Capítulo

R2

DeviceNet

Autor: Constantino Seixas Filho
UFMG – Departamento de Engenharia Eletrônica

DeviceNet

Introdução:

A rede DeviceNet classifica-se como uma rede de dispositivo, sendo utilizada para interligação de equipamentos de campo, tais como sensores, atuadores, AC/DC drives e CLPs. Esta rede foi desenvolvida pela Allen Bradley sobre o protocolo CAN (*Controller Area Network*) e sua especificação é aberta e gerenciada pela DeviceNet Foundation. CAN, por sua vez, foi desenvolvida pela empresa Robert Bosh Corp. como uma rede digital para a indústria automobilística.

Hoje existem inúmeros fornecedores de chips CAN: Intel , Motorola, Philips/Signetics, NEC, Hitachi e Siemens.

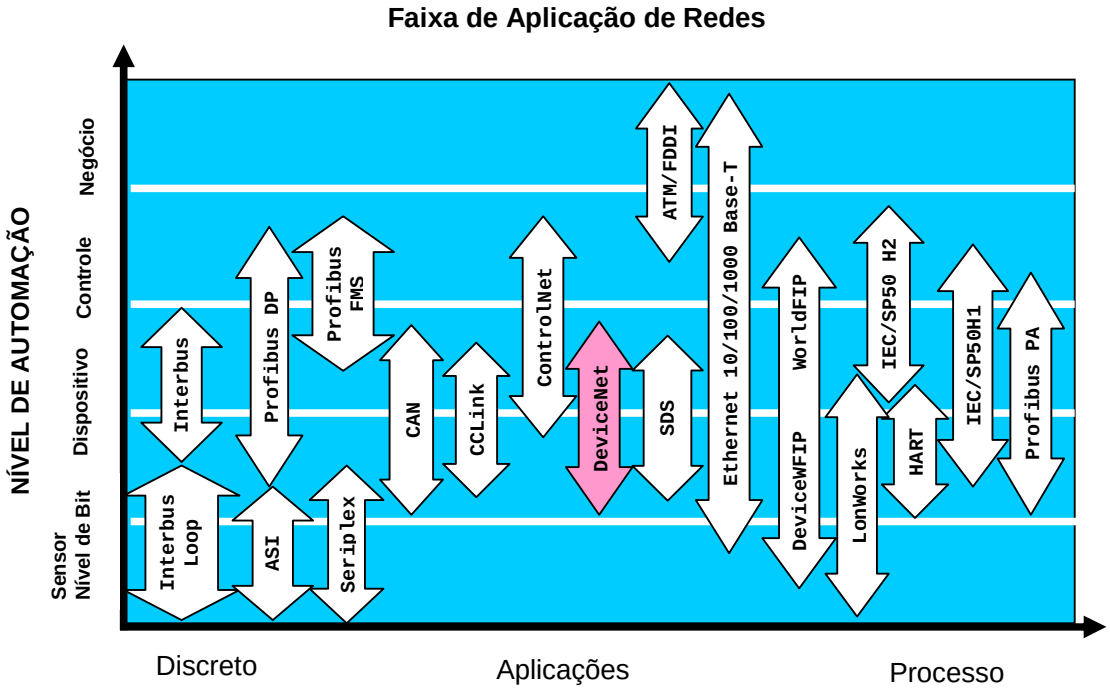


Figura 1: DeviceNet e faixa de aplicação das redes de campo

A figura 3 ilustra a relação entre CAN e DeviceNet e o stack OSI/ISO:

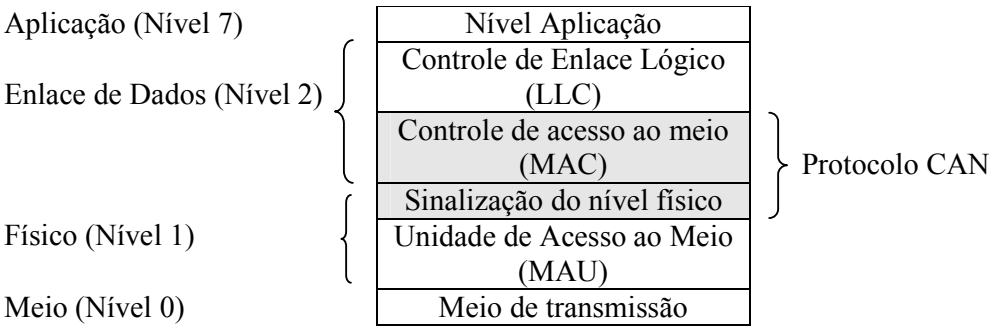


Figura 2: **DeviceNet e CAN**

Esta rede possui uma linha tronco de onde derivam *as drop lines*.

A rede DeviceNet permite a conexão de até 64 nodos. O mecanismo de comunicação é *peer to peer* com prioridade. O esquema de arbitragem é herdado do protocolo CAN e se realiza bit a bit. A transferência de dados se dá segundo o modelo produtor consumidor.

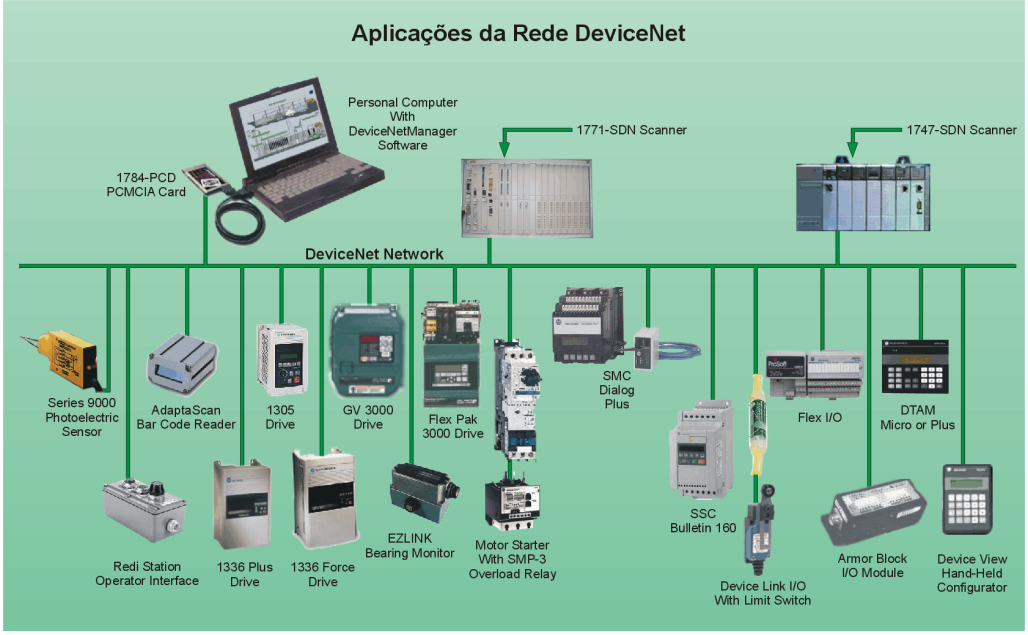


Figura 3: Aplicações da rede DeviceNet

Características do nível físico:

- Topologia física básica do tipo linha principal com derivações.
- Barramentos separados de par trançado para a distribuição de sinal e de alimentação (24VCC), ambos no mesmo cabo.
- Inserção e remoção de nodos a quente, sem necessidade de desconectar a alimentação da rede.
- Uso de opto acopladores para permitir que dispositivos alimentados externamente possam compartilhar o cabo do barramento com os dispositivos alimentados pelo barramento.
- Usa terminadores de 121 ohms em cada fim de linha.
- Permite conexão de múltiplas fontes de alimentação.
- As conexões podem ser abertas ou seladas.

