SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL
FACULDADE DE TECNOLOGIA SENAI CIMATEC
CURSO SUPERIOR DE TECNOLOGIA EM MECATRÔNICA INDUSTRIAL

SISTEMA DE AVALIAÇÃO E DIAGNÓSTICO DE FALHAS EM REDES DE COMUNICAÇÃO MODBUS/RTU EM CHÃO DE FÁBRICA

Salvador
2008

GENIVALDO CORDEIRO BARROS JR

SISTEMA DE AVALIAÇÃO E DIAGNÓSTICO DE FALHAS EM REDES DE COMUNICAÇÃO MODBUS/RTU EM CHÃO DE FÁBRICA

Trabalho de conclusão de curso apresentado à Faculdade de Tecnologia SENAI Cimatec como requisito final para obtenção do título de Tecnólogo em Mecatrônica Industrial.
Orientador: Msc. Cleber Vinícius Ribeiro de Almeida

Salvador
2008

Ficha catalográfica elaborada pela Biblioteca da Faculdade de Tecnologia

SENAI Cimatec

Barros Junior, Genivaldo Cordeiro.

Sistema de avaliação e diagnóstico de falhas em redes de comunicação modbus/rtu em chão de fábrica / Genivaldo Cordeiro Barros Junior. – Salvador, 2008.

43f.

1. Diagnóstico de falhas – Redes de comunicação. I. título

CDD 629.8

RESUMO

O desenvolvimento da eletrônica e da informática no século XX proporcionou o avanço e utilização das redes de comunicação de dados para a realização de troca de informações entre computadores. A indústria se aproveitou deste novo conceito e incorporou-o em equipamentos com o objetivo de trocar informações do processo, com algumas vantagens em relação ao método tradicional analógico, para economia de cabeamento por exemplo visando redução de custo. Em 1979 a Modicon padronizou um protocolo de comunicação aberto chamado Modbus, que inicialmente foi projetado para a comunicação serial, mas que ao passar dos anos este protocolo passou a ser amplamente utilizado em sistemas de automação industrial, e até hoje ainda o são. Apesar da grande participação desta tecnologia no mercado industrial, o domínio da mesma, principalmente, pelos profissionais de manutenção, ainda é bastante restrito. Com isso o objetivo deste trabalho é propor um sistema que possibilite a avaliação e diagnostico de falhas na comunicação entre equipamentos industriais com padrão de meio físico RS-232/485 e protocolo Modbus/RTU.

Palavras-Chaves: Rede, Meio Físico, Protocolo, Comunicação, Modbus, Chão de fábrica e CLP.

LISTA DE ACRÔNIMOS

ABNT	Associação Brasileira de Normas Técnicas
ADU	*Application Data Unit*
ASCII	*American Standard Code for Information Interchange*
CI	Circuito Integrado
CP	Controlador Programável
CLP	Controlador Lógico Programável
CMOS	*Complementary metal-oxide-semiconductor*
CRC	*Cyclic Redundancy Check*
DCE	*Data Circuit-terminating Equipment*
DTE	*Data Terminal Equipment*
EIA	*Electronic Industry Association*
GPIB	*General Purpose Interface Bus*
IHM	Interface Homem-Máquina
IEEE	*Institute of Electrical and Electronics Engineers*
IP	*Internet Protocol*
ISO	*International Standards Organization*
LAN	*Local Area Network*
LHN	*Long Haul Network*
LRC	*Longitudinal Redundancy Check*
MODEM	Modulador Demodulador
OSI	*Open Systems Interconnection*
PC	*Personal Computer*
PDU	*Protocol Data Unit*
RS	*Recommended Standard*
RTU	*Remote Terminal Unit*
WAN	*Wide Area Network*
TCP	*Transmission Control Protocol*
TIA	*Telecommunications Industry Association*
TTL	*Transistor-Transistor Logic*

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1. Introdução

Atualmente, as redes de comunicação no ambiente industrial são largamente utilizadas. Existem diversos protocolos e meios físicos que viabilizam a comunicação de dados entre equipamentos industriais. Por serem susceptíveis a erros e a defeitos que na maioria das vezes estão relacionados ao ambiente agressivo em que estão submetidos, as redes de comunicação necessitam de diagnósticos constantes.

Em função disso, observa-se que há a necessidade de avaliação e diagnóstico do funcionamento da comunicação de equipamentos industriais que utilizam as redes de comunicação. Apesar dos trabalhos técnicos e científicos encontrados, não foi identificado um método de diagnóstico de defeitos e de simulação que contemple a configuração proposta.

Assim, o objetivo deste trabalho é gerar um aplicativo para Computador Pessoal (Personal Computer - PC), que permita a realização de diagnóstico de defeitos de comunicação em equipamentos industriais. As principais contribuições deste Trabalho de Conclusão de Curso (TCC) é a proposição de um sistema que possibilite a redução do tempo de máquina parada em plantas industriais e consequentemente a diminuição dos desperdícios de insumos e matéria-prima e custo de manutenção em aplicações industriais que utilizam redes de comunicação Modbus/RTU (*Remote Terminal Unit*).

O desenvolvimento e a validação do aplicativo foram realizados nos laboratórios de microeletrônica e sistemas digitais da Faculdade de Tecnologia SENAI-CIMATEC.

1.1. Definição do Problema

Por estarem as redes de comunicação de dados Modbus/RTU inseridas em um ambiente industrial, onde as condições são bastante agressivas (calor excessivo, intempéries, vibração, poeira, umidade, interferências eletromagnéticas etc.), é comum à ocorrência de diversos defeitos. Estes são decorrentes de falhas no equipamento mestre ou escravo, de problemas com o meio físico da comunicação, ou até mesmo de deficiências na transmissão ou recepção das

informações. Por exemplo, um CLP (Controlador Lógico Programável) que foi configurado como o mestre da comunicação, obtendo dados de um transmissor de vazão, configurado como escravo, caso a comunicação entre eles for interrompida (Figura 1), apesar dos mesmos estarem ligados e funcionando, não é evidente onde o problema se encontra. A fonte do defeito poderá ser o mestre, o escravo ou o meio físico. Portanto o objetivo do **Sistema de Avaliação e Diagnóstico de Falhas em Redes de Comunicação Modbus/RTU em Chão de Fábrica** desenvolvido neste trabalho é auxiliar o técnico a descobrir qual a causa das falhas e assim poder eliminá-la.

Figura 1. Falha de Comunicação

1.2. Justificativa

Atualmente no ramo da automação industrial, o protocolo de comunicação Modbus/RTU é amplamente utilizado. Porém, um número restrito de profissionais detém o domínio desta tecnologia. Além disso, há também a carência de dispositivos (*hardware e software*) específicos para a simulação, testes e diagnósticos de equipamentos com tal especificação. Com isso, a avaliação (fazer uma verificação superficial para se certificar como está a situação) e o diagnóstico (determinar a causa precisa do problema ou defeito) de problemas é bastante lento e perde-se produtividade por parada da planta por um tempo relativamente alto. Apesar de se encontrar no mercado alguns *softwares* que se propõem a realizar as tarefas de diagnóstico e simulação, muitas vezes eles possuem custo elevado e/ou apresentam determinadas limitações de funcionamento. Por isso, este projeto propõe o desenvolvimento de um *software* para ambiente computacional que contemple as principais funções do protocolo Modbus/RTU.

Isto proporcionará a avaliação e o diagnóstico de uma rede de comunicação deste tipo em um chão de fábrica. Buscou-se uma solução simples e de baixo custo, por meio da utilização de um *hardware* já existente no mercado e sendo desenvolvido apenas o *software*.

1.3. Objetivos

Este trabalho teve como objetivo principal propor um sistema capaz de auxiliar na avaliação e diagnóstico de falhas de comunicação entre equipamentos industriais que se comuniquem através de redes baseadas no padrão de meio físico EIA/TIA-232/485 e protocolo Modbus/RTU.

1.3.1. Objetivos Específicos

Desenvolver um *software* aplicativo de Interface Homem-Máquina (IHM), para avaliação e o diagnóstico de defeitos em equipamentos industriais. Dentro deste contexto, são especificados os equipamentos constituintes do sistema, como *Notebook*, conversor de meio físico (EIA-232 para EIA-485), conversor USB para Serial e cabos de comunicação, dentre outros.

1.4. Metodologia

O trabalho foi dividido em etapas, as quais serão realizadas individualmente e seguindo a ordem e prazos definidos no cronograma de atividades. Este cronograma será constantemente atualizado, pois é através do mesmo que será feito o acompanhamento do avanço do projeto ao decorrer do tempo. Abaixo são listadas as etapas do projeto.

- Cronograma de Atividades;
- Definição de escopo;
- Pesquisa e Levantamento Bibliográfico sobre o Modbus/RTU e padrão de meio físico EIA-485;
- Estudo Teórico do Protocolo Modbus/RTU;

- Projeto da solução;
- Modelagem do *Software*;
- Programação do *Software*;
- Testes de comunicação com equipamentos mestre e escravos;
- Validação através da realização de testes rotineiros;
- Redação do TCC;
- Defesa do TCC;

2. Redes de Comunicação de Dados

Os modernos sistemas de processamento são constituídos de um grande número de computadores autônomos, porém interconectados por redes de comunicação de dados. Estes sistemas são chamados de redes de computadores (TANENBAUM,2003).

