


COICHI URANO
FERNANDO OBARA SUZUKI
HEITOR HIDEHARU NAKAMURA

8,0 (oitava zero)



JOYSTICK SENSITIVO PARA INTERAÇÃO COM O AMBIENTE VIRTUAL

Trabalho a ser apresentado à Escola
Politécnica da Universidade de São
Paulo para obtenção do título de
bacharel em Engenharia Mecatrônica.

São Paulo
1º Semestre de 2004

COICHI URANO
FERNANDO OBARA SUZUKI
HEITOR HIDEHARU NAKAMURA

JOYSTICK SENSITIVO PARA INTERAÇÃO COM O AMBIENTE VIRTUAL

Trabalho a ser apresentado à Escola
Politécnica da Universidade de São
Paulo para obtenção do título de
bacharel em Engenharia.

Área de Concentração:
Engenharia Mecatrônica

Orientador:
Prof. Dr. Tarcísio Hess Coelho

São Paulo
1º Semestre de 2004

Dedicamos este trabalho aos nossos pais, que sempre estiveram ao nosso lado nos apoiando e incentivando. Obrigado por tudo.

AGRADECIMENTOS

A nossos familiares e amigos, pela ajuda e apoio prestados não só na realização deste trabalho de formatura, mas também durante a nossa formação na Escola Politécnica.

Ao nosso amigo e orientador, professor Tarcísio, que sempre esteve ao nosso lado, não nos deixando desanimar e nos incentivando a todo momento.

A Deus, por nos conceder a oportunidade de realizar este trabalho.

A todos que, direta ou indiretamente, colaboraram na execução deste trabalho.

RESUMO

Este trabalho trata da implementação prática de um sistema háptico de interação com ambiente virtual, envolvendo o mecanismo, a eletrônica do dispositivo e a programação do ambiente virtual. O resultado obtido foi um protótipo com total integração destes três subsistemas. Este texto também tem por objetivo transmitir a experiência acerca da prática de projeto, vivida pelos autores, a futuros projetistas de sistemas hápticos.

ABSTRACT

This work deals with the development of a haptic device, which interacts with a virtual environment. The whole system is composed by a mechanism, an electronic board and a computing interface. The built prototype is a result of total integration of these subsystems. The aim of the text is also to transmit the experience acquired by authors in developing the mentioned system, to help future haptic device designers.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

<u>1. INTRODUÇÃO</u>	1
<u>2. OBJETIVO</u>	2
<u>3. DESENVOLVIMENTO DO PROJETO</u>	4
<u>3.1. Mecanismo</u>	4
<u>3.1.1. Definições Iniciais</u>	5
<u>3.1.2. Construção</u>	8
<u>3.1.3. Partes do Mecanismo</u>	10
<u>3.1.3.1. Terminal</u>	11
<u>3.1.3.2. Cadeias Cinemáticas</u>	13
<u>3.1.3.3. Base</u>	15
<u>3.1.4. Material Utilizado</u>	16
<u>3.1.5. Lista de Desenhos</u>	17
<u>3.2. Controle</u>	17
<u>3.2.1. Controle de Potência</u>	18
<u>3.2.2. Placa de inversão de sentido e habilitador de movimento</u>	28
<u>3.2.2.1. Circuito interpretador de dados que controla a</u> <u>inversão de sentido e habilitação de funcionamento</u>	31
<u>3.2.3. Circuito de percepção</u>	34
<u>3.2.3.1. Dimensionamento do Rz (resistor de proteção do</u> <u>Diodo Zener)</u>	38
<u>3.2.3.2. Dimensionamento do Rseg (resistor de segurança)</u>	39
<u>3.2.3.3. Circuito interpretador ou combinatório do circuito</u> <u>A/D</u>	39
<u>3.2.4. Motor CC</u>	40
<u>3.2.5. Técnica utilizada para a implementação das placas</u>	46
<u>3.3. Ambiente Virtual</u>	48
<u>3.3.1. Introdução</u>	48
<u>3.3.2. Especificações</u>	48
<u>3.3.3. O programa</u>	49
<u>3.3.3.1. Funções do programa</u>	49
<u>3.3.3.2. DirectX</u>	50
<u>3.3.3.3. Terminologia:</u>	51
<u>3.3.3.4. Porta Paralela</u>	53
<u>4. DISCUSSÃO E CONCLUSÃO</u>	55

ANEXO A – DESENHOS DE FABRICAÇÃO

ANEXO B – DATASHEETS DE COMPONENTES

ANEXO C – CÓDIGO-FONTE DO AMBIENTE VIRTUAL

REFERÊNCIAS BIBLIOGRÁFICAS

LISTA DE FIGURAS

<u>Figura 4.1.1 – Junta de Revolução (Maquete)</u>	6
<u>Figura 4.1.2 – Junta Esférica(Maquete)</u>	6
<u>Figura 4.1.3 – Maquete</u>	7
<u>Figura 4.1.4 – Grau de Liberdade do Protótipo</u>	8
<u>Figura 4.1.5 – Rótula Radial (Junta Esférica)</u>	11
<u>Figura 4.1.6 – Rótula Radial</u>	11
<u>Figura 4.1.7 – Comparação da Angulação</u>	12
<u>Figura 4.1.8 – Terminal de Nylon com vista superior</u>	12
<u>Figura 4.1.9 – Interligação da Cadeia Cinemática</u>	13
<u>Figura 4.1.10 – Desenho de corte do conjunto Terminal/Cadeia Cinemática</u>	14
<u>Figura 4.1.11 – Desenho Isométrico do Protótipo</u>	16
<u>Figura 4.2.1 – Diagrama de Blocos do sistema de controle</u>	18
<u>Figura 4.2.2 – Esquema elétrico da placa de potência</u>	19
<u>Figura 4.2.3 – Direções de movimento do terminal e rotações de sentido contrário quando o terminal é elevado ou abaixado</u>	20
<u>Figura 4.2.4 – Direções de movimentos e motores que funcionarão para fornecer o torque reativo em cada direção</u>	21
<u>Figura 4.2.5 – Mapeamento para simplificação das funções lógicas</u>	24
<u>Figura 4.2.6 – Diagrama lógico das funções que acionarão as placas de potência</u>	26
<u>Figura 4.2.7 – Diagrama de blocos do L298</u>	28
<u>Figura 4.2.8 – esquema de conexão do L298 para o controle do Motor CC</u>	31
<u>Figura 4.2.9 – Mapeamento para simplificar as funções de controle do L298</u>	33
<u>Figura 4.2.10 – Diagrama lógico das funções de controle do L298</u>	34
<u>Figura 4.2.11 – discretização do potenciômetro em 5 níveis</u>	35
<u>Figura 4.2.12 – circuito A/D do sensor</u>	36
<u>Figura 4.2.13 – Diagrama lógico do conversor A/D</u>	40
<u>Figura 4.2.14 – Descrição do motor encontrado na Santa Efigênia</u>	42
<u>Figura 4.2.15 – Esquema elétrico do motor</u>	44
<u>Figura 4.2.16 – Foto do Joystick com cabo extra para facilitar movimento</u>	46
<u>Figura 4.2.17 – Fotos de confecção dos circuitos impressos</u>	47

Figura 4.3.1 – Pinagem da porta paralela.....	54
---	----

LISTA DE TABELAS

<u>Tabela 4.1.1 – Tipo de Juntas Utilizadas</u>	9
<u>Tabela 4.1.2 – Cadeias Cinemáticas do Protótipo</u>	10
<u>Tabela 4.1.3 – Numeração dos Desenhos de Fabricação</u>	17
<u>Tabela 4.2.1 – Resistências que serão utilizadas em cada placa de potência para controle de corrente</u>	19
<u>Tabela 4.2.2 – Tabela da verdade para o controle do circuito de potência</u> ..	23
<u>Tabela 4.2.3 – Tabela da verdade associada aos níveis de torque dos motores CC</u>	27
<u>Tabela 4.2.4 – Valores de resistências e correntes para os níveis de torque</u>	28
<u>Tabela 4.2.5 – Dados de entrada para o controle de uma ponte H</u>	29
<u>Tabela 4.2.6 – Limites máximos dos parâmetros do L298</u>	30
<u>Tabela 4.2.7 – Tabela da verdade para o controle do L298</u>	32
<u>tabela 4.2.8 – tabela da verdade que relaciona os bits de saída do circuito do sensor com sinais utilizados no PC</u>	37
<u>Tabela 4.2.9 – Características funcionais do motor encontrado</u>	43
<u>Tabela 4.3.1 – Principais funções do Ambiente Virtual</u>	49

1. INTRODUÇÃO

Com a rapidez do avanço tecnológico, a interação do ser humano com máquinas vem se tornando cada vez mais comum no dia-a-dia, através do uso de equipamentos de alta sofisticação e complexidade. Isto se deve principalmente à enorme capacidade de processamento dos microprocessadores, no qual um simples computador pessoal pode realizar tarefas que há alguns anos atrás só podiam ser realizadas por enormes e caros computadores.

Uma destas tarefas é o chamado CAD (Computer Aided Design), no qual o computador é utilizado para desenho e visualização de objetos em três dimensões. Além do fato de ser muito mais fácil de se visualizar um desenho no computador, em três dimensões, do que um desenho feito à mão, os programas de CAD que existem atualmente possuem diversas funções que permitem a criação de objetos com apenas alguns cliques do mouse.

Atualmente, existem no mercado alguns dispositivos que substituem o mouse, facilitando e ampliando a habilidade de movimentação no ambiente virtual. Alguns mais sofisticados são chamados de "haptic devices", nos quais há uma realimentação de força para a sensação de tato, e tornando assim o movimento mais preciso e realístico. Esta denominação "haptic" se origina da língua inglesa e não se restringe somente à percepção de tato, mas também de força, calor, vibração e movimento.

2. OBJETIVO

O objetivo deste trabalho é projetar e implementar um dispositivo de movimentação em um ambiente virtual para interação com objetos tridimensionais com graus de rigidez variados, de forma que ao tocar em uma restrição virtual (parede) enquanto se movimenta pelo ambiente, o usuário possa sentir a força de reação proporcional à rigidez (force feedback – realimentação de força) do objeto. Tal dispositivo utilizará motores, que farão o papel de atuadores (força de reação) e sensores de posição (potenciômetros), sendo que estarão conectados a um computador, que simulará o ambiente virtual.

A motivação para o desenvolvimento de um joystick de interação está no fato de que o dispositivo permite ao usuário sentir através das forças reativas as restrições existentes no campo visual registrados por uma câmera ou um simulador virtual. A Principal consequência disso é a melhora da precisão do dispositivo, uma vez que o usuário sente as forças de reação impostas pelo meio externo. Deste modo, o usuário pode ter a noção do físico apresentado, desenvolvendo habilidades de manuseio em simuladores ou podendo realizar cirurgias com braços robóticos a longas distâncias. Pode-se criar sistemas controlados por um usuário que se encontra a quilômetros de distância, tendo este a noção física exata do local, manipulando o objeto a ser controlado de acordo com as restrições existentes sentidas pelo usuário. Como exemplo disso, temos mecanismos que devem ser controlados em ambientes nocivos, com apenas um joystick

que transmite ao usuário a restrição real encontrada pelo mecanismo no local de operação e evitando a contaminação do ser humano.

3. DESENVOLVIMENTO DO PROJETO

Devido à grande extensão do projeto do joystick, foram definidas três frentes de trabalho: Mecanismo, responsável pela estrutura do joystick; Controle, responsável pela interligação eletrônica do Mecanismo com o computador, e Ambiente Virtual, responsável pela confecção do programa que simula o ambiente tridimensional. A seguir, temos os materiais e métodos utilizados por cada frente de trabalho para a execução deste projeto.

3.1. Mecanismo

Para projetar e construir o Mecanismo do Joystick, utilizamos o conceito de Mecanismos Paralelos, que é definido como um mecanismo na qual a sua base é interligada com o terminal móvel (ou no nosso caso a manche do Joystick), com pelo menos duas cadeias cinemáticas independentes. O Mecanismo Paralelo oferece ao usuário uma maior precisão nos movimentos, maior rigidez e velocidade comparada aos mecanismos de cadeia aberta. Os mecanismos paralelos estão sendo cada vez mais estudados e implementados devido às vantagens já citadas, sendo utilizados em usinagens de peças, em simulações, em haptic devices e em muitas outras aplicações. Através dos Mecanismos Paralelos foi possível obter os Graus de Mobilidade necessários para o funcionamento do sistema.

3.1.1. Definições Iniciais

Foi definido na primeira etapa do projeto de formatura que o protótipo deveria ter no mínimo 3 graus de liberdade para que seja possível uma movimentação razoável dentro de um ambiente virtual em 3 dimensões. A partir dessa definição inicial foi definido para cada uma das 4 cadeias cinemáticas ativas, 5 Graus de Liberdade e 3 Graus de Liberdade na cadeia cinemática passiva.

Portanto foi construída, primeiramente uma maquete para que seja possível ter uma noção física e prática do que estaria sendo projetado, e se os graus de liberdade determinados seriam suficientes para que no final obtenhamos 3 Graus de Mobilidade para o sistema como um todo.

Na construção da maquete foram utilizados materiais de fácil manuseio como papelão, bolas de isopor, madeiras, canudo e colas, para que não detivéssemos muito tempo na sua construção, já que essa não seria a idéia principal dessa primeira etapa do projeto. Portanto para simular juntas de revolução foram utilizadas duas madeiras finas e compridas e interligamos nas suas extremidades formando dessa maneira uma das juntas (Figura 4.1.1).

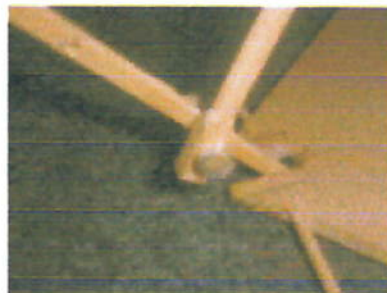


Figura 4.1.1 – Junta de Revolução (Maquete)

Utilizando canudos e as bolas de isopor foram simuladas as juntas esféricas onde se grudou a extremidade da madeira com a bola de isopor. Já que o canudo possui uma certa flexibilidade foi possível obter os 3 graus de liberdade necessária para uma junta esférica (Figura 4.1.2). A junta prismática foi obtida através da construção de uma caixa na qual havia uma madeira que atravessava o seu interior.

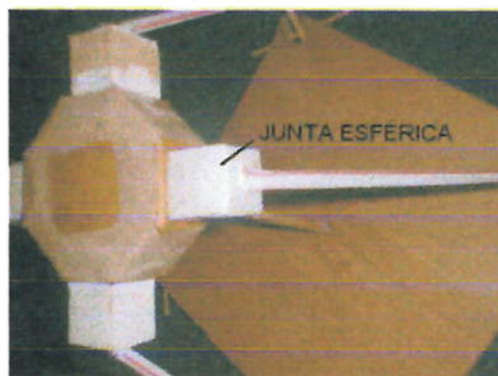


Figura 4.1.2 – Junta Esférica(Maquete)

Através da construção dessa maquete foi possível ter uma noção física e prática e pudemos perceber algumas vantagens e algumas restrições que o nosso protótipo teria se fosse construído nos mesmos moldes.

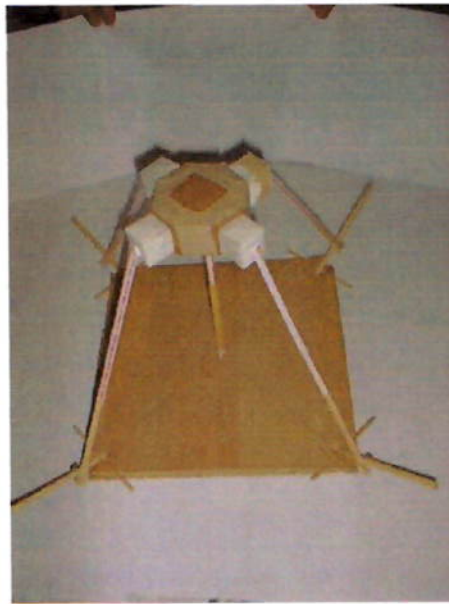


Figura 4.1.3 – Maquete

A) Vantagens

Uma das vantagens percebidas através da construção da maquete foi a simetria que o ela possuía. Essa simetria é bastante útil na obtenção dos movimentos do terminal e no seu controle. Outro ponto forte é que a maquete possuía 3 Graus de Mobilidade e cada grau nos 3 eixos de coordenada X, Y e Z, podendo o protótipo estar “andando” dentro de um ambiente virtual de 3 dimensões.

B) Restrições Encontradas

A maquete, devido a sua construção possui um certo Grau de Mobilidade, mas esse Grau de Mobilidade não é igual em todos os pontos. Se o terminal estiver na sua extremidade mais baixa, o Grau de Mobilidade nos eixos X e Y são bastante grandes, porém se estiver na extremidade mais alta, esse Grau de Mobilidade é bastante restringida. Esse problema

pode ser solucionado utilizando o comprimento correto para as barras laterais melhorando a performance da mobilidade.

3.1.2. Construção

Tendo em vista a maquete construída na primeira etapa do Projeto de Formatura no 1º Semestre, foi projetado e construído um protótipo com as mesmas características, ou seja, com o mesmo Grau de Mobilidade no Mecanismo.

Os Graus de Mobilidade do protótipo finais são: 02 (duas) rotações no eixo X e Y, e 01 (uma) translação no eixo Z, como descrito no esquema abaixo.

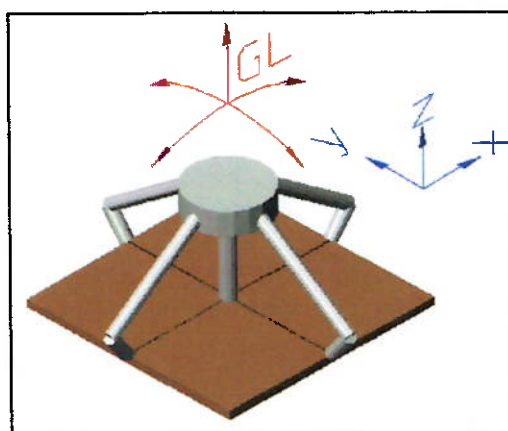


Figura 4.1.4 – Grau de Liberdade do Protótipo

Esses graus de mobilidade foram obtidos através do conjunto de grau de liberdade de cada cadeia ativa e da cadeia passiva do protótipo. Cada uma das quatro cadeias ativas possuem 5 graus de liberdade e a cadeia passiva (barra central do protótipo) possui 3 graus de liberdade. Esses graus de liberdade foram obtidos utilizando o conjunto de juntas abaixo.



Tipo de Junta	Forma Física	Utilizado no Protótipo	Graus de Liberdade
Revolução			1
Prismática			1
Esférica			3

Tabela 4.1.1 – Tipo de Juntas Utilizadas

As juntas de revolução mostrada na figura acima foram obtidas pela interligação das barras laterais menor e maior por um parafuso que os conecta. Através de duas roscas impedimos que os parafusos se soltem durante a utilização do protótipo. A junta prismática é simplesmente o manche do joystick que tem um furo no seu centro e é atravessado por uma barra de alumínio (barra centra superior). Dessa maneira o manche pode fazer o movimento de subida e descida de forma livre. A junta esférica é um terminal esférico de 3 GL que interliga o manche (terminal) das barras laterais.

As 4 cadeias cinemáticas ativas contam com duas juntas de revolução, uma na interligação da barra lateral menor com o motor CC e outra na ligação entre o barra lateral maior com a menor e uma junta esférica na extremidade do braço maior que liga a cadeia ativa com o

manche. A cadeia passiva, conta com uma junta prismática que permite o movimento de translação no eixo Z e a junta universal que fornece as rotações no eixo X e Y na manche.

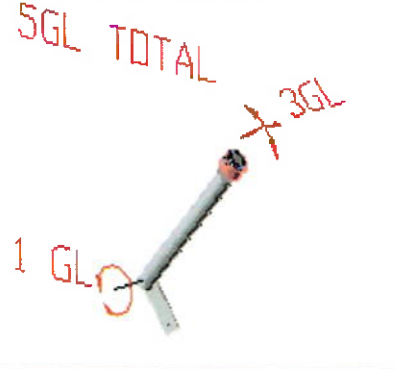
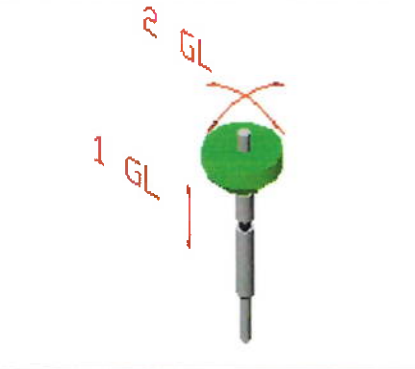
Cadeia Ativa 5 Graus de Liberdade	Cadeia Passiva 3 Graus de Liberdade
	

Tabela 4.1.2 – Cadeias Cinemáticas do Protótipo

O mecanismo, visando a melhor simetria, é constituído de 4 articulações ativas que interliga a base com o terminal. Uma articulação passiva interliga o centro do terminal com o centro da base. Com esta construção, dependendo dos movimentos, todas as articulações realizam o mesmo movimento, por exemplo, movimento de subida ou descida do terminal. Ou ainda, somente duas das articulações realiza movimentos, facilitando o feedback dos motores.

3.1.3. Partes do Mecanismo

O joystick pode ser dividida em três partes: Terminal, Cadeias Cinemáticas e Base.

3.1.3.1. Terminal

O Terminal é constituída de um cilindro usinado de Nylon, Juntas Esféricas de aço, e bronze que realiza a interligação com as hastes e manche de plástico e molas.



Figura 4.1.5 – Rótula Radial (Junta Esférica)



Figura 4.1.6 – Rótula Radial

O cilindro de Nylon possui 4 furos de diâmetro 22 mm com angulação de 90° de uma para outra para acoplar as juntas esféricas e para realizar a sua fixação foram colocados pequenos parafusos. Os furos para acoplar as juntas esféricas foram feitos com 45° em relação ao plano do chão (Figura 5) para não restringir o grau de liberdade que a junta oferece. Caso a o furo fosse realizado paralelo ao plano do chão, não seria possível estar aproveitando em sua totalidade o grau de liberdade da junta, como mostra a figura comparativa abaixo.

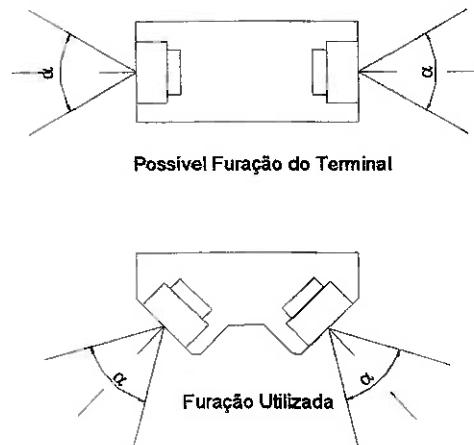


Figura 4.1.7 – Comparação da Angulação

Na etapa do projeto do protótipo, não estávamos prevendo a instalação de um manche, porém percebendo que os motores utilizados ofereciam bastante resistência, e para melhorar a controlabilidade, acoplamos um manche na parte superior da barra central superior do protótipo. O resultado obtido foi uma melhor facilidade no controle do protótipo.

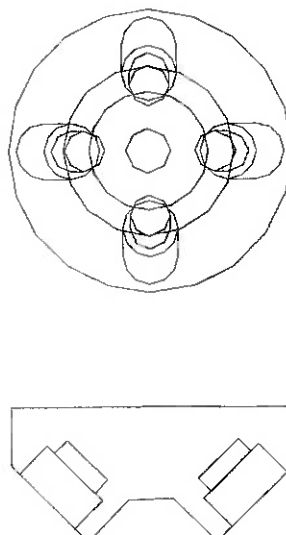


Figura 4.1.8 – Terminal de Nylon com vista superior

3.1.3.2. Cadeias Cinemáticas

A Cadeia Cinemática é constituída de Barras de Alumínios formando as cadeias cinemáticas ativas e a cadeia passiva do protótipo.

As cadeias ativas são constituídas de duas barras laterais, uma maior e uma menor que são interligadas por um parafuso e duas roscas para impedir que se soltem. As Cadeias Cinemáticas ativas são interligadas ao Terminal através de um eixo que atravessa a junta esférica. Esse eixo tem um dos lados no mesmo diâmetro que o furo da junta esférica e outro lado rosqueado como mostra a Figura. Após acoplar o eixo à junta fixamos com cola. Em uma das extremidades da barra lateral maior usinamos uma rosca interna, podendo dessa forma fazer a fixação do eixo e por fim a interligação do Terminal com a Cadeia Cinemática ativa.

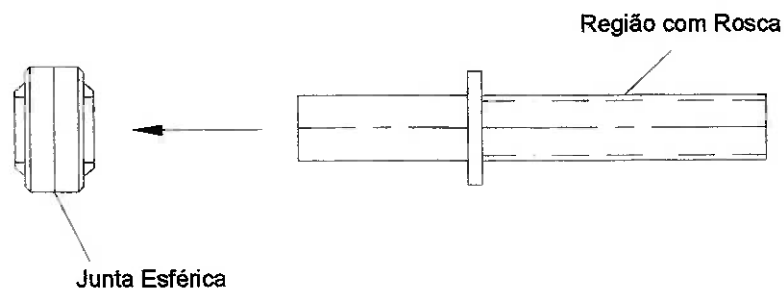


Figura 4.1.9 – Interligação da Cadeia Cinemática

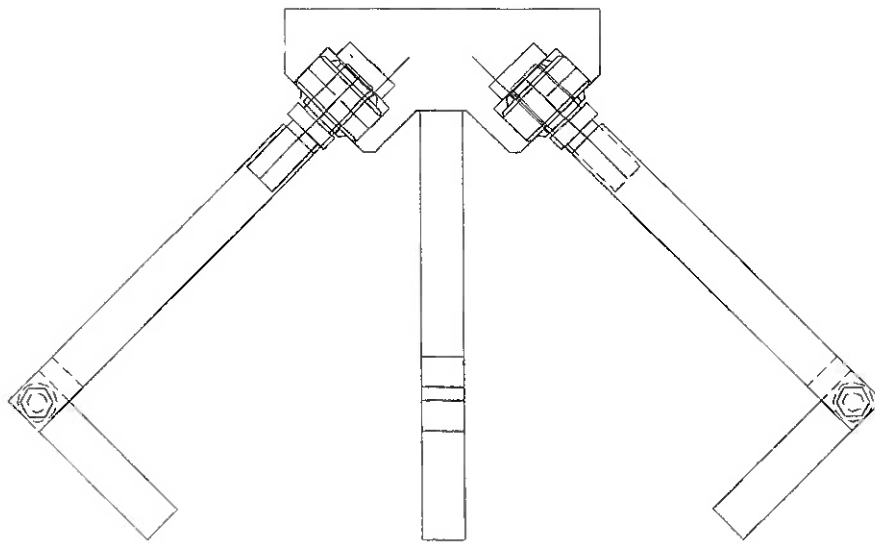


Figura 4.1.10 – Desenho de corte do conjunto Terminal/Cadeia Cinemática

Fazem parte da Cadeia Cinemática passiva a barra central superior, barra central inferior, borracha que interliga as duas barras e o suporte da barra central. O cilindro de Nylon possui um furo no seu centro que permite a passagem pelo furo da barra central superior possibilitando realizar o movimento de subida e descida. O movimento de subida do Cilindro de Nylon é limitado por um parafuso que está fixado na extremidade mais alta da barra superior e o movimento de descida é limitado por um rebaixo que a barra central superior possui.

A barra central superior e a barra central inferior são interligados por uma borracha cilíndrica que faz o papel de uma junta universal. As barras centrais possuem nas suas extremidades furos do mesmo diâmetro que a borracha cilíndrica podendo acoplar dessa maneira as barras. Além disso, existem pequenos parafusos que melhoram a fixação da borracha com as

barras. A barra central inferior é fixado a base pelo suporte da barra central que por sua vez é fixado a base por dois parafusos.

As Cadeias Cinemáticas Ativas são aquelas que provém os Graus de Liberdade ao protótipo enquanto que as Cadeias Cinemáticas são aquelas limitam o grau de liberdade, mas melhoram a sua controlabilidade.

3.1.3.3. Base

A Base é onde os motores (acionamento) e potenciômetros estão fixados através de uma cantoneira de alumínio.

A fixação dos motores é um dos itens mais importantes na construção do protótipo, pois a sua colocação errada pode afetar a simetria e o controle do protótipo. Caso o motor esteja fora do local projetado, esta folga poderá rotacionar o cilindro de Nylon do Terminal, impedindo a sua correta movimentação.

Tivemos também um cuidado maior na fixação do potenciômetro, para que o seu eixo esteja concêntrico ao eixo do motor, não afetando a sua performance.

Através dos eixos dos motores interligamos a barra lateral menor que possui um furo do mesmo diâmetro que o eixo do motor e com o parafuso os fixamos.

3.1.4. Material Utilizado

Para a fabricação e montagem do protótipo definimos como requisitos básicos dos materiais, a facilidade na usinagem e a sua leveza, pois o próprio peso do protótipo poderia atrapalhar a performance dos motores. Além desses requisitos os materiais deverão ser rígidos o suficiente para agüentar os esforços aplicados tanto pelo usuário como pelos motores.

Portanto, apesar do custo relativamente alto maior parte do protótipo foi construída em alumínio, material de baixa densidade, de fácil usinagem, que agüenta esforços relativamente altos sem se deformarem e possuem uma aparência boa depois de realizada usinagem. Além do alumínio foi utilizado cilindro de Nylon, de bronze, de borracha e chapa de madeiras como mostra a figura a seguir.

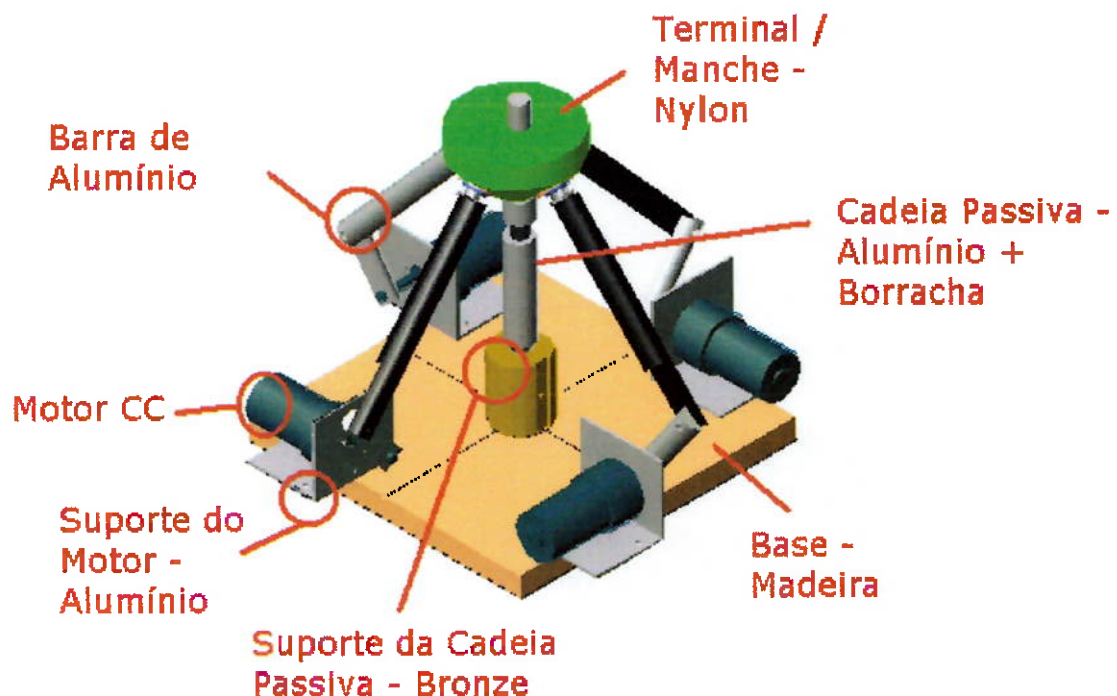


Figura 4.1.11 – Desenho Isométrico do Protótipo

Foram utilizados parafusos de diferentes tipos que serviram como elementos de fixação e como conexão entre as barras laterais de alumínio para constituir juntas de revolução.

Na extremidade do protótipo foi acrescentado um manche de plástico e molas para facilitar o seu controle.

3.1.5. Lista de Desenhos

A Tabela a seguir mostra a lista de desenhos de fabricação dos protótipos que estão em anexo a este relatório.

	Numeração	Nome	Quantidade
1	01 - F	BARRA CENTRAL INFERIOR	1
2	02 - F	BARRA CENTRAL SUPERIOR	1
3	03 - F	BARRA LATERAL MAIOR	4
4	04 - F	BARRA LATERAL MENOR	4
5	05 - F	MANCHE (TERMINAL)	1
6	06 - F	SUORTE BARRA CENTRAL	1
7	07 - F	SUORTE DO MOTOR	4
8	08 - F	SUORTE DO POTENCIÔMETRO	2
9	09 - F	TERMINAL ESFÉRICO	4

Tabela 4.1.3 – Numeração dos Desenhos de Fabricação

3.2. Controle

Somadas as experiências passadas (primeiro semestre de 2003) no desenvolvimento do controle do Joystick Sensitivo, foi elaborado um sistema que integra 2 portas paralelas de um PC, podendo-se compreender todo o sistema de acordo com o diagrama de blocos que está representado logo abaixo:

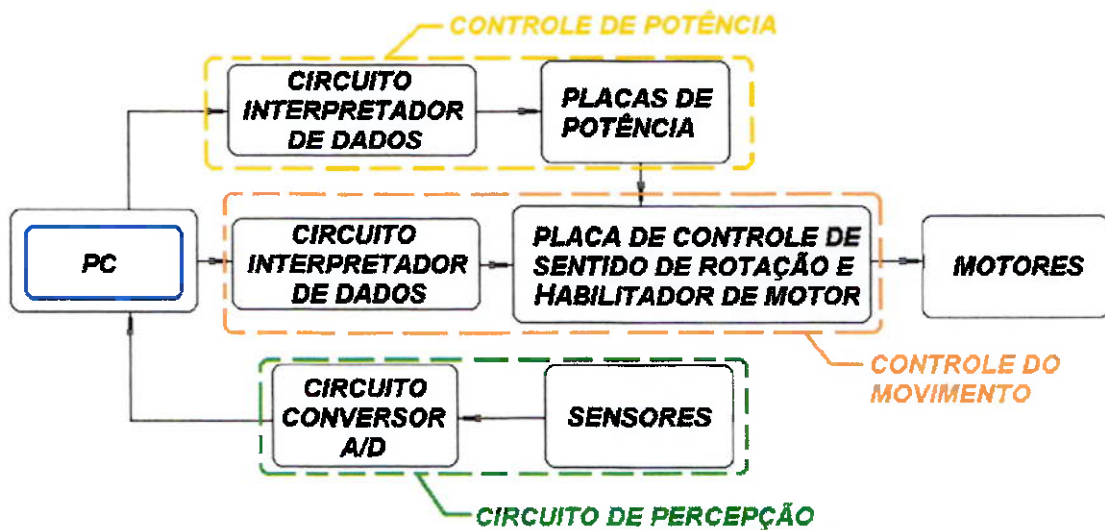


Figura 4.2.1 – Diagrama de Blocos do sistema de controle

Além das duas portas paralelas, são necessárias Uma fonte de alimentação de 5Vcc (600mA), uma de 5,8V (1A) e outra de 24Vcc (2A) para o seu perfeito funcionamento.

A seguir serão apresentados com mais detalhes cada um dos itens representados no diagrama de blocos acima que formam todo o sistema de controle do joystick sensitivo.

3.2.1. Controle de Potência

Para o controle de potência foram utilizadas 5 placas de circuito. A primeira placa de circuito, ou **circuito interpretador de dados**, é um circuito combinatório, que se utilizando do conceito de portas lógicas, definiu-se o processamento de um certo número de bits recebidos do PC para o controle das outras 4 placas que são circuitos de potência. Considerando que cada barra ou perna do dispositivo mecânico (num total de 4) precisa ter um motor, foi necessário definir o número de bits capaz de controlar todos os níveis de acionamento para cada motor. Definiu-se 5 níveis de potência para

cada motor, ou seja, 5 níveis para cada placa de potência (num total de 4 placas), o que totaliza 20 níveis de controle. As placas de potência nada mais são do que circuitos chaveados transistorizados que variam a resistência elétrica, permitindo ou não uma carga maior de corrente no motor, controlando sua intensidade de torque. O esquema elétrico de uma placa de potência está ilustrado a seguir:

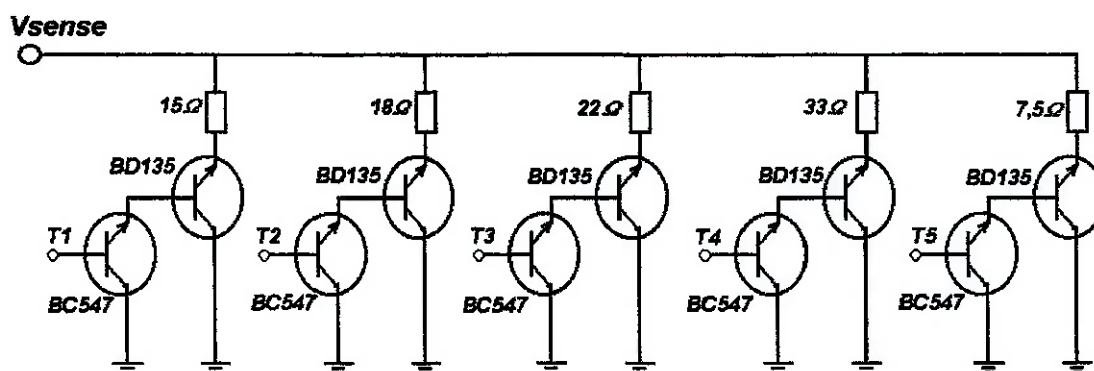


Figura 4.2.2 – Esquema elétrico da placa de potência

A tensão V_{sense} é fornecida por um outro estágio do diagrama de blocos (Placa de controle e sentido de rotação e habilitação de motor implementado com o CI L-298, como será apresentado mais adiante).