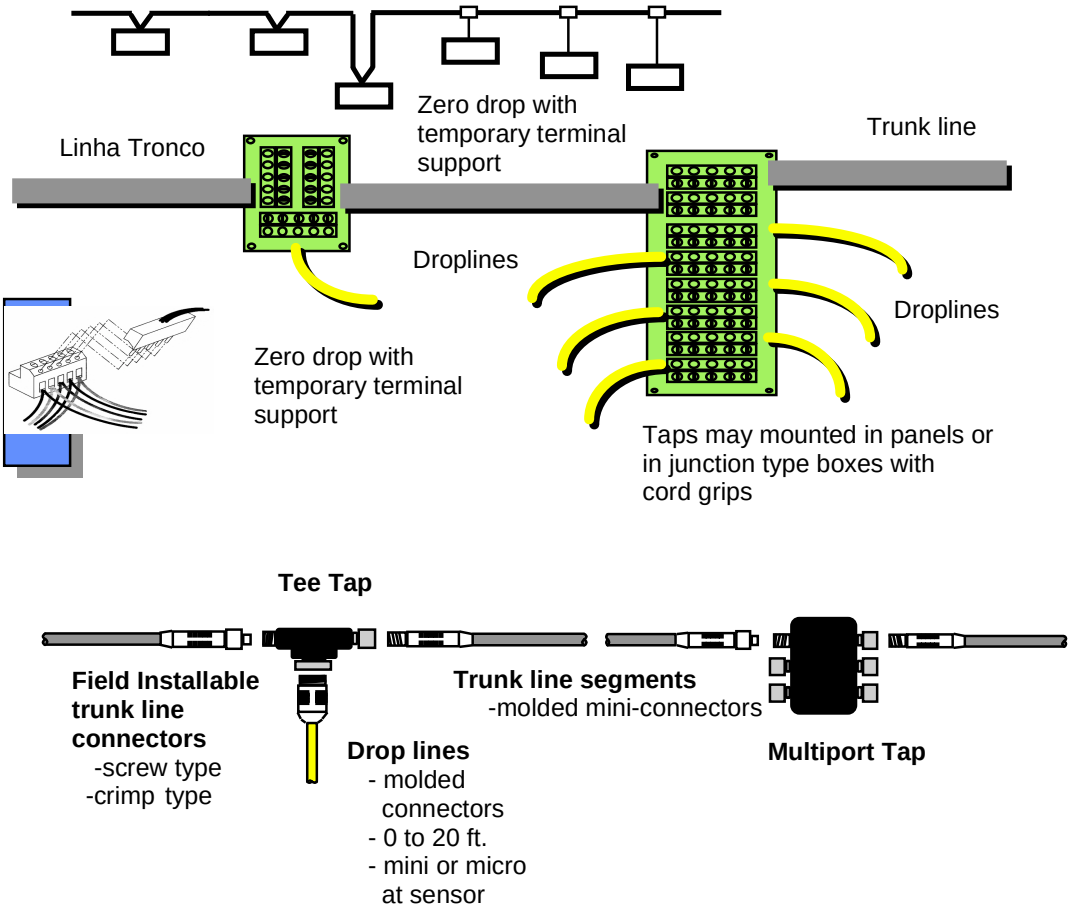


Figura 4: Caixa de conexão aberta e conexão selada

A partir de cada *dropline* vários dispositivos podem ser ligados em *daisy chain*.

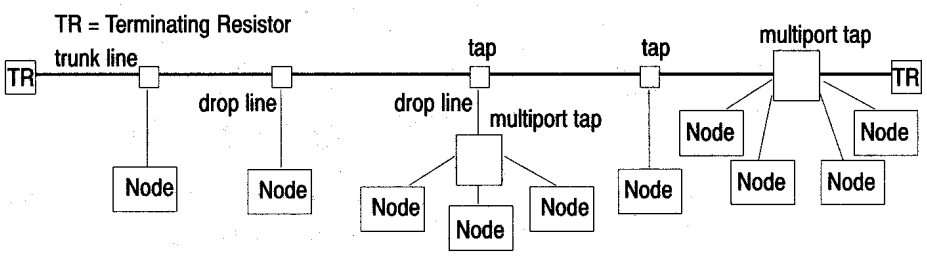


Figura 5: Topologia da rede DeviceNet

As seguintes regras devem ser obedecidas para que o sistema de cabos seja operacional:

- A distância máxima entre qualquer dispositivo em uma derivação ramificada para a linha tronco não pode ser maior que 6 metros (20 pés).
- 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. A distância se refere a distância entre dois dispositivos ou resistores de terminação.

Velocidade de transmissão	Distância Máxima (Cabo Grosso)	Distância Máxima (Cabo fino)	Comprimento da derivação	
			Máxima	Acumulada
125 Kbps	500 m	100m	6 m	156 m
250 Kbps	250 m	100m	6 m	78 m
500 Kbps	100 m	100m	6 m	39 m

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

Exemplo: Cálculo da derivação cumulativa

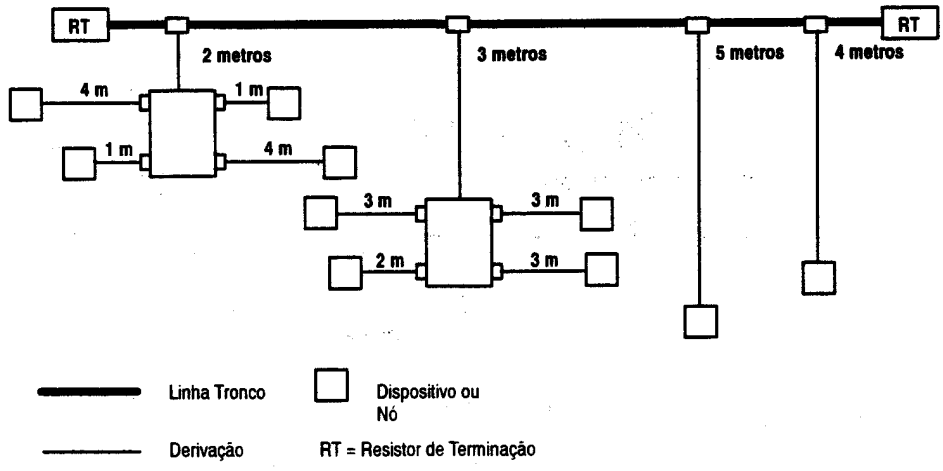


Figura 6: Cálculo da derivação cumulativa

Examine o desenho acima e complete:
O comprimento da derivação cumulativa é:
(4+1+1+4) + 2 + (3+2+3+3) + 3 + 5 + 4 = 35 m.
Existe algum nó a mais de 6 metros da linha tronco ? _____
Que taxas de comunicação podem ser usadas nesta rede ? _____

Exemplo: Cálculo da distância máxima dos cabos

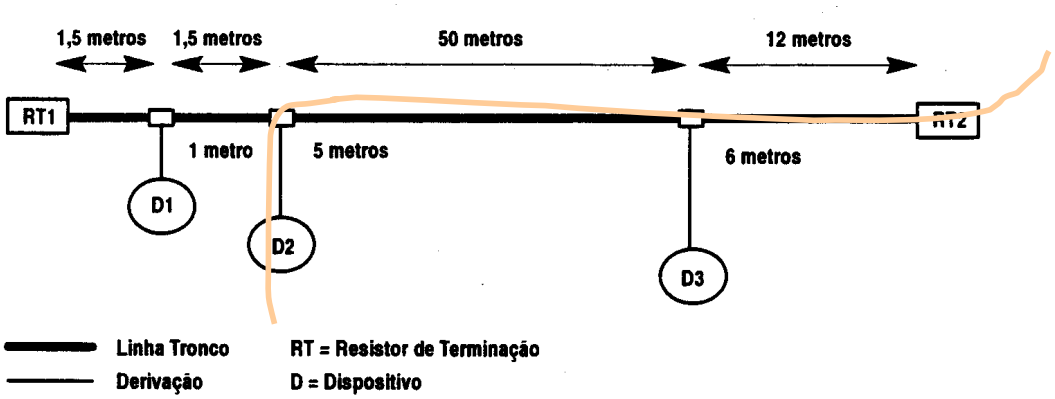


Figura 7: Cálculo da distância Máxima dos cabos

Derivação 1: Não é considerada porque seu comprimento é menor que a distância da linha tronco para o resistor de terminação (1.5 m).

Derivação 2: É considerada, já que 5 > 1,5 + 1,5.

Derivação 3: Não é considerada.

Distância máxima dos cabos = (5 m + 50 m + 12 m) = 67 metros.

Uma outra maneira de se realizar este cálculo seria avaliar a perda de tensão na rede para que a tensão na entrada de qualquer módulo não seja inferior a 21,6 V. Para isso calculamos as quedas de tensão em cada cabo considerando a resistência linear típica dos cabos, as distâncias entre os nodos e as correntes de consumo de cada equipamento. Usando Kirchoff determina-se as correntes em cada trecho e por conseqüência as quedas de tensão.

