MODBUS RTU

ÍNDICE

INTRODUÇÃO

O objetivo deste artigo é introduzir conceitos básicos do Protocolo de Comunicação de dados Modus RTU afim de instruir o usuário com os principais aspectos de sua aplicação. Hoje em dia, Modbus é um Protocolo aberto usado em muitos produtos de automação. Modbus pode ser usado em comunicações Seriais e Ethernet. Neste artigo, o alvo é o Protocolo Modbus RTU em redes seriais.

MODBUS RTU

O Protocolo de Comunicação MODBUS RTU *(Remote Terminal Unit – Unidade Terminal Remota)* é projetado para operar redes seriais baseado na configuração entre Mestre-Escravo. Desenvolvido pela Modcon em 1979, suas aplicações consistiam em sistemas BSM *(Building Management System – Sistema de gerenciamento predial)* e IAS *(Industrial Automation Systems – Sistema de Automação Industrial)* e também em sistemas SCADA *(Supervisory Control and Data Acquisition – Sistemas de Supervisão e Aquisição de Dados)*. Detalhes sobre a aplicação de cabeamento RS-485 não serão detalhados neste artigo. Mais informações sobre a rede física RS-485 poderão ser consultados no artigo Rede física RS-485. O Modbus é um protocolo de mensagens de comandos e respostas na posição de camada física do nível 7 no modelo OSI. O Modbus RTU em específico monta um pacote de dados utilizando códigos binários e inclui CRC *(Cyclic Redundancy Check – Verificação de Redundância Cíclica)* de 16 bits.

APLICAÇÕES DO PROTOCOLO

Na figura 1, temos um esquema completo das aplicações de comunicação Modbus. neste artigo, detalharemos os conceitos básicos do Protocolo Modbus RTU na rede serial RS-485.

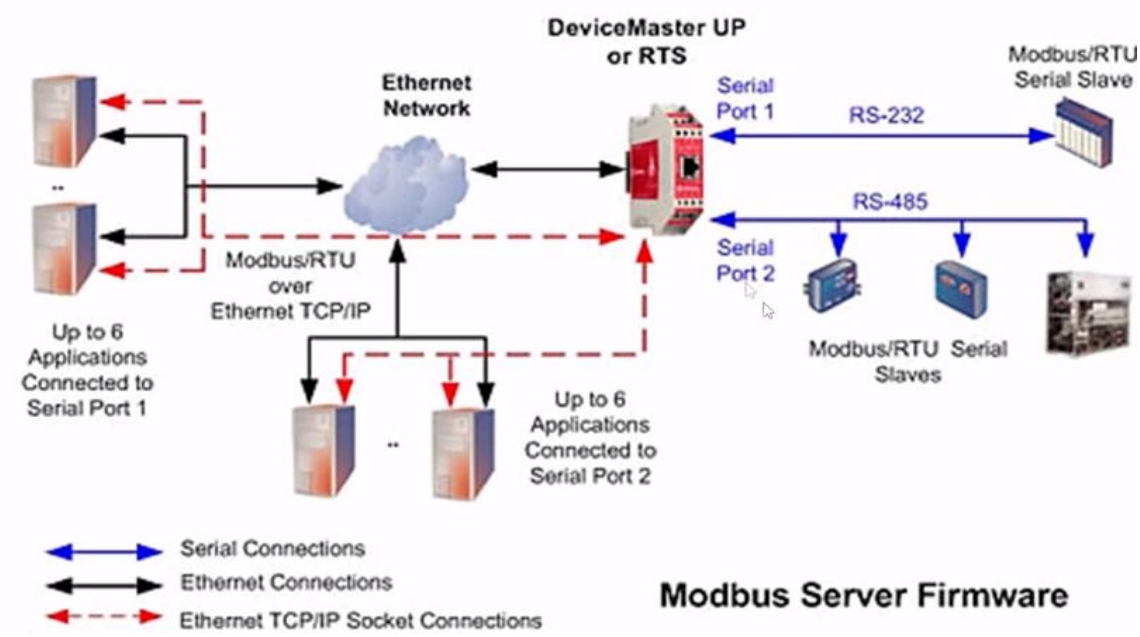

Figura 1 – Protocolo de Comunicação Modbus

MAPAS DE ENDEREÇAMENTO DE PROTOCOLO

Existem 4 tabelas de endereços onde as informações são armazenadas. Dois mapas armazenam valores discretos simples, chamados "Bobinas" *(Coils),* e dois mapas onde são armazenados valores numéricos de 16 bits, chamadas "Registradores" *(Registers)*. Para cada tipo de dado, existe um mapa apenas para leitura *(Read Only)* e um mapa para Leitura e Escrita *(Read/Write)*. Cada mapa é capaz de armazenar até 9.999 endereços.

