



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
UNIDADE ACADÊMICA DE ENGENHARIA ELÉTRICA**

DEVICENET: PROTOCOLO E APLICAÇÕES

VICTOR LHINCIO SALES BRITO

CAMPINA GRANDE, JULHO DE 2008



Biblioteca Setorial do CDSA. Fevereiro de 2021.

Sumé - PB

VICTOR LHINCIO SALES BRITO

DEVICENET: PROTOCOLO E APLICAÇÕES

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, em cumprimento às exigências para obtenção do Grau de Engenheiro Eletricista.

ORIENTADOR:
Edmar Candeia Gurjão, D. Sc., UFCG

CAMPINA GRANDE, JULHO DE 2008

VICTOR LHINCIO SALES BRITO

DEVICENET: PROTOCOLO E APLICAÇÕES

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, em cumprimento às exigências para obtenção do Grau de Engenheiro Eletricista.

Edmar Candeia Gurjão, D. Sc., UFCG

Bruno Barbosa Albert, D. Sc., UF

CAMPINA GRANDE, _____ DE JULHO DE 2008

DEDICATÓRIA

Aos meus pais, sólidos alicerces da minha formação e exemplos de vida.

À minha namorada, sempre tão atenciosa.

Às minhas irmãs, companheiras e vibrantes com meus êxitos.

AGRADECIMENTOS

Embora haja apenas um nome na capa deste trabalho de conclusão de curso, não há dúvidas que o mesmo é fruto de um esforço combinado e distribuído ao longo de minha vida de inúmeros colaboradores.

A Deus, que nas mais diversas formas, intensidades e situações sempre se mostrou vigilante e atuante na minha vida pessoal e acadêmica. Agradeço-lhe Senhor por cada oportunidade que me foi concedida e cada aprendizado que pude acumular nessas situações.

Agradeço com amor, o amor, a compreensão, a formação pessoal e a educação de qualidade que meus pais me concederam. Agradeço-lhes pelo simples fato de, em momento algum, mensurarem esforços para que alcançasse meus objetivos.

Lívia e Laísa, minhas irmãs, que acreditam no meu potencial e sempre se mostraram compreensivas nos momentos mais complicados da minha vida acadêmica.

Maiara, pelo amor dedicado e pela paciência em suportar as ausências de um acadêmico de Engenharia Elétrica.

O Professor Edmar Candeia Gurjão, pela valorosa orientação na realização deste trabalho.

Os professores e funcionários da Unidade Acadêmica de Engenharia Elétrica e da UFCG que com seu trabalho, esforço e dedicação formam bons engenheiros eletricitas.

Todos os amigos e colegas que tenho colecionado aqui na UFCG.

"Não queiras ser como aquele cata-vento dourado do grande edifício; por muito que brilhe e por mais alto que esteja, não conta para a solidez da obra. Oxalá seja sempre como um velho silhar oculto nos alicerces, debaixo da terra, onde ninguém te veja; por ti não desabará a casa."

Josemaría Escrivá

RESUMO

Um protocolo de comunicação industrial nada mais é do que uma tecnologia, empregada em processos produtivos, que disponibiliza informações sobre estes processos para entidades (pessoas ou computadores), a fim de supri-lhes como dados para tomada de decisões.

Porquanto que, através desses dados, inúmeras decisões, das mais simples às mais complexas são tomadas, é que a tecnologia empregada para captação e transmissão desses dados deve oferecer a maior confiabilidade possível, além de boa velocidade e baixo custo operacional. Neste cenário, surge em 1994, a tecnologia *DeviceNet*. Tal inovação destina-se a comunicação entre equipamentos do nível mais baixo do padrão OSI, a camada de dispositivos de medição, sensoriamento e atuação diretamente sobre os processos.

A tecnologia *DeviceNet*, baseada numa tecnologia criada com o mesmo objetivo, embora visando automóveis de alto desempenho (o protocolo CAN), alia as características de modularidade, baixo custo de implantação e operação, o que torna-a bastante interessante do ponto de vista econômico. Além disso, sua característica de ser um protocolo aberto permite que inúmeros fabricantes ofereçam equipamentos e dispositivos compatíveis com a *DeviceNet* e isto barateia e facilita ainda mais sua implantação e manipulação.

Neste trabalho objetiva-se estudar e reunir informações seguras sobre a *DeviceNet*, tendo em vista que esta tecnologia é relativamente nova e hoje desponta como uma das mais avançadas dentre aquelas destinadas à comunicação industrial. Foi feito, também, um apanhado das vantagens e desvantagens deste protocolo.

Para ilustrar, um estudo de caso foi realizado abordando a rede que foi instalada e hoje se encontra em operação nas dependências da CIPASA, uma indústria de cimentos localizada no estado da Paraíba. Ao contemplar aspectos como objetivos do projeto da rede e procedimentos de manutenção da mesma, uma análise dos benefícios da adoção da tecnologia *DeviceNet* foi feita.

LISTA DE FIGURAS

Figura 1 – Arquitetura tradicional de rede industrial	5
Figura 2 – Classificação das Redes	7
Figura 3 – Topologia da rede CAN	8
Figura 4 – Topologia da rede <i>DeviceNet</i>	15
Figura 5 – Cabos utilizados na rede <i>DeviceNet</i>	16
Figura 6 – Rede <i>DeviceNet</i> usada para exemplificação do 1º método de dimensionamento de cabos	17
Figura 7 – Aterramento de uma rede <i>DeviceNet</i>	20
Figura 8 – Processo de arbitragem na <i>DeviceNet</i>	23
Figura 9 – Método de comunicação <i>polled</i>	25
Figura 10 – Método de comunicação <i>strobed</i>	25
Figura 11 – Método de comunicação <i>cyclic</i>	26
Figura 12 – Método de comunicação <i>change of state</i>	26
Figura 13 – Tela de osciloscópio conectado a um barramento <i>DeviceNet</i> para manutenção	36

LISTA DE TABELAS

Tabela 1 – Velocidades de transmissão e comprimentos de cabo na <i>DeviceNet</i>	16
Tabela 2 – Resistividades lineares dos cabos usados na <i>DeviceNet</i>	18
Tabela 3 – Comprimento do segmento de rede versus corrente máxima para fonte única	19
Tabela 4 – Número e nome das classes de objetos <i>DeviceNet</i>	29
Tabela 5 – Alguns dos parâmetros dos inversores de frequência monitoráveis pela <i>DeviceNet</i>	35

SUMÁRIO

1.0 Introdução	1
2.0 Histórico	2
3.0 Considerações iniciais: redes para automação industrial	4
4.0 O Protocolo CAN	8
4.1 O Padrão CAN	10
4.2 Significados dos campos das palavras nos padrões CAN 2.0A e 2.0B	11
5.0 A <i>DeviceNet</i>	14
5.1 O meio físico	15
5.2 O projeto da rede <i>DeviceNet</i>	17
5.3 Controle de acesso ao meio	21
5.4 Arbitragem de colisões	22
5.5 Modelos de rede – Aquisição de dados	24
5.6 Configuração de uma rede <i>DeviceNet</i>	27
5.7 Modelo de objetos	29
5.8 Perfis de dispositivos	31
5.9 Classificação dos dispositivos compatíveis com a <i>DeviceNet</i>	32
5.10 Estados de erro	33
6.0 Estudo de caso (Rede instalada na CIPASA)	34
7.0 Conclusões	38
8.0 Referências bibliográficas	39

1.0 INTRODUÇÃO

A incessante busca de maiores índices de produtividade aliada à preocupação de que produtos de qualidade sejam fabricados e à necessidade de redução de custos dos processos produtivos incentivaram e permanecem incentivando o surgimento e aperfeiçoamento de tecnologias de troca de informações dentro das fábricas mundo afora.

É necessário que haja confiabilidade e segurança nas informações trocadas entre os equipamentos para que decisões importantes dentro de um determinado processo automatizado de produção configurem-se como decisões acertadas, e assim, se consiga maior produtividade, menores custos e mais qualidade.

No bojo do desenvolvimento de métodos computacionais para controle automatizado de processos surgiram as redes de comunicação. Tais redes visam, tão somente, fornecer caminhos para que dados do processo produtivo sejam disponibilizados àqueles elementos (homens ou equipamentos) que necessitem de tais informações no desempenho de sua tarefa.

A necessidade que a indústria tem de criar sistemas integrados, onde as informações fluam rapidamente do chão de fábrica até os sistemas corporativos, é crescente. Grande parte dos projetos atuais de automação industrial inclui a tecnologia de redes industriais para os equipamentos de campo e existem no mercado diversos protocolos.

Neste trabalho será abordada uma rede de comunicação e troca de dados industriais: A *DeviceNet*. Este protocolo apresenta um grande parque instalado no Brasil e configura-se como um dos mais confiáveis padrões de comunicação industrial disponíveis no mercado.

2.0 HISTÓRICO

Atualmente a automação industrial é vastamente aplicada na melhoria da produtividade e qualidade em processos industriais considerados repetitivos, estando presente no dia-a-dia das empresas para apoiar conceitos de Produção Otimizada, Qualidade Total e Sistemas Flexíveis de Manufatura.

Entretanto, entende-se que o conceito de automação remonta a muitos séculos antes de Cristo, ao desenvolver-se a roda. O objetivo era o mesmo dos objetivos atuais: simplificar o trabalho do homem, de forma a substituir o trabalho braçal e o esforço físico deste por outros mecanismos que pudessem desempenhar atividades repetitivas tão ou mais eficientemente quanto o ser humano. Os séculos passaram-se e os sistemas automatizados evoluíram.