Tipo do cabo	Resistividade do cabo (Ω/m)
Cabo Grosso	0,015
Cabo fino	0,069
Cabo chato	0,019

Tabela 2 – Resistividade de cabos DeviceNet

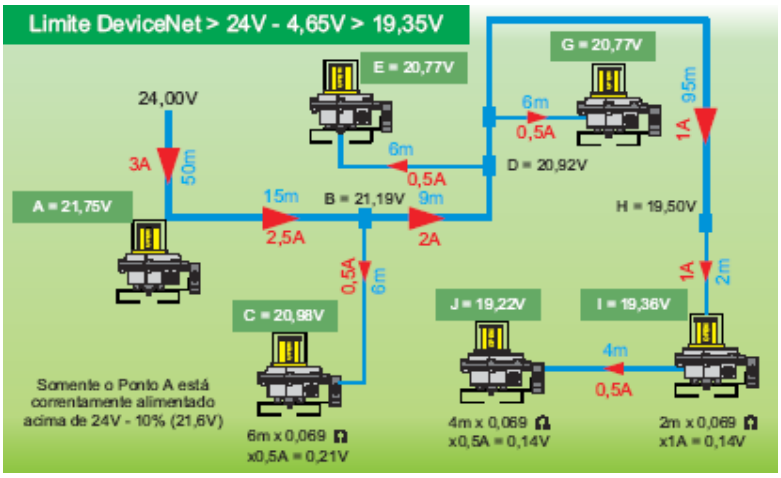


Figura 8 – Cálculo de queda de tensão numa rede DeviceNet [Sense 2001]

Colocação da fonte de alimentação

A colocação da fonte de alimentação também segue uma série de regras. Aqui examinaremos apenas alguns princípios básicos:

A corrente máxima suportada pela fonte, em um dado segmento, é função do comprimento máximo do segmento e deve obedecer à tabela abaixo:

	Cabo Grosso	Cabo Chato		Cabo Grosso	Cabo Chato
Comprimento da rede (m)	**Corrente Máxima (A)**	**Corrente Máxima (A)**	**Comprimento da rede (m)**	**Corrente Máxima (A)**	**Corrente Máxima (A)**
0	8.00	8.00	240	1.28	1.20
20	8.00	8.00	260	1.19	1.11
40	6.53	7.01	280	1.10	1.03
60	4.63	4.72	300	1.03	0.96
80	3.59	3.56	340	0.91	0.85
100	2.93	2.86	360	0.86	0.80
120	2.47	2.39	380	0.82	0.76
140	2.14	2.05	420	0.74	0.69
160	1.89	1.79	440	0.71	----
180	1.69	1.60	460	0.68	----
200	1.53	1.44	480	0.65	----
220	1.39	1.31	500	0.63	----

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

Princípios gerais a serem observados para melhorar o posicionamento da fonte:

- Mover a fonte de alimentação na direção da seção sobrecarregada
- Mover as cargas de corrente mais alta para mais próximo da fonte.
- Transferir os dispositivos de seções sobrecarregadas para outras seções.
- Diminuir o comprimento dos cabos.

Exemplo

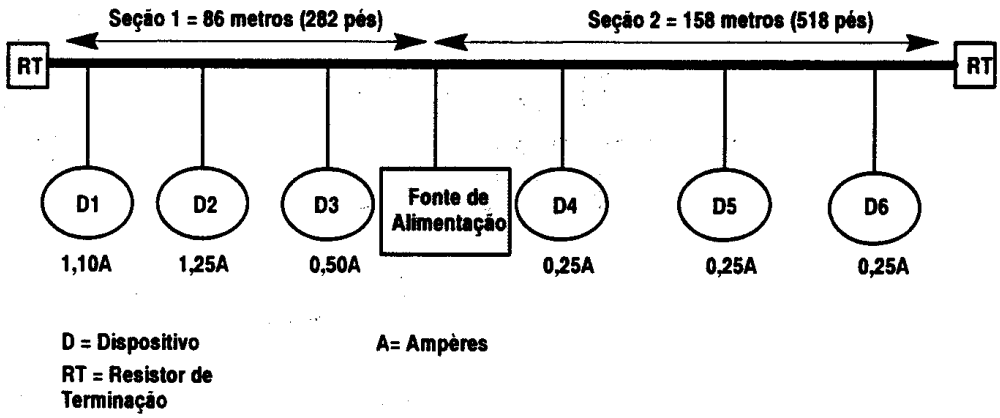


Figura 9: Posicionamento da fonte de alimentação

Vamos determinar se a fonte de alimentação está sobrecarregada ou não:

1) Somatório das correntes dos dispositivos da Seção 1:
   (1,10 + 1,25 + 0,50) = 2,85

2) Somatório das correntes dos dispositivos da Seção 2:
   (0,25 + 0,25 + 0,25) = 0,75

3) O comprimento da seção 1 é de 86 metros. Consultando a tabela para 100 metros verificamos que a corrente máxima permitida é de 2,93 A.
   O comprimento da seção 2 é de 158 metros. Consultando a tabela para 160 metros encontramos 1,89 A.

   Logo, toda a rede está operacional.

DeviceNetAssistant

A Rockwell Automation desenvolveu um aplicativo que facilita a configuração de um barramento DeviceNet. O software realiza os cálculos necessários para verificação de comprimentos de cabo, corrente, etc.

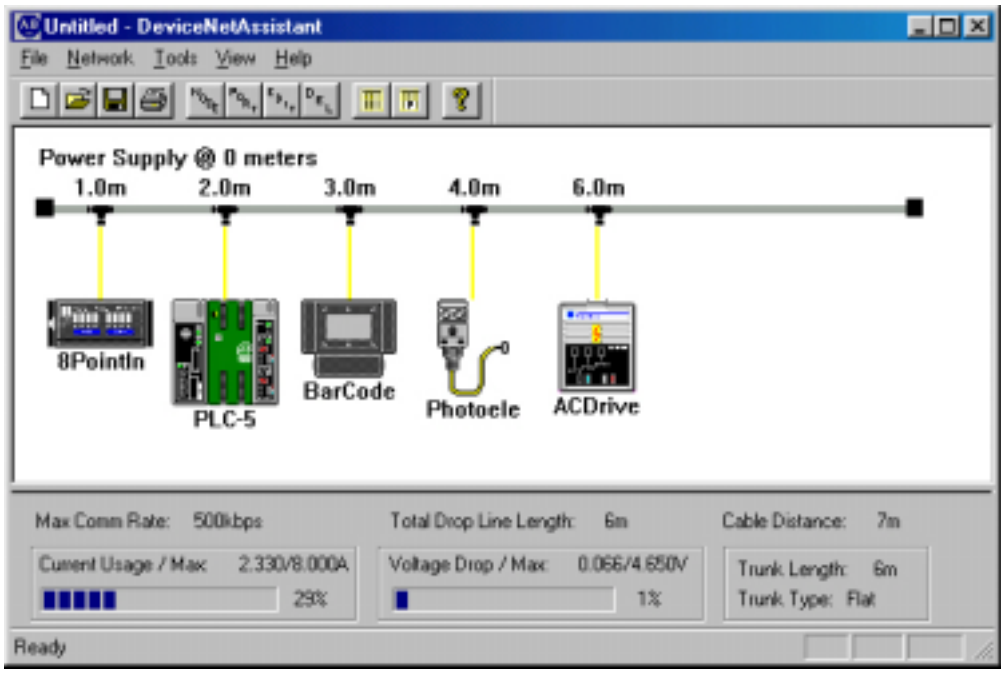


Figura 10: Tela do DeviceNetAssistant

Controle de acesso ao meio – camada DLL

Utiliza protocolo CSMA/NBA – **C**arrier **S**ense **M**ultiple **A**ccess with **N**on **D**estructive **B**itwise **A**rbitration ou CSMA/CD + AMP (*Arbitration on Message Priority*)

Através deste protocolo qualquer nó pode acessar o barramento quando este se encontra livre. Caso haja contenção, ocorrerá arbitragem bit a bit baseada na prioridade da mensagem que é função do identificador de pacote de 11 bits.

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

SOF – Start of Frame
LEN – Tamanho do campo de dados
CRC - Cyclic Redundancy Code
ACK - Acknowledgement
EOF – End of frame

Figura 11: Quadro de dados CAN

Arbitragem

Um nodo só inicia o processo de transmissão, quando o meio está livre.
Cada nó inicia um processo de transmissão e escuta o meio para conferir bit a bit se o dado enviado é igual ao dado recebido. Os bits com um valor dominante sobrescrevem os bits com um valor recessivo.

