Ana Cristina da Silva Krause

# 

# **Trabalho de Pesquisa — Modularização PWM**

## **Tema**

Modulador PWM (Pulse Width Modulation)

## **Objetivo**

Compreender o conceito e o funcionamento da modulação por largura de pulso (PWM), sua importância dentro da arquitetura de sistemas embarcados e IoT, bem como a relação com o ADC (Conversor Analógico-Digital) e a forma como ambos se complementam no controle de sinais, sensores e atuadores.

## **Contexto (com base na aula)**

Durante a aula, estudamos o processamento de sinais em microcontroladores, onde:

* As entradas (sensores) captam sinais analógicos do ambiente;
* As saídas (atuadores) respondem a esses sinais em forma digital;
* E o microcontrolador realiza o processamento utilizando conversão ADC e modulação PWM.

Essa modulação é fundamental para converter comandos digitais em sinais controláveis para atuadores como LEDs, motores e sistemas de potência, permitindo variação de intensidade, velocidade ou posição.

## **Tópicos que devem ser pesquisados e respondidos**

### **1. O que é o Modulador PWM**

Um modulador PWM é um circuito (ou algoritmo) que gera um sinal digital (ligado/desligado) com uma frequência fixa, mas com largura de pulso variável.

Ou seja: A tensão não é alterada de forma contínua;  
O tempo em que o sinal fica em nível alto (ligado) ou baixo (desligado) é que muda. Esse tipo de modulação é muito usado para controlar a potência média entregue a uma carga, como motores, LEDs, aquecedores, etc.

* Explique o significado de “Modulação por Largura de Pulso”

A Modulação por Largura de Pulso (PWM) é uma forma de controlar a energia que chega a um dispositivo (como um motor ou LED) ligando e desligando o sinal muito rápido.  
 O tempo em que o sinal fica ligado (em nível alto) e desligado (em nível baixo) muda conforme o controle que se quer fazer.  
 Assim, mesmo sendo um sinal digital (apenas ligado ou desligado), ele pode simular diferentes níveis de força ou tensão média.

* O que significa “Duty Cycle”?

O Duty Cycle (ou Ciclo de Trabalho) mostra quanto tempo o sinal fica ligado durante um ciclo completo.

Ele é medido em porcentagem (%).

25% → o sinal fica ligado só 1/4 do tempo (baixa potência);  
50% → metade do tempo ligado (potência média);  
75% → ligado quase todo o tempo (alta potência).

* Qual é a diferença entre PWM e um sinal analógico contínuo?

Um sinal analógico contínuo muda de forma suave e constante, podendo assumir qualquer valor entre o mínimo e o máximo (por exemplo, 0V a 5V);  
Um sinal PWM muda apenas entre 0V e 5V (ligado e desligado), mas a média de tempo ligada faz parecer que a tensão varia;

Ou seja, o PWM não muda o valor da tensão, mas o tempo em que ela é aplicada, gerando um efeito parecido com um sinal analógico.

### **2. Como o PWM funciona**

* Descreva como o PWM gera diferentes níveis médios de tensão.

O PWM controla a tensão média (a força média do sinal) mudando o tempo em que o sinal fica ligado.  
 Mesmo que o sinal vá sempre de 0V a 5V, o tempo em que ele fica em 5V é o que define o quanto de energia a carga (motor, LED, etc.) vai receber.

Exemplo com uma fonte de 5V:

* 25% de Duty Cycle: o sinal fica ligado 25% do tempo → tensão média ≈ 1,25V
* 50% de Duty Cycle: o sinal fica ligado metade do tempo → tensão média ≈ 2,5V
* 75% de Duty Cycle: o sinal fica ligado 75% do tempo → tensão média ≈ 3,75V

Ou seja, quanto maior o tempo ligado, maior a tensão média e maior a potência entregue ao dispositivo.

