



Universidade Federal de Ouro Preto  
Escola de Minas  
CECAU - Colegiado do Curso de  
Engenharia de Controle e Automação



Diego de Jesus Ferreira

## **Desenvolvimento de Sistema de Bombeamento de Água para Ensino Remoto**

Monografia de Graduação

Ouro Preto, 2025

Diego de Jesus Ferreira

# **Desenvolvimento de Sistema de Bombeamento de Água para Ensino Remoto**

Trabalho apresentado ao Colegiado do Curso  
de Engenharia de Controle e Automação da  
Universidade Federal de Ouro Preto como  
parte dos requisitos para a obtenção do Grau  
de Engenheiro(a) de Controle e Automação.

Universidade Federal de Ouro Preto

Orientador: Prof. Alan Kardek Rêgo Segundo

Ouro Preto

2025



MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE OURO PRETO  
REITORIA  
ESCOLA DE MINAS  
DEPARTAMENTO DE ENGENHARIA CONTROLE E  
AUTOMACAO



**FOLHA DE APROVAÇÃO**

**Diego de Jesus Ferreira**

**Desenvolvimento de Sistema de Bombeamento de Água para Ensino Remoto**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Controle e Automação

Aprovada em 21 de agosto de 2025

**Membros da banca**

Dr. Alan Kardek Rêgo Segundo - Orientador (Universidade Federal de Ouro Preto)  
Dr. João Carlos Vilela de Castro - Convidado (Universidade Federal de Ouro Preto)  
Dr. Paulo Marcos de Barros Monteiro - Convidado (Universidade Federal de Ouro Preto)

Alan Kardek Rêgo Segundo, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 25/08/2025



Documento assinado eletronicamente por **Alan Kardek Rego Segundo, PROFESSOR DE MAGISTERIO SUPERIOR**, em 25/08/2025, às 14:13, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufop.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0964232** e o código CRC **23013B4A**.

**Referência:** Caso responda este documento, indicar expressamente o Processo nº 23109.010523/2025-18

SEI nº 0964232

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163  
Telefone: 3135591533 - [www.ufop.br](http://www.ufop.br)

# Resumo

O gerenciamento inteligente de recursos hídricos é uma questão de crescente importância, dada a escassez de água e os altos índices de desperdício. Este trabalho teve como objetivo o desenvolvimento de um sistema de controle para o bombeamento de água entre duas caixas d'água, visando um melhor aproveitamento deste recurso. O método empregado foi a construção de um protótipo em um ambiente de simulação totalmente virtual, contornando a necessidade de hardware físico, que demonstra potencial para uso em contextos de aprendizagem remota. Foram utilizadas ferramentas como o microcontrolador PIC18F4520, simulado no software PICSimLab, e o ambiente de desenvolvimento MPLAB X para a criação do firmware. A comunicação entre o sistema embarcado e uma interface de supervisão foi estabelecida por meio do protocolo Modbus. Os resultados demonstraram que o sistema operou com sucesso, controlando o acionamento de uma bomba com base nos níveis dos tanques e nos parâmetros configurados pelo usuário, com modos de operação manual e automático. Conclui-se que a simulação validou a lógica de controle proposta, confirmando a viabilidade do sistema para uma futura implementação física e sua eficácia como ferramenta para a gestão hídrica.

**Palavras-chaves:** controle de nível, sistemas embarcados, simulação, PICSimLab, Modbus.

# Abstract

The intelligent management of water resources is a matter of growing importance, given the scarcity of water and high rates of waste. This work aimed to develop a control system for water pumping between two water tanks, seeking a better use of this resource. The employed method was the construction of a prototype in a fully virtual simulation environment, bypassing the need for physical hardware, which proved to be an effective approach, especially in remote learning contexts. For this, tools such as the PIC18F4520 microcontroller, simulated in the PICSimLab software, and the MPLAB X development environment for firmware creation were used. The communication between the embedded system and a supervisory interface was established using the Modbus protocol. The results demonstrated that the system operated successfully, controlling the activation of a pump based on the tank levels and user-configured parameters, with both manual and automatic operation modes. It is concluded that the simulation validated the proposed control logic, confirming the system's viability for future physical implementation and its effectiveness as a tool for water management.

**Key-words:** level control, embedded systems, simulation, PICSimLab, Modbus.

# Lista de ilustrações

|                                                                                         |    |
|-----------------------------------------------------------------------------------------|----|
| Figura 1 – Placa PICSimLab . . . . .                                                    | 24 |
| Figura 2 – Tanque em 0% . . . . .                                                       | 24 |
| Figura 3 – Tanque em 50% . . . . .                                                      | 25 |
| Figura 4 – Tanque em 100% . . . . .                                                     | 25 |
| Figura 5 – VSPE . . . . .                                                               | 25 |
| Figura 6 – Implementação de leitura dos registradores . . . . .                         | 25 |
| Figura 7 – Implementação de escrita dos registradores . . . . .                         | 26 |
| Figura 8 – Trecho de implementação no microcontrolador . . . . .                        | 26 |
| Figura 9 – Trecho de implementação das funções de leitura no microcontrolador . . . . . | 26 |
| Figura 10 – Demonstração do sistema em equilíbrio. . . . .                              | 27 |
| Figura 11 – Sistema em dequilíbrio por conta do tanque 2 . . . . .                      | 28 |
| Figura 12 – Sistema em dequilíbrio por conta do tanque 1 . . . . .                      | 28 |
| Figura 13 – Sistema em equilíbrio . . . . .                                             | 29 |

# Sumário

|     |                                                                                                        |    |
|-----|--------------------------------------------------------------------------------------------------------|----|
| 1   | INTRODUÇÃO . . . . .                                                                                   | 7  |
| 1.1 | Justificativas e Relevância . . . . .                                                                  | 8  |
| 1.2 | Materiais e Métodos . . . . .                                                                          | 9  |
| 1.3 | PICSimLab: Um Laboratório Virtual Completo para Simulação de<br>Microcontroladores . . . . .           | 10 |
| 1.4 | MPLAB X IDE: A Plataforma Central para o Desenvolvimento com<br>Microcontroladores Microchip . . . . . | 12 |
| 1.5 | Protocolo Modbus: O Padrão da Comunicação Industrial . . . . .                                         | 14 |
| 1.6 | Virtual Serial Ports Emulator (VSPE) . . . . .                                                         | 16 |
| 1.7 | Construção do supervisor . . . . .                                                                     | 17 |
| 1.8 | Objetivos . . . . .                                                                                    | 18 |
| 2   | REVISÃO DE LITERATURA . . . . .                                                                        | 19 |
| 2.1 | Profissional de engenharia de controle e automação . . . . .                                           | 19 |
| 2.2 | Sistemas embarcados . . . . .                                                                          | 20 |
| 2.3 | Utilização de laboratórios virtuais durante a pandemia da SARS-CoV-2 . . . . .                         | 20 |
| 2.4 | Utilização de cisternas no Brasil . . . . .                                                            | 21 |
| 2.5 | Domótica no Brasil . . . . .                                                                           | 22 |
| 3   | DESENVOLVIMENTO . . . . .                                                                              | 24 |
| 4   | RESULTADOS . . . . .                                                                                   | 27 |
| 5   | CONSIDERAÇÕES FINAIS . . . . .                                                                         | 30 |
|     | Referências . . . . .                                                                                  | 31 |

# 1 Introdução

Ao longo do curso de Engenharia de Controle e Automação, os estudantes percorrem o ciclo básico, comum aos cursos de engenharia, que abrange disciplinas fundamentais como cálculo diferencial e integral, álgebra linear, física, estatística e lógica de programação, entre outras. O ciclo profissionalizante, por sua vez, foca no aprendizado de teoria de controle, eletrônica, processamento de sinais e programação de sistemas embarcados, criando a base para que o profissional possa desenvolver sistemas complexos, uma tarefa que demanda conhecimento multidisciplinar.

Os Sistemas Embarcados encontram-se cada vez mais presentes em nosso dia-a-dia, e com uma utilização e importância crescente. Isso faz com que esse tipo de segmento seja fundamental para a formação do profissional na área de Engenharia de Controle e Automação. Conforme aponta [Silva et al. \(2024\)](#), dada a necessidade do mercado pela formação de profissionais da área de sistemas embarcados, diversas instituições de ensino superior têm ofertado disciplinas dessa área tanto em cursos de graduação como em cursos de pós-graduação nas Engenharias.

Com a necessidade do isolamento social imposta pela pandemia do novo coronavírus (Sars-CoV-2) o desafio de integração por meio de atividades laboratoriais tornou-se ainda maior, visto que muitas universidades passaram a ofertar essas disciplinas apenas de forma remota. Se nos laboratórios físicos de sistemas embarcados os estudantes tinham acesso à diversas plantas didáticas e contato direto com hardwares previamente adquiridos pelas instituições, com a pandemia essa interação ficou limitada ao computador pessoal do aluno ([PARREIRA et al., 2021](#)). Isso ressaltou ainda mais a importância do ensino no ambiente virtual, como softwares de simulação de projetos, possibilitando o contínuo aprendizado dos acadêmicos de forma remota, fortalecendo a capacidade de ensino em momentos de crise como o que estamos vivendo ([SOUZA; ONESSEKEN, 2020](#)).

