

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

" ASPECTOS TECNOLÓGICOS DA INTEGRAÇÃO DE SISTEMAS SENSORES EM  
SISTEMAS DE PRODUÇÃO AUTOMATIZADOS"

DISSERTAÇÃO SUBMETIDA A UNIVERSIDADE FEDERAL DE SANTA  
CATARINA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA  
MECÂNICA

PAULO SÉRGIO MARTINS PEDRO

FLORIANÓPOLIS, AGOSTO DE 1993

ASPECTOS TECNOLÓGICOS DA INTEGRAÇÃO DE SISTEMAS SENSORES EM  
SISTEMAS DE PRODUÇÃO AUTOMATIZADOS

PAULO SÉRGIO MARTINS PEDRO

ESSA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO  
TÍTULO DE MESTRE EM ENGENHARIA, ESPECIALIDADE ENGENHARIA MECÂNICA,  
ÁREA DE CONCENTRAÇÃO METROLOGIA E AUTOMAÇÃO, APROVADA EM SUA FORMA  
FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA.

PROF. HERMANN ADOLF HARRY LÜCKE, DR. ING. - ORIENTADOR  
PROF. BEREND SNOEIJER, PhD. - COORDENADOR

BANCA EXAMINADORA:



PROF. HERMANN A. H. LÜCKE, DR. ING.



PROF. CARLOS A. MARTIN, DR. ING.



PROF. MARCELO RICARDO STEMMER, DR. ING.

## AGRADECIMENTOS

Ao Departamento de Engenharia Mecânica da Universidade Federal de Santa Catarina (UFSC) e ao Centro Regional de Tecnologia em Informática (CERTI) pela infra-estrutura colocada à disposição.

Ao CnPq pela concessão da bolsa de mestrado.

Ao professor Hermann A. H. Lücke pela orientação e apoio.

Ao Departamento de Informática da Universidade Estadual de Maringá (UEM) pelas condições oferecidas para a conclusão deste trabalho.

Aos colegas do CERTI/LABMETRO e amigos da UEM que contribuíram diretamente ou não para a execução do presente trabalho.

A Deus pela provisão farta de energia e estímulo para a liquidação deste compromisso.

A minha família pelo incentivo.

Aos meus pais

J. Pedro (in memoriam) e

M. Luz Martins Pedro

## ÍNDICE

RESUMO.....	i
ABSTRACT.....	ii
ZUSAMMENFASSUNG.....	iii
GLOSSÁRIO.....	iv
1. INTRODUÇÃO.....	01
1.1 Contexto do trabalho.....	06
1.2 Estado da arte.....	09
1.2.1 Tecnologia de redes de comunicação fabril.....	09
1.2.2 Tecnologia de sensores.....	12
1.3 Proposta de trabalho.....	16
2. REQUISITOS DAS APLICAÇÕES PARA A INTEGRAÇÃO DOS SISTEMAS SENSORES EM SISTEMAS DE PRODUÇÃO.....	18
2.1 Caracterização da planta ou processo.....	18
2.1.1 Processos contínuos.....	18
2.1.2 Processos discretos.....	18
2.1.3 Processos Batch.....	19
2.2 Requisitos das aplicações.....	19
2.2.1 Número de variáveis do processo.....	19
2.2.2 Dispersão física das variáveis de processo na instalação de produção.....	20
2.2.3 Comportamento dinâmico das variáveis de processo.....	20
2.2.4 Condições ambientais.....	21
2.2.5 Tamanho da mensagem.....	22
2.2.6 Necessidade de manipulação de eventos e programas...	23
3. CLASSIFICAÇÃO DAS SOLUÇÕES PARA INTEGRAÇÃO DOS SISTEMAS SENSORES.....	24
3.1 Critério para classificação.....	24
3.2 Configurações fortemente acopladas.....	25
3.2.1 Multibus I.....	26

3.3	Configurações medianamente acopladas.....	27
3.4	Configurações fracamente acopladas.....	29
4.	A INTERFACE FIELDBUS (BARRAMENTOS DE CAMPO).....	30
4.1	Caracterização.....	30
4.2	Implicações.....	30
4.2.1	Vantagens a nível econômico.....	31
4.2.2	Vantagens a nível técnico.....	31
4.3	Arquitetura de Fieldbus.....	32
4.3.1	A Camada Física.....	34
a)	meio de transmissão.....	34
a.1)	topologia.....	34
a.2)	Tipo de codificação do sinal.....	36
a.3)	Modo de transmissão.....	38
a.4)	Taxa de transmissão.....	41
4.3.2	A Camada de Enlace.....	42
a)	Mecanismo de controle de acesso ao meio....	42
a.1)	Barramentos com MAC centralizado....	42
a.2)	Barramentos com MAC distribuído....	43
b)	Distância de Hamming (Hd).....	44
4.3.3	A Camada de Aplicação.....	44
4.3.4	Arquiteturas abertas.....	45
4.4	Exemplos de especificações de projetos Fieldbus.....	47
4.4.1	Bitbus.....	47
4.4.2	Profibus.....	48
a)	Camada Física.....	48
b)	Camada de Enlace.....	49
c)	Camada de Aplicação.....	51
d)	Implementação Profibus.....	52
d.1	Terminal Concentrador de dispositivos..	52
d.2	Sistema de medição de deslocamento do tipo laser de alinhamento com interface Profibus.....	53
d.3	Terminal com entradas/saídas digitais..	53
d.4	Estação Profibus para micro IBM-PC....	53
4.4.3	Fip.....	54
4.4.4	Projeto SP-50 (ISA).....	55
4.4.5	Sercos.....	56
4.4.6	Auto-bus.....	56

4.5	Comentários sobre o Fieldbus no contexto da automação industrial.....	58
5.	MAPEAMENTO DOS REQUISITOS DE APLICAÇÃO EM REQUISITOS DE COMUNICAÇÃO FIELDBUS - ASPECTOS DA INTEGRAÇÃO FÍSICA DE SISTEMAS SENSORES.....	65
5.1	Variáveis de Processo.....	66
5.1.1	Dispersão física das variáveis de processo	
	a) alcance x taxa de transmissão.....	66
	b) alcance x tipo de suporte.....	66
	c) alcance x tipo de codificação.....	67
	d) alcance x repetidores.....	68
5.1.2	Número de variáveis a controlar.....	68
5.1.3	Comportamento dinâmico das variáveis.....	69
	a) atraso de transferência.....	69
	a.1) tempo de processamento do protocolo.....	69
	a.2) tempo de transmissão.....	70
	a.3) comprimento do suporte de transmissão.....	70
	b) tipo de suporte de transmissão.....	71
	c) número de estações do Fieldbus.....	71
5.2	Condições ambientais.....	71
5.3	Tamanho da mensagem da aplicação.....	74
5.4	Necessidade de gerenciamento remoto de programas, eventos e acesso à variáveis.....	74
6.	A CAMADA DE APLICAÇÃO DE FIELDBUS -ASPECTOS DA INTEGRAÇÃO LÓGICA DE SISTEMAS SENSORES.....	75
6.1	Estabelecimento do Dispositivo de Campo Virtual (VFD-Virtual Field Device).....	75
	6.1.1 Objetos de comunicação.....	76
	6.1.2 Dicionário de Objetos.....	78
6.2	Definição dos serviços FMS e das funções da ALI.....	80
	6.2.1 Serviços de Gerenciamento remoto de eventos.....	82
	6.2.2 Serviços de Gerenciamento remoto de programas...	83

6.2.3	Serviços de Acesso a Domínios.....	83
6.2.4	Serviços de Acesso a Variável.....	84
6.2.5	Serviços de Gerenciamento de contexto.....	84
6.2.6	Serviços de Gerenciamento do OD.....	85
6.2.7	Serviço de Suporte VFD.....	85
6.3	Configuração do Fieldbus.....	86
7.	EXEMPLO DE APLICAÇÃO.....	88
7.1	Configuração do sistema.....	88
7.2	Operação do sistema.....	90
7.3	Estabelecimento do Dispositivo de Campo Virtual.....	92
7.3.1	Cabeçalho.....	96
7.3.2	Sub-dicionário estático de tipos (ST-OD).....	97
7.3.3	Sub-dicionário estático de objetos (S-OD).....	98
	a) Objeto medida (variável).....	98
	b) Objeto Erro_posição (evento).....	100
	c) Objeto dom_med, dom_cal (domínios).....	101
7.3.4	Sub-dicionário dinâmico Invocação de programa (DP-OD).....	103
	a) Objeto Calibração e Medição (Pi).....	103
7.4	Definição dos serviços e das funções da ALI.....	104
7.5	Representação e Acesso aos objetos processos.....	107
7.5.1	Forma de acesso ao objeto Medida.....	110
7.6	Programação da aplicação.....	113
7.7	Considerações finais sobre a camada aplicação.....	121
8.	CONCLUSÃO.....	122
9.	REFERÊNCIAS BIBLIOGRÁFICAS.....	126
10.	ANEXO A.....	135



## RESUMO

O crescente nível de automação em sistemas de produção é acompanhado pelo crescente número de dispositivos de sensoriamento e de atuação. O potencial avanço na eficiência e flexibilidade de sistemas de produção reside na integração destes dispositivos primários no sistema integrado de automação. Em contraste com os demais equipamentos em ambientes de produção, os sistemas sensores e atuadores caracterizam-se pela ampla quantidade de fornecedores, variedade de aplicações e de tipos, o que evidencia a dificuldade de se realizar a integração.

Para a integração de equipamentos de automação de maior porte (p. Ex. máquinas CNC e robôs) existem vários barramentos seriais padronizados e proprietários disponíveis tais como Mini-map e Ethernet, com interface de barramento relativamente poderosa e de alto custo. Para o grande número de pequenos dispositivos digitais utilizados em sistemas de produção, incluindo-se nestes os sistemas sensores e atuadores, tais barramentos são economicamente inviáveis e apresentam um baixo desempenho devido aos elevados requisitos de comunicação verificados.

O presente trabalho oferece um suporte à integração de sistemas sensores, mostrando o caminho para a comunicação aberta e a inserção destes no fluxo de informação dos modernos sistemas de produção automatizados. Para isto são considerados os aspectos físicos e lógicos desta integração, realizada através da chamada interface Fieldbus. Adicionalmente é apresentado um exemplo de aplicação com a respectiva análise dos passos para a utilização de um Fieldbus como ferramenta de integração de sistemas sensores.

## ABSTRACT

The increasing level of automation in production systems is accompanied by an increasing number of sensor and actuator devices (field devices). The production system's potential for a large leap forward lies here, in the inexpensive integration of numerous items of small sized automation equipment in the integrated automation system. In contrast with other devices of production environments, sensor and actuator systems are characterized by the diversity of manufacturers and types and by a variety of applications which puts in evidence the difficulty of accomplishing the desired integration.

In the integration of larger automation equipments (e.g. CNC machines and robots), both company specific and standardized serial buses such as Mini-Map and Ethernet are the current state of art. Their bus interfaces are very powerful but also very expensive. For the large number of small digital units used in automated production plants, as sensor and actuator systems, this cost is simply too high. In addition, the given requirements allow reduced performance.

This work supports the integration of field level devices, showing the way toward open communications of sensor systems and its insertion at the information flow of modern automated production systems. Two related relevant issues are being considered: the physical integration and the logical one. Both are achieved by means of a standardized Fieldbus network. Additionally, the work presents a simplified example of implementation analysing the steps to use a Fieldbus as a means of integrating sensor systems into production systems.

## ZUSAMMENFASSUNG

Das zunehmende Niveau der Automatisierung in der Produktionstechnik ist zwingend mit einer zunehmenden Zahl von Feldgeräten verbunden. Ein möglicher Fortschritt in Flexibilität und Leistung von Produktionssystemen liegt in der Einbindung dieser Größen. Im Gegensatz zu anderen Geräten in Produktionsanlagen sind einfache Sensor bzw Aktorsysteme durch eine Vielzahl von Herstellern und Anwendungen gekennzeichnet, was Schwierigkeiten bei der Einbindung dieser Geräte verursacht.

Zur Einbindung von grossen Automatisierungsgeräten (CNC Steuerung und Roboter), sind sowohl standardisierte als auch herstellerspezifische serielle Busse, wie Mini-map und Ethernet verfügbar, die mit einer sehr leistungsfähigen und teuren Schnittstelle ausgestattet sind. Für die Vielzahl kleiner, einfacher Digitalegeräte, die in Produktionssystemen benützt werden, wie z.B Sensoren und Aktoren, sind diese Busse unwirtschaftlich.

Die vorliegende Arbeit soll bei der Einbindung von Sensorsystemen Unterstützung bieten, indem sie den Weg von der herstellerspezifischen zur offenen Kommunikation von Sensorsystemen in der modernen Produktion aufzeigt. In diesem Zusammenhang müssen zwei Gesichtspunkte berücksichtigt werden: Zum einem die logische und zum anderen die physische Einbindung. Beide werden mit einer sogenannten Feldbus Schnittstelle realisiert. Die Arbeit enthält darüber hinaus ein Anwendungsbeispiel mit der Analyse der notwendigen Schritte zur Anwendung eines Feldbussystems für Sensoren und dessen Einbindung in den Informationsfluss integrierter Produktionssysteme.

GLOSSÁRIO  
DAS SIGLAS E ACROGRAFIAS ADOTADAS NO TEXTO

SIGLA	INGLÊS	PORTUGUÊS
A/D	Analog/digital	Analógico/digital
ALI	Application Layer Interface	Interface da camada de aplicação
AMH	Automated Materials Handling	Veículo automatizado de transporte
CAM	Computer Aided Manufacturing	Manufatura auxiliada por computador
CAQ	Computer Aided Quality	Qualidade auxiliada por computador
CAD	Computer Aided Design	Projeto auxiliado pelo computador
CIM	Computer Integrated Manufacturing	Manufatura integrada por computador
CAE	Computer Aided Engineering	Engenharia auxiliada pelo computador
CAP	Computer Aided Production	Produção auxiliada pelo computador
CAPP	Computer Aided Process Planning	Planejamento do processo auxiliado por computador
CLP		Controlador Lógico Programável

CRL	Communication Relations List	Lista das Relações de comunicação
CSMA/CD	Carrier Sense multiple Access/Collision Detection	Acesso múltiplo com detecção de portadora/detecção de colisão
DCDS	Distributed Control Digital System	Sistema Digital de Controle Distribuído
DP-RAM	Dual Port Random Access Memory	Memória Ram de duas portas
DSAP	Destination Service Access Point	Ponto de acesso ao serviço destino
D/A	Digital/Analog	Digital/Analógico
EMI	Eletromagnetic Interference	Interferência eletromagnética
EPA	Enhanced Performance Architecture	Arquitetura de desempenho otimizado
FMC	Flexible Manufacturing Cell	Célula flexível de manufatura
FMS	Flexible Manufacturing System	Sistema flexível de manufatura
FMS	Fieldbus Messaging System	Sistema de mensagem de Fieldbus
GT	Group Technology	Tecnologia de grupo
Hd	Hamming Distance	Distância de Hamming

IEC	International Electrotechnical Comission	Comissão Eletrotécnica In- ternacional
IS	Intermediate System	Sistema intermediário
ISA	Instrument Society of America	Sociedade Americana de Ins- trumentação
ISO	International Standardiza- tion Organization	Organização Internacional de Padronização
LAN	Local Area Network	Rede local de comunicação
LLC	Logical Link Control	Controle do enlace lógico
LLI	Lower Layer Interface	Subcamada inferior da camada de Aplicação
LSAP	Link Service Access Point	Ponto de acesso ao servi- ço de enlace
MAC	Medium Access Control	Controle de acesso ao meio
MAP	Manufacturing Automation Protocol	Protocolo de automação da manufatura
MMS	Manufacturing Message Especification	Especificação da mensagem da manufatura
NC	Numeric Control	Controle numérico
OSI	Open System Interconnection Interconection	Interconexão de sistemas abertos
OD	Object Dictionary	Dicionário de objetos

PCP	Process Control Planning	Planejamento do controle do processo
PDU	Protocol Data Unit	Unidade de dados de protocolo
RC	Robot Control	Controle de Robô
SAP	Service Access Point	Ponto de acesso ao serviço
SDU	Service Data Unit	Unidade de dados de serviço
SPC	Statistical Process Control	Controle estatístico do processo
TOP	Technical and Office Protocols	Protocolo de automação de escritório
VFD	Virtual Field Device	Dispositivo de campo virtual

## 1. INTRODUÇÃO

O crescente nível de automação na tecnologia de produção está estreitamente relacionado com a integração da tecnologia de sensores, visto que os sistemas sensores integrados constituem uma base fundamental para a supervisão, controle de máquinas como também para a garantia da qualidade.

O modelo simplificado para integração de sistemas sensores e atuadores em sistemas de produção automatizados aqui definido envolve essencialmente os elementos representados na fig. 1.1.

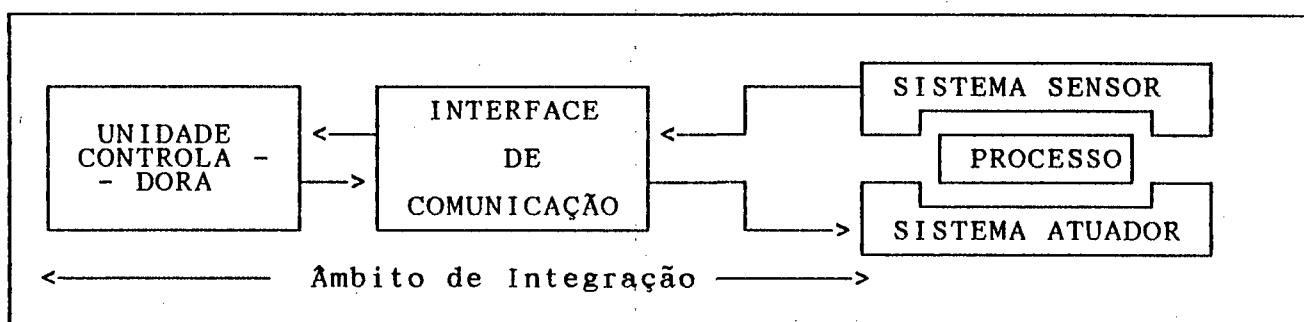


Fig. 1.1 Modelo simplificado para integração de sistemas sensores e atuadores

A unidade controladora é o elemento responsável primordialmente pela execução das tarefas de aquisição, monitoração/supervisão, e controle do processo. Baseia-se em informações que adquire deste processo em suas interações normalmente cíclicas com o(s) sistema(s) sensor(es) (SS) para exercer subsequente processamento e se necessário for, atuação sobre o mesmo através dos sistemas atuadores. A interação entre unidade controladora e os sistemas sensor e atuador se faz através de uma interface de comunicação.

Distingue-se dois aspectos de integração: A integração física (de sistemas sensores) assegura a conexão e o transporte de dados entre estes e a unidade controladora, de forma seqüencial e isenta de erros. Um passo adiante da integração física é a integração lógica, a qual permite a comunicação e a cooperação entre os programas de aplicação sendo executados respectivamente no sistema sensor e no controlador. A figura 1.2 ilustra a arquitetura de um sistema sensor virtual [1].



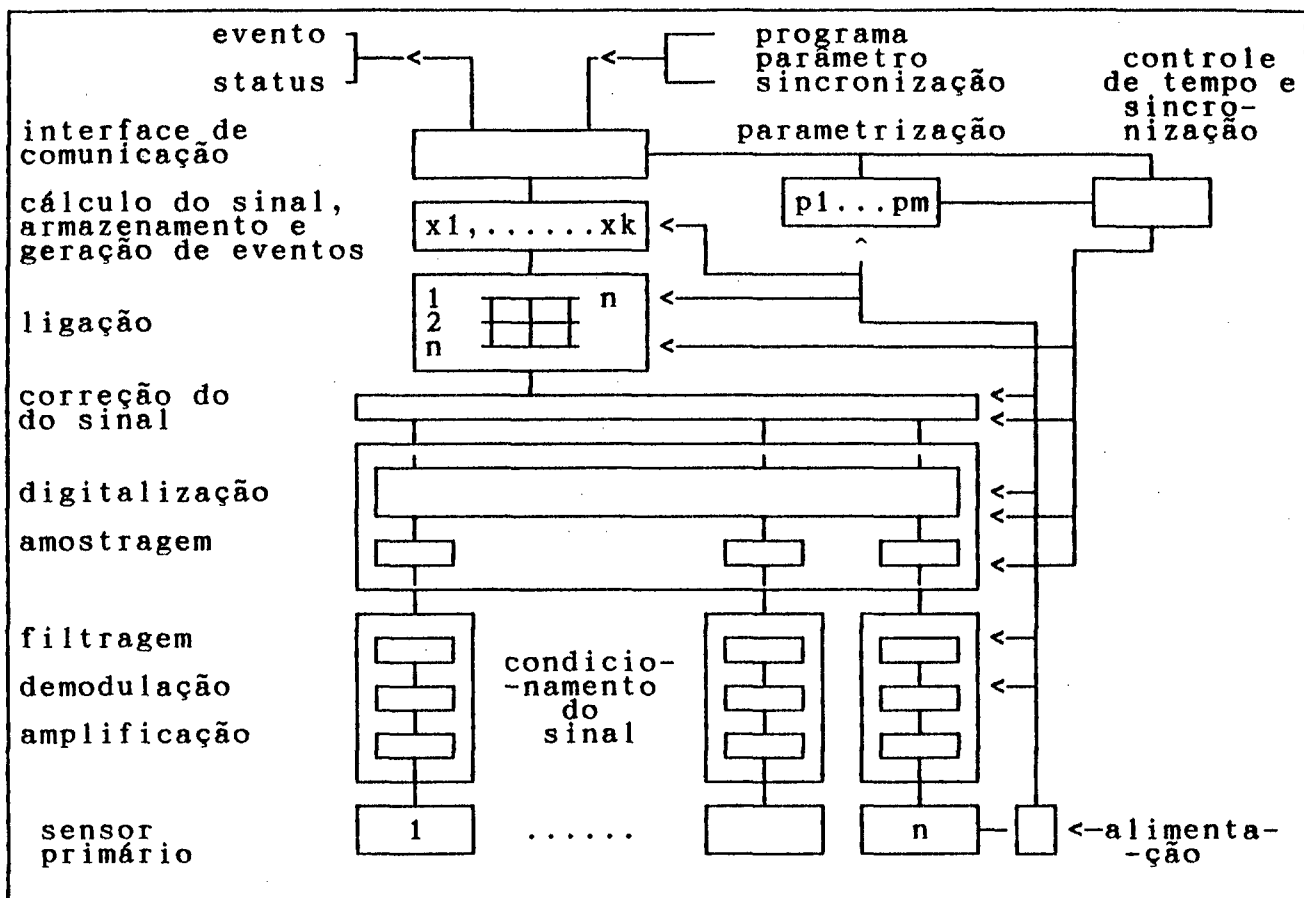


Fig 1.2 Arquitetura de um sistema sensor virtual

Os módulos que compõem esta arquitetura são os seguintes:

a) Sensores de base

Responsáveis pela interação com a grandeza de processo e geração do sinal elétrico correspondente. Já na área de sensores de base se faz presente a tecnologia digital. Tem-se por exemplo utilizado para a medição de comprimento e ângulo sistemas digitais opto-eletrônicos com bons resultados obtidos. Sensores de base com saída em frequência, apesar das reconhecidas vantagens, conseguiram até agora resultados relativamente modestos. Uma exceção são alguns sensores de temperatura a quartzo e sensores de pressão.

b) Alimentação

Este módulo está estreitamente relacionado com o sensor de base empregado. É responsável pela geração da energia auxiliar na forma

de corrente, tensão e frequência.

c) Condicionamento do sinal

Este módulo é responsável pela amplificação, filtragem e demodulação do sinal. O condicionamento do sinal é para a maioria dos sistemas sensores insubstituível, muito embora possa ser substancialmente reduzido com o desenvolvimento de sensores de melhor qualidade. Na realidade a solução ótima de condicionamento de sinal envolve uma combinação adequada de eletrônica analógica e digital. Para a amplificação de pequenos sinais dispõe-se de componentes eficientes e integrados tais como o amplificador operacional e inclusive osciladores e demoduladores integrados em uma pastilha de circuito integrado.

d) Digitalização

Há disponível para a amostragem e digitalização componentes consideravelmente eficientes, os quais representam um custo representativo para os sistemas sensores principalmente quando se objetiva baixa incerteza de medição e alta velocidade. Conversores digitais de tensão-frequência por exemplo alcançam atualmente uma boa linearidade. Uma baixa incerteza de medição no entanto compromete a velocidade da medição.

e) Correção do sinal

O módulo de correção de sinal remedia falhas as quais especificamente estão relacionadas com os sensores de base e com a respectiva eletrônica de condicionamento de sinal. Exemplos de correções a serem realizadas são falhas estáticas como erro de ponto zero, de fatores de amplificação como não linearidades e falhas dinâmicas como ruído elétrico.

#### f) Ligação

Em sistemas multisensores cada sinal adquirido isoladamente deverá ser ligado com os demais para formar uma medida. (Por exemplo a leitura de corrente e tensão para formar a medida da resistência em um condutor). O controle deste processo é realizado no módulo de ligação.

#### g) Processamento, armazenamento e geração de eventos

Para o processamento do sinal requer o sistema sensor um conjunto de funções básicas de processamento matemático, como por exemplo funções algébricas, funções elementares transcendentais e funções não elementares. Exemplos de funções específicas de processamento em sistemas sensores são transformações de coordenadas, correlação linear, funções estatísticas, correções dinâmicas (filtragem p. Ex.) e transformações de código.

O cálculo do valor da medida está relacionado com a geração de eventos, ou seja, neste módulo também são gerados eventos que são dependentes de condições pré-afixadas e dos dados obtidos no sistema sensor. Um exemplo disto é quando na supervisão de um processo determinada grandeza ultrapassa um valor limite causando o disparo de um evento alarme.

#### h) Parametrização

O módulo de parametrização tem acesso a todos os elementos funcionais e tem o propósito de armazenar os parâmetros operacionais para a configuração do sistema sensor. Exemplos de parâmetros de configuração são o fator de amplificação, taxa de amostragem, frequência de corte e tipo de função de processamento.

#### i) Controle de tempo e sincronização

Este módulo controla temporalmente os outros componentes. Dispara

por exemplo a amostragem, providencia o envio cíclico dos dados de medição para a Unidade Controladora e é responsável pelo fornecimento do tempo como grandeza.

#### j) Comunicação

O módulo de comunicação é um dos mais importantes componentes da nova geração de sistemas sensores. Representa a interface para outros sistemas (por exemplo controladores e sistemas atuadores) para a transmissão e recepção de dados, eventos, parâmetros e programas. Os aspectos de interesse relacionados a este módulo serão tratados mais especificamente à partir do capítulo 4.

Os sistemas sensores baseados nesta arquitetura podem ser classificados em cinco classes distintas:

- A classe A contém somente sensores primários ou também sensores de base;
- A classe B engloba os sensores primários com o condicionamento do sinal;
- A classe C envolve a classe B e os elementos associados à conversão do sinal para o nível digital (conversores A/D) ;
- A classe D compreende adicionalmente a correção do sinal;
- Sistemas sensores da classe E abrangem todo o âmbito de funções ilustradas na fig. 1.2 e podem agregar, a título de exemplo, a seguinte funcionalidade:
  - Auto calibração automática;
  - Aquisição de sinais mediante o acoplamento de sensores tradicionais ou integrados em chip, os quais já se encontram disponíveis para pressão, temperatura e campo magnético;
  - Digitalização e codificação dos valores de medição;
  - Filtragem do sinal digital;
  - Processamento estatístico;

- Correção automática de erros sistemáticos como não linearidades.

A integração de sistemas sensores em sistemas de produção como fica evidente requer a definição e implementação de uma interface de comunicação através da qual a informação gerada pelos sistemas sensores é enviada para o controlador onde passa a ter real significado para a aplicação. Este controlador por sua vez pode estar conectado através de uma interface de comunicação de maior capacidade aos níveis superiores de controle provendo assim um fluxo de informação universal nos modernos sistemas de produção. Embora os sistemas sensores e atuadores possam diferir entre si funcional ou arquiteturalmente, ambos interligam-se ao controlador pelo mesmo tipo de interface de comunicação, razão pela qual a análise que se segue no contexto do trabalho se estende a ambos os tipos de dispositivos de campo.

### 1.1 Contexto do trabalho

Em toda estratégia moderna de automatização estão implícitos conceitos e tecnologias preconizados pela filosofia CIM (Computer Integrated Manufacturing), cujo objetivo basicamente reside na integração das tecnologias computacionais de suporte à manufatura (as conhecidas CAx), dentro de um planejamento unificado que idealiza a otimização global da operação da empresa [2,3,4,5,6,7]. Uma visão geral do estado atual das diversas tecnologias que compõem o CIM, estimadas quanto ao seu grau de maturação tecnológica, está ilustrada na figura 1.3.

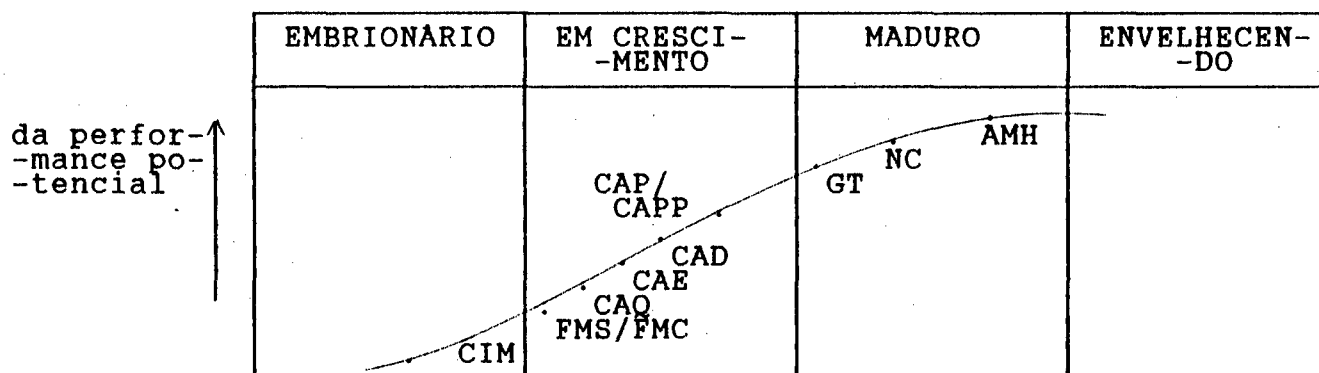


Fig. 1.3 Grau de amadurecimento das tecnologias baseadas em computador para automação industrial.

Este processo de integração exige grande esforço de uniformização de processos, fluxos (de materiais e informação), sistemas, equipamentos e bases de dados.

A figura 1.4 apresenta a clássica estrutura funcional do CIM, sendo o modelo constituído por seis níveis hierárquicos.

NÍVEIS	MISSÃO	ESCOPO	ESCALA DE TEMPO
5 ADMINISTRAÇÃO CORPORATIVA	GESTÃO DA COMPANHIA	OPERAÇÃO DA COMPANHIA	ANOS E MESES
4 PLANEJAMENTO DA PLANTA	GERENCIAMENTO DA FABRICA, USINA, ETC.	OPERAÇÃO DA FABRICA, USINA, ETC..	MESES E SEMANAS
3 AREA (FMS)	GERENCIAMENTO DO PROCESSO	OPERAÇÃO DAS AREAS	SEMANAS E DIAS
2 CÉLULA	SUPERVISÃO DO PROCESSO	OPERAÇÃO DAS UNIDADES	HORAS E MINUTOS
1 UNIDADES DE PRODUÇÃO	CONTROLE DO PROCESSO OU DE MÁQUINAS	OPERAÇÃO DO PROCESSO	MINUTOS E SEGUNDOS
0 DISPOSITIVO	CONTROLE DE DISPOSITIVOS	OPERAÇÃO DE ATUADORES	MINUTOS, SEGUNDOS E MILISEG.

Fig. 1.4 Estrutura funcional associada aos níveis hierárquicos de um sistema CIM.

No nível 5, o gerenciamento tem acesso a qualquer sinal de controle de hierarquia e do mundo exterior, de forma a obter informações para tomada de decisão. Utilizam-se computadores de grande porte para permitir a administração da corporação como um todo, incluindo tarefas de escritório e pessoal administrativo.

O nível 4 é responsável pelo planejamento de produto, processo e produção, tendo como principal atribuição as funções referentes ao planejamento e controle da produção (PCP), quando a planta estiver na fase de produção propriamente dita. A geração do plano mestre de produção e suas explosões e decomposições até se chegar às ordens liberadas para o chão de fábrica é a principal meta.

O nível 3 é incluído na hierarquia quando dois ou mais controladores do nível 2 estão incorporados ao subsistema de chão de fábrica, sendo denominada genericamente por área de produção. Por exemplo, duas células flexíveis de manufatura coordenadas/gerenciadas por um controlador formam uma área de produção denominada sistema flexível de manufatura. A principal atribuição do nível 3, dentre outras, diz respeito ao detalhamento dos planos de produção, para serem executados no chão de fábrica.

A função básica do nível 2 é de controlar/coordenar um conjunto de unidades de produção que definem uma sub-área da planta, como por exemplo uma célula de manufatura. Distúrbios que ocorram neste ou no nível inferior devem disparar ações corretivas para manter/restabelecer a seguridade e operacionalidade dos equipamentos, a segurança dos homens e a execução do plano de produção (ordens liberadas).

O nível 1 é responsável pelo controle operacional da máquina/processo incluindo as unidades de produção computadorizadas, mais especificamente os computadores industriais, controladores lógico programáveis (PLC) e comando numérico (CNC), que estão incorporados aos equipamentos de produção tais como máquinas-ferramenta, manipuladores, máquinas de medição por coordenadas.

O nível 0 representa o âmbito de interesse do trabalho, onde reside toda a problemática a ser tratada nos capítulos que seguem. Responde pelo controle interno das unidades de chão de fábrica (máquina-ferramenta, robôs, manipuladores, etc..) através de dispositivos simples de automação como sistemas sensores e atuadores (contadores, totalizadores, sensores de temperatura, torque, motores, chaves de fim de curso etc.). Neste nível estão também presentes as configurações onde tais dispositivos interagem com uma unidade de controle/monitoração através de uma interface de comunicação no controle de processos distribuídos no âmbito externo à unidade de chão de fábrica.

## 1.2 Estado da arte

Primariamente a tecnologia de redes de comunicação foi trazida para a indústria para a interligação de componentes de automação tais como CNC's, controladores de célula, robôs industriais, e num nível acima, dos computadores da área de engenharia. Recentemente com o desenvolvimento da tecnologia dos sistemas sensores, passou a se vislumbrar a utilização da tecnologia de redes também para a interligação de sistemas sensores e atuadores a seus respectivos controladores. Portanto a seguir será tratado separadamente o estado atual de cada um destas tecnologias consideradas, a de comunicação fabril e a de sistemas sensores.

### 1.2.1 Tecnologia de redes de comunicação fabril

Um dos fatores mais importantes no projeto e implementação de sistemas de CIM é a disponibilidade de redes de comunicação de dados. Diferentes protocolos e um grande número de barramentos de comunicação tem sido desenvolvidos e estão em uso pela indústria, sendo no entanto incompatíveis entre si. Esta incompatibilidade dificulta a obtenção da rede de comunicação e conseqüente integração das hierarquias de controle. [8]

O primeiro passo em direção da uniformização das interfaces de comunicação entre aplicações na manufatura automatizada, os projetos MAP na área de produção e seu complemento TOP na administração e escritórios, estabeleceram as condições necessárias para esta integração operacional idealizada (CIM), consolidando-se portanto como propostas concretas para o ambiente fabril [9,10]. Uma especificação de destacada importância no MAP é o protocolo MMS [11,12,13,14,15] (Manufacturing Messaging System), o qual especifica funções de aplicação elementares para a comunicação entre dispositivos da automação da manufatura. O âmbito de funções abrange a comunicação de CLPs, controladores NC e RC até controladores de célula.

Um sistema de comunicação convencional pode ser visto como composto de diversas camadas relacionadas a diferentes tipos de funções e serviços (ver item 4.3). A estruturação em camadas tem por obje-



-tivos aumentar a eficiência e a facilidade de utilização bem como auxiliar no domínio da complexidade. Estas camadas vão sendo construídas a partir do enlace físico de transmissão até o ponto que atendam aos requisitos do usuário. Para tornar factível a interconexão de equipamentos heterogêneos, tornou-se necessário o estabelecimento de normas e padrões para cada camada dos sistemas de comunicação. A iniciativa mais importante neste sentido partiu da ISO, que elaborou um modelo básico de referência da arquitetura de sistema de comunicação referido como OSI. A finalidade do modelo de referência OSI-ISO é prover uma base comum para a coordenação do desenvolvimento de padrões com o propósito de interconectar sistemas. O modelo de referência propõe uma arquitetura, para o sistema de comunicação, dividida em sete camadas. A figura 1.5 ilustra para cada camada os padrões de protocolos ISO que compõe o MAP/TOP e o Mini-MAP.

CAMADA	TOP	MAP	MINI-MAP
APLICAÇÃO 7	CASE/ACSE (DIS 8649) FTAM (DIS 8571) VT	CASE/ACSE (DIS 8649) FTAM (DIS 8571) MMS (DP 9506)	MMS (DP 9506)
APRESENTAÇÃO 6	DIS 8822		VAZIO
SESSÃO 5	IS 8326/8327		
TRANSPORTE 4	IS 8072/8073 CLASSE 4		
REDE 3	IS 8473 (8348/8648) SEM CONEXÃO		
ENLACE 2	LLC: IS 8802.2 TIPO 1 MAC: IS 8802.3	LLC: IS 8802.2 TIPO 1 e 3 MAC: IS 8802.4	LLC: IS 8802.2 TIPO 3 MAC: IS 8802.4
FÍSICO 1	BANDA BASE 10 e 5 Mbps	BANDAS: PORTADORA, 5 e LARGA, 10 Mbps	BANDA PORTADORA 5 Mbps

Fig. 1.5 - Âmbito dos padrões MAP/TOP E Mini-Map conforme as normas ISO.

Contrariando os objetivos iniciais do MAP/TOP, fica evidente hoje a impossibilidade de se atender a todos os requisitos de comunicação na fábrica com um tipo de solução de rede. No mínimo configuram-se desde já tres níveis de comunicação (fig. 1.6) [16,17]:

- Redes de nível 2 ou redes de gerenciamento e operação da planta: Interconectam normalmente computadores das áreas de engenharia,

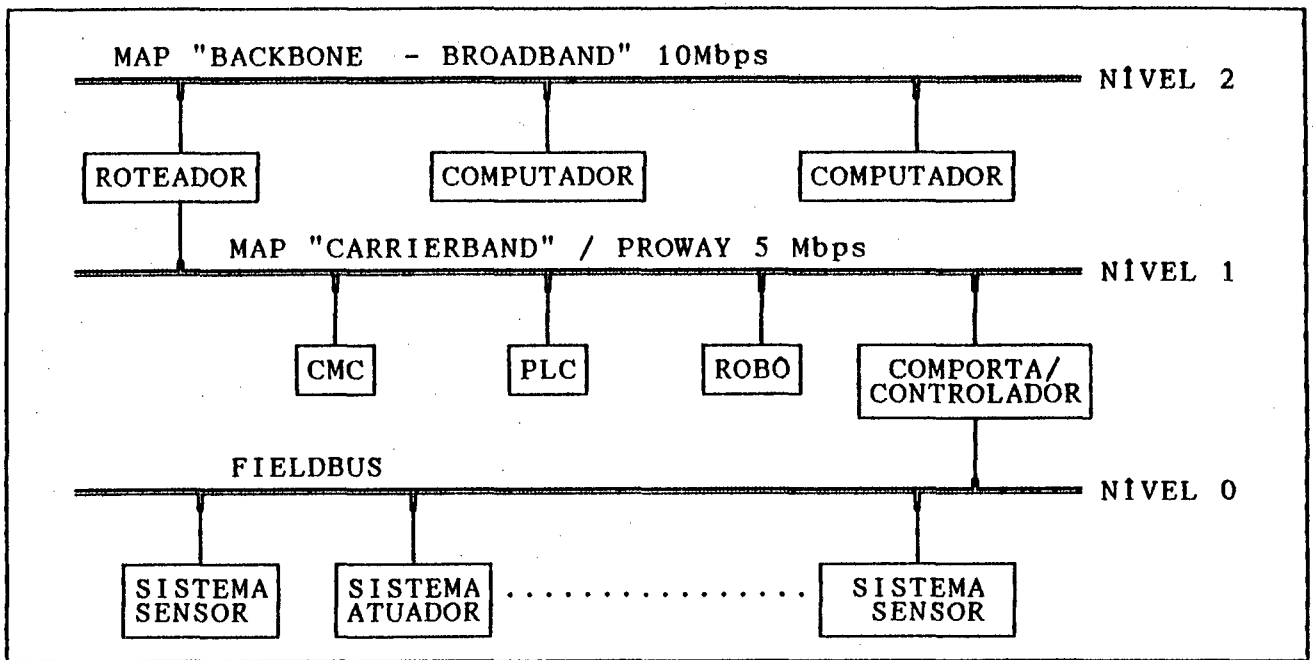


Fig. 1.6 Hierarquias de comunicação na fábrica.

planejamento a unidades de supervisão/monitoração de processo ou células na área de produção. O fluxo de dados neste nível é semelhante ao de um escritório, com um intercâmbio médio de dados intercalado com picos de demanda sem restrições de caráter temporal, e tempos de resposta em torno de 2 e 5 segundos. Particularmente a implementação da tecnologia de transmissão banda larga é suficiente para o atendimento das necessidades que se vislumbram. A utilização das redes MAP e TOP neste nível deve propiciar soluções de custo aceitável a curto e médio prazos.

- Redes de nível 1 ou redes de supervisão: Interligam principalmente unidades de supervisão/monitoração com equipamentos de automação e controle (CLP, CNC, controladores em malha fechada etc.), permitindo a carga remota de programas e a coordenação da operação destes equipamentos. Esta última função eleva as exigências tempo-real de comunicação, forçando tempos de resposta variando na faixa de 500 ms a 2 s. Neste nível deverão ganhar predominância as técnicas de transmissão de 1 canal "carrierband" 5 Mb/s, tornando os custos mais baixos. Para melhorar os tempos de resposta, foram propostas soluções de nodos compactos em dois projetos que se desenvolveram em paralelo nos últimos anos. De um lado o projeto PROWAY-C do IEC e do outro "subsets" do MAP (EPA/MAP e MINI-MAP), dentro do projeto MAP. Outras possíveis soluções constituem as redes proprietárias e "supersets"

do Fieldbus [18,19,20]. Os sistemas apresentados atendem em desempenho e preço a uma média de aplicações, com tempos de resposta entre 5 e 100 ms e mensagens mais curtas de 50 a 250 bytes.

