

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA

CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**SISTEMA DE AQUISIÇÃO E TRANSMISSÃO DE DADOS A MICROCONTROLADOR PARA
SUPERVISÃO DE TELEFONES PÚBLICOS**

Evilacy César Andrade Vieira

Campina Grande -PB

**SISTEMA DE AQUISIÇÃO E TRANSMISSÃO DE DADOS A MICROCONTROLADOR PARA
SUPERVISÃO DE TELEFONES PÚBLICOS**

Evilacy César Andrade Vieira

Dissertação apresentada à Coordenação dos Cursos de Pós-Graduação em Engenharia Elétrica da
Universidade Federal da Paraíba, em cumprimento às exigências para obtenção do grau de Mestre

Área de Concentração: Processamento da Informação / Instrumentação Eletrônica

Gurdip Singh Deep, Ph.D.

Orientador

Raimundo Carlos Silvério Freire, Doutor

Orientador

Campina Grande - PB

Dezembro de 1994



V657s Vieira, Evilacy César Andrade.
Sistema de aquisição e transmissão de dados a microcontrolador para supervisão de telefones públicos / Evilacy César Andrade Vieira. - Campina Grande, 1994. 87 f.

Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, 1994.
"Orientação : Prof. Dr. Gurdip Singh Deep, Prof. Dr. Raimundo Carlos Silvério Freire".
Referências.

1. Telefonia. 2. Eletromecânica. 3. Telefones Públicos. 4. Dissertação - Engenharia Elétrica. I. Deep, Gurdip Singh. II. Freire, Raimundo Carlos Silvério. III. Universidade Federal da Paraíba - Campina Grande (PB). IV. Título

CDU 621.395(043)

**SISTEMA DE AQUISIÇÃO E TRANSMISSÃO DE DADOS A MICROCONTROLADOR
PARA SUPERVISÃO DE TELEFONES PÚBLICOS**

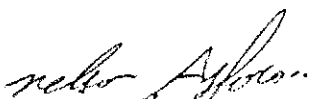
EVILACY CÉSAR ANDRADE VIEIRA

Dissertação Aprovada em 21.12.1994


GURDIP SINGH DEEP, Ph.D., UFPB
Orientador


RAIMUNDO CARLOS SILVÉRIO FREIRE, Dr., UFPB
Orientador


MARCELO SAMPAIO DE ALENCAR, Ph.D., UFPB
Componente da Banca


NELSON CAVAS ASFORA, Mestre, Eng. da EMBRATEL-PE
Componente da Banca


MISAEEL ELIAS DE MORAIS, Dr.-Ing., UFPB
Componente da Banca

CAMPINA GRANDE - PB
DEZEMBRO - 1994

Quando o trabalho desaparecer,
restará o conhecimento que fez o trabalho.
A lua sumir,
o sol que lhe dá a luz.
A vida cair,
a matéria que voltará ao cosmos.

À minha esposa Núbia,
ao bacurauzinho
e ao amigo Aléssio.

AGRADECIMENTOS

Aos professores Gurdip Singh Deep e Raimundo Carlos Silvério Freire, pela orientação e apoio dedicados ao longo deste trabalho.

Aos professores José Ewerton Pombo de Farias e Misael Elias de Moraes, pelo incentivo no início do desenvolvimento deste trabalho.

Ao engenheiro Nelson Cauás Asfora, EMBRATEL-PE, pelo apoio técnico.

Ao professor Marcelo Sampaio de Alencar, pelo apoio.

Aos funcionários das empresas TELPA, TELASA e EMBRATEL-PB, pelo auxílio técnico.

Aos funcionários da Central Telefônica da UFPB, em especial à Célia e ao Sr. Teles.

Ao CNPq pela oportunidade e auxílio concedidos para o desenvolvimento deste trabalho.

Aos amigos do Laboratório de Instrumentação e do curso de mestrado e em especial aos engenheiros Aléssio, Jorge, Stênio, Giovanni, Chaquibe, Rogério e Ilton e à professora Sheila pela incomensurável ajuda.

RESUMO

As centrais telefônicas eletromecânicas não possuem meios eficientes de supervisão dos telefones públicos a elas conectados. Apresenta-se neste trabalho o desenvolvimento dos sistemas de supervisão de telefones públicos e de auxílio ao desenvolvimento de projetos baseados no microcontrolador Intel 8031. O sistema de supervisão é subdividido em um ou mais módulos de aquisição de dados e em um programa de controle. Foi montado e testado um protótipo do módulo e uma interface homem- máquina para o programa.

ABSTRACT

The electromechanical exchanges switchings don't have efficient means of supervision of the public telephones connected to them. In this work is show the development of the systems of supervision of public telephones and of aid to the development of projects based on microcontroler Intel 8031. The system of supervision is subdivided in one or more modules of data acquisition and in a control program. It was set up and tested a module prototype and an man-machine to the program.

SUMÁRIO

RESUMO	iii
ABSTRACT	iv
SUMÁRIO	v
LISTA DE FIGURAS E TABELAS	viii
LISTA DE ABREVIACÕES E SÍMBOLOS	x
INTRODUÇÃO	1
CAPÍTULO I	6
FUNDAMENTOS DE TELEFONIA	6
1.1 - Introdução	6
1.2 - Elementos da Rede Telefônica	7
1.3 - Centrais de Comutação	7
1.4 - Sinalização de Linha	10
1.5 - Telefones Públicos	11
CAPÍTULO II	14
SISTEMA DE DESENVOLVIMENTO PARA MICROCONTROLADORES	
INTEL 8031	14
2.1 - Arquitetura	15
2.1.1 - Microcontrolador 8031 (MCU)	16
2.1.2 - Decodificador de Endereço	17
2.1.3 - EPROM de Programa	18
2.1.4 - Memória RAM de Dados	19
2.1.5 - Memória RAM de Programa	19
2.1.6 - Conversores de Tensão (Drivers RS232)	19

2.1.7 - Programa Monitor	20
2.2 - Programa de Controle da Placa de Desenvolvimento	26
2.2.1 - Buffer	28
2.2.2 - Placa	30
2.2.3 - Serial	31
2.2.4 - Dos	32
2.2.5 - Sai	33
CAPÍTULO III	35
SISTEMA DE AQUISIÇÃO DE DADOS	35
3.1 - Módulo de Aquisição e Transmissão de Dados	37
3.1.1 - Fonte de Alimentação	38
3.1.2 - Conversores TTL/RS232C e RS232C/TTL	38
3.1.3 - Microcontrolador Intel 8031 (MCU)	40
3.1.4 - Memória de Dados (RAM)	40
3.1.5 - Memória de Programa (EPROM)	41
3.1.6 - Conndicionador	41
3.1.7 - Multiplexador	42
3.1.8 - Contador	45
3.1.9 - Conversores de Nível de Tensão	45
3.2 - Programa de Controle	46
3.3 - Programa de Análise de Dados	52
CAPÍTULO IV	58
TESTES E RESULTADOS EXPERIMENTAIS	58
CONCLUSÕES	61
APÊNDICE A - MICROCONTROLADOR INTEL 8031	63
APÊNDICE B - DIAGRAMAS ESQUEMÁTICOS	74
APÊNDICE C - LISTAGEM DO PROGRAMA BÁSICO DO MÓDULO	78

REFERÊNCIAS BIBLIOGRÁFICAS

LISTA DE FIGURAS E TABELAS

Figura 1.1 - Ligação física do telefone a central	7
Figura 1.2 - Configuração de centrais telefônicas	8
Figura 1.3 - Métodos de tarifação	12
Figura 2.1 - Diagrama de blocos do Ambiente de Desenvolvimento para 8031	17
Figura 2.2a - Fluxograma da seqüência de inicialização do programa básico	22
Figura 2.2b - Fluxograma da seqüência para tratamento da interrupção da serial	24
Figura 2.2c - Fluxograma da seqüência para gravação do programa do usuário	25
Figura 2.2d - Fluxograma da seqüência para verificação do programa do usuário	26
Figura 2.2e - Fluxograma da seqüência para execução do programa do usuário	26
Figura 2.3 - Serviços e subserviços disponíveis no programa <i>TRANSFER</i>	27
Figura 2.4 - Exemplo de uma tela do programa <i>TRANSFER</i>	29
Figura 2.5 - Exemplo de tela para o subserviço <i>CARREGA</i>	30
Figura 2.6 - Exemplo de tela para o subserviço <i>APRESENTA</i>	31
Figura 2.7 - Exemplo de tela para o subserviço <i>PARÂMETROS</i>	32
Figura 2.8 - Exemplo de tela para o subserviço <i>MUDA DIR</i>	33
Figura 3.1 - Configuração do sistema proposto	36
Figura 3.2 - Diagrama de blocos do módulo de transmissão e aquisição de dados	39
Figura 3.3 - Diagrama de blocos do condicionador	42
Figura 3.4 - Diagrama de blocos da unidade de multiplexação	43
Figura 3.5 - Detalhe do diagrama de blocos do módulo	45
Figura 3.6 - Fluxograma geral do programa básico	46
Figura 3.7 - Fluxograma da rotina de aquisição e processamento de dados	49
Figura 3.8 - Fluxograma da rotina de tratamento da interrupção da serial	50
Figura 3.9 - Fluxograma da rotina de leitura dos contadores	51
Figura 3.10 - Fluxograma da rotina de inicialização de um contador	52
Figura 3.11 - Exemplo de tela do <i>SADAM - TP</i>	53

Figura 3.12 - Exemplo de tela do serviço <i>ARQUIVOS</i>	54
Figura 3.13 - Exemplo de tela para o serviço <i>COMUNICAÇÃO</i>	54
Figura 3.14 - Exemplo de tela para o serviço <i>MODEM</i>	55
Figura 3.15 - Exemplo de tela para o serviço <i>DOS</i>	56
Figura 3.16 - Exemplo de tela para o serviço <i>ESTATÍSTICA</i>	56
Figura 3.17 - Exemplo de tela para o subserviço <i>MÉDIA</i>	57
Figura 3.18 - Exemplo de tela para o serviço <i>SAI</i>	57
Tabela I: Sinalização de Linha	10
Tabela II: Mapeamento de dispositivos de E/S em memória	18
Tabela III: Vetores de interrupções do programa do usuário	22
Tabela IV: Caractere recebido pela serial x desvio do programa	23
Tabela V: Sinais básicos para interfaceamento com o modem	39
Tabela VI: Entradas ativas para os multiplexadores pares em função da seleção	44
Tabela VII: Entradas ativas para os multiplexadores ímpares em função da seleção	44
Tabela VIII: Tabela da verdade para atualização dos contadores	47

LISTA DE ABREVIACÕES E SÍMBOLOS

- ASCII** - "American Standart Code for Information Interchange";
- A/D** - Analógico para digital;
- CI**s - Circuitos Integrados;
- CPA**s - Centrais de Programa Armazenado;
- CTS** - Modem pronto para envio;
- D/A** - Digital para analógico;
- DSR** - Modem ligado;
- DTR** - Terminal de dados ligado;
- EPROM** - Memória não volátil programável por eletricidade;
- GND** - Terra;
- GNDC** - Terra da central teleônica;
- MCU** - Microcontrolador;
- RAM** - Memória volátil de leitura e escrita;
- RD** - Dado recebido;
- RTS** - Requisição para envio;
- ROM** - Memória de leitura somente;
- Sadam-TP** - Sistema de Aquisição de Dados a Microcontrolador para Telefones Públicos;
- SRF**s - Registradores de funções especiais;
- TD** - Dado a ser transmitido;
- TTL** - Lógica Transistor-Transistor - padrão de circuitos integrados;
- TELASA** - Telecomunicações de Alagoas;
- TELEBRÁS** - Sistema de Telecomunicações do Brasil;
- TELESP** - Telecomunicações de São Paulo;
- TELPA** - Telecomunicações da Paraíba e;
- TP**s - Telefones Públicos.

INTRODUÇÃO

A história da telefonia teve início com a invenção do telefone, em 1876, por GRAHAM BELL [1]. Desde então, passou-se mais de um século de evolução desta área da engenharia, oriunda das necessidades emergentes de comunicação em cada passo do crescimento da humanidade.

Um dos primeiros problemas da telefonia foi conectar dois terminais telefônicos de forma automática através de sistemas de comutação. Neste sentido, os progressos realizados foram na direção da completa automação da operação. Dispositivos eletromecânicos foram desenvolvidos, de modo a substituir a comutação manual, culminando, 30 anos após a descoberta do telefone, com a primeira central pública eletromecânica automática. Nos 60 anos seguintes, as centrais eletromecânicas continuaram a evoluir, surgindo o sistema rotativo (1915), o sistema Ericsson (1923) e o sistema de barras cruzadas (em torno de 1950). Em meados dos anos 60 e início dos anos 70, em função dos avanços conseguidos na área da eletrônica, começaram a ser desenvolvidas as primeiras centrais semi-eletrônicas. A mais recente geração de centrais de comutação, totalmente eletrônica, surgiu em fins da década de 70. Estas últimas centrais, denominadas Centrais de Programa Armazenado (CPAs), são controladas por microprocessadores, permitindo uma adaptação contínua da comutação às novas necessidades do sistema telefônico [2].

O Brasil recebeu suas duas primeiras centrais baseadas no sistema de barras cruzadas (*Crossbar*) em 1953, destinadas à automatização das localidades de Itacemópolis-SP e Muri-RJ. Estas duas centrais foram pioneiras na América Latina [1].

Atualmente, o país possui um parque instalado de centrais de comutação misto, constando de centrais eletromecânicas (derivadas das centrais *Crossbar*) e digitais. Regido por normas da TELEBRAS, desde 1988, o crescimento do parque se dá obrigatoriamente com a instalação de CPAs, que possibilitam às empresas operadoras flexibilidade, capacidade de análises lógicas e acesso remoto a informações para supervisão do sistema telefônico [2].

A troca imediata do parque instalado de centrais eletromecânicas é inviável do ponto de vista econômico. Tecnicamente as centrais eletromecânicas funcionam e os procedimentos de manutenção e operação são amplamente dominados, através da experiência das operadoras. Com isto, sua substituição deverá ser feita de forma gradual. A TELESP, maior empresa do sistema TELEBRAS, possui metas de digitalização total do parque instalado no estado de São Paulo apenas para o início do próximo século. Como as centrais eletromecânicas e as CPAs não oferecem as mesmas facilidades de supervisão, torna-se útil o desenvolvimento de sistemas de interfaceamento entre as centrais eletromecânicas e computadores digitais, de forma a melhorar o desempenho destas centrais [3, 4].

Motivação do Trabalho

Uma categoria especial de terminais telefônicos é a dos telefones públicos (TPs), sejam eles a ficha ou a cartão indutivo. Como as centrais eletromecânicas não possuem meios próprios de supervisão dos TPs, a coleta das fichas depositadas nos TPs a ficha constitui um problema de difícil solução prática. Conforme pesquisa realizada junto à TELPA (Telecomunicações da Paraíba), observou-se que as operadoras de

telecomunicações fazem tal coleta utilizando métodos estatísticos e, atualmente, procuram a terceirização deste serviço como alternativa para o problema.

Um telefone público a ficha, com seu cofre cheio, permanece normalmente inoperante, provocando prejuízos à operadora e à qualidade do serviço prestado aos usuários. Os prejuízos devem-se principalmente ao vandalismo que está relacionado, entre outros fatores, à qualidade do serviço. Danificar a pintura dos aparelhos, remover o monofone, roubar o cofre de fichas, colocar água pelas fendas do equipamento são algumas das práticas de vandalismo. Desta forma, anualmente, uma boa parte dos telefones públicos é danificada, obrigando as operadoras a manter um serviço permanente de manutenção dos aparelhos.

Por outro lado, a supervisão, através de um sistema automatizado, permite além de determinar a quantidade de fichas coletadas pelos TPs, fazer levantamentos acerca da utilização desses telefones, diagnosticar falhas por meio de métodos estatísticos, traçar roteamento de busca a cofres cheios e auxiliar no planejamento da expansão do número dos TPs em determinadas áreas. Isto possibilita a redução nos custos do sistema de coleta e melhorar a qualidade do serviço prestado à população.

Visando atender às necessidades de supervisão dos TPs pelas centrais eletromecânicas, foi considerada a possibilidade de se desenvolver um sistema de aquisição e transmissão de dados para permitir a contagem de fichas coletadas pelos cofres de TPs. Por outro lado, após alguns contatos com o pessoal técnico da TELASA (Telecomunicações de Alagoas) e da TELPA (Telecomunicações da Paraíba), descobriu-se que é uma prática das operadores a utilização da mesma sinalização para tarifação de telefones públicos a ficha ou a cartão, sejam eles conectados à centrais digitais ou eletromecânicas. Desta forma, o desenvolvimento de um sistema de supervisão baseada naquela sinalização poderia ser útil tanto às centrais eletromecânicas quando às digitais.

Uma proposta inicial do sistema de supervisão teve aceitação para apresentação no 11º Simpósio Brasileiro de Telecomunicações em Natal-RN [3], bem como no 4º Congresso Internacional de Telecomunicações, durante o TELEXPO 93, em São Paulo-SP [4]. Além disso, há uma boa possibilidade de se implantar o sistema nas empresas operadoras de telecomunicações nacionais e da América Latina, através do apoio da Fundação Parque Tecnológico do Estado da Paraíba, bem como do Programa para Desenvolvimento de *Software* para Exportação - Softex 2000, fomentado pelo CNPq.

Apesar de se ter cogitado a implementação do sistema utilizando-se um microprocessador comercial, viu-se que o uso de microcontroladores era mais adequado pois este dispositivo possui, no mesmo circuito integrado, subsistemas para processamento, aquisição e transmissão de dados, necessários a este trabalho. Procurou-se, então, um microcontrolador comercial de custo baixo e de aquisição fácil no mercado nacional, chegando-se assim ao microcontrolador Intel 8031.

Embora este tenha sido o objetivo inicial do trabalho, foi criado ainda um sistema de desenvolvimento e auxílio a projetos baseados no microcontrolador 8031, com o objetivo de se suprir a deficiência de uma ferramenta deste tipo para o desenvolvimento do trabalho.

Organização do Texto

Após esta introdução, no Capítulo 1 apresenta-se os fundamentos teóricos sobre telefonia de forma a situar este trabalho no seu âmbito. O sistema de auxílio a projetos com o 8031 é apresentado no Capítulo 2, com objetivo de descrever sua arquitetura e apresentar o seu programa de controle.

No Capítulo 3 é apresentado o sistema de supervisão proposto neste trabalho, descrevendo-se suas características como um todo e abordando-se seus subsistemas de

hardware e o programa de análise dos dados. As considerações sobre os testes do sistema e os resultados obtidos são apresentados no Capítulo 4.

Por fim, são apresentadas as conclusões seguidas pelos Apêndices A, B e C, onde são apresentados, respectivamente, o microcontrolador Intel 8031, os diagramas esquemáticos do sistema proposto e a listagem do programa básico do sistema.

FUNDAMENTOS DE TELEFONIA

Neste capítulo são abordados os fundamentos de telefonia, dando-se destaque aos elementos da rede telefônica, às centrais de comutação, à sinalização de linha trocada entre duas centrais telefônicas, bem como aos telefones públicos disponíveis atualmente.

1.1 - Introdução

A telefonia é a área da engenharia que estuda os procedimentos para o estabelecimento de um caminho de conversação entre dois usuários quaisquer, com a melhor qualidade possível [5, 6].

De início, a preocupação da telefonia, originada pela própria invenção do telefone consistia em permitir que um par de telefones se intercomunicasse. Evidentemente, as necessidades de comunicação da sociedade tornaram-se mais amplas. Desejava-se então, permitir a qualquer telefone selecionar um, dentre outros aparelhos de forma prática e econômica. Tal intento foi permitido pelo desenvolvimento da central de comutação, podendo esta ser manual, eletromecânica ou eletrônica [1].

1.2 - Elementos da Rede Telefônica

Para cada terminal telefônico sai um par de fios (linha telefônica) do Distribuidor Geral (*DG*), elemento de interligação entre a central comutadora e a planta externa, conforme mostrado pela figura 1.1.

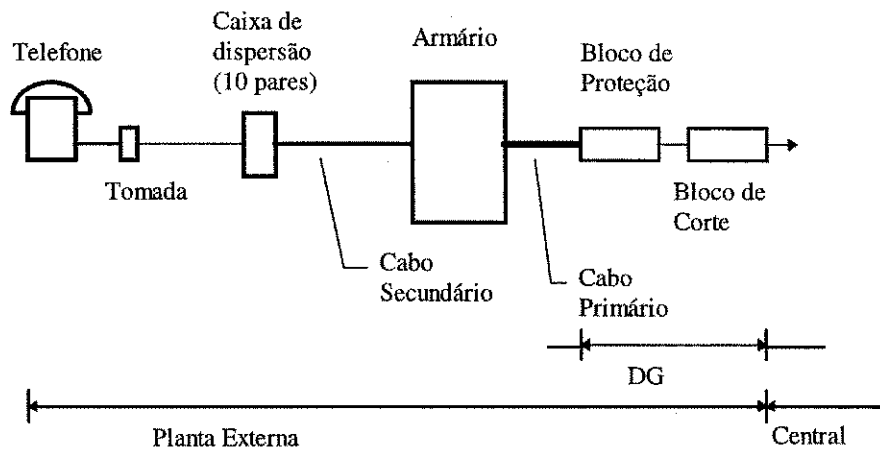


Figura 1.1. - Ligação física do telefone à central

Do *DG* sai grande quantidade de linhas telefônicas unidas em cabos primários, ou alimentadores. Este distribuidor é formado pelos blocos de proteção e de corte. Enquanto que o bloco de proteção contém os fusíveis de proteção das linhas, o bloco de corte possui os pinos de corte, onde o par de fios da linha, composto pelos fios denominados de *Linha A* e *Linha B*, é interrompido. É no bloco de corte que os circuitos de entrada da central são interligados às linhas telefônicas [1].

1.3 - Centrais de Comutação

As centrais comutadoras têm as funções básicas de concentrar, interligar, gerenciar, distribuir e tarifar as chamadas geradas pelos usuários [7]. A função de

interligar os pares de fios de dois terminais telefônicos pode ser atendida por uma única central se o número de usuários for pequeno e concentrado em uma região [1].

Com a expansão geográfica e quantitativa dos terminais telefônicos instalados, surgiu a impossibilidade técnico-econômica de se atender todos os usuários a partir de uma única central telefônica. Houve então a necessidade de uma solução economicamente viável para o serviço de comutação. A otimização econômica conduziu ao uso de centrais múltiplas, buscando-se diminuir a distância entre os usuários e a central e reduzir os custos de operação e a complexidade administrativa e operacional da rede telefônica [1].

Tecnicamente é impraticável interligar todas as centrais locais entre si, quando se tem um número razoavelmente grande de centrais. A solução mais razoável é a interconexão destas a uma outra central, que é chamada de central trânsito ou *tandem*, conforme a configuração mostrada na figura 1.2 [6]. Os usuários se interligam ao sistema telefônico através das centrais locais e estas se ligam entre si através de cabeação, denominada *Cabos Tronco*, pela qual são trocados os sinais de conversação bem como de sinalização. As ligações interurbanas são dirigidas, independentemente de seu destino, a uma central *tandem* interurbana, cuja função é interligar a rede local à rede nacional de telefonia.

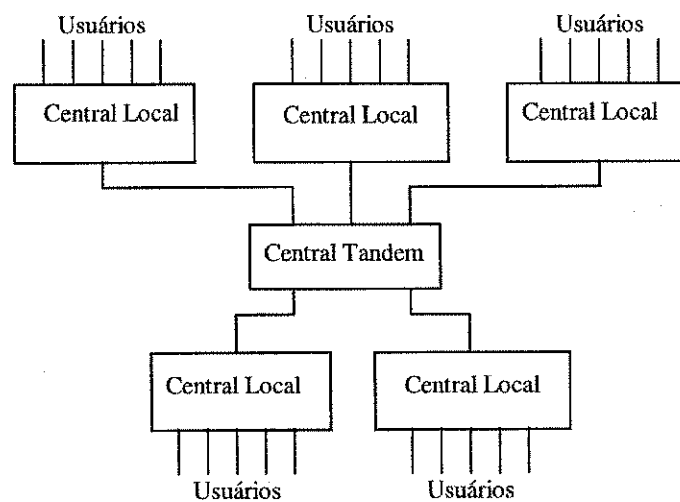


Figura 1.2. - Configuração de centrais telefônicas

Segundo a aplicação, as centrais podem se destinar ao atendimento público ou privado. As centrais privadas possuem as mesmas características das centrais públicas, a menos de algumas atribuições setoriais específicas.