Atualmente o gerenciamento inteligente da água doce é uma questão importante, pois a cada dia aumenta-se a sua escassez. Estudos demonstram que a demanda por água dobra a cada 21 anos, ao passo que a disponibilidade de água doce no mundo caiu cerca de 62% nos últimos 50 anos. Embora 3/4 do planeta Terra seja coberto por água, apenas 2,5% desse total é formado por água doce com potencial para utilização humana ([RIBEIRO; ROLIM, 2017](#)).

O Brasil é uma potência natural das águas, com 12% de toda água doce do planeta. No entanto, ela é desigualmente distribuída no país: 72% na região Amazônica, 16% no Centro Oeste, 8% no Sul e no Sudeste e 4% no Nordeste. Essa distribuição não coincide com a concentração da população no território e, apesar da abundância, 38% da água

tratada é desperdiçada, atingindo percentuais ainda mais elevados em alguns estados, como o Amapá no qual o desperdício correspondeu a 70% no ano de 2016 (PORTO; PAVÃO; NOGUEIRA, 2019).

Em 2015, durante o período de chuvas, 110 cidades mineiras declararam estado de emergência. O Sistema Paraopeba, que abastece a região metropolitana de Belo Horizonte, operou, no final de 2015, com apenas 30% da capacidade. No ano anterior, esse índice era de 78%. A escassez de recursos hídricos é latente. Medidas de restrição, estado de atenção e de alerta ao uso e à captação de águas são, de fato, uma necessidade (RIBEIRO; ROLIM, 2017).

Uma alternativa tecnológica de captação e armazenamento de água é a cisterna, que é utilizada para captar água de chuva e armazená-la durante os meses sem precipitação (AMORIM; PORTO, 2001). Ainda segundo Amorim e Porto (2001), diversas pequenas comunidades no semiárido nordestino são abastecidas por cisternas que captam água de chuva e também recebem água através de carros-pipa. Estas práticas minimizam o problema do abastecimento de água, embora seja preciso cuidados com a qualidade desta água, em relação aos riscos de contaminação biológica.

A Domótica é uma tecnologia em ascensão, diversas casas e prédios já contam com sistemas integrados de controle que proporcionam infinitas atividades visando tornar práticas cotidianas mais confortáveis e rápidas, assim como proteger o usuário de diversas ameaças inesperadas. Tais procedimentos, em sua maioria, não necessitam de presença física para controlar, além disso podem ser monitorados a distância, de qualquer parte do mundo. Segundo Hasan et al. (2018), o crescimento da domótica se deve ao recente avanço tecnológico em dispositivos eletrônicos que resultou na produção de um grande número de sistemas fáceis de usar e flexíveis que promoveram a tecnologia de vida inteligente. Os eletrodomésticos podem ser controlados remotamente por nossos dispositivos eletrônicos, como smartphones, Personal Device Assistant (PDA) e laptops, o que torna nossa vida mais fácil e confortável.

Diante do cenário exposto, o presente trabalho tem como objetivo o desenvolvimento de um sistema de controle para caixa d'água, utilizando ferramentas de simulação, a fim de contribuir para um melhor gerenciamento desse recurso hídrico.

## 1.1 Justificativas e Relevância

A crescente escassez hídrica, agravada por índices alarmantes de desperdício, representa um dos desafios mais urgentes da atualidade, demandando soluções tecnológicas inovadoras para a gestão inteligente dos recursos hídricos. Diante deste cenário, o desenvolvimento de sistemas automatizados para otimizar o uso da água em ambientes residenciais e rurais torna-se fundamental não apenas para a conservação deste recurso vital, mas

também para a promoção da sustentabilidade e melhoria da qualidade de vida. Este trabalho se justifica, portanto, pela aplicação direta dos conhecimentos de Engenharia de Controle e Automação na criação de uma solução prática e de baixo custo para um problema socioambiental de grande impacto.

A relevância acadêmica e pedagógica do projeto é igualmente significativa, especialmente ao considerar as transformações impostas pelo ensino remoto. A criação de um protótipo em um ambiente de simulação totalmente virtual, utilizando ferramentas como o PICSimLab e o MPLAB X IDE, contorna a necessidade de hardware físico e oferece uma plataforma robusta para o aprendizado prático de conceitos essenciais do curso. A abordagem permite que estudantes desenvolvam e validem lógicas de controle, explorem a programação de sistemas embarcados e compreendam a implementação de protocolos de comunicação industrial, como o Modbus, de forma acessível e segura, fortalecendo a formação de engenheiros mais preparados para os desafios tecnológicos contemporâneos.

Do ponto de vista tecnológico e social, a proposta ganha destaque ao integrar-se ao conceito de domótica, que visa automatizar práticas cotidianas para aumentar o conforto, a segurança e a eficiência. Um sistema de bombeamento automatizado, como o desenvolvido neste trabalho, não apenas previne o desperdício de água e energia, mas também aumenta a vida útil dos componentes ao evitar que a bomba opere a seco. Ao ser implementado em conjunto com cisternas para captação de água da chuva, o sistema potencializa a autonomia hídrica de residências e pequenas comunidades, alinhando-se a iniciativas de sucesso já consolidadas no Brasil e contribuindo para a resiliência frente a períodos de estiagem.

## 1.2 Materiais e Métodos

O sistema proposto foi concebido para gerenciar um conjunto de duas caixas d'água, designadas como inferior e superior. Ambas são equipadas com sensores para monitorar o nível de água em tempo real.

Para garantir a funcionalidade e a eficiência do projeto, o sistema foi desenvolvido para atender aos seguintes requisitos operacionais:

1. Modos de Operação: O sistema deve operar em dois modos distintos: manual e automático;
2. Interface de Visualização: É imprescindível que o sistema possua uma interface que permita ao usuário visualizar claramente os níveis de água em ambas as caixas;
3. Configuração de Níveis Mínimos: No modo automático, o usuário deve ser capaz de configurar e ajustar os setpoints de nível mínimo para cada uma das caixas d'água;

4. Lógica de Controle da Bomba: A bomba de água, no modo automático, só deve ser acionada quando os níveis em ambos os reservatórios estiverem de acordo com os parâmetros previamente configurados pelo usuário.

Para a implementação e simulação do sistema em um ambiente totalmente virtual, foram empregadas de ferramentas de software, cujas funções e integrações são detalhadas ao longo das próximas seções.

## 1.3 PICSimLab: Um Laboratório Virtual Completo para Simulação de Microcontroladores

O PICSimLab é um emulador de placas de desenvolvimento em tempo real que se destaca como uma ferramenta versátil e poderosa para estudantes, hobistas e profissionais da área de sistemas embarcados. Criado com o objetivo de facilitar o aprendizado e o desenvolvimento de projetos, o software permite a simulação de uma vasta gama de microcontroladores e componentes eletrônicos, oferecendo um ambiente virtual para testes e depuração antes da montagem física do hardware.

Com uma interface gráfica intuitiva, o PICSimLab possibilita a interação direta com as placas de desenvolvimento simuladas, permitindo o acionamento de chaves, a visualização de LEDs e displays, e a interação com diversos periféricos. Sua capacidade de integração com ambientes de desenvolvimento integrado (IDEs) populares, como o MPLAB X, Arduino IDE e VSCode (com PlatformIO), o torna uma ferramenta ainda mais robusta, permitindo que o código seja escrito, compilado e depurado em um fluxo de trabalho contínuo.

A utilização do PICSimLab é relativamente simples. O usuário primeiramente desenvolve o código para o microcontrolador de sua escolha em sua IDE de preferência. Após a compilação, o arquivo hexadecimal (.hex) gerado é carregado no PICSimLab. A partir daí, a simulação é iniciada, e o comportamento do hardware virtual pode ser observado e depurado em tempo real. A integração com as IDEs permite, em muitos casos, a depuração diretamente no código-fonte, facilitando a identificação e correção de falhas.

O grande atrativo do PICSimLab reside na sua extensa biblioteca de componentes e placas simuláveis. É possível simular desde microcontroladores clássicos até os mais modernos, incluindo:

- Microcontroladores;
  - Família PIC (PIC16F, PIC18F, entre outros)
  - Placas Arduino (Uno, Mega, Nano)

- Módulos ESP32
- Microcontroladores STM32
- Componentes e Periféricos;
  - Entradas: Botões (push-buttons), chaves, teclados matriciais.
  - Saídas: LEDs de diversas cores, displays de 7 segmentos, displays LCD (alfanuméricos e gráficos), buzzers.
  - Sensores: Sensores de temperatura, acelerômetros (como o ADXL345), e outros.
  - Módulos de Comunicação: Shields Ethernet (como o W5500) para simulação de conectividade de rede.
  - Outros: Memórias EEPROM, conversores analógico-digitais (ADC), motores de passo e outros atuadores

Além disso, o PICSimLab oferece placas de desenvolvimento virtuais que replicam modelos comerciais populares, bem como breadboards (matrizes de contato) que permitem a montagem de circuitos personalizados.

