



UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO

Engenharia de Computação
Universidade Federal do Vale do São Francisco, Juazeiro

Relatório

Sistema de Controle de Motor

Breno Gabriel de Souza Coelho
Daniel Alencar Penha Carvalho
Pedro Henrique Amaro Ferreira Lacerda

Juazeiro-BA, 20 /12/2024

Introdução

O objetivo do presente trabalho é implementar um sistema de controle capaz de ajustar a frequência de rotação do motor escovado presente na placa de desenvolvimento PICDEM através de um ESP32 externo. Para tal, foi utilizado um algoritmo PID implementado no microcontrolador, onde o fator de correção é calculado com base na frequência pretendida (medida em RPM e informada no início da execução) e na frequência atual de rotação do motor (medida através do sensor óptico presente na placa). Inicialmente parado, o motor ganha velocidade conforme recebe um sinal PWM enviado através do ESP32 para o pino adequado da placa. A partir disso, o sinal é ajustado conforme necessário para garantir que a velocidade de rotação medida se aproxime da pretendida.

Desenvolvimento

Componentes Utilizados

A PICDEM é uma placa de estudo equipada com vários componentes eletrônicos, incluindo um microcontrolador e periféricos como sensores e atuadores. Para este projeto, utilizamos:

- **Motor escovado DC** (da PICDEM);
- **Sensor óptico** para medição de velocidade (da PICDEM);

- **Microcontrolador ESP32**, que realiza o controle do motor e comunicação com o computador (não faz parte da PICDEM)

Arquitetura do Sistema

O sistema foi projetado para permitir comunicação eficiente entre o microcontrolador ESP32 e a placa PICDEM. As conexões principais são:

- **PWM (pino 26 do ESP32 para o pino N2 da PICDEM)**: Sinal de controle enviado para o motor (Fio Laranja);
- **Leitura do sensor óptico (pino 14 do ESP32 para o pino J7 da PICDEM)**: O ESP32 recebe o valor de pulsos gerados pelo sensor (Fio Verde);
- **GND compartilhado (GND do ESP32 para o pino VSS da PICDEM)**: Para referência elétrica comum entre os dispositivos (Fio Preto);
- **Sinal de habilitação (VIN do ESP32 para o pino P1 da PICDEM)**: Mantém o motor habilitado para operação (Fio Vermelho).

A Figura 1 apresenta o sistema completo.

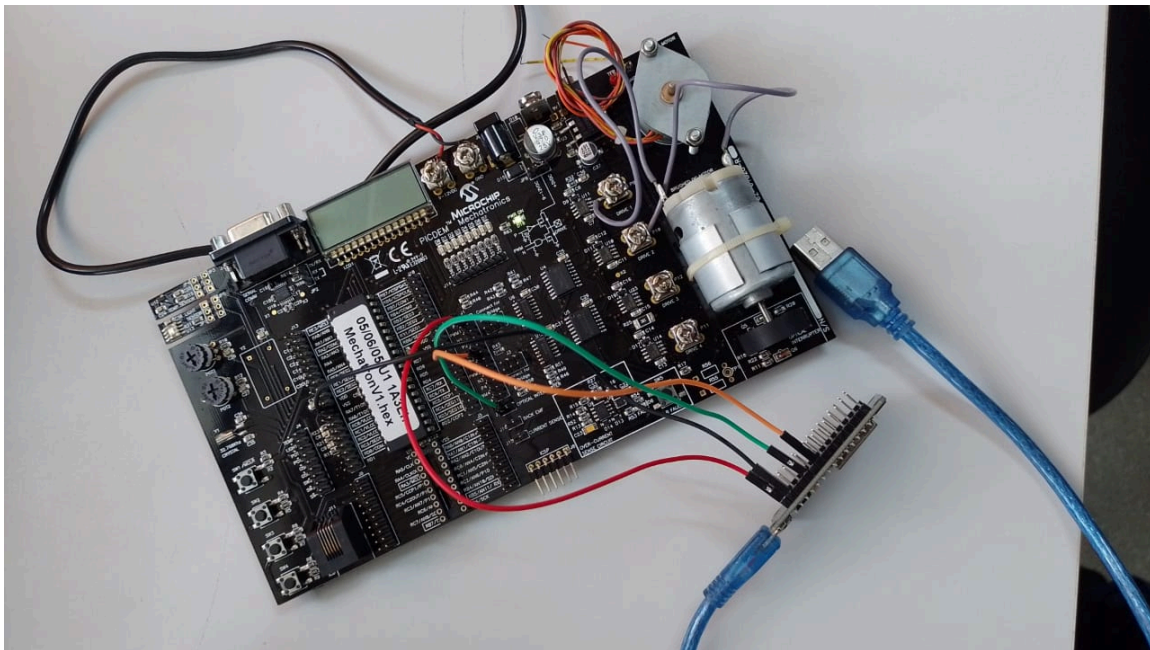


Figura 1. Montagem do sistema com a placa PICDEM e o ESP32.

