

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA E QUÍMICA DE SÃO CARLOS

“DESENVOLVIMENTO DE HARDWARE
E SOFTWARE PARA VIABILIZAR
A OPERAÇÃO DE UM
MICRODENSITÔMETRO”

Márcio Alexandre Marques

*Dissertação apresentada ao Instituto
de Física e Química de São Car-
los, Universidade de São Paulo, para
obtenção do Título de Mestre em
Física Aplicada*

ORIENTADOR: *Prof. Dr. Jan Frans Willem Slacks*

DEPARTAMENTO DE FÍSICA E CIÊNCIA DOS MATERIAIS

SÃO CARLOS

1992

OK

USP / IFQSC / SBI



8-2-000963



**IFSC UNIVERSIDADE
DE SÃO PAULO**
Instituto de Física e Química de São Carlos

Fone (0162) 72-6222
Fax (0162) 72-2218

Av. Dr. Carlos Botelho, 1465
Caixa Postal 369
CEP 13560.970 - São Carlos - SP
Brasil

MEMBROS DA COMISSÃO JULGADORA DA DISSERTAÇÃO DE MESTRADO DE Marcio Alexandre Marques APRESENTADA
AO INSTITUTO DE FÍSICA E QUÍMICA DE SÃO CARLOS, DA UNIVERSIDADE DE SÃO PAULO, EM 22.09.1992

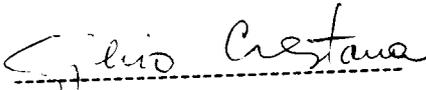
COMISSÃO JULGADORA:



Prof. Dr. Jan Frans Willem Slaets



Prof. Dr. Ulysses Borelli Thomaz Junior



Prof. Dr. Silvio Crestana

Dedico,

À minha mãe, Iracema,
pelo apoio, incentivo e
imenso amor...

Agradecimentos

Ao Prof. Dr. Jan Frans Willem Slaets, pela orientação e paciência dispensadas para que este trabalho se realizasse, e pelos ensinamentos transmitidos que muito contribuíram para minha formação profissional.

Aos professores Valentin, Gonzalo e Carlos Antônio cujos ensinamentos ofereceram-me ótimas oportunidades de aprendizado.

Ao CNPq pelo apoio financeiro.

Aos amigos do LIE, Marcos, Ivanilda e ao Lírío pelas idéias, discussões e opiniões muito proveitosas e especialmente ao Ailton pela dedicação e paciência com que sempre me auxiliou para resolver os problemas com o Photoscan.

À Benê, desenhista do Grupo de Instrumentação e Informática (GII) do IFQSC, pela prontidão, paciência, dedicação e competência com que realizou os desenhos contidos nesta dissertação.

À Lia, pela amizade e boa vontade com que sempre realizou os trabalhos de secretaria.

Ao Mauro ("Massa") e ao Mateus pela ajuda nas minhas dúvidas computacionais.

Ao amigo João Eduardo, pela paciência com que sempre solucionou minhas dificuldades com o WordPerfect.

À Adriana e a Patricia, pela utilização do microcomputador.

À Patricia e a Regina, minhas amigas desde a graduação, pela amizade, apoio e idéias muito proveitosas, em especial à Regina pela ajuda na estética final do texto e impressão deste trabalho.

Aos meus pais (Antonio e Iracema), irmãos (Manoel, Marco, Mário e Milton) e irmãs (Diva e Mathilde), pelo apoio, incentivo e oportunidade para que eu pudesse estudar e realizar este trabalho.

A todos os amigos, que direta ou indiretamente contribuíram com seu apoio e auxílio. Em especial : Amilton, Cal, Zézo, Kleber, Ana Maria, André, Gerson, Luciane, João Célio, Giulio, Neusa e Sibely.

SUMÁRIO

Lista de Figuras	i
Lista de Tabelas	iii
Resumo	iv
Abstract	v
Capítulo 1 - INTRODUÇÃO	1
Capítulo 2 - VARREDORES DE IMAGENS	3
2.1 - Introdução	3
2.2 - Requisitos Funcionais dos Varredores	5
2.3 - Aplicações dos Varredores de Imagens	7
2.4 - Classificação dos Varredores	8
2.4.1 - Evolução dos Varredores Tipo Tambor Rotatório	9
Capítulo 3 - HARDWARE	13
3.1 - Interface Paralela Desenvolvida	13
Capítulo 4 - SOFTWARE	19
4.1 - Introdução	19
4.2 - Programa Principal	20
4.2.1 - Subrotina GetSelection	22
4.3 - Rotinas do Photoscan	28
4.3.1 - Programação das 8255	28
4.3.2 - Obtenção do valor da Resolução ("RASTER")	30
4.3.3 - Comandos para movimentar o subsistema óptico do Photoscan	32
4.3.4 - Leitura dos Dados (Densidades)	34
Capítulo 5 - RESULTADOS E CONCLUSÕES	41

Referências Bibliográficas	53
Apêndice A - O Photoscan Optronics P-1000	A-1
A.1 - Introdução ao Photoscan Optronics P-1000	A-2
A.1.1 - Mecanismo de Varredura	A-3
A.1.2 - Subsistema Óptico	A-3
A.1.3 - Subsistema de Acionamento e Codificação	A-5
A.1.4 - Controles e Indicadores	A-6
A.2 - Especificações da Interface do Photoscan	A-9
Apêndice B - A Interface de Comunicação Paralela Programável - 8255	B-1
B.1 - Interface de Comunicação Paralela Programável - 8255	B-2
B.1.1 - Arquitetura Interna da 8255	B-2
B.1.2 - Descrição Funcional da 8255	B-3
B.1.3 - Modo de Operação	B-5
Apêndice C - Listagem do Programa desenvolvido	C-1
Apêndice D - Instruções para utilização do Software desenvolvido	D-1

Lista de Figuras

Figura 2.1 - Componentes de um Digitalizador de Imagens e suas Inter-relações	5
Figura 2.2 - Requisitos Funcionais dos Varredores de Imagem	6
Figura 2.3 - Varredor Tipo Tambor Rotatório	12
Figura 3.1 - Esquema da Interface Desenvolvida	16
Figura 4.1 - Diagrama em blocos do software desenvolvido	19
Figura 4.2 - Fluxograma do programa principal	21
Figura 4.3 - Ilustração das posições da linha a ser digitalizada	23
Figura 4.4 - Ilustração das posições da coluna a ser digitalizada	24
Figura 4.5a - Fluxograma da rotina GetSelection	25
Figura 4.5b - Continuação do fluxograma da rotina GetSelection	26
Figura 4.5c - Continuação do fluxograma da rotina GetSelection	27
Figura 4.6 - Configuração Funcional para o 8255 (1)	28
Figura 4.7 - Configuração Funcional para o 8255 (2)	29
Figura 4.8 - Ciclo completo para carregar dado na porta A	30
Figura 4.9 - Fluxograma para obtenção do valor da resolução ("Raster")	31
Figura 4.10 - Diagrama de sincronização para executar passo à direita ou à esquerda	32
Figura 4.11 - Fluxograma da seqüência de passos para movimentar o subsistema óptico do Photoscan	33
Figura 4.12 - Diagrama de sincronização para leitura	34
Figura 4.13 - Diagrama de sincronização para leitura de um ponto ("pixel")	35
Figura 4.14 - Fluxograma para execução da leitura de um ponto	37
Figura 4.15 - Funcionamento do monoestável	38
Figura 4.16 - Diagrama de sincronização para leitura de um dado da 8255 (2)	39
Figura 4.17 - Fluxograma para leitura dos dados	40
Figura 5.1 - "Tela" inicial	42
Figura 5.2 - Menu principal	42

Figura 5.3 - Linha a ser digitalizada (START) e posição inicial da tomada dos dados (COLUNA)	43
Figura 5.4 - Posição final da tomada dos dados (POS_FINAL)	43
Figura 5.5 - Gráfico das densidades pela distância X (resolução)	44
Figura 5.6 - Ilustração dos cursores nos extremos do gráfico a ser ampliado ("zoom") (X-scan)	45
Figura 5.7 - Ilustração do gráfico ampliado ("zoom") (X-scan)	46
Figura 5.8 - Coluna a ser digitalizada (COLUNA)	47
Figura 5.9 - Posição inicial da tomada dos dados (START) e tamanho da coluna a ser digitalizada (SIZE)	47
Figura 5.10 - Gráfico das densidades pela distância Y (resolução)	48
Figura 5.11 - Ilustração dos cursores nos extremos do gráfico a ser ampliado ("zoom") (Y-scan)	49
Figura 5.12 - Ilustração do gráfico ampliado ("zoom") (Y-scan)	50
Figura 5.13 - Armazenamento no arquivo DATA1.DEN dos dados digitalizados e as outras informações mostradas na tela	51
Figura A.1 - Diagrama em blocos do sistema óptico de leitura	A-4
Figura A.2 - Painel de controles e indicadores do Photoscan	A-7
Figura A.3 - Diagrama em blocos das ligações dos sinais do Photoscan	A-10
Figura B.1 - Arquitetura Interna da 8255	B-3
Figura B.2 - Definição da palavra de controle	B-7

Lista de Tabelas

Tabela 3.1 - Distribuição dos pinos de Entrada do Photoscan na 8255 (1)	17
Tabela 3.2 - Distribuição dos pinos de Saída do Photoscan na 8255 (2)	18
Tabela A.1a - Controles e Indicadores do Photoscan	A-8
Tabela A.1b - Continuação dos Controles e Indicadores do Photoscan	A-9
Tabela A.2a - Descrição dos pinos de Entrada	A-11
Tabela A.2b - Continuação da Descrição dos pinos de Entrada	A-12
Tabela A.3 - Dados de Entrada	A-13
Tabela A.4 - Valores da Palavra de Controle	A-13
Tabela A.5 - Descrição dos pinos de Saída	A-14
Tabela A.6 - Dados de Saída	A-15
Tabela A.7 - Valores da Resolução	A-16
Tabela B.1 - Endereçamento da 8255	B-6

Resumo

O presente trabalho foi desenvolvido para viabilizar a operação do microdensitômetro Optronics P-1000 através de um microcomputador tipo IBM-PC.

Assim, desenvolveu-se uma interface (hardware), bem como todo o Software necessário para operar o equipamento e fazer a aquisição dos dados digitalizados.

Este software, permite, também, a visualização interativa das imagens, usada para definir regiões de interesse no filme.

Abstract

The present work was developed to enable the operation of the Optronics P-1000 densitometer using a IBM-PC compatible microcomputer.

Therefore a hardware interface as well as all the needed Software to operate the equipment and execute the data aquisition was developed.

This software provides also interactive visualization and operation used to define regions of interest on the film.

Capítulo 1

INTRODUÇÃO

O objetivo dessa dissertação é descrever o desenvolvimento de hardware e software para viabilizar a utilização de um microdensitômetro automático através de um computador tipo IBM-PC.

Trata-se da utilização de um equipamento Photoscan Optronics P-1000 existente no laboratório de pesquisa, utilizado, principalmente, para a digitalização de filmes obtidos por técnicas de difração de raio-X e microscopia eletrônica.

Quando o Departamento de Física e Ciência dos Materiais (DFCM) do Instituto de Física e Química de São Carlos (IFQSC) - USP obteve o Photoscan Optronics P-1000 para uso em cristalografia, na digitalização de filmes de raio-X, utilizavam-se programas para digitalizar as imagens, que eram executados pelo computador PDP-11-45. Este estava acoplado ao microdensitômetro Optronics, que fazia a coleta dos dados digitalizados, e seu processamento era utilizado para a determinação das intensidades de reflexões em experimentos de difração de raio-X.

A desvantagem deste procedimento era a falta de uma interface gráfica para a interpretação preliminar das intensidades digitalizadas. Na então utilização do sistema, composto pelos programas computacionais e o Optronics, estes eram tratados como uma "caixa-preta", destinados apenas para aplicações em cristalografia.

Tanto a diversificação do uso do Optronics, quanto a disponibilidade de microcomputadores capazes de controlar e efetuar a aquisição dos dados digitalizados, resultaram no desejo de desenvolver hardware e software para permitir a operação do Optronics a partir do microcomputador.

A desativação do computador PDP-11-45 tomou, então, a execução deste projeto indispensável para garantir a utilização do Optronics.

Com a existência dos recursos gráficos nos microcomputadores, definiu-se no projeto uma interface com recursos gráficos capaz de visualizar interativamente as intensidades digitalizadas, facilitando assim, a determinação das regiões de interesse dos filmes.

O projeto, assim se definiu em duas partes básicas : desenvolvimento de um hardware para o qual foi feita uma interface baseada nos circuitos integrados 8255, e em circuitos de arranjos lógicos programáveis (PAL), para simplificar a geração dos sinais de controle; e, em uma segunda parte, o software, que foi desenvolvido principalmente em linguagem PASCAL fazendo uso extenso das capacidades gráficas tanto para implementação da visualização de curvas de densidades, quanto de indicadores gráficos (cursos localizados nos extremos da curva ou gráfico obtido), utilizados para determinar a região do filme a ser digitalizada, eliminando a necessidade de se efetuar medidas manuais.

A presente dissertação está dividida em 4 capítulos, os quais são brevemente descritos a seguir :

- Capítulo 2 : são apresentados os elementos básicos que compõem um digitalizador de imagens, suas principais aplicações e, um estudo sobre a evolução dos varredores tipo tambor rotatório, destacando as principais características de funcionamento e de construção deste tipo de varredor.

- Capítulo 3 : descreve a construção da interface desenvolvida (hardware).

- Capítulo 4 : apresenta as principais características e implementações do software desenvolvido, para que, em conjunto com o hardware construído, viabilizasse a operação do Optronics a partir do microcomputador.

- Capítulo 5 : são apresentados os resultados e conclusões gerais sobre o trabalho realizado, bem como sugestões para futuras expansões.

Capítulo 2

VARREDORES DE IMAGENS

2.1 - Introdução.

Segundo Castleman em {CA 79}, o processamento de uma imagem digital, requer um computador no qual a imagem será processada. Em adição, o sistema pode ter duas partes especiais : um equipamento de entrada/saída, um digitalizador de imagens e um dispositivo de exposição da imagem.

Na forma como eles usualmente ocorrem, as imagens não são diretamente tratadas para análise do computador. Desde que os computadores trabalham com uma razão numérica, então os dados ilustrados em uma imagem podem ser convertidos para uma forma numérica antes do processamento. Este processo de conversão é chamado **digitalização**. A imagem é dividida em pequenas regiões chamadas **elementos da imagem** ou "**pixel**". A mais comum subdivisão da figura é a amostragem retangular da rede. A imagem é dividida em linhas horizontais consistindo de "pixels" adjacentes. Para cada "pixel", o brilho da imagem é amostrado e quantificado. Ou seja, através da relação luz incidente/luz transmitida, pode-se ler a densidade óptica em cada "pixel". A densidade óptica D_1 , correspondente ao enegrecimento do filme, é definida como o logaritmo da razão entre a intensidade de luz incidente I_1 , pela transmitida I_2 :

$$D_1 = \log (I_1 / I_2) \qquad 0 \leq D_1 \leq \infty \qquad (1)$$

Este passo, gera um valor inteiro para cada "pixel", representando a densidade óptica da imagem naquele ponto. Feito isto para todos os "pixels", a imagem é então representada por uma matriz de inteiros. Cada "pixel" tem agora uma localização ou endereço (número da linha e número da amostra ou coluna) e um valor inteiro representando o nível de cinza

(0 - 255), onde o valor zero indica que toda a luz incidente no filme foi transmitida através dele e, o valor 255 corresponde ao enegrecimento máximo do filme, geralmente limitado até uma densidade de 2 ou 3. Esta matriz de dados digitais pode agora ser utilizada para análise e processamento no computador.

Assim, Castleman define um digitalizador de imagens como sendo um dispositivo capaz de dividir uma imagem em "pixels" e endereçá-los individualmente, medir o nível de cinza em cada "pixel", quantificar a medida produzindo um inteiro e escrever o conjunto de inteiros em um dispositivo armazenador de dados.

Para efetuar este procedimento, um digitalizador deve ter cinco elementos básicos. O primeiro é uma abertura de amostragem, que permite ao digitalizador acessar os "pixels" individualmente, ignorando o restante da imagem. O segundo elemento de um digitalizador de imagens é um mecanismo de varredura da imagem. Este processo consiste em movimentar a abertura de amostragem sobre a imagem em um padrão pré-determinado. A varredura permite que a abertura de amostragem enderece os "pixels" na ordem de um "pixel" por vez.

O terceiro elemento é um sensor, que pode medir o brilho da imagem em cada "pixel" (absorção óptica, no caso do filme), através da abertura de amostragem. O sensor é comumente um transdutor que converte a intensidade de luz em tensão ou corrente elétrica. O quarto elemento, um quantizador, converte a saída contínua do sensor em um valor inteiro. Tipicamente, o quantizador é um circuito eletrônico chamado de conversor analógico para digital (A/D). Este, produz um número que é proporcional à tensão ou corrente elétrica de entrada. Para obtenção de valores em densidades é utilizado um amplificador logarítmico.

O quinto elemento de um digitalizador de imagens é um dispositivo de armazenamento ou de saída (por exemplo um vídeo). Os valores dos níveis de cinza produzidos pelo quantizador, devem ser armazenados em um formato apropriado para um subsequente processamento feito por um computador. Tecnicamente, o dispositivo de armazenamento pode ser omitido se a imagem for processada "on-line". A digitalização da imagem frequentemente é feita no sistema "off-line" através de um computador, sendo necessário se ter um dispositivo de armazenamento. Este dispositivo pode ser uma fita ou disco magnético. Como o número de dados de uma digitalização de imagens é grande, a capacidade destes dispositivos de saída também deve ser grande. Por exemplo, um filme de 100 mm X 100 mm, digitalizado com uma resolução de 100 μm produz : $100 \times 100 \times 10 = 10^5$ ou 100 Kbytes e, no caso de uma resolução de 25 μm , 400 Kbytes.

Na figura 2.1, {NA 83} esquematiza os Digitalizadores de Imagens, bem como as inter-relações entre os vários componentes.

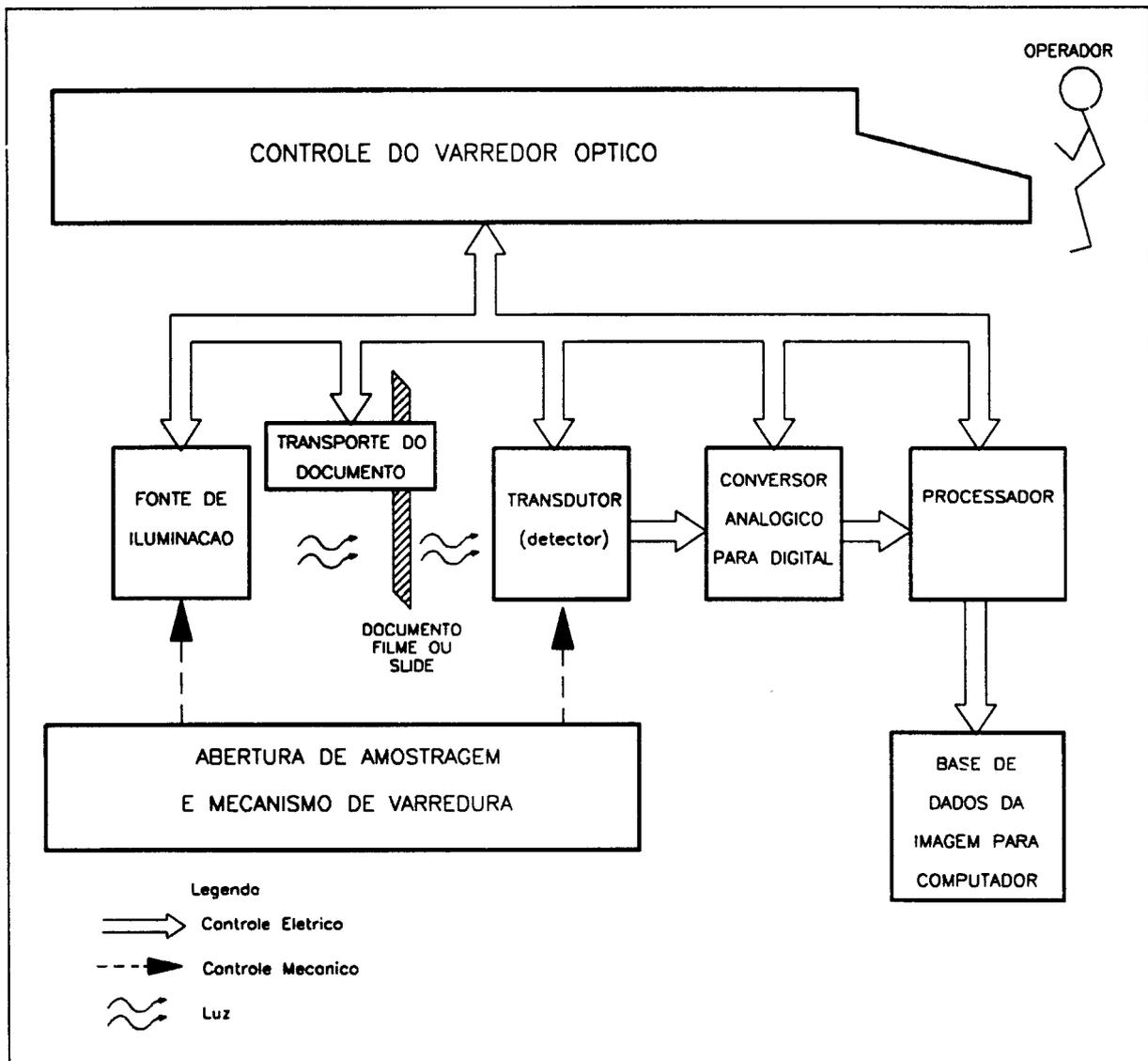


Figura 2.1 - Componentes de um Digitalizador de Imagens e suas Inter-relações

2.2 - Requisitos Funcionais dos Varredores.

Em {MO 80} são encontradas as três funções básicas associadas a qualquer varredor de imagem :

- Digitalizar áreas da imagem, áreas selecionadas ou a imagem toda.
- Fornecer informações que relacionam cada ponto com todos os outros; ou seja, considerações para localização dos pontos no espaço.
- Saída com imagem e coordenadas na forma digital para serem usadas pelo sistema ou mesmo para um armazenamento intermediário.

A figura 2.2 mostra as três funções básicas de qualquer varredor de imagens, bem como um conjunto secundário de suas funções.

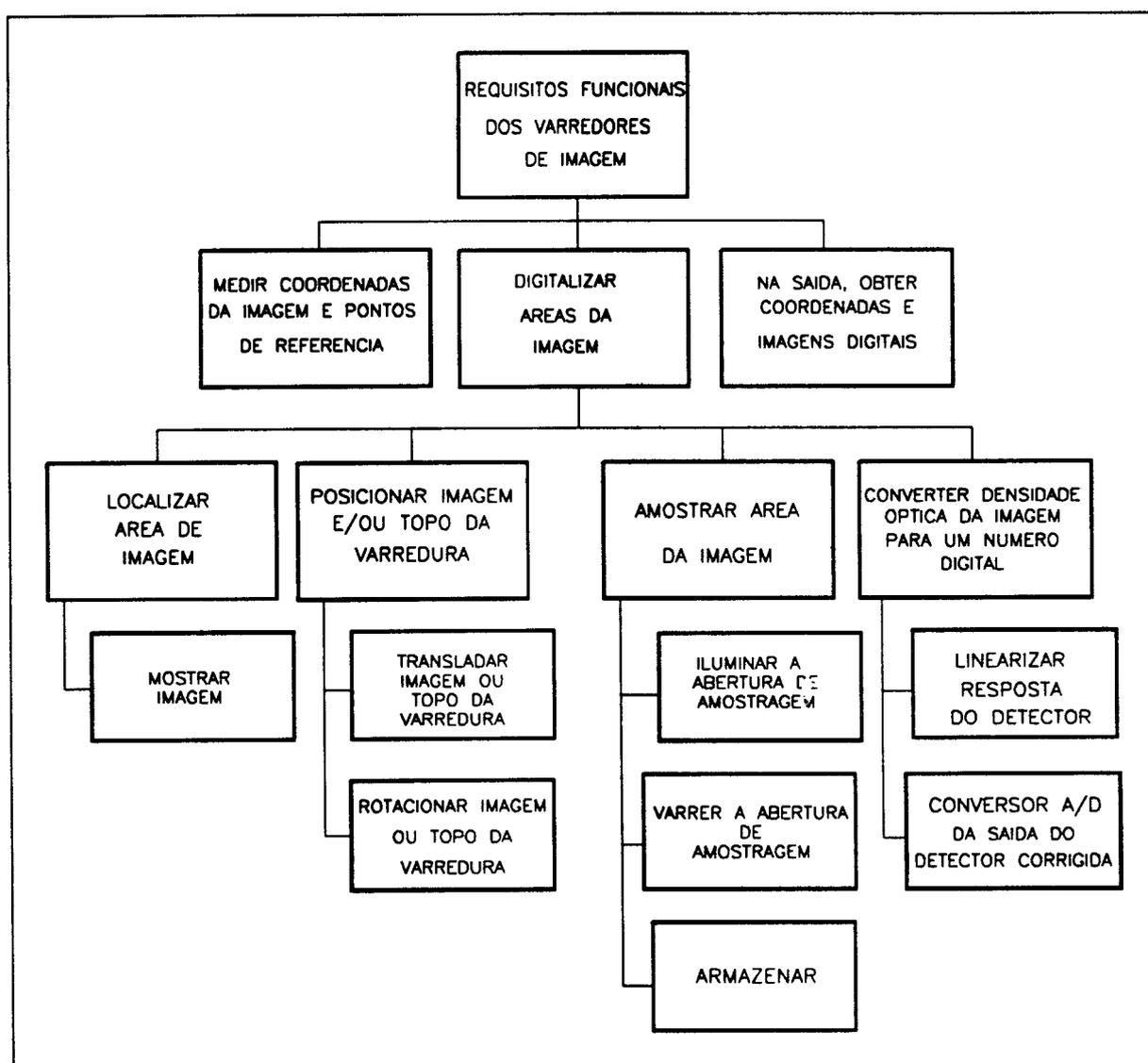


Figura 2.2 - Requisitos Funcionais dos Varredores de Imagem

Das funções principais, digitalizar áreas da imagem em questão é o requisito mais

importante, e temos associado a ele as etapas de localização, posicionamento e amostragem da área da imagem e também a conversão da densidade óptica da imagem em um número digital.

2.3 - Aplicações dos Varredores de Imagens.

Segundo {NA 83}, varredores ópticos ou digitalizadores de imagem, são usados para converter uma figura em um conjunto armazenado de números, representando a posição da distribuição da densidade óptica da figura. Eles são utilizados em :

- caráter de reconhecimento óptico
- CAD ("Computer-aided design") e desenhos
- aplicações biomédicas
- processamento de dados geográficos, incluindo leitura remota
- comunicações por fac-símiles
- impressões e publicações
- física experimental

Aplicações em engenharia requerem a digitalização de desenhos de linhas com predominância de segmentos de linha reta, tal como alguns detalhes de desenho de montagem, desenho estrutural e de arquiteturas; diagramas lógicos, circuitos e diagramas de ligações; circuitos impressos, pastilhas e esquemas de "chips". A utilização dos varredores ópticos, neste caso, se aplica para desenhos em preto e branco.

Muitas aplicações em engenharia requerem um varredor capaz de tratar uma ampla série de desenhos de vários tamanhos e uma mistura de linhas e informação alfabética.

Na área biomédica, equipamentos, tais como tomógrafos, também podem ser considerados como sistemas digitalizadores de imagens. Aqui, a transmitância de partes do corpo para o raio-X ou ultrassom em várias direções, é convertida diretamente para forma digital para subseqüente reconstrução em 2 e 3 dimensões. Equipamentos menos especializados, baseados usualmente em tecnologia de câmera de televisão, são usados para converter filme radiográfico ou seções de histologia para investigação e diagnóstico.

Em muitas aplicações geográficas, a cor é um elemento muito importante. Varredores multiespectrais e de câmeras de vidicon são usados para obter representações digitais de segmentos grandes da superfície terrestre.

Porém, uma das maiores contribuições da tecnologia dos varredores ópticos é na indústria de impressão. Desde que ilustrações coloridas estão sendo produzidas como uma sequência de impressões com impressão de tinta de várias cores nas gravuras, varredores grandes e de alta velocidade foram desenvolvidos há muitos anos atrás como preparação para separação de cores das gravuras, cartas impressas, e impressões em "offset". Alguns desses instrumentos analógicos, simplesmente reproduzem cada componente colorido da ilustração desejada; a adição de um conversor analógico para digital (A/D) transforma-o então, em dispositivos digitalizadores de imagens mais rápidos. Com o avanço das impressoras matriciais digitais de alta resolução e a rápida introdução de computadores nos outros segmentos da indústria, muitas operações de publicações estão sendo melhoradas com a tecnologia digital.

Transceptor fac-símile analógico tem sido muito usado comercialmente há muitos anos, na transmissão de imagens em preto e branco através da rede telefônica. Os varredores ópticos utilizados neste tipo de aplicação, geralmente tem uma razão de amostragem espacial bastante baixa (três a oito elementos por milímetro), mas a tendência é aumentar esta razão de amostragem e fazer transmissão digital dos dados através da rede.

Observações fotográficas são usadas em física, química e biologia, onde os varredores ópticos podem incrementar a precisão e eliminar o trabalho de digitalização manual de gráficos. Uma outra grande aplicação é a digitalização de dados da : câmara de bolhas, câmara de neblina e câmara de explosão, experiências físicas com alta energia. Varredores são também utilizados em astronomia, espectroscopia, cristalografia e fotomicrografia.

2.4 - Classificação dos Varredores.

{MO 80} descreve que em todo sistema de digitalização de imagens, a luz transmitida ou refletida de uma imagem iluminada é projetada sobre uma superfície fotosensível (fotocélula) que produz um sinal elétrico relativo à variação da intensidade luminosa sobre

uma seção da superfície. O sinal é então processado, amostrado e passa por um conversor analógico para digital (A/D) para quantizá-lo e produz um fluxo de bits digital.

Os varredores podem ser agrupados em três classes baseadas na tecnologia utilizada para iluminar a figura a ser varrida, ou no método de detecção da energia transmitida pela entrada uniformemente iluminada.

As três classes são as seguintes :

- **Varredores Eletrônicos** : podem usar dispositivos de tubo de raios catódicos (CRT) ou dispositivos tipo vidicon (tipo tubo de televisão).

- **Varredores Eletro-ópticos** : podem usar lasers, diodos emissores de luz (LED) ou lâmpadas convencionais como fontes de iluminação.

- **Varredores de Estado-sólido** : podem usar dispositivos de carga acoplada (CCD), dispositivos de carga injetada (CID), fotodiodos de carga-acoplada (CCPD) ou fotodiodo "self-scanned" (SSPD).

2.4.1 - Evolução dos Varredores Tipo Tambor Rotatório.

Foi realizado um estudo dos varredores eletro-ópticos do tipo com tambor rotatório, que representam o Photoscan utilizado neste projeto, procurando-se extrair, em síntese, as características principais de funcionamento e de construção, desse tipo de varredor.

No varredor descrito por {AB 66}, um filme de raio-X é fixo em um cilindro, que, através de um motor, fazia este rotacionar em torno de seu eixo central. Assim, o filme rotacionava e transladava no sentido longitudinal de seu eixo central, permitindo que linhas consecutivas do filme fossem varridas.

Um microscópio era utilizado para focalizar o feixe de luz, de seção retangular, sobre o filme, que depois incidia sobre uma fotomultiplicadora. O sinal que vinha da fotomultiplicadora passava por um conversor analógico para digital (A/D), que convertia o sinal elétrico em um sinal digital, que era então processado por um computador externo.

{XU 69} descreve um sistema varredor digital, que foi desenvolvido para fazer o processamento das difrações de raio-X em filmes.

Este era do tipo tambor rotatório e estava acoplado à um pequeno computador. Por

causa da rapidez e facilidade de acesso aos dados, gravados na memória do computador, a imagem do filme podia ser processada eficientemente, e automaticamente avaliada por um programa de computador.

O sistema era similar ao descrito por {AB 66}, com exceção de algumas mudanças feitas no projeto de varredura dos filmes, onde houve uma melhora, tanto na precisão do posicionamento do filme, bem como na medida da densidade óptica.