NOME DO MAPA	FUNÇÕES	ENDEREÇO	TAMANHO DE DADOS
COILS	LEITURA E ESCRITA	00001 - 09999	1 BIT
DISCRETE INPUTS	SOMENTE LEITURA	10001 - 19999	1 BIT
INPUT REGISTER	SOMENTE LEITURA	30001 - 39999	16 BITS
HOLDING REGISTERS	LEITURA E ESCRITA	40001 - 49999	16 BITS

Tabela 1 – Mapas do Protocolo Modbus

O Primeiro mapa, Coils – Read/Write, endereça dados entre 00001 a 9.999;

O segundo mapa, Discrete Inputs Read only endereça dados entre 10001 a 19999;

O terceiro mapa, input Register – read only, endereça dados entre 30001 a 39999.;

O Quarto mapa, Holding Registers – Read/Write, endereça dados entre 40001 a 49999.;

As tabelas Bobinas e Entradas Discretas contem 1 bit de dados. Basicamente, refere-se a ligado (ON) ou desligado (OFF). O Registradores é o termo para uma word *(palavra)* ou 16 bits ou 2 bytes de dados. Neste artigo, usaremos o termo 16 bits.

CÓDIGOS DE FUNÇÃO MODBUS

As funções Modbus estão divididas entre Funções de Leitura e Funções de Escrita. As mais comuns são:

FUNÇÃO LEITURA:

- 01: Read Coil Status – 0x (Leitura de Bit);
- 02: Read Input Status – 1x (Leitura de Bit);
- 03: Read Holding Register – 4x (Leitura de 16 bits);
- 04: Read Input Register – 3x (Leitura de 16 bits);
- 07: Read Exception Status – 6x (Leitura de Status);
- 20: Read File Register – 6x (Leitura de Registro de Arquivo);

FUNÇÃO DE ESCRITA:

- **05:** Force Single Coil – 0x (Escrita de Bit);
- **06:** Preset Single Register – 4x (Escrita de 16 bits Simples);
- **15:** Force Multiple Coils – 0x (Escrita de Bits);
- **16:** Preset Multiple Registers – 4x (Escrita de 16 bits);
- **21:** Write File Register – 6x (Escrita de Registro de Arquivo);

FRAMES DO MODBUS RTU

Na rede Modbus RTU, cada Escravo tem um ID, que identifica o número do Escravo na rede e pode conter de 1 até 247 endereços ID. Sabendo que o Modbus RTU opera em Half Duplex, apenas uma operação por vez é permitida e é o Mestre quem direciona a troca de mensagens. Analisaremos as duas operações:

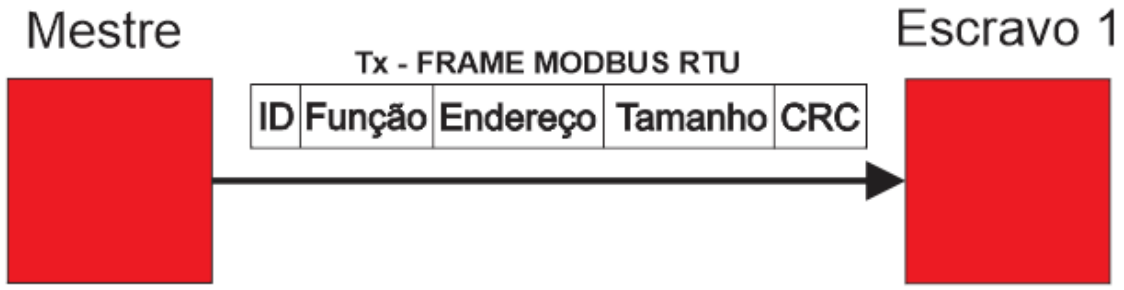

Figura 2 – Frame de envio de mensagem

Onde:

- **ID:** É o Byte do endereço de rede do Escravo que o Mestre está direcionando a mensagem;
- **FUNÇÃO:** Indica qual função Modbus está sendo utilizada;
- **ENDEREÇO:** Endereço inicial da memória formada por um conjunto de 2 Bytes;
- **TAMANHO:** Quantos dados estão sendo requisitados, ou seja, a quantidade de registros a serem lidos a partir do endereço inicial da memória. Lembrando que a quantidade de dados de registro são de 0 a 252 bytes;
- **CRC:** Verificação de erros na transmissão de dados;