* Mostre um gráfico representando diferentes duty cycles (25%, 50%, 75%):

25%: █▁▁▁█▁▁▁█▁▁▁█▁▁▁ → pouco tempo ligado (sinal fraco)

50%: ██▁▁██▁▁██▁▁██▁▁ → metade do tempo ligado (sinal médio)

75%: ███▁███▁███▁███▁ → quase sempre ligado (sinal forte)

O símbolo “█” representa o sinal ligado (5V) e o símbolo “▁” representa o sinal desligado (0V)

* Explique como a frequência do sinal influencia o comportamento da carga:

A frequência mostra quantas vezes por segundo o ciclo de ligar e desligar acontece.  
 Ela é medida em Hertz (Hz).

* Frequência baixa:  
   O ligar e desligar é mais devagar.  
   → O olho humano pode ver o LED piscando, e motores podem vibrar.
* Frequência alta:  
   O ligar e desligar é muito rápido.  
   → O LED parece aceso de forma contínua e o motor gira de forma suave.

Por isso, em aplicações práticas, o PWM usa frequências altas, para que o funcionamento pareça contínuo e estável.

### **3. Para que serve o PWM**

* Liste e explique aplicações do PWM em:

O PWM é uma forma prática e econômica de controlar energia, permitindo ajustar velocidade, brilho, temperatura ou movimento de forma precisa — algo essencial em eletrônica, robótica e IoT.

* + Controle de velocidade de motores DC:

O PWM é muito usado para aumentar ou diminuir a velocidade de motores de corrente contínua (DC).

Quando o sinal PWM tem um duty cycle pequeno, o motor recebe pouca energia e gira mais devagar.

Quando o duty cycle é maior, o motor recebe mais energia e gira mais rápido.

Exemplos: usado em ventiladores, carrinhos robôs e impressoras 3D.

* + Controle de brilho de LEDs:

Com o PWM, é possível controlar a intensidade da luz de um LED.

Duty Cycle baixo → LED fica mais fraco

Duty Cycle alto → LED fica mais forte

Mesmo sendo um sinal digital (ligado/desligado), o olho humano não percebe o piscar rápido e enxerga apenas o brilho variando suavemente. Exemplo: dimmer de luz, telas de celular, iluminação inteligente.

* + Controle de temperatura e potência elétrica:

O PWM também pode controlar resistências elétricas, aquecedores e sistemas de potência.

Ele liga e desliga rapidamente o circuito, controlando quanto tempo a corrente passa e, assim, a quantidade de calor ou energia gerada.

Exemplo: controle de aquecimento em impressoras 3D, chapas térmicas, ferros de solda automáticos.

* + Outras aplicações em robótica e IoT.

O PWM é muito usado em projetos de robótica e dispositivos IoT (Internet das Coisas), pois permite controlar vários tipos de atuadores e componentes.

Alguns exemplos:

* Servomotores: usados para mover braços robóticos ou rodas com precisão;
* Bombas e ventiladores: controle de fluxo e velocidade;
* Fontes de alimentação e inversores: regulação eficiente de energia;
* Dispositivos inteligentes (IoT): controle de brilho, motores, válvulas e sensores em casas e fábricas conectadas;

### **4. Importância do PWM**

* Por que o PWM é mais eficiente que um sinal analógico puro?

O PWM é mais eficiente porque o circuito eletrônico (como o transistor que controla o motor ou LED) só trabalha em dois estados: ligado ou desligado.

Quando está ligado, a energia é usada totalmente; quando está desligado, não há gasto.

Já em um sinal analógico puro, o circuito precisa regular a tensão continuamente, o que gera perda de energia na forma de calor.

* Qual é a vantagem de controlar energia através do duty cycle?

A principal vantagem é que o duty cycle controla o tempo em que a energia é aplicada, sem precisar mudar a tensão da fonte.

Isso torna o controle mais simples, preciso e econômico.

Exemplo:

Duty Cycle de 30% → o motor recebe pouca energia e gira devagar.

Duty Cycle de 80% → o motor recebe mais energia e gira rápido.

* Explique a importância do PWM no consumo energético e na precisão de controle.

O PWM ajuda muito a economizar energia e a controlar melhor os dispositivos.

