

ecai

Universidade Federal de Santa Catarina
Centro Tecnológico
Curso de Engenharia de Controle e
Automação Industrial

ufsc

Estudo e Implementação de um
Acionamento Microprocessado para
Motores de Passo

Monografia submetida à Universidade Federal de Santa Catarina

como requisito para a aprovação da disciplina:

EEL 5901: Projeto de Fim de Curso

Cynthia Beatriz Scheffer

Florianópolis, Fevereiro de 1997.

**Estudo e Implementação de um Acionamento
Microprocessado para Motores de Passo**

Cynthia Beatriz Scheffer

Esta monografia foi julgada no contexto da disciplina
EEL 5901: Projeto de Fim de Curso
e aprovada na sua forma final pelo
Curso de Engenharia de Controle e Automação Industrial

Banca Examinadora:

José Maria Mascheroni, M.Eng.
Orientador da Empresa

Prof. Hari Bruno Mohr
Orientador do Curso

Prof. Augusto Humberto Bruciapaglia
Responsável pela disciplina e Coordenador do Curso

Prof. João Carlos da Silva Fagundes, Avaliador

André Camargo Guedes Rodrigues, Debatedor

Antônio Carlos Pereira Filho, Debatedor

*A meus pais,
Aguinaldo e Maria José,
e aos meus irmãos.*

*A Carlos, fiel companheiro,
pelo carinho e incentivo.*

Agradecimentos

A Deus, por ter me dado a oportunidade de concluir mais este importante passo em minha vida.

Ao meu querido Carlos, pelo incentivo e amor demonstrados no dia-a-dia e pela colaboração durante a execução deste projeto.

A minha família; pelo estímulo, apoio e compreensão.

À empresa ISA Engenharia Ltda., pela oportunidade de realizar este projeto, e em especial, a José M. Mascheroni pela orientação.

A todos os colegas da ISA, pela contribuição no meu desenvolvimento pessoal e profissional.

Ao professor Augusto H. Bruciapaglia; por sua amizade, dedicação e competência sempre presentes no decorrer do curso.

Aos demais professores da UFSC responsáveis pela minha formação, principalmente ao professor Hari B. Mohr pela revisão deste trabalho.

Aos amigos do curso, pelo companheirismo.

E a todos a quem não posso citar individualmente; mas que não foram, de modo algum, menos importantes para que eu conseguisse chegar a este dia.

Resumo

Este trabalho tem por objetivo o estudo e implementação experimental de um circuito microprocessado de acionamento elétrico para motores de passo.

As especificações do circuito desenvolvido envolvem a possibilidade de acionar motores de passo de duas fases (bipolares), com máxima tensão de 50V e corrente por fase de 2A nos modos de excitação por passo completo, meio passo e micropasso. Pela excitação por micropasso, consegue-se obter até 32 micropassos por passo fundamental do motor, resultando em 6400 posições por revolução. O sistema apresenta também comando para habilitação do *driver*, para escolha de direção e entrada de frequência do passo do motor.

A implementação de *drivers* de micropasso normalmente envolve a alta tecnologia dos circuitos lógicos integrados como PAL (*Programmable Array Logic*), GAL (*Generic Array Logic*) ou ASIC (*Application Specific Integrated Circuit*). Este projeto prevê a utilização de um microcontrolador específico para coordenar as comutações de fase do motor, o sentido de rotação, a velocidade do movimento e o tipo de excitação a ser empregada.

O desenvolvimento do sistema de comando dedicado, permitindo a implementação do modo de excitação por micropasso, melhora a resolução do movimento através da variação do nível de corrente aplicada ao motor. No micropasso, as bobinas do motor devem receber corrente suficiente para incrementar posições intermediárias do rotor entre os passos completos. Cabe ao circuito de controle a escolha correta da corrente a ser entregue ao motor, o que é realizado pela aplicação de uma tensão senoidal de referência no *driver*.

Os princípios teóricos que descrevem o acionamento por micropasso são apresentados bem como a metodologia de projeto é desenvolvida. É feita a descrição sucinta do microcontrolador e do *driver* empregado, sendo detalhado o procedimento de programação para gerar o micropasso.