As tecnologias evoluíram e as exigências do mercado cresceram também. O advento da microeletrônica no século XX abriu novos horizontes para a automação industrial. Surgiram os controladores programáveis, transdutores e atuadores eletrônicos.

Buscando-se, então, por processos eficientemente produtivos, de custos inteligentemente reduzidos e onde se verifiquem simplicidade nas instalações, segurança operacional e aumento nos níveis de controle e acompanhamento do processo, desenvolveram-se as redes de computadores entre equipamentos.

A necessidade imediata era disponibilizar informações sobre os processos de forma confiável e rápida. No ambiente industrial temos ambientes hostis (temperaturas e pressões extremas, fortes campos eletromagnéticos, sujeira, etc.) e, portanto, um sistema de comunicação que atravessasse tais condições deve ter a robustez necessária para levar informações corretas.

Essas redes de comunicação devem, também, apresentar características que as tornem de fácil manutenção, evolução e operação. Aspectos como bons comportamentos

temporais, confiabilidade, proteção contra perturbações ambientais, capacidade de transmissão e interoperabilidade são requisitos básicos para as redes dos atuais processos industriais.

Na década de 1980, a empresa *Robert Bosch GmbH* desenvolveu um protocolo de comunicações visando implementação automotiva em nível intraveicular. A tecnologia CAN (*Controller Area Network*) foi desenvolvida para integrar dispositivos inteligentes em automóveis (eletrônica embarcada). Itens como injeção de combustível, frenagem, *airbags*, controle e supervisão de temperatura e pressão de óleos, refrigeração do motor e da cabine, além de ajustes automáticos de posicionamento de espelhos e bancos são alguns exemplos do uso de sistemas eletrônicos inteligentes.

O protocolo CAN, originalmente desenvolvido para o ramo automotivo, nada mais é do que uma rede de comunicação serial que consegue oferecer bom desempenho em outras aplicações a tempo crítico. Os conceitos CAN foram rapidamente absorvidos pela automação industrial e serviram de base para o desenvolvimento e aperfeiçoamento de outros protocolos de comunicação.

Assim, em 1994 surge a tecnologia *DeviceNet*. Desenvolvida pela empresa norte-americana *Allen Bradley*, a *DeviceNet* baseia-se no protocolo CAN. Atualmente, existe a ODVA (*Open DeviceNet Vendors Association*), uma organização independente que gerencia e supervisiona as especificações da *DeviceNet* ao redor do mundo.

Contando com aproximadamente 350 companhias de todo o mundo em seu quadro de membros, a ODVA se propõe a promover a difusão mundial da *DeviceNet* como rede aberta de comunicação de dados. Trabalhando conjuntamente com seus membros (vendedores, distribuidores e usuários finais) a ODVA estabelece todas as especificações para a rede e equipamentos que possam ser ligados a este protocolo. Hoje, estima-se que existam mais de 8 milhões de nós *DeviceNet* instalados mundo afora.

3.0 CONSIDERAÇÕES INICIAIS: Redes para automação industrial

A opção pela implementação de sistemas de controle baseados em redes, requer um estudo para determinação de que tipo de rede melhor se adapta à situação. Esta rede deve buscar uma plataforma de aplicação compatível com o maior número possível de seus equipamentos.

Temos, então, uma situação na qual é recomendável a utilização de arquiteturas de redes a sistemas abertos que, diferentemente de arquiteturas fechadas, nas quais apenas um fabricante lança equipamentos compatíveis com a sua própria arquitetura, o usuário poderá encontrar mais de um fabricante que forneça a solução para os seus problemas. A *DeviceNet*, sendo um protocolo aberto, torna-se uma boa e viável opção para a maioria dos processos.

O ideal, tanto para fabricantes de equipamentos de automação e controle quanto para os usuários, é a modularidade e a padronização dos equipamentos e de seus softwares. Isso, no entanto, é incompatível com um livre-mercado de fornecedores. Logo, passa a haver interesse em que a rede de comunicação entre equipamentos “fale com todo mundo”. Em outras palavras, é necessário que existam normas e técnicas aceitas universalmente. Pensando nisso, em 1983 a *International Standards Organization* (ISO) propôs o modelo de referência para a interconexão aberta de equipamentos digitais, chamado de OSI (*Open Systems interconnection Reference Model*). Daí resultaram as normas produzidas pelo IEC (*International electrotechnical commission*) e IEEE (*Institute of electrical and electronics engineers*) e seus conceitos básicos são os de protocolo, entidades e camadas, assim definidos:

1) Protocolos são conjuntos de regras que regulamentam as comunicações entre as entidades de uma determinada camada.

2) Entidades são processos de hardware ou de software que agem e cooperam para produzir serviços em uma determinada camada.

3) Camadas são níveis hierárquicos do sistema dispostos de tal maneira que cada nível agrega valor ou função aos serviços prestados pelo conjunto de níveis que lhe são inferiores. Em cada camada temos a conexão de diferentes tipos de equipamentos com suas próprias características de informação e poder para controle e intervenção (ver Figura 1).

Figura 1 – Arquitetura tradicional de rede industrial (fonte: Carlos, 2006).



a) Camada de Informação:

O nível mais alto é caracterizado por grandes volumes de fluxo de dados com constantes de tempo da ordem de grandeza de segundos (aqui o tempo não é crítico). Essencialmente de domínio da informática, esta camada destina-se exclusivamente à disseminação de dados do processo aos meios corporativos de difusão de informações. O padrão Ethernet operando com o protocolo TCP/IP é o mais comumente utilizado nesta camada.

b) Camada de supervisão e controle:

Caracteriza-se por volumes moderados de dados e constantes de tempo da ordem de milisegundos. A informação deve trafegar neste nível em tempo real para garantir a atualização dos dados nos softwares que realizam a supervisão da aplicação.

Esta camada é orientada para a integração entre unidades inteligentes, de natureza diversificada. Admite aplicações de característica contínua, baixa velocidade e alta segurança. Mensagens complexas, com razoável nível de informações de diferentes propósitos. Neste nível encontramos equipamentos supervisórios que incorporam CLPs (Controladores Lógicos Programáveis) e SDCDs (Sistemas Digitais de Controle Distribuído), por exemplo.

c) Camada de dispositivos de campo:

Nessa camada temos volumes menores de dados com constantes de tempo da ordem de grandeza de milisegundos (aqui o tempo de resposta é crucial). Esta camada da rede conecta os equipamentos de baixo nível com as partes físicas e de controle. Neste nível encontram-se os sensores discretos, contactores, dispositivos de acionamento de motores e blocos de I/O. Observam-se ações executadas no nível dos dispositivos, sem necessidade de interação com níveis superiores.

As redes de equipamentos podem ser classificadas pelo tipo de equipamento conectado a elas e o tipo de dados que flui. Esses dados podem ser observados em formato de bits, bytes ou de blocos. As redes com dados em formato de bits transmitem sinais discretos contendo simples condições ON/OFF. Já as redes com dados no formato de byte podem conter pacotes de informações digitais e/ou analógicas e as redes com dados em formato de bloco são capazes de transmitir pacotes de informação de tamanhos variados. Desta forma, classificam-se as redes quanto ao tipo de rede de equipamento e os dados que ela transporta como (ver Figura 2):

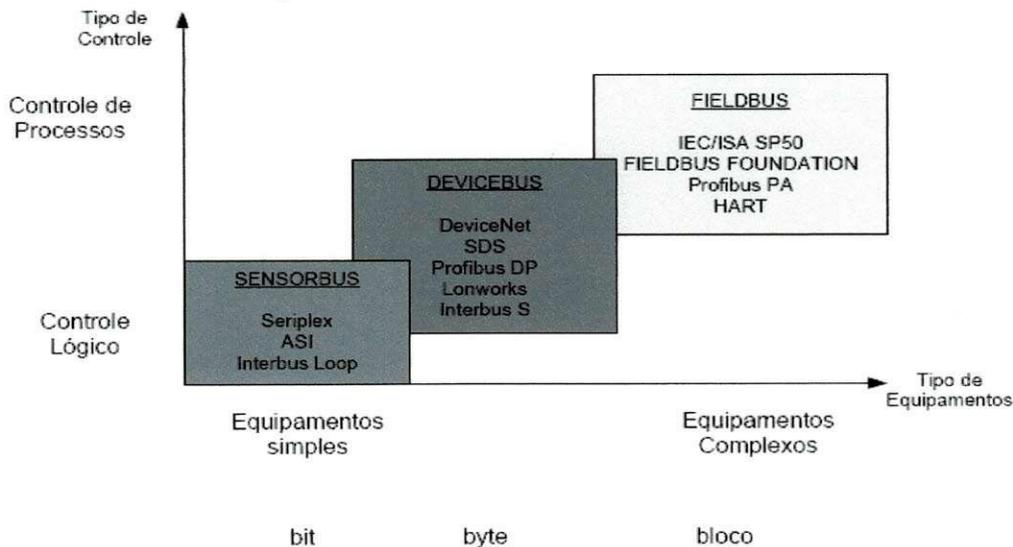
- Rede tipo *sensorbus* - dados no formato de bits;
- Rede tipo *devicebus* - dados no formato de bytes;
- Rede tipo *fieldbus* - dados no formato de pacotes de mensagens.

