

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ELETRÔNICA E DE COMPUTAÇÃO

INTERFACE DE UM SISTEMA DE RECONHECIMENTO DE OBJETOS
UTILIZANDO SENSOR DE TATO

Autor:

Lilian Kawakami Carvalho

Orientadores:

Eliana Prado Lopes Aude, Ph.D.

Henrique Serdeira, Eng.

Co-Orientador:

Jose Paulo Brafman, M.Sc.

Examinador:

Aloysio de Castro P. Pedrosa, Dr.Ing.

Examinador:

Ernesto Prado Lopes, Ph.D.

Examinador:

Mario Ferreira Martins, M.Sc.

DEL

Setembro de 2003

Agradecimentos

Agradeço aos meus pais, Itajaci e Helena, pelo apoio e incentivo que me deram para a minha formação. Aos meus irmãos, Laura e Rubens, pelo carinho e por sempre me fazerem rir nas horas de tristeza e alegria. Aos meus avós, tios e primas que sempre me deram amor, alegria e "abrigo" nas férias.

Aos meus amigos, mesmo que alguns distantes, que estiveram sempre dispostos a me ajudar. Em especial ao Fabio e Wanderley pela colaboração neste trabalho.

A todos do laboratório CONTROLAB, Eliana, Julio, Flavio, Henrique e Mario, que permitiram e contribuíram para o desenvolvimento deste projeto.

A todos os professores e colegas do DEL por ajudarem durante a minha caminhada na faculdade.

E por fim, ao CAPES, CNPq, FUJB, FAPERJ que contribuíram financeiramente para realização deste trabalho.

Resumo

Este trabalho tem como objetivo o estudo e o projeto de uma interface capaz de reconhecer objetos utilizando sensor de tato conectado a um microprocessador do tipo IBM/PC.

O projeto consiste em montar e estudar uma mão mecânica com um grau de liberdade capaz de reconhecer um conjunto de objetos regulares. O sistema é composto basicamente por três subsistemas:

- O sensor de tato;
- A interface entre o sensor e o computador;
- O subsistema capaz de reconhecer diferentes figuras pré-determinadas a partir dos dados adquiridos pela interface.

Inicialmente, procurou-se um sensor que tenha sensibilidade e granularidade suficiente para reconhecer os objetos desejados e apresente linearidade em relação aos diferentes pesos.

Com a escolha do sensor, o segundo subsistema foi dividido em duas etapas. A primeira etapa consistiu em projetar e montar um circuito eletrônico para converter a pressão no sensor em tensão, e para a confecção deste circuito foi utilizado a ferramenta Layout Plus do Orcad. A segunda etapa consistiu em projetar e montar um sistema capaz de converter os sinais analógicos de tensão em sinais digitais para serem lidos e processados pelo PC.

Finalmente, o último subsistema consiste em implementar o subsistema de reconhecimento de diferentes figuras. Este programa foi implementado utilizando a linguagem C.

Palavras-Chave

Interface, Orcad, “C”, Aquisição de Dados, Reconhecimento de objetos, Tato.

Sumário

Agradecimentos	ii
Resumo	iii
Palavras-Chave	iv
Sumário	v
1 Introdução	4
1.1 Motivação	4
1.2 Objetivo	5
1.3 Documento	6
2 Sensor de Tato	7
2.1 Sensor	7
2.1.1 Especificações do Sensor	7
2.1.2 Escolha do Sensor	7
2.2 Característica do Sensor	9
2.2.1 Conector do Sensor	9
2.2.2 Teste do Conector	14
2.2.3 Curva Característica do Sensor	14
3 Circuito de Acionamento das Células	17
3.1 Esquema Geral	17
3.2 Projeto	18
3.2.1 Amplificador	18
3.2.2 Decodificador	19

3.2.3	Multiplex Analógico	20
3.3	Simulação	20
3.3.1	Amplificador	20
3.3.2	Contador	20
3.4	Teste de Bancada	23
3.4.1	Amplificador	24
3.4.2	Contador	25
3.5	Confecção das Placas	26
3.5.1	Depuração das placas	26
4	Programa de Interface com o PC	28
4.1	Porta Paralela	28
4.1.1	O Conector DB25	29
4.1.2	Ligando e Desligando Aparelhos Externos	30
4.1.3	Escolha do Registrador	33
4.2	Conversor A/D	34
4.2.1	Funções do Conversor A/D	34
5	Reconhecimento de Objetos	36
5.1	Escolha dos Objetos a Serem Reconhecidos	36
5.2	Classificação das Figuras	36
5.3	Implementação do Reconhecimento das Figuras	37
5.4	Resultados Experimentais	38
6	Conclusões	44
6.1	Proposta de Trabalhos Futuros	45
A	Esquema das Placas	47
A.1	Esquemático	47
A.2	Layout Amplificador	48
A.3	Layout Contador	48
B	Programas	52
B.1	Programa para a Porta Paralela	52
B.2	Programa Final	54

Lista de Figuras

1.1	Robôs cooperando para realizar uma tarefa.	5
1.2	Robôs usados em telemedicina.	6
2.1	Sensor da Tekscan 5051.	9
2.2	Esquema de funcionamento do sensor da Tekscan 5051.	10
2.3	Medidas do contato do Sensor da Tekscan 5051.	10
2.4	Matriz de contatos J1 do Sensor da Tekscan 5051.	11
2.5	Matriz de contatos J2 Sensor da Tekscan 5051.	11
2.6	Esquema dos conectores.	12
2.7	Esquema do conector J3 com J1.	12
2.8	Esquema conector J4 e J5 com J1.	13
2.9	Conector do Sensor da Tekscan 5051.	13
2.10	Gráfico da resistência x peso.	15
2.11	Gráfico da pressão x peso.	16
3.1	Esquema geral do circuito.	17
3.2	Esquema detalhado do circuito para 16 células.	18
3.3	Esquema do amplificador.	19
3.4	Esquema do amplificador.	21
3.5	Simulação do amplificador para diferente Rs.	21
3.6	Esquema do contador.	22
3.7	Simulação do contador.	22
3.8	Esquema do contador em cascata.	23
3.9	Simulação do contador em cascata.	23
3.10	Esquema do contador com amplificador.	24
3.11	Simulação do contador com amplificador.	24

3.12	Esquema do filtro passa-baixa.	25
3.13	Esquema do filtro passa-baixa.	25
4.1	Significado de cada pino do DB25.	30
4.2	Esboço de funcionamento do DB25.	30
4.3	Circuito 1.	31
4.4	Circuito 2.	32
4.5	Medição do período do clock.	34
5.1	Até o sexto nível de força ativado.	39
5.2	Reconhecimento no círculo.	40
5.3	Níveis de força no círculo.	40
5.4	Reconhecimento no triângulo.	41
5.5	Níveis de força no triângulo.	41
5.6	Reconhecimento no quadrado.	42
5.7	Níveis de força no quadrado.	42
5.8	Figura desconhecida.	43
5.9	Esfera.	43
A.1	Esquemático da placa dos contadores.	47
A.2	Esquemático da placa dos amplificadores.	48
A.3	Placa dos amplificadores - BOTTON.	49
A.4	Placa dos amplificadores - TOP.	50
A.5	Placa do contador - BOTTON.	51
A.6	Placa do contador - TOP.	51

Lista de Tabelas

2.1	Dimensões do Sensor	9
2.2	Relação entre peso e resistência	15
2.3	Relação entre peso, resistência e pressão	16
4.1	Configuração da porta paralela	29
4.2	Configuração dos registradores da porta LPT1	29
4.3	Registrador de Controle	32
5.1	Número de células ativadas por figura	37
5.2	Relação entre nível de força,tensão e cor	38

Capítulo 1

Introdução

A robótica surgiu como resultado de intensa pesquisa na área de computadores e está num estágio precoce de desenvolvimento, abrangendo as áreas de controle, computação e inteligência artificial.

A grande quantidade de recursos destinados ao desenvolvimento e pesquisa em robótica já começou a dar frutos e certamente conduzirá a grandes acontecimentos no futuro. Várias ferramentas têm sido desenvolvidas e fontes de energia têm sido exploradas para substituir o homem e ajudá-lo em seu trabalho. Atualmente o homem ainda é uma parte importante do sistema por ser responsável pela tomada de decisão. O principal objetivo da robótica é liberar o ser humano de tarefas difíceis e cansativas. Para atingir esta meta muita pesquisa deve ser realizada na área de Inteligência Artificial para que o robô possa por si só tomar decisões e identificar os objetos ao seu redor.

A próxima etapa dentro da robótica depende do aparecimento de computadores que possam funcionar como cérebros. O cérebro do robô do futuro deverá ser um computador sofisticado, rápido, com grande espaço de memória e capacidade de analisar situações complicadas. É necessário também que seja pequeno e tenha baixo consumo de potência.

1.1 Motivação

Nos dias de hoje, o principal objetivo das indústrias é o desenvolvimento de plantas das linhas de produção que não requeiram a presença de pessoas. Toda a linha de produção seria controlada por computadores que estariam conectados às máquinas, robôs e sensores. O computador central executaria o planejamento de processos, fazendo a identificação e

correção de falhas.

Para obter uma interação cada vez maior entre os robôs é importante o uso do tato para o trabalho cooperativo, ou seja, para que dois robôs trabalhem em conjunto como visto na figura 1.1. Isto vem a permitir uma automatização completa da fábrica.

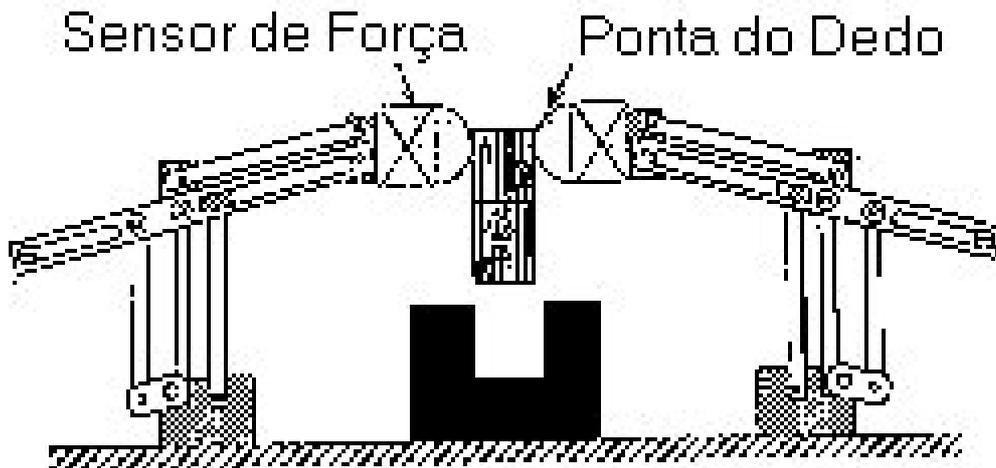


Figura 1.1: Robôs cooperando para realizar uma tarefa.

O tato em robótica é usado para obter informações do contato entre a mão mecânica e o objeto de trabalho. A informação pelo toque é usada, por exemplo, para localizar objetos e reconhecê-los, informando suas características como a geometria da superfície, textura e material, ou ainda controlar a força do manipulador para pegá-los.

Outro exemplo é o uso na telemedicina, possibilitando que um médico, mesmo à distância, consiga operar um paciente através de um robô como pode ser visto na figura 1.2.

1.2 Objetivo

O tato consiste, basicamente, em identificar o contato de algum objeto com a nossa pele. Isto é realizado por milhares de minúsculos sensores que possuímos espalhados pelo corpo. Em um robô este sentido é implementado da mesma forma, utilizando-se sensores de tato em uma mão mecânica colocando na ponta de cada dedo um desses dispositivos.

O aspecto abordado do tato neste trabalho é o reconhecimento da forma dos objetos.



Figura 1.2: Robôs usados em telemedicina.

1.3 Documento

Este trabalho consiste em construir um sistema inteligente para identificar objetos e está dividido em três partes. A primeira parte consiste na busca do sensor adequado e será tratada no capítulo 2.

A segunda parte será discutida em dois capítulos. O capítulo 3, onde está descrito o projeto e a confecção da placa responsável pelo interfaceamento com o PC e o capítulo 4 descreve o uso da porta paralela na geração do *clock* que comanda a aquisição dos dados. Este capítulo também descreve o conversor A/D e as principais funções utilizadas pelo sistema de aquisição de dados.

A terceira parte está contida no capítulo 5 e consiste na escolha do critério de classificação dos objetos. Por fim, as conclusões sobre este trabalho são apresentadas no capítulo 6.

Capítulo 2

Sensor de Tato

Como dito no capítulo 1, este capítulo descreve o sensor de tato propriamente dito. Inicialmente a especificação do sensor é descrita e em seguida o estudo de suas características é apresentado.

2.1 Sensor

O primeiro passo para o desenvolvimento do tato é dispor de um sensor com uma boa resolução.

2.1.1 Especificações do Sensor

Para simular o tato humano procurou-se um sensor organizado em forma matricial, pois montar artesanalmente vários sensores em cascata não obteria a resolução desejada. As características procuradas para o sensor são:

- forma matricial
- faixa de trabalho: sólidos pequenos de no máximo 50mm e com o peso inferior a 1kg

Esta escolha deve-se aos possíveis objetos a serem usados para o reconhecimento, pois objetos maiores e mais pesados seriam de difícil manipulação e confecção.

2.1.2 Escolha do Sensor

Buscou-se no mercado através da Internet sensores de tato que tivessem as características descritas na seção 2.1.1. Os sensores encontrados eram, normalmente, sensores de

pressão e para obter o tamanho desejado seria necessário uma cascata destes sensores. Esta decisão não foi adotada pois não teria uma boa resolução para medir a variação de pressão do objeto na matriz de sensores, tendo vários espaços em branco entre um sensor e outro.

Esta decisão foi tomada baseada em testes realizados com um protótipo anterior que foi montado com uma espécie de borracha condutiva que varia sua resistência com a força aplicada. Este tipo de material é usado na confecção dos contatos de teclado, joystick e telefones. Após a montagem de um circuito para testes, aplicaram-se pesos pré-estabelecidos sobre esta borracha condutiva medindo-se a tensão de saída, que era função de sua resistência. Foi observado que o sensor não respondia linearmente, ou seja à relação entre força e tensão não era linear. Decidiu-se, então, optar por um sensor que tivesse uma resposta do tipo ligado - desligado, isto é, se o sensor está ou não acionado.

Com o objetivo de aumentar a sensibilidade do sensor projetou-se uma estrutura do tipo prensa para aumentar a pressão sobre o objeto a ser testado.

Após inúmeros testes concluiu-se que com apenas nove células uma boa quantidade de sólidos eram classificados. O sistema final resumia-se em duas placas paralelas, cada uma com uma matriz 3x3 de sensores.

Com o funcionamento do protótipo, determinaram-se quais objetos poderiam ser reconhecidos pelo sensor. Foram escolhidos sólidos regulares onde a diferença básica entre eles era o número de sensores ativados. A confecção dos sólidos foi difícil, porque esses deveriam ter superfícies lisas e não deformáveis. A solução encontrada foi a fabricação das peças com resina e depois o polimento destas peças.

A grande vantagem deste protótipo é o seu baixo custo. Com o objetivo de classificar objetos semelhantes, não classificado pelo sistema descrito, decidiu-se optar pela compra de um sensor industrial.

Na procura de um sensor adequado aos nossos requisitos de projetos, encontrou-se um fabricante, Tekscan, que fornece uma grande variedade de sensores de tato. Os sensores são semelhantes variando o número de células e a faixas de pressão. Escolheu-se o sensor modelo 5051 (figura 2.1 com as seguintes especificações: 81 mm x 251 mm, 1936 sensores (62 sensores/cm²) e pressão máxima de 4921kgf/m², com as dimensões descritas na tabela 2.1

A matriz do sensor é formada por linhas de material condutivo na horizontal e na vertical como mostra a figura 2.2. Cada célula é definida pelo cruzamento de uma linha com uma coluna da matriz do sensor. O sensor apresenta 44 linhas e 44 colunas, tendo, no total,

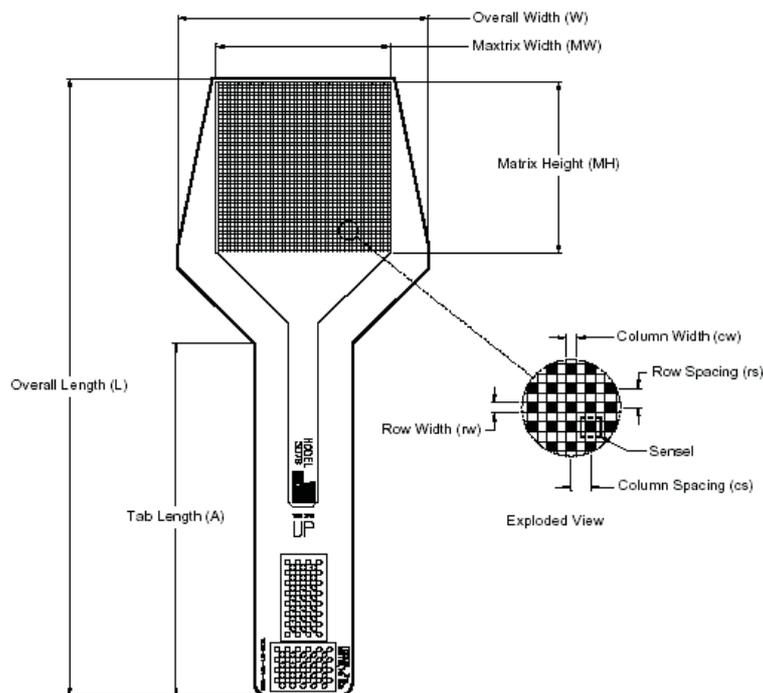


Figura 2.1: Sensor da Tekscan 5051.