* Como o sinal é rápido e preciso, ele mantém o controle estável (sem oscilações grandes).
* Gasta menos bateria, o que é ótimo para dispositivos IoT e sistemas portáteis.
* Permite ajustes finos de velocidade, brilho ou temperatura, garantindo **maior precisão e eficiência**.

### **5. Aplicação prática no Arduino**

* Como o Arduino gera um sinal PWM?

O Arduino gera sinais PWM usando temporizadores internos (timers).

Esses timers fazem o microcontrolador ligar e desligar o pino muito rápido, criando o efeito da modulação por largura de pulso.

Mesmo que o Arduino só consiga enviar sinais digitais (0V ou 5V), o tempo que o pino fica ligado muda, simulando uma tensão média variável — perfeita para controlar brilho, motores ou aquecimento.

* analogWrite(pino, valor);

pino: é o pino de saída PWM (marcado com ~).

valor: vai de 0 a 255, representando o duty cycle: 0 desligado, 127 metade do tempo ligado, 255 sempre ligado

* Quais pinos possuem suporte (marcados com ~)?

Nos modelos mais comuns, como o Arduino Uno, os pinos com suporte a PWM são marcados com o símbolo ~ ao lado do número. Esses pinos podem ser usados com o comando analogWrite() para gerar sinais PWM.

Pinos PWM no Arduino Uno: 3, 5, 6, 9, 10 e 11

* Mostre um exemplo prático com código, por exemplo, controle de brilho de um LED:

int led = 9; // Pino PWM (~9)

int brilho = 0; // Começa apagado

int passo = 5; // Quantidade de mudança no brilho

void setup() {

pinMode(led, OUTPUT); // Define o pino como saída

}

void loop() {

analogWrite(led, brilho); // Envia o valor PWM para o LED

brilho = brilho + passo; // Aumenta o brilho

if (brilho <= 0 || brilho >= 255) {

passo = -passo; // Inverte a direção (escurece ou clareia)

}

delay(30); // Espera um pouco antes de mudar o brilho

}

### **6. Interferência do PWM no ADC**

* Explique o que é o ADC (Conversor Analógico-Digital) e sua função no microcontrolador:

O ADC (Conversor Analógico-Digital) é o componente do microcontrolador que transforma sinais analógicos em números digitais.

Os sensores (como de temperatura, luz, ou umidade) geralmente enviam sinais analógicos, que variam de forma contínua.

O ADC “lê” essa tensão e converte para um valor numérico (geralmente de 0 a 1023 no Arduino) para que o microcontrolador possa entender e processar.

* Descreva como sinais PWM podem interferir em medições analógicas, caso não haja filtragem:

O PWM liga e desliga muito rápido, criando pulsos elétricos que podem gerar ruídos na alimentação ou nas linhas do circuito.

Quando o ADC tenta medir um sinal analógico próximo a esses pulsos, ele pode pegar valores errados ou instáveis, pois o ruído “bagunça” a leitura.

* Cite soluções para evitar interferências (uso de filtros RC, capacitores e separação de alimentação):

Existem várias formas de filtrar e proteger o circuito contra interferências do PWM:

1. Filtro RC (Resistor + Capacitor):  
   * Suaviza o sinal PWM, transformando-o em uma tensão média mais estável.
   * O resistor e o capacitor trabalham juntos para eliminar os picos rápidos de tensão.
2. Capacitores de desacoplamento:  
   * Colocados próximos aos pinos de alimentação do microcontrolador.
   * Reduzem ruídos elétricos vindos de motores ou outros atuadores.
3. Separação de alimentação:  
   * Usar fontes separadas ou trilhas diferentes para os circuitos analógicos (sensores) e digitais (atuadores).
   * Isso evita que o ruído dos sinais PWM chegue aos sensores.

* Explique por que é importante entender essa interferência em projetos de IoT:

Nos projetos de IoT (Internet das Coisas), os sensores e atuadores costumam trabalhar ao mesmo tempo no mesmo sistema.  
 Se o PWM causar interferência, o microcontrolador pode ler dados errados e tomar decisões incorretas, como:

* Mostrar medidas de temperatura erradas;
* Acionar motores na hora errada;
* Fazer controles automáticos funcionarem de forma imprecisa.