TESE
712

"Suporte de Comunicação Baseado em Sistema Multitarefa de Tempo Real, para Microcomputadores Compatíveis com IBM-PC e Controle Distribuído de Processos Industriais"

ESCOLA FEDERAL DE ENGENHARIA DE ITAJUBÁ

Escola Federal de Engenharia
Biblioteca
MAUÁ
Itajubá

DISSERTAÇÃO DE MESTRADO

"SUPORTE DE COMUNICACAO BASEADO EM SISTEMA MULTITAREFA DE TEMPO REAL, PARA MICROCOMPUTADORES COMPATIVEIS COM IBM-PC E CONTROLE DISTRIBUIDO DE PROCESSOS INDUSTRIAIS"

LUCIA REGINA HORTA RODRIGUES FRANCO

ORIENTADORES:

PROF. MSC. LUIZ EDIVAL DE SOUZA

PROF. MSC. JOSE ANTONIO CORTEZ

ITAJUBÁ
1992

CLASS 681.323(043.2)
CUTT. F825s
TOMO 712

DEDICATORIA

Ao meu marido e aos meus filhos que souberam compreender e superar a minha ausência durante o meu aperfeiçoamento no exterior.

Aos meus pais que sempre me incentivaram nos momentos mais difíceis.

RESUMO

Neste trabalho, apresenta-se uma implementação de um suporte de comunicação de dados de alta velocidade, para uma rede local de até 128 computadores compatíveis com IBM-PC, destinados à transmissão de dados de até 256 bytes, em aplicações industriais de supervisão em Tempo Real. Utiliza-se para isto, a técnica de programação concorrente e placas de comunicação assíncrona serial RS 422 ou RS 485. Para tanto, resumem-se os tópicos principais sobre sistemas distribuídos, redes de comunicação de dados, núcleo multitarefa em Tempo Real e a implementação do suporte de comunicação.

AGRADECIMENTOS

A todos aqueles que de alguma forma contribuiram para o meu curso de aperfeiçoamento em Nancy, de onde pude aprender novas técnicas para serem utilizadas neste trabalho.

Ao Prof. Luiz Edival de Souza por reconhecer a necessidade de renovar. Por sua dedicação, atenção e paciência , sempre disposto e accessível para discussão, no transcorrer deste trabalho.

Ao Prof. Carlos Alberto M. Pinheiro por sua sempre boa vontade em ajudar.

ABSTRACT

This work presents an implementation of a High Speed Data Communication Support, in Local Network with up to 128 computers IBM-PC compatible. This support is used in Data Transmition up to 256 bytes applied to Real Time Industrial Supervision.

Concurrent Programmation Technics and boards assynchronnous Communication RS 422 or RS 485 are used.

Following, the presentation of Main Topics about Distributed Systems, Data Communication Networks, and Real Time Multitask Kernell, the implementation of proposed Communication Support is described.

CONTEUDO

RESUMO

ABSTRACT

CONTEUDO

FIGURAS

TABELAS

CAPITULO 1 . INTRODUÇAO

CAPITULO 2 . SISTEMAS DISTRIBUIDOS PARA AUTOMAÇAO INDUSTRIAL

2.1. Introdução.....01
2.2. Sistemas Distribuídos.....01
2.2.1. Evolução Tecnológica.....01
2.2.2. Sistema de Controle de Processos.....03
2.2.2.1. Aquisição de dados, Controle e Atuação.....04
2.2.2.2. Configuração.....04
2.2.2.3. Inicialização e Atualização dos Valores de Referência.....04
2.2.2.4. Apresentação dos Dados.....04
2.2.2.5. Supervisão e Controle.....05
2.3. Interligação de Funções Distribuídas.....05
2.4. Velocidade Versus Informação.....08
2.5. Adequaçào dos PC´s aos Ambientes Industriais.....08

CAPITULO 3 . REDES DE COMUNICAÇAO DE DADOS

3.1. Introdução.....01
3.2. Meios de Transmissão.....02
3.2.1. Condutores Metálicos.....02
3.2.1.1. Par Trançado.....03
3.2.1.2. Cabo Coaxial.....04
3.2.2. Fibra Optica.....04
3.3. Tipos de Sinais.....05
3.4. Tipos de Transmissão de Sinal.....06
3.4.1. Codificação em Banda Básica.....07

3.4.2. Banda Larga.....07
3.5. Quadro Comparativo Relacionando Meios de Transmissão e Tipos de Sinais.....08
3.6. Topologia de Redes.....08
3.6.1. Malha Irregular.....09
3.6.2. Estrela.....10
3.6.3. Anel.....11
3.6.4. Barramento.....13
3.6.5. Quadro Comparativo Entre as Principais Topologias.....14
3.7. Métodos de Acesso ao Meio.....15
3.7.1. Introdução.....15
3.7.2. Multiplexagem Síncrona.....16
3.7.3. Multiplexagem Assíncrona.....16
3.7.3.1. Os Acessos Controlados.....17
3.7.3.1.1. Gerenciamento Centralizado.....17
3.7.3.1.2. Gerenciamento Distribuído.....17
3.7.3.2. Acessos Aleatórios.....20
3.7.3.2.1. Redes em Barramento.....20
3.7.3.2.2. Redes em Anel.....22
3.8. Redes Locais Industriais.....23
3.8.1. Introdução.....23
3.8.2. Meios de Transmissão.....23
3.8.2.1. Topologia.....23
3.8.2.2. Meios de Transmissão.....24
3.8.2.3. Métodos de Acesso ao Meio.....24
3.8.2.4. Modos de Transmissão.....25
3.8.3. Redes Locais Industriais no Parque Industrial.....25
3.8.4. Redes Industriais Especiais.....25

CAPITULO 4 . NUCLEO MULTITAREFAS

4.1. Introdução.....01
4.2. Estados das Tarefas.....02
4.3. Prioridade das Tarefas.....04
4.4. Visão Geral do Núcleo Multitarefas AMX 86.....05
4.4.1. Inicialização.....05
4.4.2. Procedimentos de Inicialização.....05
4.4.3. Escalonador de Tarefas.....07
4.4.4. Tarefa de Temporização do AMX e Manuseador de "Clock".....07
4.4.4.1. Controle de Temporização.....07
4.4.5. Procedimento de Temporização da Aplicação.....09
4.4.6. Procedimentos de Atendimento de Interrupção (ISP) e o Supervisor de Interrupção do AMX.....11
4.4.7. Facilidades de Comunicação e Sincronização Entre Tarefas.....14
4.4.7.1. Troca de Mensagens.....15
4.4.7.1.1. Exemplo de Utilização de

"Mailboxes" e Envelopes.....16
4.4.7.1.2. Transmissão de Mensagens.....18
4.5. Programação dos Dispositivos de I/O.....20
4.5.1. Supervisor PC do AMX (PCS).....20
4.5.2. Sistema Operacional (DOS).....20
4.5.3. IBM PC ROM BIOS ("Resident Basic I/O System").....21
4.5.4. Conclusão.....21

CAPITULO 5 . SUPORTE DE COMUNICAÇÃO

5.1. Introdução.....01
5.2. Suporte de Comunicação Proposto.....02
5.2.1. Arquitetura do Sistema.....03
5.2.2. Arquitetura do "Software" das Estações.....04
5.2.3. Arquitetura do "software" do Suporte.....04
5.2.4. Características do Suporte de Comunicação....05
5.2.5. Interfaceamento entre Aplicação e Suporte....07
5.2.6. Procedimentos de Comunicação entre Estações..08
5.2.7. Estrutura da Mensagem entre Estações.....08
5.2.8. Estrutura do "Token" entre Estações.....09
5.2.9. Programação dos Dispositivos de I/O.....10
5.2.10. A Transmissão de Mensagens.....11
5.2.11. Procedimentos de Inicialização.....13
5.2.11.1. Estação Mestre.....13
5.2.11.2. Estação Escrava.....14
5.2.11.3. Descrição dos Procedimentos de Inicialização.....14
5.2.11.3.1. Procedimento de Inicialização Início.....14
5.2.11.3.2. Procedimento de Lançamento de Rede.....15
5.2.12. Tarefas Concorrentes.....16
5.2.12.1. Estação Transmissora (origem).....17
5.2.12.1.1. Tarefa "Main".....17
5.2.12.1.2. Tarefa de Lançamento de Rede..17
5.2.12.1.3. Tarefas da Aplicação.....20
5.2.12.1.3.1. Tarefa Teclado.....20
5.2.12.1.3.2. Tarefa Impr_tr.....21
5.2.12.1.4. Tarefa Intermediária da Transmissão.....21
5.2.12.1.5. Tarefa de Transmissão RS422 / RS485.....22
5.2.12.2. Estação Receptora (destino).....23
5.2.12.2.1. Tarefa "Main".....23
5.2.12.2.2. Tarefa de Lançamento da Rede..23
5.2.12.2.3. Tarefa de Recepção RS422 / RS 485.....23
5.2.12.2.4. Tarefa Intermediária de Recepção.....25
5.2.12.2.5. Tarefas da Aplicação.....26
5.2.12.2.5.1. Tarefa Rectecla.....26

5.2.12.2.5.2. Tarefa Impr_rec.....26
5.2.12.3. Tarefas Especiais do Mestre.....26
5.2.12.3.1. Tarefa de Lançamento de Rede..27
5.2.12.3.2. Tarefa de Controle de Acesso (só existente no Mestre).....27
5.2.12.3.3. Tarefa de Recepção RS422 /RS 485 do Mestre.....29
5.2.12.4. Tarefa Especial do Escravo.....29
5.2.12.4.1. Tarefa de Reenvio de "Token".....30
5.2.12.4.1.1. Sinalização da Tarefa de Recepção RS 422/ RS 485....30
5.2.12.4.1.2. Sinalização da Tarefa de Transmissão RS422/ RS485.....30
5.3. Conclusão.....31

CAPITULO 6 - CONCLUSAO

6.1. Comentários.....01
6.2. Resultados.....01
6.3. Perspectivas Futuras.....02

BIBLIOGRAFIA

FIGURAS

Figura 2.1. Funções Industriais Interligadas 06

Figura 3.1. Meios de Transmissão 02

Figura 3.2. Tecnologia de Transmissão 06

Figura 3.3. Topologia em Malha Irregular 09

Figura 3.4. Topologia em Estrela 10

Figura 3.5. Topologia em Anel 11

Figura 3.6. Reconfiguração do Anel, após Falha de um Nó 12

Figura 3.7. Topologia em Barramento 14

Figura 3.8. Acesso ao Meio 16

Figura 3.9. Método Newhall 18

Figura 3.10. Técnica do "Token" de quadro único 19

Figura 3.11. Roteiro do "Token" num Anel Lógico 20

Figura 3.12. Quadro vazio 22

Figura 4.1. Estados de uma Tarefa 04

Figura 4.2. Operação Geral do AMX86 06

Figura 4.3. Manuseador de "Clock" 08

Figura 4.4. Atuação da Tarefa de Temporização do AMX 08

Figura 4.5. Atuação de um Procedimento Temporizado do AMX 09

Figura 4.6 Temporizadores Periódicos 10

Figura 4.7. Temporização por um Intervalo de Tempo 10

Figura 4.8. Temporização a Espera de um Evento 11

Figura 4.9. Temporização a Espera de um Evento, por um Intervalo de Tempo, que não ocorre ("time-out") 11

Figura 4.10. Atendimento à Interrupção com Disparo de uma Tarefa de Maior Prioridade que a Interrompida 13

Figura 4.11. Atendimento à Interrupção pelo ISP e pelo Supervisor de Interrupção 13

Figura 4.12. Exemplo de "Mailbox" 17

Figura 4.13. Exemplo com Envio de Mensagens 17

Figura 5.1. Arquitetura do Sistema Proposto 03

Figura 5.2. Comunicação Entre o "Hardware" e o "Software" em Cada Estação 03

Figura 5.3. Arquitetura de Controle Distribuído e Supervisão 04

Figura 5.4. Arquitetura do "Software" das Estações 05

Figura 5.5. Camadas Implementadas Pelo Suporte Proposto Segundo o Modelo OSI 05

Figura 5.6. Estrutura da Mensagem de Transmissão da Interface Entre Aplicação e Suporte 07

Figura 5.7. Estrutura de Recepção da Interface Entre Aplicação e Suporte 08

Figura 5.8. Estrutura Para Transmissão da Mensagens de Dados Entre Estações 10

Figura 5.9. Estrutura Para Transmissão do "Token" Entre Estações 10

Figura 5.10. Mensagem Armazenada em Bytes 12

Figura 5.11. Mensagem Endereçada Por Ponteiro de Ponteiro 13

Figura 5.12. Sincronização/Comunicação no Mestre 18

Figura 5.13. Sincronização/Comunicação no Escravo 19

TABELAS

Tabela 3.1. Comparação dos Meios de Transmissão 08

Tabela 3.2. Características de Cada Topologia 15

Tabela 3.3. Características das Redes no Parque Industrial 26

Tabela 4.1. Adequação dos Mecanismos de Comunicação e Sincronização 15

CAPITULO 1 . INTRODUÇAO

Este trabalho objetiva a implementação de um suporte de comunicação de dados, baseado no Sistema Multitarefa AMX86, para o desenvolvimento de aplicações em automação industrial.

Este desenvolvimento utiliza microcomputadores compatíveis IBM-PC, o suporte propriamente dito, placas de interfaces RS 422 ou RS 485, e um programa aplicativo para demonstração.

Propõe-se a utilização deste suporte nas aplicações simplificadas de supervisão em controle de processos. Nestas aplicações, esta implementação fornece uma solução de baixo custo, onde o custo de uma rede de comunicação de dados comercial e de um "software" de supervisão em tempo real inviabilizaria a automação.

No Capítulo 2, ressalta-se a importância dos sistemas distribuídos para a automação industrial através da evolução tecnológica. Aborda-se a configuração de sistemas de controle, as características exigidas dos equipamentos submetidos aos ambientes industriais e a adequação de compatíveis IBM-PC com características industrais para este fim.

No Capítulo 3, focaliza-se a importância de uma rede local de comunicação de dados para a integração da fábrica do futuro. Por esta razão, buscando-se levantar os pontos típicos para uma rede local industrial, faz-se uma rápida análise dos meios de transmissão, das tecnologias de transmissão, das topologias de uma rede e dos métodos de acesso ao meio. E desta forma, finaliza-se com uma rápida análise das características das redes locais industriais.

No Capítulo 4, descreve-se no que consiste a programação de sistemas de controle em tempo real, ressaltando-se a adequação da programação concorrente. O ambiente multitarefa é colocado em evidência. É feito um resumo do Núcleo Multitarefa em Tempo Real utilizado, ressaltando principamente os pontos utilizados no desenvolvimento da programação do suporte de comunicação de dados, tais como: estados das tarefas, controle de temporização, supervisor de interrupção, mecanismos de comunicação e sincronização entre tarefas, programação de dispositivos I/O, etc...

No Capítulo 5, faz-se uma descrição detalhada do suporte de comunicação desenvolvido, a partir de uma análise dos tipos de comunicação necessários para a utilização deste em supervisão industrial em tempo real. Descreve-se a programação de dispositivos I/O utilizada para teclado, vídeo e comunicação

assíncrona serial. Analisa-se o protocolo de comunicação utilizado. Relaciona-se a descrição dos procedimentos e tarefas desenvolvidas com as diretivas de sincronismo e comunicação, usadas para suas implementações, e apresenta-se os resultados mais importantes.

No Capítulo 6, resumem-se os resultados obtidos e as perspectivas esperadas com os subsequentes desenvolvimentos de projetos.

CAPÍTULO 2 . SISTEMAS DISTRIBUÍDOS PARA AUTOMAÇÃO INDUSTRIAL

2.1. Introdução

Procura-se, neste capítulo, caracterizar os sistemas distribuídos genérica e resumidamente, apresentar sua arquitetura e enfatizar seus tipos de tráfego de dados.

2.2. Sistemas Distribuídos

2.2.1 . Evolução tecnológica

O controle antes da revolução industrial era feito artesanalmente pelo próprio artesão ao, por exemplo, esculpir sua peça. As decisões do que fazer eram tomadas pela observação da escultura e pela comparação do seu padrão de qualidade. Deste modo, as decisões eram ditas distribuídas, i.e., ocorriam próximas à peça em produção.

Com o advento da revolução industrial, máquinas com partes mecânicas (como alavancas) foram instaladas para controlar a produção ou o processo. O operário permanecia ao seu lado para decidir como manipular estes controles. Ainda persistia o controle distribuído.

Com a evolução, a pneumática foi incorporada à planta. Possibilitando ao homem permanecer, numa sala central, visualizando as situações transmitidas pela pneumática de várias máquinas. E possibilitando o acionamento de qualquer controle à distância. Tinha-se criado o controle centralizado: o homem controlava várias máquinas à distância.

Com o avanço da eletrônica analógica e digital, a sala central foi ocupada por equipamentos de controle automático e posteriormente por computadores centrais. O controle centralizado persistia.

Com o aumento de entradas, saídas e funções, o computador tornou-se o elemento principal que controlava todas as operações na fábrica. Isto acarretou uma necessidade de maior velocidade, para poder controlar toda a fábrica satisfazendo as exigências de tempo de cada processo; maior capacidade de memória, para poder armazenar todos os programas e dados de toda a fábrica; e alta confiabilidade, para diminuir o risco de uma pane que poderia parar toda a fábrica [1] [2].

Desta forma, com a arquitetura centralizada obtinha-se entre outros[22]:

- alto custo do computador central (grande porte);

- confiabilidade aumentada com outro computador central de "back-up" (também caro);
- instalação onerosa da cablagem. Todas as ligações percorriam a fábrica até o computador central;
- dificuldade de manutenção da cablagem;
- dificuldade e alto custo no desenvolvimento do software;
- dificuldade e alto custo na alteração do software.

Com o desenvolvimento da microeletrônica (fabricação de microprocessadores em larga escala) surgem dispositivos mais baratos com capacidade de execução de algoritmos de controle, dedicados às tarefas específicas, de modo geograficamente distribuídos.

Os dispositivos de aquisição de dados, distribuídos pela fábrica, passam a ser controlados por microprocessadores que se comunicam entre si, por vias dedicadas de comunicação e entregam os seus dados a estações de arquivamento e processamento que centralizam as operações naquela área, ou que concentram os dados de toda a instalação fabril.

A arquitetura do Sistema de Controle passa a se compor por várias Estações de Controle, Aquisição de Dados e uma ou mais Estações de Supervisão interligadas por uma rede de comunicação de dados.

Estações de Supervisão também são equipadas com microprocessadores se comunicando com as Estações de Controle, Aquisição de Dados, manipulando e construindo conjuntos de informações que são organizados e apresentados aos operadores através de telas de vídeos monocromáticos ou coloridos de alta resolução. Assim, os operadores podem então ser capazes de reconhecer rapidamente a situação do processo sob sua responsabilidade, já que as telas resumem inteligentemente, apenas as informações essenciais para cada decisão, e no momento necessário, sem congestioná-las.

Estações de Supervisão e Controle ainda são dotadas de alta capacidade de processamento e são capazes de executarem elaborados algoritmos de controle e efetuarem comandos otimizadores e reguladores na área sob sua responsabilidade, reportando ao operador e aos níveis superiores as decisões tomadas.

Assim, os sistemas distribuídos descentralizam a capacidade de processamento para que um certo grau de decisões com dados e controle locais sejam realizados. Evita-se, deste modo, o custo de transmissões de dados desnecessários e a consequente perda de tempo destas transmissões.

Desta forma, retornou-se ao controle distribuído. Porém, agora, sob o controle da microeletrônica.

Com esta arquitetura obteve-se [1], entre outros:

. facilidade de substituição e alteração dos equipamentos.
. possibilidade de instalação de equipamentos de back-up mais baratos, para o aumento de confiabilidade;
. independência de cada equipamento do sistema como um todo (um equipamento com defeito não compromete todo o sistema);
. condições de conexão dos equipamentos em rede (possibilidade de se colocar somente um equipamento equivalente a um grupo de equipamentos, para trabalhar como back-up, em vez de vários equipamentos);
. modularidade, facilitando as alterações de "hardware", adequando-os às atualizações do processo;
. possibilidade de interfaces padronizadas;
. aumento de confiabilidade dos dados transmitidos, com a comunicação inteligente (protocolos);
. facilidade (relativa) de desenvolvimento, alteração e depuração dos softwares;
. diminuição de cablagem;
. disponibilidade da quantidade e de tipos de dados úteis, para as funções específicas (evitando-se atrasos e distorções na transmissão de informações desnecessárias).

Com estas vantagens, a estrutura distribuída é hoje reconhecida como ideal no mundo industrial.

2.2.2. Sistema de controle de processos

Pode-se analisar o sistema de Controle de Processos através do modelo de controle hierárquico, por este vir de encontro a organização distribuída desejada no campo industrial [1].

Este tipo de controle se divide em um conjunto de elementos de computação hierarquizados, onde cada um tem sua tarefa específica dividida em 3 níveis [3].

. Nível inferior:
Responsável pela interface com o processo.
Onde ocorre:
- a aquisição de dados;
- a transformação dos dados coletados em formatos adequados;
- a transformação de dados de controle calculados;
- a atuação no processo de acordo com os dados de controle.

. Nível intermediário:
Responsável pelo controle do processo.
Onde ocorre:
- a execução dos algoritmos de controle de acordo com os dados vindos do nível superior e inferior.

. Nível superior:
Responsável pela coordenação do controle.
Onde ocorre:

- o cálculo de valores de referência dos laços de controle;
- a execução dos algoritmos de otimização de desempenho;
- a interação homem-máquina;
- a interação homem-processo;
- a coordenação e suporte da comunicação do nível inferior;
- a supervisão (eventual);
- a otimização global (eventual).

A seguir descreve-se, de uma forma sucinta, as funções de um sistema de controle de processos [22].

2.2.2.1. Aquisição de dados, controle e atuação

A aquisição de dados consiste na leitura de sinais digitais ou analógicos, que representam os valores das variáveis de processo. Tais com: vazão, temperatura, pressão, etc...

Estes valores oscilam devido a distúrbios do processo provocados pelo próprio processo ou pelo controle na tentativa de aproximá-los dos valores de referência [3].

2.2.2.2. Configuração

Configurar é definir os tipos de algoritmos e os parâmetros destes algoritmos.

Estes parâmetros podem ser: periodicidade da leitura, periodicidade da atuação, etc...

2.2.2.3. Inicialização e atualização dos valores de referência

Inicializar os valores de referência é estabelecer estes valores para as variáveis do processo como o ideal desejado. Isto deve ocorrer após a configuração e antes da operação.

Atualizar estes valores é alterá-los durante o tempo de operação, resultante normalmente da necessidade de otimização da operação [3].

2.2.2.4. Apresentação dos dados

O acesso aos dados de diversos locais pode ocorrer sob demanda ou não.

Estes são organizados hierarquicamente, para fornecerem diversas saídas. Estas saídas podem ser fornecidas em telas de vídeo, impressoras, painéis, plotters e outros, para uso dos operadores, como também, para armazenamento em discos e fitas magnéticas. Esta organização se faz de modo a agrupar os dados, possibilitando entre outros [3]:

- a visão geral da planta, dividida por áreas;
- a visão de grupos de equipamentos;
- a visão de cada laço de controle;
- a apresentação dos alarmes ou exceções;
- a apresentação de curvas de tendências;
- a apresentação da configuração corrente;
- a apresentação do valor corrente medido;
- a indicação de falhas dos componentes, de erros de programa, de erros de configuração.

2.2.2.5. Supervisão e controle

Funções complexas com alto poder de cálculo e com grande volume de dados, implementam a supervisão e otimização.

Dentre estas funções pode-se citar [4] [3] [5]:

. otimização de consumo de matérias primas;
. gerenciamento de estratégias de produção;
. gerenciamento de estoques;
. otimização de desempenho;
. otimização de gastos de energia;
. tratamento de falhas ou alarmes;
. geração de telas (sinópticos);
. geração de relatórios;
. geração de mensagens de avisos e defeitos;
. apresentação dos dados conforme citados acima (no item 2.1.4. Apresentação dos dados).

2.3. A Interligação De Funções Distribuídas

A interligação de todas as funções distribuídas citadas acima, constitue atualmente uma meta ideal de automação rumo ao CIM (Computer Integrated Manufacturing) [5].

Uma meta que apesar das dificuldades, vem se alcançando cada vez mais com as conquistas tecnológicas no domínio das redes de comunicação. Como já é o caso das redes locais de Controladores Programáveis.

A instalação de novas células de controle e de manufatura, ou a modernização das já existentes, aumenta a produtividade, pela integração das operações e pela possibilidade de um grau cada vez mais alto de automação, visando a integração com o CIM [5].

Um desenho esquemático conforme a figura 2.1, mostra

algumas das diversas funções de equipamentos que podem ser interligados a unidades de hierarquias superiores.

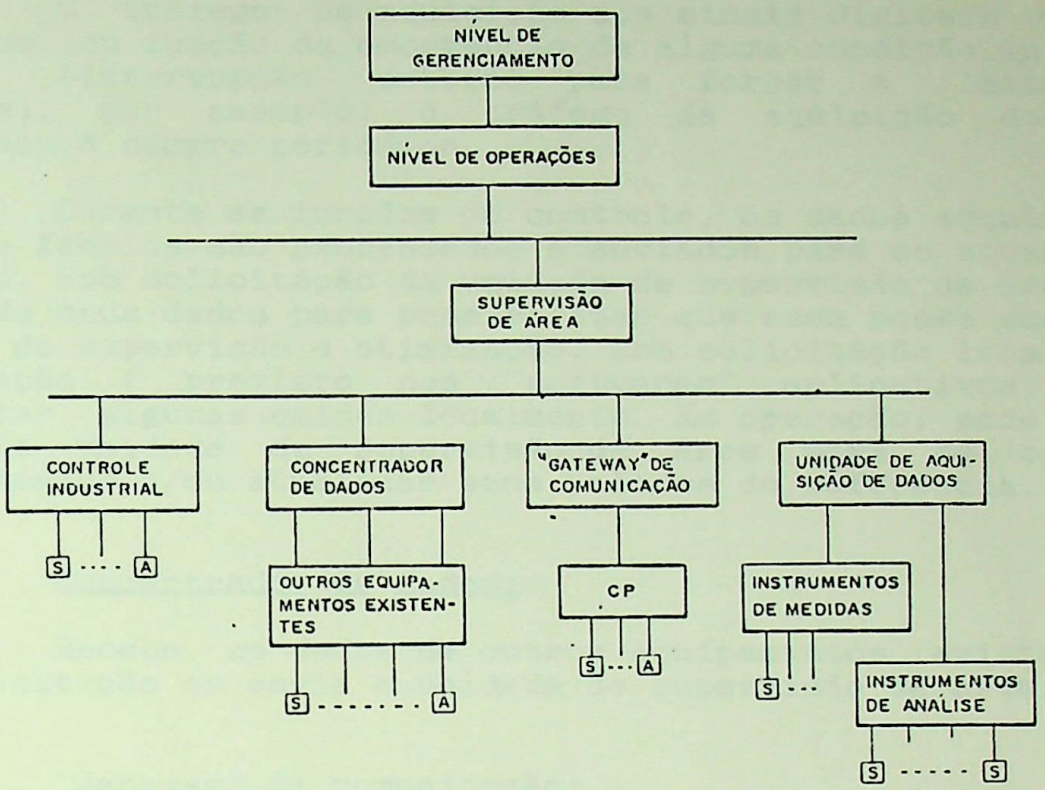


figura 2.1 - Funções industriais interligadas

Uma análise de cada função assim como do tráfego dos dados entre elas pode ser vista a seguir.