Overall Length	Overall Width	Tab Length	Matrix Width	Matrix Height	Sensel Density
L	W	A	MW	MH	
(mm)	(mm)	(mm)	(mm)	(mm)	(sensel/ cm^2)
251	81	168	56	56	62.0

Tabela 2.1: Dimensões do Sensor

1936 células. O sensor funciona de forma que uma determinada força aplicada sobre o sensor fornece uma resistência. Variando-se a força a resistência também varia.

2.2 Característica do Sensor

2.2.1 Conector do Sensor

Para ler os dados do sensor foi necessário fazer um conector especial que se encaixasse a ele, pois a soldagem direta de fios no contato do sensor os deformaria.

Foi confeccionado uma placa de circuito impresso contendo duas matrizes de contatos

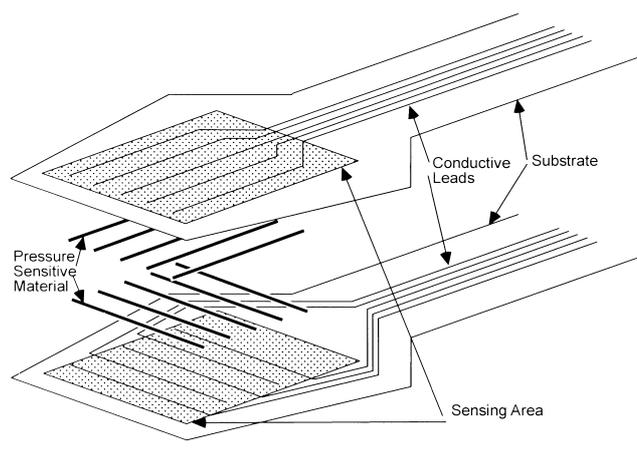


Figura 2.2: Esquema de funcionamento do sensor da Tekscan 5051.

similares às matrizes de contatos que compõem as saídas das células do sensor. A transmissão do sinal de saída das células do sensor para o circuito impresso foi obtida prensando um contato contra o outro.

Para a confecção da matriz de contato com a precisão desejada foi necessário medir o tamanho do contato e o espaçamento entre eles. Para medir o contato foi usado um paquímetro. A figura 2.3 mostra as medidas realizadas.

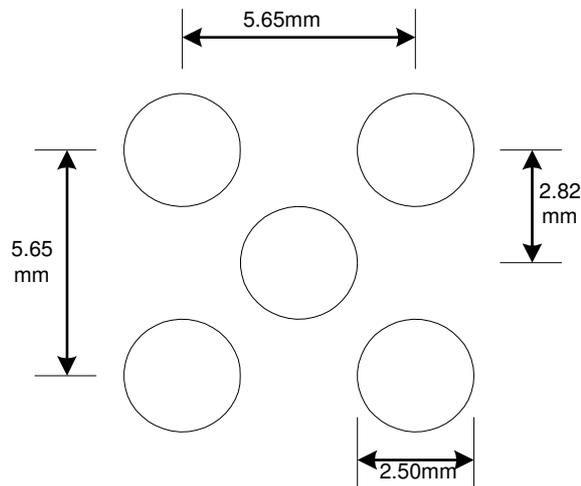


Figura 2.3: Medidas do contato do Sensor da Tekscan 5051.

Como se pode observar na figura 2.1 o sensor tem duas matrizes de contatos. Uma, que denominou-se J1, em que os contatos estão na parte superior do sensor, estando ligada às linhas horizontais e outra, que denominou-se J2, em que os contatos estão na parte inferior, estando ligada às linhas verticais.

As dimensões do J1 são 30mm x 23mm e ele apresenta 44 contatos. As dimensões do J2 são 20mm x 40.8mm e ele apresenta 53 contatos. A distância entre J1 e J2 é de 8mm.

Para desenhar os contatos do sensor foi usado o Library Manager do Orcad. O Orcad é um programa de computador utilizado para análise e confecção de circuitos eletrônicos. O programa é composto do Orcad Capture, capaz de criar diagramas esquemáticos, Orcad PSpice AD, capaz de auxiliar o usuário no projeto, simulação de circuitos eletrônicos e interpretar os resultados obtidos e o Orcad Layout Plus, onde se pode fazer o desenho de circuito impresso. Entretanto, como o programa trabalha com medidas em mils, foi necessário transformar as medidas para desenhar a matriz de contatos. A figura 2.4 representa J1 que tem 44 contatos e a figura 2.5 representa J2 com 53 contatos.

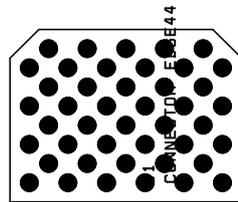


Figura 2.4: Matriz de contatos J1 do Sensor da Tekscan 5051.

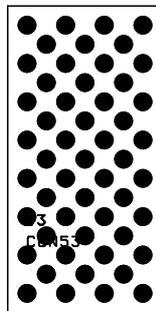


Figura 2.5: Matriz de contatos J2 Sensor da Tekscan 5051.

Após chegar a uma versão final do layout da matriz de contatos devia-se ligar os fios de saída da matriz ao amplificador para se converter a resistência lida em tensão. Para ter uma maior flexibilidade optou-se em colocar conectores do tipo 3M das matrizes de contatos do sensor para a placa do amplificador como observado na figura 2.6.

Foi escolhido o conector 3M pelo grande número de fios e a facilidade de encontrá-lo no mercado. Para isto, montou-se uma placa que foi roteada no Orcad seguindo o layout da matriz de contatos do sensor. Na visualização do roteamento da placa as duas matrizes, J1

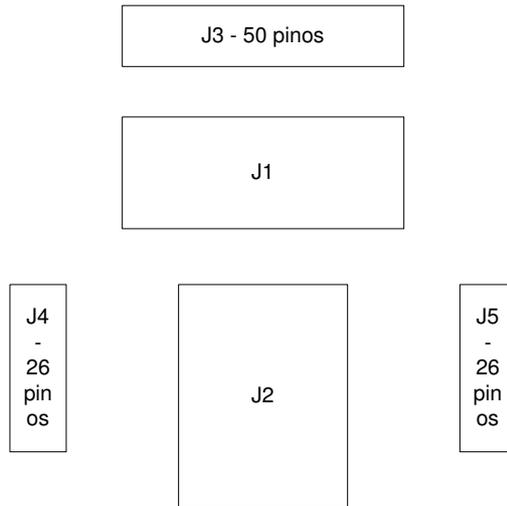


Figura 2.6: Esquema dos conectores.

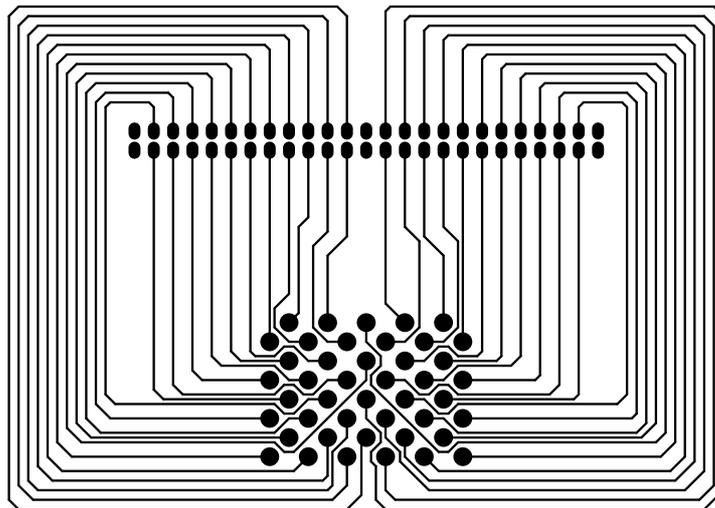


Figura 2.7: Esquema do conector J3 com J1.

e J2, tiveram os seus contatos divididos verticalmente e conectados no sentido anti-horário aos conectores 3M sendo eles, J3, J4 e J5, como pode ser visto na figura 2.7 e na figura 2.8.

Devido ao layout do sensor, as linhas horizontais estão ligadas aos contatos superiores de forma intercalada. Logo, dividiu-se o conector J3 em dois, pois metade do sinal vinha pelas linhas ímpares do sensor e a outra metade pelas linhas pares.

Os conectores J4 e J5 estão ligados aos contatos inferiores do sensor. Como as linhas verticais do sensor estão ligadas sequencialmente a J2, não foi necessário ligar os conectores J4 e J5 de forma intercalada.

As matrizes de contatos J1 e J2 estão em faces opostas do sensor. Para obter melhor

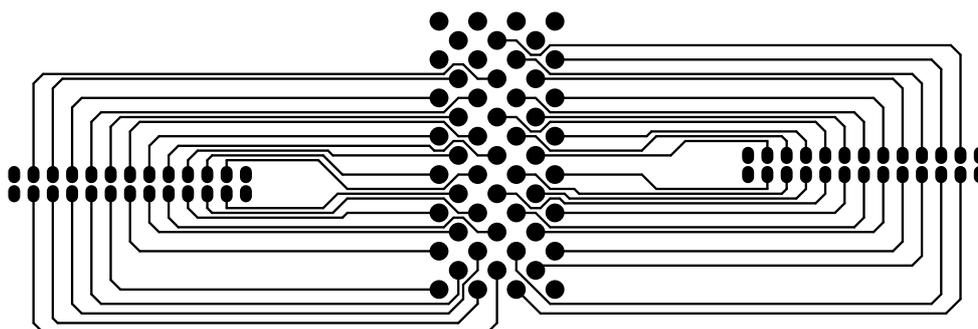


Figura 2.8: Esquema conector J4 e J5 com J1.

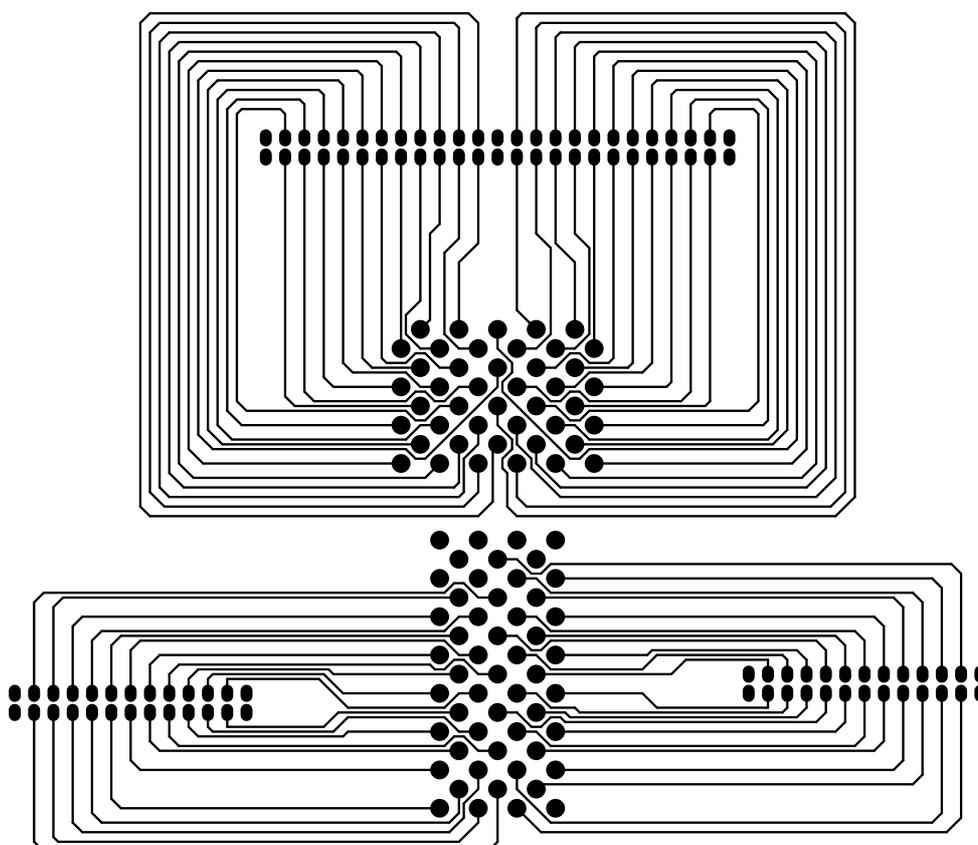


Figura 2.9: Conector do Sensor da Tekscan 5051.

encaixe e firmeza, decidiu-se duplicar a placa e ligar cada placa a um lado do sensor deixando um espaçamento entre J1 e J2 de 8mm. O contato foi feito por pressão tomando-se o cuidado para haver um encaixe perfeito. A figura 2.9 mostra o layout final.

2.2.2 Teste do Conector

Para verificar o funcionamento do conector foram realizados diversos testes. O primeiro teste foi verificar os conectores separadamente. Colocou-se a placa do conector nos contatos de J2 que esta na parte inferior do sensor e o multímetro no contato de J1 que esta localizado na parte superior do sensor e pressionou-se o sensor para medir a resistência. Esta variou de $10M\Omega$ a $7M\Omega$. Como estes valores estavam muito altos, pressionou-se os contatos e obteve-se a seguinte faixa: $11.5k\Omega$ a $7.4k\Omega$. Assim, pôde-se observar um problema de contato. Desta forma colocou-se uma esponja entre o sensor e a placa de cima e pressionou-se. Os valores de resistência obtidos com esta correção foram em torno de $14k\Omega$, concluindo-se que o resultado com a esponja estava satisfatório.

Após perceber o problema do contato, colocou-se esponja entre as 2 placas e a matriz de contatos do sensor(J1 e J2) para pressioná-los. Fazendo uma nova medida obteve-se uma resistência de $10k\Omega$ quando pressionado o sensor.

Para garantir firmeza e evitar o deslocamento das placas foram feitos furos laterais nas placas.

2.2.3 Curva Característica do Sensor

Para entender o funcionamento do sensor levantou-se sua curva característica. Para este estudo foi colocado um cubo de $20 \times 20 \text{mm}$ no canto superior direito do sensor e variou-se o peso colocado em cima do cubo, lendo-se apenas uma célula do sensor. Os resultados obtidos estão apresentados na tabela 2.2. Os dados da tabela 2.2 podem ser observados no gráfico da figura 2.10.

De acordo com as especificações do sensor, ele suporta uma pressão de até $7\text{psi}(0.49\text{kgf}/\text{cm}^2)$. Considerando que a superfície do cubo está distribuída de forma uniforme com uma área de 4cm^2 , pode-se calcular a pressão sobre o sensor. Os valores calculados estão na tabela 2.3. Plotando o gráfico para a tabela 2.3 obteve-se a figura 2.11.

Analisando o gráfico e o limite de pressão considerou-se a faixa de 0.05 até 0.25 de kgf/cm^2 , considerando que os objetos a serem trabalhados têm uma área máxima de 9cm^2 e com o peso na faixa de 200g a 1000g.

	Peso g	Resistência Ω
1	50	170.4k
2	100	80.6k
3	150	49.3k
4	200	42.4k
5	300	31.4k
6	500	36.2k
7	600	34.2k
8	700	30.9k
9	1000	16.8k

Tabela 2.2: Relação entre peso e resistência

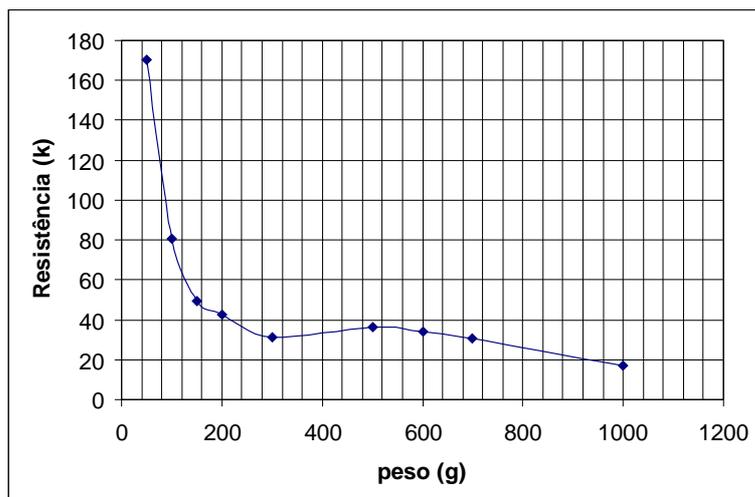


Figura 2.10: Gráfico da resistência x peso.