No Brasil, as centrais públicas são responsáveis pelo tratamento de todo o serviço básico de telefonia, bem como possibilitam a oferta de serviços especiais como hora certa, teledespertador, etc [7].

Em relação à tecnologia, o parque de centrais telefônicas instalado no Brasil consta de centrais eletromecânicas e CPAs.

Algumas das vantagens das CPAs em relação às eletromecânicas são:

- espaço físico reduzido;
- redução no consumo de energia;
- alto grau de modularidade;
- comutação com alta capacidade de tráfego;
- maior eficiência administrativa;
- medição automática de tráfego;
- tarifação automática;
- possibilidade de detecção e diagnóstico de falhas no sistema e;
- maior capacidade de supervisão do sistema telefônico.

O uso crescente e a diminuição do custo dos microcomputadores além da construção de programas de computador com interfaces homem-máquina amigáveis têm estimulado o desenvolvimento de sistemas informatizados para supervisão e controle das centrais eletromecânicas. Hoje é possível, por exemplo, encontrar-se sistemas de medição de tráfego telefônico [6] e supervisão de chamadas telefônicas [3, 4]. Desta forma, através dos microcomputadores e de sistemas de interface entre aquelas máquinas e as centrais eletromecânicas, a diferença entre as tecnologias que separa as CPAs das eletromecânicas tem diminuído.

1.4 - Sinalização de Linha

A sinalização de linha é feita por um conjunto de nove sinais, listados na tabela I, trocados entre os circuitos de entrada e saída das centrais telefônicas [8]. Pela importância neste trabalho, destes sinais, destaca-se o sinal de tarifação, o qual comanda a inversão de polaridade dos terminais dos telefones públicos, com o que se coleta uma ficha ou se diminui uma unidade de crédito do cartão indutivo. Essa inversão aparece em um dos terminais, denominado neste trabalho de *Linha B*, como um pulso de terra, com características semelhantes ao sinal de tarifação, ou seja, amplitude de 48 volts em relação à fonte de alimentação de -48 volts da central telefônica.

O sinal de tarifação tem ainda a função de acionar o contador de pulsos de tarifação local das linhas telefônicas que, nas centrais eletromecânicas, é um contador mecânico incrementado pela ação de um relé.

Tabela I: Sinalização de Linha, **A** - central que originou a chamada, **B** - central chamada

SINAL	SENTIDO*	OBJETIVO
Ocupação	A → B	Desencadear a busca a um circuito de entrada disponível
Atendimento	A ← B	Indicar o instante do atendimento do usuário chamado e desencadear o processo de tarifação da chamada em A
Desligar para Trás	A ← B	Indicar o instante em que o usuário chamado colocou o monofone no gancho
Desligar para Frente	A → B	Indicar o instante em que o usuário chamador colocou o monofone no gancho
Confirmação de Desconexão	A ← B	Confirmar a recepção do sinal de Desligar para Frente
Desconexão Forçada	A ← B	Solicitar à central A o envio do sinal de Desligar para Frente
Tarifação	A ← B	Acionar o contador de pulsos de tarifação
Bloqueio	A ← B	Indicar bloqueio ou falha nos circuitos de B
Rechamada	A → B	Acionar a rechamada do usuário em B

Ao se completar uma chamada, condição sinalizada pelo sinal de atendimento, a central chamada inicia a cadeia de tarifação. Para chamadas dentro da mesma área

numérica, denominadas chamadas locais, normalmente é utilizado o método de tarifação KARLSSON MODIFICADO, o qual leva em conta apenas o tempo da conversação. Por este método, o sinal de atendimento dá origem a um pulso no contador de pulsos da linha telefônica e a seguir, o circuito gerador de pulsos da central é conectado ao contador. Devido à aleatoriedade do primeiro pulso, este é cancelado, conforme mostrado pela figura 1.3. Para as centrais eletromecânicas, os pulsos de comando dos contadores têm duração de 150 ms \pm 30 ms e amplitude de 48 volts em relação à alimentação de -48 volts da central [1].

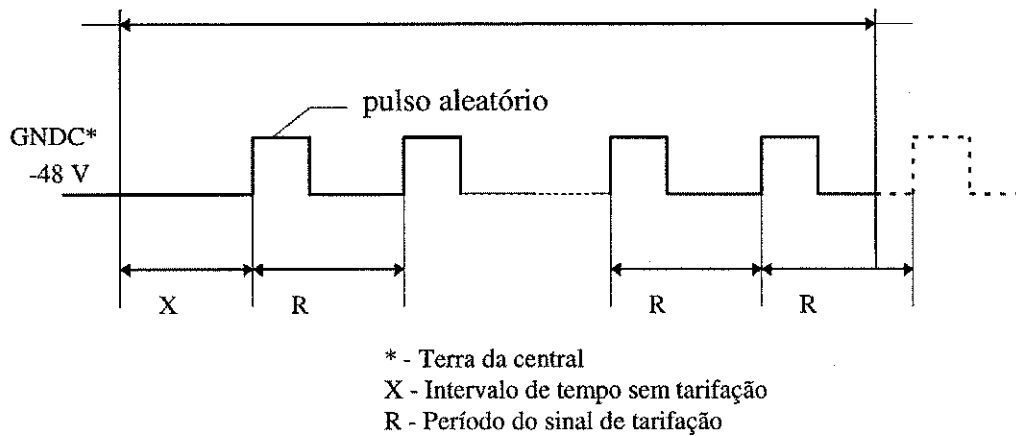
O método original de tarifação desenvolvido por KARLSSON [1], consistia apenas em conectar o gerador de pulsos da central ao contador, no instante do atendimento. Desta forma, se houvesse desligamento antes do tempo X (Figura 1.3), a chamada não seria tarifada. Uma vez que o cancelamento do primeiro pulso do gerador exige uma lógica adicional ao circuito de tarifação, algumas centrais não o cancelam, penalizando o usuário caso o tempo X seja pequeno em relação a R (Figura 1.3.).

Para a tarifação fora da área numérica, a qual leva em consideração o tempo e a distância entre as centrais, utiliza-se a multimedição por tempo e distância, método que consiste em selecionar o período de tarifação R de acordo com a distância entre as centrais. Para isto utiliza-se um banco de geradores com períodos que vão desde poucos segundos até dezenas de minutos. Os geradores são selecionados a partir da análise do número discado pelo usuário chamador [1].

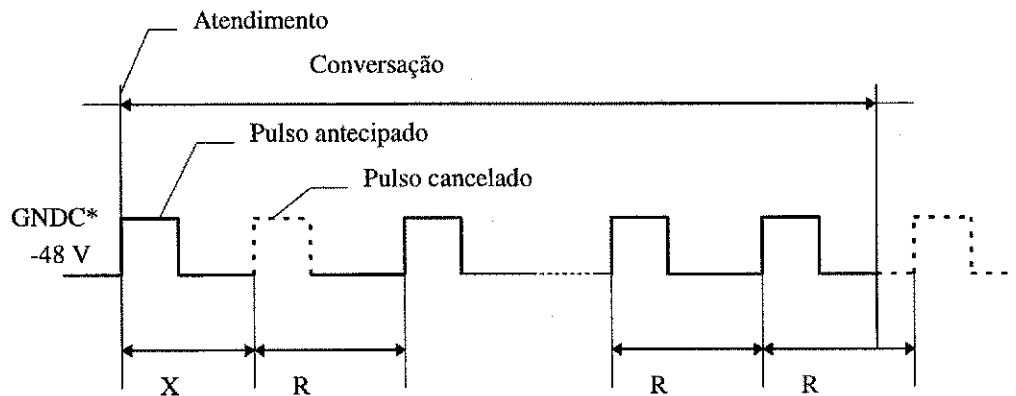
1.5 - Telefones Públicos

Os telefones públicos (TPs) constituem uma categoria de terminais telefônicos para uso público, instalados em locais de acesso à população em geral [1].

Do ponto de vista da tarifação de chamadas, os TPs são operados com fichas metálicas ou a cartão indutivo. Os TPs a cartão constituem os equipamentos mais modernos e, devido a fatores de ordem econômica e social, tais como custo elevado do equipamento, baixo poder aquisitivo da população e freqüentes depredações, ainda correspondem a uma parcela pequena dos TPs instalados. Estes telefones operam com cartões que possuem uma quantidade de células de crédito de efeito indutivo.



Método KARLSSON original



Método KARLSSON modificado

Figura 1.3 - Métodos de tarifação

Os TPs a ficha operam com pré-pagamento da chamada por fichas. Além da parte relativa ao circuito normal de transmissão telefônica, possuem circuitos para

controlar a coleta de fichas. As fichas são depositadas em uma abertura apropriada e sua validade é verificada por gabaritos dimensionais e sensores de liga metálica [1].

Para sua utilização, o usuário faz o depósito de uma ou mais fichas em uma abertura que dá acesso a um reservatório de fichas. Ao se completar uma ligação, o sinal de atendimento é enviado pela central chamada, o que dá início à cadeia de tarifação. Como se deseja que a tarifação da chamada seja feita pela coleta de fichas, para cada sinal de tarifação, a polaridade dos terminais do telefone público é invertida durante $150 \text{ ms} \pm 30 \text{ ms}$, o que provoca a transferência de uma ficha do reservatório para um cofre de coleta. Para cada ficha é feito, então, um pré-pagamento e o usuário tem um crédito inicial a seu favor. Ao repor o monofone, as fichas não transferidas são devolvidas [1]. Após se encher o cofre de coleta, o telefone permanece inoperante até a troca do cofre por outro vazio.

Como as centrais eletromecânicas não têm meios de supervisão automática dos TPs a ela conectados, a coleta dos cofres de fichas de TPs cheios é feita baseada em estatísticas do número de fichas coletadas pelo telefone em um determinado período de tempo.

Neste capítulo apresentou-se uma visão geral acerca de alguns fundamentos de telefonia, dando-se ênfase aos conceitos utilizados neste trabalho, no próximo capítulo será apresentado o sistema de desenvolvimento e auxílio a projetos baseados no microcontrolador Intel 8031.

SISTEMA DE DESENVOLVIMENTO PARA MICROCONTROLADORES INTEL 8031

A utilização crescente de microcontroladores em sistemas digitais provocou o surgimento de *softwares* emuladores destes dispositivos, bem como de placas de desenvolvimento para avaliação e testes de projetos, baseados nestes dispositivos, de forma rápida e eficiente. Embora de utilidade incontestável, os ambientes de desenvolvimento (em *software* ou *hardware*) têm custo elevado e são de difícil aquisição no mercado nacional [9].

A metodologia de desenvolvimento deste trabalho incluiu a concepção e implementação de um sistema de desenvolvimento e auxílio a projetos, ou ambiente de desenvolvimento, baseados no microcontrolador Intel 8031. O ambiente tem o objetivo de avaliar a viabilidade e desempenho dos projetos baseados naquele dispositivo, oferecendo facilidades de desenvolvimento e depuração de programas para o microcontrolador bem como testes de *hardware* dos projetos.

Esta ferramenta mostrou-se bastante útil ao trabalho desenvolvido, principalmente durante a depuração das rotinas para o microcontrolador, uma vez que o ambiente permite alterar o programa a ser executado pelo microcontrolador sem a

necessidade de se gravar novamente a memória de programa do tipo EPROM, agilizando o desenvolvimento do trabalho como um todo.

Neste capítulo é apresentada a arquitetura do sistema, bem como são feitas considerações acerca do programa básico do microcontrolador, denominado programa monitor, e do programa de controle do ambiente.

2.1 - Arquitetura

O ambiente de desenvolvimento segue a mesma filosofia dos ambientes existentes no mercado, ou seja, fornecer ao usuário todos os recursos do microcontrolador ou microprocessador em que são baseados e ainda possibilitar a utilização de ferramentas úteis no desenvolvimento do projeto tais como interfaces para comunicação serial, de acesso a dispositivos de entrada e saída de dados, programas de interface com microcomputadores para depuração de rotinas, etc.

O ambiente consta de um circuito eletrônico, chamado de placa de desenvolvimento, baseado no 8031; de um programa básico para o microcontrolador, chamado de programa monitor e; de um programa para controle do ambiente, denominado programa *TRANSFER*, compatível com microcomputadores padrão IBM-PC. O diagrama de blocos do ambiente de desenvolvimento é mostrado na figura 2.1.

A placa de desenvolvimento possui dos seguintes blocos funcionais principais:

- microcontrolador 8031 (MCU);
- decodificador de endereços;
- memória não volátil para programas (EPROM de PROGRAMA);
- memória RAM externa para dados do usuário (RAM de DADOS);
- memória RAM externa para programa (RAM de PROGRAMA);
- barramento de dados (D0-D7);
- barramento de endereços (A0-A15);

- barramento para seleção de dispositivos de E/S (S0-S7) e;
- interface para comunicações seriais síncrona e assíncrona (DB25).

Para a avaliação de projetos baseados no microcontrolador, o usuário desenvolve seu projeto de *hardware* podendo utilizar as linhas das portas P1 e P3 do 8031, dos barramentos de endereços de 16 *bits* (A0-A15), de dados de 8 *bits* (A0-A7) e de seleção de dispositivos mapeada em memória de 8 *bits* (S0-S7), além da interface para comunicação serial (DB25).

Para a avaliação e testes de programas, o programa-fonte do usuário deve ser compilado de forma a obter-se o programa-objeto em formato ASCII, o qual é transferido do microcomputador para a placa de desenvolvimento através do cabo serial de três vias conectado às interfaces para comunicação serial do microcomputador e da placa. O programa é gravado na RAM de PROGRAMAS a partir do endereço 2000H. Após a conclusão da transferência, o usuário pode solicitar ao microcontrolador, através do programa *TRANSFER*, a execução do programa.

A seguir apresenta-se cada um dos blocos funcionais da placa de desenvolvimento.

2.1.1 - Microcontrolador 8031 (MCU)

O microcontrolador é o dispositivo principal da placa de desenvolvimento. Uma descrição detalhada deste dispositivo foge ao objetivo deste trabalho, mas algumas informações básicas de interesse são apresentadas no Apêndice A.

As portas 0 e 2 do microcontrolador são utilizadas para endereçamento de memórias externas, enquanto que as portas 1 e 3 estão disponíveis para o usuário. As linhas $\overline{\text{WR}}$, $\overline{\text{RD}}$, RXD e TXD da porta 3 são utilizadas para controle de escrita, leitura

das memórias externas e para recepção e transmissão de dados pela porta serial, respectivamente.

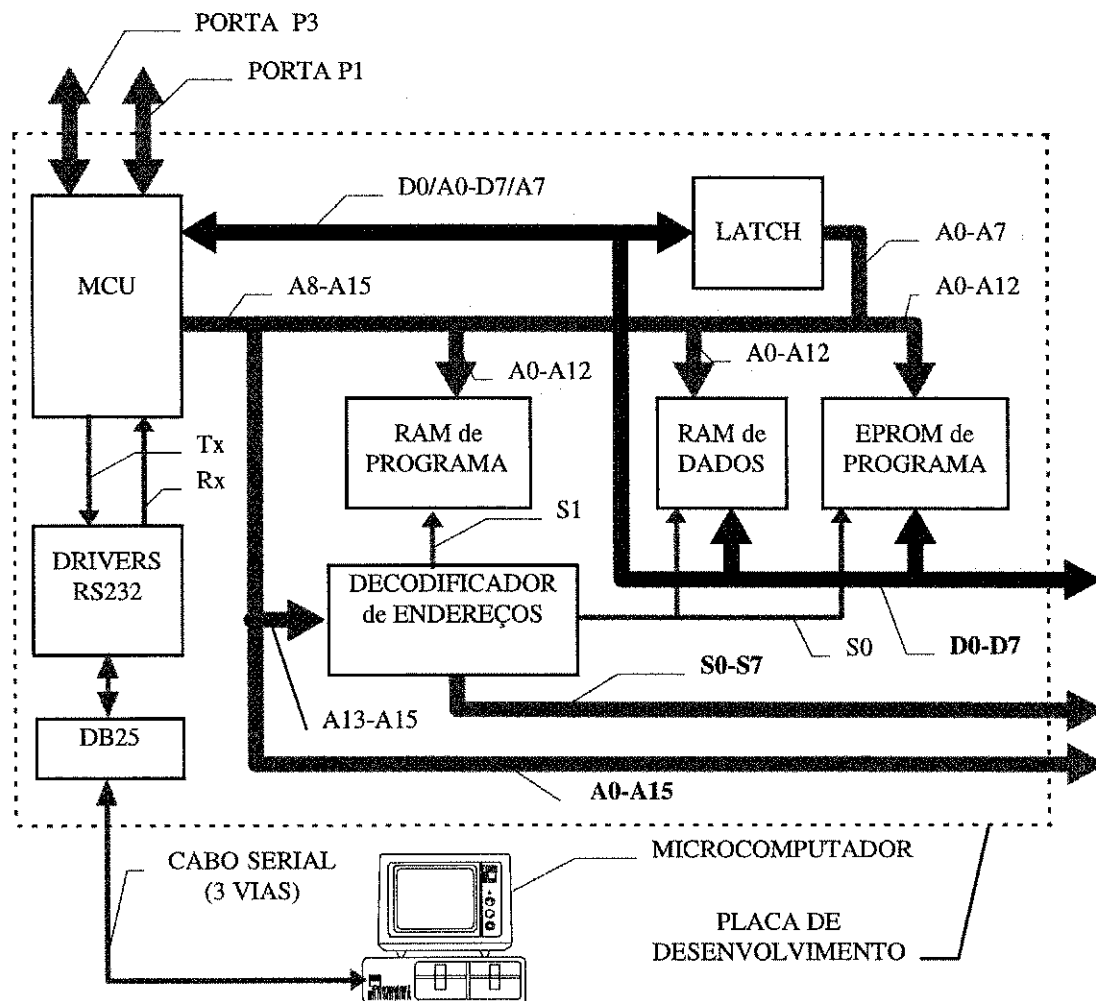


Figura 2.1 - Diagrama de blocos do Ambiente de Desenvolvimento para o 8031

2.1.2 - Decodificador de Endereços

O decodificador de endereços tem a função de habilitar uma dentre oito linhas de saída, dependendo da palavra de endereçamento de 3 *bits* (A13-A15) em sua entrada. Para sua implementação usou-se o CI 74138 [10], um circuito de seleção de custo baixo e aquisição fácil. Utilizando-se as 3 linhas mais significativas da palavra de

endereçamento de 16 *bits* do microcontrolador, como entrada do dispositivo, consegue-se dividir a capacidade total de endereçamento de 64 *kbytes* em 8 grupos de 8 *kbytes*, conforme mostrado na tabela II. Assim, um acesso a uma determinada faixa de endereços habilita a saída correspondente do decodificador, mapeando-se o acesso a dispositivos externos em memória.

2.1.3 - EPROM de Programa

O bloco EPROM de Programa é constituído de uma memória EPROM de 32 *kbytes*, o CI 27C256 [11] de tecnologia CMOS, habilitada através da linhas $\overline{\text{PSEN}}$ do microcontrolador e S0 do decodificador de endereços. Esta EPROM é mapeada na faixa de endereçamento de 0000H a 1FFFH, totalizando-se um espaço de 8 *kbytes*, inferior ao espaço de 32 *kbytes* disponível no CI. Isto deve-se ao fato de que CIs de memórias EPROMs com capacidades de 8 e 16 *kbytes* terem custo mais elevado que a utilizada no trabalho.

A função deste dispositivo é armazenar o programa básico da placa de desenvolvimento (programa monitor), o qual controla a comunicação serial entre o microcomputador e a placa de desenvolvimento, a execução de programas do usuário e o atendimento às interrupções.

Tabela II - Mapeamento de dispositivos de E/S em memória

Faixa de endereço	Saída ativa	Dispositivo habilitado
0000H-1FFFH	S0	EPROM de PROGRAMA ou RAM de DADOS, dependendo do sinal de seleção de memória de dados $\overline{\text{PSEN}}$
2000H-3FFFH	S1	RAM de PROGRAMA
4000H-5FFFH	S2	E/S do usuário
6000H-7FFFH	S3	E/S do usuário
8000H-9FFFH	S4	E/S do usuário
A000H-BFFFH	S5	E/S do usuário
C000H-DFFFH	S6	E/S do usuário
E000H-FFFFH	S7	E/S do usuário

2.1.4 - Memória RAM de Dados

A RAM de dados é uma memória RAM CMOS de 8 *kbytes*, CI 65116 [12], cuja finalidade é armazenar os dados temporários do usuário. Esta memória está mapeado na faixa de endereços de 0000H a 1FFFH e sua habilitação é feita pelas linhas \overline{WR} e \overline{RD} da porta 3 do microcontrolador.

2.1.5 - Memória RAM de Programa

O bloco Memória RAM de Programa é constituído por um dispositivo de memória RAM de 8 *kbytes*, CI 65116 [12], com a função de armazenar o programa do usuário a ser executado pelo microcontrolador, simulando uma memória de programa. Como o microcontrolador habilita a memória de programa através da linha \overline{PSEN} , a habilitação de leitura da memória RAM de programa também é feita por esta linha e a escrita na memória é habilitada normalmente pela linha \overline{WR} .

Conforme a tabela II, esta memória está mapeada na faixa de endereços de 2000H a 3FFFH. Desta forma a comutação do controle do processamento pelo programa monitor para o programa do usuário é feito desviando-se o fluxo do programa armazenado na EPROM para o primeiro endereço da memória RAM de programa, através de uma instrução de salto incondicional, a saber, LJMP 2000H.

2.1.6 - Conversores de tensão (Drivers RS232)

Os conversores são dispositivos de auxílio à comunicação serial, transformando os níveis de tensão de saída do microcontrolador, compatíveis com os níveis TTL, em valores característicos da interface serial do microcomputador, a qual é baseada no padrão RS232C, ou ainda, fazendo a operação inversa, ou seja, convertendo os níveis RS232C para valores TTL. Estas conversões permitem então, a interligação entre as interfaces seriais do microcomputador e o microcontrolador da placa de

desenvolvimento. Para a implementação deste bloco foram utilizados os CIs de interface 1488 e 1489 [13].

2.1.7 - Programa Monitor

O programa monitor realiza a interligação lógica entre os diversos blocos da placa, de forma a permitir o controle da placa de desenvolvimento bem como seu interfaceamento com um microcomputador padrão IBM-PC. Este programa está armazenado na EPROM e tem as funções básicas de:

- inicializar a placa de desenvolvimento;
- atender pedidos de interrupções;
- carregar o programa do usuário na RAM de programa;
- verificar a gravação do programa do usuário e;
- executar o programa do usuário.

A primeira função do programa monitor é inicializar a placa de desenvolvimento, seguindo a seqüência mostrada na figura 2.2a. Na inicialização, a comunicação serial é programada para modo assíncrono, taxa de transferência de 1200 bps, oito *bits* de dados, um *bit* de paridade e um *bit* de parada; as linhas para controle de acesso às memórias e para comunicação serial são colocadas em nível lógico alto; a interrupção da porta serial é habilitada com prioridade alta e as demais são desabilitadas e; o sinalizador que indica qual o programa que está em execução na placa é colocado em nível lógico 0.

Após a inicialização, o programa permanece em laço até que seja recebido um caractere pela porta serial, o que gera um pedido de interrupção ao microcontrolador. A interrupção é então atendida pelo microcontrolador conforme o fluxograma da figura 2.2b.

Uma interrupção é processada buscando-se o vetor de interrupção em memória EPROM, onde está também armazenado o programa monitor. Como o programa do usuário, armazenado em memória RAM, também pode dispor das interrupções, a rotina de tratamento de uma determinada interrupção deve ter a informação de qual programa tem o controle do processamento de forma a desviar o fluxo do programa para a rotina de tratamento individual de um dos dois programas. Esta informação é fornecida por um sinalizador armazenado na última posição de memória *bit*-endereçável, de endereço 2FH.7, chamado de *FLAG*. Desta forma, na execução do programa monitor, *FLAG* é igual a 0, caso contrário *FLAG* é colocado em 1. Como o sinalizador é utilizado na rotina de atendimento de interrupção da porta serial, sua posição de memória correspondente não deve ser utilizada pelo programa do usuário.

A rotina de tratamento de uma interrupção deve verificar o estado de *FLAG* e desviar o fluxo para um endereço da EPROM ou da RAM de programa.

No programa do usuário devem ser definidos vetores de interrupção semelhantes ao encontrados em EPROM, conforme mostrado na tabela III.

Ao final da seqüência de inicialização, o programa monitor aguarda um comando do usuário, o qual é enviado pela interface serial do microcomputador. Os comandos disponíveis ao usuário são acessados através do programa *TRANSFER*, o qual pode enviar um dos três comandos disponíveis, a saber: 1. gravação na RAM de programa; 2. verificação do conteúdo da RAM de programa e; 3. execução do programa do usuário.