Equipamento de controle:

Na área de controle de processos (figura 2.1), as variáveis envolvidas em cada processo devem ser monitoradas tipicamente a cada 100 a 200 ms. Envolvendo, assim, uma velocidade elevada entre a transmissão dos dados dos sensores para seus equipamentos de controle e a transmissão dos dados dos equipamentos de controle para seus atuadores.

Circuitos e "softwares" específicos têm sido usados, para que estas malhas de controle satisfaçam as exigências temporais dos processos físicos.

O equipamento de controle industrial é configurado antes de se colocar a planta em operação ou mesmo já em operação - reconfiguração dinâmica [18]. Ele é, então, inicializado com os valores de referência. Esta configuração e inicialização pode ser feita local ou remotamente. Se remota, as informações são passadas de um outro ponto do sistema, normalmente da unidade de supervisão da área.

Após a inicialização, ele adquire os dados, processa-os e os entrega às tarefas de controle.

O tráfego da aquisição dos sinais digitais pode ser periódico ou função da ocorrência de alguma condição interna ou externa (interrupção externa para forçar a leitura dos sensores), por exemplo: o tráfego da aquisição dos sinais analógicos é sempre periódico.

Durante as funções de controle, os dados adquiridos do chão de fábrica são processados e enviados para os atuadores no processo. Sob solicitação da unidade de supervisão da área, envia alguns de seus dados para possibilitar que essa possa executar as funções de supervisão e otimização. Sob solicitação local ou sem solicitação (previsto nos "softwares" aplicativos) pode apresentar algumas saídas localmente. Em operação, pode receber dados da unidade de supervisão de área para se configurar dinamicamente e/ou atualizar seus valores de referência.

Concentrador de dados:

Recebe os dados de outros equipamentos existentes, e sob solicitação os envia a unidade de supervisão de área.

"Gateway" de comunicação:

Implementa um mecanismo complexo para interligar redes, ou equipamentos à rede, de protocolos diferentes.

Recebe as informações de equipamentos tais como CP´s e as compatibiliza com o protocolo de comunicação da unidade de supervisão de área. Sob solicitação desta, envia os dados para ela.

Unidade de aquisição de dados:

Após configuração local ou remota, adquire os dados dos instrumentos de medidas e dos instrumentos de análise. e sob solicitação da unidade de supervisão de área, os envia, para a unidade receptora.

Unidade de supervisão de área:

A integração dos equipamentos citados anteriormente com os outros da fábrica é feita através da conexão dos mesmos com o equipamento de supervisão. Na área de controle de processos, estes últimos tem como responsabilidade, **programar** os equipamentos de controle (esporadicamente) e **fiscalizá-los** para poder mostrar, normalmente, em tela, como "andam" os processos por ele supervisionados.

Em caso de problemas que incapacitem os equipamentos de realizarem suas funções, a supervisão deve ser capaz de contorná-los, da melhor forma possível segundo as especificações de projeto: acionar equipamentos de "back-up", comandar a desativação de outros processos interligados, ativar programas alternativos, acionar alarmes, etc....

O tipo de dados entre os equipamentos de controle e supervisão, numa situação normalizada é o seguinte: equipamento de controle 1, sensor 1, valor da variável 1, equipamento de controle 2, sensor 2, valor da variável 2, etc...E nesta situação, os dados devem ser monitorados tipicamente a cada 1s aproximadamente.

Em situações de programação ou alteração de algum dado em um equipamento de controle, o fluxo de dados ocorre no sentido do equipamento de supervisão para o equipamento de controle envolvido.

De uma maneira geral, os dados que são enviados para os equipamentos trazem os objetivos da produção e as informações necessárias para atingí-los. E os que são recebidos, levam os resultados das operações: produção efetiva, qualidade constatada e dados das máquinas.

2.4. Velocidade Versus Informação

O requisito chave para o êxito na implementação destas funções integradas é a capacidade de comunicação entre cada elemento da estrutura, onde se devem analisar as características da rede a nível de velocidade e tempo de atendimento, tamanho das informações e número de estações.

Os requisitos de comunicação variam muito dentro do ambiente industrial.

A troca de pequenas quantidades de informação a altas velocidades é a necessidade básica dos equipamentos dos níveis mais baixos que realizam interface para o processo (figura 2.1).

Nos níveis mais altos, entre controle e supervisão, a comunicação exige uma troca de blocos de dados maiores a velocidades mais baixas.

2.5. Adequação dos PC´s Aos Ambientes Industriais

A partir de uma análise do ambiente de fábrica, pode-se levantar algumas condições de trabalho severas para os equipamentos nele utilizados, diferentes das condições dos ambientes refrigerados dos escritórios, citadas abaixo [7]:

. operação initerrupta dos equipamentos;
. ruídos elétricos (de até 2% do nível total de

alimentação);
. umidade relativa (10% a 90% sem condensação);
. vibrações (17Hz a 200Hz a 1,9G);
. temperatura ambiente alta (entre 35 e 60 graus C);
. choques mecânicos;
. instabilidade da tensão de alimentação (superiores a 10%);
. campos magnéticos;
. condições especiais de acondicionamento e espaço;
. ambiente agressivo (pó, fuligem, gases, respingos);
. falta de ventilação.

Até pouco tempo, as exigências para que os equipamentos utilizados na indústria satisfizessem estas condições, descartavam o uso de PC´s, por necessitarem de um "hardware" de alta confiabilidade.

No entanto, na procura da otimização da produção pela redução de custo de produção e pela melhoria de qualidade, buscando-se uma maior integração entre sistemas de controle de processos e sistemas de controle de produção na fábrica, descobriu-se que as adequações funcionais do PC favoreciam sua utilização como, por exemplo:

. ferramenta de apoio ao supervisor de fábrica;
. unidade de aquisição de dados;
. concentrador de dados em células de produção;
. interface homem-máquina para softwares de supervisão e operação;
. "gateways" para comunicação em redes no campo;
. interface homem-máquina entre controladores programáveis e outros equipamentos de controle.

A partir deste momento, a adaptação dos PC´s para o ambiente industrial se consolidou com a oferta de PC´s industriais, mais robustos, projetados com [7]:

. boa resistência mecânica;
. filtros contra poeira;
. chaves de contato liga/desliga;
. circuitos anti-surge (para evitar picos de alimentação);
. ventilação forçada;
. pressão positiva;
. RAM/ROM Disk Card (memórias de massa para locais com alta vibração);
. "Watch-dog", i.e., um temporizador por "hardware" para maior confiabilidade do processo que em operação normal do sistema é iniciado periodicamente, e em condições anormais inicializa a CPU;
. arquitetura "all-in-one", i.e., disposição do "hardware" de modo a facilitar a rápida manutenção em chão de fábrica.

E para trabalhar dentro do conceito BPP ("Back plane"

passivo), garantiram as seguintes vantagens em relação aos PC´s de escritório:

. facilidade de manutenção;
. condições de atualização para processadores mais avançados;
. incremento de confiabilidade;
. menor MTTR ("mean time to repair");
. facilidade de identificação dos problemas;
. e outras.

Adicionado-se a isto, ainda se destaca o fato de que os seus tipos de barramentos - ISA-BUS ou AT-BUS (16 bits) ou EISA-BUS (32 bits), predominam dentro do mercado mundial como padrões pela quantidade de fabricantes de "hardware" que neles se baseiam, tornando o seu "hardware" compatível com a maior parte das placas oferecidas pelos fornecedores. E como consequência destes padrões, pode-se citar a compatibilidade com as redes locais de comunicação facilmente disponíveis no mercado. O que o faz próprio para uso em estruturas distribuídas integradas, nas fábricas, por redes de comunicação. E por último, em razão de muitos "softwares" ligados aos processos industriais necessitarem de operação em tempo real, e estes serem função dos sistemas operacionais, a compatibilidade dos PC´s com DOS, UNIX ou OS/2 se tornou um fator importante para as futuras aplicações [7].

Assim, do ponto de vista destas características, ele se equiparou aos equipamentos específicos de campo já existentes tais como CP, e outros, viabilizando seu uso nos mais diversos ambientes industriais.

CAPITULO 3. REDES DE COMUNICAÇAO DE DADOS

3.1. Introdução

A utilização de computadores no processo da automação trouxe avanços no controle fabril. Mas, a necessidade de integrar toda a fábrica para se poder conhecer seu estado global foi satisfeita com o advento das redes locais de computadores [8].

Deste modo, é possível se conseguir a execução harmônica de todas as atividades no processo produtivo.

As redes de longa distância caracterizam-se por interligar equipamentos em regiões geograficamente distantes. Utilizam como meios de transmissão os mesmos meios usados em telefonia e telecomunicações. Normalmente, redes públicas. O Brasil, faz uso das redes Transdata, Renpac e outras [8].

As redes locais de comunicação de dados, representadas pelas iniciais LAN (Local Area Network), caracterizam-se por interligarem equipamentos próximos geograficamente. Interligam equipamentos situados, por exemplo, numa mesma fábrica ou num mesmo prédio [9] [8]. E desta forma, utilizam seus próprios meios de comunicação sendo geralmente privados.

Pelas suas características, vê-se que as redes indicadas para o ambiente industrial são as redes locais. E por este motivo analisadas neste capítulo.

A rede local é formada por um sistema de "hardware" e "software" que permite que equipamentos de grupos relacionados logicamente possam se comunicar à distância. Esta distância tem o alcance de alguns metros até 2 a 3 km [5].

No ambiente industrial, estes grupos podem ser formados por equipamentos de controle, controladores programáveis, equipamentos de interface homem-máquina, equipamentos de aquisição de dados, concentradores de dados, comandos numéricos, leitoras de códigos de barra, equipamentos de supervisão, etc...

As características de "hardware" de uma rede local são função da topologia e dos meios de transmissão utilizados [3] [5]. As características de "software" são função do controle de acesso ao meio. Por estes motivos segue uma análise com estes enfoques.

3.2. Meios De Transmissão

Um meio de transmissão, chamado também de canal ou suporte de transmissão, é um meio de transmissão de ondas eletromagnéticas portadoras de informações ou sinais [10].

Os meios diferem entre si em relação à banda passante, atenuação do sinal, confiabilidade e custo.

Em redes locais, existem basicamente, os seguintes tipos de meios de transmissão:

. Condutores metálicos:

- par trançado;
- cabo coaxial.

. Fibra óptica.

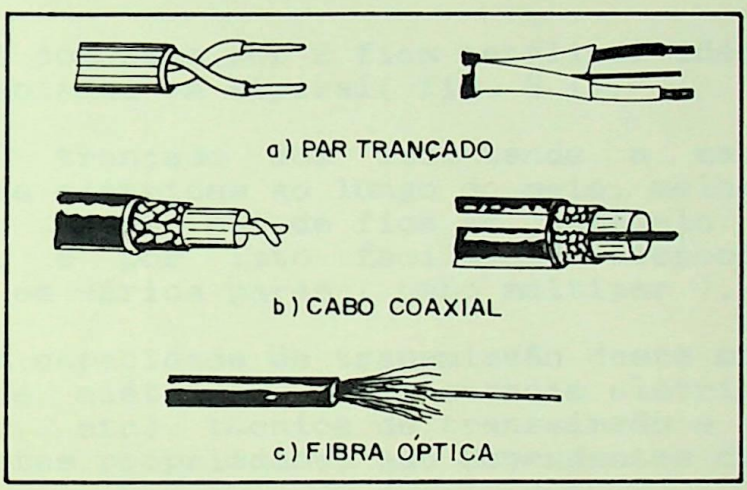


figura 3.1 - Meios de transmissão

3.2.1. Os condutores metálicos:

O condutor elementar de um sinal elétrico é constituído de 2 fios metálicos. Este condutor, ou linha, deve ter uma resistência terminal igual a impedância característica, definida como impedância de entrada de uma linha infinita, a fim de evitar o fenômeno de reflexão na extremidade. Dois tipos de condutores metálicos são utilizados: os pares trançados e os cabos coaxiais [10].

Numa visão superficial, podemos relacionar estes meios de transmissão da seguinte forma:

. par trançado: - ganho em economia de dinheiro, tempo e energia;

- perda em velocidade;

- pouca imunidade.

. cabo coaxial: - ganho em velocidade;

- ganho em versatilidade;

- perda em facilidade de instalação e manutenção;

- imunidade melhorada.

3.2.1.1. **PAR TRANÇADO**

É composto por 2 fios metálicos idênticos (em geral de cobre) enrolados em espiral(fig. 3.1).

O trançado dos fios tende a manter constantes as propriedades elétricas ao longo do meio, melhorando o desempenho em relação a um par de fios em paralelo. Comum no sistema telefônico, e por isto facilmente disponível. Podendo ser encontrado em vários pares (cabo multipar).

A capacidade de transmissão deste meio é função de suas propriedades elétricas (resistência elétrica por unidade de comprimento, etc), técnica de transmissão e qualidade do cabo. Algumas destas propriedades são dependentes do diâmetro **[10]**.

A **imunidade** pode ser melhorada pelo uso de pares trançados blindados. A blindagem diminui os efeitos dos ruídos elétricos externos e limita a emissão de radiofrequência pelo cabo. Assim, um dos maiores problemas dos cabos: diafonia ("crosstalk" - dois pares num mesmo cabo) também pode ser reduzida.

A **ligação de nós** ao cabo é extremamente simples, reduzindo ainda mais o custo de utilização (por exemplo: solda, transformadores de desacoplamento).

Fácil **manuseio** na hora da instalação devido a suas pequenas dimensões e grande maleabilidade.

A **possibilidade de conexões multiponto** é também uma vantagem. Porém, o numero de estações não deve ser muito grande.

3.2.1.2. **CABO COAXIAL**

O cabo coaxial é um meio capaz de operar de forma bidirecional composto de um condutor interno (em geral de cobre) circundado por um condutor externo (geralmente, também de cobre), que serve, ao mesmo tempo como condutor de retorno e como blindagem eletrostática. Um isolante (dielétrico: ar ou algum plástico) circunda os condutores. Duas fitas de aço podem ser aplicadas em espiral sobre o condutor externo para fornecer proteção mecânica e uma melhor blindagem magnética (cabos triaxiais). Aumentando-se, assim, a imunidade às diferenças de potencial entre os aterramentos nas extremidades do cabo.

Sua imunidade é melhor que a do par trançado. Mas geralmente seu custo é maior.

Um cuidado maior relacionado a problemas de reflexão, no momento da introdução de nós no cabo, deve ser levado em conta. E ainda, como consequência, a conexão é feita através de conectores especiais que tendem a encarecer o custo. Numa transmissão ponto-a-ponto a conexão pode ser feita por conectores mecânicos macho-fêmea (tipo BNC). Mas numa transmissão multiponto (topologia em barramento), embora permita conexões em qualquer ponto ao longo do cabo sem alterar a continuidade mecânica do cabo e nem comprometer significativamente a sua continuidade elétrica (" passividade das conexões"), os conectores devem ser bidirecionais especiais (conectores de derivação coaxial ou taps) permitindo injeção e remoção de sinal no cabo coaxial sem grandes alterações nas suas características **[10]**.

Devido às atenuações e reflexões introduzidas pelos conectores ao longo do cabo, há um compromisso entre a taxa de transmissão, comprimento do cabo e potência do sinal. Limitando, assim, o número de nós e o espaçamento mínimo entre eles.

3.2.2. **FIBRA OPTICA**

Uma fibra óptica é constituída por um guia de ondas cilíndrico composto de 2 materiais transparentes de índices de refração diferentes (o externo com índice ligeiramente superior).

A transmissão é feita através de sinal de luz codificado, na frequência do infravermelho emitidos por diodos emissores de luz ("LED´s") ou semicondutores lasers. Como a frequência de transmissão (frequência da portadora) é bem superior às frequências de sinalização usuais, a atenuação do sinal em banda básica no domínio da frequência é constante. Permitindo taxas altíssimas de transmissão. Porém, a atenuação em fibras ópticas, depende do comprimento de onda (ou frequência) da luz utilizada na transmissão.

As regiões espectrais onde esta atenuação é baixa são conhecidas como **janelas de transmissão [10].**

As características mais marcantes da fibra óptica são [8]:

. isolação eletromagnética (ruídos e "crosstalk");
. confidencialidade da transmissão;
. sensibilidade a radiações nucleares;
. unidirecionalidade;
. ganho em segurança;
. adaptável às novas tendências industriais;
. perda em compatibilidade;
. alta imunidade;
. ganho na banda passante.

Em princípio permite ligação ponto-a-ponto e multiponto. Porém, a ligação multiponto requer o uso de acopladores ópticos bem caros. Que devido aos problemas de atenuação de sinal, permite a ligação, atualmente, até 16 nós.

A fibra óptica, para uso em barramento, necessita de 2 cabos (bidirecional). Desta forma, as topologias mais adequadas para seu uso são estrela e anel.

Podemos visualizar os diferentes meios de acordo com as tecnologias disponíveis hoje em dia (Figura 3.2).

. O Sentido da Transmissão

Quando o meio de transmissão existente entre duas entidades, A e B é sempre utilizado no mesmo sentido, a ligação estabelecida entre estas entidades é dita unidirecional (ou simplex). Quando o meio é utilizado alternadamente num sentido e no outro, a ligação é então bidirecional (half duplex). Quando dois meios de transmissão são utilizados um para cada sentido, as trocas podem ocorrer simultaneamente nos dois sentidos, e a ligação é dita bidirecional (full duplex).

3.3. Tipos De Sinais

A informação a ser transmitida pode ser representada por sinais. Nos sistemas de comunicação, consideramos a natureza elétrica destes sinais.

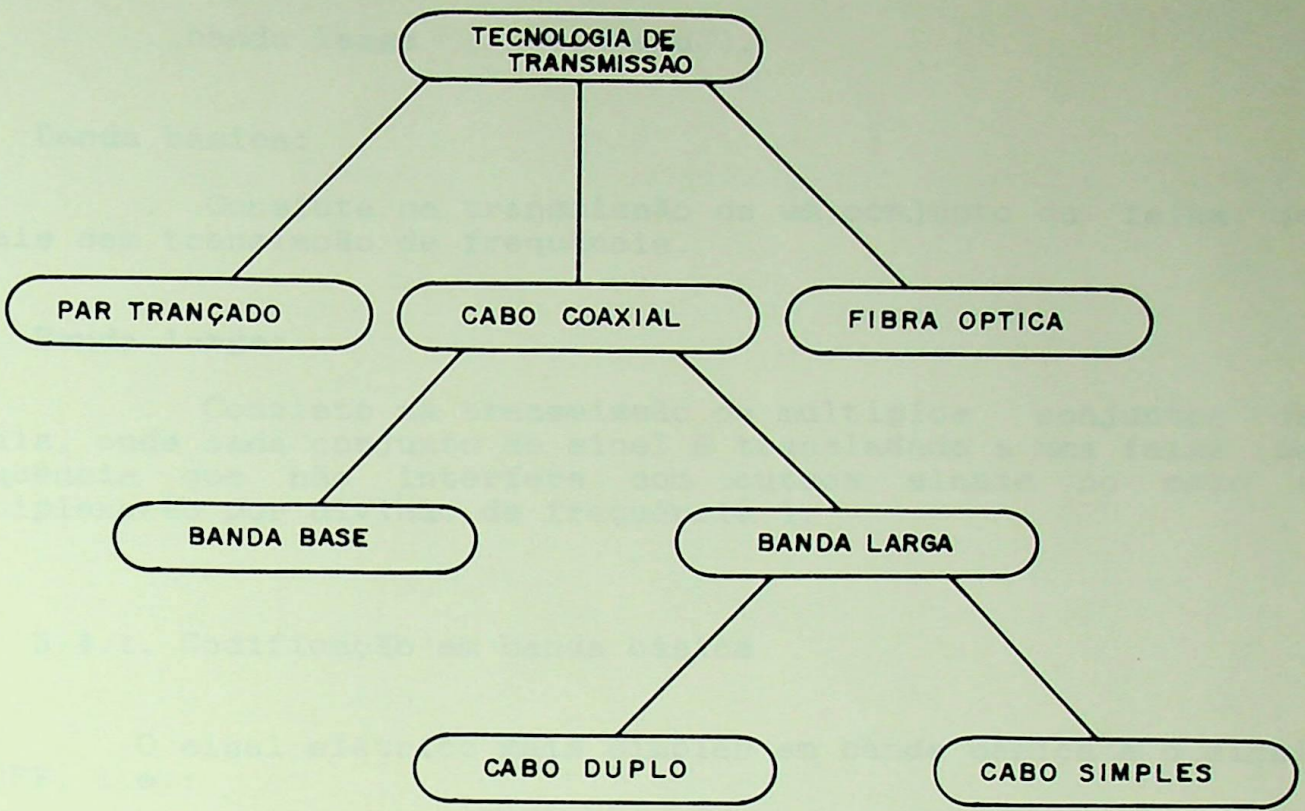


figura 3.2 - Tecnologia de transmissão

Desta forma, temos:

- Sinal analógico:

. aquele que apresenta infinitos valores possíveis de amplitude.

- Sinal digital:

. aquele que apresenta uma quantidade finita de níveis de amplitude.

Os dados de um terminal são exemplos de sinais digitais, que normalmente são submetidos a processos de codificação. Diferem dos sinais originais, porém contém a mesma informação.

3.4. Tipos De Transmissão De Sinal

Existem dois tipos utilizados na transmissão de sinal:

. banda básica ("baseband");

. banda larga ("broadband").

Banda básica:

. Consiste na transmissão de um conjunto ou faixa de sinais sem translação de frequência.

Banda larga:

. Consiste na transmissão de múltiplos conjuntos de sinais, onde cada conjunto de sinal é transladado a uma faixa de frequência que não interfere com outros sinais no meio (multiplexação por divisão de frequência).

3.4.1. Codificação em banda básica

O sinal elétrico mais simples em banda básica é o sinal ON-OFF, i.e.:

. transmissão de um bit 1 => um pulso,
. transmissão de um bit 0 => sem pulso.

Mas, de uma maneira geral, alguns meios distorcem e/ou atenuam muito os sinais digitais em grandes distâncias e em determinadas frequências. Impossibilitando a utilização destes códigos nestas condições.

Assim, é necessário a utilização de MODEM´s analógicos para que se possa transladar os sinais para outras faixas de frequência, e obter altas taxas de transmissão e longas distâncias.

3.4.2. Banda larga

A transmissão em banda larga (até 450 MHz) usa mux de divisão por frequência para dividir um canal físico único em um número de pequenos canais de frequência independentes.

Cada um destes canais pode ser utilizado ao mesmo tempo para dados, voz, vídeo, fac símile, dados de controle em tempo real.

No entanto, no controle de processos industriais e na indústria de manufatura, adota-se, em sua maioria, o sistema de banda básica pelo seu baixo custo de instalação e operação, e

maior facilidade para gerenciamento e manutenção em comparação a um sistema de banda larga.

3.5. Quadro Comparativo relacionando meios de transmissão e tipos de sinais

Um quadro comparativo entre os diversos meios de transmissão e os tipos de sinais, pode ser visto na tabela 3.1.

Característica \ Meio	Par Trançado	Cabo coaxial "baseband"	Cabo coaxial "broadband"	Fibra Óptica
Velocidade Típica	até 1Mbps	até 50Mbps	até 400 Mbps	teoricamente ilimitada
Disponibilidade de Componentes	alta disponibilidade	limitada	alta disponibilidade	bastante limitada
Custo de Componente	o mais baixo de todos	baixo	medio	alto
Complexidade de Interconexão	a mais baixa de todas	baixa	média	alta
Facilidade para ligação multiponto	baixa	média (100 nós)	alta (1000s de nós)	muito baixa
Número de nós	10s	10s a 100s	100s/canal	2 (ponto a ponto)
Relação sinal/ruído	baixa	média	média	alta
Estado da tecnologia	amadurecido	em início de desenvolvimento	em desenvolvimento	bem inicial
Distância máxima de Transmissão	poucas centenas de metros	2,5 km	300 km	100s de km

tabela 3.1 - Comparação dos meios de transmissão [5]

3.6. Topologia De Redes

A topologia de uma rede é caracterizada pela configuração das vias de transmissão existentes entre as diferentes estações.

Ela determina as seguintes características:

. a **conectividade**, que define as ligações que podem se estabelecer a partir de uma estação [5][10];

. a **difusão**, que define a possibilidade de uma estação emitir uma mensagem, para o conjunto de outras estações [10];

. a **seguranca de funcionamento**, definida pelas consequências de uma falha do meio de transmissão ou de uma estação em relação ao funcionamento da rede [5];

. os **custos de implementação [5];**

. o **desempenho do fluxo de dados [5].**

Nas redes locais de computadores, a interconexão das estações é realizada através das interfaces ou nós de comunicação.

A seguir são apresentadas as diferentes topologias existentes.

3.6.1. Malha irregular (ou "store-and-forward")

Cada nó de comunicação ou estação pode ser conectado a(s) outra(s) estação(ões), numa estrutura irregular de interconexão. Geralmente, empregada em redes de computadores à longa distância. Fornece diversas opções de caminhos para solução de problemas de trafego.