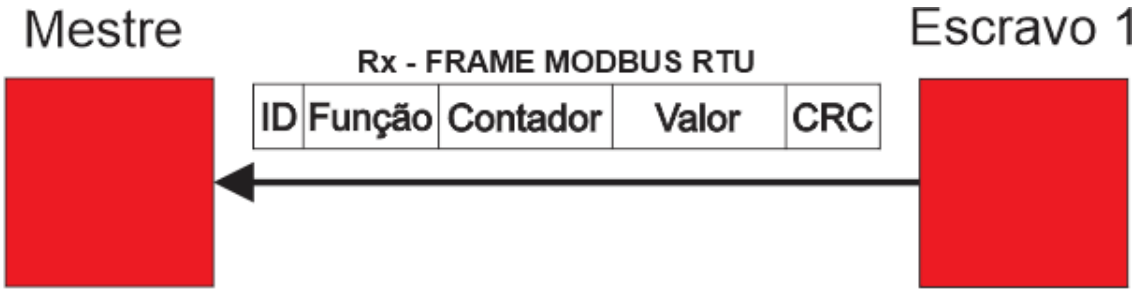

Figura 3 – Frame de recebimento de mensagem

Onde:

- **ID:** É o Byte do endereço de rede do Escravo, no frame Rx, o Escravo responde para o Mestre que ele está respondendo a mensagem que foi endereçada com o ID dele;
- **FUNÇÃO:** Indica qual função Modbus está sendo utilizada;
- **CONTADOR:** Indica a quantidade de Bytes de todos os valores recebidos;
- **VALOR:** Valor de 2 Bytes que o equipamento (Escravo) está retornando à requisição do Mestre;
- **CRC:** Verificação de erros na transmissão de dados;

ESTRUTURA DE FUNCIONAMENTO DE FRAMES DO MODBUS RTU

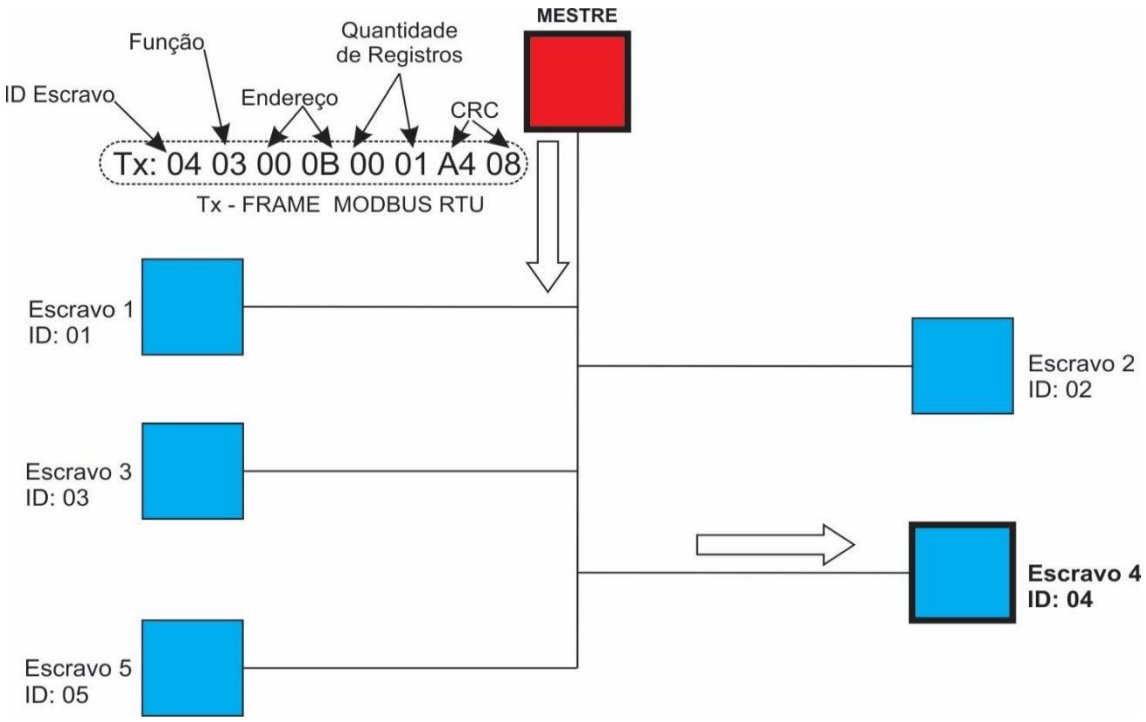

Figura 4 – Estrutura de funcionamento do frame Tx

Na figura acima, temos um exemplo de comunicação em uma configuração barramento onde o Mestre está transmitindo um pacote de dados na rede, onde:

- **ID Escravo:** No início da mensagem do exemplo, o mestre está direcionando a mensagem para o Escravo 4 (ID: 04);
- **Função:** A função que o Mestre está especificando é a 03 Read Holding Register - 4x, Leitura de word, a partir do endereço 400001;
- **Endereço:** Qual endereço no Mapa Read Holding Register que o Mestre quer acessar, no caso do exemplo, o endereço é 00 0B = 11 (O endereço do pedido está em Hexadecimal e a conversão em Decimal), ou então, simplesmente, endereço 40011;
- **Quantidade de Registros**: Neste campo podemos ler até duas variáveis, no caso do exemplo, o pedido é de apenas 1 variável;
- **CRC**: Checagem de erros;

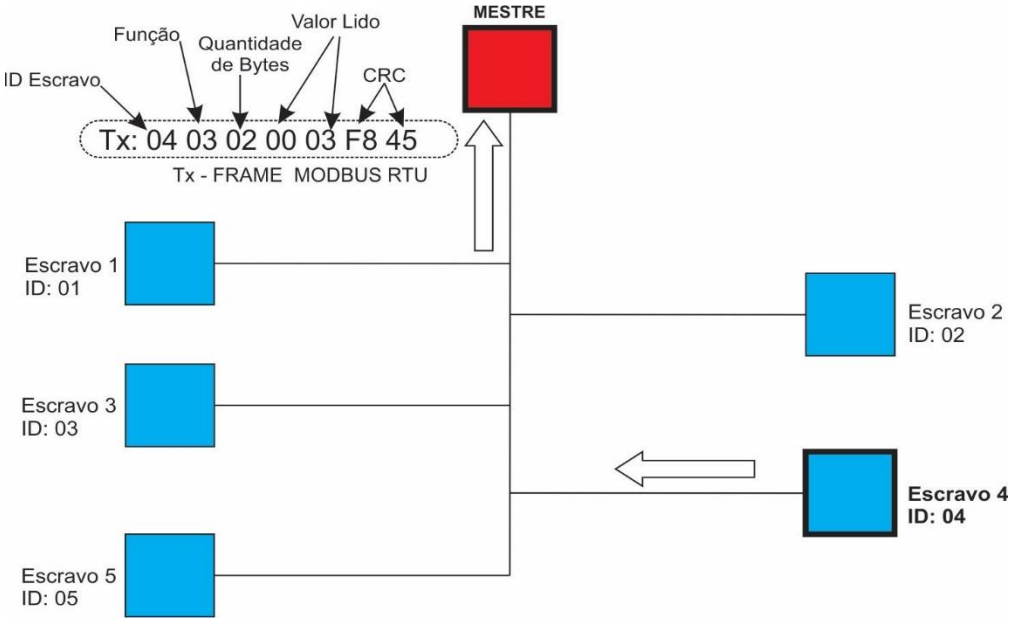

Figura 5 – Estrutura de funcionamento do frame Rx

Na figura acima, temos um exemplo de comunicação em uma configuração barramento onde o Mestre está recebendo uma resposta do Escravo do qual solicitou um comando, onde:

- ID Escravo: O Escravo se identifica para o Mestre. No caso do exemplo, o Escravo 4 (ID: 04);
- Função: O Escravo indica que executou a função requisitada, no caso do exemplo, a função executada é a 03 read Holding register – 4x (Leitura de 16 bits);
- Quantidade de Bytes: Especifica a quantidade de Bytes que o escravo está lendo, no caso do exemplo, são 2 bytes;
- Valor Lido: Caracteres da Word (16bits) lida;
- CRC: Checagem de erros;

FUNCIONAMENTO

O Protocolo Modbus tem uma unidade de dados simples, chamada PDU *(Protocol data Unit)*, que se mantém intacta nas múltiplos modos de protocolo e em diversas camadas de aplicação. O FRAME de comunicação na integra, que inclui a PDU e os campos adicionais, denominados ADU *(Application Data Unit)* que são o Cabeçalho do pacote. Somando ADU e PDU o limite máximo de dados para mensagem, no Padrão Modbus, é de 253 bytes.