-Redes de nível 0: Para a comunicação no nível 0, onde se requer a integração dos equipamentos mais simples e especializados do chão de fábrica, o MAP e MAP/EPA são inadequados por questões de custo e desempenho. Uma recente proposta de solução para este nível representam os Fieldbuses. Nos últimos anos surgiram vários projetos a nível de organizações nacionais, entre eles, o projeto MIL-STD-1553 desenvolvido pelas forças armadas americanas e retomado pelos ingleses (projeto ERA), o projeto Fip (Factory Instrumentation Protocol) executado na França num esforço conjunto da indústria e de várias universidades, o Bitbus lançado pela Intel em 1984 e o Profibus como um esforço combinado na Alemanha das empresas Bosch, Siemens e Klöckner Moeller e de várias universidades e instituições de pesquisa. Paralelamente a estes desenvolvimentos de projetos nacionais de Fieldbus percebe-se a consolidação de uma acentuada preocupação com padronização, através da qual se objetiva finalmente assegurar a interoperabilidade entre implementações de instrumentos inteligentes de múltiplos fabricantes. Cientes destas tendências grandes fornecedores da área (Rosemount, Foxboro, Siemens e muitos outros) se organizaram em torno de implementações candidatas à padronização internacional. Neste contexto, os projetos Fip e Profibus, pelo seu pioneirismo, contribuíram de forma decisiva para o projeto SP-50, em elaboração na ISA (Instrument Society of America), e que deve se tornar na proposta definitiva de um padrão Fieldbus na IEC (International Electrical Commission - comissão SC 65C/WG6). Assim, em termos de estado da arte, a padronização do Fieldbus é hoje um dos maiores focos de atenção na área de instrumentação a nível mundial. Uma visão geral dos esforços nacionais e internacionais de padronização de Fieldbus está ilustrada na Fig. 1.7 [21,22].

### 1.2.2 Tecnologia de sensores

Há hoje notadamente uma multiplicidade de técnicas conhecidas para a conversão de grandezas físicas em grandezas elétricas. Inicialmente a tecnologia de sensores preocupava-se essencialmente com a imple-

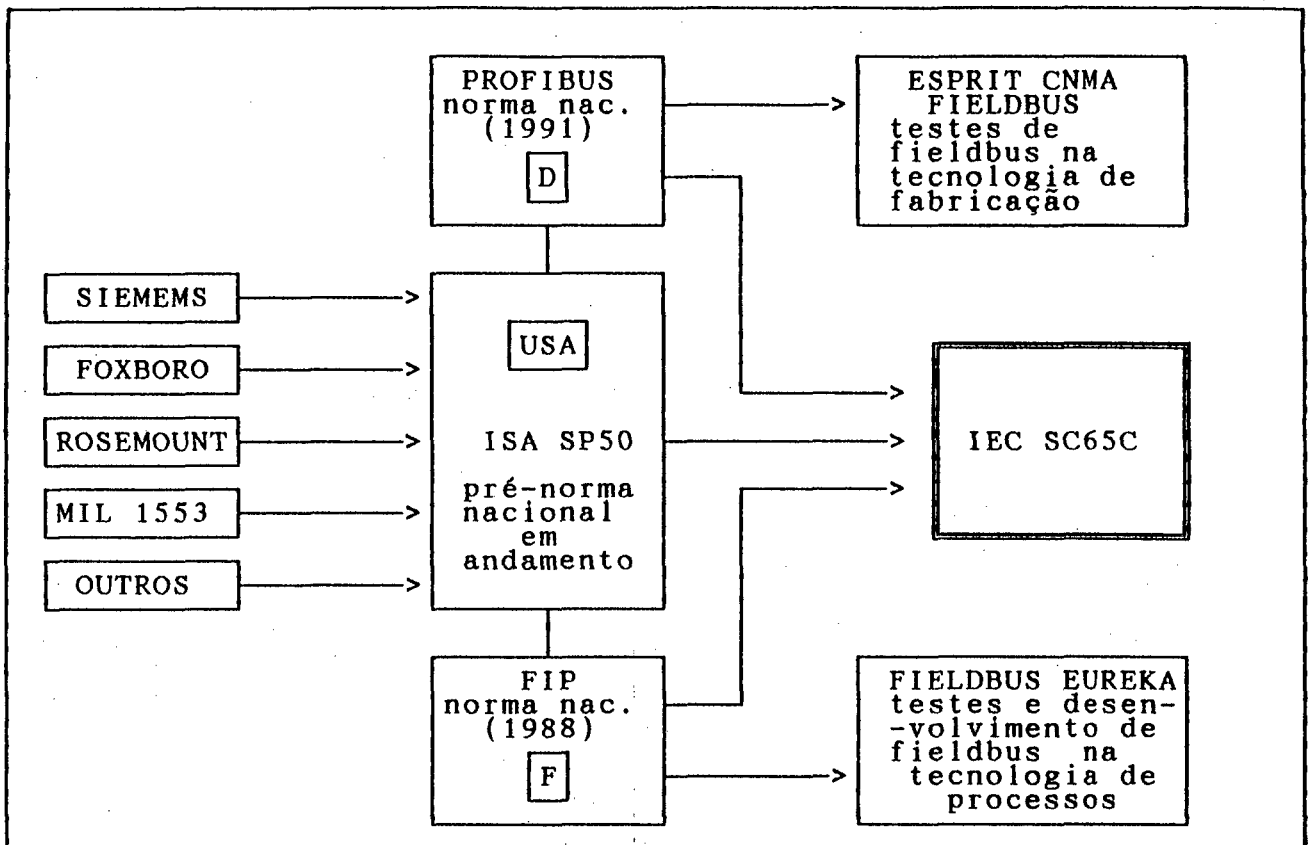


Fig. 1.7 Visão geral dos esforços de padronização do Fieldbus /Siemens/

mentação e otimização dos sensores primários. Posteriormente a maioria das soluções de sensores passaram a endereçar a usual problemática dos sensores analógicos: a obtenção de um comportamento linear ao longo da faixa de operação. Com a evolução da tecnologia digital a tecnologia de sensores foi fortemente influenciada, redundando em uma série de benefícios em termos de custo e desempenho para o sistema.

Tradicionalmente computador e sensores interligavam-se conforme ilustrado na fig. 1.8.(A). Multiplexador de sinais, amplificadores e conversores A/D estavam localizados próximos à unidade de controle e os sinais analógicos eram transmitidos remotamente desde o sensor no campo até a central. Em uma seguinte etapa evolutiva do esquema de medição os amplificadores foram deslocados para próximo do sensor com o intuito de se reduzir a sensibilidade ao ruído (fig. 1.8.(B)), e surgiram os sistemas baseados na transmissão analógica por corrente. Considerando no entanto que sistemas de porte razoável podem atingir valores de milhares de sensores, continuava a unidade de E/S

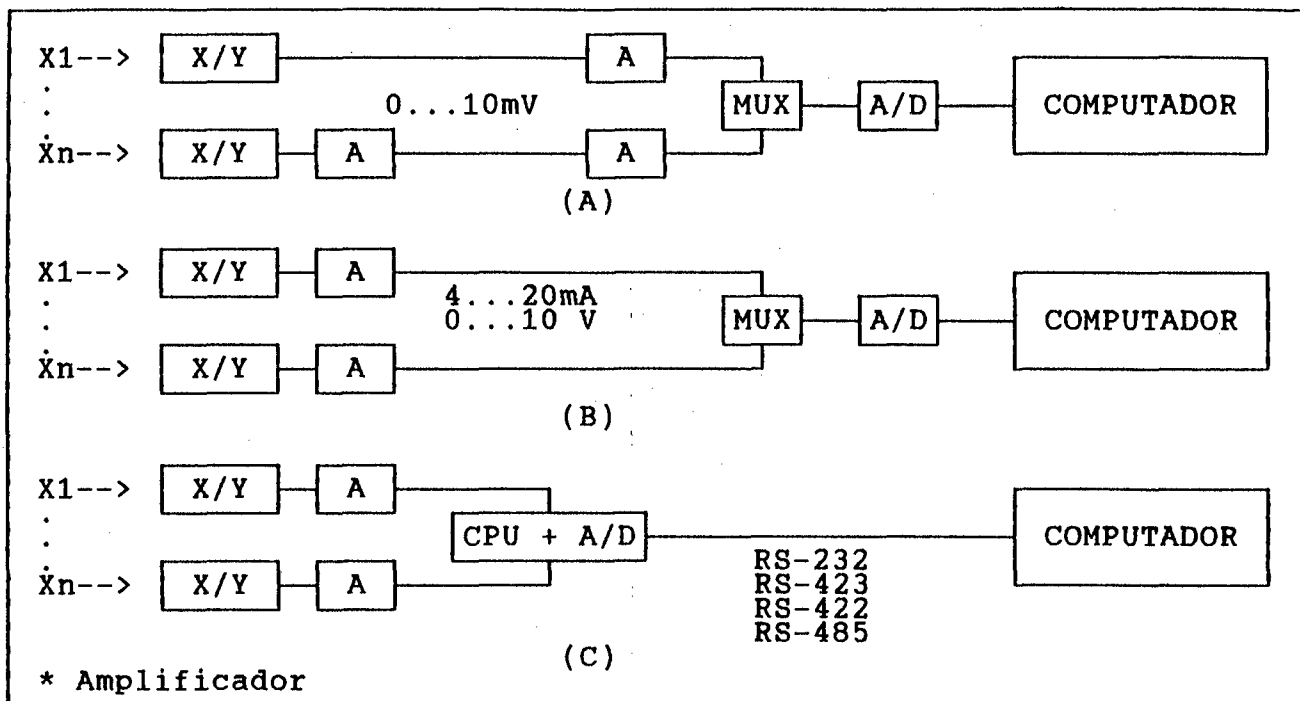


Fig. 1.8 Evolução para esquemas de medição descentralizada e digitais

do computador central um gargalo caro para a transmissão de dados de sensores, e a cablagem passava a significar um alto custo para o sistema. Adicionalmente permanecia no computador central um esforço computacional inerente às funções de processamento do sinal (correção de funções, supervisão de valores limites, e processamento estatístico dentre outros), implicando em uma degradação do desempenho como também um alto custo global do sistema. A rápida e crescente evolução das tecnologias VLSI, minimizando os custos de microprocessadores e microcontroladores, permitiu a incorporação de capacidade de processamento local junto ao sensor de campo, dando origem a uma nova geração de instrumentos de campo, denominados inteligentes. Na fig. 1.8.(C) está ilustrada a solução mais recente que envolve a digitalização do sinal no sensor, o deslocamento das funções anteriormente citadas para o campo e a multiplexagem de diversos sinais através de um meio físico comum, pelo uso de uma interface de comunicação serial digital [17].

Neste contexto, o primeiro padrão de interface a surgir foi a recomendação EIA RS-232, a qual ainda hoje é extensivamente empregada na indústria. No entanto suas desvantagens resumem-se (fig. 1.9):

-----	RS-232C	RS-423	RS-422	RS-485
modo de operação	"single-ended"	"single-ended"	diferencial	diferencial
núm. de drivers e receivers	1 driver e 1 receiver	1 driver e 10 receivers	1 driver e 10 receivers	32 drivers e 32 receivers
alcance	15 m	1200 m	1200 m	1200 m
taxa máx. de tx.	20 kb/s	100 kb/s	10 Mb/s	10 Mb/s

Fig. 1.9 Comparação sumária das especificações de interface elétrica padrão EIA.

- No baixo desempenho em termos de alcance e taxa máxima de transmissão;
- Na alta susceptibilidade relativa a ruídos industriais;
- Na possibilidade de uso de transmissão exclusivamente ponto-a-ponto.

Com o intuito de superar as limitações de alcance e velocidade foi estabelecido o padrão RS-423, o qual permite distâncias de até 1200 m (1 kbaud máx.) e taxas de até 100 Kbaud (9 m máx.). Permanecem no entanto as restrições de susceptibilidade a ruídos, as quais são somente eliminadas no padrão RS-422, através do uso de transmissão diferencial. A mais recente de todas as interfaces, padronizada pelo RS-485, atende aos mesmos requisitos do RS-422 e adicionalmente permite a conexão de até 32 "drivers" e 32 "receivers" em um barramento, permitindo a realização de um barramento efetivamente multiponto [23].

Sendo dispositivos programáveis, os sistemas sensores e atuadores inteligentes permitem uma ampla versatilidade no seu uso, viabilizando facilmente sua operação em diferentes aplicações e condições de uso. De uma forma sucinta, pode se verificar portanto no atual desenvolvimento da arquitetura dos sistemas sensores/atuadores inteligentes uma gradativa substituição dos elementos de processamento analógico remanescentes por seus equivalentes digitais (p. Ex., filtragem do sinal e linearização de curvas características dentre outros), aumento da velocidade de processamento, programação parametrizável e modular, e desenvolvimento do conceito de uma interface de comunicação apropriada e particularmente padronizada para sua in-

tegração em sistemas de produção [24,25,26,27,28].

### 1.3 Proposta de trabalho

Fundamental para a integração dos sistemas sensores e atuadores na tecnologia de produção, com o enfoque que se pretende dar neste trabalho, são os barramentos de campo ou "Fieldbuses". Pelos vários benefícios que se verifica para o sistema de produção como um todo, observa-se hoje uma intensa movimentação a nível mundial de universidades, centros de pesquisa e principalmente dos usuários e fornecedores de porte na área de instrumentação industrial, em torno do desenvolvimento e da padronização de Fieldbus, o que evidencia, por si só, a atualidade e justifica o mérito da atenção que se tem atribuído ao tema nos principais fóruns acadêmicos da área. Todos estes fatores associados à participação do Certi/Labmetro na moderna tecnologia de Fieldbus levaram de forma definitiva ao desenvolvimento do presente trabalho.

Objetiva-se realizar ao longo deste trabalho, um estudo amplo e, em tópicos que se fizer necessário, aprofundado, da problemática de integração de sistemas sensores e atuadores em sistemas de produção, apontando a virtuais usuários de instrumentação de campo os passos em direção a esta integração, tendo como ferramenta suporte de comunicação o Fieldbus.

A especificação do sistema para integração de sensores e atuadores em sistemas de produção, vista sob a perspectiva de um usuário de automação requer em primeira análise o conhecimento e a caracterização do processo de aplicação. Isto implica naturalmente no levantamento dos requisitos da planta e do processo a ser controlado. Partindo deste pressuposto o capítulo 2 objetiva estabelecer uma lista de verificação (check list) das necessidades mais comuns da maioria das aplicações de automação industrial no nível mais baixo da hierarquia dos sistemas de controle, sem a preocupação de se particularizar tais exigências para aplicações específicas. Considerando a diversidade de aplicações existentes, fica claro, portanto, que os requisitos a serem arrolados são significativos para todas elas, embora possam assumir graus variados

de importância em cada caso.

Em segundo lugar necessário se faz o conhecimento das soluções alternativas em termos tecnológicos e de mercado. O capítulo 3 classifica e descreve o elenco de configurações atualmente disponíveis para integração segundo um critério uniforme a ser definido. Os Fieldbuses correspondem a um destes itens e pelo grau de importância que assumem no âmbito do presente trabalho sua análise é realizada em maior profundidade (cap. 4) em torno das suas implicações, exemplos de soluções e arquitetura, o que servirá de subsídio para as discussões que se seguirão subseqüentemente.

Um terceiro passo em direção a especificação da integração dos sistemas sensores e atuadores que se pretende destacar neste trabalho é tratado no capítulo 5, onde se realiza um mapeamento dos requisitos da aplicação em requisitos de comunicação Fieldbus, objetivando oferecer ao usuário o suporte para a seleção de uma barramento que satisfaça adequadamente ao conjunto de necessidades mais críticas da aplicação.

O quarto e último passo para a integração diz respeito à programação da aplicação distribuída nos vários sistemas sensores/atuadores, a qual deverá servir-se do intercâmbio de informações através do Fieldbus para a realização de um determinado objetivo (monitoração, supervisão ou controle do processo), o que será tratado separadamente no capítulo 6. Busca-se adicionalmente através de um exemplo de aplicação ilustrar os conceitos e os procedimentos que compõem o projeto examinado a priori (cap. 7).

Para finalizar no capítulo 8 são tecidas as conclusões e apresentadas algumas propostas para o desenvolvimento de trabalhos subseqüentes.



## 2. REQUISITOS DAS APLICAÇÕES PARA A INTEGRAÇÃO DOS SISTEMAS SENSORES EM SISTEMAS DE PRODUÇÃO.

Há uma diversidade de fatores que afetam os requisitos das aplicações e por conseguinte a seleção da configuração do sistema de comunicação. Tais fatores compreendem as características da planta ou do processo, o estado da arte e a experiência do "staff" de manutenção, de operadores e usuários do sistema. O primeiro fator por ser predominante merece maior atenção e portanto será comentado a seguir.

### 2.1 Caracterização da planta ou processo

Em termos práticos os processos em automação industrial podem ser classificados convenientemente em tres classes distintas [29]:

#### 2.1.1 Processos contínuos

Processos na produção industrial contínuos como o refinamento de petróleo ou a produção de compostos químicos, são caracterizados por variáveis de lenta variação. Pontos de referência são ajustados, mas raramente mudam muito. Normalmente o ritmo de produção é contínuo, registrando-se poucas ocorrências de paradas temporárias. Este é o domínio dos tradicionais sistemas de controle analógico e dos sistemas de controle distribuídos baseados em microprocessadores que estão substituindo os sistemas de controle analógico.

#### 2.1.2 Processos discretos

São caracterizados principalmente pelas variáveis de estado discreto com rápida variação, freqüentemente medida na escala de ms. Os elementos de automação mais comuns são máquinas-ferramenta automatizadas, linhas de montagem e robôs industriais. Os procedimentos são trocados com freqüência e as jornadas de produção são relativamente curtas, medidas em horas ou dias. A produção de apenas uma ou poucas peças é comum, e a ocorrência de paradas tem-

porárias e partidas são freqüentes. Este é o domínio tradicional do controle baseado em relés e controladores lógicos programáveis que estão substituindo os relés.

### 2.1.3 Processos "Batch"

Ambos os processos discretos e contínuos usualmente têm, pelo menos parte do tempo, algumas características do outro. Processos da manufatura que exibem características de processos discretos e contínuos podem ser qualificados como processos "batch" [29].

## 2.2 Requisitos das aplicações ("check list")

A seguir analisa-se os requisitos mais comuns das aplicações em sistemas de produção.

### 2.2.1 Número de variáveis do processo

O número de variáveis de processo da aplicação constitui um dos mais importantes requisitos das aplicações, por definir o seu porte e conseqüentemente influenciar na escolha do tipo de interface de comunicação entre o controlador e o sistema sensor em termos de complexidade, custo e eficiência da mesma. A fig. 2.1 ilustra o número de variáveis em sistemas de controle em algumas plantas típicas em que são gerados sinais de saída e aceitos sinais de entrada [31].

APLICAÇÃO	Nº DE VARIÁVEIS DE ENTRADA		Nº DE VARIÁVEIS DE SAÍDA	
	ANALÓGICA	DIGITAL	ANALÓGICA	DIGITAL
INDÚSTRIA QUÍMICA	400	500	50	600
INDÚSTRIA DE AÇO	100	500	50	100
INDÚSTRIA DE PAPEL	40	50	5	30
GERAÇÃO DE ENERGIA	2000	6000	800	200
REFINARIA DE PETRÓ- -LEO	160	1800	---	1200

Fig. 2.1 Número de variáveis de processo para diferentes aplicações

### 2.2.2 Dispersão física das variáveis de processo na instalação de produção

Na maioria dos processos industriais as variáveis (ou os processos) não estão mais que 1000 m separados, mas há casos em que as distâncias podem ser maiores, como ilustrado na tabela da figura 2.2 [31].

APLICAÇÃO	DISPERSÃO DAS VARIÁVEIS DE PROCESSO
MINERAÇÃO	de 10 a 15 Km
AUTOMAÇÃO LABORATORIAL	no máx. 100 m
FERROVIAS/PIPELINES	centenas de kms
AUTOMAÇÃO DA MANUFATURA	1 a 25 m (máquina), 20 a 200m (célula)
AUTOMAÇÃO DE PROCESSOS CONTÍNUOS	de 50 a 1500 m

Fig. 2.2 - Dispersão das variáveis de processo (exemplo)

### 2.2.3 Comportamento dinâmico das variáveis de processo

As variáveis sob controle ou monitoração determinam fortemente o comportamento temporal da aplicação e impõem limites de desempenho sobre o sistema de comunicação.

Temperatura e umidade variam lentamente no tempo. Os valores situam-se entre 1 e 2 Hz, sendo ainda menores para umidade. Estas grandezas de processo podem ser, para efeito de controle digital do processo, amostradas a uma taxa inferior a 10 Hz, o que corresponde a um intervalo de amostragem superior a 100 ms.

Especificamente a grandeza posição é essencial para processos de fabricação. Servomotores podem ser representados no modelo matemático pelas constantes de tempo elétrica e mecânica. A constante elétrica, definida como sendo o tempo necessário para que a corrente de armadura necessita para atingir 63.2 % do valor final quando se aplica um pulso de tensão com o rotor parado, pode assumir valores de 1 ms ou mais. Já a constante mecânica, definida como sendo o tempo necessário para que o rotor atinja a velocidade de 63.2 % do seu valor final, pode apresentar valores entre 10 e 100

ms normalmente.

Torque, força, pressão, aceleração, potência e fluxo apresentam um comportamento dinâmico bastante dependente da aplicação. Este comportamento poderá ser estimado criando-se especificamente o modelo matemático do sistema e calculando-se para o mesmo as suas constantes de tempo. Pode-se dizer no entanto que em média os requisitos mais severos estão na faixa de 1 ms [30].

A tabela da figura 2.3 ilustra as características médias de alguns elementos sensores e atuadores utilizados no controle e na supervisão destas grandezas. Nesta tabela é importante observar que tais elementos são de baixo custo, geram informações de pequeno tamanho (poucos bits) e a uma elevada taxa (baixo tempo de resposta), implicando num alto volume de informação.

SENSOR/ ATUADOR	Nº DE BITS	TEMPO DE RESPOSTA	CUSTO	EXEMPLO
SENSOR ANALÓGICO RÁPIDO	12	1 ms	U\$ 50	corrente, tensão, força, rotação
SENSOR ANALÓGICO LENTO	12	1 s	U\$ 50	temperatura, fluxo
ENTRADA DIGITAL (EVENTO)	1	1 ms	U\$ 1-5	chave fim-de-curso trigger
ENTRADA DIGITAL (ESTADO)	1	20-100 ms	U\$ 1-5	identificador
ATUADOR ON/OFF	1	20 ms-1 s	-----	relés, válvulas
MOTORES	12	20 ms	U\$500	servomotores
CONTADORES	16	1 ms	U\$200	régua ótica, temporizadores, contador de rotação

Fig. 2.3 Características típicas de sensores e atuadores [32]

#### 2.2.4 Condições ambientais

O ambiente de automação de processos discretos (manufatura) é caracterizado dentre outros parâmetros pelo relativo grau de confinamento dos processos. Este confinamento aliado ao freqüente acionamento de dispositivos eletro-mecânicos tais como relés, motores elétricos, disjuntores e tiristores constituem uma fonte de interferência eletromagnética (EMI-Electromagnetic interference) que

pode interferir no desempenho do sistema de comunicação e conseqüentemente no processo controlado. A EMI interfere no processo e nos equipamentos de duas formas distintas de acoplamento: a) acoplamento por condução e b) acoplamento por irradiação. Um exemplo de acoplamento de condução é o ruído que uma fonte gera na linha de alimentação e interfere nos outros equipamentos através deste circuito. Quando a EMI é mensurável por antena, trata-se de acoplamento por irradiação. Ambas podem afetar a integridade dos dados que transitam pelo sistema de comunicação. Já o ambiente de processos tem como principal restrição para o sistema de comunicação e controle as atmosferas inflamáveis. Os parâmetros de segurança intrínseca devem assegurar que a quantidade de energia elétrica disponível em um circuito (interface de comunicação ou outro qualquer) seja muito baixa para colocar em ignição a mais inflamável mistura de ar e gás existente. A ignição pode ser causada por faíscação ou aquecimento. Com isto o projeto e a configuração do sistema deve assegurar a segurança em uso normal e sobre todas as prováveis condições de falha [33].

#### 2.2.5 Tamanho da mensagem

No nível mais próximo do processo as informações trocadas são tipicamente pequenas. O tamanho da mensagem pode ser analisado em função do tipo de interface com o processo. Para as entradas analógicas, a maioria dos conversores A/Ds possuem resolução de 8, 12 ou 16 bit. Com isto os dados não processados de sensores possuem um comprimento de no máximo 2 byte (ou seja, um número inteiro entre 0 e 65536). Através de funções de pré-processamento vários sinais de medição podem ser combinados para o uso do algoritmo de controle (p. Ex., através do processamento dos sinais do A/D em um momento torsor, número de rotações ou força), o que resulta em um número real com um conteúdo de 4 até 8 bytes. A mesma condição se verifica para as saídas analógicas, visto que os conversores D/As disponíveis implementam em média 8, 12 ou 16 bit de resolução. Os elementos de saída discreta ON/OFF tais como válvulas, relés e chaves podem ser manipulados com um bit. Por fim para entradas tipo impulso pode-se ter informações de até 4 bytes, como é o exemplo das régua óticas com até 2 m de extensão gerando 1000 pulsos por mm [30].

### 2.2.6 Necessidade de manipulação de eventos e programas

Muitas aplicações em sistemas de produção podem requerer além do acesso (leitura e escrita) a variáveis de processo remotas através de uma interface de comunicação, tarefas como a identificação da ocorrência de alarmes (eventos), a carga de programas de aplicação e parâmetros de configuração dos sistemas sensores, como também a ativação e desativação a partir do controlador da execução de programas que estão carregados nos sistemas sensores.

### 3. CLASSIFICAÇÃO DAS SOLUÇÕES PARA INTEGRAÇÃO DOS SISTEMAS SENSORES

Este capítulo primeiramente estabelece um critério para a classificação das soluções alternativas para a integração de sistemas sensores classe E em sistemas de produção, e subsequentemente realiza uma análise para cada configuração. Os Fieldbuses correspondem a uma dessas alternativas e devido à ênfase especial que lhes é atribuída neste trabalho, o exame de sua arquitetura e de exemplos é tratado em maior profundidade no capítulo 4.

#### 3.1 Critério para classificação

Para minimizar a influência na transmissão do sinal, de uma multiplicidade de fontes geradoras de ruído e sinais espúrios de tensão, tem sido extensivamente utilizado na indústria a interface de corrente 4..20 mA, pelo seu comportamento similar ao de uma fonte de corrente e pela possibilidade de se conjugar sinal e alimentação num único par de condutores. Em termos de alcance, consegue-se na prática em virtude das considerações de hostilidade do ambiente fabril, distâncias de aproximadamente 300 m [34]. No entanto a interface analógica é preponderantemente utilizada na interconexão de sistemas sensores classe A e B, razão pela qual não se estende aqui a sua análise como opção de integração.

Há notoriamente um elenco de soluções alternativas para a integração dos sistemas sensores classe E, variando cada qual enormemente quanto a estrutura e complexidade, o que evidencia a dificuldade de se encontrar um critério uniforme com o objetivo de classificação da totalidade dos casos.

Estipula-se como requisito para o critério a ser definido, que este abranja portanto um universo representativo das alternativas existentes.

O critério de classificação adotado neste contexto enfatiza uma característica funcional básica das soluções de integração, considerando-as como sistemas de processamento distribuído. Como es-

se termo é geralmente usado para descrever quase todas as situações envolvendo mais de um processador, as soluções de integração são definidas através de considerações sobre o grau de acoplamento entre os processadores (unidade controladora, sistema sensor e sistema atuador) do sistema distribuído. As soluções fortemente acopladas são definidas como sendo aquelas que interligam sistemas sensores e atuadores a uma unidade controladora fisicamente bastante próximos e acoplados em torno de um barramento de alta velocidade. Por outro lado, os sistemas fracamente acoplados caracterizam-se por vários sistemas sensores, em geral geograficamente bastante dispersos na instalação de produção, relativamente de baixa velocidade, compartilhando um suporte de comunicação organizado em forma de rede. Entre estes dois extremos estão as soluções consideradas soluções de integração medianamente acopladas, que permitem vários sistemas sensores interagirem com a unidade controladora, utilizando, por outro lado, um suporte dedicado de comunicação (fig. 3.1).

TIPO DE ACOPLAMENTO	DISTÂNCIA ENTRE UC-SS *	EXEMPLO DE SOLUÇÕES	ÂMBITO DE INTEGRAÇÃO	TIPO DE TRANSMISSÃO	MEIO FÍSICO	VELOCIDADE
FORTE	cm	VMEbus MULTIBUS	COMPUTADOR	PARALELA	BACK-PLANE	30 Mbps 40 Mbps
MÉDIO	m	IEEE-488	INSTALAÇÃO DE PRODUÇÃO OU MÁQUINA	PARALELA	CABO	4 Mbps
FRACO	Km	BITBUS FIP	INSTALAÇÃO DE PRODUÇÃO OU MÁQUINA	SERIAL	CABO FIBRA RÁDIO	375 Kbps 2.5 Mbps

Fig. 3.1 Classificação das alternativas de integração  
(\* unidade controladora - sistema sensor)

Conforme o critério estabelecido, cada classe constitui uma configuração distinta para as soluções de integração de sensores e atuadores a nível de velocidade, refletindo na prática em diferentes tipos de complexidade de interfaceamento, como ilustram os itens a seguir.

### 3.2 Configurações fortemente acopladas

O que particulariza fundamentalmente as configurações fortemente acopladas é o fato de que o sistema sensor está confinado no âmbito



da estrutura de um computador modular. A forma típica de um computador modular consiste em placas de circuito impresso que se ligam através de conectores de borda ou pinos aos soquetes de um painel de fundo ("backplane"), como ilustra a fig. 3.2.

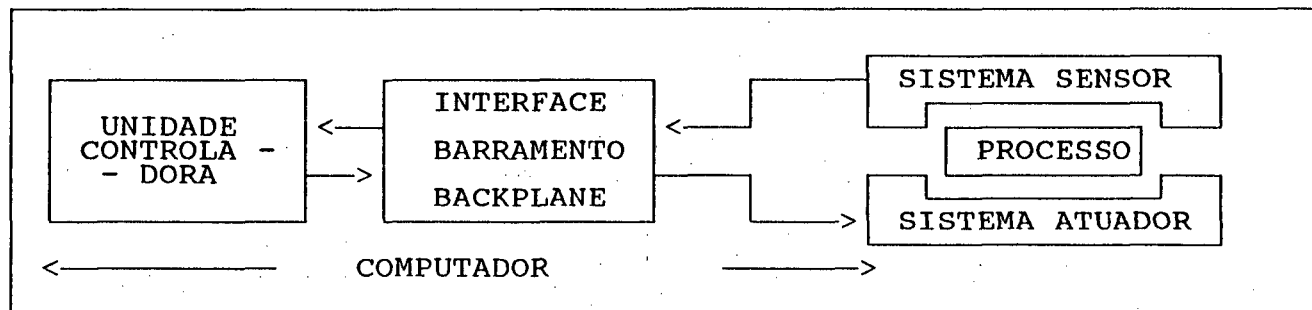


Fig. 3.2 Integração dos sistemas sensores/atuadores a nível de chassis de computador - interface "backplane" - configuração fortemente acoplada

O benefício maior resultante é bem conhecido e decorre da natureza modular do sistema em questão, possibilitando maior expansibilidade mediante a incorporação de módulos adicionais de memória e E/S, preservando-se os já existentes, à medida que crescem as exigências de capacidade de controle do processo [35].

Os barramentos de dados mais disseminados na indústria são Multibus I e II, VME e STDbus [36,37,38,39]. Devido ao elenco de tipos em vigor e por motivos de brevidade, a exposição que segue limitar-se-á à discussão sumária de uma opção apenas a título de ilustração.

### 3.2.1 Multibus I

Lançado pela Intel em 1976 e devidamente padronizado pelo projeto IEEE-796, obteve ampla aceitação na prática industrial [40]. Consta de 4 tipos distintos de sub-barramentos:

- Barramento de Sistema Multibus (ou barramento principal/primário), é utilizado nas transferências entre unidade de processamento e demais módulos conectados ao "backplane". Dispõe de 20 linhas de endereços, 16 de dados e 18 de controle. O conector é do tipo borda com um total de 86 pinos;

- Barramento de Extensão LBX (Local Bus Extension), tem o propósito de incrementar o desempenho do sistema estendendo o barramento local para acesso de recursos de memória externos ao cartão. Disposto em um conector de borda, apresenta 16 bits de dados, 24 de endereços e 8 linhas de controle. O barramento suporta transferências de até 19 Mbytes/seg. sem afetar o barramento do sistema primário;

- Barramento de Expansão SBX, o qual objetiva prover uma solução de baixo custo para a expansão da capacidade de E/S, aliviando o barramento de sistema Multibus para outras atividades. O conector está alojado na placa, como qualquer outro componente, permitindo o acoplamento de sistemas sensores/atuadores de tamanho reduzido. Padronizado pelo IEEE 959, constitui-se de 16 linhas de dados, 3 de endereços além de linhas de leitura/escrita em E/S e seleção de circuito;

- Barramento de E/S Multicanal (Multichannel I/O Bus), com a função de dispor um meio a parte do cartão CPU para transferências de DMA. Há um pino de 60 conectores alojado no centro da borda superior do cartão Multibus que prove 16 linhas de dados/endereços multiplexados e 14 linhas de controle. Este barramento alivia a atividade de E/S da CPU hospedeira e reduz a contenção entre E/S e atividades de processamento de dados no barramento de sistema. O barramento de E/S multicanal suporta até 16 dispositivos em até 8 Mbytes/s. Os dispositivos no barramento são classificados como supervisores, controladores e ouvintes de forma similar ao conceito IEEE-488 (GPIB). O tamanho do cartão Multibus é 15,24 x 30,48 cm o que permite 464,5 cm<sup>2</sup> de área para a disposição física dos componentes.

### 3.3 Configurações medianamente acopladas

Figuram neste nível as configurações que permitem um grau de distribuição das unidades de E/S na ordem de dezenas ou até centenas de metros, e velocidades em torno de 4 Mbps, a exemplo da interface paralela. Contrariamente ao barramento paralelo em "backplane", não constitui a transmissão paralela por cabo um canal interno para uso na comunicação entre unidade controladora e seus módulos sistemas

sensores, mas sim uma opção para interconexão externa de instrumentos à unidade controladora, formando um sistema. O exemplo mais notável é a interface IEEE-488, originalmente desenvolvida pela HP (HPIB). Os dispositivos conectados ao barramento enquadram-se em três categorias: locutores, que produzem e enviam mensagens de dados a um ou mais ouvintes encarregados de recebê-las e controladores, os quais gerenciam o fluxo de informação no barramento enviando comandos a todos os dispositivos [41].

O barramento em si consiste de 8 linhas de dados, 3 de controle de transferência e 5 para controle geral. Destaca-se para o fato de que o controlador é munido de um módulo de interface paralela acoplada a sua estrutura de barramento, ou seja, implicitamente na interconexão do controlador com o sistema sensor além da interface paralela há uma interface backplane interligando o controlador e o módulo interface de comunicação paralela.

A interface paralela apresenta, comparativamente à ligação serial (item 3.4), alta taxa de transferência de dados visto que o protocolo empregado utiliza o conceito de "handshaking", dispondo de linhas dedicadas ao controle de fluxo de informação, e ocorre notadamente a transferência de vários (em geral 8) bits simultaneamente a cada transmissão. Em contrapartida provê pequena capacidade quanto ao alcance e número máximo de dispositivos conectáveis ao barramento, maiores custos associados à aquisição e manutenção da cablagem. Embora constitua uma opção viável para a integração de sistemas sensores classe E, a interface paralela (fig.3.3), pelas limitações

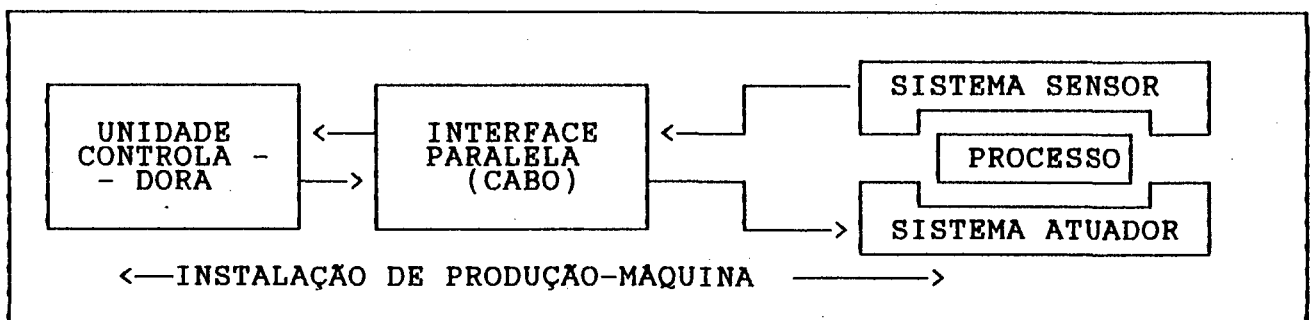


Fig. 3.3 - Integração de sistemas sensores/atuadores a nível de instalação de produção/máquina - interface paralela (cabo) - configurações medianamente acopladas

supra-citadas tem sido utilizada na interconexão apenas de instrumentos tais como voltímetros digitais, geradores de sinal, relés atuadores, contadores eletrônicos dentre outros, enfocando preferencialmente as aplicações laboratoriais em sistemas de medição e teste automatizado.

### 3.4 Configurações fracamente acopladas

Estão aqui representadas as configurações onde unidade controladora interage com o sistema sensor virtualmente através de uma rede local industrial dedicada, permitindo velocidades da ordem de 1 Mbps e distâncias superiores comparativamente ao barramento "backplane" (em geral quilômetros de distância para velocidades menores que 1 Mbps), bem como a interconexão de um maior número de dispositivos (fig. 3.4) [42,43,44,...52].

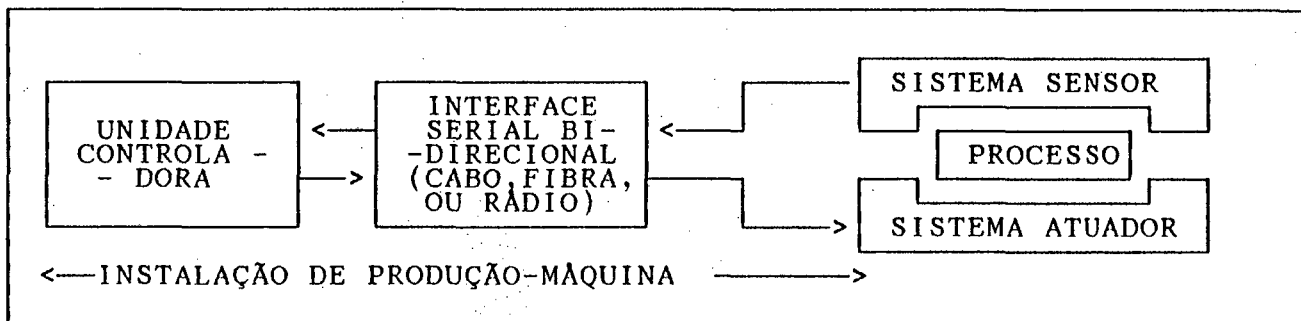


Fig. 3.4- Integração de sistemas sensores/atuadores a nível de instalação de produção/máquina - interface serial -Configurações fracamente acopladas

A interface de comunicação fracamente acoplada para a integração dos sistemas sensores classe E é realizada pelo Fieldbus, cujo fundamento geral será analisado no capítulo seguinte.

#### 4. A INTERFACE FIELDBUS (BARRAMENTOS DE CAMPO)

Com o intuito de se embasar o entendimento das discussões subseqüentes em torno da integração dos sistemas sensores, este capítulo examina preliminarmente o Fieldbus quanto ao seu objetivo, implicações e aspectos relacionados a sua arquitetura. Adicionalmente destaca-se alguns exemplos de soluções que permitem uma visão geral das opções de mercado a serem consideradas quando da especificação de um barramento.

##### 4.1 Caracterização

O Fieldbus (ou barramento de campo) consiste basicamente numa interconexão serial digital, associada aos protocolos de comunicação, especificamente otimizada para a integração de pequenos dispositivos de campo tais como sistemas sensores, atuadores e controladores (single ou multi-loop) no nível mais baixo dos sistemas de controle de processos e automação da manufatura. O Fieldbus visa deste modo a substituição das ligações tradicionais ponto-a-ponto, e dos módulos de E/S dos controladores, na interligação dos controladores com os dispositivos de campo (fig. 4.1).

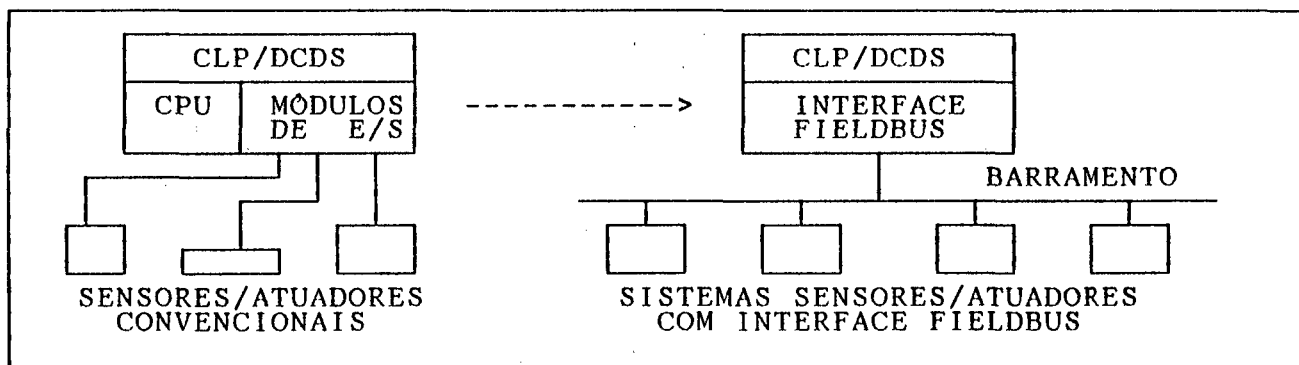


Fig. 4.1 Função do Fieldbus

##### 4.2 Implicações

Tendo como característica de comunicação a transmissão de mensagens curtas, tempos de resposta reduzidos, boa confiabilidade, segurança

e baixo custo, apresenta benefícios em relação às ligações convencionais analógicas que podem assim ser resumidos a nível econômico e técnico :

#### 4.2.1 Vantagens a nível econômico

- A redução do custo pode ser obtida pela substituição do sistema de cablagem por um meio físico compartilhado (fig. 4.1), e pela utilização de uma arquitetura de rede (item 4.3) relativamente simples. Também a substituição de todo o sistema de E/S do controlador por uma interface Fieldbus, aliada à redução da complexidade do projeto da instalação, contribuem favoravelmente neste sentido.

