

Barramentos Industriais

Projeto 02 – Dispositivo Modbus TCP

IFRS – Campus Farroupilha

Engenharia de Controle e Automação

Professor Gustavo Künzel Aluno Pedro Henrique de Assumpção



INSTITUTO FEDERAL
Rio Grande do Sul

Campus
Farroupilha

24/06/2025

ÍNDICE

- **Introdução**
- **Objetivos**
- **Justificativa**
- **Fundamentação Teórica**
- **Proposta**
- **Testes**
- **Desafios**
- **Referências**

Introdução

Utilizar conceitos de Modbus TCP e comunicação pela rede ethernet, para desenvolver uma aplicação compatível com o protocolo Modbus TCP.

O objetivo é compreender como deve ser feita a programação do protocolo nos dispositivos.

Objetivos

- **Módulo de saídas digitais (8 a 16 saídas):**
 - **Representadas por mensagens no ESP32 - Servidor;**
 - **Programa do PC - Client deve permitir envio de comando ao ESP32 - Servidor;**
 - **Função Modbus: 0x0F Write Multiple Coils.**

Justificativa

- **Desenvolver um sistema que permita o envio de comandos ao Servidor, a partir de uma conexão do Cliente, sendo o dispositivo ESP32 com placa de rede integrada para ser o Servidor. Além disso, busca-se compreender a implementação prática do protocolo e sua aplicabilidade em sistemas embarcados.**

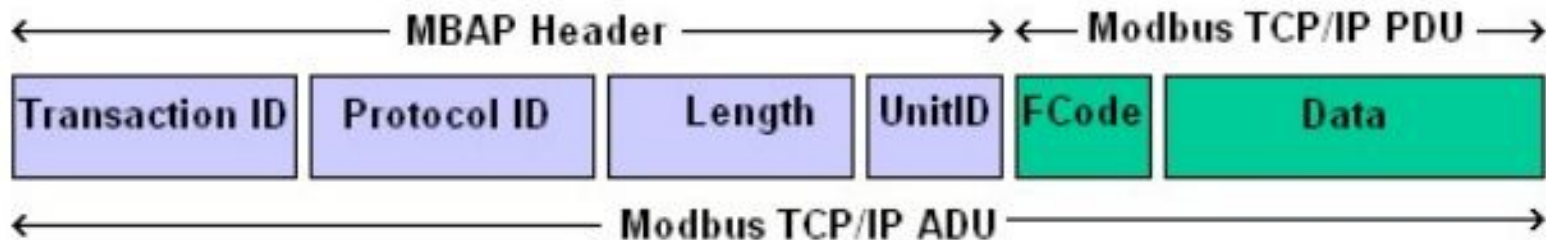
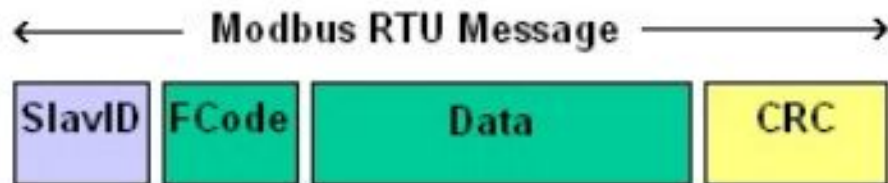
Fundamentação Teórica

Protocolo Modbus TCP

- **O protocolo é baseado em Ethernet que permite a troca de dados entre dispositivos via TCP/IP. Ele opera no modelo cliente-servidor, e encapsula os dados do Modbus tradicional (RTU) dentro de pacotes TCP/IP, preservando sua estrutura lógica de mensagens.**