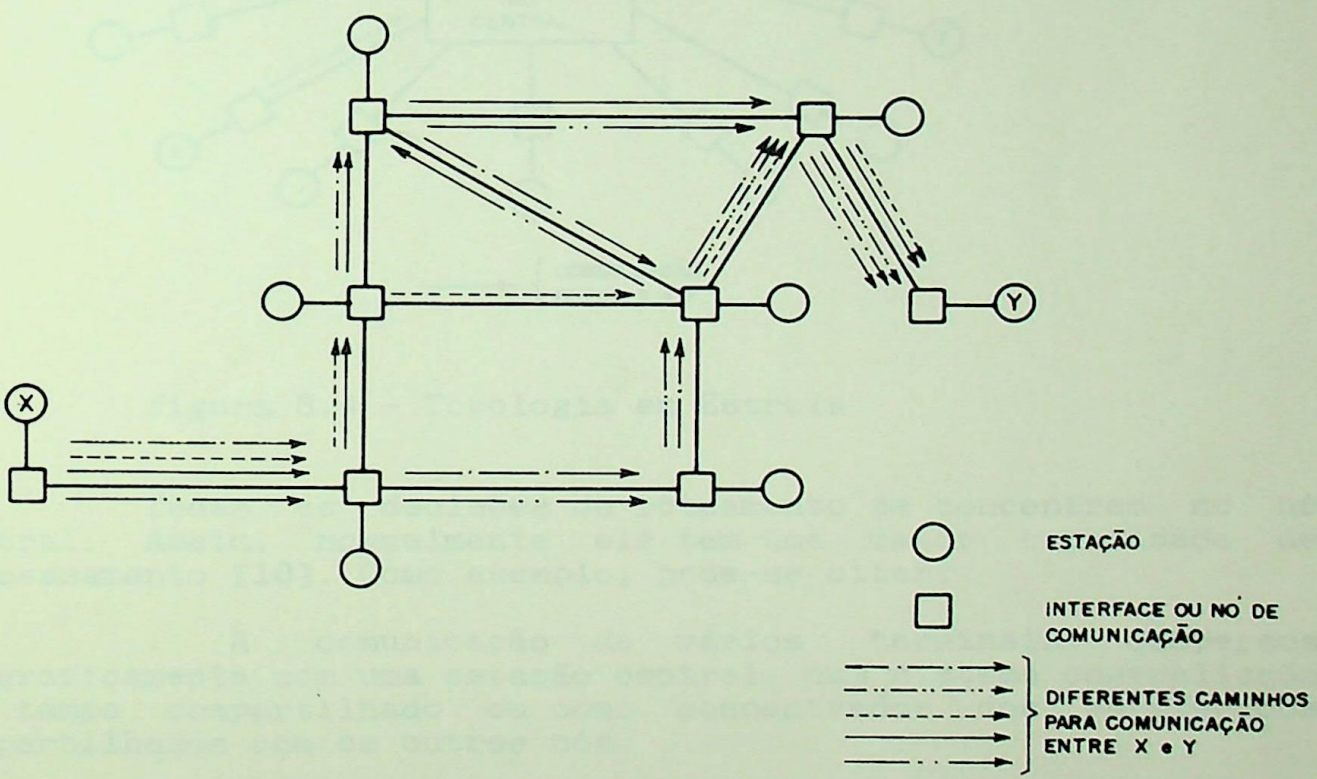


figura 3.3 - Topologia em malha irregular

Porém, nesta topologia, cada nó deve ter condições de armazenar a mensagem e de decidir (roteamento) para onde mandá-

la, em seguida. Assim, cada nó deve possuir uma considerável capacidade de **processamento e de armazenamento das mensagens** [10].

Em relação a rede local, a necessidade de armazenamento e roteamento em cada nó intermediário compromete o desempenho da rede no tocante ao **atraso**. Quanto ao **custo**, esta sofisticação de cada nó compromete a viabilidade econômica da rede local.

3.6.2. Estrela (ou ligação ponto-a-ponto)

Na topologia em estrela, cada estação está ligada a uma estação central. Não há problema de acesso pois as vias não são compartilhadas com outras estações. A comunicação ocorre aos pares, e vários pares podem se comunicar ao mesmo tempo [10][11].

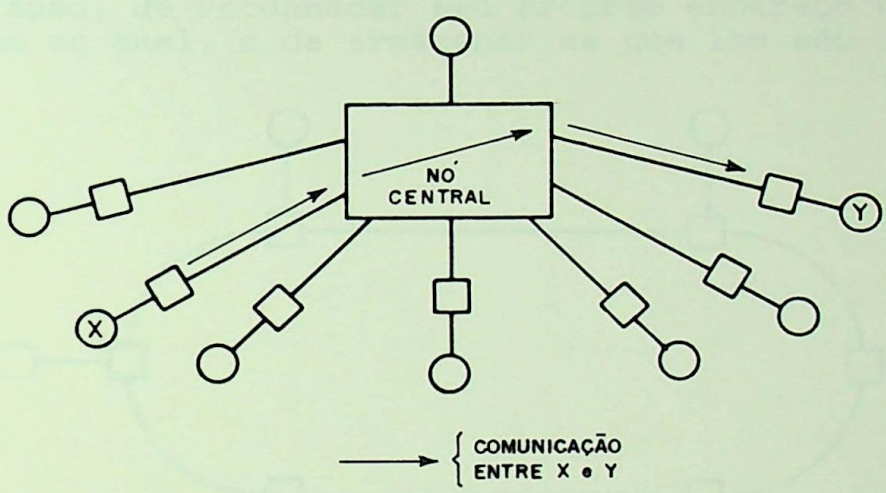


figura 3.4 - Topologia em Estrela

Todas as decisões de roteamento se concentram no nó central. Assim, normalmente ele tem uma maior capacidade de processamento [10]. Como exemplo, pode-se citar:

. A comunicação de vários terminais dispersos geograficamente com uma estação central, num sistema centralizado de tempo compartilhado ou como concentrador dos periféricos compartilhados com os outros nós.

Esta topologia nem sempre é aconselhável, devido à confiabilidade, à complexidade do nó central (que aumenta com o numero de estações que lhe serão interconectadas), e ao número de ligações físicas (custo de instalação, manutenção, etc...) [11].

É uma topologia típica de redes de telefonia privada (PBX) (

taxas de transmissão relativamente baixas) [11].

3.6.3. Anel

A topologia em anel é relativamente simples e facilmente adaptada às redes locais de computadores. Onde o fluxo de comunicação entre as estações tende a ser descentralizado (Figura 3.5).

Não há problema de roteamento.

As mensagens são enviadas do transmissor ao destinatário passando unidirecionalmente pelos nós de comunicação.

Há somente a necessidade, por parte de cada nó de comunicação, de reconhecer seu próprio endereço nas mensagens que circulam no anel, e de armazenar as que lhe são destinadas.

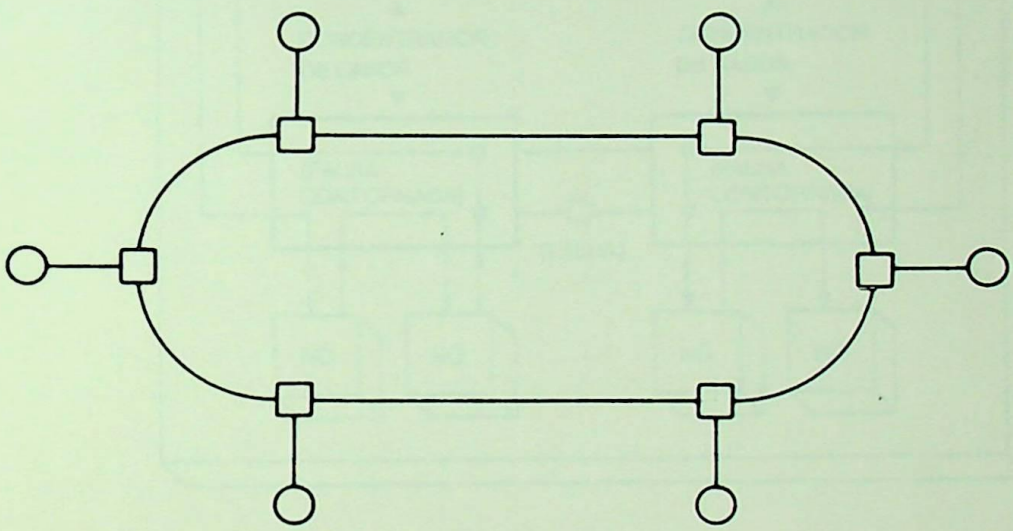


figura 3.5 - Topologia em anel [11]

O desempenho da topologia em anel em relação aos atrasos e o fluxo de dados melhora, comparado ao desempenho das topologias citadas anteriormente. Isto é devido a não necessidade de armazenamento intermediário de mensagen.

Em comparação a topologia em estrela, permite uma redução de custo e complexidade na instalação do meio físico de transmissão. Pois este é comum a todos os nós de comunicação. Constituídos por vários segmentos de transmissão ponto-a-ponto entre pares de nós de comunicação adjacentes [11].

Porém, há necessidade de se compartilhar o meio físico de transmissão, o que acarreta na necessidade da implantação de

mecanismos controladores do acesso e utilização do meio por cada estação.

Assim, na topologia em anel, os nós intermediários participam (como repetidores) da transmissão de uma mensagem entre duas estações não adjacentes. Isto faz com que a confiabilidade da rede dependa dos elementos repetidores (resolvido com anel sobressalente, com sentido reverso de transmissão, em caso de falha ou com o mecanismo físico "bypass") [12].

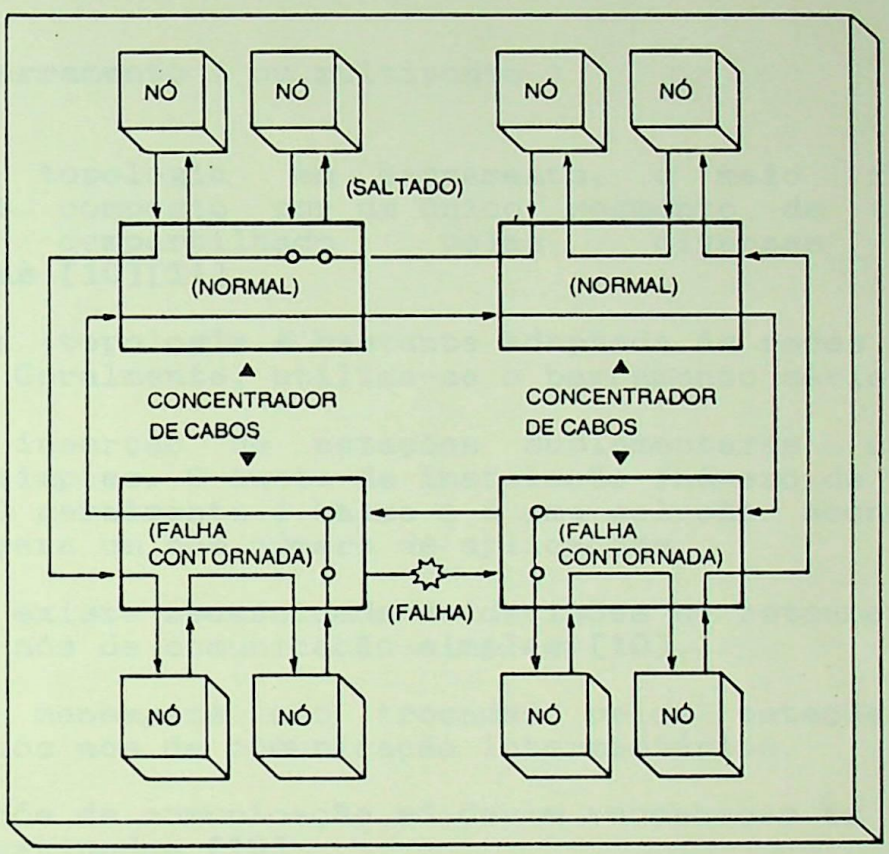


figura 3.6 - Reconfiguração do anel, após falha de um nó [12]

A topologia em anel permite um maior alcance (distância entre estações). Porém, mais estações acarretam maior retardo na rede.

Há uma facilidade em relação a confirmação da recepção da mensagem. Um bit, por exemplo, pode ser acrescentado à mensagem, de modo que ao receber a mensagem a estação receptora inverta este bit. No momento em que a mensagem retornar à estação transmissora, esta verifica se o bit foi invertido, confirmando ou não a recepção.

Há ainda uma facilidade na transmissão de mensagens de difusão: cada estação pode ler a mensagem quando esta passa por

ela.

Porém, devido a um erro, é possível , entre outras coisas, que [10]:

. não se possa saber quem deve ou quem pode transmitir (erro no controle de acesso ao anel);

. uma mensagem fique circulando indefinidamente no anel (erro de endereço de destino).

3.6.4. Barramento (ou multiponto)

Na topologia em barramento, o meio físico de transmissão é composto por um único segmento de transmissão multiponto, compartilhado pelas diversas estações interconectadas [10][11].

Esta topologia é bastante adaptada às redes locais de computadores. Geralmente, utiliza-se o barramento série.

A inserção de estações suplementares numa rede existente é simples. O custo da instalação (número de linhas e acopladores) geralmente é baixo e é uma solução economicamente interessante para um bom número de aplicações.

Não existe necessidade de decisões de roteamento, i.e., utiliza-se os nós de comunicação simples [10].

As mensagens são trocadas pelas estações sem a participação dos nós de comunicação intermediários.

Os nós de comunicação só devem reconhecer as mensagens que lhe são destinadas [10].

Não existe armazenamento intermediário das mensagens. Assim, o seu desempenho não é prejudicado.

Do mesmo modo que na topologia em anel, esta topologia simplifica a instalação e manutenção dos meios físicos de transmissão.

No entanto, há necessidade da implantação de mecanismos que disciplinem o acesso das diversas estações ao meio compartilhado (controle distribuído ou centralizado).

No caso do controle de acesso ser centralizado, a mensagem é enviada a um nó pré-determinado que a enviará ao nó de destino (procedimento semelhante a "polling" ou interrupção, de controle de vários terminais em computadores).

No caso do controle de acesso ser distribuído

(descentralizado) , cada nó é responsável por realizar parte do controle (multiplexação em frequência, tempo ou acesso com "contenção").

Em caso de falha do nó controlador, qualquer nó poderá assumir sua função (melhor que a topologia em estrela).

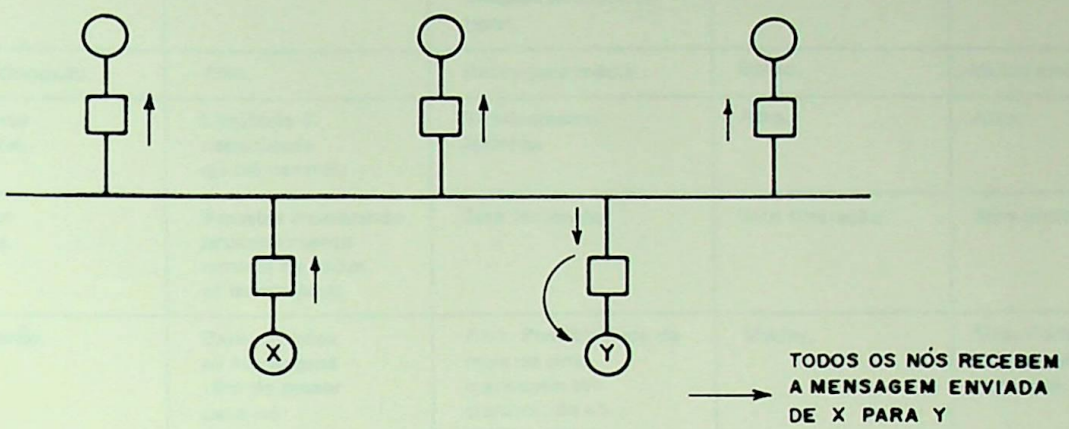


figura 3.7 - Topologia em barramento [11]

Geralmente, os nós de comunicação nesta topologia são passivos, não interferindo na confiabilidade da rede. Mas, também por isto, limitam o alcance da rede exigindo o uso de repetidores para grandes distâncias e muitos nós.

3.6.5. Quadro Comparativo entre as principais topologias

A partir de uma comparação entre as principais características de cada topologia, observa-se que, de um modo geral, cada topologia pode ser classificada conforme a tabela abaixo:

TOPOLOGIAS / CARACTERÍSTICAS	Estrela	Anel	Barra Comum	Grafos (Redes geograficamente distribuídas)
Simplicidade funcional.	A melhor de todas.	Razoável.	Razoável. Une pouco melhor do que o anel.	Extremamente complexa.
Roteamento.	Inexistente.	Inexistente no anel unidirecional. Simples sem outros tipos.	Inexistente.	Bastante complexo.
Custo de Conexão.	Alto.	Baixo para médio.	Baixo.	Muito alto.
Crescimento incremental.	Limitado à capacidade do nó central.	Teoricamente infinito.	Alto.	Alto.
Aplicações adequadas.	Aquelas envolvendo processamento central de todas as mensagens.	Sem limitação.	Sem limitação.	Sem limitação.
Desempenho.	Baixo. Todas as mensagens têm de passar pelo nó central.	Alto. Possibilidade de mais de uma mensagem ser transmitida ao mesmo tempo.	Médio.	Alto. Pode se adaptar ao volume de tráfego existente.
Confiabilidade.	Pouca confiabilidade.	Boa desde que sejam tomados cuidados adicionais.	A melhor de todas. Interface passiva com o meio.	Boa devido à existência de caminhos alternativos.
Retardo de Transmissão.	Médio.	Baixo, podendo chegar a não mais que um bit por nó.	O mais baixo de todas.	Alto.
Limitação quanto ao meio de transmissão.	Nenhuma. Ligação ponto a ponto.	Nenhuma. Ligação ponto a ponto.	Na ligação multiponto. Sua ligação ao meio de transmissão pode ser de custo elevado, como é o caso da fibra ótica.	Nenhuma. Ligação ponto a ponto.

tabela 3.2 - Características de cada topologia [11]

3.7. Métodos De Acesso Ao Meio

3.7.1. Introdução

As topologias em anel e barramento reduzem consideravelmente os problemas quanto à instalação e manutenção dos meios de transmissão. Eliminando a necessidade do roteamento intermediário das mensagens e permitindo a difusão de mensagens entre as estações. Porém, estas topologias básicas necessitam disciplinar a utilização do meio de transmissão.

Os métodos de acesso ao meio físico de comunicação estão intimamente ligados à topologia da rede.

Existem vários métodos diferentes, e alguns são específicos para redes em anel, outros específicos para redes em barramento e outros adaptados aos dois. Os métodos principais podem ser vistos na figura 3.8.

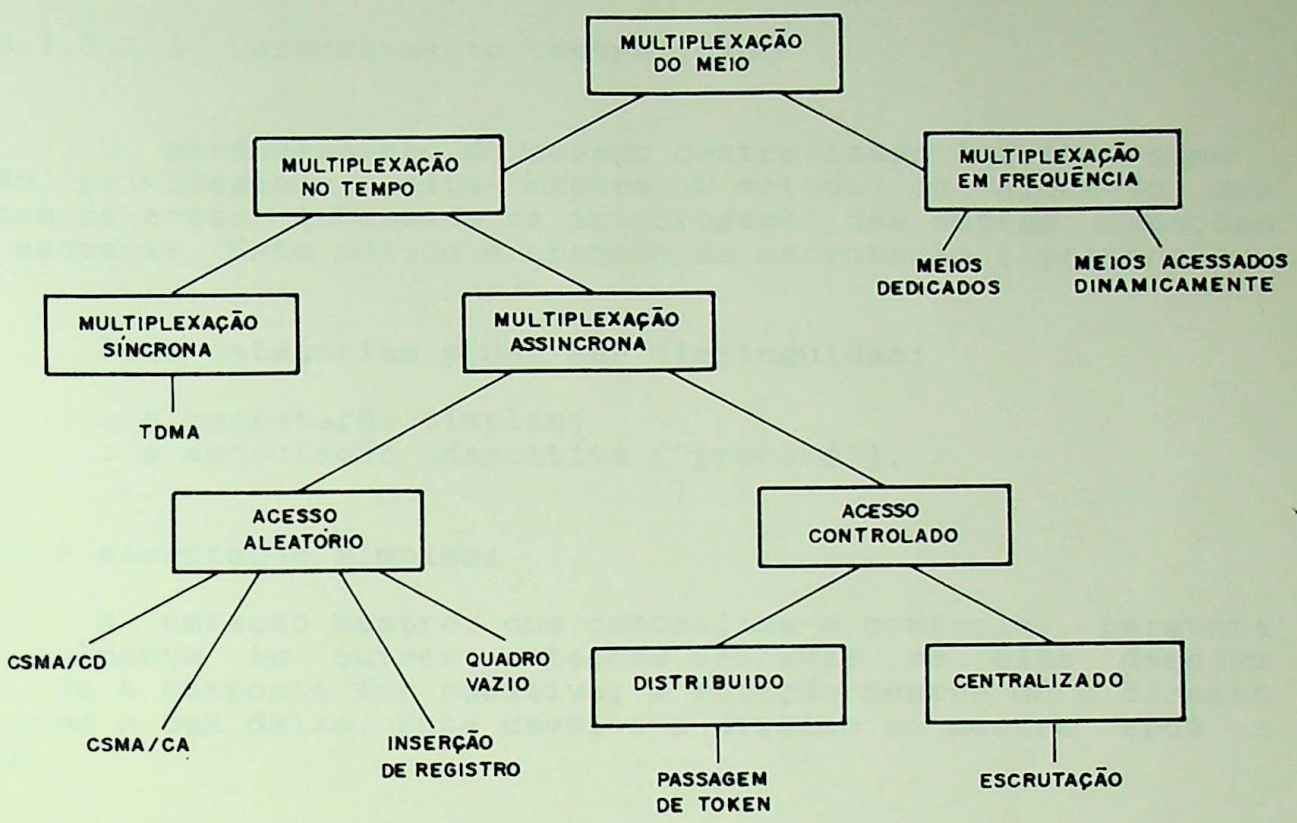


figura 3.8 - Acesso ao meio

3.7.2. Multiplexação Síncrona

A multiplexação síncrona autoriza um acesso múltiplo ao meio de transmissão por uma divisão de tempo em intervalos iguais. Estes intervalos são alocados ciclicamente para os usuários. Este é o método TDMA (Time Division Multiple Access). O endereçamento é implícito. A posição da informação no quadro estabelece quem é o emissor.

Como no ambiente industrial os eventos não são só os síncronos, esta multiplexação não se adapta, e por isto não será detalhada.

3.7.3. Multiplexação Assíncrona

Esta multiplexação se divide em: controle de acesso aleatório e controlado.

3.7.3.1. Os acessos controlados

Os acessos controlados se dividem em gerenciamento centralizado e distribuído.

3.7.3.1.1. Gerenciamento centralizado

O gerenciamento do acesso centralizado é feito numa estação privilegiada, dita mestre. O método de alocação dos direitos de acesso é baseado na interrogação das outras estações ditas escravas. Este método é chamado de escrutação ("polling").

Duas categorias podem ser distinguidas:

. a escrutação simples;
. a escrutação adaptativa ("probing").

a) A escrutação simples:

A estação mestre, que centraliza o controle, pergunta sequencialmente às outras estações escravas se elas desejam emitir. Se a resposta for positiva, a estação mestre dá o direito de acesso a uma delas. Esta devolve o direito ao mestre após a emissão.

b) A escrutação adaptativa:

A estação mestre envia uma interrogação ao conjunto de estações escravas simultaneamente. Cada estação escrava que deseja emitir, responde num intervalo de tempo que lhe é individualmente reservado. O mestre aloca então, sucessivamente o direito de acesso as escravas que responderam. O ciclo recomeça por uma nova interrogação [10].

O gerenciamento centralizado apresenta as vantagens e os inconvenientes habituais ligados a esta estrutura: simplicidade, mas dependência da estação mestre.

Um outro inconveniente, mais específico, está na perda de tempo devido ao mecanismo de interrogação. O gerenciamento centralizado puro tende a só ser empregado em redes de baixo custo de conexão (FIP, BITBUS, etc...).

3.7.3.1.2. Gerenciamento distribuído

Uma estação só pode acessar o meio quando ela for

autorizada, após a consulta entre as diferentes estações. Esta consulta se efetua por circulação de uma mensagem dando direito de acesso, chamado ficha ("token").

A duração da utilização do meio pelo possuidor do "token" é limitado, o que permite definir um tempo de espera máximo para a emissão do pacote, e de qualificar este método como determinístico.

As prioridades de emissão de pacotes podem ser gerenciadas, comparando-se a duração da última rodada do "token" na duração máxima admissível. Um pacote de prioridade alta deve ser sempre emitido, e outros pacotes, eventualmente de prioridades inferiores devem ser emitidos se houver tempo suficiente. O tempo de espera aumenta com o número de estações conectadas. Mas este tempo é praticamente independente do comprimento da rede.

a) A rede em anel:

Numa rede em anel, a ordem de passagem do "token" é imposta pela ordem de implantação no anel. A estação possuidora do "token", que deseja emitir, envia seu pacote com seu endereço e do seu destino. Duas principais variantes são possíveis quanto ao reenvio do "token" [10].

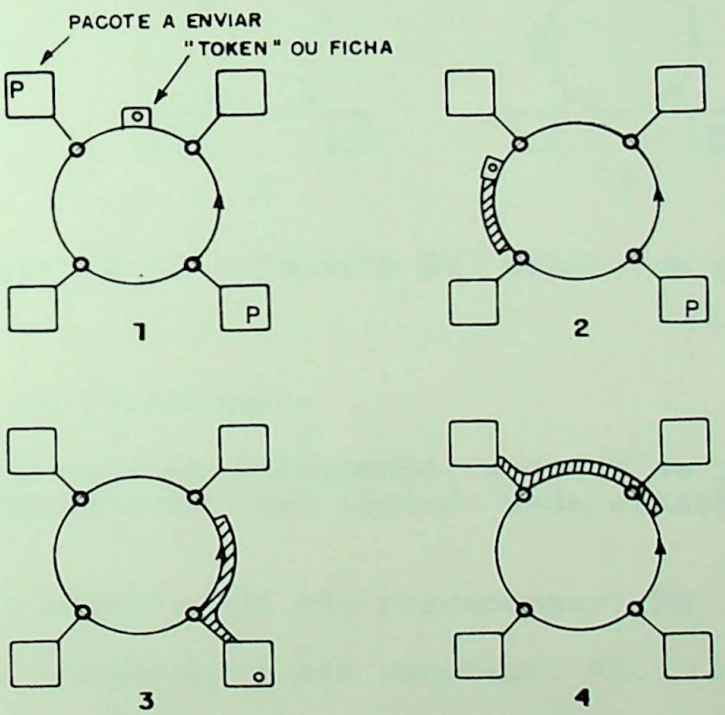


figura 3.9 - Método Newhall [10]

a.1) Ele é emitido imediatamente após o pacote(fig. 3.9), que autoriza a propagação de vários pacotes na rede. Esta técnica é chamada "Newhall" . O quadro pode ser retirado pelo destinatário ao reconhecer o seu endereço. Isto permite diminuir o tempo de transmissão entre cada estação, mas impede a difusão.

a.2) O "token" é devolvido quando o quadro retorna ao emissor (fig. 3.10). Este método tem a vantagem de não necessitar do reconhecimento sistemático do endereço do emissor, e de permitir a difusão com verificação implícita do trajeto no nível da transmissão. Esta técnica, dita quadro único, é utilizada na nova rede local da IBM, e está normalizada pela ISO (8802.5).