O aprimoramento da tecnologia empregada nos computadores e nos sistemas de comunicação de dados influenciou profundamente o modo como são organizados os sistemas computacionais e, portanto, os sistemas de automação. O conceito antigo de centro de computação tornou-se obsoleto e um novo modelo tomou seu lugar.

O termo rede de computador significa um conjunto de computadores autônomos interligados para trocar informação. A ligação não precisa ser necessariamente um par de fios de cobre, mas podem ser usadas fibras ópticas, microondas e satélites de comunicação. Uma rede com muitos computadores localizados no mesmo prédio é chamada de **Rede de Área Local** (***Local Area Network* – LAN**), em contraste com a chamada **Rede de Área Distante (*Wide Area Network* – WAN**), também chamada de **Rede de Longo Alcance** (***Long Haul Network* – LHN)** (TANENBAUM,2003).

Uma rede é uma configuração de dispositivos de processamento de dados e programa ligados para trocar informação. Ela é formada por um grupo de

nós e *links* que os interligam. Quando se impõe que os computadores sejam autônomos, se excluem os sistemas em que há uma relação clara de mestre/escravo. Se um computador pode controlar outro, este ultimo não é autônomo.

2.1. Modelo OSI

Com o advento das redes de comunicação de dados, tornou-se ideal a padronização dos equipamentos e de seus programas tanto para fabricantes de equipamentos de automação e controle quanto para os usuários. Em 1983, a Organização Internacional para Padronização (ISO - *International Standards Organization*), propôs um Modelo de Referência para Interconexão Aberta para uso universal chamado OSI (*Open Systems Interconnection Reference Model*), para ser aplicado na fabricação de equipamentos digitais (TANENBAUM,2003). Os conceitos básicos do modelo OSI são as camadas, as entidades e os protocolos:

- Camadas (*Layers*): São níveis de hierarquia, dispostos de forma que cada um presta serviço para um nível mais alto, agregando valor ou função aos serviços dos níveis mais baixos. Uma estação de trabalho pode participar, física e logicamente, de uma ou mais camadas;
- Entidades: São dispositivos de *hardware* ou de *software* que cooperam para produzir serviços em uma camada;
- Protocolos: São conjuntos de regras que regulam a comunicação entre as entidades de uma determinada camada.

O modelo de referência OSI da ISO possui sete camadas com as seguintes funções (Figura 2):

- Camada 7 – Aplicação (*Application*): Fornece recursos e os administra para realizar a transferência de dados da aplicação;
- Camada 6 – Apresentação (*Presentation*): Realiza as transformações e representações da informação;

- Camada 5 – Sessão (*Session*): Proporciona a manutenção da associação entre entidades da aplicação e controle dos diálogos;
- Camada 4 – Transporte (*Transport*): Controla o fluxo de dados e o tratamento de erros entre estações;
- Camada 3 – Gerência de Rede (*Network*): Realiza o encaminhamento (*routing*) de dados, chaveamento e outros serviços internos à rede;
- Camada 2 – Transmissão de Dados (*Data Transmission*): Realiza o controle de fluxos e de erros nos enlaces de dados simples e acesso ao meio de comunicação;
- Camada 1 – Rede Física (*Physical Network*): Proporciona a transferência de bits de dados e de sinalização.

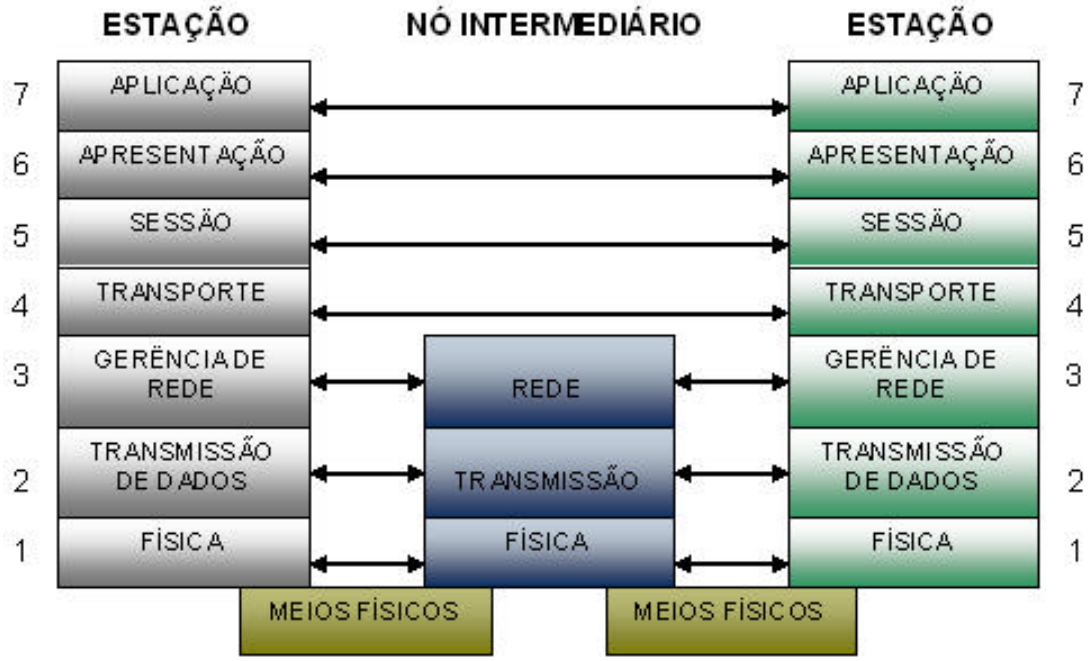

Figura 2. Modelo OSI

No nível físico, a padronização de conexões com periféricos é muito importante. Os principais padrões são:

i. IEEE488 *Bus*, também conhecido por barramento de interface de uso geral (GPIB - *General Purpose Interface Bus*), geralmente empregado para interconexão de equipamentos de medição em laboratórios;

ii. EIA RS232 *Standard*, mais utilizado para interconexão de pequena distância e;

iii. O Padrão 4-20 mA, um dos padrões mais utilizados em meios industriais. Como uma variante do padrão RS232, para situações em que as distâncias envolvidas superam 15 metros e onde se deseja comunicação multi-pontos, existe o padrão RS485 que ainda proporciona maior imunidade a ruídos externos.

2.2. Principais Redes Industriais

Além das redes de informação e de controle, pode-se verificar ainda a existência das redes de campo, que são peças fundamentais para a comunicação em qualquer processo de automação industrial.

As redes de campo atendem pelo nome genérico de *fieldbus* ou barramento de campo. Na verdade, elas se dividem em 3 tipos:

1. Redes de sensores *ou Sensorbus* - são redes apropriadas para interligar sensores e atuadores discretos tais como chaves limites (*limit switches*), contactores, desviadores, etc. São exemplos de rede *Sensorbus*: ASI da Siemens, Seriplex, CAN e LonWorks.
2. Redes de Dispositivos *ou Devicebus* - são redes capazes de interligar dispositivos mais genéricos como CLPs, outras redes remotas de aquisição de dados e controle, conversores AC/DC, relés de medição inteligentes, etc. Exemplos: Profibus-DP, DeviceNet, Interbus-S, SDS, LonWorks, CAN, ControlNet, ModbusPlus.
3. Redes de instrumentação *ou fieldbus* - São redes concebidas

para integrar instrumentos analógicos no ambiente industrial, como transmissores de vazão, pressão, temperatura, válvulas de controle, etc. Exemplos: IECSP50-H1, HART, WorldFIP, Profibus-PA.

2.3. Aplicação no Ambiente Industrial

As redes de comunicação industriais estão divididas em níveis de aplicação, onde cada nível tem uma função diferente, conforme pode ser observado na Figura 3. No nível de dispositivos, encontram-se diversos equipamentos como: atuadores de válvulas, telemetria de tanques, inversores de freqüência, computadores de vazão etc. A camada de controle é composta principalmente por CLPs e IHMs, responsáveis em realizar o controle dos sistemas e processos industriais. E por fim, a camada de informação, que através dos servidores e equipamentos de comunicação como modem (Modulador Demodulador), satélites, celulares etc., aquisitam, armazenam e transferem as informações gerenciais dos processos.

Figura 3. Nível Redes Industriais

2.4. Padrões de Meio Físico RS

2.4.1. O RS232

"RS" é o acrônimo inglês para "*Recommended Standard*", ou em português "Padrão Recomendado". Ele relata uma padronização de uma interface comum para comunicação de dados entre equipamentos, tendo sido criado no início dos anos 60, por um comitê conhecido atualmente como Associação de Indústrias de Eletrônica (EIA - *Electronic Industry Association*). Naquele tempo, a comunicação de dados compreendia a troca de dados digitais entre um computador central (*mainframe*) e terminais de computador remotos, ou entre dois terminais sem o envolvimento do computador. Estes dispositivos poderiam ser conectados através de linha telefônica, e consequentemente necessitavam de um modem em cada lado para fazer a decodificação dos sinais.