	Peso g	Resistência Ω	Pressão kgf/cm ²
1	100	80.6k	0.025
2	200	42.4k	0.050
3	300	31.4k	0.075
4	500	36.2k	0.125
5	600	34.2k	0.150
6	700	30.9k	0.175
7	1000	16.8k	0.250

Tabela 2.3: Relação entre peso, resistência e pressão

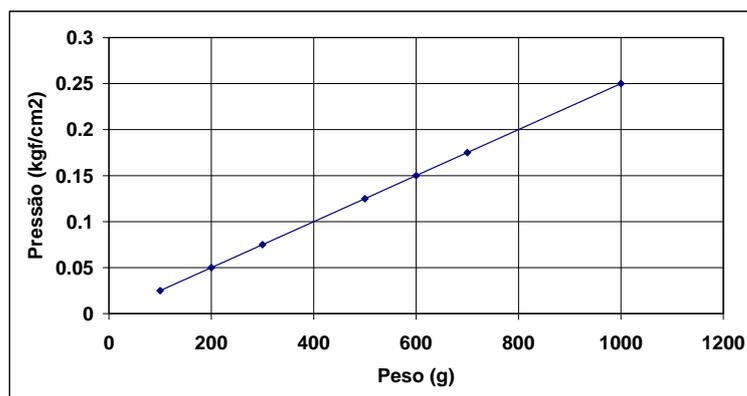


Figura 2.11: Gráfico da pressão x peso.

Capítulo 3

Circuito de Acionamento das Células

Após a familiarização com o sensor, abordar-se-á, neste capítulo, o projeto, a simulação e a confecção dos circuitos para interface com o sensor.

3.1 Esquema Geral

O esquema proposto do circuito de interface do sensor com o conversor A/D está representado na figura 3.1. O sensor fornece resistência para o circuito. Para medir esta resistência irá se fornecer corrente para o circuito e se medirá a tensão gerada no componente, obtendo-se sua resistência pela lei de Ohm. A tensão lida já é uma medida indireta da própria resistência, logo é mais simples ler diretamente a tensão. Para injetar corrente no circuito se utilizará um amplificador operacional inversor.

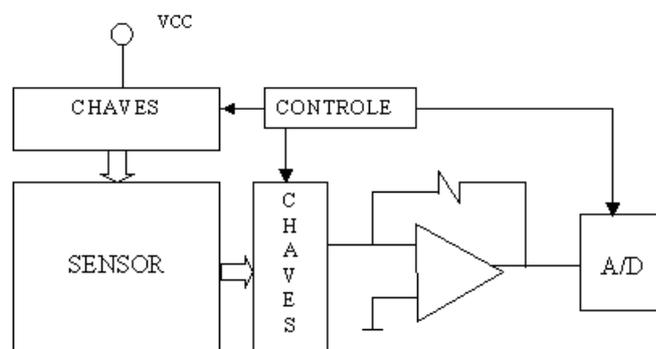


Figura 3.1: Esquema geral do circuito.

Observando o sensor como um conjunto de 1936 sensores não seria apropriado lê-los ao mesmo tempo, pois requeria um número alto de conversores. Além disso os conversores

estariam interferindo no funcionamento do circuito. Desta forma opta-se por fazer uma varredura no sensor. Esta varredura será feita alimentando as colunas, uma de cada vez, com uma tensão de referência e lendo as n linhas. A figura 3.2 mostra o esquema detalhado para uma matriz com 16 sensores. Nela pode-se observar o funcionamento da alimentação seqüencial das 4 colunas. Quando uma coluna está alimentada é feita a leitura de 4 linhas. Também pode-se observar que sendo o conversor A/D de apenas um canal, utiliza-se um multiplex para a leitura das 4 linhas.

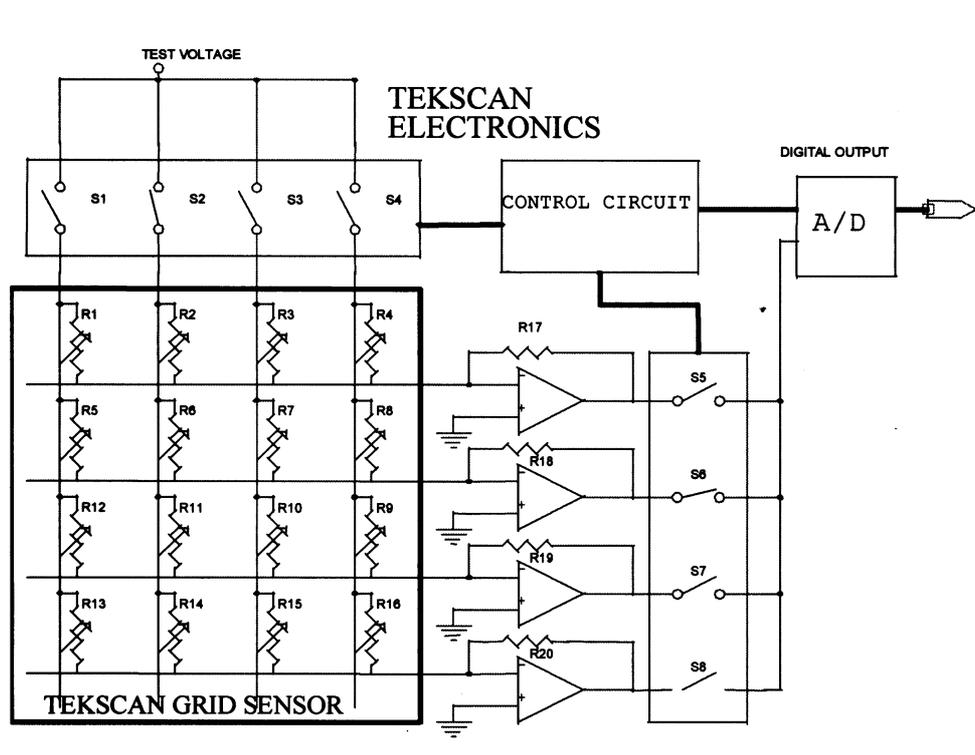


Figura 3.2: Esquema detalhado do circuito para 16 células.

3.2 Projeto

3.2.1 Amplificador

Para conseguir transformar a resistência gerada no sensor em tensão se utilizará um amplificador operacional. Analisando cada célula individualmente como um resistor variável pode-se chegar na representação do esquema na figura 3.3.

O ganho do circuito variará conforme a variação de resistência do sensor. O ganho do

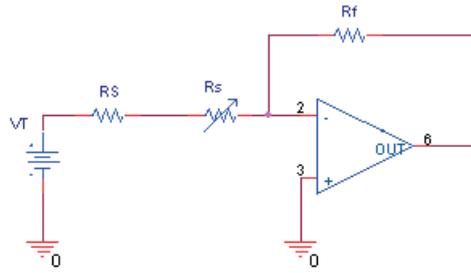


Figura 3.3: Esquema do amplificador.

amplificador inversor é dado pela fórmula 3.1 como descrito na referência [1].

$$V_o = VT \frac{R_f}{R_s} \quad (3.1)$$

A tensão de referência foi estipulada em 5V e a tensão máxima de saída em 10V, já que não se deseja que a tensão de saída ultrapasse 15V em virtude da faixa de trabalho dos amplificadores disponíveis no mercado. Com estes valores chega-se a uma relação de 2 entre R_f e R_s como mostrado na equação 3.2.

$$\frac{R_f}{R_s} = \frac{V_o}{VT} = \frac{10}{5} = 2 \quad (3.2)$$

Analisando os valores obtidos na tabela 2.2, com o intuito de se obter uma boa faixa de valores de resistência, pode-se definir uma faixa de variação entre (10k Ω ;100k Ω). Desta forma o maior ganho será obtido com a menor resistência em R_s . Como a menor resistência é de 10k Ω chega-se a um valor de R_f de 20k Ω . O valor de R_f usado, entretanto, foi de 22k Ω já que este é o valor comercial mais próximo do desejado.

Para os primeiros testes utilizou-se o amplificador operacional modelo 741 e alimentou-se o circuito com uma fonte simétrica de 12V. Esta escolha se deve ao fato do sistema ser lento, ou seja, o objeto a ser reconhecido permanecerá em contato com o sensor por quanto tempo se desejar, não necessitando de uma rápida varredura do sensor. Para este caso o amplificador operacional 741 atenderia às exigências.

3.2.2 Decodificador

Observando a figura 3.1, as chaves que fornecem a alimentação para as colunas do circuito devem chavear de forma periódica. Por este motivo foi escolhido um contador em anel que fornece periodicamente 5V de tensão nos pinos de saída.

Procurou-se um contador em anel CMOS, pois sua saída é mais estável. Foram encontrados alguns componentes atendendo esta especificação. Foi escolhido o que tinha mais canais, pois tentou-se minimizar o número de integrados usados, já que o sensor tem 44 linhas. O contador em anel escolhido foi CD4017B que gera 10 ciclos de clock. O seu datasheet se encontra no apêndice C.

3.2.3 Multiplex Analógico

Uma coluna do sensor é acionada por vez e através do multiplex analógico é selecionada uma das 44 linhas para ser lida pelo conversor A/D. Para isto procurou-se um multiplex analógico com o máximo número de canais. O escolhido foi o multiplex analógico modelo MPC506A com 16 canais. Para chavear todas as saídas são necessários 2 níveis de multiplex(4 multiplexadores). O manual do multiplex MPC506A está no apêndice C.

Como o número de entradas é elevado decidiu-se usar só uma parte do sensor para montar um protótipo e verificar seu funcionamento. Escolheu-se usar uma matriz de 256 células(16 linhas x 16 colunas), pois se conseguiria usar apenas uma camada de multiplex e 16 amplificadores, simplificando o circuito.

3.3 Simulação

3.3.1 Amplificador

Para simular o sensor foi usado o resistor variável com parâmetros que alteram o seu valor de $10\text{k}\Omega$ a $100\text{k}\Omega$ com o passo de $20\text{k}\Omega$. A figura 3.4 mostra o esquema montado para a simulação.

Os resultados obtidos pela simulação estão na figura 3.5. Pode-se observar, pelos valores de ganho, que para valores de resistência abaixo de $30\text{k}\Omega$ as curvas estão bem próximas. Isto torna menor a resolução para os pesos maiores que 700g. Assim, a faixa de valores dos pesos usados será menor que 700g.

3.3.2 Contador

Para simular o contador montou-se o esquema mostrado na figura 3.6.

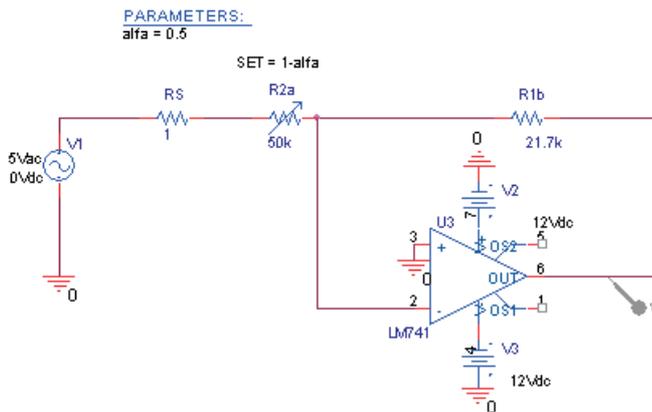


Figura 3.4: Esquema do amplificador.

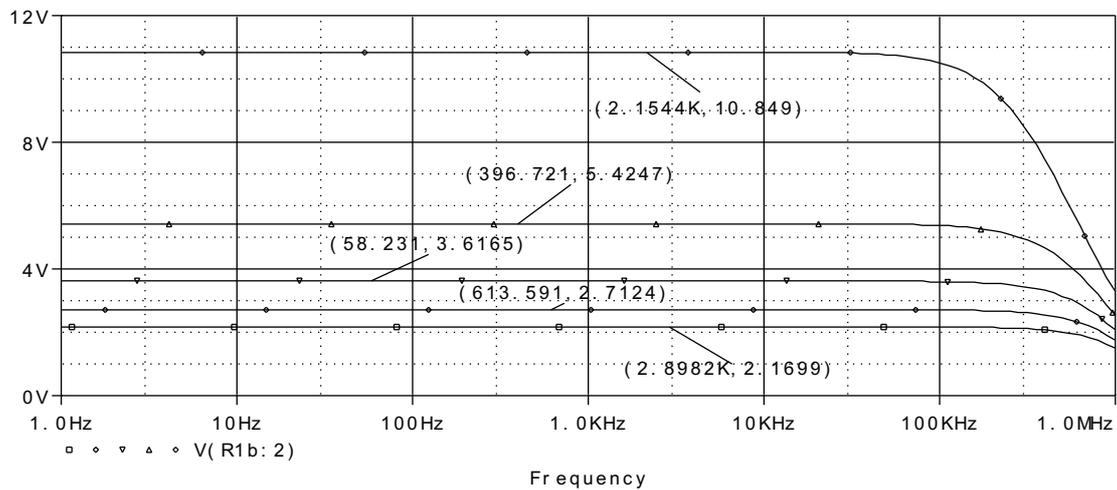


Figura 3.5: Simulação do amplificador para diferente Rs.

Os dados obtidos por simulação conferem com os valores descritos no datasheet do contador CD4017B que está no apêndice C. A figura 3.7 mostra a simulação para o contador contando até 10.

Para o contador contar até 16 precisa-se de dois integrados e uma interação entre eles. O datasheet do contador mostra uma possível configuração em cascata dos componentes. Esta configuração foi testada mas não funcionou, pois o sinal de Q0 do primeiro estágio não estava reiniciando o segundo. Para resolver isto modificou-se o circuito com um inter-travamento com clock inhibit e usando o mesmo clock para os dois integrados. Estas modificações foram implementadas e obteve-se o resultado desejado. A implementação destas modificações foi feita só com o uso de lógica NOR, pois se deseja que a lógica use somente

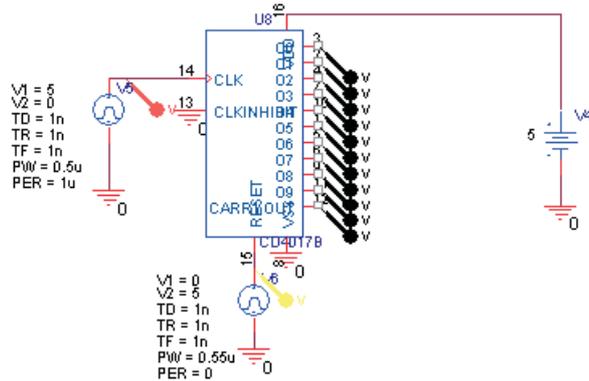


Figura 3.6: Esquema do contador.

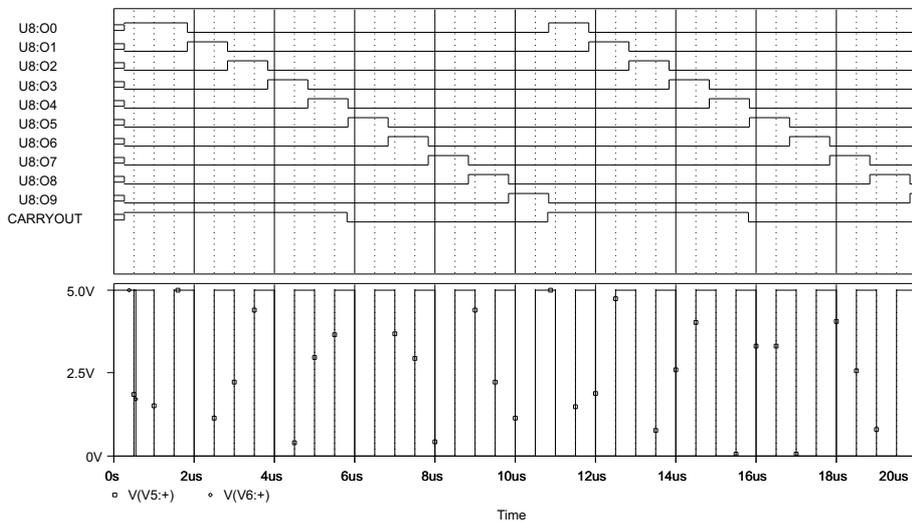


Figura 3.7: Simulação do contador.

um chip para facilitar na confecção da placa e baratear o projeto. A figura 3.8 mostra o esquema do contador em cascata com 16 ciclos de clock.

A figura 3.9 mostra a simulação para o contador contando até 16.

Após estes testes foi colocado na primeira saída do contador o amplificador operacional LM741 para verificar a integração dos dois módulos. Para simulação estipulou-se o valor de R_s em $10k\Omega$, que resulta num ganho de aproximadamente 2 no amplificador. A figura 3.10 mostra o esquema do contador em cascata com o amplificador.

A figura 3.11 mostra a simulação para o contador com o amplificador na primeira saída tendo ganho de aproximadamente 2. Isto é, a tensão de entrada no amplificador é de 5V e na saída 10V, acompanhando a periodicidade do contador.