Os resultados teóricos para diferentes programações são validados experimentalmente através de ensaios realizados em um protótipo desenvolvido do circuito de acionamento para motores de passo.

Abstract

This work is concerned with the investigation and realization of an experimental electrical driver for step motors commanded by microcontroller.

The specifications of the developed circuit include the possibility of driving bipolar stepper motors, with maximum voltage of 50V and phase current of 2A in full step, half step and microstep excitation modes. In microstepping operation, it is possible to obtain 32 microsteps for one fundamental step, resulting in 6400 microsteps per revolution. The system includes commands for enabling the driver, for choosing the direction and for setting the motor step frequency.

The implementation of microstep drivers normally involves high-technology digital logic components as PAL, GAL and ASIC. This project supports the utilization of a specific microcontroller to coordinate the motor phase computations, the direction of rotation, the velocity of motion and the excitation mode applied.

The development of a dedicated command system allows the implementation of the microstepping excitation mode, that improves the resolution of motion through the variation of the current level applied to the motor. In microstepping, the step motor windings must receive sufficient current to incremental the rotor position between full steps. The control circuit chooses the current to be applied to the motor through the application of a sinusoidal reference voltage to the driver.

The theoretical principles that describe the microstepping driver are presented as well as the methodology of the project is developed. A succinct description of the microcontroller and the driver employed is given, with a detailed view of the programming procedure to generate the microstepping.

The theoretical results for different microcontroller programs are experimentally validated using a stepper motors driver circuit prototype.

Sumário

CAPÍTULO I	CONSIDERAÇÕES GERAIS	1
1.1	INTRODUÇÃO	1
1.2	A EMPRESA	2
1.3	CONTEXTO DO PROJETO	4
CAPÍTULO II	ACIONAMENTOS ELETROMECAˆNICOS: REVISÃO BIBLIOGRÁFICA	6
2.1	INTRODUÇÃO	6
2.2	SERVOMOTORES - OS MOTORES DE PASSO	7
2.2.1	CLASSIFICAÇÃO ESTRUTURAL DE MOTORES DE PASSO	10
2.2.1.1	MOTOR DE PASSO DE RELUTÂNCIA VARIÁVEL	10
2.2.1.2	MOTOR DE PASSO DE IMÃS PERMANENTES	11
2.2.1.3	MOTOR DE PASSO HÍBRIDO	12
2.2.2	MODOS DE EXCITAÇÃO PARA MOTORES DE PASSO	16
2.3	ESTÁGIO AMPLIFICADOR DE SAÍDA - O <i>DRIVER</i>	20
2.3.1	TIPOS DE ACIONAMENTOS DE MOTORES DE PASSO	21
2.3.1.1	<i>DRIVER</i> UNIPOLAR	21
2.3.1.2	<i>DRIVER</i> BIPOLAR	22
2.3.1.3	ANÁLISE COMPARATIVA	24
2.3.2	TÉCNICA DE CONTROLE DE CORRENTE NAS FASES	25
2.3.2.1	USO DE DIODOS ZENER E RESISTORES	25
2.3.2.2	<i>DRIVER CHOPPER</i>	26
2.4	SISTEMA DE COMANDO/CONTROLE	27
2.5	CONCLUSÃO	28
CAPÍTULO III	METODOLOGIA DE PROJETO	29
3.1	INTRODUÇÃO	29
3.2	ETAPAS DO PROJETO	29
3.3	GERAÇÃO DE SINAIS SENOIDAIS	31
3.4	MODULAÇÃO POR LARGURA DE PULSOS (PWM)	35
3.4.1	A DEMODULAÇÃO DOS SINAIS PWM	37
3.5	CONCLUSÃO	38