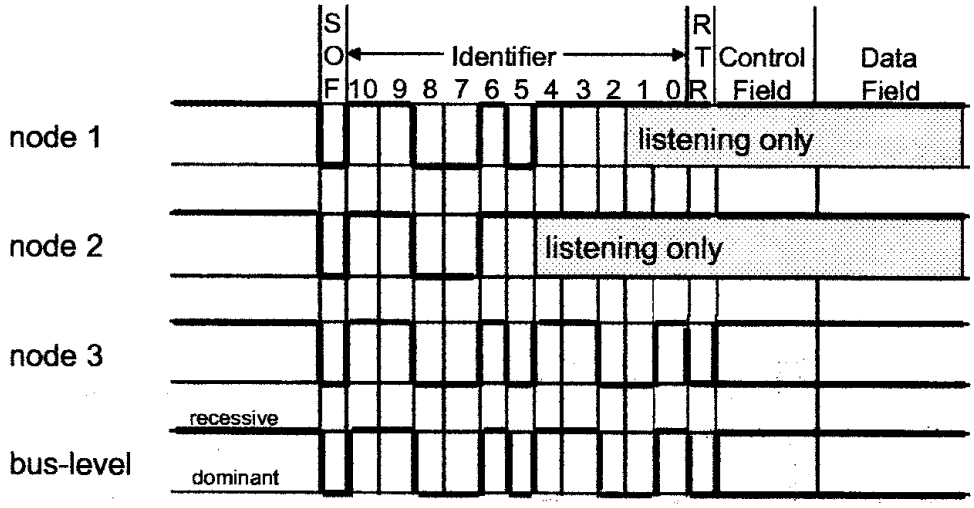


Figura 12: Processo de arbitragem

Suponha que os nodos 1, 2 e 3 iniciem a transmissão simultaneamente. Todos os nodos escrevem e lêem o mesmo bit do barramento até que o nodo 2 tenta escrever um bit recessivo (1) e lê no barramento um bit dominante (0). Neste momento o nodo 2 passa para o modo de leitura. Um pouco mais à frente o mesmo acontece com o nodo 1. Isto significa que o valor do identificador da mensagem 3 tem um menor valor binário e portanto uma maior prioridade que as demais mensagens.

Todos os nodos respondem com a ACK, dentro do mesmo slot de tempo, se eles receberam a mensagem corretamente.

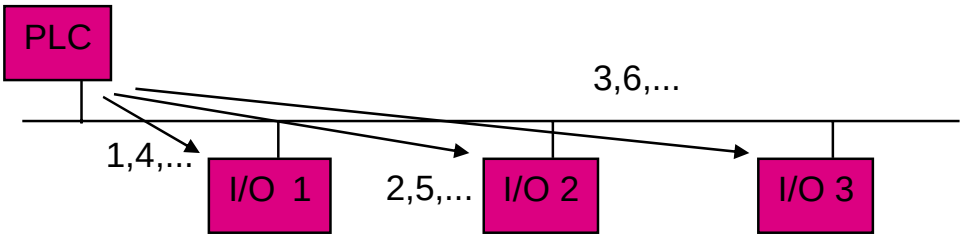
Modelo de rede

Utiliza paradigma Produtor/Consumidor que suporta vários modelos de rede:

Produtor/Consumidor

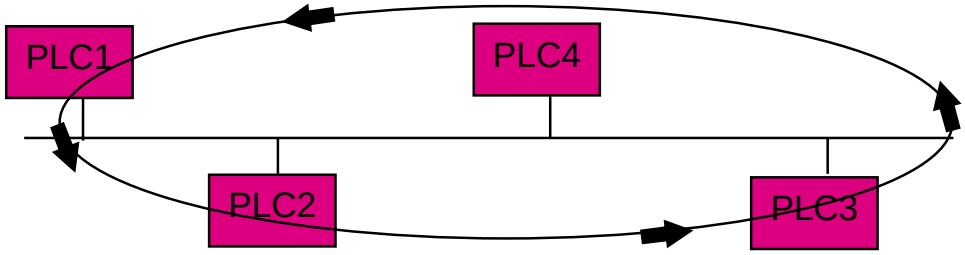
O Dado é identificado pelo seu conteúdo. A mensagem não necessita explicitar endereço da fonte e destino dos dados. Também não existe o conceito de mestre. Qualquer nodo pode iniciar um processo de transmissão. Este modelo permite gerar todos os demais:

Mestre/Escravo



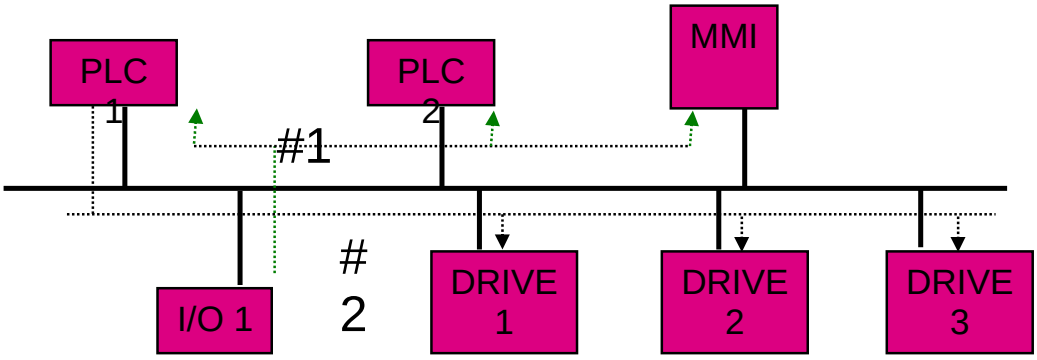
O PLC ou scanner possui a função de mestre e realiza um polling dos dispositivos escravos. Os escravos só respondem quando são perguntados. Neste sistema o mestre é fixo e existe apenas um mestre por rede.

Peer to peer



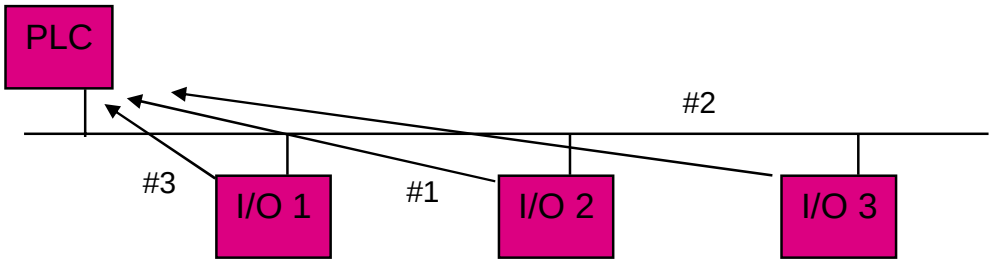
Redes peer to peer não possuem um mestre fixo. Cada nó tem o direito de gerar mensagens para a rede, quando de posse de um token. O mecanismo de passagem de token pode ser baseado na posição do nodo no anel lógico ou definido por um mecanismo de prioridades.

Multi-mestre



Uma mensagem pode alcançar diversos destinatários simultaneamente. Na figura acima a a referência de posição do rack remoto #1 é enviada para o PLC1, PLC2 e para a MMI, ao mesmo tempo. Numa segunda transação, o comando referência de velocidade é enviado aos três drives ao mesmo tempo.

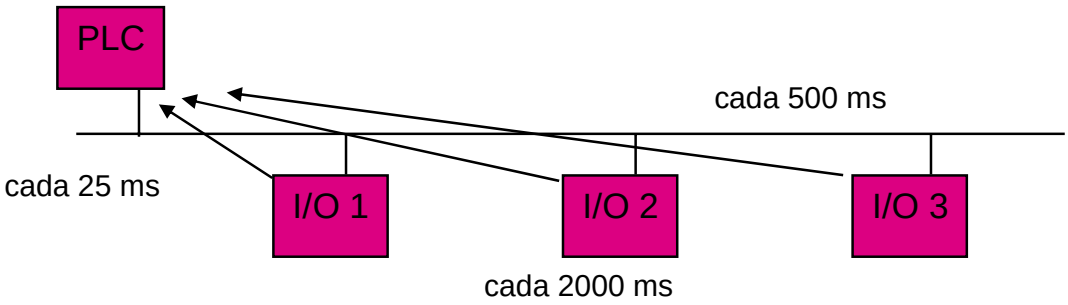
Mudança do estado do dado



12

Ao invés de termos um mestre realizando a leitura cíclica de cada dado, os dispositivos de campo enviam os dados ao mestre quando houver variação de um valor em uma variável. Também 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 sabemos se o dispositivo está vivo ou não.

Produção cíclica de dados



Os dispositivos de campo atualizam o mestre periodicamente em bases de tempo pré estabelecidas. O modo de operação: mudança de estado e produção cíclica são configuráveis nodo a nodo.