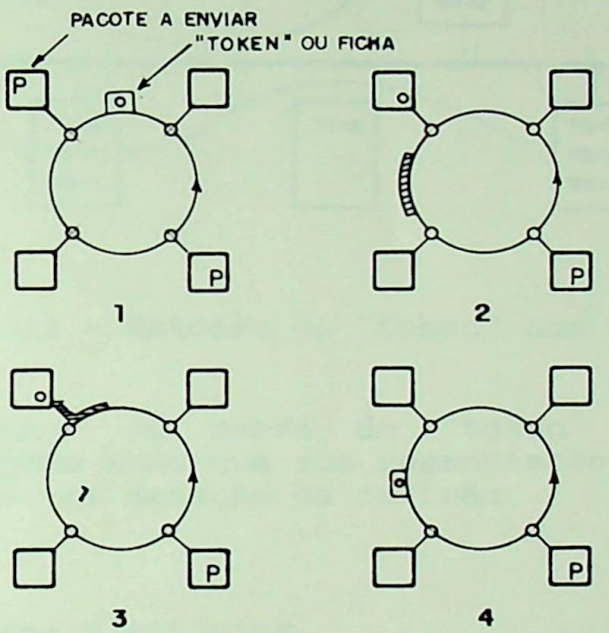


figura 3.10 - Técnica do "token" de quadro único **[10]**

b) Rede em barramento:

Numa rede em barramento, a ordem de passagem do "token" se efetua seguindo um anel lógico. Cada estação ativa guarda na memória:

. o endereço de seu predecessor: PS

. o endereço de seu sucessor: NS.

A figura 3.11, mostra o roteiro do "token" no exemplo. Observe que uma estação pode ser conectada e receber as mensagens sem ser integrada no anel. Por outro lado, uma estação possuidora do "token" pode passar o direito de acesso a outra estação **[10]**.

A reconfiguração da rede se efetua por mecanismos

apropriados:

- inserção de estações por uma mensagem interrogativa do tipo : " novo sucessor ?" emitida periodicamente;

- extração da estação por detecção de seu silêncio, e procura da estação seguinte no anel lógico por uma mensagem do tipo: " quem segue ?".

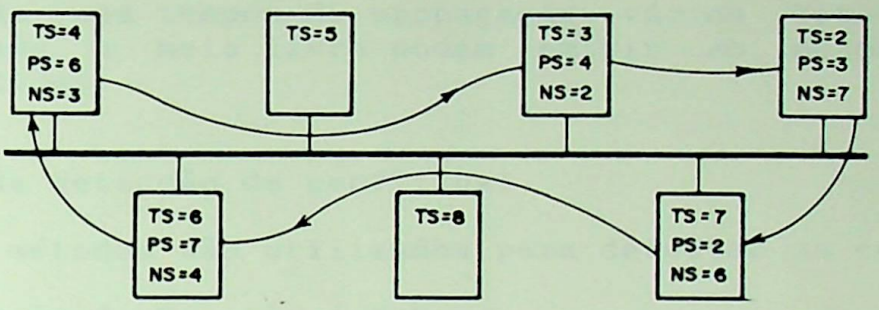


figura 3.11 - Roteiro do "token" num anel lógico [10]

A deteção da perda do "token" se efetua por temporização, do mesmo modo que sua regeneração. Esta regeneração necessita também de uma deteção de colisão.

3.7.3.2. Acessos aleatórios

São divididos em rede em barramento e em anel.

3.7.3.2.1. Redes em barramento

Este método consiste em deixar as estações competirem. Elas acessam o meio por sua própria iniciativa, e os conflitos resultantes de acessos simultâneos devem ser detectados e resolvidos [10].

Devido à sua própria natureza, o gerenciamento por competição apresenta 3 componentes principais:

1. ações efetuadas pela estação que deseja emitir se ela detecta que o meio é respectivamente livre ou ocupado;

2. meios de detecção de conflitos;

3. ações de resolução de um conflito detectado.

Inúmeros métodos de gerenciamento por competição podem

ser implementados. Porém, os mais correntes são apresentados a seguir [10]:

a) ações para emitir:

O método mais simples consiste em emitir sem detecção prévia do meio ocupado, conhecido como ALOHA puro.

O método mais utilizado consiste em emitir, somente se o meio for detectado livre, e adiar a emissão, no caso contrário.

Devido aos tempos de propagação, várias estações que tenham detectado o meio livre podem emitir ao mesmo tempo, provocando colisões.

b) Meios de detecção de conflitos:

Dois métodos são utilizados para detectar os conflitos:

b.1) Detecção de Colisão (C.D.):

Consiste em coletar o sinal transitando no meio, compará-lo com o sinal emitido, e indicar um conflito no caso de diferença.

b.2) Evitando Colisão (C.A.):

Consiste na estação receptora enviar uma indicação de recepção, e a estação transmissora detectar o conflito pela não chegada desta indicação num determinado tempo. Porém, este método só permite a difusão para as estações entre a transmissora e receptora.

c) Ações de resolução de um conflito detectado:

Dois métodos são utilizados: probabilísticos e determinísticos.

c.1) Probabilísticos (os mais conhecidos e usados):

Baseados na parada de emissão em todas as estações em conflito, e por uma política de desencadeamento das ações vistas em a). O método que dá melhor resultado consiste em diferenciar estas novas ações de um tempo função do número de tentativas de emissão do pacote correspondente. Porém, o sucesso da próxima tentativa não estará garantido.

c.2) Determinístico:

Consiste em deixar uma estação prioritária continuar a

emitir após detecção do conflito. Estes métodos são conhecidos como CSMA/DCR (Deterministic Collision Resolution).

3.7.3.2.2. Redes em anel

Dois métodos de acesso aleatório são utilizados:

a) O quadro vazio (figura 3.12).

Os pacotes (quadros de tamanhos fixos) munidos de um indicador vazio/cheio circulam em permanência no anel. Uma estação que deseja emitir espera um pacote vazio, e o completa e altera o indicador para cheio. A estação destino, necessariamente única, remete o indicador para vazio.

Este método é utilizado no anel de Cambridge (Norma IEEE 802.6).

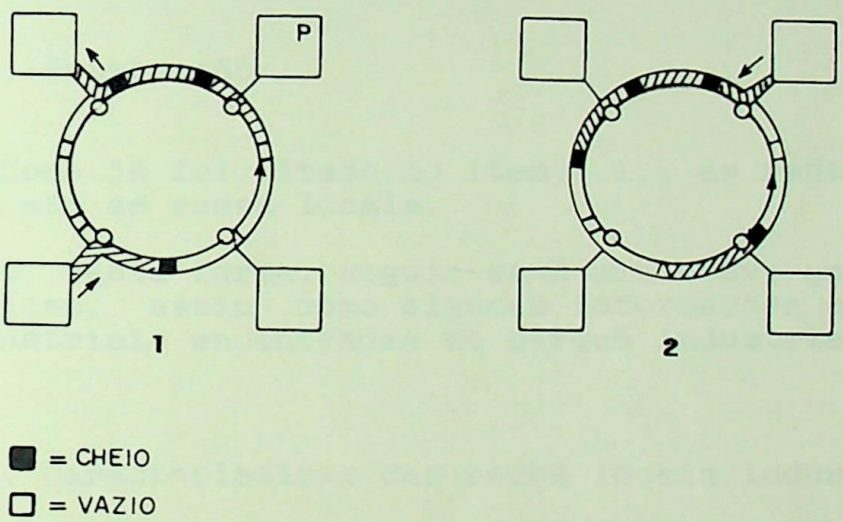


figura 3.12 - Quadro Vazio **[10]**

b) Inserção de registro:

A estação que deseja emitir um pacote, insere-o no anel no momento em que detecta o fim de um pacote em circulação. Os eventuais pacotes que chegam são retardados num registro de atraso. O pacote é extraído do anel pela estação destino ou pela estação transmissora **[10]**.

Comparação:

As técnicas de acesso aleatório garantem uma boa taxa

com uma carga pequena pois, há neste caso poucas colisões. Por outro lado, com uma carga grande, o número de colisões aumenta e a taxa pode cair muitíssimo [10].

As técnicas determinísticas do tipo "token" (ficha), oferecem uma boa taxa mesmo com muita carga. Por outro lado, com carga pequena, os tempos de acesso são mais lentos que nos casos aleatórios, devido ao atraso da passagem do "token" [10].

Atualmente, só as técnicas puramente aleatórias ou controladas estão normalizadas, mas nem sempre elas são satisfatórias (tempo de acesso muito grande ou não garantidos, taxas insuficientes, etc...) [10].

E ainda, os tipos de tráfego de algumas das aplicações tendem a se modificar ou combinar diversas técnicas puras para satisfazerem às necessidades.

3.8. Redes Locais Industriais

3.8.1. Introdução

Como já foi citado no item 3.1., as redes adequadas às indústrias são as redes locais.

E desta forma, seguir-se-á uma breve análise de suas caraterísticas, assim como algumas informações sobre as redes locais industriais encontradas no parque industrial.

3.8.2. Características das redes locais industriais

Como já mencionado, são função da topologia, meios de transmissão, métodos de acesso ao meio e modos de transmissão.

3.8.2.1. Topologia

A topologia em barramento é a melhor configuração, bem adequada a utilização no chão de fábrica. A sua disposição física é semelhante a disposição no chão de fábrica, o que torna sua instalação facilitada [5].

Esta topologia pode ser modificada rápida e facilmente, pois todas as estações tem acesso independente à linha. Isto faz com que as operações de adição e remoção de uma estação não prejudique a operação do resto da rede [5].

Sua manutenção é bem mais fácil que das outras topologias. E os fatores característicos de continuidade elétrica fazem com que seja mais confiável do que a topologia em anel [5].

3.8.2.2. Meios de Transmissão

Como o ambiente de fábrica tem alta incidência de ruído elétrico, e devido também a necessidade de alta capacidade de largura de banda, o par trançado não costuma ser recomendado para aplicações de chão de fábrica em grandes distâncias [5].

O custo médio do cabo coaxial, sua flexibilidade e sua capacidade de largura de banda, tornam-o ideal para estas aplicações. Porém, o seu uso com o sinal de banda básica em sua frequência original pode vir a competir com o ruído elétrico na mesma faixa de frequência existente no chão de fábrica [5].

Contorna-se este problema com a utilização de cabos coaxiais balanceados. Isto, no entanto, exige a utilização da interface elétrica padrão RS 422, o que garante distâncias de até 3 Km no chão de fábrica.

Como a tecnologia da fibra óptica está ainda em evolução, o seu custo é elevado. E para a instalação completa em fibras ópticas são necessários acoplamentos e derivações das fibras [5]. E nestes pontos, as perdas são muito grandes. Desta forma, ainda é mais comum sua utilização em ligações ponto-a--ponto.

É importante notar que a utilização de cabos e estruturas já existentes no local de sua instalação, como cabos de telefone, reduz consideravelmente o seu custo.

3.8.2.3. Métodos de acesso ao meio

O método de acesso ao meio controlado centralizado não é indicado para o chão de fábrica por depender da estação mestra para seu gerenciamento, e desta forma, pode comprometer toda a confiabilidade da rede em caso de pane. Uma alternativa pode ser sugerida, prevendo-se um "software" de detecção de pane na estação mestre e sua substituição automática por outra estação.

O método de acesso aleatório mais utilizado é o CSMA/CD ("Carrier Sense Multiple Access With Colision Detection"). Mas este método não é eficiente no chão de fábrica, pois a medida que a carga aumenta na rede há um crescimento exponencial das colisões. Causando uma degradação rápida no desempenho da rede [5][8][9].

O método de acesso por "token" ou ficha é determinístico. Por isto, pode-se calcular o tempo de espera para o pior caso. Desta forma, ele torna-se confiável sob carga, tornando-se o mais indicado para redes indutriais [5][8][9].

3.8.2.4. Modos de transmissão

A transmissão em banda básica por ser uma implementação mais barata tem sido largamente utilizada em redes industriais. As mais conhecidas são a Data Highways de SDCD's e Ethernet [9]. Ela também é utilizada para melhorar a caraterística sinal/ruído da transmissão em implementação onde os dados transmitidos são modulados em frequência. Esta implementação é conhecida como banda portadora [9].

3.8.3. Redes locais industriais no parque industrial

As principais características das redes locais industriais encontradas no parque industrial são listadas na tabela 3.3.

3.8.4. Redes industriais especiais

Outras redes industriais muito utilizadas são as redes proprietárias de Controladores Lógicos Programáveis (CP's). Estas redes tem como características [5][9]:

- distância máxima até 3 Km;
- tamanho do bloco de mensagem até 255 caracteres;
- número de nós de 64 a 256 bytes;
- taxa de transmissão 57,6K a 800 kbps;
- topologia em barramento;
- cabo coaxial;
- "token" ou ficha.

Outra rede muito utilizada é a ARCnet, que permite uma topologia mista em barramento e estrela. Este tipo de topologia minimiza o comprimento de cabos. É implementada por passagem de "token" ou ficha, de modo cíclico e sequencial com retenção de 24 microsegundos por nó.

Nível / Característica	Empresarial	Planta/Área	Unidade/Célula
Função	Integração nível empresa	Integração de salas de controle Grupo de trabalho	Automação em chão de fábrica
Distância	Muitos quilômetros	Mais de 350 m	Menos de 50 m
Tipo	Rede wide-área	Rede local-área	Sub-rede local *data-highway*
Conexão	X.25, CCITT, satélite, microonda	Ethernet/Map Banda base Banda larga	Linha serial Bitbus/*Field bus* Rede de CLPs
Velocidade	56 kbps a 1,5 Mbps	1 a 10 Mbps	Menor que 19,2 kbps
Protocolo	DECnet, SNA	DECnet, TCP/IP	Vários proprietários
Tipos de estações	Mainframes, Computadores departamentais	Minicomputadores Supermicros, PCs	CLPs, *single loops*, Robôs, CN, Terminais
Número de nós	Milhares	Centenas	Menos de 10
Serviços de	Transferências de arquivos, terminais *program-to-program*, acesso remoto a arquivos gerenciais de rede, compartilhamento de recursos.		Leitura/escrita de dados dispositivos *start/stop upload / download* de programa, *status* de dispositivos.

Tabela 3.3 - Características das redes no parque industrial [9]

CAPITULO 4. O NUCLEO MULTITAREFAS

4.1. Introdução

A programação do controle de processos se traduz em um número de tarefas que devem ser executadas em tempo real. Algumas tarefas devem ser executadas periodicamente ou em determinados horários [13]. E outras devem ser executadas em resposta a eventos assíncronos, resultantes dos eventos provenientes de dispositivos de aquisição de dados, atuadores, sensores, etc...

Pode-se optar, quando se dispõe de só um processador, pelo processamento sequencial. Isto é, pesquisar cada evento e rodar as tarefas uma após a outra. Porém, isto pode causar uma espera maior do que a permitida para que uma tarefa urgente seja executada, enquanto uma outra menos prioritária estiver sendo executada.

Uma outra opção, é a programação multitarefa, onde o núcleo divide o tempo do processador entre as várias tarefas. O gerenciamento deste tempo num sistema de tempo real se baseia na ordenação das tarefas em uma escala de prioridades. O microprocessador se ocupará da tarefa mais prioritária.

Este processamento é conhecido como Concorrente Dirigido por Prioridade [13].

A execução concorrente de tarefas introduz dificuldades que não existem no processamento sequencial, pois uma tarefa pode ser interrompida durante sua execução, cedendo o processador para a execução de uma outra tarefa. Como a ordem das interrupções não pode ser prevista, a ordem da execução das tarefas também não o poderá. E para uma boa cooperação entre tarefas, é necessário prover mecanismos de comunicação e sincronização entre tarefas.

A interação de duas tarefas em relação a um objeto ou é de competidores ou de produtor/consumidor.

São competidores quando necessitam exclusivamente de um objeto. Por exemplo, uma mensagem em um canal serial, não pode ser entrelaçada com outra. O mecanismo que garante o uso exclusivo de um objeto é o de exclusão mútua. Uma tarefa utiliza o objeto, e as outras esperam [13].

São do tipo produtor/consumidor, quando uma tarefa produz um objeto e outra o consome. O consumidor deve esperar o objeto ser produzido para consumí-lo. Isto exige mecanismos de sincronização [13].

Se o objeto produzido/consumido for uma mensagem, é necessário também um mecanismo de comunicação.

Dependendo da estruturação do sistema e da ordem de execução das tarefas numa programação concorrente, pode ocorrer um bloqueio nas suas execuções. Este bloqueio é chamado bloqueio mortal ou "deadlock". Isto ocorre quando recursos distintos forem concedidos de forma indiscriminada a tarefas diferentes.

No caso de exclusão mútua, o bloqueio mortal pode ocorrer quando duas tarefas necessitam do uso exclusivo de objetos que já se encontram em uso pela outra. As duas ficam bloqueadas.

No caso de sincronização, o bloqueio ocorre quando cada tarefa espera uma condição que seria estabelecida pela outra. Neste caso, as duas também ficam bloqueadas. Desta forma, o processamento concorrente reage segundo as prioridades do processo, mas sua programação deve ser feita com muito cuidado.

O núcleo multitarefas utilizado neste trabalho é o AMX 86. Sua escolha deve-se a sua utilização em várias aplicações de supervisão e controle na indústria, no Brasil e exterior. E como foi exaustivamente estudado para este trabalho, ele será detalhado a seguir, conforme [14].

4.2. Estados Das Tarefas

Neste núcleo, as tarefas podem estar em 4 diferentes estados(figura 4.1), como a seguir:

. **Inativa:**
O AMX cria as tarefas da aplicação, mas não as lança para serem executadas.

Estão disponíveis dois modos de criação de tarefas:

- Através da Tabela de Definição da Tarefa:

O AMX cria automaticamente as tarefas nela definidas, porém isto não as lança em execução. Através da Tabela de Definição da Tarefa, constrói-se uma Tabela de Controle da Tarefa (TCB) para cada tarefa, que é constantemente atualizada pelo AMX, descrevendo o estado da tarefa. O AMX retorna um identificador (id) para cada tarefa criada pela Tabela de Definição da Tarefa.

- Dinamicamente:

Pode-se criar tarefas dinamicamente nos Procedimentos de Início e/ou nas tarefas, utilizando-se uma tarefa de definição semelhante a Tabela de Definição da Tarefa [14].

. **Pronta:**

Se após a criação da tarefa, houver uma ou mais requisições suspensas para seu lançamento, a tarefa irá para o estado de pronta. Neste estado, a tarefa tem todas as condições para ser executada, dependendo apenas da disponibilidade do processador.

O lançamento da tarefa é gerado por uma das diretivas:

ajtrig - requisita a execução da tarefa;
ajsend - requisita a execução da tarefa enviando uma mensagem;
ajsenw - requisita a execução da tarefa, envia-lhe uma mensagem e espera pela resposta.

. **Execução:**

Uma tarefa estará no estado de execução se ela possuir o processador.

Uma tarefa pronta possuirá o processador toda vez que ela for de maior prioridade que a que estiver em execução ou quando não houver tarefa em execução, e ela for a de maior prioridade entre as prontas.

Sua execução pode ser interrompida por ocorrência de uma interrupção. Ao final do atendimento à interrupção, se nenhuma tarefa de maior prioridade tiver sido colocada no estado de pronta, ela retornará à execução do ponto em que foi interrompida. Se, porém, outra tarefa de maior prioridade for colocada no modo pronta(resultado da interrupção), a tarefa interrompida irá para o estado de pronta e a resultante da interrupção irá para execução.

Se a tarefa terminar sua execução, e se não houver outra requisição pendente para sua execução, ela retornará ao estado de inativa. Se houver alguma requisição pendente, ela passará para o estado de pronta e como será a de maior prioridade entre as prontas(por isto estava sendo executada) retornará à execução.

. **Espera:**

Neste estado, a tarefa fica bloqueada esperando por um evento qualquer.

Pode-se bloquear uma tarefa em execução, por um intervalo de tempo máximo até a ocorrência de um evento . Se este evento (tarefa, procedimentos de atendimento à interrupção ou temporização) não ocorrer dentro deste tempo máximo, o AMX desbloqueará a tarefa indicando que seu "time-out" ocorreu. O desbloqueio de uma tarefa seja por ocorrência do evento ou por "time-out", levará a tarefa para o estado de pronta.

4.3. Prioridade Das Tarefas

A importância de uma tarefa é expressa em termos de sua prioridade para usar o processador, relacionada a necessidade de se ativar a tarefa logo após a ocorrência de um determinado evento.

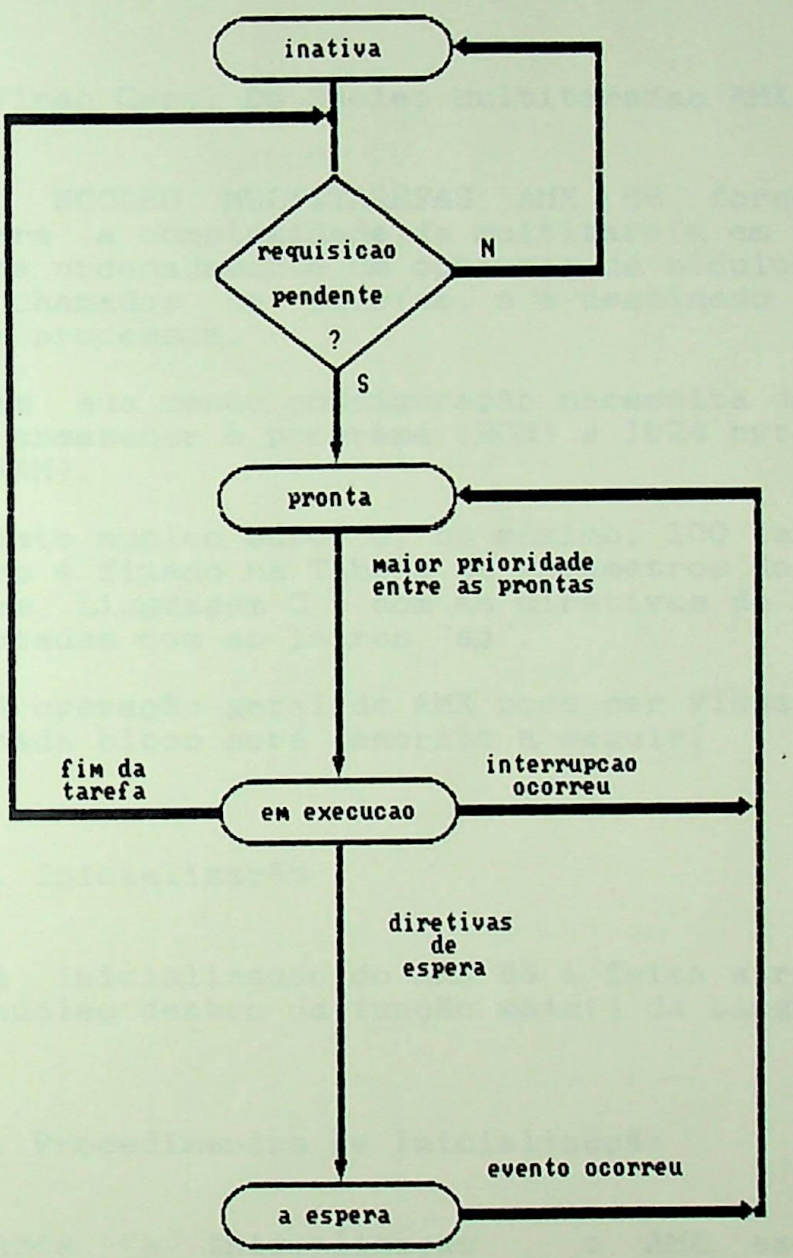


Figura 4.1 - Estados de uma Tarefa [14]

A prioridade de uma tarefa é definida no momento de sua criação e varia de prioridade 0 (mais alta) a 255 (mais baixa). As prioridades de 128 a 255 e a prioridade 0 são reservadas ao AMX. As prioridades disponíveis para as aplicações são de 1 a 127 inclusive. Entre 2 ou mais tarefas de mesma prioridade, o AMX seleciona a tarefa criada há mais tempo cronologicamente.

Uma tarefa pode trocar sua prioridade ou a(s) de outra(s) tarefa(s). Porém isto deve ser feito com muito cuidado.

4.4. Visão Geral Do Núcleo Multitarefas AMX 86

O NUCLEO MULTITAREFAS AMX 86 fornece uma solução simples para a complexidade da multitarefa em tempo real. Ele supervisiona ordenadamente um conjunto de módulos de programas de aplicação chamados de tarefas, e é destinado a ambientes de controle de processos.

Em sua menor configuração necessita de menos de 4096 bytes para armazenar o programa (ROM) e 1024 bytes para armazenar os dados (RAM).

Este núcleo suporta, no máximo, 100 tarefas no sistema (este número é fixado na Tabela de Parâmetros do Usuário - UPT), escritas em Linguagem C e com as diretivas do núcleo que são sempre iniciadas com as letras "**aj**".

A operação geral do AMX pode ser visualizada na figura 4.2, onde cada bloco será descrito a seguir.

4.4.1. Inicialização

A inicialização do AMX 86 é feita através da diretiva **ajentr** do núcleo dentro da função **main()** da Linguagem C.

4.4.2. Procedimentos De Inicialização

Após a inicialização , o AMX executa todos os Procedimentos de Inicialização descritos na Lista de Procedimentos de Inicialização. Cada um deles pode requisitar a execução de uma ou mais tarefas no sistema, mas não pode possuir diretivas do AMX que suspendam ou terminem a execução da tarefa.