Os níveis de correntes definidos nas placas de potência que acionarão o motor estão listados abaixo:

Resistência (Ω)	corrente(A)
7,5	0,56
15	0,48
18	0,45
22	0,42
33	0,35

Tabela 4.2.1 – Resistências que serão utilizadas em cada placa de potência para controle de corrente.

Para saber quantos números de bits serão necessários ao projeto para o controle, inicialmente foi necessário estabelecer quando cada motor

irá atuar. Os diagramas abaixo podem exemplificar a ação conjunta dos motores, sendo que para os movimentos na direção X e Y (na horizontal) só poderão ser utilizados apenas 2 motores para oferecer resistência ao usuário. Para o movimento vertical os quatros motores trabalharão conjuntamente, sendo que dois dos motores terão seus eixos girados num sentido e os outros dois no outro.

Para o projeto também, definimos que não se pode haver composição de dois movimentos de direções diferentes simultaneamente, ou seja, quando se decidir movimentar na direção "Y", deve-se manter o manche somente nesta direção, da mesma forma para a direção "Z" e "X". É necessário o movimento em seqüência para o correto funcionamento do dispositivo sensitivo.

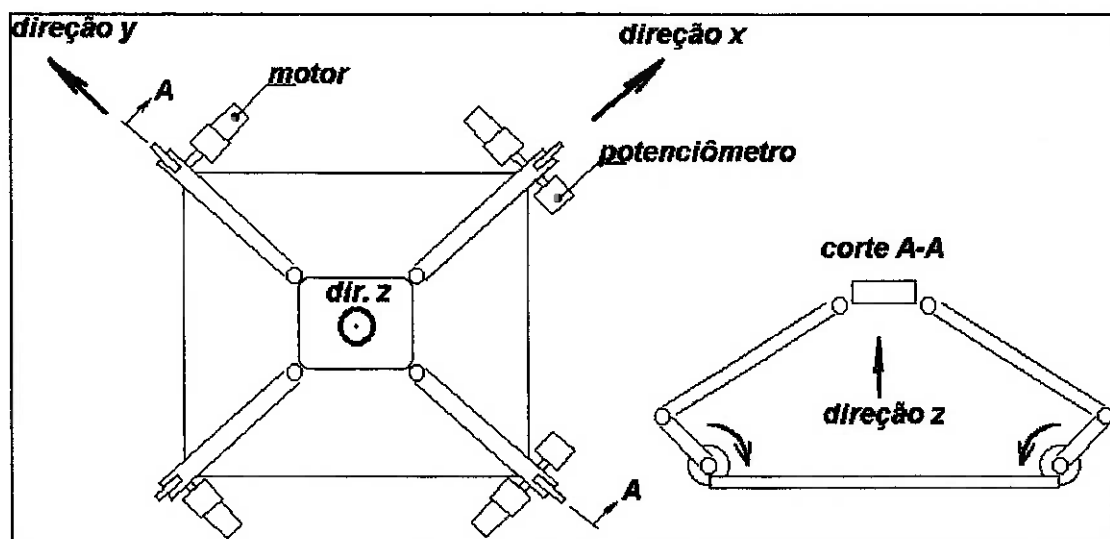


Figura 4.2.3 – Direções de movimento do terminal e rotações de sentido contrário quando o terminal é elevado ou abaixado.

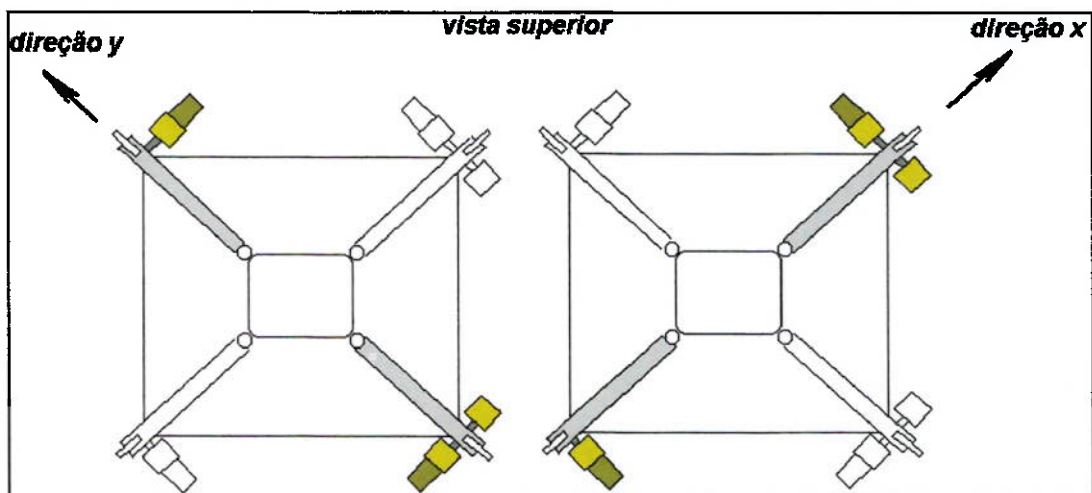


Figura 4.2.4 – Direções de movimentos e motores que funcionarão para fornecer o torque reativo em cada direção.

Os motores coloridos acima indicam que serão os acionados durante o movimento de retro-ação na direção indicada na figura. Pode-se observar também que há dois potenciômetros posicionados na figura 4.2.4. Estes serão utilizados para medir a posição angular através da variação da tensão em seus terminais quando houver giro em seu eixo. A razão de se usar apenas dois potenciômetros está no fato de que eles podem captar o movimento, tanto na direção X como na direção Y ou movimentos verticais em relação a sua base, quando há elevação ou abaixamento do terminal ou manche.

Numa das direções horizontais (x ou y) pode-se ter 10 níveis de forças de reações diferentes, combinando-se a ação dos dois motores dispostos na direção. No caso particular da direção vertical, pode-se ter somente 5 níveis de forças diferentes devido ao fato de que é necessário se ter os quatros motores funcionando ao mesmo tempo com torques de módulos iguais, para não haver ascensão ou queda do manche de forma

assimétrica. Por isso serão necessários apenas o reconhecimento de apenas 5 posições ou níveis.

A soma de todos os níveis para reconhecimento no caso da definição da potência a ser aplicada será de:

Nº.direção X + Nº.direção Y + Nº.direção Z = 25 posições diferentes

$$10 + 10 + 5 = 25$$

Para isso serão necessários 5 bits ou cinco “fios” de comando ($2^5 = 32 > 25$). As demais posições de comando não serão utilizadas, entretanto, estes poderão ser úteis caso necessitemos de uma posição ou nível para acionamento de algum outro equipamento ou acessório. No entanto, caso se necessite utilizar estas posições de comando, há o trabalho de se modelar novas funções ou o remodelamento das funções antigas o que implica no desenvolvimento de uma nova placa de circuito.

A tabela da verdade abaixo ilustra as posições de acionamento mencionado acima. Os valores b0, b1, b2, b3 e b4 representam os bits que formam uma palavra, posição ou nível, sendo b0 o bit menos significativo e b4 o bit mais significativo. Já os termos referentes a TAx, TBx, TCx e TDx são funções lógicas que dizem respeito ao acionamento de uma das 4 placas com chaveamento transistorizado que fazem controle da corrente em cada motor. As letras A, B, C e D foram utilizadas neste trabalho para simbolizar as 4 placas de potência, citando a título de exemplo que todas as

funções nomeadas como TA_x, são funções ligadas a placa de potência “A”.

Veja a tabela 4.2.2 abaixo:

	b4	b3	b2	b1	b0	TA1	TA2	TA3	TA4	TA5	TB1	TB2	TB3	TB4	TB5	TC1	TC2	TC3	TC4	TC5	TD1	TD2	TD3	TD4	TD5
0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
1	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0
2	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0
3	0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0
4	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	1	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
11	0	1	0	1	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
12	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	1	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
15	0	1	1	1	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
16	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
17	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
18	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
19	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
20	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
21	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
22	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0
23	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
24	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
31	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Tabela 4.2.2 – Tabela da verdade para o controle do circuito de potência

Na tabela 4.2.2, TA₁, TA₂, ...TD₅ são funções de circuitos combinatórios que acionam os transistores das placas de potência (A,B,C e D) saturando-os ou cortando-os. Cada transistor é acionado quando a palavra de 5 bits (b₀, b₁, ..., b₄) forma a combinação correta para a respectiva função do transistor. Cada transistor acionado significa um nível de corrente diferente (determinado por um resistor no coletor do transistor) que permitirá o controle de torque, uma vez que os motores elétricos podem ter seus torques controlados pelas correntes que os excitam.

Tendo-se a tabela acima, simplificaram-se as funções TA1, TA2...TB1, TB2, ...TD5 pelo método do mapeamento. A título de exemplo será demonstrado o método para as funções lógicas da placa de potência A.

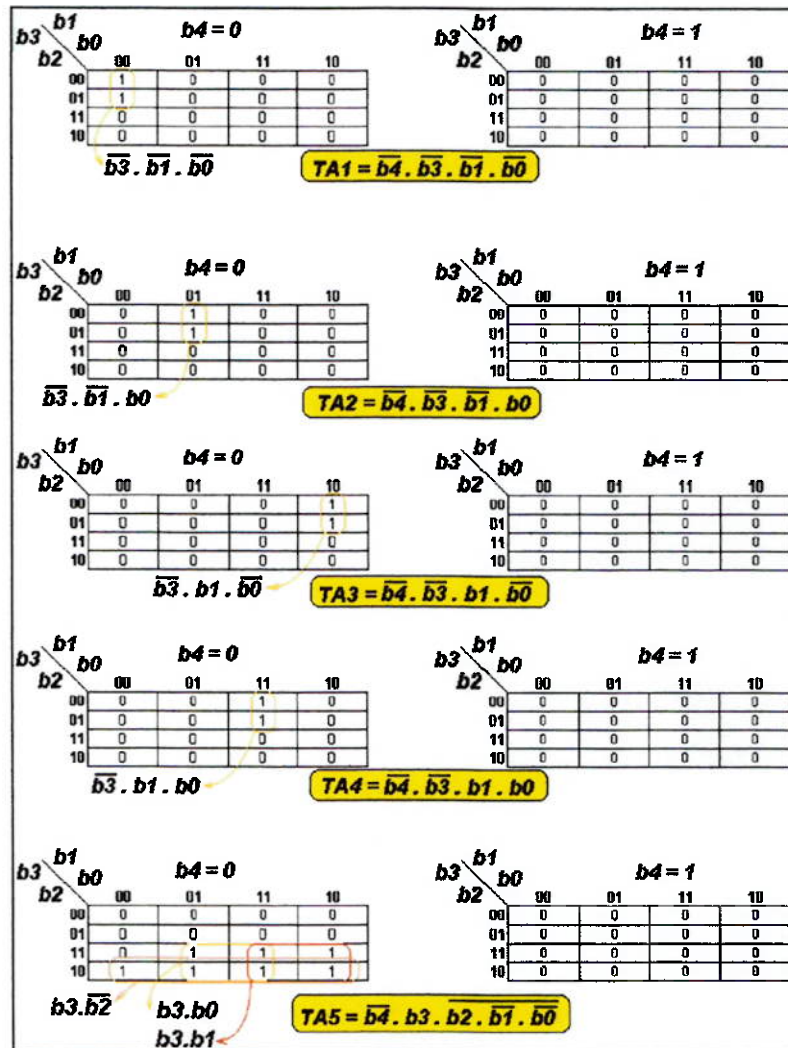


Figura 4.2.5 – Mapeamento para simplificação das funções lógicas

Obs: A função lógica TA5 foi obtida através do teorema de Morgan da seguinte forma:

$$TA5 = \overline{b_4} \cdot (\overline{b_3} \cdot \overline{b_2} + b_3 \cdot b_0 + b_3 \cdot b_1) = b_4 \cdot b_3 \cdot (b_2 + b_0 + b_1)$$

$$TA5 = \overline{\overline{\overline{b_4} \cdot b_3 \cdot \overline{\overline{b_2} \cdot b_1 \cdot b_0}}} \Rightarrow TA5 = \overline{\overline{\overline{b_4} \cdot b_3 \cdot b_2 \cdot b_1 \cdot b_0}}$$

Através das funções obtidas, pode-se determinar o diagrama lógico utilizando-se as portas NAND, através do CI 74LS00 (data sheet anexado no final deste trabalho). O resultado final pode ser visto no diagrama lógico esquematizado abaixo:

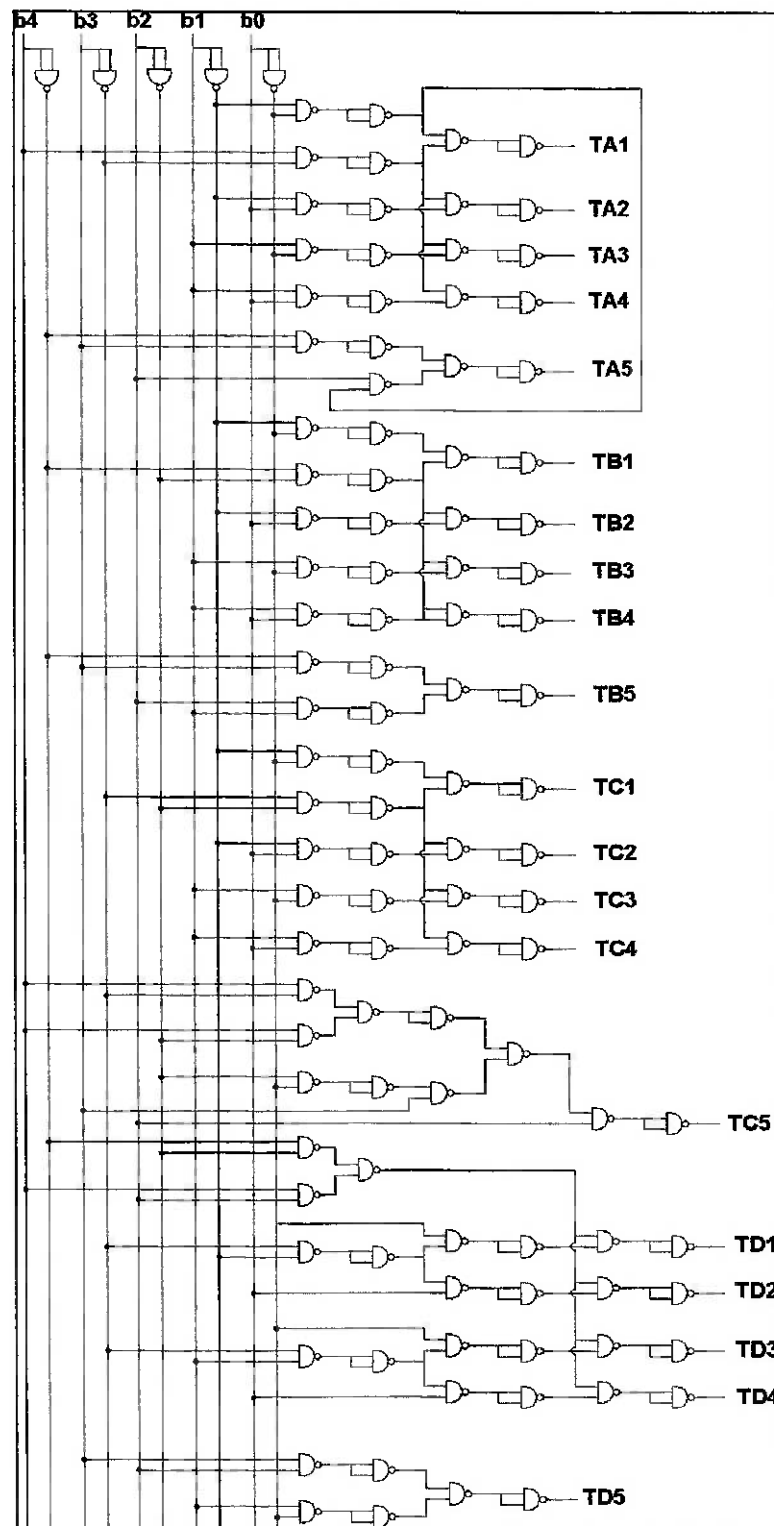


Figura 4.2.6 – Diagrama lógico das funções que acionarão as placas de potência

Os níveis lógicos com níveis de forças associados para as placas de potência estão ilustrados na tabela a seguir. Esta tabela foi utilizada para a interface entre o hardware e o software.

Nº. DECOMPL.	b4	b3	b2	b1	b0	TA1	TA2	TA3	TA4	TA5	TB1	TB2	TB3	TB4	TB5	TC1	TC2	TC3	TC4	TC5	TD1	TD2	TD3	TD4	TD5	NÍVEL DE FORÇA
0	0	0	0	0	0	1					1					1					1					TODOS - NÍVEL 5
1	0	0	0	0	1		1					1					1					1				TODOS - NÍVEL 1
2	0	0	0	1	0			1					1					1						1		TODOS - NÍVEL 2
3	0	0	0	1	1				1					1					1						1	TODOS - NÍVEL 3
4	0	0	1	0	0	1																				A - NÍVEL 5
5	0	0	1	0	1		1																			A - NÍVEL 1
6	0	0	1	1	0			1																		A - NÍVEL 2
7	0	0	1	1	1				1																	A - NÍVEL 3
8	0	1	0	0	0					1	1															A - NÍVEL 4; BNÍVEL 5
9	0	1	0	0	1					1		1														A - NÍVEL 4; BNÍVEL 1
10	0	1	0	1	0					1			1													A - NÍVEL 4; BNÍVEL 2
11	0	1	0	1	1					1				1												A - NÍVEL 4; BNÍVEL 3
12	0	1	1	0	0																					
13	0	1	1	0	1					1																A - NÍVEL 4
14	0	1	1	1	0					1					1						1				1	TODOS - NÍVEL 4
15	0	1	1	1	1					1					1											A - NÍVEL 4; BNÍVEL 4
16	1	0	0	0	0											1										C - NÍVEL 5
17	1	0	0	0	1												1									C - NÍVEL 1
18	1	0	0	1	0													1								C - NÍVEL 2
19	1	0	0	1	1														1							C - NÍVEL 3
20	1	0	1	0	0																1	1				C - NÍVEL 4; D - NÍVEL 5
21	1	0	1	0	1																1		1			C - NÍVEL 4; D - NÍVEL 1
22	1	0	1	1	0																1			1		C - NÍVEL 4; D - NÍVEL 2
23	1	0	1	1	1																1				1	C - NÍVEL 4; D - NÍVEL 3
24	1	1	0	0	0																					
25	1	1	0	0	1																					
26	1	1	0	1	0																					
27	1	1	0	1	1																					
28	1	1	1	0	0																					
29	1	1	1	0	1																					
30	1	1	1	1	0																1				1	C - NÍVEL 1; D - NÍVEL 4
31	1	1	1	1	1																1					C - NÍVEL 4

LEGENDA DOS TERMOS DA TABELA	
NÍVEL 1	SUPER FRACO
NÍVEL 2	FRACO
NÍVEL 3	MÉDIO
NÍVEL 4	FORTE
NÍVEL 5	SUPER FORTE

Tabela 4.2.3 – Tabela da verdade associada aos níveis de torque dos motores CC

O nível 1 corresponde ao nível de corrente mais baixo e o nível 5, o nível de corrente mais alto. Fazendo-se a conexão com a tabela 1, teremos a seguinte tabela:

Resistência (Ω)	Corrente (A)	NÍVEL DE TORQUE	DESCRIÇÃO DO NÍVEL
7.5	0.56	NÍVEL 5	SUPER FORTE
15	0.48	NÍVEL 4	FORTE
18	0.45	NÍVEL 3	MÉDIO
22	0.42	NÍVEL 2	FRACO
33	0.35	NÍVEL 1	SUPER FRACO

Tabela 4.2.4 – Valores de resistências e correntes para os níveis de torque

3.2.2. Placa de inversão de sentido e habilitador de movimento

A inversão do sentido de rotação do motor, necessário para o projeto, foi desenvolvida utilizando-se o CI L-298, como já foi mencionado anteriormente, utilizando-se uma configuração que permite o controle de um motor de corrente contínua. Este CI permite diversas formas de controle de motores uma vez que em seu interior está contida uma ponte H, própria para o controle de um motor de passo. Através desta ponte H é que se consegue a inversão de sentido do motor CC, manipulando-se os bits de controle.

Veja o esquema abaixo:

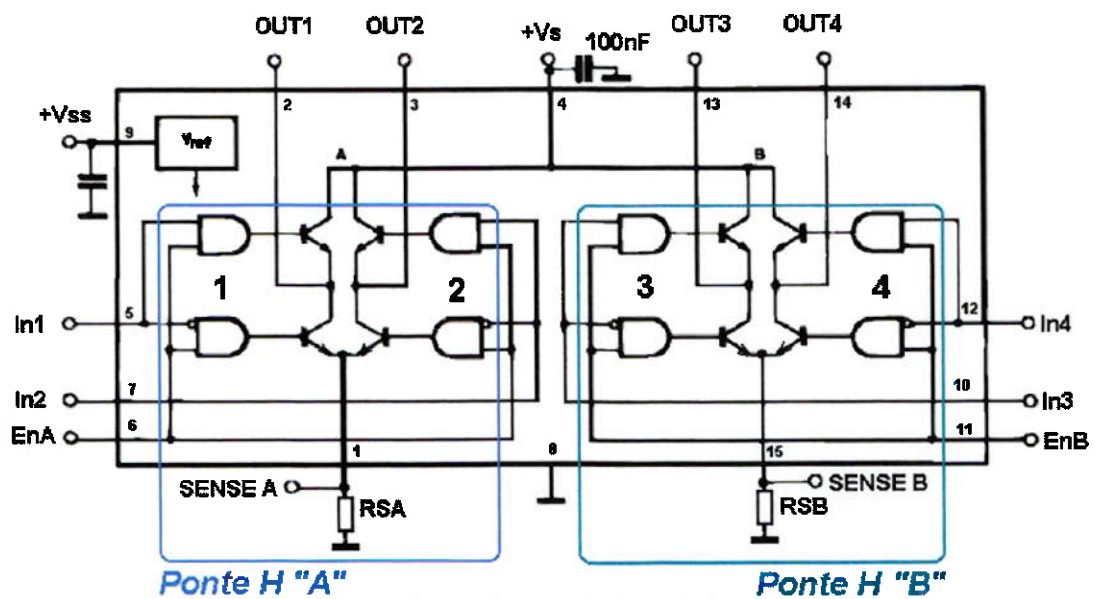


Figura 4.2.7 – Diagrama de blocos do L298

Cada CI do L298 já vem com 2 pontes H, sendo que uma ponte H é o suficiente para controlar o sentido de rotação de um motor. Para o projeto, então, foram utilizados 2 CI's L298 para o controle de 4 motores.

As saídas "OUT1", "OUT2", "OUT3" e "OUT4" do CI L298 são as saídas para as conexões dos motores CC. O seu controle é feito através dos pinos 5, 7 e 8 (respectivamente In1, In2 e EnA) para a ponte H "A", e os pinos 12, 18 e 11 (respectivamente In4, In3 e EnB) para a ponte H "B". Os pinos do In1 e In2, ou In4 e In3 são bits que controlam as pontes H, invertendo o sentido de rotação. Já os pinos EnA e EnB habilitam o funcionamento das pontes H "A" e "B", permitindo que o eixo esteja em seu neutro quando desabilitados, permitindo ao usuário o manuseio do Joystick. Veja a tabela a seguir para o controle de um motor CC, através de uma ponte H:

Inputs		Function
$V_{en} = H$	C = H ; D = L	Forward
	C = L ; D = H	Reverse
	C = D	Fast Motor Stop
$V_{en} = L$	C = X ; D = X	Free Running Motor Stop

L = Low H = High X = Don'tcare

Tabela 4.2.5 – Dados de entrada para o controle de uma ponte H

V_{en} nesta tabela equivale ao EnA ou EnB, e os inputs C e D equivalem aos pinos In1 e In2 ou In4 e In3 respectivamente. Pode-se perceber que quando C está "high", D esta "low" para um sentido de rotação e para se obter o sentido contrário, basta inverter C e D. Para o nosso projeto não se utilizou a configuração C=D. Observe que C e D citados neste

projeto não se utilizou a configuração C=D. Observe que C e D citados neste trecho não possui nenhuma conexão com as letras C e D citados anteriormente ou nas letras que serão citados mais adiante.

Os resistores RSA e RSB são os mesmos das placas de potência, ou seja, através do chaveamento transistorizado, variam-se as resistências de RSA e RSB, aumentando o fluxo de corrente e conseqüentemente o torque. Os capacitores de 100nF servem para cortar ruídos de baixa freqüência, dando maior estabilidade ao controle. V_{SS} é a tensão de alimentação do CI (5V) e V_{SS} é a tensão que cairá sobre o enrolamento do motor, sendo seu valor máximo, dado de acordo com a tabela ilustrada abaixo:

L298

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{EN}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	- Non Repetitive ($t = 100\mu s$)	3	A
	- Repetitive (80% on - 20% off, $t_{ON} = 10ms$)	2.5	A
	-DC Operation	2	A
V_{SENSE}	Sensing Voltage	-1 to 2.3	V
P_{TOT}	Total Power Dissipation ($T_{CASE} = 75^\circ C$)	25	W
T_{OP}	Junction Operating Temperature	-25 to 130	°C
T_{STG}, T_J	Storage and Junction Temperature	-40 to 150	°C

Tabela 4.2.6 – Limites máximos dos parâmetros do L298

A configuração de uma ponte H (no exemplo, a ponte H "B") para controlar os motores contínuos CC está esquematizada a seguir:

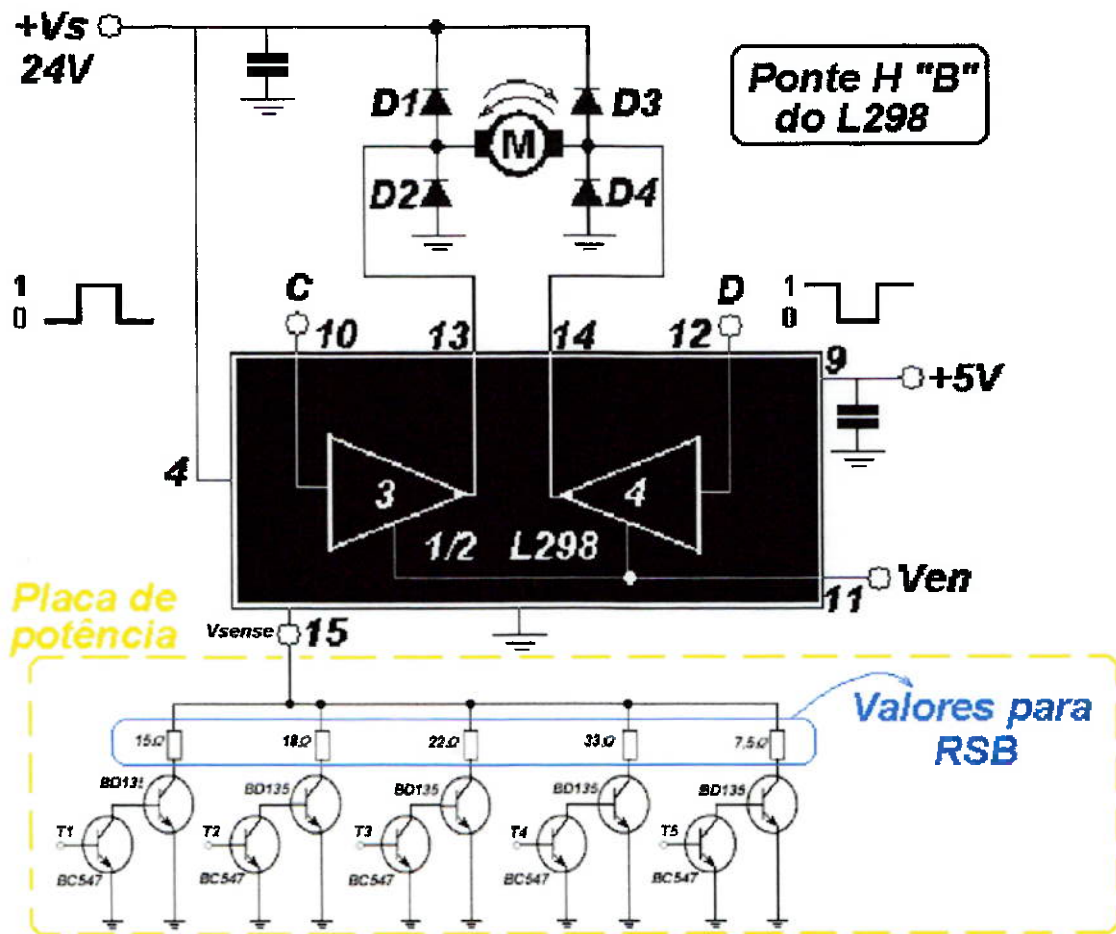


Figura 4.2.8 – esquema de conexão do L298 para o controle do Motor CC

3.2.2.1. Circuito interpretador de dados que controla a inversão de sentido e habilitação de funcionamento

O L298 é controlado através de um circuito interpretador ou combinatório de 3 bits (b0, b1 e b2) e seu funcionamento foi definido da seguinte forma:

sentido de rotação	b7	b6	b5	A	EnA	B	EnB	C	EnC	D	EnD
neutro	0	0	0	X	0	X	0	X	0	X	0
X-	0	0	1	1	1	1	1	X	0	X	0
X+	0	1	0	0	1	0	1	X	0	X	0
Y-	0	1	1	X	0	X	0	1	1	1	1
Y+	1	0	0	X	0	X	0	0	1	0	1
SOBE	1	0	1	1	1	0	1	1	1	0	1
DESCE	1	1	0	0	1	1	1	0	1	1	1
DON'T CARE	1	1	1	X	X	X	X	X	X	X	X

Motores A e B estão na abscissa

Motores C e D estão na ordenada

Tabela 4.2.7 – Tabela da verdade para o controle do L298

As funções A e B fazem parte do primeiro CI L298 e as funções C e D do segundo CI L298. Tomando-se como exemplo a função A, quando esta está em “high”, definiu-se que os pinos In1 estará em “high” e In2 estará em “low” para um sentido e quando se tiver a função A igual a “low”, ocorre a inversão dos pinos In1 e In2 com relação ao que já foi citado logo acima. Isto é equivalente para as funções B, C e D da tabela 4.2.7

Como já foi mencionado, observe que quando os Enables EnA, EnB, EnC e EnD estão em “0”, o usuário poderá movimentar livremente o joystick, livre de qualquer torque.

Tendo-se definido o funcionamento das funções A, EnA, B, EnB, C, EnC, D e EnD, resta agora minimizar as suas funções, semelhantemente ao que já foi feito para o circuito interpretador de dados ou combinatório da placa de potência. Os mapeamentos e suas simplificações estão ilustrados a seguir:

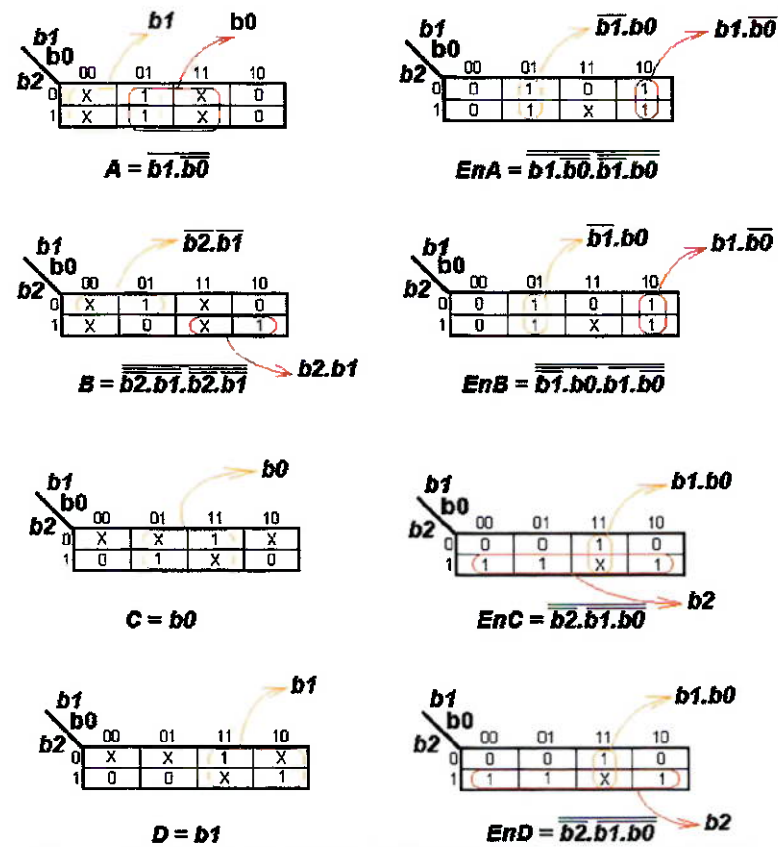


Figura 4.2.9 – Mapeamento para simplificar as funções de controle do L298

Através das funções simplificadas, montou-se o diagrama lógico utilizando-se as portas NAND do CI 74LS00, como se pode ver abaixo:

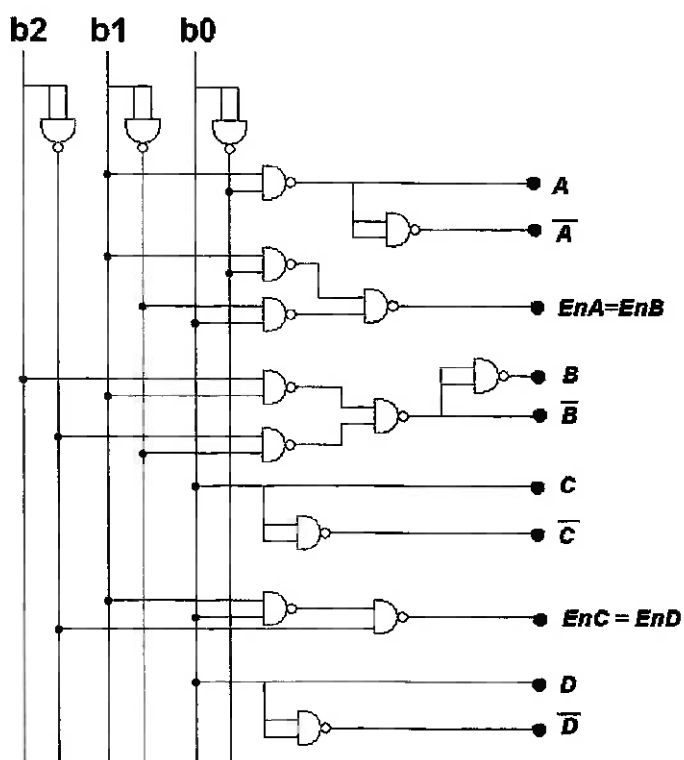


Figura 4.2.10 – Diagrama lógico das funções de controle do L298

3.2.3. Circuito de percepção

A técnica utilizada para a percepção consiste de um conversor A/D seguido de um circuito combinatório digital. Foram utilizados dois potenciômetros como sensores, e este foi discretizado em 5 estágios da seguinte forma:

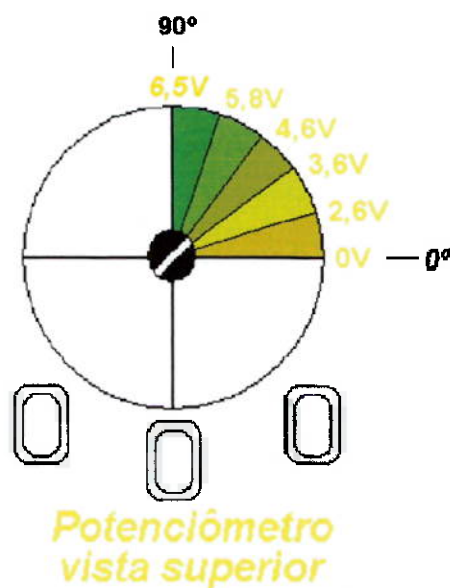


Figura 4.2.11 – discretização do potenciômetro em 5 níveis

O estágio que compreende entre as tensões 2,6V-3,6V, 3,6V-4,6V e 4,6V-5,8V fazem parte do estágio interpretado como “neutro”. Foram tomados três estágios na fase “neutra” para facilitar a detecção, uma vez que cada estágio possui apenas 22,5°, além do fato de que a parte central é mais difícil de ser reconhecida (bordas são mais fáceis de serem percebidas pelo fato de existirem limitações mecânicas do movimento).