Dessas idéias nasceu o padrão RS232, que especifica os níveis de tensões dos sinais, os comprimentos dos cabos, conectores, temporizações e funções dos sinais, um protocolo de *hardware* para troca de informações, e as conexões físicas. Há mais de 30 anos desde que essa padronização foi desenvolvida, a EIA publicou três modificações. A mais recente, EIA232E, foi introduzida em 1991. Ao lado da mudança de nome de RS232 para EIA232, algumas linhas de sinais foram renomeadas e várias linhas novas foram definidas.

Embora tenha sofrido poucas alterações, muitos fabricantes adotaram diversas soluções mais simplificadas que tornaram impossível a simplificação da padronização proposta. As maiores dificuldades encontradas pelos usuários na utilização da interface RS232 incluem pelo menos um dos seguintes fatores:

- A ausência ou conexão errada de sinais de controle, resultando em estouro do buffer (*overflow*) ou travamento da comunicação.
- Função incorreta de comunicação para o cabo em uso, resultando em inversão das linhas de Transmissão e Recepção, bem como a inversão de uma ou mais linhas de controle (*handshaking*).

Felizmente, os *drivers* utilizados são bastante tolerantes aos abusos cometidos, como por exemplo inversão de sinais e aplicação de tensão acima do especificado, e os Circuitos Integrado (CI) normalmente “sobrevivem”.

2.4.2. Definição de Sinais

Se a norma EIA232 completa for implementada, o equipamento que faz o processamento dos sinais é chamado DTE (*Data Terminal Equipment* – usualmente um computador ou um terminal), tem um conector DB25 macho, e utiliza 22 dos 25 pinos disponíveis para sinais e o terra. O equipamento que faz a conexão (normalmente uma interface com a linha telefônica) é denominado de DCE (*Data Circuit-terminating Equipment* – usualmente um modem), tem um conector DB25 fêmea, e utiliza os mesmos 22 pinos disponíveis para sinais e

terra. Um cabo de conexão entre dispositivos DTE e DCE contém ligações em paralelo, não necessitando mudanças na conexão de pinos. Se todos os dispositivos seguissem essa norma, todos os cabos seriam idênticos, e não haveria chances de haver conexões incorretas. Na Figura 4 é apresentado um esquema de conexão direta entre um PC (DTE) e um modem (DCE) e na Figura 5 é apresentado um esquema de conexão de dois PC (DTE) via modem (DCE).

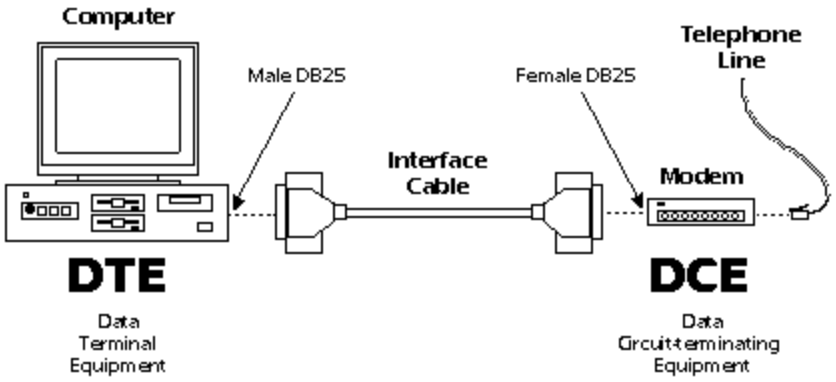

Figura 4. Esquema de Conexão direta DTE DCE

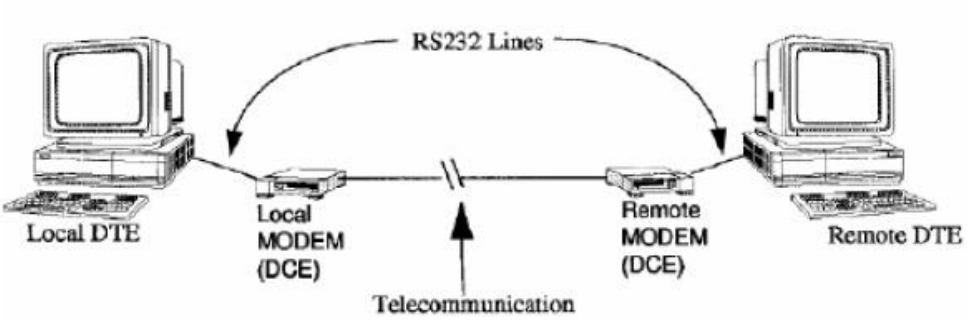

Figura 5. Esquema de Conexão modem DTE DCE

Diversos sinais são necessários para conexões onde o dispositivo DCE é um modem, e eles são utilizados apenas quando o protocolo de *software* os emprega. Para dispositivos DCE que não são modem, ou quando dois dispositivos DTE são conectados diretamente, poucos sinais são necessários. Deve-se notar nas figuras apresentadas que existe um segundo canal que inclui um conjunto de

sinais de controle duplicados. Este canal secundário fornece sinais de gerenciamento do modem remoto, habilitando a mudança de taxa de transmissão durante a comunicação, efetuando um pedido de retransmissão se erros de paridade forem detectados, e outras funções de controle. Na Figura 6a, é apresentada a definição dos sinais para um dispositivo DCE (usualmente um Modem), e na Figura 6b um dispositivo DTE, onde os sinais mais comuns são apresentados em destaque.

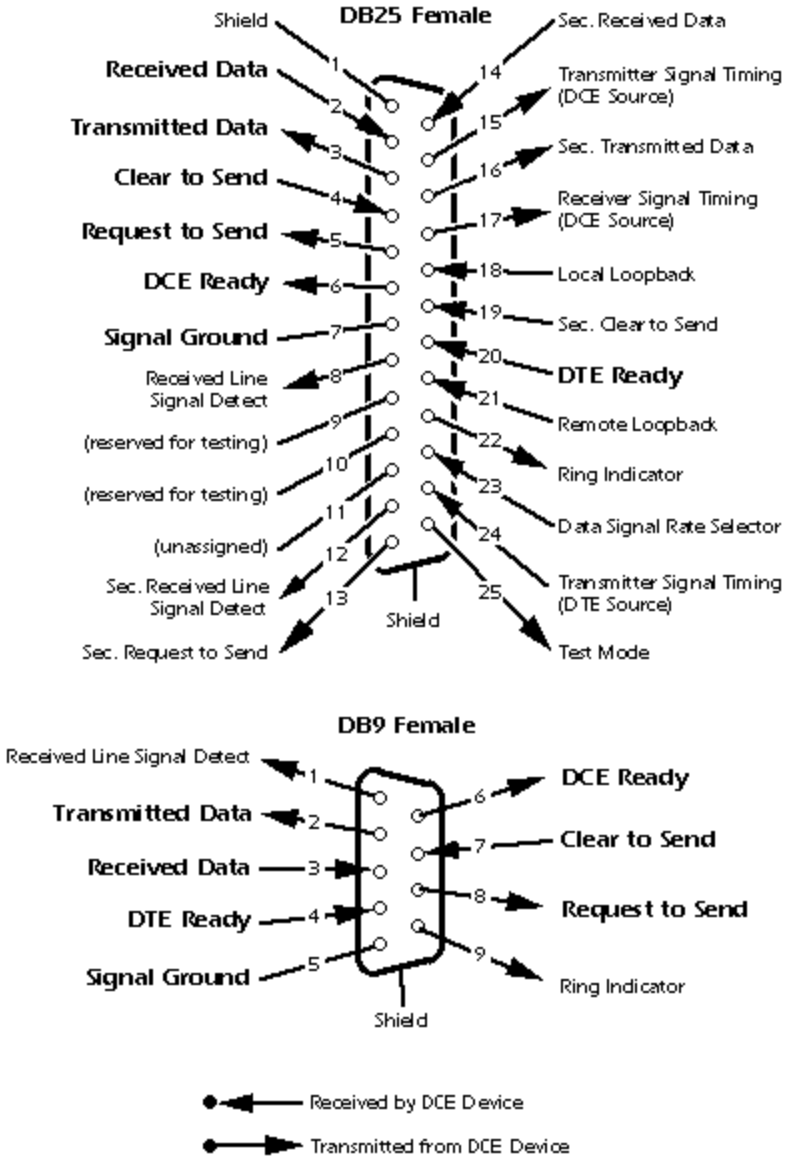

Figura 6a. Pinagem DB-25 fêmea e DB-9 fêmea RS232 (DCE)