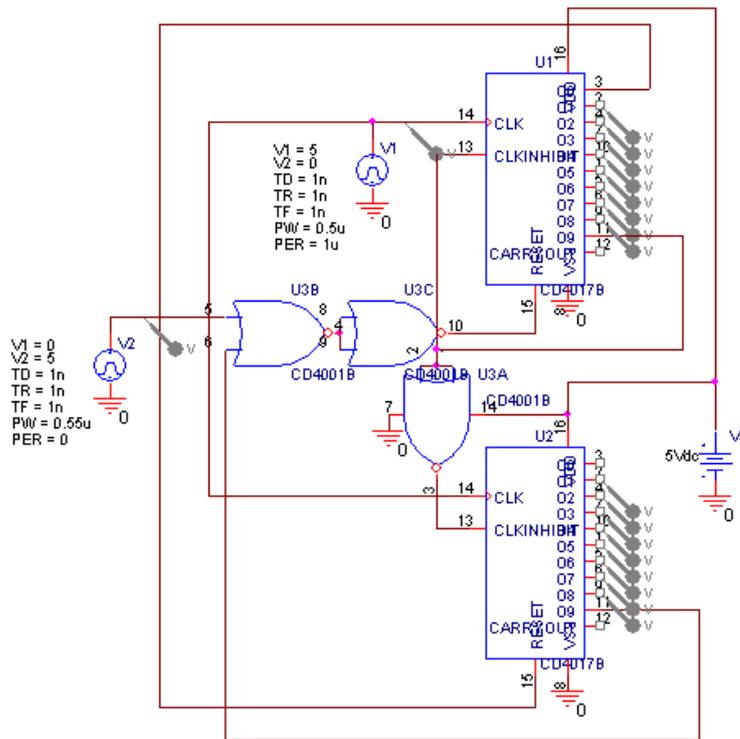


Figura 3.8: Esquema do contador em cascata.

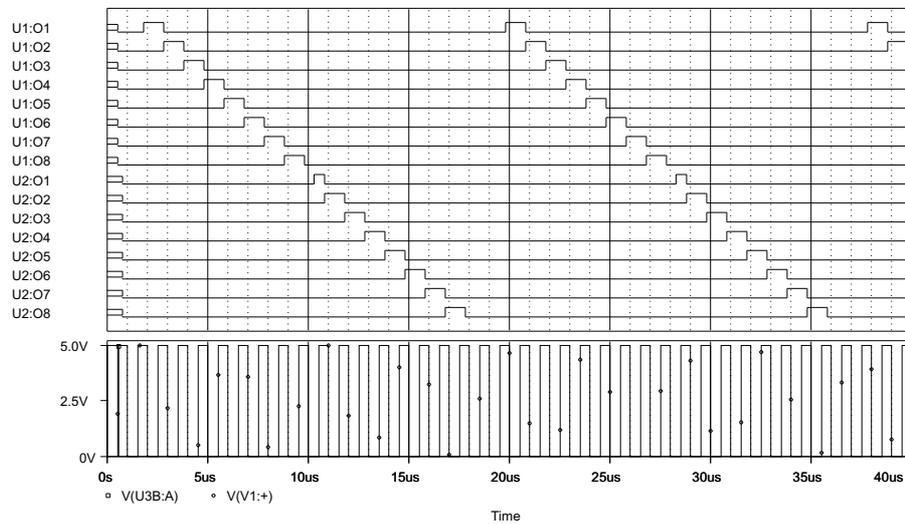


Figura 3.9: Simulação do contador em cascata.

3.4 Teste de Bancada

Depois de simular o sistema, partiu-se para a montagem dos circuitos. Para um primeiro teste foi montado os circuitos em *proto-board*.

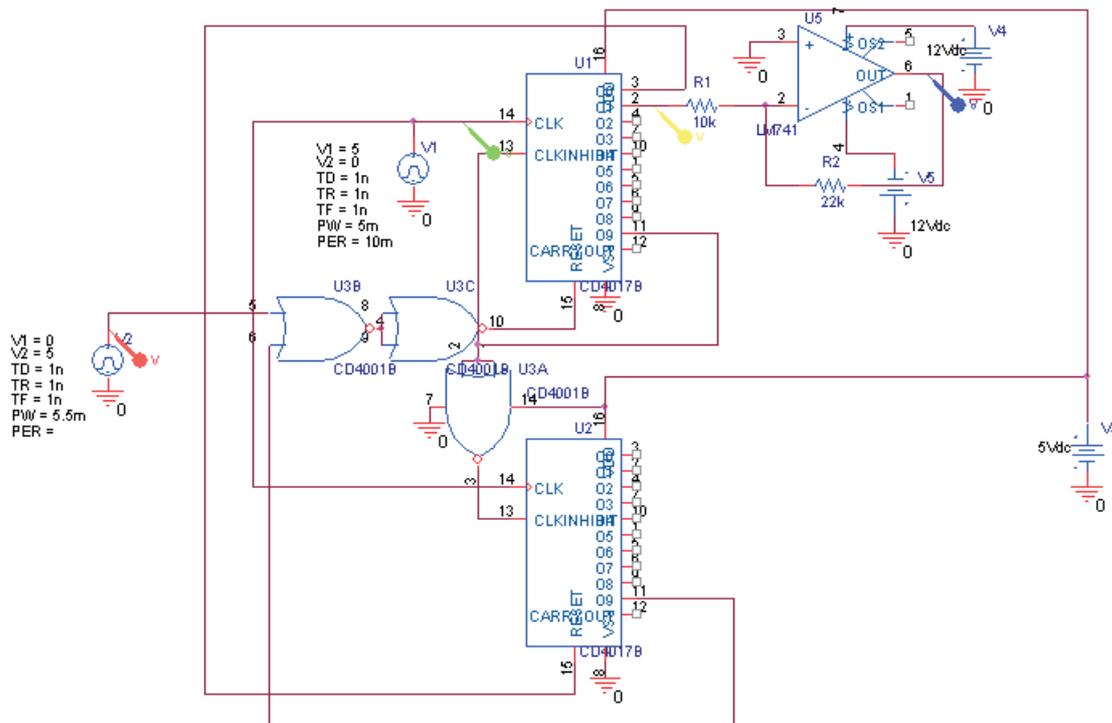


Figura 3.10: Esquema do contador com amplificador.

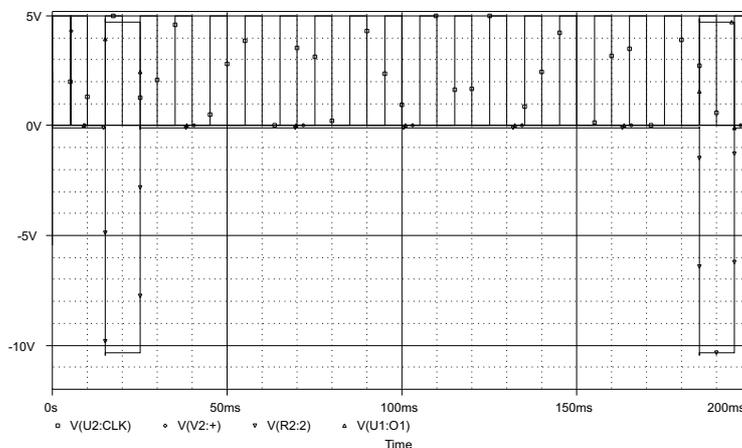


Figura 3.11: Simulação do contador com amplificador.

3.4.1 Amplificador

No teste em bancada foi usado um cubo de área fixa variando-se pesos em cima deste de 100g a 1kg. Com uma célula do sensor ligada a um amplificador mediu-se a voltagem na saída do amplificador e observou-se que os valores variavam de acordo com os valores simulados. Ao analisar com mais cuidado o sinal medido, verificou-se que este apresentava

um ruído de 100mV com uma frequência média de 83kHz. Para tentar eliminar este ruído de alta frequência colocou-se um filtro passa-baixa. A figura 3.12 mostra o esquema básico de um filtro passa-baixa como descrito na referência [1]. Esta configuração foi adotada para se economizar componentes e tempo no roteamento da placa.

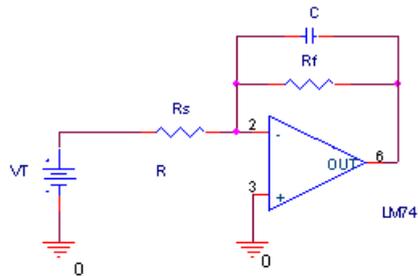


Figura 3.12: Esquema do filtro passa-baixa.

A figura 3.13 mostra o comportamento do filtro passa-baixa.

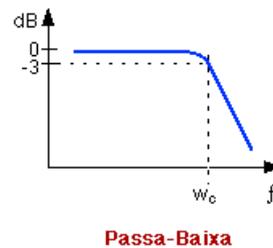


Figura 3.13: Esquema do filtro passa-baixa.

A frequência de corte do filtro é o ponto de queda de 3dB do ganho do amplificador e é dada pela equação 3.3.

$$C \cdot Rf = \frac{1}{w_c} \quad (3.3)$$

Considerando $f_c=8\text{kHz}$, uma década abaixo da frequência do ruído, o capacitor calculado foi de 0.9nF e o valor comercial mais próximo é de 1nF que nos dá $f_c=7.2\text{kHz}$.

3.4.2 Contador

Como o contador apresenta muitas saídas(16)e seu comportamento já tinha sido simulado, escolheu-se confeccionar a placa antes de testar em *proto-board*.

3.5 Confeção das Placas

O esquemático das placas foi feito no OrCad Layout Plus. O footprint dos componentes foi modificado para que suas ilhas ficassem no formato oval, pois havia uma maior área para soldar.

Como a confecção das placas seria feita de forma artesanal foram escolhidas somente duas camadas: a BOTTON como *routing* e a TOP como *jumper*. O tamanho das trilhas foi de 20mils.

O roteamento das placas foi difícil porque o comando *autorouting* do OrCad não conseguiu rotear todas as trilhas. Este roteamento foi difícil porque só havia uma face de cobre. Logo, teve-se que otimizar o posicionamento dos componentes e escolher o caminho das trilhas nas regiões mais críticas. Por fim teve-se que ligar as trilhas não roteadas manualmente. Os layouts das placas estão no apêndice A.

Para confeccionar a placa usou-se o papel *easy-peel*. Esse papel transfere o circuito impresso projetado no computador para a placa de cobre via impressora a laser. A transferência do circuito para a placa de cobre usando-se o papel *easy-peel* não é perfeita, podendo haver falhas nas trilhas. Portanto, torna-se necessário fazer uma inspeção visual e consertar as falhas.

Após a corrosão das placas foi testada a condutividade das trilhas. Estas também podiam apresentar falhas dependendo do tempo em que as placas permaneceram em contato com o percloro. As falhas foram corrigidas usando solda.

3.5.1 Depuração das placas

Depois de furar as placas e soldar os componentes, foi testado primeiro a alimentação do circuito sem os integrados. A alimentação do contador é de 5V e a alimentação da placa dos amplificadores é de 12V simétricos. Depois de verificada a correta alimentação os integrados foram inseridos e soldados na placa, verificando-se, novamente, o seu correto funcionamento.

Após o teste de alimentação do circuito testou-se o funcionamento de cada placa. Para testar a placa do contador usou-se um conjunto de leds, já que o contador apresenta 16 saídas e não seria possível monitorar mais do que 2 saídas com o osciloscópio. Com o conjunto de leds pôde-se verificar a variação de estados da saída e sua periodicidade. A

dificuldade encontrada para verificar o funcionamento do contador foi no *reset* do contador, pois ao mudar o estado de ativado para desativado ou vice e versa o pino de *reset* fica em aberto e o sistema entra em um estado indeterminado. Para resolver este problema o *reset* sempre tem que estar alimentado, seja com 5V ou 0V.

O próximo passo foi verificar o circuito dos amplificadores. Colocou-se uma tensão de referência de 5V e uma resistência fixa, e foi medido se a saída de cada amplificador apresentava o ganho esperado.

Com o bom funcionamento das duas placas, integrou-se os circuitos(contador-sensor-amplificador) e foram feitos testes. Aplicaram-se pesos conhecidos no sensor e suas resistências foram medidas. Verificou-se na tabela 2.2 se a resistência medida correspondia ao peso usado para depois verificar se o ganho era igual ao esperado. Caso a resistência não correspondesse ao peso, tornava-se necessário um ajuste da posição do peso no sensor. O resultado das medidas foi satisfatório.

Capítulo 4

Programa de Interface com o PC

Neste capítulo descreve-se como interfacear o circuito com o computador para fazer a leitura dos dados, isto é, como digitalizar as tensões vindas do circuito e sincronizar a aquisição dos dados com o contador. Para converter os sinais analógicos em digitais foi usado o conversor A/D e para sincronizar a aquisição dos dados do circuito com o PC foi usada a porta paralela.

4.1 Porta Paralela

Para resolver o problema de sincronismo entre a aquisição dos dados e o contador trabalhou-se com a porta paralela.

A porta paralela é uma interface de comunicação entre o computador e um periférico. Quando a IBM criou seu primeiro PC (Personal Computer) ou Computador Pessoal, a idéia era conectar a essa porta uma impressora, mas atualmente, são vários os periféricos que utilizam-se desta porta para enviar e receber dados para o computador (exemplos: Scanners, Câmeras de vídeo, Unidade de disco removível e outros).

A porta paralela será utilizada, de uma maneira não convencional, isto é, não será utilizada com uma impressora, mas para gerar um clock e um reset para o contador.

O computador geralmente possui 3 portas: LPT1, LPT2 e LPT3 (alguns possuem LPT4). Cada porta dessas possui 3 endereços: data, status e control. Esses endereços estão numa ordem seqüencial. Isto quer dizer que se a porta data tem o endereço 0x0378, então o endereço correspondente de status é 0x0379 e o de control é 0x037a.

Hoje em dia, os computadores estão basicamente configurados da maneira descrita na

tabela 4.1

Printer	Data Port	Status	Control
LPT1	0x0378	0x0379	0x037a
LPT2	0x0278	0x0279	0x027a
LPT3	0x03bc	0x03bd	0x03be

Tabela 4.1: Configuração da porta paralela

Em alguns computadores (principalmente os da IBM) esses valores podem mudar. Uma maneira de saber quais são os endereços configurados na máquina é verificar o setup da BIOS.

A porta usada no projeto será a LPT1 e seus registradores estão definidos como na tabela 4.2

Nome	Endereços LPT1	Descrição
Registro de Dados	0x0378	Envia um byte para a impressora
Registro de Status	0x0379	Ler o Status da impressora
Registro de Controle	0x037a	Envia dados de controle para a impressora

Tabela 4.2: Configuração dos registradores da porta LPT1

4.1.1 O Conector DB25

O DB25 é um conector que fica na parte de trás do gabinete do computador. É através deste que o cabo paralelo se conecta ao computador para poder enviar e receber dados. A figura 4.1 mostra o conector padrão DB25, com 25 pinos, onde cada pino tem um nome que o identifica.

Todas as saídas da porta paralela são lógicas, isto é, escrevendo nível lógico 1 em um bit da porta o pino correspondente terá 5V na saída. Entretanto, os pinos SELECT IN, AUTOFEED e STROBE possuem uma lógica invertida. Isto quer dizer que colocando nível lógico 1 nesses pinos corresponde a uma saída de 0v. Isso acontece porque quando não há nenhuma ligação com a porta, o computador "vê" esses pinos como nível lógico 1, já que ele "pensa" que existe uma impressora conectada. Para resolver esse problema esses pinos foram invertidos. A figura 4.2 representa o funcionamento geral do DB25.

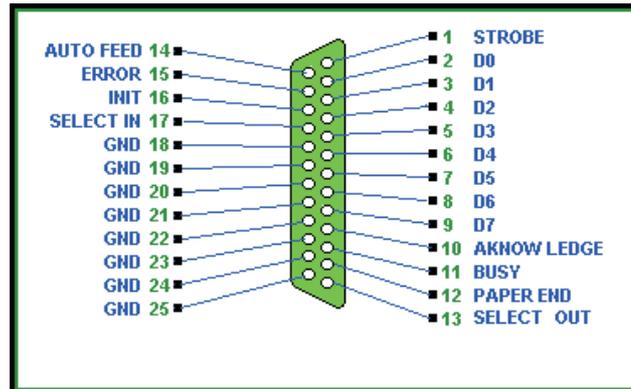


Figura 4.1: Significado de cada pino do DB25.

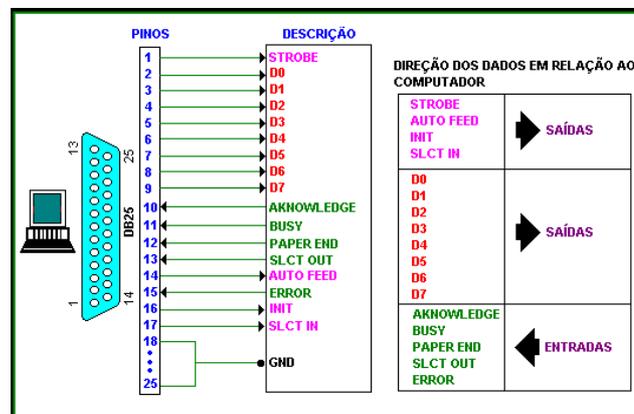


Figura 4.2: Esboço de funcionamento do DB25.

4.1.2 Ligando e Desligando Aparelhos Externos

O primeiro teste feito com a porta foi de ligar e desligar oito LEDs, conectados através de uma cabo à Porta Paralela como descrito pela figura 4.3 que chamou-se de circuito 1.

No esboço da figura 4.3, observa-se que os terminais do cátodo (K) dos LEDs estão ligados aos terminais dos resistores, que estão ligados através do cabo aos pinos do DB25.