O princípio de varredura adotado neste equipamento era o seguinte : depois de cada revolução completa do tambor, a fonte de luz e o detector, transladavam no sentido longitudinal do eixo central para fazer a varredura de uma outra linha do filme; enquanto que em {AB 66} havia a translação do tambor com o filme.

Mas, o mais importante, era que o processamento da imagem podia ser feito com mais versatilidade, pois o varredor estava conectado a um computador, cuja memória era suficiente para armazenar toda a imagem do filme.

{NO 70} apresenta um instrumento que é similar àqueles descritos por Abrahamson {AB 66} e por Xuong {XU 69}. Neste, a luz transmitida através do filme possuía um feixe de diâmetro variável de 50, 100 ou 200 μm .

O tambor rotacionava 1, 2 ou 4 revoluções por segundo e a luz transmitida pelo filme, era medida para cada deslocamento de 50, 100 ou 200 μm na direção de rotação.

Depois de cada revolução do tambor, a fonte de luz deslocava-se de 50, 100 ou 200 μm ao longo do eixo central do tambor para fazer a varredura de uma nova linha do filme.

{MA 72} descreve um varredor do tipo tambor rotatório, controlado por um computador, utilizado para medida da difração de raio-X em filmes fotográficos, para estudos cristalográficos de proteínas.

A vantagem deste sistema sobre os descritos por {AB 66}, {XU 69} e {NO 70} é que neste caso, um computador controla o varredor e as densidades são obtidas de forma "on-line".

Já nos sistemas descritos anteriormente, fazia-se a varredura do filme sequencialmente, e as densidades eram armazenadas em um dispositivo intermediário (memória) de grande capacidade para serem processadas posteriormente, e assim obter as intensidades integradas. Este tipo de sistema, chamado "off-line", em geral não é possível verificar os dados amostrados durante a execução das medidas das densidades do filme, e desta maneira não se pode otimizar o modo de leitura.

O instrumento descrito por {MA 72}, consiste de um varredor digital do tipo Photoscan modelo B, interfaceado com um computador Varian 620i.

O filme é montado num tambor que rotaciona 4 revoluções por segundo. O sistema óptico e o detector, são montados em uma rosca-sem-fim para que possam transladar na direção axial de rotação do tambor. A rosca-sem-fim é controlada por um motor de passo, onde cada incremento (passo) no filme corresponde à 12,5 μm . Um conversor analógico para digital (A/D) de 8 bits é utilizado para que os dados sejam convertidos em densidades ópticas, e assim, colocados no computador.

As rotinas que comandam o densitômetro foram feitas na linguagem de programação FORTRAN.

A figura 2.3 ilustra um varredor tipo tambor rotatório.

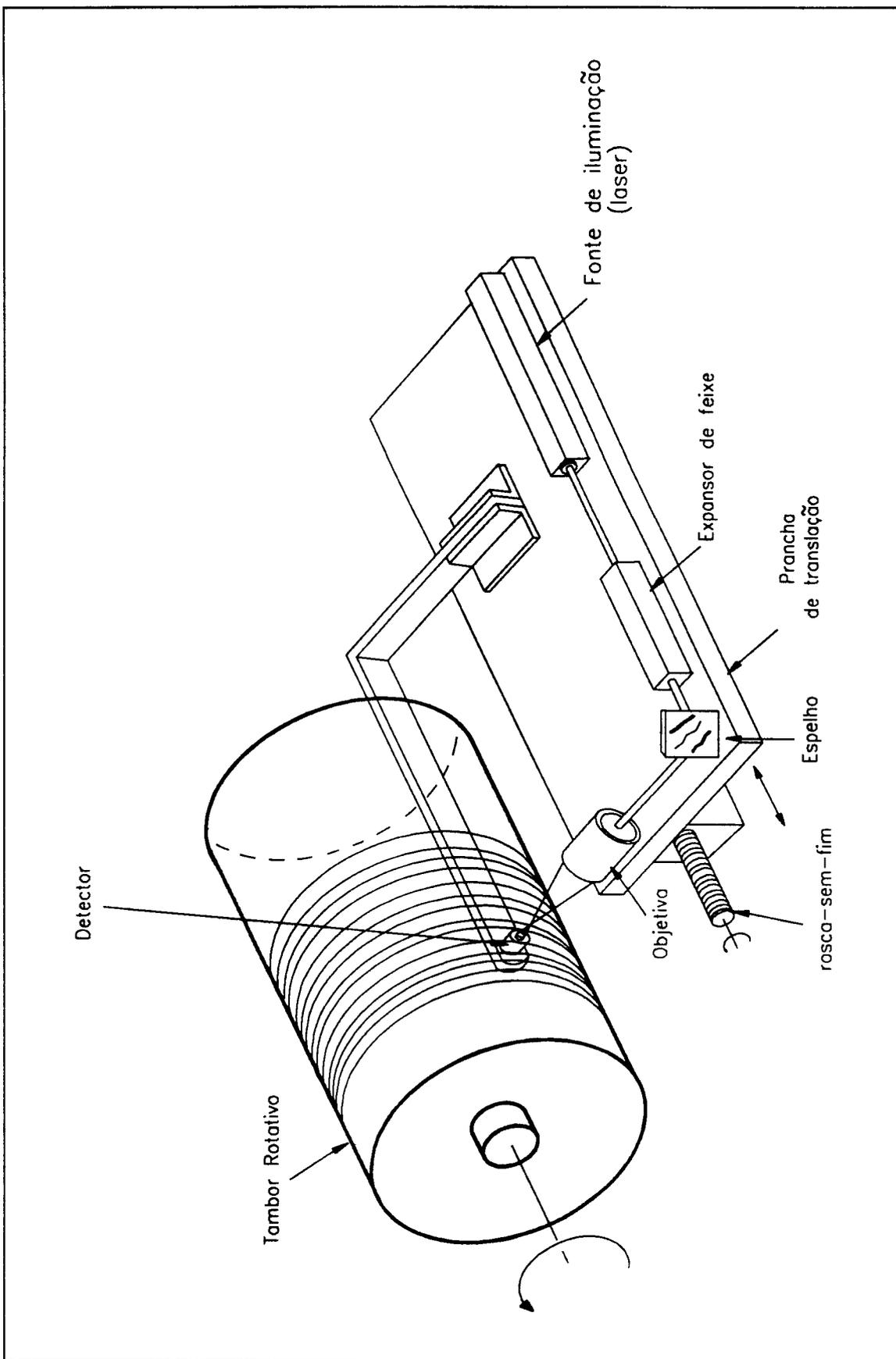


Figura 2.3 - Varredor Tipo Tambor Rotatório

Capítulo 3

HARDWARE

3.1 - Interface Paralela Desenvolvida.

A figura 3.1 ilustra o esquema da interface desenvolvida.

No circuito da figura 3.1 temos dois componentes DM74LS245N descrito em {TTL 76} e que são usados para armazenamento temporário e para fazer a comunicação entre os dutos de dados do microcomputador PC e as 8255, sendo que um deles envia sinais de controle para as 8255.

Um monoestável SN74LS123N {TTL 76} foi utilizado para que o tempo das aquisições dos dados realizadas pelo microcomputador se tornasse viável, compatibilizando especificações de temporização do Photoscan. Uma descrição de seu funcionamento, bem como o software necessário para a aquisição dos dados, são discutidos no capítulo 4.

Utilizou-se um circuito integrado PAL10L8 ("Programmable Array Logic - PAL"), que é descrito por Birkner e Coli {BI 81}, como sendo um dispositivo que implementa uma soma de produtos lógicos, através da utilização de um conjunto programável de portas "AND", cujos termos de saída alimentam um conjunto fixo de portas "OR". Sendo que a forma da soma dos produtos pode expressar qualquer função de transferência Booleana. Assim, as PALs são utilizadas por um número limitado de termos disponíveis no conjunto de portas "AND-OR".

A PAL pode ser programada, e durante a programação, metade das saídas da PAL são selecionadas para programação, enquanto as outras saídas e entradas, são usadas para endereçamento. As saídas são então ligadas para programar as outras posições. Para isto, utiliza-se um software especial que define, simula, programa e testa as PALs.

Programou-se a PAL para fazer a decodificação do endereçamento das 8255, habilitando-se então, uma das 8255 através do "chip select" (\overline{CS}).

Descreve-se abaixo, a distribuição dos pinos da PAL (ver figura 3.1), bem como as equações da programação realizada.

A2 a A10, AEN --> SAÍDAS DO BARRAMENTO DO PC
(ENTRADAS DA PAL)

S0 a S6, MS --> SAÍDAS DA PAL

Convenção : / = nega

* = AND

PAL10L8

A2 A3 A4 A5 A6 A7 A8 A9 A10 GND pinos 1 a 10
AEN S6 S5 S4 S3 S2 S1 S0 MS VCC pinos 11 a 20

$/S6=/A10*A9*A8*/A7*/A6*/A5*A4*A3*/A2*/AEN$
 $/S5=/A10*A9*A8*/A7*/A6*/A5*A4*/A3*A2*/AEN$
 $/S4=/A10*A9*A8*/A7*/A6*/A5*A4*/A3*/A2*/AEN$
 $/S3=/A10*A9*A8*/A7*/A6*/A5*/A4*A3*A2*/AEN$
 $/S2=/A10*A9*A8*/A7*/A6*/A5*/A4*A3*/A2*/AEN$
 $/S1=/A10*A9*A8*/A7*/A6*/A5*/A4*/A3*A2*/AEN$
 $/S0=/A10*A9*A8*/A7*/A6*/A5*/A4*/A3*/A2*/AEN$
 $/MS=/A10*A9*A8*/A7*/A6*/A5*/AEN$

Duas 8255 foram utilizadas na interface : uma serve como comunicação para os sinais de entrada para o Photoscan e a outra para os sinais de saída do Photoscan.

No Apêndice A encontra-se uma descrição do Photoscan Optronics P-1000 e as especificações da sua interface.

Descreve-se no Apêndice B a interface de comunicação paralela programável 8255 utilizada no hardware desenvolvido.

O endereçamento das portas das 8255, os endereços de programação e as respectivas palavras de controle, são descritas a seguir :

8255 (1)	8255 (2)
PortaA1 - 300 H	PortaA2 - 304 H
PortaB1 - 301 H	PortaB2 - 305 H
PortaC1 - 302 H	PortaC2 - 306 H
Endereço de programação - 303 H	Endereço de programação - 307 H
Palavra de controle - 80 H	Palavra de controle - BF H

A tabela 3.1, ilustra como os pinos de entrada do Photoscan são ligados nas portas da 8255 (1) e na tabela 3.2 como os pinos de saída do Photoscan são ligados nas portas da outra 8255 (2).

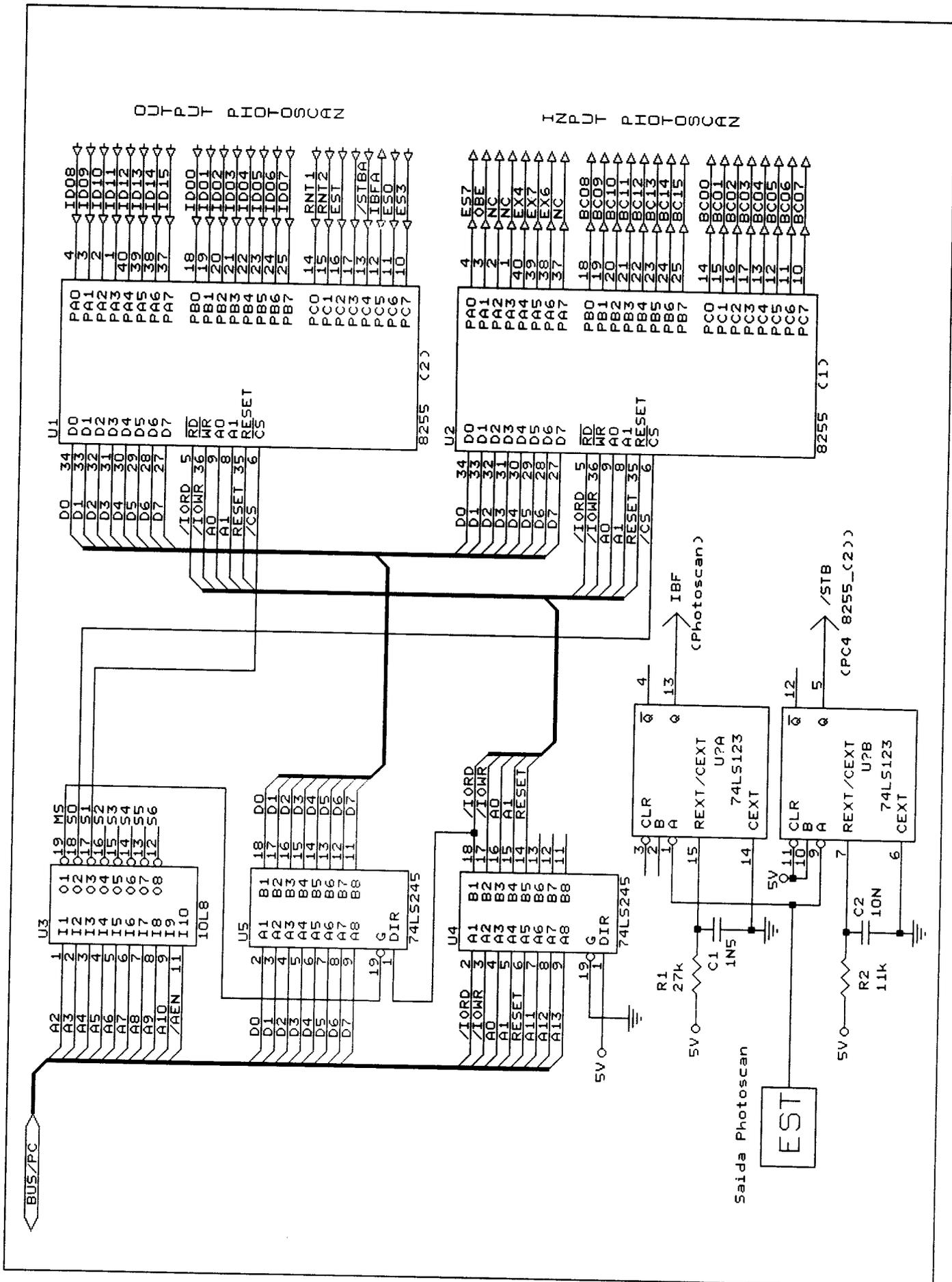


Figura 3.1 - Esquema da Interface Desenvolvida

Porta	Pinos da 8255 (1)	Nome do sinal de Entrada
A	PA0	ES7
	PA1	OBE
	PA2	N.C.
	PA3	N.C.
	PA4	EX4
	PA5	EX7
	PA6	EX6
	PA7	N.C.
B	PB0	BC08
	PB1	BC09
	PB2	BC10
	PB3	BC11
	PB4	BC12
	PB5	BC13
	PB6	BC14
	PB7	BC15
C	PC0	BC00
	PC1	BC01
	PC2	BC02
	PC3	BC03
	PC4	BC04
	PC5	BC05
	PC6	BC06
	PC7	BC07

Tabela 3.1 - Distribuição dos pinos de Entrada do Photoscan na 8255 (1)

Porta	Pinos da 8255 (2)	Nome do sinal de Saída
A	PA0	ID08
	PA1	ID09
	PA2	ID10
	PA3	ID11
	PA4	ID12
	PA5	ID13
	PA6	ID14
	PA7	ID15
B	PB0	ID00
	PB1	ID01
	PB2	ID02
	PB3	ID03
	PB4	ID04
	PB5	ID05
	PB6	ID06
	PB7	ID07
C	PC0	RNT1
	PC1	RNT2
	PC2	EST
	PC3	INTR
	PC4	/STB
	PC5	IBF
	PC6	ES0
	PC7	ES3

Tabela 3.2 - Distribuição dos pinos de Saída do Photoscan na 8255 (2)

Capítulo 4

SOFTWARE

4.1 - Introdução.

O software desenvolvido para controlar o Photoscan Optronics P-1000, foi construído em duas partes, Rotinas para Controle do Photoscan e Programa Principal, que fornece a interface ao usuário.

O Programa Principal e a maioria das rotinas foram desenvolvidas em linguagem PASCAL, pois esta possibilita uma programação mais fácil para a compreensão do usuário. A linguagem PASCAL, permite ainda a utilização de unidades compiladas (UNIT), descritas por Hergert em (HE 89), onde estas contêm rotinas que são utilizadas pelo programa principal ou por outras rotinas.

Um diagrama em blocos das principais funções do software desenvolvido é mostrado na figura 4.1.

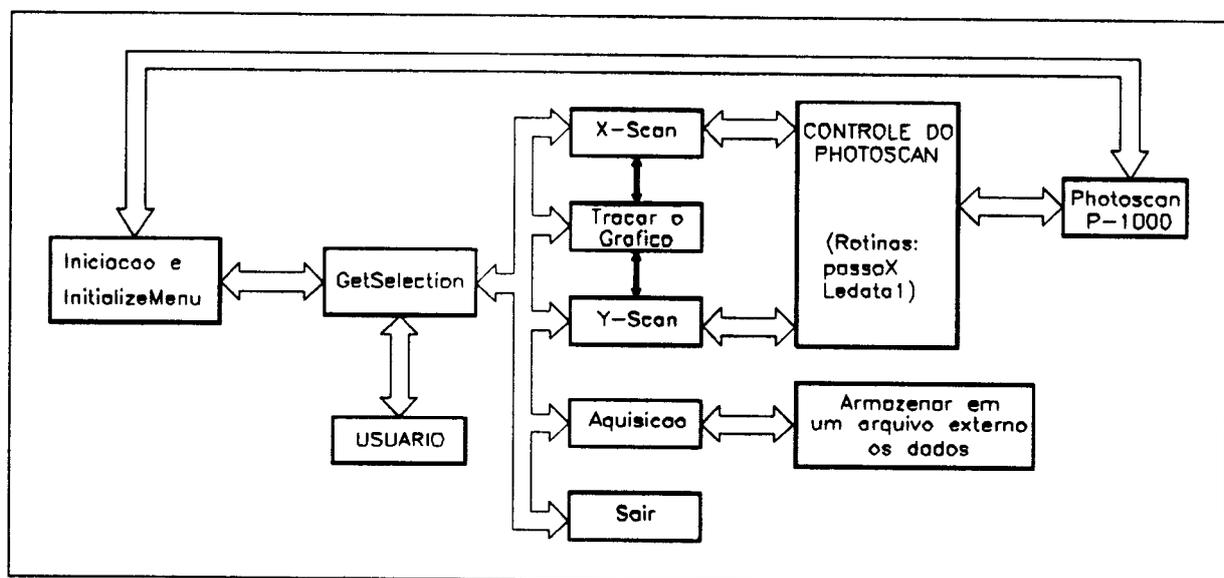


Figura 4.1 - Diagrama em blocos do software desenvolvido

Na primeira parte do programa desenvolvido é fornecido ao usuário um menu (rotina `InitializeMenu`) onde este possa optar por fazer a digitalização de uma linha do filme, de uma coluna ou fazer a aquisição dos dados digitalizados, ou seja, armazenar estes dados em um arquivo e, uma última opção serve para terminar o programa. A seleção de uma das opções descritas acima é feita através da rotina `GetSelection`.

A segunda parte, contendo as Rotinas de Controle do Photoscan, faz a programação das portas das 8255 utilizadas na interface do PC (rotina `Init55`), determina a resolução ("Raster") (rotina `GetRaster`) que está selecionada para o Photoscan (Iniciação) e, as outras rotinas desenvolvidas servem para movimentar o subsistema óptico do Photoscan para direita ou esquerda (rotina `passoX`) e fazer a leitura fotométrica do filme (densidade) (rotina `Ledata1`).

Todas estas rotinas estão contidas na unidade `Scan_Unit` utilizada pelo programa principal, com exceção da que faz a leitura dos dados que se encontra declarada na unidade `SelecaoUnit`, como uma rotina externa (`External`).

A rotina que faz a leitura dos dados do filme (densidades) foi desenvolvida em linguagem Montadora, devido a rapidez com que os dados são lidos pelo Photoscan e enviados ao microcomputador, ou seja, para viabilizar as especificações de temporização do Photoscan.

Uma listagem do programa desenvolvido, encontra-se no Apêndice C.

4.2 - Programa Principal.

São as seguintes as unidades desenvolvidas e utilizadas no programa principal :

- `MenuUnit` : contém as rotinas relacionadas com a exibição de um menu repetitivo na tela.
- `Scan_Unit` : contém as rotinas relacionadas diretamente com o Photoscan.
- `CursorUnit` : contém as rotinas relacionadas com o ajuste da curva (gráfico).
- `SelecaoUnit` : contém a rotina relacionada com a seleção de uma das opções do menu principal e a rotina `Ledata1`, declarada como `external`.

A figura - 4.2 mostra o fluxograma do programa principal.

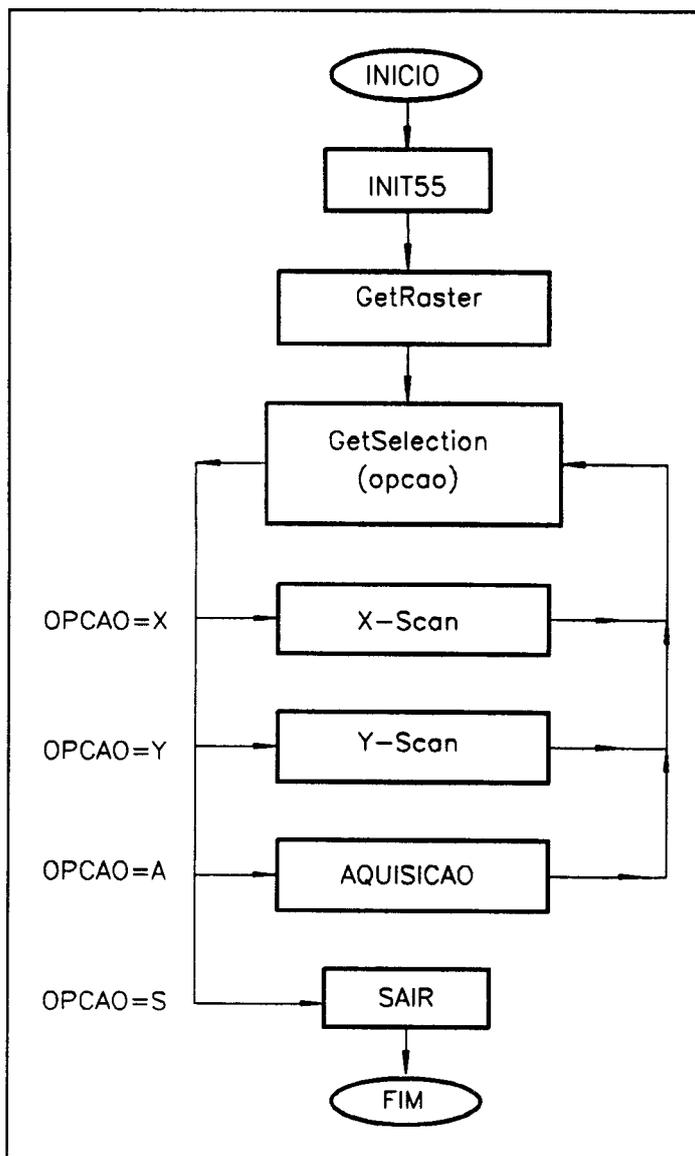


Figura 4.2 - Fluxograma do programa principal

Inicialmente, este programa faz a iniciação ou programação das portas das 8255. A seguir, é chamada a rotina GetRaster que verifica e obtém o valor da resolução ("Raster") selecionada. Depois, faz-se a chamada à rotina InitializeMenu, para exibir o menu na tela com as quatro opções disponíveis :

- X-scan - faz a digitalização de uma linha do filme
- Y-scan - faz a digitalização de uma coluna do filme
- Aquisicao - armazena os dados (densidades) da linha ou coluna digitalizada em um

arquivo

- Sair - fim do programa

Logo após, o programa faz repetidas chamadas à rotina GetSelection, onde o usuário deve fazer a seleção de uma das opções disponíveis. Cada opção escolhida é executada, e sempre retorna à rotina GetSelection. Novamente o usuário deve escolher uma das opções, exceto se a escolhida anteriormente for a opção S, pois na execução desta opção o programa termina.

No Apêndice D, encontram-se as instruções para utilização do software desenvolvido.

4.2.1 - Subrotina GetSelection.

A função desta rotina é selecionar uma das opções disponíveis no menu, escolhida pelo usuário, e executá-la.

Para fazer a seleção de uma das opções, o usuário deve pressionar a tecla da seta para cima ou tecla da seta para baixo, até que o destaque se mova para a opção desejada, ou pressionar, ainda, a primeira letra do nome da opção : X, Y, A ou S e completar a seleção pressionando a tecla ENTER. As opções são as seguintes :

X-scan - o usuário entra com a posição da linha (start) (em mm) que será digitalizada; a posição inicial da tomada dos dados (coluna) (em mm) e a posição final da tomada dos dados (pos_final) (em mm).

Com isto o subsistema óptico será deslocado para direita ou para esquerda, até a posição inicial onde será iniciada a tomada dos dados, no caso até o valor da coluna (rotina passoX). Além disto, toda vez que um dado vai ser coletado, deve-se saltar os pontos (ptos_saltar) até chegar na linha a ser digitalizada (start).

Faz-se então, chamadas a rotina Ledatal até que o número de pontos a serem coletados seja atingido, sendo que após cada chamada a rotina, o subsistema óptico é deslocado para direita ou para esquerda (rotina passoX), onde a distância (ou comprimento) deste deslocamento depende da resolução selecionada e cada passo deslocado é de 12,5 μm . Os pontos da coluna (start) são então, novamente pulados até chegar na linha de interesse.

Traça-se o gráfico das densidades pela distância X (valor da resolução) entre cada

ponto, onde o usuário tem ainda a opção de ampliar ("zoom") regiões do gráfico.

Para selecionar a região de interesse, o usuário irá movimentar dois cursores que se encontram nos extremos do gráfico; movimento este realizado com as setas para direita ou seta para esquerda, até eliminar os pontos do gráfico que não interessam.

A figura - 4.3 ilustra o tambor do Photoscan com um filme onde estão indicadas as posições descritas anteriormente.

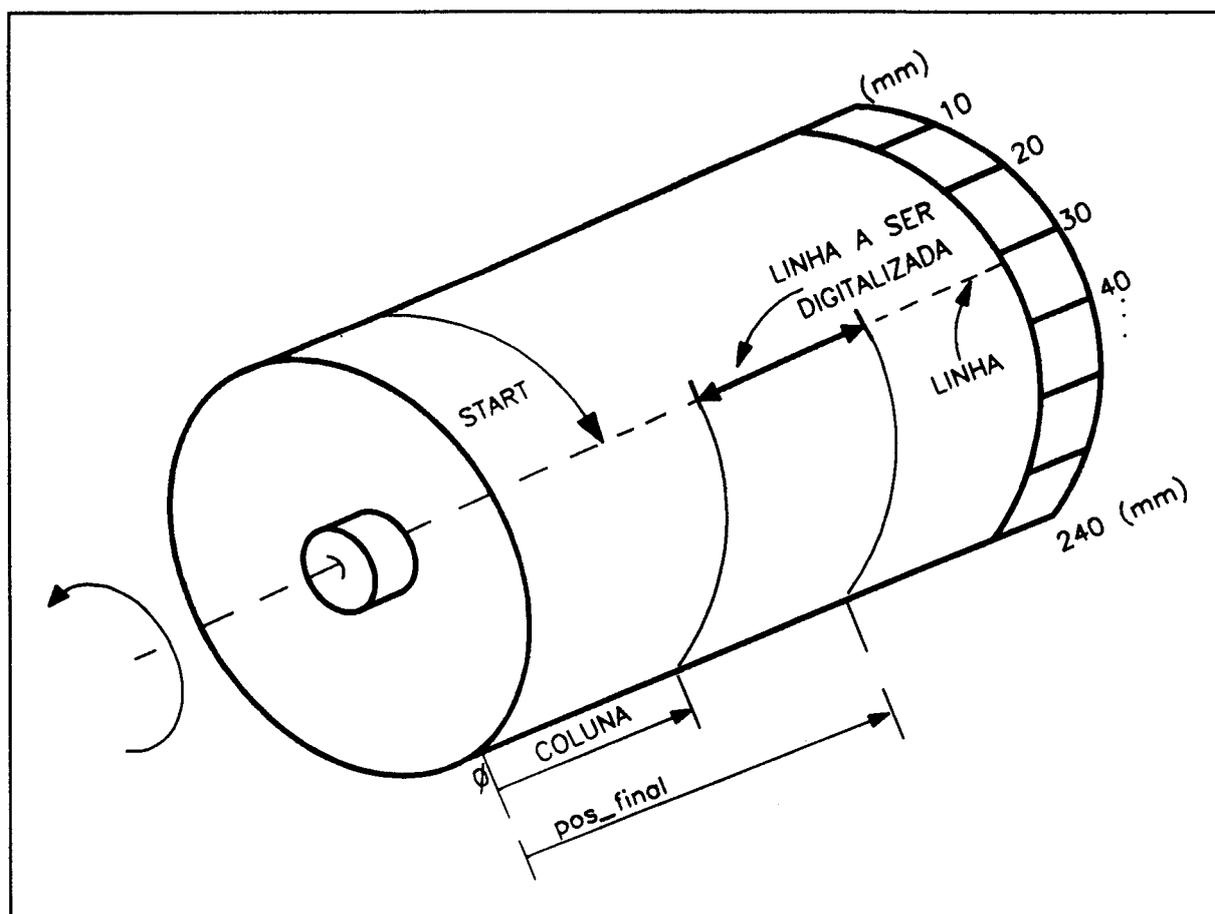


Figura 4.3 - Ilustração das posições da linha a ser digitalizada

Y-scan - para digitalizar uma coluna do filme, o usuário entra com a posição da coluna (coluna) (em mm) a ser digitalizada; a posição inicial (start) (em mm) da tomada dos dados e o tamanho da coluna (size) (em mm) a ser digitalizada.

Desta forma, o subsistema óptico será deslocado para esquerda ou para direita até que a posição da coluna a ser digitalizada seja atingida (rotina passoX). Observe, que neste caso, o usuário deve entrar com o tamanho da coluna a ser digitalizada e não com a posição final.

Antes de iniciar a coleta dos dados, deve-se saltar os pontos (ptos_saltar) da coluna a ser digitalizada até chegar na posição inicial para iniciar a digitalização da coluna (start).

Faz-se, então, uma chamada à rotina Ledatal para que os dados sejam coletados. Em seguida, traça-se o gráfico das densidades pela distância Y entre cada ponto, distância esta que depende do valor da resolução ("Raster") selecionada.

Finalmente, o usuário tem a opção de ampliar ("zoom") regiões do gráfico obtido, sendo esta ampliação realizada como descrito na opção X-scan. Uma ilustração do tambor com um filme e, as posições descritas anteriormente, estão contidas na figura - 4.4.