Os estágios que compreendem entre 0-2,6V e 5,8V-6,5V (bordas) foram definidos como superiores e inferiores. Uma explicação mais detalhada será abordada mais adiante, no capítulo que descreve o software do projeto.

Cada potenciômetro foi ligado ao conversor A/D, que nada mais é do que uma placa que converte a tensão em quatro bits de entrada. Veja esquema do circuito abaixo:

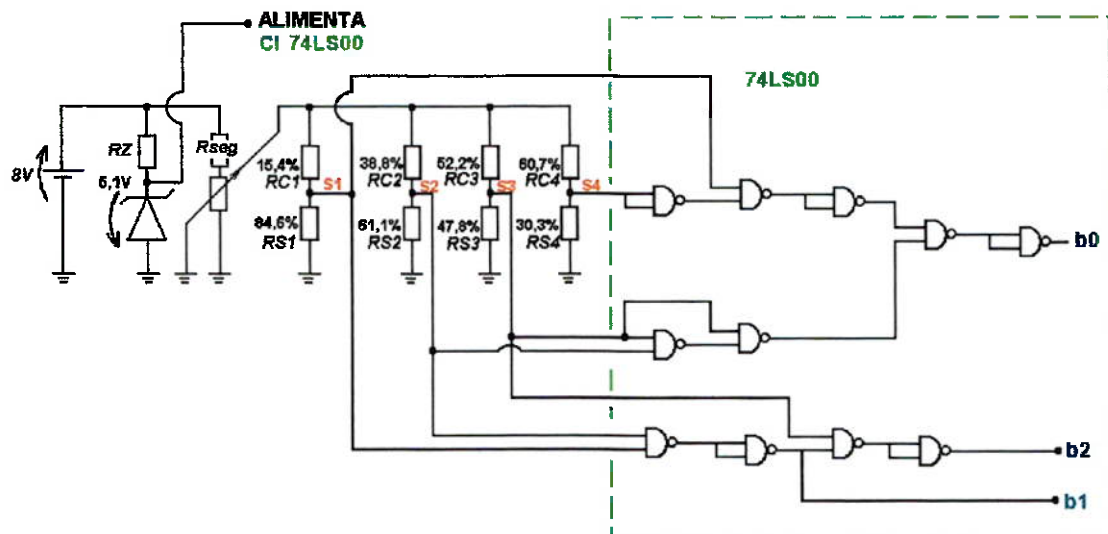


Figura 4.2.12 – circuito A/D do sensor

A discretização da tensão é feita através de quatro ramos ou estágios, sendo estes divisores de tensão implementados com resistores. Os divisores de tensão estão dimensionados de forma que apenas um estágio de divisão seja acionado quando se atinge um nível de tensão. À medida que vai se elevando a tensão através do potenciômetro, cada estágio (começando por S1) vai sendo excitado com nível lógico “1”.

Para o projeto, inicialmente, defiram-se as faixas de trabalho considerando-se (através de data-sheet) que o nível de tensão mínimo para o reconhecimento de nível lógico “1” do CI 74LS00 está em torno de 2,2V. A tabela da verdade que interpreta os dados de entrada (S1, S2, S3 e S4) está ilustrada abaixo:

	S4	S3	S2	S1	B2	B1	B0	n°. em Hexa
0	0	0	0	0	0	0	0	104
1	0	0	0	1	0	0	1	120
2	0	0	1	1	0	1	1	152
3	0	1	1	1	1	1	1	216
4	1	1	1	1	1	1	0	200

dados de entrada
dados de saída

tabela 4.2.8 – tabela da verdade que relaciona os bits de saída do circuito do sensor com sinais utilizados no PC

Quando o potenciômetro atingir 2,6V, deve-se alcançar 2,2V no pino de entrada S1 (tensão mínima de excitação do 74LS00). Para tanto, dimensionou-se os resistores da seguinte forma:

$$RS1 = VS1/V_{pot} = 2,2/2,6 = 0,846 \text{ ou } 84,6\%$$

$$RC1 = 1 - RS1 = 0,154 \text{ ou } 15,4\% \text{ (Resistor complementar do divisor de tensão)}$$

Ou seja, no ramo de S1, o resistor que garantirá a tensão de 2,2V em S1 quando o potenciômetro fornecer a tensão de 2,6V deverá possuir 84,6% da resistência do ramo. Observe que os outros ramos de tensão deve possuir uma divisão de tensão de tal forma que não seja excitados pela tensão 2,6V do potenciômetro.

Aumentando-se a tensão através do potenciômetro, até se atingir 3,6V, o pino S2 também deverá ser reconhecido como nível lógico "1" ou "high". Para isso, procede-se da mesma maneira para se dimensionar o divisor de tensão, ou seja:

$$RS2 = VS2/V_{pot} = 2,2/3,6 = 0,611 \text{ ou } 61,1\% \quad (V_{pot} = \text{Tensão fornecida pelo potenciômetro})$$

$$RC2 = 1 - RS2 = 0,389 \text{ ou } 38,9\%$$

Observe que quando S2 for excitado, S1 permanecerá excitado. Para S3 e S4, procede-se da mesma forma:

$$RS3 = VS3/V_{pot} = 2,2/4,6 = 0,478 \text{ ou } 47,8\%$$

$$RC3 = 1 - RS3 = 0,522 \text{ ou } 52,2\%$$

$$RS4 = VS4/V_{pot} = 2,2/5,8 = 0,38 \text{ ou } 38\%$$

$$RC4 = 1 - RS4 = 0,62 \text{ ou } 62\%$$

A somatória dos dois resistores por ramo deverá ser de $10K\Omega$. Quando-se atingir a tensão máxima de $6,5V$, a corrente máxima transitando pela malha será de $I_{m\acute{a}x} = 6,5V / 10K\Omega = 0,65mA$, seguro para a operação do circuito.

3.2.3.1. Dimensionamento do R_z (resistor de proteção do Diodo Zener)

O diodo utilizado foi o 1N4733A de $5,1V$, e segundo o data-sheet da FAIRCHILD, a corrente recomendada para o seu funcionamento é de $49mA$. Para isso, dimensionou-se o RZ da seguinte forma:

$$R_z = (E - V_z) / I_z = (8V - 5,1V) / 49mA = 59,18 \cong 60\Omega$$

$$PR_z = R_z \cdot I_z^2 = 144mW \Rightarrow$$

$$\therefore R_z = 60\Omega \quad \frac{1}{4}W$$

Este diodo tem apenas a função de alimentar o CI 74LS00 com tensão constante de $5,1V$. Observe que outros diodos zeners utilizados para alimentar o CI 74LS00 de outros circuitos foram dimensionados de forma semelhante.

3.2.3.2. Dimensionamento do Rseg (resistor de segurnaça)

$$R_{seg} = V_{seg}/(V_{seg} + V_{pot}) \cdot R_{pot} = 1,5 / (1,5+6,5) \cdot 4,7k = 881,25 \cong 880 \Omega$$

$$P_{rseg} = R_{seg} \cdot I_{máx}^2 = 880 \cdot (8/880)^2 = 72,72mW \Rightarrow$$

$$\therefore R_z = 880\Omega \quad \frac{1}{4} W$$

3.2.3.3. Circuito interpretador ou combinatório do circuito A/D

Este circuito (que pode ser visualizado acima na fig 4.2.12) foi implementado semelhantemente aos outros já apresentados neste trabalho simplificando-se as funções contidas na tabela da verdade apresentada na tabela 4.2.8. Entretanto para este caso, o processo é inverso aos que já foram feitos anteriormente, ou seja, o que antes era o sinal de entrada tornou-se a saída e o que era saída tornou-se entrada. A idéia para esta inversão é a de diminuir o número de bits necessário para a interface entre o computador e o circuito controlador, diminuindo de 4 para 3 pinos o número necessário para a comunicação com a porta paralela.

Para a simplificação, utilizou-se novamente o teorema de Morgan sobre as funções:

$$\begin{aligned} b_0 &= \overline{S_4} \cdot \overline{S_3} \cdot \overline{S_2} \cdot S_1 + \overline{S_4} \cdot \overline{S_3} \cdot S_2 \cdot S_1 + \overline{S_4} \cdot S_3 \cdot \overline{S_2} \cdot S_1] \\ &= \overline{S_4} \cdot S_1 (\overline{S_3} \cdot \overline{S_2} + \overline{S_3} \cdot S_2 + S_3 \cdot \overline{S_2}) \\ &= \overline{S_4} \cdot S_1 (\overline{S_3} (\overline{S_2} + S_2) + S_3 \cdot \overline{S_2}) \\ &= \overline{S_4} \cdot S_1 (\overline{S_3} + S_3 \cdot \overline{S_2}) \Rightarrow \end{aligned}$$

$$b_0 = \overline{S_4} \cdot S_1 \cdot S_3 \cdot S_3 \cdot S_2$$

$$\begin{aligned} b_1 &= S_3 \cdot S_2 \cdot S_1 + S_2 \cdot S_1 \\ &= S_2 \cdot S_1 (S_3 + 1) \Rightarrow \end{aligned}$$

$$b_1 = S_2 \cdot S_1$$

$$b_2 = \overline{S_4} \cdot S_3 \cdot S_2 \cdot S_1 + S_4 \cdot S_3 \cdot S_2 \cdot S_1 = S_3 \cdot S_2 \cdot S_1 (\overline{S_4} + S_4) \Rightarrow$$

$$b_2 = S_3 \cdot S_2 \cdot S_1$$

O resultado obtido acima pode ser novamente visualizado no esquema a seguir:

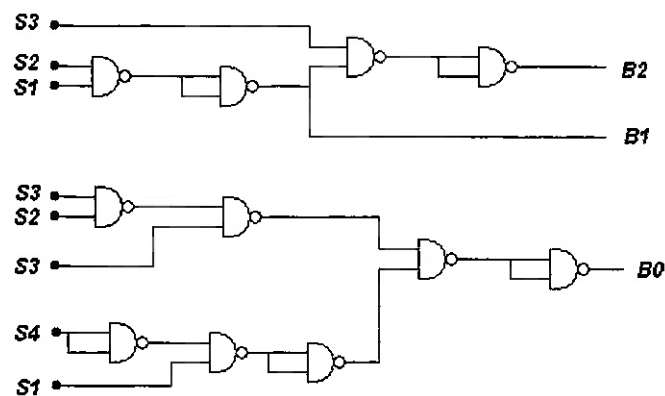


Figura 4.2.13 – Diagrama lógico do conversor A/D

3.2.4. Motor CC

Para a determinação do motor, inicialmente, foi determinado o nível de torque necessário para que o usuário do Joystick pudesse ter a sensação de retro força. Para se ter uma idéia física desta força a ser aplicada pelo motor, estimou-se que 2 kilogramas (2 pacotes de 1kg de açúcar) ou aproximadamente 20 N seria o suficiente para se ter esta sensação. Foi

estimado um braço de 10 cm com relação ao eixo do motor para que os 20 N fossem aplicados ao usuário. O torque necessário então ficou determinado como 2 N.m .

Visando a diminuição dos gastos, um motor CC que foi encontrado na Rua Timbiras, no cruzamento com a Rua Santa Efigênia foi:

DC P M G E A R M O T O R P L A T F O R M 1 . 6 1 . 0 4 6 .



SPECIAL DESIGNS:

- Available with motor I13.023 motor I13.021
- Max. reduction ratio up to 3000 : 1
- Torque capability: max. torque up to 500 mNm
- Output shaft configuration: Shape according to customers specification
- Extended temperature range: -30 °C/+85 °C
- Protection class: IP44
- Add-ons:
Encoder
Lead connections or
Plug connections
Drive dog (e. g. pinion)

Special designs are feasible deviations to the standard version. Realisation with additional investment.

The illustrated gearmotor is representative for the platform.

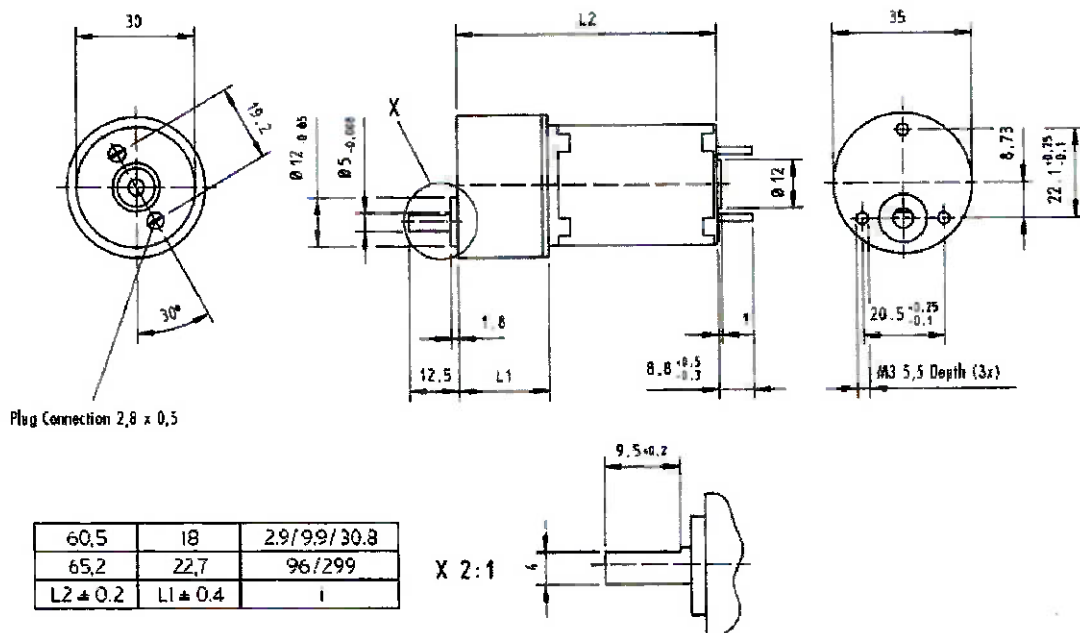


Figura 4.2.14 – Descrição do motor encontrado na Santa Efigênia.

Um motor do modelo de 35 Ohms, 24Vcc foi comprado para testes. Este motor apresenta alto torque, baixa velocidade de rotação (alta redução) ideal para as condições do projeto. Veja abaixo as características técnicas do motor.

Platform 1.61.046.	Rated Torque	Rated Speed	Rated Current	No Load Speed	Terminal Resistance	Gear Ratio	Stages	
	T_n	n_n	I_n	n_0	R	i	-	
	mlm	r.p.m.	mA	r.p.m.	Ω	:	-	
24 V	331	25	1040.0	350	1710.0	35.0	2.9	1
	332	70	335.0	300	500.0	35.0	9.9	2
	333	150	121.0	250	160.0	35.0	30.8	3
	334	300	43.5	195	52.0	35.0	96.0	4
	335	300	15.5	110	16.5	35.0	299.0	5

Tabela 4.2.9 – Características funcionais do motor encontrado.

Apesar do baixo preço, a grande dificuldade em comprar um motor na Rua Timbiras é que o vendedor só conhece dados básicos do motor como tensão de alimentação. O Torque fornecido entre outras informações fica a cargo do comprador de descobrir suas características técnicas. No caso particular deste motor, encontrou-se um catálogo eletrônico do mesmo o que permitiu a análise dos dados, entretanto devido as dificuldades já citadas, deve-se tomar cuidado ao adquirir um motor de segunda mão.

Tendo-se o motor e seus dados técnicos, estimou-se o torque máximo que este pode fornecer e a corrente necessária para o correto funcionamento. Veja os cálculos abaixo:

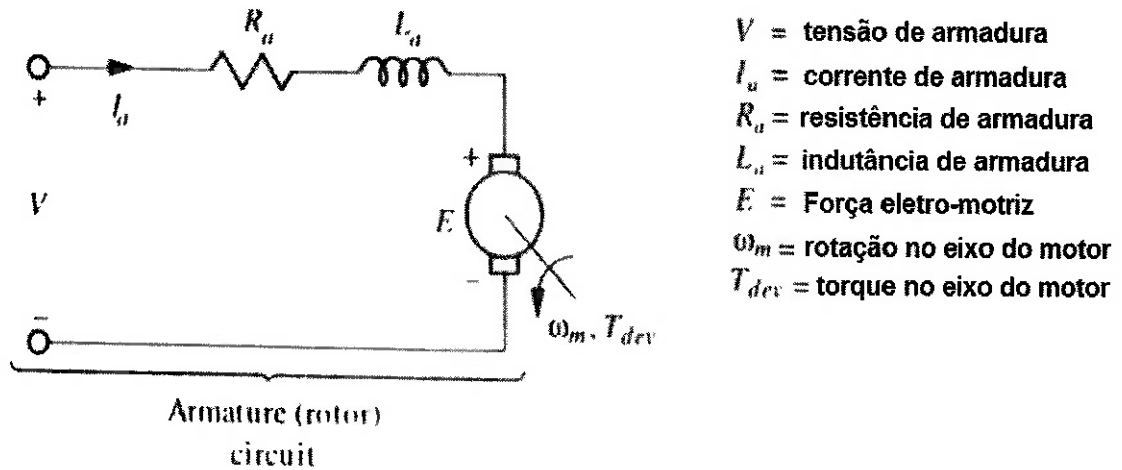


Figura 4.2.15 – Esquema elétrico do motor

Obs.: Os dados do motor foram tomados da tabela 3, onde se encontra grifado de azul.

$$E = K \cdot \Phi \cdot \omega \quad (1)$$

$$V_a = R_a \cdot I_a + E \quad (\text{regime permanente}) \quad (2)$$

$$T = K \cdot \Phi \cdot I_a \quad (3)$$

Da equação (1), tem-se:

$$E = 24 - 195 \cdot e^{-3} \cdot 35 = 17,18V \quad \Rightarrow$$

$$\omega = (96 \cdot 43,5 \cdot 2 \cdot \pi) / 60 = 437,3 \text{ rad/s}$$

$$\text{De (2): } K \cdot \Phi = E / \omega = 39,27 \cdot e^{-3}$$

$$T_{\text{eixo travado}} = K \cdot \Phi \cdot I_{\text{eixo travado}}$$

$$I_{\text{eixo travado}} = (V_a - E) / R_a$$

Como a força contra-eletromotriz é nula, tem-se:

$$I_{\text{eixo travado}} = V_a / R_a = 24 / 35 = 685,71 \text{ mA} \Rightarrow$$

$$T_{\text{eixo travado}} = 39,27 \cdot e^{-3} \cdot 685,71 \cdot e^{-3}$$

$$T_{\text{eixo travado}} = 2,5853\text{Nm} = 26,38 \text{ kgf.cm}$$

Do resultado obtido acima pode-se concluir que se o eixo movesse uma carga com um braço de 10 cm, este poderia levantar uma carga aproximada de 2,6 kg, mostrando-se dentro das expectativas iniciais (2 kg do pacote de açúcar).

O maior problema apresentado pelo motor é a sua mobilidade, ou seja, a facilidade que o motor apresenta ao usuário do haptic device de se mover quando este está movimentando o manche no ponto neutro ou situação de manche livre. Para isso, foi adaptado um cabo maior para que o usuário tenha que exercer menos força, ou seja, com menos força, este aplica o mesmo torque sobre os motores dando maior controlabilidade e conforto no uso. Veja foto do dispositivo abaixo com as alterações:



Figura 4.2.16 – Foto do Joystick com cabo extra para facilitar movimento

3.2.5. Técnica utilizada para a implementação das placas

Os layouts das placas foram desenhados através de um programa CAD (entretanto há softwares especializados na elaboração de placas de circuitos impressos) e impressos em folha de sulfite. O processo para se criar a máscara para proteger a placa de cobre contra o perclorato de ferro foi o da implementação pelo “método do ferro de passar”. Este consiste em tirar uma cópia xerox (escurecida de preferência) do desenho de layout de um circuito impresso e passar o conteúdo do papel xerox para a placa de cobre através de um processo térmico, onde as trilhas do desenho xerox são

derretidas, passando para a placa de cobre e protegendo o no momento da corrosão. Veja as fotos a seguir:



Figura 4.2.17 – Fotos de confecção dos circuitos impressos

Entretanto vale ressaltar que a passagem das trilhas não é perfeita, e se faz necessário o uso de uma caneta de retro-projetor para garantir que não haja o contato entre a trilha e o percloroeto de ferro. A vantagem do método consiste em fazer aparecer os contornos das trilhas na placa de cobre o que permite ao projetista visualizar o desenho de layout sobre a placa de cobre. Assim fica mais fácil a criação da máscara protetora contra a corrosão. A desvantagem está nos cuidados que se deve ter, como por

exemplo, ter que repassar a caneta de retro projetor sobre a placa de cobre (principalmente quando se perceber falha no momento da passagem da tinta xerox do papel para a placa de cobre ou quando se tem uma trilha muito fina) ou na atenção que se deve tomar para não cometer enganos pelo efeito "espelho" que se tem no momento de passar o desenho de circuito impresso, desde o momento da impressão, até o momento do processo térmico.

3.3. Ambiente Virtual

3.3.1. Introdução

O ambiente virtual do Joystick é um programa de computador que simula objetos tridimensionais no monitor e permite que haja interação com o usuário, movendo os objetos quando o usuário mover o joystick em alguma direção.

A interação do programa com o joystick ocorre na conexão dos sinais dos sensores e de acionamento dos motores nas portas paralelas do computador.

3.3.2. Especificações

Antes de começar a programar o ambiente virtual, foi necessário definir quais seriam suas funções principais, chegando-se na tabela abaixo.

Função	Descrição
Exibir gráficos na tela	O programa cria, simula e desenha os objetos tridimensionais do ambiente na tela.
Captura de sinais dos sensores	Através de comandos para leitura de dados na porta paralela, o sistema detecta os sinais enviados ao computador pelos sensores de movimento do Joystick.
Interpretação de dados	Com os dados recebidos na porta paralela do computador, o sistema é capaz de interpretar o movimento desejado pelo usuário.
Movimentação	Após interpretar o movimento desejado, o computador realiza a movimentação no ambiente para obter a visão desejada dos objetos. Caso o movimento seja inibido pela presença de um objeto, o sistema não faz a movimentação do ambiente.
Envio de sinais para acionamento dos motores	No caso do movimento inibido pela presença de um objeto, o sistema envia dados para a porta paralela para que os motores se movam, produzindo o contra-torque necessário para o usuário sentir a barreira do objeto.

Tabela 4.3.1 – Principais funções do Ambiente Virtual

3.3.3. O programa

3.3.3.1. Funções do programa

O programa em Visual Basic contém um formulário principal, com um objeto Timer para fazer o loop do programa; um módulo com as funções do DirectX e um módulo com as funções de acesso à porta paralela. Os nomes das principais funções do sistema são:

- Exibição dos gráficos na tela: Renderiza
- Captura dos sinais: Checaentrada
- Interpretação dos dados: Tratados
- Movimentação: Movimenta, Detectacoliso
- envio de sinais para acionamento dos motores: Enviasaida

3.3.3.2. DirectX

O ambiente virtual do joystick foi desenvolvido utilizando o software Microsoft Visual Basic 6.0.

Para a criação dos gráficos no ambiente, foi utilizado o Microsoft DirectX 8.1 SDK. O DirectX contém um conjunto de APIs (Application Programming Interfaces – Interfaces de programação de aplicativos) que fornecem acesso aos recursos avançados de hardware de alto desempenho, como processadores de aceleração de elementos gráficos 3D e placas de som. Essas APIs controlam funções de baixo nível, incluindo a aceleração de elementos gráficos bidimensionais (2D), o suporte a dispositivos de entrada, como joysticks, teclados e mouses, além do controle da mixagem e da saída de som. Os componentes que foram utilizadas no projeto são (há outros componentes, mas só estão listados os usados):

- DirectDraw – oferece suporte ao acesso extremamente rápido aos recursos de hardware acelerados do adaptador de vídeo de um computador;
- Direct3D – fornece uma interface para as funções de processamento 3D embutidas na maioria das placas de vídeo novas;
- DirectInput – dá suporte a vários dispositivos de entrada, inclusive para a tecnologia force-feedback.

O DirectX 8.1 SDK pode ser livremente baixado da internet, no site da Microsoft (<http://www.microsoft.com/directx>). Este pacote está disponível para a construção de programas em várias linguagens, mas para este projeto só foi necessária a versão para Visual Basic 6.0.

Antes de começar a programação do ambiente no Visual Basic, foi necessário especificar uma referência às bibliotecas do DirectX. Para isto, seguimos os seguintes passos:

- Clicar no menu 'Project';
- Clicar em 'References...', e uma janela irá aparecer;
- Procurar por 'DirectX 8 Type Library for Visual Basic' e clicar na caixa de seleção ao lado deste item;
- Clicar em 'OK'.

Deste modo, o Visual Basic reconhecerá todas as classes, tipos e funções contidas no DirectX.

3.3.3.3. Terminologia:

Existem alguns termos essenciais para que seja possível entender a função dos métodos do directx.

- Classe – código que descreve um Objeto.
- Objeto – Um objeto pode ter variáveis e também funções que podem ser chamadas no código em qualquer lugar. Um objeto é criado a partir de uma classe.

- Instância – Quando um objeto é criado a partir de uma classe, este é chamado como uma instância do objeto. Pode ocorrer de ter várias instâncias de um objeto criado de uma mesma classe.

- Biblioteca – Geralmente os objetos são agrupados em um arquivo, normalmente uma DLL (ynamic Link Librry – Biblioteca de Vínculo Dinâmico). O DirectX é formado por DLLs.

- Ponteiro – Variável que guarda um endereço de memória. No VB, não se usa ponteiros diretamente, mas se for usado um objeto sem ter criado uma nova instância do objeto, basicamente se está usando um ponteiro para outro objeto.

- Buffer – Palavra dada a um pedaço da memória que foi designada para um trabalho, geralmente para guardar temporariamente dados que se movem muito. Há muitos tipos de Buffer no DirectX.

- Backbuffer, Frontbuffer – São usados para guardar gráficos. O único buffer gráfico visível é o FrontBuffer, No DirectX 8, não há a necessidade de se preocupar com isso, só saber o que é (é o que se vê na tela). O back buffer é onde os gráficos vão antes de ir pro FrontBuffer. Os gráficos são desenhados no backbuffer e quando estão prontos, pede-se ao directx para que mova-o para o frontbuffer.

- Z buffer – Pedaço da memória que guarda as posições z dos objetos desenhados em uma superfície.

- Rendering device – Responsável por desenhar os gráficos, pode ser uma placa gráfica ou um software de emulação.
- Vetor – Um valor tridimensional de um ponto, com as coordenadas x , y e z .
- Transformação – uma formula que muda um vetor para outra posição.
- Matriz – As matrizes descrevem qualquer transformação.
- Translação, rotação – Tipos de transformações. Translação é um movimento na direção x , y ou z (ou em duas ou nas três direções) e rotação gira um vetor ao redor de uma origem,
- Origem – Um ponto ou um vetor que é o centro de algo.
- Mundo, Vista e Projeção – A transformação do mundo afeta todos os vetores no mundo, movendo um mundo se move tudo. A transformação da vista faz com que a camera ou olho da cena esteja no lugar certo. A transformação de projeção descreve como as cenas tridimensionais são projetadas na figura bidimensional produzida a partir dele.

3.3.3.4. Porta Paralela

A conexão do computador com a parte de controle do joystick foi feita a partir da porta paralela do computador. Foram utilizadas duas portas paralelas, sendo uma para entrada e saída de dados e outra somente para entrada de dados. Deste modo, em uma porta paralela usamos os pinos 2 a 9 para saída de dados e os pinos 11 a 13 para entrada de dados, e na outra

porta paralela usamos somente os pinos 11 a 13, para entrada de dados dos sensores.

O endereço das portas paralelas para recebimento de sinais são (em hexadecimal): 279 e 379, e o endereço para envio de sinais é 278.

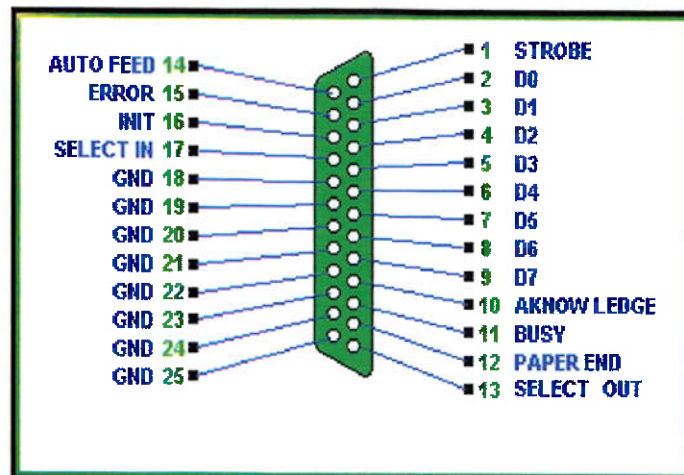


Figura 4.3.1 – Pinagem da porta paralela

4. DISCUSSÃO E CONCLUSÃO

O mecanismo implementado possui mobilidade mais restrita quando o terminal está no seu ponto mais alto, em comparação com o terminal na posição mais baixa. Essa falta de mobilidade pode ser aprimorada com a utilização de barras laterais, que compõe as cadeias cinemáticas, de diferentes comprimentos comparando com as que foram utilizadas no projeto.

Para realizar a interligação entre as barras laterais utilizamos parafusos e duas roscas. O tipo de interligação também poderia ser repensado, pois no emprego de parafusos e roscas, com o decorrer da utilização do joystick os parafusos começam a se soltar necessitando de um novo aperto.

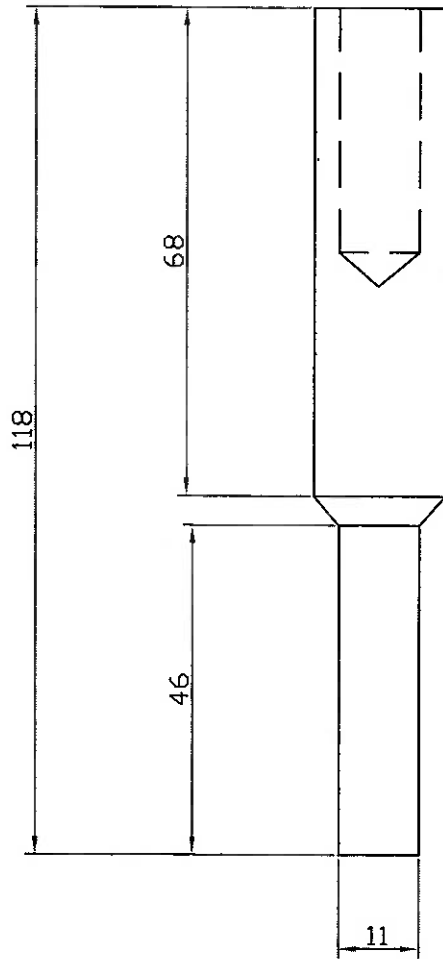
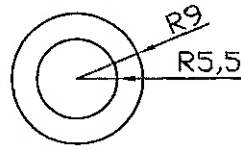
Após a montagem do protótipo, pode-se perceber que, na prática, não é possível detectar todas as posições a todo o momento, devido a um erro de simetria e uma certa imprecisão do mecanismo. A movimentação no eixo z, que necessita da combinação dos sinais dos dois sensores para a sua detecção pode ser falha devido a imprecisão do mecanismo ou folga. Para corrigir este imprevisto, deve-se utilizar mais sensores (quatro) para que o software possa avaliar todos os sinais e após isto, analisar e interpretar os dados recebidos. Uma avaliação da média dos dados recebidos permite uma maior estabilidade e clareza na definição de cada movimento visto pelo software.

Além disso, para melhorar o funcionamento de detecção, cada sensor poderia ser mais sensível, ou seja, poderia haver uma discretização maior do ângulo de movimento. Para que isto seja viável, uma idéia seria a utilização de buffers implementados com o amplificador operacional (CI 741) que permitiria uma detecção mais precisa de cada nível de tensão fornecida pelo potenciômetro. Isto permitiria o desenvolvimento de estudos para a reação de rigidez em diversos tipos de materiais a diferentes velocidades, pois pode-se ter uma idéia de afundamento de acordo com o impacto sobre a superfície virtual.

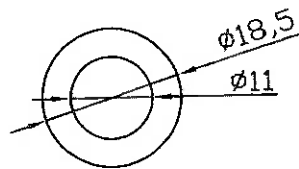
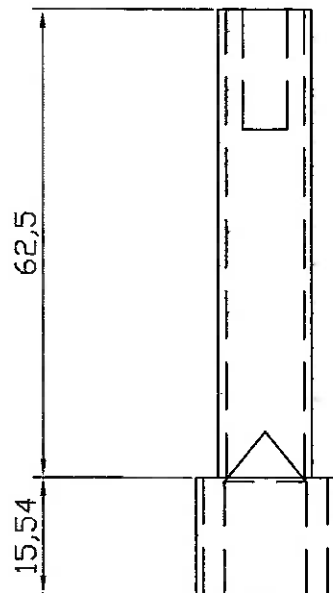
Na placa de potência também poderia se implementar os níveis de torque com mais sensibilidade, ou seja, discretizar mais os níveis de torque que se pode oferecer e colocar os motores funcionando sempre com a mesma intensidade de corrente. Isto para que por problema de assimetria ou imprecisão do mecanismo, não se promova movimentos do mancche de forma indesejada, dando mais equilíbrio ao movimento.

Apesar da facilidade de se utilizar o Visual Basic para programar, e da grande utilidade do DirectX para simular um ambiente tridimensional, este possui muitos detalhes para se configurar, tornando a programação do Ambiente Virtual em um nível mais sofisticado muito mais difícil. Com isso, a proposta inicial de se fazer um ambiente com objetos que se deformassem no momento em que o usuário os “tocasse” foi descartada.

