Código 1 - Sketch da Prática 01

4.2. Prática 02 – Entradas analógicas

Nesta prática será utilizado um potenciômetro como entrada analógica de controle para o sinal PWM. O circuito que deve ser montado é mostrado na Figura 4 e o *sketch* deve ser escrito conforme o **Erro! Fonte de referência não encontrada.**

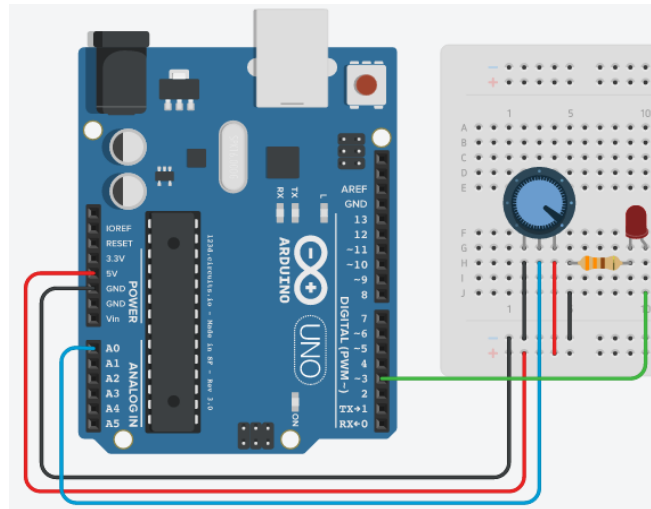


Figura 4 - Circuito da Prática 02

```
#define LED 3

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

void loop() {
  int valorPotenciometro;
  valorPotenciometro = analogRead(A0);
  analogWrite(LED, valorPotenciometro);
}
```

Código 2 - Sketch da Prática 02

O *sketch* mostrado no Código 2, apesar de compilar sem apresentar nenhum erro, causará um efeito indesejado no funcionamento do circuito de ligamento e desligamento do LED durante o giro do potenciômetro. Este efeito é causado pela diferença de bits entre o conversor analógico/digital (A/D), que possui resolução de 10 bits, presente nas entradas analógicas e o conversor digital/analógico (D/A) presente nas saídas analógicas (PWM), que possui resolução de 8 bits. É necessário que seja feita uma adaptação no código para que esse efeito desapareça. Esta adaptação fica como desafio de implementação.

5. REFERÊNCIAS

BANZI, Massimo. *Getting Started with Arduino*. 2ª ed. Sebastopol: O'Reilly, 2011.

EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. *Arduino em Ação*. 1ª ed. [S.l.]: Novatec, 2013.

MONK, Simon. *Programação com Arduino: começando com Sketches*. 1ª ed. Porto Alegre: Bookman, 2013.