#### 4.2.2 Vantagens a nível técnico

- Descentralização resultante da migração de funções de pré-processamento para o campo, aliviando assim a carga de processamento do controlador principal e portanto permitindo melhor desempenho global da aplicação;

- Facilidade de instalação e manutenção do sistema pela maior modularidade resultante;

- Potencial compatibilidade com redes de nível hierárquico superior (p. Ex., MAP/TOP);

- Aumento da consistência, confiabilidade e alcance da comunicação pelo uso de transmissão serial, digital e bidirecional, aliada à incorporação a nível de protocolos de mecanismos de identificação e recuperação de falhas inerentes ao hardware de comunicação ou erros causados através do suporte de transmissão;

- Maior expandibilidade devido a natureza modular da rede, permitindo a medida que se torna necessário, o crescimento gradual e seguro do nível de complexidade da configuração.

Como causa imediata da utilização de um meio físico no qual se multiplexa os fluxos de informação de diversos dispositivos independentes, resultam limitações a nível sistêmico na adoção do Fieldbus, particularmente no que se refere a confiabilidade e restrições tempo-real. A confiabilidade e disponibilidade podem no entanto ser substancialmente melhoradas através de redundâncias de meio físico e estações (fig. 4.2).

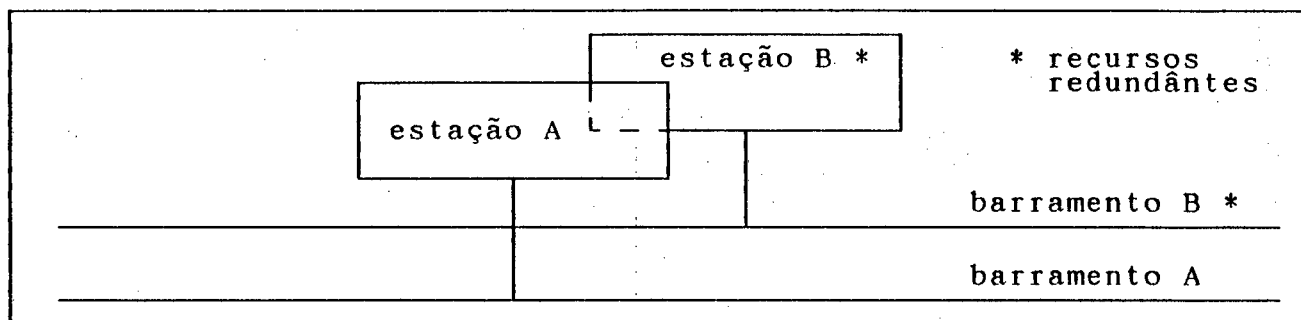


Fig. 4.2 Emprego de redundância para aumento da disponibilidade e confiabilidade do Fieldbus.

Já as restrições temporais compõem uma problemática mais complexa, envolvendo uma combinação de fatores tais como mecanismo de controle de acesso ao meio físico, taxas de transmissão e outros fatores adicionais a serem tratados no capítulo 5.

### 4.3 Arquitetura de Fieldbus

A idéia fundamental do Modelo de Referência OSI também é aplicável ao Fieldbus [53,54]. Para reduzir a complexidade de projeto, os sistemas de comunicação estão organizados em níveis ou camadas. O conceito de projeto modular por camadas é conhecido amplamente na prática de engenharia de software. Funções semelhantes são agrupadas na mesma camada, comunicando-se cada camada N com as duas vizinhas, N-1 e N+1, através de interfaces. A camada N oferece serviços para a camada imediatamente superior N+1 e utiliza para isto os serviços da camada inferior N-1. Os serviços estão disponíveis em pontos de acesso ao serviço (SAPs- Service Access Points). Os SAPs da camada N são os locais onde a camada N+1 pode acessar os serviços oferecidos. Cada SAP tem um endereço que o

identifica inequivocamente. Cada camada comunica-se com a correspondente na estação remota obedecendo a regras e convenções coletivamente denominadas de protocolo da camada N. Ao conjunto de camadas e protocolos de um barramento de campo denomina-se de arquitetura de Fieldbus. O intercâmbio de dados se dá verticalmente através da arquitetura na forma de Unidade de Dados de Protocolos (PDUs - Protocol Data Units). Ao conjunto de dados trocados nas interfaces locais vizinhas de cada camada denomina-se Unidade de Dados de Serviço (SDUs - Service Data Units). O SDU conterá o eventual PDU a ser transmitido, mas outras informações de interface poderão ser repassadas (p. ex. informação de gerenciamento de camadas, caso não existam PDU's superiores). As primitivas de serviço são comandos dirigidos, do usuário para o fornecedor do serviço ou deste para o usuário. A primitiva Request é utilizada por um usuário de serviço para requerer um determinado serviço de comunicação ao provedor. O provedor remoto de serviço deverá indicar ao usuário destino a solicitação de serviço feita pelo usuário origem através da primitiva de Indication. Esta pode ser respondida pelo usuário-destino pela primitiva Response. O provedor local confirma ao usuário origem a realização do serviço por ele antes solicitado (através da primitiva Request) com a primitiva Confirm. Todos os serviços de comunicação em Fieldbus compõe-se de seqüências de primitivas de serviço (fig 4.3).

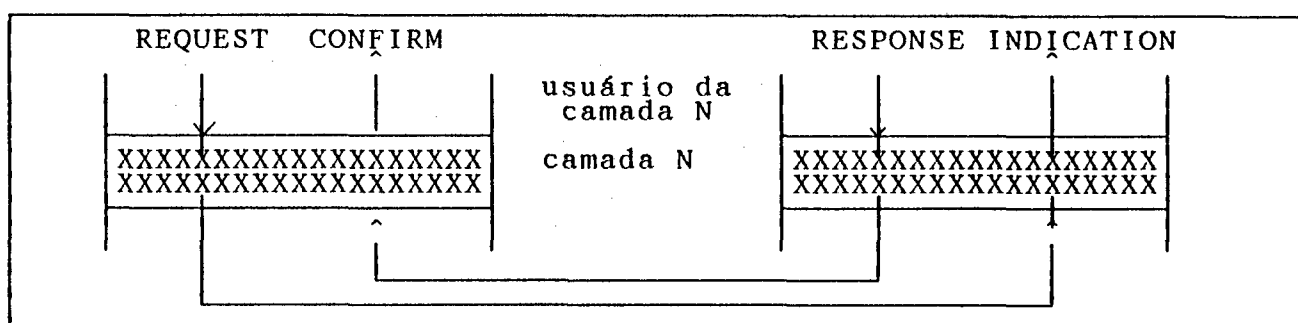


Fig. 4.3 - Primitivas de Serviço do modelo de camadas

Por questões de eficiência adota-se para o Fieldbus apenas as camadas 1, 2 e 7 do RM-OSI, a serem examinadas a seguir, sendo que a funcionalidade necessária das camadas 3 a 6 é distribuída entre as camadas remanescentes 2 e 7.



#### 4.3.1 A Camada física

A camada física ( nível 1) define as características mecânicas, elétricas, funcionais e procedimentos para a transmissão de bits entre estações. Esta é a camada de nível mais baixo na arquitetura OSI, e está diretamente associada ao meio físico interconectando os sistemas. A camada física ocupa-se, por exemplo, da regulamentação das interfaces mecânicas (número de pinos dos conectores, dimensões, etc..) e das interfaces elétricas (níveis de tensão, corrente, etc..). Os principais parâmetros da camada física discutidos a seguir são:

##### a) meio de transmissão

O meio de transmissão é aqui entendido como o conjunto de recursos físicos e de regras lógicas que permitem a transmissão entre os vários nós da rede. Além do suporte físico de transmissão ligando os diversos nós de comunicação, estão associados a topologia da rede e os elementos responsáveis pela implementação dos protocolos de comunicação ao nível de bit. Dentre estes últimos podem ser destacados o tipo de codificação do sinal e o modo de transmissão. A especificação do suporte de transmissão não é a rigor objeto desta camada, muito embora seja frequentemente incluída nesta.

##### a.1) topologia

A topologia de um Fieldbus corresponde à estrutura de interconexão física das várias estações que a compõem. Em princípio existem várias maneiras de se configurar a interconexão das estações, cada qual apresentando diferentes implicações quanto ao desenvolvimento, operação e manutenção da rede.

A fig. 4.4 destaca tres possibilidades de topologias de Fieldbus. Na topologia em barramento, a mais freqüente em sistemas Fieldbus, o meio físico de transmissão é composto por um único segmento de transmissão multiponto, compartilhado pelas várias estações

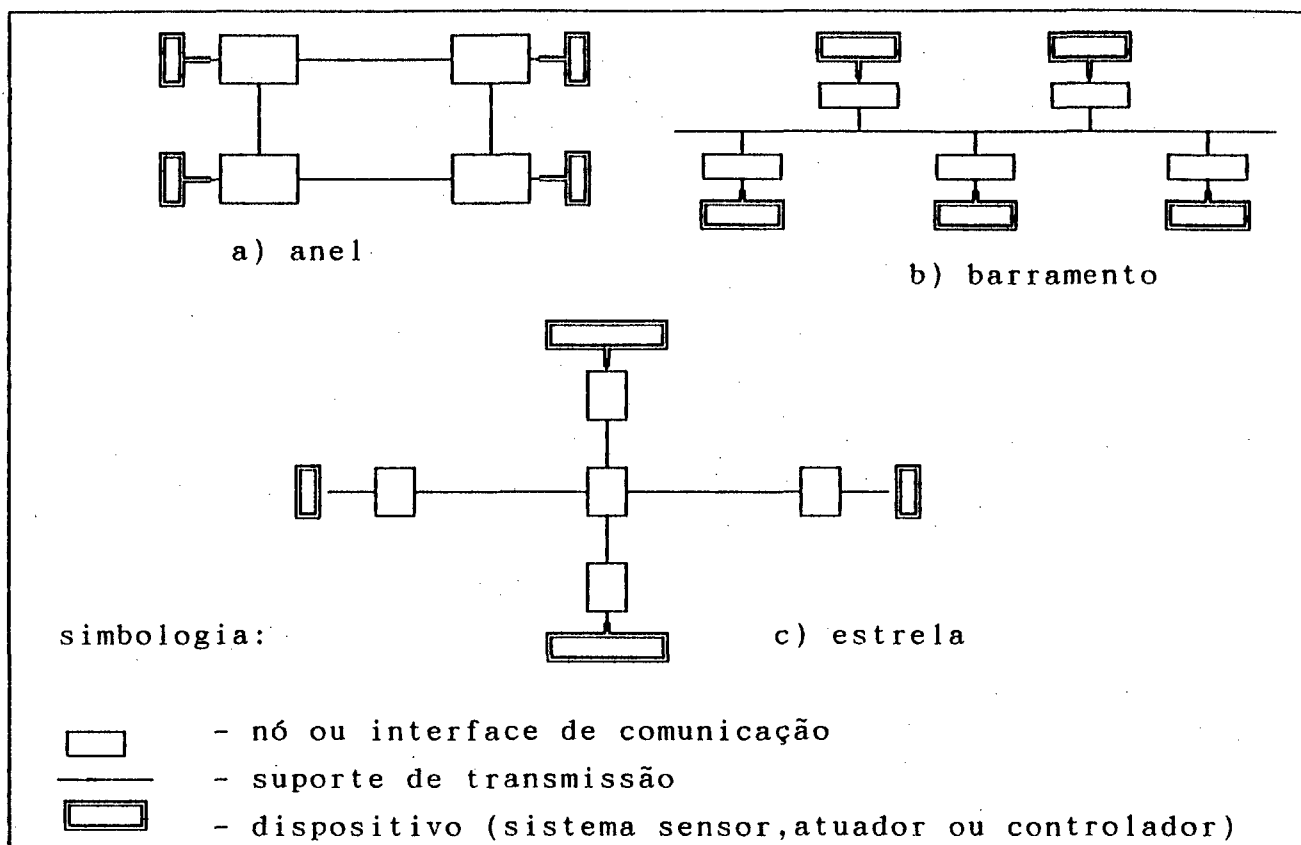


Fig. 4.4 Topologias básicas de Fieldbus

conectadas. A topologia em anel tem sua aplicação muito mais restrita em relação ao barramento. Na estrutura em anel a ligação entre as estações é do tipo ponto-a-ponto, ou seja, o meio físico de transmissão é constituído por vários segmentos de transmissão entre pares de nós de comunicação. Na transmissão da informação entre duas estações não adjacentes ocorre a participação dos nós intermediários, os quais funcionam como repetidores da mensagem até o seu destino. Outra possibilidade de estrutura de interconexão é a topologia em estrela, representada na fig. 4.4.c. Nela todas as estações conectam-se a um nó central (ligação ponto-a-ponto), responsável pelo roteamento das mensagens que são transmitidas entre duas estações quaisquer. Dentre as tres alternativas citadas, a estrela é a menos frequentemente encontrada em aplicações de Fieldbus pois além de colocar um problema de confiabilidade (vulnerabilidade da rede), a complexidade do nó de comunicação central aumenta consideravelmente com o número de nós que lhe são interconectados. Além disso, um grande número de estações implica um grande número de ligações físicas, o que pode passar a comprometer

os custos associados aos meios físicos de transmissão (instalação, aquisição etc..).

Na topologia de um Fieldbus podem ser encontrados conectados ao barramento além dos próprios dispositivos de campo, os chamados sistemas intermediários (SI), classificados em:

- Repeaters: Usados em geral em um Fieldbus para a interligação de vários dos seus segmentos, tornando possível uma maior cobertura geográfica pela mesma rede. As suas funções estão limitadas à camada 1 do Modelo OSI, pressupondo-se que todos os protocolos sejam idênticos nos segmentos interligados;

- Bridges: São dispositivos que interligam dois ou mais Fielbuses a nível de camada de enlace;

- Gateways: São dispositivos destinados à interconexão do Fieldbus a uma rede distinta. Em particular a necessária transmissão entre o Fieldbus e o MAP é resolvida com este último tipo de sistema intermediário, o qual pode ser implementado em um controlador de célula, a título de exemplo, com duas placas de comunicação, uma para a interface Fieldbus e outra para a interface MAP (fig. 4.5). A informação é convertida através de um software conversor de protocolos de um formato para outro. Em geral o custo de um Gateway (ou comporta) é determinado pela semelhança entre as camadas de aplicação. Um problema central para a comporta é o gerenciamento funcional e também temporal de ambos os lados, do que resultam custos relativamente altos.

#### a.2) Tipo de codificação do sinal

A codificação do sinal elétrico para transmissão em longas distâncias pode ser em banda básica ou banda larga [55]. Na transmissão banda larga os dados são modulados em frequências portadoras. Vários tipo de informações (p. ex. dados, voz, vídeo) podem ser simultaneamente transmitidos cada qual através de uma frequência portadora. Contrariamente a transmissão em banda larga, na transmissão em banda base os dados binários são diretamente

transmitidos pulsando-se o cabo com corrente ou tensão, ou seja, sem modulação, sendo que apenas um tipo de informação pode ser

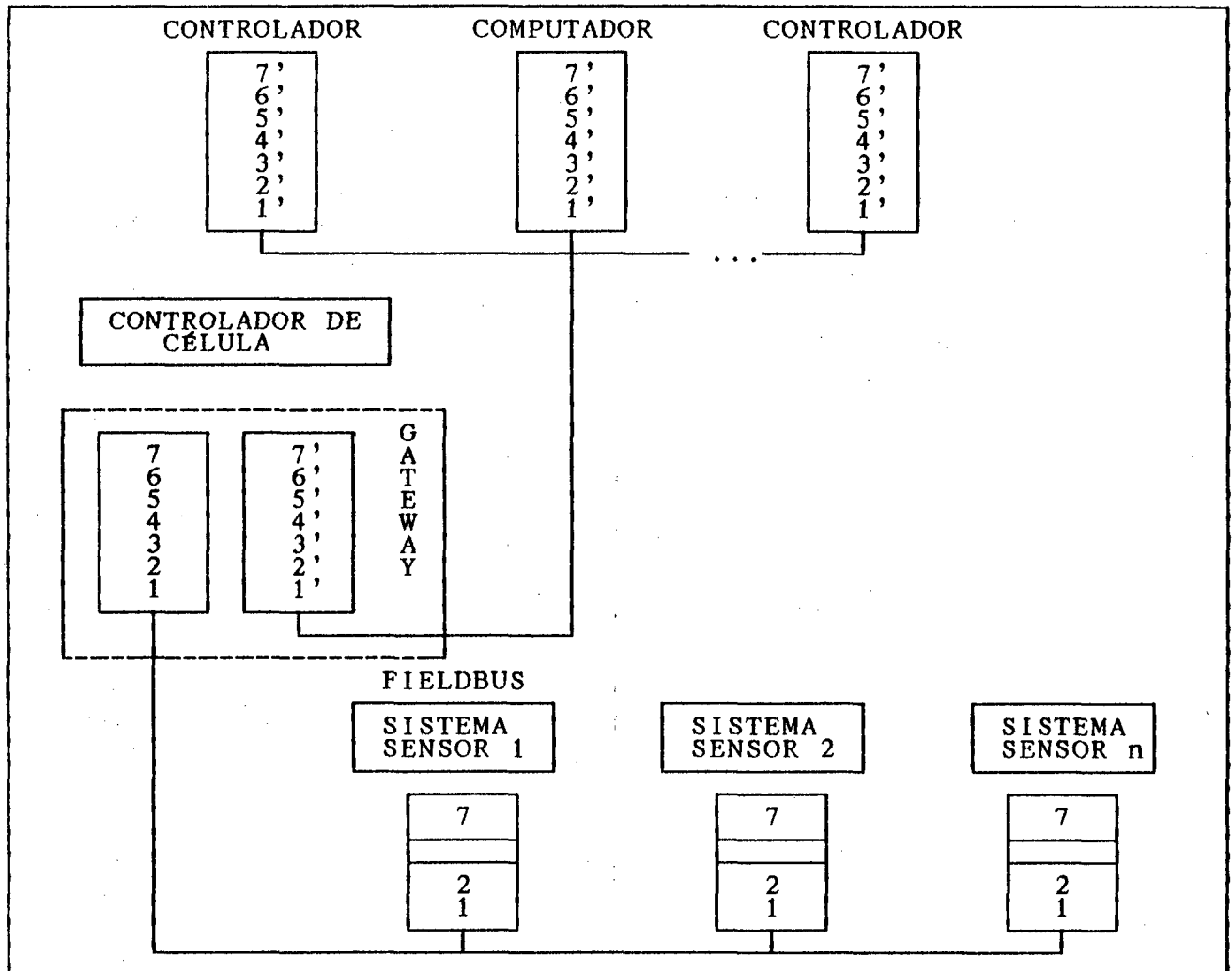


Fig. 4.5 Implementação de uma comporta MAP-Fieldbus em controlador de célula

transmitida pelo canal de comunicação em um dado instante. Exemplos de sistemas de codificação em banda base são (fig. 4.6):

- NRZ (Non-Return-to-Zero), o qual representa o 1 lógico por um pulso que permanece durante todo o intervalo de bit, e o 0 lógico é representado pela manutenção do sinal no nível de referência;

- RZ (Return-to-Zero), o qual representa o 1 lógico por um pulso que permanece durante metade do intervalo de bit, depois do que o sinal retorna para o nível de referência (0 V) durante a outra metade do intervalo de bit. O zero lógico é representado neste método pela

ausência de sinal, ficando o mesmo no nível de referência;

- NRZI (Non-Return-to-Zero-Inverted), em que um zero lógico é representado por uma mudança de polaridade do sinal e o 1 lógico é representado pela ausência de mudança na polaridade do sinal.

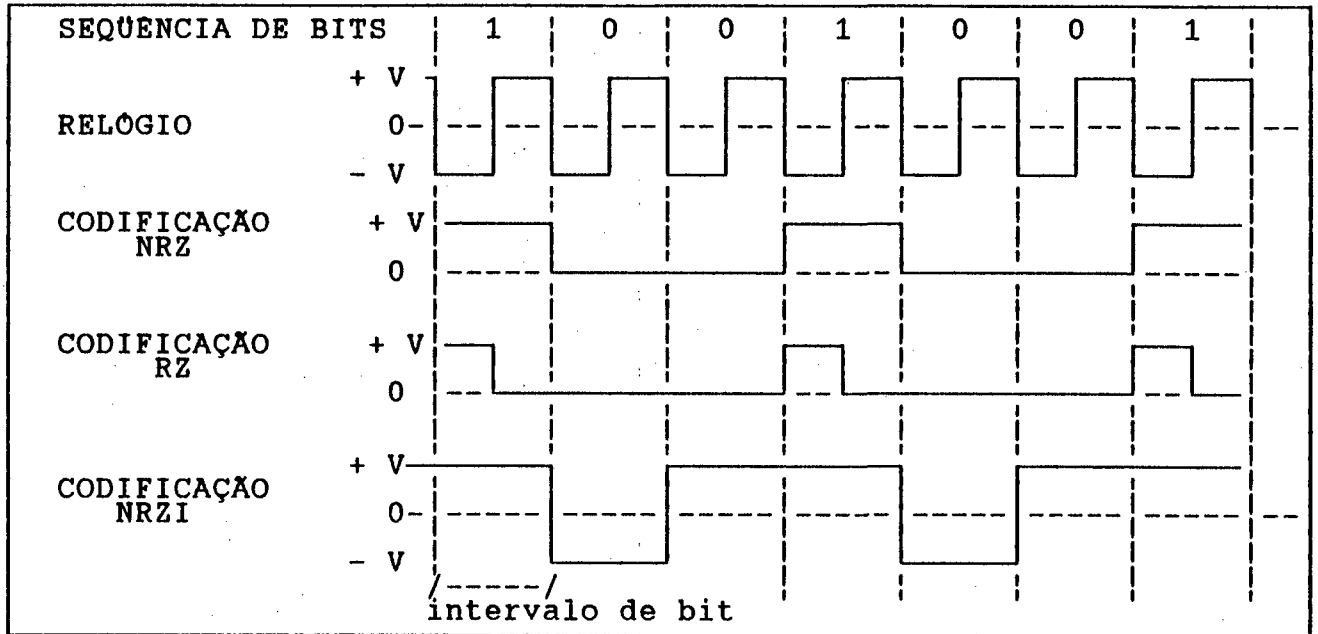


Fig. 4.6 Tipos de codificação em banda básica (exemplos)

### a.3) Modo de transmissão

Pode-se identificar os modos de transmissão síncrono e assíncrono pela maneira de se utilizar a variável tempo para separar os bits em uma transmissão. Na transmissão assíncrona, utiliza-se um elemento de sinalização para indicar o início do caractere (START) e outro para indicar o término do caractere (STOP). O START (bit de partida) corresponde a uma interrupção do sinal na linha e o STOP (bit de parada), à condição de marca ou repouso, ou seja, à existência do sinal na linha (normalmente o STOP corresponde a 1,4 ou 2,0 vezes o tempo de START), conforme ilustrado na figura 4.7.

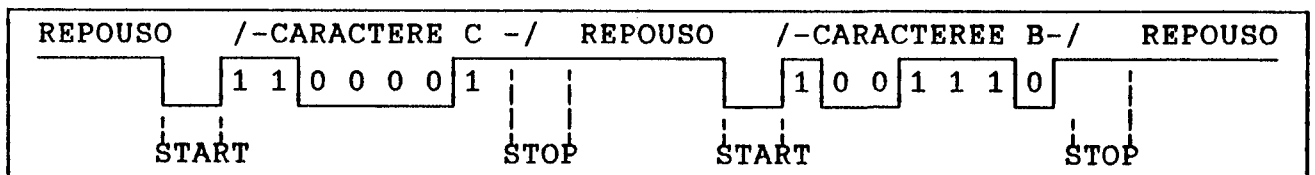


Fig. 4.7 Transmissão serial assíncrona

Pelo bit START, o receptor será avisado da transmissão de um caractere com antecedência suficiente para que possa, através de seu próprio relógio, sincronizar seus circuitos elétricos para ler cada bit no momento apropriado. O termo "Assíncrono" refere-se à irregularidade dos instantes de ocorrência dos caracteres, ou seja, o tempo decorrido entre dois caracteres (tempo de repouso) pode ser variado pelo transmissor sem que o receptor tome conhecimento. A transmissão assíncrona, apesar da emissão dos caracteres ser irregular, possui um sincronismo ao nível dos bits que compõem o caractere (obtido pela identificação do START), pois o equipamento receptor deve necessariamente conhecer os instantes que separam os bits dentro do caractere.

Na transmissão síncrona, os bits de caractere são enviados imediatamente após o anterior, não existindo START-STOP e tempo de repouso entre eles. A transmissão síncrona é estabelecida através de uma cadência para a transmissão dos bits de todo um conjunto de caracteres (bloco). Antes da transmissão de um bloco, o transmissor envia um preâmbulo que consiste numa configuração de bits de sincronização com o objetivo de colocar o equipamento receptor exatamente em fase com o mesmo. Esta configuração de bits de sincronização necessariamente deverá ser diferente de qualquer configuração de bits que possa ser enviada no bloco da mensagem. O protocolo BSC (Binary Synchronous Communication), exemplificado na fig. 4.8, mostra a presença de caracteres de sincronização, que permitirá a transmissão síncrona entre dois equipamentos.

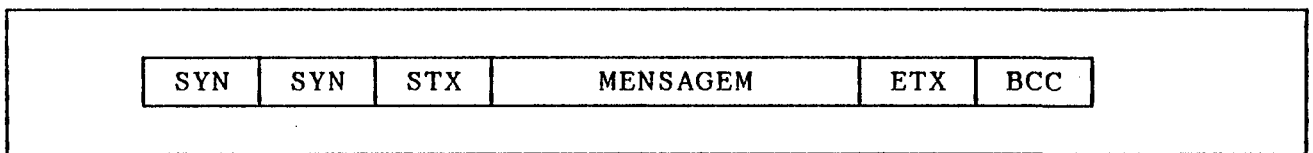


Fig. 4.8 Transmissão serial síncrona.

Os métodos de sincronização permitem se obter, no lado do receptor, uma onda de relógio sincronizada com o sinal transmitido. Pode-se em princípio, recorrer a um dos seguintes métodos de sincronização:

- utilização de um único oscilador centralizado, o qual fornece uma onda de relógio que serve de referência comum para transmissor e receptor (fig. 4.9.a);

- derivação do sinal de relógio no receptor, a partir do sinal recebido (fig. 4.9.b);
- emprego de dois osciladores de alta estabilidade, operando na mesma frequência (fig. 4.9.c).

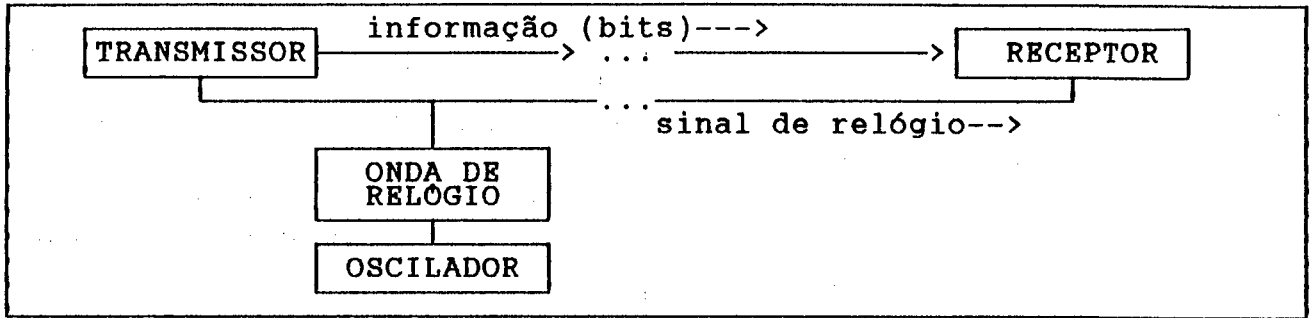


Fig. 4.9 a) Sincronização com um oscilador centralizado e suporte de transmissão paralelo

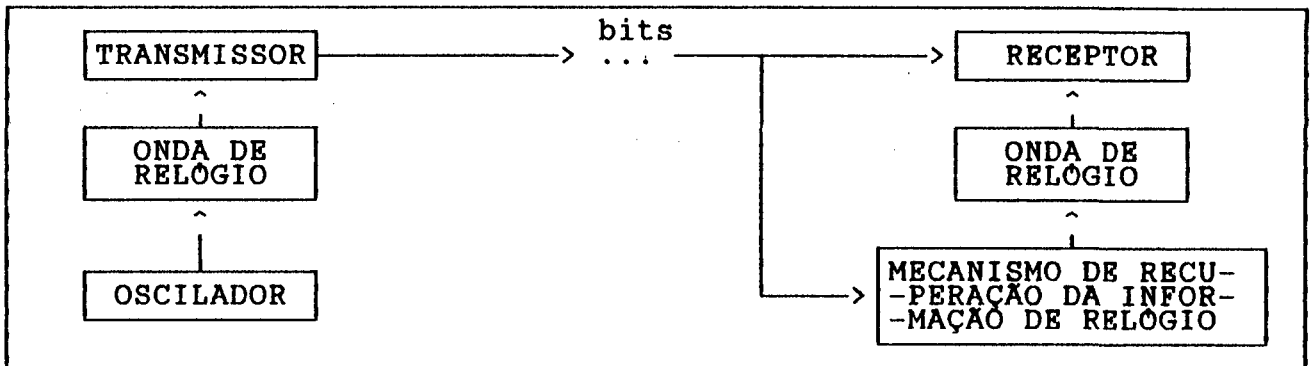


Fig. 4.9 b) Sincronização com derivação da onda de relógio a partir do sinal recebido

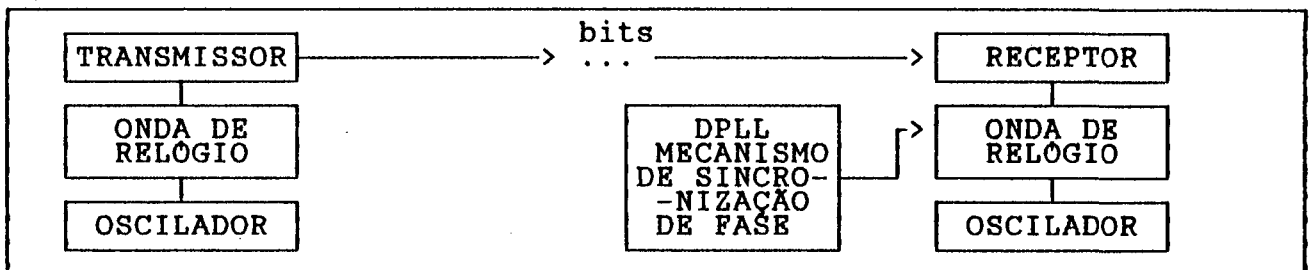


Fig. 4.9 c) Sincronização com dois osciladores independentes

No primeiro caso o padrão primário de relógio é transportado entre transmissor e receptor através de um suporte físico separado. Portanto este sistema não se adapta muito bem aos barramentos de maior

alcance, ou de alta velocidade de transmissão, por não levar em conta os diferentes tempos de propagação do sinal e das ondas de relógio transmitidas.

No segundo método utiliza-se um mesmo suporte físico para transportar a informação e o sinal de relógio, cabendo ao receptor recuperar o sinal de relógio a partir da informação recebida. Utiliza-se com este objetivo, por exemplo, a codificação RZ e NRZ.

O terceiro método baseia-se na utilização de um oscilador altamente estável, no receptor, que é resincronizado com o relógio do transmissor durante a recepção da informação através de um DPLL (digital phase-locked loop). Este método requer para a sua efetividade o uso da codificação NRZI.

#### a.4) Taxa de transmissão

A taxa de sinalização é a taxa de transmissão nominal do Fieldbus em bits por segundo (bps). Contudo o usuário está interessado no número de bit de informação que são transferidos por unidade de tempo. A taxa de transferência é inferior à taxa de sinalização pelo fato de que variáveis de comunicação são transmitidas em conjunto com os dados de usuário para o controle da comunicação, o que ocorre de forma transparente para a tarefa de aplicação. Portanto o usuário deverá especificar uma taxa de transmissão, mas a real taxa de transferência será substancialmente inferior. Aumentando-se a taxa de transmissão aumenta-se proporcionalmente a taxa de transferência, mas a partir de um valor limite o qual depende exclusivamente da implementação específica de camada física, o aumento desta é usualmente inferior que o aumento da taxa de transmissão devido ao aumento dos erros associados ao meio físico. O transmissor poderá, por exemplo, ter transmitido 30 bytes em uma seqüência de transmissão constituída de 3 blocos cada qual com 10 bytes. Se por exemplo apenas o último bloco for aceito pelo receptor, e os demais forem corrompidos pelo meio de transmissão devido às altas taxas de transmissão, a taxa de transferência será, como é de se esperar, 1/3 da taxa de transmissão.



### 4.3.2 A Camada de Enlace

A camada de Enlace de Dados (nível 2) tem por objetivo principal detectar e, possivelmente corrigir os erros que podem ocorrer nas trocas de informações (bits) a nível de camada física. Exemplos de serviços oferecidos pela camada de enlace são o SDA (Send Data with Acknowledge-envio de dados com confirmação de recepção) e o SDN (Send data with no Acknowledge-envio de dados sem confirmação de recepção). Os parâmetros de maior interesse para o usuário a nível de enlace são o controle de acesso ao meio de transmissão e a distância de Hamming, a serem analisados a seguir.

#### a) Mecanismo de controle de acesso ao meio

Devido ao fato de que em um barramento várias estações compartilham o uso do meio físico, torna-se imprescindível que o protocolo de enlace disponha de mecanismos que assegurem o acesso ordenado das estações ao meio de comunicação. O método de controle de acesso (Medium Access Control-MAC) divide as implementações de Fieldbus em duas classes diametralmente opostas:

##### a.1) Barramentos com MAC centralizado:

No caso de mecanismos onde as funções de gerência e controle de acesso ao meio de transmissão são implantadas num único nó de comunicação diz-se que o controle de acesso é do tipo centralizado. O método de controle centralizado mais frequentemente utilizado, denominado mestre-escravo, baseia-se em consultas-convite ("Polling") que o nó mestre (estação principal ou ainda estação ativa) envia aos nós escravos (estações secundárias ou estações passivas), um a um, inquirindo-as a transmitir. A ordem das consultas convite é estabelecida em uma lista armazenada no nó central. Os nós escravos consultados, caso tenham alguma mensagem para transmitir, realizam a transmissão. Caso contrário, o nó mestre é notificado (p. ex. pela ausência de resposta em tempo hábil) e passa, então, a consultar o próximo nó na lista de consulta seqüencial. Com isto somente o nó mestre, pode iniciar uma transmissão, ao passo que os nós escravos

só transmitem quando consultados pelo nó mestre.

a.2) Barramentos com MAC distribuído (ou descentralizado):

Quando as funções de gerência e controle de acesso ao meio de transmissão são distribuídos entre todos os nós de comunicação ativos, tem-se caracterizado o mecanismo de controle de acesso do tipo distribuído (Ethernet, IEEE 802.4). Este pode ser adicionalmente classificado quanto ao aspecto temporal, em MAC determinístico ou probabilístico. O mecanismo de controle de acesso é dito determinístico quando o tempo de acesso ao meio, e portanto o tempo de resposta do barramento, pode ser previsto para uma dada configuração. Um exemplo típico é o procedimento "token-bus". O "token" neste mecanismo corresponde a uma mensagem de controle endereçada ciclicamente de nó em nó, formando um anel lógico e segundo uma tabela de ordenação lógica de endereços implantada nos nós de comunicação, que confere a estação que dela se apodera o direito de acesso/transmissão pelo barramento. No caso de anomalias causadas por erros de transmissão ou por falhas do hardware e software envolvidos na comunicação, o mecanismo de controle de acesso "Token-bus" provê uma série de procedimentos de modo a permitir a manutenção da operação do anel lógico. Anomalias típicas como a perda do "token" ou multiplicidade de "tokens", uma vez detectadas por temporização ou conflitos no acesso ao meio, acionam os procedimentos de reinicialização, envolvendo mensagens de controle para a regeneração do "token" e o reestabelecimento do anel lógico. O mecanismo também possibilita a reconfiguração dinâmica do anel lógico, isto é, a inserção ou remoção de nós sem interromper o seu funcionamento normal.

Os métodos de acesso probabilístico caracterizam-se pela imprevisibilidade do tempo de acesso de uma estação ao barramento. Neste método, a exemplo do protocolo CSMA/CD, amplamente usado em redes locais no controle de acesso, todas as estações podem transmitir, ou seja, podem difundir suas mensagens no meio de transmissão quando quiserem. Entretanto, como as estações interconectadas ao barramento geram suas mensagens no tempo de modo independente (aleatório), pode acontecer que dois ou mais nós de comunicação queiram ter acesso ao meio de transmissão comum no mesmo instante. Quando isto ocorre

tem-se uma situação de conflito ou colisão no acesso do meio compartilhado. O princípio de contenção, neste caso, é baseado na capacidade do nó de comunicação detectar as situações de conflito. Uma vez detectada uma colisão no acesso ao meio, cada um dos nós de comunicação conflitantes aguarda um novo instante para retransmitir sua mensagem. Este atraso obedece, em geral, a uma distribuição aleatória a fim de minimizar a probabilidade de uma nova colisão no momento das retransmissões [54,55,56,57,58].

#### b) Distância de Hamming (Hd)

A camada de enlace dispõe de mecanismos cujo objetivo é detectar a ocorrência de erros inerentes ao meio de transmissão e recuperar, através de retransmissões, a informação perdida ou mutilada. A medida da integridade de um quadro (bloco) de informação transmitido pela camada de enlace é dada pelo parâmetro Distância de Hamming (Hd), a qual corresponde na prática ao número de bits que devem estar alterados para causar a aceitação de um quadro espúrio pela camada de enlace. Uma distância de Hamming igual a 4 significa, a título de exemplo, que a ocorrência de tres bits de erro simultâneos na mensagem ainda são detectados pela camada de enlace.

#### 4.3.3 A Camada de aplicação

Na camada de aplicação (nível 7) procura-se implementar os serviços que visam a cooperação (comunicação) dos diversos programas de aplicação (situados em locais distintos da rede) na execução de uma tarefa distribuída. Exemplos de serviços oferecidos pela camada de aplicação são Read para a leitura e Write para a escrita em variáveis de processo. A camada de aplicação define e gerencia os denominados objetos de comunicação, como por exemplo variáveis, programas, eventos e campo de dados (cap. 6).

Estas tres camadas da arquitetura de Fieldbus colocam à disposição do programa usuário os elementos necessários a comunicação. No entanto requer-se ainda, para que os sistema de comunicação seja funcional, funções para inicialização e confi-

guração, para o gerenciamento da estação assim como identificação e recuperação de falhas. Estas funções são agrupadas na camada de gerência e constituem parte integrante da arquitetura de Fieldbus (Fig. 4.10).

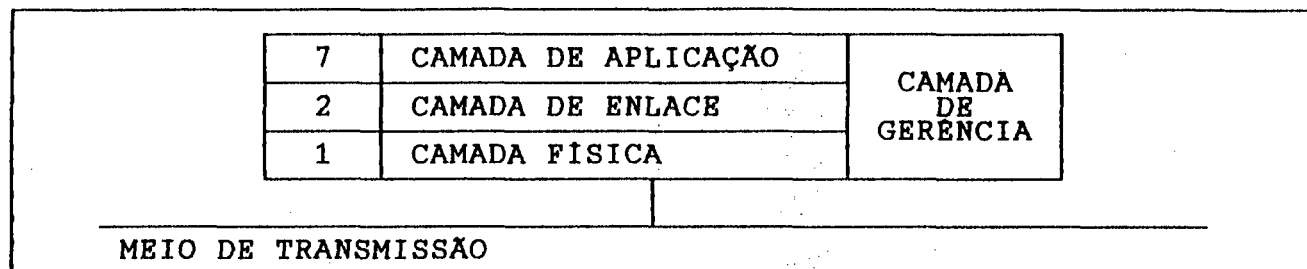


Fig 4.10 - Arquitetura de Fieldbus

A descrição acima evidencia que, sob o ponto de vista de arquitetura, a integração física de sistemas sensores é em Fieldbus realizada com as camadas 1 e 2. A informação que chega para o usuário remoto da camada de enlace, por exemplo através de uma primitiva de indicação SDA.Ind, indica unicamente a chegada de informações do parceiro de comunicação. No entanto o significado desta informação é desconhecido para o usuário desta camada, de modo a ter-se apenas o transporte de dados com a invocação dos serviços de enlace. Já a integração lógica, ou seja, a comunicação e a cooperação entre os processos de aplicação executados nos dispositivos é implementada com a funcionalidade da camada 7. Por exemplo, a ocorrência de uma primitiva de indicação Read.Ind no lado do sistema sensor, representa a comunicação que o processo de aplicação remoto faz para o processo local, do seu desejo de ler uma variável de processo especificada como parâmetro da primitiva, conforme será detalhado no capítulo 7.

#### 4.3.4 Arquiteturas abertas de Fieldbus

Fala-se de arquitetura aberta ou ainda de comunicação aberta quando existe uma especificação universalmente acessível e a interface é padronizada. Particularmente a interface Fieldbus aberta é de fundamental importância para a tecnologia de produção, por propiciar a necessária flexibilidade da configuração de tarefas como também a

redução dos custos do sistema pela possibilidade de especificação de soluções de diversos fabricantes. A desvantagem de uma interface de comunicação aberta e mais especificamente de um Fieldbus aberto é a generalidade da solução, decorrente da padronização, o que torna por conseguinte a interface mais complexa. Adicionalmente é transmitida informação redundante, o que reduz a eficiência de comunicação, conforme ilustrado na figura 4.11.