A rede *sensorbus* conecta equipamentos simples e pequenos diretamente à rede. Os equipamentos deste tipo de rede necessitam de comunicação rápida em níveis discretos e são tipicamente sensores e atuadores de baixo custo. Estas redes não se destinam a cobrir grandes distâncias, pois sua principal função é manter os custos de conexão tão baixos quanto for possível.

As redes tipo *devicebus* preenchem o espaço intermediário entre as redes *sensorbus* e *fieldbus* e podem cobrir distâncias de até 500 m. Os equipamentos conectados a esta rede terão mais pontos discretos, alguns dados analógicos ou uma mistura de ambos. Além disso, algumas destas redes permitem a transferência de blocos em uma menor prioridade comparada aos dados no formato de bytes. Esta rede tem os mesmos requisitos de transferência rápida de dados da rede de *sensorbus*, mas consegue gerenciar mais equipamentos e dados. A *DeviceNet* encaixa-se nesta classificação.

Por último, a rede *fieldbus* interliga os equipamentos de I/O mais inteligentes e pode cobrir distâncias maiores. Os equipamentos acoplados à rede possuem inteligência para desempenhar funções específicas de controle tais como laços PID, controle de fluxo de informações e processos. Os tempos de transferência podem ser longos, mas a rede deve ser capaz de comunicar-se por vários tipos de dados (discreto, analógico, parâmetros, programas e informações do usuário).

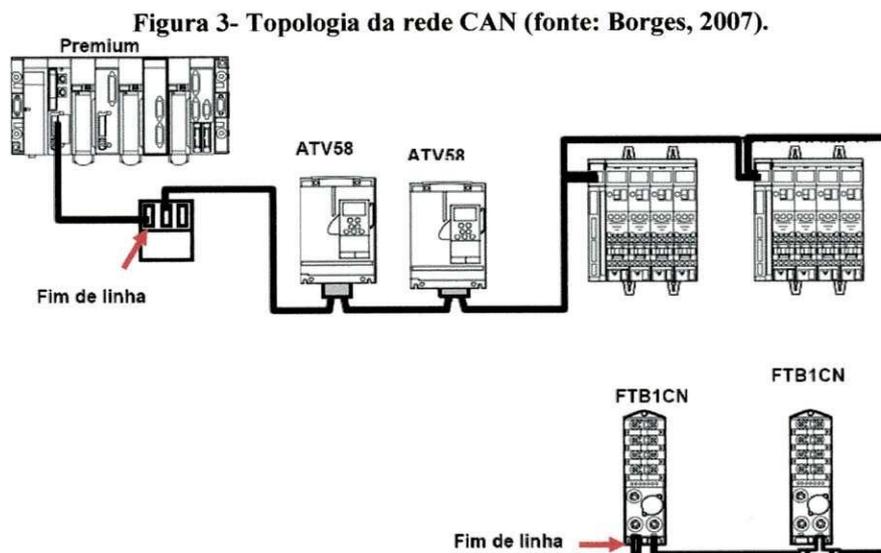
Figura 2- Classificação das Redes (fonte: Souza, 2004)



4.0 O PROTOCOLO CAN

Sendo a base de desenvolvimento do protocolo *DeviceNet*, veremos a seguir uma breve descrição do sistema CAN, seus princípios de operação e sua implementação. Este protocolo é vastamente aplicado na indústria em geral, principalmente pela sua característica de interoperabilidade entre dispositivos de fabricantes distintos, baixo custo, alta imunidade a interferências elétricas e capacidade de detecção e reparação de erros automaticamente.

CAN é um barramento de comunicação multi-mestre que apenas opera nas camadas de supervisão e de dispositivos do modelo OSI. Sua taxa de transferência máxima é da ordem de 1 Mbps. Permite uma topologia livre, em barramento ou com derivações. Identifica as mensagens e não os equipamentos. A comunicação é por eventos, o que reduz o tráfego na rede. Cada equipamento pode enviar mensagens sempre que o barramento estiver livre. A figura abaixo é um exemplo de aplicação do sistema CAN. Vemos que sua topologia é linear (barramento) que pode chegar a até 1000m. Dependendo da secção do cabo e do comprimento da rede poderemos ter de 10 kbps a 1 Mbps.



Diferentemente das redes mais tradicionais, tais como Ethernet, na CAN não há fluxo de grandes blocos de dados ponto-a-ponto ou de um nó para outro sob supervisão de

um mestre central. Numa rede CAN, muitas pequenas mensagens, como temperatura e rotação são disponibilizadas em toda a rede.

Possíveis conflitos são evitados através de uma definição de níveis de prioridade. É possível, ainda, que todos os participantes tenham os mesmos direitos, daí, é possível configurar uma rede sem mestre.

A rede CAN é um protocolo muito interessante do ponto de vista de funcionalidades e de desempenho, pois define uma metodologia de acesso ao meio físico MAC (Controle de Acesso ao Meio) e fornece como segurança os bits CRC (Verificação de redundância Cíclica), que detecta erros nos dados enviados.

4.1 O padrão CAN:

O protocolo é definido como sendo CSMA/CD+AMP (*Carrier sense multiple-access with collision detection and arbitration on message priority*, na sigla em inglês). CSMA significa que cada nó num barramento deve esperar por um período pré-definido de inatividade antes de tentar mandar uma mensagem. CD+AMP refere-se à característica da rede CAN em resolver problemas de colisão de dados através de uma decisão a bit, baseada numa prioridade pré-programada em cada mensagem no campo identificador de cada mensagem. O identificador de mais elevada prioridade sempre consegue o acesso.

A primeira versão do protocolo CAN, para aplicações de até 125kbps, apresenta um campo identificador padrão de 11 bits. A segunda versão, também com 11 bits no campo identificador, permitia taxas de transmissão entre 125 kbps e 1 Mbps. A versão mais recente do protocolo introduz 29 bits no campo identificador e trabalha a 1 Mbps. Os protocolos a 11 bits são frequentemente referidos como sendo *Standard CAN* versão 2.0A, enquanto o padrão a 29 bits é conhecido como *Extended CAN* versão 2.0B.

4.2 Significados dos campos das palavras nos padrões CAN 2.0A e 2.0B:

Utilizando o protocolo CSMA/CD+AMP qualquer nó pode acessar o barramento se o mesmo se encontrar livre. No caso de ocorrer colisão, como já foi dito, ocorrerá arbitragem bit a bit baseada na prioridade da mensagem que é função do identificador de pacote.

Assim, para as duas versões acima referidas, temos os seguintes formatos de datagramas:

Padrão CAN versão 2.0A:

S	Identificador de 11	R	I	r0	DLC	Dados (0 a 8 bytes)	CRC	ACK	E	I
O	bits	T	D						O	F
F		R	E						F	S

CAN estendida versão 2.0B:

S	Identificador	S	I	Identificador de 18	R	r1	r0	DLC	Dados (0	CRC	ACK	E	I
O	de 11 bits	R	D	bits	T				a 8 bytes)			O	F
F		R	E		R							F	S

Os significados dos campos são:

- **SOF: *Start of Frame*** - Este único bit marca o início da mensagem e é usado para sincronizar os nós num barramento ocioso.
- **Identificador de 11 bits:** Estabelece a prioridade da mensagem.
- **RTR: *Remote transmission request*** - este bit é dominante quando uma informação é solicitada de outro nó. Todos os nós recebem a requisição, mas o identificador determina o nó especificado. Os dados de resposta são então recebidos por todos os nós e usados pelos nós que necessitam.
- **IDE: *Identifier extension*** - Um bit dominante nesta posição indica que um identificador CAN padrão sem extensão está sendo transmitido. ~~Na versão 2.0B,~~

um bit recessivo nesta posição indica que existem mais bits identificadores para vir.

- r0: Bit reserva – para possível uso por uma correção padrão futura.
- DLC: *Data length code* - Esse campo de 4 bits informa o número de bytes do dado que está sendo transmitido.
- Dados: Informação – um dado de até 64 bits de tamanho pode ser transmitido.
- CRC: Teste de redundância cíclica (*Cyclic redundancy check*) - São 16 bits (15 bits mais um delimitador) que contêm o número de bits transmitidos da informação precedente para correção de erros.
- ACK: *Acknowledgement* - Reconhecimento. Todos os nós que recebem uma mensagem correta fazem este bit dominante para indicar que aquela mensagem este livre de erros. Se o nó recebe uma mensagem errada, então o bit permanece recessivo, a mensagem é descartada e o nó que envia a mensagem a reenvia após as devidas avaliações de prioridade. Dessa forma, todos os nós reconhecem a integridade de suas informações. Esse campo contém espaço para 2 bits. Um é o bit de reconhecimento e o outro funciona como delimitador.
- EOF: *End of frame* - Este campo de 7 bits indica o fim da mensagem.
- IFS: *inter-frame space* - Este campo de 7 bits informa a quantidade de tempo requerida pelo controlador para mover corretamente a mensagem recebida para sua própria posição numa área de buffer.
- SRR: *Substitute remote request* – Este bit recoloca o bit RTR na localização padrão da mensagem como um armazenador de posição no formato estendido.
- r1: Seguindo os bits RTR e r0, um bit reserva adicional foi incluído à frente do bit DLC.

Em linhas gerais, o protocolo CAN pode ser entendido como ideal para aplicações onde se verifique um grande número de mensagens curtas, num curto período de tempo, com confiabilidade e em ambientes hostis (camada de dispositivos de campo). Pelo fato de o sistema CAN ser baseado na mensagem e não no endereço, torna-se apropriado quando se necessita de dados em mais de uma localidade e a consistência dos dados é de suma importância.