Software

No ESP32

O código do ESP32 foi implementado em **Arduino IDE**, utilizando bibliotecas para comunicação serial, controle PWM, e cálculo de tempos. O fluxo básico é ler o valor de RPM transmitido pela serial por parte do usuário e depois o sistema realiza os seguintes passos em loop:

1. Calcular o RPM atual com base nos pulsos do sensor;
2. Aplicar o controle PID para ajuste do PWM;
3. Enviar o sinal PWM de volta para o driver do motor da PICDEM.

Cálculo de RPM

O sensor óptico da PICDEM gera pulsos proporcionais à velocidade do motor. O cálculo de RPM é baseado na fórmula:

$$RPM = \frac{pulsos * 1000 * 60}{PPR * intervalo}$$

Onde:

- **Pulsos:** Quantidade de pulsos gerados pelo sensor óptico;
- **Intervalo:** É a quantidade de tempo (em milissegundos) entre cada leitura da quantidade de pulsos gerados pelo sensor óptico.
- **PPR:** Número de furos no anteparo do motor que são lidos pelo sensor óptico a cada revolução do motor. No caso do motor escovado da placa PICDEM, são dois.

A lógica da equação acima é a seguinte: cada pulso produzido pelo sensor óptico é gerado pela detecção de um dos buracos presentes no disco do motor. Sendo dois buracos, sempre que ocorrem dois pulsos, houve uma rotação completa. A partir disso, se houveram **N** pulsos, é porque ocorreram **N/2** (ou **N/PPR**) rotações completas. Contudo, devemos levar em consideração que a contagem de pulsos gerados só é detectada a cada **i** milissegundos, de modo que a quantidade de pulsos por milissegundo é dada por **p/i** (pulsos medidos em “i” milissegundos dividido pelo valor de “i”. Por exemplo, se são 500 pulsos a cada 10ms, então existem 50 pulsos a cada 1ms - 500/10). Por fim, sendo o valor em RPM, devemos converter os milissegundos para minutos multiplicando por 1000 depois por 60. Matematicamente:

$$RPM = \text{pulsosPorMinuto} / PPR \quad (PPR = 2)$$

$$RPM = \text{pulsosPorMilissegundo} * 1000 * 60 / PPR$$

$$RPM = (\text{pulsosLidosNaJanelaDeTempo} / \text{janelaDeTempo}) * 1000 * 60 / PPR$$

$$RPM = \text{pulsos} * 1000 * 60 / PPR * \text{intervalo}$$

Os valores de RPM são enviados para o computador a partir do ESP32 através da porta Serial, cujo valor é acessado através de um software em Python, que renderiza um gráfico dinâmico em tempo real, como mostrado na figura 2.