CAPÍTULO IV O CIRCUITO DRIVER	39
4.1 INTRODUÇÃO	39
4.2 ASPECTOS FUNCIONAIS DO 3952	40
4.3 CONTROLE DE CORRENTE ATRAVÉS DO PWM	41
4.4 APLICAÇÃO DO 3952 COM MOTORES DE PASSO	43
4.5 CONCLUSÃO	44
CAPÍTULO V O MICROCONTROLADOR	45
5.1 INTRODUÇÃO	45
5.2 JUSTIFICATIVA DE ESCOLHA	45
5.3 O MICROCONTROLADOR PIC 16C65	46
5.3.1 CARACTERÍSTICAS DO PIC16C65	46
5.3.1.1 FREQUÊNCIA DO CLOCK	48
5.3.1.2 ESQUEMA DO CLOCK E CICLO DE INSTRUÇÕES	48
5.3.1.3 ORGANIZAÇÃO DA MEMÓRIA	49
5.3.2 ARQUITETURA	50
5.3.3 PRINCIPAIS MÓDULOS UTILIZADOS	52
5.3.3.1 PORTAS DE ENTRADA E SAÍDA	53
5.3.3.2 MÓDULOS DE TEMPORIZAÇÃO	54
5.4 MODO PWM	55
5.5 INTERRUPÇÕES	58
5.5.1 INTERRUPÇÃO EXTERNA ATRAVÉS DO RB0/INT	59
5.6 CONCLUSÃO	60
CAPÍTULO VI DESENVOLVIMENTO DO PROJETO	61
6.1 INTRODUÇÃO	61
6.2 AMBIENTE DE TRABALHO	61
6.2.1 PLACA DE DEMONSTRAÇÃO DE BAIXO CUSTO PICDEM-2	61
6.2.2 ASSEMBLER UNIVERSAL MPASM	63
6.2.3 O PROGRAMADOR PICSTART PLUS	63
6.2.4 CIRCUITO EMULADOR DE ALTA PERFORMANCE ICEPIC	64
6.3 ESPECIFICAÇÕES DO PROJETO	66
6.4 PROJETO DE <i>SOFTWARE</i>	66
6.5 PROJETO DE <i>HARDWARE</i>	76
6.6 CONCLUSÃO	77

CAPÍTULO VII RESULTADOS OBTIDOS	78
7.1 INTRODUÇÃO	78
7.2 ENSAIOS REALIZADOS	78
7.3 CONCLUSÃO	88
CAPÍTULO VIII CONCLUSÕES E PERSPECTIVAS	89
ANEXOS	91
A-1 CONVERSÃO D/A	92
A-1.1 CARACTERÍSTICAS GERAIS	92
A-1.2 INFORMAÇÃO DA INTERFACE LÓGICA	93
A-2 CIRCUITOS DE PROJETO DE <i>HARDWARE</i>	94
A-3 RESULTADOS COMPLEMENTARES	97
A-3.1 INTRODUÇÃO	97
A-3.2 FAIXA DE OPERAÇÃO DE 1 A 10 Hz	98
A-3.3 FAIXA DE OPERAÇÃO DE 10 A 100 Hz	99
A-3.4 FAIXA DE OPERAÇÃO DE 100 A 1 kHz	100
A-3.5 FAIXA DE OPERAÇÃO DE 1k A 10 kHz	101
A-3.6 FAIXA DE OPERAÇÃO DE 10 k A 100 kHz	102
REFERÊNCIAS BIBLIOGRÁFICAS	103

Lista de Ilustrações

CAPÍTULO II ACIONAMENTOS ELETROMECAÑICOS - REVISÃO BIBLIOGRÁFICA

2-1 Diagrama de blocos do controle de motores de passo	7
2-2 Motor de passo híbrido	13
2-3 Seqüência de acionamento das fases do motor de passo híbrido	15
2-4 Seqüência de excitação das fases de um motor de passo	16
2-5 Excitação das fases na seqüência <i>on phase on</i>	17
2-6 Formas de onda das correntes nas bobinas para <i>driver</i> com excitação das fases A e B em passo completo	17
2-7 Formas de onda da excitação das fases e correntes nas bobinas para o <i>driver</i> com meio passo	18
2-8 Forma das tensões de referência para obter correntes de fase senoidais no motor do passo (<i>driver</i> por micropasso)	19
2-9 Diagrama do circuito para <i>driver</i> unipolar	22
2-10 Diagrama do circuito para <i>driver</i> bipolar	22
2-11 Esquema de funcionamento da ponte-H	24
2-12 <i>Driver chopper</i>	26