O programa descrito abaixo, envia à Porta Paralela oito bytes, um a cada vez até que o usuário pressione uma tecla.

```
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#define MASCARA 0X01
```

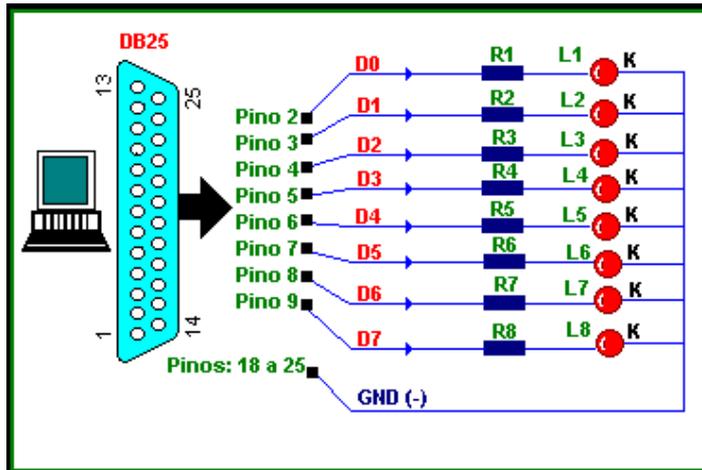


Figura 4.3: Circuito 1.

```
#define LPT1 0x378

main(void) {
    unsigned char Valor=1;
    while( Valor < 8 )
    {
        printf("LED %d aceso",Valor);
        outportb(LPT1, Valor); // Envia para a Porta LPT1
        printf("\nPressione uma tecla para ascender o próximo LED...");
        getch( );
        Valor = Valor*2; //o bit 1 é movido para a esquerda
    }
}
```

Cada bit do byte enviado à Porta Paralela está relacionado com um pino do DB5. Ao enviar o byte 00000001 o LED que acende é o que está relacionado ao primeiro estado(D0), o byte 00000010 acende o LED do segundo estado(D1) e assim sucessivamente.

O segundo teste foi verificar o funcionamento do registrador de controle, pois ele será útil na implementação do *clock* e *reset* do contador.

A figura 4.4 mostra o teste feito com o registrador de controle da LPT1 que tem o endereço 37Ah. Ao enviar um byte para este registrador, manteve-se os 4 últimos bits (nibble alto) zerados, para não interferir no registrador 378h.

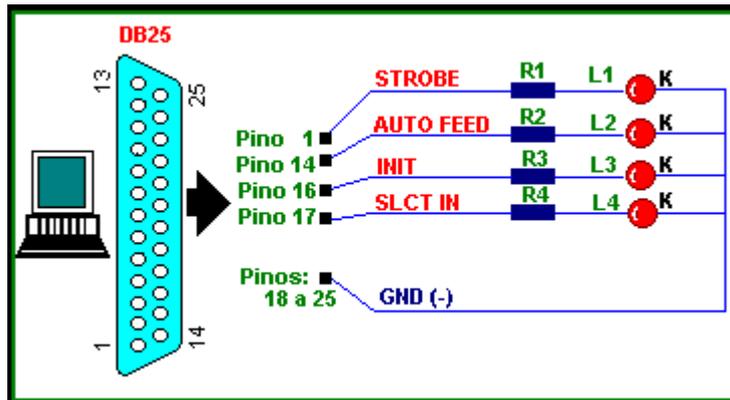


Figura 4.4: Circuito 2.

A lógica do registrador 37Ah é um pouco diferente. Para ativar os sinais Slet In, AutoFeed e Strob, será necessário colocar os bits em 0 (zero) porque eles trabalham de forma invertida: 0 (zero) ativa, 1 (um) desativa. Somente o Init trabalha na sua forma normal. A pinagem e a lógica do registrador de controle pode ser visualizada na tabela 4.3.

	SletIn	Init	AutoFeed	Strobe
Pino	17(low)	16(high)	14(low)	1(low)
Bit	3	2	1	0

Tabela 4.3: Registrador de Controle

```
//Envia para a Porta Paralela LPT1 no registro 37Ah:
```

```
#include <stdio.h>
```

```
#include <dos.h>
```

```
#define BIT0LIGALEDD1 10 //0000 1010
```

```
#define BIT1LIGALEDD2 9 //0000 1001
```

```
#define BIT2LIGALEDD3 15 //0000 1111
```

```
#define BIT3LIGALEDD4 3 //0000 0011
```

```
#define LIGATODOS 4 //0000 0100
```

```
#define DESLIGATODOS 11 //0000 1011
```

```

main(void) {
    outportb(0x37A,LIGATODOS); //Liga todos os LEDs
    getch();
    outportb(0x37A,BIT0LIGALED1); //Liga LED 1
    getch();
    outportb(0x37A,BIT1LIGALED2);
    getch();
    outportb(0x37A,BIT2LIGALED3);
    getch();
    outportb(0x37A,BIT3LIGALED4);
    getch();
    outportb(0x37A,DESLIGATODOS); //Desliga todos os LEDs
}

```

4.1.3 Escolha do Registrador

Escolheu-se para o sinal de *clock* o bit **strobe** que está localizado no pino 1 do conector DB25 e para o sinal de *reset* o bit **init** que está no pino 16, ambos do registrador de controle.

Para ativar e desativar o sinal criou-se duas funções, uma de ativação que coloca o bit em 1 e outra de desativação que coloca o bit em 0. A função do *reset* ativa o sinal e depois desativa, provocando um *reset* assíncrono no contador. A função do *clock* ativa e desativa o sinal periodicamente. Como o contador tem transição positiva, a função do *clock*, quando chamada, ativa o estado pedido. As funções implementadas podem ser vistas no apêndice B.

Ao implementar as funções teve-se o cuidado de ler o estado da porta e alterar apenas o bit do sinal com a ajuda da mascara, pois uma modificação em um dos sinais poderia alterar o estado do outro.

Os testes foram feitos no contador e verificou-se que o sinal apresentava 5V na saída com um ruído de 0.4V. Entretanto o ruído encontrado não altera o estado do contador, pois o ruído está dentro da faixa tolerada e não muda o estado de 0 para 1 ou vice e versa como se verifica no manual do contador. O tempo medido para a leitura de 16 células(1 coluna) foi de 18ms usando um PC com um processador Athlon K6-3 e está mostrado na figura 4.5.

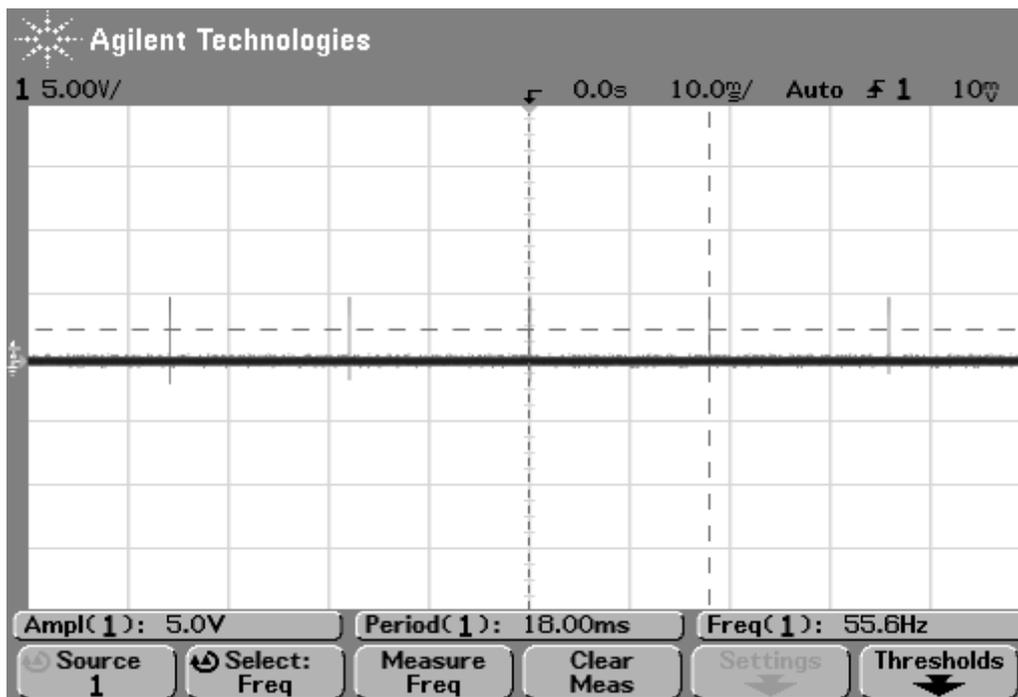


Figura 4.5: Medição do período do clock.

4.2 Conversor A/D

Para capturar os dados foi usado um cartão conversor analógico/digital. O conversor A/D utilizado foi STD-5012 que tem uma resolução de 12 Bits e apresenta um tempo de conversão de $25\mu s$. O conversor tem 16 entradas analógicas isoladas que podem ser usadas ou com 16 canais absolutos ou 8 canais diferenciais. As entradas se conectam ao exterior através do conector tipo Header de 26 pinos. O conversor apresenta fundos de escala de: $\pm 100mV$, $\pm 1V$, $\pm 5V$ e $\pm 10V$.

4.2.1 Funções do Conversor A/D

O conversor A/D já vem com funções pré-definidas. Estas funções devem ser utilizadas para inicializar o cartão e para realizar a leitura no conversor dos dados. As funções estão disponibilizadas no arquivo .obj.

As funções necessárias para funcionamento do cartão são:

- LADINIT: habilita interrupção. Neste modo os dados são lidos e armazenados em um buffer.
- LADCONF: configura o endereço da base do cartão.

- LADINI: inicializa o cartão e as variáveis internas do hardware.
- LADTMO: define o tempo de espera para que uma leitura seja aceita.
- LADMODO: configura o modo de operação do cartão. Os modos de operação são Modo Normal, Modo Trigger Externo e Modo com Interrupção.
- LADGANHO: configura o ganho do amplificador de entrada.
- Le_AD: seleciona o canal, lê os dados do conversor e retorna o valor lido.

Para o este sistema foi utilizado o Modo Normal com ganho 1. Caso ocorra algum erro estas funções retornam o parâmetro erro que será mostrado na tela por Escreve_Erro. O programa pode ser visualizado no apêndice B.

Capítulo 5

Reconhecimento de Objetos

Este capítulo introduzirá a técnica usada para o reconhecimento de objetos. Abordando a escolha dos objetos a serem utilizados no reconhecimento e serão apresentados os resultados práticos do reconhecimento dos objetos.

5.1 Escolha dos Objetos a Serem Reconhecidos

O reconhecimento dos objetos será baseado em formas regulares pré-moldadas. No início trabalhou-se com peças rígidas (cubos, pirâmides e cilindros), mas estes não ativavam o sensor de forma adequada, isto é, a forma apresentada na tela não se assemelhava à forma dos objetos. Vários testes foram feitos com diferentes objetos e percebeu-se que os objetos um pouco mais flexíveis apresentavam melhor resposta no sensor. Isto é compreendido se analisarmos o tato humano, pois a superfície da mão se molda facilmente ao objeto em contato. Entretanto, os objetos também não podiam ser muito flexíveis, pois isto fazia com que suas formas se distorcessem no sensor.

Os objetos escolhidos para se trabalhar foram de 2 dimensões e foram confeccionados com material de borracha. Como os pesos destes objetos não são suficientes para ativar as células do sensor, posicionou-se pesos em cima dos objetos para ativar as células do sensor.

5.2 Classificação das Figuras

As formas das figuras escolhidas para um primeiro reconhecimento foram o quadrado, o círculo e o triângulo por apresentarem áreas distintas para um mesmo tamanho de l , sendo

l o lado do quadrado e do triângulo e o diâmetro do círculo.

O parâmetro usado para reconhecer a figura será o número de células ativadas no sensor. Para cada figura considera-se uma faixa de valores para o número de células ativadas. O uso de faixas de valores é necessário uma vez que posicionando o objeto de diferentes maneiras no sensor, isto pode acarretar diferentes números de células ativadas. Portanto, para cada forma foram feitos muitos testes para determinar a sua faixa de número de células ativadas. Os valores encontrados para os testes realizados estão na tabela 5.1.

Forma	Valor	Faixa
Círculo	73	68-78
Triângulo	83	79-85
Quadrado	114	90-120

Tabela 5.1: Número de células ativadas por figura

5.3 Implementação do Reconhecimento das Figuras

A função que realiza o reconhecimento das figuras foi implementada em linguagem C para MS-DOS. Os valores utilizados para reconhecer os sólidos estão na tabela 5.1. Caso o valor não se encontre nas faixas definidas, o programa informa que a figura não é conhecida. Para que o sistema ative a função de reconhecimento é necessário que o objeto permaneça por 3 segundos sobre o sensor, o que corresponde a 6 ciclos de varredura, para que se inicie o reconhecimento.

Como o sensor é sensível à variação de força aplicada, também foi implementada no programa uma função para o reconhecimento de níveis de força. Os níveis de força foram definidos a partir da tabela 2.2. Desta maneira foram definidos 11 níveis de força que variam de 0.1kgf. Inicialmente o programa só mostra as células ativadas no sensor. As células só são ativadas para forças aplicadas maiores que 0.2kgf, que corresponde a tensões acima de 1.4V. Entretanto, o reconhecimento dos níveis de força pode ser gradativamente ativado pressionando-se a tecla *SPACE* do teclado. A cada toque mais um nível de força é reconhecido pelo sistema, de forma que no 11^o toque é visualizado o nível de força máximo. Com mais um toque o sistema volta a reconhecer apenas o número de células ativadas.

O programa pode ainda habilitar uma função para armazenar os dados adquiridos em um arquivo de dados(.DAT).

5.4 Resultados Experimentais

Para identificar os níveis de força foi associada a cada faixa uma cor como pode ser visto na tabela 5.2. Se os níveis de força não forem ativados a figura é reconhecida com a cor magenta.

	Força kgf	Tensão V	Cor
1	< 0.1	< 1	cinza
2	[0.1;0.2[[1;1.4[azul escuro
3	[0.2;0.3[[1.4;2.1[azul
4	[0.3;0.4[[2.1;2.3[cyan
5	[0.4;0.5[[2.3;2.5[cyan claro
6	[0.5;0.6[[2.5;2.7[verde
7	[0.6;0.7[[2.7;3.0[verde claro
8	[0.7;0.8[[3.0;3.5[amarelo
9	[0.8;0.9[[3.5;5.5[marrom
10	[0.9;1.0[[5.5;10.0[vermelho
11	> 1.0	> 10.0	vermelho claro

Tabela 5.2: Relação entre nível de força,tensão e cor

Para testar a visualização do aumento gradativo dos níveis de força, colocou-se um objeto sobre o sensor e com a tecla *SPACE* foi-se aumentando o nível máximo reconhecido. A figura 5.1 mostra até o 6^o nível de força.

Para testar o funcionamento do programa foram realizados experimentos com os objetos definidos. Os resultados obtidos para cada figura foram satisfatórios e os valores encontrados pelo acionamento das células estão dentro das faixas definidas. O primeiro teste foi realizado com o círculo e o resultado obtido está na figura 5.2. Acionando os níveis de força obtemos a figura 5.3. O teste seguinte foi realizado com o triângulo e os resultados obtidos

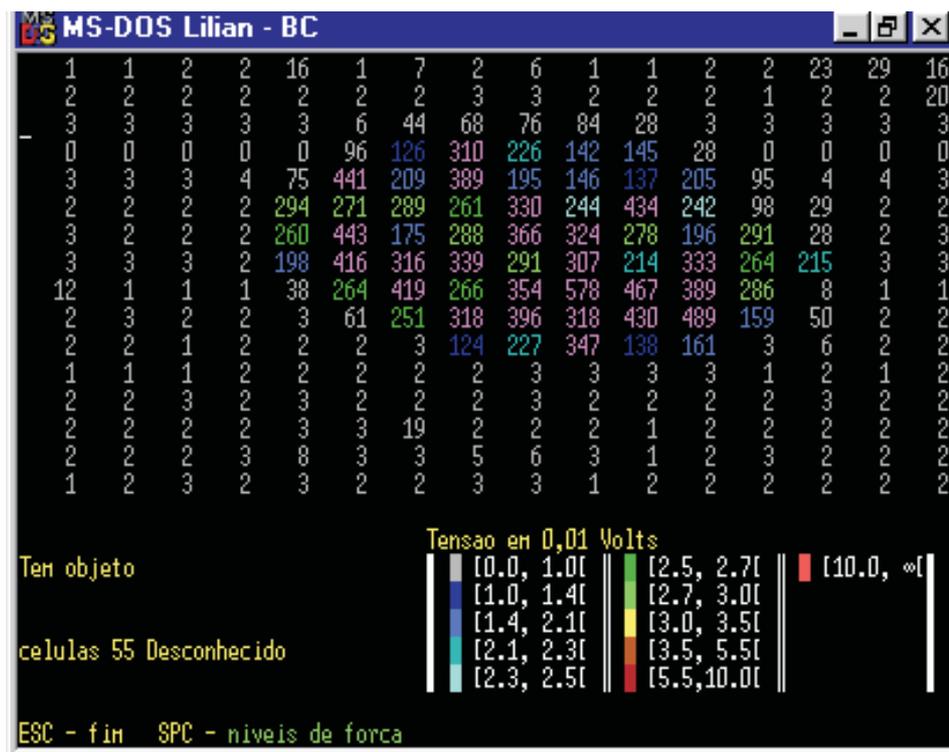


Figura 5.1: Até o sexto nível de força ativado.

estão nas figuras 5.4 e 5.5. A seguir foi realizado o teste com o quadrado e os resultados obtidos estão nas figuras 5.6 e 5.7.