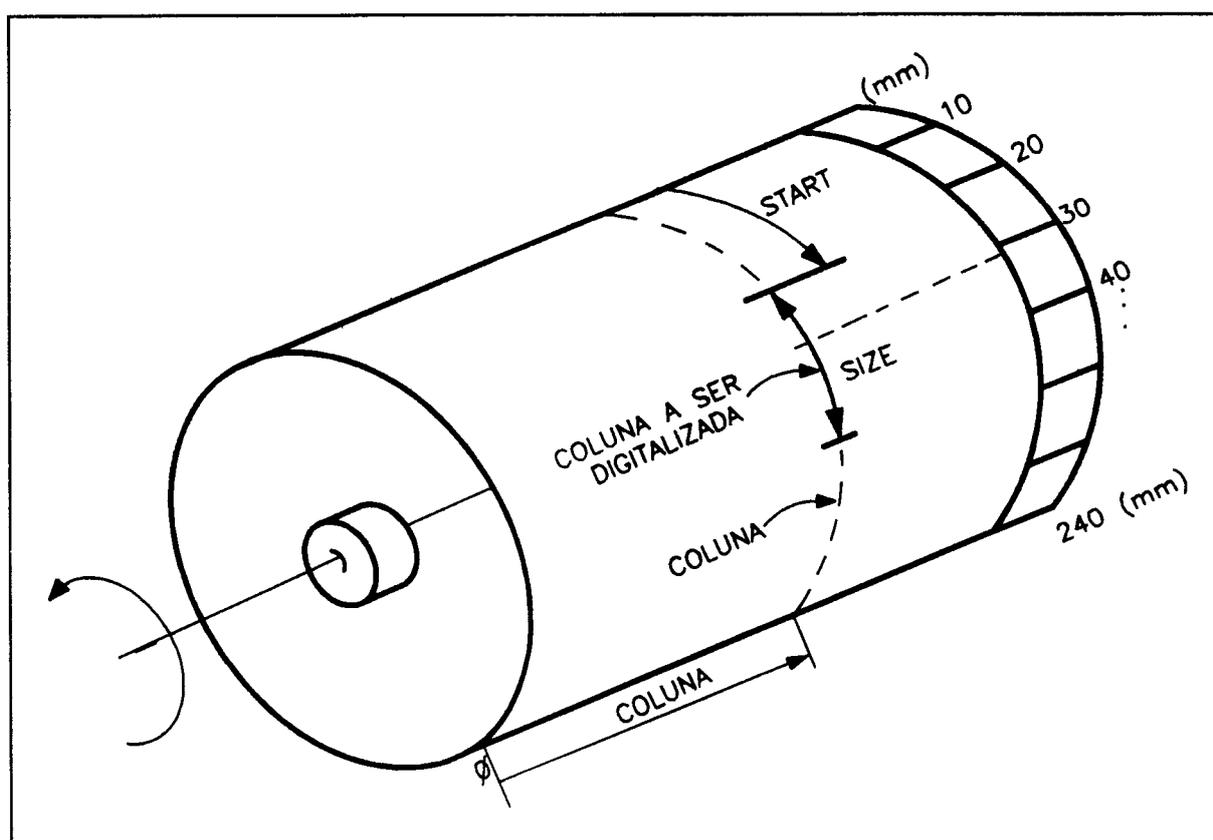


Figura 4.4 - Ilustração das posições da coluna a ser digitalizada

Aquisicao - nesta opção, o usuário poderá armazenar em um arquivo (tipo texto) os dados do filme digitalizado, bem como alguns comentários e informações sobre o filme utilizado, como por exemplo : data, se foi digitalizada uma linha ou coluna, os limites do gráfico obtido na digitalização, as coordenadas do filme onde foi feita a digitalização etc. Todas estas informações serão armazenadas em um arquivo, cujo nome será fornecido pelo usuário.

Sair - esta opção serve para terminar o programa.

O fluxograma desta rotina, encontra-se nas figuras - 4.5a, 4.5b e 4.5c.

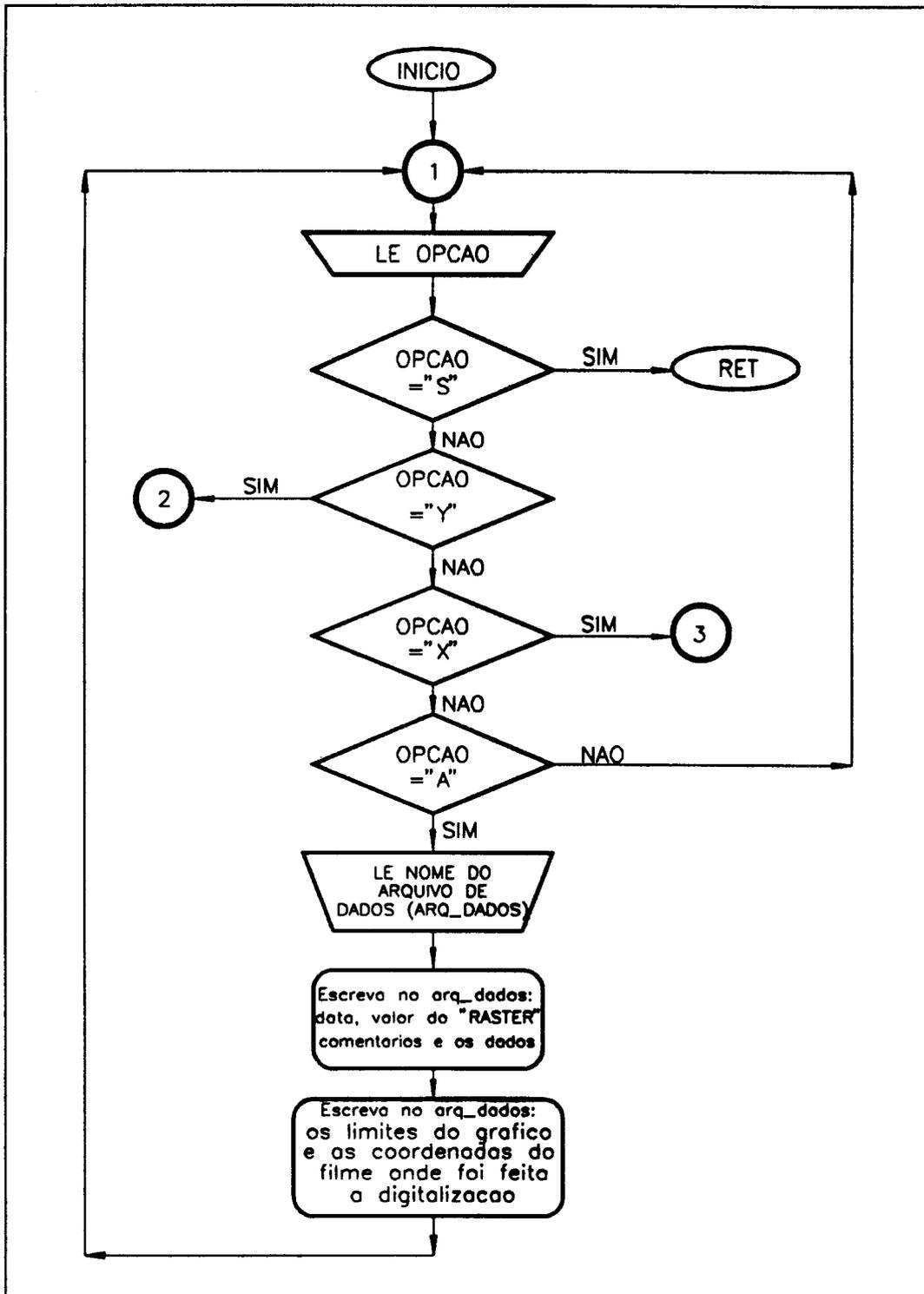


Figura 4.5a - Fluxograma da rotina GetSelection

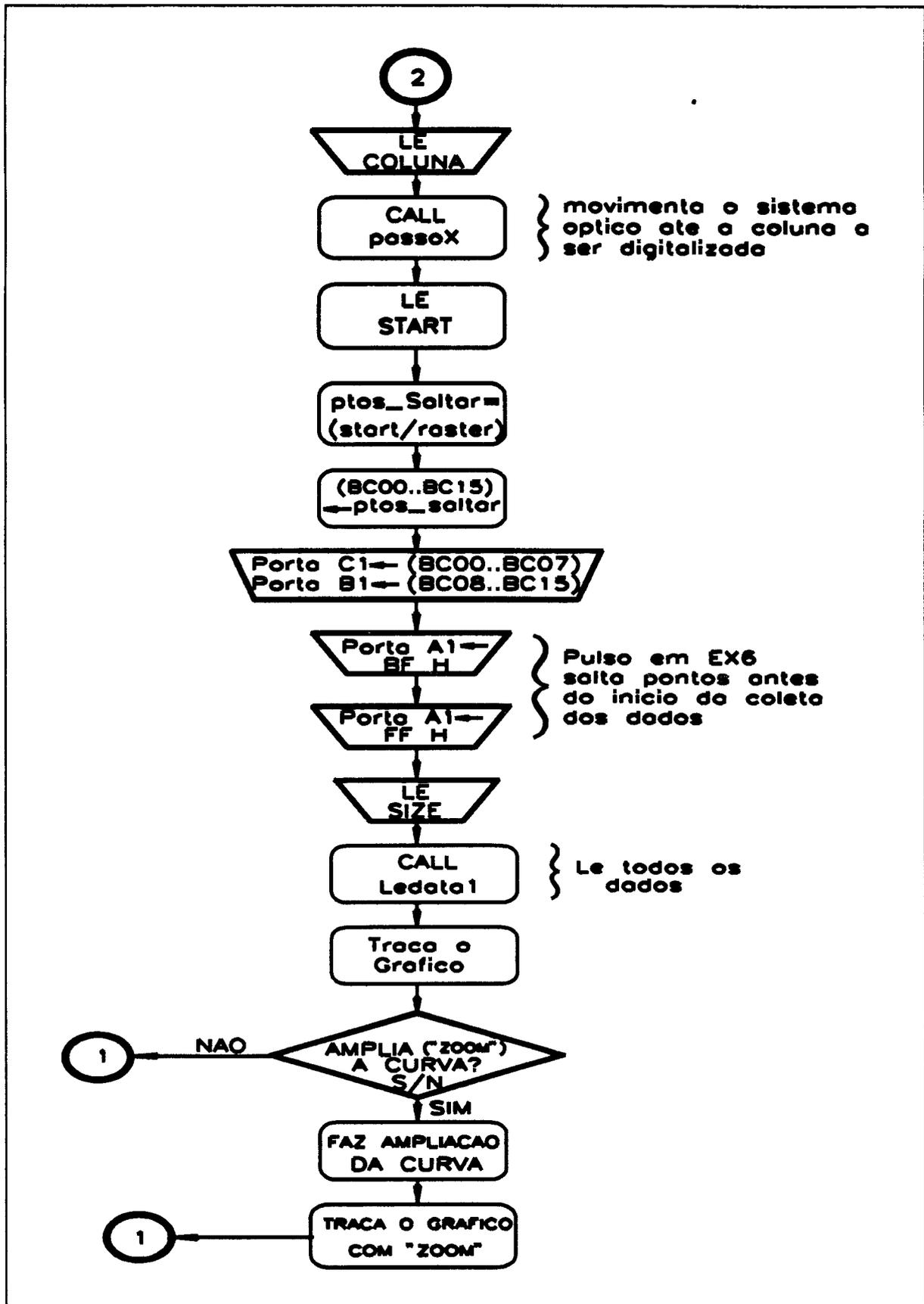


Figura 4.5b - Continuação do fluxograma da rotina GetSelection

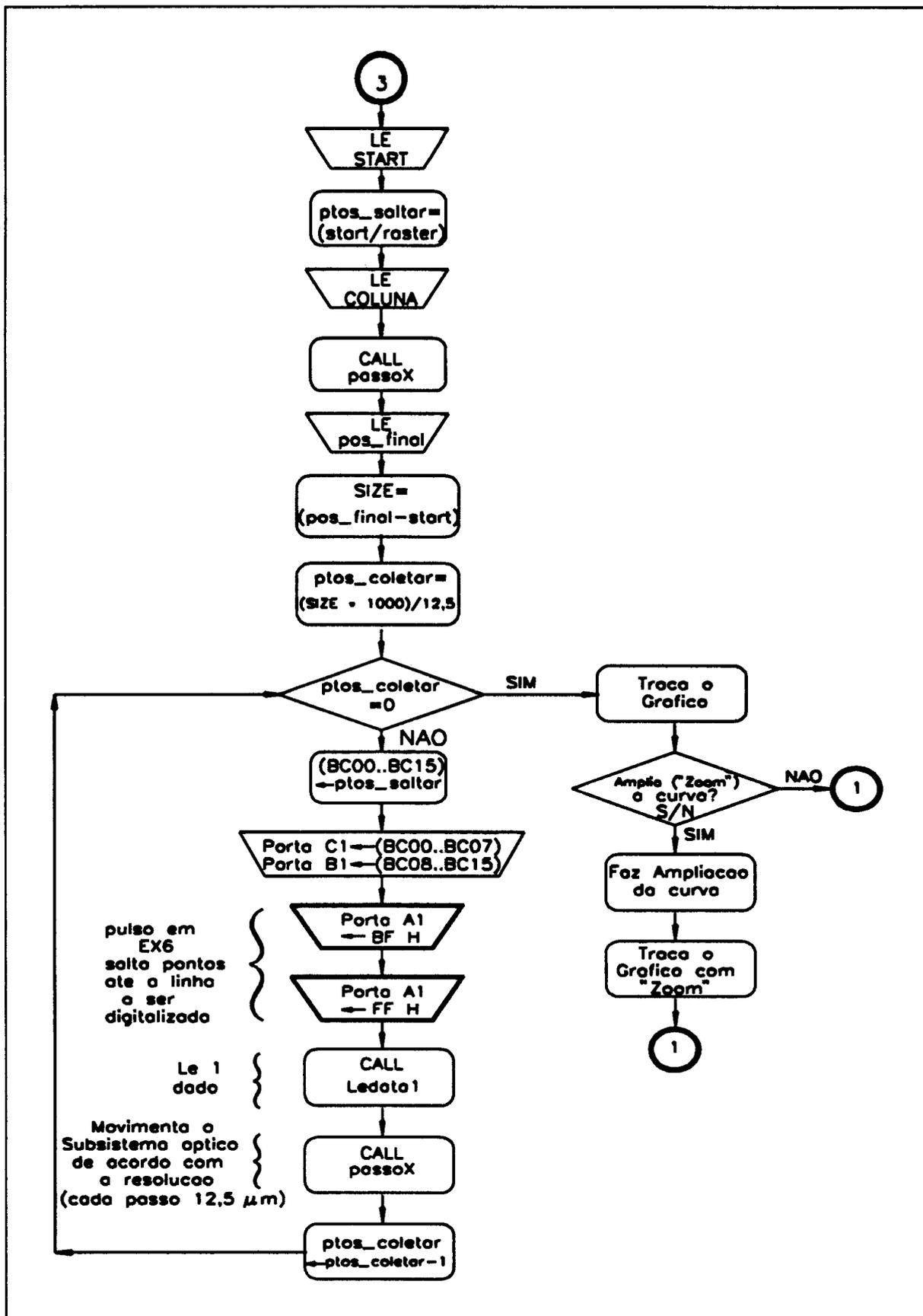


Figura 4.5c - Continuação do fluxograma da rotina GetSelection

4.3 - Rotinas do Photoscan.

4.3.1 - Programação das 8255.

Cada integrado 8255 contém três portas de 8 bits (A, B e C), sendo que cada porta deve ser programada por software como porta de entrada, ou de saída, ou como porta de controle, conforme seja conveniente (ver Apêndice B).

Cada porta permite, então, a transferência de informação do Photoscan para o microcomputador e vice-versa, conforme a maneira como for programada.

Para a 8255 (1) (Sinais de Entrada do Photoscan), o modo de operação escolhido foi o modo 0. A figura - 4.6 ilustra a configuração e a palavra de controle selecionada.

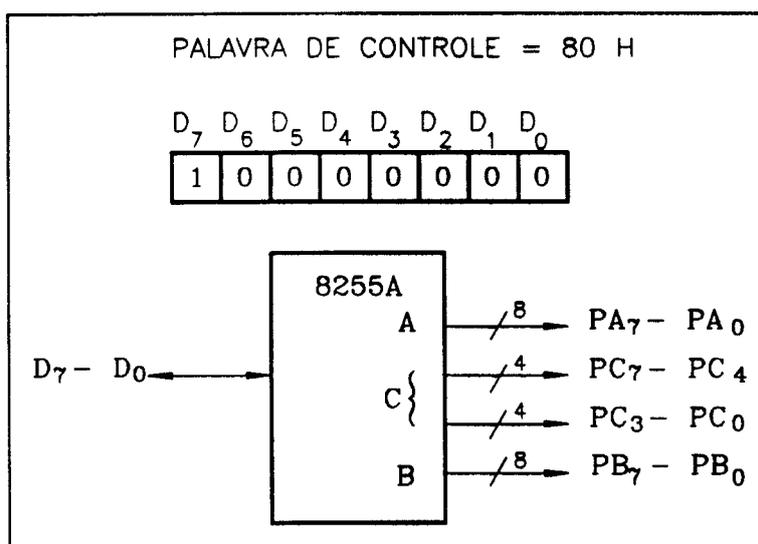


Figura 4.6 - Configuração Funcional para o 8255 (1)

No segundo integrado 8255 (2), utilizou-se o modo de operação 1 (modo 1). Na referência {IN 81} vemos que esta configuração funcional fornece uma maneira de transferência de dados de entrada/saída (I/O) para ou de uma porta especificada em associação (combinação) com sinais de controle ("handshaking"). Neste modo, as portas A e B, usam as

linhas da porta C para gerar ou aceitar estes sinais de "handshaking". A configuração selecionada, assim como a palavra de controle associada, é mostrada na figura 4.7.

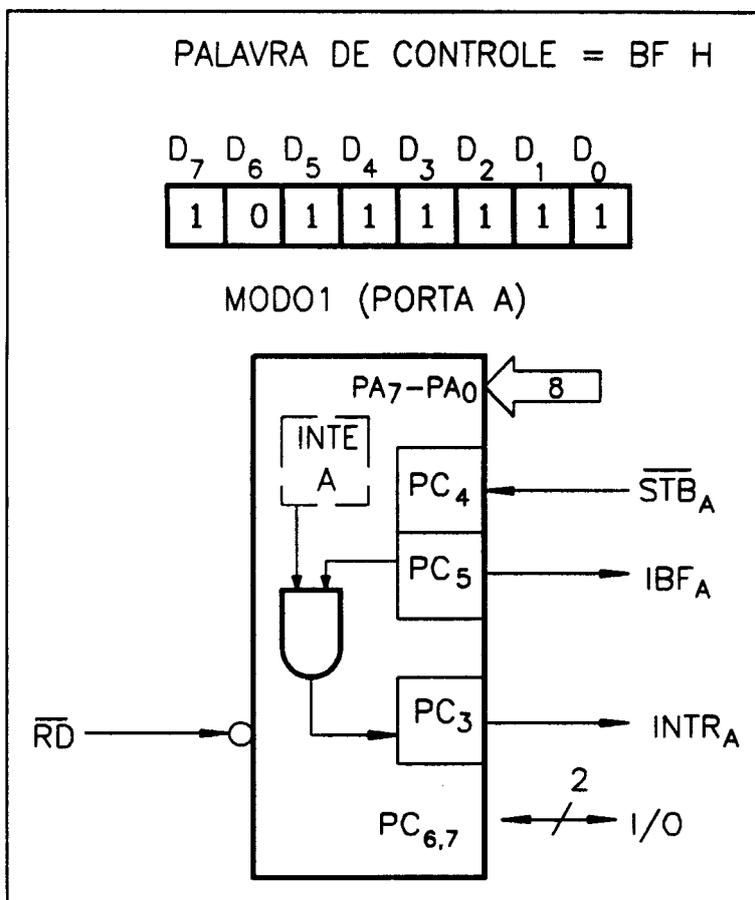


Figura 4.7 - Configuração Funcional para o 8255 (2)

Definição dos Sinais de Controle de Entrada :

- **STB ("Strobe Input")** : um nível lógico "baixo" nesta entrada carrega o dado no registro ("latch") de entrada (porta A ou B, dependendo da palavra de controle); neste caso, porta A.

- **IBF ("Input Buffer Full")** : um nível lógico "alto" nesta saída, indica que o dado foi carregado no registro de entrada; em essência, um reconhecimento. IBF é colocado em um nível lógico "alto" quando a entrada \overline{STB} está em um nível lógico "baixo" e, é reestabelecido, ou seja, colocado novamente em um nível lógico "baixo", através da borda de subida da entrada \overline{RD} (leitura).

• **INTR ("Interrupt Request")** : um nível lógico "alto" nesta saída, pode ser usado para interromper a CPU quando um dispositivo de entrada faz uma solicitação de serviço. INTR é colocado em um nível lógico "alto" quando \overline{STB} , IBF e INTE estão em um nível lógico "um" ("alto"). E este é reestabelecido através da borda de descida da entrada \overline{RD} (leitura). Este procedimento permite que um dispositivo de entrada faça um pedido de serviço da CPU, simplesmente "pulsando" (colocando) estes dados na porta de entrada (no caso a porta A).

INTE A controlado pelo bit PC4.

A figura - 4.8, ilustra o ciclo completo de carregamento de dado na Porta A ou B, escolhida de acordo com a palavra de controle.

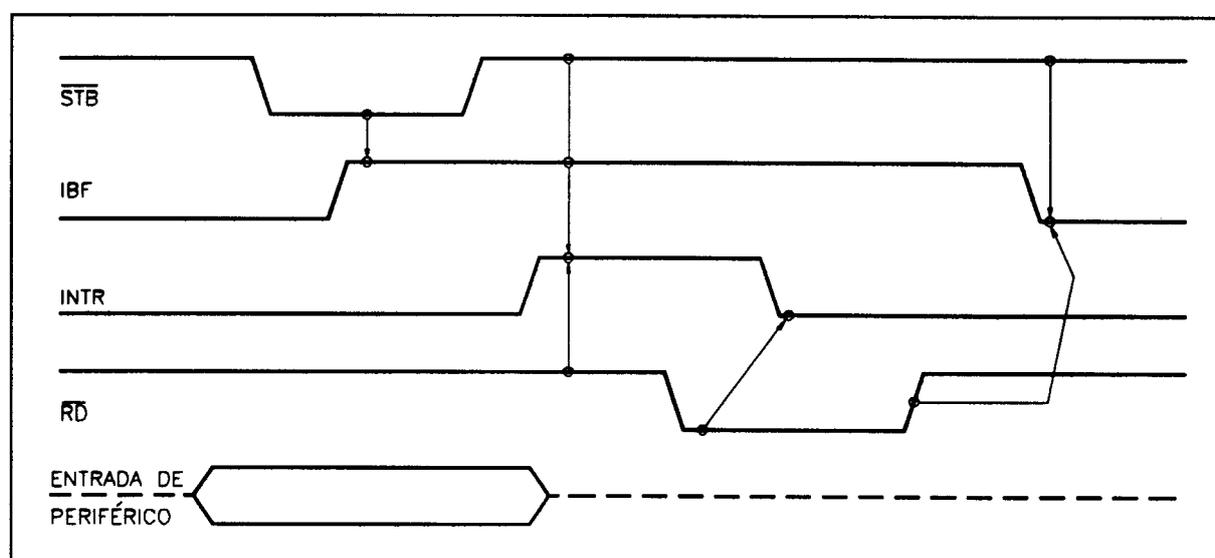


Figura 4.8 - Ciclo completo para carregar dado na porta A

4.3.2 - Obtenção do valor da Resolução ("RASTER").

O Photoscan possui 3 valores de resolução, que podem ser estendidos para 5. Estes valores determinam a velocidade de rotação do tambor bem como, a distância entre dois pontos adjacentes quando se faz a varredura (leitura das densidades) de uma coluna (Y) no filme. Quando se deseja saltar alguns pontos da coluna antes de iniciar a aquisição dos dados, a distância entre dois pontos adjacentes que devem ser pulados também é determinada por estes valores.

A tabela A.7 fornece estes valores da resolução (25, 50 e 100 μm), bem como os valores para a extensão (12,5 e 200 μm).

A figura - 4.9, mostra o fluxograma da obtenção do valor da resolução do Photoscan.

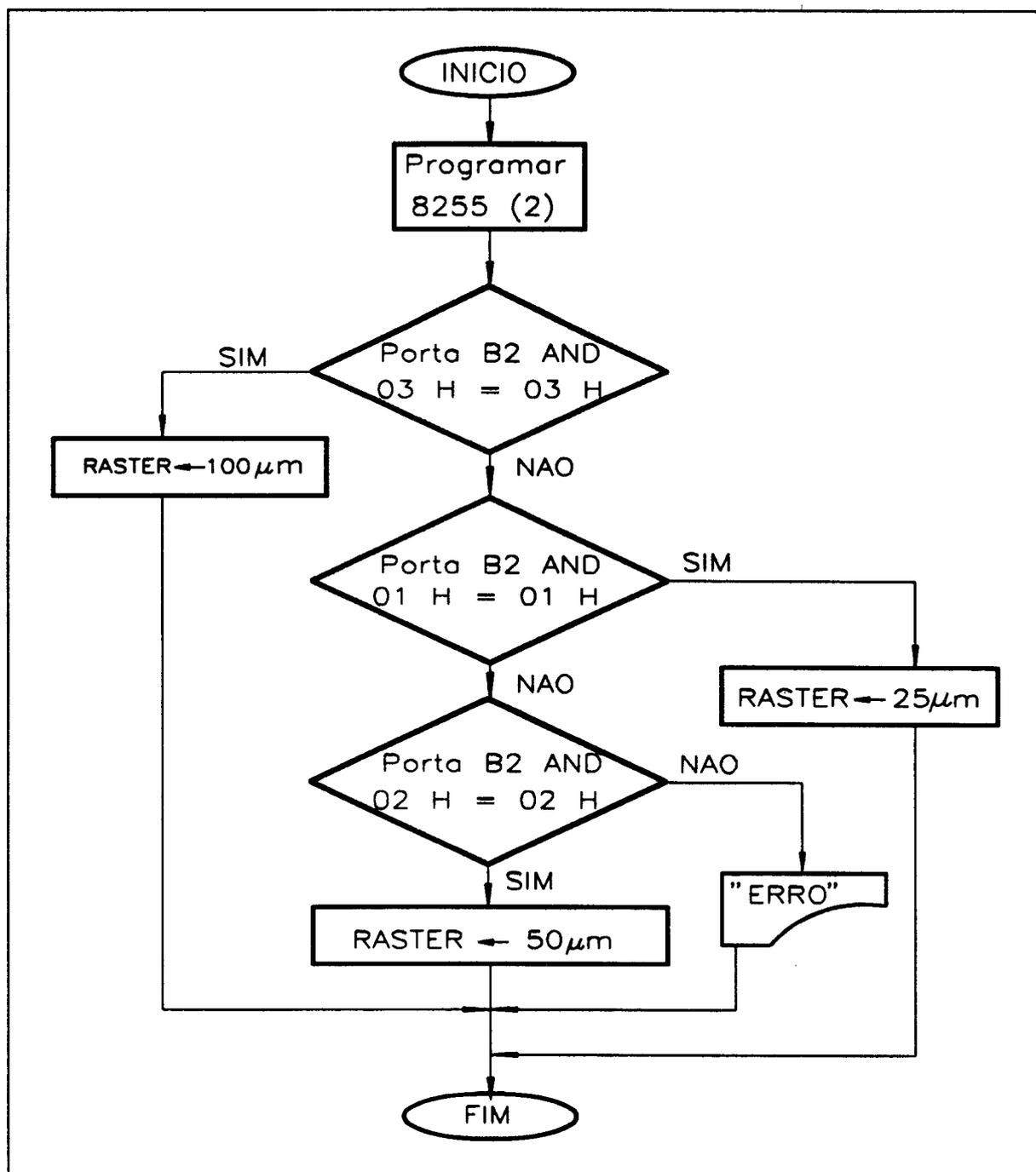


Figura 4.9 - Fluxograma para obtenção do valor da resolução ("Raster")

4.3.3 - Comandos para movimentar o subsistema óptico do Photoscan.

O movimento do subsistema óptico do Photoscan é realizado através da execução de uma seqüência de comandos que estão ilustrados no diagrama de sincronização ou distribuição da figura - 4.10. Este movimento pode ser realizado em duas direções : para direita (passo à direita) ou para esquerda (passo à esquerda).

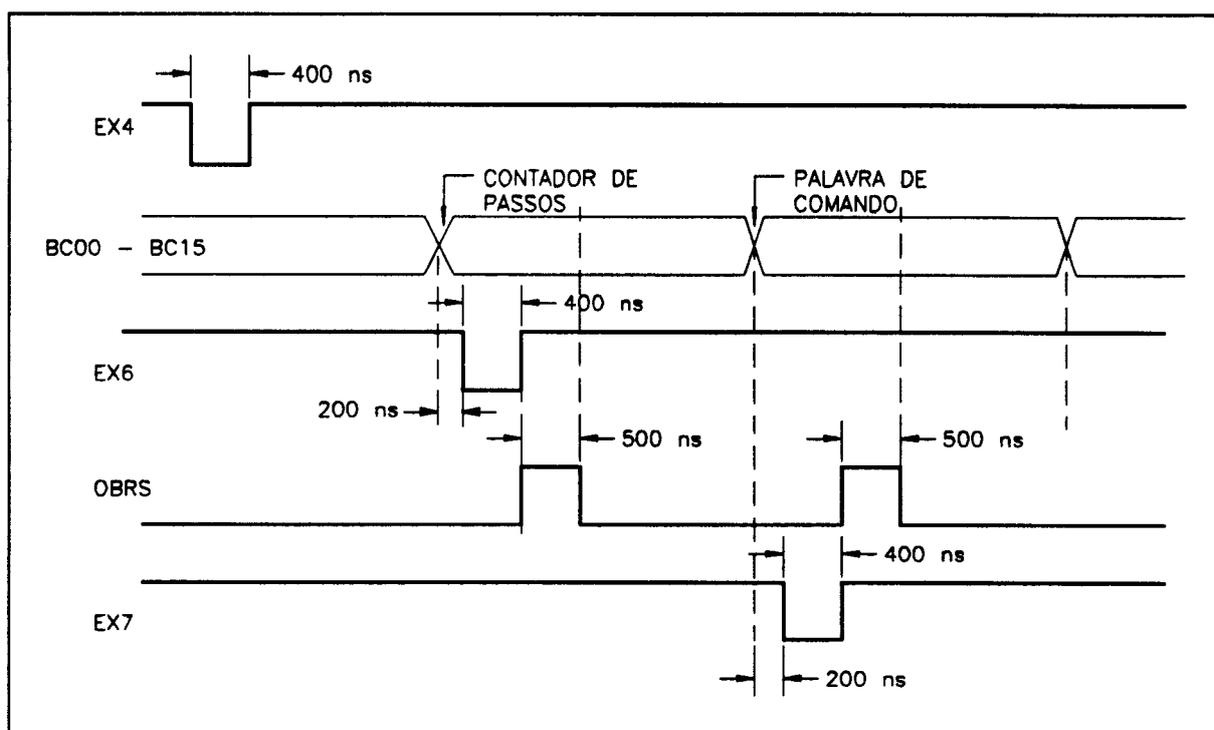


Figura 4.10 - Diagrama de sincronização para executar passo à direita ou à esquerda

A seqüência de comandos para execução deste movimento é a seguinte : executa-se um pulso, com nível lógico "baixo" em EX4. Este limpa qualquer comando que esteja em processo.

Carrega-se o número de passos (cada passo tem 12,5 μm) nas linhas de dados BC00..BC15, sendo este número fornecido em complemento de 2. Executa-se um pulso com nível lógico "baixo" em EX6, para que o dado (número de passos) contido nas linhas de dados BC00..BC15 seja carregado no contador (registro armazenador) do Photoscan.

Envia-se para o Photoscan a palavra de comando através dos 4 bits menos significativos das linhas de dados BC (BC00..BC03). Esta palavra de comando é que

determina a direção de deslocamento do subsistema óptico.

A tabela A.4 fornece os valores desta palavra. Finalmente, envia-se para o Photoscan um pulso com nível lógico baixo em EX7 para iniciar o comando especificado nas linhas BC00..BC03.

A figura - 4.11 mostra o fluxograma da seqüência de comandos para movimentar o subsistema óptico do Photoscan.