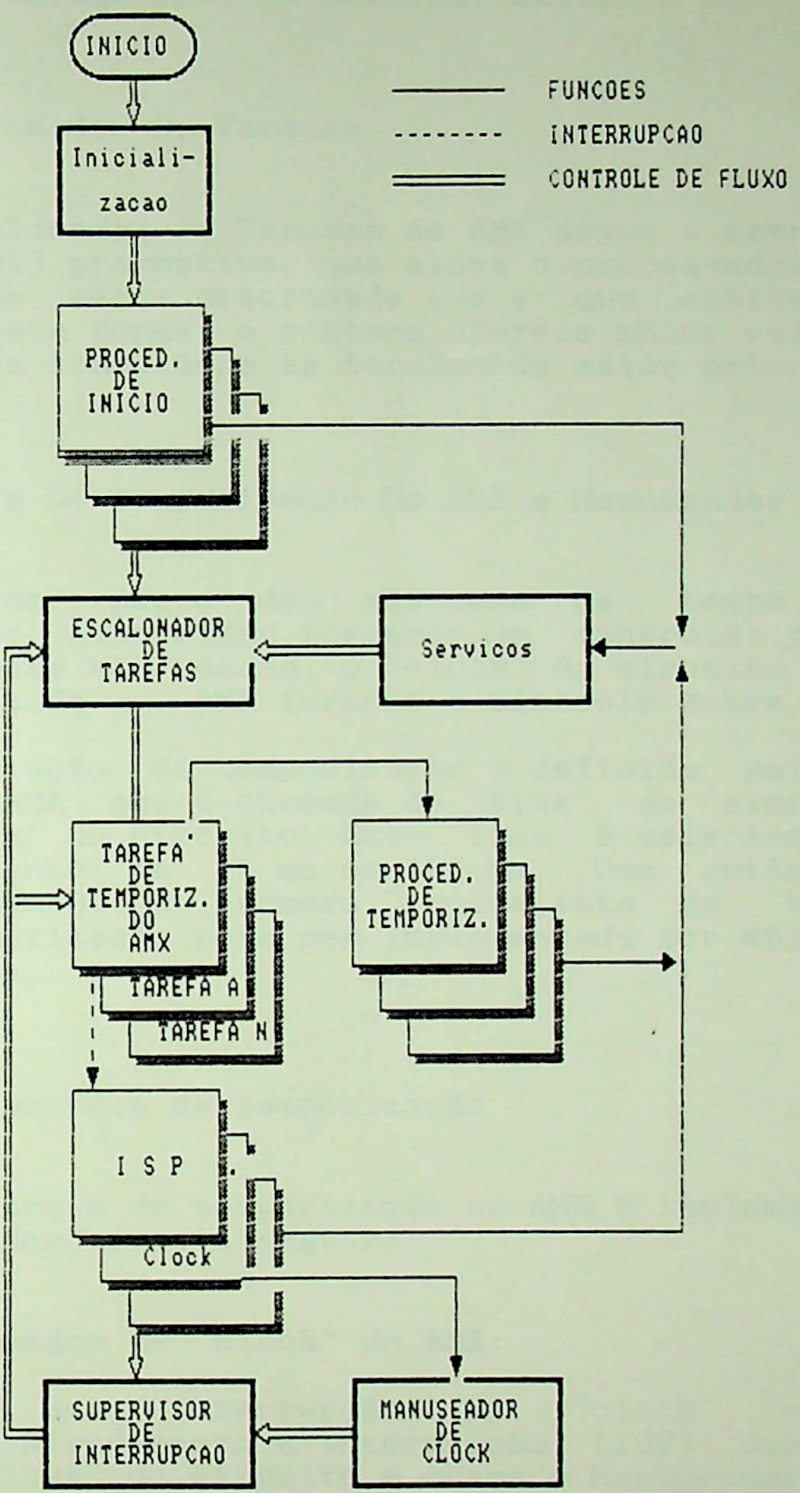


figura 4.2 - Operação Geral do AMX 86 [14]

São normalmente utilizados para inicializarem a configuração de vídeo, placas de comunicação, preparação de telas, testes de integridade de memória, etc..

4.4.3. Escalonador De Tarefas

O Escalonador de Tarefas do AMX segue a estratégia de escalonamento [21] preemptiva, que aloca o processador para uma tarefa pronta de maior prioridade que a que estiver alocada anteriormente. Desta forma, o sistema oferece maior velocidade de resposta a eventos associados às tarefas de maior prioridade.

4.4.4. Tarefa De Temporização Do AMX e Manuseador de "Clock"

A maior parte dos sistemas de tempo real é caracterizada por necessitar fornecer um controle preciso da temporização de suas atividades. O "clock" do circuito fornece a fonte de temporização e o AMX fornece o controle sobre este uso.

A resolução da temporização é definida pela unidade fundamental do AMX que é chamada de "tick" do sistema, e é derivada do "clock" do circuito. Este "tick" é selecionável pelo usuário, tipicamente de 10 ms ou 100 ms. Uma rotina do AMX converte milisegundos no número equivalente de "ticks" do sistema. Uma temporização pode ser implementada por múltiplos de "ticks" do sistema.

4.4.4.1. O controle de temporização

O controle de temporização no AMX é implementado por dois mecanismos, descritos a seguir:

. Manuseador de "clock" do AMX:

Quando uma interrupção de "clock" ocorre, o Procedimento de Atendimento à Interrupção (ISP) desabilita a interrupção do "clock" do circuito e chama o Manuseador de Clock do AMX para disparar a tarefa de Temporização do AMX, se requerida (figura 4.3). A tarefa de Temporização é disparada num intervalo de múltiplos do "tick" do sistema definido, se e somente se, existir qualquer atividade de temporização requerida pendente no sistema. Neste caso, a tarefa interrompida é suspensa e a tarefa de Temporização do AMX começa a execução.

. **Tarefa de Temporização do AMX:**

A Tarefa de Temporização do AMX é a de mais alta prioridade (zero), monitora todas as tarefas que estão no estado de espera temporizada e assegura um melhor controle de tempo das tarefas de aplicação. Como uma tarefa, num sistema multitarefa, pode ser interrompida constantemente e suspendida ocasionalmente para executar tarefas de prioridades mais altas, qualquer tentativa de temporização através de laços dará um resultado incorreto.

Cada temporizador usado pelo AMX, para manter uma tarefa em espera, é mantido na TCB desta tarefa. Os temporizadores são contadores de 16 bits, criados a qualquer tempo. A tarefa de Temporização do AMX decrementa estes temporizadores (se existir algum) nos intervalos dos "ticks" do sistema (figura 4.4). Após a tarefa de Temporização terminar, a tarefa relacionada à temporização volta à execução.

Se algum Procedimento de Temporização do AMX for requisitado (como Procedimento de Relógio ou de Data), o Procedimento de Temporização se responsabiliza em atualizar seus temporizadores (figura 4.5).

Uma vez que todos os temporizadores de tarefa ativos e os temporizadores de intervalos da aplicação tenham sido servidos, a tarefa de Temporização do AMX termina a execução.



Figura 4.3 - Manuseador de "Clock"

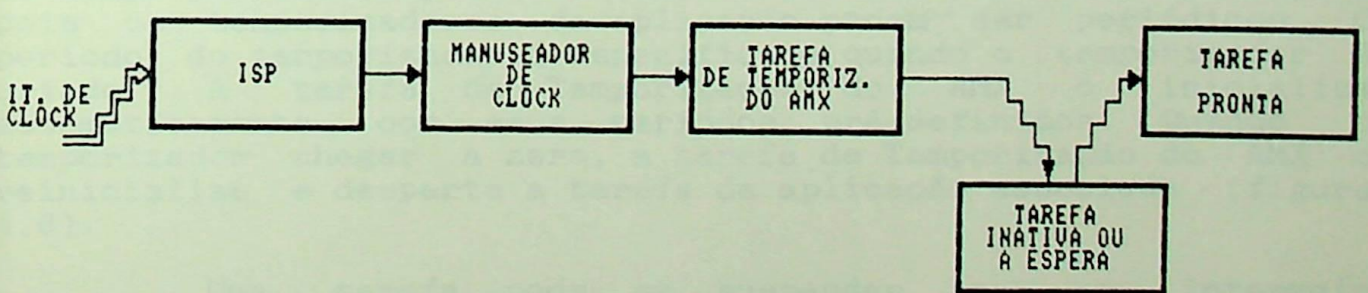


Figura 4.4 - Atuação da Tarefa de Temporização do AMX

O AMX somente pode fornecer as facilidades de temporização se a interrupção de "clock" estiver ativa.

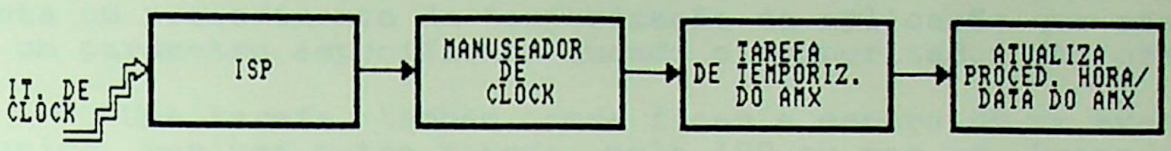


Figura 4.5 - Atuação de um Procedimento Temporizado do AMX

Em alguns casos onde, por exemplo, um periférico é extremamente rápido, pode ser necessário alterar temporariamente a prioridade da Tarefa de Temporização do AMX para uma prioridade menor e modificar a prioridade da tarefa de aplicação deste periférico para zero. Aconselha-se a tentar todas as outras possibilidades, antes de fazê-lo.

A diretiva utilizada para isto:

ajctpr - muda a prioridade da Tarefa de Temporização do AMX.

Normalmente, os dispositivos de alta velocidade são manipulados mais eficientemente por interrupções.

4.4.5. Procedimento De Temporização Da Aplicação

O AMX permite a programação de Procedimentos de Temporização definidos pelo usuário. Os temporizadores da aplicação são fornecidos pelo AMX, mas devem ser criados, inicializados, consultados e parados por qualquer tarefa ou ISP.

Qualquer processo que deva ser executado a intervalos de tempo periódicos pode ser lançado por sua tarefa de aplicação, pois os temporizadores da aplicação podem ser periódicos. O período do temporizador é especificado quando o temporizador é criado. A tarefa de Temporização do AMX o inicializa automaticamente com seus períodos pré-definidos. Quando o temporizador chegar a zero, a tarefa de Temporização do AMX o reinicializa e desperta a tarefa da aplicação associada (figura 4.6).

Uma tarefa pode se suspender por um intervalo específico. A tarefa de Temporização do AMX também serve a todos os temporizadores ativos de intervalo da aplicação. Quando um intervalo de temporização se expira, após o intervalo de tempo

ter-se expirado, a Tarefa de Temporização do AMX executa o Procedimento de Temporização associado. Este procedimento pode chamar um subconjunto de serviços do AMX para enviar mensagens para as tarefas, sinalizar eventos ou despertar tarefas do estado de espera (figura 4.7). Quando o temporizador se expira, o AMX executa um procedimento de temporização da aplicação que passa a ela, um parâmetro especificado quando o temporizador foi criado.

Uma tarefa, também, pode ficar a espera de um evento (produzido por uma outra tarefa, pelo ISP ou por um Procedimento de Temporização) que deva ocorrer dentro de um intervalo máximo específico. Se o evento ocorrer durante este intervalo, a tarefa

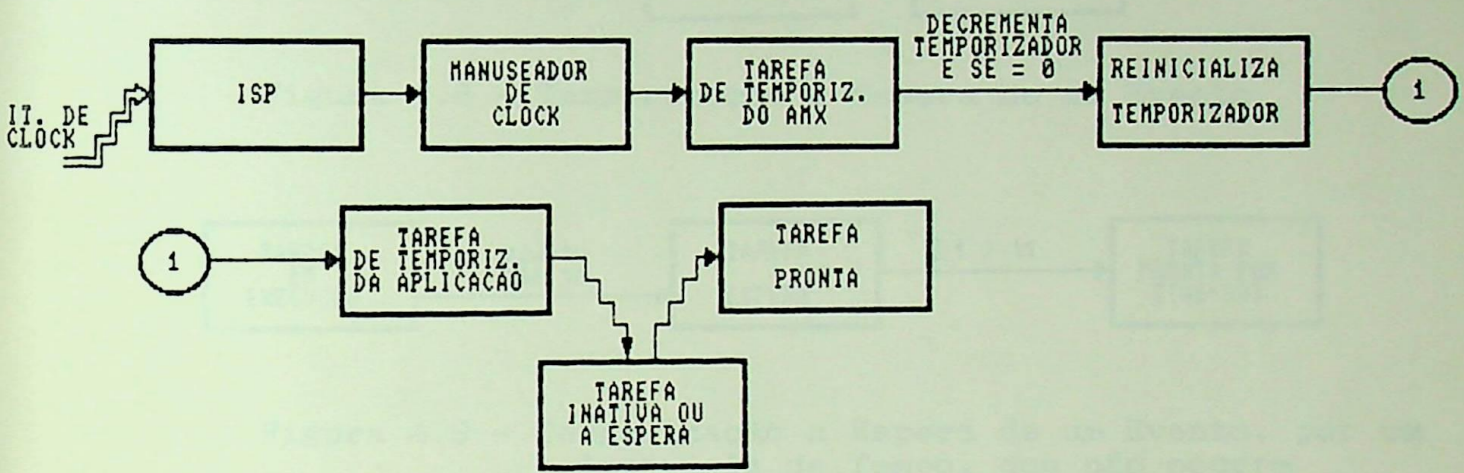


Figura 4.6 - Temporizadores Periódicos

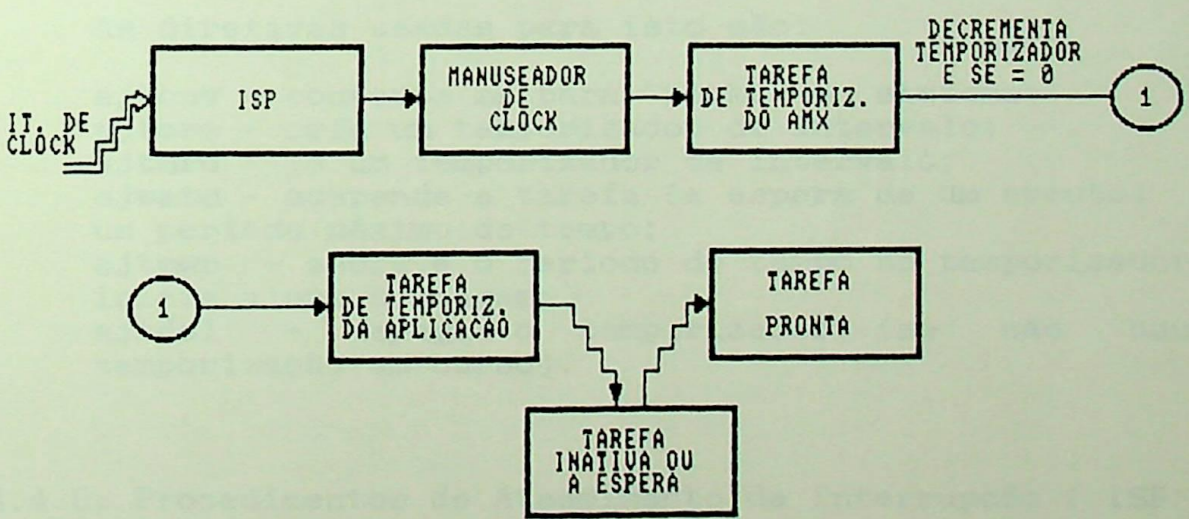


Figura 4.7 - Temporização por um Intervalo de Tempo

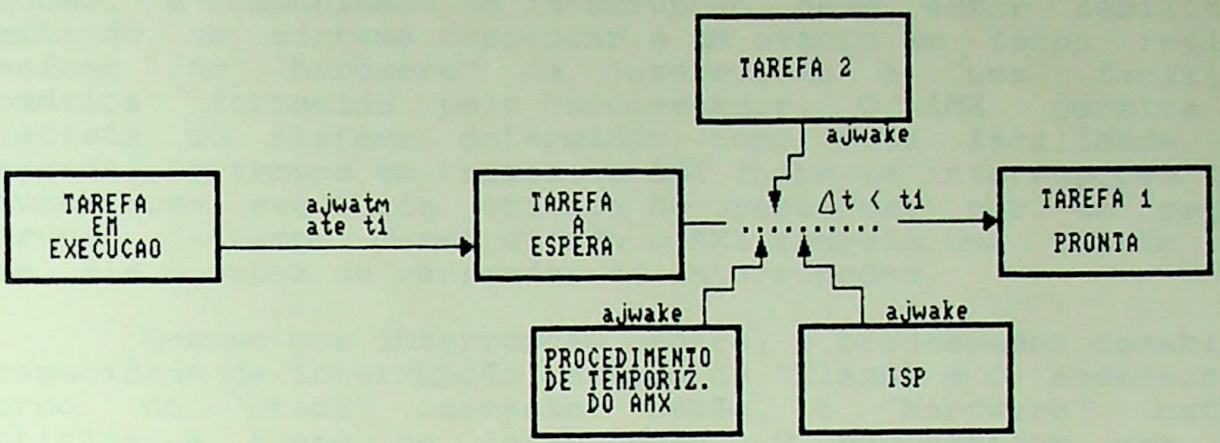


Figura 4.8 - Temporização a Espera de um Evento



Figura 4.9 - Temporização a Espera de um Evento, por um Intervalo de Tempo, que não ocorre ("time-out")

é despertada (figura 4.8). Senão, a tarefa reassume a execução com uma indicação de "time-out" (figura 4.9).

As diretivas usadas para isto são:

ajtcnv - converte ms para "ticks" do sistema;
ajtcre - cria um temporizador de intervalo;
ajtmrd - lê um temporizador de intervalo;
ajwatm - suspende a tarefa (a espera de um evento) por um período máximo de tempo;
ajtmwr - escreve o período de tempo no temporizador e inicia a sua contagem;
ajtdel - apaga o temporizador (se não houver temporização em curso).

4.4.6. Procedimentos de Atendimento de Interrupção (ISP) e o Supervisor de Interrupção do AMX

A viabilidade de sistemas multitarefas tempo real

dirigidos a eventos só é possível devido à capacidade de interrupção do processador. Assim, quando uma tarefa estiver em execução, a capacidade de interrupção deve estar habilitada, permitindo ao sistema responder a um evento em tempo real. O mecanismo de "hardware" de interrupção é uma facilidade automática fornecida pelo processador. O AMX permite ao projetista do sistema determinar como esta facilidade será empregada. De tempos em tempos, o AMX inibe as interrupções para executar uma sequência crítica de operações, por um pequeno intervalo de tempo. Mesmo quando o AMX chaveia uma tarefa para outra, ele é capaz de responder às interrupções.

Quando uma interrupção ocorre, o processador desabilita a capacidade de interrupção, estoca os "flags" e o endereço de retorno no "stack" corrente. Então, o "hardware" externo identifica a fonte de interrupção. O processador usa esta identificação para que o vetor do programa flua para um único endereço dependente do periférico. O programa localizado neste endereço é chamado de ISP.

Normalmente, um ISP salva os registros que deseja usar, atende aos periféricos, restaura os registros, habilita as interrupões e retorna à execução do programa no ponto de interrupção.

O AMX dispõe de um Supervisor de Interrupções que simplifica a construção do ISP. Assim, o ISP só controla o dispositivo e o Supervisor de Interrupções do AMX manipula o resto.

No caso de interrupção externa, a tarefa em execução é interrompida. O endereço onde ela deve recomeçar a ser executada, no fim do atendimento da interrupção é estocado no "stack" da tarefa. O endereço do ISP será procurado no endereço N*4, onde N é o tipo da interrupção ocorrida.

Quando o ISP começa a ser executado, deve encontrar uma chamada ao Supervisor de Interrupções do AMX (AAINT) que muda o "stack" corrente para o "stack" de interrupção, retornando então ao ISP. Se mais de um periférico estiver conectado na mesma fonte de interrupção externa, o ISP deve determinar qual delas provocou a interrupção. Isto pode ser feito por "polling" dos dispositivos, por "hardwares" externos, ou por outros métodos.

O ISP tendo sido executado até o fim, chama o Supervisor de Interrupções do AMX (AAINX), que ao encontrar a diretiva de retorno de interrução (IRET), checa se há alguma tarefa de maior prioridade pronta (resultado da interrupção). Se houver, a tarefa interrompida continua suspensa (Figura 4.10). Porém, pronta para a execução. O controle é dado ao Escalonador de Tarefas do AMX. Se não houver, a tarefa interrompida reassumirá o controle do processador, no ponto de interrupção, com todos seus registros restaurados do seu "stack" e a interrupção será habilitada para poder atender aos próximos eventos (Figura 4.11).

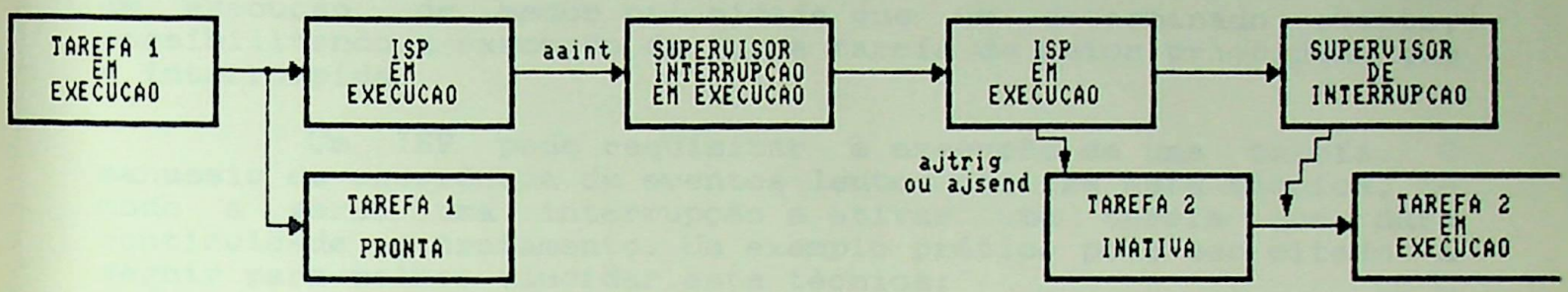


Figura 4.10 - Atendimento à Interrupção com Disparo de uma Tarefa de Maior Prioridade que a Tarefa Interrompida

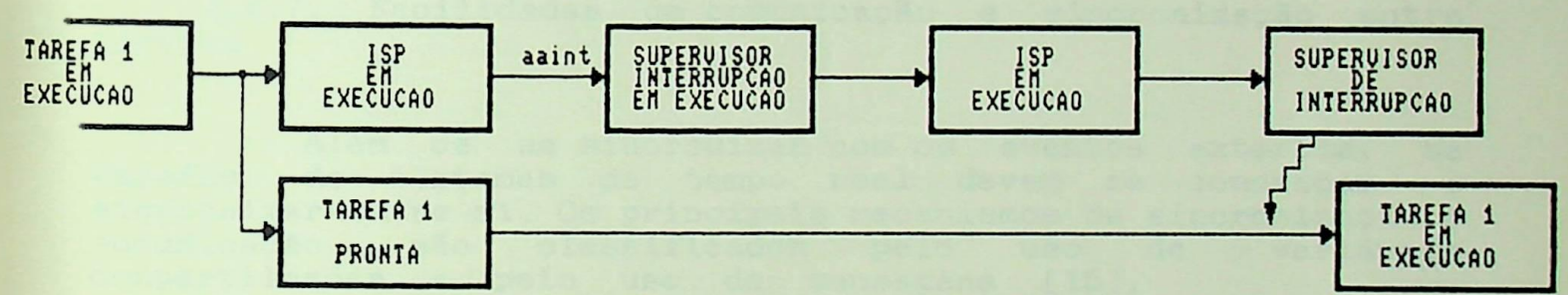


Figura 4.11 - Atendimento à Interrupção pelo ISP e pelo Supervisor de Interrupção

O Supervisor de Interrupções do AMX também permite ao ISP comunicar-se com qualquer tarefa no sistema ou lançá-las. Para isto, pode utilizar as diretivas abaixo:

ajsend - requisita a execução da tarefa enviando uma mensagem;
ajtrig - requisita a execução da tarefa;
ajwake - desperta a tarefa que espera por sua interrupção;
ajssig - sinaliza para o semáforo;
ajesig - sinaliza um evento.

Como os periféricos podem provocar uma interrupção, eles podem requisitar, indiretamente, a execução de uma tarefa através do ISP. E quando o ISP requisita uma tarefa, o AMX suspende a tarefa interrompida e o Escalonador de Tarefas do AMX decide pela tarefa pronta de maior prioridade, colocando-a em execução. É o tipo de operação que permite interromper uma tarefa em execução, de menor prioridade que um determinado evento, possibilitando a execução de outra tarefa de maior prioridade que a interrompida.

Um ISP pode requisitar a execução de uma tarefa. O manuseio de ocorrência de eventos **lentos** utiliza esta técnica, de modo a gerar uma interrupção e ativar uma tarefa que dará continuidade ao tratamento. Um exemplo prático pode ser citado a seguir para melhor elucidar esta técnica:

Um periférico envia uma interrupção indicando que um bloco de dados do periférico está pronto para ser adquirido. O ISP os recolhe numa memória e inicializa a tarefa de tratamento destes dados com um **ajtrig**. O intervalo de ocorrência de uma nova interrupção é suposto suficientemente grande para que os dados possam ser tratados.

4.4.7. Facilidades de comunicação e sincronização entre tarefas

Além de se sincronizar com os eventos externos, as tarefas de sistemas de tempo real devem se comunicar e sincronizar entre si. Os principais mecanismos de sincronização e comunicação são classificados pelo uso de variáveis compartilhadas e pelo uso de mensagens **[15]**.

Os principais mecanismos utilizados com variáveis compartilhadas são:

. semáforo **[16]**;
. região crítica **[17]**;
. monitores **[15] [18]**.

O mecanismo de comunicação e sincronização através de uso de mensagens implementado pelo AMX é na forma de primitivas de trocas de mensagens.

Uma análise resumida de cada tipo de comunicação e sincronização entre tarefas na Tabela 4.1, mostra a apropriação de cada um deles para diferentes aplicações.

Devido à análise da tabela 4.1, a implementação deste trabalho descrita no CAPITULO 5, se baseou no mecanismo de mensagens. E por este motivo, uma análise detalhada de como o AMX gerencia este mecanismo será feita a seguir.