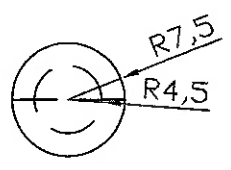
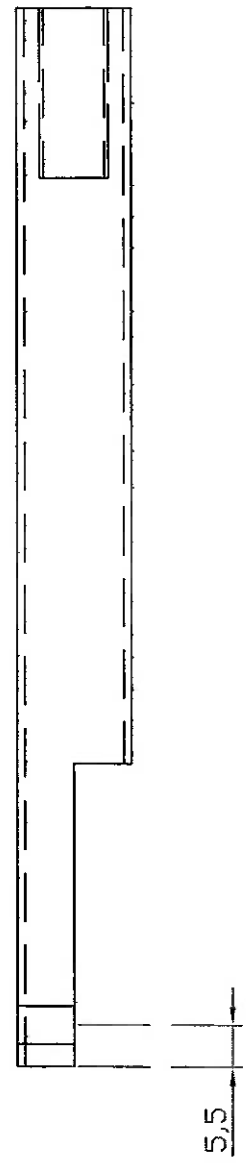
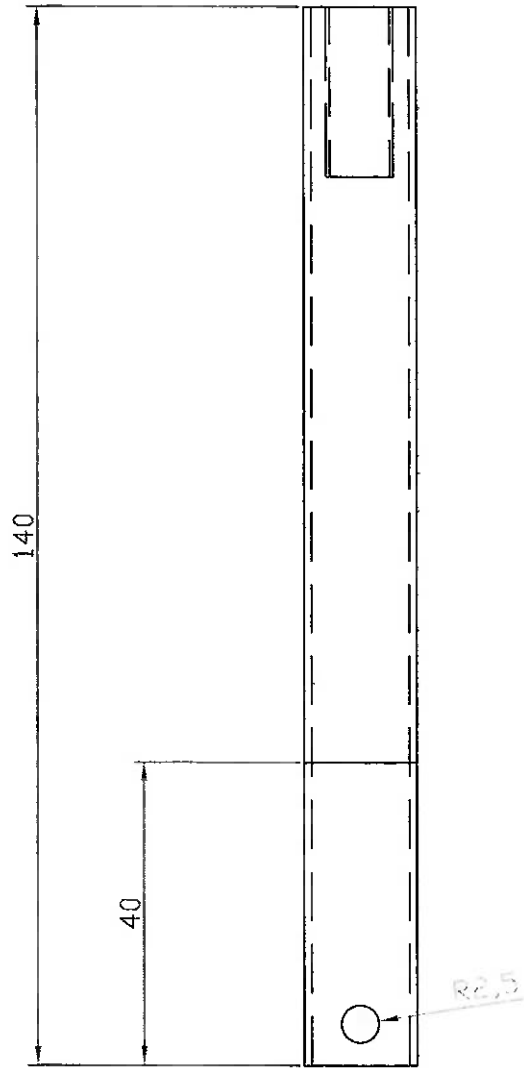
ANEXO A – DESENHOS DE FABRICAÇÃO



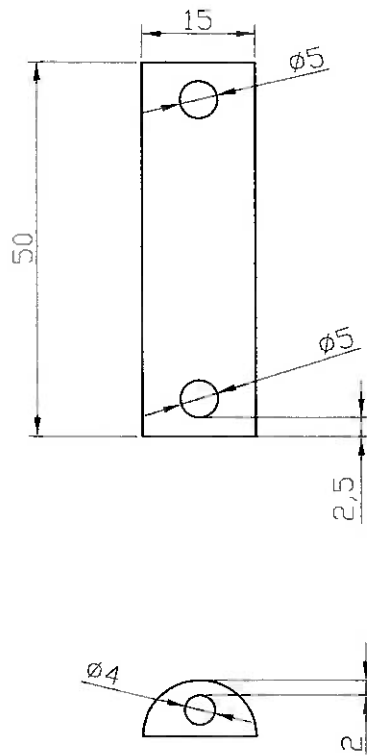
Nº 01 - F		DIMENSÕES INDICADAS EM MILÍMETRO
INTEGRANTES: Colchi Urano Fernando Obara Suzuki Heltor Hideharu Nakamura		ESCOLA POLITÉCNICA DA USP ENGENHARIA MECATRÔNICA
ESCALA 1 : 1	TÍTULO: Barra Central Inferior	TRABALHO DE CONCLUSÃO DE CURSO JOYSTICK SENSITIVO PARA INTERAÇÃO COM AMBIENTE VIRTUAL
	PROFESSOR ORIENTADOR: Pr. Dr. Tarcísio Hess Coelho	



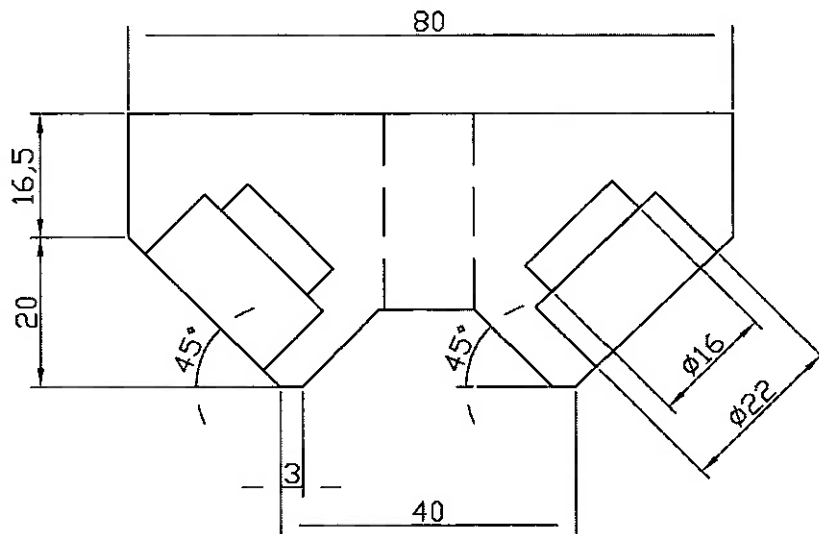
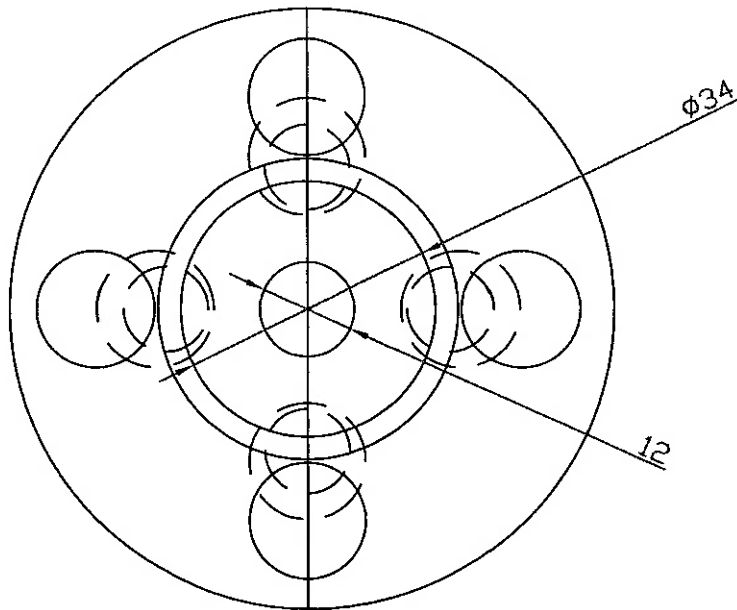
Nº 02 - F		DIMENSÕES INDICADAS EM MILÍMETRO
INTEGRANTES: Coichi Urano Fernando Obara Suzuki Heitor Hideharu Nakamura		ESCOLA POLITÉCNICA DA USP ENGENHARIA MECATRÔNICA
ESCALA	TÍTULO: Barra Central Superior	TRABALHO DE CONCLUSÃO DE CURSO JOYSTICK SENSITIVO PARA INTERAÇÃO COM AMBIENTE VIRTUAL
1 : 1	PROFESSOR ORIENTADOR: Pr. Dr. Tarcísio Hess Coelho	



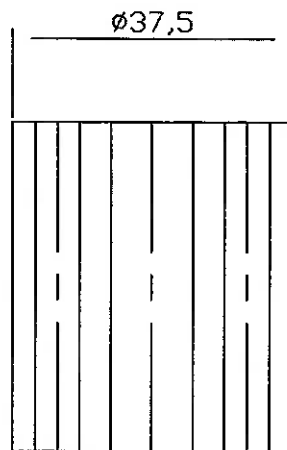
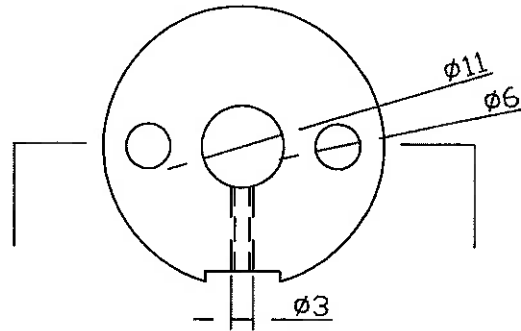
Nº 03 - F		DIMENSÕES INDICADAS EM MILÍMETRO
INTEGRANTES: Coichi Urano Fernando Obara Suzuki Helton Hideharu Nakamura		ESCOLA POLITÉCNICA DA USP ENGENHARIA MECATRÔNICA
ESCALA	TÍTULO: Barra Lateral Maior	TRABALHO DE CONCLUSÃO DE CURSO JOYSTICK SENSITIVO PARA INTERAÇÃO COM AMBIENTE VIRTUAL
	PROFESSOR ORIENTADOR: Pr. Dr. Tarcísio Hess Coelho	
1 : 1		



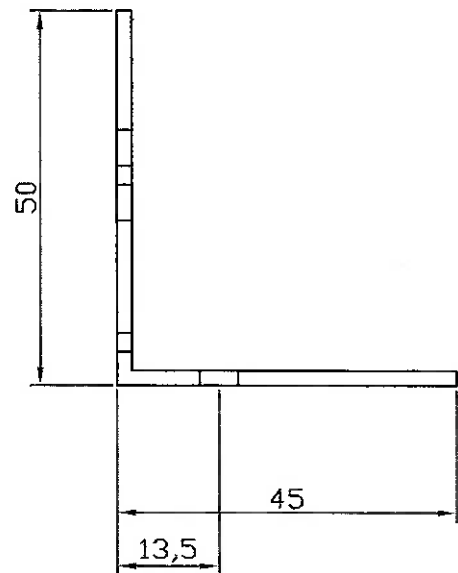
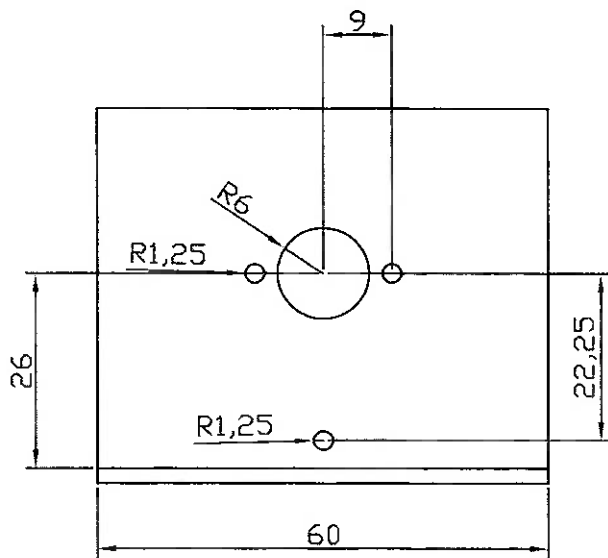
Nº 04 - F		DIMENSÕES INDICADAS EM MILÍMETRO
INTEGRANTES: Colchi Urano Fernando Obara Suzuki Heitor Hideharu Nakamura		ESCOLA POLITÉCNICA DA USP ENGENHARIA MECATRÔNICA
ESCALA	TÍTULO: Barra Lateral Menor	TRABALHO DE CONCLUSÃO DE CURSO JOYSTICK SENSITIVO PARA INTERAÇÃO COM AMBIENTE VIRTUAL
	PROFESSOR ORIENTADOR: Pr. Dr. Tarcísio Hess Coelho	
1 : 1		



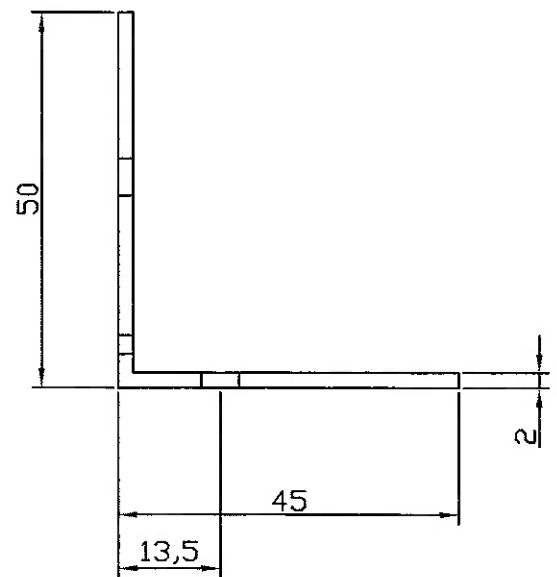
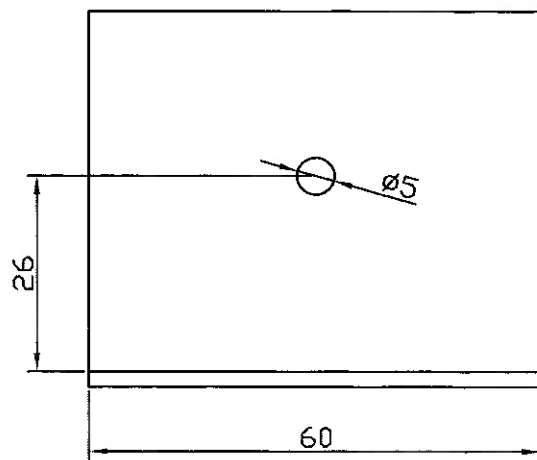
Nº 05 - F		DIMENSÕES INDICADAS EM MILÍMETRO
INTEGRANTES: Coichi Urano Fernando Obara Suzuki Heitor Hideharu Nakamura		ESCOLA POLITÉCNICA DA USP ENGENHARIA MECATRÔNICA
ESCALA	TÍTULO: Manche (Terminal)	TRABALHO DE CONCLUSÃO DE CURSO JOYSTICK SENSITIVO PARA INTERAÇÃO COM AMBIENTE VIRTUAL
	1 : 1	
	PROFESSOR ORIENTADOR: Pr. Dr. Tarcísio Hess Coelho	



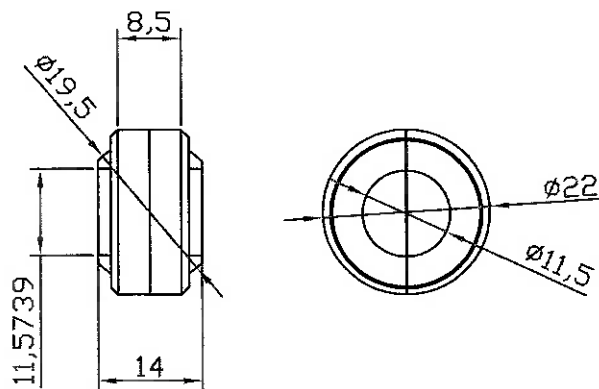
Nº 06 - F		DIMENSÕES INDICADAS EM MILÍMETRO
INTEGRANTES: Colchi Urano Fernando Obara Suzuki Heitor Hideharu Nakamura		ESCOLA POLITÉCNICA DA USP ENGENHARIA MECATRÔNICA
ESCALA	TÍTULO: Suporte Barra Central	TRABALHO DE CONCLUSÃO DE CURSO JOYSTICK SENSITIVO PARA INTERAÇÃO COM AMBIENTE VIRTUAL
1 : 1	PROFESSOR ORIENTADOR: Pr. Dr. Tarcísio Hess Coelho	



Nº 07 - F		DIMENSÕES INDICADAS EM MILÍMETRO
INTEGRANTES: Coichi Urano Fernando Obara Suzuki Heitor Hideharu Nakamura		ESCOLA POLITÉCNICA DA USP ENGENHARIA MECATRÔNICA
ESCALA	TÍTULO: Suporte do Motor	TRABALHO DE CONCLUSÃO DE CURSO JOYSTICK SENSITIVO PARA INTERAÇÃO COM AMBIENTE VIRTUAL
1 : 1	PROFESSOR ORIENTADOR: Pr. Dr. Tarcísio Hess Coelho	



Nº 08 - F		DIMENSÕES INDICADAS EM MILÍMETRO
INTEGRANTES: Colchi Urano Fernando Obara Suzuki Heitor Hideharu Nakamura		ESCOLA POLITÉCNICA DA USP ENGENHARIA MECATRÔNICA
ESCALA	TÍTULO: Suporte do Potenciômetro	TRABALHO DE CONCLUSÃO DE CURSO JOYSTICK SENSITIVO PARA INTERAÇÃO COM AMBIENTE VIRTUAL
1 : 1	PROFESSOR ORIENTADOR: Pr. Dr. Tarcísio Hess Coelho	



Nº 09 - F		DIMENSÕES INDICADAS EM MILÍMETRO
INTEGRANTES: Colchi Urano Fernando Obara Suzuki Heitor Hideharu Nakamura		ESCOLA POLITÉCNICA DA USP ENGENHARIA MECATRÔNICA
ESCALA	TÍTULO: Terminal Esférico	TRABALHO DE CONCLUSÃO DE CURSO JOYSTICK SENSITIVO PARA INTERAÇÃO COM AMBIENTE VIRTUAL
1 : 1	PROFESSOR ORIENTADOR: Pr. Dr. Tarcísio Hess Coelho	

ANEXO B – DATASHEETS DE COMPONENTES

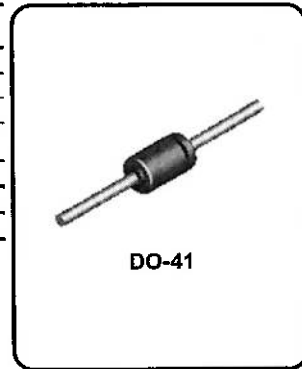
1N4728A - 1N4752A Series One Watt Zeners

Absolute Maximum Ratings*

TA = 25°C unless otherwise noted

Tolerance: A = 5%

Parameter	Value	Units
Storage Temperature Range	-65 to +200	°C
Maximum Junction Operating Temperature	+ 200	°C
Lead Temperature (1/16" from case for 10 seconds)	+ 230	°C
Total Device Dissipation	1.0	W
Derate above 50°C	6.67	mW / °C
Thermal resistance Junction to Lead	53.5	°C/W
Thermal resistance Junction to Ambient	100	°C/W
Surge Power**	10	W



*These ratings are limiting values above which the serviceability of the diode may be impaired.

**Non-recurrent square wave PW= 8.3 ms, TA= 55 degrees C.

NOTES:

- 1) These ratings are based on a maximum junction temperature of 200 degrees C.
- 2) These are steady state limits. The factory should be consulted on applications involving pulsed or low duty cycle operations.

Electrical Characteristics

TA = 25°C unless otherwise noted

Device	V _Z (V)	Z _Z (Ω)	@	I _{ZT} (mA)	Z _{ZK} (Ω)	@	I _{ZK} (mA)	V _R (V)	@	I _R (μA)	I _{SURGE} (mA)	I _{ZM} (mA)
1N4728A	3.3	10		76	400	1.0		1.0		100	1,380	276
1N4729A	3.6	10		69	400	1.0		1.0		100	1,260	252
1N4730A	3.9	9.0		64	400	1.0		1.0		50	1,190	234
1N4731A	4.3	9.0		58	400	1.0		1.0		10	1,070	217
1N4732A	4.7	8.0		53	500	1.0		1.0		10	970	193
1N4733A	5.1	7.0		49	550	1.0		1.0		10	890	178
1N4734A	5.6	5.0		45	600	1.0		2.0		10	810	162
1N4735A	6.2	2.0		41	700	1.0		3.0		10	730	146
1N4736A	6.8	3.5		37	700	1.0		4.0		10	660	133
1N4737A	7.5	4.0		34	700	0.5		5.0		10	605	121
1N4738A	8.2	4.5		31	700	0.5		6.0		10	550	110
1N4739A	9.1	5.0		28	700	0.5		7.0		10	500	100
1N4740A	10	7.0		25	700	0.25		7.6		10	454	91
1N4741A	11	8.0		23	700	0.25		8.4		5.0	414	83
1N4742A	12	9.0		21	700	0.25		9.1		5.0	380	76
1N4743A	13	10		19	700	0.25		9.9		5.0	344	69
1N4744A	15	14		17	700	0.25		11.4		5.0	304	61
1N4745A	16	16		15.5	700	0.25		12.2		5.0	285	57
1N4746A	18	20		14	750	0.25		13.7		5.0	250	50
1N4747A	20	22		12.5	750	0.25		15.2		5.0	225	45
1N4748A	22	23		11.5	750	0.25		16.7		5.0	205	41
1N4749A	24	25		10.5	750	0.25		18.2		5.0	190	38
1N4750A	27	35		9.5	750	0.25		20.6		5.0	170	34
1N4751A	30	40		8.5	1,000	0.25		22.8		5.0	150	30
1N4752A	33	45		7.5	1,000	0.25		25.1		5.0	135	27

V_F Forward Voltage = 1.2 V Maximum @ I_F = 200 mA for all 1N4700 series

TRADEMARKS

The following are registered and unregistered trademarks Fairchild Semiconductor owns or is authorized to use and is not intended to be an exhaustive list of all such trademarks.

ACE ^x ™	ISOPLANAR™
CoolFET™	MICROWIRE™
CROSSVOLT™	POP™
E ² CMOS™	PowerTrench™
FACT™	QS™
FACT Quiet Series™	Quiet Series™
FAST®	SuperSOT™-3
FAST ^r ™	SuperSOT™-6
GTO™	SuperSOT™-8
HiSeC™	TinyLogic™

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

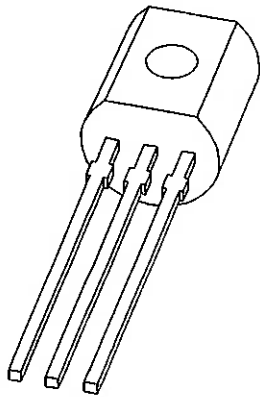
1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, or (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

PRODUCT STATUS DEFINITIONS

Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This datasheet contains preliminary data, and supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
No Identification Needed	Full Production	This datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
Obsolete	Not In Production	This datasheet contains specifications on a product that has been discontinued by Fairchild semiconductor. The datasheet is printed for reference information only.

DATA SHEET



BC546; BC547; BC548 NPN general purpose transistors

Product specification
Supersedes data of September 1994
File under Discrete Semiconductors, SC04

1997 Mar 04

NPN general purpose transistors

BC546; BC547; BC548

FEATURES

- Low current (max. 100 mA)
- Low voltage (max. 65 V).

APPLICATIONS

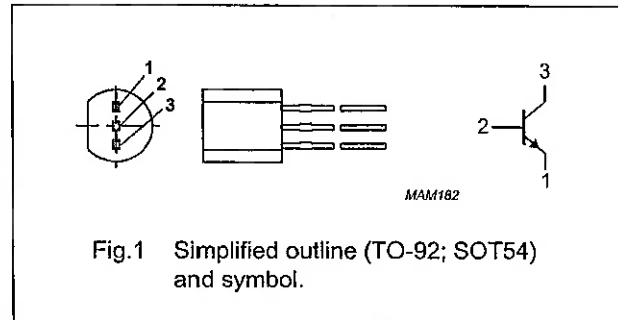
- General purpose switching and amplification.

DESCRIPTION

NPN transistor in a TO-92; SOT54 plastic package.
PNP complements: BC556, BC557 and BC558.

PINNING

PIN	DESCRIPTION
1	emitter
2	base
3	collector



QUICK REFERENCE DATA

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{CBO}	collector-base voltage	open emitter			
	BC546		–	80	V
	BC547		–	50	V
V_{CEO}	collector-emitter voltage	open base			
	BC546		–	65	V
	BC547		–	45	V
	BC548		–	30	V
I_{CM}	peak collector current		–	200	mA
P_{tot}	total power dissipation	$T_{amb} \leq 25\text{ }^\circ\text{C}$	–	500	mW
h_{FE}	DC current gain	$I_C = 2\text{ mA}; V_{CE} = 5\text{ V}$			
	BC546		110	450	
	BC547		110	800	
	BC548		110	800	
f_T	transition frequency	$I_C = 10\text{ mA}; V_{CE} = 5\text{ V}; f = 100\text{ MHz}$	100	–	MHz

NPN general purpose transistors

BC546; BC547; BC548

LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V _{CB0}	collector-base voltage	open emitter			
	BC546		–	80	V
	BC547		–	50	V
V _{CEO}	collector-emitter voltage	open base			
	BC546		–	65	V
	BC547		–	45	V
V _{EB0}	emitter-base voltage	open collector			
	BC546		–	6	V
	BC547		–	6	V
	BC548		–	5	V
I _c	collector current (DC)		–	100	mA
I _{CM}	peak collector current		–	200	mA
I _{BM}	peak base current		–	200	mA
P _{tot}	total power dissipation	T _{amb} ≤ 25 °C; note 1	–	500	mW
T _{stg}	storage temperature		–65	+150	°C
T _j	junction temperature		–	150	°C
T _{amb}	operating ambient temperature		–65	+150	°C

Note

1. Transistor mounted on an FR4 printed-circuit board.

THERMAL CHARACTERISTICS

SYMBOL	PARAMETER	CONDITIONS	VALUE	UNIT
R _{th j-a}	thermal resistance from junction to ambient	note 1	0.25	K/mW

Note

1. Transistor mounted on an FR4 printed-circuit board.

NPN general purpose transistors

BC546; BC547; BC548

CHARACTERISTICS

 $T_j = 25\text{ }^\circ\text{C}$ unless otherwise specified.

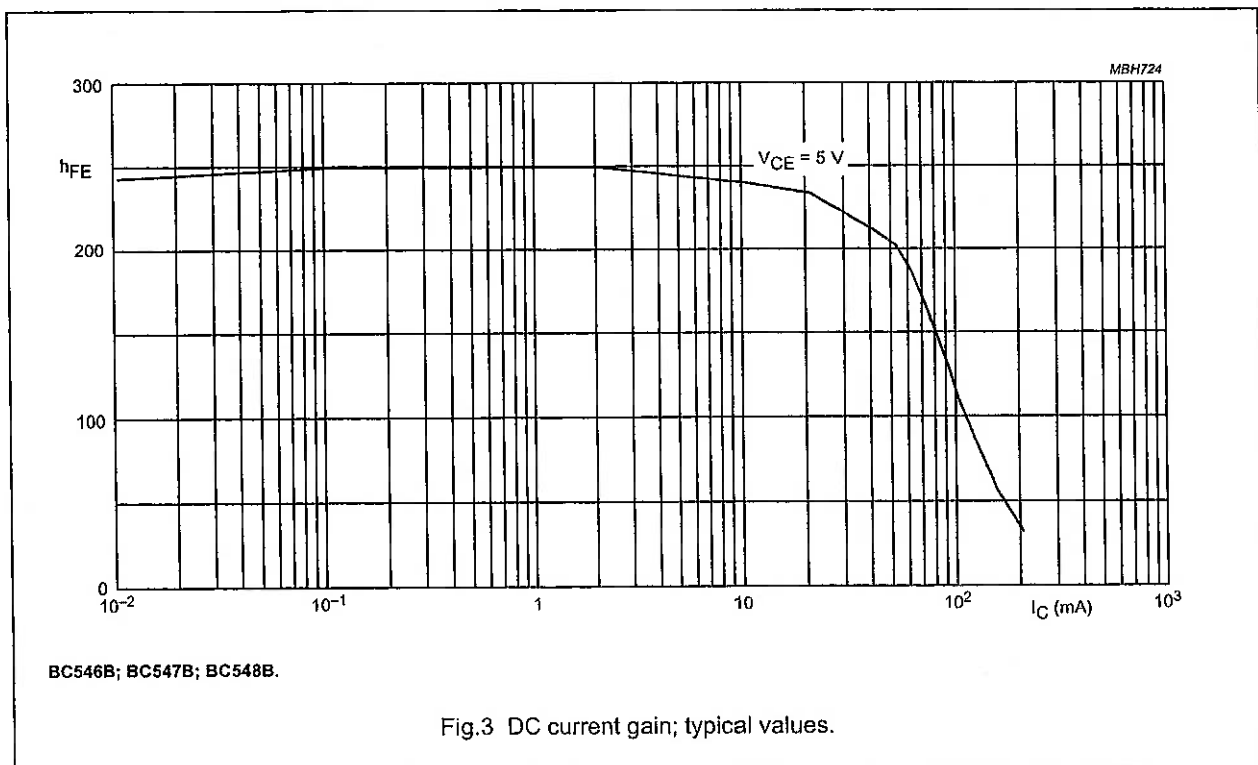
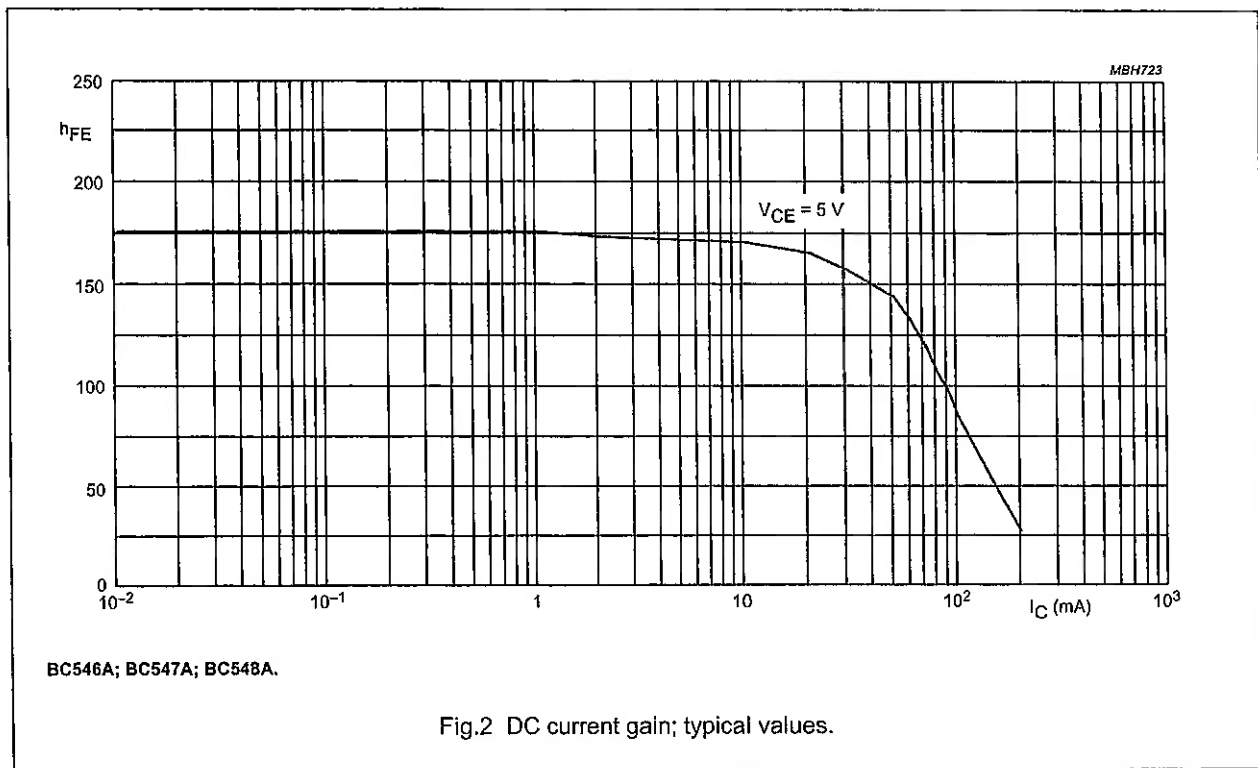
SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
I_{CBO}	collector cut-off current	$I_E = 0; V_{CB} = 30\text{ V}$	–	–	15	nA
		$I_E = 0; V_{CB} = 30\text{ V}; T_j = 150\text{ }^\circ\text{C}$	–	–	5	μA
I_{EBO}	emitter cut-off current	$I_C = 0; V_{EB} = 5\text{ V}$	–	–	100	nA
h_{FE}	DC current gain BC546A; BC547A; BC548A BC546B; BC547B; BC548B BC547C; BC548C	$I_C = 10\text{ }\mu\text{A}; V_{CE} = 5\text{ V};$ see Figs 2, 3 and 4	–	90	–	
			–	150	–	
			–	270	–	
h_{FE}	DC current gain BC546A; BC547A; BC548A BC546B; BC547B; BC548B BC547C; BC548C BC547; BC548 BC546	$I_C = 2\text{ mA}; V_{CE} = 5\text{ V};$ see Figs 2, 3 and 4	110	180	220	
			200	290	450	
			420	520	800	
			110	–	800	
			110	–	450	
V_{CEsat}	collector-emitter saturation voltage	$I_C = 10\text{ mA}; I_B = 0.5\text{ mA}$	–	90	250	mV
		$I_C = 100\text{ mA}; I_B = 5\text{ mA}$	–	200	600	mV
V_{BEsat}	base-emitter saturation voltage	$I_C = 10\text{ mA}; I_B = 0.5\text{ mA};$ note 1	–	700	–	mV
		$I_C = 100\text{ mA}; I_B = 5\text{ mA};$ note 1	–	900	–	mV
V_{BE}	base-emitter voltage	$I_C = 2\text{ mA}; V_{CE} = 5\text{ V};$ note 2	580	660	700	mV
		$I_C = 10\text{ mA}; V_{CE} = 5\text{ V}$	–	–	770	mV
C_c	collector capacitance	$I_E = I_e = 0; V_{CB} = 10\text{ V}; f = 1\text{ MHz}$	–	1.5	–	pF
C_e	emitter capacitance	$I_C = I_c = 0; V_{EB} = 0.5\text{ V}; f = 1\text{ MHz}$	–	11	–	pF
f_T	transition frequency	$I_C = 10\text{ mA}; V_{CE} = 5\text{ V}; f = 100\text{ MHz}$	100	–	–	MHz
F	noise figure	$I_C = 200\text{ }\mu\text{A}; V_{CE} = 5\text{ V};$ $R_S = 2\text{ k}\Omega; f = 1\text{ kHz}; B = 200\text{ Hz}$	–	2	10	dB

Notes

- V_{BEsat} decreases by about 1.7 mV/K with increasing temperature.
- V_{BE} decreases by about 2 mV/K with increasing temperature.

NPN general purpose transistors

BC546; BC547; BC548



NPN general purpose transistors

BC546; BC547; BC548

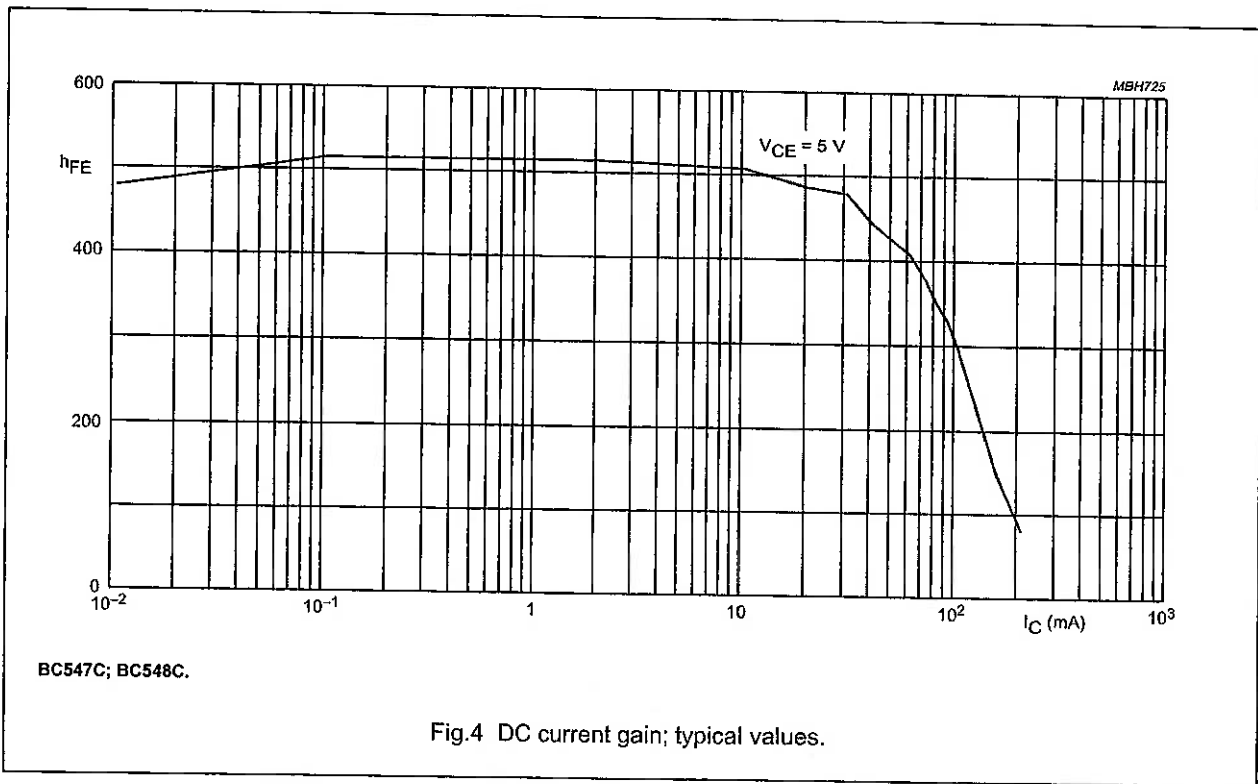


Fig.4 DC current gain; typical values.

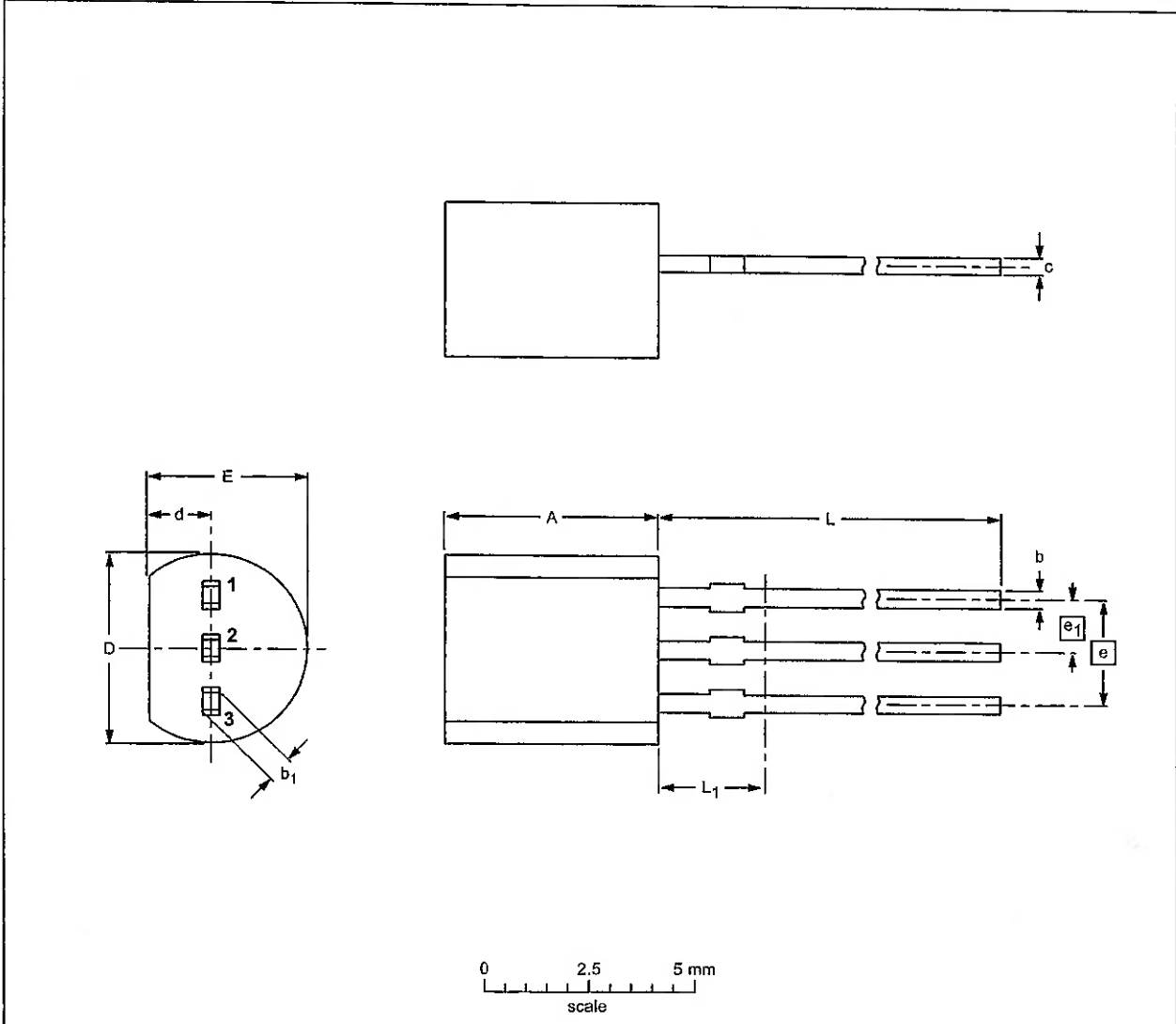
NPN general purpose transistors

BC546; BC547; BC548

PACKAGE OUTLINE

Plastic single-ended leaded (through hole) package; 3 leads

SOT54



DIMENSIONS (mm are the original dimensions)

UNIT	A	b	b ₁	c	D	d	E	e	e ₁	L	L ₁ ⁽¹⁾
mm	5.2	0.48	0.66	0.45	4.8	1.7	4.2	2.54	1.27	14.5	2.5
	5.0	0.40	0.56	0.40	4.4	1.4	3.6			12.7	

Note

1. Terminal dimensions within this zone are uncontrolled to allow for flow of plastic and terminal irregularities.

OUTLINE VERSION	REFERENCES			EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ		
SOT54		TO-92	SC-43		97-02-28

NPN general purpose transistors

BC546; BC547; BC548

DEFINITIONS

Data Sheet Status	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
Limiting values	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
Application information	
Where application information is given, it is advisory and does not form part of the specification.	