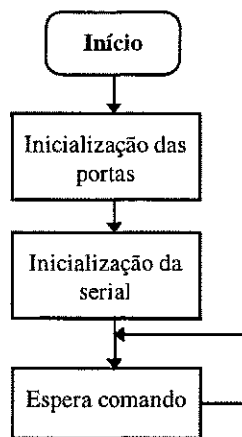


Figura 2.2a - Fluxograma da seqüência de inicialização do programa básico

Tabela III - Vetores de interrupções do programa do usuário

Interrupção	Endereço do vetor
$\overline{\text{INT0}}$	2003H
T/C0	200BH
$\overline{\text{INT1}}$	2013H
T/C1	201BH
SERIAL	2023H

Como a interrupção da serial foi habilitada na rotina de inicialização, a recepção de um caractere pela serial gera uma interrupção que será atendida pela seqüência para tratamento de interrupção, cujo fluxograma é mostrado na figura 2.2b.

Após a desabilitação de novas interrupções da serial, a rotina de tratamento verifica se a interrupção foi gerada durante a execução do programa do usuário ou não. No primeiro caso, o fluxo é desviado para o endereço 2023H e, no outro caso, verifica-se qual o caractere recebido e o fluxo do programa é desviado para uma das rotinas da tabela IV.

Tabela IV - Caractere recebido pela serial x desvio do programa

Caractere recebido	Desvio
Gravação (0E6H)	Rotina de gravação da RAM de programa
Verificação (0FCH)	Rotina de verificação da gravação
Execução (0FAH)	Rotina de execução do programa do usuário
Outros	Retorno à rotina de interrupção

O fluxograma da rotina de gravação do programa do usuário é mostrado na figura 2.2d. Esta gravação é requisitada pelo envio do comando de gravação (0E6H) pelo microcomputador. Ao receber o comando, o microcontrolador devolve o caractere para confirmação e, após a sua recepção, o microcomputador o devolve novamente. Este protocolo de confirmação é comum às demais rotinas e permite que os programas do microcomputador e da placa de desenvolvimento tenham a informação de que a comunicação serial entre os dois sistemas funciona sem problemas. Após passar pelo protocolo, o programa monitor aguarda o endereço inicial e a quantidade de *bytes* que será transferida, guardando esses valores em memória para o uso posterior da rotina de verificação. A seguir, os *bytes* são então transferidos até se chegar ao último, quando o programa monitor retorna da interrupção da serial e aguarda novo comando do usuário.

Após a gravação do programa, o microcomputador solicita sua verificação transmitindo o caractere de verificação (0FCH). Após o protocolo de confirmação, o programa monitor envia ao microcomputador o conteúdo da memória RAM de programa gravada, conforme o fluxograma da figura 2.2d.

Após a gravação e sua verificação, o programa do usuário poderá ser executado através do envio do comando de execução (0FAH), conforme fluxograma da figura 2.2e. Após o protocolo de confirmação o programa monitor primeiro coloca o sinalizador *FLAG* em 1, indicando à rotina de atendimento de interrupção da serial que o programa do usuário será executado e depois desvia o fluxo para o primeiro endereço da RAM de programa, ou seja, 2000H, dando-se início ao programa do usuário.

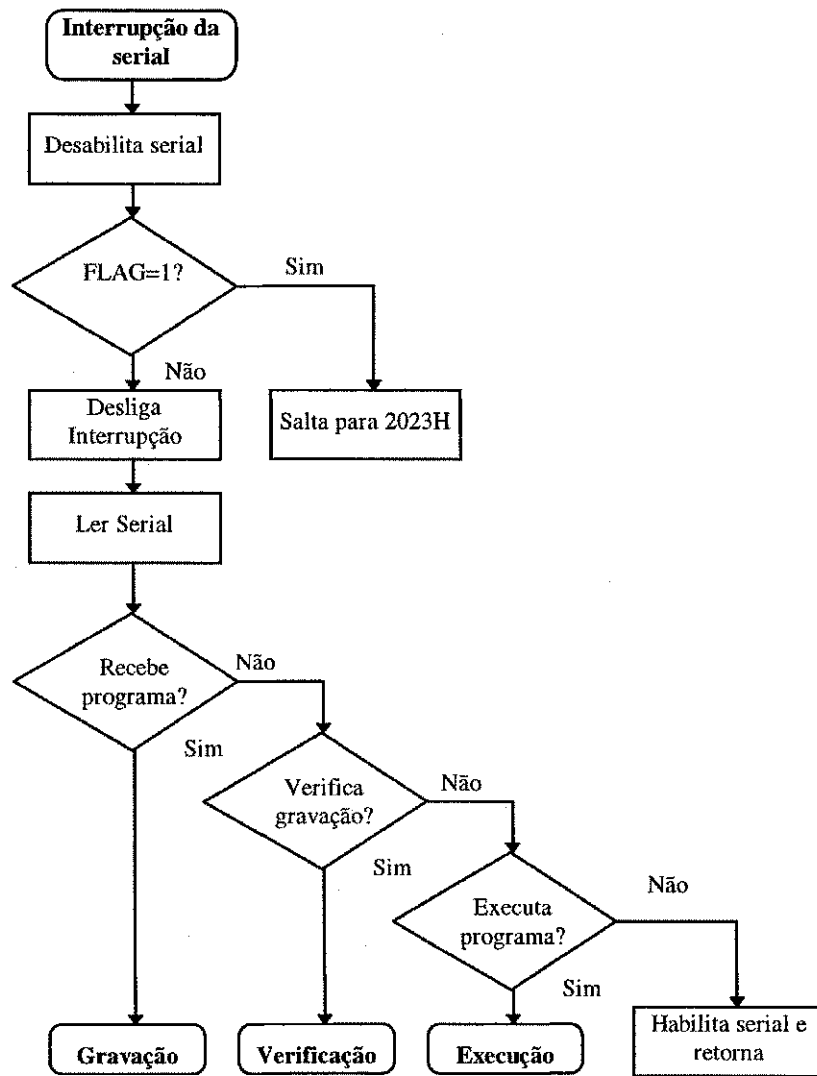


Figura 2.2b - Fluxograma da seqüência para tratamento da interrupção da serial

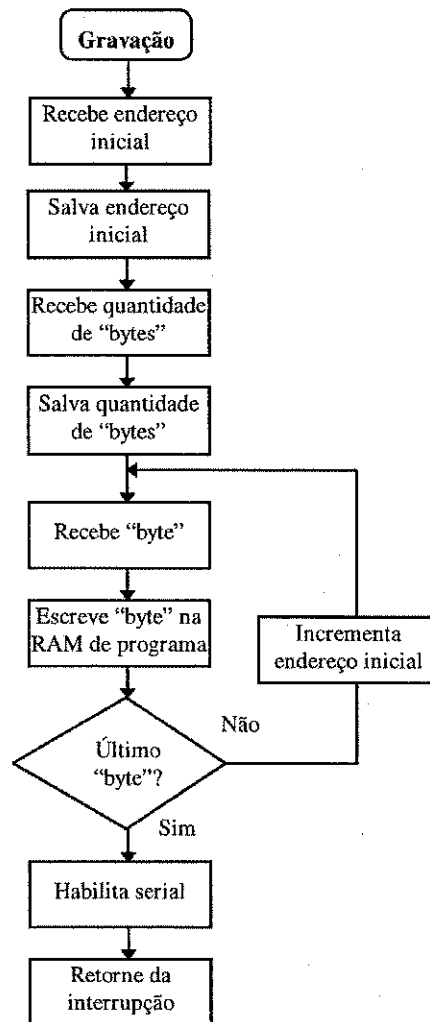


Figura 2.2c - Fluxograma da sequência para gravação do programa do usuário

Para devolver o controle do processamento ao programa monitor, o usuário deverá reinicializar a placa de desenvolvimento através de um *reset* por *hardware*, colocando a entrada RESET do microcontrolador em nível baixo, ou desviar o fluxo do programa para a rotina *INÍCIO*, que equivale a uma reinicialização da placa por programação.

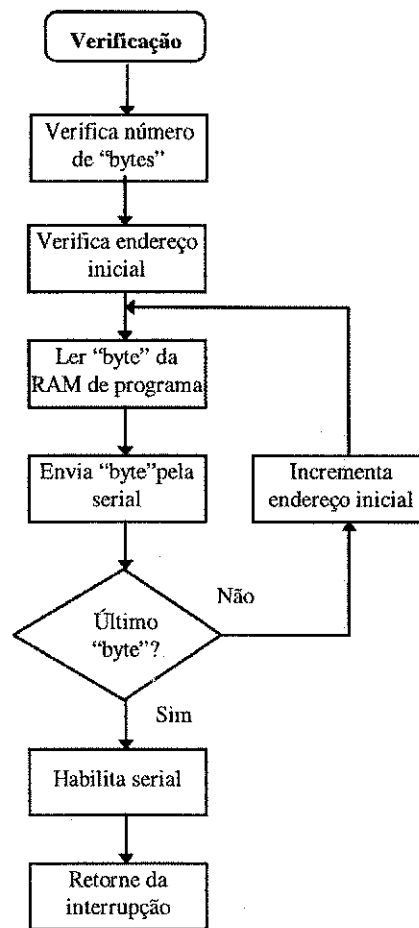


Figura 2.2d - Fluxograma da sequência para verificação do programa do usuário

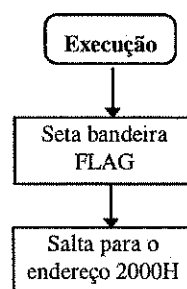


Figura 2.2e- Fluxograma da sequência para execução do programa do usuário

2.2 - Programa de Controle da Placa de Desenvolvimento

O programa de controle da placa de desenvolvimento, chamado de *TRANSFER*, tem o objetivo de fornecer uma interface homem-máquina amigável para o controle da placa através de um microcomputador compatível com o IBM-PC.

O *TRANSFER* oferece cinco serviços ao usuário, selecionáveis através da movimentação por uma barra de opções horizontal, a saber: 1. *BUFFER*, 2. *PLACA*, 3. *SERIAL*, 4. *DOS* e 5. *SAI*. Cada serviço pode possuir subserviços, cuja seleção é feita pela movimentação por *menus* verticais. A figura 2.3 apresenta todos os serviços e subserviços do programa.

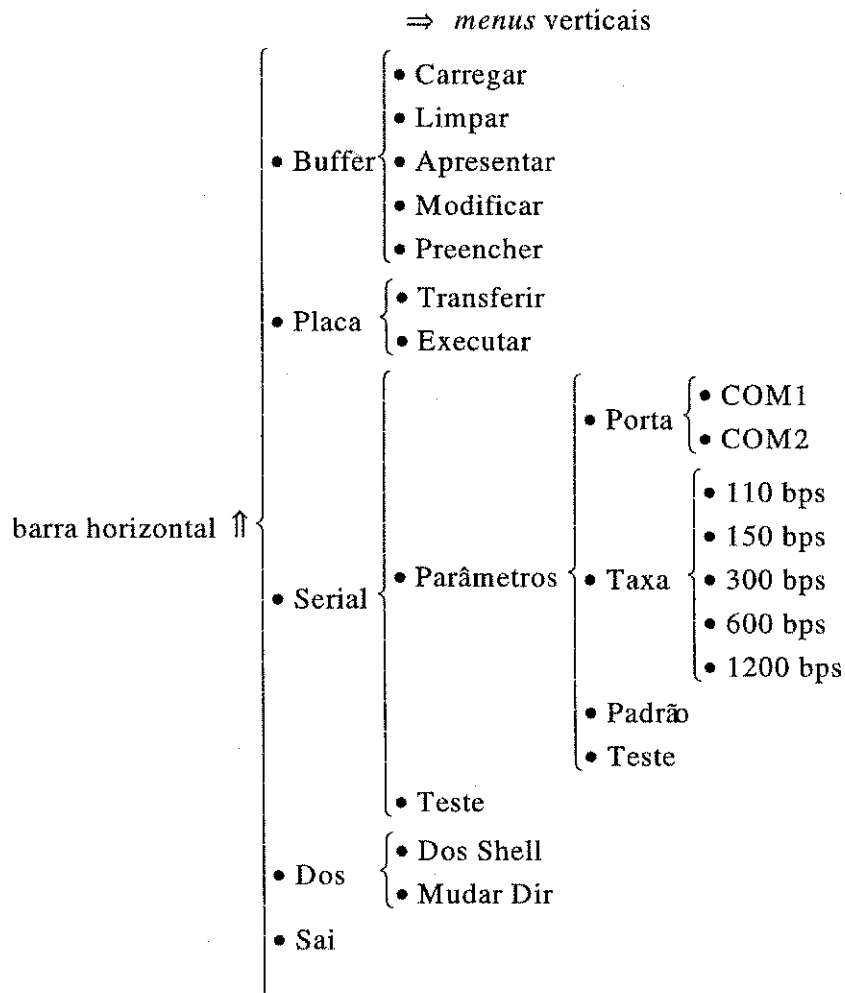


Figura 2.3 - Serviços e subserviços disponíveis no programa *TRANSFER*

Uma tela típica do *TRANSFER* é mostrada na figura 2.4, na qual pode-se observar todos os elementos de uma tela do programa. Neste exemplo, temos:

- a barra horizontal, na linha superior da tela, com um serviço *BUFFER* ativado;
- a indicação de que o programa está processando ou à espera de entrada de dados;

- o *menu* vertical do serviço *BUFFER* com o subserviço *MODIFICA* selecionado;
- a janela de apresentação e entrada de dados, no centro da tela e;
- três linhas de mensagens, nas linhas inferiores da tela, informando o subdiretório corrente, o arquivo corrente com o respectivo tamanho em *bytes* e mensagens de advertência e auxílio à navegação do usuário.

A navegação pela barra de opções horizontal é feita por meio das teclas de seta para direita e para esquerda, movimentando-se assim de um serviço para outro seqüencialmente. O serviço a ser selecionado permanece em destaque e a tecla ENTER o seleciona, abrindo o *menu* de opções verticais para aquele serviço. Pode-se ainda pressionar uma tecla de atalho para um determinado serviço, correspondente à letra em destaque para o serviço na barra horizontal. Uma vez aberto um *menu* vertical, pode-se selecionar um subserviço por processo semelhante ao utilizado na barra vertical, utilizando-se as teclas de setas para cima e para baixo e a tecla ENTER ou ainda as teclas de atalho para o subserviço requerido.

A seguir, serão apresentados todos os serviços e subserviços do *TRANSFER*, além de alguns exemplos de telas.

2.2.1 - BUFFER

O serviço *BUFFER* tem a função de manipular uma área de memória do microcomputador, denominada *buffer*, para armazenar o programa do usuário. Para tanto existem cinco subserviços: 1. *CARREGA*, 2. *LIMPA*, 3. *APRESENTA*, 4. *MODIFICA* e, 5. *PREENCHE*.

O subserviço *CARREGA* tem a função de fazer a leitura de um arquivo em formato ASCII contendo o programa-objeto do usuário. Isto é feito solicitando-se ao usuário o nome do arquivo, podendo-se alterar o subdiretório corrente do disco pressionando-se a tecla de função F2. Arquivos não encontrados ou não válidos são

rejeitados pelo programa, que permanece requisitando o nome do arquivo até que uma entrada seja válida ou o usuário pressione a tecla ESC. A tela correspondente a este subserviço é mostrada na figura 2.5.

A limpeza do *buffer*, ou seja, o seu preenchimento com o valor (0FFH) é feito através do subserviço *LIMPA*. Após a limpeza, a mensagem “Buffer está limpo. Pressione uma tecla...” será apresentada na linha de mensagens.

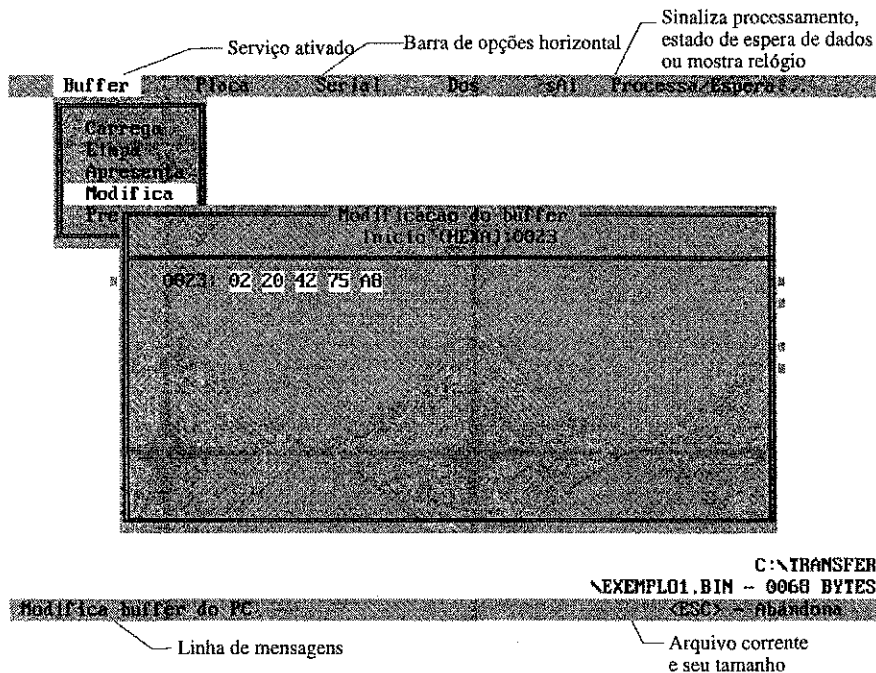


Figura 2.4 - Exemplo de uma tela do programa *TRANSFER*

O subserviço *APRESENTA* destina-se à apresentação do conteúdo de um trecho do *buffer* com endereço inicial e final fornecidos pelo usuário. Este conteúdo é apresentado em uma janela no centro da tela, conforme mostrado na figura 2.6.

No subserviço *MODIFICA*, cuja função é fazer a modificação do conteúdo do *buffer*, solicita-se ao usuário o endereço inicial para o início das alterações. De posse desse endereço, o programa apresenta seu conteúdo e espera que o usuário digite um novo conteúdo. Pressionando-se a tecla ENTER para cancelar a alteração daquele

endereço ou ESC para abandonar o subserviço. Um exemplo de tela para este serviço é apresentado na figura 2.4.

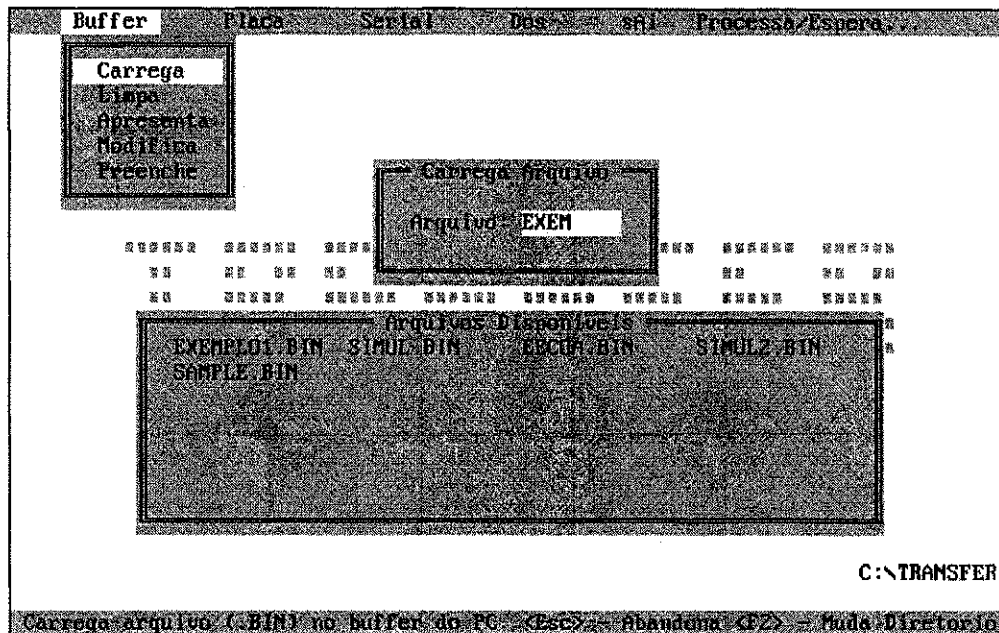


Figura 2.5 - Exemplo de tela para o subserviço *CARREGA*

Para o serviço *BUFFER*, existe ainda a opção de preencher um trecho do *buffer* com uma constante (subserviço *PREENCHE*). Para isto, são solicitados ao usuário os endereços inicial e final e o valor da constante, conforme mostrado pela figura 2.7.

2.2.2 - *PLACA*

O serviço *PLACA* oferece dois subserviços, um para a transferência do programa do usuário, previamente armazenado no *buffer* pelo subserviço *CARREGA*, do microcomputador para a placa de desenvolvimento e outro para a execução daquele programa.

A transferência é realizada pelo subserviço *TRANSFERE*, que envia todo o conteúdo do *buffer* pela porta serial do microcomputador e depois recebe o conteúdo enviado e verifica se a transferência foi realizada sem erros de transmissão. A linha de

mensagens informa ao usuário o andamento da transferência e sinaliza possíveis erros na operação.

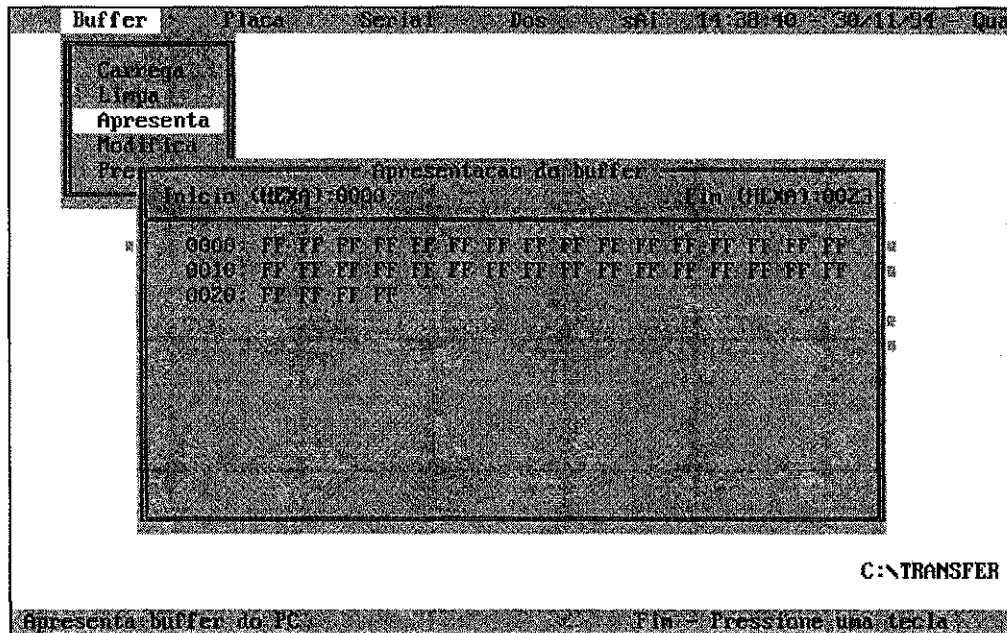


Figura 2.6 - Exemplo de tela para o subserviço *APRESENTA*

2.2.3 - *SERIAL*

O serviço *SERIAL* oferece as opções de selecionar a porta serial em que está conectada a placa de desenvolvimento, configurar a taxa de transferência de dados e testar a comunicação entre microcomputador e placa. Para isto, oferece os subserviços *PARÂMETROS* e *TESTE*.

Ao ser selecionado, o subserviço *PARÂMETROS* abre um *menu* vertical oferecendo quatro opções, a saber:

1. *PORTA* - seleciona a porta COM1 ou COM2 para a comunicação de dados;
2. *TAXA* - escolha a taxa de transferência de dados (110, 150, 300, 600 ou 1200 bps);
3. *PADRÃO* - muda parâmetros para os valores padrão, ou seja, seleciona a porta COM2 com taxa de 1200 bps e;

4. *TESTE* - verifica se a comunicação entre o microcomputador e a placa de desenvolvimento, respeitando-se os parâmetros escolhidos, está normal. Esta opção realiza um teste idêntico ao solicitado pelo subserviço *TESTE* do serviço *SERIAL*.

A escolha da porta ou da taxa de transferência é feita de formas semelhantes. Ao selecionar-se, por exemplo a opção *PORTA*, um *menu* vertical oferece as portas disponíveis e o usuário escolhe uma delas através das teclas de movimentação ou de atalho e da tecla ENTER. A tela para escolha da porta é mostrada na figura 2.7.