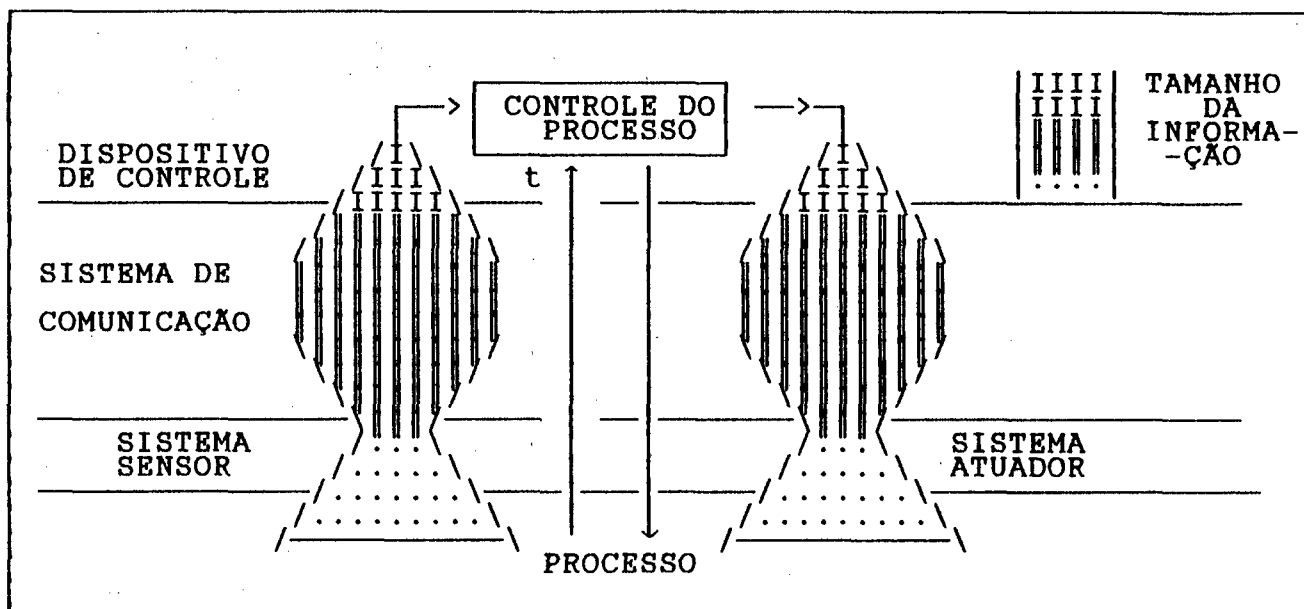


Fig. 4.11 - Quantidade de informação para o controle do processo

A parte larga tracejada ilustra a manipulação do conjunto de informação. O comprimento do elemento é proporcional ao "overhead" de processamento e transmissão, de modo que quanto mais eficiente for a solução de Fieldbus, tanto menor será a superfície tracejada. O sistema sensor, dotado de pré-processamento do sinal, comprime a informação, transmitindo unicamente os dados já pré-processados. Para se realizar a transmissão pelo barramento faz-se necessário acrescentar um conjunto de variáveis de comunicação causando um aumento no conjunto de informação. No sistema de controle apenas a informação do usuário é processada. Por isto reduz-se novamente o conjunto de informação. O efeito contrário verifica-se no controle dos elementos de atuação. A diferença de interface Fieldbus aberta e fechada manifesta-se no tamanho do elemento tracejado.

#### 4.4 Exemplos de especificações de projetos Fieldbus

Os exemplos discutidos a seguir ilustram apenas algumas possibilidades de soluções existentes, sem no entanto constituir a intenção de se esgotar o assunto, dada a diversidade de tipos e soluções atualmente disponíveis no mercado. Cita-se como exemplo o Bitbus, Profibus, Sercos, Autobus, Phoebus, Proway-C, Mil-STD 1553 e Fip, dos quais examinar-se-à na seqüência a título de ilustração os quatros primeiros, atribuindo-se no entanto maior atenção aos barramentos Profibus e Bitbus pela sua maior disseminação na indústria.

##### 4.4.1 Bitbus

Desenvolvido pela Intel em 1984, o Bitbus pode ser considerado uma interface padrão "de Facto", já dispondo de uma ampla gama de produtos e fornecedores, além de uma significativa base instalada. Permite a interconexão de até 250 dispositivos distribuídos ao longo de uma par trançado, podendo ser extendido até 13.2 Km com uma taxa de 62.5 Kbps e 10 repetidores [59,60,61,62,63]. Emprega a interface elétrica EIA-RS-485, topologia barramento, com 2 ou 4 condutores e suporta as seguintes modos de operação:

- " Self-Clocked-Mode": Este modo permite se estender a distância entre as estações pela combinação da informação de relógio e dados (NRZI) com a utilização de um único par de condutores (transmissão síncrona).
- " Synchronous-Mode": Concebido para distâncias curtas o suficiente para que a informação de relógio possa ser transmitida em um par de condutores exclusivo. Taxas de transmissão podem ser elevadas até 2.4 Mbps (transmissão síncrona).

O protocolo da camada de enlace do Bitbus adota a distância de Hamming 1 e emprega um esquema de "polling" baseado na configuração mestre-escravo, devido a simplicidade e redução de custo implicados. O protocolo de enlace consiste em um "subset" do SDLC (IBM Synchronous Data Link Control), baseado no protocolo HDLC da ISO.

Acima da camada de enlace é especificada a chamada interface RAC (Real time Access and Control), cujos serviços estão ilustrados na figura 4.12. Quando os resultados de um serviço estão disponíveis, são retornados em uma mensagem de resposta (reply).

O elemento central do Bitbus é microcontrolador 8044, o qual integra uma interface serial inteligente (SIU- Serial Interface Unit) para a implementação do protocolo de enlace em chip, obtendo-se com isto tempos de reação abaixo de 1ms. As estações do barramento contém um núcleo de tempo real (iDCX-51) através do qual as tarefas em execução utilizam os serviços de comunicação e controle de rede providos pela interface RAC para o acesso do meio de comunicação.

SERVIÇO	ACESSO	CONTROLE
RESET SLAVE		X
CREATE TASK		X
DELETE TASK		X
GET FUNCTION ID		X
RAC PROTECT		X
READ I/O	X	
WRITE I/O	X	
UPDATE I/O	X	
UPLOAD MEMORY	X	
DOWNLOAD MEMORY	X	
OR I/O	X	
AND I/O	X	
XOR I/O	X	
STATUS READ	X	
STATUS WRITE	X	

Fig. 4.12 - Comandos RAC

#### 4.4.2 Profibus

Uma importante interface aberta de Fieldbus constitui o Profibus, especificado através das normas DIN V19245 partes 1 e 2. A relação dos produtos e fabricantes, as aplicações e maiores detalhes sobre o protocolo Profibus podem ser encontrados nas referências [64,65,66,67,68,69,70,71,72]. As características básicas, dispostas de acordo com a sua arquitetura de camadas são as seguintes:

##### a) Camada Física

Uma característica fundamental do padrão Profibus a nível de camada física é a multiplicidade de topologias e taxas de transmissão objetivando suportar adequadamente várias aplicações. Inicialmente a

norma preve a adoção das seguintes especificações para a camada física:

- Topologia básica: barramento
- Interface elétrica RS-485
- meio físico: par trançado blindado (p. Ex. 24 AWG)
- codificação dos bits tipo NRZ, transmissão assíncrona
- alcance sem repetidor: 2,4 Km  
(dependente da taxa de transmissão)
- alcance com repetidor: 9,6 Km
- taxa de transmissão máx.: 500 kbps.
- Considera-se 32 estações (ativas, passivas e repetidoras) por segmento.

Especificações adicionais serão providas posteriormente para aplicações em áreas de alto risco e também para aplicações de maior desempenho.

#### b) Camada de Enlace

A camada de enlace do Profibus (FDL - Fieldbus Data Link) especifica dois níveis de distância de Hamming (Hd), 2 e 4, e adota o chamado método híbrido de controle de acesso, que combina as vantagens do método "token-passing" com o método mestre-escravo.

A figura 4.13 mostra uma estrutura Profibus típica com 3 estações ativas (mestres) e quatro estações passivas (escravas). Se, por exemplo, a estação ativa com endereço 2 recebe o "token" da estação precedente, é capaz de exercer a função mestre através do barramento e comunicar-se com todas estações ativas e passivas em uma relação mestre-escravo. A passagem cíclica do "token" pelas estações ativas constitui a formação de um anel lógico o qual deverá ser inicializado e mantido às custas de um "overhead" não justificável para as aplicações mais simples.

Além de controlar o acesso ao barramento a camada FDL também é responsável pela provisão dos seguintes serviços para transmissão e requisição acíclica de dados:



- Send Data with No Acknowledge SDN
- Send Data with Acknowledge SDA
- Send and Request Data SRD

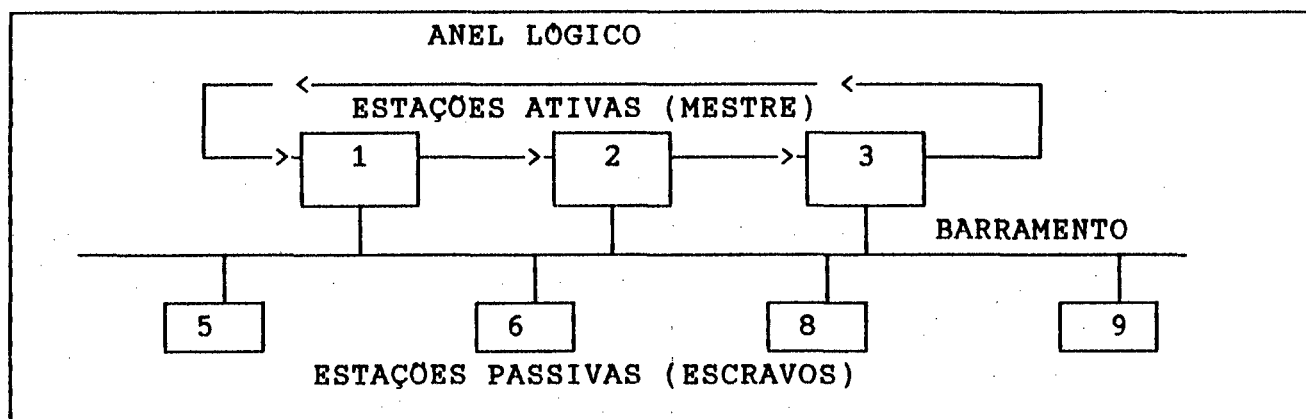


Fig. 4.13 - Método Híbrido de Acesso ao Barramento

O serviço SDN permite a realização da transmissão "Broadcast", a qual caracteriza-se pela difusão de informação de uma estação ativa para todas as demais do barramento, sem a confirmação do serviço. Exemplo de aplicação deste serviço é a sincronização de sistemas sensores no barramento para a atualização da hora.

O serviço SDA possibilita o transporte de dados de uma estação mestre para uma única estação remota. Na ausência de confirmação do recebimento da informação transmitida, dentro de um intervalo de tempo pré-especificado (slot-time), ocorre a retransmissão da mesma antes que uma confirmação negativa seja repassada para o usuário local do serviço. Este mesmo mecanismo de temporização e retransmissão é utilizado em todos os demais serviços discutidos a seguir.

O serviço SRD habilita a transmissão de dados a uma estação remota e simultaneamente requisita dados que tenham sido previamente dispostos pela mesma estação. A estação local recebe uma confirmação que provê o dado requisitado ou então informação indicando a causa da indisponibilidade deste.

Adicionalmente aos serviços acíclicos, as aplicações industriais frequentemente requerem o uso de procedimentos de transmissão

cíclica. O Profibus permite a criação de uma lista de consultas-convite (poll-list) contendo a relação das estações a serem incluídas no processo de varredura cíclica da estação mestre, liberando assim a aplicação desta tarefa.

Com o serviço cíclico CSDR (Cyclic Send data with Reply-intercâmbio de dados) habilita-se a estação local a enviar dados ciclicamente a diversas estações como também requisitar dados das mesmas. Também neste caso a PDU de confirmação prove dados ou argumentos indicando a causa da indisponibilidade dos mesmos.

Os serviços mencionados compõe-se das primitivas (como chamadas de subrotinas) Request, Indication e Confirmation. A primitiva de resposta não é necessária na camada de enlace pela característica de resposta imediata.

Cada primitiva compõe-se por sua vez da unidade de dados do usuário a ser transportada e parâmetros outros como endereço, SAP fonte e destino, status da operação e classe de serviço. Este último parâmetro define a prioridade da respectiva primitiva. Estes parâmetros são passados para a camada superior ou inferior.

### c) Camada de Aplicação

A camada de aplicação do Profibus é composta de duas subcamadas:

-FMS (Fieldbus Message Specification), que descreve os objetos de comunicação, serviços e a partir destes, a descrição do modelo do dispositivo de campo virtual (cap. 6);

- LLI (Lower Layer Interface), responsável dentre outros pelo mapeamento dos serviços FMS e FMA7 (Fieldbus Management) em serviços FDL, também pela incorporação da funcionalidade necessária das camadas 3 a 6;

Adicionalmente a camada de gerência FMA7 (Fieldbus Management) fornece serviços de gerência local ou remotos para o reconhecimento e remediação de faltas, acesso a variáveis de comunicação da camada

de enlace, carga e leitura da CRL (item 6.3), identificação das estações de comunicação (item 6.2.7) como também para o estabelecimento e encerramento de conexões (item 6.2.5).

#### d) Implementação Profibus

O projeto e desenvolvimento do protótipo de hardware de uma estação escrava baseada no microcontrolador Intel 8051 (ver anexo A) e a implementação do protocolo de controle de acesso ao meio são resultados integrantes do presente trabalho, o qual insere-se no contexto maior do projeto Certi PADCT-Fieldbus [73]. A configuração global deste está ilustrada na fig. 4.14 e congrega os seguintes elementos:

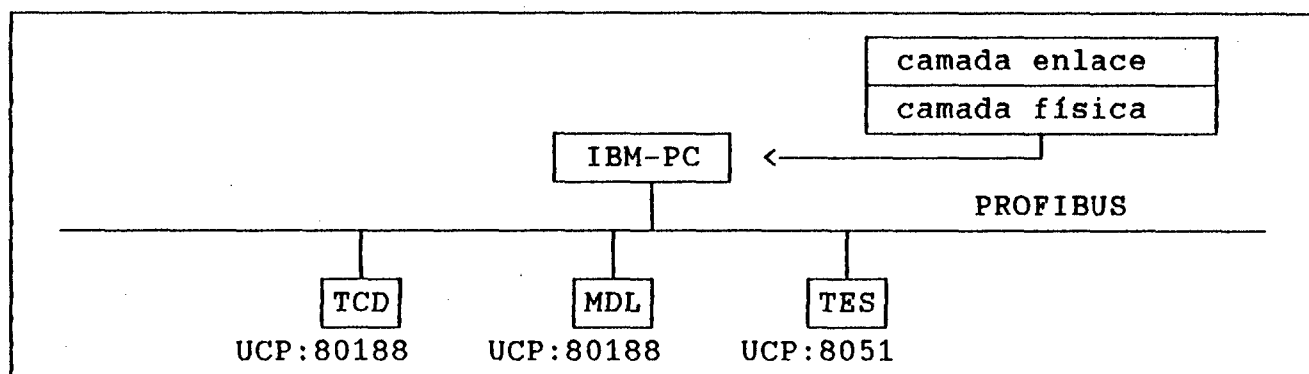


Fig. 4.14 - Configuração adotada - Fieldbus Certi/PADCT

##### d.1) Terminal Concentrador de dispositivos

Baseado no microcontrolador Intel 80188, o terminal concentrador de dispositivos (TCD) é um dispositivo com interface Profibus incorporada. Esta interface inclui tanto para o TCD como também para os demais elementos da configuração (MDL e TES) os circuitos associados à camada física (transceptores) e os protocolos da camada de enlace. O TCD pode ser utilizado na integração de instrumentação, tais como transdutores, controladores, totalizadores e sistemas de medição que dispõem exclusivamente de interfaces convencionais (4-20 mA, 0-24 V, TTL, EIA-RS-232C). Suas características básicas são:

- interface com o operador (teclado e display);

- 12 saídas digitais em coletor aberto (0-30 V) e 12 entradas opto-acopladas para interfacear com processos em chão de fábrica (monitorar pontos digitais);
- 16 entradas (em tensão ou corrente) analógicas multiplexadas.

#### d.2) Sistema de Medição de deslocamento do tipo Laser de alinhamento

Baseado no microcontrolador 80188, este sistema (MDL) está dotado de interface Profibus, podendo ser utilizado na realização de ensaios geométricos de máquinas, equipamentos, estruturas e componentes mecânicos, nas atividades de montagem, manutenção e qualificação. O sistema destina-se à realização dos ensaios de retilineidade e seus derivados, como ortogonalidade, planicidade, alinhamento e paralelismo, atendendo à verificação de movimentos e superfícies.

#### d.3) Terminal com entradas/saídas digitais

Baseado no microcontrolador Intel-8051, constitui uma estação simplificada (TES) utilizando as próprias portas de E/S digital da UCP para o interfacimento com chaves ON/OFF, fim-de-curso, sistemas de medição com saídas em BCD (por exemplo, medidores de deslocamento eletro-óptico) e relés dentre outras possibilidades (ver anexo A).

#### d.4) Estação Profibus para microcomputador IBM-PC

Esta estação consiste de uma interface diretamente acoplada ao barramento do PC, permitindo ao usuário ter acesso a primitivas de comunicação através de uma biblioteca de software. Esta estação atuará, sob o ponto de vista da aplicação, como controladora da aplicação, e sob o ponto de vista de comunicação como uma estação mestre. Sua operação abrangerá a leitura dos sistemas sensores, o processamento da informação, a atuação em sistemas atuadores além da interação com o operador do ensaio através de telas e menus.

#### 4.4.3 Fip

Fip é o padrão Francês desenvolvido por um grupo de empresas de alta tecnologia dentre as quais destaca-se CGEE Alstom e Telemecanique [74]. Processos de aplicação considerados pelo Fip são essencialmente cíclicos (periódicos) com atualização de dados em instantes bem definidos. As principais características do Fip a nível de camada Física são:

- taxa de transmissão: 1 Mbits/s (31,25 Kbits/s e 2,5 Mbits/s para aplicações específicas);
- alcance: até 2000 m;
- número de estações: até 256;
- suporte de transmissão: par trançado blindado ou fibra óptica;

Dentro do projeto Fip já estão disponíveis três componentes ASIC (circuitos integrados de aplicação específica). O FULLFIP, desenvolvido pela Cegelec, é um co-processador de comunicação que suporta todos os protocolos Fip, incluindo as funcionalidades da camada de aplicação. O FIPART provê funções de transmissão e recepção em um barramento FIP e comporta apenas a camada 1 e uma parte da camada 2. O FIPIU, por fim, desenvolvido pela Telemecanique, suporta os protocolos FIP das camadas 1 e 2, bem como a função de árbitro do barramento, podendo ser usado com um ou dois processadores, com os quais compartilha a memória; enfim, ele comporta o conjunto de serviços, mas de uma forma mais simplificada. O Fip consiste num barramento de campo relativamente revolucionário, baseado no modelo produtor-consumidor. Os consumidores podem ser algoritmos, sistemas de supervisão, histórico do SDCD dentre outros. Nem produtores nem consumidores em princípio tomam a iniciativa de transmitir informações porque se todos estes elementos começarem a colocar dados na rede ou fazer solicitações estará se excedendo a capacidade de transmissão de dados pelo barramento que em Fieldbus devido aos requisitos de baixo custo tem de ser pequena. O Fip então adota a figura do controlador de barramento, definindo um elemento especial na rede, o árbitro, que passa a comandar as ações, requisitando aos produtores que coloquem os dados na rede devido a existência de consumidores que irão consumir os dados. O Fip foi concebido para integração de instrumentação de campo e é um projeto

com características tempo-real e que veio a ter grande impacto no ISA SP-50.

#### 4.4.4 Projeto SP-50 (ISA)

Previendo a possível invasão de seu mercado por Fip e/ou Profibus, os fabricantes norte-americanos trataram de criar, dentro da ISA (Instrument Society of America), o projeto SP-50, atualmente em discussão, congregando fabricantes e usuários para estabelecer um protocolo padrão a ser seguido por todos quantos quisessem fornecer para empresas dos EUA [17]. Pela importância do mercado americano, o SP-50 acabou por atrair para si também o pessoal responsável pelo Fip e Profibus, que agora disputam, via SP-50, a fixação das suas respectivas soluções como padrão reconhecido internacionalmente. O projeto da ISA SP-50 é, de fato, o único com chances de abrangência internacional, na opinião de especialistas, pois hoje estão nele envolvidos praticamente todos os órgãos que elaboraram projetos nacionais, como Fip e Profibus, tentando uma solução de compromisso que, se obtida, deverá ser aceita no mundo todo. O SP-50 procura atender aos requisitos da área de processos (Fieldbus H1) e automação da manufatura (Fieldbus H2), conforme resumido abaixo:

##### a) Fieldbus SP-50 H1 (Automação de processos)

- taxa de transmissão: 31,25 Kbits/s;
- alcance: até 1900 m;
- número de estações: até 32;
- suporte de transmissão: par trançado;
- topologia: barramento e estrela.

##### b) Fieldbus SP-50 H2 (Automação da manufatura)

- taxa de transmissão: 1 Mbps;
- alcance: até 750 m;
- número de estações: até 30;
- suporte de transmissão: par trançado;
- topologia: barramento.

#### 4.4.5 Sercos

Consiste no resultado de um esforço conjunto do VDW (Verein Deutscher Werkzeug Maschinen Hersteller) e ZVEI (Zentral Verband Elektrotechnik und Elektronik Industrie) para o desenvolvimento de um sistema de comunicação entre controle numérico e acionamentos de uma máquina ferramenta [75,76]. O Fieldbus Sercos (Serial Real-time Communication System) tem sua concepção voltada para tarefas de controle de momento, velocidade e posição. Suas características básicas são:

- Suporte de transmissão: fibra ótica
- Topologia: Anel
- Alcance: limitado em 30 m
- Taxa de transmissão: até 4 Mbaud
- Tipo de codificação: banda base
- Método de acesso: mestre-escravo

Devido a adoção da topologia anel cada estação comporta-se com um repetidor, sem o que a operação do barramento não se viabiliza. Adicionalmente verifica-se a dependência do ciclo de varredura ao número de estações em operação no anel e a necessidade de sincronização do processamento de dados com o ciclo de transmissão.

#### 4.4.6 Auto-bus

Os mais recentes sistemas em eletrônica embarcada vem adquirindo um crescente nível de sofisticação, o que se pode verificar através da tendência para a incorporação de microprocessadores dedicados ao controle das diversas funções dos automóveis, os quais se interagem através de um sistema de comunicação. Este sistema de comunicação constitui-se um nicho de aplicação bastante expressivo para o Fieldbus, com a problemática que apresenta em termos de espaços, cablagem e confiabilidade. Neste contexto o Auto-bus CAN (Controller Area Network) e o VAN (Vehicle Area Network) representam um exemplo significativo de solução voltada para o acoplamento de subsistemas

em automóveis [77,78,79,80]. Suas características são a nível de camada física:

- taxa de transmissão: 1 Mbaud
- alcance: 40 m
- topologia: barramento
- suporte de transmissão: par trançado
- tipo de codificação: banda base

O método de Acesso empregado é uma variante do CSMA/CD, com arbitrio de prioridade. Na ocorrência de conflitos transmite-se primeiramente a informação de maior prioridade. A implementação das tarefas de comunicação é suportada pelo componente integrado Intel 82526 ou Philips 82C900, com o que torna-se possível alcançar-se um tempo de resposta abaixo de 420 microssegundos. Adicionalmente o sistema oferece uma distância de Hamming 6 e, sob o aspecto econômico a expectativa é de preços relativamente baixos devido a sua produção em escala que se preconiza.

A fig. 4.15 resume algumas características comparativas de alguns dos projetos citados. Convém nela destacar a existência de barramentos desenvolvidos para classes específicas de aplicação.

FIELD BUS	SERCOS	AUTO-BUS	BITBUS	PROFIBUS
CARACTERÍSTICA				
TOPOLOGIA	ANEL	BUS	BUS	BUS
MODO DE TRANSMISSÃO	-----	-----	SÍNCRONO	ASSÍNCRONO
TIPO DE CODIFICAÇÃO	BANDA BASE	BANDA BASE	BANDA BASE	BANDA BASE
TAXA DE TRANSMISSÃO	até 4 Mbps	1 Mbaud	2,4 Mbps máx.	500 kbps máx.
CONTROLE DE DE ACESSO	mestre-escravo	CSMA/CD	mestre-escravo	híbrido
Hd	-----	6	1	2 e 4
SUPORTE DE TRANSMISSÃO	fibra-ótica	par trançado	par trançado	par tranç. fibra-ótica
ALCANCE	max.30m	40 m	1200 m	2.4 km máx.

Fig. 4.15 - Tabela comparativa de várias soluções de Fieldbus



Aplicações em processos confinados, onde distância não é fator relevante mas sim a velocidade, são melhor atendidas com o barramento Sercos. Contrariamente, aplicações na automação da manufatura e em controle de processos relativamente dispersas, que não requeiram do barramento altas velocidades, podem ser servidos satisfatoriamente com o barramento Profibus. Para aplicações que requeiram alta integridade na transmissão da informação, por estarem sujeitas a uma série de ruídos, voltadas para a integração de dispositivos em unidades móveis (eletrônica embarcada), foi concebido especificamente o barramento Auto-Bus, que oferece a maior distância de Hamming ( e portanto maior confiabilidade). E finalmente para aplicações em ambientes locais relativamente pouco ruidosos com requisitos médios de velocidade, o Bitbus constitui uma significativa alternativa a se considerar. Fica claro portanto desde já que a existência de nichos de aplicações, como também de barramentos específicos que são concebidos para se atender exclusivamente a tais classes de aplicações, não se limitam aos anteriormente citados - paralelamente a isto percebe-se a impossibilidade de se atender às aplicações em automação industrial com um único padrão internacional de Fieldbus.

#### 4.5 Comentários sobre Fieldbus no contexto da automação industrial

O conhecimento da atual panorâmica do Fieldbus, incluindo mercado, suas tendências e expectativas, é sem dúvida extremamente importante para o usuário ou fornecedor que, a curto, médio ou longo prazo, visualiza um investimento nesta tecnologia. Para isto pretende-se aqui sucintamente apontar e questionar os possíveis caminhos futuros da automação do chão de fábrica, tendo em perspectiva o Fieldbus, tomando para isto em consideração as seguintes observações[81]:

- a) Fip e Profibus foram, são e serão projetos nacionais, na visão de alguns especialistas, o que não impede, evidentemente, que empresas de outros lados do mundo venham a eles aderir. O único projeto atualmente com chances de ser internacional seria o projeto ISA SP-50. Infelizmente esse projeto ainda pode demorar. Mas de qualquer forma o projeto já está hoje muito claro sobre os

níveis físicos e de enlace;

- b) O objetivo de qualquer dos projetos, FIP, Profibus ou SP-50 é obviamente corrigir a distorção existente no MiniMAP, ou seja, criar uma camada de aplicação inspirada certamente no MMS, mas também uma especificação de protocolo que considere a ausência das camadas de rede, transporte, sessão e apresentação do modelo ISO, e que seja mais próxima do usuário de chão de fábrica, que não é um analista de sistemas;
- c) Muitas empresas estão esperando o Fieldbus ISA SP-50, notadamente as norte-americanas. Parte das empresas alemãs estão esperando o que a IEC definir. Então quando se tiver os testes do SP-50, haverá uma discussão em torno do que é melhor, qual foi o desempenho, porque não usar e qual o caminho a seguir. Acredita-se então que a partir deste ano haverá bases para discussão do destino do Fieldbus;
- d) Os alemães basearam-se no padrão IEEE 802.4, o Tokenbus, mas simplificaram, acrescentaram algumas primitivas de comunicação e especificaram um protocolo MMS na camada 7 um pouco simplificado. E produziram um projeto nada de revolucionário, e ao contrário do que se vislumbrava inicialmente, com uma especificação mais adequada para o nível de célula, e que certamente tem grandes chances de vir a substituir o barramento MiniMap nesse campo. Mas não é um projeto com características de tempo real e com a simplicidade que se verifica no FIP, portanto não é propriamente adequado para os instrumentos de campo. A tendência é cada vez mais FIP ser visto claramente como sendo um Fieldbus, enquanto o Profibus aproxima-se mais de um barramento controlador de células;
- e) Muito se tem questionado sobre a proposta de se pensar em algo de ponta como o Fieldbus no Brasil de hoje, onde tanta coisa há ainda a ser feita e onde, em muitos casos não se instaurou nem mesmo uma cultura básica sobre a automação industrial. A posição da maioria dos potenciais usuários nacionais de Fieldbus no momento é a de querer conhecer o Fieldbus, testá-lo e verificar seu desempenho, analisando as tendências para o futuro. Estão

utilizando uma miscelânea de redes, tais como Arcnet, Datahighway, terminais de aquisição de dados em células com supervisão através de redes proprietárias Ethernet. Não se pretende esperar pelo barramento ideal, que pode ser inimigo da perfeição neste caso. Então está se pendendo bastante para padrões de fato e sistemas de implementação relativamente fácil. A tendência é se querer a coisa mais perto de sistemas abertos quanto possível;

- f) Por parte dos fabricantes há ainda uma postura de espera por essa tecnologia para se fazer alguma coisa. O nosso mercado acompanha tudo com distância, a ponto de pouco se preparar para isso. O próprio usuário, na hora em que tiver que comprar o sistema Fieldbus, frequentemente questionará sobre a preparação da sua equipe e sobre o tempo e custo necessários para educar esse pessoal;
- g) Verifica-se a que antes de se colocar uma tecnologia nova, é preciso resolver os problemas internos existentes nas empresas nacionais. Por exemplo, cita-se a implantação de CEP, que é algo que pode ser implementado com Fieldbus, na coleta de dados dos sistemas sensores, para buscar a informação necessária para o controle do processo. Mas esbarra-se na falta de cultura do pessoal para CEP. Verifica-se que está se discutindo um padrão que está no futuro enquanto as empresas brasileiras, devido às próprias deficiências do país, da estrutura econômica, não conseguem avançar muito rapidamente. Os fabricantes estão lá na frente, há algumas empresas muito avançadas, como multinacionais, talvez algumas estatais, e um grande número de empresas que poderiam estar utilizando o Fieldbus mas, antes de usar, precisam ter um trabalho básico internamente. Esse é o grande problema das empresas nacionais. Ao tentar trabalhar com um aquisitor de base de dados para a parte de CEP é grande a oferta e a multiplicidade de dispositivos e sistemas, embora não se saiba sequer se o operário saberá usá-los no chão de fábrica, pois falta cultura para isso;
- h) O Fieldbus é aberto e cria uma oportunidade para várias empresas, para os usuários em geral, mas tira a fatia de mercado que é restrita aos grandes fabricantes. A maioria dos grandes fabricantes não quer um Fieldbus intercambiável, mas o pequeno

fabricante que está no SP-50, e cujo voto tem o mesmo peso que o do grande está dizendo que quer;

i) Há diferenças de mercados para aceitação de qualquer que seja o Fieldbus que se firme. Hoje, por exemplo, tem-se o Profibus se alastrando na Alemanha e o Fip consolidado na Europa - várias pesquisas em outros países além da França estão sendo feitas à base de Fip; a Itália e a Noruega estão em peso com o Fip - e tem-se o SP-50, que conseguiu chamar a atenção a nível internacional. Tanto a IEC, que está trabalhando junto com a ISA, quanto dos franceses - o pessoal que trabalhou no Fip hoje está no SP-50, como também do pessoal do Profibus, mais relutante mas também apresentando trabalhos e discutindo o SP-50. Mas apesar dessa tendência pelo SP-50 no momento, acredita-se haver um problema sério depois, porque existe uma diferença muito grande entre os mercados norte-americano e europeu, ambos muito grandes. A unificação do mercado Europeu atraiu a atenção global de usuários e fornecedores, porque ele será mais forte que o norte-americano. Só que o mercado europeu é muito exigente em termos de segurança. As normas da Cenelec, da Europa, são 50% mais rigorosas que as regras de órgão dos EUA e Canadá, o que caracteriza uma diferenciação de mercado muito grande. E além disso, o mercado europeu tem uma confiabilidade muito maior que a do norte-americano. Então depois dessa discussão toda de qual é o padrão, na implementação pode acontecer uma rejeição ao protocolo devida à falta de confiabilidade. Sabe-se hoje que o mercado alemão prefere pagar mais pelos seus produtos a importar. É uma proteção, mas é também uma questão de confiabilidade, porque ele sabe qual o nível de segurança do que ele faz. Essa poderá ser uma reação estendida ao mercado europeu em relação ao SP-50 ;

j) A existência de um dois ou tres padrões é melhor que 25 ou 30 padrões e melhor que nenhum. Com certeza, se o limite for em torno de três, quatro ou cinco, o mercado irá fornecer Bridges para fazer a conexão de um com o outro. Isso fará com que diferentes padrões de Fieldbus possam conviver extremamente bem numa planta. Eventualmente o custo de treinamento da equipe de manutenção, o custo de preparação, será maior se utilizar mais de um padrão. Mas de qualquer maneira eles conviverão. O difícil é

não se ter padrão ou se ter uma quantidade muito grande, onde essa viabilidade de Bridges ou Gateways fica muito complicada. Hoje em cada sistema que vai ser integrado é preciso fazer um trabalho de software praticamente novo, porque tem um novo padrão, ou então um sistema proprietário, e há um desenvolvimento de software quase que caso a caso. Com um número pequeno de padrões vai haver uma escala maior de sistemas intermediários para interconexão de diferentes tecnologias de Fieldbus;

- k) A questão da utilização do fieldbus na interligação de sensores e atuadores no âmbito interno a uma máquina (unidade integrada de produção) ainda é objeto de estudo. Um dos grandes benefícios do Fieldbus, a redução da cablagem, notadamente não é tão significativo para a automatização de máquinas ferramenta, por exemplo, devido as distâncias serem relativamente pequenas se comparadas às distâncias das instalações de produção. Também o custo das interfaces acaba ficando mais caro em relação as tradicionais ligações analógicas ponto-a-ponto, ao se considerar os custos dos protocolos de comunicação e dos componentes integrados que deverão implementá-los. Estes custos tendem naturalmente a se reduzirem com a produção em escala de circuitos integrados com protocolo Fieldbus. Também a integração em hardware dos protocolos de comunicação permitirão um melhor desempenho temporal do Fieldbus;

- l) Como hoje o Fieldbus está, tanto se pode levá-lo para nível de célula quanto para baixo, a nível de processos. Sua finalidade no entanto seria resolver o problema de chão de fábrica, por isso se chama Fieldbus, e não se preocupar com a parte de cima, para a qual já existem muitas soluções. Se o Fieldbus cumprir a finalidade para a qual foi idealizado, não haverá problema para convivência em outros níveis com os DCDSs de hoje ou do futuro. Pode-se afirmar com certeza de que a rede proprietária que está hoje no DCDS, com grande probabilidade vai se transformar numa rede aberta. A boa resposta para essa questão seria esta: o Fieldbus não vai destruir o DCDS, mas vai ser um elemento muito importante para essa próxima geração de DCDSs, que é certamente uma geração muito aberta. Não haverá mais aqueles blocos

monolíticos que existem hoje;

- m) O Fieldbus não precisa ser um padrão para provocar uma evolução com uma grande revolução. Fieldbus é uma idéia que, na hora em que atingir o mercado, com tres, quatro, cinco padrões, vai mudar esse mercado, vai mudar a forma de se projetar um CLP, a forma de projetar um DCDS, mudar as instalações das plantas além da forma do operador trabalhar;
- n) Em algumas áreas (celulose por exemplo), grandes empresas nacionais estão no mesmo nível tecnológico que as escandinavas, que tem o maior nível tecnológico do mundo. Não é o caso da maioria das empresas do Brasil, mas é o caso pelo menos deste tipo de empresa. E para continuar competitiva, ela deve continuar evoluindo. É interessante que, a partir do momento em que se chegou ao nível de competitividade do primeiro mundo, é muito difícil continuar copiando, porque não há mais o que copiar;
- o) O Fieldbus tem uma proposta de conviver com alguns dos padrões já existentes, por exemplo, o 4-20 mA, e isso daria o que se pode chamar de direitos iguais de comunicação aos elementos de campo, mais próximos do processo, com os elementos lá de cima. Toda e qualquer proposta de padrão terá que conviver com as outras durante muito tempo;
- p) O Comitê ISA SP-50 introduziu outra camada, chamada de camada do usuário (PCUL-process control user layer). Quem quer ser interconectável, não vai usar nada do que existe nas camadas do usuário e de aplicação e comunica-se diretamente ao nível da camada de enlace. Já o outro nível permite que os elementos, além de interconectáveis, também interajam e, para isso, as mensagens trocadas tem que obedecer a uma semântica rigorosa. Bem, esses equipamentos terão que interagir através de uma camada de aplicação já sofisticada, tipo MMS, que serve justamente para permitir a interoperabilidade dos instrumentos interconectados. Mas ainda veio alguém e disse: só isso não chega, eu quero intercambiar. Já faço isso no 4-20 mA, tenho que fazer no digital, ou seja, a placa PID apresenta defeito, é trocada por outra de qualquer fabricante. Bem isso colocou um problema

gravíssimo, levando a necessidade de se criar uma espécie de camada do usuário que só serve para esses instrumentos intercambiáveis. Quer dizer, os que apenas se interconectam ou interoperam, se comunicam via as camadas normais que já conhecemos. Quem quer ser intercambiável, comunica-se via essa camada do usuário, mas, se não quer, pode ir pela camada de aplicação (interoperação) ou pela de enlace (interconexão);

q) O preço de conexão de um sistema sensor com entradas e saídas em tensão/corrente dotado de interface Fieldbus, para produtos existentes no mercado atual, fica na faixa de 700 a 800 US\$, sendo que o custo de fabricação estimado está na faixa de 200 US\$. Sistemas sensores dedicados (transmissores inteligentes) com interface Fieldbus tendem a ser mais caros devido a tecnologia de sensor embutido, como é o exemplo de sistemas sensores de pressão (ou transmissores de pressão inteligentes), cujo preço está situado em torno de 1400 US\$.

r) Não se tem ilusões de que o Fieldbus vá logo para uma indústria, pois sabe-se que a maioria dos problemas brasileiros hoje são resolvidos, por exemplo, com Ethernet, que conta com uma ampla base instalada, vários fornecedores, custo acessível e opera relativamente bem em tempo real.

## 5 MAPEAMENTO DOS REQUISITOS DE APLICAÇÃO EM REQUISITOS DE COMUNICAÇÃO FIELDBUS - ASPECTOS DA INTEGRAÇÃO FÍSICA DE SISTEMAS SENSORES

Um terceiro passo em direção a integração de sistemas sensores em sistemas de produção consiste na determinação dos requisitos da solução de interface de comunicação a ser selecionada. Tais requisitos podem ser identificados a partir da análise dos requisitos do processo de aplicação em consideração. Este capítulo objetiva examinar de que forma o processo de aplicação, cujos requisitos foram relacionados no capítulo 2, pode influenciar nas características e funcionalidade das camadas física e enlace da arquitetura Fieldbus, por estarem diretamente relacionadas com a integração física de sistemas sensores. Com isto pretende-se dispor de uma base para a seleção de uma solução de Fieldbus mais apropriada às necessidades de integração dos sistemas sensores.

A seguir discute-se o relacionamento entre os requisitos das aplicações com os requisitos Fieldbus sumarizados na fig. 5.1.

REQUISITO DA APLICAÇÃO	REQUISITO DE COMUNICAÇÃO CORRESPONDENTE
DISPERSÃO FÍSICA DAS VARIÁVEIS DO PROCESSO	ALCANCE DA REDE <ul style="list-style-type: none"> <li>- taxa de transmissão</li> <li>- tipo de suporte de transmissão</li> <li>- tipo de codificação do sinal</li> <li>- número de repetidores</li> </ul>
NÚMERO DE VARIÁVEIS DO PROCESSO	NÚMERO DE ESTAÇÕES DO FIELDBUS
COMPORTAMENTO DINÂMICO DAS VARIÁVEIS DO PROCESSO	ATRASO DE TRANSFERÊNCIA DO FIELDBUS <ul style="list-style-type: none"> <li>- tempo de processamento do protocolo</li> <li>- tempo de transmissão</li> <li>- comprimento do suporte de Tx. (atraso de propagação)</li> </ul> TIPO DE SUPORTE DE TRANSMISSÃO NÚMERO DE ESTAÇÕES DO FIELDBUS
CONDIÇÕES AMBIENTAIS DA INSTALAÇÃO DE PRODUÇÃO	TIPO DE SUPORTE DE TRANSMISSÃO DISTÂNCIA DE HAMMING (Hd) REDUNDÂNCIA DE MEIO FÍSICO E ESTAÇÕES
TAMANHO DA MENSAGEM DA APLICAÇÃO	LARGURA DE BANDA DO FIELDBUS
NECESSIDADE DE GERENCIAMENTO REMOTO DE EVENTOS, PROGRAMAS, E MANIPULAÇÃO DE VARIÁVEIS	SERVIÇOS DA CAMADA DE APLICAÇÃO

Fig. 5.1 Mapeamento dos requisitos de aplicação



## 5.1 Variáveis de processo

Tanto o número de variáveis a controlar em um processo como também a própria dispersão geográfica das mesmas sobre a área da planta incide essencialmente sobre as características da camada física, conforme analisa o item seguinte.

### 5.1.1 Dispersão física das variáveis de processo sobre a instalação de produção

A utilização de transmissão digital por si só permite ao Fieldbus uma boa característica de alcance relativamente aos tradicionais sistemas analógicos. Já a adoção da topologia em barramento tende a limitar o alcance, visto que as estações conectam-se ao Fieldbus como elementos passivos, ou sejam, não regeneram o sinal de transmissão como no caso das estações na topologia em anel. Esta mesma limitação constitui uma vantagem quanto a confiabilidade do Fieldbus, considerando que a perda de uma estação não compromete a operação global do sistema de comunicação.

A definição do alcance da implementação de Fieldbus pode influir, em última análise, na especificação da taxa de transmissão, do tipo de suporte de transmissão, do tipo de codificação do sinal e do número de repetidores do barramento como analisado a seguir.

#### a) Relação alcance x taxa de transmissão

O alcance máximo permissível pela configuração está relacionado com a taxa de transmissão e esta dependência é expressa pelo produto taxa de transmissão x distância [54,55].

#### b) Relação alcance x tipo (características) do suporte de transmissão

O produto taxa de transmissão x distância ( e portanto o alcance) varia em função do tipo de suporte de transmissão empregado, se cabo

coaxial, par trançado ou fibra ótica. Para um mesmo tipo de cabo, a bitola e a sua capacitância influenciam o máximo comprimento do barramento para uma dada taxa. Quanto maior a bitola e menor a capacitância do cabo, maiores serão as distâncias permissíveis. A fig. 5.2 ilustra a relação alcance x taxa de transmissão para o par trançado.