A confinação das falhas é outro grande benefício do protocolo CAN. Nós defeituosos são automaticamente desligados do barramento e isto assegura a operação contínua do sistema. Essas e outras características constituem a robustez e a flexibilidade dos sistemas cujo controle se dá através de um protocolo tipo CAN.

As principais desvantagens do protocolo residem em limitações oriundas da sua dinâmica de funcionamento. Em comparação com outras redes, os sistemas CAN têm menor taxa de transferência de dados. A necessidade de sincronização dos bits do protocolo CAN também atua como fator limitante. Além disso, transmitir mensagens em forma de grandes pacotes de informação torna-se complicado, embora o protocolo suporte fragmentação de dados maiores que 8 bytes.

5.0 A DEVICENET

Como já foi dito, a *DeviceNet* nada mais é do que um sistema de barramento de campo para dispositivos industriais. Esta rede tem a capacidade de melhorar a qualidade e a taxa de transmissão de informação advindas dos dispositivos de chão-de-fábrica para os sistemas de controle, supervisão e informação com substancial redução nos custos de implantação (*wiring costs*).

A tecnologia *DeviceNet* tem característica de utilizar protocolo abertos e, por isso, permite a comunicação entre equipamentos de fabricantes distintos. Este sistema tem, por base, o bem sucedido protocolo digital CAN para comunicação entre seus componentes. Para comunicação com as camadas superiores do padrão OSI, a *DeviceNet* dispõe do, também consagrado, protocolo CIP (*Common industrial protocol*).

Entretanto, diferentemente do padrão CAN, que emprega a lógica CSMA/CD+AMP para resolução de problemas de colisão no barramento, o protocolo *DeviceNet* utiliza a tecnologia CSMA/NBA (*Carrier sense multiple-access with Non-destructive Bitwise Arbitration*, na sigla em inglês). Essa tecnologia incorpora aquela utilizada no padrão CAN e ainda permite que, no caso de colisão de dados no barramento, não haja perda da informação que foi suprimida para transmissão daquela que era mais importante naquele momento e todas as informações serão transmitidas dentro de um tempo fixo, de forma que o desempenho em tempo real da rede seja garantido.

O sistema é capaz de interconectar os mais diversos aparelhos e equipamentos usados na indústria moderna, tais como sensores, atuadores, dispositivos para acionamentos de motores e CLPs. Cada dispositivo ou controlador será um nó da rede digital. A *DeviceNet* suporta até 64 nós lógicos e a capacidade de remover e recolocar dispositivos da rede sem desativá-la constitui uma das vantagens desta tecnologia.

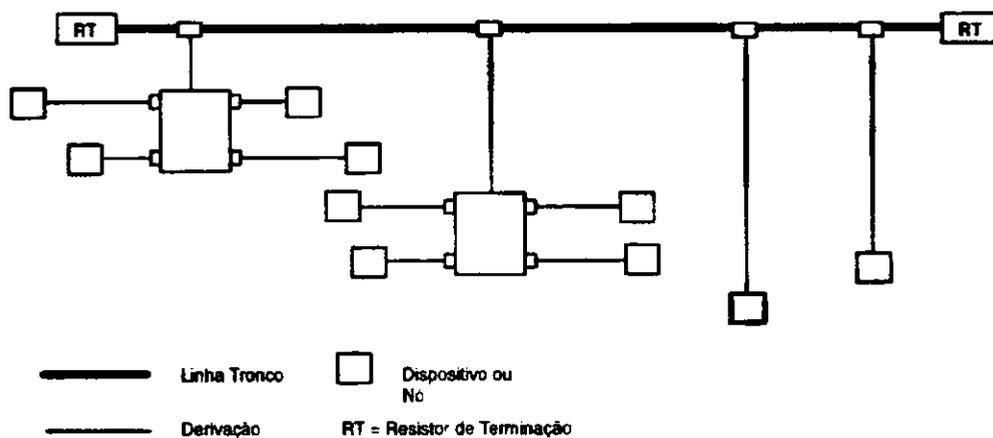
5.1 O meio físico:

A *DeviceNet* utiliza e admite apenas uma topologia para interconexão dos dispositivos. O formato usado é ilustrado na figura 4 e é definido por um barramento (cabo tronco ou cabo principal) com derivações necessariamente executadas a partir deste ramo principal, num máximo de 64 nós. A inserção ou remoção desses nós é feita a quente e não é necessário desconectar a alimentação da rede. É possível a utilização de opto acopladores para que dispositivos que sejam alimentados externamente (*soft-starters* e válvulas solenóides, por exemplo) possam compartilhar o barramento com dispositivos alimentados pelo próprio barramento. A *DeviceNet* também permite o uso de múltiplas fontes de alimentação.

A rede utiliza dois pares de fios, sendo um par destinado a comunicação e o segundo par para alimentação (24V cc) dos dispositivos conectados ao sistema. É possível que haja ainda uma blindagem externa dos pares, pelo uso de uma fita de alumínio e uma blindagem geral do cabo através de uma malha trançada com fio de dreno.

Os sinais de comunicação trabalham com a técnica de tensão diferencial para os níveis lógicos. O objetivo é diminuir a interferência eletromagnética, que atuará igualmente nos dois fios. Essa técnica, aliada a blindagem dos cabos, tende a garantir a integridade da informação.

Figura 4- Topologia da rede *DeviceNet* (fonte: Filho, 2005).



Existe uma padronização das cores dos fios: o par de alimentação é vermelho (V+) e preto (V-). O par de comunicações é branco para o sinal chamado de *CAN High* e azul para o *CAN Low*. Existem no mercado três tipos de cabos utilizáveis num sistema *DeviceNet*: O cabo tronco, também conhecido por cabo grosso, tem diâmetro externo de 12,5 mm. Também temos o cabo fino com diâmetro externo de 7 mm e o chamado cabo *flat* que possui um perfil chato e é recomendável para utilização por conectores especiais com tecnologia de perfuração visando reduzir o tempo de montagem (figura 5).

Figura 5 – Cabos (grosso, fino e flat) utilizados na rede *DeviceNet* (fonte: Padovan, 2004).



Temos, também, a presença de resistores de terminação da ordem de 121Ω e $\frac{1}{4}$ watt em cada fim de linha para que não haja reflexão de sinais. Eles são conectados entre os dois fios do par de comunicações (azul e branco).

Quanto à velocidade da comunicação, temos que, no caso da *DeviceNet*, a mesma varia de acordo com o tipo de cabo e o comprimento das derivações. A tabela abaixo mostra as restrições quanto ao comprimento dos cabos em função da taxa de transmissão adotada para a troca dos dados e do tipo de cabo escolhido:

Tipo de cabo	Função do cabo	Taxa de transmissão		
		125 kbps	250 kbps	500 kbps
Cabo grosso	Tronco	500 m	250 m	100 m
Cabo fino	Tronco	100 m		
Cabo flat	Tronco	380 m	200 m	75 m
Cabo fino	Derivações simples	6 m		
Cabo fino	Derivações cumulativas	156 m	78 m	39 m

Tabela 1: Velocidades de transmissão e comprimentos de cabo na *DeviceNet*.

5.2 O projeto da rede *DeviceNet*:

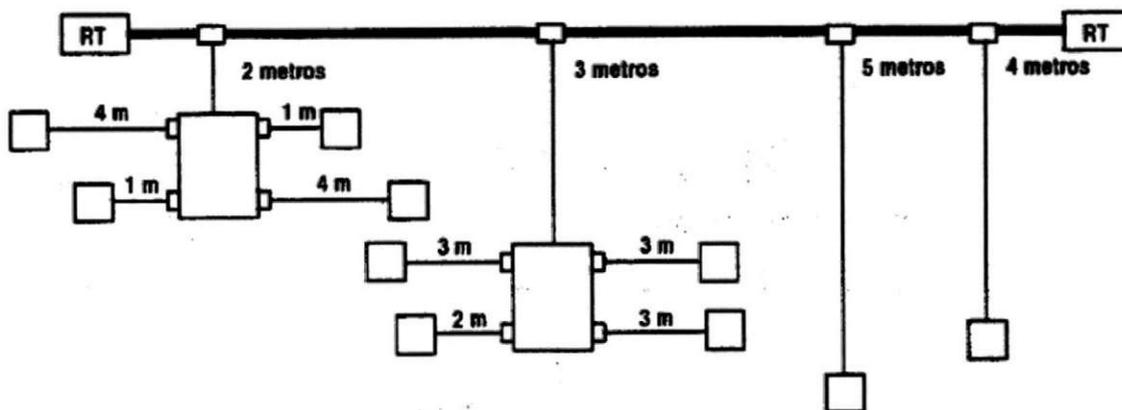
Abaixo, apresentam-se duas técnicas de dimensionamento de cabos para a *DeviceNet*. Ambas as técnicas respeitam as duas regras básicas abaixo:

- A distância máxima entre qualquer dispositivo em uma derivação ramificada para o cabo tronco não pode ser maior que 6 metros.
- A distância entre dois pontos quaisquer na rede não pode exceder a distância máxima dos cabos permitida para a taxa de comunicação e tipo de cabo utilizado conforme a tabela 1.