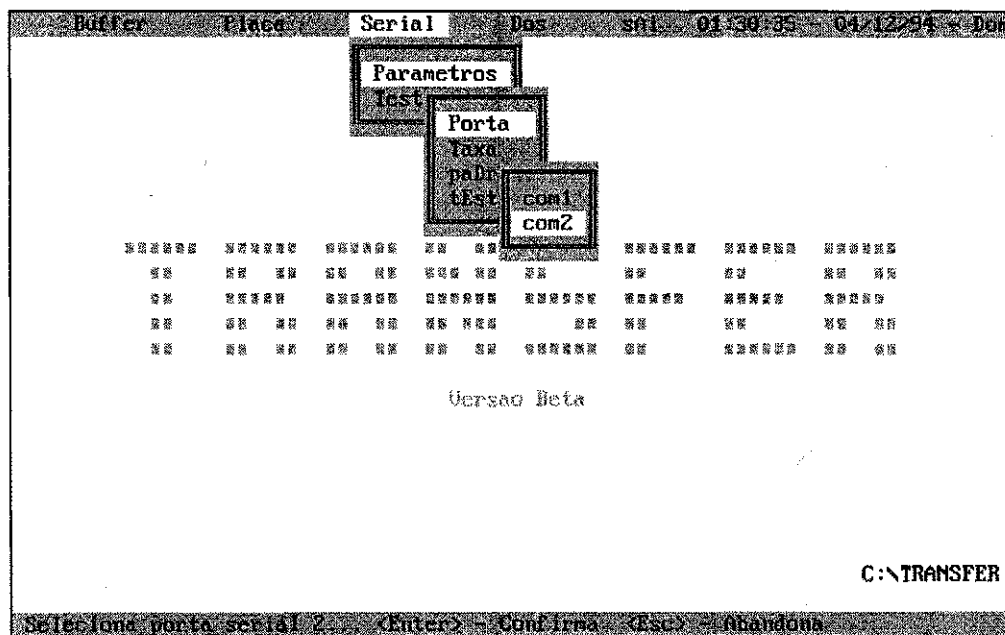


Figura 2.7 - - Exemplo de tela para o subserviço *PARÂMETROS*

2.2.4 - *DOS*

Como o *TRANSFER* não possui serviços internos de edição e compilação de programas fontes para o 8031, o serviço *DOS* permite ao usuário suspender temporariamente a execução do *TRANSFER* e voltar ao sistema operacional, de onde podem ser executados programas com aqueles serviços. Desta forma, para depurar um programa, o usuário pode executá-lo através do *TRANSFER* e utilizar aplicativos do sistema operacional para alteração e recompilação do programa. Após obter o programa-

objeto novo, pode-se voltar ao *TRANSFER* e executá-lo por meio dos serviços *BUFFER* e *PLACA*, abordados anteriormente.

Um outro subserviço do *DOS*, o *MUDA DIR*, troca o subdiretório corrente de forma a permitir a busca de arquivos do usuário em qualquer ponto das unidades de disco rígido ou flexível do microcomputador. Este subserviço pode ser chamado ainda à partir do subserviço *CARREGA*, pressionando-se a tecla de função F2.

A troca para um determinado subdiretório é feita digitando-se seu nome e uma lista de todos os subdiretórios disponíveis é apresentada na janela central da tela, conforme mostrado na figura 2.8.

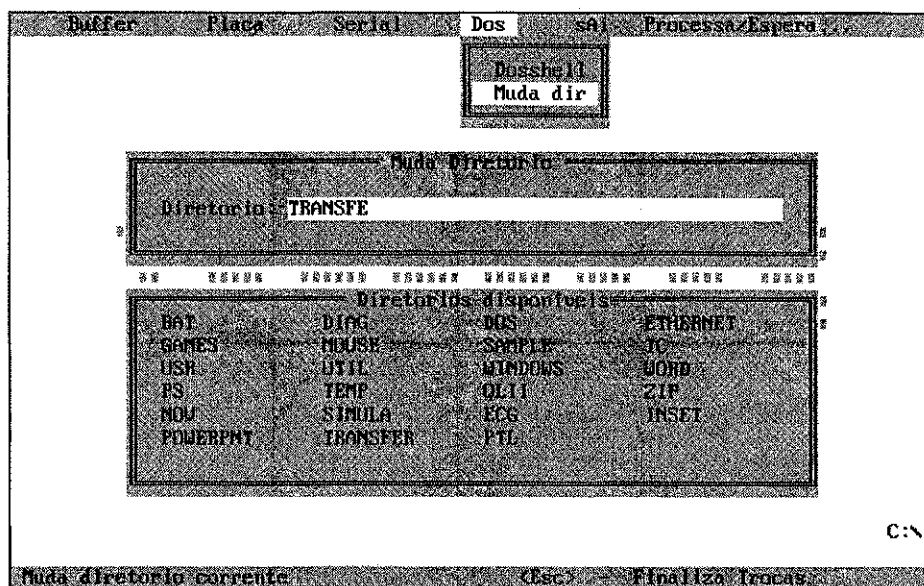


Figura 2.8. - Exemplo de tela para o subserviço *MUDA DIR*

2.2.5 - *SAI*

A opção *SAI* da barra horizontal permite ao usuário abandonar o *TRANSFER*, perdendo-se o conteúdo armazenado no *buffer*. Para isto, a mensagem "Encerrar programa... Confirma? (S/N)" é apresentada na linha de mensagens e a confirmação do usuário é aguardada até ser pressionada a tecla S para sim e N para não.

O sistema apresentado neste capítulo auxiliou o desenvolvimento do sistema para supervisão de telefones públicos proposto neste trabalho, mostrado no próximo capítulo.

SISTEMA DE AQUISIÇÃO DE DADOS

Neste capítulo é apresentado o sistema proposto neste trabalho, o qual tem a função básica de fazer o interfaceamento entre centrais de comutação telefônica eletromecânicas e microcomputadores, permitindo a supervisão dos telefones públicos conectados a estas centrais.

O sistema está subdividido em um ou mais módulos de aquisição e transmissão de dados e em um programa de aquisição dos dados enviados pelos módulos a um microcomputador, através de linha telefônica discada. A figura 3.1 mostra a configuração deste sistema.

Uma central eletromecânica pode suportar até 10.000 linhas telefônicas. Desse total de linhas, no máximo 2% é destinado aos telefones públicos, conforme norma da TELEBRAS. Assim, os módulos foram projetados de forma a permitir a contagem de pulsos de inversão de polaridade de até 200 TPs, armazenando a contagem atualizada em memória RAM de 2 *kbytes*. Como o limite máximo de fichas que pode ser coletado por um TP está em torno de 1700, para cada telefone está associado um contador de 16 *bits* formado pela junção de 2 *bytes* de memória.

Um programa básico permanece em execução contínua nos módulos, fazendo a aquisição de dados e aguardando um comando, originado pelo microcomputador, de

leitura dos dados armazenados ou de reinicialização de um determinado contador.

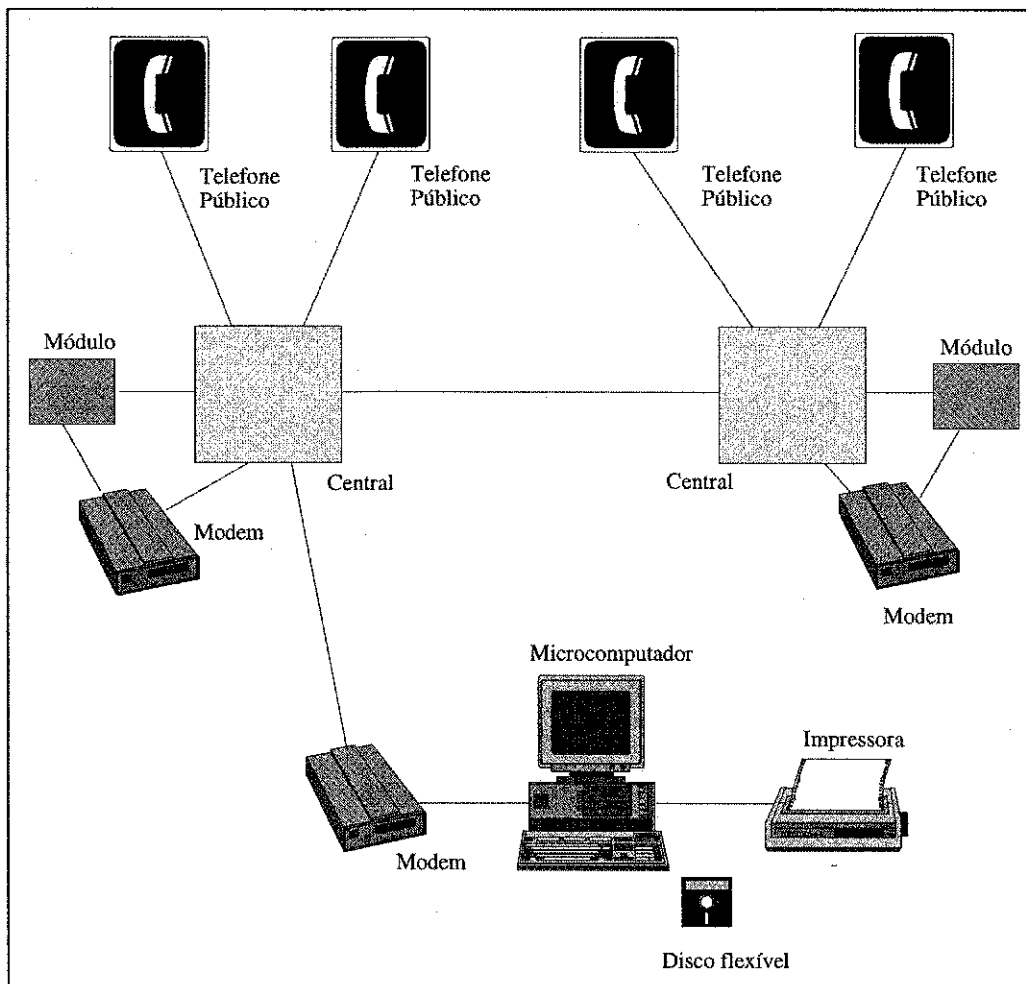


Figura 3.1. - Configuração do sistema proposto

Para se fazer a leitura dos dados de um determinado módulo, o programa de aquisição de dados, chamado de *SADAM-TP*, envia um comando de discagem para o modem conectado ao microcomputador. O modem chamado deve ser programado internamente para atender a chamada e, desta forma, conectar o microcomputador ao módulo através de uma determinada linha telefônica.

Uma vez estabelecida a conexão, solicita-se a transferência de dados por meio de um protocolo semelhante ao desenvolvido para a comunicação entre o programa *TRANSFER* e a placa de desenvolvimento. A leitura dos contadores é obtida transferindo-se 400 *bytes* da memória RAM do módulo para o microcomputador,

correspondentes aos conteúdos dos contadores dos telefones interligados ao módulo. A reinicialização de contadores é feita de forma semelhante, ou seja, o programa envia um comando de reinicialização e o número do contador, que varia de 0 a 199 correspondendo a 200 TPs; e, então o programa básico do microcontrolador coloca tal contador em zero.

A partir da contagem lida pelo *SADAM-TP*, pode-se determinar quais os TPs a ficha que estão com cofres cheios ou em vias de esgotamento ou, para o caso de TP a cartão, a quantidade de créditos utilizados desde a última leitura.

Pode-se ainda ter a indicação de possíveis telefones defeituosos, selecionando-se aqueles cujas contagens não sofreram alteração de uma leitura para outra.

O *SADAM-TP* é constituído de serviços para gerência de dados e comunicação remota de dados. Esse programa caracteriza-se principalmente por sua versatilidade em controlar inúmeros módulos a partir de um único microcomputador, podendo centralizar a supervisão dos telefones em um ponto onde são tomadas decisões de gerência [3].

Nas próximas seções serão analisados em detalhes o módulo de aquisição e transmissão de dados, o programa básico bem como a interface homem-máquina do *SADAM-TP*.

3.1 - Módulo de Aquisição e Transmissão de Dados

O módulo de aquisição e transmissão de dados é um sistema baseado no microcontrolador Intel 8031 que, como sugere sua designação, realiza os serviços básicos de aquisição e transmissão de dados. A figura 3.2 mostra o módulo em diagrama de blocos, o qual é constituído das seguintes unidades funcionais:

- fonte de alimentação;
- conversores TTL/RS232C e RS232C/TTL;

- microcontrolador (MCU);
- memória de dados (RAM);
- memória de programa (EPROM);
- condicionador;
- multiplexador;
- contador binário e;
- conversores de nível de tensão.

3.1.1- Fonte de Alimentação

A fonte de alimentação recebe as tensões de -48 volts e terra da central telefônica (GNDC) e fornece ao módulo todas as tensões necessárias ao seu funcionamento, a saber: +5, +12, -12, GND, -36 e -48 volts. As tensões de saída são obtidas deslocando-se a referência de terra da central para uma tensão próxima de -21 volts em relação a GNDC e utilizando-se circuitos integrados reguladores de tensão para obter as tensões de +12, +5 e -12 volts. A tensão de -36 volts é obtida a partir de um diodo zener de 15 volts x 1 watt e de um seguidor de emissor, conforme mostrado no diagrama esquemático desta unidade, mostrado no Apêndice B.

3.1.2 - Conversores TTL/RS232C e RS232C/TTL

A unidade de conversão de níveis TTL para RS232C e vice-versa tem a função de interligar o microcontrolador, que utiliza níveis de tensão padrão TTL, ao modem, que usa níveis compatíveis com a interface RS232C. Sua implementação foi baseada em CIs de interface 1488 e 1489 [13]. Esta unidade possui um conector DB-25 com pinagem padrão RS232C [14] para interligar-se a um modem comercial.

Para realizar a comunicação, o modem conectado ao módulo deve ser programado de forma a ficar em estado de espera e receber chamadas telefônicas

automaticamente. Ele necessita, para sua interligação com o terminal de dados, neste caso o microcontrolador, de quatro sinais de controle (RTS, CTS, DSR e DTR), dois sinais de dados (TD e RD) e um sinal de referência de tensão de terra (GND), conforme mostrados na tabela V [15]. Todos esses sinais são providos pela unidade de conversão.

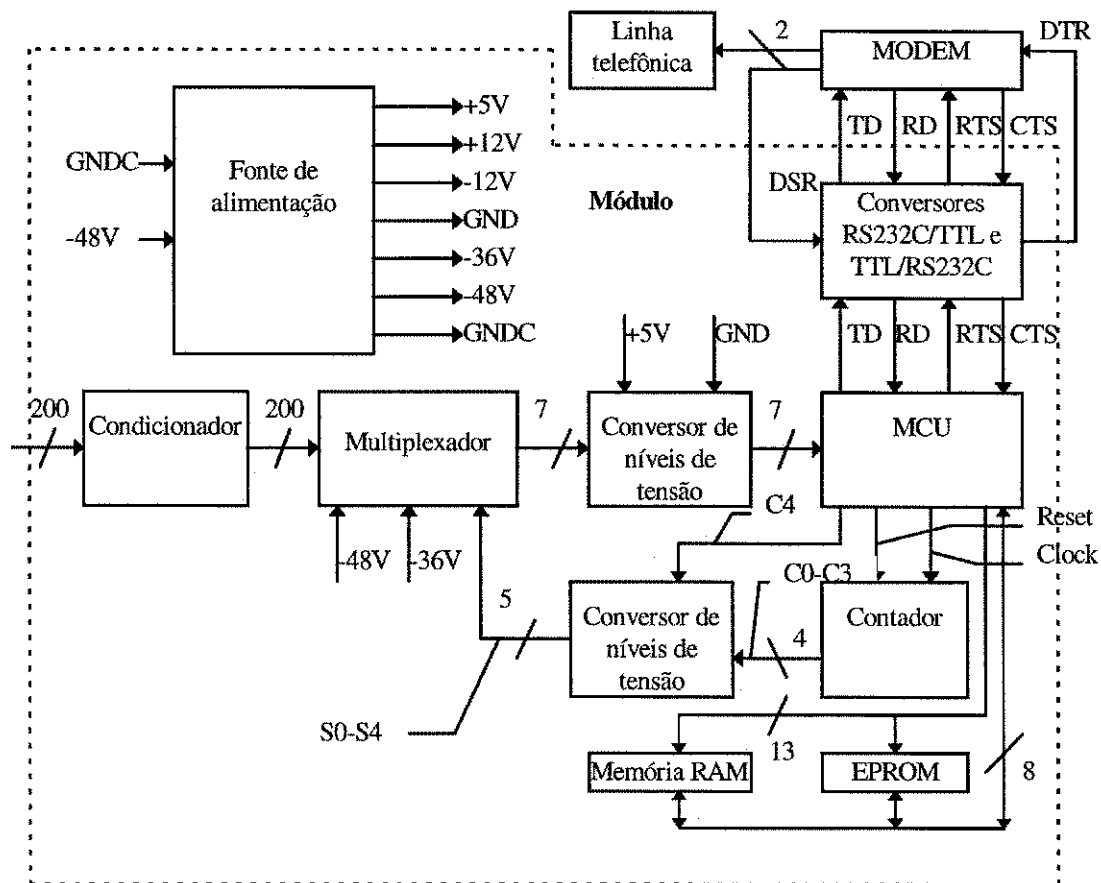


Figura 3.2 - Diagrama de blocos do módulo de transmissão e aquisição de dados

Tabela V - Sinais básicos para interfaceamento com o modem

Sinal	Fonte	Função
RTS	terminal de dados	requisitar permissão para enviar dados
CTS	modem	responder ao sinal de RTS, informando que o modem está pronto para receber os dados
DSR	modem	sinalizar que o modem está ligado
DTR	terminal de dados	sinalizar que o terminal de dados está ligado
TD	terminal de dados	dado a ser transmitido
RD	modem	dado recebido
GND	-	referência de terra para os demais sinais

3.1.3 - Microcontrolador Intel 8031 (MCU)

O dispositivo principal do módulo de aquisição e transmissão de dados é o microcontrolador Intel 8031. Como o detalhamento deste dispositivo foge ao objetivo deste trabalho, nesta seção apresenta-se apenas comentários acerca dos subsistemas do microcontrolador utilizados no desenvolvimento do módulo. Uma abordagem adicional sobre o dispositivo é encontrada no Apêndice A deste trabalho.

As portas 0 e 2 são utilizadas para endereçamento e leitura de memória. Sete linhas da porta P1 foram utilizadas para a entrada de dados que correspondem a pulsos de tarifação de TPs, ou seja, se a linha estiver em nível lógico baixo (0 volts) há a presença de um pulso de tarifação para um TP correspondente, caso contrário a linha permanece em nível alto (5 volts). A linha restante (P1.7) é utilizada para controle do modem, fornecendo o sinal RTS. Os pinos da porta 3 são todos empregados para funções diversas:

- P3.0 - entrada do sinal RD;
- P3.1 - saída do sinal TD;
- P3.2 - saída de reinicialização do contador;
- P3.3 - entrada do sinal CTS, proveniente do modem;
- P3.4 - saída do sinal de relógio para o contador de seleção do multiplexador;
- P3.5 - saída do *bit* mais significativo do contador de seleção do multiplexador;
- P3.6 - saída do sinal de leitura de memória RAM e;
- P3.7 - saída do sinal de escrita em memória RAM.

3.1.4 - Memória de Dados (RAM)

A memória de dados tem a finalidade de armazenar a contagem dos pulsos de tarifação dos 200 TPs além dos dados necessários ao processamento pelo programa

básico. Como cada cofre de TP pode armazenar em torno de 1700 fichas, utilizou-se um contador de dois *bytes* para cada TP totalizando 400 *bytes* de contadores. Além disto, cerca de 100 *bytes* foram usados para propósitos gerais. Assim escolheu-se um CI de 2 *kbytes* de RAM, o CI 65116 [12], por possuir capacidade de armazenamento superior à necessária e ser encontrado no comércio local com facilidade.

Esta unidade é ativada pelos sinais de leitura e escrita da memória de dados (\overline{RD} e \overline{WR}) provenientes do microcontrolador. A palavra de endereçamento desta unidade é formada por 11 *bits*, sendo os 3 *bits* mais significativos (A8-A10) fornecidos pela porta P2 do microcontrolador e os 8 *bits* menos significativos (A0-A7) pela porta P0 multiplexados no tempo com 8 *bits* de dados (D0-D7). A demultiplexação de endereços e dados é feita com o auxílio de um circuito integrado de trava (*latch*), o CI 74LS373 [10].

3.1.5 - Memória de Programa (EPROM)

O programa básico foi armazenado em um CI de memória EPROM de 8 *kbytes*, o 27C64 [11], pela facilidade de obtenção no comércio local, apesar deste programa ocupar menos de um *kbyte* de memória.

Esta memória é ativada através da linha \overline{PSEN} do microcontrolador. Seu endereçamento é feito de forma semelhante ao da memória de dados, sendo que, para esta unidade, estão disponíveis 13 *bits* para endereçamento, distribuídos em 5 *bits* mais significativos e 8 *bits* menos significativos provenientes das portas P2 e P0 do microcontrolador, respectivamente.

3.1.6 - Condicionador

Como já foi discutido em seções anteriores, a inversão de polaridade dos terminais de um TP corresponde a um pulso de terra (GNDC) com duração de 150 ms \pm

30 ms na *Linha B* do telefone. Para monitorar 200 desses pulsos, o módulo utiliza a unidade de multiplexação, que é alimentada por tensões de -36 e -48 volts. Desta forma, sinais na faixa de -48 a -36 volts podem ser conectados diretamente à unidade.

Para tornar os pulsos de terra compatíveis com a unidade de multiplexação, utilizou-se um condicionador de sinais, conforme o diagrama de blocos da figura 3.3. O condicionador rejeita o sinal de toque de campainha de 25 Hz e 75 V_{pp} e converte o pulso de terra em um pulso com amplitude de 1 volt em relação à tensão de -48 volts, podendo ser conectado diretamente à entrada da unidade de multiplexação.

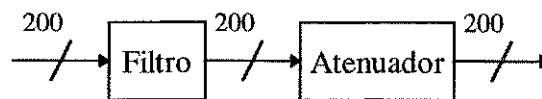


Figura 3.3 - Diagrama de blocos do condicionador de sinais

3.1.7 - Multiplexador

A unidade de multiplexação é formada por um arranjo de 13 multiplexadores *CMOS* de 16 entradas agrupados de tal forma a se obter um multiplexador de 208 entradas para 6 ou 7 saídas, conforme a palavra de seleção (figura 3.4).

Os multiplexadores são numerados de 0 a 12 e distribuídos em 7 multiplexadores pares (M0, M2,..., M12) e 6 ímpares (M1, M3,..., M11). Os multiplexadores pares são selecionados pela palavra de seleção (CS0-CS4) de 00000B a 01111B, ou seja de 0 a 15, resultando em 7 entradas selecionadas para cada um dos valores da palavra de seleção, ou 112 entradas selecionadas pelo grupo par de multiplexadores (tabela VI).

O grupo ímpar de multiplexadores (tabela VII) é selecionado para (CS0-CS4) de 10000B a 11111B, ou seja de 16 a 31, selecionando-se então 6 entradas, para cada valor da palavra. Desta forma, o grupo ímpar seleciona um total de 96 entradas. Somando-se os números de entradas dos grupos ímpar e par, totaliza-se 208. Entretanto,

as 6 últimas entradas dos multiplexadores ímpares não são utilizadas como também as duas penúltimas dos multiplexadores M9 e M11, o que reduz o valor de entradas seleccionáveis para 200 e torna desnecessária a palavra de seleção de valor 11111B.