CAPÍTULO III METODOLOGIA DE PROJETO

3-1 Tabela <i>look-up</i> para geração do seno discretizado	31
3-2 Tabela para geração do 1ºquadrante da função seno (32 valores)	33
3-3 Circuito básico para a demonstração do PWM	35
3-4 Saída V_o para diferentes valores de <i>duty cycle</i> com freqüência de chaveamento fixa	36
3-5 Filtro passa-baixa de 2ª ordem	37
3-6 Formas de onda do sinal PWM e demodulado	37
3-7 Sinal filtrado de saída do PWM	38

CAPÍTULO IV O CIRCUITO DRIVER

4-1 Diagrama funcional do 3952	40
4-2 Correntes circulando no <i>driver</i> do 3952 para diferentes modos de decaimento	42
4-3 Formas de onda de corrente de carga para diferentes modos de decaimento	42

CAPÍTULO V O MICROCONTROLADOR

5-1 Mapa de memória de programa e <i>stack</i> do PIC16C65	50
5-2 Diagrama de blocos da arquitetura do PIC16C65	52
5-3 Saída do sinal em PWM	56
5-4 Diagrama de blocos do módulo CCP operando no modo PWM	56
5-5 Tabela de frequências para o PWM (com <i>clock</i> de 20MHz)	58

CAPÍTULO VI DESENVOLVIMENTO DO PROJETO

6-1 Fluxograma geral do gerenciamento do <i>driver</i> de micropasso	69
6-2 Tabela de valores de tensões de fases nos sentidos: (a) horário e (b) anti-horário	70
6-3 Fluxograma da subrotina de fases	70
6-4 Fluxograma da subrotina de atendimento a interrupção	72
6-5 Fluxograma da subrotina do PWM	74
6-6 Fluxograma da subrotina de conversão D/A	75
6-7 Foto do protótipo do circuito de acionamento para motores de passo	77

CAPÍTULO VII RESULTADOS

7-1 Formas de onda do sinal PWM e micropasso para o <i>driver</i> com PWM a 156.3kHz	81
7-2 Formas de onda do sinal de referência (seno) para o <i>driver</i> com PWM a 156.3kHz	81
7-3 Formas de onda da seqüência de fases para o <i>driver</i> com PWM a 156.3kHz	82
7-4 Sobreposição do sinal de corrente e fase para o <i>driver</i> com PWM a 156.3kHz	82
7-5 Formas de onda do sinal PWM e micropasso para o <i>driver</i> com PWM a 78.12kHz	83
7-6 Formas de onda do sinal de referência (seno) para o <i>driver</i> com PWM a 78.12kHz	83
7-7 Formas de onda da seqüência de fases para o <i>driver</i> com PWM a 78.12kHz	84
7-8 Sobreposição do sinal de corrente e fase para o <i>driver</i> com PWM a 78.12kHz	84
7-9 Forma de onda do sinal de micropasso para o <i>driver</i> com conversores D/A	86
7-10 Formas de onda do sinal de referência (seno) para o <i>driver</i> com conversores D/A	86
7-11 Formas de onda da seqüência de fases para o <i>driver</i> com conversores D/A	87
7-12 Sobreposição do sinal de corrente e fase para o <i>driver</i> com conversores D/A	87

ANEXOS

A-1 Diagrama de blocos do conversor D/A	93
A-2.1 Circuito da fonte de alimentação do <i>driver</i> de micropasso	95
A-2.2 Circuito de amplificação do <i>driver</i> de micropasso	96

A-3.1 Entrada de passo a 5.3 Hz - velocidade do motor: 0.05 rpm	98
A-3.2 Entrada de passo a 53 Hz - velocidade do motor: 0.5 rpm	99
A-3.3 Entrada de passo a 530 Hz - velocidade do motor: 5 rpm	100
A-3.4 Entrada de passo a 5.3 kHz - velocidade do motor: 50 rpm	101
A-3.5 Entrada de passo a 53 kHz - velocidade do motor: 500 rpm	102



Considerações Gerais

1.1 Introdução

Antes do aparecimento dos microprocessadores, os circuitos elétricos para o acionamento de motores eram desenvolvidos analogicamente, sem nenhum tipo de controle. As estruturas de comando eram simples e implementadas por componentes com gatilho como os diodos, transistores e tiristores.