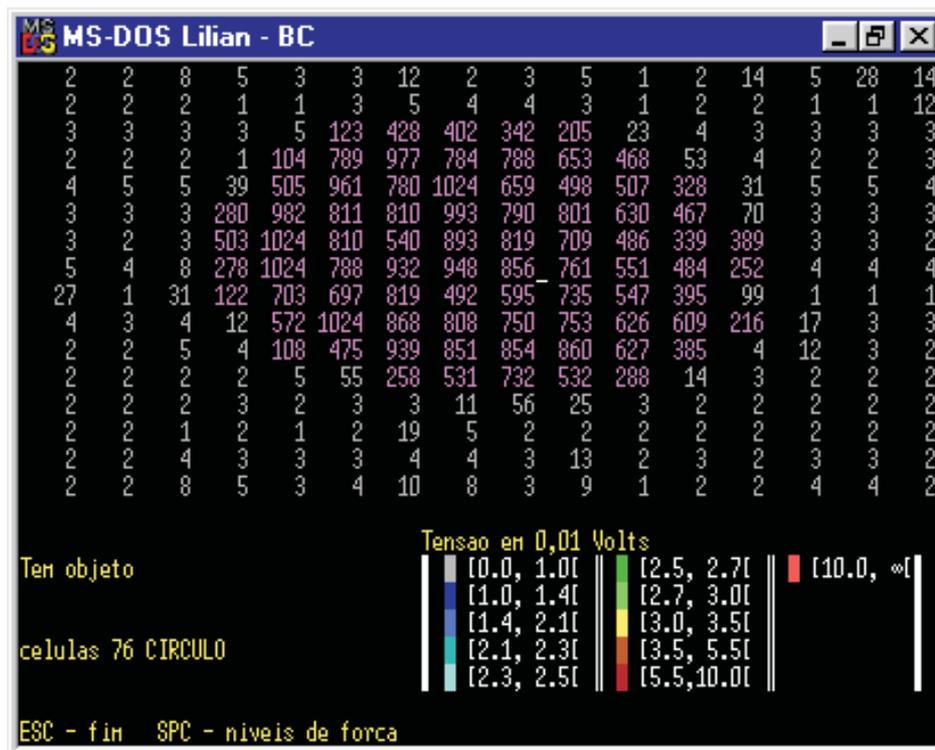


Figura 5.2: Reconhecimento no círculo.

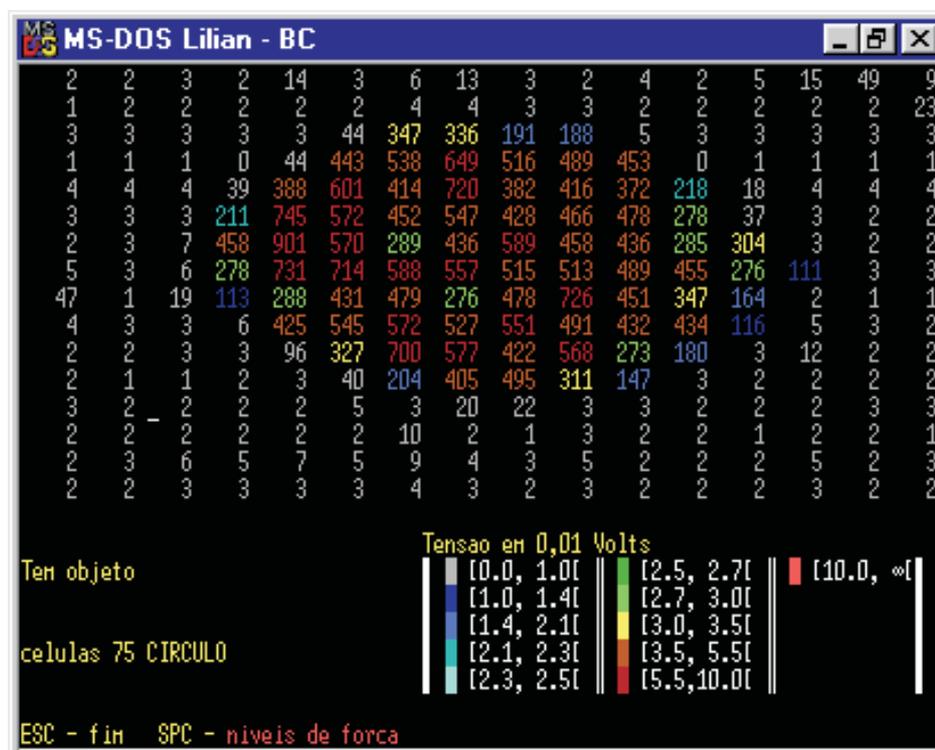


Figura 5.3: Níveis de força no círculo.

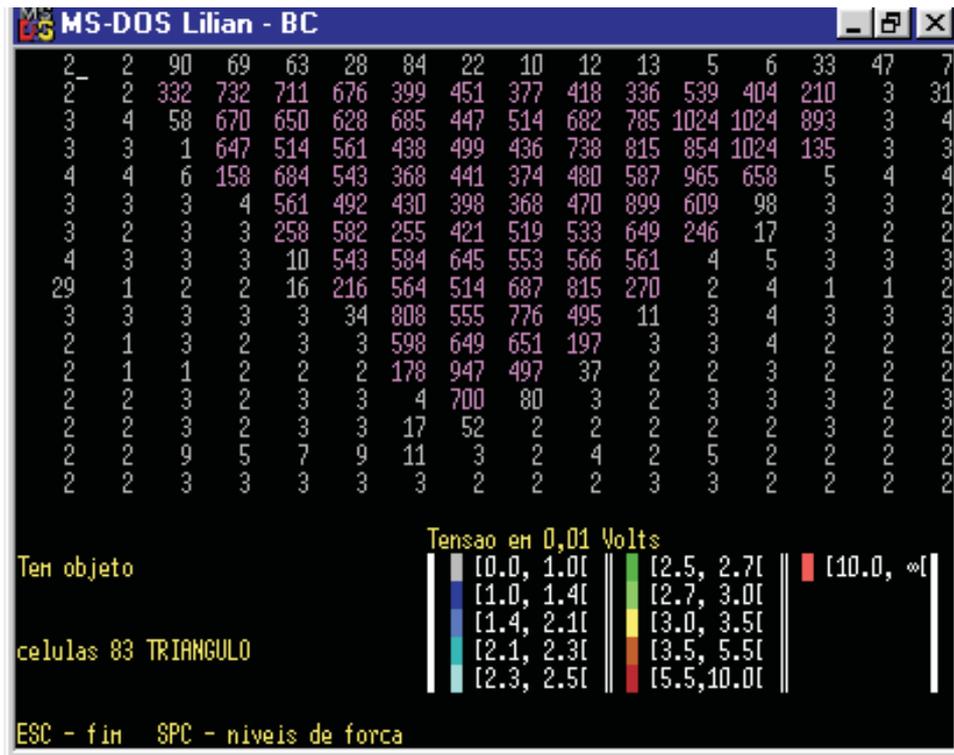


Figura 5.4: Reconhecimento no triângulo.

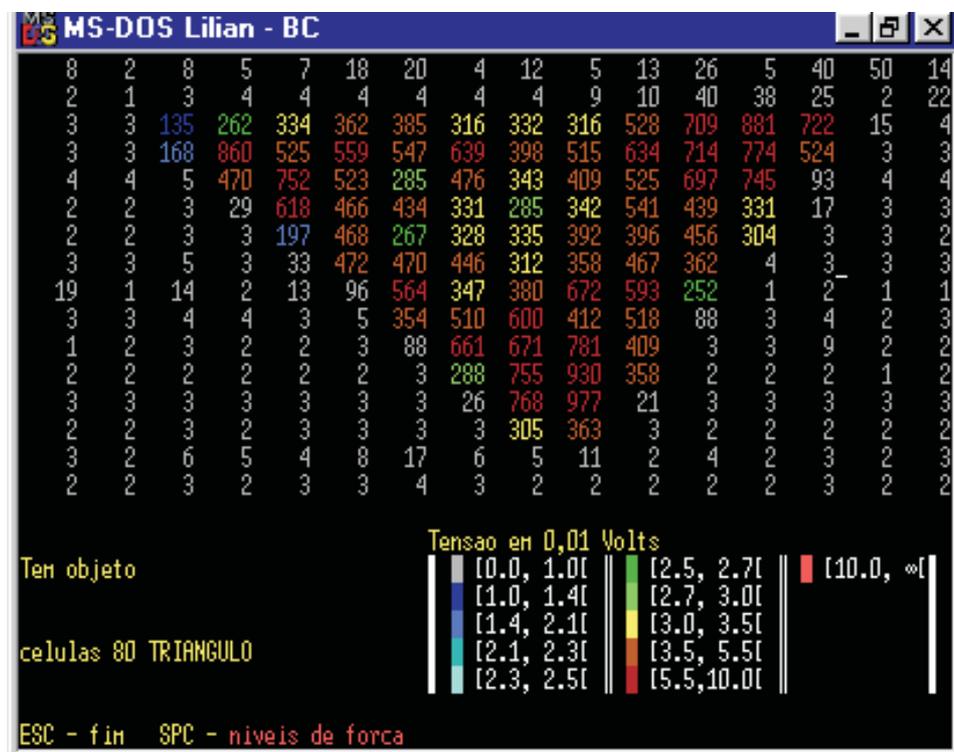


Figura 5.5: Níveis de força no triângulo.

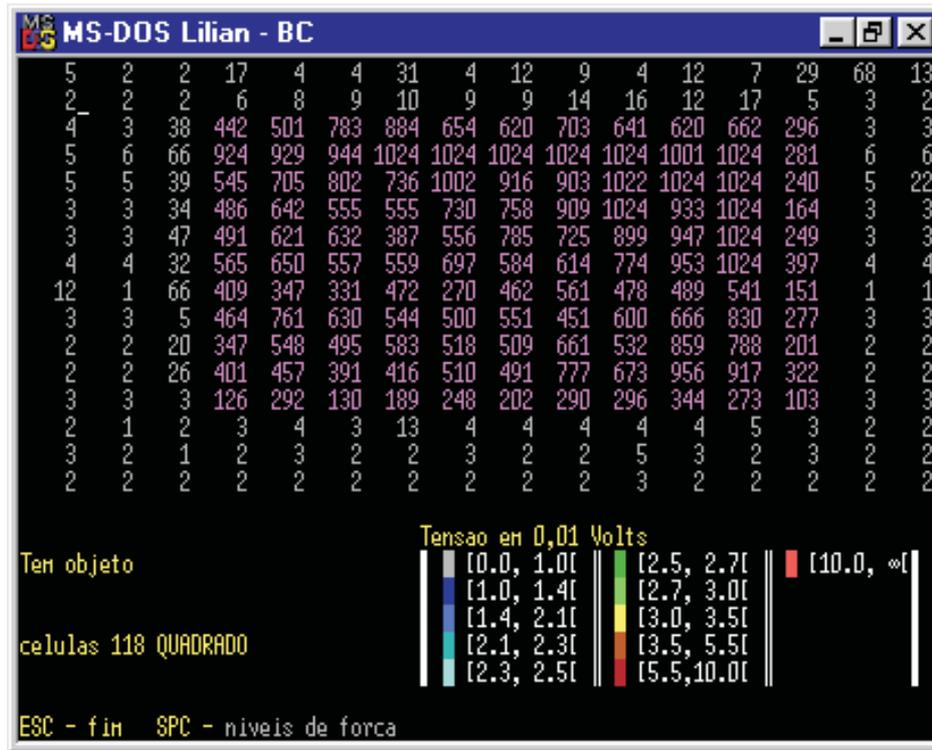


Figura 5.6: Reconhecimento no quadrado.

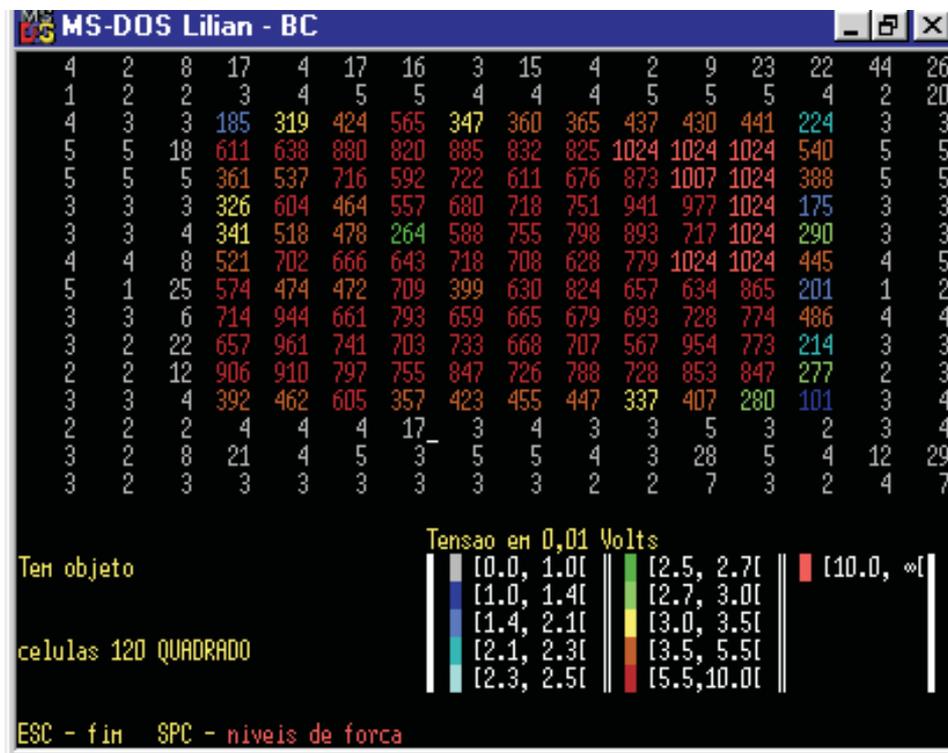


Figura 5.7: Níveis de força no quadrado.

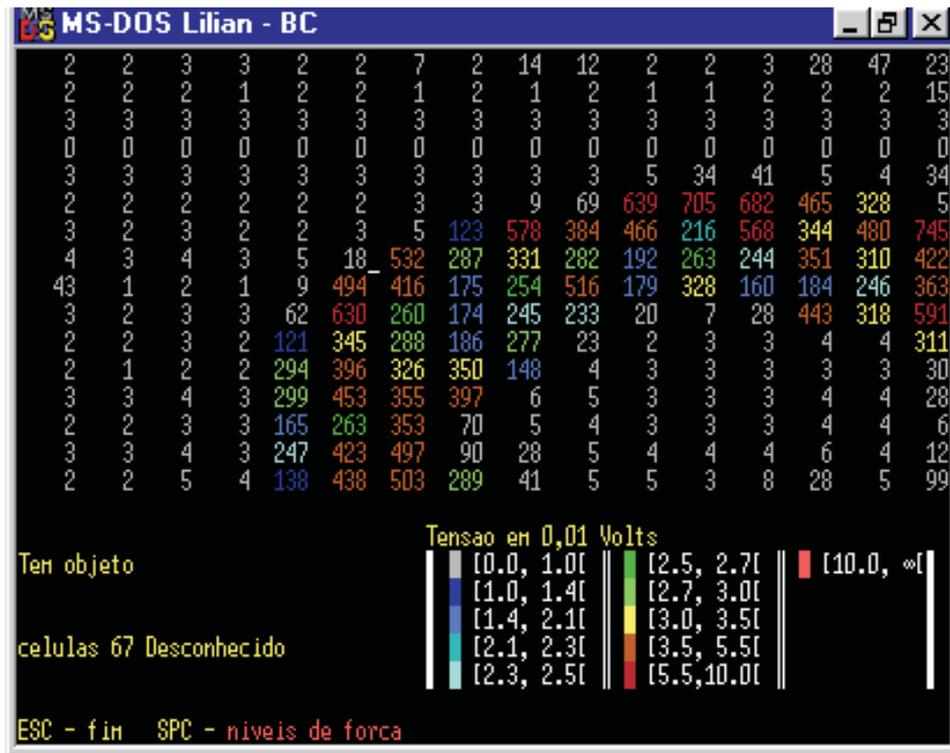


Figura 5.8: Figura desconhecida.