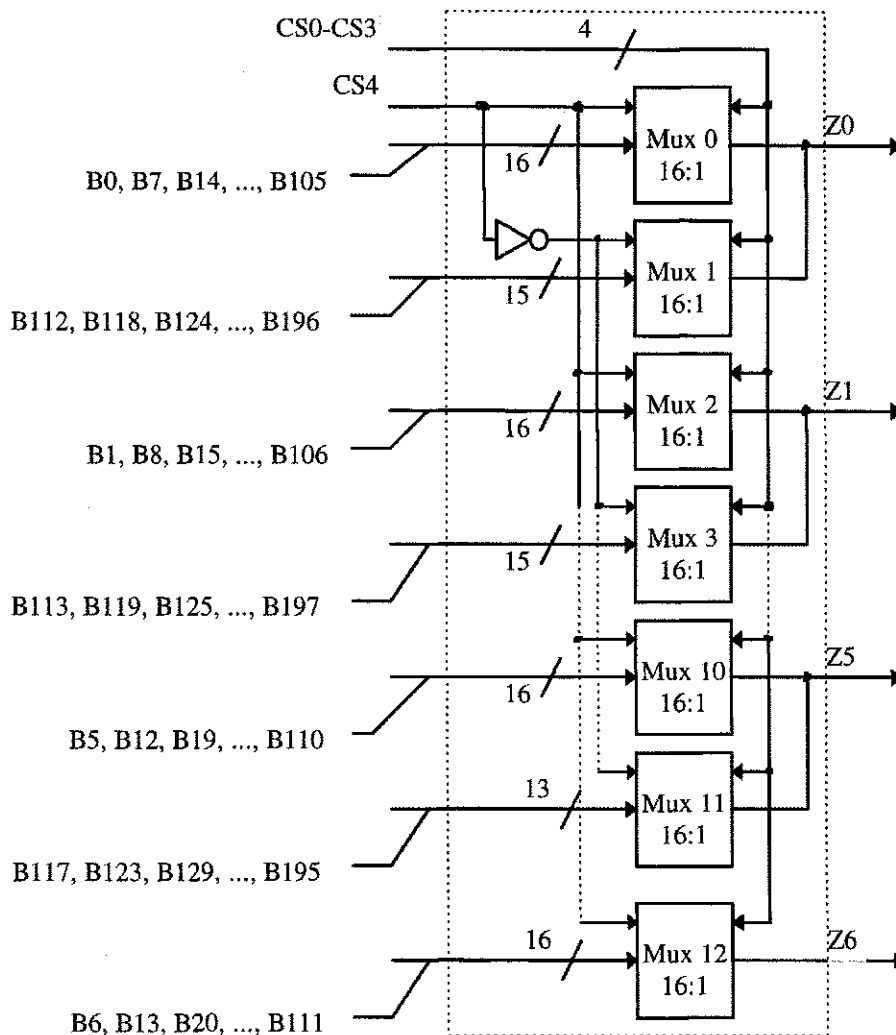


Figura 3.4 - Diagrama de blocos da unidade de multiplexação

Para proceder a monitoração dos 200 TPs, o sistema faz a seleção na unidade de multiplexação através da palavra de seleção e lê as 7 ou 6 linhas seleccionadas. Caso a leitura represente uma inversão de polaridade, o conteúdo do contador do TP correspondente à entrada é incrementado em uma unidade, passando-se a seguir, para a próxima linha, sucessivamente até ser reinicializado um novo ciclo de monitoração.

Tabela VI - Entradas ativas para os multiplexadores pares em função da seleção

Multiplexador/ Seleção	M0	M2	M4	M6	M8	M10	M12
00000B (00)	000	001	002	003	004	005	006
00001B (01)	007	008	009	010	011	012	013
00010B (02)	014	015	017	018	018	019	020
00011B (03)	021	022	023	024	025	026	027
00100B (04)	028	029	030	031	032	033	034
00101B (05)	035	036	037	038	039	040	041
00110B (06)	042	043	044	045	046	047	048
00111B (07)	049	050	051	052	053	054	055
01000B (08)	056	057	058	059	060	061	062
01001B (09)	063	064	065	066	067	068	069
01010B (10)	070	071	072	073	074	075	076
01011B (11)	077	078	079	080	081	082	083
01100B (12)	084	085	086	087	088	089	090
01101B (13)	091	092	093	094	095	096	097
01110B (14)	098	099	100	101	102	103	104
01111B (15)	105	106	107	108	109	110	111

Tabela VII - Entradas ativas para os multiplexadores ímpares em função da seleção

Multiplexador/ Seleção	M1	M3	M5	M7	M9	M11
10000B (16)	112	113	114	115	116	117
10001B (17)	118	119	120	121	122	123
10010B (18)	124	125	126	127	128	129
10011B (19)	130	131	132	133	134	135
10100B (20)	136	137	138	139	140	141
10101B (21)	142	143	144	145	146	147
10110B (22)	148	149	150	151	152	153
10111B (23)	154	155	156	157	158	159
11000B (24)	160	161	162	163	164	165
11001B (25)	166	167	168	169	170	171
11010B (26)	172	173	174	175	176	177
11011B (27)	178	179	180	181	182	183
11100B (28)	184	185	186	187	188	189
11101B (29)	190	191	192	193	194	195
11110B (30)	196	197	198	199	X	X
11111B (31)	X	X	X	X	X	X

X - entrada não utilizada

3.1.8 - Contador

A unidade de contagem é constituída por um contador binário de 4 *bits* cuja função é fornecer parte da palavra de seleção da unidade de multiplexação (C0-C3). A utilização desta unidade permitiu o uso de apenas 3 linhas do microcontrolador para realizar a seleção do multiplexador, ou seja, duas linhas fornecem o sinal de relógio (*clock*) e de zeramento (*reset*) do contador e a terceira o *bit* mais significativo (C4) da seleção da unidade de multiplexação. Com isto economiza-se duas linhas do microcontrolador, as quais são usadas para outros propósitos. Uma vez que as linhas (C0-C4) são compatíveis com níveis TTL, utilizou-se conversores de níveis de tensão para se obter a palavra de seleção dos multiplexadores (CS0-CS4).

3.1.9 - Conversores de Nível de Tensão

Os conversores de tensão são blocos funcionais, mostrados no diagrama de blocos da figura 3.5, que convertem os níveis de tensão de -48 e -36 volts para os níveis de GND e 5 volts respectivamente e vice-versa. Desta forma os multiplexadores podem ser interligados com os demais circuitos do módulo, compatíveis com os níveis lógicos TTL.

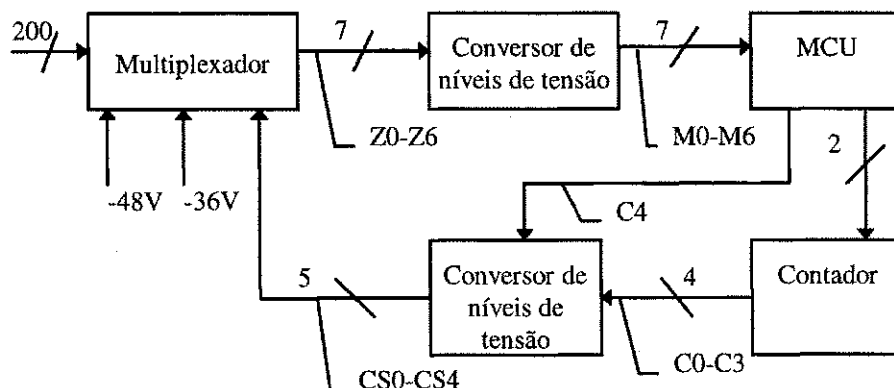


Figura 3.5 - Detalhe do diagrama de blocos do módulo

Estas unidades transformam os sinais de seleção dos multiplexadores (C0-C4), provenientes do contador e do microcontrolador, em níveis de seleção de -36 e -48 volts (CS0-CS4) e os sinais de saída dos multiplexadores (Z0-Z6) para o padrão TTL (M0-M6). Essas conversões são realizadas usando-se acopladores ópticos [16].

3.2. - Programa de Controle

Cada módulo de aquisição e transmissão de dados possui um programa básico sendo executado por seu microcontrolador. Este programa realiza as funções básicas de inicialização, aquisição, processamento e transmissão de dados, conforme o fluxograma geral da figura 3.6. A transmissão de dados é feita pela rotina de tratamento de interrupções da serial, de forma que o programa executa a aquisição e o processamento de dados de forma cíclica até ocorrer um pedido de interrupção do canal serial provocado pela chegada de um caractere pelo canal.

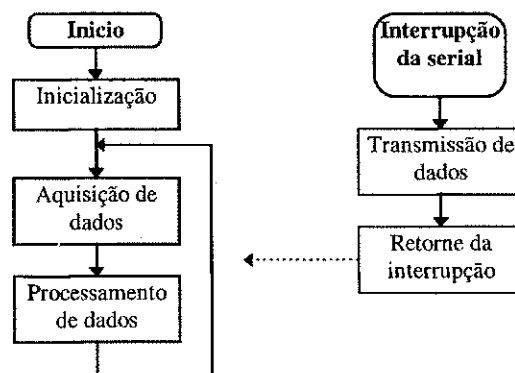


Figura 3.6 - Fluxograma geral do programa básico

A inicialização do módulo é feita pelo acionamento da chaves de alimentação ou de *reset*, dando origem as seguintes tarefas:

1. programação dos vetores das interrupções;
2. configuração dos parâmetros padrões da comunicação serial;
3. habilitação das portas de entrada e saída de dados;
4. inicialização dos contadores dos TPs e RAM de dados e;

5. habilitação da interrupção do canal serial.

Após a inicialização, todos os contadores dos TPs estão zerados e segue-se o laço de aquisição e processamento de dados.

A aquisição é feita selecionando-se a unidade de multiplexação e lendo-se, através da porta P1 do microcontrolador, 7 ou 6 saídas dos conversores de níveis de tensão, sendo que uma saída de nível lógico 0 equivale a uma inversão de polaridade na *Linha B* (queda de ficha) do TP correspondente.

A inversão de polaridade tem duração de $150 \text{ ms} \pm 30 \text{ ms}$. Considerando-se o pior caso, tem-se uma duração de 120 ms, então o microcontrolador deverá fazer a aquisição do sinal na *Linha B* de cada TP pelo menos a cada 120 ms de forma a garantir que a monitoração do TP seja feita sem perda de informação.

Por outro lado, considerando-se que o microcontrolador consegue monitorar cada TP a períodos muito inferiores a 120 ms, a inversão de polaridade de um mesmo TP será observada várias vezes antes de seu término. Desta forma, a rotina de aquisição de dados deve considerar apenas a transição do sinal de inversão, ou seja, em termos do sinal de saída dos conversores, uma transição de nível lógico alto (5 volts) para baixo (GND). Assim, o programa básico deverá guardar o estado anterior do sinal e compará-lo com o estado atual. Caso ocorra uma transição, o contador correspondente ao TP será incrementado de 1. A tabela da verdade com a variável *ATUALIZA*, que representa a necessidade de se incrementar um determinado contador, em função dos estados *ANTERIOR* e *ATUAL* do sinal é apresentada na tabela VIII.

Tabela VIII - Tabela da verdade para atualização dos contadores

ANTERIOR	ATUAL	ATUALIZA
0 (GND)	0 (GND)	0
0 (GND)	1 (5 volts)	0
1 (5 volts)	0 (GND)	1
1 (5 volts)	1 (5 volts)	0

Da tabela VIII, tem-se que:

$$ATUALIZA = ANTERIOR \cdot \overline{ATUAL} \quad (1)$$

Usando-se o teorema de De Morgan, tem-se de (1):

$$\overline{ATUALIZA} = \overline{ANTERIOR} + ATUAL \quad (2)$$

A implementação de (2) é feita pela rotina de aquisição e processamento de dados, cujo fluxograma é mostrado na figura 3.7.

Como mostrado anteriormente, a palavra de seleção dos multiplexadores pode ter 31 valores possíveis. Para cada seleção, o microcontrolador aguarda 400 μ s para que o sinal de seleção se estabilize, de forma a ser garantida uma leitura confiável. Para cada seleção, dependendo do estado do *bit* de seleção CS4, o microcontrolador faz a análise de 6 ou 7 *bits*, sendo que cada *bit* corresponde ao estado de um TP. A análise resulta em uma transição válida (descida) ou não do sinal correspondente à inversão de polaridade. Caso seja uma transição válida, são feitas 15 amostras da mesma linha, a intervalos de 7,6 μ s, desconsiderando-se então a inversão se o nível do sinal voltar ao nível alto durante a amostragem. Caso a inversão persista durante a amostragem, ela será confirmada no próximo ciclo de leitura da linha, ou seja, uma inversão só é válida se a mesma persistir durante dois ciclos de amostragem. Esta estratégia de amostragem permitiu filtrar ruídos aleatórios presentes na linha telefônica.

O intervalo entre ciclos de amostragem de uma mesma linha varia dentro da faixa de 18,8 ms a 48,2 ms, o que garante, mesmo no pior caso, uma amostragem sem perda de informação de todas as 200 linhas de interesse, uma vez que uma inversão de polaridade tem duração mínima de 120 ms.

A rotina de aquisição e processamento permanece em laço até ocorrer um pedido de interrupção do canal serial do microcontrolador. Esta interrupção é gerada quando chega um caractere pelo canal [17], indicando um pedido possível de transferência de dados entre o microcomputador e o módulo. Após o atendimento do

pedido de interrupção, o fluxo do programa volta ao controle da rotina de aquisição e processamento.

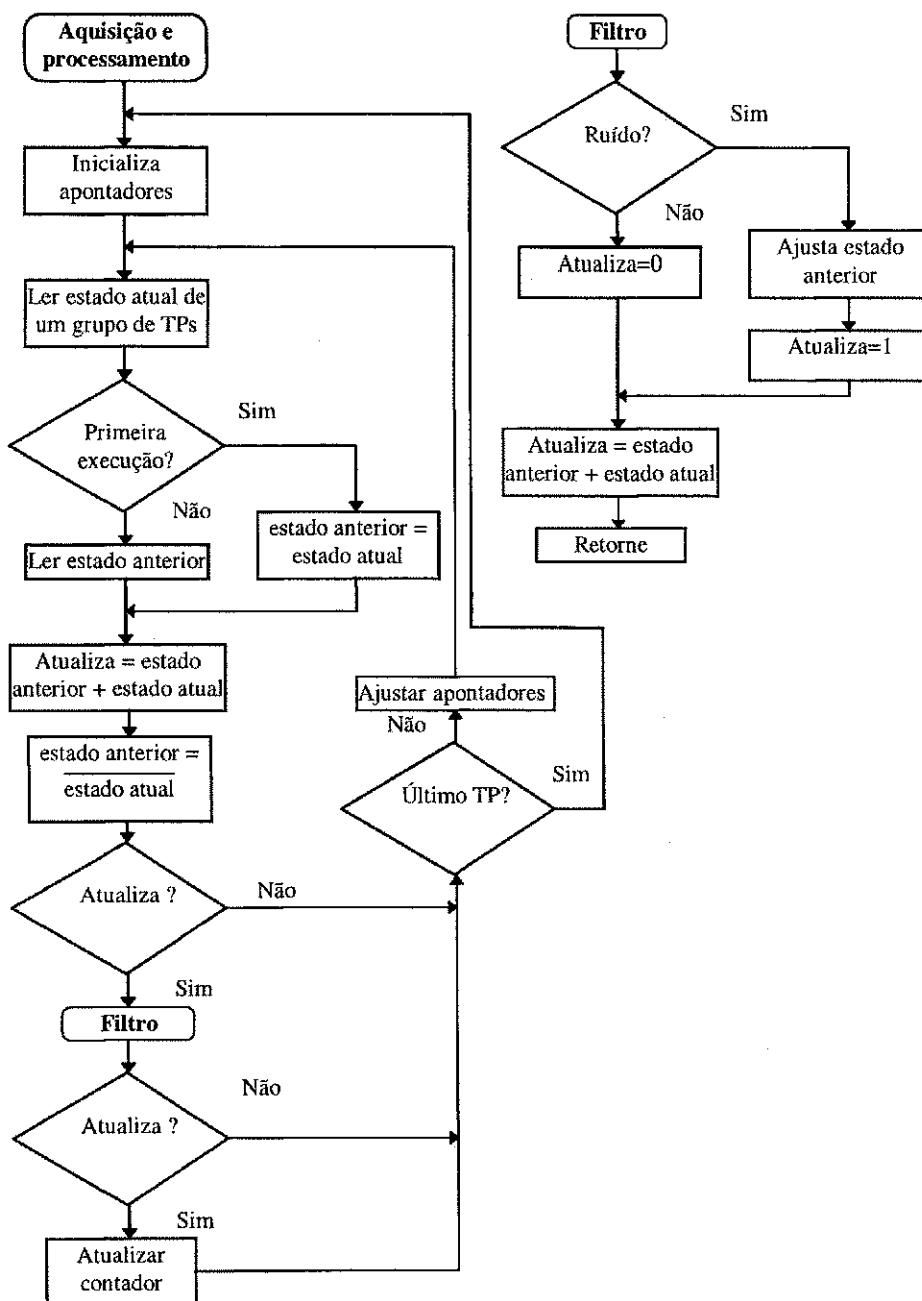


Figura 3.7 - Fluxograma da rotina de aquisição e processamento de dados

A rotina de interrupção da porta serial, cujo fluxograma é mostrado na figura 3.8, executa as funções básicas de transferir o conteúdo da memória RAM de dados referente à contagem atual dos 200 TPs, ou reinicializar o contador de um determinado

TP. Qualquer uma das funções pode ser requisitada pelo microcomputador através do envio de um dos caracteres de comando disponíveis, a saber: 0DCH (*LEITURA*) para leitura de dados ou 0E6H (*ESCRITA*) para reinicialização de um contador.

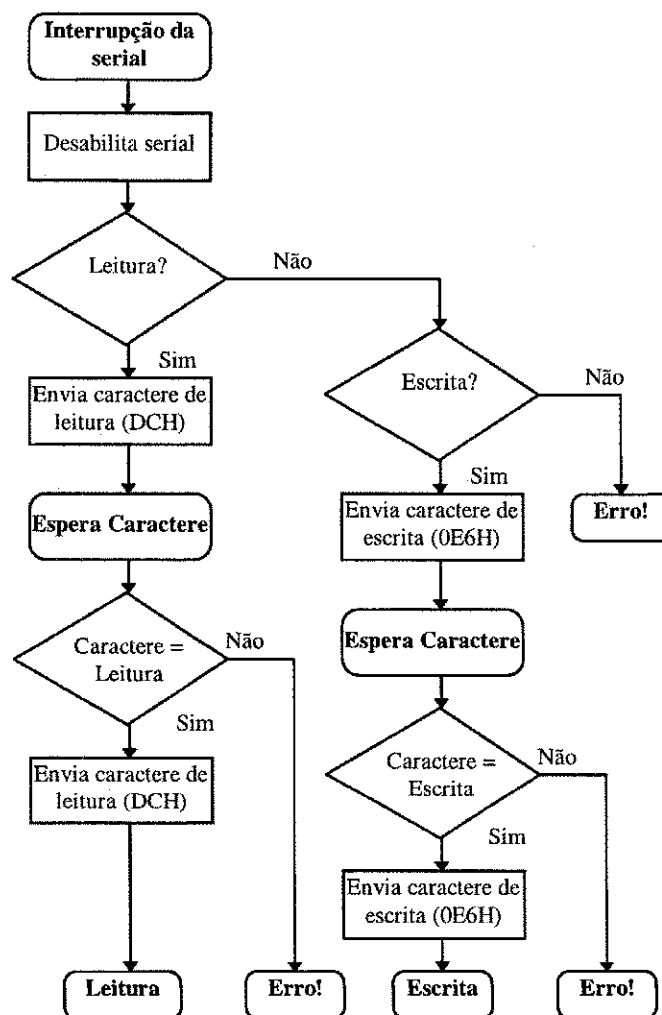


Figura 3.8 - Fluxograma da rotina de tratamento da interrupção da serial

A leitura da RAM de dados com os conteúdos dos contadores dos TPs é feita transferindo-se 10 blocos de 40 *bytes* pela interface serial, conforme mostrado pelo fluxograma da figura 3.9. Ao final de cada bloco, o microcontrolador aguarda que o microcomputador informe se algum *byte* do último bloco apresentou erro de paridade, através do caractere de sinalização de erro 0F0H ou se não houve erros, por meio do caractere 0E1H, que significa “prossiga com a transferência do próximo bloco”. Caso seja recebido um caractere de erro, o último bloco é retransmitido e espera-se uma nova

confirmação. O bloco pode ser retransmitido até cinco vezes, mas se o erro persistir, a transmissão é abandonada. Após o envio do último *byte* do último bloco, o microcontrolador transmite o caractere 0FAH, significando “fim de transmissão” e retorna à rotina de aquisição e processamento de dados.

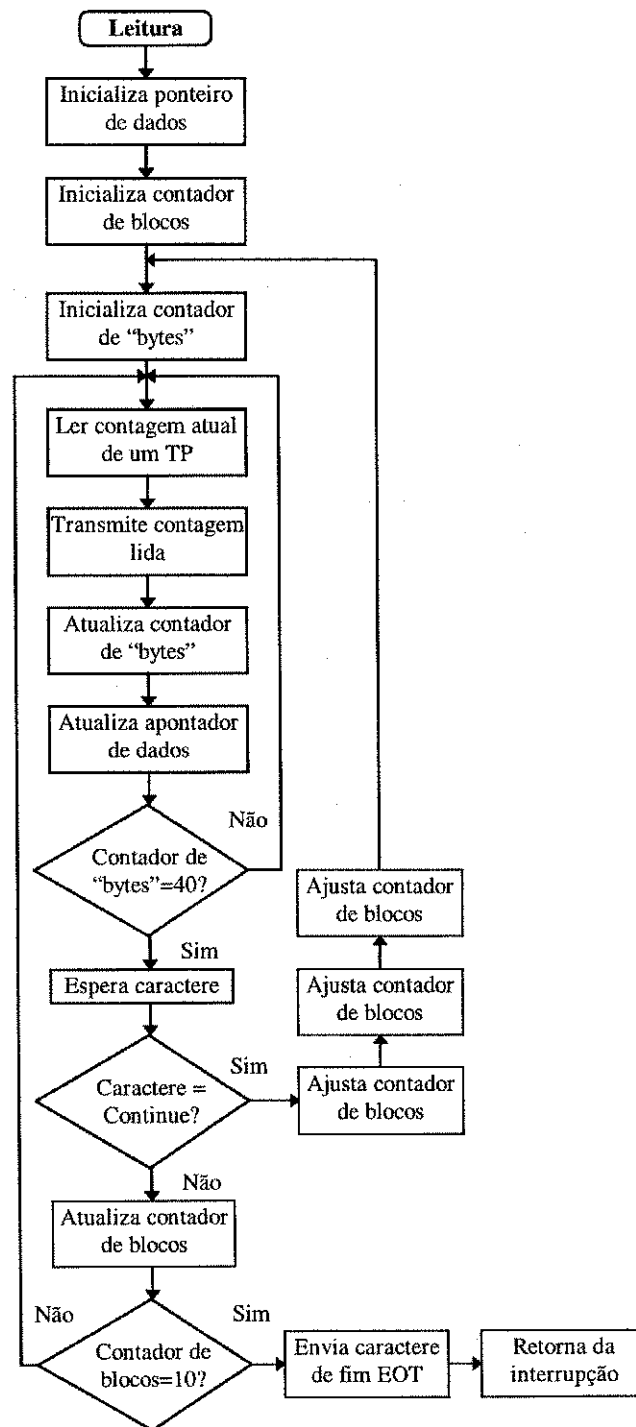


Figura 3.9 - Fluxograma da rotina de leitura dos contadores

A inicialização de um contador é feita enviando-se o comando de escrita para o microcontrolador. Após a confirmação do comando, o microcontrolador aguarda que seja enviada a posição do TP no módulo. Ao receber tal posição, o microcontrolador calcula os endereços correspondentes ao contador do TP e coloca seus conteúdos em zero, conforme o fluxograma da figura 3.10.

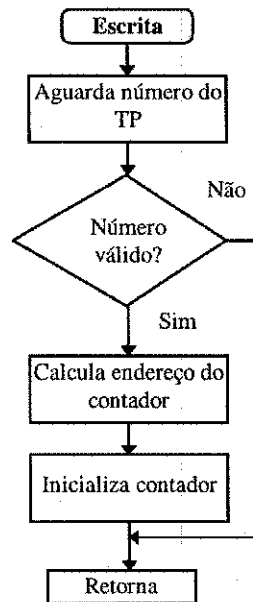


Figura 3.10 - Fluxograma da rotina de inicialização de um contador

3.3 - Programa de Análise dos Dados

O programa de análise de dados, chamado de *SADAM-TP*, permite ao usuário uma supervisão dos TPs conectados aos diversos módulos de aquisição e transmissão de dados do sistema proposto. Para isto, definiu-se 6 serviços básicos:

1. *ARQUIVOS*;
2. *COMUNICAÇÃO*;
3. *MODEM*;
4. *DOS*;
5. *ESTATÍSTICAS* e;
6. *SAI*

As telas da interface homem-máquina são semelhantes às apresentadas pelo programa de controle da placa de desenvolvimento, dispensando maiores informações acerca da navegação pelo programa. Cada um dos elementos de uma tela típica do *SADAM-TP* é mostrado na figura 3.11.