Com o advento dos microcontroladores, surge um estágio importante de integração que recebe sinais de controle de um processador e, aliado à eletrônica de potência que alimenta o motor, permite controlar seu movimento. Esta estreita relação entre os processadores, a eletrônica e os motores permitiu o aparecimento da automação industrial na área de acionamentos eletromecânicos.

Este projeto foi executado nesta área da automação, tendo por objetivo o desenvolvimento de um circuito de acionamento microcontrolado para motores de passo. A utilização de um dispositivo inteligente, capaz de comandar diversos sinais simultaneamente, permite a execução de um sistema de acionamento bastante eficiente, excitado através de micropassos.

O acionamento por micropasso surge da necessidade de posicionamento mais preciso entre os passos fundamentais do motor, possibilitando o emprego de motores de passo (que são de fácil controle, pois não exigem realimentação de posição para comandar seu movimento), em sistemas tradicionalmente acionados por motores de corrente contínua, cujas leis de comportamento são bastante conhecidas e utilizadas. Com o uso de

microcontroladores, obtêm-se: multiplicidade de funções, flexibilidade, agilidade e rapidez além de maior confiabilidade no acionamento. Isto tornou possível o uso desta moderna técnica de acionamento de motores de passo, que necessita de controle preciso das correntes de excitação do motor, a fim de permitir movimentações suaves e controladas para qualquer faixa de velocidade desejada.

Nos últimos anos, a empresa ISA Engenharia Ltda., onde se desenvolveu este projeto, tem-se dedicado a realização de projetos de automação de sistemas acionados principalmente por motores de passo. Muitos deles, utilizam *drivers* de micropasso importados, em vista da escassez deste produto no mercado nacional. Estes *drivers* normalmente são implementados através da alta tecnologia de dispositivos lógicos programáveis como as PAL (*Programmable Array Logic*), GAL (*Generic Array Logic*) e ASIC (*Application Specific Integrated Circuit*), o que torna o desenvolvimento do sistema mais complexo.

Este projeto se propõe a realização de um circuito de acionamento de motores de passo através de um *driver* bipolar totalmente implementado através de um microcontrolador específico; a fim de se obter uma resolução de posicionamento de 32 micropassos por passo, em tempo real. Pretende-se com isto atingir o nível de desempenho do sistema importado [Compumotor, 93], substituindo-o; o que representaria redução dos custos de desenvolvimento da empresa em seus projetos futuros.

Este capítulo segue com uma breve discussão sobre a empresa ISA Engenharia, sendo posteriormente apresentado um roteiro do trabalho.

1.2 A Empresa

Este projeto foi realizado na empresa **ISA Engenharia Ltda. - Integração de Sistemas e Automação**, situada no Centro Empresarial para Laboração em Tecnologias Avançadas - o CELTA - no Pólo Tecnológico de Florianópolis - o TECNÓPOLIS.

A empresa atua basicamente na área de automação de máquinas e equipamentos, tendo por objetivo encurtar a distância tecnológica verificada no mercado de equipamentos para

automação da manufatura, prestando serviços na forma de:

- projeto de soluções para problemas de automação;
- especificação de equipamentos de informática industrial - comandos numéricos computadorizados (CNC's), microcomputadores industriais, controladores lógicos programáveis (CLP's), servossistemas, motores de passo e sensores - nacionais ou importados;
- desenvolvimento de *software* aplicativo e de *hardware* específico quando a aplicação o requer;
- projeto e construção de comandos pneumáticos;
- projeto e construção de dispositivos mecânicos e de mecânica de precisão, entre outros.

A ISA Engenharia assessora empresas na elevação do nível tecnológico de seus processos, equipamentos e pessoal, visando o aumento de qualidade e produtividade. A ISA Engenharia também se vê presente na atual tendência de terceirização que se verifica nas empresas, como uma empresa disposta a assumir, total ou parcialmente e na forma de parcerias, problemas de engenharia que requeiram maior especialização.