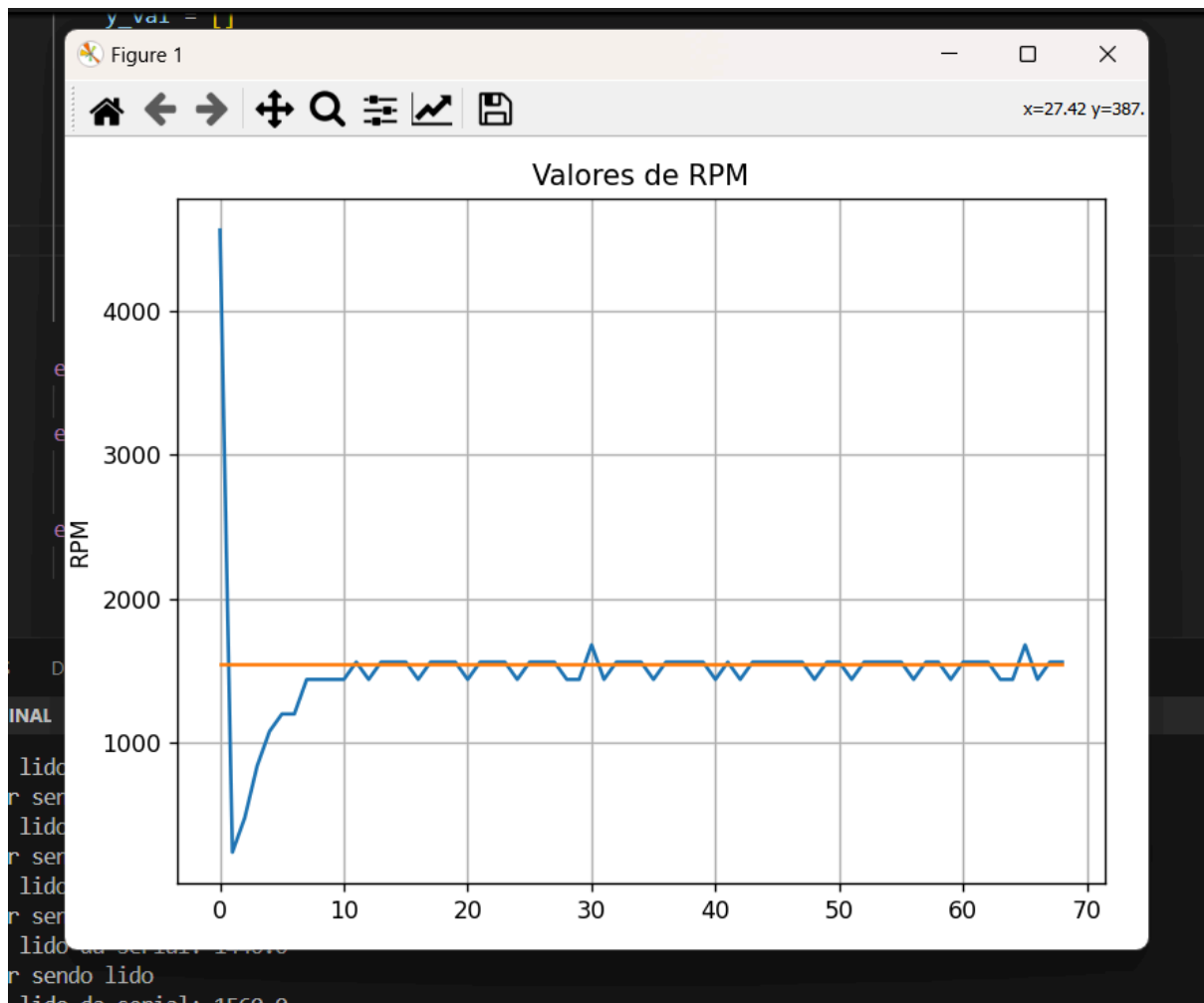


Figura 2. Gráfico gerado pelo software em Python. A linha azul representa o valor de RPM medido pelo sensor ao longo do tempo. A linha amarela é o valor alvo.

Controle PID

O controle da velocidade do motor utiliza um algoritmo **PID (Proporcional-Integral-Derivativo)**. Os valores das constantes foram ajustados empiricamente para garantir um desempenho satisfatório. O processo de controle segue as etapas:

1. **Cálculo do erro:** Diferença entre o RPM desejado e o medido.
2. **Geração do sinal de controle:** O PID calcula um novo valor de PWM baseado no erro.
3. **Ajuste do motor:** O valor PWM é enviado ao motor para corrigir sua velocidade.

No Python

O programa em Python se comunica com o ESP32 através da serial e recebe os valores de RPM do sensor óptico. Esse programa usa esses valores para gerar um gráfico em tempo real da velocidade do motor. É possível visualizar muito bem como é o ajuste do PID no PWM. O código utiliza bibliotecas como **pyserial** e **matplotlib**

para receber os dados via serial e atualizar em tempo real o gráfico de RPM, respectivamente.

Repositório do projeto com os códigos e manual de execução: [GitHub - Motor Control System](#).

Testes

Os testes foram realizados para validar:

1. **Comunicação entre ESP32 e PICDEM:** Verificando a integridade dos dados recebidos pelo sensor;
2. **Precisão do cálculo de RPM:** Comparando com um tacômetro de referência;
3. **Estabilidade do controle PID:** Avaliando o tempo de resposta e o overshoot.

Resultados

O sistema demonstrou:

- Controle estável do motor com convergência rápida para o RPM desejado ou valor flutuante próximo do valor desejado;
- Monitoramento dinâmico e responsivo dos valores medidos;
- Integração bem-sucedida entre ESP32 e PICDEM, utilizando comunicação eficiente e confiável.

Conclusão

O projeto ilustra o potencial do ESP32 para controle de motores em sistemas embarcados, utilizando sensores externos e algoritmos de controle clássico. Além disso, a integração com um programa Python fornece uma interface intuitiva para monitoramento em tempo real. O sistema desenvolvido pode ser expandido para aplicações mais complexas, como controle de velocidade em múltiplos motores ou automação industrial.