Fundamentação Teórica

Quadros da mensagem



Fundamentação Teórica

Campo	Seção	Descrição	Tamanho (bytes)	Exemplo (Hex)
Transaction ID	MBAP Header	ID da transação definido pelo cliente	2	00 01
Protocol ID	MBAP Header	Sempre 0x0000 para Modbus	2	00 00
Length	MBAP Header	Nº de bytes restantes (Unit ID + PDU = 1 + 7)	2	00 08
Unit ID	MBAP Header	Identificação do dispositivo alvo	1	01
Function Code	PDU	Código da função (0x0F = Write Multiple Coils)	1	0F
Starting Address	PDU	Endereço inicial dos coils	2	00 13
Quantity of Coils	PDU	Quantidade de coils a escrever (10)	2	00 0A
Byte Count	PDU	Nº de bytes nos valores dos coils	1	02
Output Values	PDU	Bits para os coils (ex: 1,0,1,0,1,0,1,0,1,0)	2 ou +	55 01

Proposta

Componentes do Sistema

- **O sistema é composto por um dispositivo ESP32-WROOM-32, configurado como servidor Modbus TCP. A comunicação entre cliente e servidor é estabelecida via Wi-Fi, utilizando a porta 502 e o endereço IP dinâmico atribuído ao servidor pela rede. A estrutura das mensagens segue o padrão definido pelo protocolo Modbus TCP.**

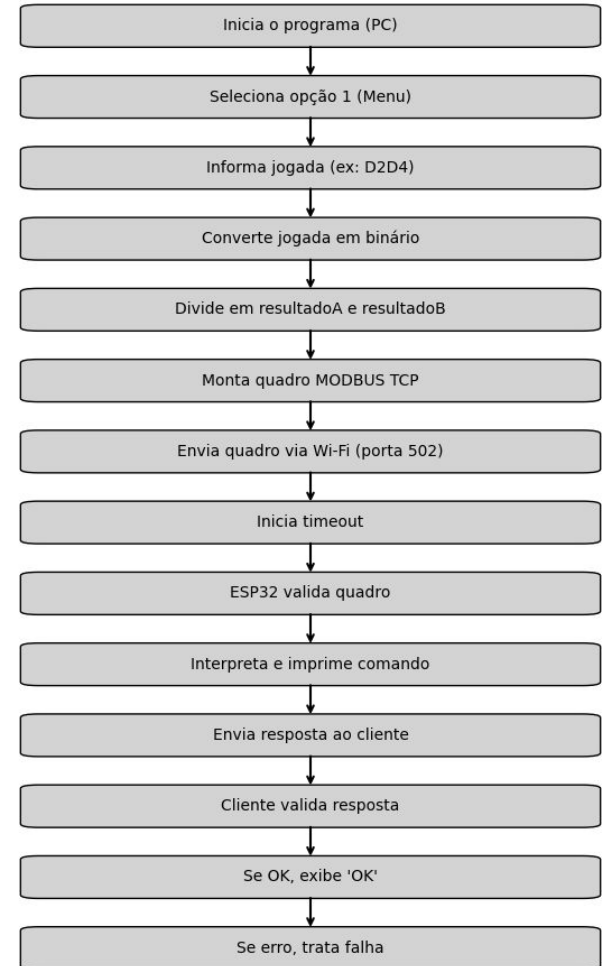
Proposta

Matriz de LEDs

- **A matriz de LEDs 8x8 possui 64 LEDs dispostos em linhas e colunas. É controlada por meio de registros que determinam quais LEDs devem ser acesos, permitindo a criação de diversos padrões visuais.**

Fluxograma explicado

1. Usuário executa o programa em C no terminal Linux.
2. Usuário seleciona a opção 1 no menu (Write Multiple Coils).
3. Usuário informa a jogada de xadrez (ex: D2D4).
4. Programa converte a jogada para um padrão binário.
5. Programa divide os 16 bits em dois bytes: `resultadoA` e `resultadoB`.
6. Quadro MODBUS TCP é montado com função 0x0F.