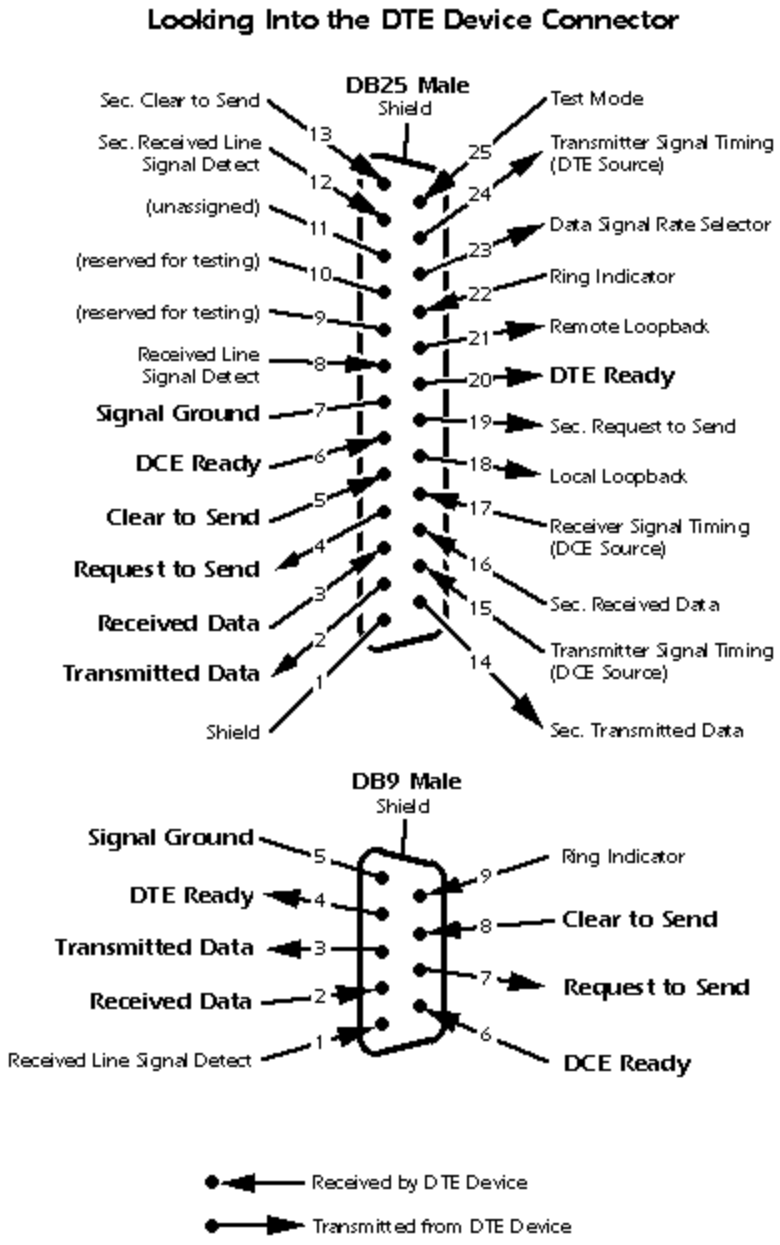

Figura 6b. Pinagem DB-25 macho e DB-9 macho RS232 (DTE)

Os sinais de temporização de transmissão e recepção são utilizados somente quando o protocolo de transmissão utilizado for síncrono. Para protocolos assíncronos, padrão 8 bits, os sinais de temporização externos são desnecessários. Os sinais que implicam em um direção. Como *Transmit Data* e *Receive Data*, são nomeados do ponto de vista dos dispositivos DTE. Se a norma EIA232 for seguida a risca, estes sinais terão o mesmo nome e o mesmo número

de pino do lado do DCE. Infelizmente, isto não é feito na prática pela maioria dos engenheiros, provavelmente porque em alguns casos torna-se difícil definir quem é o DTE e quem é o DCE. A Figura 7 apresenta a convenção utilizada para os sinais mais comuns.

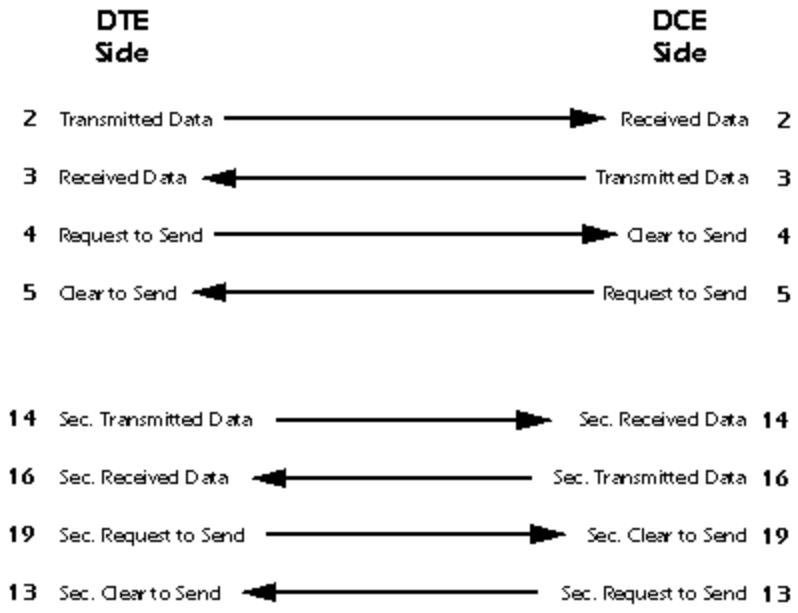

Figura 7. Principais sinais DTE DCE

2.4.3. Sinal de Terra Comum

A norma EIA232 inclui a referência de terra no pino 5 do conector DB-9 e pino 7 do DB-25, e é freqüentemente conectada ao Pino 1 a blindagem do cabo que envolve os demais condutores. Sinais de tensão dos dados, temporizações e controle são medidos com relação a esse terra comum. Equipamentos que usam a interface RS232 não podem ser utilizados em aplicações que exijam que os dois equipamentos (mestre e escravo) estejam eletricamente isolados. Nestes casos isoladores ópticos podem ser usados para garantir a isolação. Contudo, isso não é mencionado ou incluído na especificação da norma EIA232.

2.4.4. Características do Sinal

Sinais com tensão entre –3 volts e –25 volts com relação ao terra (pino 7) são considerados nível lógico “1” (condição marca), e tensões entre +3 volts e +25 volts são considerados nível lógico “0” (condição espaço). A faixa de tensões entre –3 volts e +3 volts é considerada uma região de transição para o qual o estado do sinal é indefinido. Na Figura 8 são representados estes níveis de tensão.

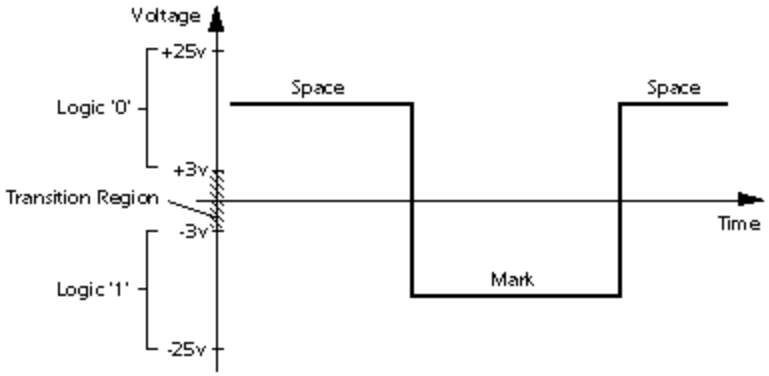

Figura 8. Característica de Sinal RS232

2.4.5. Conversores de nível TTL – RS232

A maioria dos equipamentos digitais utilizam níveis TTL, que significa ***T****ransistor-****T****ransistor – ****L****ogic* (Lógica Transistor-Transistor em português), ou CMOS, que significa ***C****omplementary ****M****etal ****O****xide ****S****emiconductor* (Semicondutor de Óxido-Metal Complementar). Portanto, o primeiro passo para conectar um equipamento digital a uma interface RS232 é transformar níveis TTL (0 a 5 volts) em RS232 e vice-versa. Isto é feito por conversores de nível. Existe uma variedade grande de equipamentos digitais que utilizam o *driver* 1488 (TTL ? ? RS232) e o *receiver* 1489 (RS232 ? ? TTL). Estes CIs contém 4 inversores de um mesmo tipo, sejam *drivers* ou *receivers*.

O *driver* necessita de duas fontes de alimentação +7,5 volts a +15 volts e –7,5 volts a –15 volts. Isto é um problema onde somente uma fonte de +5 volts é utilizada. Um outro CI que está sendo largamente utilizado é o MAX232 (da Maxim). Ele inclui um circuito de *charge pump* (bomba de carga) capaz de gerar tensões de +10 volts e –10 volts a partir de uma fonte de alimentação simples de

+5 volts, bastando para isso alguns capacitores externos. Este CI também possui 2 *receivers* e 2 *drivers* no mesmo encapsulamento. Nos casos onde serão implementadas somente as linhas de transmissão e de recepção de dados, não seria necessário 2 *chips* e fontes de alimentação extras.

2.5. Comunicação RS485

A comunicação RS485 funciona em modo diferencial. Ou seja, a diferença entre as tensões na linha informarão se o mestre está transmitindo níveis lógicos 1 ou 0. A RS485 suporta a comunicação *half-duplex* e *full-duplex* sendo que para a primeira há a necessidade da utilização de um cabo par-trançado, enquanto na segunda são necessários dois pares de cabos. Este tipo de comunicação pode alcançar grandes distâncias de cabo, podendo chegar até 1200m funcionando a uma taxa de transmissão de 9600 bps (bits por segundo). Conforme o *baud-rate* (taxa de transmissão) aumenta, o tamanho do cabo diminui, pois quanto maior a freqüência, maior a atenuação do sinal ao longo do cabo.