O PICSimLab oferece suporte a diversos protocolos de comunicação, permitindo a simulação de interações complexas entre o microcontrolador e outros dispositivos. Os principais protocolos que podem ser utilizados nas simulações são:

- UART (Universal Asynchronous Receiver-Transmitter): Essencial para a comunicação serial, frequentemente utilizada para depuração e comunicação com computadores ou outros módulos.
- I<sup>2</sup>C (Inter-Integrated Circuit): Um protocolo de comunicação serial síncrona amplamente utilizado para a comunicação com uma vasta gama de sensores, memórias e outros periféricos.
- SPI (Serial Peripheral Interface): Outro protocolo de comunicação serial síncrona, conhecido por sua alta velocidade e utilizado em aplicações que demandam rápida transferência de dados, como displays gráficos e cartões de memória.
- RS232: Um padrão de comunicação serial mais antigo, mas ainda relevante em muitas aplicações industriais e de automação, também pode ser simulado.

Neste trabalho foi utilizado o PIC18F4520 que pertence à família PIC18, ele se destaca pela sua performance computacional e um conjunto de instruções otimizado para compiladores em linguagem C. Ele é baseado em uma arquitetura Harvard RISC (Reduced Instruction Set Computer) de 8 bits. As principais características de sua arquitetura são:

- **Arquitetura Harvard:** Possui barramentos separados para dados e instruções, o que permite o acesso simultâneo a ambos e resulta em um processamento mais rápido. O barramento de instruções é de 16 bits, enquanto o de dados é de 8 bits
- **Conjunto de Instruções Otimizado:** Oferece um conjunto de instruções estendido, opcionalmente, que é otimizado para código reentrante e compiladores em linguagem C, tornando a programação de alto nível mais eficiente.
- **Velocidade de Processamento:** Pode operar com um clock de até 40 MHz, o que, com sua arquitetura, permite atingir uma performance de até 10 MIPS (Milhões de Instruções por Segundo), já que uma instrução é executada a cada quatro ciclos de clock.
- **Multiplicador de Hardware:** Inclui um multiplicador de hardware de 8x8, que acelera operações matemáticas, um recurso importante em algoritmos de processamento de sinais e controle.

## 1.4 MPLAB X IDE: A Plataforma Central para o Desenvolvimento com Microcontroladores Microchip

O MPLAB X IDE é o Ambiente de Desenvolvimento Integrado (IDE) oficial e gratuito da Microchip Technology. Trata-se de uma plataforma de software robusta e multiplataforma (compatível com Windows, macOS e Linux) que serve como o principal ecossistema para a criação de projetos com a vasta gama de microcontroladores (MCUs) e controladores de sinais digitais (DSCs) da empresa, incluindo as populares famílias PIC®, AVR® e SAM.

Baseado na plataforma open-source NetBeans, o MPLAB X oferece uma interface moderna e um conjunto completo de ferramentas que integram todas as fases do desenvolvimento de um sistema embarcado.

O MPLAB X IDE é utilizado para gerenciar todo o ciclo de vida de um projeto de firmware, desde a escrita do código até a programação do dispositivo físico. Suas principais funcionalidades incluem:

- **Edição de Código:** Possui um editor de texto inteligente com recursos como auto-completar, coloração de sintaxe, navegação por código e detecção de erros em tempo real, agilizando a escrita de código em Assembly, C ou C++.
- **Gerenciamento de Projetos:** Permite organizar múltiplos arquivos, configurações de hardware e versões de compiladores dentro de um mesmo projeto de forma estruturada.

- **Simulação:** Inclui um poderoso simulador de software que permite testar e depurar o código sem a necessidade do hardware físico. É possível simular o comportamento dos pinos de I/O, periféricos internos e até mesmo injetar estímulos para verificar a resposta do programa.
- **Depuração (Debugging):** É a sua função mais crítica. Em conjunto com ferramentas de hardware como os programadores/depuradores MPLAB PICKIT™, ICD ou JTAG ICE, o MPLAB X permite depurar o código em tempo real, diretamente no microcontrolador. O desenvolvedor pode executar o código passo a passo, inspecionar variáveis e registros, e definir breakpoints para analisar o comportamento do firmware em execução no hardware real.
- **Programação de Dispositivos:** Integra-se perfeitamente com as ferramentas de gravação da Microchip para programar o arquivo compilado (.hex) na memória do microcontrolador.

O MPLAB X IDE foi projetado para funcionar de forma integrada com a família de compiladores MPLAB XC da própria Microchip. Esses compiladores são otimizados para gerar um código altamente eficiente para cada arquitetura de microcontrolador. A escolha do compilador depende da família do dispositivo que você está utilizando:

- **MPLAB XC8:** É o compilador para todos os microcontroladores de 8 bits, o que inclui as famílias PIC10, PIC12, PIC16, PIC18 e também os microcontroladores AVR®.
- **MPLAB XC16:** Destina-se aos microcontroladores de 16 bits, que englobam as famílias PIC24 e os controladores de sinais digitais dsPIC®.
- **MPLAB XC32/XC32++:** É o compilador C/C++ para os microcontroladores de 32 bits, como as famílias PIC32 e os MCUs baseados em ARM®, como a série SAM

Além dos compiladores XC, o MPLAB X também suporta os assemblers (montadores) da Microchip, que vêm inclusos nos pacotes dos compiladores XC (como o mpasmx ou o mais novo pic-as). Para projetos com microcontroladores baseados em ARM, o IDE também pode ser configurado para utilizar a toolchain GNU ARM Embedded.

Cada compilador XC possui uma versão gratuita (Free) que oferece otimizações básicas e todas as funcionalidades necessárias para a maioria dos projetos. Existem também licenças pagas (PRO) que habilitam níveis de otimização de código mais avançados, resultando em um firmware menor e mais rápido, algo crucial para aplicações comerciais e com restrições de recursos.

Neste trabalho foi utilizado o compilador XC8 v2.31 que é uma versão específica do compilador C da Microchip para seus microcontroladores de 8 bits (famílias PIC® e AVR®). Lançada no final de 2021, esta versão faz parte de uma fase de maturação do compilador após a grande mudança arquitetônica introduzida na versão 2.0.

## 1.5 Protocolo Modbus: O Padrão da Comunicação Industrial

O Modbus é um dos protocolos de comunicação mais difundidos e utilizados no mundo da automação industrial e de sistemas de controle. Criado em 1979 pela Modicon (hoje parte da Schneider Electric), ele foi desenvolvido como uma forma simples e eficaz de permitir que dispositivos eletrônicos trocassem informações entre si. Apesar de sua idade, sua robustez, simplicidade e natureza aberta garantiram sua relevância até os dias de hoje, sendo um pilar na comunicação em chão de fábrica.

O Modbus é um protocolo de camada de aplicação, o que significa que ele define as regras para a estrutura das mensagens e a forma como os dados são trocados, independentemente do meio físico por onde a comunicação ocorre. Ele foi projetado para ser um protocolo de solicitação-resposta e opera em um modelo Mestre-Escravo (ou mais modernamente, Cliente-Servidor).

Sua aplicação é vasta, sendo encontrado em praticamente todos os segmentos que envolvem automação:

- Automação Industrial: É o seu principal campo de atuação. Conecta Controladores Lógicos Programáveis (CLPs), Inversores de Frequência, Interfaces Homem-Máquina (IHMs), sensores, atuadores e outros dispositivos em um mesmo sistema de controle.
- Automação Predial: Utilizado em sistemas de aquecimento, ventilação e ar-condicionado (HVAC), controle de iluminação e sistemas de segurança.
- Monitoramento de Energia: Empregado para ler dados de medidores de energia, relés de proteção e gerenciadores de consumo.
- Saneamento: Controle de bombas, medição de vazão e monitoramento de reservatórios em estações de tratamento de água e esgoto.

O funcionamento do Modbus é baseado em uma hierarquia. Em uma rede, existe um único dispositivo Mestre (Cliente) e um ou vários dispositivos Escravos (Servidores).

1. A Iniciativa é do Mestre: A comunicação sempre começa com o Mestre. Ele envia uma mensagem (uma requisição) para um Escravo específico, utilizando seu endereço único na rede;

2. A Tarefa do Escravo: O Escravo endereçado recebe a mensagem, interpreta a requisição, executa a ação solicitada e, em seguida, envia uma mensagem de resposta de volta ao Mestre. Os outros Escravos na rede ignoram a mensagem que não foi endereçada a eles.
3. Ações Possíveis: As requisições do Mestre são para acessar os dados do Escravo, que são organizados em quatro tipos básicos de registradores:
  - Coils (Bobinas): Leitura e escrita de valores digitais (ligado/desligado).
  - Discrete Inputs (Entradas Discretas): Leitura de valores digitais (apenas leitura).
  - Holding Registers (Registradores de Retenção): Leitura e escrita de dados de 16 bits (valores analógicos, configurações, etc.).
  - Input Registers (Registradores de Entrada): Leitura de dados de 16 bits (apenas leitura).