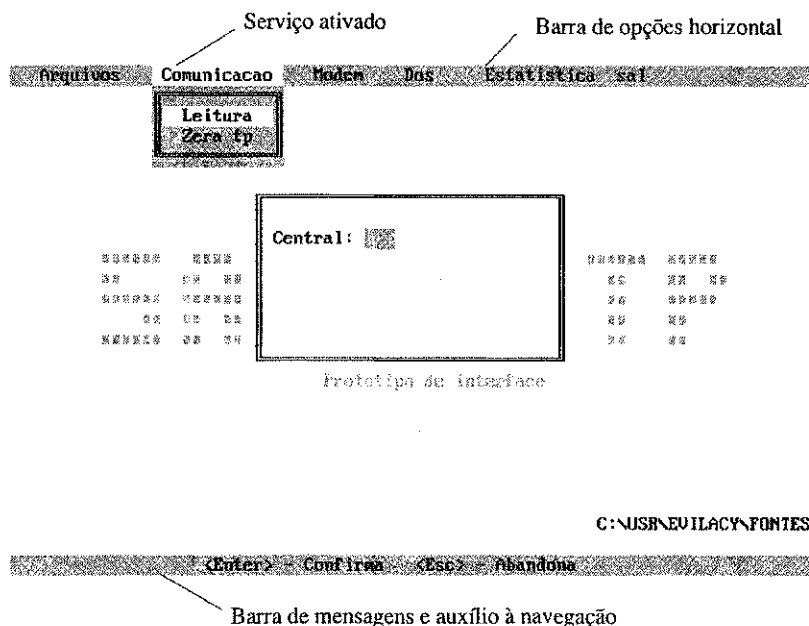


Figura 3.11 - Exemplo de tela do *SADAM-TP*

O serviço *ARQUIVOS*, tem a função de manipular os arquivos de TPs e Centrais, no que diz respeito às funções ou subserviços de: incluir, editar, excluir e alterar registros destes arquivos. Uma tela deste serviço é mostrada na figura 3.12.

Os registros destes arquivos contêm as informações necessárias para a supervisão dos telefones bem como dados cadastrais das centrais telefônicas. Para os TPs, são armazenados o número de sua linha telefônica, endereço, localização no módulo, tipo de tarifação utilizada (a cartão ou a ficha), o resultado e a data da última contagem além da contagem acumulada no período de um mês. Para as centrais são armazenados o seu prefixo, o número da linha do modem ao qual está conectado seu módulo de aquisição de dados, os parâmetros da comunicação serial, o endereço e o número de TPs conectados.

O próximo serviço é o **COMUNICAÇÃO** com os subserviços de fazer a leitura dos contadores em uma determinada central e de reinicialização da contagem de um determinado telefone, conforme mostrado na figura 3.13.

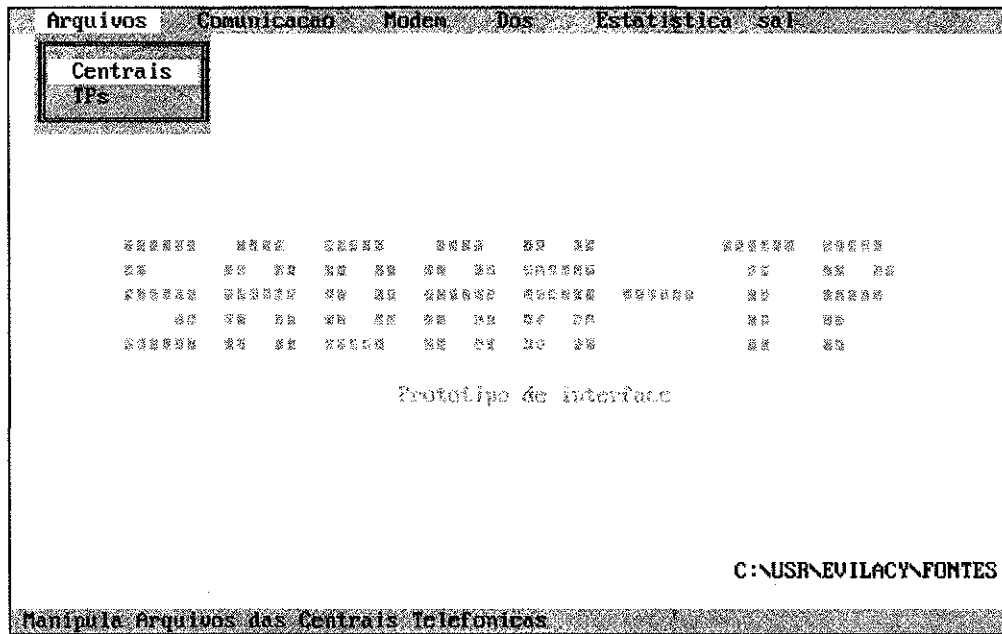


Figura 3.12 - Exemplo de tela do serviço **ARQUIVOS**

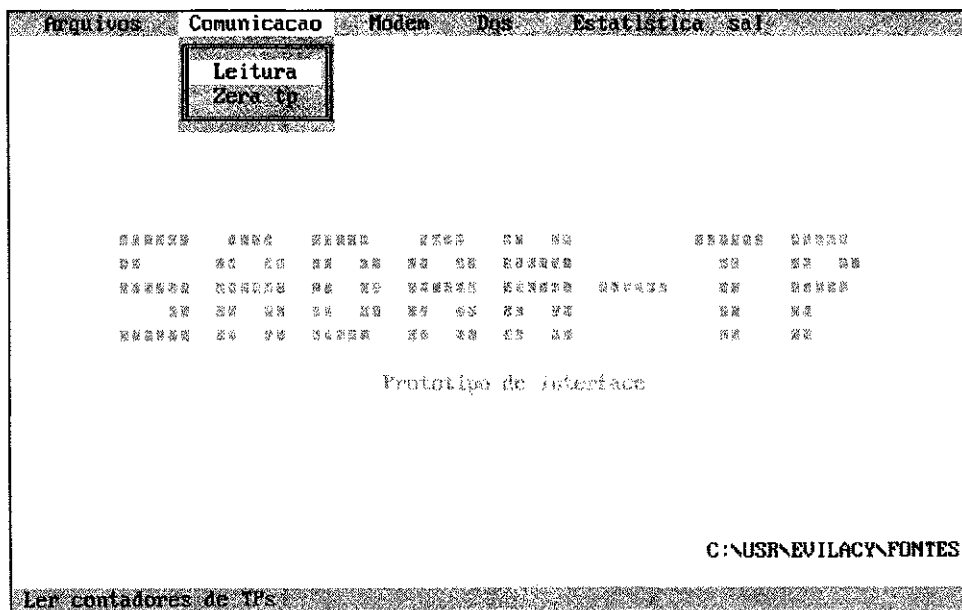


Figura 3.13 - Exemplo de tela para o serviço **COMUNICAÇÃO**

Para proceder a leitura, o *SADAM-TP* solicita ao usuário o prefixo da central onde está o módulo de aquisição e transmissão de dados cujo conteúdo será lido. De posse do prefixo, o programa busca em um arquivo de centrais o número da linha telefônica que está conectada ao módulo, realiza a discagem do número, recebe o conteúdo dos contadores de todos os telefones conectados ao módulo de interesse e atualiza o arquivo de TPs correspondente.

Para que a comunicação seja feita com sucesso, é necessário fornecer ao *SADAM-TP* o número da porta ativa em que está conectado o modem, o que é feito através do serviço *MODEM*, cujo exemplo de tela é mostrado na figura 3.14.

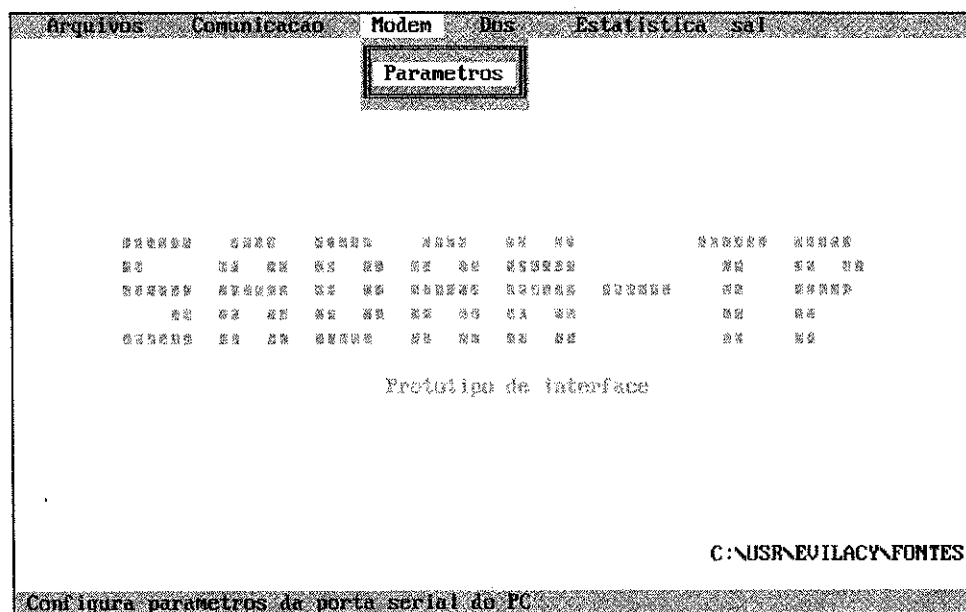


Figura 3.14 - Exemplo de tela para o serviço *MODEM*

O quarto serviço do programa permite o acesso temporário ao sistema operacional *DOS*, permitindo ao usuário executar aplicativos ou comandos fora do ambiente *SADAM-TP* e depois retornar ao sistema. A tela para este serviço é apresentada na figura 3.15.

O serviço *ESTATÍSTICA* permite que se faça uma análise histórica dos dados coletados pelo *SADAM-TP*, determinando os telefones a ficha com cofres cheios, os TPs

defeituosos e o tempo médio entre coletas de um grupo de telefones de interesse, conforme mostrado na figura 3.16.

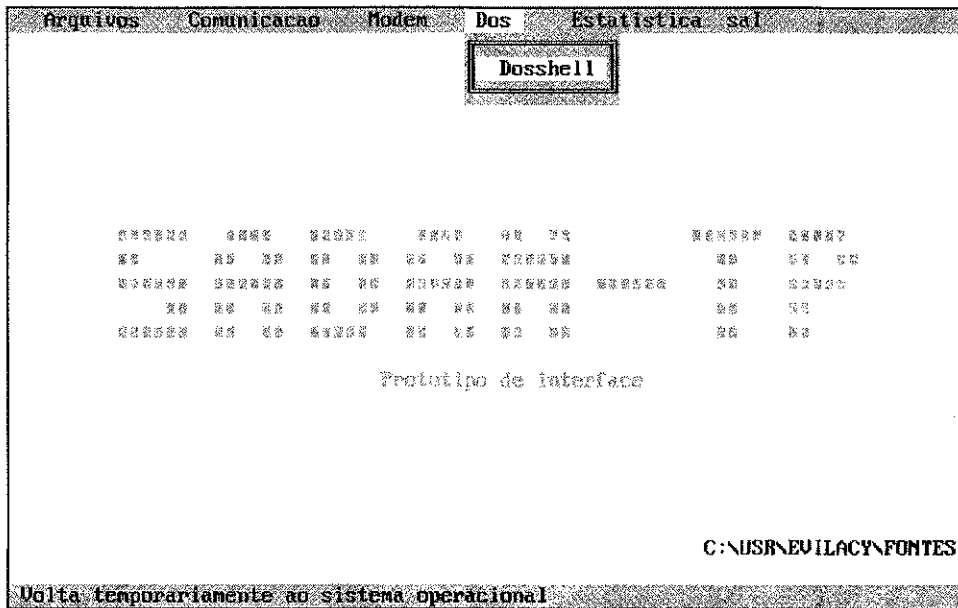


Figura 3.15 - Exemplo de tela para o serviço *DOS*

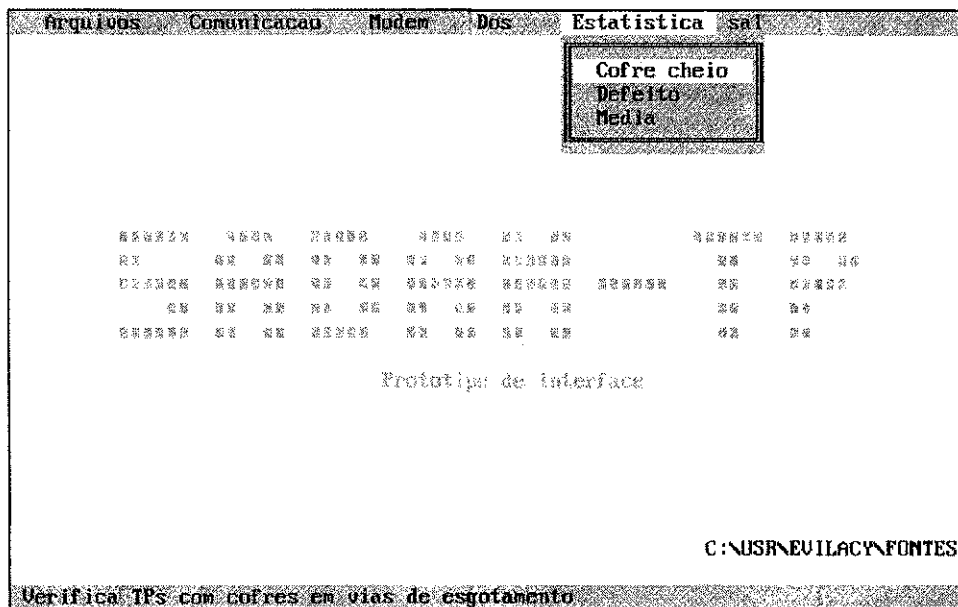
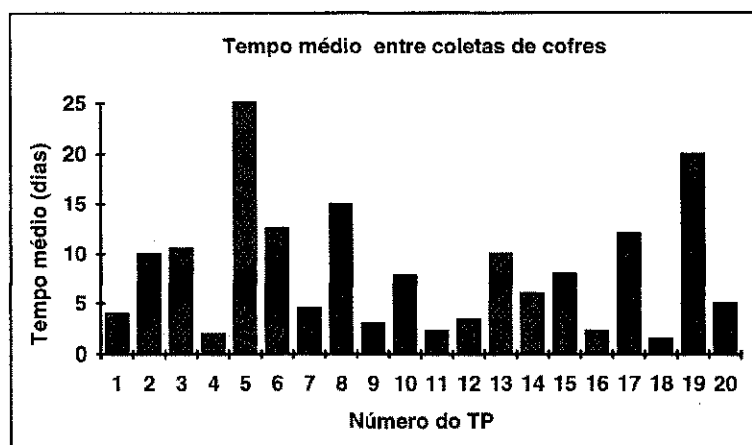
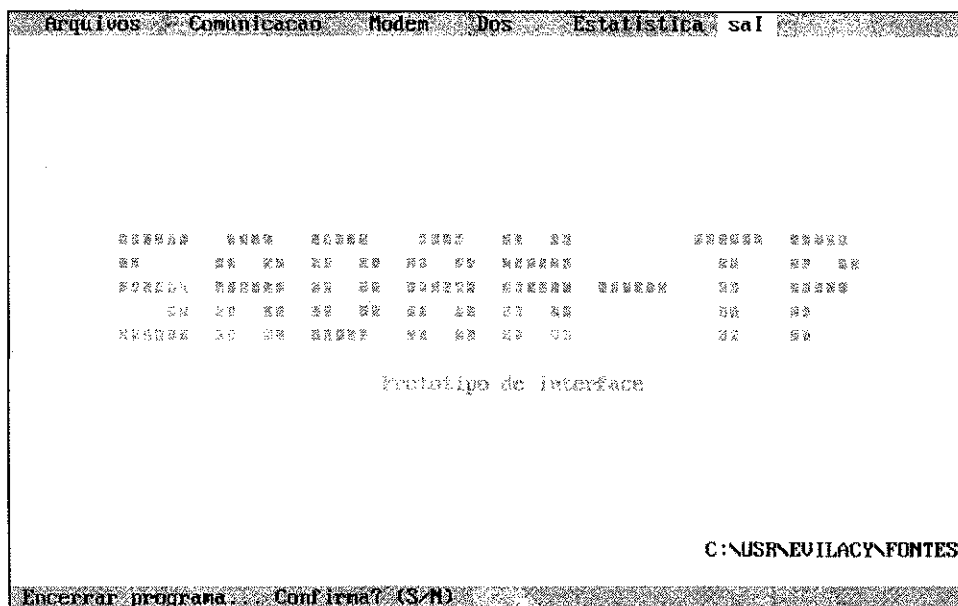


Figura 3.16 - Exemplo de tela do serviço *ESTATÍSTICA*

A figura 3.17 mostra o exemplo de uma tela para o subserviço *MÉDIA*, que informa a média de dias entre a coleta e o enchimento de cofre de TPs.

Figura 1.17 - Exemplo de tela para o subserviço *MÉDIA*

Finalmente, o serviço *SAI*, apresentado na figura 3.18, permite ao usuário abandonar o *SADAM-TP*. Para isto, a mensagem *Encerrar programa... Confirma? (S/N)* é apresentada na linha de mensagens e a confirmação do usuário é aguardada até ser pressionada as teclas *S* ou *N*.

Figura 3.18 - Exemplo de tela para o serviço *SAI*

No próximo capítulo são apresentadas considerações sobre os testes do sistema e resultados obtidos.

TESTES E RESULTADOS EXPERIMENTAIS

Os testes do sistema proposto foram feitos utilizando-se o ambiente de desenvolvimento para o 8031, uma ferramenta de simulação do microcontrolador Intel 8031, um protótipo para monitoração de 80 TPs além de um protótipo do programa *SADAM-TP* com as funções básicas de aquisição de dados via interface serial. Neste capítulo apresenta-se a metodologia de testes do sistema e os resultados obtidos.

Em uma primeira etapa, foram realizados os testes do programa, simulando a execução de suas rotinas individualmente através de uma ferramenta de *software* para simulação do microcontrolador 8031. Essa ferramenta avaliou as condições de operação do programa principalmente em relação aos tempos de execução das rotinas e da aquisição de dados.

Com isto obteve-se os períodos de tempo máximo e mínimo entre duas amostragens do sinal de inversão de um mesmo TP, considerando-se todas as condições do processamento. No pior caso, ou seja, as monitorações de todas as entradas do multiplexador acusam inversão de polaridade, o período foi de 48,2 ms e; para o melhor caso, ou seja, apenas um telefone apresenta inversão de polaridade, o período resultou em 18,8 ms. Como a inversão de polaridade tem duração de 120 ms, garante-se, nos dois casos, que a amostragem é feita sem perda de informação.

Após a simulação do programa monitor, usou-se a placa de desenvolvimento para a avaliação de desempenho e depuração do programa em condições reais de processamento, com o que se chegou à versão do programa apresentada no apêndice C deste trabalho.

Para os testes de *hardware*, foi montado um protótipo do módulo de aquisição e transmissão de dados com capacidade de monitoração de 80 TPs. A unidade de multiplexação foi montada utilizando-se 5 multiplexadores de 16 entradas e uma saída, arranjados de forma semelhante ao descrito na seção 3.1.7. Com o objetivo de se obter condições semelhantes de monitoração da configuração completa do sistema, foi usada uma palavra de seleção dos multiplexadores de 00 a 31, varrendo-se assim todos os 200 TPs, mesmo tendo-se apenas 80 entradas disponíveis.

Em uma primeira etapa, o protótipo foi testado em laboratório. Para isto, utilizou-se duas fontes de alimentação simétricas de ± 12 volts em série, obtendo-se com o arranjo uma fonte de 48 volts, que corresponde à alimentação da central telefônica, além de um gerador de pulsos de 48 volts de amplitude e 120 ms de duração, para simulação da inversão de polaridade nos TPs.

Após os testes em laboratório, prosseguiu-se a avaliação do sistema na central telefônica da UFPB, que possui um telefone público conectado ao seu *DG*. Este telefone foi conectado simultaneamente a 4 entradas do multiplexador, simulando-se a ocorrência simultânea de inversões em 4 TPs. Nesta avaliação, foram feitas ligações com o TP e observou-se que a contagem do módulo permaneceu igual ao número de fichas depositadas.

Por fim, testou-se a comunicação de dados, à taxa de 1200 bps via Modem, usando-se dois modem's de mesa, modelo Parks - UP1220C. Além disto, testou-se rotinas para uso de um modem de placa compatível com o IBM-PC.

Para o teste da comunicação, um programa executado no microcomputador envia um caractere e aguarda que a placa devolva o *complemento de um* do caractere recebido. A cada grupo de 1000 caracteres enviados, o programa exhibe o percentual de dados errados.

Numa primeira fase, utilizou-se dois ramais da central telefônica da universidade obtendo-se em torno de 4% de dados errados, provavelmente devido à presença de ruído na linha, escutado normalmente durante conversações. Posteriormente foram utilizadas duas linhas telefônicas das centrais 321 e 333, obtendo-se um erro inferior a 0,5 %. Considerando-se que o módulo envia 400 caracteres, distribuídos em 10 blocos de 40 caracteres, tem-se em média 2 caracteres errados, que equivale à possibilidade de se retransmitir um bloco até duas vezes, o que não constitui problema, uma vez que um determinado bloco pode ser retransmitido até 15 vezes.

Através desses testes, observou-se que os objetivos básicos do sistema, ou seja, a amostragem dos TPs e a transmissão da contagem do número de inversões, foram obtidos de forma satisfatória.

A seguir serão apresentadas as conclusões e sugestões para a continuidade do trabalho.

CONCLUSÕES

Apresentou-se neste trabalho um sistema de aquisição e transmissão de dados para supervisão de telefones públicos, baseado no microcontrolador Intel 8031. O sistema é subdividido em um ou mais módulos de aquisição de dados interligados a um microcomputador via modem, por meio de uma linha telefônica discada. No microcomputador é executado um programa, chamado de programa *SADAM-TP*, para a análise de dados. Foi ainda apresentado um ambiente de desenvolvimento para microcontroladores Intel 8031.

Foi montado um protótipo para monitoração de 80 TPs no intuito de se testar o desempenho do sistema. Os teste do programa básico do microcontrolador foi feito com auxílio de uma ferramenta de simulação e do ambiente de desenvolvimento.

Este trabalho tenta suprir uma lacuna no que diz respeito à supervisão dos telefones públicos de centrais eletromecânicas.

O sistema foi testado em laboratório e na central telefônica da universidade com resultados satisfatórios, haja visto que desempenhou suas funções básicas de aquisição e transmissão de dados de acordo com o esperado no trabalho. A aquisição de dados foi feita sem perda de informação para o pior caso, ou seja, todos os telefones apresentam inversão de polaridade simultaneamente enquanto que a comunicação entre

módulo e microcomputador foi realizada com uma margem de erro aceitável pelo sistema.

Como um trabalho científico que tem pretensões de resultar em um produto de mercado e melhorar a qualidade de um serviço prestado à população, seu desenvolvimento deverá prosseguir no sentido de construir um protótipo completo do sistema, a ser testado exaustivamente nas centrais eletromecânicas, desenvolver o desenho da placa de circuito impresso de seu *hardware*, criar uma interface de painel para o equipamento definitivo, além de se adaptar o programa *SADAM-TP* de forma a atender as necessidades das concessionárias de telecomunicações. Além disto, sugere-se para a continuação do trabalho:

-expandir o sistema para se fazer a monitoração dos pulsos de tarifação de todas as linhas telefônicas de uma central eletromecânica, de forma a automatizar o plano de tarifação das ligações locais;

-inserir no módulo de aquisição e transmissão de dados uma interface com a linha telefônica que permita abandonar o uso de um modem de mesa externo e;

-implantar no sistema um código corretor de erros de forma a evitar a retransmissão de blocos de dados no caso de erros de transmissão e aumentar a taxa de transferência de dados para valores acima de 1200 bps.

Microcontrolador Intel 8031

Os microcontroladores têm facilitado o desenvolvimento de sistemas digitais pois concentram, no mesmo circuito integrado da unidade central de processamento, portas de entrada e saída de dados, interfaces para comunicação de dados, memórias voláteis e não voláteis, conversores A/D e D/A, temporizadores e outros dispositivos necessários a muitos sistemas de monitoramento e controle. Pode-se dizer que o microcontrolador representa um microcomputador de especificações modestas integrado em uma única pastilha.

Várias famílias de microcontroladores estão disponíveis no mercado, com as mais diversas especificações. Os microcontroladores da família Intel 8031 são bastante utilizados na indústria devido ao seu custo baixo e facilidade de aquisição no mercado interno. Neste trabalho, escolheu-se a versão 80C31 da família devido aos fatores mencionados e ao seu consumo baixo de energia.

Este apêndice faz análise breve da arquitetura do microcontrolador, dando-se ênfase ao bloco da Interface para Comunicação Serial, com o objetivo de facilitar o entendimento do texto.