Distância (m)

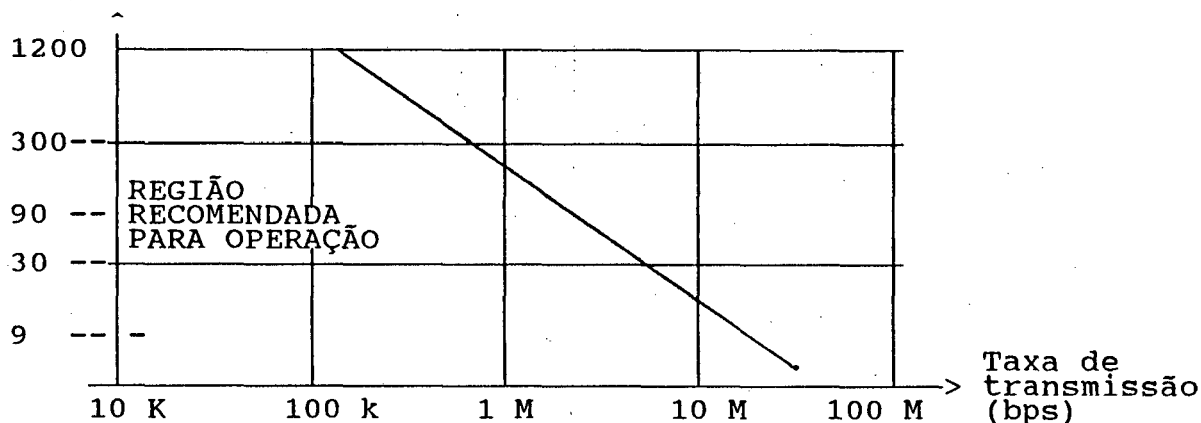


Fig. 5.2 Produto taxa de transmissão x velocidade - par trançado

Como se observa no gráfico, aumentos no alcance da rede só se tornam possíveis mediante a redução na taxa de transmissão. Assim, por exemplo, é possível se transmitir em um quilômetro de par trançado (tipo 24 AWG, sinal ON-OFF NRZ) em taxas da ordem de 300 Kbps. Uma transmissão a 4 Mbps é limitada a distâncias inferiores a 100 metros.

### c) Relação alcance x tipo de codificação do sinal

A capacidade de alcance depende adicionalmente do tipo de codificação do sinal utilizado. Embora o Fieldbus seja originalmente concebido para a operação em banda base, ideal para aplicações localmente dispersas envolvendo distâncias menores que 1 Km, algumas propostas de Fieldbus definem modulação em banda larga permitindo distâncias muito maiores (45 Km para Profibus a 9.6 kbps com 15 km por segmento).

#### d) Relação alcance x repetidores

Embora os três últimos fatores considerados possam ser configurados ou especificados de forma a se aumentar a capacidade de alcance da rede, é no entanto o uso de repetidores a solução mais frequentemente adotada para a extensão da capacidade de alcance da rede e também o número de estações, quando se tem em mente atender a configurações com determinado número de variáveis de controle dispersos em toda a planta.

#### 5.1.2 Número de variáveis do processo

Embora não se possa estabelecer uma relação uniforme, o número de variáveis de controle da aplicação normalmente traduz-se em termos de rede na especificação do número máximo de estações suportadas pelo Fieldbus. Com a redução do custo dos microprocessadores, cada sistema sensor ou atuador pode ser considerado um nó ou estação no Fieldbus. Se o sistema sensor for dedicado (ou seja, só lê uma grandeza a exemplo dos transmissores de pressão muito utilizados na área de processos), e se o processo tem um porte razoável (na ordem de milhares de variáveis) o número potencial de estações em uma planta pode ser portanto substancial e o Fieldbus deve ser especificado de modo a prover a capacidade de interconexão de todas estas estações ao barramento.

A maioria das propostas e implementações de Fieldbus define um número relativamente pequeno de estações por segmento. Barramentos baseados na interface elétrica EIA-RS-485, por exemplo, limitam, pela especificação da norma, o número máximo de estações em 32. Este número é reduzido em parte porque as estações comportam-se como cargas, e os circuitos alimentadores do barramento possuem uma capacidade limitada de fornecimento de corrente elétrica. Por outro lado grande número de estações no Fieldbus exigiria correspondentemente alto nível de corrente para a operação, a qual poderia comprometer a segurança intrínseca do sistema para muitas aplicações sobretudo na área de processos contínuos.

### 5.1.3 Comportamento dinâmico das variáveis do processo

Conforme visto no capítulo 2, cada variável de processo tem um comportamento de variação no tempo que pode estar na ordem de segundos ou milisegundos. As variáveis analógicas lentas, por exemplo, podem ser controladas normalmente sem maiores restrições temporais para o sistema de controle, visto que a velocidade de processamento da arquitetura do dispositivo, com o estado atual da tecnologia, aliado aos tempos de resposta dos barramentos de campo atendem em média às necessidades das aplicações. O controle das variáveis rápidas, tanto analógicas como digitais, cuja variação se encontra na ordem dos milisegundos, impõem no entanto requisitos tempo-real críticos ao Fieldbus. É importante destacar que, de todos os requisitos de aplicação mencionados, este é provavelmente o de maior complexidade para o usuário ao se considerar que ele se desdobra nos seguintes requisitos de comunicação discutidos a seguir:

#### a) Atraso de transferência

Definido como sendo o tempo entre o instante que a tarefa de aplicação na estação transmissora indica ao sistema de comunicação que tem algo para enviar, e a tarefa de aplicação na estação receptora é notificada que a mensagem chegou, o atraso de transferência depende dos seguintes parâmetros:

##### a.1) Tempo de processamento do protocolo de comunicação

O desempenho de um Fieldbus depende fundamentalmente da complexidade dos protocolos da camada de enlace e de aplicação. Atrasos podem ocorrer em uma estação como resultado de várias tarefas que geram informações a serem transmitidas no mesmo canal de transmissão, ou para serem processadas pelos procedimentos de comunicação. No entanto um dos fatores mais determinantes das restrições temporais de utilização do Fieldbus é o método de controle de acesso ao barramento (Medium Access Control - MAC). Os métodos de acesso descentralizados são considerados relativamente de alta complexidade para aplicações de Fieldbus [20], devido a interfaces complexas (custo),

desempenho reduzido devido a funções de gerenciamento extensivas, e devido a dificuldade em se obter um tempo de ciclo constante necessário para as mensagens periódicas (varredura de variáveis de processo). A vantagem das técnicas de controle de acesso centralizado para redes Fieldbus em relação ao acesso descentralizado "token-bus" está na inexistência de falhas associadas à perda ou duplicação do "token", a remediação das quais implica em uma considerável interrupção do processo de transmissão da informação. No mecanismo de acesso mestre-escravo o tamanho do código de programa é relativamente pequeno e pode ser armazenado na memória interna de um microcontrolador, o que interessa particularmente às aplicações em sistemas de produção que requerem estações de tamanho reduzido.

Um forte argumento contra o controle de acesso centralizado é a reduzida disponibilidade sistêmica, visto que a operação de todo o Fieldbus depende em uma estação mestre a qual constitui um ponto singular de falha. Contudo este argumento pode ser superado através de esquemas de redundância tanto em termos de interfaces de rede das estações mestre quanto ao meio físico empregado.

#### a.2) Tempo de transmissão

As altas taxas de transmissão e o pequeno tamanho da informação levam a uma redução do tempo de transmissão da informação no meio físico ( $\text{tempo de transmissão} = \frac{\text{comprimento da info.}}{\text{taxa de transmissão}}$ ). Devido a alta atenuação do sinal de alta frequência este se torna mais sensível ao ruído eletromagnético, portanto aumenta o requisito de EMI com o aumento da taxa de transmissão. Disto resulta maior custo de componentes, cablagem e blindagem. Via de regra uma taxa de transmissão de 1 até 2 Mbps é o suficiente para a maioria das aplicações. Um quadro de 8 bytes a uma taxa de 2 Mbps consome, a título de exemplo, 32 microsegundos de transmissão.

#### a.3) Comprimento do suporte de transmissão

O atraso de transferência do sistema de comunicação Fieldbus depende, além do tempo de transmissão e do protocolo de comunicação, do

comprimento do cabo. Para condutores de cobre o atraso de propagação do sinal é de 10 ns/m. Além disto o uso de repetidores influencia negativamente o atraso de propagação. Para os sistemas de produção na automação da manufatura a distância entre as estações (sistemas sensores, sistemas atuadores e controladores) é relativamente pequena e o atraso de propagação é pouco afetado com o comprimento do suporte de transmissão.

#### b) Tipo de suporte de transmissão

A susceptibilidade aos ruídos industriais pode afetar significativamente o comportamento tempo-real do Fieldbus, considerando que toda a informação mutilada pelas interferências do meio de transmissão deve ser retransmitida. Desta forma o tempo de consulta (ou tempo de polling) de um sistema sensor pode ser elevado a múltiplos do tempo normal. No item 5.2 será examinado alguns procedimentos que o usuário deverá considerar em relação ao tipo de suporte de transmissão para se diminuir a susceptibilidade aos ruídos.

#### c) Número de estações do Fieldbus

Em um sistema mestre-escravo o mestre deverá consultar todas as estações conectadas para a leitura dos dados. Um grande número de estações irá afetar negativamente o comportamento tempo-real do sistema global, devido ao elevado "overhead" no nó mestre.

### 5.2 Condições ambientais

As condições ambientais se traduzem em requisitos de confiabilidade a um Fieldbus. Esta confiabilidade pode ser adaptada ao tipo de ambiente ajustando-se fatores mais tipicamente a nível de camada física, e em menor grau a nível de enlace.

O uso de transmissão digital por si só já melhora a integridade dos dados em relação aos tradicionais sistemas analógicos. Entretanto, em aplicações de sistemas de produção onde são transmitidas

várias mensagens de controle para sistemas atuadores e é realizada a aquisição de dados de sistemas sensores, além de afetar o comportamento tempo-real do Fieldbus como analisado no item anterior, o erro de um bit causado por falhas no meio de transmissão (associadas a EMI por irradiação ou condução) pode ter conseqüências catastróficas para a planta do sistema de produção, contrariamente aos sistemas de comunicação nos níveis mais altos da hierarquia de comunicação. Com isto tem-se dado um destaque especial para a confiabilidade de sistemas Fieldbus, e o usuário não pode desprezar este aspecto na especificação e configuração da rede. Muitas implementações de Fieldbus atualmente em vigor consideram a existência de várias opções de meio físico para atender aos requisitos de confiabilidade da diversidade de aplicações existentes na automação, implementadas para uma mesma camada de enlace.

Das opções de suporte de transmissão, a fibra ótica é notadamente a que assegura total imunidade a ruídos eletromagnéticos e radio-freqüência, o que constitui uma grande vantagem na implementação de um Fieldbus confiável. Outros benefícios associados constituem o seu pequeno peso, pequena atenuação do sinal em todas as freqüências e alto produto taxa de transmissão x alcance. Contudo as dificuldades tecnológicas em se obter um segmento de transmissão multiponto associado a seu custo relativamente alto tem limitado o seu uso em barramentos, e favorecido o seu emprego nas topologias em anel.

O par trançado é considerado um suporte de transmissão com razoável imunidade ao ruído e a melhor solução sob o aspecto de custo para barramentos operando em ambientes locais e internos, sem fontes importantes de ruído impulsivo. O emprego de uma adequada blindagem e isolamento galvânica da estação ao condutor é no entanto imprescindível.

Já os cabos coaxiais, devido a sua inerente estrutura blindada, apresentam uma imunidade ao ruído superior à dos pares trançados, o que favorece a sua utilização em ambientes eletricamente ruidosos. No entanto é mais frequentemente empregado nas redes de nível hierárquico superior, visto que nestes níveis o custo das estações co-

nectadas é dominante [56,57,58].

Visto que as falhas de transmissão no meio físico não podem ser completamente eliminadas, o usuário deverá especificar, com o intuito de melhorar a confiabilidade da informação transmitida, o Fieldbus que ofereça a maior distância de Hamming possível. Um valor maior para a distância de Hamming significa que a probabilidade de reconhecimento de uma falha de transmissão é correspondentemente maior. Alguns Fieldbuses possuem  $H_d=1$ , outros 2 ou 4, e os mais seguros apresentam  $H_d=6$ . Uma outra forma de se aumentar a confiabilidade já em termos de disponibilidade é através do uso de redundância de recursos de meio físico e de estações.

O procedimento de transmissão (tipo de codificação) tem uma substancial influência na susceptibilidade aos ruídos. Através dos procedimentos de transmissão em banda larga consegue-se uma alta imunidade para maiores distâncias. Requer-se no entanto um custo adicional para os circuitos moduladores e demoduladores. Este procedimento como se verifica é pouco adequado, pelo seu alto custo, para sistemas sensores e atuadores simples, embora alguns estudos de viabilidade neste âmbito ainda devam ser realizados.

A codificação banda base possibilita um fácil acoplamento das estações sem custos adicionais de circuitos, o que favorece sua utilização em sistemas Fieldbus. A imunidade a ruídos é no entanto inferior.

O requisito imunidade a ruídos no nível 0 de comunicação deve ser considerado também sob o aspecto de custo. Por esta razão tem sido privilegiado atualmente o uso da interface elétrica EIA-RS-485 com dois condutores. A imunidade a ruídos do RS 485 resulta do fato de que a informação é transmitida simultaneamente aos dois condutores diferenciais, sendo que o receptor recebe apenas o nível diferencial.

Ruídos que surgem na rede de alimentação e afetam diretamente as estações conectadas ao Fieldbus podem ser melhorados com filtros de rede e correspondente blindagem.



Para cabos com alta transmissão de dados parâmetros como amortecimento, capacidade, impedância de transferência são importantes. A isolação galvânica (de potencial) pode ser obtida com optoacoplador ou através de transformadores.

### 5.3 Tamanho da mensagem da aplicação

O requisito tamanho da mensagem da aplicação normalmente é atendido pelas soluções de Fieldbus em vigor, não implicando em maiores restrições para o sistema de comunicação, considerando o pequeno tamanho das informações que são geradas a nível de processos. Além de traduzir-se para uma determinada taxa de transmissão no tempo de transmissão da informação, como analisado no item 5.1.3 (A.2), o tamanho da mensagem da aplicação combinado com o número de variáveis a controlar e a taxa de amostragem permite estimar-se a largura de banda do canal de transmissão, que deverá ser previamente conhecida para efeito de especificação da solução de Fieldbus. Por exemplo, em uma aplicação com 64 sistemas sensores, cada qual monitorando uma variável de processo do tipo ponto flutuante com 3 byte, com uma taxa de amostragem de 10 amostras/seg., irá requerer uma taxa de transferência de 16 kbps ( $64 \times 3 \times 8 \times 10 = 15360$  bit). Incluindo as variáveis de comunicação dos protocolos a taxa de transmissão poderá alcançar por exemplo 70 kbps para o Profibus e 45 kbps para o Bitbus. Considera-se para esta última estimativa a existência de 6 byte de variáveis de comunicação para a transmissão dos três byte de dados do sistema sensor ( $64 \times (3+6) \times 8 \times 10 / 1024 = 45$  kpbs).

### 5.4 Necessidade de gerenciamento remoto de programas (carga, partida e parada), eventos e acesso remoto à variáveis do processo

Este requisito tem implicações na funcionalidade da camada de aplicação a qual deverá, utilizando-se da funcionalidade das camadas inferiores, prover ao usuário serviços de comunicação específicos para se realizar tais operações de forma transparente e confortável (ou seja, isolando o usuário de aplicação dos detalhes e complexidade de operação do sistema de comunicação Fieldbus). Estes e outros aspectos serão oportunamente discutidos no capítulo seguinte.

## 6. A CAMADA DE APLICAÇÃO DE FIELDBUS - ASPECTOS DA INTEGRAÇÃO LÓGICA

Fundamental para o usuário de Fieldbus na integração (comunicação) de processos de aplicação distribuídos, por exemplo, em vários sistemas sensores, é a camada de aplicação e a correspondente interface da camada de aplicação (ALI), por constituírem ambas a interface entre o programa de aplicação e o sistema de comunicação.

Este capítulo, baseado na norma DIN 19245 T2 [66], objetiva verificar os conceitos e apontar os aspectos relacionados a estas camadas de interesse imediato para um usuário de Fieldbus, fornecendo-lhe uma base sobre a qual possa nortear a construção da aplicação. Este projeto obedece a passos bem definidos sendo cada qual contemplado com um item neste capítulo. Busca-se adicionalmente ilustrar a realização em Fieldbus do último requisito relacionado no capítulo anterior, o gerenciamento remoto de programas e eventos, a carga de programas e o acesso remoto a variáveis de processo.

### 6.1 Estabelecimento do Dispositivo de Campo Virtual (VFD-Virtual Field Device)

O primeiro e importante passo do usuário para o projeto da aplicação é o estabelecimento do dispositivo de campo virtual.

Um dispositivo virtual, visto pelo usuário, representa genericamente uma abstração de uma classe de dispositivos reais. Exemplos de dispositivos virtuais podem ser memórias virtuais, terminais (virtuais) e naturalmente dispositivos de campo. Cada classe de dispositivos virtuais é diferenciada por um conjunto de características que permitem a troca mútua de informação numa área de comunicação não homogênea.

Em Profibus, o modelo do dispositivo de campo virtual (VFD) representa a parte de um processo de aplicação real, constituída por objetos processos, que pode ser visível e gerenciável por meio dos serviços Profibus. O modelo de comunicação Profibus suporta, portanto, o modo de trabalho orientado a objeto, de forma que um processo

de aplicação trabalha com objetos processos, como por exemplo variáveis, programas e alarmes, os quais contém atributos, regras, e descrições de operações sobre objetos. Cita-se a título de exemplo a partida de um programa. "Start" é a operação aplicada sobre o objeto "programa". Da mesma forma "Read" é uma possível operação a ser aplicada ao objeto "Variável".

Para a implementação prática do conceito de dispositivo de campo virtual, sem o que a operação do Fieldbus não é possível, o usuário deverá identificar quais são os objetos processos existentes em seu processo de aplicação, associá-los a objetos de comunicação e subsequentemente subscreve-los juntamente com sua descrição no chamado dicionário de objetos (OD), conforme será analisado nos dois subitens seguintes. A partir de então, o processo de aplicação que coloca a funcionalidade do seu VFD, por meio de um serviço, a disposição de um processo de aplicação remoto passa a ser chamado de Servidor. Contrariamente, ao processo de aplicação (AP) que utiliza a funcionalidade de um VFD de um processo remoto, denomina-se Cliente. O cliente é também um processo de aplicação que executa um pedido de serviço Profibus, sendo que o AP servidor executa este serviço e possivelmente envia uma resposta (nos serviços confirmados). A Interface da camada de aplicação (ALI) define as regras do cliente. As regras do servidor são descritas com a ajuda do modelo do dispositivo de campo virtual, de acordo com a figura 6.1.

#### 6.1.1 Objetos de comunicação

Ao ser inserido no dicionário de objetos, o objeto processo torna-se um objeto de comunicação e é indiretamente tornado público para outros processos de aplicação. Os objetos de comunicação representam objetos processos existentes tornando possível operações sobre estes objetos via serviços Profibus. Parâmetros dos sistemas sensores, valores de medição de grandezas de processo e valores de atuação são representados no Fieldbus pelo objeto de comunicação Variável (Variável simples, Array e Record). Mensagens e alarmes são representados pelo objeto Evento. A representação de áreas de memória contíguas resulta no objeto Domínio. Diagramas de fluxo de processo por exemplo podem ser mapeados em objetos Lista de variáveis. Este com-

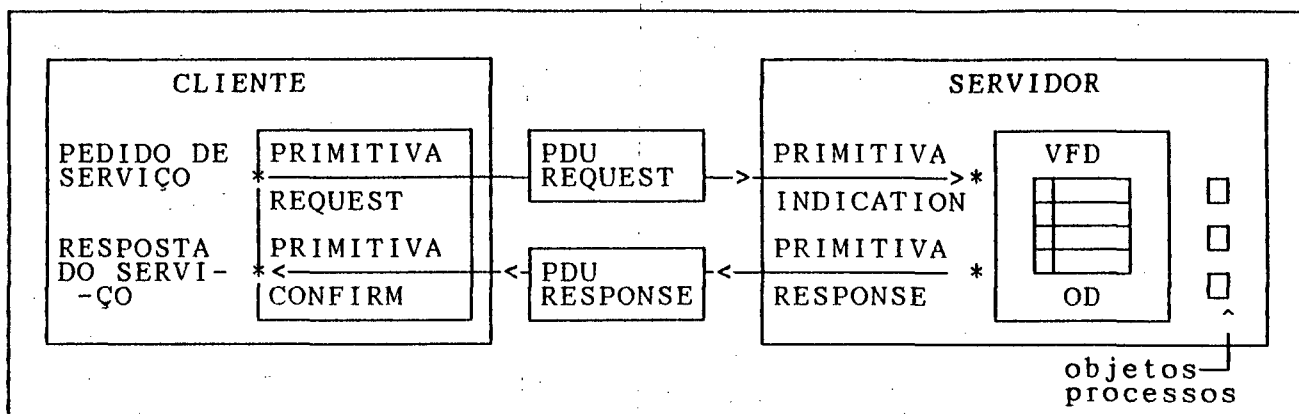


Fig. 6.1 Modelo Cliente/Servidor no Profibus

põe-se de objetos variáveis já definidos estaticamente no dicionário na fase de planejamento (projeto) do sistema. A representação de um programa executável juntamente com blocos de dados se faz através do objeto Invocação de Programa. Este constitui-se da combinação de domínios em uma única unidade que contém o programa executável e parâmetros associados.

O acesso aos objetos de comunicação da aplicação que são visíveis ao sistema de comunicação pode ser efetuado através de um endereço simples denominado de índice do objeto (endereçamento lógico) ou então através do próprio nome do objeto (endereçamento simbólico).

Os objetos de comunicação podem ser declarados no dicionário de objetos de um parceiro de comunicação durante a fase de projeto, quando o VFD não está pronto para se comunicar, ou dinamicamente durante a fase de operação. Verifica-se que a maioria dos objetos no nível 0 são raramente modificados, geralmente em situações de falha operacional. Por um outro lado a definição de objetos para grandes quantidades de dados através do Fieldbus é um processo relativamente lento, comprometendo a eficiência da transmissão. Os objetos Variável, Evento e Domínio podem ser definidos apenas durante a fase de projeto (descrição estática), ao passo que as listas de variáveis e Invocações de Programa podem ser definidos na etapa de execução (descrição dinâmica). A descrição de um objeto consiste, dentre outros elementos, do índice, do nome, do tipo de objeto e facultativamente do comprimento do objeto. As descrições de objetos podem ser lidas de, ou escritas no dicionário de objetos de parceiros de co-

municação uma ou várias vezes mediante o emprego de serviços FMS apropriados (ver item 6.2.6). A real troca de dados do usuário somente requer a indicação do índice na chamada de serviço.

### 6.1.2 Dicionário de Objetos

Os objetos de comunicação deverão ser inscritos no OD conforme a estrutura ilustrada na figura 6.2.

índice	Dicionário de Objetos
0	Cabeçalho (OD-Header)
1 . i	DICIONÁRIO DE TIPOS DE DADOS (ST-OD) (objetos tipos de dados)
K . N	DICIONÁRIO ESTÁTICO (S-OD) (objetos variáveis, domínios e eventos)
P . T	DICIONÁRIO DE LISTA DE VARIÁVEIS (DV-OD) (DINÂMICO) (objetos lista de variáveis)
V . Z	DICIONÁRIO DE INVOCAÇÕES DE PROGRAMA (DP-OD) (DINÂMICO) (objetos invocação de programa)

Fig. 6.2 Estrutura do dicionário de objetos

A estrutura do dicionário de objetos está descrita no seu cabeçalho, em Profibus denominado de OD- header, o qual deve ser armazenado com o índice 0 e pode ser lido pelos parceiros de comunicação. O primeiro subdicionário de objetos é o dicionário estático de tipos de dados (ST-OD). Este contém a descrição dos tipos de dados e a descrição das estruturas de dados dos objetos variáveis inscritos no dicionário estático (S-OD). Esta descrição está estaticamente definida e não é passível de alterações. Adicionalmente pode ser lida remotamente pelo parceiro de comunicação mediante serviços apropriados. O subdicionário estático de tipos deverá ser iniciado com o índice 1, de forma que as demais descrições de objetos identifiquem-se com índices crescentes. O número de índices existentes neste subdicionário é fixado no cabeçalho. Distingue-se ainda dois tipos de subdicionários de tipos. No primeiro poderão ser inscritos a partir

do índice 1 e conforme a necessidade os seguintes tipos padrões definidos pela norma Profibus:

- Boolean
- Integer (8,16,32 bit)
- Unsigned (8,16,32 bit)
- Floating-Point (IEEE Standard 754)
- Octect-string
- Visible-string (ISO 646)
- Date
- Time of Day e
- Time difference

O segundo tipo deverá estar disposto imediatamente subjacente e conter a descrição de tipos e estruturas definidas pelo usuário.

O próximo subdicionário, o subdicionário estático de objetos, deverá conter a descrição dos objetos Variável (simples, array e record), Evento e Domínio. A definição dos objetos subscritos neste subdicionário deve ser efetuada apenas na etapa de planejamento. Cada entrada descritora de um objeto contém informação sobre o nome, o índice, a classe do objeto, o comprimento e os direitos de acesso a escrita/leitura dentre outros.

O subdicionário Lista de Variáveis Dinâmica (DV-OD) contém a descrição dos objetos Lista de variáveis. Esta descrição contém além dos atributos já mencionados para o caso dos objetos variáveis, uma relação dos índices dos objetos variável inscritos no subdicionário estático que compõem a lista de variável. Cada objeto Lista de variável ocupa uma entrada do subdicionário e é identificado pelo correspondente índice. Através deste pode-se com uma única operação de escrita ou leitura acessar a lista de variáveis, evitando-se o "overhead" que se incorreria caso fosse necessário o endereçamento individual de cada objeto da lista.

O subdicionário Invocação de Programa dinâmico contém a descrição dos objetos Invocação de Programa, os quais podem estar pré-definidos (definição estática-fase de configuração) ou serem dinamicamente inscritos (descrição dinâmica-fase de ope-

ração) no subdicionário mediante serviços FMS.

Da mesma forma que o subdicionário Lista de Variáveis, a descrição dos objetos Invocação de Programa compõe-se de informações mais gerais como por exemplo direitos de acesso e identificação da classe de objeto, dentre outros atributos, além de uma relação de índices de objetos Domínio, pré-definidos no subdicionário estático, que compõem o objeto Invocação de Programa.

## 6.2 Definição dos serviços FMS e das funções da ALI

Em termos de implementação prática adota-se, por questões de eficiência operacional e também para a redução do custo e complexidade dos dispositivos, apenas um subconjunto dos serviços FMS na funcionalidade dos sistemas sensores. Com isto o usuário deverá verificar, a partir dos objetos de comunicação definidos no item 6.1, quais são os serviços FMS, necessários à execução da aplicação, que deverão estar disponíveis na funcionalidade do sistema sensor.

A figura 6.3 ilustra para cada tipo de objeto de comunicação quais os serviços que podem sobre ele atuar.

OBJETO	SERVIÇOS
VARIÁVEL	PUT-OD, READ, WRITE, READ-WITH-TYPE, WRITE-WITH-TYPE, INFORMATION-REPORT, INFORMATION-REPORT-WITH-TYPE
DOMÍNIO	PUT-OD, CREATE-PI, DELETE-PI, INITIATE-DOWNLOAD-SEQ., DOWNLOAD-SEGMENT, TERMINATE-DOWNLOAD-SEQ., REQUEST-DOMAIN-DOWNLOAD, INITIATE-UPLOAD-SEQ., UPLOAD-SEGMENT, TERMINATE-UPLOAD-SEQUENCE, REQUEST-DOMAIN-UPLOAD.
EVENTO	EVENT-NOTIFICATION, EVENT-NOTIFICATION-WITH-TYPE, ACKNOWLEDGE-EVENT-NOTIFICATION, ALTER-EVENT-CONDITION-MONITORING, PUT-OD
INVOCACÃO DE PROGRAMA	PUT-OD, CREATE-PROGRAM-INVOCATION, DELETE-PROGRAM-INVOCATION, START, STOP, RESUME, RESET, KILL
LISTA DE VARIÁVEIS	PUT-OD, READ, WRITE, WRITE-WITH-TYPE, READ-WITH-TYPE, INFORMATION-REPORT, INFORMATION-REPORT-WITH-TYPE, DEFINE-VARIABLE-LIST, DELETE-VARIABLE-LIST

Fig. 6.3 Relação objeto/serviços de comunicação.

Na operação do sistema o programa de aplicação não irá interagir di-

retamente com os serviços da camada de aplicação, mas com uma biblioteca de funções fornecidas pela Interface da Camada de Aplicação (ALI). Desta forma o usuário ao ativar uma determinada função da ALI fará com que esta requisi-te o serviço correspondente à camada de aplicação, como ilustrado na figura 6.4.

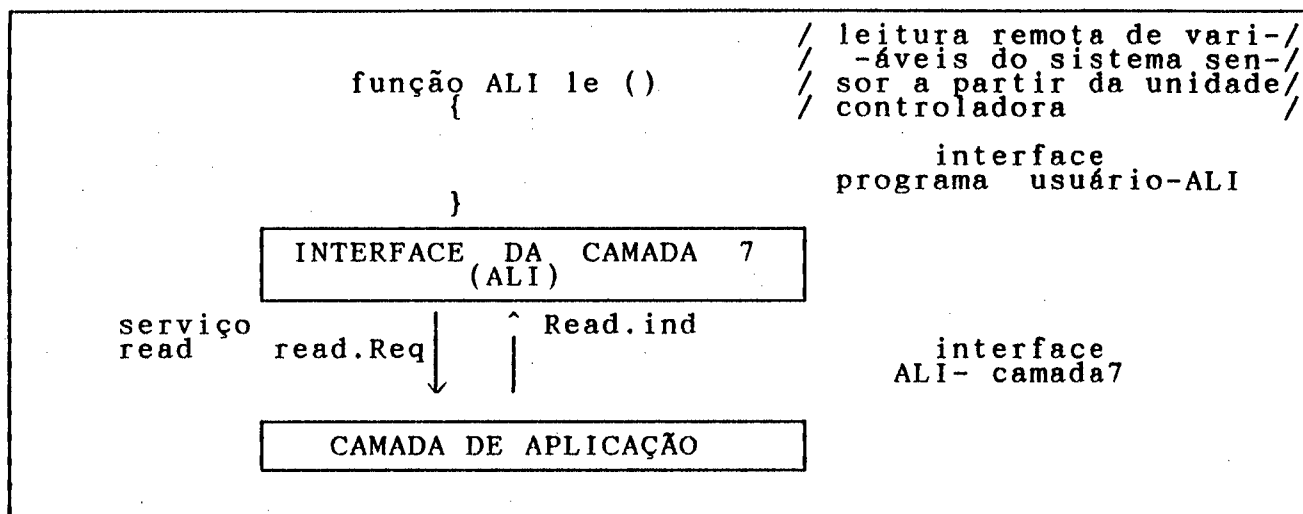


Fig. 6.4 - Exemplo de mapeamento das funções ALI em serviços FMS

Uma dada função da ALI poderá ativar um ou mais serviços FMS dependendo da implementação. Além de constituir uma interface específica para o dispositivo, a ALI é também responsável pelo mapeamento dos objetos processos existentes localmente em objetos conhecidos no Profibus (objetos de comunicação).

O padrão Profibus define tres grupos funcionais de serviços:

A) O primeiro, denominado de serviços de aplicação, consiste de todos os serviços com os quais os objetos de comunicação da aplicação podem ser acessados. Com os serviços de aplicação pode-se satisfazer o último requisito da aplicação mencionado no capítulo anterior, a necessidade de acesso remoto a variáveis, gerenciamento de eventos e programas. Exemplos destes são os serviços "Read" para a leitura ou "Write" para escrita de variáveis, ou o serviço "Stop" para o término da execução de uma Invocação de Programa e ainda "Event\_notification" para a notificação da ocorrência de um evento remoto;



b) Os serviços do segundo grupo (serviços de administração), também utilizados pelo usuário apenas se aplicam a objetos ou a relações de comunicação do Dispositivo de campo virtual VFD. Um objeto pode ser a descrição de um objeto de comunicação no dicionário de objetos por exemplo. Um possível serviço a atuar sobre este objeto é "Get-OD" para a leitura de uma entrada do dicionário de objetos. Para iniciar e encerrar uma conexão de uma relação de comunicação utiliza-se os serviços Initiate e Abort;

c) O terceiro grupo (serviços de gerenciamento de rede) é constituído de serviços para a manutenção da operação técnica da rede. São serviços que definem por exemplo, o tamanho dos buffers de mensagens, reinicializam camadas ou carregam e modificam a lista das possíveis relações de comunicação.

A seguir discute-se os serviços de aplicação e administração pela importância que assumem no projeto da aplicação.

#### 6.2.1 Serviços de gerenciamento remoto de eventos (aplicação)

A classe Gerenciamento de eventos oferece basicamente serviços que enviam mensagens urgentes de um dispositivo para outro. A definição das condições causadoras do evento (por exemplo, a presença de um sinal de nível alto na entrada de uma determinada porta de E/S deverá acionar um evento, chamando o serviço "Event\_notification" para sinalizar ao controlador da aplicação falha funcional do sistema sensor) é tarefa do usuário. Sendo a condição satisfeita, o processo de aplicação chama o serviço Event-notification que transmite para o parceiro de comunicação a informação do evento. A transmissão do tipo é realizada com o serviço Evento-notification-with-type. Com o serviço Alter-event-condition-monitoring pode-se bloquear ou liberar a transmissão pelo Fieldbus da informação de evento. Para se responder a uma mensagem evento dispõe o FMS do serviço Acknowledge-event-Notification.

### 6.2.2 Serviços de gerenciamento remoto de Programas (Invocação de programa)(aplicação)

Permite agregar-se domínios para formar um programa executável, além de controlar a sua execução ou mesmo apagá-lo. Este programa é acessível pelo índice. Com os serviços Start e Stop pode se iniciar e suspender respectivamente a execução de um programa. O serviço Resume tem por função ativar novamente um programa com execução suspensa. Para se reinicializar um programa suspenso tem-se a disposição o serviço Reset. Kill leva o programa do seu estado momentâneo para o estado em que este não pode mais ser executado, mas apenas apagado (estado "unrunnable"). Com o serviço Create-program-invocation pode-se dinamicamente agregar domínios, descritos no dicionário de objetos, para formar um programa executável. Este passa a ser acessável através de um endereço lógico. Para se apagar o objeto Invocação de Programa utiliza-se o serviço Delete-program-invocation. Esta classe oferece serviços que permitem a criação, controle de execução e deleção de programas executáveis.

### 6.2.3 Serviços de Acesso a Domínios (aplicação)

A classe de serviços Acesso a Domínios (Domain Access) contém serviços para a transferência de blocos de dados de domínios entre o chamado cliente e o servidor.

Com o serviço Initiate-Download-Sequence o cliente informa ao servidor o início do transporte de dados (parâmetros, programas..) para um domínio deste. Com o serviço Download-segment o servidor solicita ao cliente o transporte de um segmento de dados. Na PDU de Resposta a este serviço, o cliente envia os dados juntamente com um sinalizador que acusa a existência ou não de mais dados para completar o "download". Não havendo mais dados, o servidor informa com o serviço Terminate-Download-segment que a carga de dados em seu domínio está encerrada. A requisição de uma operação de Download é realizada pelo servidor com o serviço Request-Domain-Download. O transporte de dados de um domínio do servidor para o cliente obedece à mesma lógica descrita para o "download", porém utilizando-se os serviços Initiate-Upload-Sequence, Upload-segment, Terminate-Upload-Sequence

e Request-Domain-Upload.

#### 6.2.4 Serviço de Acesso a Variável (aplicação)

A classe de serviço Acesso a Variável oferece serviços que permitem o acesso a variáveis simples, records, arrays e Lista de variáveis. Além disso os objetos de comunicação do tipo Variable-list podem ser dinamicamente criados ou apagados. Os serviços Read e Write possibilitam a leitura e escrita de variáveis, respectivamente, descritas no dicionário de objetos. Para a leitura e escrita da variável objeto com a sua descrição de tipo devem ser utilizados os serviços Read-with-Type e Write-with-Type. Caso haja a necessidade de se transmitir um objeto variável a todas as estações conectadas ao Fieldbus, tem-se a disposição o serviço Information-Report (Broadcast). Com os serviços Define-Variable-List pode-se criar Listas de variáveis em tempo de execução (dinamicamente) constituídas de variáveis objetos definidas no dicionário de objeto estático. Delete-Variable-List apaga o objeto lista de variável especificado.

#### 6.2.5 Serviços de Gerenciamento de Contexto (administração)

Os serviços de gerenciamento de contexto servem para o gerenciamento de conexões de aplicação. O contexto que é necessário para o estabelecimento de uma conexão consiste em todos os acordos entre os parceiros de comunicação a respeito de uma relação de comunicação. Esta informação está armazenada na descrição do Dicionário de Objetos e na Lista de Relações de Comunicação da camada 7 (item 6.3). Através do serviço Initiate pode-se estabelecer uma conexão. Todos os demais serviços orientados a conexão devem ser precedidos inicialmente pela etapa de estabelecimento de conexão utilizando este serviço. O encerramento de uma conexão existente entre dois parceiros de comunicação pode ser efetuado pelo serviço Abort, de forma que para se ter novamente a comunicação dos parceiros em serviços orientados a conexão faz-se necessário reestabelecer a conexão com o serviço Initiate. Mediante o serviço Reject o protocolo FMS rejeita uma Unidade de dados de protocolo de recepção, na ocorrência de falha do protocolo de transmissão, em situação de erro na chamada ou res-

posta de um serviço, na indisponibilidade do serviço ou ainda na violação do contexto acordado.

#### 6.2.6 Serviços de Gerenciamento do Dicionário de Objetos (administração).

Os serviços de gerenciamento do dicionário de objetos permitem a leitura e escrita do dicionário de objeto fonte do parceiro de comunicação. Se a descrição dos objetos de comunicação em vigor são modificadas, os parceiros de comunicação envolvidos devem ser informados, o que pode ser feito encerrando as conexões envolvidas. Com o serviço Get-OD pode-se ler uma ou mais descrições de objeto do dicionário de objetos. Para isto deve-se especificar como parâmetro da chamada do serviço o índice ou nome da descrição de objeto desejada ou pode-se, opcionalmente, ler todas as descrições de objeto. O serviço Put-OD possibilita a escrita de uma ou mais descrições de objetos no dicionário de objetos do parceiro de comunicação remoto. O número de descrições de objeto a serem transportados com os serviços desta classe depende do comprimento de Pdu disponível. O acesso a escrita de um OD deve ser precedido do serviço Initiate-Put-OD e encerrado com o serviço Terminate-Put-OD .

#### 6.2.7 Serviço de Suporte ao Dispositivo de Campo Virtual - VFD (administração)

Os serviços VFD-Support permitem o intercâmbio de informações sobre os dispositivos conectados ao Fieldbus. O serviço Status requer a transmissão do estado lógico e físico de um dispositivo. A transmissão do estado do dispositivo (sem que tenha sido requisitado pelo parceiro de comunicação remoto) pode ser realizada utilizando-se o serviço Unsolicited-Status. Com o serviço Identify pode-se requisitar informações para a identificação de um VFD.

### 6.3 Configuração do Fieldbus

Uma importante atribuição do usuário no sentido de se operacionalizar a comunicação entre os processos de aplicação é o projeto das relações de comunicação, a ser realizada na fase de configuração do Fieldbus.

Uma relação de comunicação entre dois ou mais processos de aplicação é completamente definida por um conjunto de parâmetros de comunicação que descrevem as características da comunicação, tais como o tipo de parceiros de comunicação (mestre-mestre ou mestre-escravo), a periodicidade (cíclica ou acíclica), o endereçamento do processo de aplicação local e remoto e o tamanho das filas de transmissão e recepção. Cada relação de comunicação que o processo de aplicação local mantém com outros processos remotos ocupa uma entrada da chamada Lista de Relações de Comunicação (CRL), identificada por um índice (CR), e pode ser acessada para leitura ou escrita com serviços oferecidos pela camada de gerência (FMA7). Com o serviço Read-CRL pode se ler uma entrada da CRL especificando-se na chamada do serviço a CR correspondente a entrada. A carga na CRL dos parâmetros de comunicação configurados pelo usuário deverá ser realizada com a seqüência de serviços Initiate-Load-CRL, Load-CRL e Terminate-Load-CRL. Cada relação de comunicação constitui, na gíria de redes de comunicação, canais lógicos ou ainda conexões pré-projetadas. Antes que se possa transferir propriamente informações pelo barramento, faz-se necessário ativar/habilitar tais conexões, o que pode ser realizado com o serviço Initiate. Após o fim da fase de transferência de dados, tais conexões poderão ser encerradas com o serviço Abort.

A Lista de Relações de Comunicação está definida em termos de arquitetura de Fieldbus na camada 7. Além de carregar a CRL, há também um conjunto de parâmetros de comunicação a nível de camada de enlace (p. Ex., taxa de transmissão, tempo de circulação do "token", tempo de espera pela confirmação dos serviços) a serem carregados antes da operação do barramento, utilizando-se para isto os serviços de gerência.

Visto que estamos primariamente interessados na programação da apli-

cação, os aspectos específicos inerentes ao projeto das relações de comunicação e à configuração dos parâmetros de comunicação da camada de enlace não serão aqui considerados, para o que sugere-se a consulta das referências [65,66].

## 7. EXEMPLO DE APLICAÇÃO

Com o intuito de ilustrar em maior grau de profundidade o projeto da aplicação integrada por meio de um Fieldbus (Profibus), analisado no capítulo precedente, será primeiramente apresentado um exemplo hipotético de uma aplicação constituída de tres processos, conforme apresentado na fig. 7.1 e analisado a seguir.

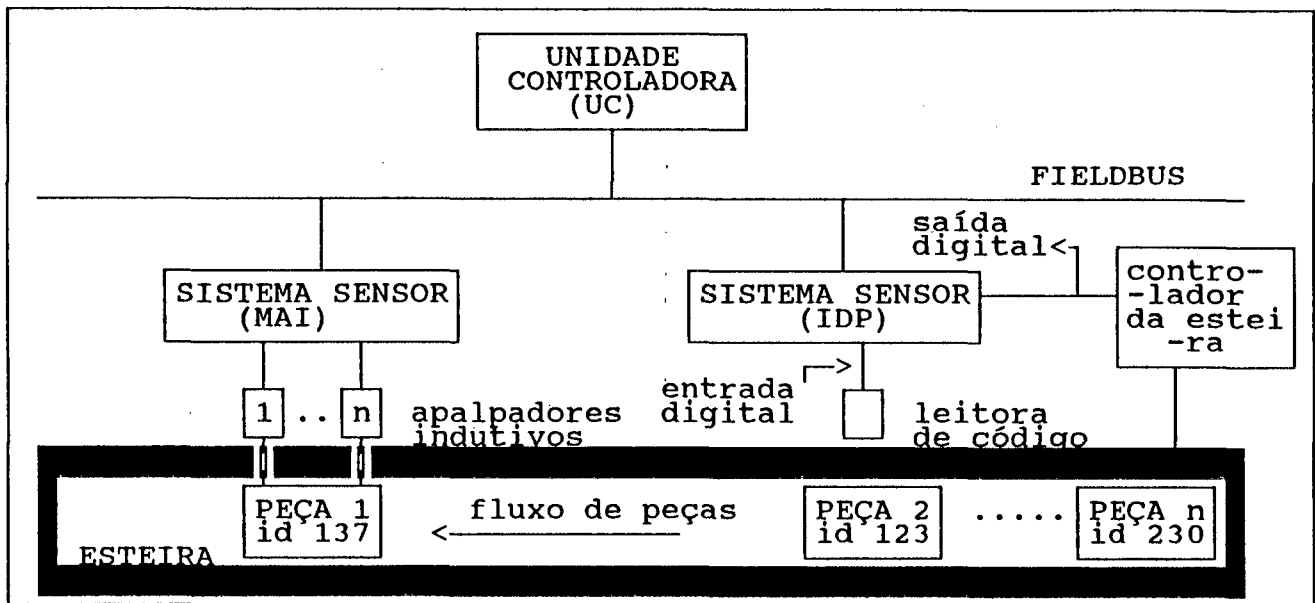


Fig. 7.1 Exemplo de aplicação

### 7.1 Configuração do sistema

A configuração ilustrada pode ser utilizada para a supervisão das características relevantes de peças de trabalho no controle estatístico do processo (CEP). Com este objetivo tais características são primeiramente adquiridas e posteriormente processadas estatisticamente de forma que ao se ultrapassar determinados valores limites de tolerância o processo deva sofrer uma intervenção corretiva. As informações relevantes sobre o processo são apresentadas a título de exemplo na forma de cartas estatísticas, as quais são armazenadas e transmitidas para o sistema de garantia da qualidade.