LIFE SUPPORT APPLICATIONS

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

Philips Semiconductors – a worldwide company

Argentina: see South America

Australia: 34 Waterloo Road, NORTH RYDE, NSW 2113,
Tel. +61 2 9805 4455, Fax. +61 2 9805 4466

Austria: Computerstr. 6, A-1101 WIEN, P.O. Box 213,
Tel. +43 1 60 101, Fax. +43 1 60 101 1210

Belarus: Hotel Minsk Business Center, Bld. 3, r. 1211, Volodarski Str. 6,
220050 MINSK, Tel. +375 172 200 733, Fax. +375 172 200 773

Belgium: see The Netherlands

Brazil: see South America

Bulgaria: Philips Bulgaria Ltd., Energoproject, 15th floor,
51 James Bourchier Blvd., 1407 SOFIA,
Tel. +359 2 689 211, Fax. +359 2 689 102

Canada: PHILIPS SEMICONDUCTORS/COMPONENTS,
Tel. +1 800 234 7381

China/Hong Kong: 501 Hong Kong Industrial Technology Centre,
72 Tat Chee Avenue, Kowloon Tong, HONG KONG,
Tel. +852 2319 7888, Fax. +852 2319 7700

Colombia: see South America

Czech Republic: see Austria

Denmark: Prags Boulevard 80, PB 1919, DK-2300 COPENHAGEN S,
Tel. +45 32 88 2636, Fax. +45 31 57 1949

Finland: Sinikalliontie 3, FIN-02630 ESPOO,
Tel. +358 9 615800, Fax. +358 9 61580/xxx

France: 4 Rue du Port-aux-Vins, BP317, 92156 SURESNES Cedex,
Tel. +33 1 40 99 6161, Fax. +33 1 40 99 6427

Germany: Hammerbrookstraße 69, D-20097 HAMBURG,
Tel. +49 40 23 53 60, Fax. +49 40 23 536 300

Greece: No. 15, 25th March Street, GR 17778 TAVROS/ATHENS,
Tel. +30 1 4894 339/239, Fax. +30 1 4814 240

Hungary: see Austria

India: Philips INDIA Ltd, Shivsagar Estate, A Block, Dr. Annie Besant Rd.
Worli, MUMBAI 400 018, Tel. +91 22 4938 541, Fax. +91 22 4938 722

Indonesia: see Singapore

Ireland: Newstead, Clonskeagh, DUBLIN 14,
Tel. +353 1 7640 000, Fax. +353 1 7640 200

Israel: RAPAC Electronics, 7 Kehilat Saloniki St, TEL AVIV 61180,
Tel. +972 3 645 0444, Fax. +972 3 649 1007

Italy: PHILIPS SEMICONDUCTORS, Piazza IV Novembre 3,
20124 MILANO, Tel. +39 2 6752 2531, Fax. +39 2 6752 2557

Japan: Philips Bldg 13-37, Kohnan 2-chome, Minato-ku, TOKYO 108,
Tel. +81 3 3740 5130, Fax. +81 3 3740 5077

Korea: Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL,
Tel. +82 2 709 1412, Fax. +82 2 709 1415

Malaysia: No. 76 Jalan Universiti, 46200 PETALING JAYA, SELANGOR,
Tel. +60 3 750 5214, Fax. +60 3 757 4880

Mexico: 5900 Gateway East, Suite 200, EL PASO, TEXAS 79905,
Tel. +9-5 800 234 7381

Middle East: see Italy

Netherlands: Postbus 90050, 5600 PB EINDHOVEN, Bldg. VB,
Tel. +31 40 27 82785, Fax. +31 40 27 86399

New Zealand: 2 Wagener Place, C.P.O. Box 1041, AUCKLAND,
Tel. +64 9 849 4160, Fax. +64 9 849 7811

Norway: Box 1, Manglerud 0612, OSLO,
Tel. +47 22 74 8000, Fax. +47 22 74 8341

Philippines: Philips Semiconductors Philippines Inc.,
106 Valero St. Salcedo Village, P.O. Box 2108 MCC, MAKATI,
Metro MANILA, Tel. +63 2 816 6380, Fax. +63 2 817 3474

Poland: Ul. Lukiska 10, PL 04-123 WARSZAWA,
Tel. +48 22 612 2831, Fax. +48 22 612 2327

Portugal: see Spain

Romania: see Italy

Russia: Philips Russia, Ul. Usatcheva 35A, 119048 MOSCOW,
Tel. +7 095 755 6918, Fax. +7 095 755 6919

Singapore: Lorong 1, Toa Payoh, SINGAPORE 1231,
Tel. +65 350 2538, Fax. +65 251 6500

Slovakia: see Austria

Slovenia: see Italy

South Africa: S.A. PHILIPS Pty Ltd., 195-215 Main Road Martindale,
2092 JOHANNESBURG, P.O. Box 7430 Johannesburg 2000,
Tel. +27 11 470 5911, Fax. +27 11 470 5494

South America: Rua do Rocio 220, 5th floor, Suite 51,
04552-903 São Paulo, SÃO PAULO - SP, Brazil,
Tel. +55 11 821 2333, Fax. +55 11 829 1849

Spain: Balmes 22, 08007 BARCELONA,
Tel. +34 3 301 6312, Fax. +34 3 301 4107

Sweden: Kottbygatan 7, Akalla, S-16485 STOCKHOLM,
Tel. +46 8 632 2000, Fax. +46 8 632 2745

Switzerland: Allmendstrasse 140, CH-8027 ZÜRICH,
Tel. +41 1 488 2686, Fax. +41 1 481 7730

Taiwan: Philips Semiconductors, 6F, No. 96, Chien Kuo N. Rd., Sec. 1,
TAIPEI, Taiwan Tel. +886 2 2134 2870, Fax. +886 2 2134 2874

Thailand: PHILIPS ELECTRONICS (THAILAND) Ltd.,
209/2 Sanpavuth-Bangna Road Prakanong, BANGKOK 10260,
Tel. +66 2 745 4090, Fax. +66 2 398 0793

Turkey: Talatpasa Cad. No. 5, 80640 GÜLTEPE/ISTANBUL,
Tel. +90 212 279 2770, Fax. +90 212 282 6707

Ukraine: PHILIPS UKRAINE, 4 Patrice Lumumba str., Building B, Floor 7,
252042 KIEV, Tel. +380 44 264 2776, Fax. +380 44 268 0461

United Kingdom: Philips Semiconductors Ltd., 276 Bath Road, Hayes,
MIDDLESEX UB3 5BX, Tel. +44 181 730 5000, Fax. +44 181 754 8421

United States: 811 East Arques Avenue, SUNNYVALE, CA 94088-3409,
Tel. +1 800 234 7381

Uruguay: see South America

Vietnam: see Singapore

Yugoslavia: PHILIPS, Trg N. Pasica 5/v, 11000 BEOGRAD,
Tel. +381 11 625 344, Fax. +381 11 635 777

For all other countries apply to: Philips Semiconductors, Marketing & Sales Communications,
Building BE-p, P.O. Box 218, 5600 MD EINDHOVEN, The Netherlands, Fax. +31 40 27 24825

Internet: <http://www.semiconductors.philips.com>

© Philips Electronics N.V. 1997

SCA53

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner.

The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Printed in The Netherlands

117047/00/02/pp12

Date of release: 1997 Mar 04

Document order number: 9397 750 01853

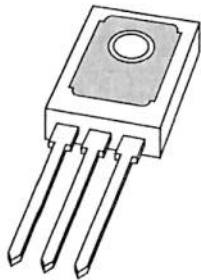
Let's make things better.

**Philips
Semiconductors**



PHILIPS

DATA SHEET



BD135; BD137; BD139 NPN power transistors

Product specification
Supersedes data of 1997 Mar 04

1999 Apr 12

NPN power transistors

BD135; BD137; BD139

FEATURES

- High current (max. 1.5 A)
- Low voltage (max. 80 V).

APPLICATIONS

- Driver stages in hi-fi amplifiers and television circuits.

DESCRIPTION

NPN power transistor in a TO-126; SOT32 plastic package. PNP complements: BD136, BD138 and BD140.

PINNING

PIN	DESCRIPTION
1	emitter
2	collector, connected to metal part of mounting surface
3	base

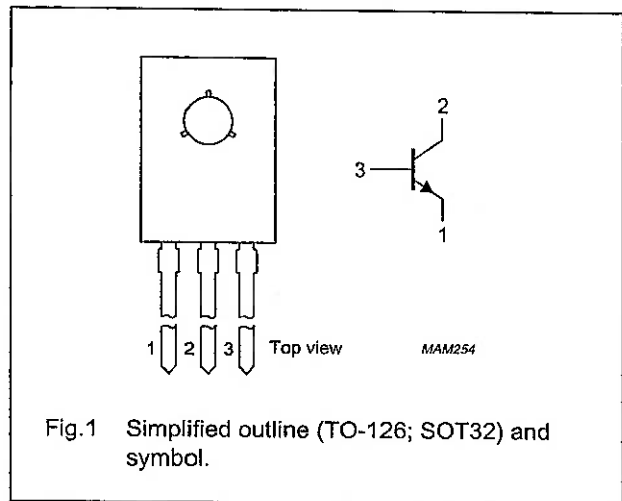


Fig.1 Simplified outline (TO-126; SOT32) and symbol.

LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V _{CBO}	collector-base voltage	open emitter			
	BD135		—	45	V
	BD137		—	60	V
	BD139		—	100	V
V _{CEO}	collector-emitter voltage	open base			
	BD135		—	45	V
	BD137		—	60	V
	BD139		—	80	V
V _{EBO}	emitter-base voltage	open collector	—	5	V
I _C	collector current (DC)		—	1.5	A
I _{CM}	peak collector current		—	2	A
I _{BM}	peak base current		—	1	A
P _{tot}	total power dissipation	T _{mb} ≤ 70 °C	—	8	W
T _{stg}	storage temperature		−65	+150	°C
T _J	junction temperature		—	150	°C
T _{amb}	operating ambient temperature		−65	+150	°C

NPN power transistors

BD135; BD137; BD139

THERMAL CHARACTERISTICS

SYMBOL	PARAMETER	CONDITIONS	VALUE	UNIT
$R_{th\ j-a}$	thermal resistance from junction to ambient	note 1	100	K/W
$R_{th\ j-mb}$	thermal resistance from junction to mounting base		10	K/W

Note

1. Refer to TO-126; SOT32 standard mounting conditions.

CHARACTERISTICS

$T_j = 25\text{ °C}$ unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
I_{CBO}	collector cut-off current	$I_E = 0; V_{CB} = 30\text{ V}$	–	–	100	nA
		$I_E = 0; V_{CB} = 30\text{ V}; T_j = 125\text{ °C}$	–	–	10	μA
I_{EBO}	emitter cut-off current	$I_C = 0; V_{EB} = 5\text{ V}$	–	–	100	nA
h_{FE}	DC current gain	$V_{CE} = 2\text{ V};$ (see Fig.2)				
		$I_C = 5\text{ mA}$	40	–	–	
		$I_C = 150\text{ mA}$	63	–	250	
	$I_C = 500\text{ mA}$	25	–	–		
	DC current gain BD135-10; BD137-10; BD139-10 BD135-16; BD137-16; BD139-16	$I_C = 150\text{ mA}; V_{CE} = 2\text{ V};$ (see Fig.2)	63 100	– –	160 250	
V_{CEsat}	collector-emitter saturation voltage	$I_C = 500\text{ mA}; I_B = 50\text{ mA}$	–	–	0.5	V
V_{BE}	base-emitter voltage	$I_C = 500\text{ mA}; V_{CE} = 2\text{ V}$	–	–	1	V
f_T	transition frequency	$I_C = 50\text{ mA}; V_{CE} = 5\text{ V};$ $f = 100\text{ MHz}$	–	190	–	MHz
$\frac{h_{FE1}}{h_{FE2}}$	DC current gain ratio of the complementary pairs	$ I_C = 150\text{ mA}; V_{CE} = 2\text{ V}$	–	1.3	1.6	

NPN power transistors

BD135; BD137; BD139

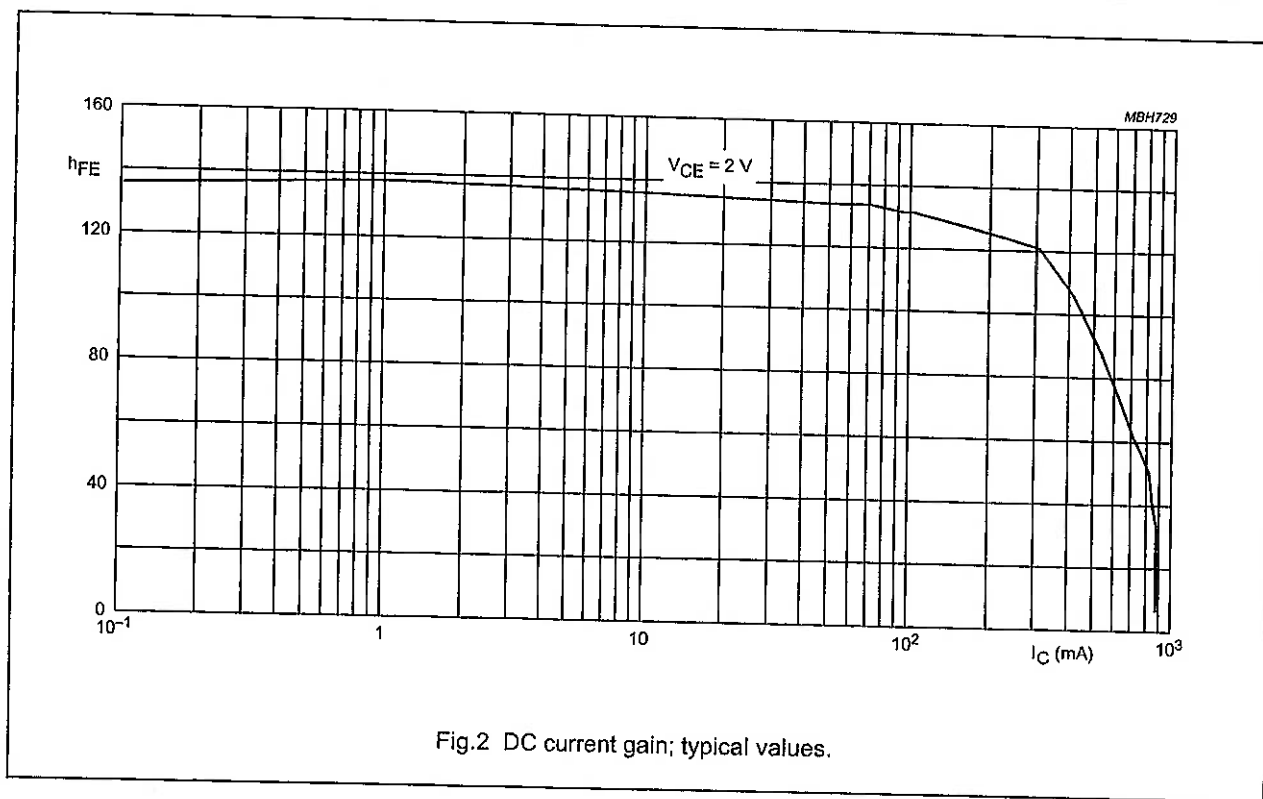


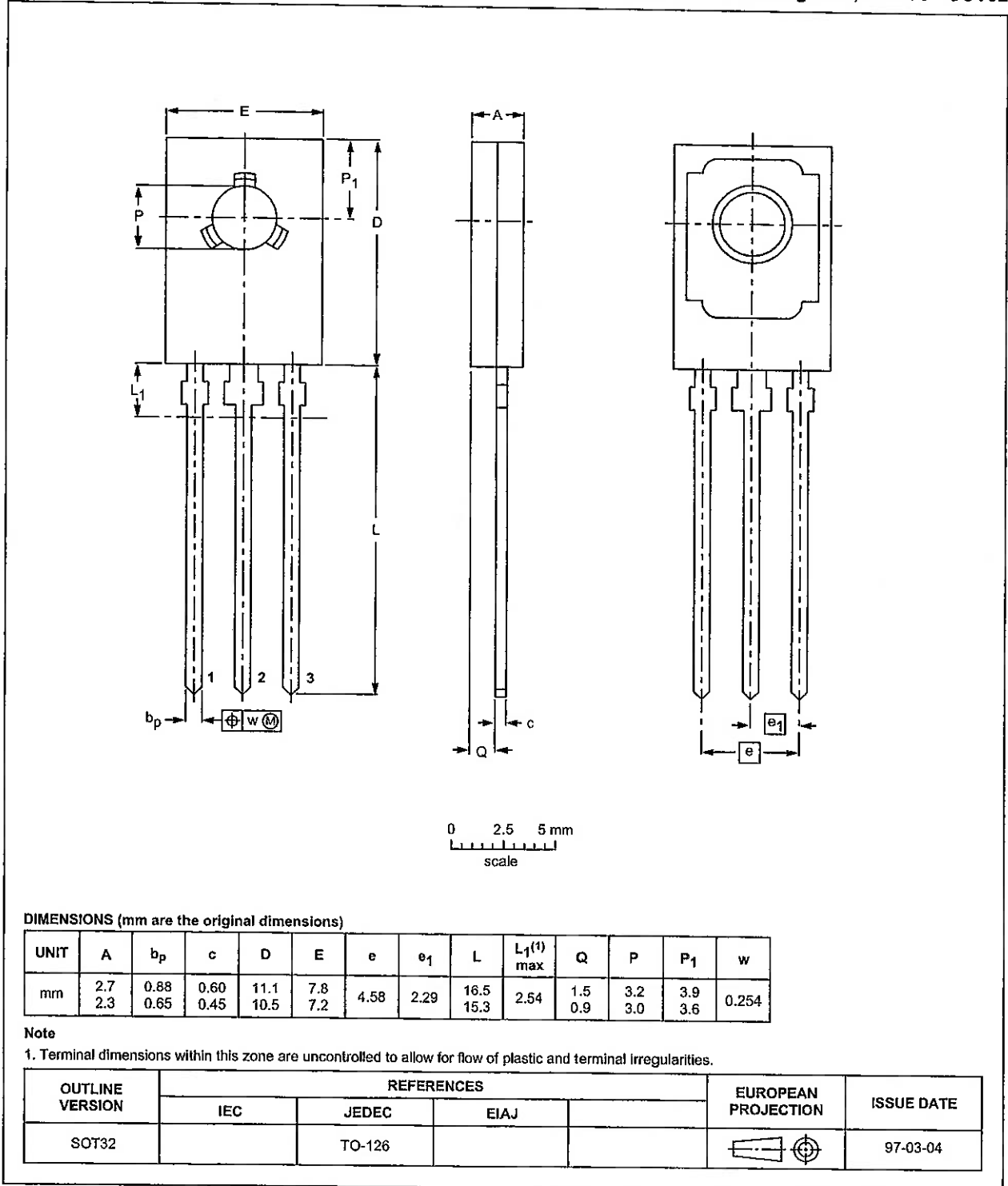
Fig.2 DC current gain; typical values.

NPN power transistors

BD135; BD137; BD139

PACKAGE OUTLINE

Plastic single-ended leaded (through hole) package; mountable to heatsink, 1 mounting hole; 3 leads SOT32



NPN power transistors

BD135; BD137; BD139

DEFINITIONS

Data Sheet Status	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
Limiting values	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
Application information	
Where application information is given, it is advisory and does not form part of the specification.	

LIFE SUPPORT APPLICATIONS

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

Philips Semiconductors – a worldwide company

Argentina: see South America

Australia: 34 Waterloo Road, NORTH RYDE, NSW 2113,
Tel. +61 2 9805 4455, Fax. +61 2 9805 4466

Austria: Computerstr. 6, A-1101 WIEN, P.O. Box 213,
Tel. +43 1 60 101 1248, Fax. +43 1 60 101 1210

Belarus: Hotel Minsk Business Center, Bld. 3, r. 1211, Volodarski Str. 6,
220050 MINSK, Tel. +375 172 20 0733, Fax. +375 172 20 0773

Belgium: see The Netherlands

Brazil: see South America

Bulgaria: Philips Bulgaria Ltd., Energoproject, 15th floor,
51 James Bouchier Blvd., 1407 SOFIA,
Tel. +359 2 68 9211, Fax. +359 2 68 9102

Canada: PHILIPS SEMICONDUCTORS/COMPONENTS,
Tel. +1 800 234 7381, Fax. +1 800 943 0087

China/Hong Kong: 501 Hong Kong Industrial Technology Centre,
72 Tat Chee Avenue, Kowloon Tong, HONG KONG,
Tel. +852 2319 7888, Fax. +852 2319 7700

Colombia: see South America

Czech Republic: see Austria

Denmark: Sydhavnsgade 23, 1780 COPENHAGEN V,
Tel. +45 33 29 3333, Fax. +45 33 29 3905

Finland: Sinikalliontie 3, FIN-02630 ESPOO,
Tel. +358 9 615 800, Fax. +358 9 6158 0920

France: 51 Rue Carnot, BP317, 92156 SURESNES Cedex,
Tel. +33 1 4099 6161, Fax. +33 1 4099 6427

Germany: Hammerbrookstraße 69, D-20097 HAMBURG,
Tel. +49 40 2353 60, Fax. +49 40 2353 6300

Hungary: see Austria

India: Philips INDIA Ltd, Band Box Building, 2nd floor,
254-D, Dr. Annie Besant Road, Worli, MUMBAI 400 025,
Tel. +91 22 493 8541, Fax. +91 22 493 0966

Indonesia: PT Philips Development Corporation, Semiconductors Division,
Gedung Philips, Jl. Buncit Raya Kav.99-100, JAKARTA 12510,
Tel. +62 21 794 0040 ext. 2501, Fax. +62 21 794 0080

Ireland: Newstead, Clonskeagh, DUBLIN 14,
Tel. +353 1 7640 000, Fax. +353 1 7640 200

Israel: RAPAC Electronics, 7 Kehilat Saloniki St, PO Box 18053,
TEL AVIV 61180, Tel. +972 3 645 0444, Fax. +972 3 649 1007

Italy: PHILIPS SEMICONDUCTORS, Piazza IV Novembre 3,
20124 MILANO, Tel. +39 2 6752 2531, Fax. +39 2 6752 2557

Japan: Philips Bldg 13-37, Kohnan 2-chome, Minato-ku,
TOKYO 108-8507, Tel. +81 3 3740 5130, Fax. +81 3 3740 5077

Korea: Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL,
Tel. +82 2 709 1412, Fax. +82 2 709 1415

Malaysia: No. 76 Jalan Universiti, 46200 PETALING JAYA, SELANGOR,
Tel. +60 3 750 5214, Fax. +60 3 757 4880

Mexico: 5900 Gateway East, Suite 200, EL PASO, TEXAS 79905,
Tel. +9-5 800 234 7381, Fax +9-5 800 943 0087

Middle East: see Italy

Netherlands: Postbus 90050, 5600 PB EINDHOVEN, Bldg. VB,
Tel. +31 40 27 82785, Fax. +31 40 27 88399

New Zealand: 2 Wagener Place, C.P.O. Box 1041, AUCKLAND,
Tel. +64 9 849 4160, Fax. +64 9 849 7811

Norway: Box 1, Manglerud 0612, OSLO,
Tel. +47 22 74 8000, Fax. +47 22 74 8341

Pakistan: see Singapore

Philippines: Philips Semiconductors Philippines Inc.,
106 Valero St. Salcedo Village, P.O. Box 2108 MCC, MAKATI,
Metro MANILA, Tel. +63 2 816 6380, Fax. +63 2 817 3474

Poland: Ul. Lukiska 10, PL 04-123 WARSZAWA,
Tel. +48 22 612 2831, Fax. +48 22 612 2327

Portugal: see Spain

Romania: see Italy

Russia: Philips Russia, Ul. Usatcheva 35A, 119048 MOSCOW,
Tel. +7 095 755 6918, Fax. +7 095 755 6919

Singapore: Lorong 1, Toa Payoh, SINGAPORE 319762,
Tel. +65 350 2538, Fax. +65 251 6500

Slovakia: see Austria

Slovenia: see Italy

South Africa: S.A. PHILIPS Pty Ltd., 195-215 Main Road Martindale,
2092 JOHANNESBURG, P.O. Box 7430 Johannesburg 2000,
Tel. +27 11 470 5911, Fax. +27 11 470 5494

South America: Al. Vicente Pinzon, 173, 6th floor,
04547-130 SÃO PAULO, SP, Brazil,
Tel. +55 11 821 2333, Fax. +55 11 821 2382

Spain: Balmes 22, 08007 BARCELONA,
Tel. +34 93 301 6312, Fax. +34 93 301 4107

Sweden: Kottbygatan 7, Akalla, S-16485 STOCKHOLM,
Tel. +46 8 5985 2000, Fax. +46 8 5985 2745

Switzerland: Allmendstrasse 140, CH-8027 ZÜRICH,
Tel. +41 1 488 2741 Fax. +41 1 488 3263

Taiwan: Philips Semiconductors, 6F, No. 96, Chien Kuo N. Rd., Sec. 1,
TAIPEI, Taiwan Tel. +886 2 2134 2886, Fax. +886 2 2134 2874

Thailand: PHILIPS ELECTRONICS (THAILAND) Ltd.,
209/2 Sanpavuth-Bangna Road Prakanong, BANGKOK 10260,
Tel. +66 2 745 4090, Fax. +66 2 398 0793

Turkey: Talatpasa Cad. No. 5, 80640 GÜLTEPE/ISTANBUL,
Tel. +90 212 279 2770, Fax. +90 212 282 6707

Ukraine: PHILIPS UKRAINE, 4 Patrice Lumumba str., Building B, Floor 7,
252042 KIEV, Tel. +380 44 264 2776, Fax. +380 44 268 0461

United Kingdom: Philips Semiconductors Ltd., 276 Bath Road, Hayes,
MIDDLESEX UB3 5BX, Tel. +44 181 730 5000, Fax. +44 181 754 8421

United States: 811 East Arques Avenue, SUNNYVALE, CA 94088-3409,
Tel. +1 800 234 7381, Fax. +1 800 943 0087

Uruguay: see South America

Vietnam: see Singapore

Yugoslavia: PHILIPS, Trg N. Pasica 5/v, 11000 BEOGRAD,
Tel. +381 11 62 5344, Fax. +381 11 63 5777

For all other countries apply to: Philips Semiconductors,
International Marketing & Sales Communications, Building BE-p, P.O. Box 218,
5600 MD EINDHOVEN, The Netherlands, Fax. +31 40 27 24825

Internet: <http://www.semiconductors.philips.com>

© Philips Electronics N.V. 1999

SCA63

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner.

The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Printed in The Netherlands

115002/00/03/pp8

Date of release: 1999 Apr 12

Document order number: 9397 750 05576

Let's make things better.

**Philips
Semiconductors**

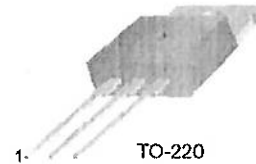


PHILIPS

TIP41 Series(TIP41/41A/41B/41C)

Medium Power Linear Switching Applications

- Complement to TIP42/42A/42B/42C



1. Base 2. Collector 3. Emitter

NPN Epitaxial Silicon Transistor

Absolute Maximum Ratings $T_C=25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Value	Units
V_{CBO}	Collector-Emitter Voltage: TIP41	40	V
	: TIP41A	60	V
	: TIP41B	80	V
	: TIP41C	100	V
V_{CEO}	Collector-Emitter Voltage: TIP41	40	V
	: TIP41A	60	V
	: TIP41B	80	V
	: TIP41C	100	V
V_{EBO}	Emitter-Base Voltage	5	V
I_C	Collector Current (DC)	6	A
I_{CP}	Collector Current (Pulse)	10	A
I_B	Base Current	2	A
P_C	Collector Dissipation ($T_C=25^\circ\text{C}$)	65	W
P_C	Collector Dissipation ($T_a=25^\circ\text{C}$)	2	W
T_J	Junction Temperature	150	$^\circ\text{C}$
T_{STG}	Storage Temperature	- 65 ~ 150	$^\circ\text{C}$

Electrical Characteristics $T_C=25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Max.	Units
$V_{CEO(sus)}$	* Collector-Emitter Sustaining Voltage	$I_C = 30\text{mA}, I_B = 0$	40		V
	: TIP41				
	: TIP41A				
	: TIP41B				
I_{CEO}	Collector Cut-off Current	$V_{CE} = 30\text{V}, I_B = 0$ $V_{CE} = 60\text{V}, I_B = 0$		0.7	mA
	: TIP41/41A : TIP41B/41C				
I_{CES}	Collector Cut-off Current	$V_{CE} = 40\text{V}, V_{EB} = 0$ $V_{CE} = 60\text{V}, V_{EB} = 0$ $V_{CE} = 80\text{V}, V_{EB} = 0$ $V_{CE} = 100\text{V}, V_{EB} = 0$		400	μA
	: TIP41				
	: TIP41A				
	: TIP41B				
I_{EBO}	Emitter Cut-off Current	$V_{EB} = 5\text{V}, I_C = 0$		1	mA
	: TIP41/41A : TIP41B/41C				
h_{FE}	* DC Current Gain	$V_{CE} = 4\text{V}, I_C = 0.3\text{A}$ $V_{CE} = 4\text{V}, I_C = 3\text{A}$	30	75	
	: TIP41/41A : TIP41B/41C				
$V_{CE(sat)}$	* Collector-Emitter Saturation Voltage	$I_C = 6\text{A}, I_B = 600\text{mA}$		1.5	V
$V_{BE(sat)}$	* Base-Emitter Saturation Voltage	$V_{CE} = 4\text{V}, I_C = 6\text{A}$		2.0	V
f_T	Current Gain Bandwidth Product	$V_{CE} = 10\text{V}, I_C = 500\text{mA}$	3.0		MHz

* Pulse Test: PWS300 μs , Duty Cycles2%

Typical Characteristics

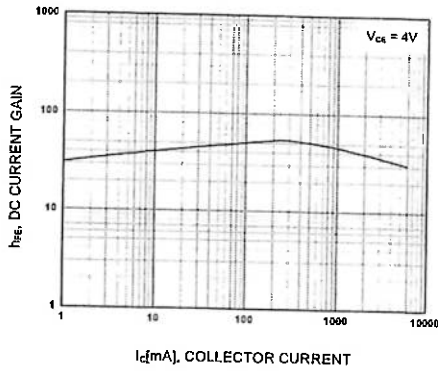


Figure 1. DC current Gain

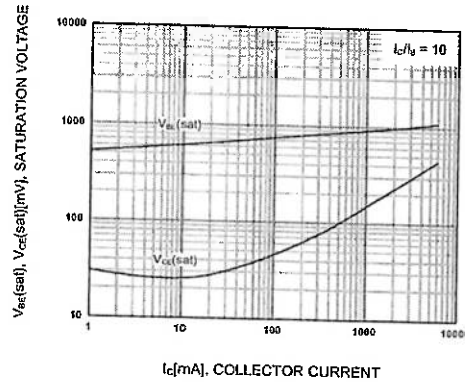


Figure 2. Base-Emitter Saturation Voltage
Collector-Emitter Saturation Voltage

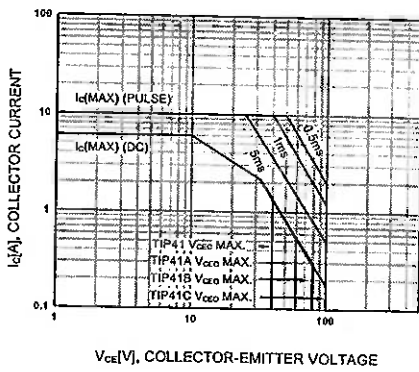


Figure 3. Safe Operating Area

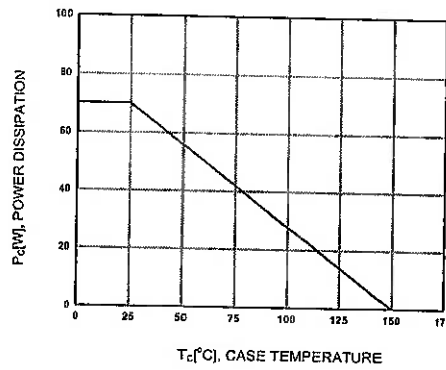
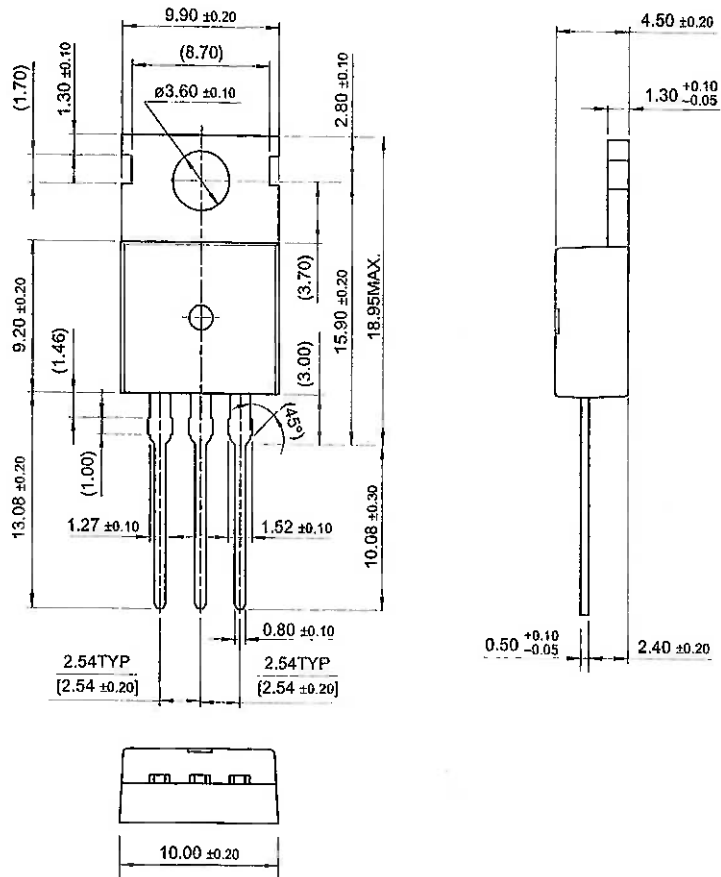


Figure 4. Power Derating

Package Dimensions

TIP41 Series (TIP41/A/41B/41C)

TO-220



Dimensions in Millimeters

TRADEMARKS

The following are registered and unregistered trademarks Fairchild Semiconductor owns or is authorized to use and is not intended to be an exhaustive list of all such trademarks.

ACE ^x ™	HiSeC™	SuperSOT™-8
Bottomless™	ISOPLANAR™	SyncFET™
CoolFET™	MICROWIRE™	TinyLogic™
CROSSVOLT™	POP™	UHC™
E ² CMOS™	PowerTrench®	VCX™
FACT™	QFET™	
FACT Quiet Series™	QS™	
FAST®	Quiet Series™	
FAST ^r ™	SuperSOT™-3	
GTO™	SuperSOT™-6	

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR INTERNATIONAL.

As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, or (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

PRODUCT STATUS DEFINITIONS

Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This datasheet contains preliminary data, and supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
No Identification Needed	Full Production	This datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
Obsolete	Not In Production	This datasheet contains specifications on a product that has been discontinued by Fairchild semiconductor. The datasheet is printed for reference information only.