A.1 - Arquitetura do 8031

O 8031 é um microcontrolador Intel de 8 *bits* encapsulado em um circuito integrado de 40 pinos e capaz de operar com relógio de até 12 MHz. Sua arquitetura interna, mostrada na figura A.1, apresenta basicamente [17]:

- RAM de 128 *bytes*;
- 21 registradores de funções especiais (SRFs);
- 4 *kbytes* de ROM;
- 4 portas de entrada e saída de dados (E/S);
- 2 temporizadores/contadores de 16 bits;
- Interface para comunicação serial;
- 6 fontes de interrupções e;
- Unidade de Controle e Temporização.

A.1.1 - Memórias RAM e ROM

A memória RAM do microcontrolador está subdividida em blocos com finalidades específicas. O primeiro bloco, do endereço 000H a 01FH é composto por quatro bancos de oito registradores de 8 *bits*. A faixa de 020H a 02FH contém posições de memória cujos bits podem ser endereçados individualmente através de instruções específicas. O restante da memória, de 030H a 07FH também está disponível para o usuário, mas deve-se ter critério ao utilizá-la pois o ponteiro da pilha de dados (*Stack Pointer*) aponta, após o *RESET* do microcontrolador, para o primeiro endereço desta faixa.

O acesso à faixa *bit*-endereçável é feita através dos endereços de 000H (primeiro *bit* da posição de memória 020H), 001H (segundo *bit* da posição de 020H) até 07FH (oitavo *bit* da posição 02FH) ou ainda por meio de mnemônicos associando-se aos *bits* da posição 020H os endereços 020H.0, 020H.1,...,020H.7 e, assim, sucessivamente.

A memória ROM corresponde a um bloco de 4 *kbytes* de memória de programa gravada em fábrica com o conteúdo desejado pelo usuário. Caso esta memória não seja utilizada, a linha de seleção de endereçamento externo (\overline{EA}) deve ser colocada em nível de tensão baixo (0 Volts).

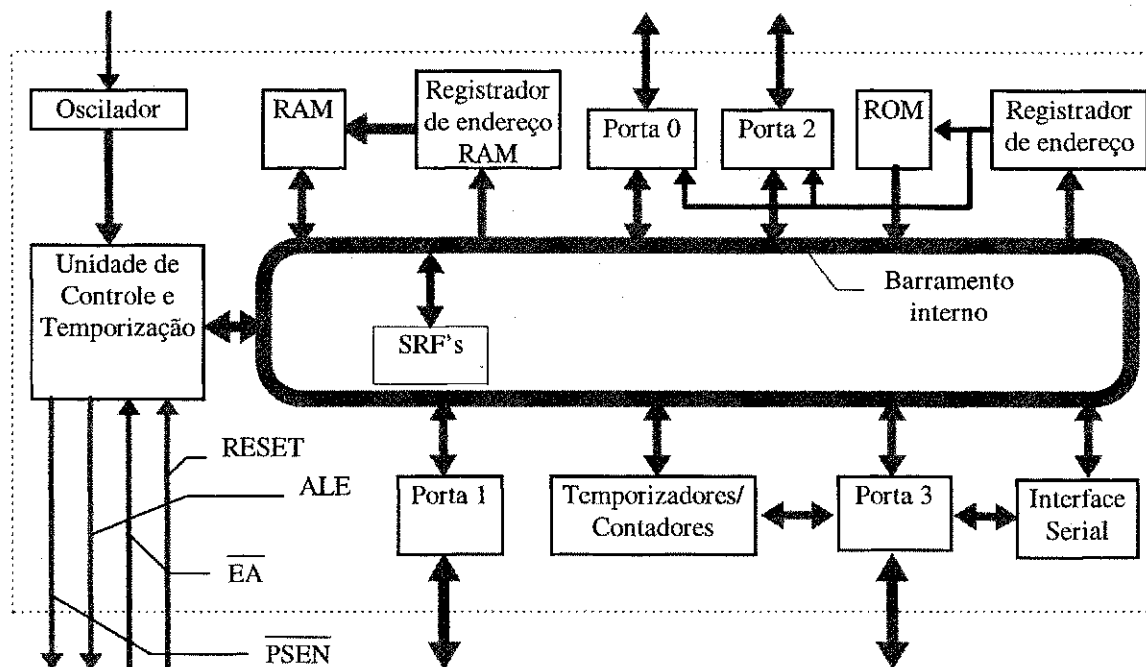


Figura A.1 - Diagrama de bloco básico do microcontrolador 8031

A.1.2 - Registradores de Funções Especiais

Os Registradores de Funções Especiais (SRFs) são posições de memória referentes aos registradores de dados temporários (*ACC e *B); *status* do programa (*PSW); ponteiros do sistema (SP, DPL e DPH); portas de E/S de dados (*P0, *P1, *P2 e *P3); controle das interrupções (*IP e *IE); controle dos temporizadores/contadores (TMOD, *TCON, TH0, TL0, TH1 e TL1); controle e acesso à porta serial (*SCON e SBUF); e de controle dos modos de operação de baixo consumo do microcontrolador (PCON) [17], conforme mostrado pela tabela A.I.

A.1.3 - Portas de Entrada e Saída de Dados

O 8031 possui 32 linhas de entrada e saída de dados (E/S) distribuídas em 4 portas de 8 bits, numeradas de 0 a 3.

As portas 0 e 2 do microcontrolador podem ser utilizadas para fins gerais, mas quando o acesso à memória externa estiver habilitado, o microcontrolador faz uso destas portas para endereçar até 64 *kbytes* de memória externa de dados ou de programas. Enquanto a porta 0 provê as 8 linhas de endereçamento menos significativas (A0-A7), multiplexadas com 8 linhas de dados (D0-D7), a porta 2 fornece as 8 linhas de endereçamento mais significativas (A8-A15).

A porta 1 é utilizada para fins gerais, enquanto que a porta 3 tem suas linhas associadas a funções especiais do microcontrolador, a saber:

- porta 3, linha 0 (P3.0) - Entrada serial de dados (RXD);
- porta 3, linha 1 (P3.1) - Saída serial de dados (TXD);
- porta 3, linha 2 (P3.2) - Entrada de interrupção externa n° 0 ($\overline{\text{INT0}}$);
- porta 3, linha 3 (P3.3) - Entrada de interrupção externa n° 1 ($\overline{\text{INT1}}$);
- porta 3, linha 4 (P3.4) - Entrada para o temporizador/contador n° 0 (T0);
- porta 3, linha 5 (P3.5) - Entrada para o temporizador/contador n° 1 (T1);
- porta 3, linha 6 (P3.6) - Seleção de leitura de memória ($\overline{\text{WR}}$) e;
- porta 3, linha 7 (P3.7) - Seleção de escrita em memória ($\overline{\text{RD}}$).

O acesso aos 64 *kbytes* de memória de dados bem como aos 64 *kbytes* de memória de programa, utiliza o mesmo barramento de endereços. Para isto, são utilizadas três linhas da porta 3, de forma que os sinais de habilitação das memórias são independentes, ou seja, a memória de dados é acessada através dos sinais de escrita ($\overline{\text{WR}}$) e leitura ($\overline{\text{RD}}$) enquanto que os programas são acessados pelo sinal de seleção de memória de programa ($\overline{\text{PSEN}}$).

Tabela A.I - Registradores de Funções Especiais

Símbolo	Nome	Endereço	Função
*ACC	Acumulador	0E0H	Armazena dados temporários
*B	Registrador B	0F0H	Armazena dados temporários
*PSW	Palavra de <i>status</i> do programa	0D0H	Sinaliza resultados de operações
SP	Ponteiro para pilha de dados	081H	Contém o endereço corrente da pilha de dados
DPL	Byte menos significativo do apontador de dados	082H	Contém a parte menos significativa do endereço corrente de um dado em memória
DPH	Byte mais significativo do apontador de dados	083H	Contém a parte mais significativa do endereço corrente de um dado
*P0	Porta 0	080H	Mapeia a Porta 0 em memória
*P1	Porta 1	090H	Mapeia a Porta 1 em memória
*P2	Porta 2	0A0H	Mapeia a Porta 2 em memória
*P3	Porta 3	0B0H	Mapeia a Porta 3 em memória
*IP	Controle de prioridade das interrupções	0B8H	Controla a ordem de prioridade das interrupções
*IE	Controle de habilitação das interrupções	0A8	Liga/desliga interrupções
TMOD	Controle dos modos dos temporizadores/contadores	089H	Programa os modos de operação dos temporizadores/contadores
*TCON	Controle dos temporizadores/contadores	088H	Contém sinalizadores dos temporizadores/contadores
TH0	Byte mais significativo do temporizador/contador 0	08CH	Contém a parte mais significativa do valor atual do temporizador/contador n° 0
TL0	Byte menos significativo do temporizador/contador 0	08AH	Contém a parte menos significativa do valor atual do temporizador/contador n° 0
TH1	Byte mais significativo do temporizador/contador 1	08DH	Contém a parte mais significativa do valor atual do temporizador/contador n° 1
TL1	Byte menos significativo do temporizador/contador 1	08BH	Contém a parte menos significativa do valor atual do temporizador/contador n° 1
*SCON	Controle do canal serial	098H	Programa os modos de operação da interface serial e contém seus sinalizadores de estado
SBUF	Buffer do canal serial	099H	Mapeia o canal serial para transmissão e recepção de dados
PCON	Registrador de controle do modos de operação do sistema	087H	Dobra taxa de transmissão do canal serial e altera modo de operação de baixo consumo do sistema

* Registradores *bit*-endereçáveis

Para se habilitar qualquer uma das funções especiais das linhas acima, deve-se ativar o *bit* do endereço 0B0H (endereço da porta 3) correspondente à linha desejada. Da mesma maneira, pode-se configurar uma determinada linha de uma das portas como entrada de dados.

A.1.4 Temporizadores/Contadores

Existem dois temporizadores/contadores (T/Cs) de 16 bits disponíveis no 8031. Estes dispositivos podem operar como temporizadores ou contadores de eventos externos em 4 modos distintos e são habilitados por um sinal externo ou através de programação. O resultado da contagem de cada um dos T/Cs é armazenado nos pares de registradores TH0/TL0 para o T/C 0 e TH1/TL1 para o T/C 1.

O controle dos T/Cs é feita pelos registradores de funções especiais TMOD e TCON e o resultado da contagem é armazenada nos registradores TH0 (8 *bits* mais significativos do T/C0), TL0 (8 *bits* menos significativos do T/C0), TH1 e TL0.

Existem quatro modos de operação dos T/Cs, numerados de 0 a 3. No modo 0, cada T/C é transformado em um T/C de 13 *bits*. O modo 1 é semelhante ao modo 0, mas o T/C opera com todos os seus 16 *bits*. O modo 2 é normalmente utilizado para gerar taxas de transmissão da porta serial. Nele, os T/Cs são de 8 *bits* com recarga automática do início da contagem, ou seja, após o estouro, a contagem é reinicializada em um valor definido pelo usuário. O último modo desativa o T/C1 e transforma o T/C0 em dois T/Cs de 8 *bits* cada.

A figura A.2. mostra os registradores TMOD e TCON. O significado de cada um de seus *bits* é explicado a seguir:

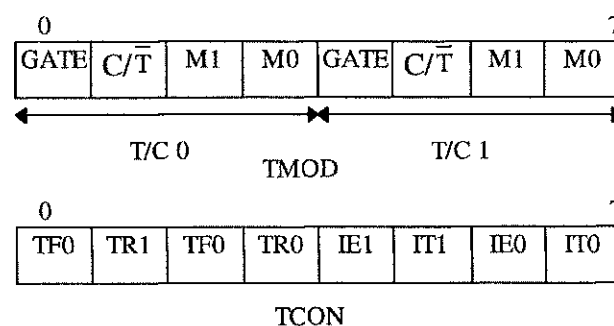


Figura A.2 - Registradores TMOD e TCON

- GATE - Define a fonte de habilitação do T/C, ou seja, se o T/C será habilitado por programação ou por *hardware*;

- $\overline{C/T}$ - Seleciona cada um dos T/Cs para operar como temporizador ou como contador;
- M1,M0 - Selecionam o modo de operação do T/C;
- TFX - Sinalizador de estouro (*overflow*) do T/Cx;
- TRx - Habilita o T/Cx;
- IEx - Sinalizador de requisição de interrupção externa em \overline{INTx} e;
- ITx - Seleciona sensibilidade da linha \overline{INTx} (borda de descida ou nível baixo).

A.1.5 Interface para Comunicação Serial

A comunicação serial de dados envolve dois sistemas de comunicação capazes de enviar ou receber dados serialmente. Esta comunicação só faz sentido se houver sincronismo entre transmissor e receptor. De outra forma, o receptor não poderia identificar os limites de início e fim de cada *bit* de dados que chega pelo canal serial. As soluções para sincronizar transmissor e receptor definem os dois modos de comunicação serial: síncrono e assíncrono [17]. No modo síncrono, o transmissor gera o sinal de sincronismo para o receptor, enquanto que no modo assíncrono a informação de sincronismo está embutida no próprio caractere transmitido ou seja, os caracteres são enviados em seqüências contendo 1 *bit* de inicialização seguido dos *bits* de dados (caractere) e 1 ou 2 *bits* de fim de caractere.

A interface serial do 8031 (porta serial) é do tipo duplex, ou seja, pode transmitir ou receber dados simultaneamente oferecendo um modo de comunicação síncrona e três modos assíncronos [17], a saber:

- modo 0 - Registrador de Deslocamento. Os dados são enviados ou recebidos através da linha RXD (P3.0) e o sinal de relógio é provido pela linha TXD (P3.1) e a taxa de transferência é fixada em 1/12 da frequência do relógio do microcontrolador;
- modo 1 - Comunicação assíncrona. Os dados são transferidos pela linha TXD em blocos de 10 *bits*: 1 bit de inicialização, 8 *bits* de dados e 1 *bit* de parada com taxa de

transferência programável. A recepção é feita pela linha RXD e a taxa de transferência é programável pelo usuário;

- modo 2 - Comunicação assíncrona. Neste modo, são transferidos blocos de 11 bits de dados, sendo 1 *bit* de início, 9 *bits* de dados e 1 *bit* de parada. Como o registrador da serial possui apenas 8 *bits*, o nono *bit* é armazenado, na posição RB8 do registrador SCON ou na transmissão, na posição TB8 do mesmo registrador. A taxa de transferência pode ser de 1/32 ou 1/64 da frequência do relógio do microcontrolador e;
- modo 3 - Comunicação assíncrona. Este modo tem as características semelhantes ao anterior, mas a taxa de transmissão pode ser programada pelo usuário.

A taxa de transferência nos modos 1 e 3 é obtida pela taxa de estouro do T/C1. Este T/C é normalmente configurado para o modo 2 e a taxa de transferência é obtida por (3), onde Clock é a frequência de operação do microcontrolador, SMOD é o estado do *bit* de mesmo nome no registrador PCON (endereço 087H) e TH1.

$$\text{Taxa} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{Clock}}{12 \times (256 - \text{TH1})} \quad (3)$$

Rescrevendo-se (3), tem-se a expressão para TH1:

$$\text{TH1} = 256 - \frac{2^{\text{SMOD}} \times \text{Clock}}{\text{Taxa} \times 32 \times 12} \quad (4)$$

Para possibilitar a operação duplex, a interface possui um buffer para transmissão e outro para recepção, mapeados em memória pelo mesmo registrador (SBUF) de endereço 099H. A leitura em SBUF acessa o registrador de recepção, enquanto que a escrita acessa o registrador de transmissão. Existem ainda outros registradores associados à interface: Os registradores de controle da serial SCON (endereço 098H) e o registrador PCON (endereço 087H), cujo *bit* 7 (SMOD) dobra a taxa de transferência de dados quando setado.

O registrador SCON possui oito bits de controle da porta serial, conforme mostrado pela figura A.3.

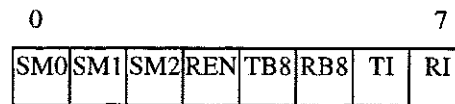


Figura A.3 - Registrador de controle da porta serial (SCON)

Cada uma das posições do registrador SCON tem uma função especial, a saber:

- SM0 SM1 - Seleciona modo de operação;
- SM2 - Habilita a comunicação multiprocessador nos modos 2 e 3;
- REN - Habilita a recepção;
- TB8 - Nono bit de dados a transmitir;
- RB8 - Nono bit de dados recebido;
- TI - Sinalizador de fim de transmissão e;
- RI - Sinalizador de fim de dado recebido

A.1.6 - Fontes de Interrupções

O 8031 possui seis fontes de interrupções vetorizadas, com controle de prioridade, sendo três externas, através das linhas $\overline{INT0}$ e $\overline{INT1}$ da porta 3 e de RESET e três internas, à partir do T/C0, do T/C1 ou da interface serial [17]. Cada fonte, com exceção do RESET (não mascarável), pode ser habilitada individualmente ou não através do registrador IE, podendo-se ainda desabilitar todas de uma vez. O registrador IE e as funções de seus bits são mostrados à seguir.

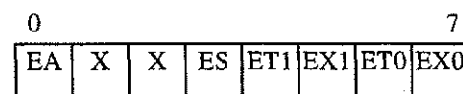


Figura A.4 - Registrador de habilitação das interrupções (IE)

- EA - Desabilita todas as interrupções quando colocado em zero. Se em nível 1, permite a escolha de interrupções habilitadas através dos demais *bits* do registrador IE;
- X - Não utilizado;

- ES - Habilita ou não a interrupção do canal serial;
- ETx - Habilita ou não a interrupção do T/Cx e;
- EXx - Habilita ou não a interrupções externas $\overline{\text{INTx}}$

O controle de prioridade de interrupções é feito de forma que uma interrupção de prioridade maior interrompe uma de prioridade menor, mesmo que esta tenha chegado primeiro. Se a interrupção que chegar for de nível igual ou menor da que já está sendo atendida, ficará aguardando o fim desta para ser servida [17]. O controle de prioridade das interrupções é feito pelo registrador IP, mostrado pela figura A.5. O significado de cada um de seus *bits* é mostrado à seguir.

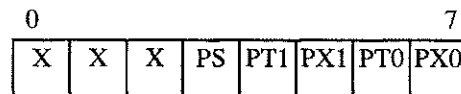


Figura A.5 - Registrador de controle de prioridade das interrupções

- X - Não utilizado;
- PS - Atribui ou não prioridade alta a serial;
- PTx - Atribui ou não prioridade alta à interrupção do T/C1 e;
- Pxx - Atribui ou não prioridade alta à interrupção externa $\overline{\text{INTx}}$.

Os vetores de interrupções estão localizados em posições de memória conforme a tabela A.II.

Interrupção	Endereço do vetor
RESET	000H
$\overline{\text{INT0}}$	003H
T/C0	00BH
$\overline{\text{INT1}}$	013H
T/C1	01BH
SERIAL	023H

Tabela A.II - Vetores de Interrupção

A.1.7 - Unidade de Controle e Temporização

A Unidade de Controle e Temporização gera os sinais de controle para separar as palavras de endereço/dados multiplexadas na porta 0 (ALE) e de busca de instrução em memória de programa externa ($\overline{\text{PSEN}}$). Esta unidade recebe os sinais de inicialização (RESET) e de existência de memórias externas conectadas ao microcontrolador ($\overline{\text{EA}}$).

Diagramas esquemáticos

Diagrama esquemático da placa de controle do módulo

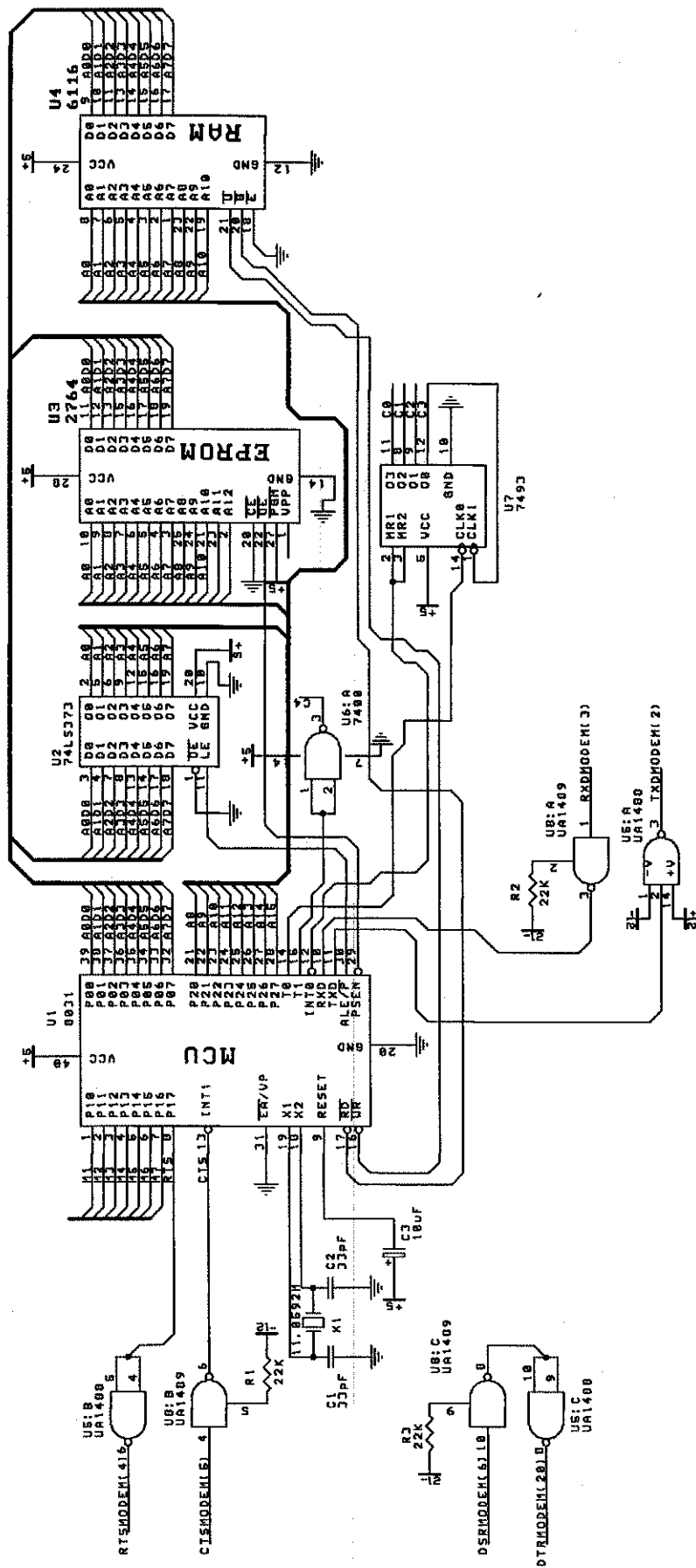


Diagrama esquemático da placa de multiplexação para 80 TPs

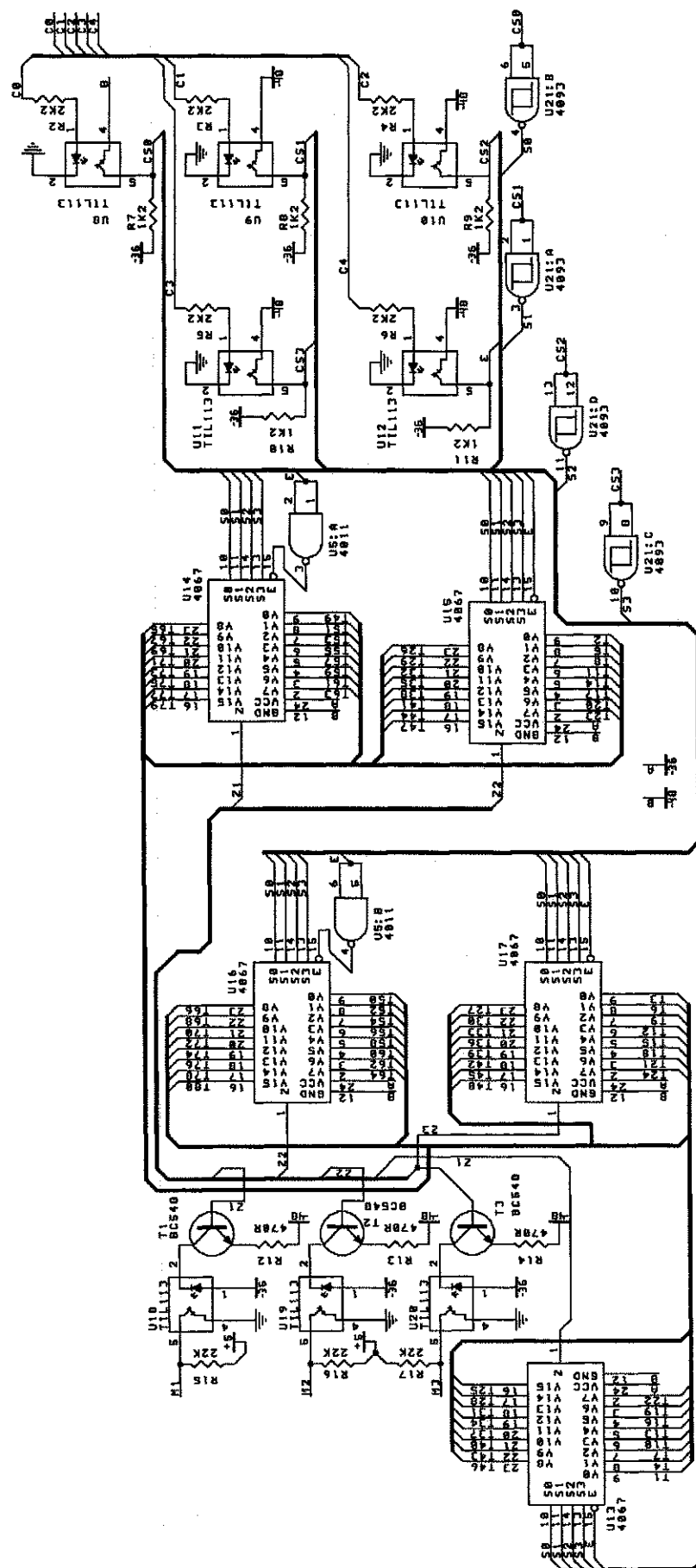
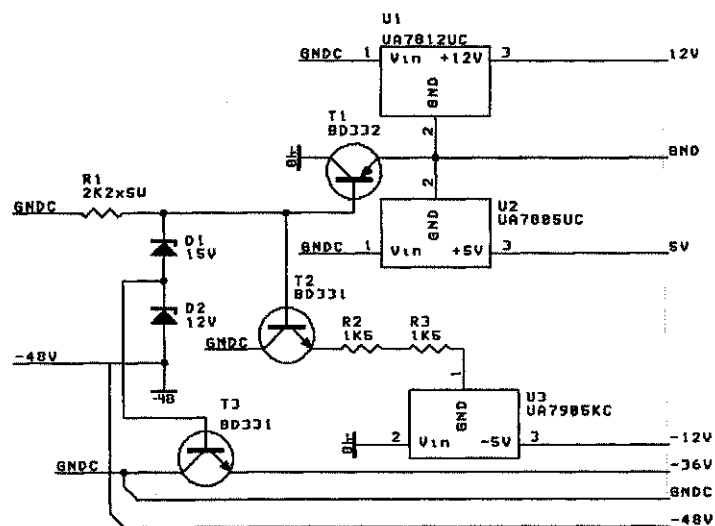