A estrutura da mensagem Modbus é simples e bem definida, contendo geralmente:

- Endereço do Escravo: Para quem a mensagem se destina (1 a 247).
- Código da Função: O que o Escravo deve fazer (ex: "Ler Holding Registers").
- Dados: Informações necessárias para a função (ex: endereço do registrador a ser lido e a quantidade)
- Verificação de Erro (Checksum): Um cálculo para garantir que a mensagem não foi corrompida durante a transmissão.

Existem três variações (ou modos) principais do protocolo:

1. Modbus RTU (Remote Terminal Unit): É a implementação mais comum. Utiliza comunicação serial (tipicamente sobre os padrões físicos RS-485 ou RS-232) e transmite os dados em um formato binário compacto, o que a torna eficiente. A verificação de erro é feita por CRC (Cyclic Redundancy Check).
2. Modbus TCP/IP: É a versão do Modbus para uso em redes Ethernet. As mensagens Modbus são encapsuladas dentro de pacotes TCP/IP. Isso permite que o protocolo utilize a infraestrutura de redes locais (LANs) e até a internet. Ele não precisa de um checksum próprio, pois a camada TCP já garante a integridade dos dados.
3. Modbus ASCII: Também para comunicação serial, mas os dados são transmitidos como caracteres ASCII legíveis. É menos eficiente e mais lento que o RTU, sendo raramente utilizado em novas aplicações.

A longevidade e a popularidade do Modbus se devem a um conjunto de vantagens claras e objetivas:

- **Simplicidade:** É um protocolo extremamente fácil de entender, implementar e diagnosticar. Sua estrutura lógica e direta reduz o tempo de desenvolvimento e facilita a manutenção.
- **Protocolo Aberto e Livre:** O Modbus é um padrão aberto, de domínio público e livre de royalties. Qualquer fabricante pode implementá-lo em seus equipamentos sem custo de licenciamento.
- **Baixo Custo:** A simplicidade do protocolo se traduz em um menor requisito de hardware (processamento e memória) nos dispositivos, tornando-os mais baratos.
- **Alta Interoperabilidade:** Por ser um padrão de fato no mercado, ele garante que dispositivos de diferentes fabricantes possam se comunicar sem problemas, dando flexibilidade ao projeto.
- **Flexibilidade de Meios Físicos:** Funciona sobre diferentes infraestruturas, desde um simples par de fios trançados (RS-485) até redes Ethernet complexas (TCP/IP), adaptando-se a diversos cenários.
- **Robustez e Confiabilidade:** É um protocolo testado e comprovado por décadas de uso em ambientes industriais severos, o que o torna extremamente confiável para aplicações críticas.

Neste trabalho será utilizado as bibliotecas Easymodbus<sup>1</sup> e NModbus<sup>2</sup>, que são implementadas em C Sharp e que será utilizada pelo supervisorio. Já no sistema embarcado será utilizada a biblioteca Modbus Slave For PIC18F<sup>3</sup>, implementada em C.

## 1.6 Virtual Serial Ports Emulator (VSPE)

O Virtual Serial Ports Emulator (VSPE), desenvolvido pela Eterlogic, é um software utilitário para Windows projetado para criar e gerenciar uma variedade de dispositivos seriais virtuais. Sua função principal é criar portas de comunicação COM virtuais que podem ser interligadas de diversas formas, permitindo a comunicação entre aplicações de software que precisam de uma porta serial, mesmo sem a existência de uma porta física no computador.

---

<sup>1</sup> [Rossmann-Engineering e Cheng \(s.d., EasyModBus - Acesso em 20 de Agosto de 2025\)](#).

<sup>2</sup> [Turin e Brey \(s.d., NModbus - Acesso em 20 de Agosto de 2025\)](#).

<sup>3</sup> [Walsh \(s.d., EasyModBus - Acesso em 20 de Agosto de 2025\)](#).

Em essência, o VSPE atua como um “encanador” digital para dados seriais, permitindo que você conecte a “saída” de um programa à “entrada” de outro de maneira virtual e flexível.

O VSPE permite criar diferentes tipos de dispositivos virtuais, sendo os mais comuns:

- **Connector:** Cria um par de portas COM virtuais interligadas (ex: COM5 e COM6). Tudo o que um programa escreve na COM5 aparece na COM6, e vice-versa. É como ter um cabo serial “nulo”(null-modem) dentro do próprio computador.
- **Splitter:** Permite que múltiplos programas acessem os dados de uma única porta serial física. Os dados que chegam à porta física são “divididos” e enviados para todas as aplicações conectadas a uma porta virtual.
- **Pair:** Semelhante ao Connector, cria um par de portas virtuais para interligar duas aplicações.
- **TCP/UDP Server/Client:** Permite que uma porta COM (física ou virtual) se comunique através de uma rede TCP/IP. Isso possibilita, por exemplo, que um software em um computador acesse um dispositivo serial conectado a outro computador na rede.

A combinação do PICSimLab com o VSPE é extremamente poderosa para o desenvolvimento e depuração de projetos embarcados que envolvem comunicação serial (UART/RS232). Ela permite simular o cenário completo de um microcontrolador se comunicando com um software no PC, sem a necessidade de nenhum hardware físico (nem o microcontrolador, nem o conversor USB-Serial).

O PICSimLab possui um periférico chamado “UART/USART” que pode ser configurado para se conectar a uma porta serial do computador. É aqui que o VSPE entra em cena para criar a “ponte” virtual.

## 1.7 Construção do supervisor

O Visual Studio Community<sup>4</sup> é um Ambiente de Desenvolvimento Integrado completo (IDE), moderno e gratuito da Microsoft. Ele oferece um conjunto robusto de ferramentas para a criação de aplicações para desktop, web, nuvem, dispositivos móveis e jogos. Para o mundo da automação e engenharia, ele se destaca como uma plataforma excepcional para a construção de sistemas supervisórios customizados, principalmente através de sua tecnologia de formulários. Com ele é possível criar aplicações desktop para

<sup>4</sup> [VISUAL...](#) (s.d., Visual Studio Community - Acesso em 20 de Agosto de 2025).

Windows de forma visual e intuitiva usando a tecnologia Windows Forms (WinForms). Este é um framework que permite arrastar e soltar componentes visuais em um formulário, construindo a interface gráfica do seu supervisor de maneira rápida e eficiente. A linguagem de programação mais comumente utilizada para isso é o C# (C Sharp), que é poderosa, versátil e relativamente fácil de aprender.

## 1.8 Objetivos

- Objetivo geral: Desenvolver um sistema de controle de nível para caixas d'água, por meio de ferramentas de simulação, com a finalidade de promover um sistema que auxilie no uso mais eficiente e racional da água.
- Objetivos específico:
  - Desenvolver uma interface de visualização do sistema;
  - Desenvolver uma lógica de proteção para evitar que a bomba trabalhe a seco.

## 2 Revisão de literatura

### 2.1 Profissional de engenharia de controle e automação

O Engenheiro de Controle e Automação é um profissional que atua diretamente na área tecnológica, sujeita à rápidas mudanças e, por conta disso, necessita estar em constante atualização para acompanhar a evolução científica e tecnológica concernente à área de atuação ([HAHN, 2021](#)). Para [Kamp \(2019\)](#), as universidades têm um papel importante no preparo de seus alunos, que não poderá ser apenas imediatista para um primeiro emprego, mas de preparar para o aprendizado e o trabalho durante a vida toda. Conforme o autor, os graduados não devem apenas aprender a lidar com o mundo em mudança, mas sendo eles próprios agentes de transformação, desafiando-se a desenvolver habilidades e mentalidades para prosperar no mundo da Indústria 4.0 que irá evoluir para uma 'Sociedade 5.0', onde o conhecimento não será apenas criado por humanos, mas cada vez mais por algoritmos em máquinas inteligentes, movidas a sensores remotos e uma abundância de dados ([HAHN, 2021](#)).

Conforme dito por [Pascarelli e Nascimento-e-Silva \(2020\)](#) um dos conhecimentos exigidos do Engenheiro de Controle e Automação pelo mercado de trabalho é a modelagem de sistemas automatizados. Esse conhecimento consiste na representação, física ou simbolicamente, de um sistema ou parte dele. A modelagem é necessária para que testes de funcionamento e adequações sejam feitos antes que determinada solução seja colocada em prática. Como consequência, esse procedimento reduz as possibilidades de erros, falhas e desconformidades ao permitir que elas sejam detectadas antes da implementação. Em linha com essa competência profissional, o sistema de bombeamento proposto neste trabalho foi modelado utilizando técnicas de virtualização e simulação. A criação de um protótipo inteiramente virtual, empregando ferramentas como o PICSimLab e o MPLAB X IDE, permitiu que todos os testes de funcionamento e adequações na lógica de controle fossem realizados de forma segura e eficiente, exemplificando a aplicação prática da modelagem no ciclo de desenvolvimento de um sistema de automação.