PROBLEMA	MECANISMOS APROPRIADOS
Exclusão Mútua	Região Crítica / Monitor
Troca de sinais de sincronismo	Semáforo
Troca de dados	Mensagens
Sincronização condicional	Região Crítica / Monitor
Comunicação e Sincronização Sist. Distribuidos	Mensagens

Tabela 4.1 - Adequação dos Mecanismos de Comunicação e Sincronização [19]

4.4.7.1. Troca de Mensagens

O AMX possibilita a formação de fila de mensagens para as tarefas se comunicarem. Esta facilidade permite que as mensagens sejam empilhadas num "**mailbox**" de forma controlada [16]. Um "**mailbox**" é uma região de memória fornecida para cada tarefa, geralmente organizada sob forma de fila de espera, onde a troca de mensagens entre tarefas se efetua de modo indireto [20]. Deste modo, o AMX permite que tanto a tarefa transmissora continue sua execução como que se suspenda até que a tarefa receptora tenha sido executada como resposta à sua transmissão. A tarefa transmissora especifica a prioridade na qual a mensagem deve ser armazenada. Os níveis de prioridades (0 - mais alto, 1, 2 ou 3 - mais baixo) determinam em qual dos "mailboxes" a tarefa receptora receberá a mensagem. A profundidade de cada nível de prioridade do "mailbox" é especificada pelo usuário.

O AMX mantém **envelopes**, que são formados por blocos de 16 bytes, usados para armazenarem e transmitirem as mensagens de uma tarefa para a outra. O número máximo de envelopes permitidos pelo AMX é de 4095, também definido pelo usuário. Uma mensagem no AMX é formada por 12 bytes. Ao executar uma instrução de envio de mensagem, o AMX indica um ponteiro para o bloco de um envelope livre, move para dentro dele os parâmetros da mensagem do transmissor e adiciona o ponteiro para o "mailbox" da tarefa de destino de acordo com o nível da prioridade especificada. Se o transmissor for uma tarefa, seu identificador é acrescentado ao bloco (nos 4 bytes restantes).

Quando a tarefa receptora se inicia, os parâmetros do envelope são transferidos para o "stack" da tarefa e o envelope é automaticamente liberado para reutilização.

Se uma tarefa que recebeu uma mensagem termina, o AMX verifica se quem a enviou foi uma tarefa e se ela está esperando pelo fim da tarefa receptora. Se estiver, libera a tarefa transmissora da espera, tornando-a pronta.

As diretivas utilizadas são:

ajsend - requisita a execução da tarefa e envia-lhe uma mensagem;
ajsenw - requisita a execução da tarefa, envia-lhe uma mensagem e espera pela resposta.

4.4.7.1.1. Exemplo de Utilização de "Mailboxes" e Envelopes

Um exemplo de utilização de "mailbox" e envelopes será descrito a seguir, conforme a figura 4.12, supondo-se a seguinte situação:

- A tarefa A está sendo executada e no seu "stack" está a mensagem X (recebida da tarefa X).

- Três mensagems I, J, K estão nos envelopes do "mailbox" da tarefa A com prioridade 1 e duas mensagens, L, M estão nos envelopes do "mailbox" da tarefa A com prioridade 3, a espera de serem copiados no "stack" desta tarefa.

- Após uma interrupção, a tarefa Y (de prioridade maior) se tornou pronta e o Escalonador de Tarefas a selecionou para possuir o processador.

- A tarefa Y requisita envio da mensagem Y (prioridade 1) para a tarefa A.

- O AMX copia os parâmetros da mensagem Y no envelope e o acrescenta na fila do "mailbox" de prioridade 1.

Ao término ou bloqueio da tarefa Y, a tarefa A retorna à execução, processando a mensagem X. Ela deverá ser executada mais 3 vezes colhendo as mensagens I, J , K e Y sucessivamente no seu "stack" até que a mensagem Y seja processada. Se não tiver havido nenhuma outra requisição de envio de mensagem, as mensagens L e M seguirão a mesma rotina (figura 4.13).

Como se pode notar neste exemplo, as mensagens são recebidas e processadas pela tarefa de recepção em ordem de prioridade e entre as de mesma prioridade, em ordem de chegada.

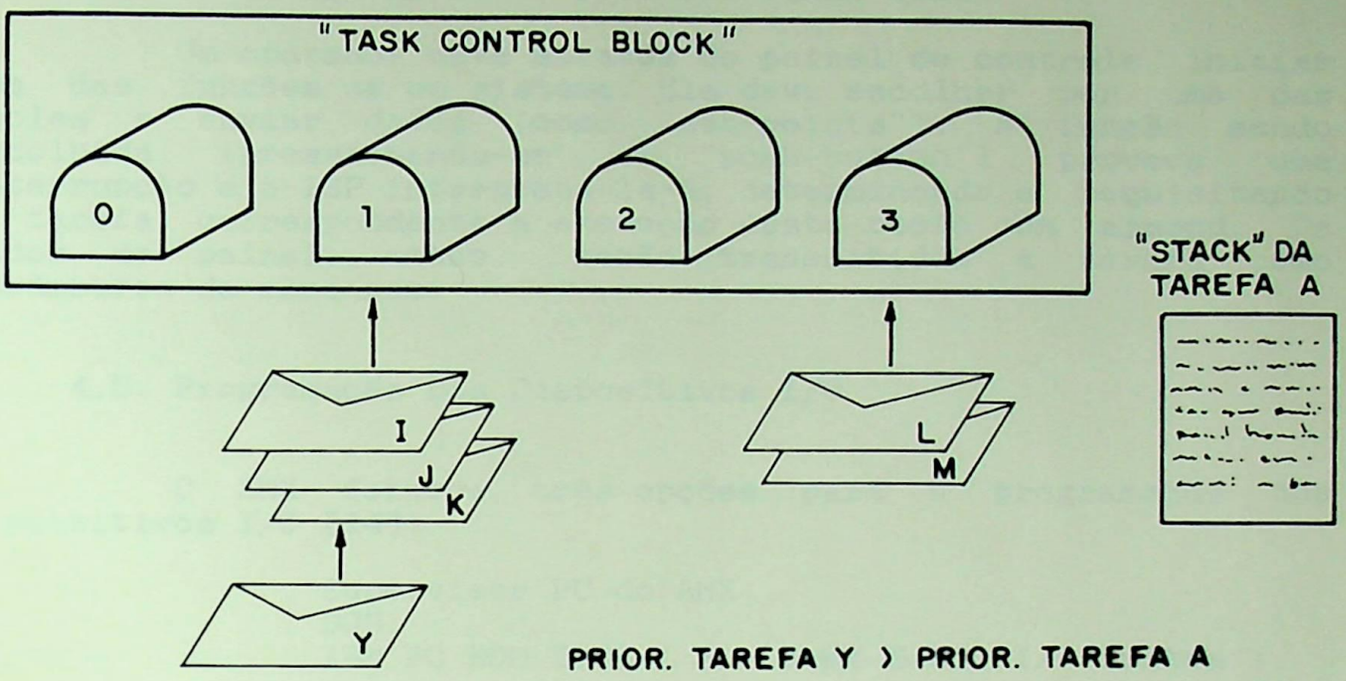


Figura 4.12 - Exemplo de "mailbox" [14]

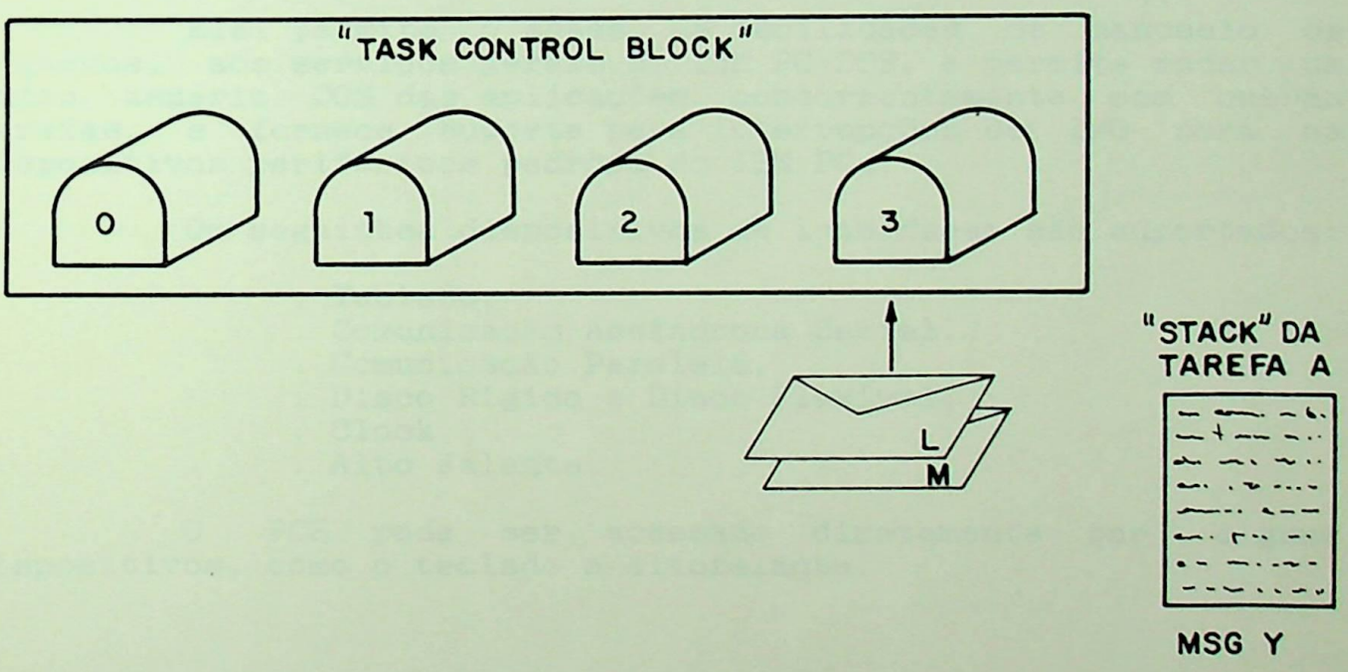


Figura 4.13 - Exemplo com envio de mensagens [14]

Um exemplo prático pode ser visto como:

Um operador deve através do painel de controle iniciar uma das funcões de um sistema. Ele deve escolher por uma das teclas e enviar dados (como "set-points"). A função sendo escolhida (pressionando-se um "push-button") provoca uma interrupção e o ISP interpreta-la-á, determinando e requisitando a tarefa correspondente a execução desta tecla com **ajsend**. Os dados do painel, então, serão transmitidos à tarefa como parâmetros da mensagem.

4.5. Programação Dos Dispositivos I/O

O AMX fornece três opções para a programação dos dispositivos I/O **[14]**:

- Supervisor PC do AMX,
- DOS,
- IBM PC ROM BIOS (Resident Basic I/O System).

4.5.1. Supervisor PC do AMX (PCS)

O Supervisor PC do AMX coloca a disposição usuários do AMX com um interface para os dispositivos de I/O nos compatíveis IBM-PC **[14]**.

Ele permite o acesso às facilidades de manuseio de arquivos, aos serviços gerais do IBM PC DOS, e permite rodar um único usuário DOS das aplicações, concorrentemente com outras tarefas, e fornece suporte para interrupções de I/O para os dispositivos periféricos padrões do IBM PC.

Os seguintes dispositivos de interfaces são suportados:

- Teclado,
- Comunicação Assíncrona Serial,
- Comunicação Paralela,
- Disco Rígido e Disco Flexível,
- Clock,
- Alto Falante.

O PCS pode ser acessado diretamente por alguns dispositivos, como o teclado e altofalante.

4.5.2. Sistema Operacional (DOS)

As chamadas às funções do DOS podem ser usadas no AMX

para acessarem:

. Teclado,
. Vídeo,
. Comunicação Assíncrona Serial,
. Comunicação Paralela,
. Disco Rígido e flexível.

Todas as operações de disco devem ser programadas por chamadas às funções do DOS. Mais de uma tarefa pode usar o DOS para arquivos I/O em disco. No entanto, cada tarefa ao utilizar o DOS, deve reservá-lo, executar as operações no disco e então liberá-lo. Esta reserva do DOS é feita por serviços do PCS.

4.5.3. IBM PC ROM BIOS

O IBM PC ROM BIOS ("Resident Basic I/O System") fornece chamadas I/O separadas (interrupção por "software") para os mesmos dispositivos mencionados acima. No entanto, muitas chamadas ao ROM BIOS são interceptadas e manuseadas diretamente pelo PCS.

4.5.4. Conclusão

O método de programação do dispositivos I/O depende diretamente da aplicação. Normalmente, deve ser criada uma tarefa separada para cada dispositivo físico I/O serial conectado à Comunicação Assíncrona Serial. Do mesmo modo, para cada impressora conectada à Comunicação Paralela, já que as tarefas sendo separadas por cada dispositivo físico, permitem operações I/O simultâneas em todos os dispositivos. As chamadas ao ROM BIOS devem, então, ser usadas por estas tarefas para operações nestes dispositivos.

CAPITULO 5 . SUPORTE DE COMUNICAÇAO

5.1. Introdução

Pelo exposto nos capítulos anteriores, para a automação completa de uma fábrica é necessária a sua integração. O que significa ter acesso a todos os dados e informações do processo. Para isto, a interligação dos diferentes equipamentos é indispensável. Neste sentido, a rede de comunicação de dados se torna o elemento principal da integração do sistema assumindo a responsabilidade de transportar os dados e informações do processo para uma sala central, onde estes poderão ser adequadamente tratados.

Deste modo, os diferentes tipos de dados de cada área da fábrica e seus diferentes usos, exigem diferentes especificações destas redes.

Deseja-se neste trabalho atuar no nível de supervisão e controle de processos, onde estas especificações podem ser definidas a partir das características listadas a seguir:

. Mensagens curtas:

De acordo com o item 2.3, as mensagens entre os níveis de supervisão e chão de fábrica são de tamanhos reduzidos, tipicamente de até 256 bytes. O que simplifica a atuação de uma rede de comunicação de dados para estes níveis, dispensando funções tais como: concatenação de mensagens e transferência de arquivos.

. Restrições temporais:

Estas mensagens necessitam de uma comunicação rápida, exigindo uma velocidade de transmissão elevada, função do programa aplicativo e dos mecanismos de comunicação de dados, que para satisfazerem estas exigências, devem responder às premissas de um sistema em tempo real. Para tanto, somente um ambiente multitarefa os satisfaz.

. "Softwares" de Supervisão e Controle de Processos para equipamentos "stand-alone":

Os "softwares" disponíveis no mercado, rodando em núcleos multitarefas em cada estação, não dispõem de mecanismos de comunicação entre as estações. Desta forma, são impedidos de trabalharem integrados à fábrica. Estes mecanismos de comunicação de dados, indispensáveis a esta integração, devem transmitir os

dados pedidos pelo programa aplicativo, sem atrasar consideravelmente sua execução. O que justifica a utilização de uma comunicação em ambientes multitarefa em tempo real.

. **Custo:**

As redes de comunicação de dados poderiam implementar estes mecanismos, mas as redes comerciais existentes, por não se restringirem a estes níveis de aplicação, são implementadas por um "hardware" e um "software" adicional que oferecem funções para diversos aplicativos (inclusive para transmissão de grandes pacotes de mensagens) e nem sempre respeitam as restrições temporais dos processos industriais. E pela sua complexidade, para aplicações no chão de fábrica, o seu custo é alto , e nestes casos, inviabiliza sua utilização.

Deste modo, um conjunto de funções mais simples pode ser especificado, de forma a atender a estas restrições.

Neste trabalho, desenvolveu-se este conjunto de funções que associado a placas de comunicação é chamado de Suporte de Comunicação baseado em Sistema Multitarefa de Tempo Real para Computadores Compatíveis com IBM-PC e Controle Distribuído de Processos Industriais, que tem entre outras, a responsabilidade de possibilitar:

. que um "software" instalado em um equipamento envie dados do programa aplicativo que estiver sendo executado, segundo seu controle de acesso ao meio de transmissão, para o "hardware" de comunicação;
. que um "hardware" responsável pela comunicação envie os dados recebidos ao meio de transmissão;
. que um "hardware", em outro equipamento, também responsável pela comunicação, receba os dados enviados pelo meio de transmissão e os envie para um "software" de comunicação;
. que este "software" receba os dados do "hardware" de comunicação e os envie para o programa aplicativo que estiver sendo executado no equipamento.

5.2. O Suporte de Comunicação Proposto

Devido as características já citadas no item 2.5, os equipamentos escolhidos sobre os quais rodam as tarefas que implementam este suporte, foram os compatíveis IBM-PC. O "hardware" escolhido para o sistema de comunicação de dados é baseado em placas comerciais, de custo relativamente baixo e que permite a conexão dos microcomputadores através de um canal serial padronizado.

5.2.1. Arquitetura do Sistema

A arquitetura do sistema sobre o qual roda o suporte de comunicação proposto é formada por até 127 estações escravas e 1 estação mestre. As estações são interconectadas por um meio físico do tipo par de fios trançados, sendo que em cada estação é colocada uma placa de comunicação que suporte o padrão RS422/RS485. A figura 5.1 ilustra a topologia adotada de acordo com as análises do Capítulo 3: topologia em Barramento. A figura 5.2 ilustra a comunicação entre o "hardware" e "software" de cada estação.

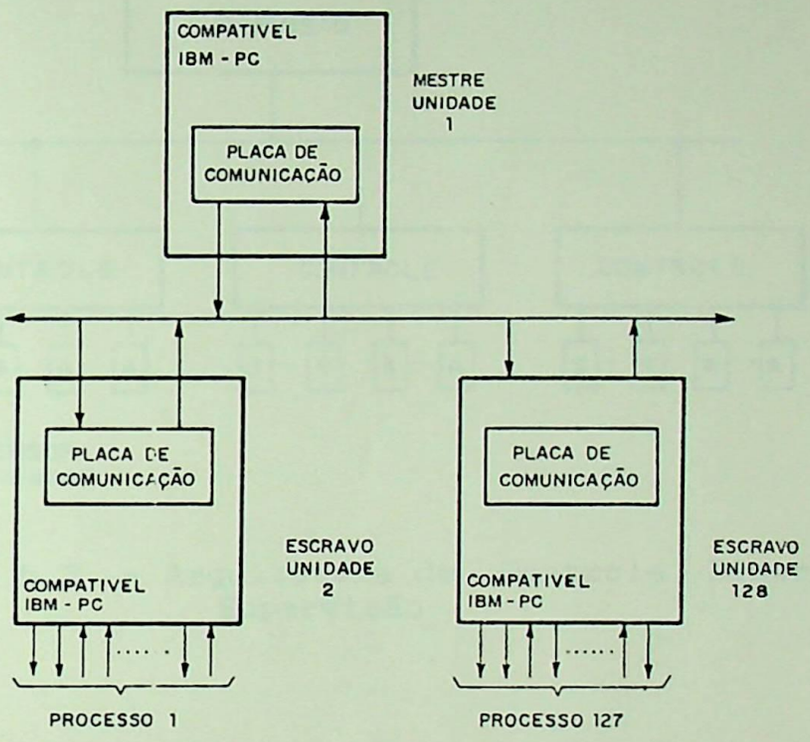


Figura 5.1 - Arquitetura do Sistema Proposto

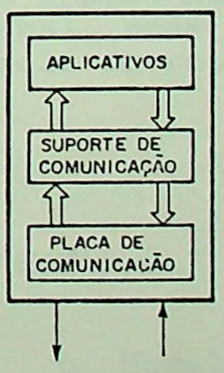


Figura 5.2 - Comunicação entre o "hardware" e o "software" em cada estação

Deve-se ressaltar que a unidade 1 poderia implementar a função de supervisão, e as outras unidades poderiam implementar as funções de controle conforme a figura 5.3. Nesta configuração, a unidade 1 poderia ser um compatível IBM-PC completo (vídeo colorido, winchester, impressora, e outros) realizando a função de supervisão, enquanto as outras unidades poderiam ser placas industriais "all-in-one", com "softwares" específicos embutidos e recursos de "hardware" mínimos para Controle de uma área ou setor de uma planta.

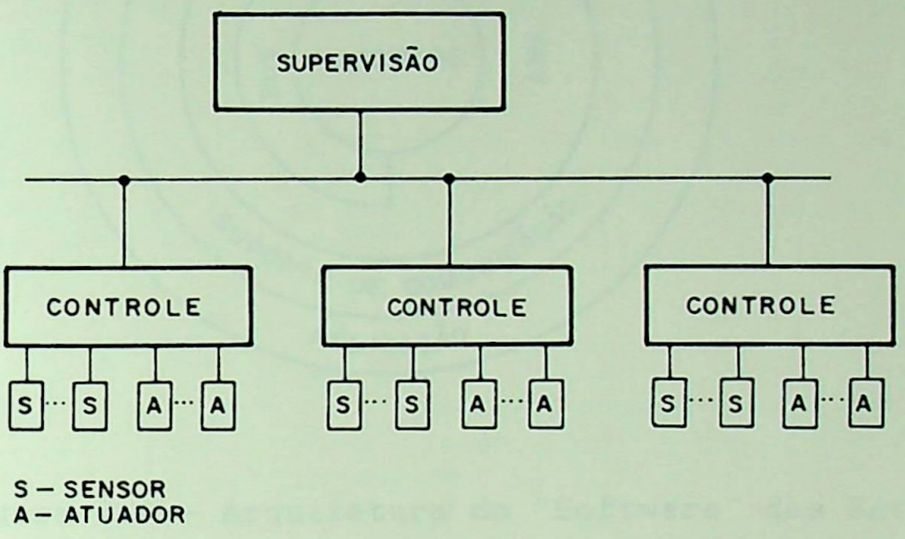


Figura 5.3 - Arquitetura de Controle Distribuído e Supervisão

5.2.2. Arquitetura do "Software" das Estações

A arquitetura do "software" das estações é composta pelo Suporte de comunicação e pelo Núcleo Multitarefa AMX86 de Tempo Real, em cima do qual roda o programa da aplicação propriamente dito (de controle ou supervisão), conforme a figura 5.4.

5.2.3. Arquitetura do "Software" do Suporte

De acordo com o Modelo OSI (Open System Interconnection) de 7 Camadas proposto pela ISO (Internnation Standardlization Organization), este Suporte implementa as Camadas 1 e 2 - Camada Física e Camada de Enlace de Dados,

respectivamente, interfaceando diretamente com a Camada da Aplicação, conforme a figura 5.5.

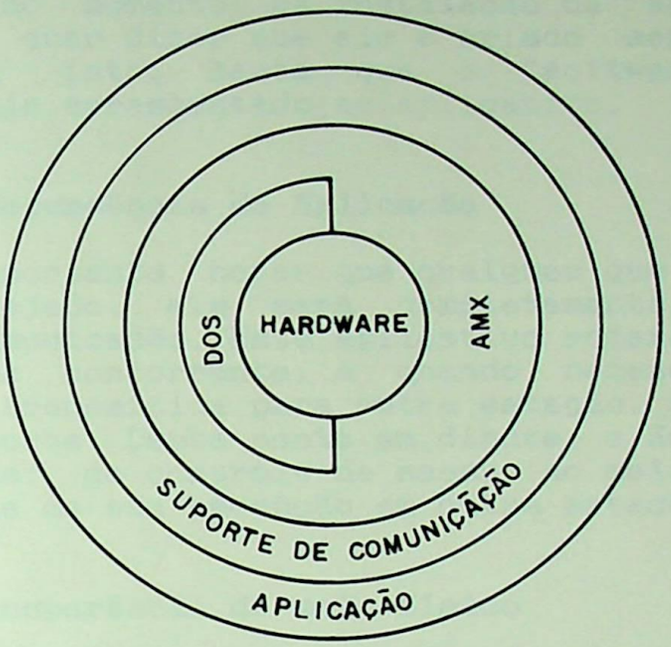


Figura 5.4 - Arquitetura do "Software" das Estações

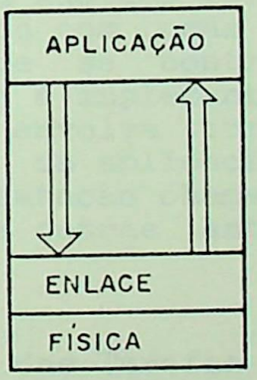


Figura 5.5 - Camadas Implementadas pelo Suporte Proposto Segundo o Modelo OSI

5.2.4. Características do Suporte de Comunicação

Algumas características importantes constituiram o Suporte de Comunicação ora proposto, tais como:

. Criação dinâmica

Este Suporte foi concebido de forma a ser criado dinamicamente no momento da instalação da aplicação em cada estação. Isto quer dizer que ele é criado sem intervenção do usuário. Para isto, basta que o "software" do Suporte desenvolvido seja acrescentado ao aplicativo.

. Independência da Aplicação

É importante notar que qualquer que seja o programa aplicativo desejado, ele será completamente independente do Suporte de Comunicação. Este aplicativo rodará sobre o Núcleo AMX86 de forma concorrente, e quando necessário, enviará a mensagem a ser transmitida para outra estação, a uma tarefa pré-definida do Suporte. Deste ponto em diante, o Suporte assumirá a responsabilidade: do controle de acesso ao meio, da transmissão desta mensagem e de sua recepção em outra estação.

. Transparência do meio físico

O meio físico é acessado através das placas de comunicação, automaticamente programadas pelo Suporte. Desta forma, sua utilização é transparente para o usuário.

. Transparência do Controle de Acesso ao Meio

Como a topologia adotada foi a de Barramento, e nela o meio físico é compartilhado com todas as estações multiplexado no tempo, há necessidade de se controlar o seu acesso para transmissão. Este controle é implementado pelo Suporte, de forma que o usuário não se envolva com esta função durante a programação ou execução da aplicação. O gerenciamento deste controle é feito por uma estação chamada Mestre, que decide pelo melhor momento em que as outras estações, chamadas Escravas, devem transmitir.

. Identificação das Tarefas Comunicantes

A informação a ser transmitida deve ser acompanhada de quem a enviou e para onde deve ir. A identificação automática das tarefas do programa aplicativo que desejam se comunicar, é uma das funções do Suporte.

. Formatação da mensagem

O dado a ser transmitido é acrescido de várias outras informações necessárias para sua transmissão adequada. Cada uma destas informações é relacionada ao seu tipo, segundo sua posição

relativa num determinado formato pré-definido. O Suporte realiza esta formatação.

. Integridade dos dados

Mais importante do que a transmissão de um dado, é a certeza da integridade dos dados recebidos. Para isto, o Suporte assume a coordenação automática de um mecanismo de verificação dos dados. O mecanismo escolhido foi baseado no "checksum" ou verificação a partir da soma dos dados.

5.2.5. Interfaceamento entre Aplicação e Suporte

Quando uma tarefa da aplicação deseja enviar uma mensagem para uma tarefa localizada em outra estação, ela deve ser transmitida para o suporte numa estrutura que contenha as seguintes informações conforme a figura 5.6, utilizando a instrução:

ajsend (gcb.idtskintr,0,pp_msg_aplic_int);

onde:

pp_msg_aplic_int será o ponteiro de ponteiro desta estrutura.