Este meio utiliza a estrutura de comunicação mestre-escravo, onde há uma máquina que faz a pergunta (mestre) e os escravos respondem. Sendo que o pacote de dados (*frame*) trafega por toda a rede até encontrar o equipamento escravo que tenha mesmo endereço de identificação.

O cabo para a comunicação em RS485 é composto de dois fios, sendo um destes chamado de A e o outro de B. a seguir-se uma tabela que mostra os estados lógicos das linhas A e B de acordo com o dado que o transmissor queira enviar:

DRIVER

INPUT D	ENABLE DE	OUTPUTS	
		A	B
H	H	H	L
L	H	L	H
X	L	Z	Z

Figura 9. Estado Lógico Sinais RS485

Verifica-se que, quando o transmissor (INPUT D) fica em alto, a linha A fica mais positiva que a B e o inverso ocorre quando o estado inverte. Nota-se que também há uma linha de controle chamada DE e quando a mesma fica em nível lógico baixo, o barramento fica em alta impedância.

Para que o receptor identifique um sinal válido, a diferença entre os terminais A e B deve ser maior que 200 mV. Entre 200mV e –200mV o sinal é indefinido.

Na Figura 10, tem-se um gráfico que mostra a transmissão do conteúdo binário 01001 a uma taxa de 9600 bps no barramento RS485.

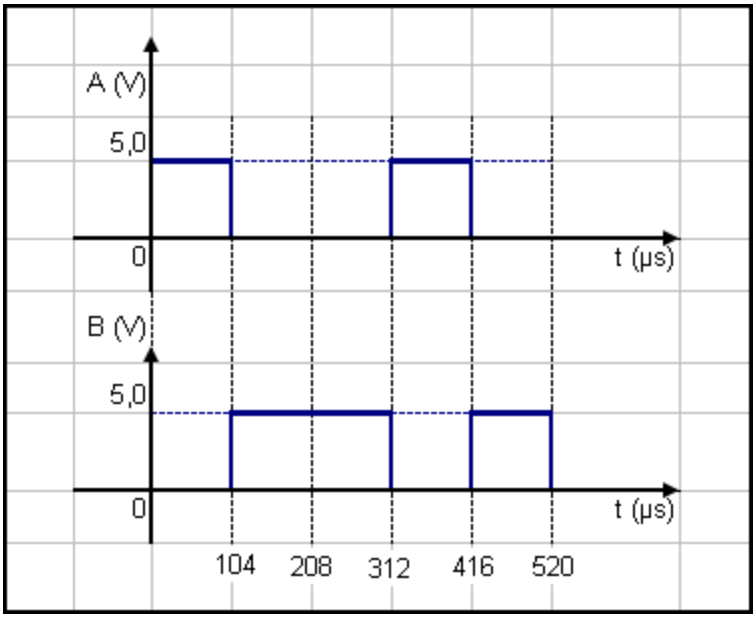

Figura 10. Transmissão de Dados RS485

3. Protocolo Modbus

3.1. Definição

MODBUS é um protocolo aberto de comunicação industrial de fato, inicialmente desenvolvido pela Modicon ® e introduzido no mercado em 1979, atual *Shneider Eletric*, para comunicação tanto entre CPs (Controladores Programáveis), instrumentos e atuadores, como entre CPs e IHMs. Devido ao baixo custo e simplicidade, tornou-se um dos protocolos mais utilizados na indústria.

Atualmente, o padrão é mantido pela MODBUS-IDA [1], uma sociedade independente e sem fins lucrativos, que possui como membros usuários e fornecedores de equipamentos de automação.

Este protocolo de mensagem situa-se na sétima camada do modelo OSI e conecta dispositivos em uma arquitetura Cliente/Servidor ou Mestre/Escravo. O quê oferece serviços especificados por códigos de funções em transações tipo requisição e resposta, que constituem os elementos do PDU (*Protocol Data Unit*). Esta comunicação é implementada através de diferentes tipos de redes e camadas físicas, tais como TCP/IP sobre *Ethernet* ou transmissão serial assíncrona (EIA/TIA-232, 422 ou 485).

Em função do mecanismo de transmissão, o MODBUS-IDA classifica o protocolo Modbus em quatro tipos:

1. MODBUS RTU: transmissão serial através de formato binário.
2. MODBUS ASCII: transmissão serial utilizando caracteres ASCII. Existe um maior consumo de recursos em relação ao item anterior.
3. MODBUS TCP/IP: utiliza como meio de comunicação o padrão *Ethernet*.
4. O MODBUS Plus é um protocolo fechado sob o domínio da *Schneider Eletric*.

A Figura 11 ilustra a flexibilidade do uso do MODBUS para conexão de diversos dispositivos. A utilização do MODBUS não se resume apenas aos dispositivos mostrados na Figura 11, pois com a utilização de *gateways* [2], pode-se realizar conexão com outros tipos de redes, que utilizam diferentes protocolos.

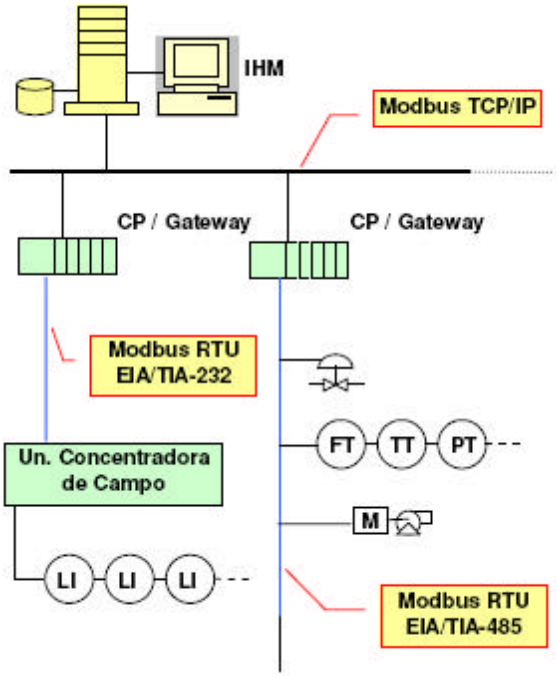

Figura 11. Conexão de Dispositivos Modbus

3.2. Descrição do Protocolo

A Figura 12 ilustra a comparação do protocolo com o modelo OSI. Pode-se observar que o Modbus/RTU atende apenas às camadas física, enlace e de aplicação do modelo OSI.

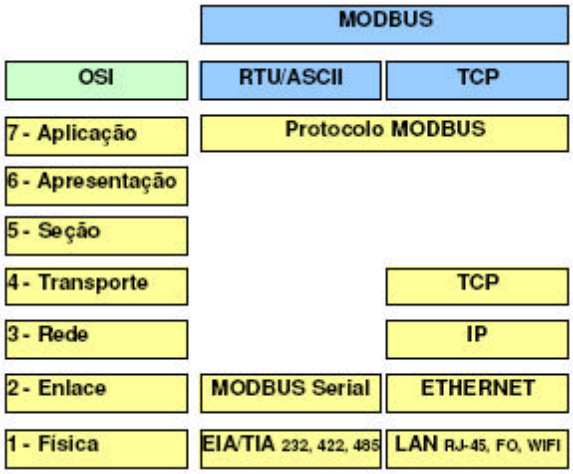

Figura 12. Camadas do Protocolo MODBUS.

O MODBUS é essencialmente um protocolo de troca de mensagens para uma arquitetura Cliente/Servidor (Mestre/Escravo), cuja comunicação, que é iniciada pelo Cliente (mestre), se faz através de *frames* conforme a Figura 13.

Figura 13. Visão geral do *frame* genérico de comunicação.

O núcleo básico da comunicação é a Unidade de Dados de Protocolo **(PDU - *Protocol Data Unit*),** , que é definido para a camada de aplicação e independente das camadas inferiores. Já a Unidade de Dados de Aplicação **(ADU - *Application Data Unit*),** , que é formada do PDU acrescida de campos que são dependentes do tipo de rede e barramento utilizado.

O Código da Função é determinado pelo Cliente (Mestre) e possui tamanho de 1 byte, com valores válidos de 1 a 255 decimal, sendo que 0 (zero) não é um código de função válido e os valores compreendidos no intervalo de 128

a 255 são reservados para respostas a exceções. Este código determina que tipo de ação o Servidor (Escravo) deve executar. Sub funções podem ser adicionadas para definir múltiplas ações.

O campo de dados do PDU que é enviado ao Servidor conté m informações adicionais sobre a tarefa a ser executada que complementam o código de função, especificando endereços, itens discretos ou registros, quantidades a serem “manipulados” e quantidade de bytes existe ntes no campo. Dependendo da situação, este campo pode ter comprimento nulo, quando a função a ser executada não possui nenhum tipo de informação complementar para o seu entendimento.

A Figura 14 ilustra os processos de comunicação entre as partes para uma transação realizada com sucesso e para outra realizada com erro.