12

Nestes dois últimos tipos de mensagens o consumidor deve enviar uma ACK ao produtor. Para gerenciar o envio de mensagens de múltiplos consumidores, o ACK *handler object* deve ser utilizado.

Mensagens

O identificador CAN é utilizado para estabelecer a prioridade do nó no processo de arbitragem e é usado pelos nodos que recebem a mensagem para filtrar as mensagens do seu interesse.

A rede DeviceNet define dois tipos de mensagens: mensagens de entrada e saída e mensagens explícitas.

Mensagens de entrada/saída

São dados de tempo crítico orientados ao controle. Elas permitem o trânsito de dados entre uma aplicação produtora e uma ou mais aplicações consumidoras.

As mensagens possuem campo de dados de tamanho de 0 a 8 bytes que não contém nenhum protocolo, exceto para as mensagens de I/O fragmentado, onde o primeiro byte da mensagem é usado para o protocolo de fragmentação. O significado de cada mensagem é função do identificador CAN. Antes que mensagens utilizando este ID possam ser enviadas, tanto o dispositivo emissor quanto o receptor devem ser configurados.

Identificador	Dado	CRC
	<- 0..8 bytes ->	

Quando a mensagem supera os 8 bytes, existe um serviço de fragmentação de mensagens que é aplicado. Não existe limite no número de fragmentos.

Mensagens explícitas

São utilizadas para transportar dados de configuração e diagnóstico ponto a ponto. Estas mensagens possuem baixa prioridade. Elas constituem uma comunicação do tipo pergunta/resposta geralmente utilizadas para realizar a configuração de nodos e o diagnóstico de problemas. O significado de cada mensagem é codificado no campo de dados.

Mensagens explícitas também podem ser fragmentadas.

	7	6	5	4	3	2	1	0
0	Cabeçalho da mensagem							
1	Corpo da Mensagem							
2								
3								
4								
5								
6								

	7	6	5	4	3	2	1	0
0	Cabeçalho da Mensagem							
1	Protocolo de fragmentação							
2	Corpo de Mensagem Fragmentada							
3								
4								
5								
6								

7			7	

Figura 13: Quadro de dados: mensagem explícita não fragmentada x fragmentada

A definição do comportamento de um dispositivo inserido na rede é definida pelo Device Profile.

DeviceNet divide os 11 bits do identificador CAN em quatro grupos:
Os três primeiros grupos contém dois campos, um campo de 6 bits para o MAC ID (6 bits <-> 64 endereços) e o restante para o MESSAGE ID. Os dois campos combinados formam o CONECTION ID.

Group ID

Bits de identificação											HEX RANGE	USO DO ENDEREÇO
10	9	8	7	6	5	4	3	2	1	0		
0	Group 1 Msg ID				Source MAC ID						000-3ff	Grupo de mensagens 1
1	0	MAC ID						Group 2 Message ID			400-5ff	Grupo de mensagens 2
1	1	Group 3 Message ID			Source MAC ID						600-7bf	Grupo de Mensagens 3
1	1	1	1	1	Group 4 Message ID (0-2f)						7c0-7ef	Grupo de Mensagens 4
1	1	1	1	1	1	1	X	X	X	X	7f0-7ff	Idenficadores Inválidos
10	9	8	7	6	5	4	3	2	1	0		

Figura 14: Grupos de mensagens DeviceNet

Predefined Master/Slave Connection Set

Em aplicações Master slave com dispositivos simples, não existe necessidade de configuração dinâmica de conexões entre os dispositivos. Neste caso pode-se usar um conjunto especial de identificadores conhecidos como *Predefined Master/Slave Connection Set*. O tipo e a quantidade de dados a serem gerados por estes dispositivos simples é conhecido em tempo de configuração.

As mensagens do grupo 2 são utilizadas na definição destes identificadores. Neste grupo, o MAC ID não é especificado como Source MAC ID, o que possibilita utilizá-lo como Destination ID. O group ID e o MAC ID estão localizados nos primeiros 8 bits da mensagem o que permite sua filtragem por chips antigos do protocolo CAN, que só trabalham com 8 bits.

Um mestre, desejando se comunicar com diversos escravos, pode pedir emprestado o endereço do destino da mensagem e usar o campo de MAC ID para este fim.

BITS DE IDENTIFICAÇÃO											**Descrição**
10	9	8	7	6	5	4	3	2	1	0	
0	Group 1 Msg ID				Source MAC ID						**Group 1 Messages**

0	1	1	0	1	Source MAC ID					Slaves I/O Change of State or Cyclic Message
0	1	1	1	0	Source MAC ID					Slave's I/O Bit-Strobe Response Message
0	1	1	1	1	Source MAC ID					Slave's I/O Poll Response Message
1	0	**MAC ID**				Group 2 Message ID				**Group 2 Messages**
1	0	MAC ID				0	0	0		Master's I/O Bit-Strobe Command Msg
1	0	MAC ID				0	0	1		Reserved for Master's Use – Use in TBD
1	0	MAC ID				0	1	0		Master's change of State Ack Message
1	0	MAC ID				0	1	1		Slave's Explicit Response Messages
1	0	MAC ID				1	0	0		Masgter's Connected Explicit Request Msg
1	0	MAC ID				1	0	1		Master's I/O Poll Cmd/Change of Satte/Cyclic Msg.
1	0	MAC ID				1	1	0		Group 2 Only Unconnected Explicit Req. Msgs.
1	0	MAC ID				1	1	1		Duplicate MAC ID Check Messages

Figura 15: Mensagens do Predefined Master/Slave Connection set

O Modelo de Objetos

O modelo de objetos fornece um gabarito e implementa os **atributos** (dados), **serviços** (métodos ou procedimentos) e **comportamentos** dos componentes de um produto DeviceNet.

O modelo prevê um endereçamento de cada atributo consistindo de quatro números: o **endereço do nodo** (MAC ID), o **identificador da classe de objeto**, a **instância**, e o **número do atributo**. Estes quatro componentes de endereço são usados com uma mensagem explícita para mover dados de um lugar para outro numa rede DeviceNet. A tabela a seguir indica o ranges que estes endereços podem ocupar:

Endereço	Menor	Maior
Nó	0	63
Classe	1	65535
Instância	0	65535
Atributo	1	255

As classes de objeto são identificadas por um número conforme tabela abaixo:

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

Os principais objetos definidos são:

Objeto Identidade

Cada produto DeviceNet terá uma única instância do objeto identidade. Os atributos serão:

Atributos	Serviços
• VendorID • Device Type • Product Code • Revision • Status • Serial Number • ProductName • Status	• Get_attribute_Single • Reset

Objeto Roteador de Mensagens

Cada produto DeviceNet terá uma única instância do objeto roteador de mensagem (Instância #1). O componente roteador de mensagens é o componente de um produto que passa mensagens explícitas para outros objetos. Ele em geral não possui nenhuma visibilidade externa na rede DeviceNet.

Objeto Assembly

Cada produto DeviceNet terá uma única ou múltiplas instâncias do objeto assembly. O propósito deste objeto é agrupar diferente atributos (dados) de diferentes objetos de aplicação em um único atributo que pode ser movimentado com uma mensagem única.

Objetos de Conexão

Cada produto DeviceNet terá tipicamente pelo menos dois objetos de conexão. Cada objeto de conexão representa um ponto terminal de uma conexão virtual entre dois nodos numa rede DeviceNet. Uma conexão se chama *Explicit Messaging* e a outra *I/O Messaging*. Mensagens explícitas contém um endereço do atributo, valores de atributo e código de serviço descrevendo a ação pretendida. Mensagens de I/O contém apenas dados. Numa mensagem de I/O toda a informação sobre o que fazer com o dado está contida no objeto de conexão associado como a mensagem de I/O.

Objetos de Parametrização

Este objeto é opcional e será usado em dispositivos com parâmetros configuráveis. Deve existir uma instância para cada parâmetro configurável. 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.

Todo dispositivo usualmente possui pelo menos um objeto de aplicação. Existem vários objetos de aplicação padrões na biblioteca de objetos DeviceNet.