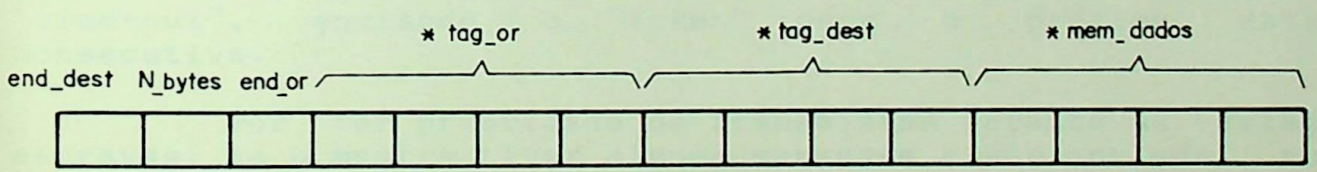


Figura 5.6 - Estrutura da Mensagem de Transmissão da Interface entre Aplicação e Suporte

E da mesma forma, quando uma tarefa da aplicação espera receber uma mensagem de uma tarefa localizada em outra estação, ela deve ser iniciada com um ponteiro para uma estrutura, a fim de recebê-la do suporte. A mensagem estará contida numa estrutura conforme a figura 5.7.

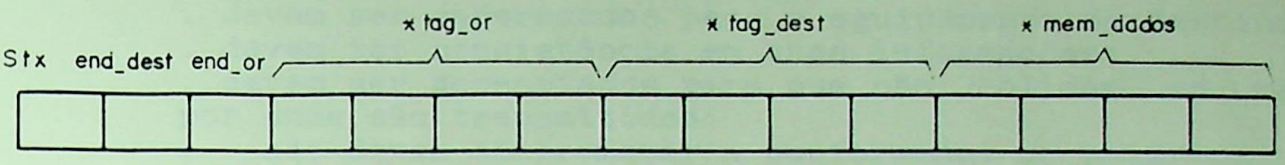


Figura 5.7 - Estrutura da Mensagem de Recepção da Interface entre Aplicação e Suporte

5.2.6. Procedimentos de Comunicação Entre Estações

Uma estação escrava só pode iniciar a transmissão de dados mediante o recebimento de uma mensagem especial padronizada, chamada de "token", enviada pela estação mestre. A partir de seu recebimento, a estação escrava executa algumas tarefas que formatam os dados e os envia ao destino. No final desta transmissão o "token" é reenviado para a estação mestre. Ao receber o reenvio do "token", a estação mestre o envia para a próxima estação escrava, que seguirá o mesmo procedimento já descrito para este tipo de estação.

Após a transmissão do "token" para uma estação escrava, o mestre espera sua resposta por um tempo pré-determinado. Para que, em caso de falha na transmissão do "token" ou em caso de falha da estação escrava, o sistema não fique eternamente a espera da resposta desta estação, o mestre saí da espera por "time-out", enviando o "token" para a próxima estação consecutiva.

Por ter prioridade de transmissão perante as estações escravas, se o mestre tiver alguma mensagem a ser enviada, antes de enviar o "token" para cada estação, ele se dá o direito de enviá-la primeiro.

Como a mensagem enviada para a supervisão é para atualização das variáveis, e esta atualização ocorre a cada 1 segundo no máximo, caso haja erro na sua transmissão, o sistema a despreza, e espera o próximo ciclo de atualização.

5.2.7. Estrutura da Mensagem Entre Estações

Algumas características das mensagens entre estações podem ser levantadas analisando-se que, tanto no modo supervisão, programação, como no modo de controle executado pelo equipamento

de supervisão, os dados:

. devem ser endereçados para o equipamento de destino;
. devem ter consistência em suas informações;
. devem ser gerenciados para que não "colidam" no meio por onde são transmitidos;
. não devem comprometer a performance do programa de controle ou supervisão que está sendo executado em cada equipamento, ao serem transmitidos e recebidos.

Assim, a mensagem de dados ou o "token" enviados de uma estação para a outra são acrescidos de informações que formam a estrutura do protocolo de comunicação do Suporte desenvolvido (figura 5.8). Esta estrutura da mensagem de dados é formada por:

. **STX**: inicia o protocolo - 1 byte ;
. **end_dest**: endereço da estação de destino - 1 byte ;
. **N_bytes**: número de bytes do protocolo menos 1. É igual ao N_bytes_dados (número de bytes de dados - da mensagem de dados, propriamente dita, a transmitir) mais 16 bytes (de cabeçalho do protocolo). (N_bytes_dados + 16) bytes;
. **end_or**: endereço da estação de origem - (1 byte);
. **tag_or**: 5 bytes de caracteres do "tag" da tarefa de origem da transmissão (4 caracteres alfanuméricos mais 1 caracter "NULL");
. **tag_dest**: 5 bytes de caracteres do "tag" da tarefa de destino da transmissão (4 caracteres alfanuméricos mais 1 caracter "NULL");
. **dados**: número de caracteres da mensagem de dados, propriamente dita, terminada pelo caracter "NULL" (\0) - N_bytes_dados bytes;
. **chksum**: 1 inteiro resultado do "checksum" do protocolo (exceto o STX) - 2 bytes;
. **ETX**: finaliza o protocolo - 1 byte.

5.2.8. Estrutura do "Token" Entre Estações

A estrutura para transmissão do "token" é formada por (figura 5.9):

. **STX**: inicia o protocolo - 1 byte ;
. **end_dest**: endereço da estação de destino acrescido de 128 (identificador de "token") - 1 byte;
. **chksum**: 1 inteiro resultado do "checksum" do protocolo (exceto o STX) - 2 bytes;
. **ETX**: finaliza o protocolo - 1 byte.

É importante esclarecer que para identificar o início

da recepção, é necessário que se detecte o caracter **STX**. Desta forma, foi desnecessário incluí-lo no chksum.

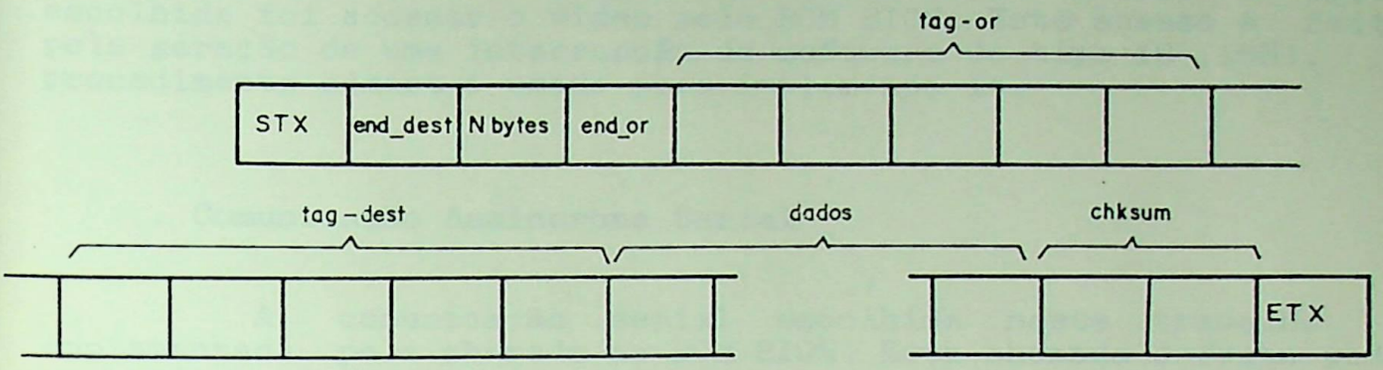


Figura 5.8 - Estrutura para Transmissão da Mensagem de Dados Entre Estações

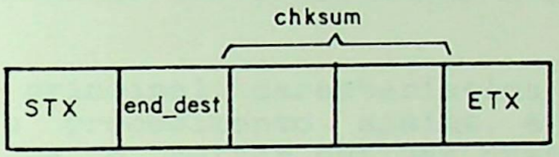


Figura 5.9 - Estrutura para Transmissão do "Token" Entre Estações

5.2.9. Programação dos Dispositivos I/O

Como mencionado no item 4.5. , o método de programação I/O depende diretamente da aplicação. Os dispositivos I/O utilizados neste trabalho foram o teclado, vídeo e interface de comunicação assíncrona serial. E cada um deles foi programado como a seguir.

. Teclado

O Supervisor PC do AMX (PCS) fornece um manuseador para teclado que obtém o carater: o procedimento **ajkbgc**. Este procedimento gera 2 caracteres de 8 bits, quando uma tecla ou combinação de teclas pressionada(s) for(em) válida(s).

O primeiro código identifica a tecla específica ou a combinação de teclas pressionadas. O segundo código é equivalente ao código ASCII da(s) tecla(s) pressionada(s).

. **Vídeo**

O PCS não interfere nas chamadas de vídeo. A opção escolhida foi acessar o vídeo pelo ROM BIOS. Este acesso é feito pela geração de uma interrupção de software do tipo 16 (10H). O procedimento **ajsint** é usado para implementá-lo.

. **Comunicação Assíncrona Serial**

A comunicação serial escolhida neste trabalho é implementada pela chamada ao ROM BIOS. Esta chamada é feita pela geração de uma interrupção de software do tipo 20(14H). Uma inicialização deve ser feita separadamente para programação da linha de comunicação, dos caracteres a serem transmitidos e recebidos e para detectar o "status" do dispositivo. O PCS intercepta todas as interrupções do tipo 20(14H) e as manuseia diretamente.

A principal característica na transmissão de um caracter pelo procedimento **ajsint**, é o de bloquear a tarefa transmissora se o buffer estiver cheio, e de a desbloquear, reassumindo a execução quando o buffer estiver vazio.

E sua principal característica na recepção, é o de bloquear a tarefa de recepção se o buffer estiver vazio, e de a desbloquear, reassumindo a execução quando um caracter chegar no buffer.

5.2.10. A Transmissão de Mensagens

Uma análise mais detalhada da aplicação das diretivas utilizadas para transmitirem as mensagens será feita a seguir.

```
unsigned int id_tarefa;
int prioridade;
int sts;
  .
  .
  .
sts = ajsend ( id_tarefa, prioridade, end_msg);

     ou

sts = ajsenw ( id_tarefa, prioridade, end_msg);
```

onde:

id_tarefa => identificador da tarefa receptora;

prioridade => prioridade da mensagem;
end_msg => endereço onde se encontra armazenada a mensagem;
sts => "status" da chamada.

Um detalhe muito importante nestas diretivas é que a mensagem recebida pela tarefa receptora será a cópia da mensagem armazenada em end_msg, e não o endereço da mensagem. Como a mensagem pode ser no máximo de 12 bytes, se houver necessidade de um tamanho maior, um método que pode ser utilizado é o de se passar como mensagem um ponteiro de um ponteiro para a primeira posição da mensagem.

Um exemplo será dado a seguir, para esclarecer melhor este método, supondo-se uma mensagem do tipo (figura 5.10):

```
struct { int A;
         int B;
         char C;
         int *D;
         char *E;
       }msg;
struct msg *p;
```

Neste caso, a mensagem ocuparia 13 bytes (A => 2, B => 2, C => 1, *D => 4, *E => 4). E deste modo, enviar **p** como endereço da mensagem seria impossível. Uma solução seria enviar um ponteiro para o ponteiro **p** (Figura 5.11):

```
struct msg **pp;
sts = ajsend (id_tarefa_receptora,prioridade,pp);
```

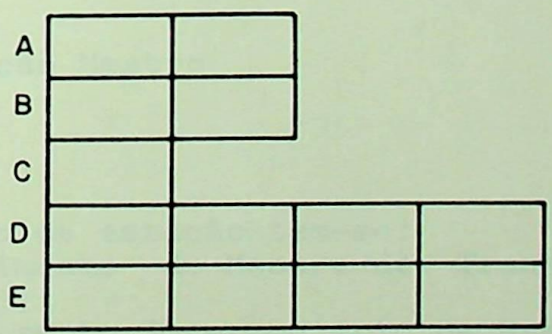


Figura 5.10 - Mensagem armazenada em bytes

A tarefa receptora receberia então o conteúdo apontado por **pp**, que seria o próprio **p**:

```
tarefa receptora (p)
struct msg *p;
{
      .
      .
      .
}
```

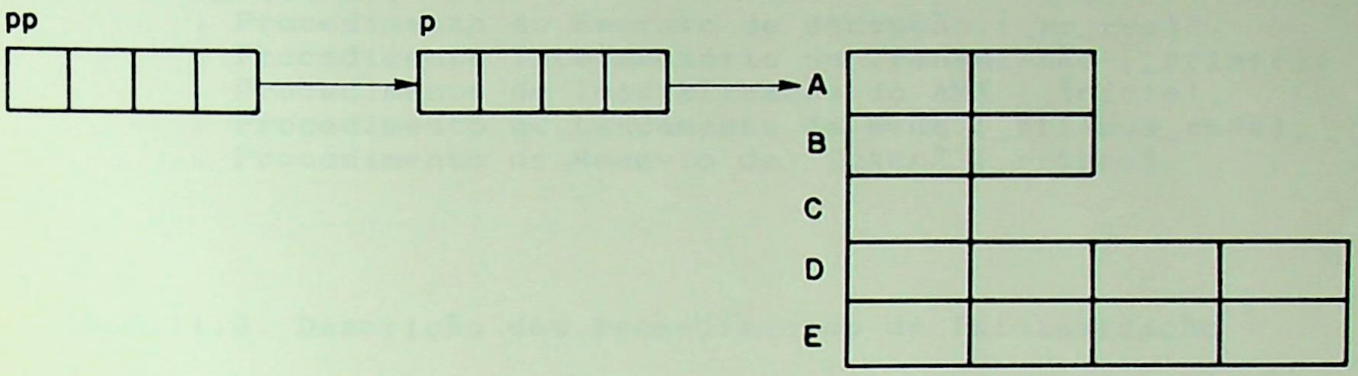


Figura 5.11 - Mensagem endereçada por ponteiro de ponteiro

Esta solução foi constantemente adotada na comunicação entre tarefas na implementação deste Suporte.

5.2.11. Procedimentos de Inicialização

Os Procedimentos de Inicialização se diferenciam conforme o tipo de estação: mestre ou escrava.

5.2.11.1. Estação Mestre

Neste tipo de estação tem-se:
. Procedimento do Mestre de Transmissão RS 422/485 (_rrmtrRS);
. Procedimento Intermediário de Recepção (_rrinrec);
. Procedimento de Controle de Acesso ao Meio (_rrctrl);
. Procedimento do Mestre de Recepção (_rrmestre_rec);
. Procedimento Intermediário de Recepção (_rrmestre_rec);
. Procedimento Intermediário de Transmissão (_rrintr);
. Procedimento de Inicialização do AMX (_inicio);
. Procedimento de Lançamento da Rede (_rrlanca_rede).

5.2.11.2. Estação Escrava

Neste tipo de estação tem-se:
. Procedimento Intermediário de Recepção(_rrinrec);
. Procedimento do Escravo de Transmissão RS 422/485 (_rrtrRS);
. Procedimento do Escravo de Recepção (_rr_rec);
. Procedimento Intermediário de Transmissão (_rrintr);
. Procedimento de Inicialização do AMX (_início);
. Procedimento de Lançamento da Rede (_rrlanca_rede);
. Procedimento de Reenvio do "Token" (_rrtken).

5.2.11.3. Descrição dos Procedimentos de Inicialização

Todos os Procedimentos de Inicialização, com exceção do Procedimento de Inicialização Início, utilizam-se da diretiva **ajntsk** para criar dinamicamente a respectiva tarefa, como a seguir:

```
void rr...()

{ int sts;
  sts=ajntsk(&tdt...,&gcb.idtsk...);
      .
      .
      .
}
```

onde:

tdt... é um ponteiro para uma estrutura compatível com a Tabela de Definição de Tarefa para a tarefa ..., criada dinamicamente;

gcb.idtsk... é um ponteiro para o identificador da tarefa ..., criada dinamicamente, definido no arquivo de definições **tesedefe.h** ou **tesedefm.h**.

5.2.11.3.1. Procedimento de Inicialização Início

O Procedimento de Inicialização Inicio localiza as

tarefas da aplicação e as associa aos seus identificadores, através da diretiva **ajutsk**, como a seguir:

```
unsigned int tecl_id,impr_tr_id,impr_rec_id,rec_id;
void inicio()

{ unsigned int sts;
  sts= ajutsk(&tecl_id,"tecl");
          .
          .
          .
  sts= ajutsk(&impr_tr_id,"imtr");
          .
          .
  sts= ajutsk(&impr_rec_id,"imre");
          .
          .
  sts= ajutsk(&rec_id,"recs");
          .
          .
}
```

onde:

sts	é o status da diretiva ajutsk;
tecl_id	é um ponteiro para o identificador da tarefa Teclado da Aplicação;
impr_tr_id	é um ponteiro para o identificador da tarefa Impressão de Transmissão da Aplicação;
impr_rec_id	é um ponteiro para o identificador da tarefa Impressão de Recepção da Aplicação;
rec_id	é um ponteiro para o identificador da tarefa de Recepção de Teclado da Aplicação;
tecl,imtr, imre,recs	são os "tags" das respectivas tarefas.

5.2.11.3.2. Procedimento de Lançamento de Rede

Além da criação dinâmica da respectiva tarefa, como descrita anteriormente, este procedimento dispara a Tarefa de Lançamento de Rede pela diretiva **ajtrig**, como a seguir:

```
void rrlanca()

{ int sts;
```

```
    sts=ajntsk(&tdtlanc,&gcb.idtsklanca);
            .
            .
    sts= ajtrig(gcb.idtsklanca);
            .
            .
}
```

onde:

tdtlanc — é um ponteiro para uma estrutura compatível com a Tabela de Definição de Tarefa para a tarefa Lançamento de Rede, criada dinamicamente;

gcb.idtsklanca — é um ponteiro para o identificador da Tarefa Lançamento de Rede criada dinamicamente, definido no arquivo de definições **tesedefe.h** ou **tesedefm.h**.

5.2.12. Tarefas Concorrentes

O Suporte de Comunicação instalado em cada equipamento é composto, entre outros, por um núcleo multitarefas em tempo real, sobre o qual rodam as diversas tarefas, listadas abaixo:

a - Tarefa "Main";
b - Tarefa de Lançamento da Rede;
c - Programa da Aplicação (dividido em várias tarefas de acordo com seu objetivo):

Uma aplicação com a finalidade de testar a comunicação de dados, foi desenvolvida em paralelo ao desenvolvimento do Suporte de comunicação, como a seguir:

c.1 - Tarefa Teclado;
c.2 - Tarefa Impr_tr;
c.3 - Tarefa Rectecla;
c.4 - Tarefa Impr_rec;

d - Tarefa Intermediária da Transmissão;
e - Tarefa de Transmissão RS 422/RS 485;
f - Tarefa de Recepção RS 422/RS 485;
g - Tarefa Intermediária de Recepção;

A estação mestre, além das tarefas listadas acima possui:

h - Tarefa de Controle de Acesso.

A estação escrava, além das tarefas listadas acima (com exceção do item h) possui:

i - Tarefa de Reenvio do "Token".

A seguir será feito o detalhamento das tarefas a partir de uma sequência de transmissão/recepção da mensagem entre a estação de origem e a de destino (Figuras 5.12 e 5.13).

5.2.12.1. Estação Transmissora (origem)

5.2.12.1.1. Tarefa "Main"

Todo programa escrito na Linguagem C, se inicia pela função **main()**. Neste trabalho, uma tarefa foi alocada para ela, e se inseriu a diretiva **ajentr** para forçar a entrada no ambiente multitarefa do AMX, como a seguir:

```
main()

{ char * result;
  int sts;
  struct amxupts *upt;
  ajupt(&upt);
  sts= ajentr(1,upt,&result);
          .
          .
          .
}
```

onde:

sts	é o status de retorno da diretiva **ajentr**;
result	é um ponteiro para 4 bytes da informação retornada da chamada **ajexit**;
upt	é um ponteiro para a Tabela de Parâmetros do Usuário;
1	permite a saída do AMX pela diretiva **ajexit**.

5.2.12.1.2. Tarefa de Lançamento da Rede

Ativada por seu procedimento de inicialização, inicializa a placa de Comunicação Assíncrona Serial, pela diretiva **ajsint**. Dispara a tarefa de recepção RS 422/485 e a tarefa Teclado da Aplicação, com a diretiva **ajtrig**. A partir

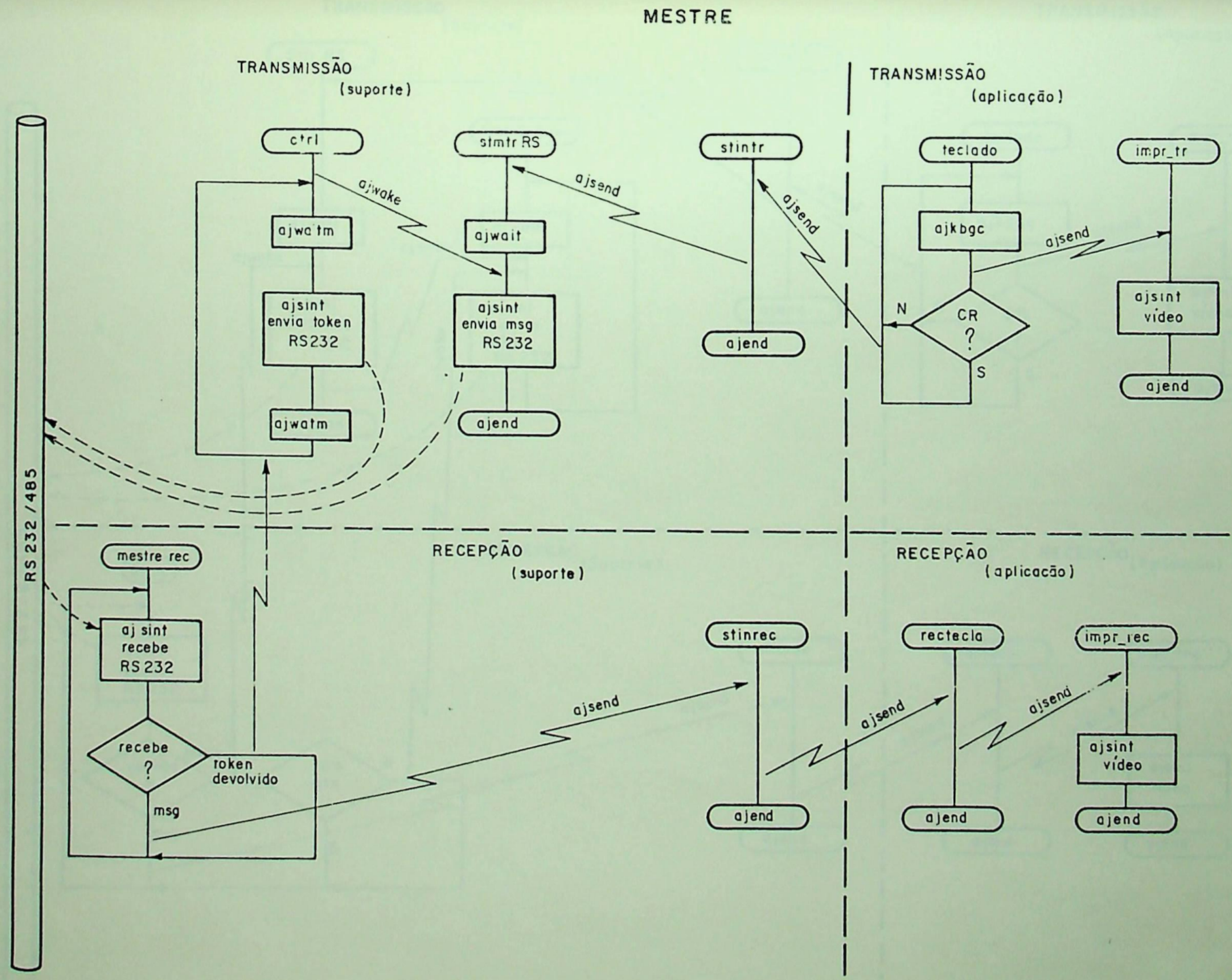


Figura 5.12 - Sincronização/Comunicação no Mestre

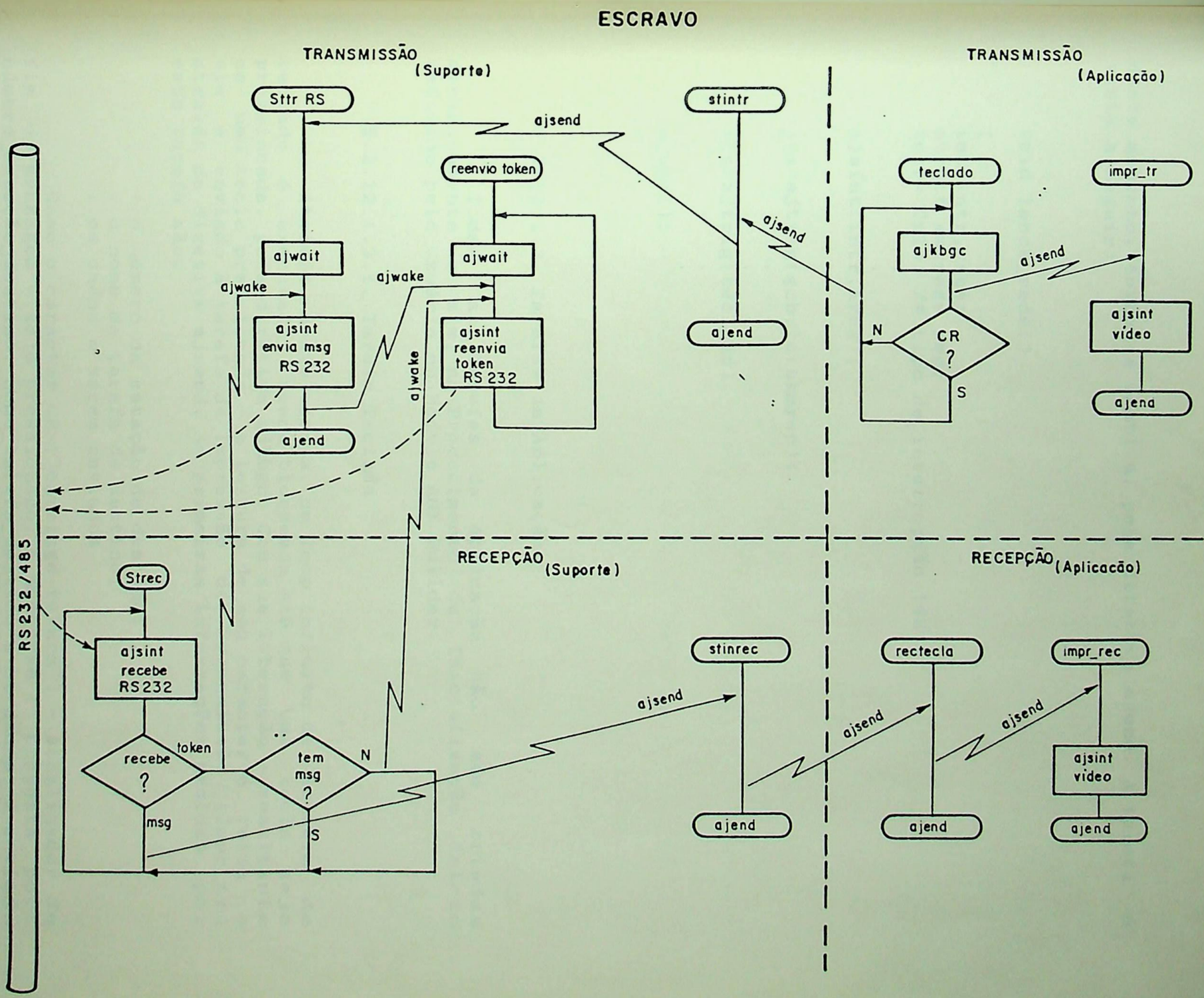


Figura 5.13 - Sincronização/Comunicação no Escravo

deste momento, torna-se inativa, pela diretiva **ajend**. A tarefa é vista a seguir:

```
void lanca_rede()

{
int intr, sts;
struct amxregs reg;
intr= 0x14; /* tipo de interrupção 14H */
          .
          .
ajsint(intr,&reg);
          .
          .
sts=ajtrig(gcb.idtskmrec);
          .
          .
sts=ajtrig(tecl_id);
          .
          .
ajend();
}
```

5.2.12.1.3. Tarefas da Aplicação

Todas as Tarefas da Aplicação não são criadas dinamicamente e sim, no Procedimento de Inicialização Início definido pelo usuário no Módulo AMX Builder.