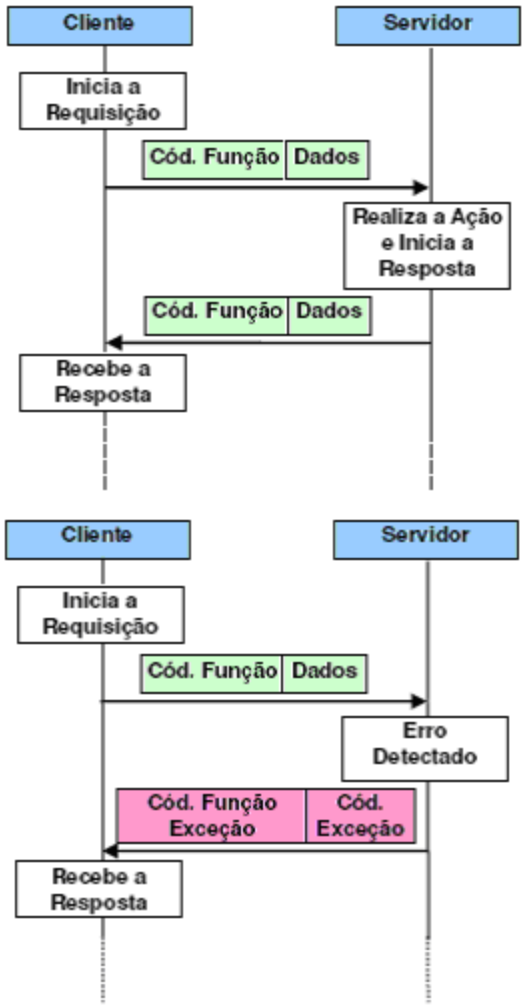

Figura 14. Processo Típico de Comunicação.

Caso o processo de comunicação ocorra sem problema, o Servidor (Escravo) retorna para o Cliente (Mestre) um eco do código da função de requisição. Caso alguma anormalidade ocorra, é retornado um código similar ao código de requisição, com o bit mais significativo posicionado no nível lógico 1. Este valor é obtido através da soma de 0x80 (128) ao código da função requisitada. Em resumo, as funções válidas encontram-se no intervalo [1, 127] com os respectivos códigos de erro no intervalo [129 **(1+128)**, 255 **(127+128)**].

O Cliente (Mestre) deve implementar também uma rotina de detecção

de tempo de resposta, que trate as situações para as quais o retorno do Servidor nunca ocorra.

3.3. MODBUS Serial

A Figura 15 ilustra as topografias possíveis para o MODBUS Serial em função do tipo de conexão.

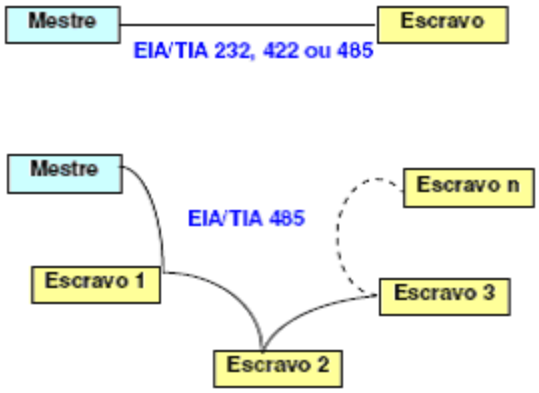

Figura 15. Topologias possíveis para o MODBUS Serial.

O padrão EIA/TIA 232 é utilizado para comunicação ponto a ponto de curta distância. O 422 é uma extensão do 232, que permite comunicação bidirecional, maiores distâncias e velocidades. Já o 485, além da distância maior, permite a formação de uma rede.

O tamanho do ADU para o MODBUS RTU é limitado a 256 *bytes* (variável de 8 bits), em decorrência do limite imposto pelas versões projetadas para redes EIA/TIA-485. A Figura 16 ilustra a composição do ADU para o MODBUS serial.

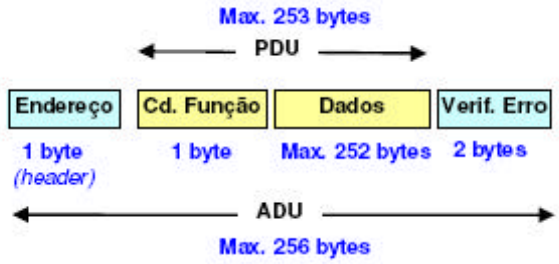

Figura 16. Detalhe do frame para MODBUS Serial.

O MODBUS Serial – RTU a comunicação binária permite uma compactação da informação. As mensagens são iniciadas e concluídas com um intervalo de silêncio mínimo de 3,5 bytes e deverá ter uma pausa máxima entre bytes de 1,5 byte. A verificação de erro é feita através de um CRC (*Cyclic Redundancy Check*), verificação de redundância cíclica, de dois *bytes*.

Na comunicação através do MODBUS Serial – ASCII, cada *byte* do frame é codificado através de 02 caracteres ASCII, representando a notação hexadecimal de cada *byte* (0-9, A-F). A verificação de erro é obtida através de LRC (*Longitudinal Redundancy Check*). Cada mensagem possui três caracteres a mais, pois é iniciada com um caractere ‘:’ (0x3A) e finalizadas com um caractere de retorno de carro ‘CR’ (0x0D) e um caractere de avanço de linha ‘LF’ (0x0A). Pausas de transmissão entre caracteres de até 1 segundo podem ocorrer. O PDU terá, então, 2 caracteres para o código de função e um máximo de 2 x 252 caracteres para dados. Para o endereço, faz-se necessário 2 caracteres e para a verificação de erro outros 4 caracteres, resultando em um ADU de até 513 caracteres. O *byte* de endereçamento pode assumir valores no intervalo [0, 255], sendo que:

a) 0 – endereço para comunicação *broadcast*.

b) [1, 247] – endereços para comunicação *unicast*.

c) [248, 255] – endereços reservados.

O mestre é único em uma rede, não possui endereço e sempre inicia a comunicação. Os escravos devem possuir endereços únicos no intervalo de 1 a

247, só emitem mensagem após a solicitação do mestre e nunca se comunicam entre si.

Na **comunicação *broadcast***, que representa o envio de uma mensagem do mestre para todos os dispositivos escravos pertencentes à rede, não existe resposta dos escravos que individualmente devem aceitar o comando do mestre, o qual deverá ser uma função de escrita. Já na **comunicação *unicast*** as mensagens são endereçadas a apenas um equipamento escravo especifico e realizadas através de requisições e respostas entre o mestre e o escravo respectivo.

No MODBUS Serial, tanto o mestre quanto o escravo possuem estados definidos, ilustrados nas Figuras 17 e 18, respectivamente. Quando o mestre envia uma mensagem em *broadcast*, o tempo de atraso "*Turnaround Delay*" deve ser suficiente para que os escravos executem a ação solicitada em menor tempo que o *timeout* para comunicação *unicast*. Em geral, o primeiro situa-se na faixa dos décimos de segundos e o outro na faixa de segundos.

Os escravos devem implementar contadores de erros para permitir diagnósticos do sistema.

Figura 17. Estados do Mestre.

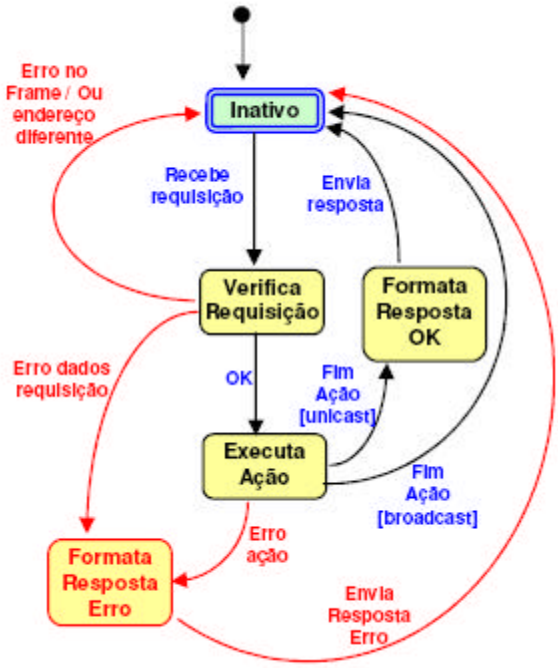

Figura 18. Estados do Escravo.

Taxas de transmissão de 9,6 Kbps e 19,2 Kbps (padrão) são requeridas.

No entanto, é possível realizar conexões desde 1,2 Kbps a mais de 115 Kbps, dependendo do tipo de padrão físico, quantidade de dispositivos e distância entre eles.

3.4. Modelo de Dados

A Tabela 1 ilustra os tipos de dados do MODBUS.

Tabela1. Modelo de dados do MODBUS.

Nome	Tipo	Acesso
Entrada Discreta (Status)	1 bit	Apenas Leitura
Saída Discreta (Coils)	1 bit	Escrita / Leitura
Registro de Entrada (Input Registers)	16 bits	Apenas Leitura
Registro Retentivo (Holding Registers)	16 bits	Escrita / Leitura

A Entrada Discreta é constituída por um único bit e pode ser associada a um contato seco de relé ou mesmo uma chave que se encontra aberta ou fechada, assumindo os níveis lógicos 0 ou 1.

A Saída Discreta é similar à Entrada discreta, no entanto pode ser escrita ou lida. Sua analogia imediata ao mundo real conduz às bobinas (*coils*) dos relés.