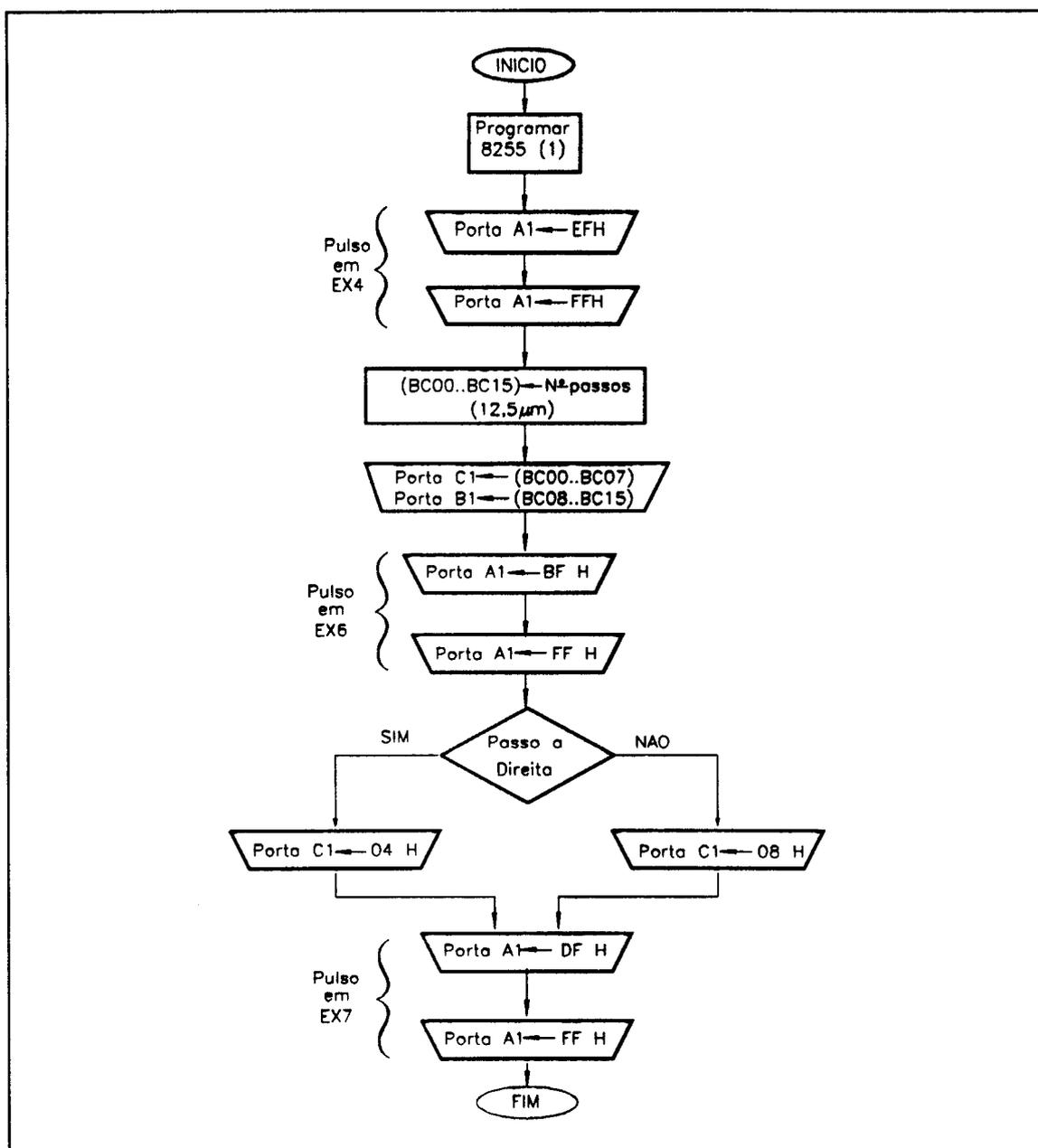


Figura 4.11 - Fluxograma da seqüência de passos para movimentar o subsistema óptico do Photoscan

4.3.4 - Leitura dos Dados (Densidades).

Para realizar a leitura ou medida fotométrica do filme (densidade), deve-se primeiramente executar uma seqüência de comandos para leitura que estão ilustrados no diagrama de sincronização da figura - 4.12.

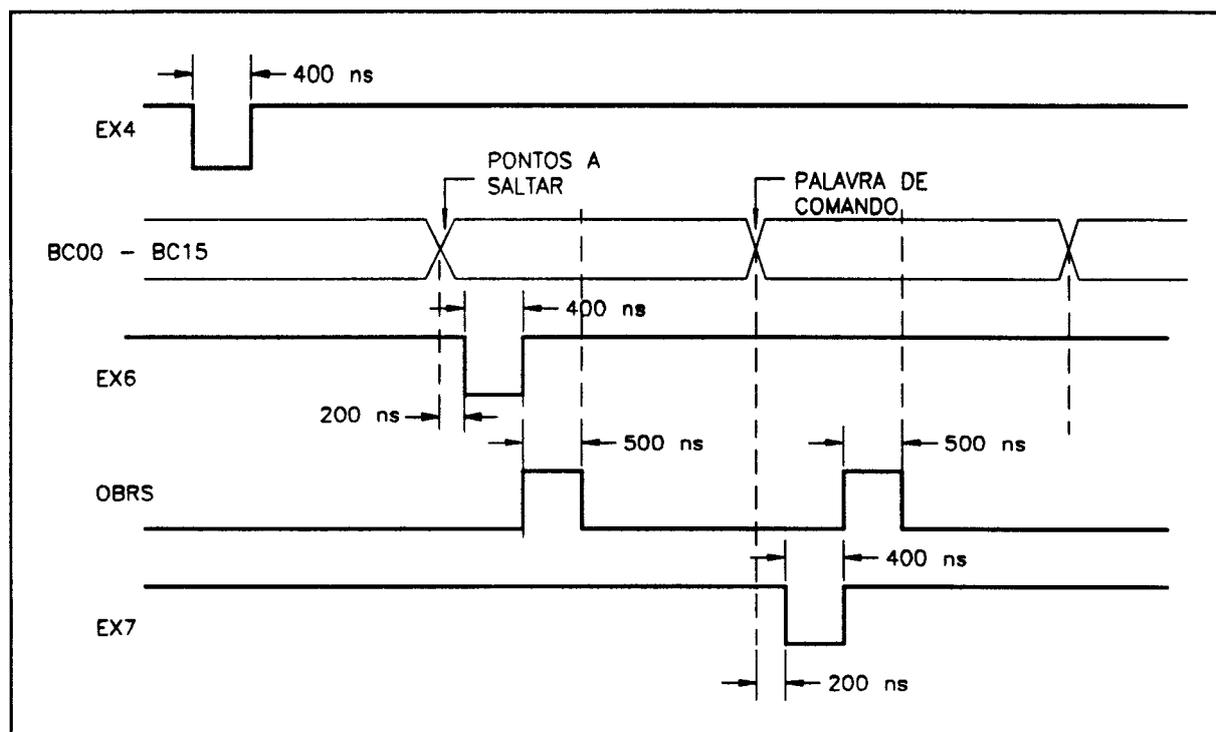


Figura 4.12 - Diagrama de sincronização para leitura

Depois, realiza-se os passos que estão contidos no diagrama de sincronização da figura - 4.13.

Os passos para a execução da leitura de um ponto ("pixel") do filme são realizados da seguinte maneira: inicialmente deve-se verificar e saltar o número de pontos para iniciar a leitura dos dados, ou seja, verificar qual a posição a partir da origem circunferencial do tambor onde será iniciada a tomada dos dados (a origem se situa na parte superior do tambor; ponto zero da escala do tambor).

O número de pontos para saltar é determinado de acordo com o valor da resolução ("Raster") selecionada. Assim, carrega-se nas linhas de dados BC00..BC15 o número de pontos (em complemento de 2) que deve-se saltar antes de iniciar a leitura dos dados (pontos).

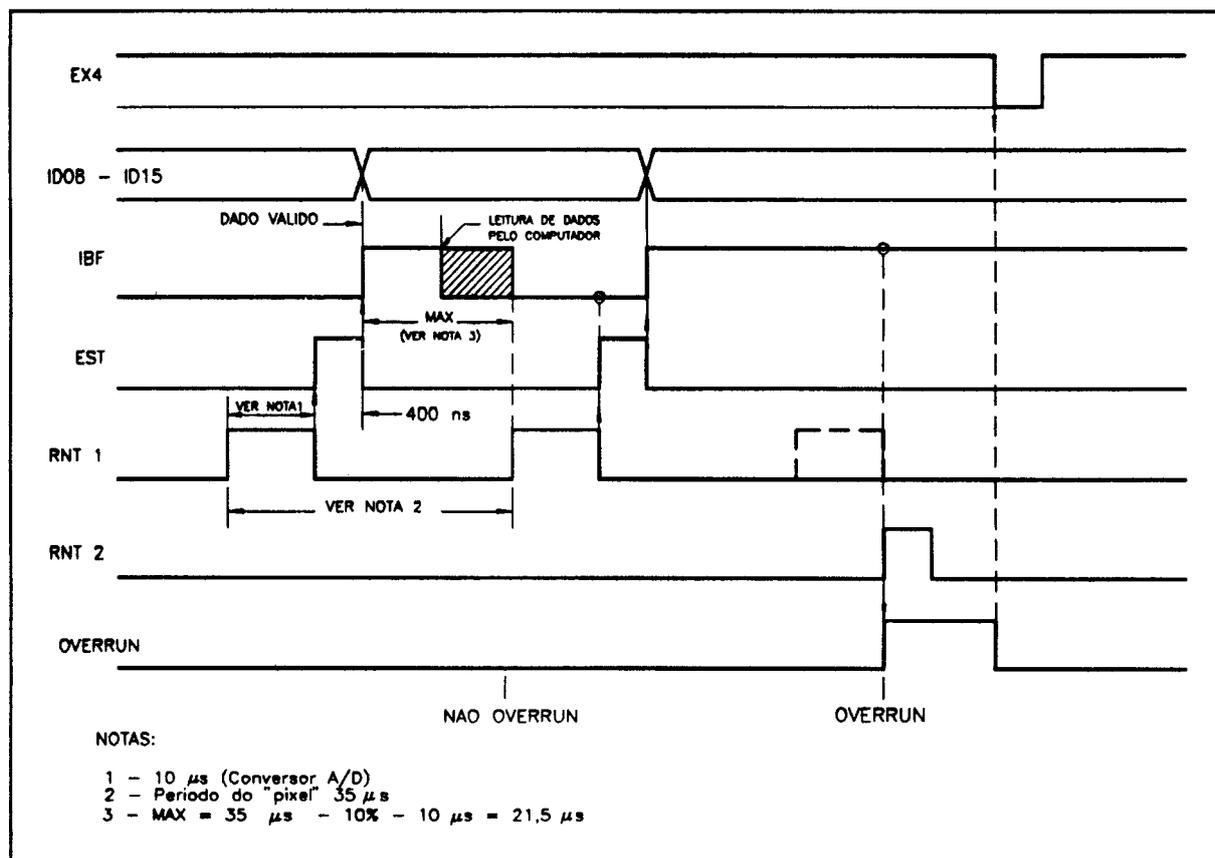


Figura 4.13 - Diagrama de sincronização para leitura de um ponto ("pixel")

Executa-se um pulso com nível lógico "baixo" em EX6 para carregar o dado (número de pontos a saltar) contido nas linhas de dados BC00..BC15 no contador do Photoscan.

Envia-se para o Photoscan a palavra de comando (neste caso 0E H) para realizar leitura dos dados (ver tabela A.4) através dos 4 bits menos significativos das linhas de dados BC (BC00..BC03). Envia-se para o Photoscan um pulso com nível lógico baixo em EX7 para iniciar o comando especificado nas linhas BC00..BC03 (ver figura - 4.12).

Desta forma, quando o Photoscan faz a leitura de um dado (ponto) do filme, este envia o sinal RNT1 (pulso com nível lógico "alto") indicando a transferência do dado para as linhas de dados ID08..ID15.

Depois, envia o pulso EST com nível lógico "alto" indicando que o dado contido nas linhas de dados (ID08..ID15) é válido. Assim, o sistema controlador (microcomputador), através de um software, envia o pulso IBF com nível lógico "alto" para "avisar" o Photoscan a aceitação e leitura (aquisição) do dado.

Caso ocorra um erro ("overrun"), o Photoscan envia o pulso RNT2 com nível lógico

"alto", indicando a sua ocorrência. A existência de um estado de "overrun" inibe futuras transferências de dados.

Quando o número de dados (pontos) tiver sido transmitido pelo Photoscan, executa-se um pulso com nível lógico "baixo" em EX4 para terminar a informação (ver figura - 4.13).

A figura - 4.14 mostra um fluxograma da seqüência de comandos ilustrados nas figuras - 4.12 e 4.13, onde o Photoscan faz a leitura de um ponto ("pixel") do filme.

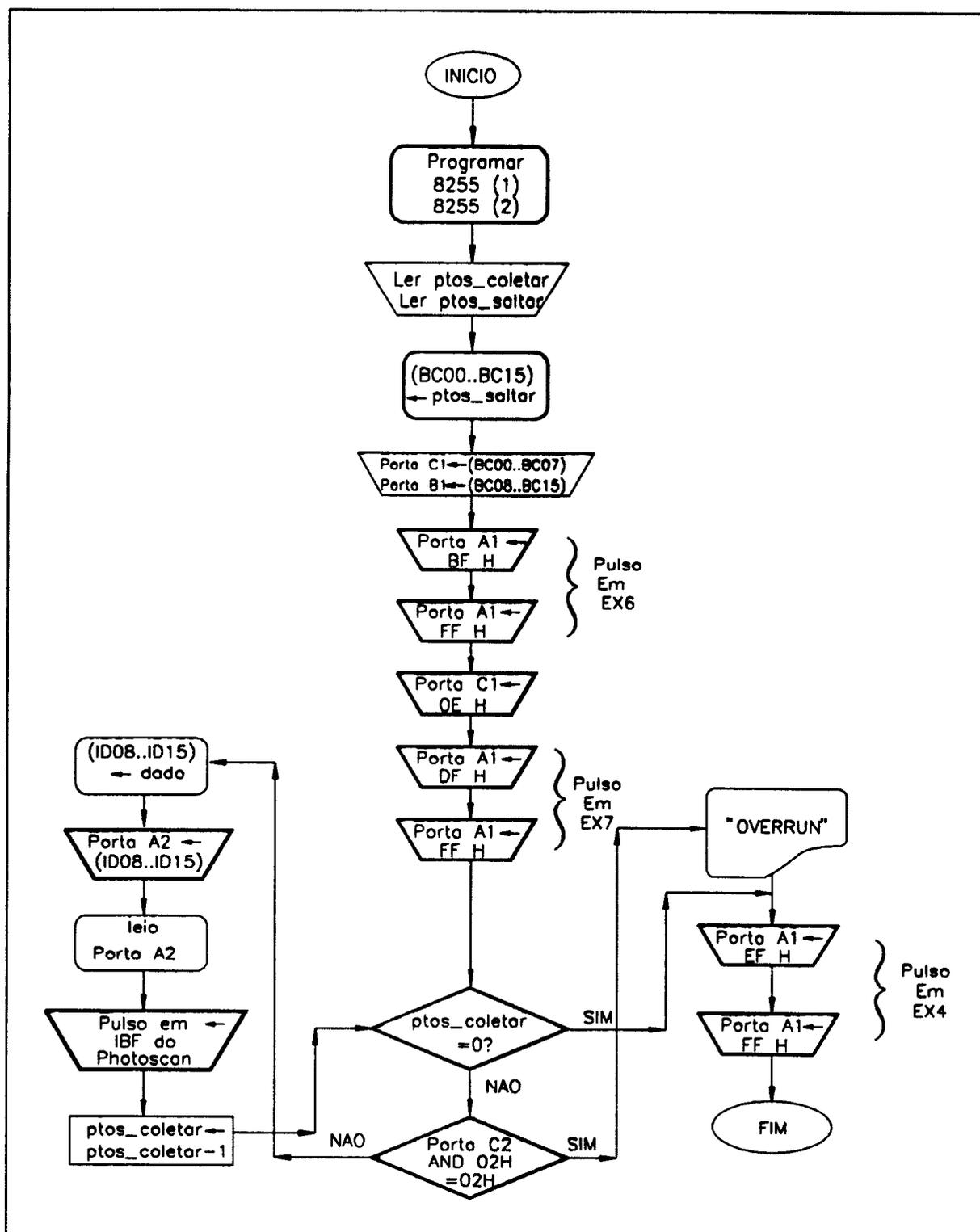


Figura 4.14 - Fluxograma para execução da leitura de um ponto

Devido ao fato do Photoscan enviar os dados (seqüência de passos descritos nas figuras - 4.12 e 4.13) para o microcomputador com muita rapidez, tornou-se necessário a colocação de um monoestável na placa da interface para que, desta forma, o tempo disponível, para a aquisição do dado pelo microcomputador, se tornasse maior. Ao mesmo tempo, utilizou-se na 8255 (2) o modo 1 de programação (descrito anteriormente).

Na figura - 4.15 está ilustrado o princípio de funcionamento do monoestável utilizado para auxiliar na leitura dos dados (densidades) do filme, realizada pelo Photoscan, e conseqüente coleta, feita pelo microcomputador.

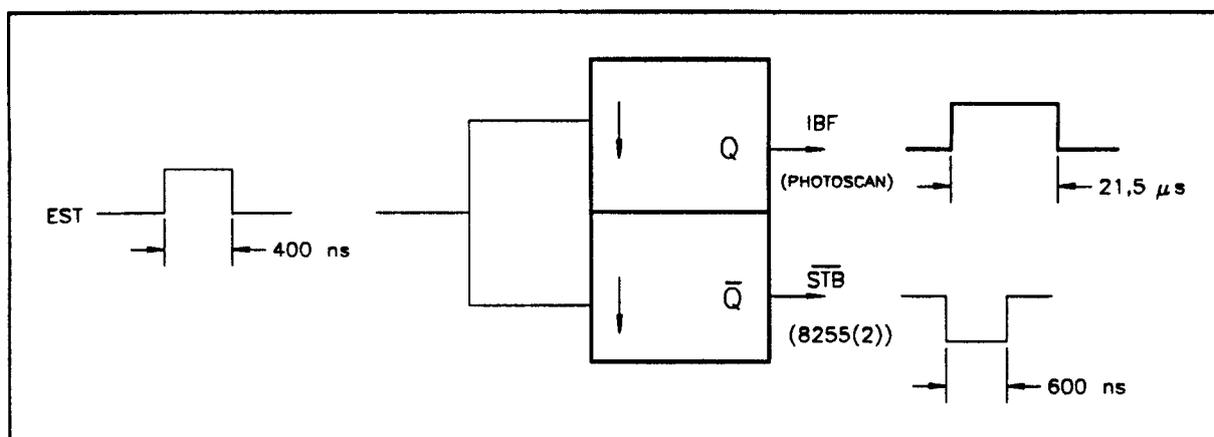


Figura 4.15 - Funcionamento do monoestável

Na borda de descida do pulso EST enviado pelo Photoscan ao monoestável, este enviará para 8255 (2) o pulso \overline{STB} para que o dado seja carregado na porta A da 8255 (2).

Quando a 8255 (2) carrega o dado na Porta A2, IBF (8255 (2)) passa para um nível lógico "alto", indicando que há um dado na Porta A2.

Desta forma, verifico se IBF da 8255 (2) está em um nível lógico "alto", para então fazer a leitura do dado na Porta A2 (IBF (8255 (2)) tem largura suficiente para que o microcomputador consiga fazer a leitura do dado). Ao mesmo tempo o monoestável também envia o pulso IBF (Photoscan) ao Photoscan indicando para este a aceitação e leitura do dado (contido na porta A da 8255 (2)).

A figura - 4.16 mostra o diagrama de sincronização e a figura - 4.17 o fluxograma da leitura do dado da 8255 (2) realizada pela microcomputador.

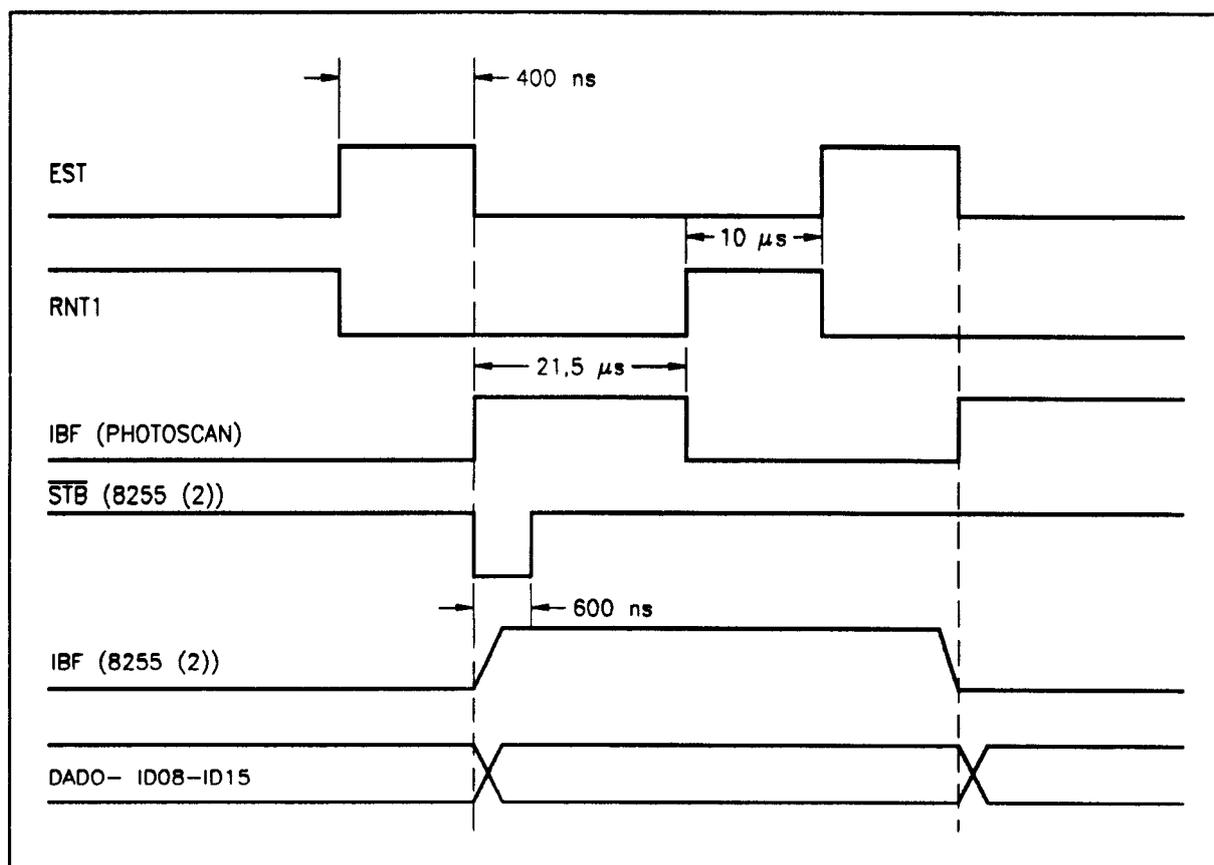


Figura 4.16 - Diagrama de sincronização para leitura de um dado da 8255 (2)

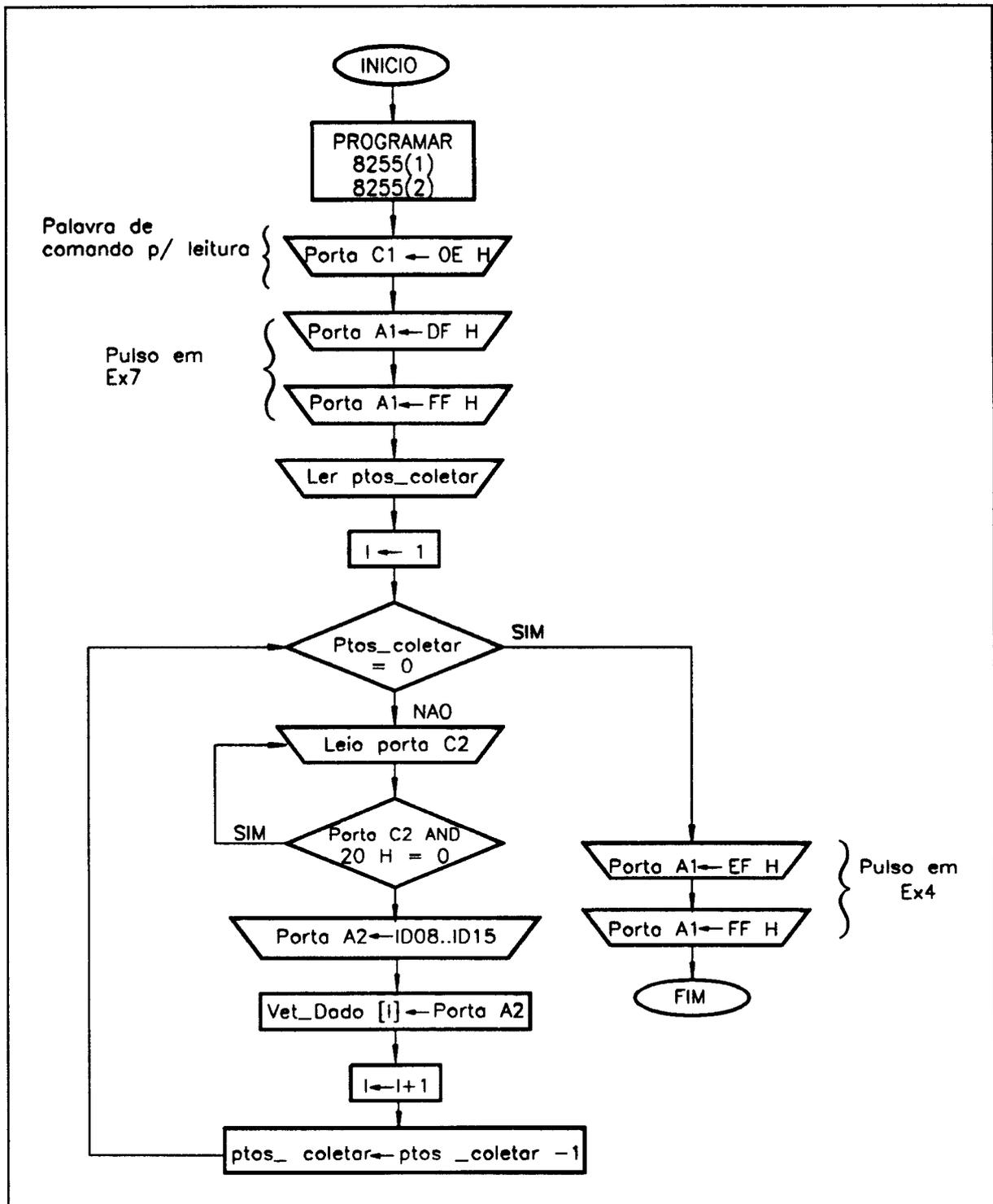


Figura 4.17 - Fluxograma para leitura dos dados

Capítulo 5

RESULTADOS E CONCLUSÕES

É apresentado a seguir um exemplo da execução do software desenvolvido.

A figura 5.1 ilustra a "tela" inicial do programa e, o menu principal está ilustrado na figura 5.2.

Selecionando-se a opção de digitalizar uma linha do filme (X-scan), o usuário entra com a posição da linha (start) que será digitalizada e a posição inicial (coluna) e final (pos_final) da tomada dos dados (ver figuras 5.3, 5.4 e 4.3).

Um gráfico das densidades pela distância x entre cada ponto será traçado (figura 5.5) e o usuário tem a opção de ampliar ("zoom") o gráfico obtido, eliminando alguns pontos que não interessam (figura 5.6).

A figura 5.7 ilustra o gráfico ampliado.

O tempo de aquisição dos dados neste exemplo (200 pontos coletados (5 mm) e resolução de 25 μm) foi de aproximadamente 20 segundos.

Na opção de digitalizar uma coluna do filme (Y-scan), o usuário entra com a posição da coluna (coluna) a ser digitalizada, a posição inicial (start) da tomada dos dados e o tamanho da coluna (size) a ser digitalizada (ver figuras 5.8, 5.9 e 4.4).

Analogamente como no caso da digitalização de uma linha do filme, será traçado um gráfico das densidades pela distância y entre cada ponto (figura 5.10) com a opção de ampliar ("zoom") a curva obtida (figura 5.11).

A figura 5.12 ilustra o gráfico ampliado.

Neste caso (200 pontos coletados (5 mm) e resolução de 25 μm), o tempo para se realizar a aquisição dos dados digitalizados foi de aproximadamente 10 milissegundos.

Finalmente, na opção de aquisição, será armazenado no arquivo DATA1.DEN (cujo nome é fornecido pelo usuário) os dados digitalizados e as outras informações mostradas na figura 5.13.

```
** PROGRAMA PARA DIGITALIZAR FILMES **
```

```
X_Min_Filme = 0 [ mm ]  
X_Max_Filme = 100 [ mm ]  
Y_Min_Filme = 0 [ mm ]  
Y_Max_Filme = 100 [ mm ]
```

Posicione o subsistema optico em : X = 0

Coloque o botao RASTER no valor desejado.

E pressione a barra de espaco quando estiver pronto.

Figura 5.1 - "Tela" inicial

MENU PRINCIPAL

```
** Opcoes para Digitalizacao de filmes **
```

```
X -SCAN - DIGITALIZA UMA LINHA (X).  
Y -scan - digitaliza uma coluna (Y)  
Aquisicao dos dados.  
Sair.
```

(Use ou X Y A S para destacar uma opcao.
depois, pressione <ENTER> para completar a selecao.)