A área de sistemas embarcados constitui um campo de atuação fundamental para o profissional de Engenharia de Controle e Automação. Nesse contexto, o estudo de ([SILVA et al., 2024](#)) revela que o desenvolvimento e a depuração de firmware e software são as atividades mais recorrentes para os profissionais do setor. Tal constatação evidencia a necessidade de que a formação acadêmica, para atender às demandas do mercado de trabalho, seja direcionada prioritariamente ao aprimoramento dessas competências.

## 2.2 Sistemas embarcados

Um sistema embarcado, também conhecido como sistema embutido, é um sistema computacional projetado para executar uma tarefa específica como parte de um sistema maior. Diferente de um computador de propósito geral, como um *desktop*, que é projetado para uma variedade de tarefas e para ser programado pelo usuário final, um sistema embarcado é dedicado a uma função ou a um conjunto limitado de funções.

Esses sistemas são compostos pela união de hardware e software. O hardware é o núcleo do sistema e geralmente consiste em um microcontrolador ou microprocessador, memória e periféricos de entrada e saída. O software, conhecido como firmware, é o conjunto de instruções armazenado na memória que define o comportamento do sistema.

Para [Braun et al. \(2014\)](#), um sistema embarcado é um sistema técnico que opera em um ambiente físico, e tem como objetivo controlar seu ambiente utilizando-se de variáveis que fazem referências as propriedades físicas ou técnicas do ambiente.

[Rahman \(2014\)](#) indica que sistemas embarcados são um conjunto de hardware e software, que determina a funcionalidade de microprocessadores e outros dispositivos programáveis que são utilizados para controlar equipamentos eletrônicos, elétricos e eletromecânicos e sub-sistemas.

[Ossada et al. \(2012\)](#) definem sistemas embarcados como sistemas com a finalidade de controlar um ambiente ou dispositivo físico, diferindo-se em diversos aspectos de um computador para propósitos gerais.

Conforme [Sousa et al. \(2015\)](#), dada a variedade de aplicações para sistemas embarcados, não existe um consenso na conceitualização, mas existe um conjunto de características que são comuns a esses sistemas, com isso ele define um sistema embarcado como sendo um firmware, que está acoplado a um dispositivo físico e que realiza tarefas específicas diferentes de um sistema de propósito geral.

## 2.3 Utilização de laboratórios virtuais durante a pandemia da SARS-CoV-2

A pandemia global da COVID-19, uma doença respiratória aguda causada pelo coronavírus SARS-CoV-2, foi oficialmente declarada pela Organização Mundial da Saúde (OMS) em 11 de março de 2020. Caracterizada pela alta transmissibilidade e mortalidade significativa, a doença impôs a necessidade de medidas de isolamento social em escala global, o que impactou profundamente todos os setores da sociedade, incluindo a educação superior e, conseqüentemente, as atividades laboratoriais práticas.

Se nos laboratórios físicos de sistemas embarcados os estudantes tinham acesso à

diversas plantas didáticas e contato direto com hardwares previamente adquiridos pelas instituições, com a pandemia essa interação ficou limitada ao computador pessoal do aluno. Já os laboratórios virtuais utilizam uma abordagem puramente via software para realizar a simulação da dinâmica de diversos tipos de plataformas físicas (PARREIRA et al., 2021).

Sobre a eficiência na utilização dos laboratórios virtuais nos cursos de ensino superior, temos alguns estudos como em (NICOLETE et al., 2021) que apontam bons resultados no processo de ensino-aprendizado, uma vez que a ferramenta levou, por exemplo, à diminuição dos altos custos, assim como à criação e manutenção de novos espaços de laboratórios físicos. As práticas telepresenciais favoreceram ainda os níveis de motivação entre discentes em épocas de distanciamento. No estudo de (SERRANO; MARTINEZ; CLAVIJO, 2021), estudantes do segundo ano de Engenharia de uma universidade pública da Argentina realizaram tarefas de diagnóstico usando laboratório online, com as experiências desenvolvidas com recursos virtuais e seu uso em condições de não-presencialidade nas aulas promoveu aprendizagens significativas.

Em 5 de maio de 2023, a OMS declarou o fim da emergência de saúde pública de importância internacional para a COVID-19, marcando o início de uma nova fase de convivência com o vírus. A normalização das atividades ocorreu de forma gradual, com a flexibilização das restrições de distanciamento, o retorno ao trabalho presencial e, notadamente, a retomada completa das aulas em formato presencial nas instituições de ensino. Embora a pandemia tenha terminado, as lições aprendidas e as tecnologias adotadas durante o período, como os laboratórios virtuais, consolidaram-se como ferramentas valiosas e permanentes no processo de ensino-aprendizagem, oferecendo modelos híbridos e mais flexíveis para a educação.

## 2.4 Utilização de cisternas no Brasil

As cisternas são reservatórios destinados à captação e armazenamento de água, principalmente da chuva. Funcionando como um sistema de aproveitamento hídrico, elas coletam a água que escorre de telhados e outras superfícies por meio de calhas e a direcionam para um compartimento fechado, onde fica protegida de contaminação e evaporação. A água armazenada, por não ser potável, é ideal para fins não-domésticos, como a irrigação de plantas, lavagem de pisos e carros, e descargas sanitárias, representando uma economia significativa no consumo de água tratada.

No Brasil, o uso de cisternas é uma realidade consolidada e de extrema importância, especialmente em regiões que sofrem com a escassez de água, como o Semiárido nordestino. Nessas áreas, a cisterna transcende a função de simples reservatório e se torna um elemento fundamental para a segurança hídrica e a qualidade de vida de milhões de famílias.

Segundo o estudo de Lima et al. (2016) o programa “Um Milhão de Cisternas

Rurais” (P1MC), iniciado em 2003, é um destaque, pois por meio da articulação no Semi-Árido Brasileiro, um fórum que reúne ONGs e entidades da sociedade civil, o projeto construiu mais de 294 mil cisternas e beneficiou mais de 313 mil famílias.

O P1MC contabiliza aproximadamente 300 mil cisternas construídas, beneficiando diretamente 300 mil famílias, ou mais, já que, em alguns casos, uma cisterna atende a mais de uma família, e com um índice de aceitação pelas famílias contempladas de 97,31%. Este programa ofertou às famílias rurais do Semiárido brasileiro um volume correspondendo a 4,8 bilhões de litros de água, superior à capacidade de armazenamento dos açudes São José do Egito II - 4.6 bilhões de litros de água - e o açude Araripina - 3,7 bilhões de litros de água - ambos no estado de Pernambuco (Lima Brito e Araújo (2011)).

Neste cenário de aproveitamento hídrico, o sistema de controle desenvolvido neste trabalho se apresenta como uma solução tecnológica complementar e de grande valor. O protótipo, que gerencia o bombeamento entre um tanque inferior e um superior, pode ser diretamente aplicado em residências que utilizam cisternas. Nessa aplicação, a cisterna de captação de água da chuva atuaria como o reservatório inferior (tanque 1), enquanto a caixa d’água principal da casa funcionaria como o reservatório superior (tanque 2). A principal contribuição do sistema é a automação inteligente do processo, que não só mantém o reservatório superior abastecido, mas também implementa uma lógica de proteção essencial: a bomba só é acionada se houver água suficiente na cisterna, evitando que o equipamento trabalhe a seco e sofra danos. Dessa forma, o projeto agrega eficiência, segurança e otimiza a gestão da água armazenada, potencializando os benefícios do uso de cisternas.

## 2.5 Domótica no Brasil

Segundo Lima Brito e Araújo (2011), a palavra “domótica” resulta da junção da palavra latina “Domus”, que significa casa, com “Robótica”, que pode ser entendido como controle automatizado de algum processo ou equipamento.

Desta maneira, podemos entender a domótica como sendo a integração de tecnologias em uma residência para automatizar e controlar processos, visando aumentar o conforto, a segurança, a eficiência energética e a comunicação. No Brasil, o que antes era visto como um luxo futurista, tem se tornado uma realidade cada vez mais presente e acessível, transformando a maneira como os brasileiros interagem com suas casas.

Basicamente, a domótica conecta os mais variados sistemas de uma casa – iluminação, climatização, segurança, entretenimento e eletrodomésticos – a uma central de controle. Essa central pode ser acessada remotamente por meio de smartphones ou tablets, permitindo que o morador gerencie seu lar de qualquer lugar do mundo.