O Registro de Entrada possui sua informação armazenada em 16 bits, o que equivale a uma palavra (*word*), que pode representar tanto um número inteiro, quanto um número de ponto flutuante no formato IEEE 754 com 32 ou 64 bits, 2 ou 4 palavras. Um exemplo de dispositivo do mundo real que poderia ser associado a este modelo seria um transmissor ou conversor analógico digital. O Registro Retentivo possui comportamento idêntico ao Registro de Entrada, com a diferença de que além de permitir leitura, permite escrita. Um exemplo de dispositivo físico do mundo real que incorpora este modelo é um *set-point* (valor desejado) de um atuador industrial.

O mapeamento da memória do endereço MODBUS para o endereço do

dispositivo (Escravo / Servidor) pode ser feito para blocos independentes ou um único bloco como mostra a Figura 19. Este mapeamento, que é definido pela aplicação que é executada no dispositivo, deve ser fornecido pelo fabricante ou projetista do mesmo.

Este mapeamento serve de base para o desenvolvimento e configuração da aplicação executada no Cliente (Mestre).

3.5. Funções

Existem três categorias de funções:

a) Funções Públicas: são funções bem definidas, publicamente documentadas pela comunidade MODBUS-IDA e tendo a garantia de serem únicas, possuem testes de conformidade. Esta categoria inclui não só funções já definidas como reserva códigos para futuras definições.

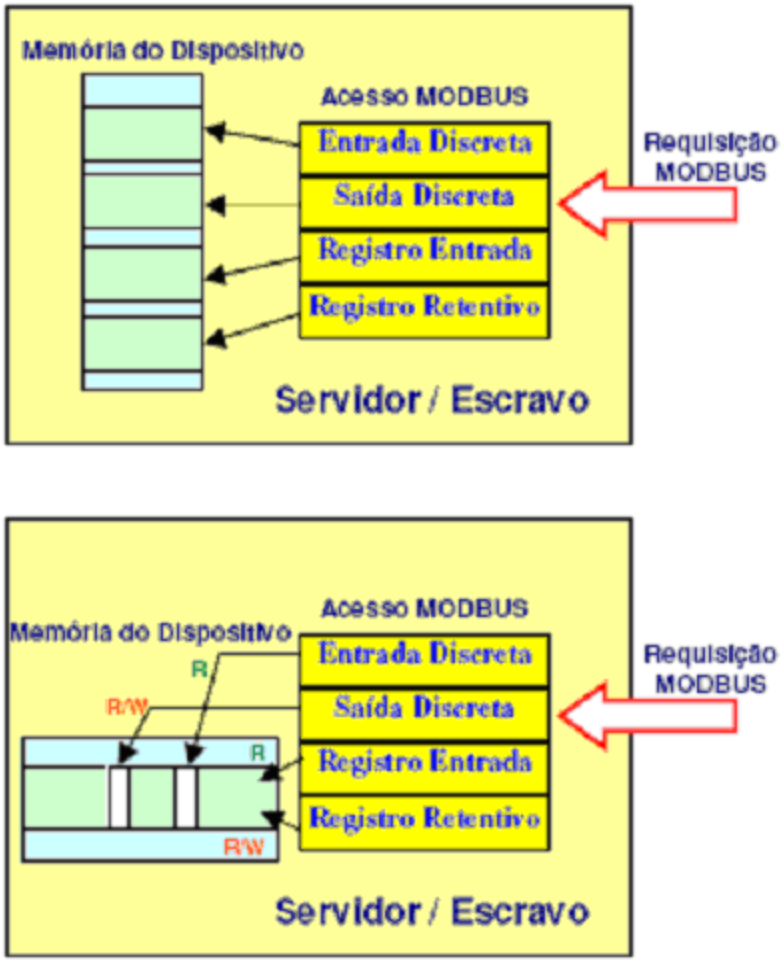

Figura 19. Mapeamento de Memória.

b) Funções Definidas pelo Usuário: são funções customizadas que não possuem garantia de serem únicas.

c) Funções Reservadas: são funções legadas cujos códigos são utilizados por algumas empresas não fazendo parte da atual especificação da MODBUS-IDA.

A MODBUS IDA classifica as funções com públicas ou definidas pelo usuário (Figura 20).

Normalmente as funções exigem um complemento, como sub códigos ou endereços dos dados. A tabela 2 ilustra a distribuição dos códigos destas categorias, as funções reservadas não são mostradas por não fazerem parte da

especificação. Na prática, as funções mais utilizadas são as de acesso a dados.

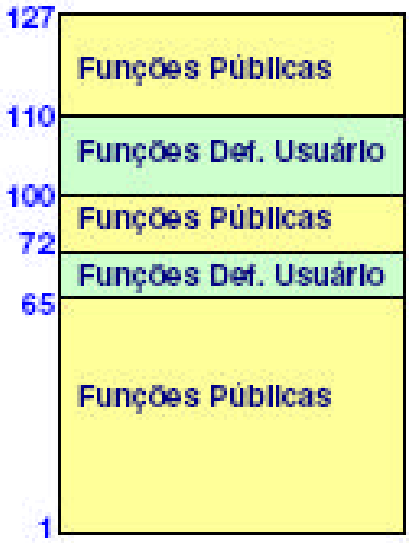

Figura 20. Classes de função do MODBUS – IDA.

Tabela2. Principais funções utilizadas.

	Tipo	Função	Código
bits	Entradas Discretas (Status)	Leitura múltipla	02
bits	Saídas Discretas (Coils)	Leitura múltipla	01
bits		Escrita única	05
bits		Escrita múltipla	15
words	Registros Entrada (Input Registers)	Leitura múltipla	04
words	Registros Retentivos (Holding Registers)	Leitura múltipla	03
words		Escrita única	06
words		Escrita múltipla	16

4. Modelo do Sistema Proposto

4.1. Exemplo de Arquitetura de Redes Modbus/RTU

Na Figura 21, é apresentada a arquitetura geral de um sistema de automação composto por uma comunicação multiponto RS485 e comunicações ponto-a-ponto em RS232, RS422 e RS485, todos utilizando o protocolo

Modbus/RTU.

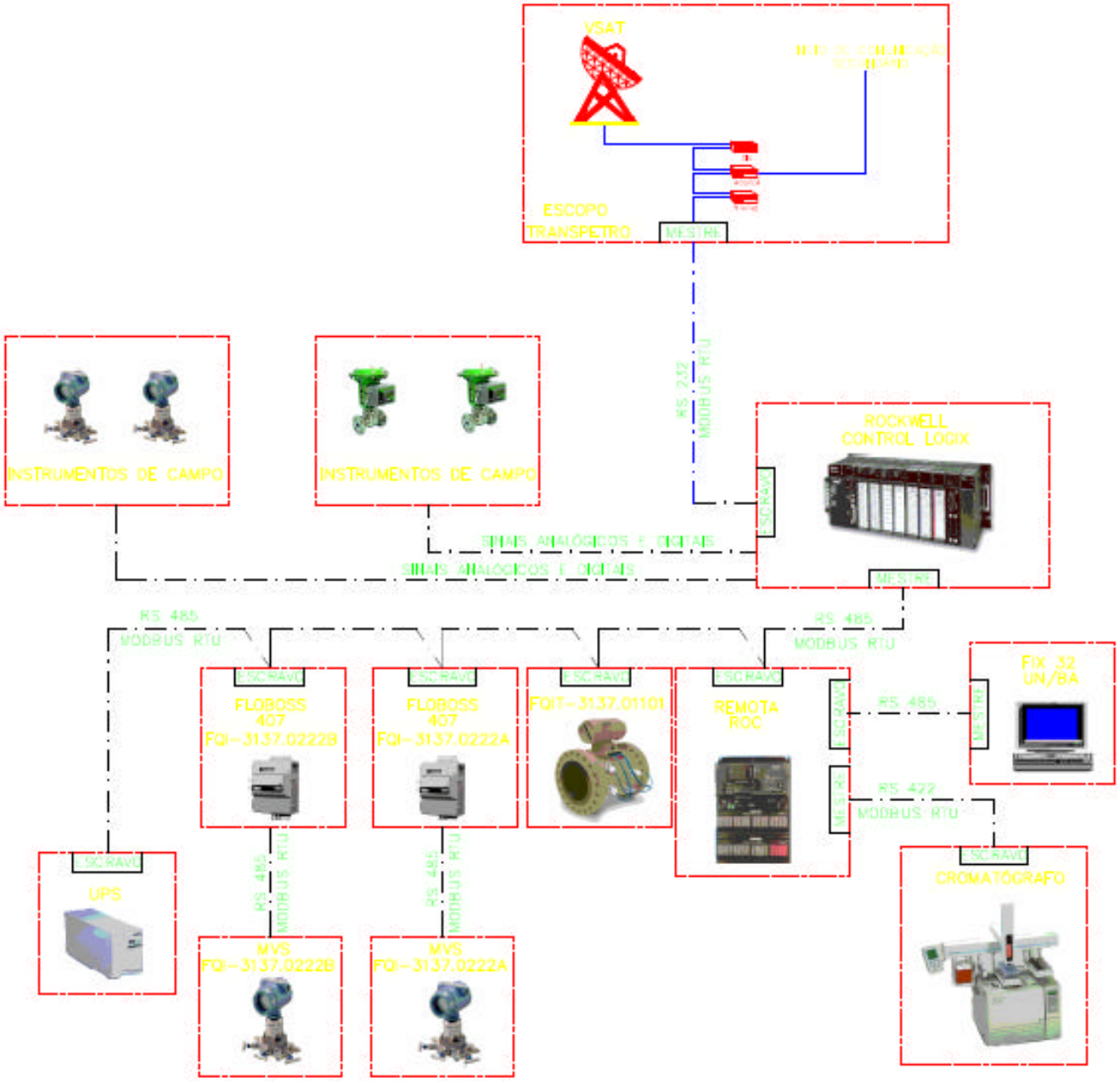

Figura 21. Exemplo de arquitetura de rede industrial.