Figura 5.2 - Menu Principal

```

Digite a posicao
(em mm) da linha
a ser varrida :
START = 40

COLUNA = 1.20
Digite a posicao
inicial (mm) para
inicio da tomada
dos dados :
COLUNA = 5

```

Figura 5.3 - Linha a ser digitalizada (START) e posição inicial da tomada dos dados (COLUNA)

```

Digite a posicao
(em mm) da linha
a ser varrida :
START = 40

COLUNA = 1.20
Digite a posicao
inicial (mm) para
inicio da tomada
dos dados :
COLUNA = 5
AGUARDE...
Digite a posicao
final em (mm)
POS_FINAL = 10

```

Figura 5.4 - Posição final da tomada dos dados (POS_FINAL)

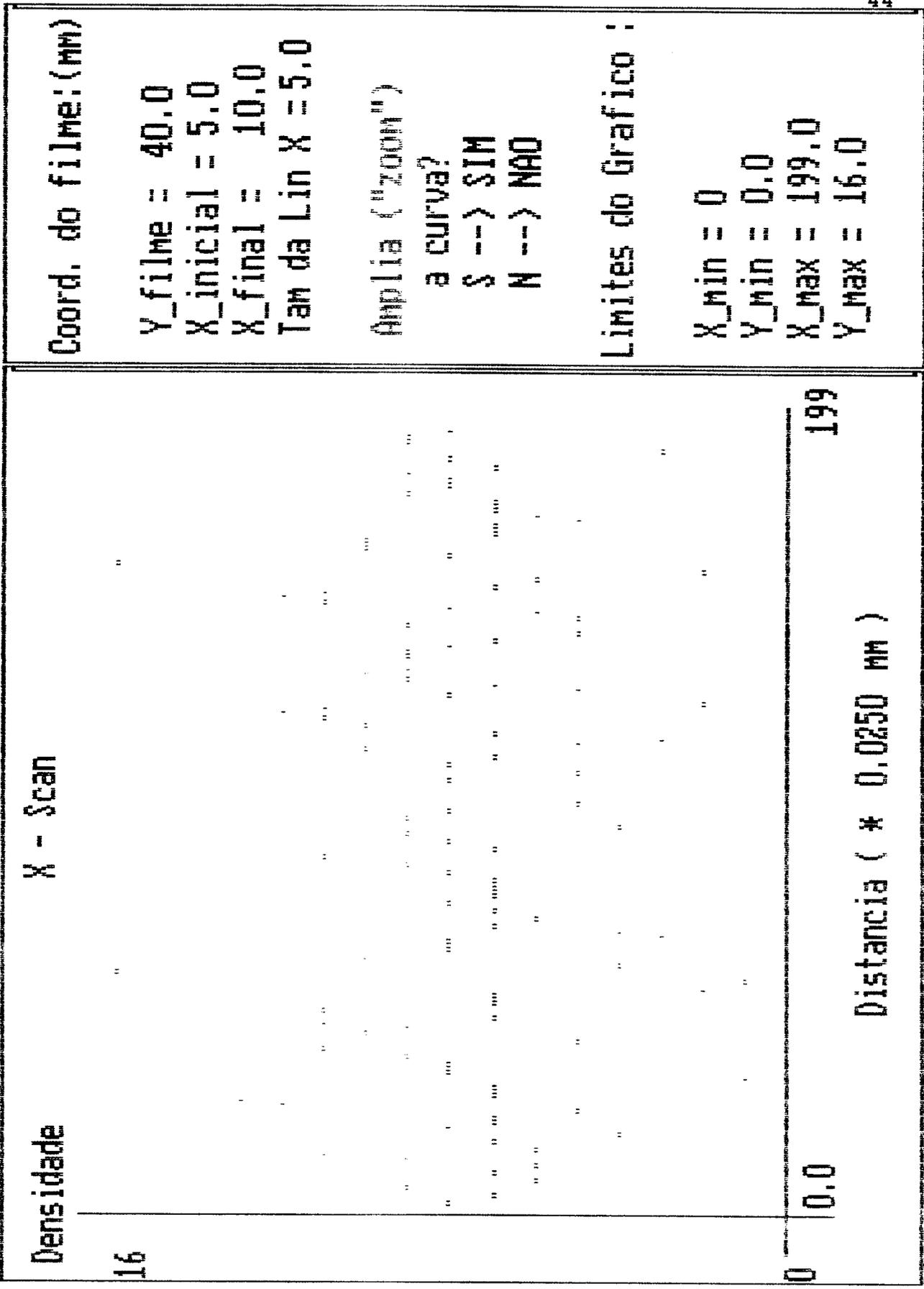


Figura 5.5 - Gráfico das densidades pela distância X (resolução)

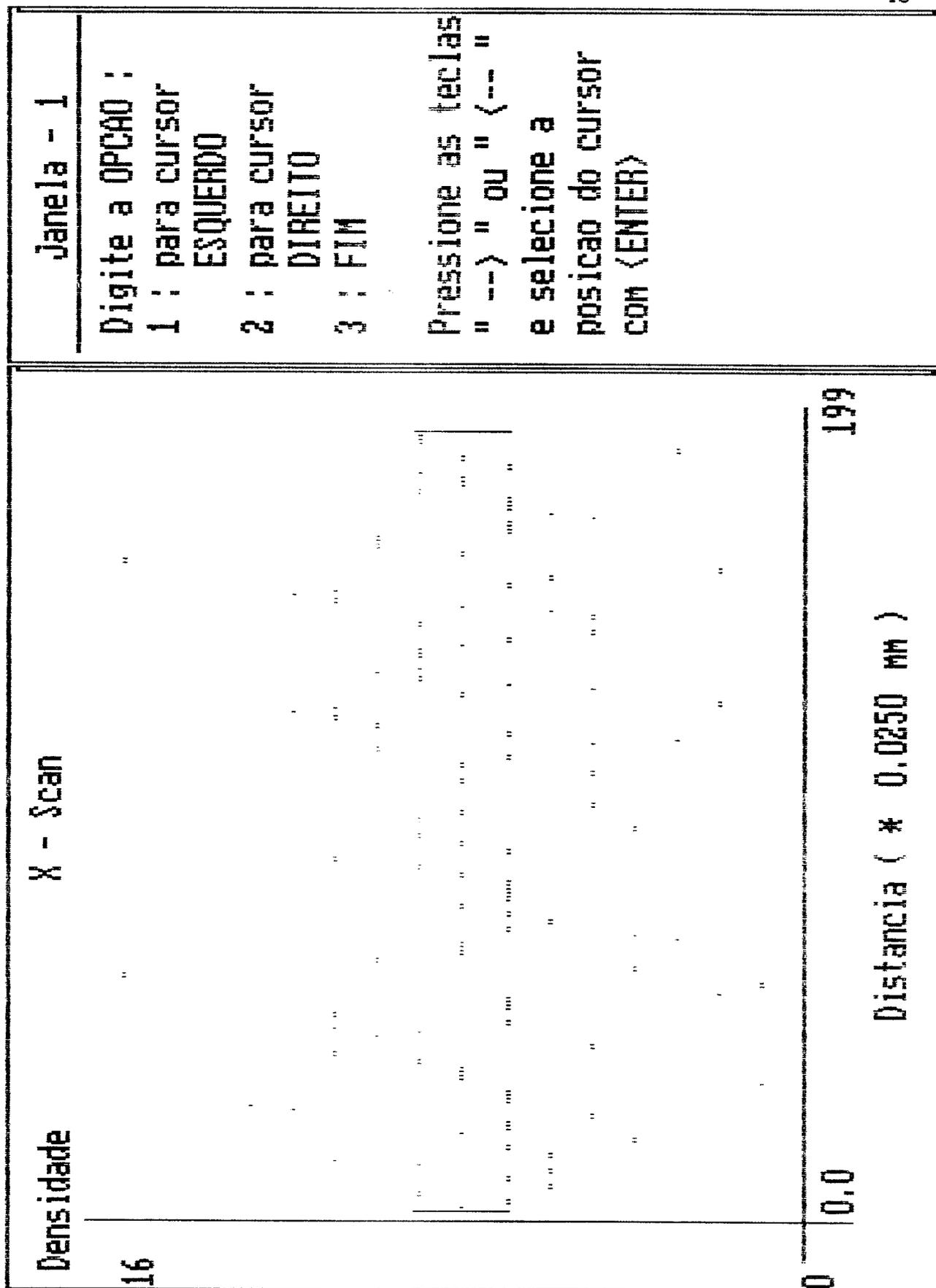


Figura 5.6 - Ilustração dos cursores nos extremos do gráfico a ser ampliado ("zoom") (X-scan)

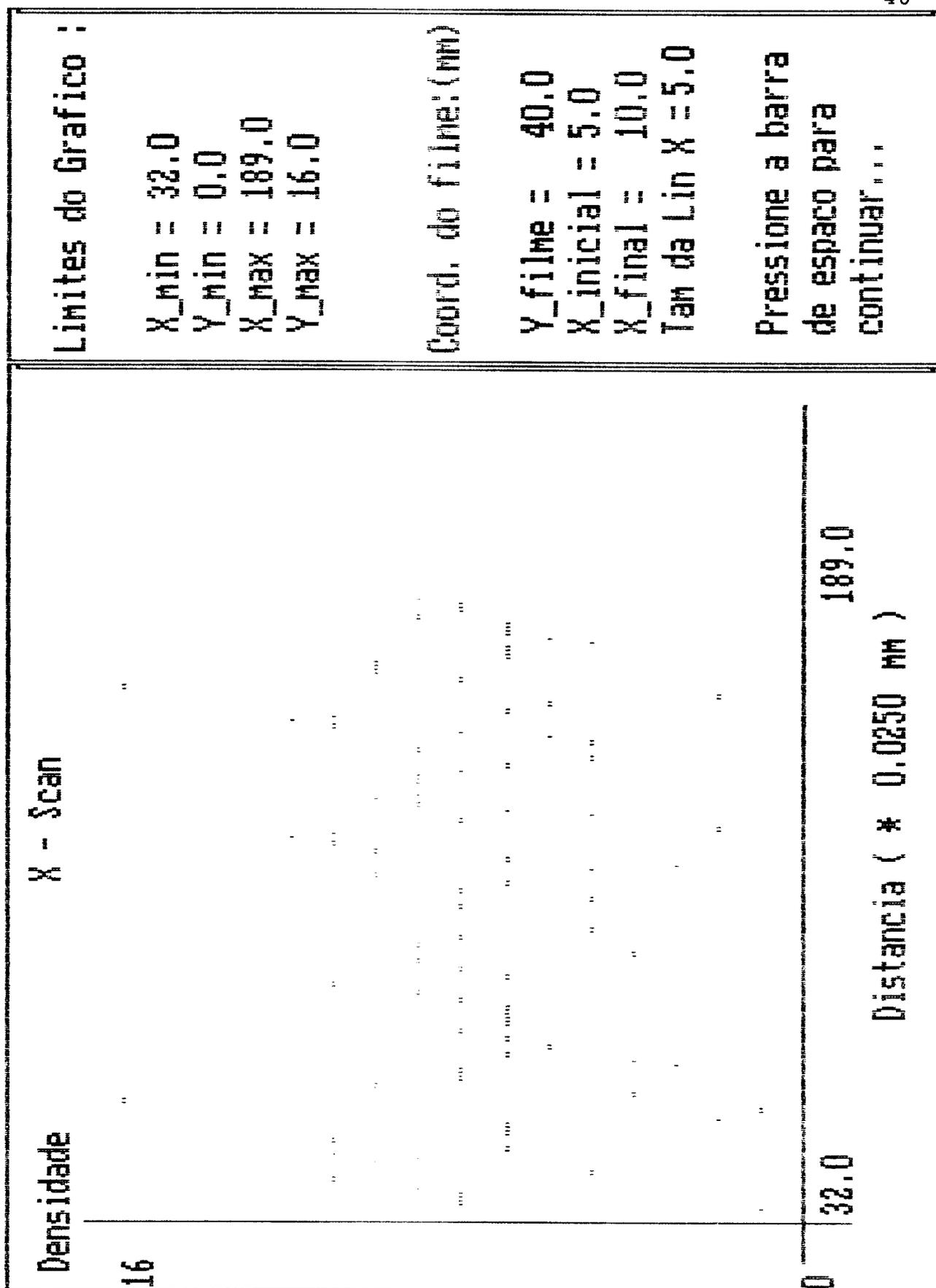


Figura 5.7 - Ilustração do gráfico ampliado ("zoom") (X-scan)

COLUNA = 10.00
Digite a posição
(em mm) da coluna a
ser digitalizada ;
COLUNA = 15

Figura 5.8 - Coluna a ser digitalizada (COLUNA)

COLUNA = 10.00
Digite a posição
(em mm) da coluna a
ser digitalizada ;
COLUNA = 15
AGUARDE...
Digite a posição
inicial (mm) para
início da tomada
dos dados e o
tamanho da coluna
(mm) dos dados.
START = 30
SIZE = 5

Figura 5.9 - Posição inicial da tomada dos dados (START) e tamanho da coluna a ser digitalizada (SIZE)

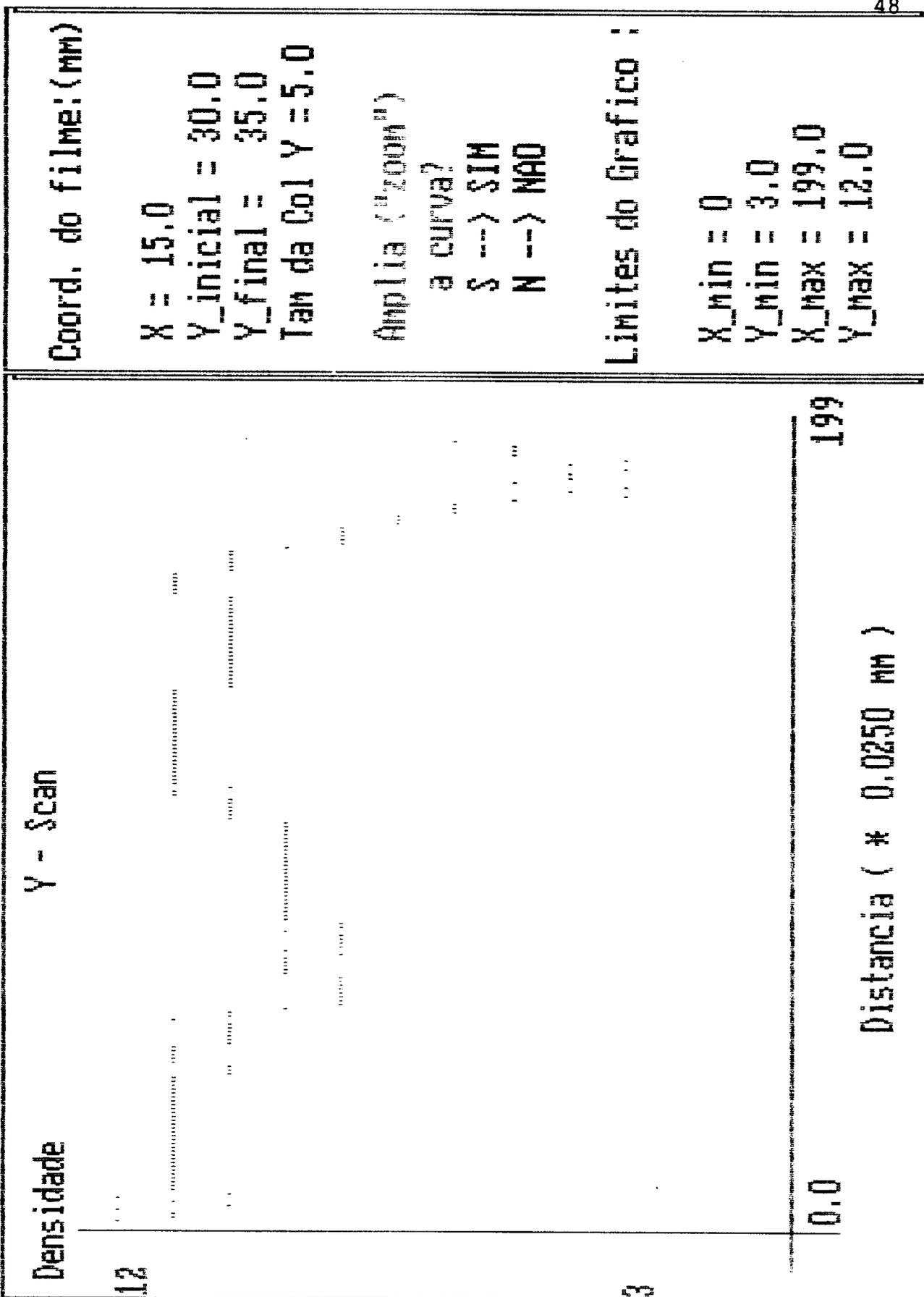


Figura 5.10 - Gráfico das densidades pela distância Y (resolução)

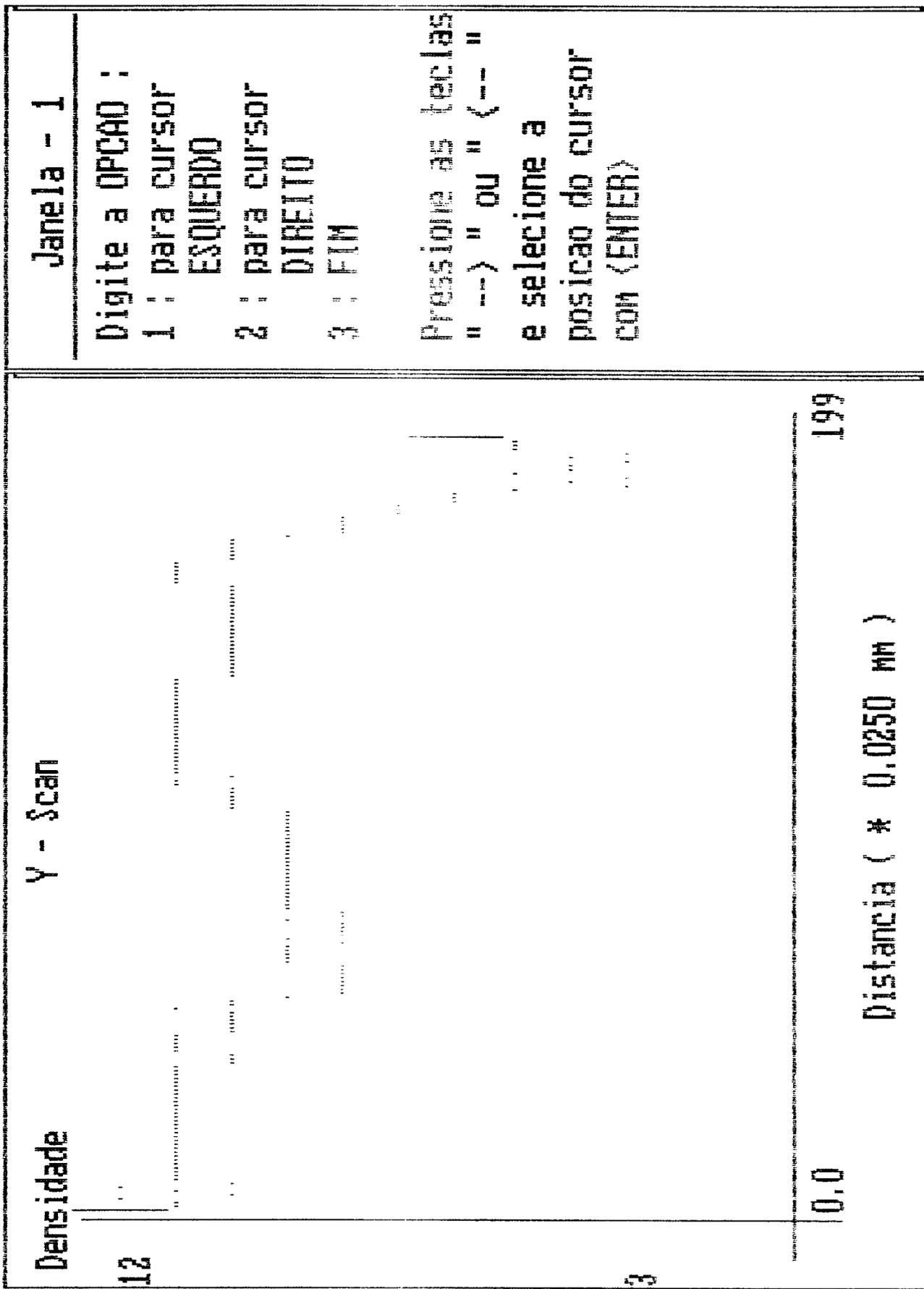


Figura 5.11 - Ilustração dos cursores nos extremos do gráfico a ser ampliado ("zoom") (Y-scan)

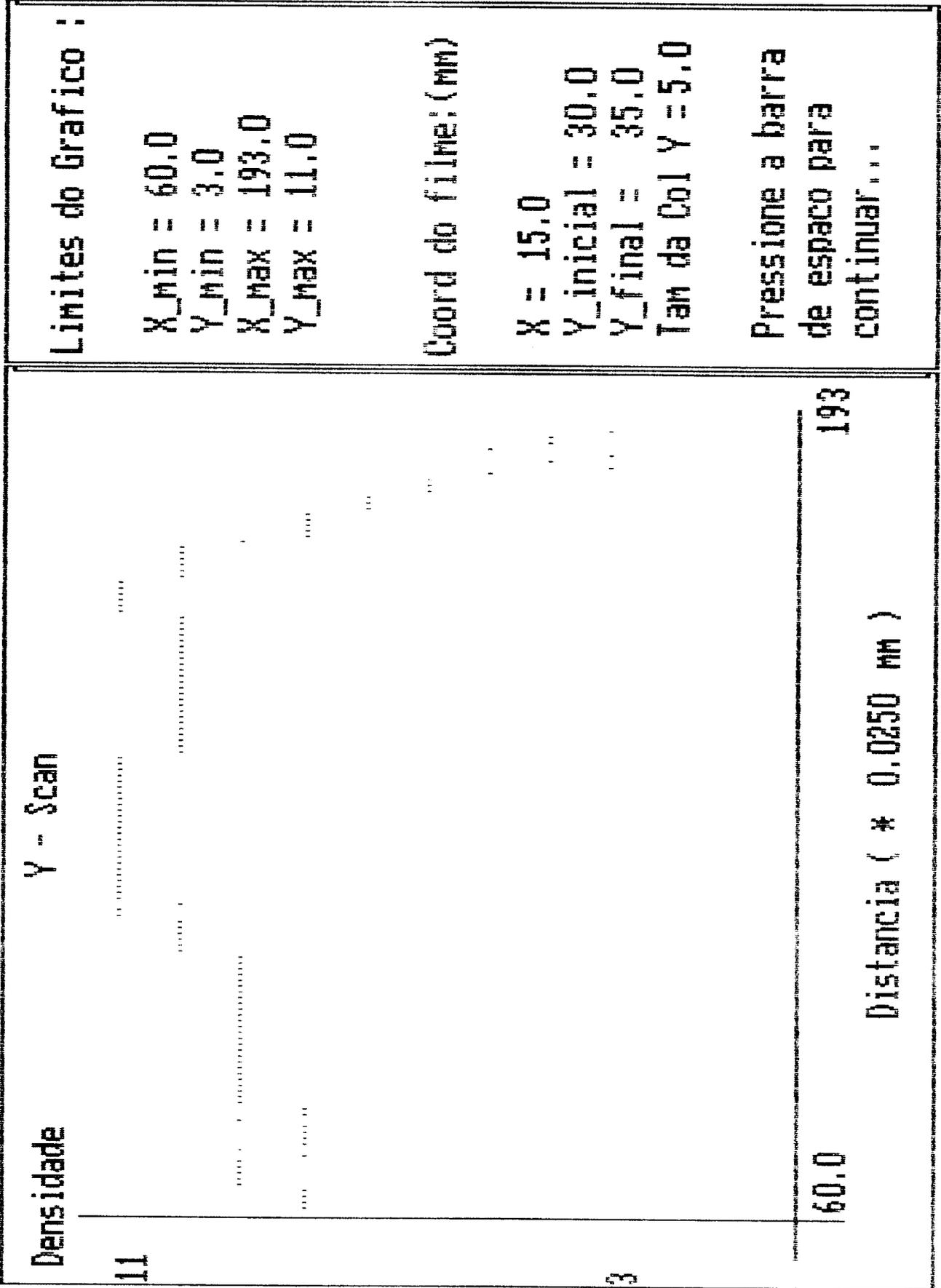


Figura 5.12 - Ilustração do gráfico ampliado ("zoom") (Y-scan)

Digite nome do arquivo para armazenar os dados : DATA1.DEN

Scan em : Y

RASTER = 0.0250 mm

Data : 12-11-1992

Comentarios : (texto com 70 caracteres)
Dissertacao

Limites do Grafico :

Coord. do filme:(mm)

X_min = 60.0

X = 15.0

Y_min = 3.0

Y_inicial = 30.0

X_max = 193.0

Y_final = 35.0

Y_max = 11.0

Tam da Col Y = 5.0

Pressione a barra de espaco para retornar ao menu.

Figura - 5.13 - Armazenamento no arquivo DATA1.DEN dos dados digitalizados e as outras informações mostradas na tela

Pode-se concluir, então, que tanto o projeto de hardware quanto o software desenvolvidos, atingiram as metas propostas. A interface desenvolvida foi capaz de gerar todos os sinais de controle e aquisição necessários e, dentro dos padrões exigidos pelo Photoscan. Para tais ajustes, foram necessários a implementação da rotina de aquisição de dados em linguagem Montadora, afim de atingir a velocidade necessária para viabilizar a aquisição de dados sem perda de pontos.

Na parte de software, destacaram-se a parte interativa e gráfica que não existiam.

A não disponibilidade no início do projeto de terminais e interfaces gráficas para microcomputadores PC suportando os 256 níveis de intensidades, inviabilizou a visualização de regiões de intensidades. Com a disponibilidade destes novos recursos, como futuro trabalho, o programa poderá ser expandido para suportar a visualização de partes da imagem em 2 dimensões.

Estas modificações, no entanto, não eliminam a necessidade de visualizar linhas de intensidades para interpretação mais precisa destas na imagem.

A escolha da linguagem PASCAL ajudou tanto na autodocumentação do programa quanto facilitará a implementação de futuras alterações por se tratar de uma linguagem de alto nível, estruturada e padronizada.

REFERÊNCIAS BIBLIOGRÁFICAS

- {AB 66} - ABRAHAMSSON, S. A computer controlled film scanner *Journal of Scientific Instruments*, v.43, p.931-33, 1966
- {BI 81} - BIRKNER, J. ; COLI, V. PAL - Programmable Array Logic Handbook 1981
- {CA 79} - CASTLEMAN, K. R. Digital image processing New Jersey, Prentice-Hall, 1979. 429p
- {HE 89} - HERGERT, D. Dominando o turbo Pascal 5. Rio de Janeiro, Ciência Moderna, 1989.
- {IN 81} - INTEL Peripheral design handbook. 1981
- {MA 72} - MATTHEWS, B. W.; KLOPFENSTEIN, C. E.; COLMAN, P. M. A computer controlled film scanner for X-ray crystallography. *Journal of Physics E*, v.5, p.353-59, 1972.
- {MO 80} - MONTOURI, J. S. Image scanner technology. *Photogrammetric Engineering and Remote Sensing*, v.46, n.01, p.49-61, Jan 1980.
- {NA 83} - NAGY, G. Optical scanning digitizers. *IEEE Computer*, v.16, n.05, p.13-23, may 1983.
- {NO 70} - NOCKOLDS, C. E.; KRETSINGER, R. H. An evaluation of a rotating drum densitometer and its application to precession photographs of protein crystals. *Journal of Physics E*, v.3, p.842-46, 1970
- {TTL 76} - TTL Data Book : for design engineers. 2.ed. s.l., Texas Instruments, c1976.

{VI 81} - VISCONTI, Antonio C. J. Franceschini. **Microprocessadores 8080/8085:**
hardware. 7.ed. São Paulo, Érica, 1981. v.1

{XU 69} - XUONG, N. An automatic scanning densitometer and its application to
x-ray crystallography. **Journal of Physics E. Série 2**, v.2, p.485-89,
1969

Apêndice A - O Photoscan Optronics P-1000

A.1 - Introdução ao Photoscan Optronics P1000

O Photoscan é descrito em sua documentação, como sendo um sistema leitor digital de filmes de alta resolução. O sistema mecânico eletro-óptico, incorpora um varredor eletro-óptico com um tambor rotatório e um motor de passo.

O modo de entrada, converte as densidades fotométricas do negativo do filme para valores digitais podendo transferir estas informações para um computador ou um outro dispositivo armazenador de dados. Para digitalizar um negativo do filme, este é montado sobre uma abertura num tambor cilíndrico que rotaciona em alta velocidade (ver figura 2.3).

A densidade óptica do negativo do filme é lida por um sistema óptico incoerente, que está sempre no eixo para garantir linearidade, estabilidade e repetibilidade. O sistema óptico consiste de uma fonte de iluminação, cuja luz, passando através de uma abertura, incide sobre o filme e atinge um fotodetector, que faz a medida. As densidades de imagem (ou de cinza) do negativo do filme, ao longo de uma circunferência do tambor (eixo Y), são medidas através de uma resolução (campo de exploração) pré-selecionada.

O Photoscan possui três valores de resolução : 25, 50 e 100 μm , podendo ser estendido para cinco. Depois de cada revolução, há um deslocamento do sistema óptico na direção axial (eixo X), de acordo com a resolução selecionada, sendo este controlado por software. Este deslocamento pode ser para direita ou esquerda, sendo cada passo deslocado de 12,5 μm e, o processo se repete até que a área de interesse seja varrida.

O sistema eletronicamente correto, tem densidade zero em cada revolução pela introdução de uma abertura (fenda) sobre o caminho da luz. A saída do fotodetector é amplificada por um amplificador logarítmico, que alimenta um conversor analógico para digital (A/D).

Os dados digitais são então transferidos para um computador externo através da interface. Cada ponto de densidade usa oito bits, permitindo que a extensão das densidades seja dividida em 256 níveis de cinza.

A.1.1 - Mecanismo de Varredura

O mecanismo de varredura consiste de um tambor giratório de formato cilíndrico, no qual um negativo de filme é montado para ser lido. Um filme é fixo sobre um corte da superfície cilíndrica que provém um índice de referência do filme e é esticado sobre este corte.

O tambor pode rotacionar em três velocidades, dependendo da resolução selecionada. As velocidades estão na razão de 1 : 2 : 4, acompanhando a velocidade de conversão do conversor A/D para as resoluções de 25, 50 e 100 μm (ou 2 : 4 : 8) com respeito uma com a outra; a maior velocidade corresponde a maior resolução.

Como o tambor rotaciona, o sistema óptico dentro e debaixo do tambor digitaliza a densidade do negativo do filme, fazendo uma varredura em forma de uma linha no filme. Esta linha varrida, constitui o eixo Y no sistema de coordenadas cartesianas convencionais.

Depois de cada revolução, ou varredura no eixo Y, o sistema óptico é deslocado uma distância predeterminada (valor da resolução) e é feita a varredura de uma outra linha. A translação linear do sistema óptico fornece as informações ao longo do eixo X.

A.1.2 - Subsistema Óptico

O subsistema óptico é formado e montado sobre um carro que é movido linearmente ao longo do eixo X. A extremidade superior avança sobre o tambor (paralelamente e dentro deste) e transporta o iluminador óptico. A extremidade inferior transporta o detector óptico. O detector óptico é fixo em relação ao iluminador óptico e o carro óptico movimenta-se, transportando assim, ambos na direção X.

A figura A.1 ilustra o sistema óptico de leitura. O iluminador consiste de uma fonte de luz (tungstênio), uma lente de colimação e uma roda de abertura. A lente de colimação focaliza a luz com a abertura selecionada sobre o filme. A roda de abertura é dentada para permitir a seleção de aberturas de diferentes tamanhos (25, 50 e 100 μm).

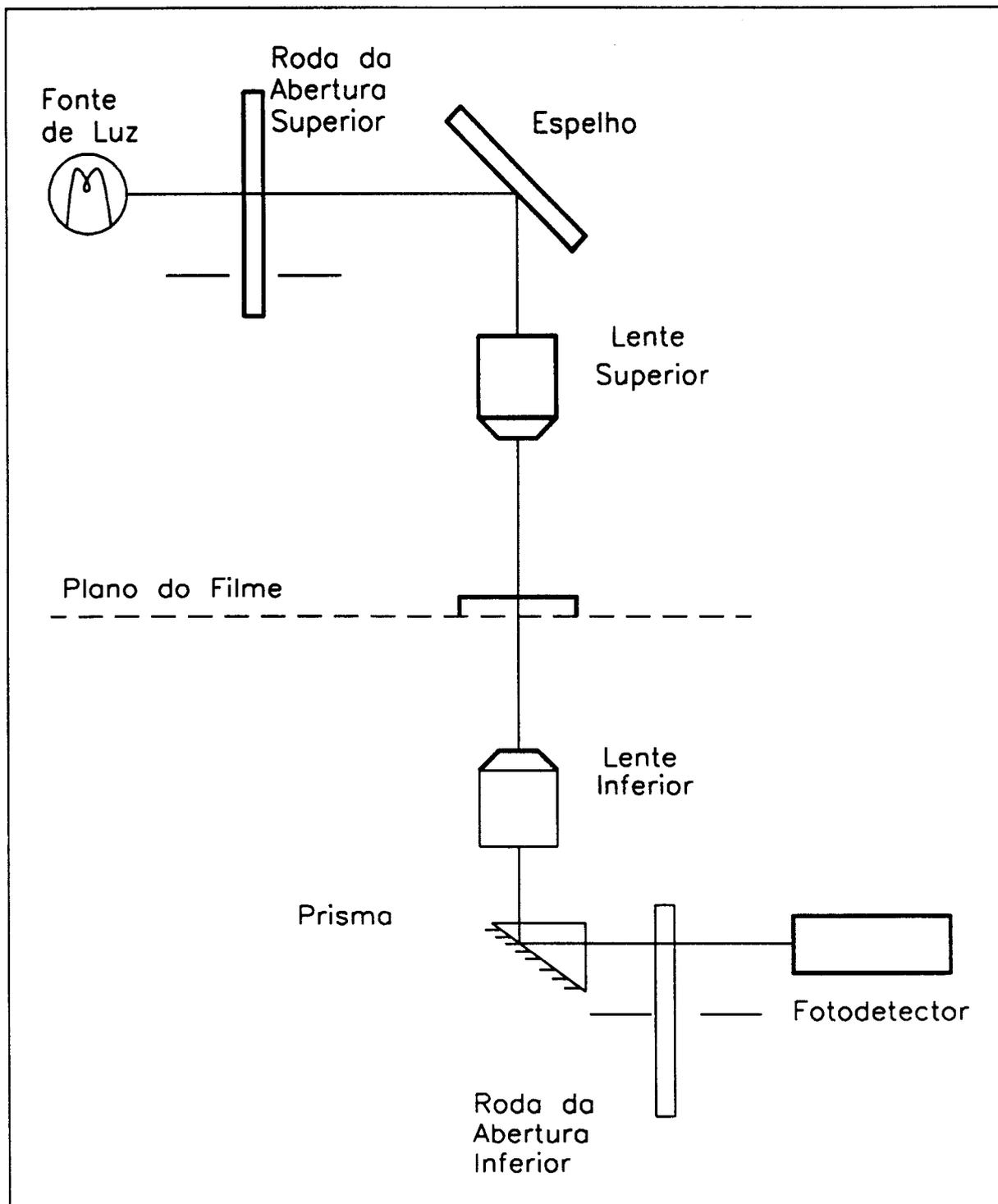


Figura A.1 - Diagrama em blocos do sistema óptico de leitura

A luz passa pela abertura e é, em seguida, focalizada por uma lente de tal forma que o ponto de luz que incide no plano do filme seja um quinto do tamanho da abertura. A configuração do iluminador garante uma iluminação uniforme sobre o ponto de interesse.

O detector consiste de uma fotomultiplicadora (fotodetector) e uma roda de abertura. A abertura é selecionável, e é similar àquela usada no iluminador. Esta abertura é reduzida para metade do tamanho, por uma lente.