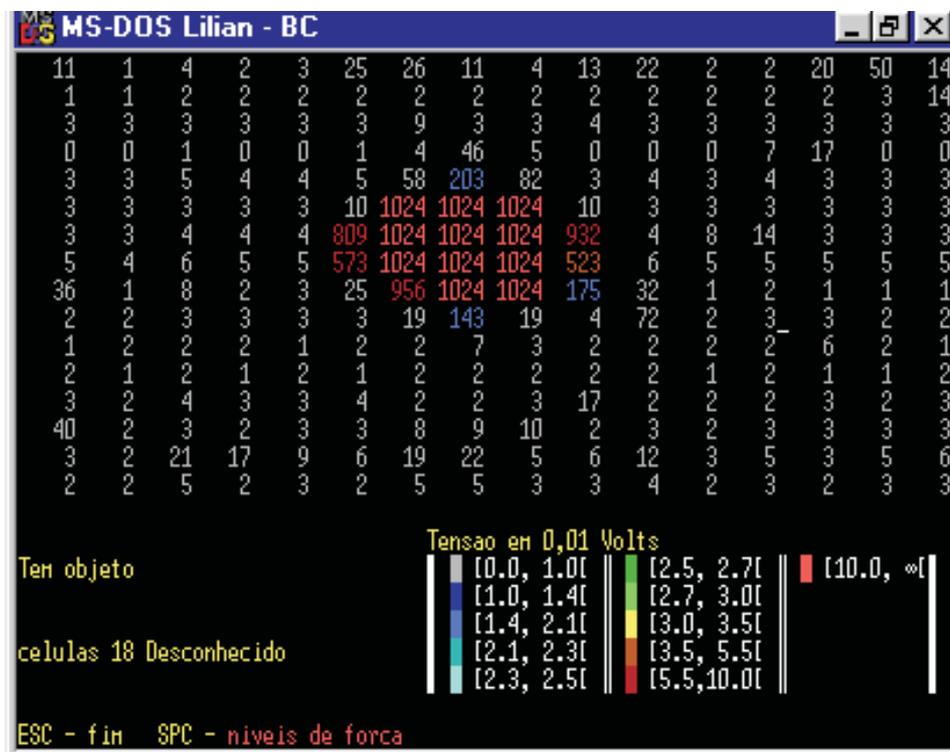


Figura 5.9: Esfera.

Capítulo 6

Conclusões

Neste trabalho foi realizada uma interface capaz de reconhecer objetos utilizando sensor de tato conectado a um microprocessador do tipo IBM/PC.

O sistema é composto basicamente por três subsistemas: o sensor de tato; a interface entre o sensor e o computador; e o subsistema capaz de reconhecer diferentes figuras pré-determinadas a partir dos dados adquiridos pela interface.

O sensor de tato utilizado foi o sensor do fabricante TEKSCAN modelo 5051. Apesar de se utilizar apenas uma parte do sensor este apresentou uma boa resolução no reconhecimento das figuras.

O segundo subsistema consistiu em projetar e montar um circuito eletrônico para converter a pressão no sensor em tensão e converter este sinal analógico em sinal digital para ser lido e processado pelo PC. Para fazer a sincronização da leitura de cada coluna do sensor com o conversor A/D foi utilizada a porta paralela.

O terceiro subsistema consistiu em implementar o reconhecimento de diferentes figuras (círculo, quadrado e triângulo) através do número de células acionadas. Este subsistema foi implementado utilizando a linguagem C.

Os resultados obtidos foram satisfatórios e a resolução obtida foi bem acima da desejada com o protótipo. Os ruídos intrínsecos ao sistema foram filtrados de forma que não interferiram nos resultados obtidos.

Com este trabalho, foi dado um passo inicial no reconhecimento de objetos. O sistema foi elaborado de forma a ser ampliado de modo simples e efetivo, caso novas técnicas sejam implementadas. Estas podem ser facilmente incluídas, devido ao alto grau de complexidade do sensor, permitindo assim a constante evolução deste trabalho.

Por fim, deve-se dizer que este trabalho não deve ser visto como algo acabado. Da mesma forma que a tecnologia está sempre em desenvolvimento, cabe aos futuros desenvolvedores interessados em dar continuidade a este trabalho, implementar novas técnicas para o reconhecimento de objetos.

6.1 Proposta de Trabalhos Futuros

Como proposta para trabalhos futuros pode-se citar:

- o melhor aproveitamento da capacidade do sensor construindo mais duas placas para utilizar todas as células do sensor;
- reconhecer outras formas e melhorar o algoritmo de reconhecimento refinando a técnica de reconhecimento;
- utilizar outras técnicas, como Lógica Fuzzy e redes neuronais, para o reconhecimento dos objetos;
- utilizar o sensor para outras aplicações como verificação da distribuição de força.

Referências Bibliográficas

- [1] SEDRA, A. S., SMITH, K. C., *Microelectronic Circuits*. Oxford University Press, 1997.
- [2] EVERETT, H. R., *Sensor for Mobile Robots: Theory and Application*. 1 ed. A K Peters Ltd, 1995.
- [3] ZELENOVSKY, R., MENDONÇA, A., *Hardware e Interfaceamento*. 2 ed. MZ Editora Ltda., 1999.
- [4] RODRIGUES, W. D., *Curso de OrCAD Layout*. UFRJ.
- [5] TERROSO, A. R., “Apostilas sobre OrCad - PUCRS”, <http://diana.ee.pucrs.br/terroso/Apostilas/apostilas.html>.
- [6] “Robótica - Porta Paralela”, <http://www.mrshp.hpg.ig.com.br/rob/paralela.htm>.
- [7] MESSIAS, A. R., “Porta Paralela”, <http://www.rogercom.com/pparalela/introducao.htm>.

Apêndice A

Esquema das Placas

A.1 Esquemático

A figura A.2 mostra o esquemático da placa dos amplificadores e a figura A.1 mostra o esquemático da placa dos contadores.

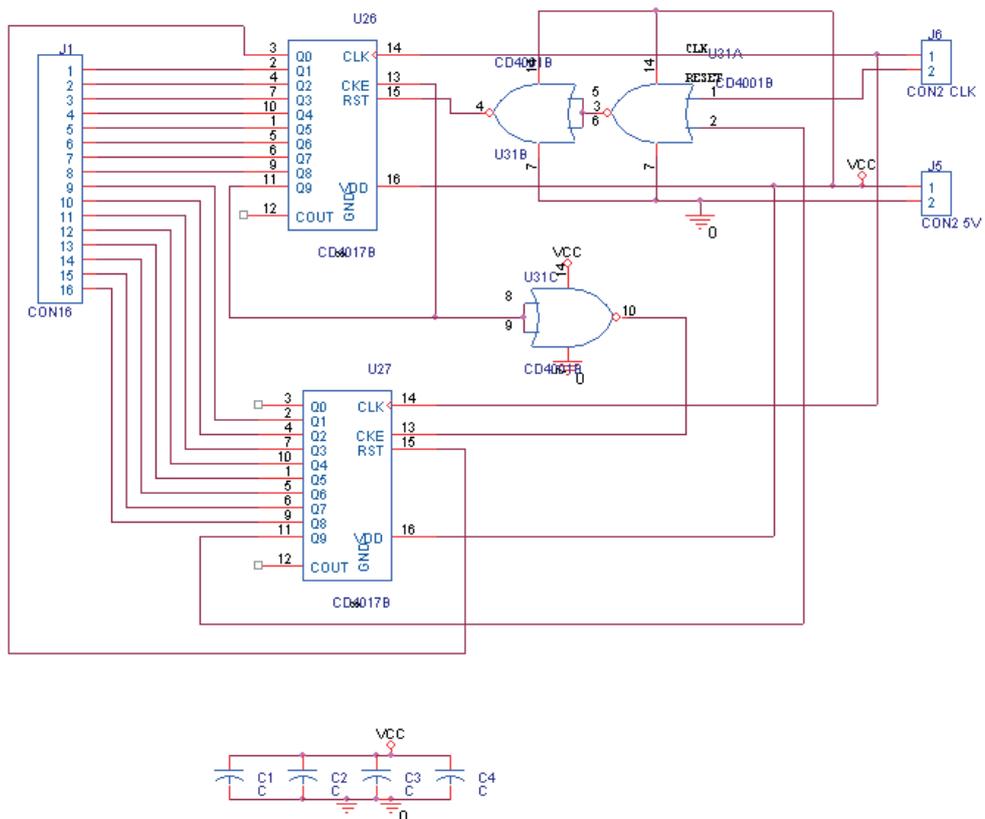


Figura A.1: Esquemático da placa dos contadores.

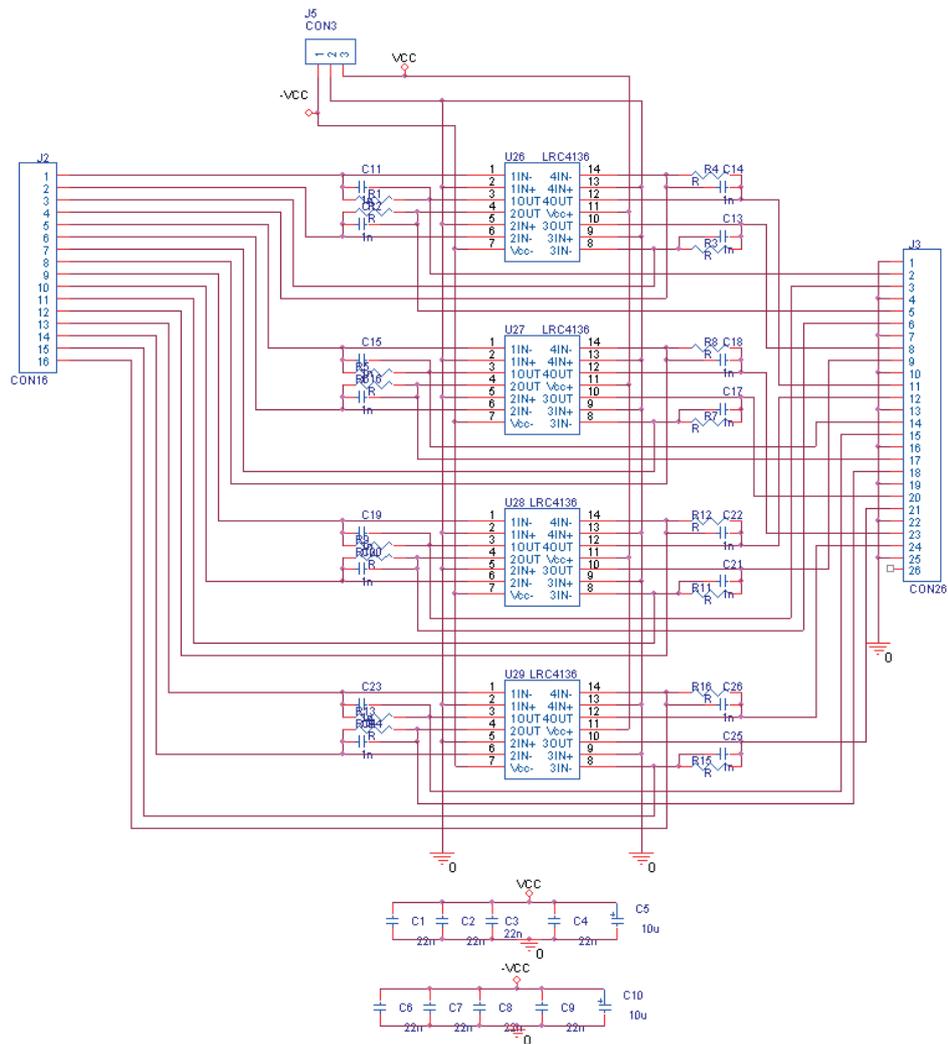


Figura A.2: Esquemático da placa dos amplificadores.

A.2 Layout Amplificador

A figura A.3 mostra o layout da face BOTTON da placa do amplificador. A figura A.4 mostra o layout da face TOP da placa do amplificador.

A.3 Layout Contador

A figura A.5 mostra o layout da face BOTTON da placa do contador. A figura A.6 mostra o layout da face TOP da placa do contador.

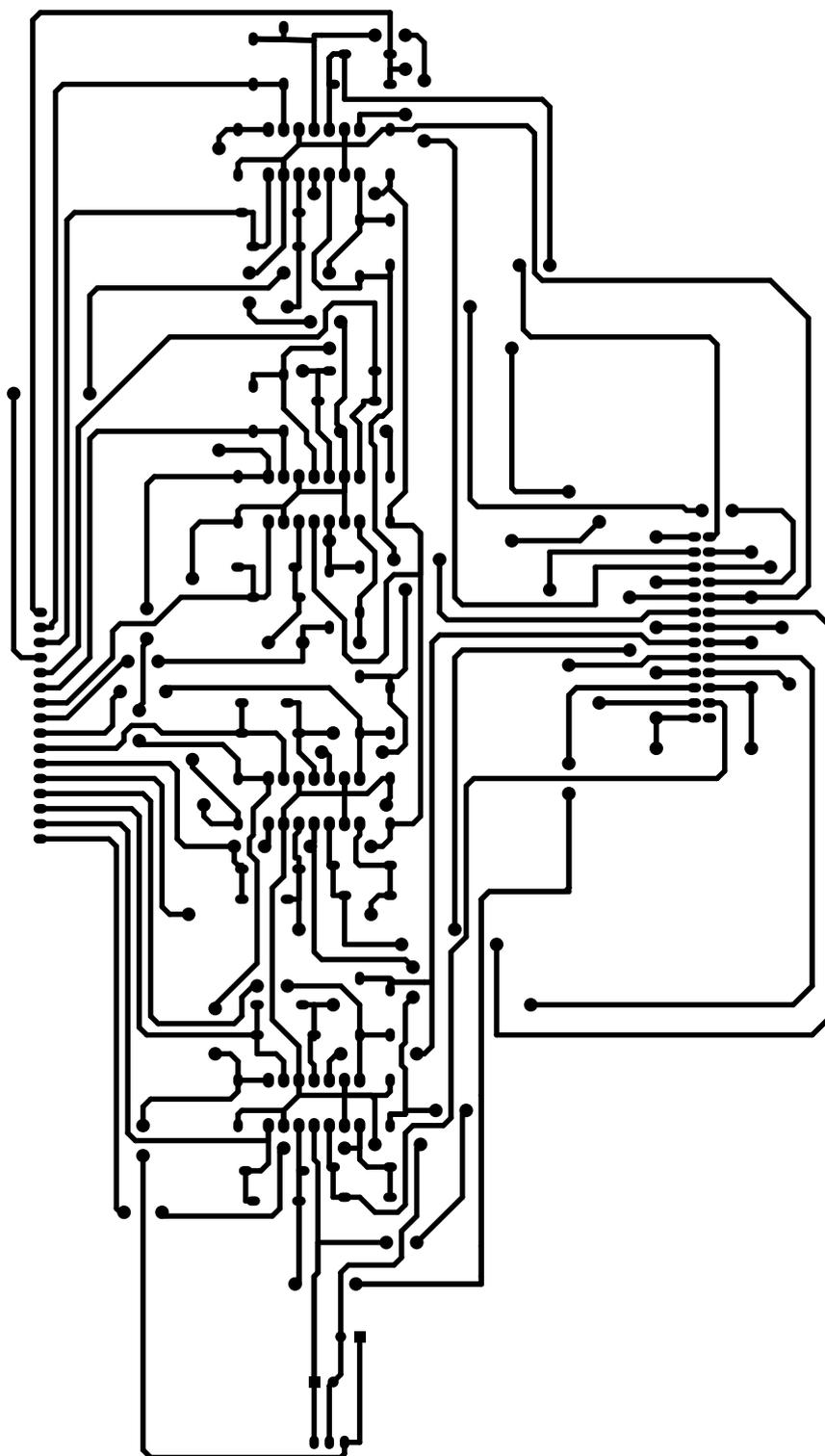


Figura A.3: Placa dos amplificadores - BOTTON.

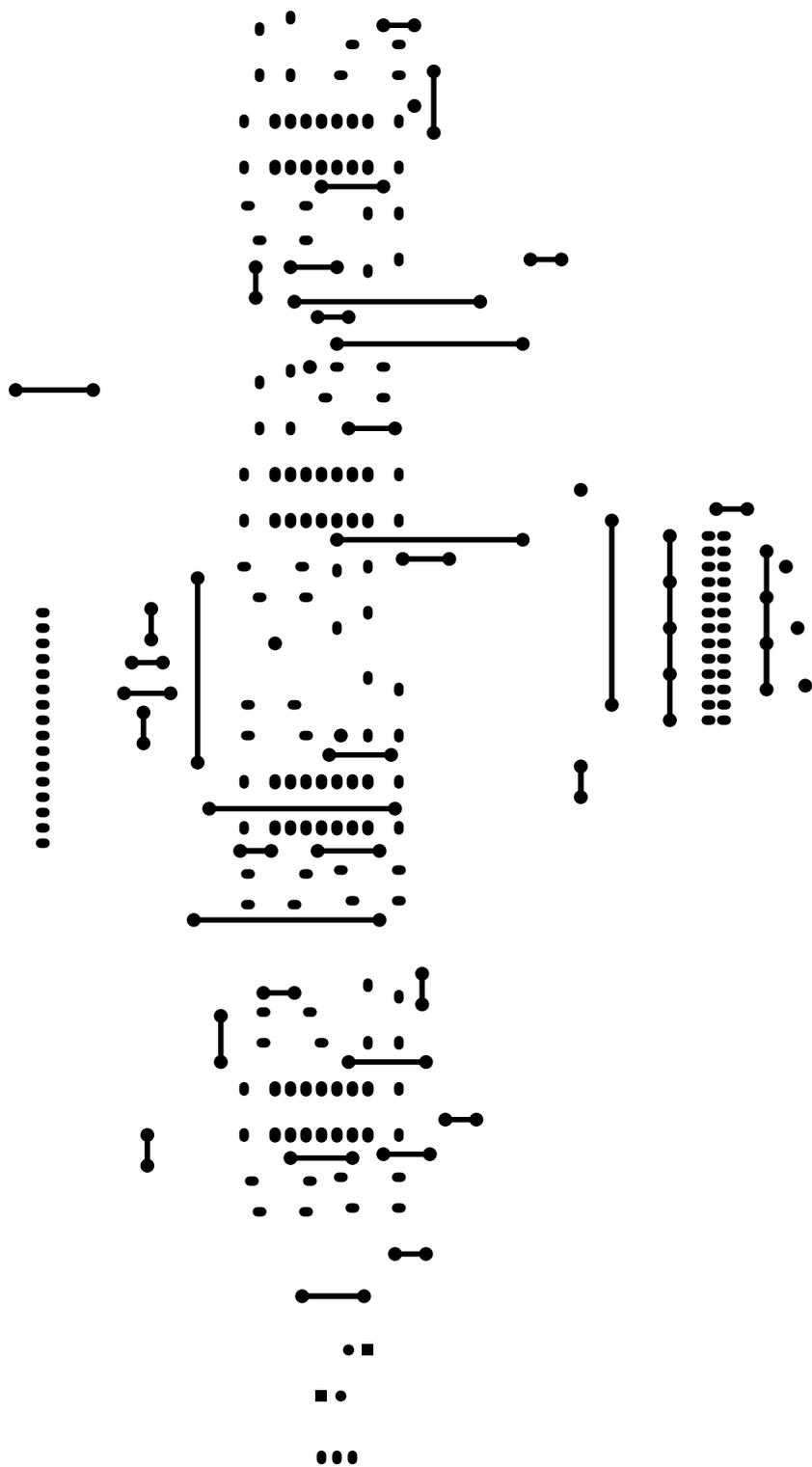


Figura A.4: Placa dos amplificadores - TOP.