1º Método - Derivação cumulativa:

Usaremos a rede de dispositivos mostrada na figura 6 para exemplificar como esse método se aplica para o dimensionamento dos cabos usados para obtermos uma boa operacionalidade da rede *DeviceNet* mostrada:

Figura 6 – Rede *DeviceNet* usada para exemplificação do 1º método de dimensionamento de cabos.(fonte: Filho, 2005)



Da análise da figura acima vemos que a derivação cumulativa é de $\{(4+1+1+4)+2\} + \{(3+3+2+3)+3\} + 5 + 4 = 35\text{m}$. Observa-se, também, que nenhum nó

encontra-se a mais de 6 m do cabo tronco. Da tabela 1, concluímos que qualquer uma das taxas de transmissão mostradas pode ser usada para este sistema.

2º método - Método da queda de tensão:

O segundo método baseia-se na restrição de que a tensão de entrada em qualquer dispositivo *DeviceNet* não pode ser inferior a 21,6V. Como a tensão de alimentação da rede é 24V, se tivermos a informação da resistividade linear de cada tipo de cabo (tabela 2), poderemos calcular a queda de tensão em cada trecho considerando a corrente de trabalho de cada módulo, a distância entre os nós e utilizando as leis de Kirchhoff.

Tipo de cabo	Resistividade (Ω/m)
Cabo grosso	0,015
Cabo fino	0,069
Cabo flat	0,019

Tabela 2: Resistividades lineares dos cabos usados na *DeviceNet*.

Outro dispositivo que também segue regras para sua conexão é a fonte de alimentação do sistema. Para tal, devem-se ser observados alguns princípios básicos para determinar o posicionamento correto da fonte na rede:

- 1) Mover a fonte alimentação no sentido da seção mais carregada;
- 2) Mover as cargas de corrente mais alta para mais próximo da fonte;
- 3) Transferir dispositivos de seções sobrecarregadas para outras seções;
- 4) Diminuir o comprimento dos cabos.

Em um dado segmento, a corrente máxima suportada pela fonte, depende do comprimento desse segmento e do tipo de cabo utilizado. A tabela a seguir deve ser obedecida para o correto dimensionamento da rede:

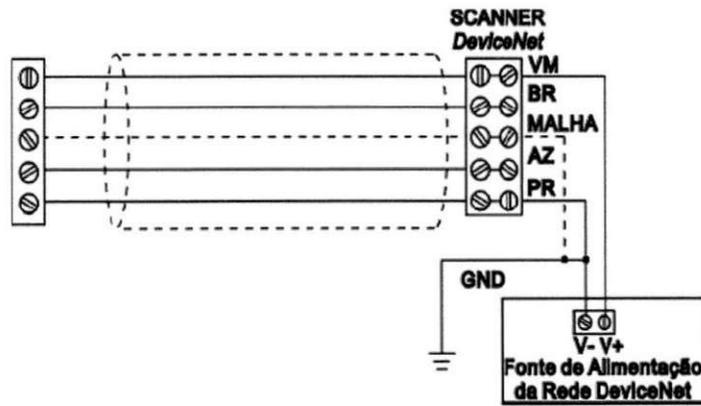
Comprimento do cabo (m) / tipo do cabo	Cabo grosso	Cabo flat
0	8,00	8,00
20	8,00	8,00
40	6,53	7,01
60	4,63	4,72
80	3,59	3,56
100	2,93	2,86
120	2,47	2,39
140	2,14	2,05
160	1,89	1,79
180	1,69	1,60
200	1,53	1,44
220	1,39	1,31
240	1,28	1,20
260	1,19	1,11
280	1,10	1,03
300	1,03	0,96
340	0,91	0,85
360	0,86	0,80
380	0,82	0,76
420	0,74	0,69
440	0,71	-
460	0,68	-
480	0,65	-
500	0,63	-

Tabela 3: Comprimento do segmento de rede versus corrente máxima para fonte única.

Assim como as mais difundidas redes industriais, a *DeviceNet* necessita de aterramento e em um único ponto. Isto independe do número de fontes de alimentação. O

cabo *DeviceNet* possui um fio de dreno para interligar a malha do cabo quando a rede passar por caixas de distribuição. Como vemos na figura seguinte, no ponto do aterramento deve-se conectar a malha e o negativo da fonte de alimentação (-V).

Figura 7 – Aterramento de uma rede *DeviceNet* (fonte: Padovan, 2004).



5.3 Controle de acesso ao meio:

Sendo uma tecnologia de barramento de dados baseado no sistema CAN, a *DeviceNet* utiliza-se do protocolo CSMA/CD+AMP (*Carrier Sense Multiple Access with Collision Detection and Arbitration on Message Priority*) e o incrementa, ao lançar mão da tecnologia NBA (*Non-destructive Bitwise Arbitration*). O mecanismo CD (em português, detecção de colisão) faz com que os nós recebam e emitam dados ao mesmo tempo, razão pela qual o CSMA/CD é também conhecido por “*Listen While Talk*” (escute enquanto conversa). Uma colisão é identificada pela comparação entre as amplitudes do sinal recebido e do sinal enviado. Assim, quando se escuta algo diferente do que foi emitido, detecta-se uma colisão no barramento.

Pelo uso deste protocolo, qualquer nó do barramento poderá acessar o mesmo se ele estiver livre. No caso de dois ou mais nós requisitarem o uso do barramento simultaneamente, ou seja, no caso de colisão, haverá arbitragem bit a bit baseada na prioridade da mensagem (AMP). O nível de prioridade é informado no campo do identificador de pacotes de 11bits de cada mensagem. O mecanismo NBA, permite ainda que não se perca informações daqueles nós que foram suprimidos durante a arbitragem da colisão. A figura abaixo ilustra o datagrama numa palavra de dados de um dispositivo *DeviceNet*:

S	Identificador	Campo	Tamanho	Dados: 0 a 8 bytes	C	A	E
O	de 11 bits (campo de	de	(LEN)	(Campo de dados)	R	C	O
F	arbitragem)	controle			C	K	F

Como já vimos:

SOF: bit de início da mensagem;

LEN: Tamanho do campo de dados;

CRC: Teste de redundância cíclica;

ACK: Reconhecimento;

EOF: Fim da mensagem.

5.4 Arbitragem de colisões:

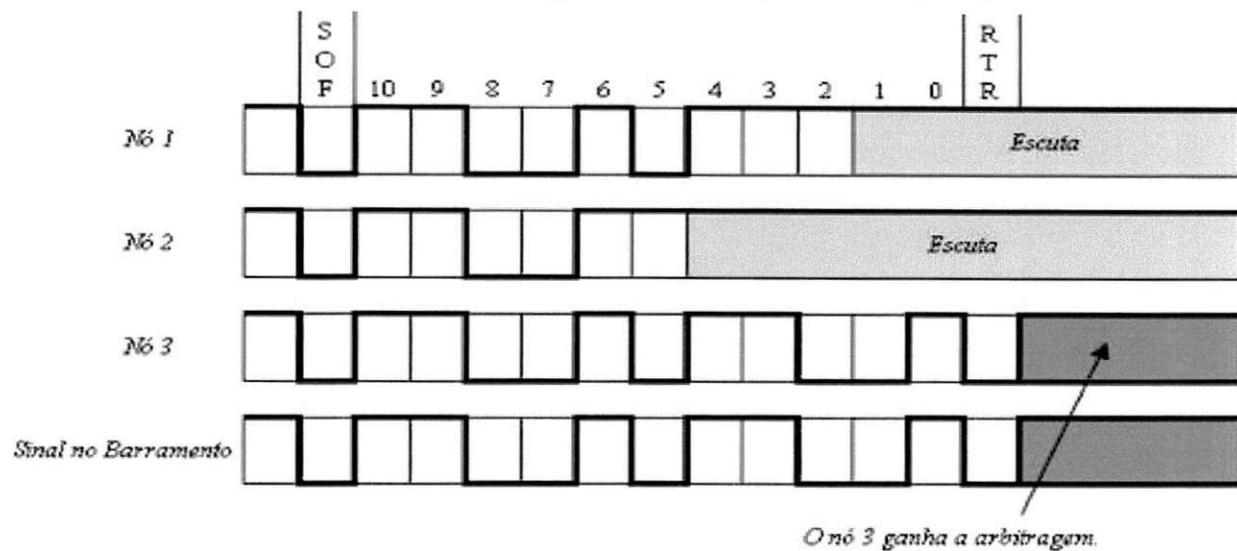
O processo de arbitragem (resolução de situação de colisão) baseia-se no conceito de dominância para garantir que a mensagem mais importante terá prioridade para ocupar o barramento e que a mensagem suprimida não será destruída durante o processo de arbitragem. Por este conceito entende-se que os sinais do barramento podem ter dois estados diferentes: bit recessivo e bit dominante. Logicamente, o bit dominante prevalece sobre o bit recessivo.

Uma vez que o protocolo utiliza um sistema de detecção de portadora, os nós sempre verificam se existe algum dado trafegando na rede antes de tentar enviar suas próprias mensagens. Estando a rede em repouso, aquele nó pode iniciar sua transmissão normalmente.

