**Vitor Amadeu Souza**

Desenvolvendo um medidor

# Decibelímetro MODBUS

RS232 no STM32F103 programado no Arduino

**© 2020 by Cerne Tecnologia e Treinamento Ltda.**

**© 2020 by Vitor Amadeu Souza**

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.** Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

**Outubro de 2020**

Direitos reservados por:
Cerne Tecnologia e Treinamento Ltda
*Produção: Cerne Tecnologia e Treinamento*
*E-mail da Empresa: cerne@cerne-tec.com.br*
*Home Page: www.cerne-tec.com.br.com.br*
*Atendimento ao Consumidor: sac@cerne-tec.com.br*
*Contato com o Autor: vitor@cerne-tec.com.br*



**FEITO NO BRASIL**

**Dedicatória**

Como nos meus outros livros, dedico este livro a minha querida esposa Renata Leal.

***"O tempo cura as dores e as querelas porque mudamos e não somos mais a mesma pessoa."***

**Blaise Pascal**

**Kits Didáticos e Gravadores da Cerne Tecnologia**

A Cerne tecnologia têm uma linha completa de aprendizado para os microcontroladores da família PIC, 8051, Holtek, dsPIC, ARM e etc. Veja os detalhes de cada um nas figuras abaixo:




**Kit Cerne MODBUS**

- Microcontrolador PIC16F628A
- Botões
- Leds
- Gravação ICSP
- Display LCD
- Comunicação RS485 MODBUS
- E muito mais!

Uma linha completa de componentes para o desenvolvimento de seus projetos eletrônicos como displays, PICs, botões, leds, cristais, etc. Visite a nossa página na Internet, no endereço **www.cerne-tec.com.br** e conheça melhor nossos serviços e produtos.

Cerne
Conhecimento para o Desenvolvimento

**www.cerne-tec.com.br**

# Sumário

**Capítulo I – Metodologia de desenvolvimento ..... 6**

1. Introdução ..... 6

**Capítulo II – MODBUS SLAVE ..... 7**

1. Introdução ..... 7
2. Tipos de quadros ..... 8
3. Modos de transmissão ..... 10
4. Comunicação RS232 ..... 12
5. Software de Comunicação MODBUS ..... 18
6. Protocolo de Comunicação ..... 20
7. Programa para o Microcontrolador ..... 25
8. Calculando o CRC ..... 30
9. Ajustando o MODBUS Test Pro ..... 31
10. Programa para o Arduino ..... 35

**Capítulo III – Decibelímetro ..... 41**

**Capítulo IV – Sensor MODBUS ..... 57**

# Capítulo I

# Metodologia de desenvolvimento

## 1. Introdução

Esta literatura é uma continuação da obra *ARM no Arduino – Com base no modelo STM32F103C8 (2019)* do mesmo autor e editora, onde a proposta é desenvolver um sensor medidor seguindo o título da obra que forneça o resultado de tal medição através da rede MODBUS RS232 programado no Arduino.

A placa didática utilizada foi a *Blue Pill* STM32F103, onde os recursos para testar os circuitos propostos são conectados através das conexões disponíveis na lateral do kit. Tal kit está à venda no site www.cerne-tec.com.br.

# Capítulo II

# MODBUS SLAVE

## 1. Introdução

O protocolo MODBUS é um dos mais usados a nível industrial atualmente. O mesmo é usado por CLPs e Computadores para monitorarem, por exemplo, uma rede de periféricos como sensores, válvulas, entradas analógicas e saídas analógicas através desta rede. Veja abaixo algumas características deste protocolo:

- Utilizado para comunicação entre CLPs;
- Define a estrutura de quadros, não de meio físico;
- Usualmente utilizado sobre RS232, RS485 ou Ethernet;
- Arquitetura do tipo Mestre/Escravo;
- Permite Broadcast;
- Apenas dois tipos de quadros.

O protocolo MODBUS foi desenvolvido pela Modicon Industrial Automation Systems, hoje Schneider, para comunicar um dispositivo mestre com outros dispositivos escravos. Embora seja utilizado normalmente sobre conexões seriais padrão RS-232, ele também pode ser usado como um protocolo da camada de aplicação de redes industriais tais como TCP/IP sobre Ethernet e MAP. Este é talvez o protocolo de mais larga utilização em automação industrial, pela sua simplicidade e facilidade de implementação.

## 2. Tipos de quadros

Os tipos de quadros ou frames disponíveis para o MODBUS são os seguintes:

- Quadro de consulta;
- Endereço;
- Código da função;
- Dados;
- Verificação de erro.

Ou seja, o quadro que pode fazer a consulta, por exemplo, em um sensor ou CLP contém nele um campo que identifica o endereço no qual o mestre deseja se comunicar, neste caso com o escravo podendo ser do endereço de 1 a 247. Caso o endereço seja 0, é identificado como um tipo de mensagem broadcast, onde todos os elementos da rede podem receber tal dado. Em seguida, é informado o código da função que identifica o que o frame em si faz. Seguido do identificador da função vem a parte de dados, onde neste há o pacote de dados referente as informações que serão enviadas. Finalmente há a parte de verificação de erros, onde nesta parte há checagem de dados enviados, a fim de verificar se foi enviado corretamente. O próximo frame é o de resposta, composto dos seguintes itens:

- Quadro de resposta;
- Endereço;
- Confirmação (Eco do código de função enviado);
- Dados;
- Verificação de erro.

## 3. Modos de Transmissão

Existem dois tipos de modos de transmissão usados no MODBUS. O primeiro deles chama-se modo do tipo ASCII e outro do tipo RTU. No mercado industrial e em geral, o modo do tipo RTU é o mais usado, sendo menos o do tipo ASCII. No modo de transmissão ASCII há o desperdício de dados do canal, pois cada caracter enviado pela rede é enviado como um caracter ASCII enquanto que no modo RTU, o dado é transmitido em forma de bits. Observe abaixo como é formado o frame do tipo RTU:

| Start | Endereço | Função | Dados | CRC | END |
|---|---|---|---|---|---|
| Silêncio 3..5 chars | ← 8 bits→ | ← 8 bits→ | ←N x 8 bits→ | ←16 bits→ | Silêncio 3..5 chars |

Observe primeiramente que temos um período de silêncio, onde sempre que as linhas de comunicação ficarem sem tráfego de dados pelo tempo de até 3,5 caracteres, o escravo pode entender que a rede entrou em estado idle, ou seja, de repouso. Como a comunicação utilizada terá uma taxa de comunicação de 9600 bps, pode observar que o tempo de transmissão de 1 bit será de :

$$1/9600 = 104\mu s$$