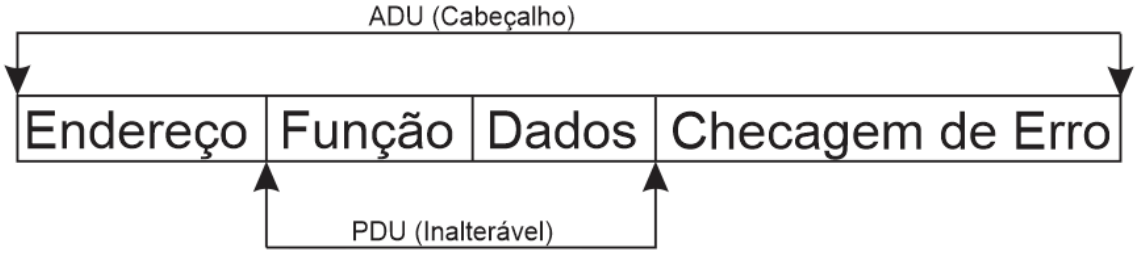

Figura 6 – Identificação de PDU e ADU

Portanto, dependendo do tipo de função e/ou de dado que será utilizado na comunicação, a tabela 2 do padrão Modbus os seguintes limites de dados de bloco em cada comunicação

Funções Modbus	Descrição de função	Limite de dados
03.04	Leitura de múltiplos registros de 16 bits	250 Bytes (125 Registros)
16	Escrita de múltiplos registros de 16 bits	247 Bytes (123 Registros)
01.02	Leitura de múltiplos bits	250 Bytes (2000 bits)
15	Escrita de Múltiplos bits	247 Bytes (1968 bits)
20	Leitura de Registros de arquivos	248 Bytes (124 Registros)
21	Escrita de Registros de arquivo	244 Bytes (122 Registros)

Tabela 2 – Limite de dados de função

Outra observação importante do mensagem completa, é que o Mestre envia uma Condição de Início *(Start Condition, ou simplesmente, Start)* para todos os escravos da rede. assim como a Condição de Início, o endereço do Escravo (ID) também é enviado a todos os equipamentos da rede. Quando o Escravo requisitado recebe a mensagem contendo seu endereço, ele é ativado. O tamanho da Condição de Início consiste em 28 bits ou 3,5 caracteres. (Cada Caractere contém 8 bit ou 1 byte). E após o envio do pacote de mensagem, o Mestre envia ao Escravo endereçado na mensagem (ID) a Condição de Término *(Stop Condition, ou simplesmente, Stop)* que também contém 16 bits (ou 3,5 caracteres).

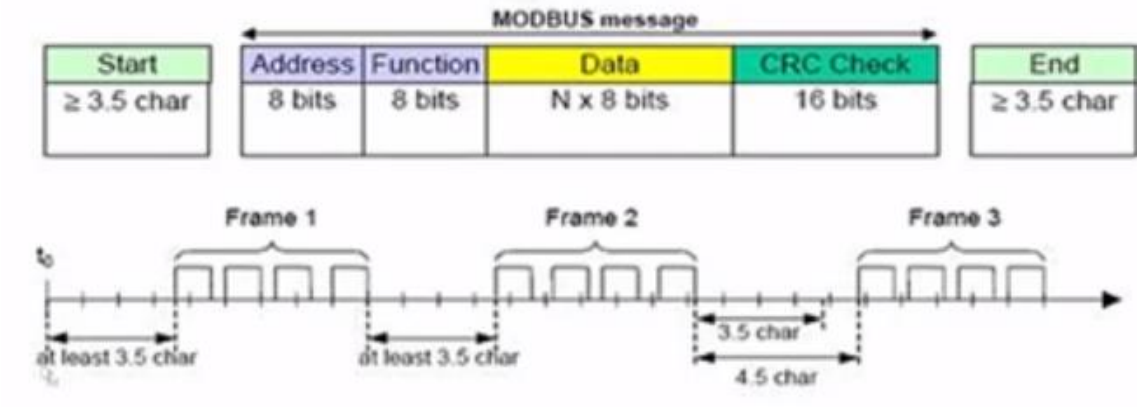

Figura 7 – Protocolo Completo

REFERÊNCIA

MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b.pdf

ELIPSE TUTORIAL DRIVER MODBUS