Entretanto, no caso de dois ou mais nós requisitarem simultaneamente o meio, temos uma colisão. O mecanismo de resolução funciona como se descreve a seguir: quando o barramento torna-se ocioso, cada estação com mensagem pendente iniciará sua transmissão. Cada nó inicia um processo de transmissão e observa o barramento para conferir bit a bit se o dado enviado é igual ao dado recebido. Durante a transmissão do campo identificador (campo de arbitragem), se um nó transmitir um bit recessivo e em seguida ler um bit dominante, significa que ocorreu uma colisão com uma mensagem de maior prioridade e, conseqüentemente, este nó abortará a transmissão da sua mensagem. Aguardará, então, o final daquela mensagem prioritária para tentar acessar o meio novamente. O nó de onde parte a mensagem de maior prioridade irá prosseguir o envio de dados sem perceber qualquer colisão e, assim, aquela transmissão terá sucesso.

Embora existam normas que classificam os tipos de identificadores, é possível que um programador de dispositivos crie seu próprio sistema de identificadores. Abre-se a possibilidade de se selecionar alguns bits do campo identificador para estabelecer um de dispositivo. Isto, no entanto, é desaconselhável, pois tornará esta classe de dispositivos incompatível com outros equipamentos fabricados sob as normas e os padrões da ODVA.

Figura 8 – Processo de arbitragem na DeviceNet (Fonte: Sebbe, 2005).



5.5 Modelos de rede – Aquisição de dados:

Diversos dispositivos *DeviceNet* (sensores e atuadores) são projetados para desempenhar uma determinada função, na qual dependendo do tipo e da quantidade de dados, o dispositivo atuará como produtor ou consumidor. O conceito de produtor-consumidor foi adotado pelo sistema visando eliminar trocas desnecessárias de informações. Nessa concepção, temos que um elemento “produz” a informação e a coloca no barramento. Os elementos que necessitam daqueles dados os “consomem”. Ou seja, múltiplos nós podem simultaneamente consumir os dados de um mesmo produtor.

Fundamentalmente, a *DeviceNet* emprega um modelo de comunicação entre pares (*peer-to-peer*). Entretanto, é possível também que se operacionalize um esquema simplificado de comunicação baseado numa relação mestre/escravo ou multi-mestre. Este método simplifica o movimento das mais freqüentes mensagens usadas nos sistemas de controle.

O dado é identificado pelo seu conteúdo e a mensagem não necessita de explicitar endereços de fonte e destino. Neste paradigma, não existe necessariamente a figura do mestre. Qualquer nó poderá iniciar um processo de transmissão. Necessitamos, ainda, que se configure algum dos métodos de comunicação abaixo para que a troca de informações seja correta:

Polled message: Neste método, o mestre (fixo e único), por exemplo, um cartão *scanner* da rede montado no controlador, gera mensagens de requisição de dados individuais a cada escravo (*modelo ponto a ponto*) e a resposta do escravo é direcionada ao mestre. Os nós escravos só respondem quando são perguntados. A taxa de amostragem do mestre é determinada pelo número de escravos na lista de *scan* daquele mestre (ver figura 9).

Strobed message: Neste método o mestre gera uma requisição tipo *multicast* no barramento da rede e todos os escravos com comunicação *strobed* respondem um após o

outro. Temos, portanto, uma requisição geral do mestre e respostas individuais de cada escravo *strobed* (figura 10).

Figura 9 – Método de comunicação *polled* (fonte: Padovan, 2004).

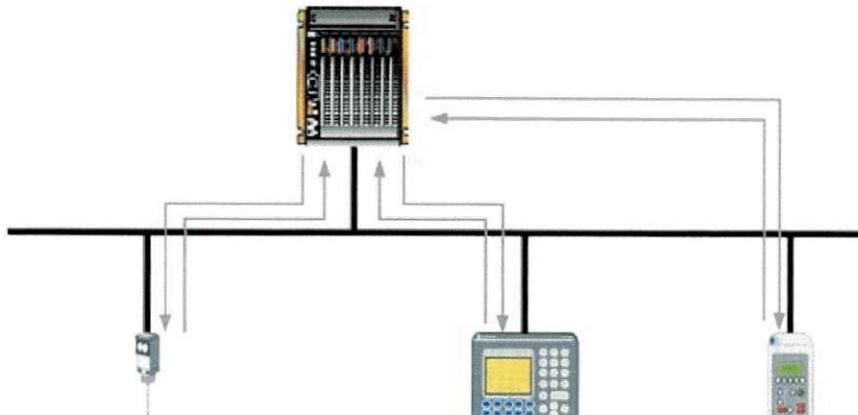
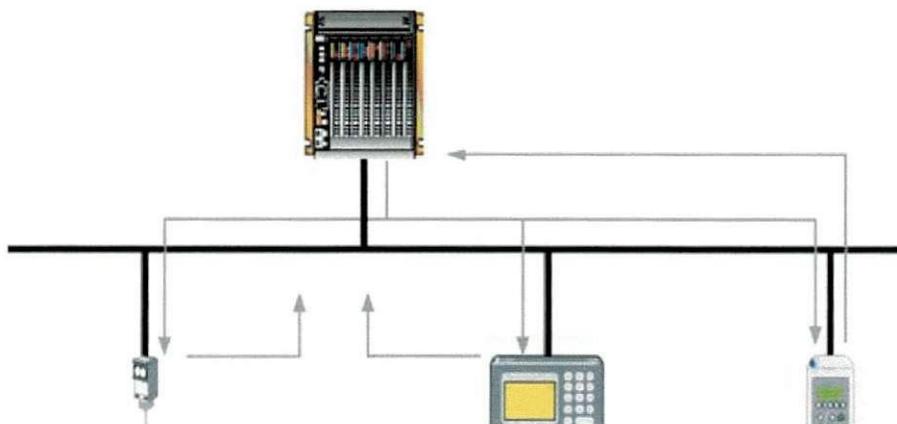
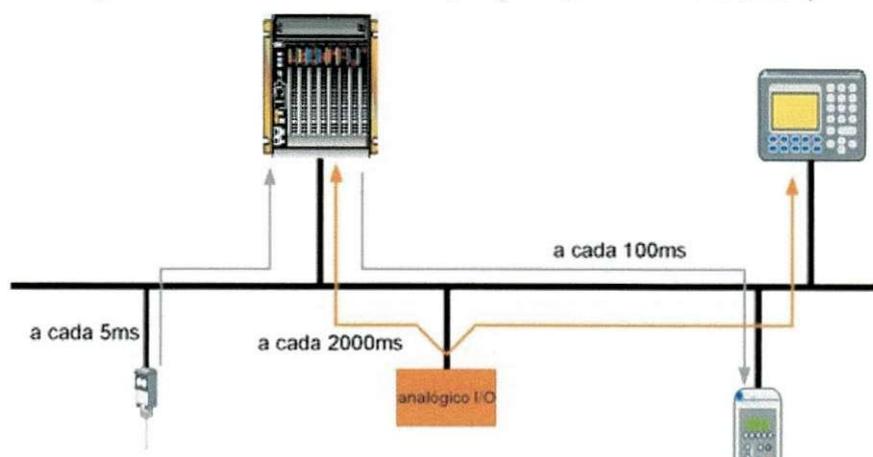


Figura 10 – Método de comunicação *strobed* (fonte: Padovan, 2004).



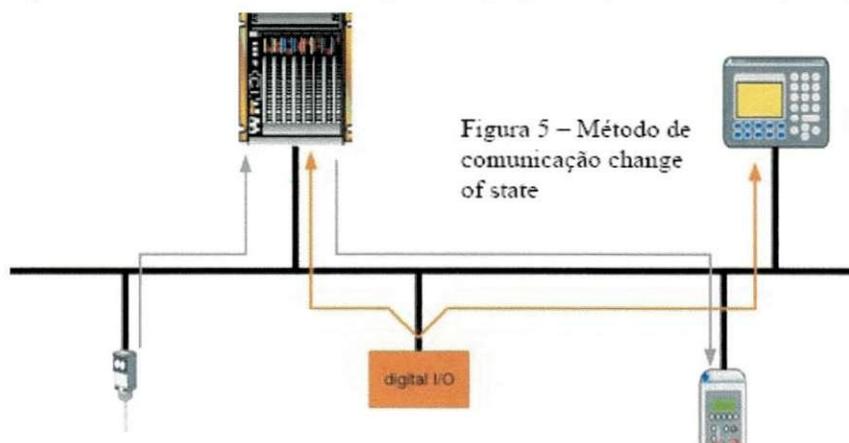
Cyclic message: Neste sistema, o escravo atualiza seus dados ao mestre da rede em intervalos de tempo pré-definidos. Este método é largamente utilizado em aplicações onde a variação de determinado ponto não necessita de atualização instantânea. A transferência cíclica é vantajosa porque ao se configurar o intervalo de tempo, poderemos aperfeiçoar a comunicação ao fazer os dados serem transmitidos numa taxa adequada aos dispositivos. Dependendo da aplicação este método pode reduzir o tráfego no barramento e usar mais eficientemente a banda de passagem (ver figura 11).

Figura 11 – Método de comunicação *cyclic* (fonte: Padovan, 2004).



Change of state: Ao invés de termos um mestre realizando a leitura cíclica de cada dado, os dispositivos de campo enviam os dados ao mestre tão logo ocorra variação de um valor em uma variável. Além disso, é possível configurar uma mensagem de *heart beat*. O dispositivo envia uma mensagem quando um dado variou ou quando o sistema ficar sem comunicar por um período de tempo determinado. Desta forma, o mestre sabe se o escravo continua ativo ou não. Este método é eficiente porque reduz significativamente o tráfego de dados na rede e faz com que os recursos sejam economizados ao evitar-se o processamento de dados antigos (ver figura 12).