4.2. Método de Aplicação

Na Figura 22, tem-se a simplificação de um sistema onde se experimenta o funcionamento de uma comunicação do tipo ponto-a-ponto entre

um CLP (mestre da comunicação) e um transmissor de vazão (escravo da comunicação).

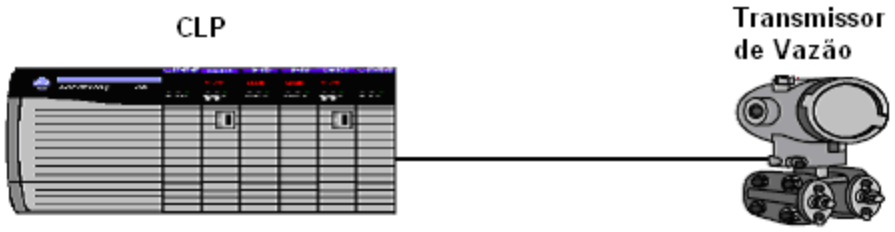

Figura 22. Comunicação CLP – Transmissor.

4.2.1. Cenário ? 01 - Ensaio com o Mestre Modbus

No primeiro cenário simula-se o funcionamento do transmissor de vazão (escravo da comunicação) se comunicando com um CLP (mestre da comunicação), conforme a Figura 23.

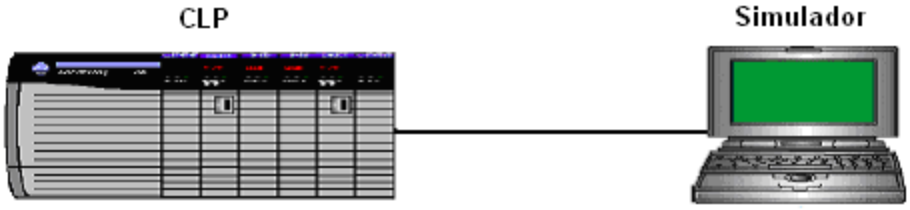

Figura 23. Comunicação CLP - Simulador

4.2.2. Cenário ? 02 - Ensaio com o Escravo Modbus

Já no segundo cenário foi simulado o funcionamento do CLP (mestre da comunicação) se comunicando com o transmissor de vazão (escravo da comunicação), conforme a Figura 24.

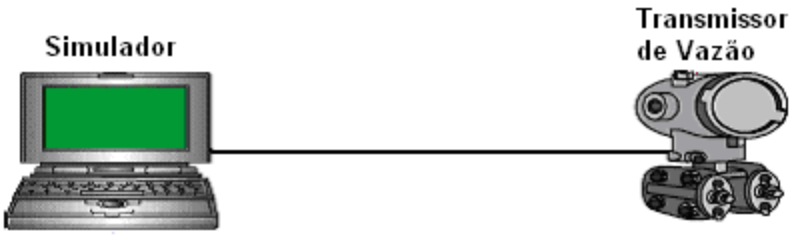

Figura 24. Comunicação Simulador – Transmissor.

4.3. Simulador

Foi desenvolvido um *software* aplicativo para operar em computadores no ambiente *Windows*. Este aplicativo contém as principais funções e características de equipamentos mestre ou escravo Modbus/RTU e poderá simular a comunicação entre eles.

4.3.1. Funcionalidade

Do ponto de vista de *software*, o sistema desenvolvido possui dois componentes principais: o protocolo de comunicação e a IHM para a realização da configuração e visualização dos dados monitorados. O *software* dispõe de propriedades de configuração da porta de comunicação serial, modo de funcionamento, mestre ou escravo, funções e endereços de registradores Modbus e outras.

Para a realização da comunicação, foi utilizado o protocolo de comunicação serial Modbus/RTU e o padrão de meio físico RS232, podendo este meio físico ser convertido através de um conversor RS232/RS485, quando o equipamento que queira comunicar possua este tipo de padrão.

Quando o computador utilizado não dispuser de porta de comunicação serial, deverá ser utilizado um conversor USB/Serial.

4.4. Análise e Discussão dos Resultados

No desenvolvimento do trabalho pretende-se fazer uma análise crítica e discussão do funcionamento da rede Modbus/RTU em chão de fábrica, apresentando o comportamento da mesma, os principais defeitos observados, os

pontos positivos e negativos, e quantificar a redução do tempo de parada de equipamentos que o sistema poderá contribuir.

4.5. Perspectivas

4.5.1. Impactos Esperados

Dentre os impactos esperados, pode-se citar o impacto tecnológico, onde uma nova ferramenta e metodologia será disponibilizada possibilitando através de sua utilização a realização de treinamento de profissionais, e em decorrência deste temos também um ganho econômico considerável atrelado ao fato de que as manutenções destes sistemas poderão ser realizadas em menor tempo, e com isso espera-se uma redução do tempo de parada de produção, culminando em menor perda de matéria prima e insumos de processos, e ainda com o uso deste sistema os próprios técnicos das empresas que possuem este tipo de rede podem reparar os defeitos dos equipamentos, sem ter que a empresa arcar com altos custos de contratação de serviços de suporte técnico dos fabricantes dos mesmos, e o mais importante, uma vez que é uma solução de baixo custo pode ser facilmente implementada.

5. Cronograma de Atividades

No cronograma de atividades, tabela 3, foram listadas as principais etapas para a realização do trabalho, contendo os prazos estimados para a sua realização.

Tabela 3. Cronograma de Atividades.

Etapas	Atividades	Período								
		2007			2008					
		Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun
1	Definição de escopo									
2	Pesquisa e Levantamento Bibliográfico									
3	Estudo Teórico do Protocolo Modbus									
4	Projeto da Solução									
5	Programação do Software									
6	Testes									
7	Validação									
8	Monografia									
9	Defesa do TCC									

Legenda	
Previsto	
Realizado	

6. Conclusão

O *software* implementado possui as características principais necessárias para avaliação e diagnostico de uma comunicação Modbus/RTU. Os equipamentos utilizados apresentaram um ótimo funcionamento, os testes realizados foram considerados satisfatórios, por isso o sistema proposto pode ser utilizado no ambiente industrial como ferramenta auxiliar para avaliação e diagnostico de falhas de comunicação em redes com padrão RS232/485 e protocolo Modbus/RTU.

Referências

ALVES, W. P. **C++Builder 6**: desenvolva aplicações para Windows. 2ª ed. São Paulo: Érica, 2007.

ALPHA INSTRUMENTOS. **Comandos de pesagem para Modbus RTU/ASCII**. São Paulo : 2004. (Rev. 2).

Anybus Fieldbus & Industrial Ethernet. Solutions.

Disponível em: http://www.hms-networks.com. Acessado em: 3 set. 2007.

BEZERRA, M. A. D. **Protocolo Modbus.** Universidade Federal da Bahia, Programa de Pós-Graduação em Mecatrônica. Artigo Técnico, 20 de Setembro de 2007.

Canzian, Edmur. **Comunicação serial - RS232**. CNZ Engenharia e Informática. Disponível em: http://www.cnz.com.br. Acessado em: 01 set. 2007.

JAMOD. Disponível em: http://jamod.sourceforge.net/. Acessado em: 10 set. 2007.

KRON INSTRUMENTOS ELÉTRICOS. **MKM-01 - Protocolo Modbus**: manual do cliente. São Paulo: Alpha Instrumentos, 2004. (Relatório Técnico - Rev. 1.1).

LIMA, Valter. **Manual prático do seu PC**. São Paulo: Érica, 1999.

MANZANO, J. A. N. G. **Estudo dirigido de C++Builder 6**. São Paulo: Érica, 2003.

MODBUS-IDA. **Modbus application protocol specification:** v1.1a. North Grafton-MA: 2004. (Relatório Técnico). Disponível me HTTP: http://www.modbus-ida.org. Acesso em 28 de Dezembro de 2006.

MODBUS Application Protocol Specification.V1.1b. MODBUS-IDA. 2006. Disponível em http://www.modbus-ida.org.

MODBUS over serial line specification and implementation guide V1.0. MODBUS-IDA. 2002. Disponível em http://www.modbus-ida.org.

NATALE, Ferdinando. **Automação industrial**. São Paulo: Érica, 2000.

SOUZA, David José de. **Desbravando o PIC**. São Paulo: Érica, 2000.

TANENBAUM, A.S **Computer networks**.4ª Ed. New Jersey: Prentice Hall,1996.

_____. **Redes de computadores**. 4ª Ed, Rio de Janeiro: Campus, 2003.

WinTECH. Disponível em: http://www.win-tech.com/. Acessado em: 21 set. 2007.