5.2.12.1.3.1. Tarefa Teclado

Através desta tarefa, um loop infinito de leitura de teclado é executado, sendo bloqueado até que uma tecla seja pressionada, pela diretiva **ajkbgc**. Com sua liberação - resultante de uma tecla pressionada - a leitura de seu caracter é feita e ele é enviado à tarefa de Impressão de Transmissão (Impr_tr) através da diretiva **ajsend**. As primeiras informações pedidas por esta tarefa são:

- o número da estação de destino;
- o nome da tarefa de destino;
- os dados a serem enviados.

Caso o caracter CR ("carriage return") - sinalizador de fim de mensagem - seja pressionado, a mensagem (proposta por: número_estação_destino, nome_tarefa_destino e os dados) é enviada à tarefa Intermediária de Transmissão, também pela diretiva

ajsend, como a seguir:

```
void teclado()

{int tecla,sts;
 struct msg msg_int,*p_msg_int,**pp_msg_int;
for(;;)
     {
for(;tecla<>CR;)
     {
          .
     .
          tecla=ajkbgc();
     .
     .
     sts=ajsend(impr_tr,0,&tecla);
     }
     /*  estoca-se  cada  tecla  na  estrutura  da  mensagem
                          msg_int */
               .
     sts=ajsend(gcb.idtskintr,0,pp_msg_int);
               .
     }
}
```

5.2.12.1.3.2. Tarefa Impr_tr

Através desta tarefa - disparada toda vez que recebe um caracter vindo da tarefa Teclado - a impressão do caracter na tela é implementada pela interrupção de vídeo (10H), pela diretiva **ajsint**. O que provoca a execução da diretiva **ajend**, que a finaliza, tornando-a inativa.

```
void impr_tr(ch)
unsigned char ch;

{    .
     .
 sts=ajsint(intr,&reg);
     .
ajend();
}
```

5.2.12.1.4. Tarefa Intermediária da Transmissão

Através desta tarefa - disparada pela recepção da

mensagem vinda da tarefa Teclado - uma adequada alocação dos bytes é feita. Posteriormente, a mensagem realocada é enviada para a tarefa de transmissão RS 422/485, pela diretiva **ajsend.** A tarefa Intermediária torna-se inativa, pela diretiva **ajend**, como a seguir:

```
void stintr(p)
struct msg *p;

{
 int sts;
      .
      .
 sts=ajsend(gcb.idtskmtrRS,0,pp_msg_intr_tr);
      .
      .
 ajend();
}
```

onde:

0	é a prioridade do "mailbox" para o qual é enviada a mensagem;
pp_msg_intr_tr	é o ponteiro do ponteiro da mensagem a ser enviada.

5.2.12.1.5. Tarefa de transmissão RS 422/RS 485

Disparada pela recepção da mensagem vinda da tarefa intermediária de transmissão. É bloqueada pela diretiva **ajwait**, a espera da permissão de acesso ao meio ("token") enviado pela tarefa de controle de acesso pela diretiva **ajwake**. Este envio só ocorre no início de cada fatia de tempo destinada a cada estação. Após o recebimento do "token", o canal serial RS 422/485 é utilizado através da interrupção (14H) da transmissão serial, para o envio da mensagem para a estação destino, pela diretiva **ajsint**, como visto a seguir:

```
void sttrRS(p_msg_tr)
struct msg_intr *p_msg_intr;

{struct amxregs rege;
 int sts;
      .
      .
 ajwait();
      .
```

```
        .
        .
 sts=ajsint(0x14,&rege);
        .
        .
 sts=ajsint(0x14,&rege);
        .
 ajend();
}
```

5.2.12.2. Estação Receptora (destino)

5.2.12.2.1. Tarefa "Main"

Esta tarefa é a mesma descrita no item 5.2.12.1.1.

5.2.12.2.2. Tarefa de Lançamento da Rede

Ativada por seu procedimento de inicialização, inicializa a placa de Comunicação Assíncrona Serial, pela diretiva **ajsint**. Dispara a tarefa de recepção RS 422/485 e a tarefa Teclado da Aplicação, com a diretiva **ajtrig**. A partir deste momento, torna-se inativa, pela diretiva **ajend**. A tarefa é vista a seguir:

```
void lanca_rede()

{
int intr, numero_estacao,sts;
struct amxregs reg;
intr= 0x14; /* tipo de interrupção 14H */
          .
          .
ajsint(intr,&reg);
          .
          .
sts=ajtrig(gcb.idtskmrec);
          .
          .
sts=ajtrig(tecl_id);
          .
          .
ajend();
}
```

5.2.12.2.3. Tarefa de Recepção RS 422/RS 485

Esta tarefa é ativada pela tarefa de Lançamento de Rede. É um "loop" infinito que fica bloqueado pela diretiva **ajsint**, a espera da interrupção de recepção RS 422/485, que ocorre quando uma palavra chega por este canal. Pode receber o "token" ou uma mensagem. Desta forma, executa diversos testes. Estes testes podem ser:

a) Se o primeiro byte é a palavra **STX** (início de uma mensagem).

b) Se a informação recebida é mensagem. Neste caso, verifica se o segundo byte é seu endereço (endereço_destino). Se for, aloca os outros carateres de acordo com o terceiro byte (número de bytes da mensagem), e calcula o "checksum". Verifica se os bytes recebidos correspondentes ao "chksum" conferem com o "checksum" calculado. Se forem, envia a mensagem à tarefa Intermediária de Recepção, pela diretiva **ajsend**,

c) Se a informação recebida é o "token" (permissão enviada pelo mestre para que transmita se houver mensagem a espera). Caso o seja, a tarefa verifica se o segundo byte é seu endereço. Se for, faz o "checksum". Verifica, em seguida, se a tarefa de Transmissão RS 422/485 está a espera do "token"(se há mensagem a ser transmitida), pela diretiva **ajwake**. Testa se há mensagem pela verificação do "status" desta diretiva. Se não houver executa a diretiva **ajwapt**, para conservar a tarefa de Transmissão RS 422/485 bloqueada, e executa a diretiva **ajwake** para disparar a tarefa Reenvio do "Token" para a estação mestre. Logo após, retorna ao início do "loop" bloqueada a espera de recepção de mensagem pela placa de Comunicação, como a seguir:

```
void strec()

{
int intr,sts;
struct amxregs reg;
intr= 0x14; /* tipo de interrupção 14H */

for(;;)

     {
          .
          .
/* recebimento de caracter de STX até ETX */
     ajsint(intr,&reg);
          .
          .
     ajsint(intr,&reg);
               .
               .
     ajsint(intr,&reg);
               .
               .
```

```
        /* recepção de ETX, fim de recepção da mensagem */
        /* se for mensagem */
        /* faz chksum */
        sts=ajsend(gcb.idtskinrec,0,pp_msg_RSrec);
             .
        /* se for token */
        /* faz chksum */
        sts=ajwake(gcb.idtsktrRS);
             .
        /* testa se tem mensagem a espera */
        /* se não tiver */
        sts=ajwapt(gcb.idtsktrRS);
        sts=ajwake(gcb.idtsktken);
             .
        /* verifica se tarefa do token está pronta */
        /* se não anula ajwake */
        sts=ajwapt(gcb.idtsktken);
             .
        }

}
```

5.2.12.2.4. Tarefa Intermediária de Recepção

Ativada pelo recebimento da mensagem enviada pela tarefa de Recepção RS 422/485, retira todos os bytes anexados na tarefa Intermediária de Transmissão - para a transmissão e verificação de transmissão, descobre de qual tarefa é o "tag" da tarefa destino da aplicação, pela diretiva **ajutsk**, e a envia os bytes restantes, pela diretiva **ajsend**. No caso da aplicação desenvolvida para demonstrar este Suporte, a tarefa será Rectecla. Torna-se inativa em seguida.

```
void stinrec(p_msg_RSrec)
struct msg_RS *p_msg_RSrec;

{ int sts;
  unsigned int taskid;
          .
          .
  sts= ajutsk(&taskid,(*p_msg_RSrec).tag_dest);
          .
          .
  sts= ajsend(taskid,0,pp_msg_RSrec);
          .
          .
  ajend();
}
```

5.2.12.2.5. Tarefas da Aplicação

5.2.12.2.5.1. Tarefa Rectecla

Ativada pelo recebimento da mensagem enviada pela tarefa Intermediária de Recepção, é responsável pelo envio dos caracteres à tarefa da Aplicação Impr_rec - Impressão da Recepção, que é feita pela diretiva **ajsend**.Tornando-se inativa posteriormente pela diretiva **ajend**, como a seguir:

```
void rectecla(p_msg)
struct msg_inrec *p_msg;

{
unsigned char *p_char;
/*    p_char    aponta    para    cada    caracter    recebido
consecutivamente até o último (CR) */
for(;p_char<> CR;)
     {
     .
     sts=ajsend(impr_rec,0,p_char);
          .
ajend();
     }
}
```

5.2.12.2.5.2. Tarefa Impr_rec

Ativada pelo recebimento dos caracteres vindos da Tarefa da Aplicação Rectecla. Imprime na tela estes caracteres, pela diretiva **ajsint**. E torna-se inativa, em seguida, pela diretiva **ajend**, como a seguir:

```
void impr_rec(ch)
unsigned char ch;

{    .
     .
 sts=ajsint(intr,&reg);
     .
ajend();
}
```

5.2.12.3. Tarefas Especiais do Mestre

As tarefas descritas anteriormente são comuns a todas

as estações, permitindo que tanto possam ser transmissoras quanto receptoras. A única exceção se encontra na estação mestre, que embora incorpore todas as funções anteriores, possui algumas especiais.

5.2.12.3.1. Tarefa de Lançamento de Rede

Além do que foi exposto no item 5.2.12.1.2. esta tarefa é diferenciada por ainda disparar a tarefa de Controle de Acesso ao Meio e preencher uma tabela que indicará quais as estações estão ativas na rede, a partir da solicitação ao operador para que entre pelo teclado com os seus respectivos números, colhidos pelo AMX, pela diretiva **ajkbgc**. No momento em que o operador determinar o fim destas entradas, torna-se inativa, pela diretiva **ajend**. A tarefa é vista a seguir:

```
void lanca_rede()

{
int intr, numero_estacao,sts;
struct amxregs reg;
intr= 0x14; /* tipo de interrupção 14H */
          .
          .
ajsint(intr,&reg);
          .
          .
numero_estacao=ajkbgc();
          .
          .
          .
sts=ajtrig(gcb.idtskmrec);
          .
          .
sts=ajtrig(gcb.idtskctrl);
          .
          .
sts=ajtrig(tecl_id);
          .
          .
ajend();
}
```

5.2.12.3.2. Tarefa de Controle de Acesso (só existente no mestre)

Esta tarefa é disparada pela diretiva **ajtrig** da Tarefa Lançamento de Rede do Mestre. E existem três situações atendidas

por esta tarefa, vistas a seguir.

1) Envio da mensagem do mestre (se esta existir) para uma das estações escravas, através da execução da diretiva **ajwake** para a Tarefa Transmissão RS 422/485 do Mestre. E em seguida, a execução da diretiva **ajwatm**, onde a Tarefa de Controle de Acesso espera por **t1** para a mensagem ser enviada por essa tarefa.

2) Se a Tarefa de Transmissão RS 422/485 do Mestre não tiver mensagem a ser enviada, a diretiva **ajwapt** anula a diretiva **ajwake**, interrompendo a espera do término da fatia de tempo para a transmissão da mensagem. E ocorre, então, o envio do "token" para a próxima estação escrava da sequência e a espera por um intervalo , em que a estação possuidora do "token" enviará sua mensagem, de tempo fixo **t2**.

3) Quando a estação escrava receptora do "token" não possuir mensagem a ser enviada, ela reenvia o "token" para a estação mestre. O reenvio do "token", é uma interrupção da espera do término da fatia de tempo da estação escrava e recomeço do ciclo da Tarefa de Controle de Acesso do Mestre.

A tarefa de recepção RS 422/485 do Mestre, por conseguinte, recebe-o e sinaliza para a tarefa de Controle de Acesso ao Meio, pela diretiva **ajwake**, liberando-a da espera por **t2**.

```
void ctrl()

{ unsigned char numero_estacao, estado_estacao;
  int sts;
        .
  for(;;)
     {
     for(numero_estacao=0;numero_estacao<128;numero_estacao++)
          {if(estado_estacao==ativa)
               {sts=ajwake(gcb.idtskmtrRS);
                if(sts==AERWKP) /* sem msg para enviar */
                ajwapt(gcb.idtskmtrRS);
                else{ sts=ajwapr();
                    /*  espera por  transmissao da  mensagem
                                    do mestre */
                      sts=ajwatm(t1);
                    }
               /* envio do "token" para a proxima estacao */
               for(..;..;..)
                    {          .
                               .
                      sts=ajsint(intr,&reg);
                               .
                    }
               sts=ajwapr();
               /*  espera por  transmissao do "token"  e  da
                             mensagem do escravo */
```

```
                    sts=ajwatm(t2);
                    }
               }
          }
     }
```

5.2.12.3.3. Tarefa de Recepção RS 422/485 do Mestre

Esta tarefa é diferenciada da tarefa equivalente nas estações escravas, pelo fato de poder receber uma mensagem a ele destinado ou o **reenvio do "token"**, ao invés de poder receber uma mensagem ou o **"token"**. Assim que recebe o reenvio do "token", sinaliza para a tarefa de Controle de Acesso ao Meio, despertando-a pela diretiva **ajwake**. No caso de ser uma mensagem, tudo se passa como nas outras estações.

```
void mestre_rec()

{ struct msg msg_rec, *p_msg_rec, **pp_msg_rec;
  char tecla;
                    .
                    .
  for(;;)
     {
     for(;tecla!=ETX);)
          {
     ajsint(intr,&reg);
                    .
          if( msg )/* verifica se é msg */
          {/* se for msg */
          /*  faz chksum e transmite para a tarefa
            recepção intermediária */
               sts=ajsend(gcb.idtskinrec,0,*p_msg_rec);
               }
          else  /* se for "token" devolvido para o mestre */
               sts=ajwake(gcb.idtskctrl);
     }
     }
}
```

5.2.12.4. Tarefa Especial do Escravo

A Tarefa de Reenvio de "Token" se torna especial por só existir na estação escrava, com a finalidade de implementar algumas particularidades, vistas a seguir.

5.2.12.4.1. Tarefa de Reenvio de "Token"

É disparada pela Tarefa de Lançamento de Rede da estação escrava com a diretiva **ajtrig**. A diretiva **ajwait** faz com que ela espere por uma sinalização recebida da:

. Tarefa de Recepção RS 422/485;
. Tarefa de Transmissão RS 422/485.

5.2.12.4.1.1. Sinalização da Tarefa de Recepção RS 422/485

A sinalização da Tarefa da Recepção RS 422/485 indica que recebeu o "token", mas que a Tarefa de Transmissão RS 422/485 não estava bloqueada a espera desta permissão de transmissão. Esta sinalização ocorre como resultado da diretiva **ajwake** executada pela Tarefa de Recepção RS 422/485. e tem como consequência o reenvio do "token" para a estação mestre, ocasionado pela execução da diretiva **ajsint**. A tarefa se inicia pela diretiva **ajwapr** para forçar a anulação de qualquer diretiva **ajwake** que possa ter ocorrido antes da recepção do "token".

5.2.12.4.1.2. Sinalização da Tarefa de Transmissão RS 422/485

A sinalização da Tarefa de Transmissão RS 422/485, indica que o envio da mensagem desta estação pela RS 422/485, terminou e para apressar o envio do "token" para a próxima estação, desperta-se a Tarefa de Reenvio de "Token" pela diretiva **ajwake**.

```
void reenvio_token()

{for(;;)
      {     .
            .
       ajwapr();
       ajwait();
       for(;enquanto reenvio_token não acabar;)
           {          .
                      .
            sts=ajsint(intr,&reg);
                      .
                      .
           }
      }
}
```

5.3. Conclusão

Conforme as características dos dados, do tráfego destes dados e das aplicações desejadas no chão de fábrica, foi desenvolvido um Suporte de Comunicação específico para estes fins, visando uma simplificação das Redes de Comunicação Comerciais disponíveis, de forma a viabilizar a integração de novas aplicações e das já existentes no ambiente industrial.

Este Suporte foi implementado com caraterísticas especiais que, entre outras, provê: independência da aplicação, transparência do meio físico e transparência do Controle de Acesso ao Meio.

Entre outras aplicações, este Suporte, permite ao operador visualizar no vídeo da estação de trabalho, de forma dinâmica, o comportamento do processo frente as pertubações e/ou variações nas variávies envolvidas, tais como: "set-point" e parâmetros dos controladores.

CAPÍTULO 6 . CONCLUSÃO

6.1. Comentários

Este Suporte caracteriza-se por trabalhar sobre um núcleo multitarefas de tempo real, e possibilitar, de acordo com a aplicação, a alteração de parâmetros e informações vindos e enviados para outra(s) estação(ões), sem que se interrompa o processamento em cada estação. Desta forma, não compromete a performance de qualquer que seja o programa aplicativo, mesmo que ele seja um programa em tempo real. Isso corresponde a característica necessária para utilizá-lo em supervisão e controle de processos, conforme já ressaltada no item 5.1. Introdução.

De uma maneira geral, todos os CP's dispõem de placas de comunicação padrão com os microcomputadores compatíveis com IBM-PC. Mas, normalmente, diferentes CP's implementam diferentes protocolos de comunicação. Por isto, normalmente, não se comunicam entre si.

No caso citado anteriormente, uma outra importante aplicação pode ser obtida, utilizando este Suporte com estes microcomputadores, como interfaces entre diferentes protocolos de CP's. Onde, por exemplo, um ou mais CP's de fabricantes diferentes podem estar conectados a um microcomputador, aqui chamado Unidade 1. O mesmo sendo possível com outro microcomputador denominado Unidade 2. E o Suporte interligando as duas unidades. Deste modo, os microcomputadores poderiam fazer o papel de supervisão dos CP's e poderiam trocar informações pelo Suporte. Assim, possibilita-se a comunicação entre eles e a consequente integração dos seus diversos fabricantes num mesmo sistema.

6.2. Resultados

Para a obtenção dos resultados desta implementação, utilizou-se a aplicação desenvolvida para demonstração e placas de comunicação RS 232. Devida a estas últimas não possibilitarem ligações em estruturas multipontos, a adição de MODEM's para a devida isolação foi necessária .

Como a velocidade de comunicação dos MODEM's é baixa, a sua máxima velocidade de comunicação operacional definiu a máxima velocidade de operação do Suporte de comunicação, nesta implementação. A velocidade neles programada foi de 1200 Bauds.

O protocolo de comunicação implementado pelas placas RS 232 foi programado para uma comunicação com 1 bit de "stop", 8 bits de dados e paridade ímpar.

Na versão atual do Suporte foi previsto um possível aprimoramento com a programação entre tarefas com respostas de confirmação de recebimento de mensagem. Desta forma, a implementação desta confirmação ficou indicada no programa entre comentários, facilitando as futuras inserções necessárias para sua devida implementação.

Numa topologia em barramento com acesso ao meio por "token", onde se faz uma divisão de tempo entre cada nó de comunicação, o número de nós que se incorporam à rede influencia, naturalmente, no tempo de transmissão de uma mensagem de uma estação à outra.

Os resultados descritos em seguida foram colhidos num sistema composto por 3 nós (3 microcomputadores).

O tempo obtido a partir do instante do envio da mensagem de uma estação até a recepção por outra estação, foi de aproximadamente 1 segundo. Permitindo, ainda, a execução de outras tarefas concorrentes, como por exemplo, a impressão instantânea na tela de um carater em resposta a qualquer tecla pressionada pelo operador.

O tempo do envio do "token" pela estação mestre até a recepção da mensagem pela estação escrava também foi de menos de 1 segundo.

6.3. Perspectivas Futuras

Como ressaltado no item anterior, os tempos de resposta são influenciados pela velocidade de transmissão. Nos resultados já descritos, as características elétricas do MODEM limitaram estes tempos. Uma melhora considerável pode ser conseguida com a utilização de placas RS 485 em substituição das placas RS 232. Estas placas permitem as ligações multiponto e atingem altas velocidades de transmissão, diminuindo o tempo de transmissão no sistema. Vale ressaltar que a compatibilidade entre a placa RS 232 e RS 485 no modo de programação é integral, garantindo o bom funcionamento em suas substituições.

Um projeto de controle de processos e supervisão, já em andamento, tende a concretizar a integração de CP's de diferentes fabricantes através do Suporte desenvolvido, utilizando os microcomputadores na supervisão, e os CP's no controle de processos, conforme citado no item 6.1. Comentários. Pedidos de alterações de parâmetros e de programações provenientes de um

controle em um CP, poderão, por exemplo, ser enviadas (de acordo com a aplicação) de um microcomputador para o outro microcomputador, através do Suporte e transmitidas para o segundo CP. Isso, no entanto, não comprometerá a operação em tempo real dos programas de supervisão que poderão estar rodando nos microcomputadores.

BIBLIOGRAFIA

[1] SCHWARZ, G. Computadores em Tempo Real - Uma Introdução aos Conceitos Básicos. COPPE/UFRJ, p. 459-68, 1981.

[2] FRANCO, L.R.H.R. O Barramento de Campo FIP (Factory Instrumentation Protocol). Anais do V Seminário de Pesquisa da EFEI, p. 283-9, 1990.

[3] NOGUEIRA, J.M.S. Aspectos de Comunicação em Sistemas Distribuídos para Aplicações de Automação Industrial - Tese de Doutorado da Universidade Estadual de Campinas, Campinas, 1985.

[4] FRANCO, L.R.H.R. Desenvolvimento de uma Rede Local em Microcomputadores Compatíveis com IBM-PC - Anais do VI Seminário de Pesquisa da EFEI, Itajubá, p. 61-5, 1992.

[5] COSTA, C. Aplicação de Redes Locais de CLP's no Chão de Fábrica - Um Passo Fundamental Rumo ao CIM. Automação e Indústria, p.14-8, agosto/1989.

[6] SOUZA, L.E. Um Suporte para a Configuração Estática de Sistemas Distribuídos Utilizando a Abordagem Por Linguagem: Projeto e Implementação. Dissertação de Mestrado da Universidade Federal de Santa Catarina, Florianópolis, 1988.

[7] BARBARISI, J. O Avanço dos PC's nos Processos de Automação Industrial, Instec, p. 33-4, novembro/1991.

[8] LOBIANCO JR., W. Redes Locais na Indústria: A Tendência Natural, Automação e Indústria, p. 27-9, agosto/setembro/1987.

[9] SAITO, A.Y. Experiência de Aplicações de Redes Industriais de Comunicações. Instec, p. 54-60, agosto/1991.

[10] THOMESSE, J.P. Les Reseaux Locaux Industriels. Collection Novotique, Lausanne, 1985.

[11] GIOZZA, W.F.; ARAUJO, J.F.M. ; BELTRÃO, J.A.; SAUVÉ, J.P. Redes Locais de Computadores. Tecnologia e Aplicações, Mc Graw Hill, São Paulo, 1986.

[12] LIZUKA, K. Ligação Micro-Mainframe, Atlas, São Paulo, 1987.

[13] BRUSTOLONI, J.C.F. Técnicas de Sincronização e Comunicação entre Tarefas Concorrentes. Anais do Segundo CONAI, p. 483-95, 1985.

[14] AMX86 Multitasking Executive - Reference Manual . KADAK Products Ltd., 1987.

[15] ANDREWS, R.G. e SCHNNEIDER, F.S. Concepts and Notations for Concurrent Programming. Computing Surveys - 15(1) - 3/11 - março/1983.

[16] SCHIPER, A. Programmation Concurrente,Presses Polytechniques Romandes, 1986.

[17] HANSEN, P.B. Operating Systems Principles. Prentice Hall Inc., New Jersey, 1973.

[18] HOARE, C.A.R. Monitors: An Operating System Structuring Concept., CACM, outubro/1974.

[19] SANTOS, S. M. Programação Concorrente: Mecanismos de Comunicação e Sincronização de Processos. Tese de Doutorado da Quarta Escola de Computação, São Paulo, julho/1984.

[20] NUSSBAUMER, H. Informatique Industrielle II - Introduction à l'Informatique du Temps Réel, Presses Polytechniques Romandes, Lausanne, 1987.

[21] GONZALES, M.J.Is. Deterministic Processor Scheduling. Computing Surveys, 9(3) 173/204, setembro/77.

[22] NOGUEIRA, J.M.S. Sistemas Distribuídos e Aplicações da Automação Industrial, Anais do Segundo CONAI, São Paulo,p. 283-94, 1985.

EFEI - BIBLIOTECA MAUÁ
8200712
NÃO DANIFIQUE ESTA ETIQUETA