Figura 12 – Método de comunicação *change of state* (fonte: Padovan, 2004).



O modo de comunicação pode ser configurado individualmente para cada escravo, e inclusive, numa mesma rede, pode haver métodos distintos de comunicação. Como já sabemos, neste caso, para evitar colisões no barramento, é utilizada a técnica CSMA/NBA.

5.6 Configuração de uma rede *DeviceNet*

Como já foi dito, a rede *DeviceNet* é uma rede industrial de protocolo aberto. Em outras palavras, existem vários fabricantes que podem fornecer equipamentos, portanto existem vários softwares configuradores de instrumentos. Para tornar viável a uniformização das configurações e a comunicação devemos observar os seguintes aspectos:

1) EDS:

Como existem vários softwares configuradores e diversos fabricantes de equipamentos, o configurador da rede deve propiciar o setup básico dos equipamentos e para tanto precisa “conhecer” o equipamento a ser configurado. Para isto existe a EDS (*Electronic Data Sheet*), um arquivo eletrônico disponível nos sites dos fabricantes. O objetivo é caracterizar o conjunto de funcionalidades presentes em determinado dispositivo e permitir uma rápida configuração dos sistemas computacionais.

As ferramentas de configuração de alto nível fazem uso destes arquivos para tornar visíveis informações de produtos de múltiplos fornecedores. Estes arquivos são disponibilizados em formato ASCII e descrevem atributos essenciais do dispositivo: nome, tipo, faixas de operação, método de comunicação, unidades de engenharia, tipos de dados, etc.

2) *Scan List*:

O software de configuração deve instruir o controlador e a placa de *scanner*, que se comunica com a rede, sobre todos os equipamentos que compõem a rede. Devem-se identificar os equipamentos por seu endereço na rede, que pode ser configurado através do software de configuração da rede.

3) Memória:

Após definido o *scan list*, deve-se definir o espaço de memória no scanner para cada elemento, e o espaço para cada escravo será definido com base no número de bytes que o

escravo necessita para as entradas e saídas, e este endereço definido é o que irá ser usado em toda a programação da estratégia de controle da planta.

5.7 Modelo de objetos

O modelo de objetos fornece uma definição e implementa os atributos (dados), serviços (métodos ou procedimentos) e comportamentos dos componentes de um dispositivo compatível com a rede *DeviceNet*.

Este padrão prevê um endereçamento para cada atributo do objeto que consiste de em quatro números:

- o endereço do nó;
- o identificador da classe do objeto;
- a instância;
- o número do atributo.

Estes quatro componentes de endereço são usados juntamente a uma mensagem explícita para que se possa mover dados de um nó para outro numa rede *DeviceNet*. Por sua vez, as classes de objeto são identificadas conforme a tabela abaixo:

Número da classe de objeto	Nome da classe de objeto
1	Identidade
2	Roteador de Mensagens
2	DeviceNet
4	Assembly
5	Conexão
6	Parametrização

Tabela 4: Número e nome das classes de objetos *DeviceNet*.

Essas classes possuem as seguintes informações:

- Objeto Identidade: ID do vendedor, data de fabricação, número serial do dispositivo.

- Roteador de mensagens: Este objeto existe para rotear mensagens e respostas explícitas através do objeto conector.
- Objeto *DeviceNet*: Contém atributos que identificam a porta, taxa de transmissão, endereço *DeviceNet*, ID do vendedor e outros parâmetros físicos de operação.
- Objeto *Assembly*: O propósito deste objeto é agrupar diferentes atributos de diferentes objetos de aplicação em um único atributo que pode ser movimentado com uma mensagem única.
- Objeto de conexão: Possui atributos de controle do processamento de mensagens I/O e mensagens explícitas. Entre outras funções, estes parâmetros controlam a frequência com que se produzem dados e os caminhos pelos quais os objetos de conexão encontram os dados.
- Objetos de parametrização: Este objeto é opcional e será usado em dispositivos com parâmetros configuráveis. Uma ferramenta de configuração necessita apenas endereçar o objeto de parametrização para acessar todos os parâmetros. Opções de configuração que são atributos do objeto de parametrização devem incluir: valores, faixas, texto e limites.

5.8 Perfis de dispositivos

O padrão *DeviceNet* reúne especificações técnicas que vão além de definições de protocolo e de conexão física dos dispositivos. Através deste instrumento, a ODVA, com o objetivo de garantir a interoperabilidade entre equipamentos de diferentes fabricantes, define modelos padrões para todos os tipos de dispositivos que podem ser ligados ao barramento *DeviceNet*.

Esses perfis demarcam as características mínimas que cada dispositivo deve possuir para serem considerados compatíveis. Desta forma, fotocélulas, inversores, atuadores de válvulas e sensores poderão ser conectados ao barramento *DeviceNet* sem problemas. Um perfil de equipamento compatível deve possuir obrigatoriamente as seguintes seções:

- Definição do modelo de objeto do dispositivo: Enumera-se todas as classes de objeto presentes no equipamento, o número de instâncias e como cada objeto interfere no comportamento das interfaces públicas.
- Definição do formato de dados de I/O do produto: Geralmente inclui a definição de um objeto da classe *Assembly* que contém o endereço (classe, instância e atributo) dos componentes de dados desejados.
- Definição dos parâmetros configuráveis do objeto e das interfaces públicas para estes parâmetros: Estas informações estão incluídas no EDS.

5.9 Classificação dos dispositivos compatíveis com a *DeviceNet*:

Os equipamentos que podem ser ligados a um barramento *DeviceNet* basicamente classificam-se em:

Dispositivos de Entrada/Saída (I/O): A maior parte dos aparelhos compatíveis com o protocolo *DeviceNet* pertence a esta categoria. Eles são responsáveis por transmitir informações para um dispositivo mestre e estão disponíveis no mercado em diversos formatos e tamanhos.

Dispositivos mestres: Normalmente são Controladores programáveis ou computadores pessoais. Estes aparelhos alocam dispositivos escravos e transferem dados entre mestres e seus escravos.

Dispositivos Gateways: Dispositivos que convertem dados de outro protocolo para o protocolo *DeviceNet*. A grande dificuldade encontrada por um Gateway é o mapeamento de dados de outro protocolo para a estrutura de objetos aceita pela *DeviceNet*.

5.10 Estados de erro

O protocolo *DeviceNet*, por sua característica de modularidade de dispositivos, pode implementar diagnósticos em algumas falhas que podem ocorrer no sistema. Eis algumas dos estados mais comuns que a rede pode assumir e seus respectivos diagnósticos:

- NON-EXISTANT: O dispositivo foi desligado devido a algum erro interno ou comando remoto;
- UNALLOCATED: O dispositivo foi ligado corretamente à rede, mas ainda não “pertence” a nenhum dispositivo mestre;
- TIMED OUT: Mensagens falharam para chegar a uma ou mais conexões com o dispositivo mestre;
- FAULTED: O dispositivo detectou um erro interno ou recebeu uma mensagem duplicada de endereço para resposta;
- BUSOFF: Neste estado o dispositivo detectou muitos erros e removeu-se da rede. Este erro é comum quando ocorrem problemas nos circuitos do dispositivo.

6.0 ESTUDO DE CASO (Rede instalada na CIPASA)

Para detalhar a aplicabilidade e as vantagens do uso do protocolo *DeviceNet* será feito um estudo de caso. O protocolo foi adotado na Votorantim Cimentos unidade CIPASA, uma empresa fabricante de cimentos. Nosso estudo focará as vantagens e as novas possibilidades que este sistema disponibilizou aos operadores daquela fábrica.

A fábrica em questão utilizava outras tecnologias para comunicação industrial (DH+, *Ethernet*, *ControlNet*), entretanto, na mais recente expansão, foi adotada a tecnologia mais moderna existente no momento. A *Allen-Bradley*, como fornecedora dos equipamentos para controle do processo produtivo da CIPASA, sugeriu a aplicação do padrão *DeviceNet*, para comunicação daquela nova área fabril com sua CPU de controle. Este padrão coexiste e opera normalmente com os outros padrões que já existiam. A interface entre os mesmos é feita através de adaptadores e das várias CPUs.

Basicamente, a nova área fabril requereu o uso do protocolo *DeviceNet* para interligação de inversores de frequência, que pode ser encarado como uma das especialidades dessa tecnologia. Com o padrão *DeviceNet* é possível, não apenas consultar e ajustar os parâmetros dos dispositivos através de um *link* do sistema de remotas como também, configurar ou reconfigurar os dispositivos através desse *link* utilizando uma interface computadorizada baseada no sistema *Windows*.

As aplicações em indústrias de cimento de um dispositivo acionador de motores incluem regulagem de velocidade do suprimento de ar do processo e de ventiladores. Essa regulagem reduz o custo com energia elétrica e fornece um produto de qualidade mais constante. Os acionadores de motores também são usados para variar a velocidade de rotação dos moinhos e mudar a taxa de alimentação de material.

Adiante temos uma tabela que informa alguns dos parâmetros de inversores de frequência que podem ser monitorados através da *DeviceNet*:

Medições	Diagnósticos
Corrente de saída	Última falha
Tensão de saída	Status do dispositivo
Potência de saída	Status das entradas
Frequência de saída	Frequência da fonte

Tabela 5: Alguns dos parâmetros dos inversores de frequência monitoráveis pela *DeviceNet*.