DM74LS00 Quad 2-Input NAND Gate

General Description

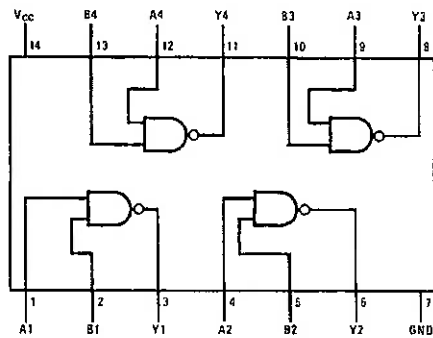
This device contains four independent gates each of which performs the logic NAND function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS00M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS00SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS00N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$$Y = \overline{AB}$$

Inputs		Output
A	B	Y
L	L	H
L	H	H
H	L	H
H	H	L

H = HIGH Logic Level
L = LOW Logic Level

Absolute Maximum Ratings(Note 1)

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	0°C to +70°C
Storage Temperature Range	-65°C to +150°C

Note 1: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the Electrical Characteristics tables are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Recommended Operating Conditions

Symbol	Parameter	Min	Nom	Max	Units
V _{CC}	Supply Voltage	4.75	5	5.25	V
V _{IH}	HIGH Level Input Voltage	2			V
V _{IL}	LOW Level Input Voltage			0.8	V
I _{OH}	HIGH Level Output Current			-0.4	mA
I _{OL}	LOW Level Output Current			8	mA
T _A	Free Air Operating Temperature	0		70	°C

Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
V _I	Input Clamp Voltage	V _{CC} = Min, I _I = -18 mA			-1.5	V
V _{OH}	HIGH Level Output Voltage	V _{CC} = Min, I _{OH} = Max, V _{IL} = Max	2.7	3.4		V
V _{OL}	LOW Level Output Voltage	V _{CC} = Min, I _{OL} = Max, V _{IH} = Min, I _{OL} = 4 mA, V _{CC} = Min		0.25	0.5	V
I _I	Input Current @ Max Input Voltage	V _{CC} = Max, V _I = 7V			0.1	mA
I _{IH}	HIGH Level Input Current	V _{CC} = Max, V _I = 2.7V			20	μA
I _{IL}	LOW Level Input Current	V _{CC} = Max, V _I = 0.4V			-0.36	mA
I _{OS}	Short Circuit Output Current	V _{CC} = Max (Note 3)	-20		-100	mA
I _{COH}	Supply Current with Outputs HIGH	V _{CC} = Max		0.8	1.6	mA
I _{CCL}	Supply Current with Outputs LOW	V _{CC} = Max		2.4	4.4	mA

Note 2: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 3: Not more than one output should be shorted at a time, and the duration should not exceed one second.

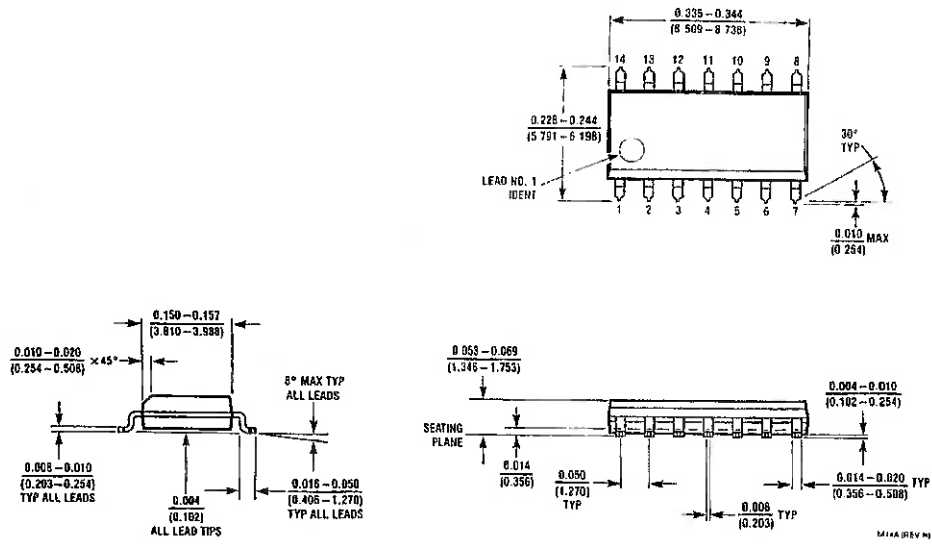
Switching Characteristics

at V_{CC} = 5V and T_A = 25°C

Symbol	Parameter	R _L = 2 kΩ				Units
		C _L = 15 pF		C _L = 50 pF		
		Min	Max	Min	Max	
t _{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	3	10	4	15	ns
t _{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	3	10	4	15	ns

Physical Dimensions inches (millimeters) unless otherwise noted

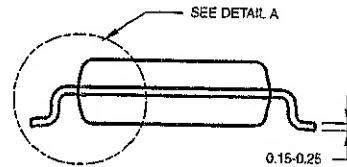
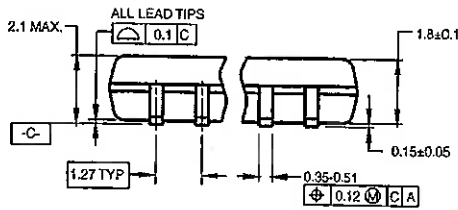
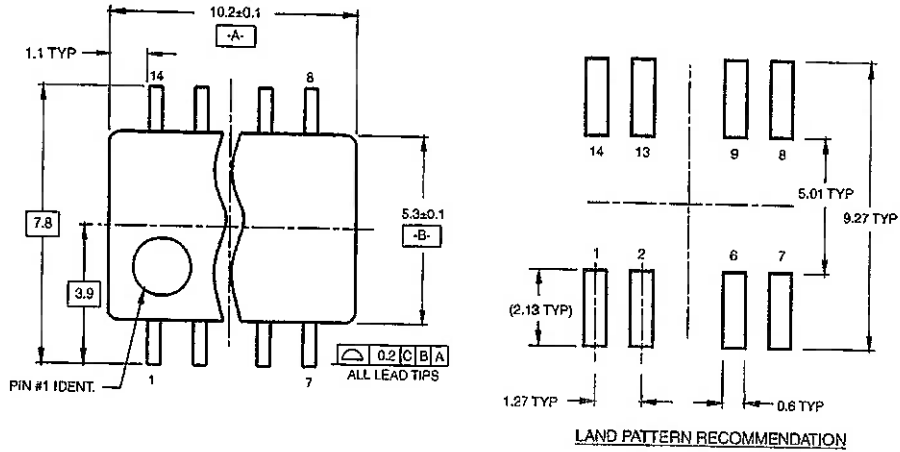
DM74LS00



14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
Package Number M14A

M14A (REV. N)

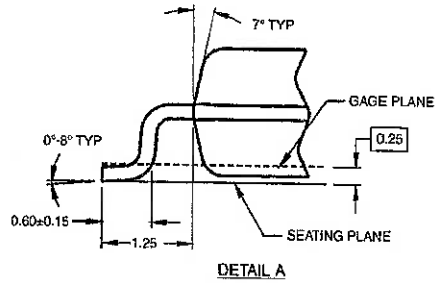
Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



DIMENSIONS ARE IN MILLIMETERS

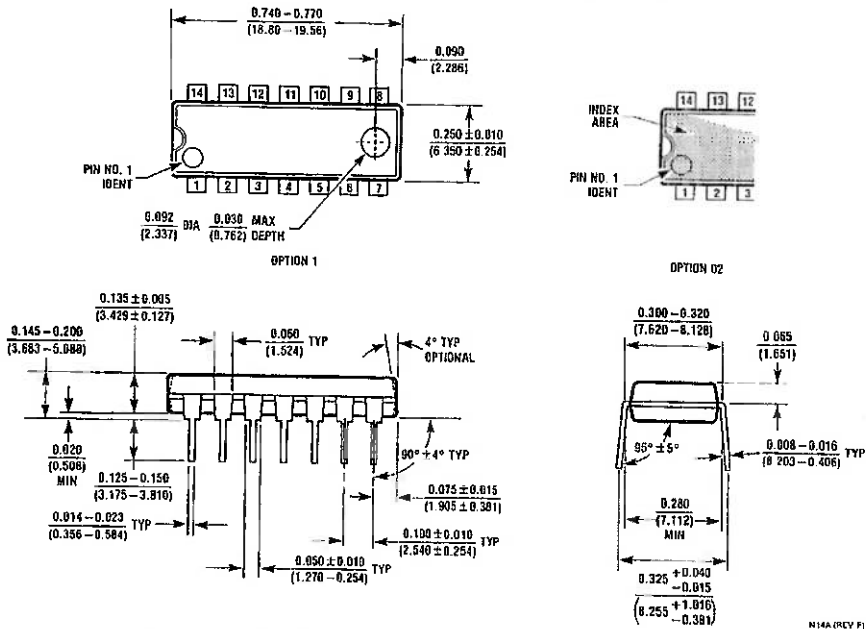
- NOTES:
 A. CONFORMS TO EIAJ EDR-7320 REGISTRATION, ESTABLISHED IN DECEMBER, 1998.
 B. DIMENSIONS ARE IN MILLIMETERS.
 C. DIMENSIONS ARE EXCLUSIVE OF BURRS, MOLD FLASH, AND TIE BAR EXTRUSIONS.

M14DRevB1



14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide Package Number M14D

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



**14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide
Package Number N14A**

Fairchild does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and Fairchild reserves the right at any time without notice to change said circuitry and specifications.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

www.fairchildsemi.com

SEMICONDUCTOR Ics

SGS-Thomson – L298N Dual full bridge driver

Order Code	Maxim Ref. No.	Description	Pin-Package
82-0202	L298N	Dual full bridge driver	M/Watt

**Dual full bridge driver****82-0202**Revision A
09/02/2000

The enclosed information is believed to be correct, Information may change 'without notice' due to product improvement. Users should ensure that the product is suitable for their use. E. & O. E.

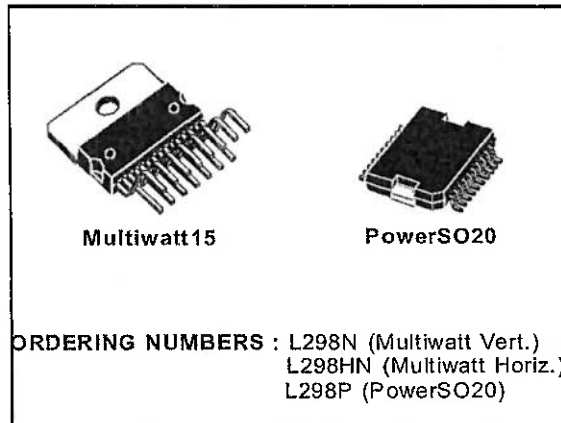
Page 1 of 14

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

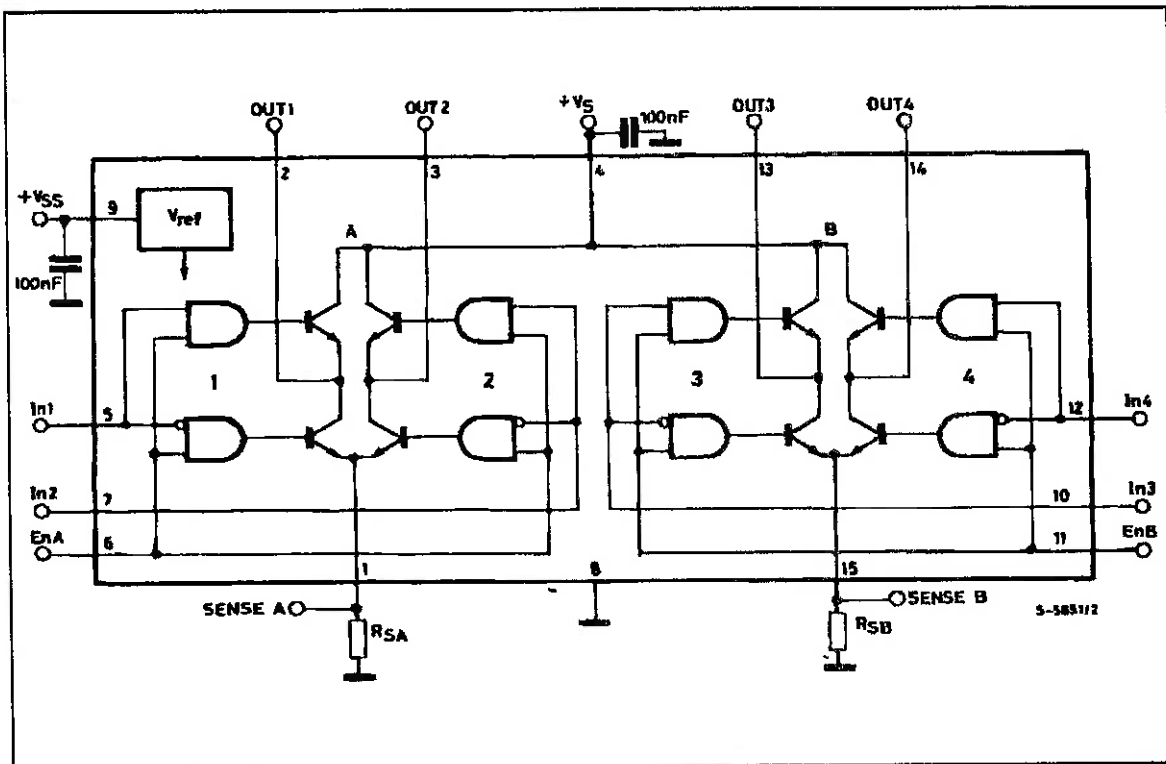
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

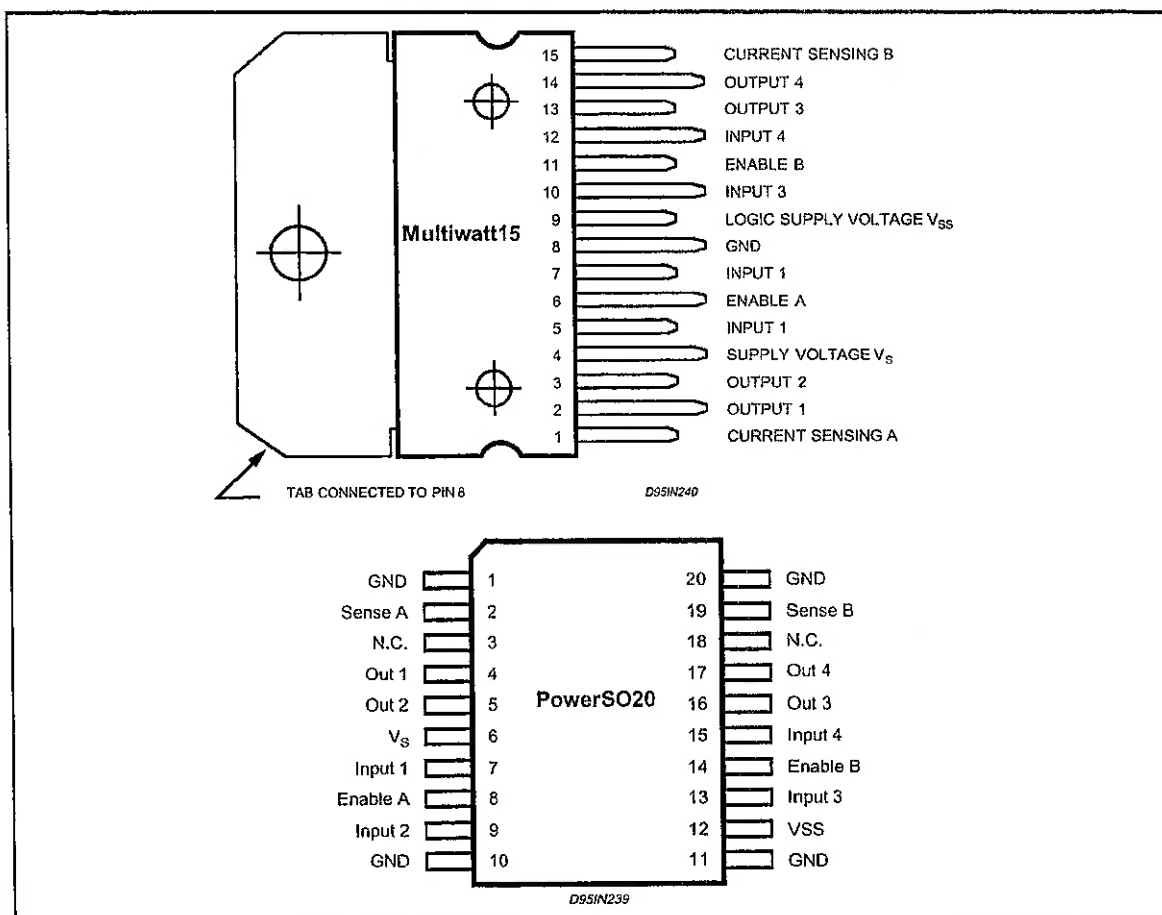
BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	- Non Repetitive ($t = 100\mu s$)	3	A
	- Repetitive (80% on -20% off; $t_{on} = 10ms$)	2.5	A
	- DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter	PowerSO20	Multiwatt15	Unit
$R_{th j-case}$	Thermal Resistance Junction-case	Max. -	3	$^\circ C/W$
$R_{th j-amb}$	Thermal Resistance Junction-ambient	Max. 13 (*)	35	$^\circ C/W$

(*) Mounted on aluminum substrate



PIN FUNCTIONS (refer to the block diagram)

MW, 15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A 100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _S	Supply Voltage (pin 4)	Operative Condition	V _{IH} +2.5		46	V
V _{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I _S	Quiescent Supply Current (pin 4)	V _{en} = H; I _L = 0	V _i = L V _i = H	13 50	22 70	mA mA
		V _{en} = L	V _i = X		4	mA
I _{SS}	Quiescent Current from V _{SS} (pin 9)	V _{en} = H; I _L = 0	V _i = L V _i = H	24 7	36 12	mA mA
		V _{en} = L	V _i = X		6	mA
V _{iL}	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V _{iH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V _i = L			-10	μA
I _{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	V _i = H ≤ V _{SS} - 0.6V		30	100	μA
V _{en} = L	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
V _{en} = H	Enable High Voltage (pins 6, 11)		2.3		V _{SS}	V
I _{en} = L	Low Voltage Enable Current (pins 6, 11)	V _{en} = L			-10	μA
I _{en} = H	High Voltage Enable Current (pins 6, 11)	V _{en} = H ≤ V _{SS} - 0.6V		30	100	μA
V _{CEsat(H)}	Source Saturation Voltage	I _L = 1A I _L = 2A	0.95	1.35 2	1.7 2.7	V V
V _{CEsat(L)}	Sink Saturation Voltage	I _L = 1A (5) I _L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V _{CEsat}	Total Drop	I _L = 1A (5) I _L = 2A (5)	1.80		3.2 4.9	V V
V _{sens}	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T ₁ (V _i)	Source Current Turn-off Delay	0.5 V _i to 0.9 I _L (2); (4)		1.5		μs
T ₂ (V _i)	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		0.2		μs
T ₃ (V _i)	Source Current Turn-on Delay	0.5 V _i to 0.1 I _L (2); (4)		2		μs
T ₄ (V _i)	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.7		μs
T ₅ (V _i)	Sink Current Turn-off Delay	0.5 V _i to 0.9 I _L (3); (4)		0.7		μs
T ₆ (V _i)	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.25		μs
T ₇ (V _i)	Sink Current Turn-on Delay	0.5 V _i to 0.9 I _L (3); (4)		1.6		μs
T ₈ (V _i)	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.2		μs
f _c (V _i)	Commutation Frequency	I _L = 2A		25	40	KHz
T ₁ (V _{en})	Source Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (2); (4)		3		μs
T ₂ (V _{en})	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		1		μs
T ₃ (V _{en})	Source Current Turn-on Delay	0.5 V _{en} to 0.1 I _L (2); (4)		0.3		μs
T ₄ (V _{en})	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.4		μs
T ₅ (V _{en})	Sink Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (3); (4)		2.2		μs
T ₆ (V _{en})	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.35		μs
T ₇ (V _{en})	Sink Current Turn-on Delay	0.5 V _{en} to 0.9 I _L (3); (4)		0.25		μs
T ₈ (V _{en})	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.1		μs

- 1) Sensing voltage can be -1 V for t ≤ 50 μsec; in steady state V_{sens} min ≥ -0.5 V.
- 2) See fig. 2.
- 3) See fig. 4.
- 4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

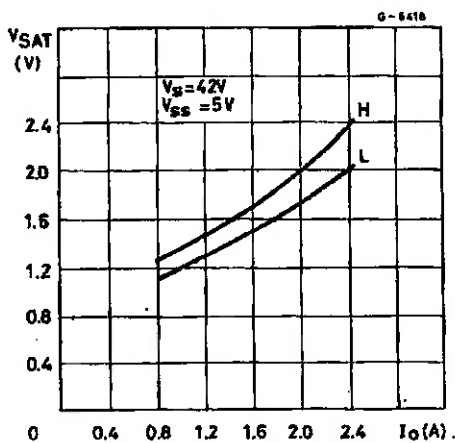
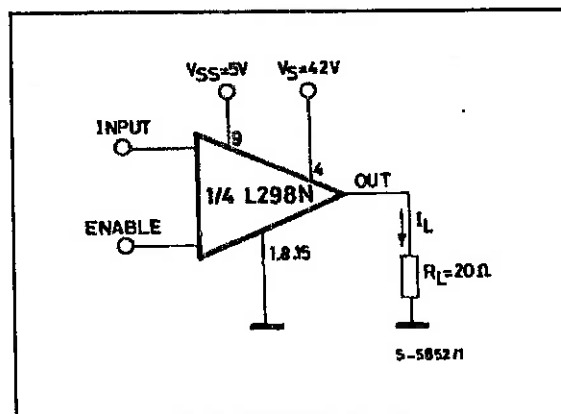


Figure 2 : Switching Times Test Circuits.



Note: For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

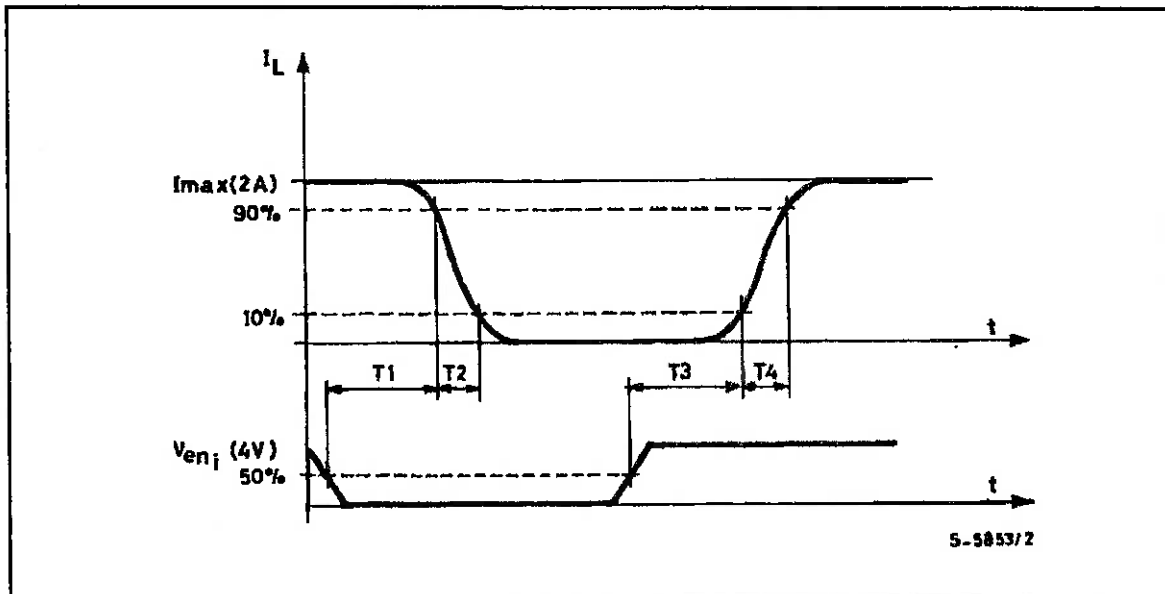
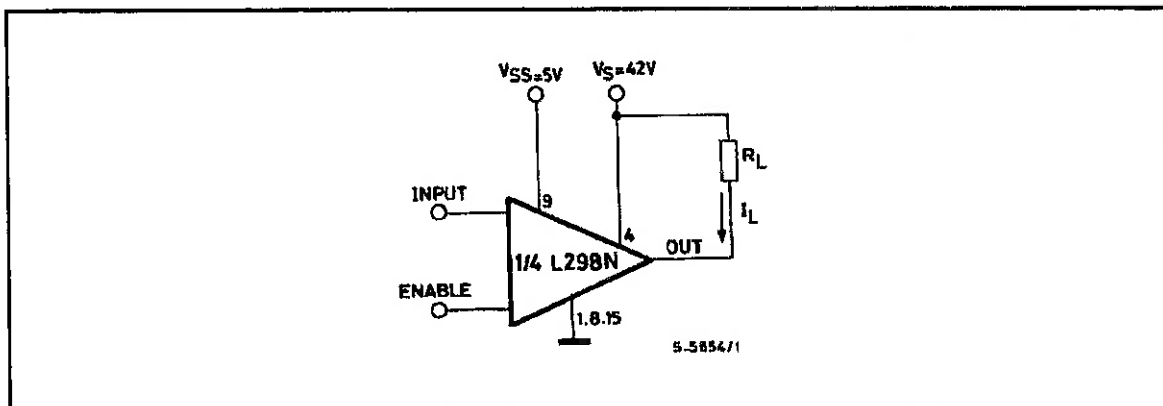


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
 For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

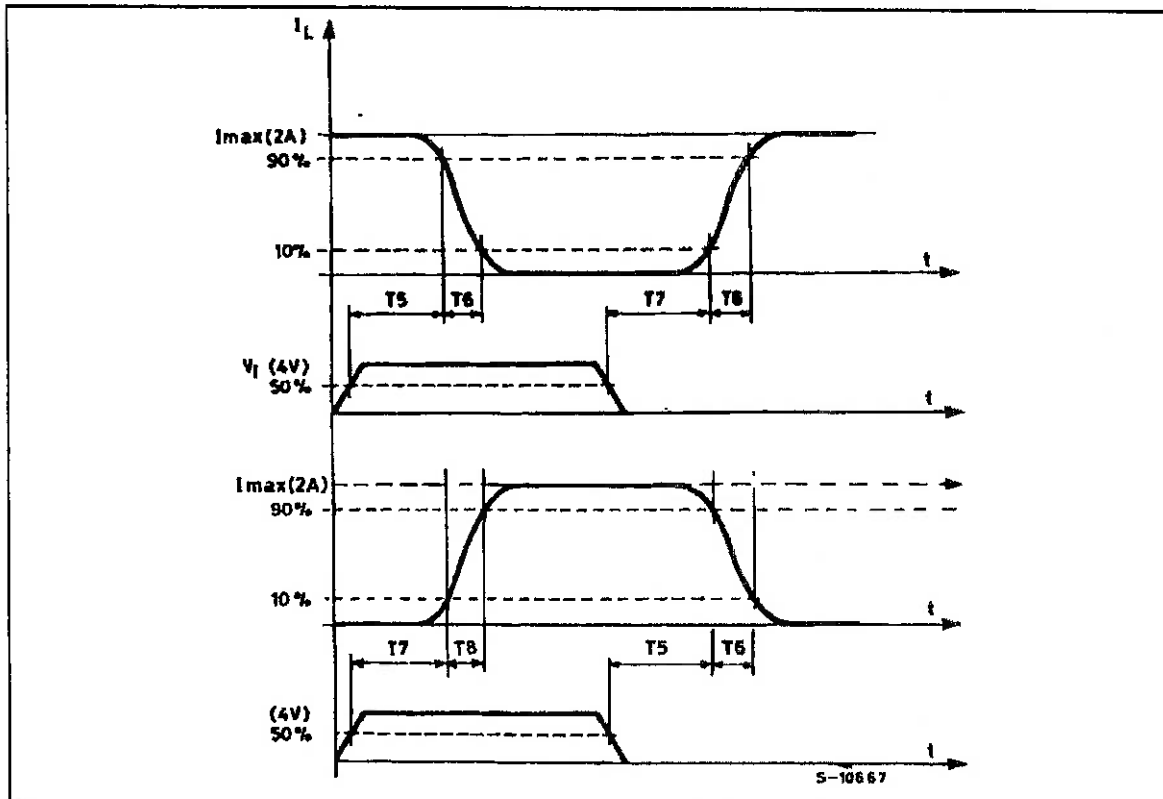


Figure 6 : Bidirectional DC Motor Control.

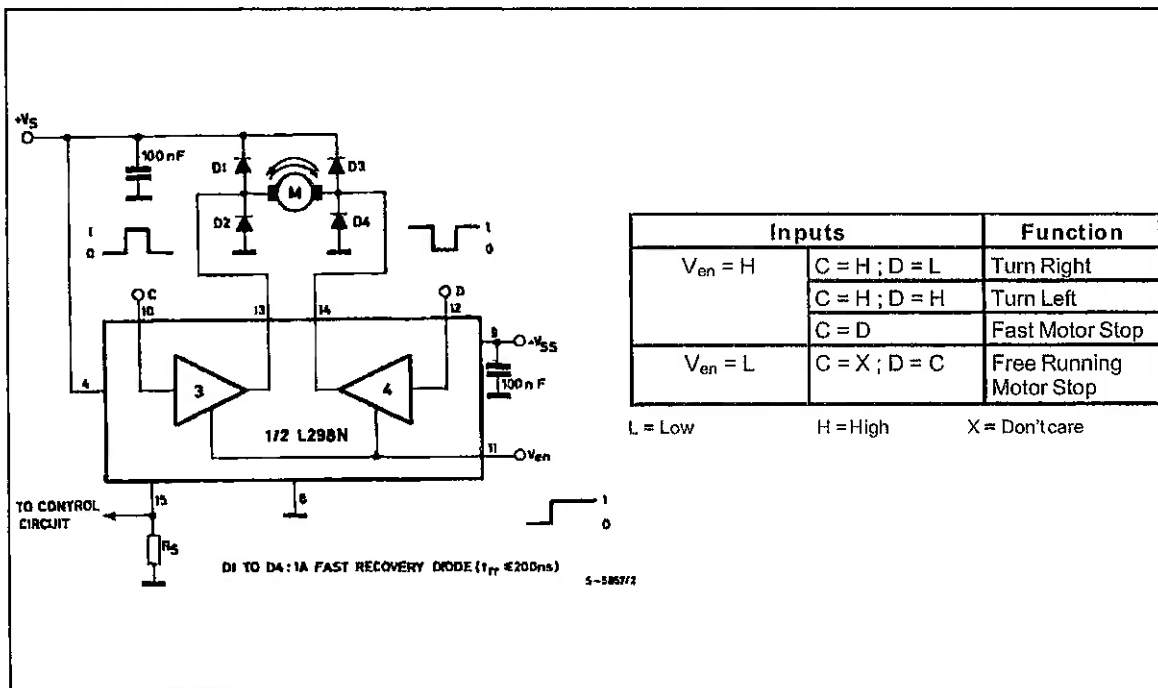
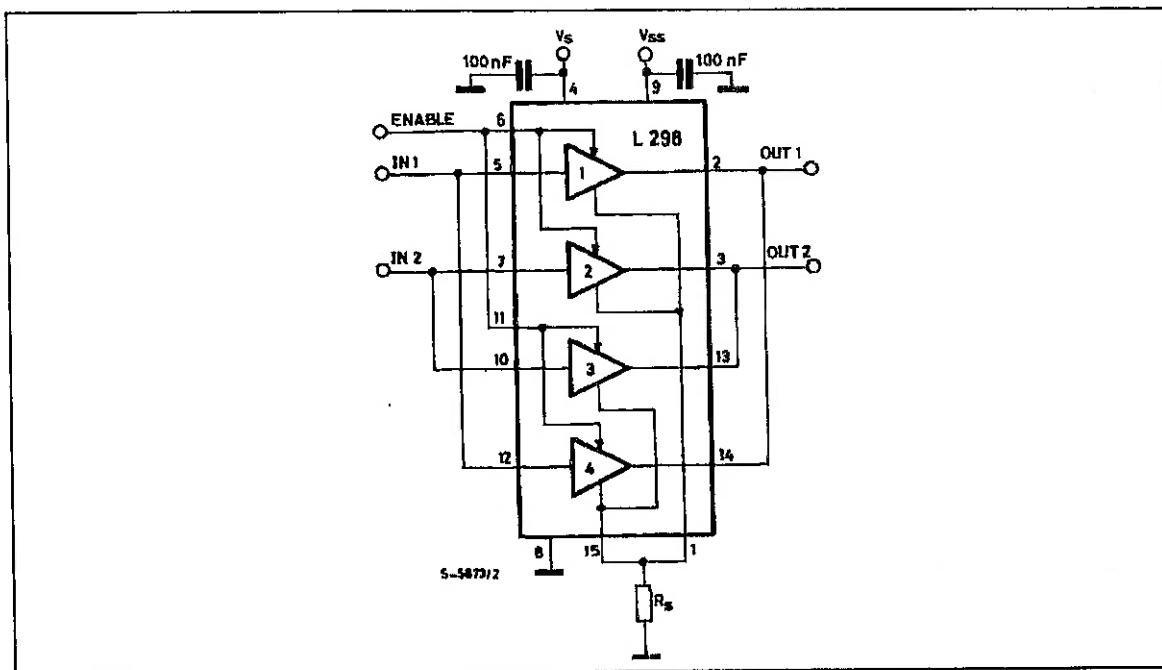


Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differential mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output: an external resistor (R_{SA} ; R_{SB}) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are In_1 ; In_2 ; EnA and In_3 ; In_4 ; EnB . The In inputs set the bridge state when The En input is high; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both V_s and V_{ss} , to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of V_s that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off: Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements ($t_{rr} \leq 200$ nsec) that must be chosen of a VF as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped; Schottky diodes would be preferred.

L298

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.