Quando se fala em domótica, há consenso em dizer que o que se busca é comodidade,

melhor gestão e acessibilidade. Acessibilidade é um tema que vem sendo abordado na sociedade contemporânea e começa a ser abordada na legislação dos países, principalmente para os portadores de deficiência, enfermos, idosos e crianças. Dessa forma a domótica pode melhorar muito a qualidade de vida e também facilitar a gestão do local domotizado por meio de interfaces com smartphones, tablets e outros equipamentos eletrônicos. Dessa forma a domótica terá cada vez mais espaço nas economias pelos benefícios que pode proporcionar e com a evolução contínua da tecnologia, cada vez mais haverá possibilidade de integração e gestão por meio da domótica ([LIMA BRITO; ARAÚJO, 2011](#)).

Nesse contexto, o sistema de controle de bombeamento desenvolvido neste trabalho se enquadra como uma aplicação prática de domótica. O projeto incorpora seus elementos fundamentais ao automatizar uma tarefa residencial — o gerenciamento de caixas d'água — por meio de um sistema embarcado que utiliza sensores e uma lógica de controle para tomar decisões. Ao visar o aumento da eficiência, a economia de recursos hídricos e a proteção de equipamentos, a solução se alinha diretamente aos objetivos de conforto, segurança e melhor gestão que caracterizam a domótica no ambiente residencial.

### 3 Desenvolvimento

A validação do sistema de controle de bombeamento foi realizada inteiramente no ambiente de simulação, onde a interação entre a interface de supervisão e o microcontrolador PIC18F4520, emulado no PICSimLab, pôde ser observada em tempo real. A configuração inicial da simulação, apresentada na figura 1, destaca os componentes virtuais essenciais para a operação: o display LCD, que exibe o estado atual do sistema; os potenciômetros, que simulam os sensores de nível dos tanques; e o relé 1, responsável pelo acionamento da bomba.

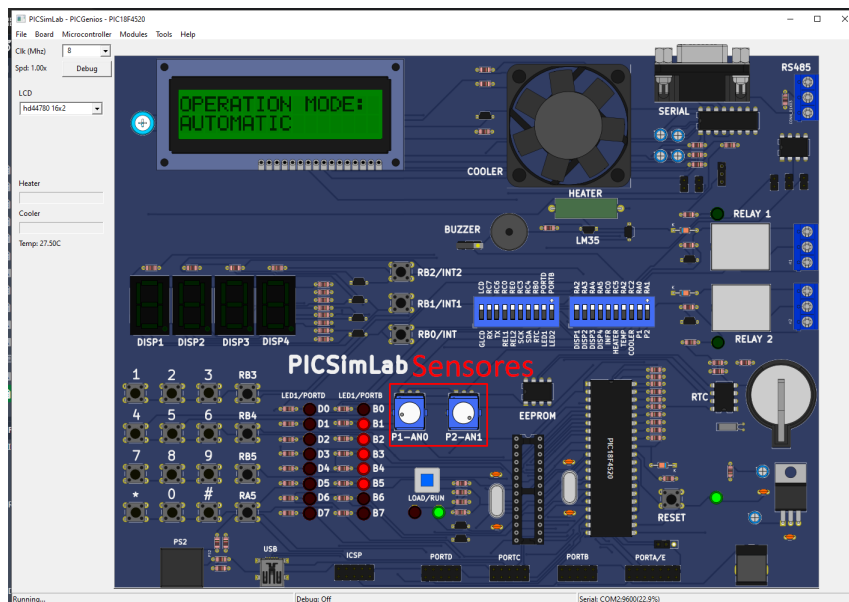


Figura 1 – Demonstração da placa e a indicação dos sensores de níveis dos tanques

Para representar o nível dos tanques de água, utilizaram-se técnicas de projeção de imagens, conforme as figuras 2, 3 e 4.

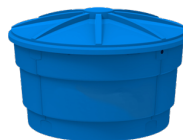


Figura 2 – Representação do tanque de água com 0% da capacidade

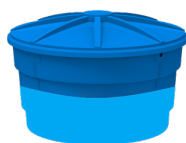


Figura 3 – Representação do tanque de água com 50% da capacidade



Figura 4 – Representação do tanque de água com 100% da capacidade

O VSPE foi utilizado para parear a COM2 (utilizada pelo PICSIMLAB) com a COM6 (utilizada pelo supervisor), conforme a figura 5.

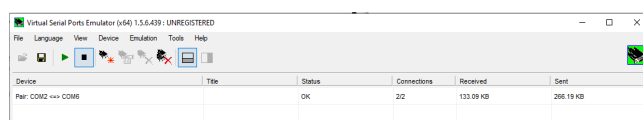


Figura 5 – Configuração do VSPE

No supervisor, o protocolo Modbus foi implementado a partir de uma rotina de leitura e escrita nos registradores, conforme os trechos de código apresentados nas figuras 6 e 7. Já no microcontrolador, foi utilizada a implementação demonstrada nas figuras 8 e 9.

```
private void Loop()
{
    Task.Run(async delegate
    {
        try
        {
            while (true)
            {
                ushort[] registers = _Master.ReadHoldingRegisters(_SlaveId, _StartAddress, _NumberOfPoints);
                for (int i = 0; i < registers.Length; i++)
                {
                    if (registers[i] != _Registers[i])
                    {
                        IMsgUart msg = CreateMsg((RegisterTypeByPosition)i, registers);
                        if (msg != null)
                        {
                            Program.UartService.TreatMessage(msg);
                        }
                    }
                }
                _Registers = registers;
                await Task.Delay(1000);
            }
        }
        catch (Exception e)
        {
            throw;
        }
    });
}
```

Figura 6 – ModBus - Leitura dos registradores

```

1 reference
public void WriteRegisters(IMsgUart msg)
{
    try
    {
        short startAddress = GetStartAddress(msg);
        if (startAddress != -1)
        {
            var propsInfo = msg.GetType().GetProperties();
            if ((propsInfo != null) && (propsInfo.Length > 0))
            {
                for (int i = 0; i < propsInfo.Length; i++)
                {
                    ushort data = Convert.ToInt16(propsInfo[i].GetValue(msg, null));
                    _Master.WriteSingleRegister(_SlaveId, (ushort)(startAddress + i), data);
                }
            }
        }
    }
    catch (Exception e)
    {
        throw;
    }
}

```

Figura 7 – ModBus - Escrita dos registradores

```

while(1)
{
    CLEARDT();

    last_pump_status_received = holdingReg[0];
    last_operation_mode_received = holdingReg[1];

    if (modbusMessage)
    {
        decodeIt();
        Receive_Message();
    }

    Read_Level();

    Debounce(PORTB, bt_operation_mode, &bt_operation_mode_press, &filter_bt_operation_mode, Change_Mode);

    if (operation == AUTOMATIC) Automatic_Mode(); else Manual_Mode();

    Debounce(PORTB, bt_incr_min_lv1, &bt_min_lv1_incr_press, &filter_bt_min_lv1_incr, Min_lv1_Incr);
    Debounce(PORTB, bt_incr_min_lv2, &bt_min_lv2_incr_press, &filter_bt_min_lv2_incr, Min_lv2_Incr);

    if (refresh_display) Refresh_Screen();
}
return;

```

Figura 8 – ModBus - Implementação no microcontrolador

```

void Read_Level()
{
    int adcTank1 = adc_amostra(0);
    int adcTank2 = adc_amostra(1);

    tank1.level = (adcTank1 / 10.23); // %
    tank2.level = (adcTank2 / 10.23); // %

    holdingReg[4] = tank1.level;
    holdingReg[5] = tank2.level;
}

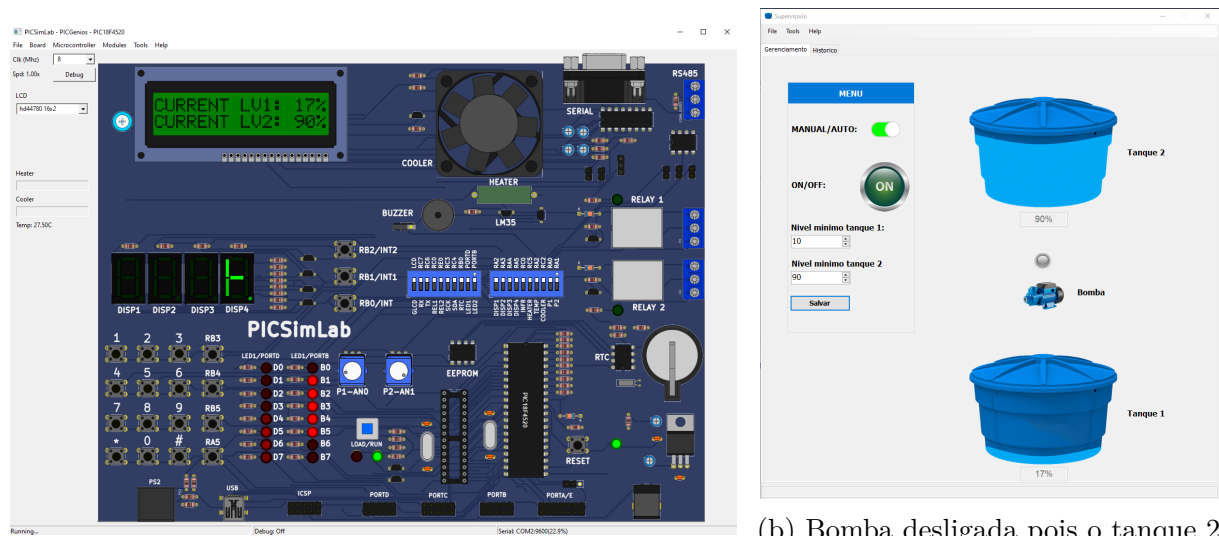
void Receive_Message()
{
    Sync_Registers();
    if (holdingReg[0] == 1) SetBit(&LATC, pump_relay);
    else ClearBit(&LATC, pump_relay);
    operation = holdingReg[1];
    tank1.min_level = holdingReg[2];
    tank2.min_level = holdingReg[3];
}