Apesar de na fábrica em questão, o protocolo *DeviceNet* ser usado para acionamento de apenas inversores de frequência, é possível fazer o mesmo para *Soft-starters* e relés de sobrecarga. Estes equipamentos, usualmente, fornecem partida e parada programadas de motores, baseadas em limitações de corrente e condições de carregamento. A configuração destes dispositivos é geralmente feita localmente ou através de um *link* de remotas. O protocolo pode ser usado para fazer isso da mesma forma que faz para os inversores.

Esse controle sobre a partida e sobre a parada de motores é útil na redução dos esforços mecânicos nas instalações e nos picos de energia demandados. Alguns dos aspectos dos motores podem ser acusados ou monitorados pelo uso do protocolo são: corrente média, desbalanceamento entre fases, subcarga, reversão de fases, partidas por hora, status da saída e razões de problemas, entre outros.

À medida que equipamentos da camada de dispositivos vão sendo aceitos na comunidade industrial, os benefícios da relação comunicação de dispositivos/custo de aplicação vão sendo percebidos. Existe uma gama enorme de equipamentos de custos e requisitos de comunicação variáveis. Por exemplo, uma chave de proximidade tem baixo custo, alta velocidade, e necessita de uma pequena largura de banda da rede para trabalhar. Já um transmissor de pressão necessita de uma capacidade maior de transmissão dos pacotes de informação, entretanto, seu custo é baixo e não requer muita velocidade.

A Votorantim Cimentos mantém um conjunto de padrões corporativos que devem ser seguidos à risca na realização de procedimentos dentro das fábricas do grupo. Estes padrões são elaborados seguindo normas e legislações vigentes, recomendações dos

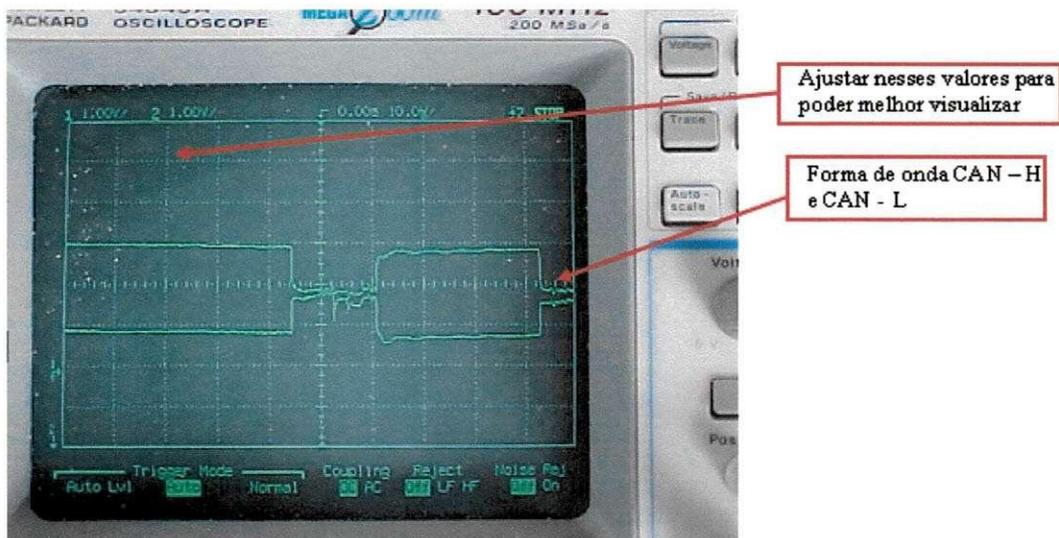
fornecedores de equipamentos e metas de econômicas e de produtividade dos equipamentos envolvidos, de tal forma que as atividades por eles descritas tenham o máximo de eficiência.

Entre esses padrões, encontram-se aqueles relacionados a atividades de manutenção e dentre os padrões gerenciais que envolvem atividades de manutenção, encontra-se o que estabelece e regulamenta os procedimentos de atividades de manutenção nas redes *DeviceNet* existentes nas fábricas.

O padrão em questão descreve rotinas de manutenção preventiva e preditiva na rede instalada e contempla verificação de:

- Integridade de rede (através de visualização de formas de onda num osciloscópio);
- Tensão de alimentação dos dispositivos (tensão mínima é 21,6 V);
- Tensão de rede (*CAN High* e *CAN Low*);
- Presença de ruído na rede;
- Aterramento da rede;
- Integridade dos resistores de terminação;

Figura 13 - Tela de osciloscópio conectada a um barramento *DeviceNet* para manutenção (fonte: Branquinho, 2006).



O objetivo dessas ações, como é de se esperar, é garantir a integridade, a confiabilidade e a operacionalidade da rede instalada e dos dispositivos a ela conectados.

O protocolo *DeviceNet* está focado em baixo custo de implantação, operação e manutenção, grande número de nós e aplicações de alta velocidade. Podemos concluir, portanto, que o sistema de comunicação industrial em questão aplica-se muito bem em fábricas de cimentos, como em muitas outras fábricas, uma vez que pode fornecer:

- Melhorias na eficiência do projeto, instalação, manutenção e operação do sistema;
- Melhor capacidade de decisão ao fazer o processo de produção de cimento ser mais transparente, em termos de acesso e confiabilidade dos dados;
- Maior flexibilidade para adicionar, retirar e reconfigurar o sistema de controle, pois o mesmo terá capacidade de se adaptar às novas necessidades.
- A manutenção preventiva em tempo real, não só da rede, mas de todos os equipamentos conectados pela rede.
- Diagnóstico de problemas mais rápido e fácil.

7.0 CONCLUSÕES

Em termos gerais, a *DeviceNet* tem um desempenho melhor, se comparada a outros protocolos, quando se trata de sistemas de controle com pequenas mensagens e/ou priorização das mesmas, características identificadas como cruciais para redes de atuação direta sobre os processos industriais, ou seja, dispositivos da camada mais baixa do sistema OSI.

Um aspecto muito importante que deve ser observado é a importância de termos um projeto bem feito e principalmente uma instalação de acordo com os requisitos que a rede industrial necessita. A instalação de redes sem antes desenvolver um pré-projeto, leva a resultados operacionais desagradáveis relativos ao desempenho da aplicação, dificultando assim o diagnóstico de problemas e a manutenção.

Toda a futura funcionalidade da rede *DeviceNet* será garantida por um projeto detalhado e que mostre todos os elementos pertencentes a rede com seus respectivos modelos, localizações e identificações, bem como as derivações e terminações existentes. Um fluxograma de redes é a principal ferramenta para uma manutenção segura e rápida evitando assim horas de produção interrompidas por falta de documentação.

Tendo seus requisitos básicos atendidos, o protocolo *DeviceNet* pode oferecer aos usuários grandes benefícios, pois reúne todas as características de uma rede industrial de baixo custo de instalação e diagnóstico remoto, além de dispor de uma programação similar a de um CLP convencional, não trazendo nenhuma dificuldade e de oferecer uma diversidade muito grande de equipamentos e de fornecedores também.

Assim como para o protocolo CAN, a desvantagem da *DeviceNet* reside na sua taxa de transferência de dados. A necessidade de utilização de um bit de sincronização também atua como fator de limitação da tecnologia. A dificuldade de transmissão de pacotes grandes de informação faz da tecnologia *DeviceNet* aconselhável apenas para comunicação no nível da camada de dispositivos, para a qual foi, de fato, projetada.

8.0 Referências bibliográficas:

- Bagschik, P., *An Introduction to CAN*, 2000;
- Barbosa, L. R. G., *Rede CAN*, 2003;
- Branquinho, G. A., Rodrigues, J. A., *Padrão Operacional Corporativo Votorantim Cimentos Rede DeviceNet – Manutenção*, 2006.
- Borges, F. *Documento técnico Schneider Electric – Redes de automação industrial*, 2007;
- Carlos, L., *Automação Industrial II*, 2006
- Corrigan, S., *Introduction to the Controller Area Network (CAN)*, 2002;
- Filho, C. S. *Arquiteturas de sistemas de automação – Uma introdução*, 2005;
- Filho, C. S. *DeviceNet*, 2005;
- Fonseca, J. A., *Redes de comunicação em ambientes industriais*, 1999;
- Guedes, L. A., *Redes para automação industrial*, 2006;
- Horner APG, *DeviceNet Network Communication Option Board for use with Reliance Electric GV3000 AC Drive*, 1998;
- Myiazaki, Y., Fukumaru, H., *Open Control Systems and DeviceNet applications*, 2000.
- Padovan, M. A. D. G., Rossit, R., *Redes Industriais – DeviceNet*, 2004;
- Parisi, A. *DeviceNet – Specification & configuration*, 2006;
- Rinaldi, J., *DeviceNet Unplugged – A “view under the hood” for end users*, 2004;
- Sebbe, R. D., *Redes industriais baseadas na tecnologia CAN: DeviceNet, SDS e CANOPEN*, 2005;
- Silva, M. E., *Curso de Automação Industrial*, 2007;
- Souza, A. J., de Oliveira, L. C., *Automação industrial*, 2003;
- Souza, F. C. *Foundation Fieldbus*, 2004;
- Sun, F., Zhao, Y., Liu, Y., *The Analysis of the Key performances of DeviceNet.*, 2006;
- Tristão, I. M., de Mattos, J. C. B., *Avaliação de barramentos industriais e sua aplicação em um estudo de caso*, 2003;
- Witte, G., *Open Networks as a Foundation for Cement Plant Automation*, 1996;