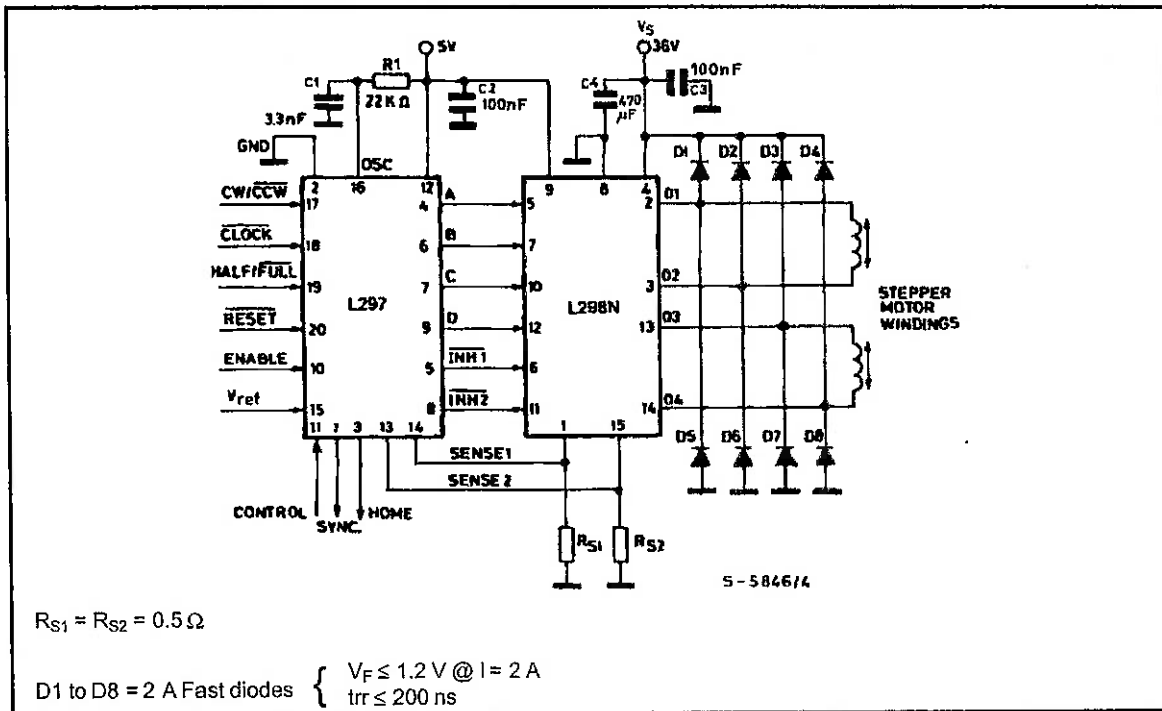


Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

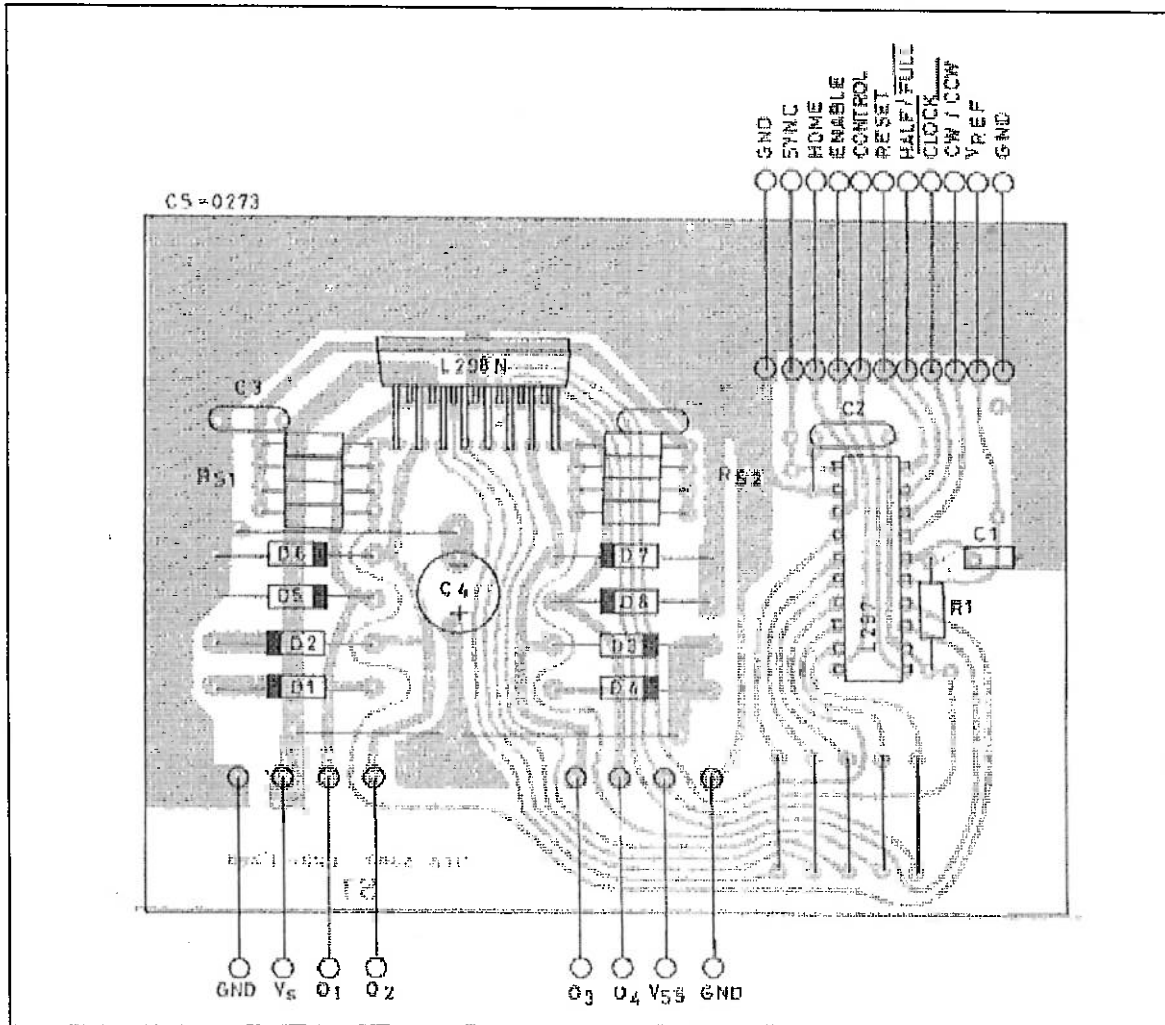
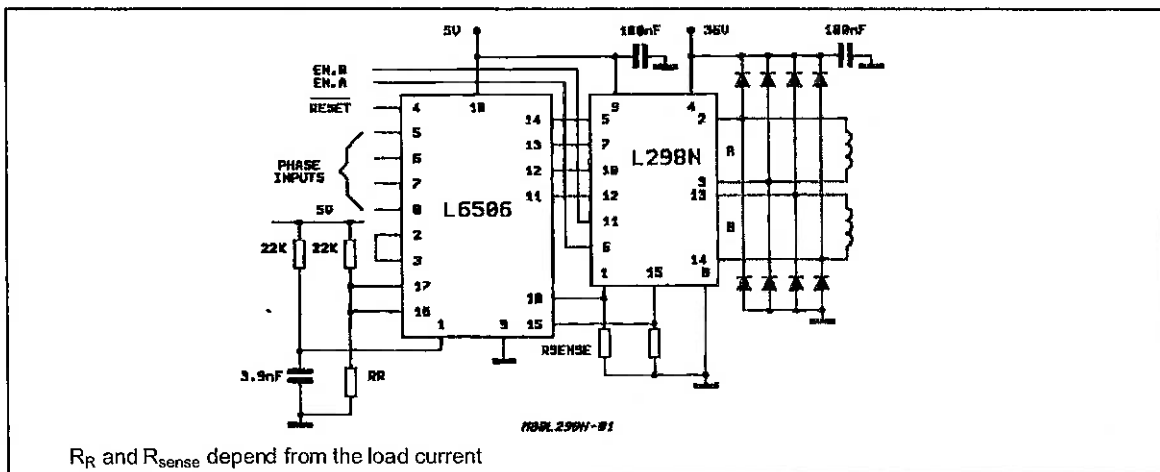
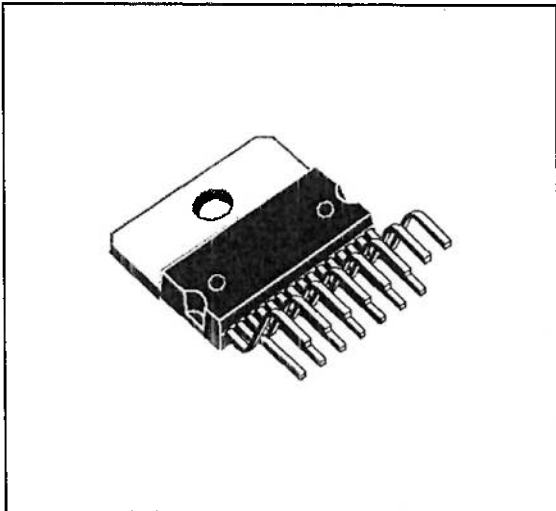


Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.

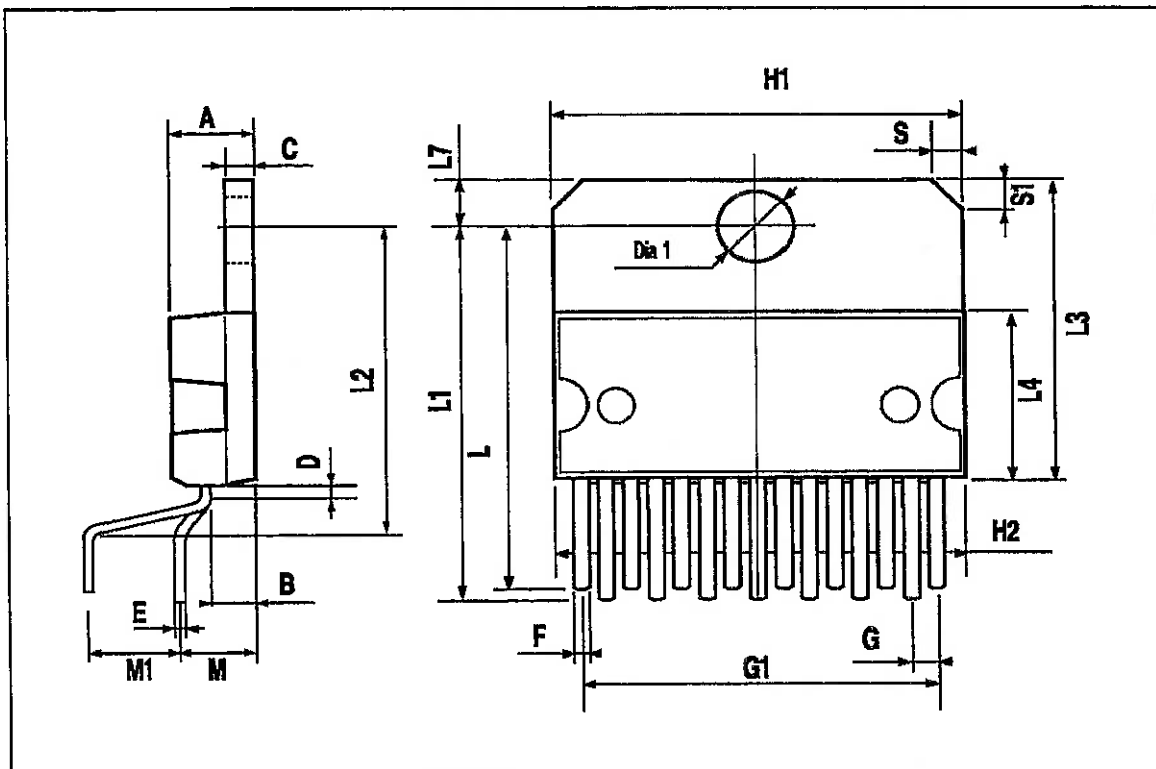


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA

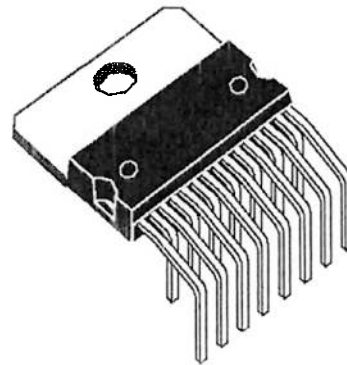


Multiwatt15 V

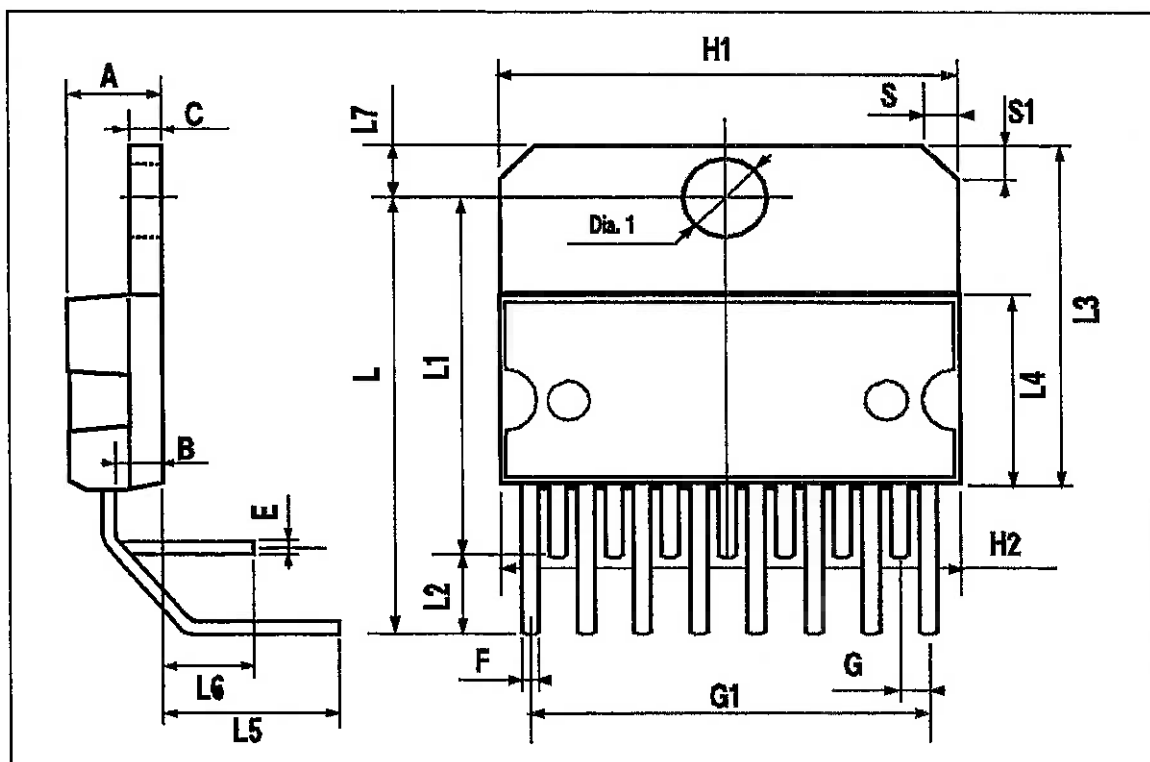


DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



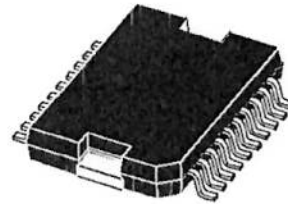
Multiwatt15 H



DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10° (max.)					
S	8° (max.)					
T		10			0.394	

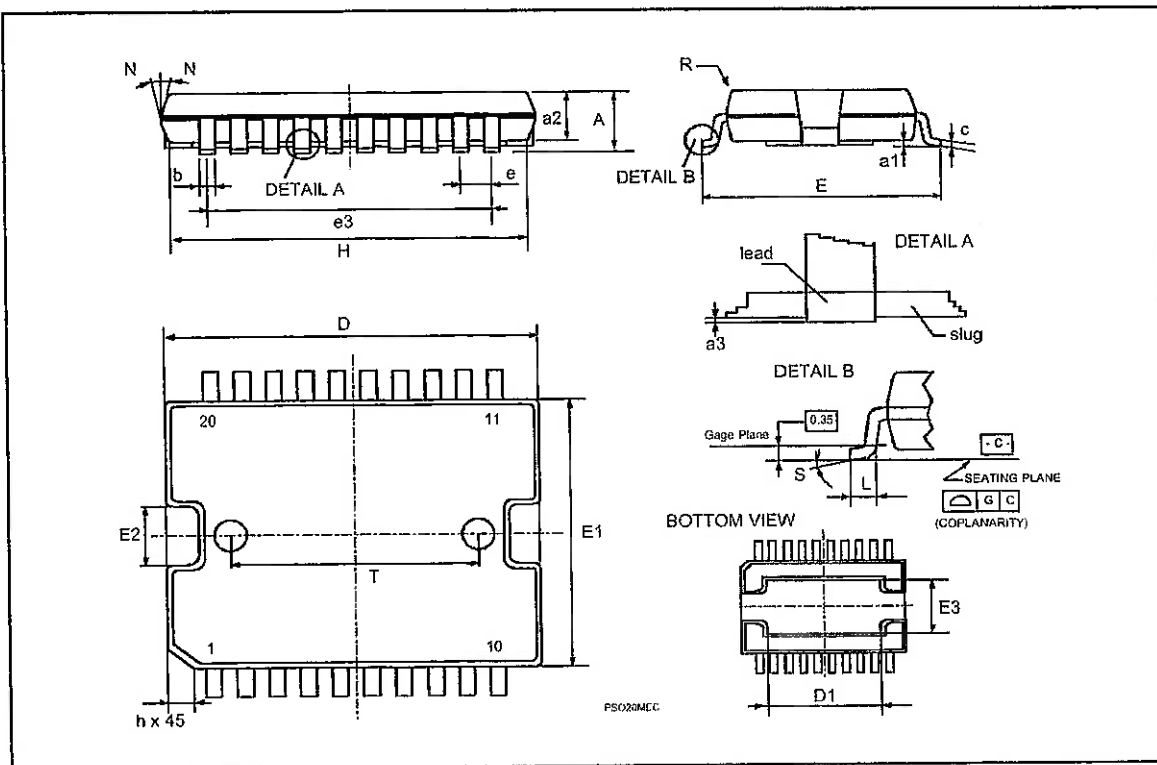
(1) "D and F" do not include mold flash or protrusions.
 - Mold flash or protrusions shall not exceed 0.15 mm (0.006").
 - Critical dimensions: "E", "G" and "a3"

OUTLINE AND MECHANICAL DATA



JEDEC MO-166

PowerSO20



Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics
© 1998 STMicroelectronics – Printed in Italy – All Rights Reserved
STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - Canada - China - France - Germany - Italy - Japan - Korea - Malaysia - Malta - Mexico - Morocco - The Netherlands -
Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.
<http://www.st.com>

**ANEXO C – CÓDIGO-FONTE DO AMBIENTE
VIRTUAL**

Arquivo frmPrincipal.frm

Option Explicit

```

' Public Estavel As Boolean
' Public inTempl As Integer
' Public inTempl_2 As Integer
' Public inTemp2 As Integer
' Public inTemp2_2 As Integer

Dim objetoMesh As D3DXMesh ' Objeto da classe D3DXMesh, será o objeto do ambiente
                             ' virtual
Dim fl_Init As Boolean ' Indica que o D3d foi inicializado
Dim fl_Minimized As Boolean ' Indica se a janela de visualização está
                             ' minimizada

Dim fl_Tecla(256) As Boolean ' Indica se a tecla foi pressionada, tem como
                             ' parametro o valor ASCII do caracter
Dim fl_Load As Boolean ' Indica que a inicialização do ambiente foi
                             ' realizada com sucesso

' O programa é inicializado nesta função
Private Sub Form_Load()
On Error GoTo Erro

    ' Variáveis de inicialização do sistema
    dimensao = 2
    delta = 0.01
    ViewPoint = vec3(0, 0, -1)
    CameraPoint = vec3(0, 0, 0)

    ' Mostra a janela principal
    Me.Show

    ' Deixa as mensagens do Windows serem executadas
    DoEvents

    ' Formata o título da janela
    Me.Caption = "Trabalho de Formatura: Joystick Sensitivo para Interação com " & _
                "Ambiente Virtual"

    ' Inicializa o DixectX
    Init

    ' Inicia os timers
    Call DXUtil_Timer(TIMER_start)
    Timer1.Enabled = True

    ' Encerra a função
    Exit Sub

Erro:
    MsgBox Err.Description
End Sub

' Processa os eventos de tecla pressionada
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)

    fl_Tecla(KeyCode) = True

    ' Verifica a tecla pressionada
    Select Case KeyCode
    ' Caso a tecla pressionada for o Esc, finaliza o programa
    Case vbKeyEscape: Out IoData2, 0: Unload Me

    ' Caso a tecla F2 for pressionada, para o timer e carrega o formulário para seleção
    ' dos Devices
    Case vbKeyF2
        DXUtil_Timer TIMER_STOP
        Menu_Reinicia_Click
        DXUtil_Timer TIMER_start

```

```
        RestoreDeviceObjects
    End Select

End Sub

' Processa os eventos de tecla liberada
Private Sub Form_KeyUp(KeyCode As Integer, Shift As Integer)
    fl_Tecla(KeyCode) = False
End Sub

' Processa os eventos de clique do mouse
Private Sub Form_MouseDown(Button As Integer, _
    Shift As Integer, x As Single, y As Single)
    ' Se o botão for o direito, mostrar o menu
    If Button = 2 Then Me.PopupMenu MENU_POPUP
End Sub

' Trata a midança no tamanho da janela
Private Sub Form_Resize()
    ' Se o D3D não tiver iniciado, sai do método
    If Not fl_Init Then Exit Sub

    ' Se o programa estiver minimizado, pára o timer e sai do método
    If Me.WindowState = vbMinimized Then
        DXUtil_Timer TIMER_STOP
        fl_Minimized = True
        Exit Sub
    ' Se maximizar a tela, reinicia o timer
    Else
        If fl_Minimized = True Then
            DXUtil_Timer TIMER_start
            fl_Minimized = False
        End If
    End If

    ' Não deixa a janela ficar muito pequena
    If Me.ScaleWidth < 10 Then
        Me.Width = Screen.TwipsPerPixelX * 10
        Exit Sub
    End If
    If Me.ScaleHeight < 10 Then
        Me.Height = Screen.TwipsPerPixelY * 10
        Exit Sub
    End If

    ' Reinicia e redimensiona o backbuffer para o tamanho da janela
    D3DUtil_ResizeWindowed Me.hwnd

    ' Como todas as propriedades se perdem quando se reinicia, reconfigura as variáveis
    g_lWindowWidth = Me.ScaleWidth
    g_lWindowHeight = Me.ScaleHeight

    RestoreDeviceObjects
End Sub

Public Sub Init()

    ' Chama a função de inicialização do módulo D3DUtil
    fl_Init = D3DUtil_Init(Me.hwnd, True, 0, 0, D3DDEVTYPE_HAL, Nothing)

    ' Se houver falha na inicialização, termina o programa
    If fl_Init = False Then End

    fl_Load = False

    ' Deleta os objetos dos devices
    DeleteDeviceObjects

    ' Inicia os objetos dos devices
    InitDeviceObjects
```

```
' Inicia as fontes
InitFont

' Reatura os objetos dos devices
RestoreDeviceObjects

' Se tudo foi carregado com sucesso, configura a variável para verdadeiro
fi_Load = True

' Deixa as mensagens do Windows serem executadas
DoEvents

End Sub

' Chamado quando a aplicação está sendo terminada, ou o device está sendo mudado, esta
' função deleta qualquer objeto dependente do device
Sub DeleteDeviceObjects()
' Limpa a variável
Set objetoMesh = Nothing
End Sub

' Inicializa os objetos da cena
Function InitDeviceObjects() As Boolean
On Error GoTo Erro

' Cria o objeto com as dimensões especificadas
Set objetoMesh = g_d3dx.CreateBox(g_dev, dimensao, dimensao, dimensao, Nothing)

Exit Function

Erro:
MsgBox "Não foi possível carregar o Objeto!"
End Function

' Inicia a fonte do programa para escrita na tela
Public Sub InitFont()
Dim fnt As StdFont
Set fnt = New StdFont

Dim fntDesc As IFont
Set fntDesc = fnt

' Tipo da fonte
fnt.Name = "Arial"

' Tamanho da fonte
fnt.Size = 8

' Negrito
fnt.Bold = True

Set fonteAmbiente = g_d3dx.CreateFont(g_dev, fntDesc.hFont)
End Sub

' Restaura objetos e estados da memoria depois que o device foi criado ou alterado.
Sub RestoreDeviceObjects()

' Configura a cena padrão
D3DUtil_SetupDefaultScene

' Configura os renderstates - a renderização dos objetos
With g_dev
.SetRenderState D3DRS_CULLMODE, D3DCULL_NONE
.SetRenderState D3DRS_RANGEFOGENABLE, 1

' Cria uma luz direcional
Dim light As D3DLIGHT8
D3DUtil_InitLight light, D3DLIGHT_DIRECTIONAL, -1, 0.5, 1
.SetLight 1, light
.LightEnable 1, 1
.SetRenderState D3DRS_LIGHTING, 1
```

```

End With
End Sub

' Ciclo que se repete a cada 10 milissegundos
Private Sub Timer1_Timer()

    ' Checa a entrada dos sinais dos sensores, caso tenha sido selecionado o joystick
    ChecaEntrada

    ' Chama a rotina de tratamento de dados
    TrataDados

    ' Envia o sinal para acionamento dos motores
    EnviaSaida IoData2, saidaLpt1

    ' Chama a rotina para movimentação do ambiente
    Movimenta

    ' Se foi possível renderizar, mostra na tela
    If Renderiza Then D3DUtil_PresentAll g_focushwnd
End Sub

' Trata os dados ou do teclado ou dos sensores do joystick
Public Sub TrataDados()
    Select Case sensor1
        Case -1
            Select Case sensor2
                Case -1 ' Os dois sensores girando para a esquerda -> Sobe
                    If Not DetectaColisao(0, 0, 1) Then
                        MudaVisao 0, 0, delta
                    Else
                        saidaLpt1 = motDesce + acTodos2
                    End If
                Case 0 ' Somente o primeiro sensor girando para a esquerda -> X Negativo
                    If Not DetectaColisao(-1, 0, 0) Then
                        MudaVisao -delta, 0, 0
                    Else
                        saidaLpt1 = motXNeg + acEixoX1
                    End If
                Case 1 ' Os dois sensores girando em sentidos opostos - Nada
                    saidaLpt1 = motNeutro
            End Select
        Case 0
            Select Case sensor2
                Case -1 ' Somente o segundo sensor girando para a esquerda -> Y Negativo
                    If Not DetectaColisao(0, 1, 0) Then
                        MudaVisao 0, delta, 0
                    Else
                        saidaLpt1 = motYNeg + acEixoY2
                    End If
                Case 0 ' Os dois sensores parados - Nada
                    saidaLpt1 = motNeutro
                Case 1 ' Somente o segundo sensor girando para a direita -> Y Positivo
                    If Not DetectaColisao(0, -1, 0) Then
                        MudaVisao 0, -delta, 0
                    Else
                        saidaLpt1 = motYPos + acEixoY5
                    End If
            End Select
        Case 1
            Select Case sensor2
                Case -1 ' Os dois sensores girando em sentidos opostos - Nada
                    saidaLpt1 = motNeutro
                Case 0 ' Somente o primeiro sensor girando para a direita -> X Positivo
                    If Not DetectaColisao(1, 0, 0) Then
                        MudaVisao delta, 0, 0
                    Else
                        saidaLpt1 = motXPos + acEixoX5
                    End If
                Case 1 ' Os dois sensores girando para a direita -> Desce
                    If Not DetectaColisao(0, 0, -1) Then
                        MudaVisao 0, 0, -delta
                    Else

```

```
        saidaLpt1 = motSobe + acTodos2
    End If
End Select
End Select
End Sub

' Faz a movimentação do ambiente virtual - usa matrizes de transformação
Sub Movimenta()
On Error GoTo Erro

    ' Cria uma matriz a partir dos dados da camera e da visao
    D3DXMatrixLookAtLH g_viewMatrix, CameraPoint, ViewPoint, vec3(0, 1, 0)

    Exit Sub

Erro:
    MsgBox Err.Description
End Sub

' Chamada uma vez por frame, esta função configura o resder state, limpa o viewport e
' renderiza a cena.
Function Renderiza() As Boolean
On Error GoTo Erro
    Dim hr As Long

    Renderiza = False

    ' Verifica o estado do device
    hr = g_dev.TestCooperativeLevel
    If hr = D3DErr_DeviceNotReset Then
        g_dev.Reset g_d3dpp
        RestoreDeviceObjects
    End If

    ' Não renderiza se o device não está preparado
    If hr <> 0 Then Exit Function
    Renderiza = True

    ' Limpa o backbuffer
    D3DUtil_ClearAll &HFF&

    With g_dev
        ' Faz a transformação
        .SetTransform D3DTS_VIEW, g_viewMatrix

        .BeginScene

        ' Desenha o objeto
        objetoMesh.DrawSubset 0

        ' Desenha a posição do cursor
        DesenhaPosicao

        ' Desenha o cursor
        DesenhaCursor

        ' Desenha o status da colisão
        DesenhaColisao direcao, sentido

        .EndScene
    End With
    direcao = ""
    sentido = ""

    Exit Function

Erro:
    MsgBox Err.Description
End Function

' Escreve na tela a posição atual do cursor
Public Sub DesenhaPosicao()
    Dim TextRect As RECT
```

```

' Parametros de localização do texto
TextRect.Top = 1
TextRect.Left = 1
TextRect.bottom = 16
TextRect.Right = 300
' Escreve na tela
g_d3dx.DrawText fonteAmbiente, &HFFFFFF0000, _
    "Posição Cursor: " & " X=" & FormatNumber(CameraPoint.x, 2) & _
    " Y=" & FormatNumber(CameraPoint.y, 2) & _
    " Z=" & FormatNumber(CameraPoint.z, 2), _
    TextRect, DT_TOP Or DT_LEFT
End Sub

' Escreve na tela o cursor
Public Sub DesenhaCursor()
    Dim TextRect As RECT

    ' Parametros de localização do texto
    TextRect.Top = 358
    TextRect.Left = 510
    TextRect.bottom = 400
    TextRect.Right = 530
    ' Escreve na tela
    g_d3dx.DrawText fonteAmbiente, &HFFFFFF0000, "|", TextRect, DT_TOP Or DT_LEFT

    ' Parametros de localização do texto
    TextRect.Top = 363
    TextRect.Left = 510
    TextRect.bottom = 400
    TextRect.Right = 530
    ' Escreve na tela
    g_d3dx.DrawText fonteAmbiente, &HFFFFFF0000, "|", TextRect, DT_TOP Or DT_LEFT

    ' Parametros de localização do texto
    TextRect.Top = 368
    TextRect.Left = 510
    TextRect.bottom = 400
    TextRect.Right = 530
    ' Escreve na tela
    g_d3dx.DrawText fonteAmbiente, &HFFFFFF0000, "|", TextRect, DT_TOP Or DT_LEFT

    ' Parametros de localização do texto
    TextRect.Top = 358
    TextRect.Left = 503
    TextRect.bottom = 400
    TextRect.Right = 540
    ' Escreve na tela
    g_d3dx.DrawText fonteAmbiente, &HFFFFFF0000, "___", TextRect, DT_TOP Or DT_LEFT
End Sub

' Escreve na tela o status da colisão
Public Sub DesenhaColisao(dir As String, sen As String)
    Dim TextRect As RECT

    ' Parametros de localização do texto
    TextRect.Top = 1
    TextRect.Left = 600
    TextRect.bottom = 50
    TextRect.Right = 900
    ' Escreve na tela
    g_d3dx.DrawText fonteAmbiente, &HFFFFFF0000, "Colisão -> Direção: " & dir & _
    " -- Sentido: " & sen, _
    TextRect, DT_TOP Or DT_LEFT
End Sub

' Verifica se houve a colisão com os objetos do ambiente
' Parametros: flX - direção e sentido do eixo X
'             flY - direção e sentido do eixo Y
'             flZ - direção e sentido do eixo Z
Public Function DetectaColisao(flX As Single, flY As Single, flZ As Single) As Boolean
    DetectaColisao = False

```

```

direcao = ""
sentido = ""
If flX <> 0 Then
    flX = flX * delta
    If CameraPoint.x + flX <= -(dimensao / 2 - delta) Then
        direcao = "X"
        sentido = "Neg"
        DetectaColisao = True
    ElseIf CameraPoint.x + flX >= (dimensao / 2 - delta) Then
        direcao = "X"
        sentido = "Pos"
        DetectaColisao = True
    End If
ElseIf flY <> 0 Then
    flY = flY * delta
    If CameraPoint.y + flY <= -(dimensao / 2 - delta) Then
        direcao = "Y"
        sentido = "Neg"
        DetectaColisao = True
    ElseIf CameraPoint.y + flY >= (dimensao / 2 - delta) Then
        direcao = "Y"
        sentido = "Pos"
        DetectaColisao = True
    End If
ElseIf flZ <> 0 Then
    flZ = flZ * delta
    If CameraPoint.z + flZ <= -(dimensao / 2 - delta) Then
        direcao = "Z"
        sentido = "Neg"
        DetectaColisao = True
    ElseIf CameraPoint.z + flZ >= (dimensao / 2 - delta) Then
        direcao = "Z"
        sentido = "Pos"
        DetectaColisao = True
    End If
End If
End Function

' Função que muda o valor do vetor da camera e da visão, de acordo com o movimento
' detectado
Public Sub MudaVisao(x As Single, y As Single, z As Single)
    ViewPoint = vec3(ViewPoint.x + x, ViewPoint.y + y, ViewPoint.z + z)
    CameraPoint = vec3(CameraPoint.x + x, CameraPoint.y + y, CameraPoint.z + z)
End Sub

' Quando o usuario clicar no menu Reiniciar Orientação, reinicia a visualização do
' ambiente
Private Sub Menu_Reinicia_Click()
    ViewPoint = vec3(0, 0, -1)
    CameraPoint = vec3(0, 0, 0)
    EnviaSaida IoData2, motNeutro
End Sub

' Função para executar métodos antes de fechar o programa
Private Sub Form_Unload(Cancel As Integer)
    D3DUtil_Destroy
    DeleteDeviceObjects

    ' Termina o programa
End
End Sub

```

Arquivo mdlDirectX.bas

```

' Short list of usefull functions
'   D3DUtil_Init           first call to framework
'   D3DUtil_LoadFromFile  loads an x-file
'   D3DUtil_SetupDefaultScene  setup a camera lights and materials
'   D3DUtil_SetupCamera    point camera
'   D3DUtil_SetupMediaPath  set directory to load textures from
'   D3DUtil_PresentAll    show graphic on the screen
'   D3DUtil_ResizeWindowed  resize for windowed modes
'   D3DUtil_ResizeFullscreen  resize to fullscreen mode
'   D3DUtil_CreateTextureInPool  create a texture

Option Explicit

Private Declare Function GetWindowRect Lib "user32.dll" (ByVal hwnd As Long, _
    ByVal rect As RECT) As Long
Private Declare Function GetClientRect Lib "user32.dll" (ByVal hwnd As Long, _
    ByVal rect As RECT) As Long
Private Declare Function SetWindowPos Lib "user32.dll" (ByVal hwnd As Long, _
    ByVal hwndafter As Long, _
    ByVal x As Long, _
    ByVal y As Long, _
    ByVal cx As Long, _
    ByVal cy As Long, _
    ByVal options As Long) As Long
Private Declare Function SetWindowLongA Lib "user32.dll" (ByVal hwnd As Long, _
    ByVal nIndex As Long, _
    ByVal newVal As Long) As Long
Private Declare Function GetWindowLongA Lib "user32.dll" (ByVal hwnd As Long, _
    ByVal nIndex As Long) As Long

' User Defined types
' Type that hold info about a display mode
    Public Type D3DUTIL_MODEINFO
        lWidth As Long           'Screen width in this mode
        lHeight As Long         'Screen height in this mode
        format As CONST_D3DFORMAT 'Pixel format in this mode
        VertexBehavior As CONST_D3DCREATEFLAGS 'Whether this mode does SW
        ' or HW vertex processing
    End Type
' Type that hold info about a particular back buffer format
    Public Type D3DUTIL_FORMATINFO
        format As CONST_D3DFORMAT
        usage As Long
        bCanDoWindowed As Boolean
        bCanDoFullscreen As Boolean
    End Type
Public Type D3DUTIL_DEVTYPEINFO
    'List of compatible Modes for this device
    lNumModes As Long
    Modes() As D3DUTIL_MODEINFO
    'List of compatible formats for this device
    lNumFormats As Long
    FormatInfo() As D3DUTIL_FORMATINFO
    lCurrentMode As Long 'index into modes array of the current mode
End Type
' Type that holds info about adapters installed on the system
    Public Type D3DUTIL_ADAPTERINFO
        'Device data
        DeviceType As CONST_D3DDEVTYPE 'Reference, HAL
        d3dcaps As D3DCAPS8 'Caps of this device
        sDesc As String 'Name of this device
        lCanDoWindowed As Long 'Whether or not this device
        ' can work windowed
        DevTypeInfo(2) As D3DUTIL_DEVTYPEINFO 'format and display mode
        ' list for hal=1 for ref=2

        d3dai As D3DADAPTER_IDENTIFIER8
        DesktopMode As D3DDISPLAYMODE
        'CurrentState
        bWindowed As Boolean 'currently in windowed mode
        bReference As Boolean 'currently using reference rasterizer
    End Type

```

```

' see D3DUtil_Timer
Public Enum TIMER_COMMAND
    TIMER_RESET = 1           '- to reset the timer
    TIMER_start = 2          '- to start the timer
    TIMER_STOP = 3           '- to stop (or pause) the timer
    TIMER_ADVANCE = 4        '- to advance the timer by 0.1 seconds
    TIMER_GETABSOLUTETIME = 5 '- to get the absolute system time
    TIMER_GETAPPTIME = 6     '- to get the current time
    TIMER_GETELAPSEDTIME = 7 '- to get the ellapsed time
End Enum

-----
' Variáveis Globais
-----

Public g_Adapters()          As D3DUTIL_ADAPTERINFO ' Array of Adapter infos
Public g_lCurrentAdapter     As Long                ' current adapter (index into infos)
Public g_lNumAdapters        As Long                ' size of the g_Adapters array
Public g_EnumCallback        As Object              ' object that defines VerifyDevice function

Public g_dx                   As DirectX8           ' Root objects for DX8
Public g_d3dx                 As D3DX8              ' Root objects for d3dx
Public g_d3d                  As Direct3D8          ' Root object for d3d
Public g_dev                   As Direct3DDevice8   ' D3D device
Public light0                  As D3DLIGHT8         ' light type usefull in imediate pane
Public light1                  As D3DLIGHT8         ' light type usefull in imediate pane
Public fonteAmbiente          As D3DXFont           ' Objeto da classe D3DXFont

Public g_d3dCaps               As D3DCAPS8          ' Current caps of g_dev
Public g_devType               As CONST_D3DDEVTYPE  ' Current device type (hard. or soft.)
Public g_behaviorflags         As Long              ' Current VertexProcessing (hard. or soft.)
Public g_focushwnd             As Long              ' Current focus window handle
Public g_d3dpp                 As D3DPRESENT_PARAMETERS ' Current presentation parameters

Public g_lWindowWidth         As Long                ' backbuffer width of windowed state
Public g_lWindowHeight        As Long                ' backbuffer height of windowed state
Public g_WindowRect           As RECT                ' size of window (including title bar)

Public g_fov                   As Single             ' view frustrum field of view
Public g_aspect                As Single             ' view frustrum aspect ratio
Public g_znear                  As Single             ' view frustrum near plane
Public g_zfar                   As Single             ' view frustrum far plane

Public g_identityMatrix        As D3DMATRIX          ' Filled with Identity Matrix after
' D3DUtil_Init
Public g_worldMatrix            As D3DMATRIX          ' Filled with current world matrix
Public g_viewMatrix             As D3DMATRIX          ' Filled with current view matrix
Public g_projMatrix             As D3DMATRIX          ' Filled with current projection matrix

Public ViewPoint               As D3DVECTOR          ' Vetor que indica a posição da vista
Public CameraPoint             As D3DVECTOR          ' Vetor que indica a posição da câmera

Public fl_TecladoJoystick      As Boolean           ' Variável que define se o sistema será
' acionado pelo teclado ou pelo joystick
' Teclado=>false / Joystick=>>true

Public delta                    As Single            ' Menor unidade de movimentação do joystick
Public dimensao                 As Single            ' Tamanho do objeto
Public direcao                  As String            ' Variável que define qual a direção da movimentação
Public sentido                   As String            ' Variável que define qual o sentido da movimentação

-----
' Constantes Globais
-----

Public Const g_pi = 3.1415

' Esta função cria os objetos: DirectX8, Direct3D8, Direc3DDevice8.
'
' Parametros:
' bWindowed      Fullscreen or windowed
' hwnd           Starting hwnd to display graphics in This can be changed with a call
'                to Reset if drawing to multiple windows For full screen operation
'                the hwnd will have to belong to a top level window