```

Figura 9 – ModBus - Implementação das funções de leitura dos registradores no microcontrolador

## 4 Resultados

O sistema foi projetado com dois modos de operação, manual e automático, configuráveis através da interface de supervisão, que também permite ao usuário definir os setpoints de nível mínimo para ambos os tanques (figura 10b). A Figura 10 ilustra um cenário de equilíbrio no modo automático.

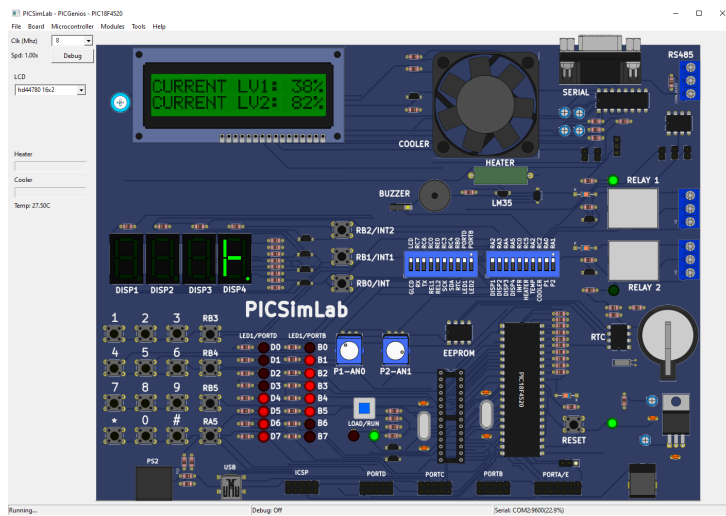


(a) Display da placa PICSimLab informando os níveis atuais dos tanques

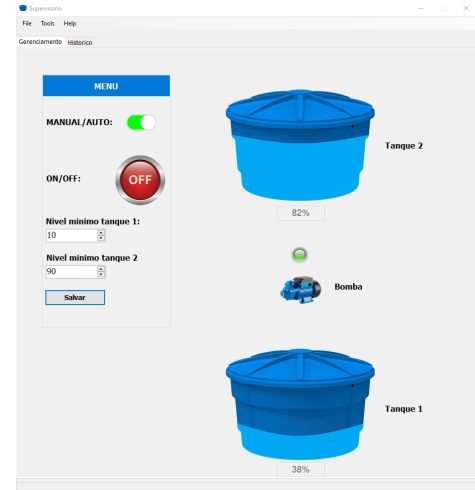
(b) Bomba desligada pois o tanque 2 esta com 90% de sua capacidade e seu valor mínimo é de 90%

Figura 10 – Demonstração do sistema em equilíbrio.

Conforme as figuras 11, 12 e 13, é possível observar que o sistema aciona a bomba de acordo com a necessidade e, caso o tanque 1 que representa uma cisterna, fique abaixo do valor mínimo, a bomba é desligada, evitando que a bomba trabalhe a seco.

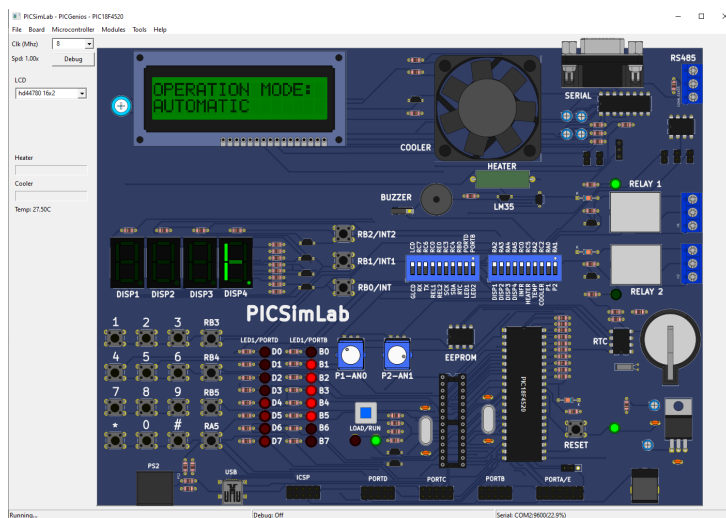


(a) Display da placa PICSimLab informando os níveis atuais dos tanques

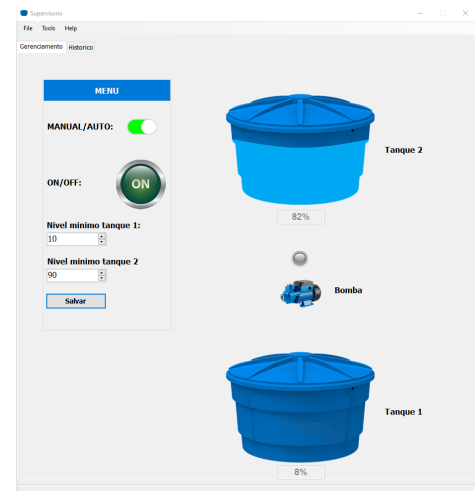


(b) Bomba ligada pois o tanque 2 esta com 82% de sua capacidade e seu valor mínimo é de 90%

Figura 11 – Sistema em equilíbrio por conta do tanque 2

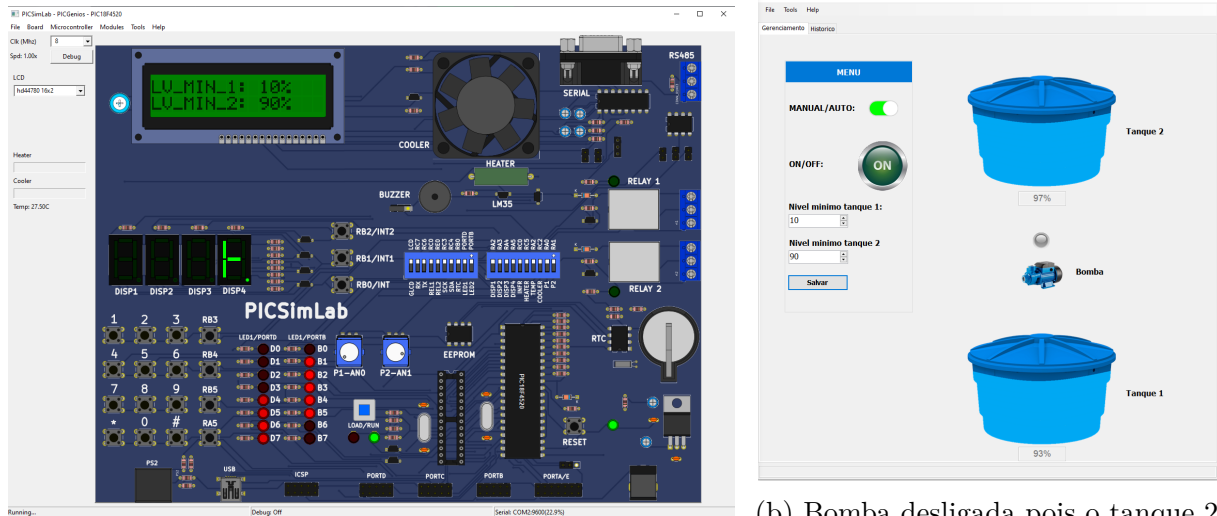


(a) Display da placa PICSimLab informando o modo de operação



(b) Bomba desligada pois o tanque 1 esta com 8% de sua capacidade e seu valor mínimo é de 10%

Figura 12 – Sistema em equilíbrio por conta do tanque 1



(a) Display da placa PICSimLab informando os níveis configurados

(b) Bomba desligada pois o tanque 2 esta com 97% de sua capacidade e seu valor mínimo é de 90%

Figura 13 – Sistema em equilíbrio

A eficácia da lógica de controle automático é demonstrada na Figura 11. Ao simular uma queda no nível do tanque 2 para 82%, valor inferior ao setpoint de 90%, o sistema respondeu prontamente. A interface de supervisão (Figura 11b) atualizou o status da bomba para "ligada"(ON), e o relé correspondente na placa virtual foi acionado, validando a comunicação via Modbus e a correta execução do firmware no microcontrolador.