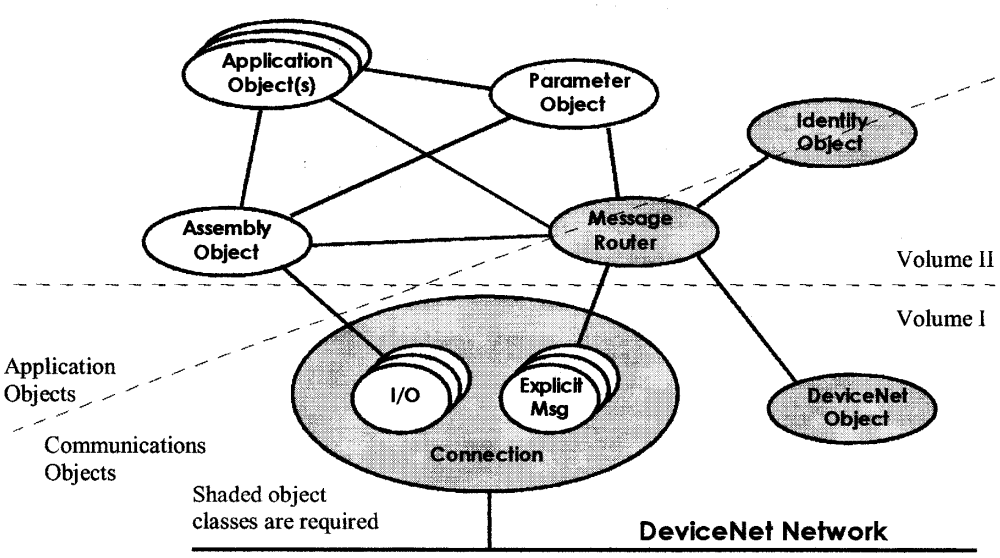


Figura 16: Modelo de objetos DeviceNet

Electronic Data Sheet (EDS)

Um fornecedor de um instrumento DeviceNet para obter seu certificado de conformidade, deve fornecer as informações de configuração de um dispositivo de diversas formas:

- Uma folha de dados impressa
- Uma folha de dados eletrônica (*Electronic Data Sheet*s ou EDS)
- Lista de parâmetro dos objetos
- Combinação das três alternativas anteriores

Electronic Data Sheets são arquivos de especificação associados a um dispositivo. Seu objetivo é definir o conjunto de funcionalidades presentes em um dispositivo e permitir uma rápida configuração dos sistemas computacionais de nível mais alto.

As ferramentas de configuração de alto nível fazem uso destes arquivos para tornar visíveis informações de produtos de múltiplos fornecedores.

Estes arquivos têm formato ASCII e incluem a descrição de atributos essenciais do instrumento como: nome, faixas de operação, unidades de engenharia, tipos de dados, etc. Alguns destes atributos constituem requisitos mínimos para aquela classe de instrumento. Outros são atributos específicos de um fornecedor.

Exemplo EDS

Perfil de um AC Drive

Publicado nas especificações	**A-B** start/stop fwd/rev accel/decel	**Mitsubishi** start/stop fwd/rev accel/decel	**Magnetek** start/stop fwd/rev accel/decel
Adicionado pelo vendedor	**A-B** Unida. Eng Cálculo de potência	**Mitsubishi** Lingua Estrang. Cálculo de temperatura	**Magnetek** Nenhum

Figura 17: *Electronic Data Sheet*

Exemplo Parte do EDS de um sensor fotoelétrico

```
$ DeviceNet 9000 Photoelectric Sensor
$
$ Description: The following file is the EDS for the Allen-Bradley
$              DeviceNet 9000 Photoelectric Sensor
$
```

```
$ Author:       BJT
$ Date:         11/28/94
$
$ Edit History: BJT   11/28/94      Created
$               BJT   11/30/94      Support Rev C
$               BJT   04/10/95      Added Output and Margin
$               BJT   06/01/95      MaxInst should equal # of parameters in EDS
$               BJT   06/23/95      Added IO Info section
$               BJT   8/16/96       Add COS and Diagnostic Mode

[File]
        DescText = "DeviceNet 9000 Photoelectric Sensor EDS File";

        CreateDate = 11-22-94;
        CreateTime = 11:00:00;
        Revision = 1.0;         $ EDS file revision

[Device]
        VendCode = 1;
        ProdType = 6;
        ProdCode = 10;

        MajRev = 2;
        MinRev = 1;

        VendName = "Allen-Bradley";
        ProdTypeStr = "Photoelectric Sensor";
        ProdName = "Series 9000 - Transmitted Beam Receiver";

        Catalog = "42GNR-9000-QD1";

[IO_Info]
        Default = 0x0004;               $ Strobe only
        PollInfo = 0, 0, 0;             $ Not supported
        StrobeInfo = 0x000, 0, 0;       $ Not Supported
        COSInfo = 0x0004, 1, 1;         $ Use Input1 and Output1 for COS
        Input1 =
                1,                      $ 1 byte
                2,                      $ 2 bits used
                0x0002,                 $ Strobe only
                "Sensor Output & Margin", $ Name
                6,                      $ Path size
                "20 04 24 01 30 03", $ Path to ID value attribute
                "Output & Margin from Sensor.  Output value is Bit 0. Margin indication is Bit 1"; $
Help string

[ParamClass]
        MaxInst = 4;
        Descriptor = 0x09;

[Params]
        Param1 =                        $ Operate Mode
                0,                      $ Data Placeholder
                6, "20 0e 24 01 30 08",$ Path size and Path to Operate Mode Attribute
                0x02,                   $ Descriptor - (Support Enumerated Strings)
                4, 1,                   $ Data Type and Size - (16 bit word)
                "Operate Mode",         $ Name
                " ",                    $ Units (Not Used)
                "LIGHT OPERATE [DARK OPERATE] - The output is \"on\"[\"off\"] when \n"
                " the photoelectric sensor(receiver) sees light generated by the light \n"
                " source(emitter).  The output is \"off\"[\"on\"] when the target object breaks \n"
```

```
        " the light beam between source and receiver.  The default is Light Operate.",  $
Help
        0,1,0,          $ min, max, default values
        1,1,1,0,        $ mult, div, base, offset scaling (Not Used)
        1,1,1,0,        $ mult, div, base, offset links (Not Used)
        0;              $ decimal places
```

Perfis de dispositivos

A especificação DeviceNet define muito mais que a conexão física e protocolos. Define também modelos padrões para tipos de dispositivos. O objetivo final é promover a intercambialidade e interoperabilidade entre dispositivos de diferentes fabricantes.

Os perfis de dispositivos definem os requisitos mínimos que cada dispositivo: push button, fotocélulas, atuadores de válvulas pneumáticas, etc. devem possuir para serem considerados compatíveis.

Um perfil de dispositivo deve conter as seguintes seções:

- Definição do modelo de objeto do dispositivo: Lista todas as classes de objeto presentes no equipamento, o número de instâncias em cada classe, como cada objeto afeta o comportamento das interfaces públicas para cada objeto.

- 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 este parâmetros. Estas informações são incluídas no EDS.

Exemplo: Sensor fotoelétrico

Tipo do objeto	Quantidade
Identidade	1
Roteador de Mensagem	1
DeviceNet	1
Conexão	2 (1 explícito, 1 I/O)
Assembly	1
Parametrização	1 (opcional)
Sensor de Presença	1

Conexão do host à rede DeviceNet.

A conexão do host à rede DeviceNet pode se dar através de uma linha serial RS_232, utilizando o adaptador de comunicação 1770- KFD. A conexão pode ser feita em qualquer ponto da rede da DeviceNet.