Exemplos de sistemas sensores fundamentais para esta aplicação são o terminal multiplexador de apalpadores indutivos (MAI), o qual dedica-se à leitura de um número variado de transdutores de deslocamento

para a aquisição das características geométricas e dimensionais de peças de trabalho, e o terminal identificador de peças (IDP), cuja função consiste na leitura do código de identificação de cada peça. A exceção da Unidade de Tratamento de Sinal (UTS), ambos os terminais possuem a arquitetura representada na fig. 7.2. No microcontrolador a comunicação entre a

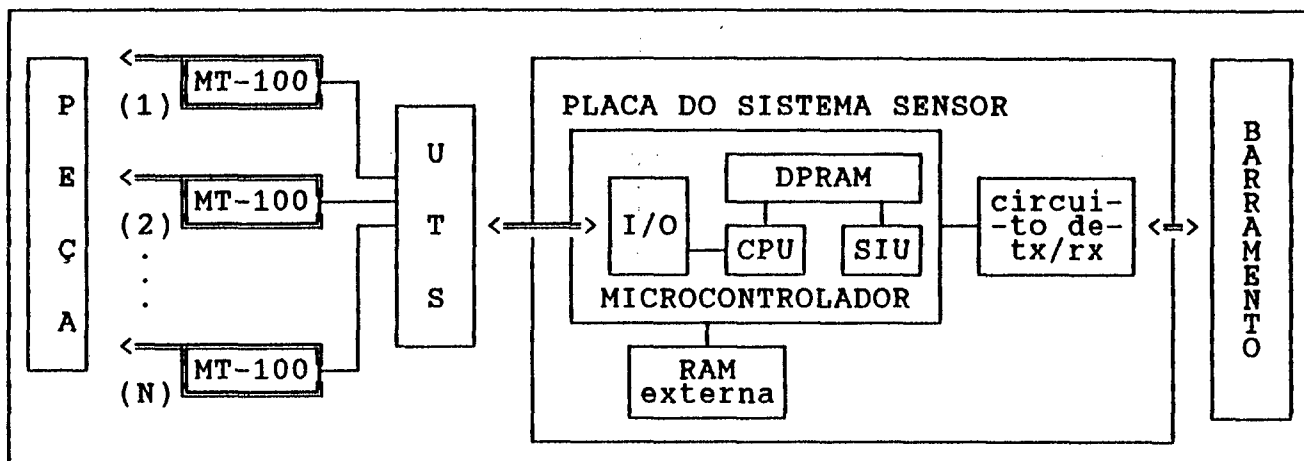


Fig. 7.2 - Configuração básica do terminal multiplexador

CPU e a interface serial (SIU) se dá via memória DP-RAM e a ligação da interface serial ao barramento se faz através de circuitos transceptores (TxD/RxD) com isolamento por optoacoplamento. A diferença entre os dois sistemas sensores reside basicamente na interface com o processo. Como interface para o processo o sistema sensor MAI possui essencialmente uma interface analógica com conversor A/D e multiplexador para as entradas, por exemplo, nas faixas de 0 a 20 mA, 0 a + 5V, 0 a 10 V ou +\ - 10 V. Com isto podem ser acoplados todos os transdutores que fornecem o sinal de medição em corrente ou tensão através de uma cadeia de pré-amplificação realizada pela UTS.

Já o IDP possui uma Unidade de Tratamento do Sinal que converte a leitura do código de identificação de peça (id) em uma palavra digital de 8 bits que informa ao sistema, dentre 256 possibilidades, qual o tipo de peça a ser medida. Adicionalmente o IDP possui uma saída digital que opera como comando para o controle da esteira. A escrita nesta saída de qualquer valor diferente de zero fará com que a esteira se desloque de uma



distância fixa e pré-ajustada e pare automaticamente de forma que uma peça fique posicionada frente ao MAI para medição e a sua antecedente esteja concomitantemente diante do IDP para reconhecimento. A Unidade Controladora (UC) para o exemplo em questão consiste de um micro-computador compatível com IBM-PC ou um computador industrial (VME, STD, Multibus etc.) que se interliga ao Fieldbus através de uma interface de "hardware" diretamente acoplada ao seu barramento. A interação entre Unidade Controladora e os dois sistemas sensores conectados ao Fieldbus ocorre ciclicamente, para a aquisição dos valores dos transdutores.

## 7.2 Operação do sistema

A configuração opera na instalação de produção desenvolvendo o ciclo representado na Fig. 7.3. Após a partida do sistema, envolvendo basicamente as operações de inicialização das variáveis de comunicação e da aplicação, a Unidade Controladora estará apta a trocar informações com os demais sistemas sensores e portanto poderá executar a aplicação. Inicialmente a Unidade Controladora requisita ao IDP o código de identificação da peça presente sobre a esteira. O valor 0 indica a inexistência de peças sobre a esteira e portanto o fim do lote de peças a inspecionar. Subseqüentemente um novo comando enviado da Unidade Controladora ao IDP solicita o acionamento da esteira para se efetuar o posicionamento da peça recém identificada frente ao MAI. Para se prosseguir a operação, deverá ser considerado o tipo de peça a ser medida. Se esta constituir um novo tipo de peça em relação a precedente (ou seja, um outro código de identificação), ou ainda se for a primeira peça do lote) será necessário se proceder de antemão a carga remota da Unidade Controladora para o MAI dos programas de calibração e de medição, conforme o tipo de peça, juntamente com parâmetros operacionais do sistema sensor tais como "offset", fatores de correção, fator de amplificação, faixa de operação, tensão de alimentação ou mesmo lista de parâmetros para a interpolação. Em seguida a Unidade Controladora dirige um novo comando ao Sistema sensor MAI para que este realize a calibração antes do início da medição, para somente então solicitar o pedido de medição da peça. Estando a peça mal posicionada para a medição, o que pode ser

facilmente identificado localmente no MAI através da leitura de um transdutor específico para detecção de erro de posição da peça, o MAI envia um alarme para a Unidade Controladora, a qual paralisa o processo durante um intervalo de tempo para a intervenção corretiva e manual na aplicação. Uma vez recebida pela Unidade Controladora, a medida da peça é processada estatisticamente, com o que encerra-se um ciclo completo de operação do sistema.

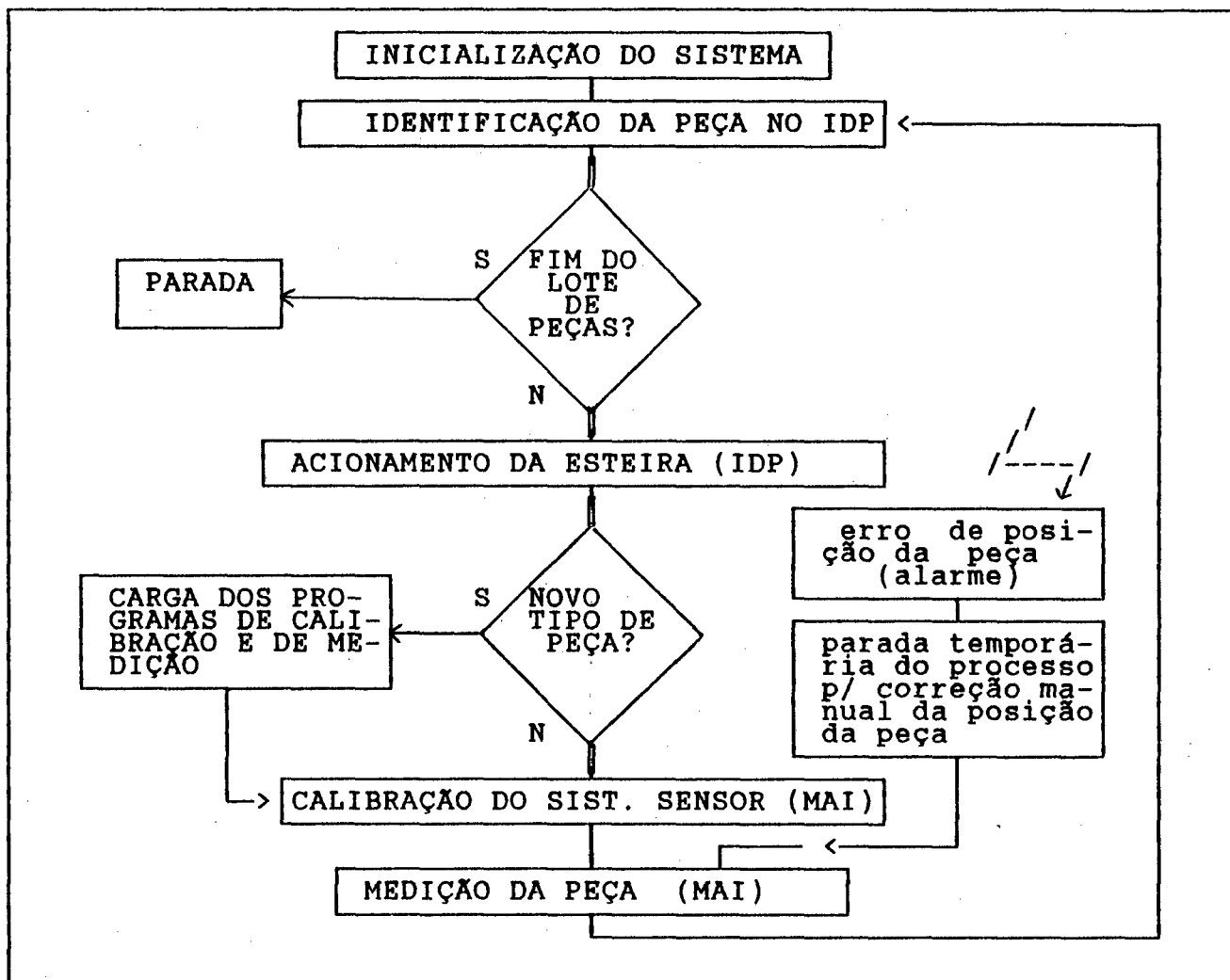


Fig. 7.3 Ciclo de operações do sistema

A seguir será ilustrada a construção desta aplicação no ambiente aberto Fieldbus conforme os passos definidos no capítulo anterior.

### 7.3 Estabelecimento do Dispositivo de Campo Virtual

Com base na descrição da aplicação disposta anteriormente, os sistemas sensores deverão apresentar os seguintes objetos processos:

#### - Sistema sensor IDP

1. Um objeto Variável simples "código", para leitura (entrada digital) a partir da Unidade Controladora do código de identificação da peça;
2. Um objeto Variável simples "posição", para a escrita na saída digital do IDP do comando de acionamento da esteira (posicionamento da peça a ser medida);

#### - Sistema sensor MAI

3. Um objeto Domínio " dom\_cal ", destinado a armazenar o programa de calibração e parâmetros operacionais do sistema sensor associados à calibração;
4. Um objeto Domínio " dom\_med ", designado para armazenar o programa de medição e parâmetros operacionais correspondentes;
5. Um objeto Invocação de programa " calibração", para calibração do MAI antes do início da operação de medição;
6. Um objeto Invocação de programa "medição", para iniciar a operação de medição no MAI;
7. Um objeto Variável Array "medida", para ler e armazenar as medições realizadas pelo MAI;
8. Um objeto Evento "Erro\_posição", que indicará à Unidade Controladora o eventual erro de posicionamento da peça a ser medida.

Os dicionários de objetos dos sistemas sensores, contendo o cabeçalho e uma entrada para cada um destes objetos, estão respectivamente

te ilustrados nas figs. 7.4 e 7.5. A Unidade Controladora deverá conter uma cópia destes objetos em seu OD. Particularmente a inicialização do dicionário de objetos do MAI será discutida na seqüência segundo sua organização em subdicionários. A construção do dicionário de objetos fonte do sistema sensor IDP segue a mesma sistemática e portanto dispensa aqui maiores considerações.

CABEÇALHO DO DICIONÁRIO DE OBJETOS (HEADER)											
object index	rom/ram flag	name length	access protection supported	version OD							
0	false	16	true	02							
local addr. header	ST_OD length	local address ST_OD									
&cab_addr	1	&st_OD_addr(valor da variável apontador)									
first index S_OD	S_OD length	local address S_OD									
2	4	&s_OD_addr (valor da variável apontador)									
first index DV_OD	DV_OD length	local address DV-OD									
0	0	0									
first index DP_OD	DP-OD length	local address DP-OD									
6	2	&dp_OD_addr(valor da variável apontador)									
SUBDICIONARIO ESTÁTICO DE TIPOS (ST-OD)											
object index	object code	descrição									
01	data -type	Integer8									
SUBDICIONÁRIO ESTÁTICO DE OBJETOS (S-OD)											
object index	object code	pass word	access group	access rights	Data ty-pe index	Length	local address	number of elements	array name		
02	array	43	40	R,Rg	01	01	&med_ad	10	medida		
object index	object code	pass word	access group	access rights	index event	Length	enabled		event name		
03	event	30	0f	W	01	01	true		erro_posição		
object index	object code	pass word	access group	access rights	Max octets	Upload State	local address	domain state	domain name	counter	
04	domain	35	50	W, Wg	4096	inexiste	&p_cal	existent	dom_cal	0	
05	domain	35	50	W, Wg	4096	inexiste	&p_med	existent	dom_med	0	
SUBDICIONÁRIO DINÂMICO INVOCAÇÃO DE PROGRAMAS (DP-OD)											
object index	object code	pass word	access group	access rights	number of domains	dele-table	local address	reus-able	Pi state	Pi name	list of domain index
06	Pi	86	50	Sg,Hg	1	false	&ob_med	true	idle	medição	5
07	Pi	86	50	Sg,Hg	1	false	&ob_cal	true	idle	calibra-ção	4

Fig. 7.4 Dicionário de objetos do MAI

CABEÇALHO DO DICIONÁRIO DE OBJETOS (HEADER)								
object index	rom/ram flag	name length	access protection supported	version OD	local addr. OD_header			
0	false	16	true	02	&cab_addr			
-----	ST_OD length	local address ST_OD						
-----	1	&st_OD_addr(valor da variável apontador)						
first index S_OD	S_OD length	local address S_OD						
2	2	&s_OD_addr (valor da variável apontador)						
first index DV_OD	DV_OD length	local address DV-OD						
0	0	0						
first index DP_OD	DP-OD length	local address DP-OD						
0	0	0						
SUBDICIONÁRIO ESTATICO DE TIPOS (ST-OD)								
object index	object code	Descrição						
01	data -type	Integer8						
SUBDICIONÁRIO ESTATICO DE OBJETOS (S-OD)								
object index	object code	pass word	access group	access rights	data ty-pe index	Length	local address	variable name
02	var	00	00	Ra	01	01	&codigo_ad	código
object index	object code	pass word	access group	access rights	data ty-pe index	Length	local address	variable name
03	var	30	0f	W	01	01	&posicao_ad	posição

Fig. 7.5 Dicionário de objetos do IDP

### 7.3.1 Cabeçalho

O cabeçalho do dicionário de objetos do MAI está ilustrado nas figuras 7.6 e 7.7. O primeiro campo corresponde ao índice do cabeçalho, que deverá sempre ser inicializado com 0. O segundo campo define o atributo ROM/RAM-Flag, do tipo booleano, que especifica a possibilidade (true) ou não (false) de modificações no dicionário de objetos. Exemplos de alterações do dicionário de objetos são a inscrição remota de novos objetos mediante o serviço Put-OD ou a deleção de objetos já existentes com o mesmo serviço Put-OD, porém especificando para o código do objeto a ser apagado o valor "null". Por questões de simplificação, o dicionário de objetos neste exemplo não é passível de modificações dinâmica e remotamente a partir de serviços FMS emitidos pela Unidade Controladora, motivo pelo qual este parâmetro é inicializado com "false". Havendo a necessidade de se definir mais objetos com a expansão da aplicação, um novo dicionário de objetos deverá ser definido e regravado em memória EPROM. O terceiro atributo indica o comprimento máximo permissível dos nomes, definidos na descrição de cada objeto, utilizados para o endereçamento simbólico (por nomes) dos mesmos. Neste caso atribui-se o valor 16 (hexadecimal) suficiente para definir no OD o nome de todos os objetos da aplicação em questão.

Cada subdicionário pode ser acessado a partir de apontadores para o próprio endereço específico de implementação, como também através de informação sobre o número máximo de entradas ocupáveis. Tais informações estão definidas respectivamente para cada subdicionário no cabeçalho através dos campos "Local-address" e "OD-length". Excluindo-se o subdicionário estático de tipos (ST-OD), todos os demais necessitam adicionalmente para o seu gerenciamento da informação do índice do primeiro objeto a ocupar o subdicionário, a qual está disposta no cabeçalho na forma do campo "First Index". Outras informações contidas no OD-Header incluem a versão do dicionário de objetos, a definição quanto a proteção (true) ou não (false) do acesso dos parceiros de comunicação aos objetos locais (item 7.3.3) e o próprio endereço local de implementação do cabeçalho.

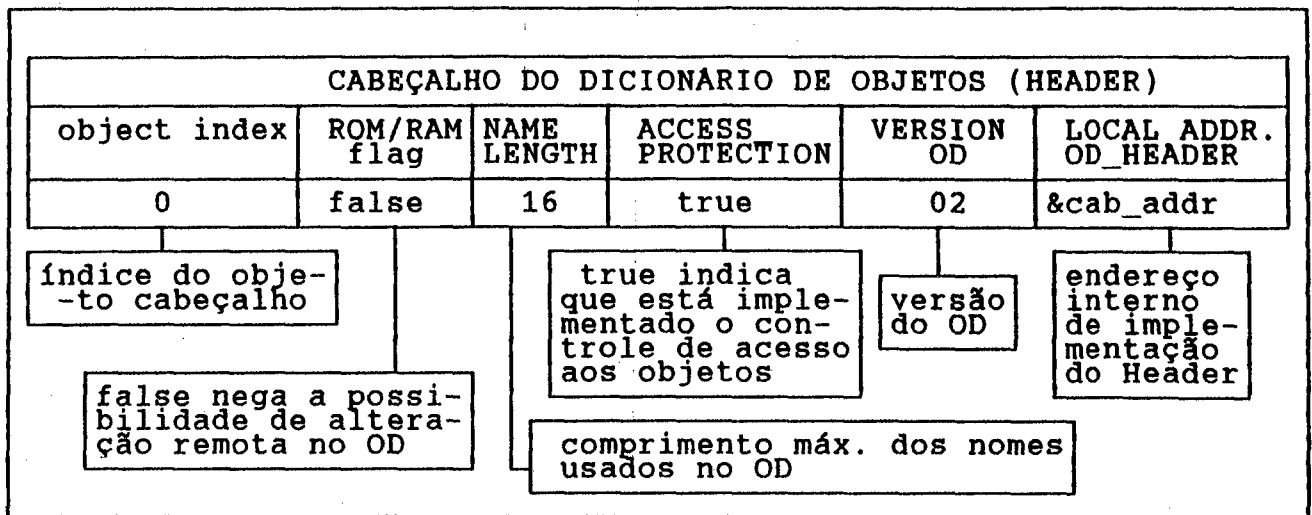


Fig. 7.6.A Cabeçalho do OD - sistema sensor MAI

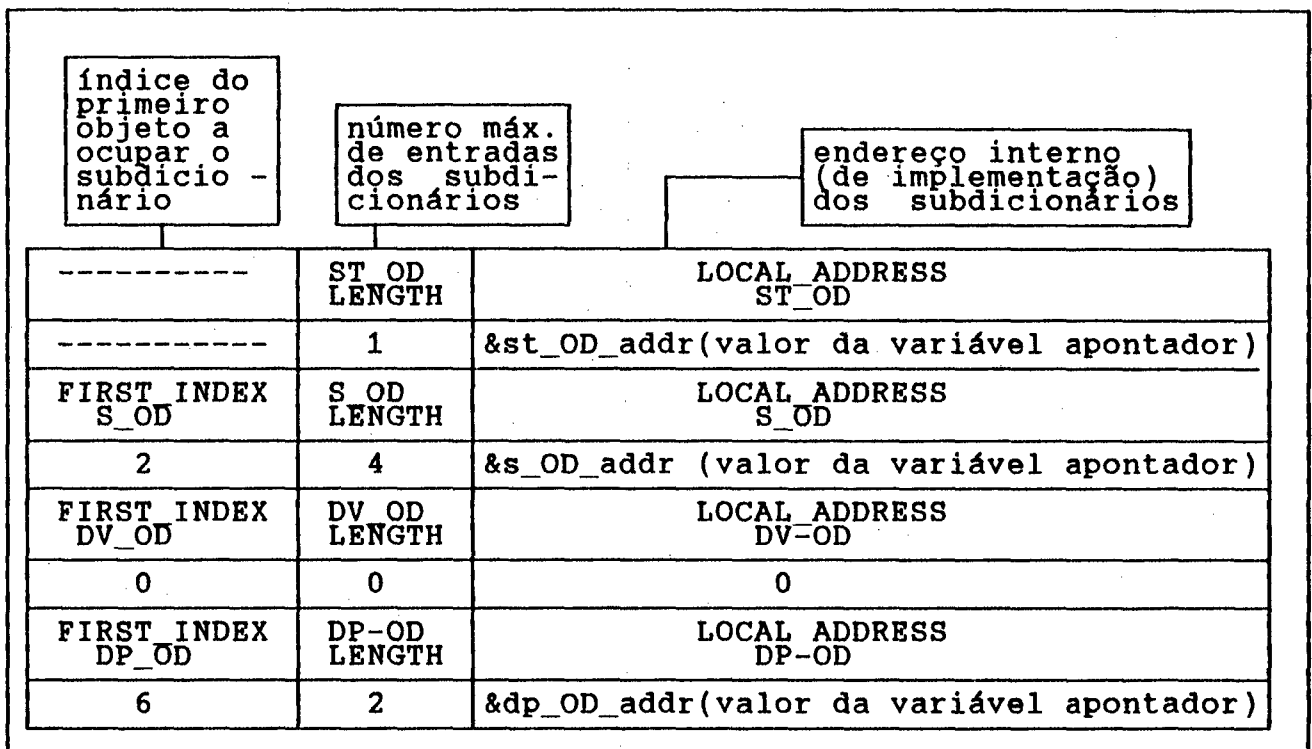


Fig. 7.6.B Cabeçalho do OD - sistema sensor MAI (continuação)

### 7.3.2 Subdicionário estático de tipos (ST-OD)

Este subdicionário terá apenas uma entrada, com índice 1, que define o tipo byte (0..255) a ser utilizado na descrição de tipo do objeto processo "medida", considerado neste exemplo um Array de elementos tipo byte.



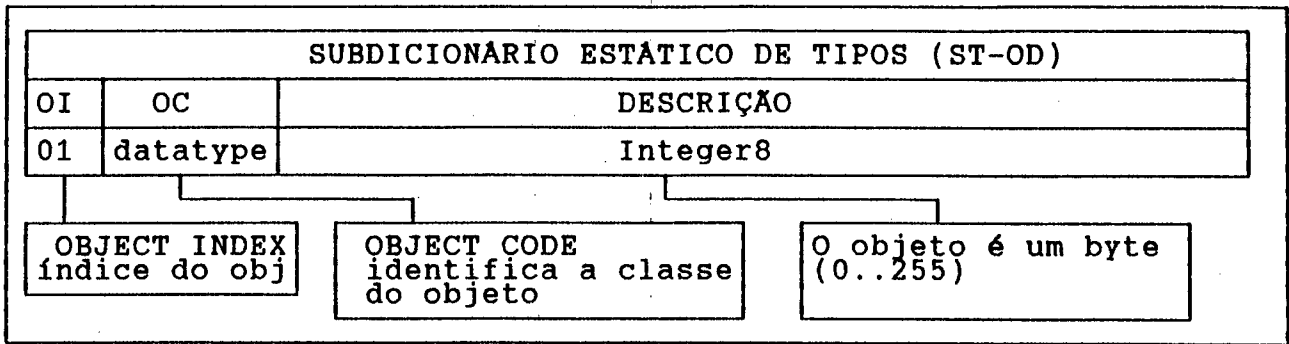


Fig. 7.7 Objeto tipo de dados

### 7.3.3 Subdicionário estático de objetos (S-OD)

Este subdicionário compõe-se dos objetos Medida (Array), Erro-posição (Evento), dom-cal e dom-med (Domínios), cada qual analisado a seguir quanto a sua descrição para este exemplo.

#### a) objeto Medida (Array)

A fig. 7.8 sumariza os valores atribuídos à descrição do objeto processo medida, o qual corresponde ao objeto de comunicação Array.

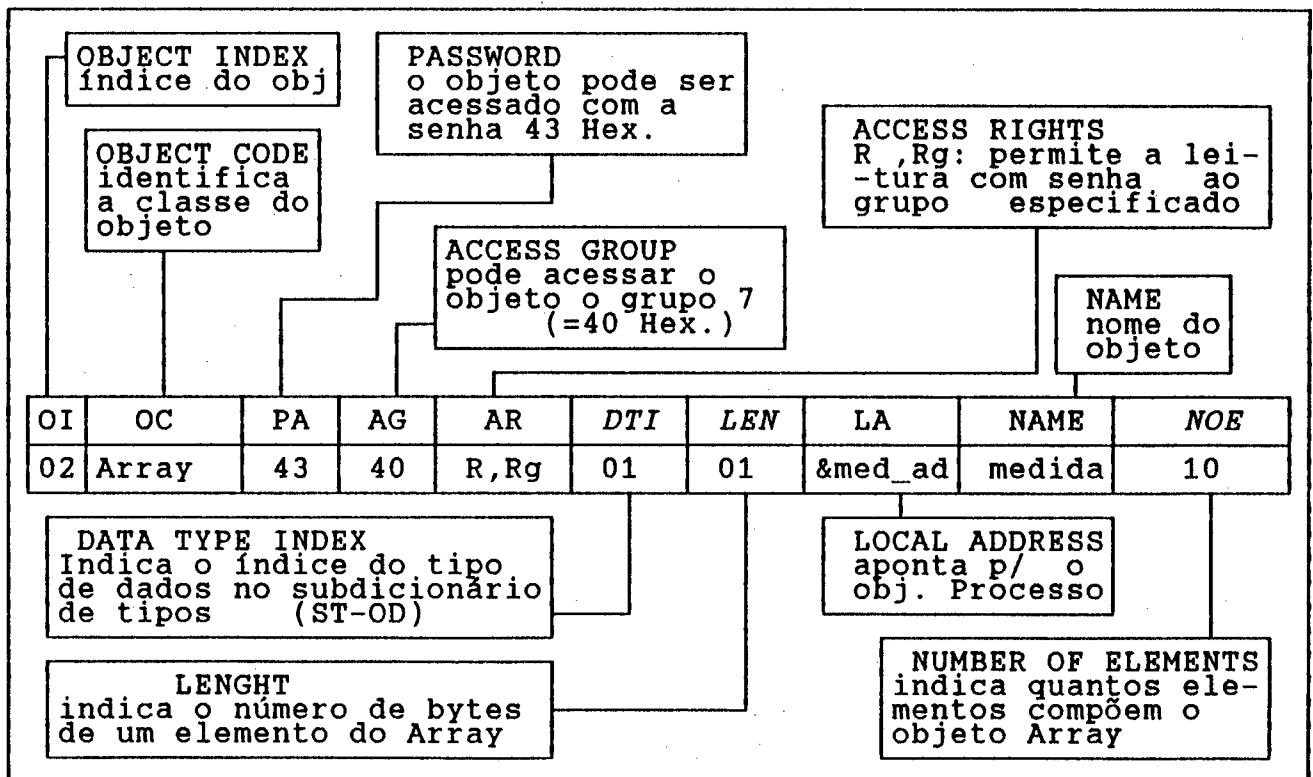


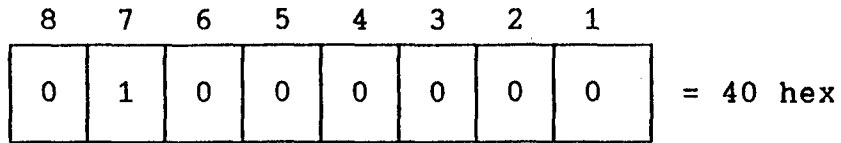
Fig. 7.8 Entrada do Subdicionário estático de objetos-objeto Array

O primeiro atributo que compõe a descrição do objeto Array é o índice do objeto, que corresponde na prática ao valor que o identifica inequivocamente perante os demais objetos do dicionário. Visto que os objetos estão relacionados no OD através de índices crescentes, este objeto deve assumir o índice 2. O atributo Object-code especifica a classe a que pertence o objeto, neste caso Array, e serve para o gerenciamento da ALI. Name especifica o nome do objeto, utilizado nas chamadas de serviço que utilizam o endereçamento simbólico. Destaca-se em particular os atributos Password, Access-group e Access-righths, os quais implementam o mecanismo de proteção de acesso aos objetos. Ao tentar estabelecer uma conexão com o sistema sensor, através do serviço Fms-Initiate, a Unidade Controladora informará qual a sua senha e grupo de acesso e obterá do sistema sensor na resposta ao serviço os parâmetros correlatos. Uma vez estabelecida a conexão, sempre que for executado um serviço FMS será verificado o direito de acesso da Unidade Controladora ao objeto processo. A seqüência de primitivas da fig. 7.9, tomando em consideração o objeto medida definido na fig. 7.4, ilustra o funcionamento do mecanismo de proteção de acesso. A Unidade Controladora na fase de operação requisita com a primitiva de serviço de leitura (Read.Reg) a leitura do objeto de índice 2 (objeto medida) do MAI. A primitiva de resposta (Read.Res) confirma positivamente a realização do serviço, pois a Unidade Controladora tem o "password" 43 e pertence ao grupo 7 (ambos valores arbitrados), requisitos de acesso exigidos para a leitura do objeto medida .

Serviço Initiate executado na Unidade Controladora (cliente) na fase de inicialização do sistema para o estabelecimento de conexão com o MAI (servidor)	
primitiva de serviço	parâmetros passados na primitiva de serviço
---> Initiate.Reg (requisição de serviço)	Access-protection-supported(UC)=false Password(UC)=43 Access-group(UC)=40 hex (grupo 7)
<-- Initiate.Res (resposta ao serviço)	Access-protection-supported(MAI)=true Password(MAI)=0 Access-group(MAI)=0
Serviço de leitura da medição do MAI (fase de operação)	
--> Read.Reg <-- Read.Res(+)	Index=02 Os processos com password 43 do grupo 7 podem ler o objeto

Fig. 7.9 Exemplo de funcionamento do mecanismo de proteção de acesso a objetos

Nota:



└─> bit 7 ligado implica em grupo de acesso 7

O mecanismo de proteção contra o acesso aos objetos funciona de forma similar para os demais objetos de comunicação, com a ALI verificando sempre, antes da atuação sobre o objeto, se está sendo respeitado pelo cliente da relação o direito de acesso ao objeto. O que varia apenas para cada classe de objeto é o tipo de direito de acesso.

O atributo *Local\_address* aponta para o endereço real do objeto processo no sistema sensor e será tratado mais especificamente no item 7.5. Examinando-se a fig. 7.4 verifica-se que todos os atributos discutidos até aqui são comuns à maioria dos objetos de comunicação. Sendo assim as descrições que se seguirão em torno dos objetos do exemplo em questão considerará, para evitar-se redundância, apenas os atributos específicos (destacados em *itálico*) de cada classe de objetos.

Finalmente o atributo *Data-type-index* indica o índice do tipo de dados relacionado no subdicionário de tipos, que descreve o objeto Array. *Length* especifica o comprimento em Bytes de um elemento do Array e *Number-of-elements* o número de elementos que compõem o Array.

#### b) Objeto Erro-posição (Evento)

O atributo *Index-event-data* (fig. 7.10) especifica o índice do objeto tipo de dados do parâmetro *Event-data*. Este parâmetro é utilizado no serviço *FMS-Event-notification* para comunicar o erro de posição da peça. *Length* representa o comprimento em número de bytes da informação passada através do parâmetro *Event-data*.

O atributo Enabled especifica o estado do objeto Evento, devendo ser inicializado com o valor "true", pelo qual, habilita-se o envio da informação de evento à Unidade Controladora na ocorrência de erro de posição. Este atributo pode ter seu valor remotamente alterado para "false", nas implementações que incluem o serviço Alter-event-condition-monitoring, bloqueando no servidor o envio das informações de evento.

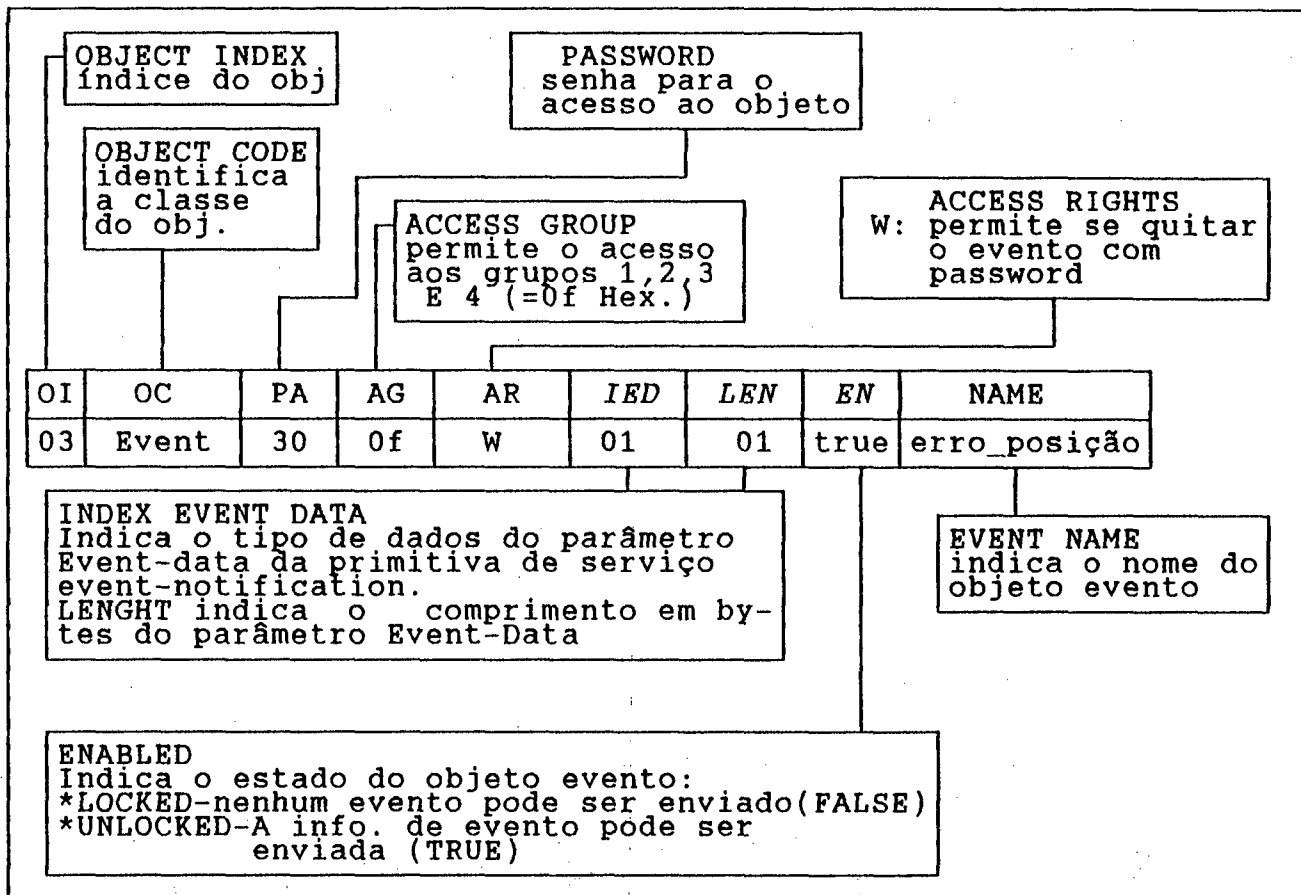


Fig. 7.10 Entrada do subdicionário estático de objetos- obj. Evento

c) Objeto " dom\_cal " e " dom\_med" (domínios)

Os atributos específicos do objeto Domínio, destacados na fig. 7.11, são Max. Octets, Upload State, Domain State e Counter. O primeiro atributo mencionado, Max Octets, indica o número máximo de bytes que compõe o Domínio. Admitindo-se programas de calibração e de medição de no máximo 4 kbytes, incluindo-se os parâmetros operacionais, este atributo terá de ser inicializado para cada objeto domínio com o equivalente valor hexa-decimal 1000 Hex.

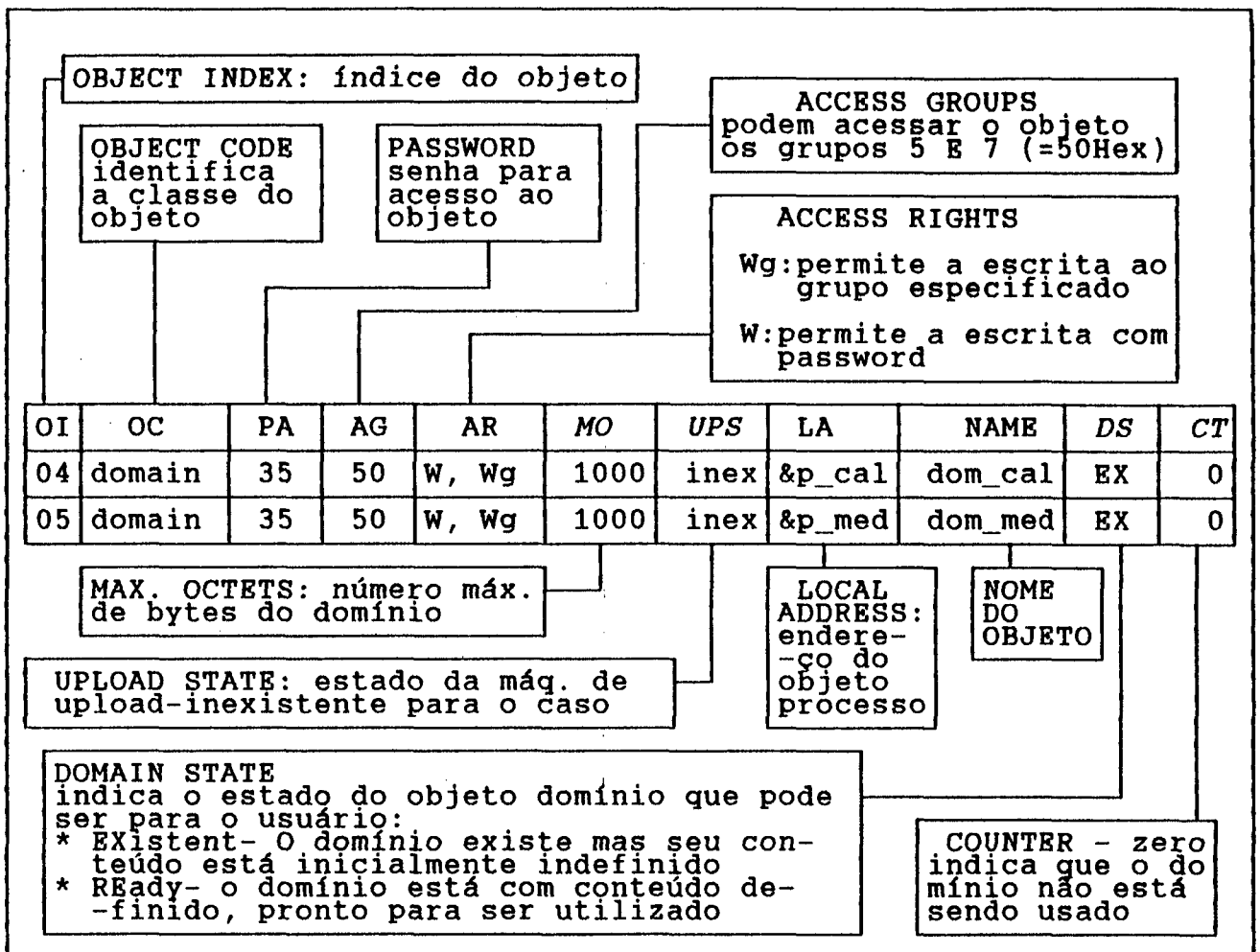


Fig. 7.11 Entrada do Sub-dicionário estático de objetos - objeto Domínio

O segundo atributo específico do objeto Domínio é definido em norma para ser utilizado no processo de Upload (carga de blocos de dados do sistema sensor para a Unidade Controladora) e deve ser atualizado pela ALI de modo a indicar o estado corrente deste processo. Nesta implementação deverá ser inicializado com "Inexistent", visto que a ALI exemplo não implementa o serviço de Upload. Domain-state deverá ser utilizado pelo programador usuário para indicar qual o estado inicial do Domínio. Para o exemplo em questão será "existent", visto que os domínios (dom\_cal e dom\_med) estão definidos no dicionário de objetos, mas o conteúdo ainda deverá ser carregado remotamente pela Unidade Controladora. Caso os programas e respectivos parâmetros estejam gravados em memória ROM local no sistema sensor em um objeto Domínio definido, já disponíveis para serem utilizados, o atributo Domain State deverá ser inicializado com o valor "ready".

Por fim, o último atributo específico para o objeto Domínio, Counter, especifica o número de usos de um Domínio, servindo apenas para a gerência de Domínio a sua interpretação e uso. Deverá ser inicializado com zero indicando à gerência do sistema que o Domínio não está inicialmente sendo utilizado.

#### 7.3.4 Sub-dicionário dinâmico Invocação de programa (DP-OD)

Este sub-dicionário contém apenas os objetos calibração e medição, definidos estaticamente na fase de configuração do sistema.