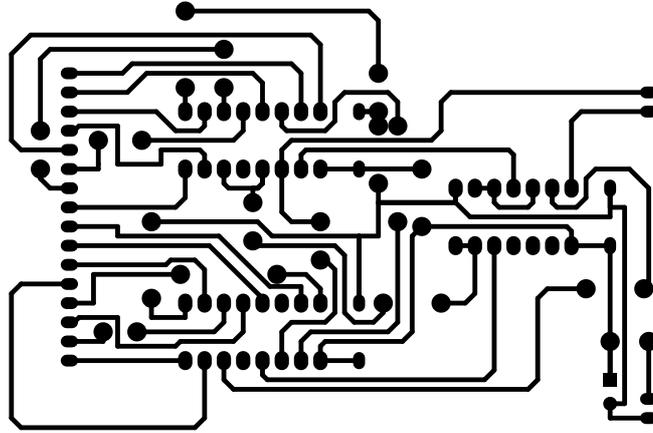


Figura A.5: Placa do contador - BOTTON.

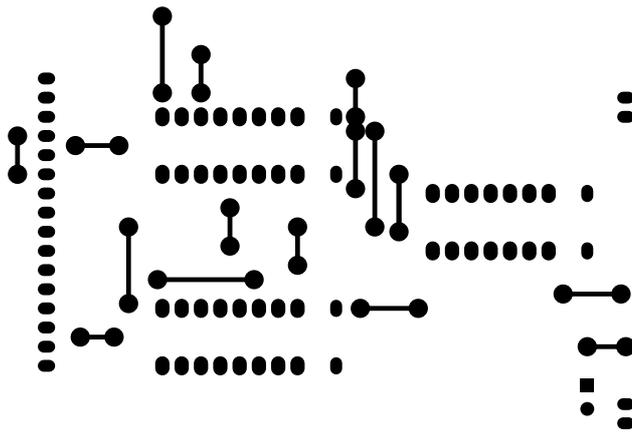


Figura A.6: Placa do contador - TOP.

Apêndice B

Programas

B.1 Programa para a Porta Paralela

O programa em C que implementa o funcionamento do *clk* e do *reset*. Quando pressiona-se uma tecla o sistema é inicializado.

```
#include <conio.h>
#include <dos.h>
#include <math.h>
#include <stdio.h>

#define N_CANAIS      16
#define FAIXA        20.475

#define CONTLPT1     0x37A
#define LIGACLK      0x01 // 0000 0001
#define LIGARESET    0x04 // 0000 0100
#define LIGATODOS    4 // 0000 0100
#define DESLIGATODOS 11 // 0000 1011

#define MASCARA      0x0B // 0000 1011

unsigned char byte;
```

```

void Reset_PIO(void) {
    //Ler o estado da porta
    byte=((inportb(CONTLPT1)&0x00F)^MASCARA);
    //Ativa o RESET
    byte|=LIGARESET;
    printf("RESET em 1..\n");
    outportb(CONTLPT1,(byte&0x0F)^MASCARA);

    //Ler o estado da porta
    byte=((inportb(CONTLPT1)&0x00F)^MASCARA);
    //Desativar o RESET
    byte=(byte&~LIGARESET);
    printf("RESET em 0..\n");
    outportb(CONTLPT1,(byte&0x0F)^MASCARA);
}

```

```

void Clk_PIO(void) {
    //Ler o estado da porta
    byte=((inportb(CONTLPT1)&0x00F)^MASCARA);
    //Ativa o CLK
    byte|=LIGACLK;
    printf("CLK em 1..\n");
    outportb(CONTLPT1,(byte&0x0F)^MASCARA);

    //Ler o estado da porta
    byte=((inportb(CONTLPT1)&0x00F)^MASCARA);
    //Desativar oCLK
    byte=(byte&~LIGACLK);
    printf("CLK em 0..\n");
    outportb(CONTLPT1,(byte&0x0F)^MASCARA);
}

```

```

void main ()
{
    clrscr ();
    while(1)
    {
        Clk_PIO();
        delay(100);
        if (kbhit())
        {
            Reset_PIO();
            getch();
        }
    }
}

```

B.2 Programa Final

Este programa agrega as funções acima e a parte de reconhecimento de objetos.

```

/*****
Programa: Teste4.c Autor: Lilian Kawakami Carvalho Data: 1/6/2003
Atualizacao: 30/8/2003
*****/

#include <conio.h>
#include <dos.h>
#include <math.h>
#include <stdio.h>

#include "adda.h"
#include "janelas.h"

//Definicao dos dados do A/D

```

```

#define END_CARTAO 0x360
#define NUM_CARTAO 0
#define TIMEOUT 0
#define MODO_NORMAL 0
#define GANHO_2 1 // Ganho do sinal igual a 2
#define GANHO_1 0 // Ganho do sinal igual a 1

#define UNIPOLAR 0
#define BIPOLAR 1
#define OPERACAO BIPOLAR

//A faixa de leitura do A/D em V
#define FAIXA 20.475

//Numero de Linhas e Colunas do Sensor
#define N_LINHA 16
#define N_COLUNA 16

//Definicao da PIO
#define CONTLPT1 0x37A
#define LIGACLK 0x01 // 0000 0010
#define LIGARESET 0x04 // 0000 0100
#define MASCARA 0x0B // 0000 1011

//Define a tecla ESC
#define ESC 27

//Define o tipo Boolean
#define BOOLEAN unsigned int
#define FALSE 0
#define TRUE 1

```

```

//0 numero niveis de forca
#define N_FAIXA 11

//variaveis globais
int erro;          //A/D
int canal = 0;     //A/D
unsigned char byte; //var PIO
unsigned char tecla = 0;
int cor[N_FAIXA] = {
    LIGHTGRAY,
    BLUE,
    LIGHTBLUE,
    CYAN,
    LIGHTCYAN,
    GREEN,
    LIGHTGREEN,
    YELLOW,
    BROWN,
    RED,
    LIGHTRED
}; //cores -> niveis de forca

float nivel[N_FAIXA-1] = {
    1.0,
    1.4,
    2.1,
    2.3,
    2.5,
    2.7,
    3.0,
    3.5,
    5.5,

```

```
10
};
```

```
float dados[N_LINHA] [N_COLUNA];
```

```
/******
```

```
Funcao: Retorna os erros do conversor A/D
```

```
*****/
```

```
void Escreve_Erro (int e)
```

```
{
```

```
    switch (e)
```

```
    {
```

```
        case 0 : printf ("%1d - Operacao completada com sucesso\n", e);
```

```
            break;
```

```
        case 1 : printf ("%1d - Erro no cartao ou cartao ausente\n", e);
```

```
            break;
```

```
        case 2 : printf ("%1d - Parametro invalido\n", e);
```

```
            break;
```

```
        case 3 : printf ("%1d - Time-out de leitura\n", e);
```

```
            break;
```

```
        case 4 : printf ("%1d - Buffer vazio\n", e);
```

```
            break;
```

```
        case 5 : printf ("%1d - Erro de configuracao\n", e);
```

```
            break;
```

```
        default: printf ("%d - Erro desconhecido ou sem erro\n", e);
```

```
    }
```

```
}
```

```
/******
```

```
Funcao: Inicializa o conversor A/D
```

```
*****/
```

```

void Inicia_AD (void)
{
    LADINIT (&erro);
    Escreve_Erro (erro);
    LADCONF (NUM_CARTAO, END_CARTAO);
    Escreve_Erro (erro);
    LADINI (NUM_CARTAO);
    Escreve_Erro (erro);
    LADTMO (NUM_CARTAO, TIMEOUT);
    Escreve_Erro (erro);
    LADMODO (NUM_CARTAO, MODO_NORMAL);
    Escreve_Erro (erro);
    LADGANHO (NUM_CARTAO, GANHO_1);
    Escreve_Erro (erro);

    delay (10);
}

/*****
Funcao: Le os dados do conversor A/D
*****/

float Le_AD(int canal)
{
    int valor;

    valor=0;
    LADSELCANAL (NUM_CARTAO, canal);
    delay(1);
    LADLER (NUM_CARTAO, &valor);
}

```

```

if (OPERACAO == BIPOLAR)
    if (!(valor & 0x0800))
        valor &= 0x0FFF;
    else
        valor |= 0xF000;
    else
        valor = (valor & 0xFFF) ^ 0x0800;

return (FAIXA*(float)valor/4095);
}

/*****
Funcao: Inicializa o contador pela porta paralela
*****/

void Reset_PIO(void)
{
    //Ler o estado da porta
    byte=((inportb(CONTLPT1)&0x00F)^MASCARA);
    //Ativa o RESET
    byte|=LIGARESET;
    outportb(CONTLPT1,(byte&0x0F)^MASCARA);

    //Ler o estado da porta
    byte=((inportb(CONTLPT1)&0x00F)^MASCARA);
    //Desativar o RESET
    byte=(byte&~LIGARESET);
    outportb(CONTLPT1,(byte&0x0F)^MASCARA);
}

/*****

```

Funcao: 0 clk do circuito pela porta paralela

*****/

```
void Clk_PIO(void)
{
    //Ler o estado da porta
    byte=((inportb(CONTLPT1)&0x00F)^MASCARA);
    //Ativa o CLK
    byte|=LIGACLK;
    outportb(CONTLPT1,(byte&0x0F)^MASCARA);

    //Ler o estado da porta
    byte=((inportb(CONTLPT1)&0x00F)^MASCARA);
    //Desativar oCLK
    byte=(byte&~LIGACLK);
    outportb(CONTLPT1,(byte&0x0F)^MASCARA);

}
```

Funcao: Escolhe a cor pelo nivel de forca

*****/

```
int setacor (float valor_v)
{
    int i;

    for (i=0;i < (N_FAIXA-1);i++)
    {
        if (valor_v <= nivel[i])
            return i;
    }
}
```

```

    return i;

}

/*****
Funcao: Escreve os dados no arquivo
*****/

void EscreveArq (int n_aqs)
{
    FILE *lista;
    int linha, coluna;

    lista = fopen ("dados.dat", "a+");
    fprintf (lista, "\nAquisisao %d :\n\n",n_aqs);
    for (linha = 0; linha < N_LINHA; linha++)
    {
        for (coluna = 0; coluna < N_COLUNA; coluna++)
            fprintf (lista, "%7.3f ", dados [linha] [coluna]) ;
        fprintf (lista, "\n");
    }
    fclose (lista);
}

/*****
Funcao: Verifica qual objeto pelo no celulas ativadas
*****/

void VerificaObj(void)
{
    int linha, coluna, cel;

```

```

cel=0;
for (linha = 0; linha < N_LINHA; linha++)
{
    for (coluna = 0; coluna < N_COLUNA; coluna++)
    if (fabs(dados [linha] [coluna])>1.4)
        cel++;
};

cprintf("celulas %d ",cel);
if ((68<=cel)&(cel<=78))
    cprintf("CIRCULO");
else if ((79<=cel)&(cel<=85))
    cprintf("TRIANGULO");
else if ((90<cel)&(cel<=120))
    cprintf("QUADRADO");
    else cprintf("Desconhecido");
}

/*****
Funcao: Principal
*****/

void main ()
{
    int modo_faixa=0;
    int x,l;
    int linha,coluna;
    float valor_V;
    int v, n, ciclo;
    BOOLEAN obj;
    //variavel da janela
    int j_dados, j_status, j_obj, j_leg, j_com;

```

```

//Inicializa a placa A/D
clrscr ();
Inicia_AD ();

//Inicializa a PIO - RESET
Reset_PIO();

clrscr();

//Definir Janelas
j_dados = DefineJanela(1,1,80,N_LINHA+1,YELLOW,BLACK);
j_status = DefineJanela(1,N_LINHA+2,20,N_LINHA+5,YELLOW,BLACK);
j_obj = DefineJanela(1,N_LINHA+6,25,N_LINHA+8,YELLOW,BLACK);
j_leg = DefineJanela(36,N_LINHA+2,80,N_LINHA+8,YELLOW,BLACK);
j_com = DefineJanela(1,25,80,25,YELLOW,BLACK);

//Montando janela barra de comandos
AtivaJanela(j_com);
cprintf("ESC - fim SPC - niveis de forca");

//Montando janela Legendas
AtivaJanela(j_leg);
cprintf("Tensao em 0,01 Volts\r\n");
for (l=0;l<N_FAIXA/2;l++)
{

    textcolor(WHITE);
    cprintf("\xdd ");
    textcolor(cor[l]);
    cprintf ("\xdb ");
    textcolor(WHITE);

```

```

cprintf("[%3.1f,%4.1f[ \xba",nivel[l-1],nivel[l]);
textcolor(cor[N_FAIXA/2+1]);
cprintf (" \xdb " );
textcolor(WHITE);
cprintf("[%3.1f,%4.1f[ \xba",nivel[(N_FAIXA/2+1)-1],nivel[N_FAIXA/2+1]);
if (l==0)
{
textcolor(cor[N_FAIXA-1]);
cprintf (" \xdb " );
textcolor(WHITE);
cprintf("[%3.1f, \xec[\xdd\r\n",nivel[N_FAIXA-2]);
}
else
{
cprintf("          \xdd\r\n");
}
}

//contador
n=0;
ciclo=0;
obj=FALSE;

while(tecla != ESC)
{

n++;

AtivaJanela(j_dados);
gotoxy(1,1);
x=1;

```

```

for (coluna=0;coluna<N_COLUNA;coluna++)
{
    Clk_PIO();

    for (linha=0;linha<N_LINHA;linha++)
    {
//le os dados do A/D
valor_V=Le_AD(linha);
gotoxy(x,linha+1);
dados [linha] [coluna] = valor_V;

//verifica se uma cel esta ativa
if (fabs(valor_V)>1.4) obj=TRUE;

//Seleciona a cor
v=setacor(fabs(valor_V));

//Cor-modod por faixa ou unico
if (v > modo_faixa)
    textcolor(LIGHTMAGENTA);
else
    textcolor(cor[v]);

//Escreve na tela os dados
cprintf ("%5.0f",fabs(valor_V)*100);
    }

    x=wherex();
}

//verifica se alguma tecla foi ativada
if (kbhit())

```

```

{
    tecla = getch();
    modo_faixa++;
    if (modo_faixa == N_FAIXA)
modo_faixa = 0;

    //muda a cor de niveis de forca
    AtivaJanela(j_com);
    gotoxy(19,1);
    textcolor(cor[modo_faixa]);
    cprintf("niveis de forca");
}

```

```

AtivaJanela(j_status);

```

```

gotoxy(1,2);

```

```

if (obj==FALSE)

```

```

{
    clreol();
    cprintf("Nao tem objeto\r\n");
    AtivaJanela(j_obj);
    gotoxy(1,1);
    clreol();
    ciclo=0;
}

```

```

else

```

```

{
    // Verificando se tem objeto
    gotoxy(1,2);
    textcolor(YELLOW);
    clreol();
    cprintf("Verificando\r\n");
}

```

```

obj=FALSE;
ciclo++;
if (ciclo>=6)
{
    //Tem objeto
    gotoxy(1,2);
    clreol();
    cprintf("Tem objeto\r\n");
    AtivaJanela(j_obj);
    gotoxy(1,1);
    clreol();
    //Chama a funcao classifica obj
    VerificaObj();
    //armazenas os dados lidos
    //2EscreveArq(ciclo-5);

};
};
Reset_PIO();

}
}

```

Apêndice C

Manuais