```

```

' AdapterIndex      The index of the display card to be used (most often 0)
' ModeIndex        Ignored if bWindowed is TRUE Otherwise and index into the
'                  g_adapters(AdapterIndex).Modes array for a given width height and
'                  backbuffer format. Use D3DUtil_FindDisplayMode to obtain an mode
'                  index given a desired height width and backbuffer format
' CallbackObject   Can be Nothing or an object that has implemented
'                  VerifyDevice(usageflags as long ,format as CONST_D3DFORMAT)
' Remarks: caps for the device are passed to VerifyDevice in the g_d3dcaps global
Public Function D3DUtil_Init(hwnd As Long, bWindowed As Boolean, AdapterIndex As Long, _
                        modeIndex As Long, devtype As CONST_D3DDEVTYPE, _
                        CallbackObject As Object) As Boolean

    On Local Error GoTo errOut

    ' Initialize the DirectX8 and d3dx8 objects
    If g_dx Is Nothing Then Set g_dx = New DirectX8
    If g_d3dx Is Nothing Then Set g_d3dx = New D3DX8

    ' Create the Direct3D object
    Set g_d3d = g_dx.Direct3DCreate

    ' Call the sub that builds a list of available adapters,
    ' adapter device types, and display modes
    Call D3DEnum_BuildAdapterList(CallbackObject)

    If bWindowed Then
        D3DUtil_Init = D3DUtil_InitWindowed(hwnd, AdapterIndex, devtype, True)
    Else
        D3DUtil_Init = D3DUtil_InitFullscreen(hwnd, AdapterIndex, modeIndex, _
                                                devtype, True)
    End If

    Exit Function

errOut:
    Debug.Print "Failed D3DUtil_Init"
End Function

' This function creates the following objects: DirectX8, Direct3D8, Direc3DDevice8.
'
' Params:
' hwnd            Starting hwnd to display graphics in This can be changed with a call
'                  to Reset if drawing to multiple windows For full screen operation
'                  the hwnd will have to belong to a top level window
' AdapterIndex    The index of the display card to be used (most often 0)
' DevType         Indicates if user wants HAL (hardware) or REF (software) rendering
' bTryFallbacks  True if wants function to attempt to fallback to the reference device
'                  on faulure and display dialogs to that effect
Function D3DUtil_InitWindowed(hwnd As Long, AdapterIndex As Long, _
                        devtype As CONST_D3DDEVTYPE, _
                        bTryFallbacks As Boolean) As Boolean

    On Error GoTo errOut

    'save the current adapter
    g_lCurrentAdapter = AdapterIndex

    Dim d3ddm As D3DDISPLAYMODE

    ' Initialize the present parameters structure
    ' to use 1 back buffer and a 16 bit depth buffer
    ' change the autoDepthStencilFormat if you need stencil bits
    With g_d3dpp
        .BackBufferCount = 1
        .EnableAutoDepthStencil = 1
        .AutoDepthStencilFormat = D3DFMT_D16
        .hDeviceWindow = hwnd
    End With

    g_focushwnd = hwnd

    Dim rc As RECT
    Call GetClientRect(g_focushwnd, rc)
    g_lWindowWidth = rc.Right - rc.Left
    g_lWindowHeight = rc.bottom - rc.Top
    Call GetWindowRect(g_focushwnd, g_WindowRect)

```

```

With g_Adapters(g_lCurrentAdapter)
  ' If running windowed, set the current desktop format
  ' as the format the device will use.
  ' note the the current mode and backbuffer width and height
  ' information is ignored by d3d
  Call g_d3d.GetAdapterDisplayMode(g_lCurrentAdapter, d3ddm)

  ' figure out if this format supports hardware acceleration
  ' by looking it up in our format list
  g_behaviorflags = D3DEnum_FindInFormatList(g_lCurrentAdapter, devtype, _
      d3ddm.format)

  If g_behaviorflags <= 0 Then g_behaviorflags = _
      D3DCREATE_SOFTWARE_VERTEXPROCESSING

  g_d3dpp.BackBufferFormat = d3ddm.format
  g_d3dpp.BackBufferWidth = 0
  g_d3dpp.BackBufferHeight = 0

  g_d3dpp.Windowed = 1
  g_d3dpp.SwapEffect = D3DSWAPEFFECT_DISCARD

  .bWindowed = True
  .DeviceType = devtype

  g_devType = devtype
End With

'Try to create the device now that we have everything set.
On Local Error Resume Next
Set g_dev = g_d3d.CreateDevice(g_lCurrentAdapter, devtype, g_focushwnd, _
    g_behaviorflags, g_d3dpp)

If Err.Number Then

  If bTryFallbacks = False Then Exit Function

  'If a HAL device was being attempted, try again with a REF device instead.
  If g_devType = D3DDEVTYPE_HAL Then
    Err.Clear

    'Make sure the user knows that this is less than an optimal 3D environment.
    MsgBox "No hardware support found. Switching to reference rasterizer.", _
        vbInformation

    'reset our variable to use ref
    g_Adapters(g_lCurrentAdapter).DeviceType = D3DDEVTYPE_REF
    g_Adapters(g_lCurrentAdapter).bReference = True
    g_devType = D3DDEVTYPE_REF
    Set g_dev = g_d3d.CreateDevice(g_lCurrentAdapter, D3DDEVTYPE_REF, _
        g_focushwnd, g_behaviorflags, g_d3dpp)

  End If
End If

If Err.Number Then
  'The app still hit an error. Both HAL and REF devices weren't created. The app
  ' will have to exit at this point.
  MsgBox "No suitable device was found to initialize D3D. " & _
      "Application will now exit.", vbCritical
  D3DUtil_InitWindowed = False
  End
  Exit Function
End If

'update our device caps data
g_dev.GetDeviceCaps g_d3dCaps

'set any state we need to initialize
D3DXMatrixIdentity g_identityMatrix

'set the reference flag if we choose a ref device
With g_Adapters(g_lCurrentAdapter)
  If .DeviceType = D3DDEVTYPE_REF Then
    .bReference = True
  Else

```

```

        .bReference = False
    End If
End With

D3DUtil_InitWindowed = True
Exit Function

errOut:
    Debug.Print "Failed D3DUtil_Init"

End Function

' This function creates the following objects: DirectX8, Direct3D8,
' Direc3DDevice8.
'
' Params:
' hwnd           Starting hwnd to display graphics in This can be changed with a call
'                 to Reset if drawing to multiple windows For full screen operation
'                 the hwnd will have to belong to a top level window
' AdapterIndex   The index of the display card to be used (most often 0)
' ModeIndex      index into the g_adapters(AdapterIndex).Modes for width
'                 height and format
' DevType        Indicates if user wants HAL (hardware) or REF (software) rendering
' bTryFallbacks  True if wants function to attempt to fallback to the reference device
'                 on failure and display dialogs to that effect
Function D3DUtil_InitFullscreen(hwnd As Long, AdapterIndex As Long, modeIndex As Long, _
                                devtype As CONST_D3DDEVTYPE, _
                                bTryFallbacks As Boolean) As Boolean

    On Error GoTo errOut

    Dim ModeInfo As D3DUTIL_MODEINFO
    Dim rc As RECT

    'save the current adapter
    g_lCurrentAdapter = AdapterIndex

    g_focushwnd = hwnd

    'switching from windowed to fullscreen so save height and width
    If g_d3dpp.Windowed = 1 Then
        Call GetClientRect(g_focushwnd, rc)
        g_lWindowWidth = rc.Right - rc.Left
        g_lWindowHeight = rc.bottom - rc.Top
        Call GetWindowRect(g_focushwnd, g_WindowRect)
    End If

    ' Initialize the present parameters structure to use 1 back buffer and a 16 bit
    ' depth buffer change the autoDepthStencilFormat if you need stencil bits
    With g_d3dpp
        .BackBufferCount = 1
        .EnableAutoDepthStencil = 1
        .AutoDepthStencilFormat = D3DFMT_D16
        .hDeviceWindow = g_focushwnd
    End With

    With g_Adapters(g_lCurrentAdapter)
        With .DevTypeInfo(devtype)
            ModeInfo = .Modes(modeIndex)
            g_behaviorflags = .Modes(modeIndex).VertexBehavior

            g_d3dpp.BackBufferWidth = ModeInfo.lWidth
            g_d3dpp.BackBufferHeight = ModeInfo.lHeight
            g_d3dpp.BackBufferFormat = ModeInfo.format
            g_d3dpp.SwapEffect = D3DSWAPEFFECT_FLIP
            g_d3dpp.Windowed = 0
            .lCurrentMode = modeIndex
        End With

        .bWindowed = False
        .DeviceType = devtype

        If g_behaviorflags = 0 Then g_behaviorflags =
D3DCREATE_SOFTWARE_VERTEXPROCESSING
        g_devtype = devtype
    End With
End Function

```

```

End With

'Try to create the device now that we have everything set.
On Local Error Resume Next
Set g_dev = g_d3d.CreateDevice(g_lCurrentAdapter, devtype, g_focushwnd, _
                             g_behaviorflags, g_d3dpp)

If Err.Number Then
    If bTryFallbacks = False Then Exit Function

    'If a HAL device was being attempted, try again with a REF device instead.
    If g_devType = D3DDEVTYPE_HAL Then
        Err.Clear
        'Make sure the user knows that this is less than an optimal 3D environment.
        MsgBox "No hardware support found. Switching to reference rasterizer.", _
            vbInformation
        'reset our variable to use ref
        g_Adapters(g_lCurrentAdapter).DeviceType = D3DDEVTYPE_REF
        g_devType = D3DDEVTYPE_REF
        Set g_dev = g_d3d.CreateDevice(g_lCurrentAdapter, D3DDEVTYPE_REF, _
            g_focushwnd, g_behaviorflags, g_d3dpp)
    End If
End If

If Err.Number Then
    'The app still hit an error. Both HAL and REF devices weren't created.
    ' The app will have to exit at this point.
    MsgBox "No suitable device was found to initialize D3D. " & _
        "Application will now exit.", vbCritical
    D3DUtil_InitFullscreen = False
    End
    Exit Function
End If

'update our device caps data
g_dev.GetDeviceCaps g_d3dCaps

'set any state we need to initialize
D3DXMatrixIdentity g_identityMatrix

'set the reference flag if we choose a ref device
With g_Adapters(g_lCurrentAdapter)
    If .DeviceType = D3DDEVTYPE_REF Then
        .bReference = True
    Else
        .bReference = False
    End If
End With
D3DUtil_InitFullscreen = True
Exit Function

errOut:
    Debug.Print "Failed D3DUtil_InitFullscreen"
End Function

Sub D3DUtil_InitLight(lgt As D3DLIGHT8, ty As CONST_D3DLIGHTTYPE, _
                    x As Single, y As Single, z As Single)
    lgt.Type = ty
    lgt.diffuse = ColorValue4(1, 1, 1, 1)
    lgt.Attenuation1 = 0.01
    lgt.Attenuation0 = 0
    lgt.Attenuation2 = 0
    lgt.Range = 99999999
    lgt.Direction.x = x
    lgt.Direction.y = y
    lgt.Direction.z = z
    lgt.Position.x = x
    lgt.Position.y = y
    lgt.Position.z = z
End Sub

' Helper function that initializes some default lighting and render states
,
```

```

' remarks:
' sets defaults for:
' g_fov, g_aspect, g_znear, g_zfar, g_identityMatrix, g_projMatrix,
' g_ViewMatrix, g_worldMatrix
' set device state for project view and world matrices
' set device state for 2 directional lights (0 and 1)
' set device state for a default grey material
Public Sub D3DUtil_SetupDefaultScene()

    ' Parâmetros para a visualização da tela em modo 1024x768
    g_fov = 4
    g_aspect = 768 / 1024
    g_znear = 0.0001
    g_zfar = 100

    If g_lWindowHeight <> 0 And g_lWindowWidth <> 0 Then
        g_aspect = g_lWindowHeight / g_lWindowWidth
    End If

    D3DXMatrixPerspectiveFovLH g_projMatrix, g_fov, g_aspect, g_znear, g_zfar
    g_dev.SetTransform D3DTS_PROJECTION, g_projMatrix

    D3DXMatrixLookAtLH g_viewMatrix, CameraPoint, ViewPoint, vec3(0, 1, 0)
    g_dev.SetTransform D3DTS_VIEW, g_viewMatrix

    D3DXMatrixIdentity g_identityMatrix
    g_dev.SetTransform D3DTS_WORLD, g_identityMatrix

    'default light0

    light0.Ambient = ColorValue4(1, 1, 1, 0.5)
    light0.diffuse = ColorValue4(1, 1, 1, 0.5)
    light0.Type = D3DLIGHT_DIRECTIONAL
    light0.Range = 10000
    light0.Direction.x = -1
    light0.Direction.y = -1
    light0.Direction.z = -1
    D3DXVec3Normalize light0.Direction, light0.Direction
    g_dev.SetLight 0, light0
    g_dev.LightEnable 0, 1 'true

    'default light1

    light1.Ambient = ColorValue4(1, 0.3, 0.3, 0.3)
    light1.diffuse = ColorValue4(1, 1, 1, 1)
    light1.Type = D3DLIGHT_DIRECTIONAL
    light1.Range = 10000
    light1.Direction.x = 1
    light1.Direction.y = -1
    light1.Direction.z = -1
    D3DXVec3Normalize light1.Direction, light1.Direction
    'g_dev.SetLight 1, light1
    'g_dev.LightEnable 1, 1 'true

    'set first material
    Dim material0 As D3DMATERIAL8
    material0.Ambient = ColorValue4(1, 0.2, 0.2, 0.2)
    material0.diffuse = ColorValue4(1, 0.5, 0.5, 0.5)
    material0.power = 10
    g_dev.SetMaterial material0

    With g_dev
        Call .SetRenderState(D3DRS_AMBIENT, &H10101010)
        Call .SetRenderState(D3DRS_CLIPPING, 1) 'CLIPPING IS ON
        Call .SetRenderState(D3DRS_LIGHTING, 1) 'LIGHTING IS ON
        Call .SetRenderState(D3DRS_ZENABLE, 1) 'USE ZBUFFER
        Call .SetRenderState(D3DRS_SHADEMODE, D3DSHADE_GOURAUD)
    End With

End Sub

' Params
' fromV world space vector of camera position

```

```

'   toV      world space vector of position camera is looking toward
'   upV      world space vector of cameras up direction
' Remarks
'   effects g_viewMatrix and device ViewMatrix state
'   Make sure upV is different than the direction of sight
Sub D3DUtil_SetupCamera(fromV As D3DVECTOR, toV As D3DVECTOR, upV As D3DVECTOR)
    D3DXMatrixLookAtLH g_viewMatrix, fromV, toV, upV
    g_dev.SetTransform D3DTS_VIEW, g_viewMatrix
End Sub

'   hWnd device window
'DOCL
' Remarks
'   use when already in windowed mode to resize the backbuffer
'   do not use to switch from fullscreen to windowed mode
Function D3DUtil_ResizeWindowed(hWnd As Long) As Boolean
    On Local Error GoTo errOut

    Dim d3dpp As D3DPRESENT_PARAMETERS
    Dim rc As RECT

    d3dpp = g_d3dpp

    If d3dpp.Windowed = 0 Then Exit Function

    g_focushwnd = hWnd
    Call GetClientRect(g_focushwnd, rc)
    g_lWindowWidth = rc.Right - rc.Left
    g_lWindowHeight = rc.bottom - rc.Top
    Call GetWindowRect(g_focushwnd, g_WindowRect)

    d3dpp.BackBufferWidth = 0 'g_lWindowWidth
    d3dpp.BackBufferHeight = 0 'g_lWindowHeight
    d3dpp.hDeviceWindow = hWnd
    d3dpp.Windowed = 1

    g_dev.Reset d3dpp

    g_d3dpp = d3dpp

    g_Adapters(g_lCurrentAdapter).bWindowed = True

    D3DUtil_ResizeWindowed = True
    Exit Function

errOut:
    Debug.Print "D3DUtil_ResizeWindowed failed-make sure width and height are in pixels"
    If Err.Number <> 0 Then 'The call to reset failed.
        D3DUtil_ResizeWindowed = False
        MsgBox "Could not reset the Direct3D Device." & vbCrLf & _
            "This sample will now exit.", vbOKOnly Or vbCritical, "Failure"
    End
    Exit Function
End If
End Function

' Params
'   col      color to clear the backbuffer
' Remarks
'   convenience function that assumes a rendertarget with a zbuffer and
'   no stencil
Function D3DUtil_ClearAll(col As Long)
    g_dev.Clear 0, ByVal 0, D3DCLEAR_TARGET Or D3DCLEAR_ZBUFFER, col, 1, 0
End Function

' Params
'   hWnd     hWnd to present to
' Remarks
'   Convenience function that presents the contents of a backbuffer to an hWnd
Function D3DUtil_PresentAll(hWnd As Long)
    On Local Error Resume Next
    g_dev.Present ByVal 0, ByVal 0, hWnd, ByVal 0

```

```

'some video cards leave the divide by zero flag set after Present or EndScene
'here we force a divide by zero to force vb to reset the flag so the next
'math operation doesnt give a divide by zero error
Dim dummy As Single
'  dummy = dummy / 0
Err.Clear
End Function

' Used to intialized a list of valid adapters and display modes
'
' Params:
' EnumCallback - can be Nothing or an object that has implemented
'               VerifyDevice(usageflags as long ,format as CONST_D3DFORMAT)
'               ussgeflags can be
'               D3DCREATE_SOFTWARE_VERTEXPROCESSING
'               D3DCREATE_HARDWARE_VERTEXPROCESSING
' Remarks:
' caps for the device are passed to VerifyDevice in the g_d3dcaps global
Public Function D3DEnum_BuildAdapterList(EnumCallback As Object) As Boolean
    On Local Error GoTo errOut

    Dim lAdapter As Long

    ' empty the list
    Call D3DEnum_Cleanup
    ' create d3d and dx objects if not already created
    If g_dx Is Nothing Then Set g_dx = New DirectX8
    If g_d3d Is Nothing Then Set g_d3d = g_dx.Direct3DCreate
    If g_d3dx Is Nothing Then Set g_d3dx = New D3DX8
    ' save callback
    Set g_EnumCallback = EnumCallback
    ' Make space for new adapter
    g_lNumAdapters = g_d3d.GetAdapterCount
    ReDim g_Adapters(g_lNumAdapters)
    ' Loop through all the adapters on the system
    For lAdapter = 0 To g_lNumAdapters - 1
        ' build a list of valid backbuffer formats
        D3DEnum_BuildValidFormatList lAdapter, D3DDEVTYPE_HAL
        D3DEnum_BuildValidFormatList lAdapter, D3DDEVTYPE_REF
        ' build a list of valid display modes for those formats
        D3DEnum_BuildDisplayModeList lAdapter, D3DDEVTYPE_HAL
        D3DEnum_BuildDisplayModeList lAdapter, D3DDEVTYPE_REF
        ' get the adapter identifier
        g_d3d.GetAdapterIdentifier lAdapter, 0, g_Adapters(lAdapter).d3dai
    Next

    D3DEnum_BuildAdapterList = True
    Exit Function
errOut:
    Debug.Print "Failed D3DEnum_BuildAdapterList"
End Function

Private Sub D3DEnum_BuildDisplayModeList(lAdapter As Long, devtype As CONST_D3DDEVTYPE)
    Dim lMode As Long
    Dim NumModes As Long
    Dim DisplayMode As D3DDISPLAYMODE

    With g_Adapters(lAdapter).DevTypeInfo(devtype)
        ' Reset the number of validated display modes to 0
        .lNumModes = 0
        ' Get the number of display modes
        ' Note this list includes refresh rates
        ' a display mode is a size and format (ie 640x480 X8R8G8B8 60hz)
        NumModes = g_d3d.GetAdapterModeCount(lAdapter)
        ' Allocate space for our mode list
        ReDim .Modes(NumModes)
        ' Save the format of the desktop for windowed operation
        Call g_d3d.GetAdapterDisplayMode(lAdapter, g_Adapters(lAdapter).DesktopMode)
        ' Loop through all the display modes
        For lMode = 0 To NumModes - 1
            ' Get information about this adapter in all the modes it supports
            Call g_d3d.EnumAdapterModes(lAdapter, lMode, DisplayMode)
            ' filter out low resolution modes
        Next
    End With
End Sub

```

```

    If DisplayMode.Width < 640 Or DisplayMode.Height < 400 Then GoTo Continue
    ' filter out modes allready in the list that might differ only in refresh
rate
    If -1 <> D3DEnum_FindInDisplayModeList(lAdapter, devtype, DisplayMode) _
        Then GoTo Continue
    ' filter out modes with formats that arent confirmed to work
    ' see BuildFormatList and ConfirmFormatList
    If -1 = D3DEnum_FindInFormatList(lAdapter, devtype, DisplayMode.format) _
        Then GoTo Continue
    ' At this point, the modes format has been validated,
    ' is not a duplicate refresh rate, and not a low res mode
    ' Add the mode to the list of working modes for the adapter
    .Modes(.lNumModes).lHeight = DisplayMode.Height
    .Modes(.lNumModes).lWidth = DisplayMode.Width
    .Modes(.lNumModes).format = DisplayMode.format
    .lNumModes = .lNumModes + 1

Continue:
    Next
    End With
End Sub

```

```

Private Sub D3DEnum_BuildValidFormatList(lAdapter As Long, devtype As CONST_D3DDEVTYPE)
    Dim lMode As Long
    Dim lUsage As Long
    Dim NumModes As Long
    Dim DisplayMode As D3DDISPLAYMODE
    Dim bCanDoWindowed As Boolean
    Dim bCanDoFullscreen As Boolean

    With g_Adapters(lAdapter).DevTypeInfo(devtype)
        ' Reset the number of available formats to
        .lNumFormats = 0
        ' Get the number of display modes
        ' a display mode is a size and format (ie 640x480 X8R8G8B8 60hz)
        NumModes = g_d3d.GetAdapterModeCount(lAdapter)
        ReDim .FormatInfo(NumModes)
        ' Loop through all the display modes
        For lMode = 0 To NumModes - 1
            ' Get information about this adapter in all the modes it supports
            Call g_d3d.EnumAdapterModes(lAdapter, lMode, DisplayMode)
            ' See if the format is already in our format list
            If -1 <> D3DEnum_FindInFormatList(lAdapter, devtype, DisplayMode.format) _
                Then GoTo Continue
            ' Check the compatibility of the format
            lUsage = D3DEnum_CheckFormatCompatibility(lAdapter, devtype, _
                DisplayMode.format, bCanDoWindowed, bCanDoFullscreen)
            ' Usage will come back -1 if VerifyDevice reject format
            If -1 = lUsage Then GoTo Continue
            ' Add the valid format and usage
            .FormatInfo(.lNumFormats).format = DisplayMode.format
            .FormatInfo(.lNumFormats).usage = lUsage
            .FormatInfo(.lNumFormats).bCanDoWindowed = bCanDoWindowed
            .FormatInfo(.lNumFormats).bCanDoFullscreen = bCanDoFullscreen
            .lNumFormats = .lNumFormats + 1
        Next
    End With

Continue:
    Next
    End With
End Sub

```

```

Private Function D3DEnum_CheckFormatCompatibility(lAdapter As Long, _
    DeviceType As CONST_D3DDEVTYPE, _
    format As CONST_D3DFORMAT, _
    ByRef OutCanDoWindowed As Boolean, _
    ByRef OutCanDoFullscreen As Boolean)

```

```

    As Long
    On Local Error GoTo errOut
    D3DEnum_CheckFormatCompatibility = -1
    Dim d3dcaps As D3DCAPS8

```

```

Dim flags As Long

' Filter out incompatible backbuffers
' Note: framework always has the backbuffer and the frontbuffer (screen) format
' matching
OutCanDoWindowed = True: OutCanDoFullscreen = True
If 0 <> g_d3d.CheckDeviceType(lAdapter, DeviceType, format, format, 0) Then _
    OutCanDoWindowed = False
If 0 <> g_d3d.CheckDeviceType(lAdapter, DeviceType, format, format, 1) Then _
    OutCanDoFullscreen = False
If (OutCanDoWindowed = False) And (OutCanDoFullscreen = False) Then Exit
Function

' If no form was passed in to use as a callback
' then default to software vertex processing

' Get the device capabilities
g_d3d.GetDeviceCaps lAdapter, DeviceType, g_d3dCaps
g_Adapters(lAdapter).d3dcaps = g_d3dCaps
' If user doesnt want to verify the device (didnt provide a callback)
' fall back to software
D3DEnum_CheckFormatCompatibility = D3DCREATE_SOFTWARE_VERTEXPROCESSING
If g_EnumCallback Is Nothing Then Exit Function
' Confirm the device for HW vertex processing
flags = D3DCREATE_HARDWARE_VERTEXPROCESSING
D3DEnum_CheckFormatCompatibility = flags
If g_d3dCaps.DevCaps And D3DDEVCAPS_HWTRANSFORMANDLIGHT Then
    If g_EnumCallback.VerifyDevice(flags, format) Then Exit Function
End If
' Try Software VertexProcessing
flags = D3DCREATE_SOFTWARE_VERTEXPROCESSING
D3DEnum_CheckFormatCompatibility = flags
If g_EnumCallback.VerifyDevice(flags, format) Then Exit Function
' Fail
D3DEnum_CheckFormatCompatibility = -1
Exit Function

errOut:
End Function

' Used to release any reference to the callback object passed in and deallocate memory
Public Sub D3DEnum_Cleanup()
    Set g_EnumCallback = Nothing
    ReDim g_Adapters(0)
End Sub

Private Function D3DEnum_FindInDisplayModeList(lAdapter As Long, _
    devtype As CONST_D3DDEVTYPE, _
    DisplayMode As D3DDISPLAYMODE) As Long

    Dim lMode As Long
    Dim NumModes As Long

    NumModes = g_Adapters(lAdapter).DevTypeInfo(devtype).lNumModes
    D3DEnum_FindInDisplayModeList = -1

    For lMode = 0 To NumModes - 1
        With g_Adapters(lAdapter).DevTypeInfo(devtype).Modes(lMode)
            If .lWidth = DisplayMode.Width And _
                .lHeight = DisplayMode.Height And _
                .format = DisplayMode.format Then
                D3DEnum_FindInDisplayModeList = lMode
                Exit Function
            End If
        End With
    Next
End Function

Private Function D3DEnum_FindInFormatList(lAdapter As Long, _
    devtype As CONST_D3DDEVTYPE, _
    format As CONST_D3DFORMAT) As Long

    Dim lFormat As Long
    Dim NumFormats As Long

```

```

NumFormats = g_Adapters(lAdapter).DevTypeInfo(devtype).lNumFormats
D3DEnum_FindInFormatList = -1

For lFormat = 0 To NumFormats - 1
    With g_Adapters(lAdapter).DevTypeInfo(devtype).FormatInfo(lFormat)
        If .format = format Then
            D3DEnum_FindInFormatList = .usage
            Exit Function
        End If
    End With
Next
D3DEnum_FindInFormatList = -1
End Function

Sub D3DUtil_Destroy()
    Dim emptycaps As D3DCAPS8
    Dim emptyrect As RECT
    Dim emptypresent As D3DPRESENT_PARAMETERS

    'D3DUtil_ReleaseAllTexturesFromPool

    Set g_dx = Nothing
    Set g_d3dx = Nothing
    Set g_dev = Nothing
    Set g_EnumCallback = Nothing

    g_focushwnd = 0
    g_behaviorflags = 0
    g_lWindowWidth = 0
    g_lWindowHeight = 0

    g_d3dCaps = emptycaps
    g_d3dpp = emptypresent
    g_WindowRect = emptyrect

    ReDim g_Adapters(0)
    g_lNumAdapters = 0
End Sub

' Desc: Performs timer operations. Use the following commands:
'       TIMER_RESET           - to reset the timer
'       TIMER_START           - to start the timer
'       TIMER_STOP            - to stop (or pause) the timer
'       TIMER_ADVANCE         - to advance the timer by 0.1 seconds
'       TIMER_GETABSOLUTE TIME - to get the absolute system time
'       TIMER_GETAPPTIME      - to get the current time
'       TIMER_GETELAPSEDTIME  - to get the elapsed time between calls
Function DXUtil_Timer(command As TIMER_COMMAND) As Single

    On Local Error Resume Next

    Static m_bTimerInitialized As Boolean
    Static m_bUsingQPF As Boolean
    Static m_fSecsPerTick As Single
    Static m_fBaseTime As Single
    Static m_fStopTime As Single
    Static m_fLastTime As Single

    Dim fTime As Single

    ' Initialize the timer
    If (False = m_bTimerInitialized) Then
        m_bTimerInitialized = True
    End If

    If m_fLastTime > Timer Then command = TIMER_RESET 'For the midnight wrap
    fTime = Timer

    ' Reset the timer
    If (command = TIMER_RESET) Then
        m_fBaseTime = fTime
        m_fStopTime = 0
        m_fLastTime = 0
    End If
End Function

```

```
        DXUtil_Timer = 0
        Exit Function
    End If

    ' Return the current time
    If (command = TIMER_GETAPPTIME) Then
        DXUtil_Timer = fTime - m_fBaseTime
        Exit Function
    End If

    ' Start the timer
    If (command = TIMER_start) Then
        m_fBaseTime = m_fBaseTime + fTime - m_fStopTime
        m_fLastTime = m_fLastTime + fTime - m_fStopTime
        m_fStopTime = 0
    End If

    ' Stop the timer
    If (command = TIMER_STOP) Then
        m_fStopTime = fTime
    End If

    ' Advance the timer by 1/10th second
    If (command = TIMER_ADVANCE) Then
        m_fBaseTime = m_fBaseTime + fTime - (m_fStopTime + 0.1)
    End If

    ' Return ellapsed time
    If (command = TIMER_GETELAPSEDTIME) Then
        DXUtil_Timer = fTime - m_fLastTime
        m_fLastTime = fTime
        If DXUtil_Timer < 0 Then DXUtil_Timer = 0
        Exit Function
    End If
    DXUtil_Timer = fTime
End Function

' Params
' a r g b values valid between 0.0 and 1.0
' Return Value
' a filled D3DCOLORVALUE type
Function ColorValue4(a As Single, r As Single, g As Single, b As Single) As
D3DCOLORVALUE
    Dim c As D3DCOLORVALUE
    c.a = a
    c.r = r
    c.g = g
    c.b = b
    ColorValue4 = c
End Function

' Params
' x y z vector values
' Return Value
' a filled D3DVECTOR type
Function vec3(x As Single, y As Single, z As Single) As D3DVECTOR
    vec3.x = x
    vec3.y = y
    vec3.z = z
End Function
```

Arquivo mdIParalela.bas

```

' Inp and Out declarations for direct port I/O in 32-bit Visual Basic
Public Declare Function Inp Lib "inpout32.dll" Alias "Inp32" (ByVal PortAddress _
                                                    As Integer) As Integer
Public Declare Sub Out Lib "inpout32.dll" Alias "Out32" (ByVal PortAddress As Integer, _
                                                    ByVal Value As Integer)

' Endereços de I/O - Status
Public Const IoData1 = "&h278"      ' Endereço de I/O da porta paralela 1 - LPT1
Public Const IoData2 = "&h378"      ' Endereço de I/O da porta paralela 2 - LPT2
Public Const IoSensor1 = "&h279"    ' Endereço de Status da porta paralela 1 - LPT1
Public Const IoSensor2 = "&h379"    ' Endereço de Status da porta paralela 2 - LPT2

' Constantes de reversão do motor
Public Const motNeutro = 0
Public Const motXNeg = 32
Public Const motXPos = 64
Public Const motYNeg = 96
Public Const motYPos = 128
Public Const motSobe = 160
Public Const motDesce = 192

' Níveis de força do acionamento dos motores
Public Const acTodos1 = 1
Public Const acTodos2 = 2
Public Const acTodos3 = 3
Public Const acTodos4 = 14
Public Const acTodos5 = 0
Public Const acEixoX1 = 9
Public Const acEixoX2 = 10
Public Const acEixoX3 = 11
Public Const acEixoX4 = 15
Public Const acEixoX5 = 8
Public Const acEixoY1 = 21
Public Const acEixoY2 = 22
Public Const acEixoY3 = 23
Public Const acEixoY4 = 30
Public Const acEixoY5 = 20

' Constantes dos sensores - os valores correspondem ao valor rcebido na entrada da
' porta paralela
Public Const sensorRotNeg = 104      ' Rotação para a esquerda
Public Const sensorRotNeutro1 = 120  ' Neutro
Public Const sensorRotNeutro2 = 216  ' Neutro
Public Const sensorRotPos = 200      ' Rotação para a direita

' Variáveis
Public sensor1      As Integer      ' Rot Esq=-1; Parado=0; Rot Dir=1
Public sensor2      As Integer      ' Rot Esq=-1; Parado=0; Rot Dir=1
Public entradaLpt1  As Integer      ' Variavel q guarda o valor do sensor 1
Public entradaLpt2  As Integer      ' Variavel q guarda o valor do sensor 2
Public saidaLpt1    As Integer      ' Sinal de controle dos motores - da Paralela 1

' Função que verifica a entrada da porta paralela
Public Sub ChecaEntrada()

    ' Variáveis recebem a entrada dos sensores 1 e 2
    entradaLpt1 = Inp(IoSensor1)
    entradaLpt2 = Inp(IoSensor2)

    ' Verifica o valor da entrada do sensor 1
    Select Case entradaLpt1
        Case sensorRotNeg: sensor1 = -1
        Case sensorRotNeutro1: Case sensorRotNeutro2: sensor1 = 0
        Case sensorRotPos: sensor1 = 1
        Case Else: sensor1 = 0
    End Select

    ' Verifica o valor da entrada do sensor 2
    Select Case entradaLpt2
        Case sensorRotNeg: sensor2 = -1
        Case sensorRotNeutro1: Case sensorRotNeutro2: sensor2 = 0
        Case sensorRotPos: sensor2 = 1
    End Select

```

```
        Case Else: sensor2 = 0
    End Select

End Sub

' Função que envia o sinal na saída da porta paralela
Public Function EnviaSaida(saida As String, sinal As Integer)
    Out saida, sinal
End Function
```

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] RICCÓ, M. H. **Interface Homem-Máquina Inteligente Aplicada a sistemas Cad.** 1997. Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo. São Paulo, 1997.
- [2] CIPELLI, ANTÔNIO M., MARKUS, O., SANDRINI, W. **Teoria e Desenvolvimento de Projetos de Circuitos Eletrônicos.** Editora Érica.
- [3] CAPUANO, F. G., MARINO, MARIA A. M. – **Laboratório de Eletricidade e Eletrônica, teoria e prática** – 8ª edição. Editora Érica.
- [4] GAJSKI, DANIEL D. **Principles of Digital Design.** Ed. Prentice Hall.
- [5] COGDELL, J. R. **Foundations of Electric Power.** Ed. Prentice Hall.
- [6] MALVINO, ALBERT P. **Eletrônica – volume 1.** Ed. McGraw-Hill.
- [7] MSDN. Contém uma biblioteca de artigos e exemplos de utilização das ferramentas Microsoft. Disponível em: <http://msdn.microsoft.com>. Acesso em: 03 maio 2003.
- [8] PLANET SOURCE CODE. Exemplos de programação em Visual Basic. Disponível em: <http://planet-source-code.com> . Acesso em: 13 abril 2003.
- [9] DIRECTX. São Paulo. Apresenta material para download do DirectX e consulta de artigos. Disponível em: <http://www.microsoft.com/directx>. Acesso em: 03 maio 2003.
- [10] COMUNICAÇÃO PARALELA. Dados sobre porta paralela do computador. Disponível em: http://www.geocities.com/SiliconValley/Server/9964/comun_paralela.html. Acesso em: 21 setembro 2003.
- [11] PARALLEL PORT INTERFACING TUTORIAL. Dados sobre porta paralela do computador. Disponível em: <http://www.logix4u.net/parallelport1.htm>. Acesso em: 21 setembro 2003.
- [12] ROBÓTICA - PORTA PARALELA. Dados sobre porta paralela do computador. Disponível em: <http://www.mrshp.hpg.ig.com.br/rob/paralela.htm>. Acesso em: 21 setembro 2003.
- [13] ROBÓTICA - PORTA PARALELA – PROGRAMANDO EM VISUAL BASIC. Dados sobre porta paralela do computador. Disponível em: <http://www.mrshp.hpg.ig.com.br/rob/portavisual.htm>. Acesso em: 21 setembro 2003.

[14] PARALLEMIC – THE PARALLEL MECHANISMS INFORMATION CENTER. Apresenta informações sobre mecanismos paralelos e suas aplicações. Disponível em <<http://www.parallemic.org>>. Acesso em: 20 abril 2003.

[15] VR LAB UNIV. OF TSUKUBA. Apresenta informações sobre “haptic devices” e exemplos de aplicação. Disponível em: <http://intron.kz.tsukuba.ac.jp/vrlab_web/index_e.html>. Acesso em: 22 abril 2003.

[16] BÜEHLER MOTOR. Site de fabricante de motorreduzidores CC com catálogos e data sheets. Disponível em: <www.buehler.com>. Acesso em: 13 abril 2003.

[17] FAIRCHILD. Fabricante de componentes eletrônicos que possui diversos data-sheets. Disponível em: <<http://www.fairchildsemi.com>>. Acesso em: 13 abril 2003.