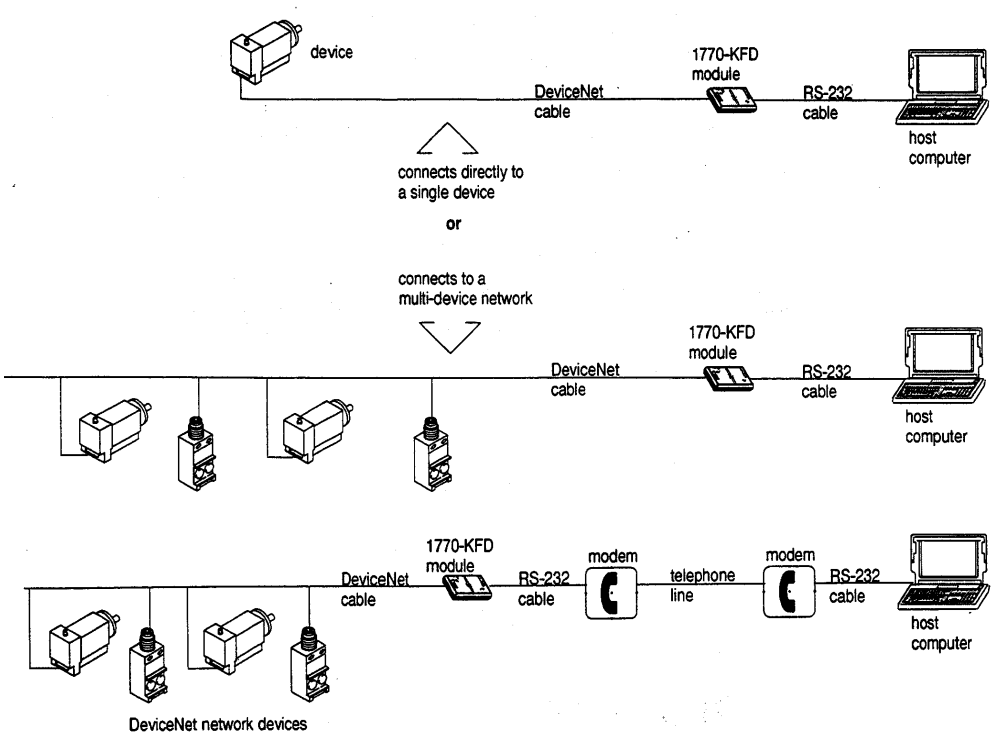


Figura 18: Comunicação da DeviceNet com computador hospedeiro

A comunicação se dá através do protocolo DF1 orientado a caracter, muito semelhante aos protocolos estudados no capítulo 3. O protocolo DF1 combina as características de protocolos da subcategoria D1 (transparência de dados) e F1 (transmissão full-duplex com respostas *embedded*) da norma ANSI x3.28.

Indica começo da mensagem		Contém CAN ID e mensagem DeviceNet	
DF1	PCCC	DeviceNet	DF1
	Comando e contador de bytes de código		Termina mensagem e realiza teste de validade

Figura 19: Formato do pacote de dados

Um segundo envelope representado pelo protocolo PCCC é acrescentado. Dos 5 bytes deste envelope, apenas dois são utilizados: o comando 0x0C que significa mensagem DeviceNet e o contador de pacotes que deve ser incrementado a cada mensagem.

O formato completo da mensagem é mostrado na figura 19. Os dados sombreados são os dados fixos da mensagem.

Camada	Nome	Tipo	Descrição
DF1	DLE	USINT	DLE = 0x10
	STX	USINT	STX = 0x02
PCCC	DST	USINT	Destino = 0 (não usado)
	SRC	USINT	Fonte = 0 (não usado)
	CMD	USINT	Comando = 0x0C (mensagem DeviceNet)
	STS	USINT	Status = 0 (não usado)
	TNSW	UINT	Contador de pacotes. Incrementado a cada mensagem (2 bytes)
Dados	CAN ID	UINT	Identificador CAN
	Dados	vetor de USINT	Dados CAN (Formato DeviceNet, max 8 bytes)
DF1	DLE	USINT	DLE = 0x10
	ETX	USINT	ETX = 0x03
	BCC	USINT	*Block Check Character*

Figura 20: Formato de mensagem do protocolo do 17770-KFD

A comunicação host com o módulo 1770-KFD se dá através de mensagens pré formatadas. O host sempre se comunica com um objeto do módulo. Por exemplo para resetar o módulo devemos nos comunicar com objeto RS-232 e para definir o endereço do nodo devemos nos comunicar com o objeto DeviceNet.

Para inicializar o módulo os seguinte passos devem ser seguidos:

	Processo	Comando	Valor em hexa	Objeto local	Class code do objeto (hex)
Reinicializando o módulo	Reset módulo	reset	05	RS-232	C8
Inicializando o módulo pela primeira vez	Serial link autobaud	DLE ENQ	10 05	N/A	N/A
	Stop service	stop	07	DeviceNet	03
	Define endereço do nodo	set_atribute_single	10	DeviceNet	03
	Define Baud rate	set_atribute_single	10	DeviceNet	03
	Inicia serviço	start	06	DeviceNet	03

Nos exemplos abaixo será assumido que o nodo host tem endereço 0x3E = 62. Este endereço foi escolhido arbitrariamente.

MENSAGENS PADRÕES:

MSG = Reset do módulo

Mensagem enviada

DLE	STX	PCCC	CAN ID	Dados DeviceNet	DLE	ETX	BCC
10	02		FFFF	3E = endereço do nodo 62	10	03	XX
				05 reset			
				C8 RS-232 object			
				01 instância 1			

Resposta

DLE	STX	PCCC	CAN ID	Dados DeviceNet	DLE	ETX	BCC
10	02		FFFF	3E = endereço do nodo 62	10	03	XX
				85 resposta ao reset			

MSG = Serial link autobaud

Mensagem enviada

DLE	ENQ	PCCC	CAN ID	Dados DeviceNet	DLE	ETX	BCC
10	05						

Resposta

DLE	NAK
10	15

Stop Service

Mensagem enviada

DLE	STX	PCCC	CAN ID	Dados DeviceNet	DLE	ETX	BCC
10	02		FFFF	3E = endereço do nodo 62	10	03	XX
				07 stop			
				03 objeto DeviceNet			
				01 instância 1			

Resposta

DLE	STX	PCCC	CAN ID	Dados DeviceNet	DLE	ETX	BCC
10	02		FFFF	3E = endereço do nodo 62	10	03	XX
				87 resposta ao stop			

Define endereço do nodo

O endereço do módulo deve ser igual ao endereço do host (o host e módulo constituem um device único).

Mensagem enviada

DLE	STX	PCCC	CAN ID	Dados DeviceNet	DLE	ETX	BCC
10	02		FFFF	3E = endereço do nodo 62	10	03	XX
				10 set_attribute_single			
				03 objeto DeviceNet			
				01 instância 1			
				01 atributo 1			
				3E 62			

Resposta

DLE	STX	PCCC	CAN ID	Dados DeviceNet	DLE	ETX	BCC
10	02		FFFF	3E = endereço do nodo 62	10	03	XX
				90 resposta ao cmd set_attribute_single			

Define Baud Rate

Mensagem enviada

DLE	STX	PCCC	CAN ID	Dados DeviceNet	DLE	ETX	BCC
10	02		FFFF	3E = endereço do nodo 62	10	03	XX
				10 set_attribute_single			
				03 objeto DeviceNet			
				01 instância 1			
				02 atributo 1			
				00 baud rate			

baud rates:
00 = 125 kbps / 01 = 250 kbps / 02 = 500 kbps

Resposta

DLE	STX	PCCC	CAN ID	Dados DeviceNet	DLE	ETX	BCC
10	02		FFFF	3E = endereço do nodo 62	10	03	XX
				90 resposta ao cmd set_attribute_single			

Start service

Mensagem enviada

DLE	STX	PCCC	CAN ID	Dados DeviceNet	DLE	ETX	BCC
10	02		FFFF	3E = endereço do nodo 62	10	03	XX
				06 start			
				03 objeto DeviceNet			
				01 instância 1			

Resposta

DLE	STX	PCCC	CAN ID	Dados DeviceNet	DLE	ETX	BCC
10	02		FFFF	3E = endereço do nodo 62	10	03	XX
				86 resposta ao start			

Para receber mensagens DeviceNet conectadas o host deve criar filtros denominados *screeners* no módulo de comunicação. *Screeners* devem ser criados para cada dispositivo da rede do qual o host deseja receber mensagens conectadas. Para cada dispositivo um *screener* deve ser criado. Os screeners permanecem até serem apagados pelo host ou quando o módulo fica fora de linha.

Criando screeners

Criar um screener para um dispositivo DeviceNet com endereço 22 = 0x16.

Mensagem enviada

DLE	STX	PCCC	CAN ID	Dados DeviceNet	DLE	ETX	BCC
10	02		FFFF	3E = endereço do nodo 62	10	03	XX
				08 create			
				CB objeto Link			
				00 instância 0			

baud rates:
00 = 125 kbps / 01 = 250 kbps / 02 = 500 kbps

Resposta

DLE	STX	PCCC	CAN ID	Dados DeviceNet	DLE	ETX	BCC
10	02		FFFF	3E = endereço do nodo 62	10	03	XX
				88 resposta a create			
				0100 ID da instância do objeto tipo link criado			

Apagando screeners

Mensagem enviada

DLE	STX	PCCC	CAN ID	Dados DeviceNet	DLE	ETX	BCC
10	02		FFFF	3E = endereço do nodo 62	10	03	XX
				09 apaga			
				CB objeto Link			
				01 instância 1			
				16 endereço do nodo			

Resposta

DLE	STX	PCCC	CAN ID	Dados DeviceNet	DLE	ETX	BCC
10	02		FFFF	3E = endereço do nodo 62	10	03	XX
				09 resposta a delete			

Bibliografia

[Franco 98] Lucia Regina Horta Rodrigues Franco / Victor Marinescu. Buses Actualización. Editorial Control S.R.L, 1998.