##### a) Objetos Calibração e Medição (Invocação de programa-Pi)

Reusable, Pi-state, Number-of-domains, List-of-Domain-Index e Deletable, listados na fig. 7.12, constituem os atributos específicos do objeto Invocação de Programa. Reusable especifica qual o estado a ser assumido pelo objeto Pi, descrito em PI-state, após o término de sua execução. Visto que a aplicação sob enfoque pode requerer uma nova calibração no ciclo subsequente dependendo do tipo de peça, o objeto Calibração deverá ser especificado com o parâmetro TRUE com o intuito de que, ao fim da calibração, o Pi volte novamente para o estado IDLE e esteja apto a executar novas calibrações posteriormente. O usuário de aplicação deverá apenas especificar no OD o estado inicial do objeto, cabendo ao gerenciamento de Invocação de programa a atualização dos estados conforme os eventos associados à chegada de primitivas de indicação FMS e demais condições definidas pelo protocolo de aplicação.

O atributo Pi-state deverá ser inicializado com "Idle", que corresponde ao estado da Pi antes do início do programa de calibração, e será atualizado pela ALI conforme o controle da execução de tais programas. "Number-of-domains" especifica o número de domínios que compõem a Pi e List-of-domain-index indica qual a relação de tais domínios, especificados através de seus índices. Para o caso em questão a Pi Calibração consiste apenas de um único domínio, Prog-calibração, cujo índice é 5. Deletable determina se a Pi pode ser apagada (true) ou não (false) com o

serviço Delete-program-Invocation. Considerações semelhantes são válidas para a Pi Medição.

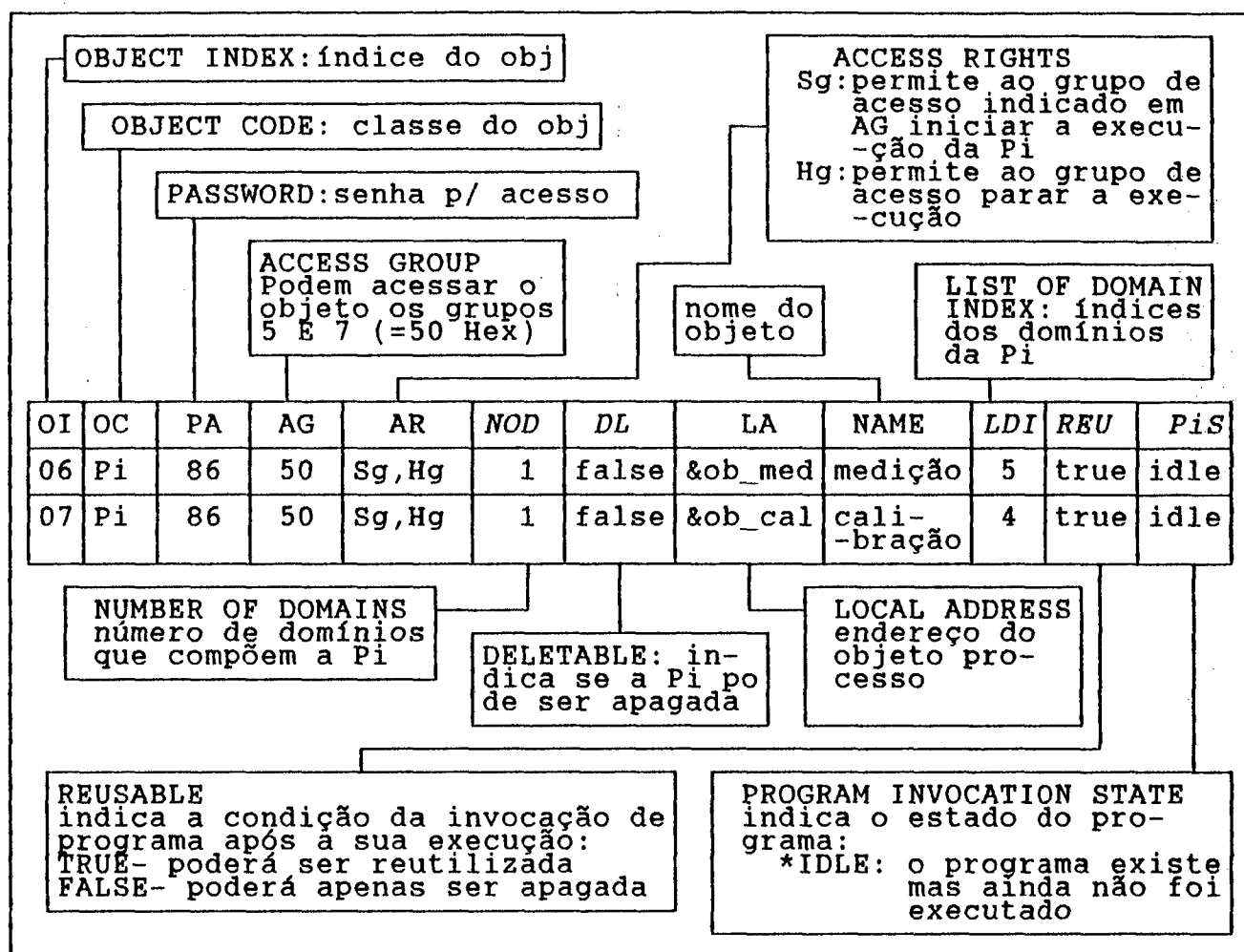


Fig. 7.12 Entradas do subdicionário dinâmico de objetos (DP-OD)  
Objeto Invocação de programa

#### 7.4 Definição dos serviços e das funções da ALI

Para a manipulação dos objetos de comunicação definidos no item anterior, deverá o usuário dispor dos serviços Fms Read e Write, para a leitura e escrita dos objetos variável; serviços Start, Stop, Resume e Kill para gerenciamento do programa de calibração e medição, além dos serviços de gerenciamento de Domínio (Initiate-Download-sequence, Download Segment e Terminate-Download-segment) para carga dos parâmetros da Unidade Controladora para o sistema sensor MAI. Para a notificação do fim da medição torna-se essencial a utilização do serviço Event-Notification. Também deverão ser previstos serviços

para a configuração do sistema de comunicação, mais especificamente os serviços Initiate e Abort para o estabelecimento e encerramento de conexões respectivamente entre a Unidade Controladora e cada sistema sensor.

Como analisado no capítulo 6, a tarefa usuário não deve interagir diretamente com os serviços FMS, mas sim com uma biblioteca de funções que constitui a ALI, a ser fornecida pelo fabricante do sistema como parte integrante do software. Para o exemplo em questão, a ALI utilizada baseia-se na chamada MMSI (Manufacturing Messaging Specification Interface) [82,83,84,85], tendo sua sintaxe modificada e o conjunto de funções restringido à sete funções que correspondem à funcionalidade mínima a ser utilizada nos sistemas sensores desta aplicação. Cada uma destas funções contém um código de retorno do tipo inteiro que traduz o resultado do pedido de serviço. Exemplos de erros na chamada da função, expressos por um código identificador, são parâmetros de chamada incorretos, ausência de recursos (p. Ex. memória) para a efetuação do pedido, e serviço FMS não suportado na funcionalidade do Fieldbus. Caso o pedido de serviço tenha sido realizado com sucesso, por exemplo com todos os parâmetros de chamada corretos, o código de retorno da função acusará o valor "SUCCESS". Do contrário, o valor de retorno exprimirá a causa da falha na invocação do serviço desejado.

A maioria das funções possui um parâmetro de saída "Result" que expressa o resultado final da execução do serviço, que pode constituir tanto a indicação da causa da falha na execução do serviço em caso negativo ou em uma confirmação do sucesso da operação em caso positivo. Exemplos de erros na execução da função são erros de hardware e impedimento de acesso ao objeto por falta de direitos de acesso.

Existem dois tipos de funções de biblioteca na interface FMS:

- a) Funções principais que mapeiam serviços FMS
- b) Funções de suporte que auxiliam o usuário

Todas as funções principais são projetadas para serem chamadas síncrona ou assíncronamente (servindo o parâmetro RETURN-EVENT\_NAME pa-



ra indicar qual destes dois modos está sendo usado). Uma exceção é a função Notifica, a qual corresponde ao serviço FMS-event-notification que não é confirmado, sendo portanto sempre assíncrona. Todas as funções de suporte podem somente ser chamadas sincronamente. As funções principais utilizadas no exemplo sob perspectiva estão representadas na fig. 7.13. Indica-se abaixo de cada função relacio-

FUNÇÃO	ENTRADA	SAIDA	COMENTÁRIO	TIPO
1. Download {INITIATE-DS} {DOWNLOAD SEG.} {TERMINATE-DS}	cr,data, ev_ret	Result.Info. fms_errorTy- -pe	Carrega parame- -tros ou código no sistema sensor	S,A
2 .Executa_prog (START) 3 .Para_prog (STOP) 4 .Retorna_prog (RESUME)	cr,oi,ev_ret	Result.Info. pi_errorTy- -pe	inicia } execução para } de pro- retorna } -grama	S,A
5 . Le (READ)	cr,oi,ev_ret	Result.Info .Read	lê objeto variá- -vel do ss	S,A
6 . Escreve (WRITE)	cr,oi,ev_ret	-----	escreve em objeto variável do ss	S,A
7. Notifica (EVENT-NOTIF.)	cr,oi,priori- -dade,ev-num	Result.Info fm_indica- -tion. Enotifica- -tion	Notifica a ocor- -rência de evento	A

Fig. 7.13 - Funções de aplicação da biblioteca ALI para o exemplo considerado

Obs.:

- oi - índice do objeto
- cr - índice da relação de comunicação
- ev\_ret - nome do evento a ser notificado na função espera
- prioridade - especifica a prioridade do evento, que pode ser alta (high) ou baixa (Low)
- ev-num - Event-number, especifica o número do evento, identificando-o perante outros eventos da aplicação
- data - especifica os dados carregados para o sistema sensor

nada e entre parenteses, qual o serviço FMS utilizado na sua implementação. Esta indicado ainda na tabela para cada função, em qual parâmetro de saída estará contido o resultado da função. O campo tipo indica a forma de chamada da função. Na chamada síncrona (S) de

uma função ocorre o bloqueio da tarefa usuário até o final da execução do serviço, quando então pode se verificar no parâmetro de saída associado ao serviço o resultado final da execução do mesmo. Já nas funções assíncronas (A) a tarefa usuário se suspende apenas até a verificação do pedido de serviço, após o que o usuário poderá através do código de retorno verificar o sucesso ou falha do encaminhamento do pedido de execução do serviço e encaminhar subsequentemente novos pedidos de serviços ao sistema de comunicação. As funções que podem ser chamadas assincronamente incluem como parâmetro de entrada o parâmetro Return-event-name através do qual pode-se associar um evento com o final da execução do serviço. O usuário terá acesso a esta notificação após chamar a função "wait" (fig. 7.14) e, somente então após a notificação poderá verificar o resultado final do serviço no parâmetro Result, passado na chamada da função assíncrona. A realização de uma chamada síncrona se dá mediante a atribuição ao parâmetro Return-event-name do valor "synchronous".

```

unsigned short wait (dT, ev_ret)
    Dt      dT;
    char    *ev_ret;
}
Onde:
dT - intervalo de tempo de espera pela ocorrência do evento
ev_ret - nome do evento cuja ocorrência é esperada (este nome-
        de ve ser fornecido na chamada da função assíncrona).

```

Fig. 7.14 Definição da função espera

Como se pode observar a ALI exemplo oferece uma parte da funcionalidade definida em norma, considerada necessária e indispensável para a execução da aplicação especificada, tendo como objetivo a redução da complexidade do sistema sensor resultante.

## 7.5 Representação e acesso aos objetos processos

A definição dos procedimentos que atuam sobre um objeto processo são responsabilidade do programador da aplicação e podem ser implementados de várias formas. Por exemplo, o objeto calibração pode ter seu acesso realizado segundo número de leituras e número de ciclos de medição a serem especificados pelo usuário apropriadamente de acordo com suas necessidades particulares. A ativação no sistema sensor dos

procedimentos que atuam sobre os objetos processos de aplicação é de responsabilidade do sistema de comunicação e não deve haver a intervenção do programa usuário. No exemplo a ilustrar, define-se que cada objeto processo deve ser representado por uma variável global com apontadores para os procedimentos que atuam sobre uma classe de objetos.

O endereço da variável que representa o objeto processo deve ser colocada no Descritor do Objeto (OB) do Dicionário de Objetos, especificamente no atributo Local-address de cada objeto, possibilitando assim a execução das operações sobre os objetos processos. Desta forma o atributo Local-Address assume o seguintes valores, dependendo da classe de objetos em consideração:

- Objeto Domínio: Local-address apontará para a função, definida pelo usuário, que fará a carga do programa na memória local do sistema sensor;

- Objeto Variável: Local-address apontará para a função de leitura ou escrita, dependendo da inicialização do sistema, da variável de processo;

- Objeto Invocação de Programa: Local-Address apontará para a função responsável pelo gerenciamento do programa representado por este objeto;

- Objeto Evento: Não há na descrição deste objeto, para este exemplo, a variável Local\_address, pois não há funções para que um usuário cliente atue sobre um objeto evento remoto, e sim para que um usuário servidor notifique a sua ocorrência. Portanto não há a necessidade de que o sistema de comunicação atue sobre este objeto, com isto não há uma variável representando o objeto evento.

Esta forma de acesso aos objetos processos é exemplificada na fig. 7.15 para a classe de objetos variável, considerando as ações realizadas desde a chamada da função, correspondente execução da mesma no sistema sensor e obtenção da resposta da função pela aplicação. A forma de acesso as demais classes de objetos segue a mesma idéia exposta a seguir.

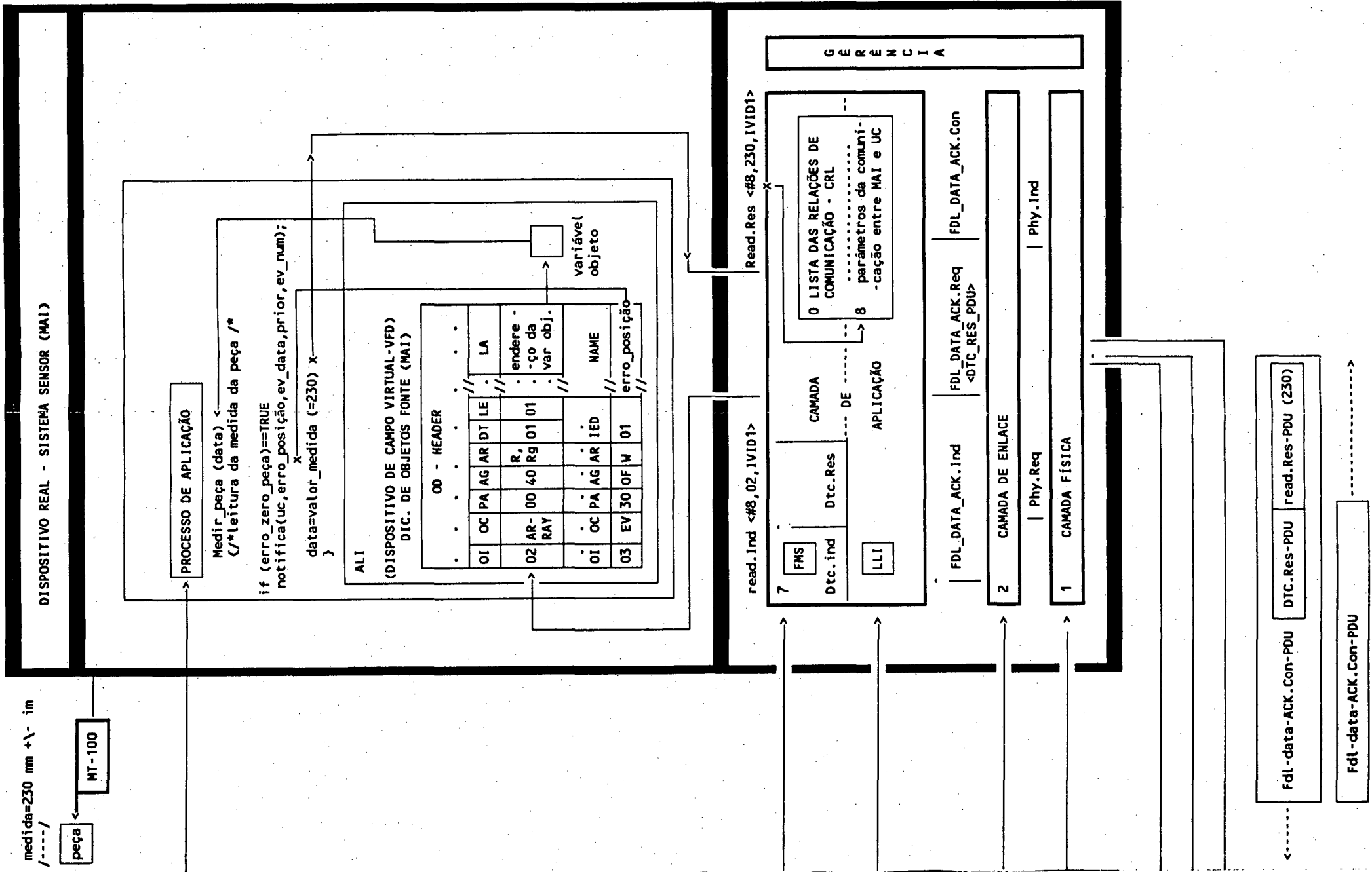
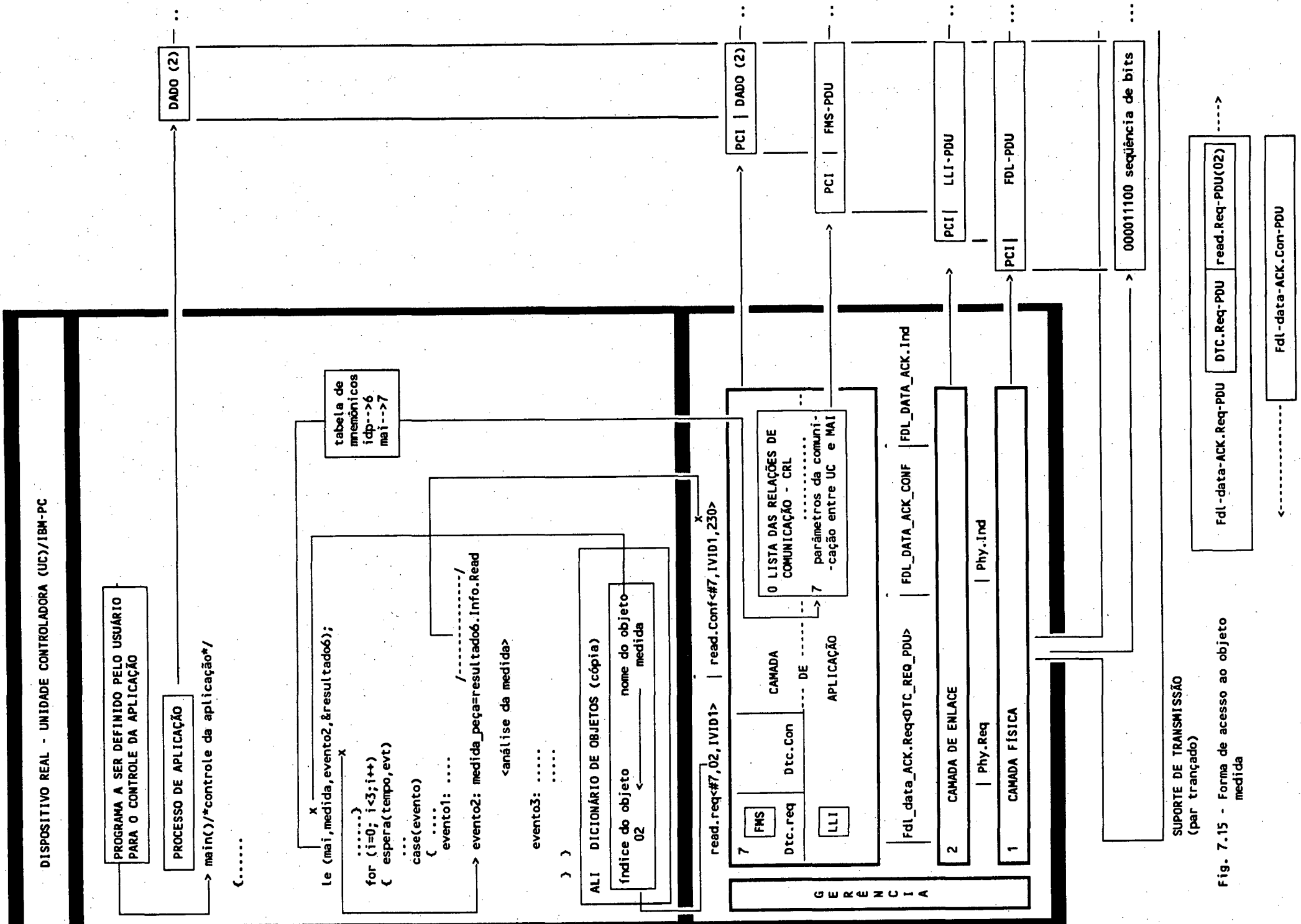


Fig. 7.15 - Forma de acesso ao objeto medida

### 7.5.1 Forma de acesso ao Objeto Medida (Classe Variável)

Com o intuito de se obter do MAI o valor da medida da peça, o algoritmo executado na Unidade Controladora utiliza a função ALI de leitura (Le), conforme destacado simplificadamente na fig. 7.15. O primeiro parâmetro da função representa o mnemônico do índice da relação de comunicação traduzido em uma tabela de mnemônicos para o valor numérico 7. O segundo é o nome do objeto de comunicação a ser lido que representa o objeto processo medida da peça. O terceiro define o tipo de chamada de função (síncrona ou assíncrona) conforme discutido no item 7.4 e finalmente o último parâmetro é o parâmetro de saída, passado pelo usuário na chamada da função, onde estará disponível para o usuário, após a execução final do serviço, o valor da medida da peça.

Com a chamada da função, o nome do objeto de comunicação é convertido pela ALI, através da cópia do dicionário de objetos do MAI que mantém o controlador, no índice do objeto. A ALI gera uma identidade de solicitação (InvokeId), que identifica inequivocamente perante o sistema de comunicação um pedido de serviço, e envia a requisição do serviço de leitura (FMS-Read) à camada de aplicação com a primitiva de serviço FMS-Read-Request. Como resultado desta operação o serviço de rede (camada 7) é informado de que deve enviar um comando de leitura de variável ao parceiro de comunicação remoto. O protocolo FMS pré-afixa um cabeçalho, denominado de PCI (Protocol Control Information) ao índice do objeto, contendo caracteres codificados que especificam a requisição de uma operação de leitura. Com isto está formada a FMS-PDU, ou Unidade de dados de protocolo FMS, a qual será transmitida e interpretada pela correspondente subcamada FMS no sistema sensor MAI. Em seguida o FMS gera o pedido de serviço DTC (Data Transfer Confirmed) para a subcamada LLI com a primitiva de serviço DTC.Request, passando como parâmetro na primitiva a FMS-PDU. Para que a FMS-PDU possa ser transmitida para a camada de aplicação no MAI, necessário se faz conhecer o endereço da estação MAI, que a identifica perante outros dispositivos e sistemas sensores que estejam conectados ao Fieldbus, o endereço do processo de aplicação, dentre outros processos executados no MAI, responsável pela leitura da peça, o tipo de comunicação (mestre-mestre), o tamanho dos buffers de transmissão e recepção, além de um conjunto de outros pa-

râmetros que definem o canal lógico ou a relação de comunicação entre os processos de aplicação executados no controlador e no MAI. Tais parâmetros, definidos pelo usuário na fase de configuração do Fieldbus, estão disponíveis para o protocolo da subcamada LLI na entrada 7 da CRL, informado na chamada da função Le com o parâmetro Referência de Comunicação. A LLI utiliza parte destes parâmetros para alocar recursos internos no sistema de comunicação (por Ex., memória) e outra parte para constituir a PCI desta subcamada a qual formará, juntamente com a FMS-PDU, a LLI-PDU.

Subsequentemente o protocolo LLI mapeia o serviço DTC no serviço de enlace SDA, enviando à camada de enlace a primitiva de serviço correspondente FDL\_DATA\_ACK.Req, tendo como parâmetro a LLI-PDU (DTC-REQ-PDU). A camada de enlace, além de acrescentar um cabeçalho (PCI) insere um sufixo à DTC-REQ-PDU, que consiste em um campo de FCS (Field-Check-Sequence) para possibilitar a detecção de erro no destino. Com isto esta formada a FDL-PDU, ou mais frequentemente o denominado quadro de transmissão que é transmitido com a ajuda da camada física para o MAI.

Como fica claro, o quadro de transmissão transporta todos os demais PDU's. Na fig. 7.15 é importante perceber que, embora a transmissão real da informação seja vertical, cada camada está organizada como se fosse horizontal. Virtualmente a camada de aplicação no controlador entrega a sua FMS-PDU diretamente à camada de aplicação no MAI (o que também é válido para as demais camadas) e, do seu ponto de vista, o fato de que a mensagem deva ser entregue à camada de enlace é um detalhe técnico. Por outro lado destaca-se o fato de que a camada de enlace (ou outra qualquer) não tem conhecimento de qual parte dos dados que lhe são entregues pela camada de aplicação é o cabeçalho, e o que é dados ou comando, e nem deve ter este conhecimento. O processo de encapsulação da informação do usuário (no caso o índice do objeto) inicia-se na camada de aplicação e é repetido até a camada física, onde o conjunto de informações (dados do usuário + variáveis de comunicação) é efetivamente transmitido. Na prática da implementação nenhum SDU é repassado através da interface entre as camadas. O que ocorre são chamadas entre as tarefas vizinhas com a repassagem dos parâmetros citados. Por exemplo, a simples transmissão de um apontador de um buffer poderá permitir à ta-

refa chamada a leitura dos restantes dados nele colocados. Como se percebe ainda, nenhum protocolo das camadas da arquitetura transmite o PDU correspondente à entidade par-remota. De fato esse PDU será repassado para a camada subjacente, e assim sucessivamente até que se alcance a camada física. Em virtude dos serviços executados em cada camada, ocorrerá um envelopamento gradativo de SDU's com PCI, formando-se os PDU's sucessivos. Na camada física o PDU é um bit isolado da informação, ou melhor, o conjunto de sinais resultantes da codificação de símbolos isolados da camada de enlace.

No sistema sensor MAI, o quadro de informação é recebido e notificado à camada de enlace. Se a EMI ou qualquer outra fonte de erros provocar durante a transmissão a mutilação do quadro completo, alterar um único bit ou um conjunto de bits, a camada de enlace no MAI detectará que o valor no FCS recebido não confere com o FCS localmente calculado e a mensagem recebida não será repassada para a camada de aplicação, e será gerada para a Unidade Controladora uma confirmação negativa. Em hipótese contrária, ou seja, caso a transmissão do controlador para o MAI seja bem sucedida, a camada de enlace no sistema sensor responde com uma confirmação positiva que leva a uma primitiva de confirmação do serviço SDA (FDL-DATA-ACK.Conf) no lado do controlador. No lado do MAI, a camada de enlace retira o sufixo (FCS) e o prefixo (PCI de enlace) e repassa para a camada de aplicação (subcamada LLI) a informação resultante, o que causará a ocorrência de uma primitiva de indicação de serviço FDL-DATA-ACK.Ind. Similarmente, a subcamada LLI retira a informação PCI-LLI e repassa para a subcamada FMS a informação, através de uma primitiva DTC.Ind. O FMS retira a sua informação de protocolo e entrega para a ALI, através de uma FMS-Read.Ind, o comando de leitura do objeto de índice 2. A ALI verifica a partir de então, na entrada 2 do dicionário de objetos fonte, se os requisitos exigidos para o acesso (password ou grupo de acesso) são satisfeitos, sendo que em caso negativo, uma resposta de erro FMS é devolvida ao controlador ao invés do valor da medida. Do contrário, a ALI com o auxílio do dicionário de objetos associa o índice do objeto ao atributo Local Address. Este aponta para a função definida pelo usuário, que fará conforme a implementação da interface de processo, a leitura das medidas da peça. A ALI então dá início à execução de tal função e o valor da leitura é armazenado no endereço apontado pelo parâmetro

"data". Dentro do procedimento, o usuário deverá, utilizando recursos locais da implementação do MAI, verificar se a peça está corretamente posicionada. Se for detectado erro de posicionamento, o processo de aplicação chama a função FMS-event-notification para enviar um alarme ao controlador. Esta função tem como parâmetros o índice do objeto evento, o número do evento e a informação de evento a ser enviada a Unidade Controladora. A ALI gera a primitiva FMS-Read.Res, contendo o valor da medida da peça como parâmetro. A camada de aplicação constrói a Read-res-pdu, similarmente ao processo executado na transmissão, e a repassa para a LLI com a primitiva DTC.Res. A LLI gera o pedido de serviço SDA, com a primitiva FDL-DATA-ACK Req, e a FMS-PDU.Res é transportada ao requisitor do serviço FMS de leitura, a Unidade Controladora. A resultante SDA-REQ-PDU transporta os dados para a UC, sendo que a sua recepção na camada de enlace causa a ocorrência da primitiva FDL-DATA-ACK.Ind. Esta é respondida imediatamente com uma confirmação que irá causar na camada de enlace do MAI a primitiva FDL-DATA-ACK.Conf. A READ-RES-PDU é entregue à LLI, no lado do controlador, com a primitiva FDL-DATA-ACK.Ind, e ao FMS é entregue com uma DTC.Con. O FMS por sua vez entrega a ALI o valor da medida, com a primitiva Read.Conf. Este valor é depositado no parâmetro de saída da função Result1.info.Read, a disposição do usuário para processamento estatístico.

## 7.6 Programação da Aplicação

Um maior detalhamento do programa de controle da aplicação da fig. 7.15 está ilustrado na figura 7.16.

```

linha  main () /* controle da aplicação */
01      {
02      .
03      .
04      inicializa (); /* função de inicialização do sistema */
05      .
06      .
07      for (;;) /* operação cíclica */
08      .
09      {
10      /* leitura do código de identificação da peça */
11      .
12      if (le(idp,codigo,synchronous,&Result1)!=SUCCESS)
13      printf ("\n Falha no pedido do serviço de leitura");
14      .

```



```

15     if (Result1.Result!=SUCCESS) /* result - param. de saída: indica o sucesso da execução */
16         printf ("\n Falha na execução do serviço de leitura");
17     else
18
19         peça_id=*Result1.Info.Read; /* result.info.read - param. de saída: indica o valor lido */
20
21 /* Posicionamento da peça frente ao MAI - escrita no IDP */
22
23     if (escreve(idp,posicao,synchronous,&Result2)!=SUCCESS)
24         printf ("\n Falha no pedido do serviço de escrita");
25     if (Result2.Result!=SUCCESS)
26         printf ("\n Falha na execução do serviço de escrita");
27     else
28         printf ("\n Operação de escrita bem sucedida ");
29
30 /* novo_tipo_peça() - função para verificar se a peça per-
31 tence a um novo tipo em relação a precedente - retorna o
32 valor TRUE caso favorável */
33
34 if (novo_tipo_peça==TRUE)
35 {
36     /* carga de programa remota, cujo programa de calibração é descrito pela var. Prog*/
37
38     if (download(mai,dm_cal,synchronous,prog_cal[peça_id],Fms_output_FAR*)&Result3)!=SUCCESS)
39         printf ("\n Falha no pedido de carga do programa de calibração");
40     if (Result3.Result!=SUCCESS)
41         printf ("\n Falha na execução da carga do programa de calibração");
42     else
43         printf ("\n Sucesso na carga do programa de calibração");
44
45     if (download(mai,dm_med,synchronous,prog_med[i],(Fms_output_FAR*)&Result4)!=SUCCESS)
46         printf ("\n Falha no pedido de carga do programa de medição");
47     if (Result4.Result!=SUCCESS)
48         printf ("\n Falha na execução da carga do programa de medição");
49     else
50         printf ("\n Sucesso na carga do programa de medição");
51 }
52 if (ultima_peça==TRUE)
53 {
54     printf ("\n Fim do lote de peças - operação concluída ");
55     break;
56 }
57
58
59     if (executa_prog(mai,calibracao,evento1,&Result5)!=SUCCESS)!!
60
61         (executa_prog(mai,medicao,evento2,&Result6)!=SUCCESS)!!
62         (fms_ireceive(mai,evento3,&Result7,&nome_serviço)!=SUCCESS))
63         printf("\n Falha de comunicação com o MAI");
64
65 for (i=0;i<3;i++)
66 {
67     if (wait(intervalo_espera,&evt_name)!=SUCCESS)
68     {
69         printf("\n Falha de wait ");
70         /* análise */
71     }
72     else
73     {
74         switch (evt_name)
75         {
76             case evento1:
77                 {
78                     if (Result5.Result!=SUCCESS)
79                         printf ("\n Falha na execução da calibração");
80                     else
81                         printf ("\n Operação de calibração bem sucedida");
82                     break;
83                 }
84             case evento2:
85                 {
86                     if (Result6.Result!=SUCCESS)
87                         printf("\n Falha na operação de medição");
88                     else
89                         {
90                             if (le(mai,medida,synchronous,&result8)!=SUCCESS)

```

```

91     printf("\n Falha no pedido do serviço de leitura");
92     if (result8.Result!=SUCCESS)
93         printf("\n Falha na leitura das medições realizadas");
94     else
95         medida_peça=(signed long)Result8.Info.Read;
96         /* análise da medida*/
97     }
98     break;
99 }
100
101 case evento3:
102 {
103     if (Result7.Result!=SUCCESS)
104         printf ("\n Falha de indicação");
105     else
106     {
107         switch (evt_name)
108         {
109             case ABORT:
110             {
111                 printf("\n Indicação de Abort");
112                 /* informações contidas em Result7.info.FM_indication.abort_info */
113                 break;
114             }
115             case FMS_event_notification:
116             {
117                 if (Result7.Info.Fm_indication.Enotification==erro_posição)
118                 {
119                     printf ("\n ERRO DE POSICIONAMENTO");
120                     /* parada temporizada para reajuste manual da peça */
121                 }
122             }
123         }
124     }

```

Fig. 7.16 Programa de controle da aplicação

O procedimento Inicializa () realiza um conjunto de passos necessários para se colocar o Fieldbus na fase de operação (também fase de transferência de dados), tais como a carga da CRL local e remota em cada sistema sensor, inicialização do VFD-MAI e VFD-IDP, além do estabelecimento de conexão com os sistemas sensores. Após a fase de inicialização, o programa de aplicação entra em um laço de execução, somente interrompido após a medição da última peça de trabalho. A primeira operação consiste na leitura do código da peça, através da chamada síncrona da função ALI-le, bloqueando a aplicação enquanto não chega a resposta do código da peça. A declaração da linha 12 encaminha o pedido de serviço de leitura ao sistema de comunicação, verificando com o código de retorno da função se não houve erros no encaminhamento do pedido de leitura.

Com a chegada da resposta do sistema sensor é executado a declaração da linha 15. O usuário primeiramente deve verificar se o serviço foi corretamente executado examinando o parametro result1.Result. Caso tenha ocorrido a leitura com sucesso, as informações do código da

peça estarão disponíveis, de acordo com a definição da ALI exemplo, no parametro Result1.Info.Read. Em seguida o posicionamento da peça é feito escrevendo-se uma palavra de controle de um byte de comprimento no IDP. Utiliza-se a função ALI-Escreve, e o teste do pedido de serviço e posterior verificação da execução do serviço é realizado da mesma forma que na função anterior. A função Novo\_tipo\_peça () é específica de implementação e deverá retornar o valor True caso a peça a ser medida representa um novo tipo em relação à precedente, exigindo portanto carga dos programas de calibração e medição conforme o tipo. Se necessária, a carga do programa ocorrerá com a chamada da função Download, especificando como Prog\_med[i] e Prog\_cal[i] respectivamente os programas e dados de medição e calibração serem carregados no MAI considerando a peça de código "peça\_id". A função Ultima\_peça retorna o valor True caso não haja mais peças a serem inspecionadas, quando então através da instrução da linha 55 finaliza-se o laço de execução do programa, encerrando portanto a inspeção completa de um lote de peças. Os dois pedidos subsequentes, a solicitação do início da operação de calibração no MAI e posterior medição da peça são assíncronos, o que vale dizer que a ALI encaminha o pedido ao sistema de comunicação e continua executando o fluxo normal do programa de controle da aplicação. Para isto o usuário deverá informar na chamada da função o nome dos eventos que será associado a chegada da confirmação do fim da calibração (evento1), do fim da operação de medição (evento2) ou da ocorrência de alarme de erro de posicionamento da peça (evento3). A função fms\_ireceive() (linha 62) é uma função de suporte e serve para habilitar o sistema de comunicação a receber indicações de serviço FMS, tais como a ocorrência de um alarme remoto mediante a função Event\_notification, ou um pedido de encerramento da conexão (Abort), enviado pelo próprio sistema de comunicação para o usuário, caso haja falha no protocolo de comunicação. O uso da função Wait permitirá se aguardar a chegada da resposta dos serviços enviados assíncronamente. A chegada de uma confirmação ativará um evento específico, e por conseguinte a execução do código correspondente ao tratamento da informação de evento. A chegada da resposta da função Executa\_prog calibração irá ativar o evento 1. A chegada da resposta da função Executa\_prog medição irá ativar o evento 2. Na ocorrência deste, deverá ser executada a função ALI-le, sobre o objeto medida, para a leitura das medições anteriormente executadas. Após o teste da realização bem

sucedida do serviço, o valor da medida é carregado para a variável medida-peça, lendo-se o parâmetro da ALI Result8.Read.Info. A informação da medida da peça pode ser então processada conforme a aplicação, para o controle estatístico do processo. Podem ser geradas cartas estatísticas a serem enviadas ao nível de controle imediatamente superior, mediante uma rede de maior porte, desde que se disponha de uma comporta. O evento 3 representa a chegada de uma Indicação de serviços, que pode ser uma indicação de "abort" da conexão, originada pelo próprio sistema de comunicação devido a falha do protocolo, ou então uma indicação de alarme através do serviço Event-notification, acusando erro de posicionamento da peça.

O programa da fig. 7.16, a ser executado na própria Unidade Controladora, deverá ser compilado e logicamente ligado com uma biblioteca de funções que constitui a própria ALI. O programa .EXE resultante deverá ser carregado na unidade controladora (IBM-PC) e chamado a partir do sistema operacional DOS, conforme ilustra a figura 7.17.

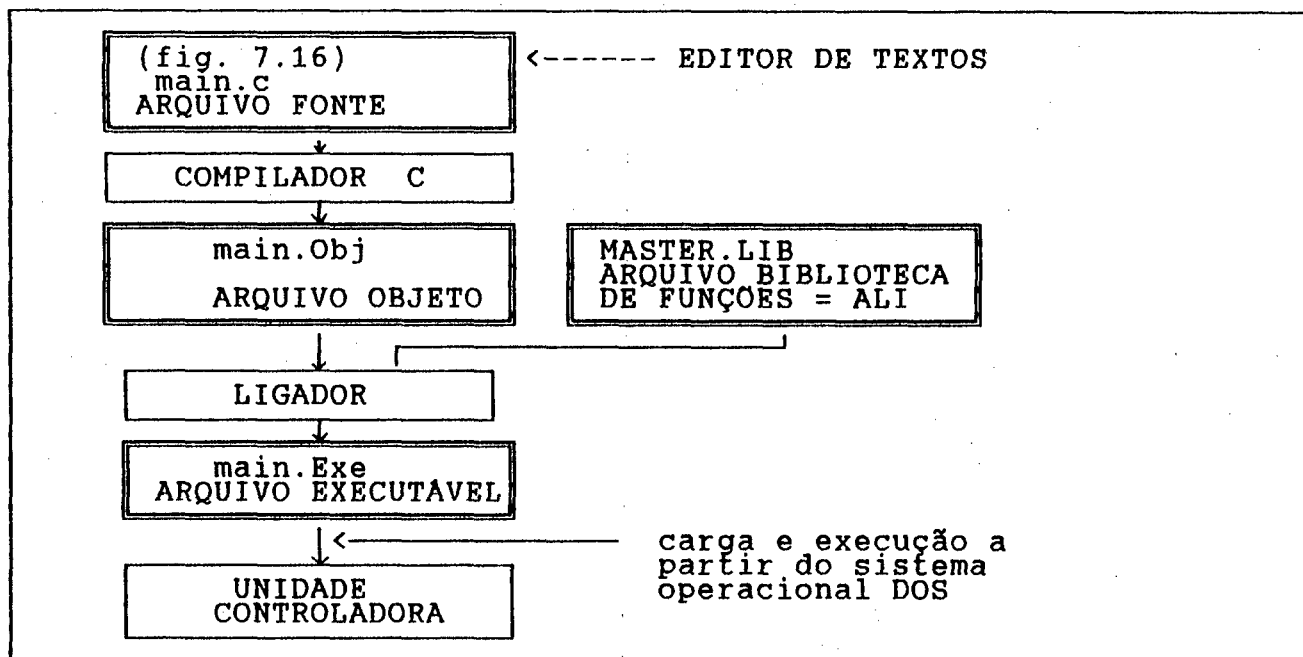


Fig. 7.17 - Procedimento de geração do arquivo executável a partir do arquivo fonte com o programa de controle da aplicação

Para o sistema sensor MAI (e também IDP), o programa fonte conterà declarações de inicialização do dicionário de objetos, de variáveis de comunicação das camadas de protocolos, além de procedimentos de

acesso aos objetos definidos pelo usuário. O programa executável será gerado de forma semelhante ao descrito na figura 7.17, com a ressalva de que deverá se utilizar um compilador que gere código de máquina para o microprocessador utilizado na arquitetura do sistema sensor (por exemplo, NEC V25, 8051 ou 80188). O programa executável resultante deverá ser gravado em EPROM e será inicializado assim que a placa for energizada com tensão de alimentação, a partir do que o sistema sensor ficará aguardando os comandos FMS emitidos pela unidade controladora e os executará sequencialmente conforme as definições do protocolo Profibus.

Para que o programa de aplicação possa ser executado no lado dos sistemas sensores, o usuário terá de considerar os seguintes passos sumarizados abaixo:

1. Declaração dos objetos de comunicação: Em um módulo criado pelo usuário, e que posteriormente terá de ser ligado com a biblioteca de funções da ALI, o programador da aplicação declara os objetos de comunicação segundo os tipos que constituem o VFD (ver exemplo de tipo do objeto de comunicação Array na fig. 7.18);
2. Definição das operações (funções em C ou C++) de acesso ao objeto: O usuário define, conforme o hardware da placa do sistema sensor, de acordo com o tipo de operação de E/S que pretende efetuar e ainda conforme a aplicação (por exemplo, quantas leituras formará a medida) a operação (função ou procedimento da linguagem de programação) que fará a aquisição da variável;
3. Definição dos objetos processos: Cada objeto processo deve ser representado por uma variável global "object\_variable", com apontadores para os procedimentos/funções que atuam sobre uma classe de objetos (ver também item 7.5);
4. Inicialização das operações de acesso ao objeto: Cada objeto processo deve apontar para um procedimento ou função que efetivamente atue sobre o processo de aplicação, lendo ou escrevendo em E/S ou memória. Deve-se portanto indicar para qual procedimento deve o objeto processo apontar;

5. Inicialização dos objetos de comunicação: O usuário deve especificar em software valores aos atributos dos objetos de comunicação.