A luz do iluminador passa pelo filme, pela abertura e vai para a fotomultiplicadora. A saída da fotomultiplicadora proporciona uma corrente analógica proporcional à luz que incide na fotomultiplicadora. Deve-se notar, que a luz transmitida através do filme, deve ser transformada para uma quantidade logarítmica para poder ser expressada em densidade, conforme expresso pela equação (1).

A.1.3 - Subsistema de Acionamento e Codificação

O tambor do varredor é movimentado em três velocidades por um motor síncrono, através de correias e polias. A velocidade do movimento depende da resolução selecionada e varia de acordo com a configuração do sistema. A velocidade de movimento está na razão de 1 : 2 : 4 (no mesmo caso 2 : 4 : 8) com a mais alta velocidade correspondendo ao mais alto valor da resolução.

Esta rotação do tambor, permite ao sistema óptico varrer o filme ao longo de uma linha na direção circunferencial (eixo Y). Conforme já descrito, o filme é montado sobre uma janela no tambor, tal que o caminho da luz do iluminador para o detector é completo quando o filme está entre os dois subsistemas ópticos (iluminador e detector).

Normalmente um adaptador de filme é utilizado para cada tamanho de filme, garantindo assim uma focalização precisa da abertura sobre o filme.

Próximo à janela do filme no tambor, tem-se uma fenda. A luz passando sobre a fenda, quando está desobstruída, equivale a uma densidade zero, utilizada como valor de referência.

Um codificador é acoplado diretamente no eixo do tambor, gerando dois sinais. O primeiro pulso ocorre uma vez a cada revolução do tambor na referência zero da fenda do tambor. Este pulso é a coordenada Y (densidade de referência). O segundo sinal é um trem de pulsos indicando a posição precisa do tambor onde os valores das densidades serão amostrados e registrados.

O carro óptico é deslocado em pequenos passos paralelos ao eixo do tambor usando um parafuso de avanço preciso (parafuso com uma rosca-sem-fim) e um motor de passo. Este movimento produz a coordenada X do filme. No equipamento, para se produzir um incremento básico x de $12,5 \mu\text{m}$, é utilizado um parafuso com um passo de $2,5 \text{ mm}$ e um motor de passo que dá 200 passos por revolução.

O movimento na direção X é limitado por uma parada final, que é ajustável. Um interruptor limita o movimento, antes que o carro atinja a parada mecânica. No painel, lâmpadas de sinalização são utilizadas para alertar o operador de que o interruptor elétrico foi disparado.

A.1.4 - Controles e Indicadores

Os controles do operador e indicadores estão no painel de controle localizado no topo da mesa de comandos do Photoscan. Este painel está ilustrado na figura A.2.

As tabelas A.1a e A.1b identificam os controles e indicadores e dão uma breve descrição de suas funções.

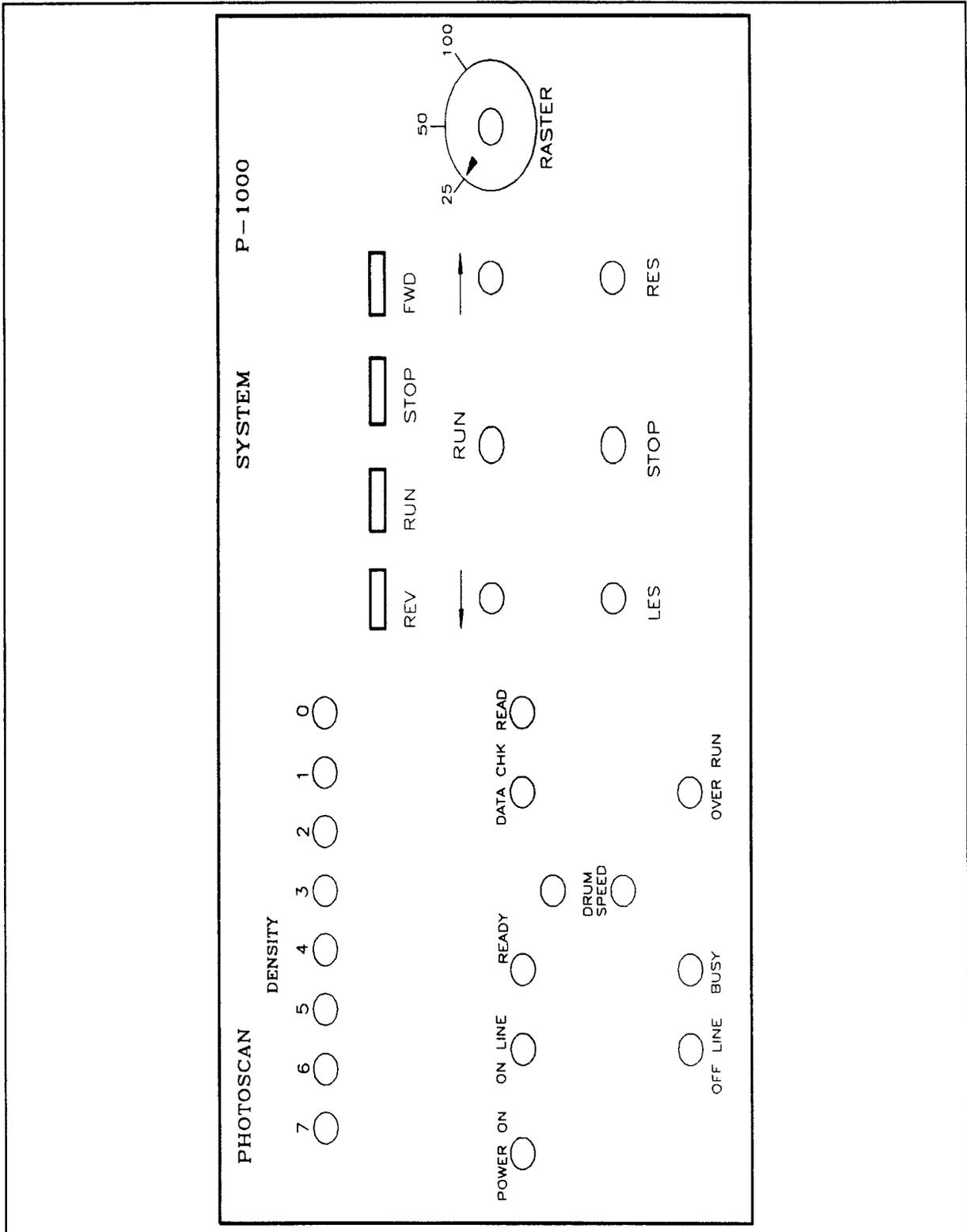


Figura A.2 - Painel de controles e indicadores do Photoscan

Controle ou Indicador	Função
lâmpada POWER ON	Indica que o Photoscan está ligado/desligado
lâmpada LES	Parada final a esquerda -indica que o subsistema óptico está nesta posição
lâmpada RES	Parada final a direita -indica que o subsistema óptico está nesta posição
Indicador READY	Indica que a leitura de dado do Photoscan está habilitada e tem dado para transferir
Chave STOP e luz indicadora	Usada para parar o movimento do carro do subsistema óptico ou a rotação do tambor
Chave RUN e luz indicadora	Usada para rotacionar o tambor
Chave FWD e luz indicadora	Usada para mover o carro do subsistema óptico para direita
Chave REV e luz indicadora	Usada para mover o carro do subsistema óptico para esquerda
lâmpada OVER-RUN	Indica que o dado fornecido pelo Photoscan não foi lido em tempo
lâmpadas DENSITY	Palavra digital de 8 bits representando a densidade de leitura em qualquer ponto
lâmpada DATA CHK	Sem função
lâmpada OFF LINE	Indica que o Photoscan não está pronto para fazer leitura
lâmpada ON LINE	Indica que o Photoscan está pronto para fazer leitura

Tabela A.1a - Controles e Indicadores do Photoscan

Controle ou Indicador	Função
lâmpadas DRUM SPEED	Inferior - indica que o tambor não está rotacionando ou não atingiu a velocidade de rotação correta Superior - indica que o tambor está rotacionando com velocidade correta
lâmpada BUSY	Indica transferência de dado
lâmpada READ	Indica leitura de dado
Chave giratória RASTER	Seleciona o intervalo entre dois pontos na direção circunferencial (Y) bem como a velocidade de rotação do tambor
Chave ¹ de sensibilidade	Seleciona alcance da densidade 0 até 2 ("up") ou 0 até 3 ("down")

Tabela A.1b - Continuação dos Controles e Indicadores do Photoscan

A.2 - Especificações da Interface do Photoscan

O Photoscan Optronics P-1000, possui uma interface que pode ser ligada a um sistema controlador, como por exemplo, um microcomputador, o qual fará todo controle sobre este. A interface está dividida em duas partes, onde em uma temos os dados de entrada para o controle do Photoscan e na outra, os dados de saída fornecendo sinais de controle e o valor da densidade óptica medida.

A figura A.3 ilustra um diagrama em blocos simplificado de como os sinais de entrada e saída do Photoscan foram ligados aos circuitos integrados 8255.

¹ Localizada dentro da abertura de acesso do Photoscan no lado direito do gabinete eletrônico

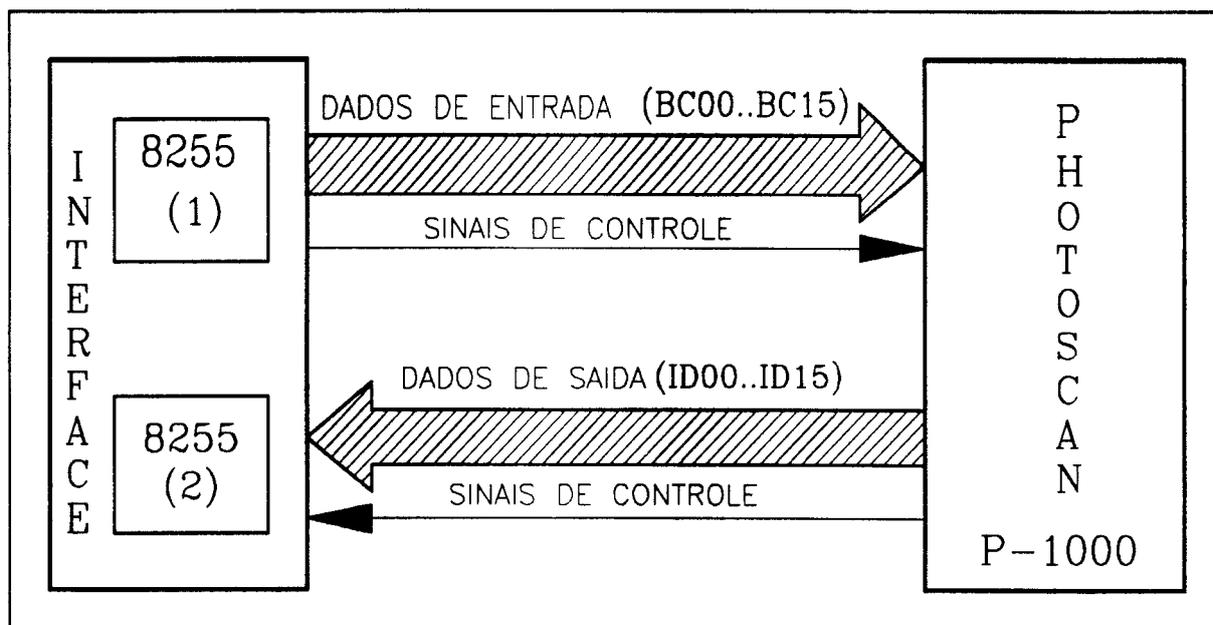


Figura A.3 - Diagrama em blocos das ligações dos sinais do Photoscan

Sinais de Entrada :

Os sinais para o Photoscan, vindos de um sistema controlador, entram através de um conector denominado **J8**. O conector é do tipo Amphenol Part #57-40500.

Nas tabelas A.2a e A.2b, os sinais de entrada são descritos na ordem em que seus pinos são distribuídos.

Os dados de entrada, com a descrição de seus sinais, estão contidos na tabela A.3 e a tabela A.4 define os valores, em hexadecimal, dos bits da palavra de controle.

Sinais de Saída :

Os sinais saem do Photoscan para o sistema controlador, através de um conector similar ao dos dados de entrada, mas rotulado como **J9**.

Na tabela A.5, temos a descrição dos sinais de saída de acordo com a distribuição dos seus pinos.

A distribuição dos bits dos dados de saída está descrita na tabela A.6. A tabela A.7, define os valores, em hexadecimal, dos bits dos dados de saída para obtenção da resolução escolhida no painel do Photoscan.

Pino #J8	Nome Sinal	Descrição do Sinal
1	BC00	DADOS DE ENTRADA
2	BC01	
3	BC02	
4	BC03	
5	BC04	
6	BC05	
7	BC06	
8	BC07	
10	EX7	Pulso de 400 ns (nível lógico baixo). Executa comando especificado em BC00 - BC03.
11	EX6	Pulso de 400 ns (nível lógico baixo). Carrega dado BC00 - BC15 no "buffer" do Photoscan.
13	EX4	Pulso de 400 ns (nível lógico baixo). Limpa qualquer comando que esteja em processo.
19	ES7	Bit de "status" - condição de dados excedidos. Este indica que uma transferência de dados foi requerida pelo Photoscan depois que o controlador (microcomputador) fez a leitura ou, que um fim de passo foi encontrado enquanto um passo estava em progresso. Inibe qualquer futura transferência de dados.

Tabela A.2a - Descrição dos pinos de Entrada

Pino #J8	Nome Sinal	Descrição do Sinal
20	IBF	"Buffer" de entrada cheio (IBF); é gerado pelo controlador depois de EST e é reestabelecido quando o controlador está lendo mais dados transferidos.
23	OBE	"Buffer" de saída vazio (OBE) é um sinal gerado pelo sistema controlador, para indicar que o dado em BC é válido.
26	BC08	DADOS DE ENTRADA
27	BC09	
28	BC10	
29	BC11	
30	BC12	
31	BC13	
32	BC14	
33	BC15	

Tabela A.2b - Continuação da Descrição dos pinos de Entrada

Nome Sinal	Descrição do Sinal
BC15	8 bits - exposição do filme
BC14	
BC13	
BC12	
BC11	
BC10	
BC09	
BC08	
BC07	Expansão opcional para exposição do dado
BC06	
BC05	Não utilizados
BC04	
BC03	Palavra de Comando
BC02	
BC01	
BC00	

Tabela A.3 - Dados de Entrada

BC03	BC02	BC01	BC00	Valor em Hexa	Função
0	1	0	0	4	Passo DIREITA
1	0	0	0	8	Passo ESQUERDA
1	1	1	0	E	Leitura DENSIDADE
1	1	0	1	D	Escrita no Filme
1	1	0	0	C	Não Operação

Tabela A.4 - Valores da Palavra de Controle

Pino #J9	Nome Sinal	Descrição do Sinal
1	ID00	DADOS DE SAÍDA
2	ID01	
3	ID02	
4	ID03	
5	ID04	
6	ID05	
7	ID06	
8	ID07	
14	ES3	Transporte óptico não "stepping". Esta condição existirá depois que um comando de passo é executado ou quando o comando do número de passos é completado.
17	ES0	Indica que o sistema não está aceitando um comando de leitura. Tambor não roda.
19	RNT1	Pulso de 10 μ s. Indica transferência de dados.
20	RNT2	Pulso de 500 ns. Indica que uma condição de erro ocorreu. A condição de erro pode ser determinada examinando-se a palavra de "status".
23	EST	Pulso de 500 ns, indicando que o dado do Photoscan é válido.
26	ID08	DADOS DE SAÍDA
27	ID09	
28	ID10	
29	ID11	
30	ID12	
31	ID13	
32	ID14	
33	ID15	

Tabela A.5 - Descrição dos pinos de Saída

Nome Sinal	Descrição do Sinal
ID15	8 bits - densidade do dado
ID14	
ID13	
ID12	
ID11	
ID10	
ID09	
ID08	
ID07	Opcional para densidade de 10 bits
ID06	
ID05	Ativado parada final à direita ("RIGHT END-STOP")
ID04	Ativado parada final à esquerda ("LEFT END-STOP")
ID03	Filme escuro
ID02	Resolução
ID01	
ID00	

Tabela A.6 - Dados de Saída

ID02	ID01	ID00	valor em Hexa	Resolução (μm)
0	0	0	0	12,5
0	0	1	1	25 ^a
0	1	0	2	50 ^a
0	1	1	3	100 ^a
1	0	0	4	200

Tabela A.7 - Valores da Resolução

^a Valores disponíveis no Photoscan P-1000

Apêndice B - A Interface de Comunicação Paralela Programável - 8255

B.1 - Interface de Comunicação Paralela Programável - 8255

Segundo Visconti {VI 81}, para a comunicação de dados entre a unidade central de processamento e os dispositivos de entrada e saída, os dados devem passar através de circuitos especiais que são as interfaces de entrada e saída. Nestes circuitos, um dado proveniente de um dispositivo de entrada e saída é armazenado para posterior leitura pela unidade central de processamento ou vice-versa.

Certos dispositivos ou certas aplicações, necessitam de interfaces de comunicação paralela, isto é, os dados de comunicação são formados por diversos bits e são transferidos todos os bits simultaneamente.

Entre as interfaces paralelas, a 8255 descrita no Peripheral Design Handbook da Intel {IN 81}, é um dispositivo de entrada e saída (I/O), programável com propósitos gerais, desenvolvido para ser utilizado com microprocessadores da linha Intel. A interface possui 24 pinos de I/O que podem ser programados individualmente em dois grupos de 12 pinos e utilizados em três modos principais de operação. Estes modos de operação serão descritos posteriormente.

B.1.1 - Arquitetura Interna da 8255

A figura B.1 ilustra a arquitetura interna da 8255. Segundo Visconti {VI 81}, as três portas de comunicação da interface paralela 8255 são divididas em dois grupos : o grupo A, formado pela porta A e os quatro bits mais significativos da porta C (PC7 - PC4), e o grupo B, formado pela porta B e os quatro bits menos significativos da porta C (PC3 - PC0). Cada controlador destes grupos recebe os comandos da unidade de controle e os envia para as portas associadas a ele.

A unidade de controle interna decodifica as variáveis de controle do sistema e gera os comandos para o buffer do duto de dados e para os controladores de grupo. A programação é feita por meio da palavra de controle fornecida pelo sistema.

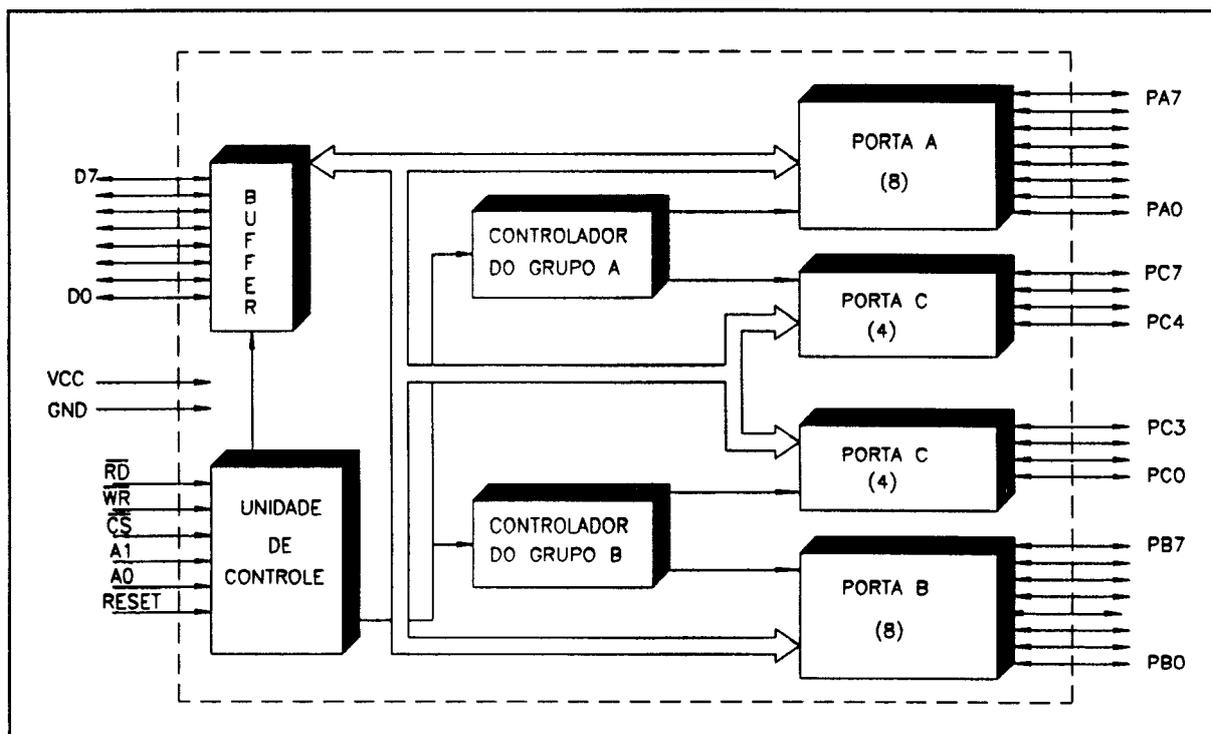


Figura B.1 - Arquitetura Interna da 8255

Os dados enviados pela unidade central de processamento serão recebidos pela interface paralela através do duto de dados e enviados aos dispositivos de entrada/saída por uma das portas conforme selecionado pela unidade de controle. De maneira inversa, os dados recebidos pela porta selecionada serão enviados à unidade central de processamento pelo duto de dados. Assim, é viabilizado por esta interface a interligação da unidade central de processamento com os dispositivos de entrada/saída.

B.1.2 - Descrição Funcional da 8255

O 8255 tem como função, servir como um componente de entrada e saída (I/O), de propósito geral, viabilizando o interfaceamento entre equipamentos periféricos e o duto de um sistema de microcomputador. A configuração funcional da 8255 é programada por software, dispensando lógica externa para fazer a interface entre dispositivos ou estruturas periféricas.

BUFFER : (Registro de Armazenamento)

Este registro possui oito bits com três estados bidirecionais, que são utilizados para fazer a interface entre a 8255 e um sistema de duto de dados. Os dados são transmitidos ou recebidos pelo registro, sob execução de instruções de entrada ou saída, pela Unidade Central de Processamento (CPU). As palavras de controle e de estado dos dados são também transferidas através do registro do duto de dados.

UNIDADE DE CONTROLE (Leitura/Escrita e Controle Lógico)

A função deste bloco é manipular todas as transferências internas e externas dos dados, bem como das palavras de controle ou estado. Ela aceita entrada de endereços da CPU e do duto de controle, distribuindo assim, comandos para os dois controladores.

 \overline{CS} (Seleção de "chip") :

Um nível lógico baixo na entrada deste pino, habilita a comunicação entre a 8255 e a CPU.

 \overline{RD} (Leitura):

Um nível lógico baixo na entrada deste pino, habilita a 8255 a enviar dados ou informação de estado para a CPU através do duto de dados. Em essência, permite que a CPU faça leitura da 8255.

 \overline{WR} (Escrita)

Um nível lógico baixo na entrada deste pino, habilita a CPU para escrever dados ou palavra de controle na 8255.

A0 e A1 :

Estes sinais de entrada, em conjunto com as entradas de leitura (\overline{RD}) e escrita (\overline{WR}), controlam a seleção de uma das três portas ou o registrador da palavra de controle. Eles são normalmente conectados aos bits menos significativos do duto de endereço (A0 e A1).

RESET :

Um nível lógico alto na entrada deste pino, limpa os registradores de controle e todas

as portas (A, B e C) são colocadas no modo de entrada.

B.1.3 - Modo de Operação

Conforme descrito por Visconti {VI 81} a interface de comunicação paralela 8255 é um dispositivo programável pelo microprocessador e pode operar em três modos distintos :

Modo 0 - Neste modo, suas portas estarão programadas para serem basicamente entrada ou saída de dados.

Modo 1 - Aqui, apenas as portas A e B serão entrada ou saída de dados, porém elas são sincronizadas por meio de sinais de controle especiais ligados à porta C.

Modo 2 - Neste, a porta B não é usada, a porta A constituirá um duto de dados bidirecional de entrada e saída e os bits da porta C são usados para sinais de controle de maneira semelhante ao modo 1.

A interface paralela 8255 pode operar em um destes modos, conforme sua programação, ou ainda, ela pode trabalhar com combinação destes modos, isto é, uma porta pode trabalhar em um modo e outra porta em outro modo, simultaneamente.

Para trabalhar com a 8255, a unidade central de processamento necessita apenas de duas instruções : uma para enviar um dado através do duto de dados para a interface e outra para receber um dado. Porém, este dado pode ser enviado para o registrador da palavra de controle; neste caso, o dado enviado será uma palavra para a programação da interface, ou então, o dado pode ser enviado ou recebido para qualquer uma das três portas e, neste caso, o dado será enviado ou proveniente diretamente de um dispositivo externo.

Vamos analisar como a unidade central de processamento seleciona uma das três portas ou o registrador da palavra de controle. Isso é feito por meio dos dois bits de endereço (A0 e A1), que são conectados da unidade central de processamento à interface. Os registradores são selecionados conforme as combinações dos bits de endereço, como pode ser visto na tabela B.1

A1	A0	Registrador
0	0	Porta A
0	1	Porta B
1	0	Porta C
1	1	Controle

Tabela B.1 - Endereçamento da 8255

As variáveis de controle \overline{RD} e \overline{WR} definirão se a operação é de leitura ou escrita, respectivamente, na interface.

Assim sendo, a comunicação da unidade central de processamento com a interface, é feita por meio das duas instruções de enviar ou receber dados da interface, sendo que as próprias instruções irão acionar adequadamente as variáveis de leitura ou escrita; estas instruções deverão conter o endereço, conforme o registrador desejado.

A programação da 8255 é feita através de uma palavra de controle que é enviada da unidade central de processamento para o registrador da palavra de controle. Esta palavra de controle é formada por oito bits (D7 - D0) e a definição de cada bit está mostrada na figura B.2.

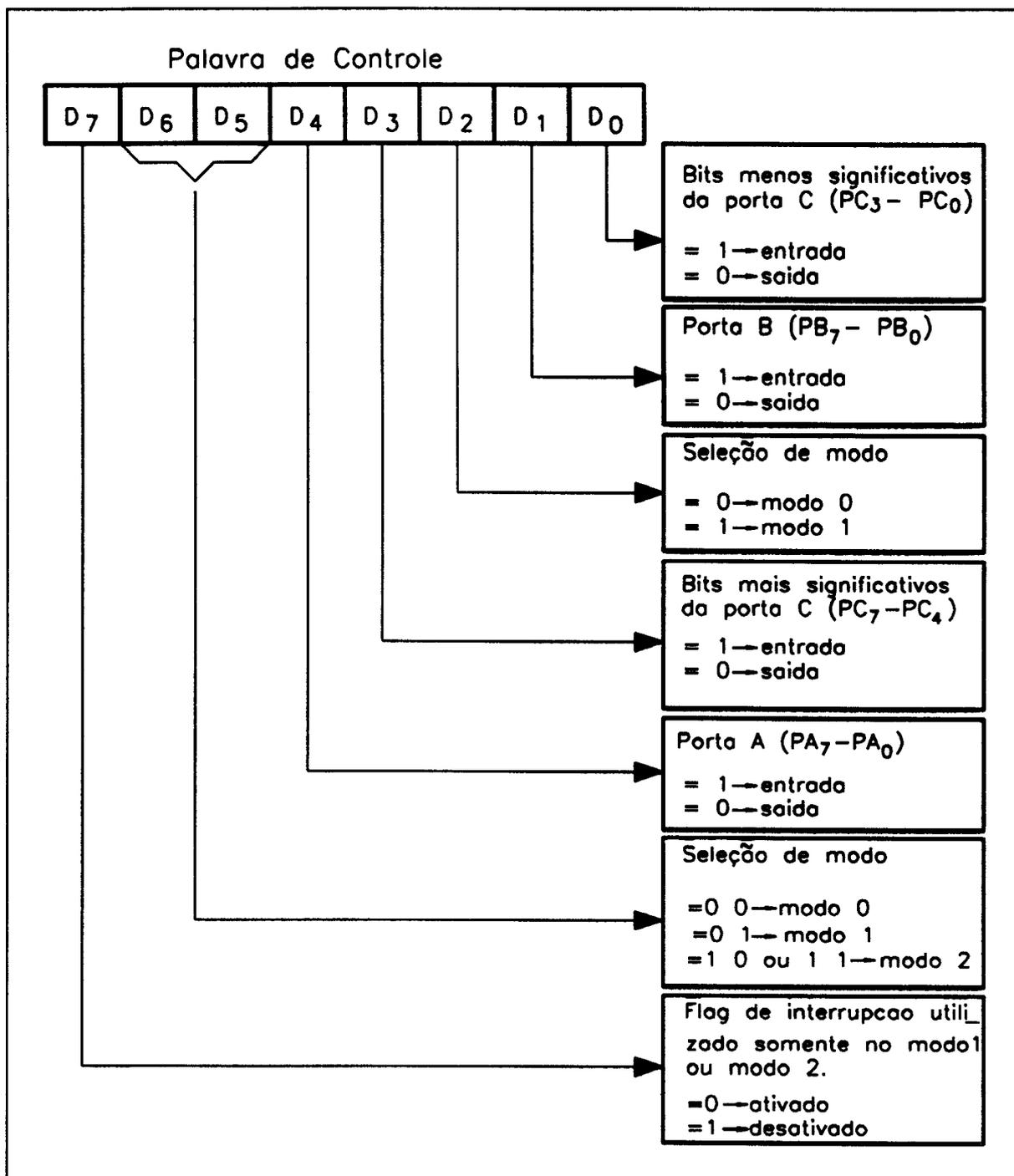


Figura B.2 - Definição da palavra de controle

Apêndice C - Listagem do Programa desenvolvido

PROGRAM Digitalizacao ;

```
{ *****
*
* Este programa fornece um acesso dirigido por menu para acessar
* quatro funcoes distintas :
*
*   - X-scan  --> digitaliza uma linha (X) do filme
*   - Y-scan  --> digitaliza uma coluna (Y) do filme
*   - Aquisicao --> armazena dados em um arquivo
*   - Sair    --> fim do programa
*
* Nas duas primeiras opcoes, traca-se o grafico da linha ou
* coluna digitalizada e o usuario ainda tem a opcao de ampliar
* ("zoom") uma regio da curva obtida.
* Na terceira opcao, o usuario digita o nome do arquivo onde os
* dados serao armazenados ( em disco ).
*
* As caracteristicas do menu sao acessadas pelas teclas das setas
* e opcoes destacadas.
*
* O programa utiliza ainda algumas UNITS :
*
*   - MenuUnit  --> contem rotinas sobre os menus
*   - CursorUnit --> contem rotinas sobre o movimento dos curso-
*                   res no ajuste da curva
*   - Scan_Unit  --> contem rotinas do scanner
*   - SelecaoUnit --> contem rotinas para execucao da opcao
*                   escolhida
*
***** }
```

USES CRT, DOS, GRAPH, MENUUNIT, CURSORUNIT, SCAN_UNIT, SELECAOUNTIT ;

```
{ *****
PROGRAMA PRINCIPAL
***** }
```

begin

ClrScr ;

{* Cria a primeira janela (texto) *}

SetWindow1 (2, 2, 78, 24) ;

ReverseVideo (TRUE) ;

GoToXY (20, 4) ;

writeln ('** PROGRAMA PARA DIGITALIZAR FILMES **') ;

ReverseVideo (False) ;

GoToXY (25, 7) ;

```

writeln ( 'X_Min_Filme = 0 [ mm ] ' );
GoToXY ( 25, 8 );
writeln ( 'X_Max_Filme = 100 [ mm ] ' );
GoToXY ( 25, 9 );
writeln ( 'Y_Min_Filme = 0 [ mm ] ' );
GoToXY ( 25, 10 );
writeln ( 'Y_Max_Filme = 100 [ mm ] ' );
GoToXY ( 15, 13 );

ReverseVideo ( True );
writeln ( 'Posicione o subsistema optico em : X = 0 ' );

GoToXY ( 15, 15 );
writeln ( 'Coloque o botao RASTER no valor desejado.' );

GoToXY ( 15, 17 );
write( 'E pressione a barra de espaco quando estiver pronto. ' );
ReverseVideo ( False );

GoToXY ( 70, 17 );
Repeat
  begin
    inSpace := Readkey ;
    if inSpace <> ' '
      then write ( bell ) ;
    end ;
  Until inSpace = ' ' ;

Init55 ;

GetRaster ;

GoToXY ( 25, 19 );
write ( 'RASTER = ', raster:0:4, ' mm' );

GoToXY ( 18, 22 );
write( 'Pressione a barra de espaco para continuar ... ' );
ReverseVideo ( False );

GoToXY ( 65, 22 );
Repeat
  begin
    inSpace := Readkey ;
    if inSpace <> ' '
      then write ( bell ) ;
    end ;
  Until inSpace = ' ' ;

SetWindowP ( 2, 4, 78, 24 );

InitializeMenu ;