Diagrama esquemático da fonte de alimentação



Listagem do Programa Básico do Módulo

```

-----
;PROGRAMA BASICO DO MODULO SADAM-TP
-----
;Ultima alteracao: 10/08/94 - 09:44
;Ultimo teste:      05/12/94 - 19:30
;Obs:
-----

-----
;definicao da taxa de transferencia de dados
-----
;valores para clock de 6.000 MHz
;BAUD_300      EQU      204      ;byte controlador da taxa de tranmissao
;BAUD_600      EQU      230      ;idem ao equate anterior, para 600 bauds
;TAXA          EQU      BAUD_300 ;taxa de transmissao em 300 BAUDS

;valores para cristal de 8.000 Mz
;BAUD_300      EQU      186
;BAUD_600      EQU      221
;BAUD_1200     EQU      239
;BAUD_2400     EQU      247
;TAXA          EQU      BAUD_600

;Valores para cristal de 11.0592M
BAUD_300      EQU      64
BAUD_600      EQU      160
BAUD_1200     EQU      208
BAUD_2400     EQU      232
BAUD_4800     EQU      244
BAUD_9600     EQU      250
TAXA          EQU      BAUD_600

-----
;mapa de utilizacao da memoria
-----
ATUAL         EQU      020H      ;estado atual da linha dos TP's
FLAG         EQU      021H      ;sinalizador de possivel contagem de ficha
ANTL         EQU      022H      ;endereço do estado anterior do TP
ANTH         EQU      023H      ;endereço do estado anterior do TP
FICHAL       EQU      024H      ;endereço do contador de fichas do TP
FICHAH       EQU      025H      ;endereço do contador de fichas do TP
MASCARA      EQU      026H      ;endereço da mascara para selecionar bit de
TP
N_AMOSTRAL   EQU      027H      ;endereço do numero de amostras feitas no
TP
N_AMOSTRAH   EQU      028H      ;endereço do numero de amostras feitas no
TP

```

```

;-----
;mapa de utilliacao da memoria bit-enderecavel
;-----
CLOCK      EQU      0B4H      ;endereço da linha de clock
S4         EQU      0B5H      ;endereço da linha do bit S4 da selecao do
MUX
FLAG0      EQU      078H      ;flag de primeira leitura no modulo
FLAG1      EQU      079H      ;flag de inicio de comunicacao serial
RTS        EQU      P1.7      ;controle de requisicao para transmissao
CTS        EQU      P3.3      ;resposta do modem ao RTS
;-----
;constantes utilizadas no programa
;-----
MAXESPERA  EQU      0200      ;maximo numero de lacos de espera
MAXERROPAR EQU      015       ;maximo numero de erros de paridade
MAXERRO    EQU      005       ;maximo numero de erros de transmissao
CRISTAL    EQU      11        ;valor do cristal da placa em MHz
RETARDO    EQU      400       ;retardo em microsegundos
TEMPO      EQU      ((RETARDO*CRISTAL)/48-2);byte controlador do tempo do delay,
;setado para aproximadamente 47

microsegundos
LEITURAS   EQU      015       ;numero de amostras para filtragem de ruído
de linha
PILHA      EQU      030H      ;endereço inicial da pilha
N_TPS      EQU      20        ;numero de TP's por FRAME
N_FRAME    EQU      10        ;numero de frames
MAXBIT     EQU      07        ;maximo numero de bits validos
MINBIT     EQU      MAXBIT-01  ;minimo de bits validos

;-----
;mapa de utilizacao dos registradores
;-----
CONT1      EQU      R0        ;contador da selecao dos multiplexadores
CONT2      EQU      R1        ;contador de bits deslocados
CONT3      EQU      R2
CONTERRO   EQU      R4
ERROPARIDADE EQU      R5      ;contador de erros de paridade
INIBLOCOL  EQU      R6        ;parte baixa do apontador para endereço
;inicial do bloco de memoria inicial
INIBLOCOH  EQU      R7        ;idem ao anterior, parte alta
DECURSO    EQU      B         ;maximo numero de espera por

;-----
;alfabeto do protocolo de comunicacao:
;-----
LEITURA   EQU      0DCH      ;comando de leitura dos contadores
CONTINUE   EQU      0E1H      ;comando de continue a enviar
ESCRITA    EQU      0E6H      ;comando de escrita nos contadores
ERRO       EQU      0F0H      ;caractere de erro geral
EOT        EQU      0FAH      ;caractere de fim de transmissao

;-----
;programacao dos vetores de interrupcoes
;e do endereço inicial do programa
;-----
INICIO     CODE      026H      ;inicio da rotina de inicializacao
RESTART    CODE      000H      ;endereço do vetor de reset
INT_0      CODE      003H      ;endereço de vetor nao utilizado
TC0        CODE      00BH      ;endereço de vetor nao utilizado
INT_1      CODE      013H      ;endereço de vetor nao utilizado
TC1        CODE      01BH      ;endereço de vetor nao utilizado
IS         CODE      023H      ;endereço do vetor da serial

;-----
;vetores de interrupcao
;-----
ORG        RESTART          ;vetor de reset
          LJMP PRINCIPAL

ORG        INT_0            ;vetor nao utilizado
          LJMP PRINCIPAL

ORG        TC0              ;vetor nao utilizado
          LJMP PRINCIPAL

ORG        INT_1            ;vetor nao utilizado
          LJMP PRINCIPAL

```

```

ORG          TCl                ;vetor nao utilizado
              LJMP PRINCIPAL

ORG          IS                  ;vetor da serial
              LJMP INT_SERIAL

;-----
;Inicio do programa basico
;-----

ORG          INICIO

PRINCIPAL:   MOV IE, #000H        ;desabilita todas as interrupcoes
              MOV IP, #010H        ;prioridade a interrupcao da serial
              MOV SP, #PILHA        ;inicializa pilha
              SETB RXD              ;inicializa pino RXD
              SETB TXD              ;inicializa pino TXD
              SETB WR                ;inicializa pino WR
              SETB RD                ;inicializa pino RD
              MOV P1, #07FH         ;inicializa porta 1
              CLR RS0                ;seleciona banco de registradores 0
              CLR RS1
              ACALL RSET
              CLR FLAG0              ;zera bit FLAG0
              SETB FLAG1            ;seta bit FLAG1
              MOV A, #00H           ;zera acumulador
              MOV CONT1, #000H      ;zera CONT1
              MOV DPL, #000H        ;zera apontador de dados
              MOV DPH, #002H

;-----
;LACO DE INICIALIZACAO DOS CONTADORES DE FICHAS
;-----
              MOV R0, #200          ;numero de TPs

LACO0:       MOVX @DPTR, A          ;inicializa um contador
              INC DPTR
              MOVX @DPTR, A
              INC DPTR              ;aponta para proximo contador
              DJNZ R0, LACO0        ;volta ao laco ateh ultimo TP

;-----
;LACO DE INICIALIZACAO DOS CONTADORES DE AMOSTRAS
;-----
              MOV R0, #200          ;numero de TPs
              MOV DPL, #020H        ;ajusta apontador
              MOV DPH, #000H
              MOV A, #002H          ;valor do numero de amostragens

LACO1:       MOVX @DPTR, A          ;inicializa valor
              INC DPTR              ;aponta para proximo endereco
              DJNZ R0, LACO1        ;volta ao laco ateh ultimo TP

;-----
;INICIALIZACAO DA SERIAL NO MODO 3, TAXA DEFINIDA POR TAXA
;-----

SERIAL:      MOV PCON, #10000000B   ;seleciona SMOD em 1 (duplica taxa de
transmissao)
              MOV TMOD, #00100000B ;temporizador 1 no modo 2 e como contador
              MOV TH1, #TAXA        ;define taxa de transmissao
              SETB TR1              ;inicializa contador 1 da CPU
              MOV SCON, #11010000B ;habilita serial para transmissao no modo 3
              MOV IE, #10010000B   ;habilita interrupcao da serial
              SETB RTS              ;liga recepcao do MODEM
              SETB CLOCK            ;seta sinal de clock

;-----
;LACO EXTERNO DE LEITURA DE TPS, FAZ A VARREDURA DOS MULTI
;PLEXADORES
;-----

LACO2:       MOV ANTL, #000H        ;inicializa apontador de estado anterior
              MOV ANTH, #000H
              MOV FICHAL, #000H    ;inicializa apontador de fichas
              MOV FICHAH, #002H

```

```

MOV N_AMOSTRAL,#020h      ;inicializa apontador de amostras
MOV N_AMOSTRAH,#000h
MOV R6, #000H             ;guarda inicio do apontador de fichas
MOV R7, #002H
MOV R0, #000H             ;inicializa variavel CONT1
CLR S4                    ;inicializa bit S4 da selecao

LACO3:                    ;carrega data_pointer em ANT
MOV DPL,ANTL
MOV DPH,ANTH
ACALL DELAY               ;executa retardo de tempo para es-
                          ;tabilizacao do sinal de selecao
                          ;dos multiplexadores
MOV ATUAL,P1              ;ler porta 1
JNB FLAG0,INICIAFLAG     ;se bit FLAG0 = 0, zera FLAG
MOVX A,@DPTR              ;ler estado anterior
ORL A,ATUAL               ;compara com atual
MOV FLAG,A                ;armazena resultado em FLAG

DESVIO1:                  ;ler estado atual
MOV A,ATUAL
CPL A                     ;complementa
MOVX @DPTR,A             ;salva resultado em estado anterior
INC ANTL                 ;incrementa apontador do estado do TP
MOV R1, #000H            ;inicializa variavel CONT2
MOV DPL,FICHAH           ;carrega data_pointer em FICHA
MOV DPH,FICHAH
MOV MASCARA,#00000001B   ;inicializa mascara

LACO4:                    ;verifica FLAG
MOV A, FLAG
RRC A                     ;carrega bit LSB de FLAG em carry
MOV FLAG, A              ;devolve FLAG
INC N_AMOSTRAL           ;incrementa apontador de amostras do TP
JNC FILTRO                ;se sinal de TP estah em 0, filtra sinal

DESVIO2:                  ;atualiza data_pointer
INC DPTR
INC DPTR

DESVIO3:                  ;desloca mascara a esquerda
MOV A,MASCARA
RL A                       ;
MOV MASCARA, A           ;devolve mascara
INC R1                    ;atualiza variavel CONT2
JB S4, CONTINUA1         ;se bit S4 da selecao for 0 compare CONT2
CJNE R1,#MAXBIT,LACO4    ;com MAXBIT, senao compare com MINBIT
SJMP SAI1                 ;sai do LACO3

CONTINUA1:                ;se nao atingir MAXBIT ou MINBIT, entao
CJNE R1,#MINBIT,LACO4    ;continua LACO3

SAI1:                     ;incrementa variavel CONT1
INC R0
CJNE R0,#016,CONTINUA2   ;se CONT1 for igual a 16, entao S4=1
SETB S4

SAI2:                     ACALL INC_CONTADOR
MOV FICHAH,DPL
MOV FICHAH,DPH
SJMP LACO3

CONTINUA2:                CJNE R0,#031,SAI2
SETB FLAG0
ACALL INC_CONTADOR
SJMP LACO2

ATUALIZA:                 INC DPTR
MOVX A,@DPTR
ADD A, #001H
MOVX @DPTR,A
DEC DPL
MOVX A,@DPTR
ADDC A, #000H
MOVX @DPTR,A
SJMP DESVIO2

INICIAFLAG:              MOV FLAG,ATUAL
SJMP DESVIO1

```

```

;-----
;ROTINA PARA FILTRAGEM DO SINAL DA LINHA DO TP:
;SE HOUVER RUIDO, DESCONSIDERA DESCIDA DO SINAL
;-----

```

```

FILTRO:      SETB RSO                ;seleciona banco de registradores 1
             MOV R0,#LEITURAS      ;inicializa r0 com numero de leituras

LER:         MOV A,P1              ;ler estado da porta P1
             ANL A,MASCARA          ;seleciona TP de interesse
             JNZ AJUSTA_AMOSTRA    ;desconsidera descida do sinal
             DJNZ R0,LER            ;volta a leitura
             CLR RSO               ;seleciona banco de registradores 0

             DEC N_AMOSTRAL        ;atualiza apontador de amostras para
             ;ultimo TP lido
             PUSH DPH              ;salva DPTR
             PUSH DPL
             MOV DPH,N_AMOSTRAH    ;carraga DPTR com apontador de amostras
             MOV DPL,N_AMOSTRAL
             MOVX A,@DPTR          ;ler numero de amostras do TP
             DEC A                 ;decrementa numero de amostras
             JNZ RETORNA_ESTADO    ;se numero de amostras diferente de zero
             ;desvia para retorna_estado
             MOV A,#002H           ;reinicializa numero de amostras

RETORNA_ESTADO: MOVX @DPTR,A       ;recupera numero de amostras
               INC N_AMOSTRAL      ;recupera apontador de amostras
               POP DPL             ;recupera DPTR
               POP DPH

               CJNE A,#02H,ESTADO_ANTER ;se numero de amostras eh dois, atualiza
               AJMP ATUALIZA       ;atualiza contador

ESTADO_ANTER: DEC ANTL             ;recupera estado do TP
               PUSH DPH           ;salva DPTR
               PUSH DPL
               MOV DPH,ANTH
               MOV DPL,ANTL
               MOVX A,@DPTR       ;ler estado do TP
               XRL A,MASCARA      ;desconsidera descida do sinal
               MOVX @DPTR,A
               POP DPL
               POP DPH
               INC ANTL           ;atualiza ANTL
               CLR RSO            ;seleciona banco de registradores 0
               AJMP DESVIO2       ;volta ao processamento normal

AJUSTA_AMOSTRA: DEC N_AMOSTRAL    ;atualiza apontador de amostras para
               ;ultimo TP lido
               PUSH DPH           ;salva DPTR
               PUSH DPL
               MOV DPH,N_AMOSTRAH ;carraga DPTR com apontador de amostras
               MOV DPL,N_AMOSTRAL
               MOV A,#02H
               MOVX @DPTR,A
               POP DPH
               POP DPL
               INC N_AMOSTRAL
               SJMP ESTADO_ANTER

INC_CONTADOR: CLR CLOCK
               NOP
               NOP
               SETB CLOCK
               RET

T_ESCRITA:   AJMP T_ESCRITA

;-----
;SUBROTINA PARA TRATAMENTO DE INTERRUPCAO DA SERIAL, COM TRA-
;TAMENTO DE ERRO DE PARIDADE OU DECURSO DE TEMPO
;-----

INT_SERIAL:  CLR RTS              ;solicita transmissao
             CLR EA              ;desabilita interrupcoes
             PUSH DPL            ;salva contexto
             PUSH DPH
             PUSH PSW
             PUSH ACC

```



```

MOV A,R2
PUSH ACC
MOV A, SBUF ;recebe caractere transmitido
CLR RI
CJNE A,#LEITURA,T_ESCRITA ;se char != LEITURA vai p/ T_ESCRITA

ACALL TRANSMITE ;ecoa LEITURA
ACALL ESPERA ;espera confirmacao de LEITURA
CJNE A, #LEITURA, SAI3 ;abandona se houve erro

MOV DPL, R6
MOV DPH, R7
MOV R2,#00H

LACO5: MOV R6,DPL
MOV R7,DPH
MOV R3,#00H
MOV R4,#00H

LACO6: MOVX A,@DPTR ;laco interno de transmissao
ACALL TRANSMITE ;transmite dados de 20 Tps,
INC DPTR ;correspondendo a 40 bytes
MOVX A,@DPTR
ACALL TRANSMITE
INC DPTR
INC R3
CJNE R3,#N_TPS, LACO6
ACALL ESPERA ;espera confirmacao para continuar
CJNE A,#CONTINUE, SUBERRO
INC R2
CJNE R2, #N_FRAME, LACO5 ;laco externo de transmissao
MOV A,#EOT ;envia Fim de Transmissao
ACALL TRANSMITE

MOV R6, #000H
MOV R7, #002H

SAI3: POP ACC
MOV R2,A
POP ACC
POP PSW
POP DPH
POP DPL
CLR RI
CLR TI
SETB EA
RETI

SUBERRO: CJNE A,#ERRO, SAI3
INC R4
CJNE R4,#MAXERRO, PULA1
AJMP SAI3

PULA1: MOV DPL, R6
MOV DPH, R7
AJMP LACO5

ESPERA: CLR RI
MOV R5, #000H

VOLTAESPERA: MOV B, #000H

TESTERI: JNB RI, PULA2
MOV A,SBUF
CLR RI
ACALL VERPARIDADE
JC ERROPAR
RET

PULA2: ACALL DELAY
ACALL DELAY

INC B
PUSH ACC
MOV A,B

```

```

                CJNE A, #MAXESPERA ,VOLTARI
                POP ACC

SAI4:          DEC SP
                DEC SP
                SJMP SAI3

VOLTARI:       POP ACC
                SJMP TESTERI

TESCRITA:      CJNE A, #ESCRITA, SUBERRO2
                ACALL TRANSMITE
                ACALL ESPERA
                CJNE A, #ESCRITA, SAI3
                ACALL ESPERA
                CJNE A, #200, TESTA          ;se A>199 C=0 senao C=1

TESTA:         JNC SUBERRO2
                MOV DPH, #002H
                MOV B, #002H
                MUL AB
                MOV DPL, A
                MOV A, B
                ADD A, DPH
                MOV DPH, A
                MOV A, #000H
                MOVX @DPTR, A
                INC DPTR
                MOVX @DPTR, A

                AJMP SAI3

ERROPAR:       INC R5
                CJNE R5, #MAXERROPAR ,TRANSMERRO
                SJMP SAI4

TRANSMERRO:    MOV A, #ERRO
                ACALL TRANSMITE
                SJMP VOLTAESPERA

;-----
;SUBROTINA PARA TESTE DE PARIDADE: SE PARIDADE ERRADA,
;C=1
;-----

VERPARIDADE:   PUSH ACC
                MOV C, P
                CLR A
                MOV 26H.0, C
                MOV C, RB8
                ADDC A, 26H          ;(realiza XOR entre 26H.0 e TB8)
                RRC A
                POP ACC
                RET

;-----
;SUBROTINA PARA TRANSMISSAO DE UM CARACTERE VIA SERIAL COM
;GERACAO DE BIT DE PARIDADE PAR
;-----

TRANSMITE:     CLR RTS              ;liga transmissao do MODEM
                PUSH ACC

ESPERA_CTS:    MOV ACC, 100          ;espera no maximo 100*400us=120 ms
                ACALL DELAY
                ACALL DELAY
                ACALL DELAY
                MOV C, CTS          ;le CTS
                JNC SAI_ESPERA
                DJNZ ACC, ESPERA_CTS ;volta a esperar CTS
                POP ACC
                AJMP SAI4          ;abandona se houver decurso de tempo

SAI_ESPERA:    POP ACC
                MOV C, P
                MOV TB8, C          ;modem ativou CTS

```

```
TRANS:      MOV SBUF,A
            JNB TI, TRANS
            CLR TI
            SETB RTS                ;desliga transmissao do MODEM
            RET

SUBERRO2:   MOV A, #ERRO
            ACALL TRANSMITE
            AJMP SAI3

;-----
;SUBROTINA DE RETARDO DE TEMPO
;-----

DELAY:      SETB RS1
            NOP
            MOV R2,#TEMPO
LAC07:      NOP
            NOP
            DJNZ R2,LAC07
            CLR RS1
            RET

;-----
;SUBROTINA DE RESET DO CONTADOR DA UNIDADE DE MULTIPLEXACAO
;-----

RSET:       CLR P3.2                ;reset do contador
            SETB P3.2
            NOP
            NOP
            NOP
            CLR P3.2
            RET

            END

;-----
;FIM DO PROGRAMA BASICO DO MODULO SADAM-TP
;-----
```

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Ferrari, A. M., *Telecomunicações: Evolução e Revolução*, São Paulo:Érica, 1991.
- [2] Moreira, M. M., *Progresso Técnico e Estrutura de Mercado: O Caso da Indústria de Teleequipamentos*, Rio de Janeiro:UFRJ, 1989.
- [3] Fernandes, S.F. de L.& Barros, A.T. de & Vieira, E.C.A. et alli, *Um Gerenciador de Telefones Públicos - SADAM-TP*, Anais do 11° Simpósio Brasileiro de Telecomunicações, Brasil,1993.
- [4] Vieira, E.C.A. & Barros, A.T. de & Fernandes, S.F. de L. et alli, *Telefonia Pública e a Viabilidade das Centrais Eletromecânicas*, Anais do 4° Congresso Internacional de Telecomunicações e Teleinformática, Brasil,1993.
- [5] Souza, M., *Tráfego Telefônico*, Telebrasil, Revista Brasileira de Telecomunicações, Maio/Junho, 1978.
- [6] de Albuquerque, Fábio Cunha, *Sistema Hierárquico com Unidade Remota Aplicado à Medição de Tráfego Telefônico*, UFPB, Brasil, 1989.
- [7] Neto, V.C., Rattes, T.M.F., *Telefonia em Sistemas Locais: Tópicos Avançados*, São Paulo: Érica, 1991.
- [8] Telebrás, *Sinalização de Linha*, Manual de Circulação Interna, Brasil.
- [9] Costa. C. & Vieira, E.C.A. et alli, *Placa de Desenvolvimento para 8031 de Baixo Custo*, Anais da 46ª Reunião da Sociedade Brasileira para o Progresso da Ciência (SBPC), Brasil, 1994.
- [10] Texas Instruments, *TTL Data Book*, USA, 1991.
- [11] Texas Instruments, *Mos Memory Data Book*, USA, 1991.
- [12] Motorola, *Motorola Memory Data Manual*, USA, 1980.
- [13] National, *Interface Databook*, USA, 1986.
- [14] Campbell, J., *RS 232: Técnicas de interface*, Rio de Janeiro: EBRAS, 1990.

- [15] Mizrahi, V.V., *Treinamento em Linguagem C*, São Paulo: Makron Books, 1993.
- [16] General Electric, *Optoelectronics*, USA, 1976.
- [17] Silva, V.P., *Microcontrolador 8051*, São Paulo: Érica, 1990.