# Trabalho Prático

Marcos Paulo de Oliveira Pereira
*21.1.1090*
*Engenharia da Computação - UFOP*
João Monlevade, Brasil
marcos.pop@aluno.edu.br

Matheus Martins Nunes
*22.1.8027*
*Engenharia da Computação - UFOP*
João Monlevade, Brasil
matheus.mn@aluno.ufop.edu.br

***Resumo*—Esse relatório descreve o processo de criação e desenvolvimento de um sistema de controle de vazão, contando com a ideia inicial, testes em simulação e montagem prática. Ele lista todos os componentes e ferramentas utilizadas, além de fazer ligação direta com os conteúdos aprendidos durante o semestre.**

***Palavras-chave*—tinkercad, vazão, microcontrolador, bomba, sensor, erro, PWM**

## I. Introducão

O sistema de controle de vazão é fundamental tanto no ambiente industrial quanto no ambiente doméstico. Através dele, é possível garantir a eficiência e precisão do fluxo de fluidos.

Neste trabalho prático proposto, foi desenvolvido um sistema de controle de vazão utilizando o microcontrolador Arduino UNO. Sua escolha se deve ao baixo custo, versatilidade e grande suporte da comunidade, que facilitou a implementação do projeto.

## II. Objetivos

O principal objetivo deste projeto é aplicar os conhecimentos adquiridos ao longo da disciplina , bem como outros aprendizados ao longo do curso, no desenvolvimento de um sistema de controle de vazão. Além disso, o projeto busca reforçar a compreensão sobre a integração de hardware e software, aprimorar habilidades em programação embarcada e estimular a resolução de problemas práticos por meio da automação.

## III. Metodologia

### *A. Abordagem inicial:*

Inicialmente, o projeto foi simulado na plataforma Tinkercad, que nos possibilitou testar o circuito e o código antes da implementação prática do sistema. Após a simulação, o sistema foi montado fisicamente, com uso de sensores e atuadores conectados ao Arduino com objetivo de controlar a vazão da água.

Fez-se necessário a realização de mudanças pontuais no projeto final, devido aos detalhes dos componentes que são fundamentais para que o sistema funcione de forma correta, detalhes esses que não são possíveis de simular no TinkerCad, devido às suas limitações. Portanto, o circuito projetado de maneira teórica sofreu adaptações necessárias para que fosse transposto para uma aplicação prática funcional.

Inicialmente, simulamos um sistema que controla a vazão de saída da bomba com base na comparação entre:

- Vazão de entrada: Simulada por um potenciômetro (0–1023) convertido para 0–50 L/min.
- Vazão de saída desejada: 10 L/min.

Lógica tomada como base para o desenvolvimento:

- Se vazaoEntrada > 10 L/min → PWM reduzido.
- Se vazaoEntrada < 10 L/min → PWM aumentado.

Lembrando que o valor de 10 L/min é setado por uma variável de controle e pode ser qualquer um, em termos de simulação.

Essa abstração do circuito sugeria que o sistema funcionaria de forma independente. Ajustando diretamente uma vazão de saída hipotética.

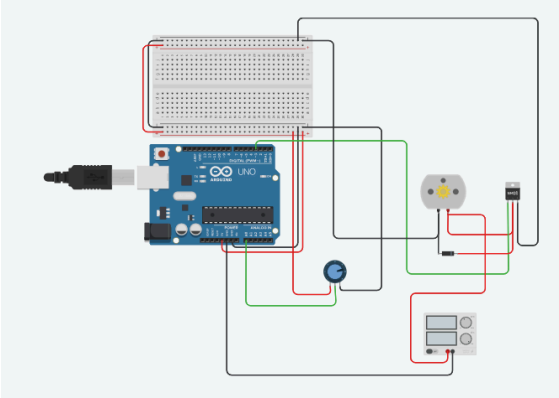

**Imagem 1 - Simulação no TinkerCad**

https://www.tinkercad.com/things/fJxeaeN9Lwv-fabulous-fyyran/editel?sharecode=-Vt37JIhxmXSozLc3WI6_VfZkFLfABki0MEOpoHXSM0

Flow Range:100L/H-/1800H-L/H

| Flow (L/H) | Freza(HZ) | Erro range |
|---|---|---|
| 120L/H | 16 | ±10 |
| 240L/H | 32.5 | |
| 360L/H | 49.3 | |
| 480L/H | 65.5 | |
| 600L/H | 82 | |
| 720L/H | 90.2 | |

**Imagem 2 - Tabela que relaciona a frequência dos pulsos e a vazão em L/H**

A adaptação do modelo de controle foi necessária devido às restrições físicas do sistema. A fonte regulável de 12V foi ajustada para fornecer 3A, garantindo que a bomba pudesse operar na faixa de vazão desejada. Com isso, conseguimos estabilizar a vazão próxima ao valor estipulado de 10L/min.

*B. Implementação Real*

Ao nos depararmos com implementação prática, percebemos que essa abordagem não era a mais trivial, pois o controle de vazão que desejávamos envolvia controlar a vazão que a bomba geria sob o sistema como um todo; no nosso caso, um balde de água, onde ela seria submersa. Com isso, notamos que os conceitos "vazão de entrada" e "vazão de saída" não foram pensados da maneira que queríamos, já que estaríamos trabalhando com a vazão que a bomba gera ao estar submersa e a partir daí pedir e controlar a vazão que seria gerada no sensor. Diante dessa verdade, reformulamos o modelo para focar exclusivamente na variável de controle desejada. A nova abordagem estabelece um valor desejado para a vazão e ajusta o PWM da bomba com base no erro entre esse valor e a vazão real medida. O cálculo do erro é feito pela expressão:

$$erro = vazão\ controlada - vazão\ de\ entrada$$

Para corrigir esse erro, utilizamos um controlador proporcional-integral (PI) que ajusta a variável de controle de forma proporcional ao erro a cada iteração do microcontrolador (no nosso caso, 1s). A equação é:

$$PWM = Kp * erro + Ki * \sum erro$$

Os valores de Kp = 2,5 e Ki = 1,5 foram determinados de forma empírica para garantir uma resposta adequada. Também estabelecemos limites para o PWM entre 140 e 255, pois os valores abaixo de 140 não acionam o circuito de ativação da bomba, pois não geram uma tensão suficiente para ligar o relé. O sensor de vazão opera por pulsos, e a conversão para L/min seguiu a proporção empírica de 8.2 pulsos por L/min.