```

```
{* Exibe o menu recursivo. *}

pos1 := 0 ;
resp1 := false ;
Repeat
  GetSelection (done) ;
Until done ;
Window ( 1, 1, 80, 25 ) ;
ClrScr;
end.
```

```
Unit MenuUnit ;
```

```
INTERFACE
```

```
USES CRT, DOS, GRAPH ;
```

```
TYPE
```

```
activities = ( x_scan, y_scan, aquisicao, quit );
```

```
{* O tipo activityRecord contem campos de informacao sobre as opcoes  
do menu.  *}
```

```
activityRecord = RECORD
```

```
    row,                { * linha e coluna * }  
    column : BYTE;      { * posicoes das opcoes do menu * }  
    menuString : STRING[35] { * string das opcoes do menu * }  
end;
```

```
CONST
```

```
columnPos = 24;  
optionDisplay = ' X Y A S ' ;
```

```
{* O array de atividades contem um record de informacoes sobre  
cada opcao do menu.  *}
```

```
activity : ARRAY[activities] OF activityRecord =
```

```
(( row: 8; column: columnPos;  
   menuString: 'X -scan - digitaliza uma linha (X).'),  
 ( row: 9; column: columnPos;  
   menuString: 'Y -scan - digitaliza uma coluna (Y).'),  
 ( row: 10; column: columnPos;  
   menuString: 'Aquisicao dos dados.' ),  
 ( row: 11; column: columnPos;  
   menuString: 'Sair.' ));
```

```
{* Codigo ASC dos cantos e linhas horizontal e vertical *}
```

```
UpLeftCorner = #201 ;  
HorzBar = #205 ;  
UpRightCorner = #187 ;  
VertBar = #186 ;  
LowLeftCorner = #200 ;  
LowRightCorner = #188 ;
```

```
VAR
```

```
currentSelection : activities ;  
Ch, inSpace : char ;  
i, x1, y1, x2, y2 : Integer ;
```

```

FUNCTION UpperCase ( inString : String ) : String ;
PROCEDURE ViewPort ( x1, y1, x2, y2 : Integer ) ;
PROCEDURE ReverseVideo ( status : Boolean ) ;
PROCEDURE HighlightSelection ;
PROCEDURE InitializeMenu ;
PROCEDURE Continue ;
PROCEDURE RemoveHighlight ;
PROCEDURE SelectNextActivity ;
PROCEDURE SelectPreviousActivity ;
PROCEDURE SetWindow1 ( x1, y1, x2, y2 : Integer ) ;
PROCEDURE SetWindowP ( x1, y1, x2, y2 : Integer ) ;

```

IMPLEMENTATION

```

{ *****
  FUNCTION UpperCase
  *****

```

Retorna com uma versao em letras maiusculas do argumento STRING que receber.

```
*)
```

```

FUNCTION UpperCase ( inString : String ) : String ;

```

```

Var

```

```

  i : Integer ;
  outString : String ;

```

```

begin

```

```

  outString := '' ;
  for i := 1 to length ( inString ) do
    begin
      outString := outString + UpCase ( inString[i] )
    end ;
  UpperCase := outString
end ;

```

```

{ *****
  PROCEDURE ReverseVideo
  *****

```

Troca o status de exibicao na tela para o video reverso ou normal, dependendo do valor do argumento BOOLEAN que a procedure recebe.

```
*)
```

```

PROCEDURE ReverseVideo ( status : Boolean ) ;

```

```

begin

```

```

if status then
  begin
    TextColor ( Black );
    TextBackground ( White )
  end
else
  begin
    TextColor ( White );
    TextBackground ( Black )
  end
end;

```

```

{ *****
PROCEDURE HighlightSelection
*****

```

Destaca a selecao do menu atual ou seja, exibe a selecao no modo de video reverso, e todas em letras maiusculas.

```

* }

```

```

PROCEDURE HighlightSelection;

```

```

begin
  ReverseVideo ( true );
  with activity[currentSelection] do
    begin
      GoToXY ( column, row );
      writeln ( UpperCase ( menuString ))
    end;
  ReverseVideo ( false );
  GoToXY (66,17)
end;

```

```

{ *****
PROCEDURE InitializeMenu
*****

```

Exibe o menu na tela e ajusta a currentSelection para a primeira opcao do menu.

```

* }

```

```

PROCEDURE InitializeMenu ;

```

```

VAR
  option : activities ;

```

```

begin
  clrscr;

```

```

  { * Exibe o titulo do menu * }

```

```

ReverseVideo ( true );
GoToXY (20,5);
writeln('** Opcoes para Digitalizacao de filmes **');
ReverseVideo ( false );

{ * Exibe as opcoes do menu *}

for option := x_scan to quit do
  with activity[option] do
    begin
      GoToXY(column, row);
      writeln(menuString)
    end;

{ * Estabelece e destaca a selecao atual. *}

currentSelection := x_scan ;
HighlightSelection;

{ * Exibe as instrucoes do teclado. *}

GoToXY (columnPos - 12, 16);
write('Use ');
ReverseVideo (true);
write(#24, ' ', #25, ' ', #26, ' ', #27, ' ');
ReverseVideo (false);
write(' ou ');
ReverseVideo (true);
write(optionDisplay);
ReverseVideo (false);
write(' para destacar uma opcao,');
GoToXY(columnPos - 10, 17);
write('depois, pressione <ENTER> para completar a selecao.');
```

end;

```

{ *****
PROCEDURE Continue
*****

Pede ao usuario para pressionar a barra de espaco, para retornar ao
menu depois de uma determinada atividade do menu ter sido selecionada
e executada.
*}

PROCEDURE Continue ;

begin
  GoToXY(15,24);
  write('Pressione a barra de espaco para retornar ao menu.');
```

```

    inSpace := ReadKey
    Until inSpace = ' ';
end;

```

```

{ *****
PROCEDURE RemoveHighlight
*****

```

Restaura uma opção rejeitada do menu para exibição normal, com letras maiúsculas e minúsculas.

```

* }

```

```

PROCEDURE RemoveHighlight ;

```

```

begin
  with activity[currentSelection] do
    begin
      GoToXY (column, row);
      writeln(menuString)
    end
  end;
end;

```

```

{ *****
PROCEDURE SelectNextActivity
*****

```

Destaca a próxima opção do menu abaixo na lista, respondendo as teclas das setas para baixo ou para direita pressionadas no teclado.

```

* }

```

```

PROCEDURE SelectNextActivity ;

```

```

begin
  RemoveHighlight ;
  if currentSelection = quit then
    currentSelection := x_scan
  else
    currentSelection := SUCC (currentSelection);
  HighlightSelection
end;

```

```

{ *****
PROCEDURE SelectPreviousActivity
*****

```

Destaca as opções do menu anteriores na lista, respondendo as teclas das setas para cima ou para esquerda pressionadas no teclado.

```

* }

```

```
PROCEDURE SelectPreviousActivity ;
```

```
begin
  RemoveHighlight ;
  if currentSelection = x_scan then
    currentSelection := quit
  else
    currentSelection := PRED (currentSelection);
  HighlightSelection
end;
```

```
{ *****
  PROCEDURE SetWindow1
  *****
```

Define uma janela em modo texto, com coordenadas superior X1 e Y1, e inferiores X2 e Y2. Esta janela e do tamanho da tela.

```
*}
```

```
PROCEDURE SetWindow1 ( x1, y1, x2, y2 : Integer ) ;
```

```
begin
  Window ( x1-1, y1-1, x2+1, y2+1 ) ;
  ClrScr ;
  Window ( 1, 1, 80, 25 ) ;

  GotoXY( x1-1, y1-1 ) ; { * desenha linha horz. superior da janela * }
  Write(UpLeftCorner) ;
  for i := x1 to x2 do
    write(HorzBar) ;
  write(UpRightCorner) ;

  for i := y1 to y2 do begin { * desenha linha vert. da janela * }
    GoToXY ( x1-1, i ) ; write(VertBar) ;
    GoToXY ( x2+1, i ) ; write(VertBar) ;
  end ;

  GoToXY ( x1-1, y2+1 ) ; { * desenha linha horiz. inferior da jan.* }
  Write(LowLeftCorner) ;
  for i := x1 to x2 do
    write(HorzBar) ;
  write(LowRightCorner) ;

  Window ( x1, y1, x2, y2 ) ;
end ;
```

```
{ *****
  PROCEDURE SetWindowP
  *****
```

Define uma janela em modo texto, com coordenadas superior X1 e Y1, e inferiores X2 e Y2. Esta janela e do tamanho da tela e contem o menu principal.

```
*)
```

```
PROCEDURE SetWindowP ( x1, y1, x2, y2 : Integer ) ;
begin
  Window ( x1-1, y1-1, x2+1, y2+1 ) ;
  ClrScr ;
  Window ( 1, 1, 80, 25 ) ;

  GoToXY ( x1-1, y1-1 ) ; { * desenha borda da janela *}
  Write(UpLeftCorner) ;
  for i := x1 to x2 do
    write(HorzBar) ;
  write(UpRightCorner) ;

  for i := y1-2 to y2 do begin
    GoToXY ( x1-1, i ) ; write(VertBar) ;
    GoToXY ( x2+1, i ) ; write(VertBar) ;
  end ;
  GoToXY ( x1-1, y2+1 ) ;
  Write(LowLeftCorner) ;
  for i := x1 to x2 do
    write(HorzBar) ;
  write(LowRightCorner) ;

  GotoXY( x1-1, y1-3 ) ; { * linha horz. superior *}
  Write(UpLeftCorner) ;
  for i := x1 to x2 do
    write(HorzBar) ;
  write(UpRightCorner) ;

  GotoXY(x1+25,y1-2); { * escreve texto da janela *}
  TextBackground(Lightgray) ;
  TextColor(Black) ;
  writeln(' MENU PRINCIPAL ');
  TextColor(LightGray) ;
  TextBackground(Black) ;

  Window ( x1, y1, x2, y2 ) ;
end ;
```

```
{ *****
PROCEDURE ViewPort
***** }
```

Abre uma janela grafica com as coordenadas de entrada e desenha uma moldura do tamanho da janela. *}

```
PROCEDURE ViewPort ( x1, y1, x2, y2 : Integer ) ;  
begin  
  Rectangle ( x1, y1, x2, y2 ) ;  
  Rectangle ( ( x1 + x2 ), ( y1 + 2 ), ( x2 - 2 ), ( y2 - 2 ) ) ;  
  SetViewPort ( x1, y1, x2, y2, ClipOn ) ;  
end ;  
  
end.
```

```

UNIT SelecaoUnit ;

INTERFACE

{$L Ledata1 }

USES CRT, DOS, GRAPH, MENUUNIT, CURSORUNIT, SCAN_UNIT ;

TYPE

    vet_Dados = array [1..4000] of byte ;

    vet_Dados_G = array [1..400] of real ;

    arq_dados = string[12] ;

CONST

    { * O conjunto menuChars contem todos os caracteres validos que
      o usuario pode pressionar em resposta ao menu.      * }

    menuChars: set of char = ['X', 'Y', 'A', 'S', nullchar, enter] ;

    { * O conjunto cursorScanCodes contem os codigos para as quatro
      teclas das setas.      * }

    cursorScanCodes : set of char = [upArrow, leftArrow, rightArrow,
                                     downArrow] ;

    { * Limites do filme ( em milimetros ) * }

    x_min_f = 0 ;
    x_max_f = 100 ;
    y_min_f = 0 ;
    y_max_f = 100 ;

    num_ptos_X = 400 ;    { * numero ptos do eixo X * }
    num_ptos_Y = 145 ;    { * numero ptos do eixo Y * }
    UM = #49 ;           { * numero 1 * }
    DOIS = #50 ;         { * numero 2 * }
    TRES = #51 ;         { * numero 3 * }

VAR

    Dados, Dados1 : vet_Dados ;

    Dados_G : vet_Dados_G ;

    nome_arq : arq_dados ;

```

```

dados_d : text ;

done, resp, resp1 : BOOLEAN ;

data : string[10] ;

coment : string[70] ;

Ch, opcao1, opcao, opcao2, opcao3,

opcao4 : char ;          { * opcoes dos cursores e setas * }

driverVar , modeVar,    { * var utilizadas no InitGraph * }

k, j, q, z, i,         { * var contadoras * }

i_max,

ptos_X,                { * num de ptos em X ( para ajuste ) * }

k_max,                 { * contem indice do vetor com den max * }

i_min, k_min,         { * contem indice do vetor com den min * }

jerr, complem2,

ptos_saltar, ptos_coletar,

coleta, t : INTEGER ;

den_max, den_min,     { * densidades maximas e minimas * }

fator_Y, fator1_Y,    { * fatores de escala para o eixo Y
                      sem e com "zoom" ( respectivamente ) * }

den1_max, den1_min,   { * densidades maxima e minima com "zoom" * }

coluna, start, limite,

size, pos2, dx_pto, dx1,

pos1, pos_final : REAL ;

texto1 : string[30] ;

```

```

PROCEDURE GetSelection (VAR quitSignal : BOOLEAN ) ;
PROCEDURE ledatal ;

```

IMPLEMENTATION

```
{ *****
  PROCEDURE ledatal
  *****
```

Rotina em Assembler para fazer a coleta dos dados (densidades).
E declarada como EXTERNAL.

```
*}
```

```
procedure ledatal ; external ;
```

```
{ *****
  PROCEDURE GetSelection
  *****
```

Aceita uma selecao do menu pelo teclado, e executa o programa
apropriado em resposta.

```
*}
```

```
procedure GetSelection (VAR quitSignal : BOOLEAN ) ;
```

```
CONST
```

```
  firstChars = 'XYAS' ;
```

```
VAR
```

```
  inChar : CHAR ;
```

```
begin
```

```
  quitSignal := FALSE ;
```

```
  { * Espera o usuario pressionar uma tecla valida em resposta ao
    menu.  * }
```

```
  Repeat
```

```
    inChar := UPCASE (READKEY) ;
```

```
    if NOT ( inChar IN menuChars ) then
```

```
      write (bell)
```

```
  Until ( inChar IN menuChars );
```

```
  { * Responde apropriadamente a digitacao  * }
```

```
  CASE inChar OF
```

```
    'X', 'Y', 'A', 'S' :
```

```
      begin
```

```
        RemoveHighlight ;
```

```
        currentSelection := activities(POS(inChar.firstChars) - 1) ;
```

```
        HighlightSelection
```

```
      end;
```

nullChar :

```
begin
  inChar := READKEY ;
  if inChar IN cursorScanCodes then
    CASE inChar OF
      upArrow, leftArrow : SelectPreviousActivity ;
      downArrow, rightArrow : SelectNextActivity
    end
  else
    write (bell)
  end;
end;
```

{ * Executa um programa quando o usuario pressionar a tecla ENTER
para confirmar a selecao do menu atual * }

enter :

```
begin
  if currentSelection = quit then
    quitSignal := TRUE
  else
    begin
      ClrScr;

      Case currentSelection of
        x_scan,
        y_scan : begin
          resp1 := true ;

          driverVar := 0 ;
          InitGraph (driverVar, modeVar, '');

          { *****
            Abre janela - 1
            ***** }

          ViewPort ( 470, 0, 639, 199 ) ;

          if ( currentSelection = y_scan )
          then { * varredura da coluna * }
          begin
            opcao2 := 'y' ;
            Repeat { * repita 02 * }
            begin
              Repeat { * repita 01 * }
              begin
                OutTextXY(10,10,'COLUNA = ');
                Str(pos1:0:2,texto);
                OutTextXY(100,10,texto);
                OutTextXY ( 15, 20, 'Digite a posicao ');
                OutTextXY ( 15, 28, '(em mm) da coluna a');
              end
            end
          end
        end
      end;
    end;
  end;
```

```

OutTextXY ( 15, 36, 'ser digitalizada :');
OutTextXY ( 15, 48, 'COLUNA = ' );
GoToXY ( 70, 4 );

readln ( coluna );

if ( coluna > 100 ) or ( coluna <= 0 )
then
begin
jerr := 1 ;
ClearViewPort ;
SetViewPort ( 0, 0, 639, 199,ClipOn);
ViewPort ( 470, 0, 639, 199 );
end

else { * else 02 * }
begin
jerr := 0 ;
pos2 := coluna ;
if ( pos1 = 0 )
then
begin
passo := 'd' ;
pos1 := coluna ;
end
else { * else 01 * }
begin
if pos1 < pos2
then
begin
passo := 'd' ;
end
else
begin
passo := 'e' ;
end ;

coluna := Abs ( coluna - pos1 ) ;
pos1 := pos2 ;
end ; { * fim do else 01 * }

OutTextXY ( 15, 58, 'AGUARDE... ' ) ;

n_passosX := Round (( coluna * 1000 ) / 12.5 );
passoX ( n_passosX, passo ) ;

t := ( Round (coluna/3) ) * 1000 ;
Delay (t) ;

end ; { * fim else 02 * }

end ; { * fim do Repita 01 * }

```

```

Until ( jerr = 0 ) ;

Repeat  { * repita 03 * }
begin
  OutTextXY ( 15, 70, 'Digite a posicao' ) ;
  OutTextXY ( 15, 80, 'inicial (mm) para' ) ;
  OutTextXY ( 15, 90, 'inicio da tomada ' ) ;
  OutTextXY ( 15, 100, 'dos dados e o ' ) ;
  OutTextXY ( 15, 110, 'tamanho da coluna' ) ;
  OutTextXY ( 15, 120, '(mm) dos dados. ' ) ;
  OutTextXY ( 15, 135, 'START = ' ) ;
  GoToXY ( 70, 15 ) ;

  readln ( start ) ;

  if (start <= 100) and (start >= 0)
  then
    begin
      OutTextXY ( 15, 152, 'SIZE = ' ) ;
      GoToXY ( 70, 17 ) ;

      readln ( size ) ;
      if (size < 0)
      then
        resp := false
      else
        begin
          limite := 100 - start ;
          if size > limite
          then
            begin
              resp := false ;
            end
          else
            begin
              resp := true
            end ;
          end ;
        end ;

      end  { * fim do then * }

    else
      begin
        resp := false ;
      end ;

    ClearViewPort ;
    SetViewPort(0,0,639,199.ClipOn);
    ViewPort ( 470, 0, 639, 199 ) ;

  end ; { * fim do repita 03 * }
Until ( resp = true ) ;

```

```

ptos_saltar := Round( start / raster );
ptos_coletar := Round( size / raster );

complem2 := Not ( ptos_saltar ) + 1 ;
Port[$301] := Hi ( complem2 ) ;
Port[$302] := Lo ( complem2 ) ;

Port[$300] := $BF ; { * pulso em EX6 * }
Port[$300] := $FF ;

Ledata1 ;

Repeat { * repita 05 * }
begin
  OutTextXY(15,63,'Digitaliza outra');
  OutTextXY(15,73,'  coluna?');
  OutTextXY(15,83,' ( S / N )');
  opcao3 := Readkey ;
  if (opcao3 = 'S') or (opcao3 = 's')
  then
    begin
      ClearViewPort ;
      SetViewPort(0,0,639,199,ClipOn);
      ViewPort ( 470, 0, 639, 199 );
    end
  else
    begin
      if (opcao3 <> 'N') and (opcao3 <> 'n')
      then
        write ( bell )
      end ;
    end ;
  end ; { * fim do repita 05 * }

Until ( opcao3 = 'S' ) or (opcao3 = 's') or
  ( opcao3 = 'N' ) or (opcao3 = 'n') ;

end ; { * fim do Repita 02 * }
Until ( opcao3 = 'N' ) or ( opcao3 = 'n' ) ;

end { * fim do then varredura da coluna * }

else { * varredura da linha * }
begin
  opcao2 := 'x' ;
  Repeat { * repita 00 * }
  begin
    Repeat { * repita 01 * }
    begin
      OutTextXY(15,20,'Digite a posicao ');
      OutTextXY(15,28,'(em mm) da linha ');
      OutTextXY(15,36,'a ser varrida : ');
      OutTextXY ( 15, 48, 'START = ' );
    end
  end
end

```

```

GoToXY ( 71, 4 ) ;

readln ( start ) ;

if (start <= 100) and (start >= 0)
then
begin
  resp := true ;
end
else
begin
  resp := false ;
  ClearViewPort ;
  SetViewPort(0,0,639,199,ClipOn);
  ViewPort ( 470, 0, 639, 199 ) ;
end ;

end ; { * fim do Repita 01 * }
Until ( resp = true ) ;

ptos_saltar := Round( start / raster );

Repeat { * repita 03 * }
begin
  OutTextXY(10,70,'COLUNA = ');
  Str(pos1:0:2,texto);
  OutTextXY(100,70,texto);
  OutTextXY(15,80,'Digite a posicao');
  OutTextXY(15,90,'inicial (mm) para');
  OutTextXY(15,100, 'inicio da tomada ');
  OutTextXY(15,110, 'dos dados : ' );
  OutTextXY(15,120,'COLUNA = ');
  GoToXY ( 71, 13 ) ;

  readln ( coluna ) ;

  if (coluna <= 100 ) and (coluna > 0)
  then
  begin
    resp := true ;
    pos2 := coluna ;
    if pos1 = 0
    then
    begin
      passo := 'd' ;
      pos1 := coluna ;
    end
    else
    begin
      if pos1 < pos2
      then
      begin

```

```

        passo := 'd' ;
    end
else
    begin
        passo := 'e' ;
    end ;

    coluna := Abs ( coluna - pos1 ) ;
    pos1 := pos2 ;
end ;

OutTextXY(15,130, 'AGUARDE... ');

n_passosX := Round(( coluna * 1000 ) / 12.5 ) ;

{ * ptos p/ andar ate inicio da linha * }
passoX ( n_passosX, passo ) ;

t := ( Round (coluna/3) ) * 1000 ;
Delay ( t ) ;

end { * fim do then * }

else
    begin
        resp := false ;
        ClearViewPort ;
        SetViewPort(0,0,639,199,ClipOn);
        ViewPort ( 470, 0, 639, 199 ) ;
    end ;

end ; { * fim do Repita 03 * }
Until ( resp = true ) :

Repeat { * repita 3.1 * }
begin
    OutTextXY(15,140, 'Digite a posicao ');
    OutTextXY(15,150, 'final em (mm) ');
    OutTextXY ( 15,169, 'POS_FINAL = ' ) ;
    GoToXY ( 73, 19 ) ;

    readln ( pos_final ) ;

    limite := 100 - pos1 ;
    if (pos_final > limite) or (pos_final < 0)
    then
        begin
            resp := false ;
            ClearViewPort ;
            SetViewPort(0,0,639,199,ClipOn);
            ViewPort ( 470, 0, 639, 199 ) ;

```

```

        OutTextXY(15,110,'START = ');
        Str(coluna:0:1,texto);
        OutTextXY(70,110,texto);
    end
else
    begin
        resp := true ;
        ClearViewPort ;
        SetViewPort(0,0,639,199,ClipOn);
        ViewPort ( 470, 0, 639, 199 ) ;
    end ;

end ; { * fim do Repita 3.1 * }
Until ( resp = true ) ;

size := Abs ( pos1 - pos_final ) ;

ptos_coletar := Round( size / raster);

if pos1 < pos_final
then
    begin
        passo := 'd' ;
    end
else
    begin
        passo := 'e' ;
    end ;

pos1 := pos_final ;

complem2 := Not ( ptos_saltar ) + 1 ;

coleta := ptos_coletar ;
ptos_coletar := 1 ;

if raster = 0.025
then
    n_passosX := 2
else
    begin
        if raster = 0.050
        then
            n_passosX := 4
        else
            n_passosX := 8 ;
        end;

OutTextXY ( 15, 162, 'AGUARDE... ' ) ;
OutTextXY ( 15,172,'Fazendo leitura');
OutTextXY ( 15,182,'dos dados...');

```

```

for q := 1 to coleta do
  begin

    Port[$300] := $EF ; { * pulso em EX4 * }
    Port[$300] := $FF ; { * limpa comando * }

    Port[$301] := Hi ( complem2 ) ;
    Port[$302] := Lo ( complem2 ) ;

    Port[$300] := $BF ; { * pulso em EX6 * }
    Port[$300] := $FF ;

    Ledata1 ;

    Dados1 [q] := Dados [1] ;

    passoX ( n_passosX, passo ) ;

    Delay(100) ;

  end ;

for i := 1 to coleta do
  begin
    Dados[i] := Dados1[i] ;
  end ;

ptos_coletar := coleta ;

Repeat { * repita 04 * }
begin
  OutTextXY(15,63,'Digitaliza outra');
  OutTextXY(15,73,' linha ?');
  OutTextXY(15,83,' ( S / N )');
  opcao3 := Readkey ;
  if (opcao3 = 'S') or (opcao3 = 's')
  then
    begin
      ClearViewPort ;
      SetViewPort(0,0,639,199,ClipOn);
      ViewPort ( 470, 0, 639, 199 );
    end
  else
    begin
      if (opcao3 <> 'N') and (opcao3 <> 'n')
      then
        write ( bell ) ;
      end ;
    end ;
end ; { * fim do Repita 04 * }

Until ( opcao3 = 'S' ) or (opcao3 ='s') or
( opcao3 = 'N' ) or (opcao3 ='n') ;

```

```

end ; { * fim do Repita 00 * }
Until ( opcao3 = 'N' ) or ( opcao3 = 'n' ) ;

end ; { * fim do else varredura da linha * }

ClearViewPort ;
SetViewPort ( 0, 0, 639, 199, ClipOn );
ViewPort(470,0,639,199);

SetViewPort ( 0, 0, 639, 199, ClipOn ) ;

{ *****
  Abre janela - 2 ( desenha grafico )
  ***** }

ViewPort ( 0, 0, 468, 199 ) ;
line ( 15, 170, 450, 170 ) ; { * eixo horiz * }
line ( 35, 17, 35, 180 ) ; { * eixo vert * }

OutTextXY (140,185,'Distancia ( *') );

if ( currentSelection = x_scan )
then
  begin
    OutTextXY ( 210, 5, 'X - Scan' ) ;
  end
else
  begin
    OutTextXY ( 210, 5, 'Y - Scan' ) ;
  end ;

Str ( raster:0:4, texto ) :
OutTextXY(260,185,texto);
OutTextXY (320,185,'mm ') );

OutTextXY ( 10, 7, 'Densidade' ) ;

{ *****
  Verificar densidades MAX. e MIN.
  ***** }

den_min := Dados[1] ;
den_max := Dados[1] ;
i_max := 1 ;
i_min := 1 ;
for i := 2 to ptos_coletar do
  begin
    if Dados[i] > den_max
    then
      begin
        den_max := Dados[i] ;
        i_max := i ;

```

```

    end ;
  if Dados[i] < den_min
  then
    begin
      den_min := Dados[i] ;
      i_min := i ;
    end
  end ;

{*****
Fator de Escala para os pontos Y
*****}

fator_Y := num_ptos_Y / den_max ;

{*****
Deslocamento para os pontos X
e num. ptos p/ tracar o grafico
*****}

if ptos_coletar > 400
then
  begin
    dx := 1 ;
    dx_pto:=Int(ptos_coletar / num_ptos_X) + 1
  end
else
  begin
    dx := Int ( num_ptos_X / ptos_coletar ) ;
    dx_pto := 1 ;
  end ;

{*****
Calculo dos ptos e Desenho do Grafico
*****}

x := 40 ;
j := 1 ;
i := 1 ;
while j <= ptos_coletar do
  begin
    Y[i] := 170 - Round ( Dados[j] * fator_Y ) ;
    PutPixel ( x, Y[i], 15 ) ;
    Dados_G[i] := Dados[j] ;
    x := Round ( x + dx ) ;
    j := Round ( j + dx_pto ) ;
    i := i + 1 ;
  end ;
x := Round ( x - dx ) ;

```

```

j := Round ( j - dx_pto ) ;
i := i - 1 ;

{ * limites do grafico * }
denl_min := den_min ;
denl_max := den_max ;

{ * limite esquerdo do cursor ESQUERDO * }
x_L_corr := 40 ;

{ * primeiro pto Y do cursor ESQUERDO * }
y_L_corr := 1 ;

{ * limite direito do cursor DIREITO * }
x_R_corr := x ;

{ * primeiro pto Y do cursor DIREITO * }
y_R_corr := i ;

{ * limite superior dos cursores * }
x_medio := Round ( ( x_R_corr +
                    x_L_corr ) / 2 ) ;

{ *****
Escala do eixo X
***** }

Str ( (((x_L_corr - 40) * dx_pto )/dx):0:1,texto ) ;
OutTextXY(x_L_corr,174,texto);

Str ( (((x_R_corr - 40) * dx_pto )/dx):0:1,texto);
OutTextXY(x_R_corr, 174,texto);

{ *****
Escala do eixo Y
***** }

Y[i_max] := 170 - Round ( Dados[i_max]
                        * fator_Y );

Y[i_min] := 170 - Round ( Dados[i_min]
                        * fator_Y );

Str(Dados[i_max], texto ) ;
OutTextXY(2, Y[i_max], texto);

Str(Dados[i_min], texto ) ;
OutTextXY(2, Y[i_min], texto);

```

```

SetViewPort ( 0, 0, 639, 199, ClipOn ) ;

{ *****
  Abre janela - 1
  ***** }

ViewPort ( 470, 0, 639, 199 ) ;
ClearViewPort ;
SetViewPort ( 0, 0, 639, 199, ClipOn ) ;
ViewPort ( 470, 0, 639, 199 ) ;

{ *****
  Limites do Grafico
  ***** }

OutTextXY ( 5, 130, 'Limites do Grafico :' ) ;

OutTextXY ( 15, 150, 'X_min = 0 ' ) ;

OutTextXY ( 15, 160, 'Y_min =' ) ;
Str ( den_min:0:1, texto1 ) ;
OutTextXY ( 80, 160, texto1 ) ;

OutTextXY ( 15, 170, 'X_max = ' ) ;
Str ( ( ( x - 40 ) * dx_pto )/dx):0:1, texto1 ) ;
OutTextXY ( 80, 170, texto1 ) ;

OutTextXY ( 15, 180, 'Y_max =' ) ;
Str ( den_max:0:1, texto1 ) ;
OutTextXY ( 80, 180, texto1 ) ;

OutTextXY ( 15, 80, 'Amplia ("zoom")' ) ;
OutTextXY ( 15, 90, ' a curva?' ) ;
OutTextXY ( 15, 100, ' S --> SIM ' ) ;
OutTextXY ( 15, 110, ' N --> NAO ' ) ;

{ *****
  Coordenadas do filme que foi feito Scan
  ***** }

if ( currentSelection = x_scan )
then
begin
  OutTextXY ( 5, 10, 'Coord. do filme:(mm)' ) ;

  OutTextXY ( 15, 30, 'Y_filme =' ) ;
  Str ( start:0:1, texto1 ) ;
  OutTextXY ( 100, 30, texto1 ) ;

```

```

OutTextXY ( 15, 40, 'X_inicial =' );
Str ( pos2:0:1, texto1 );
OutTextXY ( 110, 40, texto1 );

OutTextXY ( 15, 50, 'X_final =' );
Str ( pos_final:0:1, texto1 );
OutTextXY ( 110, 50, texto1 );

OutTextXY ( 15, 60, 'Tam da Lin X =' );
Str ( size:0:1, texto1 );
OutTextXY ( 130, 60, texto1 );
end
else
begin
OutTextXY ( 5, 10, 'Coord. do filme:(mm)' );

OutTextXY ( 15, 30, 'X =' );
Str ( pos1:0:1, texto1 );
OutTextXY ( 50, 30, texto1 );

OutTextXY ( 15, 40, 'Y_inicial =' );
Str ( start:0:1, texto1 );
OutTextXY ( 110, 40, texto1 );

OutTextXY ( 15, 50, 'Y_final =' );
Str ( (start+size):0:1, texto1 );
OutTextXY ( 110, 50, texto1 );

OutTextXY ( 15, 60, 'Tam da Col Y =' );
Str ( size:0:1, texto1 );
OutTextXY ( 130, 60, texto1 );

end ;

SetViewPort ( 0, 0, 639, 199, ClipOn );

Repeat
opcao1 := ReadKey ;

Case opcao1 of
's', 'S' : begin
opcao4 := 's' ;
{ *****
janela - 1
***** }

ViewPort ( 470, 0, 639,199);
ClearViewPort ;
SetViewPort(0,0,639,
199,ClipOn);
ViewPort ( 470, 0, 639,199);
OutTextXY(50,6,