A representação destes passos está representada em software para a linguagem C na fig. 7.19.

```

01
02
03     typedef struct Object_variable
04     {
05         Error_class      *(*read)(Uint8*);
06         Error_class      *(*write)(Uint8*,Uint8);
07     }Object_variable;
08
09     typedef unsigned char  Uint8;
10     typedef unsigned short Uint16;
11
12     typedef struct Octet_string
13     {
14         Uint16      size;
15         Uint16      *element;
16     } Octet_string;
17
18     /******
19     /* Definição de tipos dos objetos de comunicação que
20     /* constituem o dicionário de objetos - exemplo para o
21     /* objeto variável array
22     /******
23
24     typedef struct Variable_Array
25     {
26         Uint8      object_index;
27         Uint8      object_code;
28         Uint8      Password;
29         Uint8      access_group;
30         Uint8      access_rights;
31         Uint16     data_type_index;
32         Uint8      length;
33         Uint8      *local_adress;
34         Uint8      number_of_elements;
35         Octet_string array_name;
36     };
37
38     /* definição de tipos dos demais objetos do OD */
39
40     .
41     .
42
43

```

Fig. 7.18. Definições utilizadas pela biblioteca de funções da ALI

```

01 /* Módulo de inicialização do VFD */
02
03
04
05 /******
06 /* 1. Declaração do obj. de comunicação "medida" */
07 /******
08     variable_array medida;
09
10 /******
11 /* 2. Definição da função de acesso ao objeto medida */
12 /* A função de acesso deve ter este formato abaixo */
13 /******

```

```

14 /*****
15
16 Error_class  *read_medida(data)
17   Uint8      *data;
18 {
19 /*****
20 /* corpo do procedimento de leitura da medição, a ser */
21 /* definido pelo usuário conforme o hardware especí- */
22 /* -fico e conforme a aplicação em questão */
23 /*****
24   return ((Error_class*)0);
25 }
26
27 /*****
28 /* 3. Definição do objeto processo (obj_medida) - usado */
29 /*   p/ apontar para a função de leitura "read_medida", */
30 /*   na chegada de uma primitiva Read.Ind da Unidade */
31 /*   Controladora */
32 /*****
33
34   object_variable obj_medida;
35
36   inicia_vfd_mai()
37   {
38
39 /*****
40 /* 4. Inicialização das operações sobre cada objeto pro- */
41 /*   -cesso - Obj_medida passa a apontar para a */
42 /*   função de leitura da medida (read_medida) */
43 /*****
44
45   obj_medida.read=read_medida;
46
47 /* Inicialização dos demais objetos processos */
48
49
50
51 /* inicialização do dicionário de objetos fonte */
52
53 /*****
54 /* 5. Inicialização do objeto de comunicação "medida" */
55 /*****
56
57   medida.object_index=02;
58   medida.object_code=array;
59   medida.Password=43;
60   medida.access_group=40;
61   medida.access_rights=R&Rg;
62   medida.data_type_index=01;
63   medida.length=01;
64   medida.local_adress=&obj_medida;
65   medida.number_of_elements=10;
66   medida.array_name=medida;
67
68   /* inicialização dos demais objetos de comunicação */
69
70   .
71 }
72 prog_user()
73 {
74   .
75
76   inicia_vfd_MAI();
77
78 /*****
79 /* o fluxo de programa deve entrar neste ponto em um laço */
80 /* de espera dos comandos FMS oriundos da Unidade Controla- */
81 /* -dora */
82 /*****
83
84 }

```

Fig. 7.19 - Exemplo de implementação do Dispositivo de Campo Virtual (VFD) em software

## 7.7. Considerações finais sobre a camada de aplicação de Fieldbus

A camada de aplicação analisada provê a estruturação e a padronização da troca de informação em um Fieldbus, de forma que múltiplas aplicações desenvolvidas por diferentes usuários possam interoperar. A interoperabilidade requer a padronização das variáveis de processo. Por exemplo, como há uma definição padrão para a variável medida (a descrição do objeto array), então o MAI pode ser trocado pelo de outro fabricante e ainda assim prover a necessária informação para o controle estatístico do processo. Um passo adiante da interoperabilidade é a intercambiabilidade dos dispositivos. A intercambiabilidade requer a padronização do dispositivo. Para que os dispositivos sejam intercambiáveis devem exibir as mesmas características dinâmicas e prover a mesma informação no mesmo formato.

O dicionário de objetos, que contém a descrição de todos os objetos do VFD, converte o código das variáveis comuns a todos os elementos do Fieldbus ao endereço específico onde o procedimento de acesso a variável está realmente armazenado no dispositivo. O endereço físico de tais procedimentos e a forma de implementação de cada um diferem de um dispositivo para o outro, dependendo da arquitetura (hardware) do dispositivo, das necessidades do usuário ou ainda do fabricante do sistema sensor. No exemplo de aplicação em discussão, admitamos que o procedimento de leitura da medida (Medir) seja armazenado no endereço 1500 Hex. Se o fabricante revisa o projeto do MAI, alterando o seu mapa de memória, as características do hardware da placa, incluindo os endereços de E/S da interfaces de processo (A/D's, D/A's etc.) e o endereço físico de tal procedimento é alterado para 1600 Hex, o usuário deverá apenas mudar no atributo Local-Address do OD o endereço para 1600 H, além de redefinir o procedimento de acesso a variável. Se o antigo MAI for trocado, o usuário programador poderá acessar a medida da peça com o mesmo comando "Le", de modo que todos os aspectos que dizem respeito ao sistema de comunicação permanecem inalterados. O efeito da mudança no projeto está confinado no MAI. O dicionário de objetos portanto habilitará o usuário a conectar diferentes sistemas sensores e dispositivos de campo heterogêneos a controladores utilizando o Fieldbus.



## 8. CONCLUSÃO

Junto com as vantagens advindas da introdução dos sistemas sensores e atuadores inteligentes nos modernos sistemas de produção, é necessário destacar as dificuldades ligadas aos aspectos de comunicação. A interconexão dos diversos sistemas sensores e atuadores às unidades de supervisão na manufatura e em processos tornou necessária a definição e utilização de uma interface Fieldbus padronizada que possibilitasse a comunicação aberta na área dos sistemas de produção. Esta solução aberta contribui para a interoperabilidade de dispositivos de campo de diferentes fabricantes possibilitando a integração das mais diversas atividades do ciclo industrial num processo de CIM, desta forma proporcionando o aumento da produtividade e da qualidade dos produtos, requisitos indispensáveis para a competição mundial.

Realizou-se ao longo deste trabalho um estudo amplo e, em tópicos que se fizeram necessários, aprofundado, da problemática de integração operacional de sistemas sensores e atuadores em sistemas de produção, apontando a virtuais usuários de instrumentação de campo os passos para esta integração, tendo como ferramenta suporte de comunicação o Fieldbus. Para isto considerou-se fundamental o levantamento dos requisitos da aplicação, o conhecimento e classificação das opções de integração, os requisitos de comunicação bem como o próprio projeto da aplicação e a configuração do sistema.

Na análise dos documentos e normas que se procedeu procurou-se isolar o quanto possível as especificações funcionais e de desempenho de caráter sistêmico e interno ao Fieldbus (que somente interessam aos implementadores) daqueles parâmetros importantes que são visíveis pelo usuário, dispondo assim para um usuário de automação uma base sobre a qual ele possa nortear, a nível técnico, as ações necessárias para o estabelecimento da integração operacional de sistemas sensores e atuadores utilizando configurações baseadas em Fieldbus. Estes parâmetros são todos orientados para o usuário. Diferem de parâmetros operacionais como relação sinal/ruído ou tempo de espera nas filas de transmissão, que tradicionalmente tem sido utilizados para avaliar serviços de comunicação. Potenciais usuários de Fieldbus estão principalmente

interessados nas manifestações externas dos parâmetros desta rede com o intuito de selecionar a implementação que oferece o melhor desempenho para os requisitos particulares da sua aplicação e a um preço que possa justificar o investimento.

A fim de selecionar uma configuração que proporcione o melhor desempenho global, é necessário se identificar e considerar a relação entre os requisitos da aplicação. Estes requisitos não são todos independentes, de modo que a tentativa de se atender a um deles pode ocasionar a degradação dos demais. Portanto um entendimento destes relacionamentos é necessário para se realizar efetivamente os "trade-offs" necessários na configuração do sistema para a integração. Objetivou-se aqui esclarecer o significado destes requisitos de aplicação e discutir suas implicações no sistema de comunicação.

Ao passo que estes requisitos não são uma lista exaustiva para se comparar as redes Fieldbus, estão entre os mais significativos. As diferenças no desempenho destes barramentos, de acordo com estes parâmetros, geralmente se traduzirão em diferenças nos preços. Uma análise paramétrica de barramentos alternativos deverá habilitar o empresário identificar o custo real de diferentes níveis de desempenho e, portanto, ajudá-lo a selecionar o barramento mais efetivo em termos econômicos para os seus requisitos particulares da aplicação.

Outro aspecto fundamental no contexto deste trabalho foi a análise dos conceitos envolvidos na camada de aplicação da arquitetura Fieldbus, tendo como base a proposta Profibus. Ela se compõe de objetos de comunicação, um dicionário de objetos para gerenciamento e endereçamento destes objetos, de serviços de comunicação e dois componentes de interface. A camada de aplicação permite, devido a sua estrutura de objetos e serviços, o necessário e essencial fluxo de informação para redes locais MAP. Requisitos de compatibilidade com MAP em processo de produção podem ser basicamente atendidos com o uso de protocolos e serviços da camada de aplicação inspirados em MMS, como é o caso do Profibus. Esta solução possibilita um fluxo de informação ao nível superior da hierarquia de comunicação, onde estão as redes MAP, minimizando os custos de adaptação como também

facilitando a interoperabilidade entre os diversos sistemas conectados ao barramento. Muitos dos barramentos disponíveis (p. Ex Bitbus) oferecem ao programa de aplicação simples funções de transporte sem nenhuma preocupação com compatibilidade com os níveis superiores. Neste caso o fluxo de informação vertical, se necessário for, pode ser provido através de estações mestre gateways, incorrendo no entanto em maior complexidade e custo de adaptação. Em certos casos este fluxo vertical é indesejável e desnecessário por perturbar as funções de controle, a exemplo dos barramentos utilizados no controle interno das funções de uma máquina.

Para facilitar o entendimento dos conceitos da camada de aplicação e salientar todas as etapas de projeto da aplicação, foi apresentado o exemplo de um sistema destacando a descrição do dispositivo de campo virtual (VFD). Neste aspecto particular o estudo realizado permitiu desta forma o entendimento dos serviços e objetos, como uma primeira etapa para utilização do sistema.

Destaca-se a relativa dificuldade na compreensão e uso da funcionalidade da camada de aplicação de Fieldbus, considerando que o usuário de chão de fábrica não é um analista de sistemas. Embora a camada de aplicação e a ALI isolam o usuário dos detalhes de comunicação, facilitando a programação da aplicação, cria por outro lado a necessidade de um trabalho prévio de educação da equipe de trabalho para que esta possa compreender os conceitos e detalhes inerentes ao dispositivo de campo virtual, o que se faz indispensável quando da configuração e operação do sistema. Este aspecto fica ainda mais evidente quando se confronta o Fieldbus com as ferramentas já há muito tempo disponíveis e consolidadas, de uso relativamente fácil na programação de aplicações típicas do chão de fábrica;

Em função de possibilitar uma visão abrangente do Fieldbus, as perspectivas para a continuidade deste trabalho são promissoras. Uma possibilidade interessante de um trabalho mais a nível de implementação em células de manufatura, consiste no estudo de utilização do barramento Profibus na interligação de componentes de automação da manufatura. Este desenvolvimento teria possibilidades

de aproveitamento até mesmo a nível de posterior instalação na indústria. Também o estudo da implementação de um sistema de Controle Estatístico do Processo (CEP) utilizando um Fieldbus na interligação dos diversos sistemas sensores constitui uma proposta de trabalho digna de consideração.

Acredita-se haver ainda um caminho considerável a se vencer em termos de perspectivas de padronização, até a consecução e aceitação ampla pela comunidade de usuários de um padrão internacional de Fieldbus. Destaca-se em particular a existência de vários tipos de barramentos de campo cada qual otimizado especificamente para classes de aplicações, para as quais a existência de um padrão internacional de Fieldbus pouco influenciará na decisão pela escolha de um barramento. No entanto mesmo considerando estes aspectos é fundamental a realização de um padrão internacional de Fieldbus, pelos benefícios resultantes inerentes à padronização bem como para a ampla adoção do conceito de Fieldbus pelos fabricantes de sistemas sensores e dispositivos de campo de uma forma geral.

## 9. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Komischke, M. Einbindung von Sensorsystemen in den Informationsfluss integrierter Produktionssysteme; Tese de doutorado; VDI-Verlag, Aachen, 1988.
- [2] Certi CIM - Computer Integrated Manufacturing - Aspectos Relevantes para a Sua Implementação; Certi; 9/1991.
- [3] Groover, M.P. Automation, Production Systems and Computer Integrated Manufacturing. Prentice - Hall, 1987.  
Zimmers, E. E.
- [4] Bernold, T. CIM - Communication/Standardisation/Interfaces; North-Holland, 213.p; 1988.
- [5] Browne, M. Understanding CIM and Making it Work; Production Engineer; 12/1985. "Optimal" Operating Rules for Automated Manufacturing Systems; IEEE Transactions on Automated Control, v. AC-27, n.1; p. 80-6; 02/1982.  
Buzacotti, J.A
- [6] Hirsch, B. E. Comunicação no Ambiente de Manufatura: Aspectos Intra e Interorganizacional; 5º JIAI - Jornada Internacional de Automatização Industrial, São Paulo; 8/1989.
- [7] Weston, R. H. Métodos e Ferramentas para a Integração CIM; Boletim Sobracon nº 42; p.33-42; 1989.
- [8] Lepikson, H. A Padronização e Interação das Unidades de Fabricação, Inspeção e Manipulação de uma Célula Flexível de Manufatura. Dissertação de Mestrado, Labmetro/Ufsc, 210p, Florianópolis; 04/1990.

- [9] Trotter, G. J.                   New Map Networks Can Accomodate Existing Plant Systems; Control Engineering; p. 105-108; 11/1986
- [10] Mendes, M.J.                   Comunicação Fabril e o Projeto MAP/TOP; Editora Kapeluz; Argentina; 1989.
- [11] Hagar, M. L.                   The RS-511 Manufacturing Messaging Service Nears Reality for Layer Seven; Control Engineering; p. 32-34; 10/1986.
- [12] Sweeton, D. C.                 Using RS-511 to Meet the Requirements of Real-Time Communications in the Factory; Control Engineering; p. 111-112; 11/ 1986.
- [13] Pinho, A. N.                   Um Estudo da Especificação de Mensagem da Manufatura (MMS) do Protocolo de Comunicação para Ambientes Industriais Map; Dissertação de mestrado, UFSC/EEL; 12/1988.
- [14] ISO1                           DIS 9506/1: Manufacturing Message Specification - Service Definition
- [15] ISO2                           DIS 9506/2: Manufacturing Message Specification - Protocol Specification
- [16] Desmons, D.                   Field Bus in the Numerical Control of Machine-Tools: How to Get the Best of It; Decotignie, J. D.                 International Workshop on Industrial Automation Systems, Tokyo, Japão; 2/ 1987.
- [17] Mendes, M. J.                 Redes Chão de Fábrica e o Projeto Fieldbus - Curso de Redes de Comunicação de Dados em Ambiente Fabril; 9/1991.
- [18] Wood, G.                       Industrial Local Area Networks, CEN/CENELEC; P. 22-23; Brussel; 11/1988.

- [19] Graube, M. The Carrier Band Network and Mini-Map; Low Cost Solutions; Control Engineering; p. 30-31; 10/1986.
- [20] Pimentel, J. Communications Network for Manufacturing; Prentice Hall, 1990.
- [21] Engel, H.O. Feldbus Normung 1990; Automatisierungstechnische Praxis atp; p. 271-277; 6/90.
- [22] Heiler, K.-U. Einsatzgebiete für den Feldbus; VDI Berichte 728; p. 1-21; 03/1989.
- [23] Schneider, H.-J. AICHEMA 91: Sensorsysteme für die Betrieb-messtechnik und Kommunikation im Feld; Automatisierungstechnische Praxis atp 33; p. 511-528; 10/1991.
- [24] Nantais, T. Lindsay, D. The " Hard Dollars " of Intelligent Transmitters; Control Engineering; p. 6-8; 6/1990.
- [25] Babb, M. Smart Transmitters in Distributed Control: New Performance and Benefits; Control Engineering; p. 120-23; 9/1986.
- [26] Schneider, H.-J. INTERKAMA 89: Sensorsysteme für die Betriebmesstechnik; Automatisierungstechnische Praxis atp 32; p. 55-70; 02/1990.
- [27] Schneider, H.-J. AICHEMA 88: Sensorsysteme für die Betriebmesstechnik; Automatisierungstechnische Praxis atp 30; p. 417-429; 9/1988.
- [28] Raab, H. Digitale Sensoren und Sensorsysteme: Welche Informationen sollen sie liefern ?; Automatisierungstechnische Praxis atp 30; p. 534-544; 11/1988.

- [29] kompass, E. J. Batch Control Systems 1990: PLCs versus DCs; Control Engineering; p. 89-91; 06/1990.
- [30] Stemmer, M.R. Einsatzmöglichkeiten digitaler Feldbus-systeme in geschlossenen, maschineninternen Regelkreisen - Ein Beitrag zur Automatisierung der Qualitätsprüfung; Tese de doutorado; TH Aachen; 07/1991.
- [31] Prince, S.M. Communication Requirements of a Distributed Computer Control Systems; IEE Proceedings, v. 128; p. 21-34; 01/1981.
- [32] Decotignie, J.-D. Pleinevaux, P. Fieldbus in the Hierarchy of Factory Communications: The Limits of a Classical Approach; Workshop on Factory Communications; p. 3-15; 1987.
- [33] Vincent, P. Setting the Field Bus Standard; C & I; p. 107-111; 5/1988.
- [34] Garret, P. H. Analog I/O Design - Acquisition: Conversion: Recovery. Prentice-Hall, Virginia, USA, 1981.
- [35] Michael F. Microprocessors in Industry. New York, Van Hordeski, P.E. Nostrand Reinhold 1984, 523 p.
- [36] Babb, M. (et all) VMEbus - Based Control; Control Engineering; p. 69-95; 08/1987.
- [37] Babb, M. (et all) Multibus-Based Control; Control Engineering; p.113-136; 03/1987.
- [38] Seibert, I. (et all) Std Bus-Based Control; Control Engineering; p. 91-117; 12/1991.



- [39] Laduzinsky, A.J.      STD Bus-Based Control; Control Engineering; p. 53-78; 12/1985.  
(et. all)
- [40] Floyd, R.E.            Data Buses Simplify Microprocessor System  
Stabley, R. C.            Integration for Industrial Control; Control Engineering; p. 75-78; 11/1985.
- [41] Page, C. F.            GPIB for VME Communication; Control Engineering; p. 65-66; 8/1990.
- [42] Kazahaya, M.         Field Bus: New Standard ?; Control Engineering; p.21-24; 10/1987.
- [43] Pfeifer, T.            Ziele und Anwendungen von Feldbussystemen;  
Heiler, K.-U.            Automatisierungstechnische Praxis atp 29;p. 549-557; 12/1987.
- [44] Kazahaya, M.         Addressing and Preselected Variables of the  
Field Bus; In Tec; p. 11 - 24; 9/1987.
- [45] Wood, G.C.            Field Bus, a Developing Low Level Industrial LAN Standard; Presented at EFOC/LAN, Amsterdam, Netherlands, june 23-27,1986.
- [46] Phinney, T.            Fieldbus - The Botton-up Approach to an Open DCS; Control Engineering; p. 52-5; 3/1991
- [47] Pleger, J.            Kommunikationssystem Feldbus; Automatisierungstechnische Praxis atp 28; p. 223-227; 5/1986.
- [48] Pleger, J.A.H.        Feldbus für die Verfahrenstechnik - Anforderungen an Bus und Buskomponenten in der Chemischen Industrie; Automatisierungstechnische Praxis atp 31; p. 164 - 213; 4/1989.

- [49] Decotignie, J.-D. Pleinevaux, P. Fieldbus in the Hierarchy of Factory Communications: The Limits of a Classical Approach; Workshop on Factory Communications; p. 3-15; 1987.
- [50] Komischke, M. Feldbus zur Vernetzung einfacher Automatisierungskomponenten, etz, Vol. 108, nº 15; 1987.
- [51] Wood, G .G. Current Fieldbus Activities; Computer Communications; p. 51-55; 3/1986.
- [52] Warrior, J. Structure and Flexibility for Fieldbus Messaging; Control Engineering; p. 18-20; 10/1988.
- [53] Pimentel, J. Communication Architectures For Fieldbus Networks; Control Enginnering; p.74-78; 8/1989.
- [54] Castilho, N. A. Introdução aos Sistemas Distribuídos - Campinas, Editora Papirus; 1886.
- [55] Giozza, W.F. Redes locais de computadores: tecnologia e (et. all) aplicações. São Paulo: McGraw-Hill, 1986.
- [56] Soares, L.F.G. Redes Locais. Rio de Janeiro: Campus, 1986.
- [57] Tarouco, L.M.R. Redes de Comunicação de dados. Rio de Janeiro: LTC - Livros Técnicos e Científicos Editora S.A., 1984.
- [58] Tanenbaum, A. S. Computer Networks. USA: Prentice Hall international, 1988.
- [59] Babb, M. High Speed Serial Bus Puts Personal Computer in Distributed Control; Control Engineering; p.62-63; 10/1986.

- [60] INTEL Corp. The Bitbus Interconnect Serial Control Bus Specifications; In Distributed Control Modules; 1984.
- [61] Bartos, F.J. Bitbus Finds a Niche in Communication for Control; Control Engineering; p. 90-93; 5/1988.
- [62] Bartos, F.J. Bitbus Is Here Now, But User Awareness Varies Greatly; Control Engineering; p. 84-7; 4/1989.
- [63] Sturm, L.  
Werner, B.  
Rohrmann, P. Analyse und Bewertung des Bitbus für die automatisierte Fertigung; Automatisierungstechnische Praxis; p. 617-627;12/1991.
- [64] Göddertz, J. Profibus-Protokolle; VDI Berichte nº 28; p. 47-69; 1989.
- [65] Profibus Nutzerorganisation Profibus Vornorm DIN V 19245 Teil 1, version 5.0, Alemanha 02/90.  
  
Profibus: Process Fieldbus Produkte und Dienstleistungen.
- [66] Profibus Nutzerorganisation Profibus Vornorm DIN V 19245 Teil 2, version 5.0, Alemanha 02/90.
- [67] Göddertz, J. Profibus: Klöckner Moeller Technical publication for industry and commerce; 10/1990.
- [68] Göddertz, J. PROFIBUS: Kommunikationsmedium der mittleren und unteren Feldgeräteebene, etz Band 109, Heft 7/8; p. 316-321; 1988.

- [69] Göddertz, J.                    Der PROFIBUS als Medium zur offenen Kom -  
Heidel, H.                            -munikation im Feldbereich, Open Communi-  
    -cation on site with PROFIBUS; kongressband  
    11. Internationaler Kongress INTERKAMA, 11.  
    e 12/10/1989.
- [70] Kaster, L.                        PROFIBUS in der Gebaudeautomatisierung;  
Zebermann, Chr.                    Hard und Software; 4/1988.
- [71] Kaster, L.                        PROFIBUS in der nicht eigensicheren Verfah-  
Zebermann, Chr                    renstechnik; CIM Management; 5/1989.  
Rake, H.
- [72] Pfeifer, T.                      PROFIBUS-Anwendungen in der Fertigungstech-  
    nik; Industrieanzeiger nº 40; 5/1989.
- [73] CERTI                            Desenvolvimento de um Fieldbus para inter-  
    ligaçao de dispositivos de mediçao, atuaçao  
    e controle (PC-78) - Relatório final. CERTI  
    - Florianópolis-SC.
- [74] Thomesse, J. P.                Factory Instrumentation Protocol: Model,  
(et all)                              Products and Tools; Control Engineering;  
    p. 65-67; 9/1991.
- [75] Bärnreuther, B.                Mit LAN Über MAP zu integrierten Produk-  
    tionssystemen; Automatisierungstechnische  
    Praxis atp; p.462-470; 10/1987.
- [76] Bärnreuther, B.                EMO 89: Automatisierungstechnik, CIM und  
    Datenkommunikation; Automatisierungstech-  
    nische Praxis atp; p. 81-82; 02/1990.
- [77] Färber, G.                      Entwicklungstendenzen der Mikroelektronik  
    und der Informationstechnik; Automatisie-  
    rungstechnische Praxis atp, nº 31; p. 400-  
    407; 1989.

- 78] Lawrenz, W. Auto-Busse in der Industrie: Einsatz bei Feldbus- und Sensor/aktorvernetzung, Teil 1, Elektronik nº 12; p. 134-137; 6/1990 - Teil 2, Elektronik nº 13; p. 63-69; 06/1990.
- 79] Etschberger, K. Zimmermann, P. Vom Auto in die Industrie: CAN - ein Buskonzept für den industriellen Einsatz; Elektronik nº 12; p.109-114; 6/1990.
- 80] Etschberger, K. Bogotyrow, K. Fleischer, S. Buscontrollerbaustein für echtzeitfähige Netze; Elektronik nº 12; p. 79-83; 6/1990.
- 81] Crestani, M. S. O Fieldbus: a nova tecnologia que revoluciona a automação e a disputa pelo seu padrão; EM ; p. 20-53; 07/1991.
- 82] MAP/TOP Application Program Interface Model Description and Interface Specification Requirement - Map/Top Application Interface Subcommittee; 03/1987.
- 83] MAP/TOP MMS Application Interface Specification - - Map/Top Application Interface Technical Subcommittee; 3/1987.
- 84] MAP/TOP Application Interface Support Functions - - Map/Top Application Interface Technical Subcommittee; 3/1987.
- 85] LCMI Implementação da camada de Aplicação de um Protótipo Fieldbus. Relatório final. LCMI - Florianópolis-SC, junho/1992.
- 86] Böttcher, M. Endl, H. PROFIBUS - Portierungshandbuch Schicht 7; Softing GmbH Version 1.00; 03/1990.
- 87] Softing GmbH PROFIBUS - Protokollsoftware Schicht 2 - Version 2.00; 08/1990.

## 10. ANEXO A - PLACA DO SISTEMA SENSOR "TERMINAL DE ENTRADAS E SAÍDAS DIGITAIS"

Com o intuito de se partir para a fundamentação prática do conceito de integração de sistemas sensores, desenvolveu-se no contexto deste trabalho o protótipo simplificado da placa do sistema sensor "terminal de entradas e saídas digitais". A disponibilidade de interface serial digital torna-o uma interface fracamente acoplada para integração de sistemas sensores a ser potencialmente utilizada em aplicações tais como a descrita no capítulo 7, exercendo o papel dos sistemas sensores MAI e/ou IDP, bastando para isso adaptar ao hardware já implementado a interface com o processo adequada conforme o tipo de sistema sensor. Adicionalmente, constitui uma plataforma de hardware sobre a qual pode-se realizar novos desenvolvimentos, por exemplo, considerando processadores mais velozes e com maior capacidade de processamento.

O Terminal de entradas e saídas digitais constitui uma placa de propósito geral para leitura ou escrita de variáveis digitais tais como sinais ON/OFF e sinais BCD. Dispõe de transceptores EIA-RS-485 permitindo a sua conexão em rede a outras placas ou então a um controlador tal como o definido e descrito no item 4.4.2(d.4). A seguir discutir-se-a em maiores detalhes os aspectos de hardware e software relacionados ao projeto do Terminal.

### A.1 DESCRIÇÃO DO CIRCUITO

O diagrama do circuito do sistema sensor está disposto na fig. A.1. O núcleo da placa é o microcontrolador 8051, o qual pode operar com "clock" de até 12 Mhz (16 Mhz em versões mais recentes), e suas características de "hardware" e "software" permitem usá-lo como um poderoso controlador, com até 64 Kbytes de memória de dados e mais 64 Kbytes de memória de programa, isoladamente. Suas características de "hardware" incluem capacidade de expansão da memória de programa (inicialmente 4 kbytes internamente) para até 64 kbytes totais, colocando mais 60 kbytes externos, ou então utilizar 64 kbytes de memória externa; permite também expandir a memória RAM dos 128 bytes internos para mais 64 Kbytes externos; possui quatro portas de E/S de

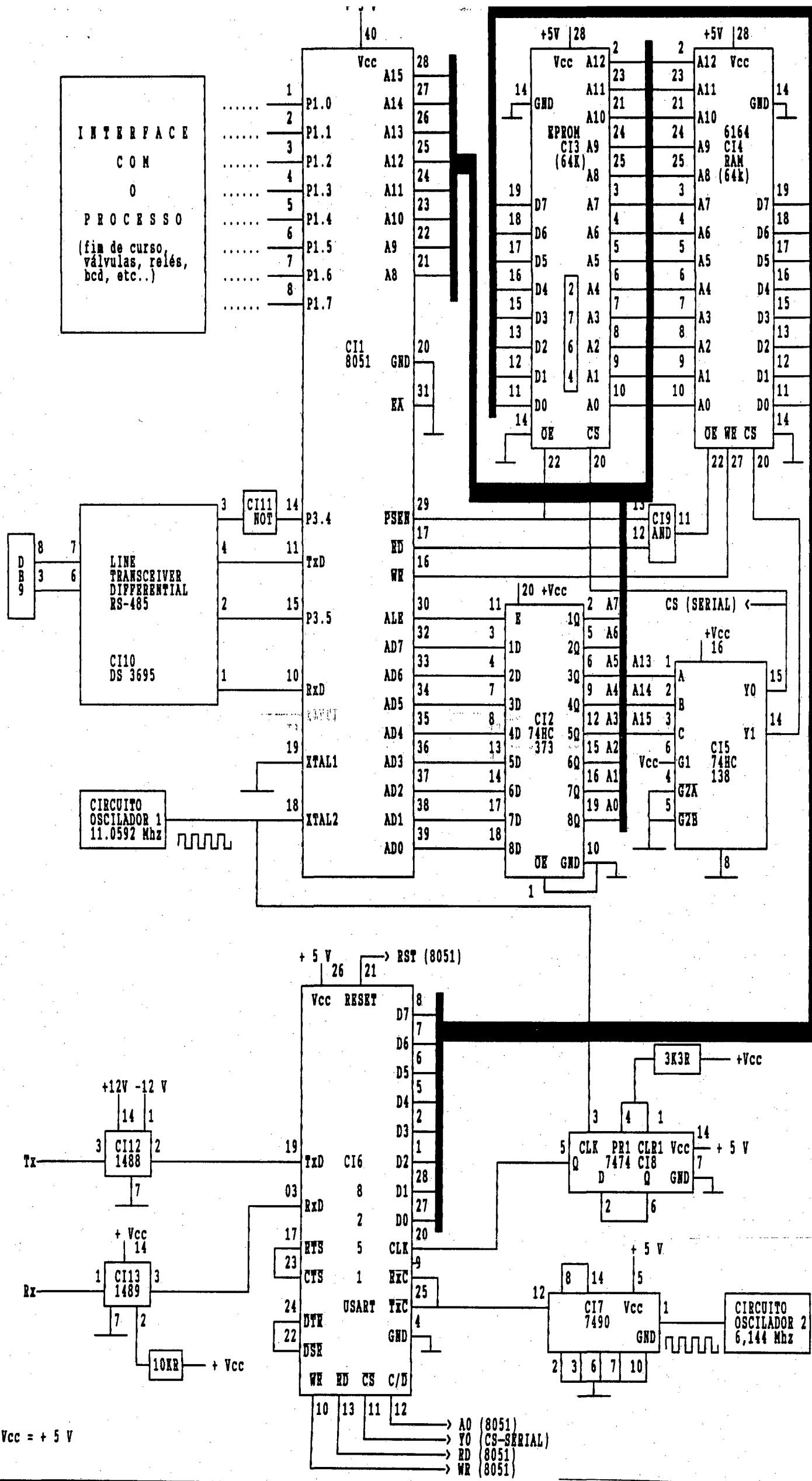


Fig. A.1 Esquema elétrico da placa sistema sensor Terminal de Entradas e Saídas Digitais.

8 bits cada uma, bits estes individualmente endereçáveis; interrupção com estrutura "nesting" (ou seja, uma interrupção pode interromper outra que já estiver sendo atendida, desde que tenha maior prioridade) com 5 fontes selecionáveis e níveis de prioridade a escolher; dois temporizadores/contadores de 16 bits; oscilador de "clock" interno, bastando um cristal e dois capacitores; e também um canal de comunicação serial do tipo UART (transmissor/receptor assíncrono) "full-duplex", que permite também a expansão de E/S. Suas facilidades de programação permitem a execução de complexas operações aritméticas e lógicas (multiplicação, divisão, permuta e deslocamento de bits,...), trabalhar com bancos de registradores nominais e inclusive trabalhar com 128 bits individualmente endereçáveis na RAM. O sinal ALE do 8051 (Address latch enable - pino 30) é utilizado para separar os sinais do barramento de dados dos sinais multiplexados AD0-AD7. O "latch" usado é do tipo 74HC373 (CI-2, octal-flip-flops tipo D) com "buffers" de saída 3-state. No primeiro ciclo de máquina, o microprocessador coloca nesses pinos o byte menos significativo de endereço, que corresponde aos bits A0-A7 e leva o pino ALE a nível lógico 1, de modo que o "latch" coloque em suas saídas esses bits, e logo em seguida leva o pino ALE ao nível lógico 0, para que esses bits fiquem "travados" na saída do "latch". Esses bits estarão fixos porque a saída estará habilitada pelo pino OE=0 (Output Enable - habilita a saída). Nesse instante, o microprocessador coloca nos pinos AD0-AD7, o byte de dado, que passará a ser usado pelo sistema diretamente.

A entrada EA (External Address Enable) do 8051 é permanentemente baixa para habilitar o controlador a buscar o código de programa de posições de memória externas na faixa de endereços de 0000H a 0FFFH. Aqui o código do programa está armazenado em uma EPROM, CI3. A interface serial é formada pelo conversor de nível EIA-RS-485, que é conectado diretamente a entrada serial e pinos da porta de saída do microcontrolador 8051.

O circuito conta ainda com um CI interface serial 8251 acoplado a um "driver" RS-232-C que pode ser utilizado para a interligação a um terminal assíncrono de interface com o operador. Na fase de prototipagem e desenvolvimento de aplicativos simples este terminal pode ser utilizado para depuração de programas e testes. Na fase de



operação pode ser utilizado por exemplo para a indicação do estado da estação, valores de medições e do estado do processo. Alternativamente pode-se substituir a RS-232-C por uma RS-485 provendo assim na fase de operação a redundância do canal serial, para se obter maior confiabilidade, o que é especificação opcional na norma Profibus.

Esta arquitetura caracteriza-se por ser a de menor custo para estações escravas por utilizar um único processador para a execução das tarefas de comunicação e da aplicação, sendo portanto os transceptores o único "hardware" de comunicação dedicado.

## A.2 INTERFACE COM O PROCESSO

Na porta do microcontrolador podem ser agregados a título de exemplo os seguintes tipos de interface com o processo:

- relés: Utilizados principalmente para tarefas de controle sem realimentação, a exemplo do controle de abertura e fechamento de garras de robôs, gabinete de máquinas ferramentas, ativação ou desativação do líquido refrigerante;

-servo-válvulas: As válvulas hidráulicas e pneumáticas são utilizadas em malha fechada, onde a rápida movimentação é necessária ou onde o acionamento elétrico por motivos de segurança intrínseca não pode ser realizado;

- motores: São utilizados principalmente como atuadores em malha fechada para o controle de posição, momento torsor, força e rotação. Quando um motor de corrente alternada ou motor de passo é utilizado, pode-se gerar um pulso para o acionamento do motor no próprio microcontrolador, e este pode ser acoplado a sua porta de E/S;

- medidor de deslocamento eletro-ótico: Transdutores comercialmente disponíveis oferecem a possibilidade de leitura do sinal através de uma interface BCD. Tais valores podem ser lidos diretamente conectando-se a interface BCD à porta de entrada do microprocessador sem qualquer eletrônica adicional. Quaisquer outros instrumentos de

medição além dos transdutores, como por exemplo paquímetros, que ofereçam saída em BCD, podem ser interligados a porta do microcontrolador 8051.

### A.3 PROTOCOLOS DE COMUNICAÇÃO

Em relação ao protocolo de comunicação, pode-se adotar para a placa em questão as seguintes estratégias:

a) Desenvolver um protocolo simplificado específico: Esta alternativa é apenas justificável quando se objetiva uma maior eficiência operacional, buscando-se implementar um protocolo com a funcionalidade mínima necessária para se atender aos requisitos particulares de uma classe de aplicações, normalmente não satisfeitos pelos protocolos atualmente disponíveis;

b) Implementar protocolo padronizado, a exemplo do Profibus: Esta opção se desdobra em um conjunto de tarefas cuja complexidade pesa desfavoravelmente para sua adoção. A implementação de um protocolo de comunicação baseando-se nas especificações da norma requer um substancial esforço de desenvolvimento (horas/homem) ao se considerar todas as etapas envolvidas, partindo do estudo das normas do protocolo, especificação e implementação do sistema até a integração e testes.

c) Adaptar um protocolo já existente: Tais protocolos já se encontram implementados e testados, como é o caso do Protocolo Profibus desenvolvido pela firma Softing [86,87]. Devido a capacidade limitada de endereçamento de memória de programa da CPU 8051, este protocolo deverá ter a funcionalidade mínima necessária para a execução e considerar apenas a implementação das camadas física e enlace de Profibus. O trabalho de adaptação aqui resume-se basicamente na integração do software do protocolo de comunicação na placa aonde será executado, minimizando-se assim o custo e o esforço de desenvolvimento em relação a alternativa precedente.

d) Adaptar um protocolo implementado em "firmware", como é o caso do Bitbus. Neste caso requer-se substituição do microcontrolador 8051

na placa do sistema sensor pelo componente BEM (8044 - Bitbus Enhanced Microcontroller) da Intel. Esta modificação requer mínimas alterações, haja visto que o BEM possui basicamente a mesma arquitetura interna que o 8051, utilizando inclusive esta CPU e uma interface serial síncrona a qual executa um subconjunto do protocolo SDLC. Visto que este circuito já agrega o protocolo Bitbus em Firmware, esta alternativa representa dentre as demais o menor esforço de desenvolvimento para a obtenção de uma solução funcional de sistema sensor.

#### A.4 POSSIBILIDADES DE MODIFICAÇÕES/OTIMIZAÇÃO

As seguintes modificações podem ser facilmente realizadas nesta placa sensor no sentido de otimizar a sua operação:

- Sensor de queda de tensão: Muitas aplicações podem requerer o armazenamento/salvamento dos dados de medição em situações de queda de alimentação da placa. Os seguintes procedimentos podem ser realizados para este fim:

- a) Utilização de memória RAM com acumulador ou bateria: Representa uma alternativa bastante aceitável a nível de custos, porém a confiabilidade restringe-se à duração da bateria;
- b) Utilização de uma memória eletricamente apagada ou gravada como a EAROM ou EEPROM (Electrical Erasable Programmable ROM): A principal desvantagem é o longo tempo de acesso as posições de memória;
- c) Utilização de uma memória não volátil (NVRAM - Non Volatile RAM): Consiste em uma combinação entre RAM e EAROM. Na ocorrência de queda de tensão os dados da RAM são em poucos milissegundos copiados para a EAROM. Com a volta da tensão de alimentação os dados são novamente copiados para a RAM. Sob o ponto de vista técnico esta é a melhor solução, no entanto os componentes são mais caros.

- isolamento por optoacoplador: A interface elétrica RS-485 pode ser galvanicamente isolada do restante "hardware" da placa. Os dados e linhas de controle são realizados por meio de optoacopladores e a fonte de tensão por meio de um conversor DC/DC de 5 V;
- inserção de um teclado e um display para interfaceamento com o operador: Pode ser utilizado para a inicialização da placa dos valores de variáveis de comunicação e variáveis da operação da placa como também para a visualização no local da instalação de produção dos valores das variáveis digitais sendo monitoradas;
- acréscimo de um conversor A/D ou D/A respectivamente para leitura e atuação sobre variáveis analógicas;
- chaves "dip-switch" para leitura do endereço da estação e da taxa de transmissão: Com isto durante a fase de inicialização da placa tais chaves seriam lidas para se obter o valor do endereço da estação no barramento, configurado pelo usuário, bem como o próprio fator de multiplicação a ser utilizado na inicialização da USART para se obter a taxa de transmissão;
- Acréscimo de maior capacidade de memória: Para armazenar os dados obtidos, ou para se acomodar um protocolo de comunicação de maior complexidade, a memória do cartão pode ser expandida simplesmente agregando-se adequadamente novos circuitos de memória RAM/EPROM ao barramento de dados/ endereços e pinos de controle de leitura e escrita do microcontrolador.

#### A.5 PROGRAMA MONITOR

Foi desenvolvido um programa monitor para a placa o qual permite as funções usuais para a depuração de programas aplicativos que podem ser desenvolvidos para a placa. Tal monitor possui as seguintes funções:

- carga remota de programas e dados: Permite a partir de um computador IBM-PCxT/AT ou compatível carregar-se os programas aplicativos para a memória local da placa sistema sensor com a

interface serial USART;

- pesquisa de "strings" na memória: Procura na memória, no campo de endereços especificado ( endereço inicial e final), uma seqüência de no máximo 16 bytes especificados na linha de comando;
- transferência de dados: Blocos de dados, especificados pelo endereço inicial e comprimento em bytes, podem ser transferidos para uma outra área de memória especificada;
- preenchimento de memória: Uma área de memória especificada é preenchida com o valor de um byte passado na linha de comando;
- execução de programas: Possibilita aos aplicativos carregados na placa serem executados a partir do endereço especificado. Um possível segundo endereço especificado na linha de comando representa um ponto de parada de execução do programa (breakpoint);
- Ajuda: Este comando apresenta uma tela contendo uma referência dos demais comandos do monitor, ilustrando a sintaxe dos mesmos e a descrição da função e forma de utilização.