[AB 1770-KFD] DeviceNet RS-232 Interface Module Communication Protocol; Cat No 1770-KFD; Allen Bradley

[AB DN-6.7.2] DeviceNet Cable System; Planning and Installation Manual CatNo DN-6.7.2, Allen Bradley

[Rockwell 99] Manual do aluno; Rede DeviceNet – Manutenção e Localização de falhas; ABT-N100-TSM20PT; Agosto 99; Rockwell Automation

[Romito 96] DeviceNet-Technical overview - Ray Romito, DeviceNet Trainer for Rockwell Automaiton/Allen-Bradley (first presented to SI/OEM User Group on April 30, 1996)

[ODVA Overview] DeviceNet- ODVA DeviceNet Technical Overview - http://www.odva.org/10_2/05_fp_tech.htm

[Sense 2001] Sense – Curso Redes Industriais DeviceNet – www.sense.com.br, 2001.

Sites a serem visitados

http://www.odva.org

Exercícios

1) Dada a situação do sistema de esteiras da figura seguinte, determine:

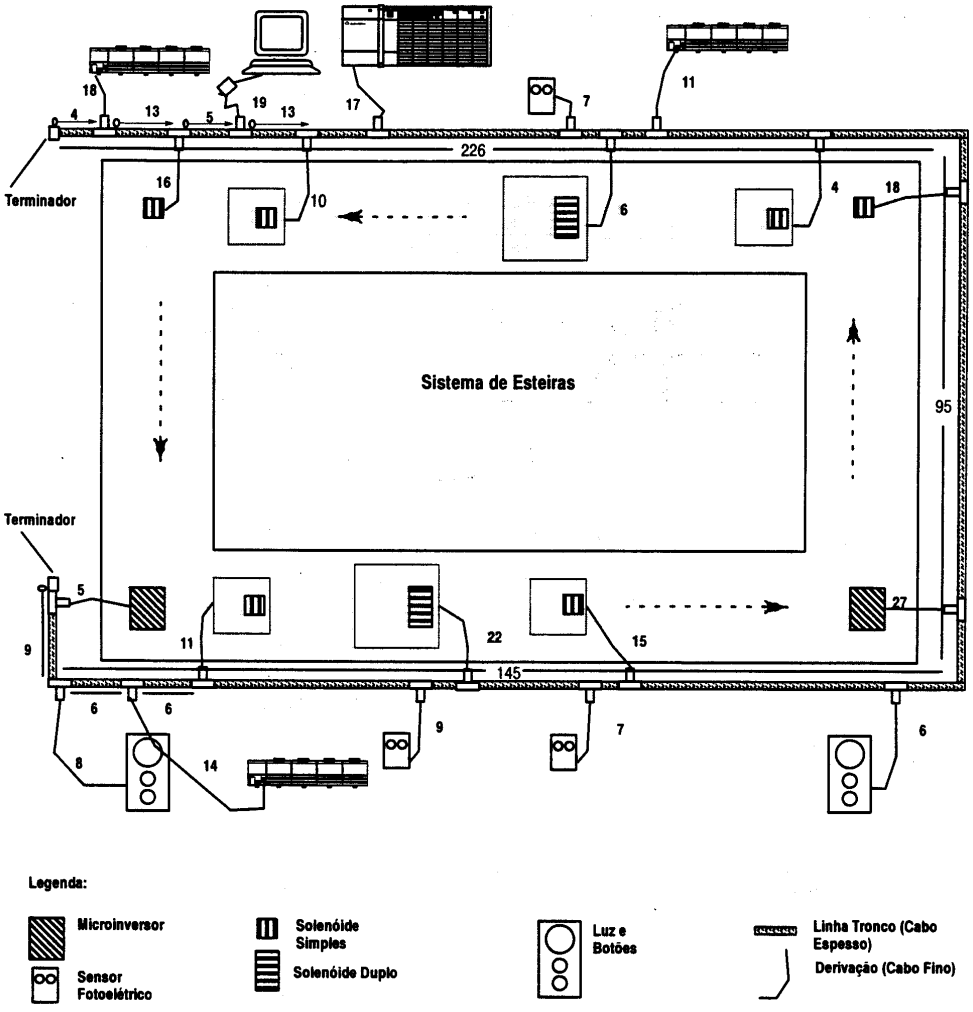
   a) Qual é o comprimento da linha tronco ?

   b) Existe algum comprimento ilegal de derivação no sistema ?

   c) Qual o comprimento de derivação cumulativa ?

   d) Qual o comprimento máximo dos cabos ?

   e) Qual a taxa máxima de comunicação permitida ?



2) O polinômio verificador do protocolo CAN é:

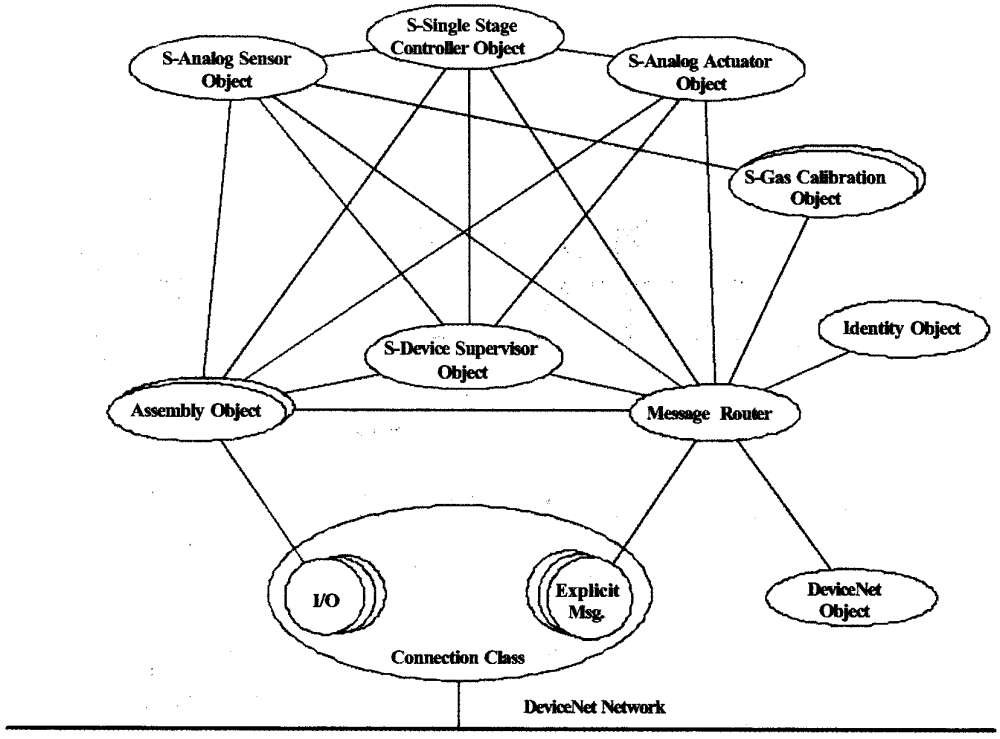
$$X^{15} + X^{14} + X^{10} + X^{8} + X^{7} + X^{4} + X^{3} + 1$$

Desenhe o circuito para realizar a divisão por este polinômio.

3) Marque Verdadeiro ou Falso:

( ) O uso do modelo produtor consumidor diminui a necessidade de banda de passagem da rede.
( ) É impossível implementar o mecanismo de *unsolicited message* na rede DeviceNet.
( ) I/O messages utilizam identificadores de alta prioridade.
( ) O *overhead* do processo de fragmentação é de 12,5% independente do número de fragmentos.
( ) O *device profile* de um dispositivo contém ma descrição apenas dos aspectos de conexão física e de comunicação do produto, mas não especifica as funcionalidades da camada de aplicação.
( ) Por se tratar de um protocolo da família CSMA/CD é de se esperar que ocorra um aumento do número de colisões quando a carga na rede aumenta.
( ) Nesta rede, assim como na ControlNet, é necessário parar a rede para se realizar a inserção de um novo nodo.

4) Examine o modelo de objeto *de Mass Flow Controller* (MFC), retirado de um Device Profile de exemplo, e indique quais os objetos de aplicação presentes.



5) Pesquise sobre o Device Link ControlNet DeviceNet no site indicado e descreva este dispositivo.