```

```

        'Janela - 1 ');
line ( 5, 15, 165, 15 );

OutTextXY(15,20,
    'Digite a OPCA0 : ');
OutTextXY (15,30,
    '1 : para cursor' );
OutTextXY ( 15, 40,
    '  ESQUERDO ' );
OutTextXY ( 15, 50,
    '2 : para cursor' );
OutTextXY ( 15, 60,
    '  DIREITO' );
OutTextXY ( 15, 70,
    '3 : FIM ' );
OutTextXY ( 15, 90,
    'Pressione as teclas');
OutTextXY ( 15, 100,
    "' --> " ou " <-- ");
OutTextXY ( 15, 110,
    'e selecione a' );
OutTextXY ( 15, 120,
    'posicao do cursor' );
OutTextXY ( 15, 130,
    'com <ENTER>' );

SetViewPort(0,0,639,
    199,ClipOn);

{ *****
  Abre janela - 2 ( desenha grafico )
  ***** }

ViewPort( 0, 0, 468, 199 );

{ * cursor inicial * }
line (x_L_corr, Y[y_L_corr]+10,
    x_L_corr, Y[y_L_corr]-10);

{ * cursor final * }
line (x_R_corr, Y[y_R_corr]+10,
    x_R_corr, Y[y_R_corr]-10);

{ * cursor inicial * }
line (x_L_corr, Y[y_L_corr]+10,
    x_L_corr, Y[y_L_corr]-10);

{ * cursor final * }
line (x_R_corr, Y[y_R_corr]+10,
    x_R_corr, Y[y_R_corr]-10);

```

```

{*****
  Opcoes dos movimentos dos cursores
*****}

```

```

Repeat
  opcao := ReadKey ;
  Case opcao of
    UM : begin { * cursor esquerdo *}
      Repeat
        seta := ReadKey ;
        if seta <> #0
        then
          begin
            if seta <> enter
            then write ( bell )
          end
        else
          begin
            seta := ReadKey ;
            le_tecla1 ( seta ) ;
          end ;
        Until seta = enter ;
      end ;

    DOIS : begin { * cursor dir *}
      Repeat
        seta := ReadKey ;
        if seta <> #0
        then
          begin
            if seta <> enter
            then write (bell)
          end
        else
          begin
            seta := ReadKey ;
            le_tecla2 (seta);
          end ;
        Until seta = enter ;
      end ;

    else
      begin
        if ( opcao <> TRES )
        then
          write ( bell ) ;
        end ;
      end ; { * fim Case *}
  Until ( opcao = TRES ) ;

```

```

{*****
  Abre janela - 2 ( desenha grafico com "zoom" )
  *****}

  ViewPort ( 0, 0, 468, 199 ) ;
  ClearViewPort ;
  SetViewPort (0,0,639,199,ClipOn);
  ViewPort ( 0, 0, 468, 199 ) ;

  {* eixo horizontal *}
  line ( 15, 170, 450, 170 ) ;

  {* eixo vertical *}
  line ( 35, 17, 35, 180 ) ;

  OutTextXY (140,185,'Distancia ( *') ;

  if ( currentSelection = x_scan )
  then
    begin
      OutTextXY ( 210, 5, 'X - Scan' ) ;
    end
  else
    begin
      OutTextXY ( 210, 5, 'Y - Scan' ) ;
    end ;

  Str ( raster:0:4, texto ) ;
  OutTextXY(260,185,texto);
  OutTextXY (320,185,'mm ');

  OutTextXY ( 10, 7, 'Densidade' ) ;

  ptos_X := 0 ;
  denl_max := Dados_G[y_L_corr] ;
  denl_min := Dados_G[y_L_corr] ;
  k_max := y_L_corr ;
  k_min := y_L_corr ;

  for k := (y_L_corr+1) to y_R_corr do
  begin
    ptos_X := ptos_X + 1 ;
    if Dados_G[k] > denl_max
    then
      begin
        denl_max := Dados_G[k] ;
        k_max := k ;
      end;
    if Dados_G[k] < denl_min
    then
      begin
        denl_min := Dados_G[k] ;

```

```

        k_min := k ;
    end
end ;

```

```

{ *****
  Fator de Escala para os pontos Y ("zoom")
  ***** }

```

```

    fator1_Y := num_ptos_Y / den1_max ;

```

```

{ *****
  Deslocamento para os pontos X ("zoom")
  ***** }

```

```

    dx1 := Int ( num_ptos_X / ptos_X ) ;

```

```

{ *****
  Calculo dos ptos e Desenho do Grafico com "zoom"
  ***** }

```

```

    x := 40 ;
    for k := y_L_corr to y_R_corr do
    begin
        Y[k] := 170-Round(Dados_G[k]*fator1_Y);
        PutPixel ( x, Y[k], 15 ) ;
        x := Round ( x + dx1 ) ;
    end ;
    x := Round ( x - dx1 ) ;

```

```

{ *****
  Escala do eixo X ("zoom")
  ***** }

```

```

    Str((( x_L_corr - 40 ) * dx_pto)/dx):0:1,texto) ;
    OutTextXy ( 40, 174, texto ) ;

```

```

    Str((( x_R_corr -40) * dx_pto)/dx):0:1,texto) ;
    OutTextXy ( x, 174, texto ) ;

```

```

{ *****
  Escala do eixo Y ("zoom")
  ***** }

```

```

Y[k_max] := 170 - Round ( Dados_G[k_max]
                        * fator1_Y );

```

```

Y[k_min] := 170 - Round ( Dados_G[k_min]

```

```

* fator1_Y );

Str(Dados_G[k_max]:0:0, texto );
OutTextXY(2, Y[k_max], texto);

Str(Dados_G[k_min]:0:0, texto );
OutTextXY(2, Y[k_min], texto);

SetViewPort (0,0,639,199,ClipOn);

{ *****
  Abre janela - 1
  ***** }

ViewPort ( 470, 0, 639, 199 );
ClearViewPort ;
SetViewPort (0,0,639,199,ClipOn);
ViewPort ( 470, 0, 639, 199 );

{ *****
  Limites do Grafico com "zoom"
  ***** }

OutTextXY(5,10,'Limites do Grafico :');

OutTextXY ( 15, 30, 'X_min = ' );
Str(((( x_L_corr - 40 ) * dx_pto)/dx):0:1,texto1);
OutTextXY ( 80, 30, texto1 );

OutTextXY ( 15, 40, 'Y_min =' );
Str ( den1_min:0:1, texto1 );
OutTextXY ( 80, 40, texto1 );

OutTextXY ( 15, 50, 'X_max = ' );
Str((((x_R_corr - 40 ) * dx_pto)/dx):0:1,texto1);
OutTextXY ( 80, 50, texto1 );

OutTextXY ( 15, 60, 'Y_max =' );
Str ( den1_max:0:1, texto1 );
OutTextXY ( 80, 60, texto1 );

{ *****
  Coordenadas do filme que foi feito Scan
  ***** }

if ( currentSelection = x_scan )
then
begin
  OutTextXY ( 5, 90, 'Coord. do filme:(mm)' );

```

```

OutTextXY ( 15, 110, 'Y_filme =' );
Str ( start:0:1, texto1 );
OutTextXY ( 110, 110, texto1 );

OutTextXY ( 15, 120, 'X_inicial =' );
Str ( pos2:0:1, texto1 );
OutTextXY ( 110, 120, texto1 );

OutTextXY ( 15, 130, 'X_final =' );
Str ( pos_final:0:1, texto1 );
OutTextXY ( 110, 130, texto1 );

OutTextXY ( 15, 140, 'Tam da Lin X =' );
Str ( size:0:1, texto1 );
OutTextXY ( 130, 140, texto1 );
end
else
begin
OutTextXY ( 5, 90, 'Coord do filme:(mm)' );

OutTextXY ( 15, 110, 'X =' );
Str ( pos1:0:1, texto1 );
OutTextXY ( 50, 110, texto1 );

OutTextXY ( 15, 120, 'Y_inicial =' );
Str ( start:0:1, texto1 );
OutTextXY ( 110, 120, texto1 );

OutTextXY ( 15, 130, 'Y_final =' );
Str ( (start+size):0:1, texto1 );
OutTextXY ( 110, 130, texto1 );

OutTextXY ( 15, 140, 'Tam da Col Y =' );
Str ( size:0:1, texto1 );
OutTextXY ( 130, 140, texto1 );
end ;

OutTextXY(15,160,'Pressione a barra ');
OutTextXY(15,170,'de espaco para');
OutTextXY(15,180,'continuar...');
Repeat
begin
inSpace := Readkey ;
if inSpace <> ' '
then write ( bell );
end ;
Until inSpace = ' ';

opcaol := 'n' ;

end ; { * fim do SIM *}

```

```

else
  begin
    if ( opcao1 <> 'n' ) and ( opcao1 <> 'N' )
    then
      begin
        write (bell ) ;
      end ;

    end ;

  end ; { * fim case opcao1 *}

Until ( opcao1 = 'n' ) or ( opcao1 = 'N' ) ;

CloseGraph ;

end; { * x_scan *}

aquisicao : begin
  Window ( 1, 1, 80, 25 ) ;
  ClrScr;
  SetWindow1 (2, 2, 78, 24 ) ;

  if resp1 = true
  then
    begin
      GoToXY ( 12, 2 ) ;
      write ( 'Digite nome do arquivo ' ) ;
      write ( 'para armazenar os dados : ' ) ;
      readln ( nome_arq ) ;

      Assign ( dados_d, nome_arq ) ;
      Rewrite ( dados_d ) ;

      writeln ( dados_d, 'Arquivo de dados : ' ) ;
      writeln ( dados_d ) ;

      GoToXY ( 15, 3 ) ;
      write ( 'Scan em : ' ) ;
      if opcao2 = 'x'
      then
        writeln ( 'X' )
      else
        writeln ( 'Y' ) ;

      write ( dados_d, 'Scan em : ' ) ;
      writeln ( dados_d, opcao2 ) ;

      GoToXY ( 15, 5 ) ;
      write ( 'RASTER = ', raster:0:4, ' mm' ) ;

      write ( dados_d, 'RASTER = ' ) ;

```

```

write ( dados_d, raster:0:4, ' mm' );
writeln ( dados_d );

write ( dados_d, 'Data : ' );

GoToXY ( 50, 5 );
write ( 'Data : ' );
readln ( data );

writeln ( dados_d, data );
writeln ( dados_d );
write ( dados_d, 'Comentarios : ' );

GoToXY ( 15, 7 );
write ( 'Comentarios : ( texto com ' );
writeln ( '70 caracteres ' );
GoToXY ( 25, 8 );
readln ( coment );

writeln ( dados_d, coment );
writeln ( dados_d );

GoToXY ( 5, 12 );

{*****
  Limites do Grafico
  *****}
if opcao4 = 'n'
then
begin { * Grafico sem "zoom" * }
  writeln ( 'Limites do Grafico : ' );
  writeln :
  writeln ( ' X_min = 0 ' );
  writeln ( ' Y_min = ', den_min:0:1 );
  writeln ( ' X_max = ',
            ((( x - 40 ) * dx_pto)/dx):0:1);

  writeln ( ' Y_max = ', den_max:0:1 );
  writeln(dados_d, ' Limites do Grafico :');
  writeln(dados_d );
  writeln(dados_d, ' X_min = 0 ' );
  writeln(dados_d, ' Y_min = ',den_min:0:1);
  writeln(dados_d, ' X_max = ',(( x - 40 )/dx):0:1);
  writeln(dados_d, ' Y_max = ',den_max:0:1);
end { * fim do then * }
else
begin
  { * Grafico com "zoom" * }
  writeln ( 'Limites do Grafico :');
  writeln :
  writeln ( ' X_min = ',
            (((x_L_corr - 40 ) * dx_pto)/dx):0:1);

```

```

writeln ( '    Y_min = ', den1_min:0:1 );
writeln ( '    X_max = ',
          (((x_R_corr - 40)*dx_pto)/dx):0:1);
writeln ( '    Y_max = ', den1_max:0:1);

writeln(dados_d,' Limites do Grafico :');
writeln(dados_d);
writeln(dados_d,' X_min = ',
          (((x_L_corr - 40)*dx_pto)/dx):0:1);
writeln(dados_d,' Y_min = ', den1_min:0:1 );
writeln(dados_d,' X_max = ',
          (((x_R_corr - 40)*dx_pto)/dx):0:1);
writeln(dados_d,' Y_max = ', den1_max:0:1);
end ;

writeln (dados_d) ; writeln (dados_d);

{*****
  Coordenadas do filme que foi feito Scan
  *****}

GoToXY(45,12) ;
if ( opcao2 = 'x' )
then
begin
  writeln ('Coord. do filme:(mm)' );
  writeln ;
  GoToXY(45,14) ;
  writeln ( '    Y_filme = ',start:0:1 );
  GoToXY(45,15) ;
  writeln ( '    X_inicial = ', pos2:0:1 ) ;
  GoToXY(45,16) ;
  writeln ( '    X_final = ', pos_final:0:1 ) ;
  GoToXY(45,17) ;
  writeln ( '    Tam da Lin X = ', size:0:1 ) ;

  writeln (dados_d, ' Coord. do filme:(mm)' );
  writeln (dados_d) ;
  writeln (dados_d, '    Y_filme = ',start:0:1 ) ;
  writeln (dados_d, '    X_inicial = ', pos2:0:1 ) ;
  writeln (dados_d, '    X_final = ', pos_final:0:1 ) ;
  writeln (dados_d, '    Tam da Lin X = ', size:0:1 ) ;
end
else
begin
  writeln ( 'Coord. do filme:(mm)' );
  writeln ;
  GoToXY(45,14) ;
  writeln ( '    X = ', pos1:0:1 ) ;
  GoToXY(45,15) ;
  writeln ( '    Y_inicial = ', start:0:1 ) ;

```

```

GoToXY(45,16) ;
writeln ( '   Y_final = ', (start+size):0:1 ) ;
GoToXY(45,17) ;
writeln ( '   Tam da Col Y = ', size:0:1 ) ;

writeln (dados_d, ' Coord. do filme:(mm)' ) ;
writeln (dados_d) ;
writeln (dados_d, '   X = ', pos1:0:1 ) ;
writeln (dados_d, '   Y_inicial = ', start:0:1 ) ;
writeln (dados_d, '   Y_final = ', (start+size):0:1 ) ;
writeln (dados_d, '   Tam da Col Y = ', size:0:1 ) ;
end ;

writeln (dados_d) ;
for i := 1 to ptos_coletar do
  begin
    write (dados_d, Dados[i], '   ');
    end ;

close ( dados_d ) ;

end { * fim do then *}

else
begin
  GoToXY (12,2) ;
  writeln('Precisa-se antes fazer a');
  GoToXY (12,3) ;
  writeln('digitalizacao de uma linha ou');
  GoToXY (12,4) ;
  writeln('coluna...');
end ;

window ( 1, 1, 80, 25 ) ;

Continue ;

end; { * fim da aquisicao *}

end; { * fim CASE *}

Window(1,1,80,25);
ClrScr;
SETWINDOWP(2,4,78,24);
InitializeMenu ;
end { * fim else *}
end { * fim enter *}
end { * fim CASE *}
end; { * fim GetSelection *}

END.

```

```

Unit Scan_Unit ;

INTERFACE

USES CRT, DOS, GRAPH ;

Var

    n_passosX,          { * num. de passos a serem deslocados * }

    compl2, p : Integer ;

    texto : string ;

    raster : real ;     { * valor da resolucao (raster) * }

    passo : char ;     { * direcao do passo * }

    PROCEDURE Init55 ;
    PROCEDURE GetRaster ;
    PROCEDURE passoX ( var n_passosX : integer ; var passo : char ) ;

IMPLEMENTATION

{ *****
  PROCEDURE Init55
  *****

  Inicializa as 8255 com as respectivas palavras de controle

  *}

PROCEDURE Init55 ;
begin

{ * Inicializacao da 8255 ( IN e OUT Scanner )

  Enderecamento dos portos :

  (1) - 8255 --> J8 --> IN Scanner      (2) - 8255 -> J9 -> OUT Scanner
                OUT 8255                    IN 8255

  PortA1 := $300                PortA2 := $304
  PortB1 := $301                PortB2 := $305
  PortC1 := $302                PortC2 := $306
  $303 --> endereco de programacao    $307 --> endereco de programacao

  *}

```

```

Port[$303] := $80 ; { * palavra de controle p/ SAIDA da 8255
                    IN no Scanner
                    todas as portas ( A, B, C ) sao SAIDAS * }

Port[$307] := $BF ; { * palavra de controle p/ ENTRADA na 8255
                    OUT no Scanner
                    A porta A e ENTRADA para os dados do Scanner
                    A porta B e ENTRADA de sinais do Scanner e
                    a porta C e utilizada para sinais de
                    "handshaking" * }

Port[$300] := $FF ;      { * inicializo PortaA1 com FF * }

Port[$301] := $FF ;      { * inicializo PortaB1 com FF * }

Port[$302] := $FF ;      { * inicializo PortaC1 com FF * }

```

```
end ;
```

```
{ *****
  PROCEDURE GetRaster
  *****
```

```
Obtem o valor da resolucao (RASTER) em mm .
```

```
* }
```

```
PROCEDURE GetRaster ;
begin
  if (( Port[$305] and $03 ) = $03 )
  then
    begin
      raster := 100e-3 ; { * em mm * }
    end
  else
    begin
      if (( Port[$305] and $01 ) = $01 )
      then
        begin
          raster := 25e-3 ; { * em mm * }
        end
      else
        begin
          if (( Port[$305] and $02 ) = $02 )
          then
            begin
              raster := 50e-3 ; { * em mm * }
            end
          else
            begin

```

```

        raster := 0.0 ;
        GotoXY (25,20) ;
        write('ERRO') ;
    end ;
end;
end ;
end;

```

```

{*****
Procedure passoX
*****}

```

Rotina para obtencao do numero de passos (em X) a serem deslocados ate chegar na coluna desejada para fazer a varredura.

```

n_passosX : numero de passos ( 12.5 micron )
passo : direcao do passo: E --> ESQUERDA
        D --> DIREITA

```

```

*)

```

```

procedure passoX ( var n_passosX : integer ; var passo : char ) ;
begin

```

```

    Port[$300] := $EF ;    { * pulso em EX4, limpa comando processo * }
    Port[$300] := $FF ;

```

```

    { * Carrego numero de passos em complemento de 2 em BC15-BC00 * }

```

```

    compl2 := Not(n_passosX) + 1 ;    { * faz o complemento 2 do numero * }
        { * NOT - troca 0 -> 1 e 1 -> 0 * }
        { * depois soma 1 ao resultado * }

```

```

    Port[$301] := Hi (compl2) ;    { * coloca o byte mais significativo * }
        { * em PortB * }

```

```

    Port[$302] := Lo (compl2) ;    { * coloca o byte menos significativo * }
        { * em PortC * }

```

```

    Port[$300] := $BF ;    { * Pulso em EX6, carrega dado BC15-BC00 * }
    Port[$300] := $FF ;    { * no buffer do Scanner * }

```

```

    if ( passo = 'D' ) or ( passo = 'd' )

```

```

    then

```

```

        begin

```

```

            Port[$302] := $04 ; { * command word, passo a RIGHT * }

```

```

        end

```

```

    else

```

```

        begin

```

```

            Port[$302] := $08 ; { * command word, passo a LEFT * }

```

```

        end ;

```

```
Port[$300] := $DF ;    { * Pulso em EX7, executa comando do passo *}  
Port[$300] := $FF ;    { * linhas de dados BC03-BC00 *}
```

```
end ;
```

```
END . { * fim da Scan_Unit *}
```

```
DATA SEGMENT WORD PUBLIC
```

```
    EXTRN Dados : BYTE , ptos_coletar : WORD
```

```
DATA ENDS
```

```
CODE SEGMENT BYTE PUBLIC
```

```
    ASSUME CS : CODE , DS : NOTHING , ES : DATA
```

```
    PUBLIC Ledata
```

```
PORTA1 equ 0300h
PORTB1 equ 0301h
PORTC1 equ 0302h
CTR1 equ 0303h
```

```
PORTA2 equ 0304h
PORTB2 equ 0305h
PORTC2 equ 0306h
CTR2 equ 0307h
```

```
Ledata PROC NEAR
```

```
    PUSH BP
    PUSH DS
```

```
    MOV AL,00Eh ;command word (Read Density )
    MOV DX,PORTC1
    OUT DX,AL
```

```
    MOV AL,0DFh ;PULSO EM EX7 AL <-- DFh
    MOV DX,PORTA1 ;EXECUTA LEITURA DX <-- 300h ,porto A
    OUT DX,AL ;DOS DADOS saida de dados pelo porto A
```

```
    MOV AL,0FFh
    MOV DX,PORTA1
    OUT DX,AL
```

```
    CLI ;seta flag de Interrupcao p/ 0
```

```
    CLD ;seta flag de Direcao (DF) p/ 0, desta
        ;forma, autoincrementa o ponteiro usado
        ;em operacoes de string
```

```
    MOV AX,Seg DATA
    MOV ES,AX
    MOV CX,ptos_coletar ;carrega contador
    MOV DI,Offset Dados
```

```
Lp1: MOV DX,PORTC2
```

```
Lp2: IN AL,DX ;AL <-- conteudo do buffer indicado
```

```
                                ;por DX
AND  AL,020h                    ;verifico se IBF (8255) esta ALTO
JZ   Lp2
MOV  DX,PORTA2                  ;move dado (PORTA2) para DX
IN   AL,DX
STOSB                           ;carrega o valor do acumulador (dado)
                                ;para memoria enderecada por DI
Loop Lp1                         ;decrementa ptos_coletar e verifica se
                                ;ptos_coletar e 0 (zero)

STI                               ;seta flag de interrupcao p/ 1

STD                               ;seta flag de direcao p/ 1

MOV  AL,0EFh                    ;pulso em EX4 (limpa qualquer comando)
MOV  DX,PORTA1                  ;em processo
OUT  DX,AL

MOV  AL,0FFh
MOV  DX,PORTA1
OUT  DX,AL

POP  DS
POP  BP

RET

Ledata ENDP

CODE  ENDS
      END
```

```
UNIT CursorUnit ;
```

```
INTERFACE
```

```
USES CRT, DOS, GRAPH ;
```

```
TYPE
```

```
pontos_Y = array [1..450] of integer ;
```

```
CONST
```

```
{* Caracteres ASCII para o null, enter e bell *
```

```
nullChar = #0 ;
```

```
enter = #13 ;
```

```
bell = #7 ;
```

```
{* Caracteres ASCII para a teclas das setas *
```

```
upArrow = #72 ;
```

```
leftArrow = #75 ;
```

```
rightArrow = #77 ;
```

```
downArrow = #80 ;
```

```
Var
```

```
inSpace, seta : char ;      {* opcoes dos cursores e setas *
```

```
x,                          {* pto X para plotar *
```

```
x_L_corr, x_R_corr,        {* coordenadas do cursor direito *
```

```
y_L_corr, y_R_corr,        {* coordenadas do cursor esquerdo *
```

```
x_medio : Integer ;        {* posicao media entre os cursores *
```

```
dx : REAL ;                 {* deslocamento dos ptos no eixo X *
```

```
Y : pontos_Y ;             {* vetor que contem os Y para serem  
plotados *
```

```
PROCEDURE le_tecla1 ( seta : char ) ;
```

```
PROCEDURE le_tecla2 ( seta : char ) ;
```

```
IMPLEMENTATION
```

```
{*****
```

```
Procedure le_tecla1
```

Procedimento relacionado com o cursor ESQUERDO do grafico.
Verifica se a tecla pressionada e valida e movimenta o cursor esquerdo do grafico, verificando seus limites.

*)

```

procedure le_tecla1 ( seta : char );
begin
  Case seta of
    RightArrow : begin
      x_L_corr := Round ( x_L_corr + dx );
      if x_L_corr < x_medio
      then
        begin
          SetColor ( black );
          line ( Round (x_L_corr - dx), Y[y_L_corr] + 10,
                Round (x_L_corr - dx), Y[y_L_corr] - 10 );
          SetColor ( white );
          PutPixel ( Round (x_L_corr - dx),Y[y_L_corr],15);
          y_L_corr := y_L_corr + 1 ;
          line ( x_L_corr, Y[y_L_corr] + 10,
                x_L_corr, Y[y_L_corr] - 10 );
        end
      else
        begin
          write ( bell );
          x_L_corr := Round ( x_L_corr - dx );
        end
      end ;

    LeftArrow : begin
      x_L_corr := Round ( x_L_corr - dx );
      if x_L_corr < 40
      then
        begin
          write ( bell );
          x_L_corr := Round ( x_L_corr + dx );
        end
      else
        begin
          SetColor ( black );
          line ( Round (x_L_corr + dx), Y[y_L_corr] + 10,
                Round (x_L_corr + dx), Y[y_L_corr] - 10 );
          SetColor ( white );
          PutPixel ( Round (x_L_corr + dx),Y[y_L_corr],15);
          y_L_corr := y_L_corr - 1 ;
          line ( x_L_corr, Y[y_L_corr] + 10,
                x_L_corr, Y[y_L_corr] - 10 );
        end;
      end ;
    end ;
  end ;

```

```
end ; { * fim CASE * }
end ;
```

```
{ *****
  Procedure le_tecla2
  *****
```

Procedimento relacionado com o cursor DIREITO do grafico.
Verifica se a tecla pressionada e valida e movimenta o cursor direito do grafico, verificando seus limites.

```
*}
```

```
procedure le_tecla2 ( seta : char ) ;
begin
```

```
  Case seta of
```

```
    RightArrow : begin
```

```
      x_R_corr := Round ( x_R_corr + dx ) ;
```

```
      if x_R_corr > x
```

```
      then
```

```
        begin
```

```
          write ( bell ) ;
```

```
          x_R_corr := Round ( x_R_corr - dx ) ;
```

```
        end
```

```
      else
```

```
        begin
```

```
          SetColor ( black ) ;
```

```
          line ( Round ( x_R_corr - dx ), Y[y_R_corr] + 10,
```

```
                Round ( x_R_corr - dx ), Y[y_R_corr] - 10 ) ;
```

```
          SetColor ( white ) ;
```

```
          PutPixel ( Round ( x_R_corr - dx ), Y[y_R_corr], 15 ) ;
```

```
          y_R_corr := y_R_corr + 1 ;
```

```
          line ( x_R_corr, Y[y_R_corr] + 10,
```

```
                x_R_corr, Y[y_R_corr] - 10 ) ;
```

```
        end;
```

```
      end ;
```

```
    LeftArrow : begin
```

```
      x_R_corr := Round ( x_R_corr - dx ) ;
```

```
      if x_R_corr < x_medio
```

```
      then
```

```
        begin
```

```
          write ( bell ) ;
```

```
          x_R_corr := Round ( x_R_corr + dx ) ;
```

```
        end
```

```
      else
```

```
        begin
```

```
          SetColor ( black ) ;
```

```
          line ( Round ( x_R_corr + dx ), Y[y_R_corr] + 10,
```

```
                Round ( x_R_corr + dx ), Y[y_R_corr] - 10 ) ;
```

```
          SetColor ( white ) ;
```

```
PutPixel ( Round (x_R_corr + dx),Y[y_R_corr],15);
y_R_corr := y_R_corr - 1 ;
line ( x_R_corr, Y[y_R_corr] + 10,
      x_R_corr, Y[y_R_corr] - 10 ) ;
end;
end ;

end ; { * fim CASE *}
end ;

END .
```

Apêndice D - Instruções para utilização do Software desenvolvido

- Ligar o Photoscan através da chave POWER ON localizada no compartimento interno.

- Coloque o filme a ser digitalizado no tambor do Photoscan.

- Com o tambor parado (sem rotacionar), através da chave REV, posicione o subsistema óptico na posição 0, em X, observando a escala contida no Photoscan, parando o movimento com a chave STOP (ver figura A.2).

- Posicione a chave RASTER na posição de resolução desejada.

- Através da chave RUN, coloque o tambor para rotacionar. Aguarde alguns segundos até que o tambor atinja a velocidade de rotação correta (lâmpadas DRUM SPEED superior e ON LINE acendem).

- Posicione a chave 2D/3D (localizada no compartimento interno do Photoscan) na posição desejada.

- Execute o programa.

- Será exibida uma "tela" com os limites do filme que pode ser digitalizado e pede-se ao usuário para posicionar o subsistema óptico em $X=0$ e selecionar a resolução desejada (figura 5.1).

- Será exibido na tela um menu com as seguintes opções :

X-scan - faz a digitalização de uma linha do filme.

Y-scan - faz a digitalização de uma coluna do filme.

Aquisicao - armazena os dados (densidades) da linha ou coluna digitalizada em um arquivo.

Sair - fim do programa.

A figura 5.2 ilustra o menu descrito anteriormente.

- Selecione uma das opções do menu, pressionando a tecla da seta para cima ou a tecla da seta para baixo, até que o destaque se mova para a posição desejada, ou pressione, ainda, a primeira letra do nome da opção : X, Y, A, ou S e complete a seleção pressionando a tecla ENTER.

- De acordo com a opção escolhida, deve-se seguir os seguintes passos :

X-scan - o usuário entra com a posição da linha (start) (em mm) que será digitalizada, e a posição inicial da tomada dos dados (coluna) (em mm) (figura 5.3).

Em seguida ele digitará a posição final da tomada dos dados (pos_final) (em mm) (ver figuras 5.4 e 4.3)

Aguarde alguns instantes até que todos os pontos (densidades) sejam coletados.

Observação : se algum erro ocorrer e, o subsistema óptico tocar a parada final de um dos lados, deve-se imediatamente desligar o Photoscan através da chave POWER ON.

Em seguida, o usuário tem a opção de digitalizar uma outra linha do filme, lembrando que se ele quiser armazenar os dados digitalizados anteriormente em um arquivo, deverá escolher a opção N, pois desta forma será traçado o gráfico das densidades pela distância X entre cada ponto, e ele optará por fazer ou não uma ampliação ("zoom") de regiões do gráfico obtido (figura 5.5).

Para ampliar a curva, o usuário irá movimentar dois cursores que se encontram nos extremos do gráfico, sendo este movimento realizado com a seta para direita ou seta para esquerda, até eliminar os pontos do gráfico que não interessam (figura 5.6).

Um novo gráfico será traçado e os novos limites do gráfico, bem como as posições do filme onde foi feita a digitalização serão mostradas na tela (figura 5.7).

O menu será novamente exibido na tela para que o usuário possa fazer a escolha de uma nova opção.

Y-scan - o usuário entra com a posição da coluna (coluna) (em mm) a ser digitalizada (figura 5.8); depois, digita a posição inicial (start) (em mm) da tomada dos dados e o tamanho da coluna (size) (em mm) a ser digitalizada (ver figuras 5.9 e 4.4).

Em seguida, o usuário seguirá os mesmos passos descritos para o caso da digitalização

de uma linha do filme (figuras 5.10, 5.11 e 5.12).

Aquisicao - nesta opção, o usuário armazenará em um arquivo (tipo texto), cujo nome será fornecido pelo usuário e, lembrando que se houver alguma coisa neste arquivo será perdida, os dados do filme digitalizado, bem como alguns comentários e informações sobre o filme utilizado, como por exemplo : data, se foi digitalizado uma linha ou coluna, os limites do gráfico obtido na digitalização, as coordenadas do filme onde foi feita a digitalização etc (figura 5.13).

Sair - esta opção serve para terminar o programa.

Se o usuário quiser rodar novamente o programa, ele deverá parar a rotação do tambor do Photoscan, através da chave STOP e colocar o subsistema óptico na posição 0 (zero), em X, através da chave REV.

Em seguida, através da chave RUN, ele deve colocar o tambor para rotacionar e aguardar alguns segundos até que o tambor atinja a velocidade de rotação correta.

Depois disto, o usuário deve então, executar novamente o programa.