Utilizamos com base o datasheet do sensor para traçar essa proporção:

Componentes Utilizados:

- Arduino Uno: processa os dados do sensor de vazão e controla a bomba.
- Sensor de fluxo YF-S201: mede a vazão real da água.
- Bomba d'água ZYW680 DC 12V: ajusta a vazão conforme necessário.
- Relé: controla o acionamento da bomba.
- Transistor BC337 e diodo: protegem o circuito ao controlar o relé.
- Resistores (220Ω E 12KΩ): usados para ajuste e proteção dos sinais elétricos.
- Fonte Externa 12V: Para realizar a alimentação da bomba.

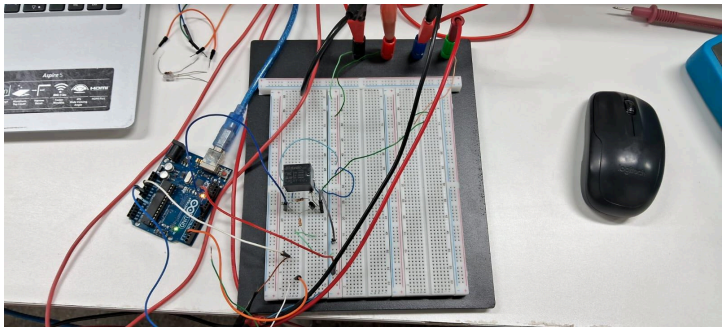

**Imagem 3 - Circuito com todos os componentes**

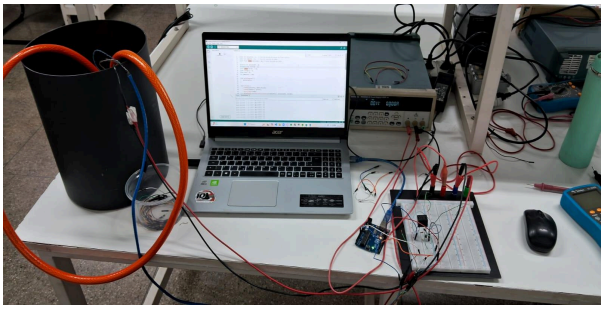

**Imagem 4 - Sistema de controle completo, com a bomba submersa**

## IV. Resultados

Ao testar o circuito com a configuração da fonte com 12V e 3A, que possibilitam para que a bomba bombeie na faixa de vazão desejada, os testes comprovaram a eficácia do projeto, fazendo com que os resultados obtidos sempre estivessem muito próximos a 10L/min, que é a vazão de saída desejada.

Vazao: 7.50 L/min | PWM: 140
Vazao: 8.20 L/min | PWM: 160
Vazao: 9.10 L/min | PWM: 200
Vazao: 10.50 L/min | PWM: 240
Vazao: 11.00 L/min | PWM: 255
Vazao: 10.80 L/min | PWM: 255
Vazao: 10.50 L/min | PWM: 255
Vazao: 10.20 L/min | PWM: 255

Através dos resultados obtidos, é possível perceber que o projeto atendeu aos requisitos pré-definidos anteriormente, que era de um sistema de controle de uma variável específica, no nosso caso a vazão.

O código fonte completo do projeto pode ser encontrado no repositório: https://github.com/emipe09/ControleDeVazao

## V. Considerações Finais

Esse projeto foi muito importante para a nossa graduação, nos possibilitando colocar em prática vários conceitos aprendidos não apenas na matéria “microprocessadores e microcontroladores”, mas também diversas outras disciplinas ao longo do curso.

O projeto foi realizado com êxito, tanto na simulação quanto na prática. Apesar das dificuldades que a dupla apresentou ao adaptar o circuito simulado para o circuito real, foi possível finalizar o projeto de forma bem sucedida.

Trabalhos em que são colocados em prática aplicações que são comuns no mundo real fortalecem ainda mais o aprendizado do aluno. O sistema de controle de vazão pode ser aplicado em um sistema de irrigação de plantas, por exemplo, em que a vazão da água pode ser regulada de acordo com a necessidade da planta, otimizando o consumo e garantindo maior eficiência no uso dos recursos.

## Referências Bibliográficas


[1] L GUSTAVO-ROBÓTICA🤖INTELIGÊNCIA ARTIFICIAL🔮. Projeto 38 - Sensor de FLUXO e Medidor de Vazão com Arduino - Projetos Jetsons. Disponível em: <https://www.youtube.com/watch?v=AqRSCihZ34k>. Acesso em: 30 mar. 2025.

[2] DANIEL GHISLENI. Video 19 - TASMOTA - Medindo vazão com sensor de fluxo. Disponível em: <https://www.youtube.com/watch?v=Kpz_CuaT9g0>. Acesso em: 30 mar. 2025.

[3] YF-S201 – Sensor de Fluxo. Datasheet. Disponível em: https://www.makerhero.com/img/files/download/YF-S201-Datasheet.pdf. Acesso em: 30 mar. 2025.