Um dos requisitos críticos do projeto era a implementação de uma lógica de segurança para proteger a bomba contra o funcionamento a seco. A Figura 12 exibe a atuação bem-sucedida dessa proteção. Embora o tanque 2 estivesse com o nível baixo, o sistema identificou que o tanque 1, que representa a cisterna de captação, atingiu um nível de apenas 8%, abaixo do mínimo configurado de 10%. Como resultado, o controle desativou a bomba para evitar danos, demonstrando que a lógica de decisão considera as condições de ambos os reservatórios antes de autorizar o bombeamento. Finalmente, a Figura 13 mostra o sistema retornando a um estado de equilíbrio após o enchimento do tanque 2, que atingiu 97% de sua capacidade, levando ao desligamento automático da bomba, concluindo um ciclo de operação com sucesso.

## 5 Considerações finais

Com base nos resultados apresentados, a simulação do sistema de controle de bombeamento de água demonstrou ser eficaz, operando de forma confiável e atendendo a todos os requisitos funcionais estabelecidos. A lógica de controle, tanto no modo manual quanto no automático, funcionou conforme o esperado, com a interface de supervisão permitindo o monitoramento e a configuração dos parâmetros em tempo real, validando a eficácia da comunicação via protocolo Modbus.

A validação do sistema em ambiente virtual confirma que a arquitetura proposta é totalmente viável para ser aplicada em cenários reais, representando uma solução de baixo custo e alta eficiência para a melhoria da gestão de recursos hídricos. A capacidade de evitar que a bomba opere a seco, ao monitorar o nível do tanque inferior, e de manter o reservatório superior em um nível adequado, reforça seu potencial para economizar água e energia, além de aumentar a vida útil dos componentes.

Para trabalhos futuros, a evolução natural deste projeto seria a sua transição do ambiente virtual para o mundo real. Recomenda-se a construção de um protótipo físico, o que envolveria as seguintes etapas:

- Desenvolvimento de uma Placa de Circuito Impresso (PCB): Projetar e confeccionar uma PCB dedicada para abrigar o microcontrolador PIC18F4520, os relés de acionamento da bomba, os circuitos de interface para os sensores e a comunicação serial;
- Implementação com Sensores Reais: Substituir os potenciômetros da simulação por sensores de nível reais (como boias elétricas ou sensores ultrassônicos) para a medição nos tanques;
- Montagem e Testes em Bancada: Integrar todos os componentes de hardware e software em uma bancada de testes para validar na prática tudo o que foi simulado, aferindo o comportamento do sistema sob condições reais de operação.

A concretização dessas etapas permitiria não apenas a validação final do projeto, mas também a sua aplicação prática em residências, pequenas propriedades rurais ou em sistemas de captação de água da chuva, contribuindo efetivamente para o uso mais consciente e sustentável da água.

# Referências

- AMORIM, MCC de; PORTO, Everaldo Rocha. Avaliação da qualidade bacteriológica das águas de cisternas: estudo de caso no município de Petrolina-PE. *Anais do 3º Simpósio Brasileiro de Captação de Água de Chuva no Semi-Árido. Campina Grande-PB, ABCMAC*, 2001. Citado 2 vezes na página 8.
- BRAUN, Peter et al. Guiding requirements engineering for software-intensive embedded systems in the automotive industry: The REMsES approach. *Computer Science-Research and Development*, Springer, v. 29, n. 1, p. 21–43, 2014. Citado 1 vez na página 20.
- HAHN, Margarida. *A formação do Engenheiro de Controle e Automação*. 2021. Tese (Doutorado). Citado 2 vezes na página 19.
- HASAN, Mehedi et al. Smart home systems: Overview and comparative analysis. In: IEEE. 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN). 2018. P. 264–268. Citado 1 vez na página 8.
- KAMP, Aldert. Science & technology education for 21st century Europe. *Task Force on Science and Technology Education for the 21st Century*, 2019. Citado 1 vez na página 19.
- LIMA, Leiliane Silva Lopes et al. Utilização de cisternas pelas comunidades rurais no semiárido brasileiro. In: I CONGRESSO INTERNACIONAL DA DIVERSIDADE DO SEMIÁRIDO. 2016. Citado 1 vez na página 21.
- LIMA BRITO, Luiza Teixeira de; ARAÚJO, Janaína Oliveira de. 10883-Água de chuva armazenada em cisterna pode incrementar qualidade nutricional da dieta das famílias. *Cadernos de Agroecologia [Volumes 1 (2006) a 12 (2017)]*, v. 6, n. 2, 2011. Citado 3 vezes nas páginas 22, 23.
- NICOLETE, Priscila Cadorin et al. A motivação de estudantes ao utilizar laboratórios online para aprendizagem experiencial de circuitos elétricos durante a pandemia do Covid-19. *Revista Novas Tecnologias na Educação*, v. 19, n. 2, p. 152–162, 2021. Citado 1 vez na página 21.
- OSSADA, Jaime Cazuhiro et al. GERSE: Guia de Elicitação de Requisitos para Sistemas Embarcados. In: WER. 2012. Citado 1 vez na página 20.
- PARREIRA, Paulo Felipe Possa et al. Laboratório virtual de sistemas embarcados: uma abordagem de ensino remoto usando softwares gratuitos. In: 1. SIMPÓSIO Brasileiro de Automação Inteligente-SBAI. 2021. v. 1. Citado 2 vezes nas páginas 7, 21.
- PASCARELLI, Clara Victória Ferreira; NASCIMENTO-E-SILVA, Daniel. Conhecimentos do engenheiro de controle e automação exigidos pelo mercado. Brasil, 2020. Citado 1 vez na página 19.

PORTO, Antonio José Maristrello; PAVÃO, Bianca Borges Medeiros; NOGUEIRA, Rafaela. Regulação da água no Brasil: estratégias atuais, desafios e uma nova possibilidade regulatória. *Revista de Direito Econômico e Socioambiental*, v. 10, n. 2, p. 297–325, 2019. Citado 1 vez na página 8.

RAHMAN, Aedah Abd. Requirements engineering approach for real-time and embedded systems: a case study of android-based smart phone devices. In: PROCEEDINGS of the 8th International Conference on Ubiquitous Information Management and Communication. 2014. P. 1–5. Citado 1 vez na página 20.

RIBEIRO, Luiz Gustavo Gonçalves; ROLIM, Neide Duarte. Planeta água de quem e para quem: uma análise da água doce enquanto direito fundamental e sua valoração mercadológica. *Revista Direito Ambiental e sociedade*, v. 7, n. 1, 2017. Citado 2 vezes nas páginas 7, 8.

ROSSMANN-ENGINEERING; CHENG, Howard. *EasyModbusTCP.NET*. Disponível em: <https://github.com/rossmann-engineering/EasyModbusTCP.NET>. Citado 1 vez na página 16.

SERRANO, Graciela M; MARTINEZ, Carlos; CLAVIJO, Silvia. Empleo de recursos virtuales para la enseñanza de circuitos eléctricos en ciclo básico de Ingeniería: un estudio comparativo. *Revista de Enseñanza de la Física*, v. 33, n. 2, p. 487–494, 2021. Citado 1 vez na página 21.

SILVA, Taynan Roger et al. Abordagem de ensino remoto com softwares gratuitos em laboratório virtual de sistemas embarcados na UFOP. In: COSTA JUNIOR, Ademar Gonçalves da et al. (Ed.). *Práticas pedagógicas remotas em Engenharia: Fundamentos e estudos de casos*. 1. ed. São Paulo: Blucher, 2024. v. 1. P. 145–176. Citado 2 vezes nas páginas 7, 19.

SOUSA, Aêda et al. Elicitação e Especificação de Requisitos em Sistemas Embarcados: Uma Revisão Sistemática. *CIbSE*, p. 606, 2015. Citado 1 vez na página 20.

SOUZA, Érika Duarte de; ONESSEKEN, Maurici. *Qual o método para a escolha dos programas mais adequado no ensino de engenharia elétrica*. 2020. B.S. thesis – Universidade Tecnológica Federal do Paraná. Citado 1 vez na página 7.

TURIN, Dmytro; BREY, Edward. *NModbus*. Disponível em: <https://github.com/NModbus/NModbus>. Citado 1 vez na página 16.

VISUAL Studio Community. Disponível em: <https://visualstudio.microsoft.com/vs/community/>. Citado 1 vez na página 17.

WALSH, Sam. *Modbus-Slave-for-PIC18F*. Disponível em: <https://github.com/whchoi/Modbus-Slave-for-PIC18F>. Citado 1 vez na página 16.