Fluxograma explicado

7. Quadro é enviado via Wi-Fi pela porta 502.

8. Contagem de timeout é iniciada.

9. ESP32 recebe e valida o quadro.

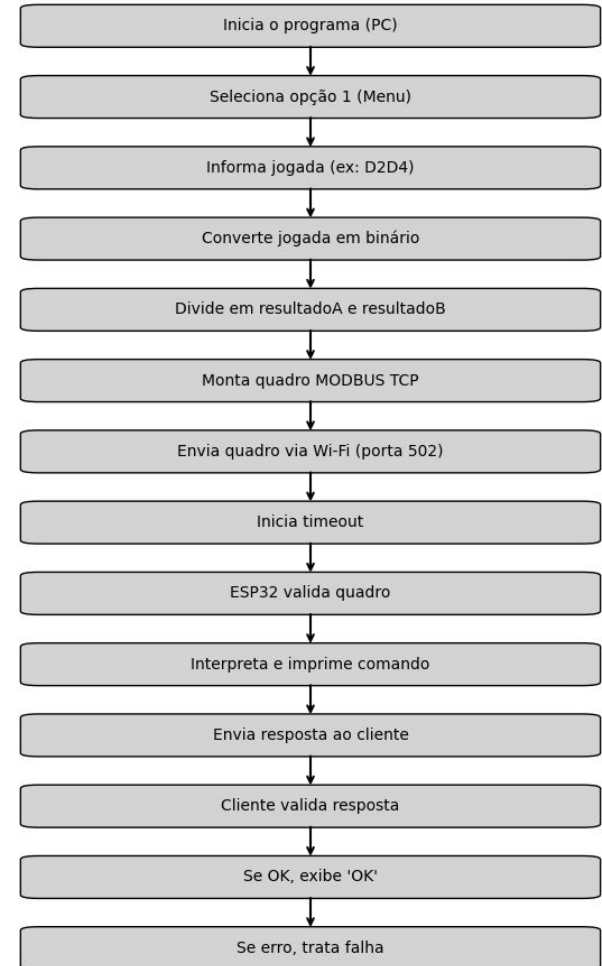
10. ESP32 interpreta os dados e imprime no Serial Monitor.

11. ESP32 envia quadro de resposta ao cliente.

12. Cliente valida a resposta recebida.

13. Se válida, exibe mensagem "OK".

14. Se inválida ou sem resposta, trata o erro.



Testes

Procedimentos de Teste

- **Foram realizados testes para verificar a correta comunicação entre o cliente e o servidor. Os testes confirmaram a eficácia da implementação através da impressão em ambos do mesmo binário.**

Testes

Rotinas de Falhas

- **O menu do Cliente possui rotinas que tratam erros como endereço inválido e ausência de resposta do escravo. Se ocorrer falha, o sistema exibe mensagens de erro e permite nova tentativa, sem travar a execução.**

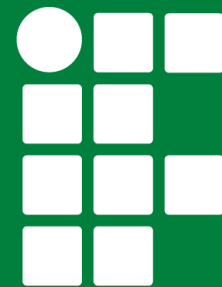
Desafios

Melhorias de Configuração

Identificou-se que pode ser aprimorado com a parametrização de variáveis atualmente estáticas, como o endereço de rede, senha, e porta serial. Torná-las configuráveis aumentaria a flexibilidade do sistema e permitiria múltiplas execuções com diferentes dispositivos sem alteração no código-fonte.

Referências

- **MODBUS ORGANIZATION. MODBUS Messaging on TCP/IP Implementation Guide V1.0b. [S.l.], 2006. Disponível em: https://modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf. Acesso em: 22 jun. 2025.**
- **EMODBUS. ModbusServer TCP – eModbus documentation. Disponível em: <https://emodbus.github.io/modbusserver-tcp>. Acesso em: 22 jun. 2025.**
- **PEDRO HENRIQUE DE ASSUMPÇÃO. ModbusTCP. GitHub, 2025. Disponível em: <https://github.com/pedrohdea/ModbusTCP>. Acesso em: 22 jun. 2025.**
- **ESPRESSIF SYSTEMS. ESP32 – TCP/IP Server Example. Documentation. [S.l.], [s.d.]. Disponível em: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/lwip.html#tcp-server>. Acesso em: 22 jun. 2025.**



**INSTITUTO
FEDERAL**

Rio Grande
do Sul

Campus
Farroupilha