A ISA Engenharia está capacitada a oferecer soluções de automação para clientes dos mais diversos setores industriais. Tem experiência acumulada nos seguintes setores: metal mecânico, de mecânica de precisão, de plásticos, de móveis e de vestuário, entre outros.

Entre seus clientes, pode-se destacar:

- EMBRACO - Empresa Brasileira de Compressores S. A.- Joinville, SC (Automação)
- Fornos JUNG Ltda. - Blumenau, SC (Consultoria, Automação)
- CERTI - Fundação Centro Regional de Tecnologia em Informática - Florianópolis, SC (Treinamento)
- Fundação de Ensino, Tecnologia e Pesquisa - São Bento do Sul, SC (Treinamento)
- Grupo Rudnick S. A. - São Bento do Sul, SC (Consultoria, Treinamento)
- IMARIBO S. A. Ind. E Com. - Monte Carlo, SC (Automação)
- Industrias Augusto Klimmek S.A. - São Bento do Sul, SC (Automação)
- Interfibra Industrial S.A. - Joinville, SC (Automação)
- SICAP Escapamentos - São Bento do Sul, SC (Treinamento)

- Mitutoyo - Suzano, SP (Automação)

Dentre os serviços da empresa se destacam aqueles onde é necessário um estreito relacionamento interdisciplinar entre as áreas de mecânica de precisão e eletrônica, e onde o nível de complexidade na integração/automação requer profissionais com domínio da técnica de projeto e conhecimento do estado da arte em automação da manufatura.

1.3 Contexto do Projeto

O projeto do circuito de acionamento microcontrolado para motores de passo se encontra bastante enquadrado no contexto das disciplinas oferecidas no Curso de Engenharia de Controle e Automação Industrial. Disciplinas como Microprocessadores e Acionamentos Elétricos para Automação possibilitaram acesso ao conhecimento da área e forneceram subsídios para o desenvolvimento do projeto. O restante foi conquistado às custas do esforço pessoal na busca de informações complementares em áreas não cursadas, como a Eletrônica de Potência.

Em vista disto, o trabalho é apresentado em capítulos, onde se propõe capacitar o entendimento pleno do desenvolvimento do sistema.

O capítulo II traz uma revisão bibliográfica dos assuntos imprescindíveis para a análise teórica e o entendimento do funcionamento e implementação do circuito de acionamento, bem como as referências iniciais sobre os componentes a serem utilizados.

O Capítulo III descreve a metodologia do projeto, fazendo uma análise das alternativas de solução do problema proposto.

O Capítulo IV faz uma abordagem sobre o circuito *driver* utilizado, analisando-o quanto aos sinais de controle que devem ser programados via microcontrolador para apresentar maior suavidade de resposta na operação por micropasso.

O Capítulo V apresenta o microcontrolador que irá gerenciar o sistema de controle do circuito de acionamento, mostrando sua arquitetura, organização de memória e principais recursos de programação utilizados para realizar este projeto.

No Capítulo VI, as especificações do projeto tanto para o desenvolvimento *de hardware* como de *software* são apresentados, sendo feita uma descrição do fluxograma de programação do microcontrolador a fim de gerar o micropasso. Os circuitos de *hardware* desenvolvidos para a construção de um protótipo do circuito também são mostrados.

No Capítulo VII são apresentados os resultados experimentais obtidos para diferentes programações do microcontrolador, através de um protótipo implementado.

Capítulo II



Acionamentos Eletromecânicos Revisão Bibliográfica

2.1 Introdução

Os sistemas eletromecânicos de acionamento de motores cumprem a função de interface entre um sistema elétrico e um sistema mecânico. Analisando-se pelo fluxo de energia, eles são conversores de energia elétrica em energia mecânica e vice-versa. Ao se considerar o fluxo de informações, eles convertem as informações de comando que vêm na forma de sinais elétricos de corrente e/ou tensão em sinais mecânicos de torque (ou força) e velocidade (ou deslocamento). Para que se estabeleça o fluxo de energia, o acionamento está ligado de um lado a uma fonte de energia (rede elétrica) e do outro ao sistema mecânico, também denominado de carga [Lichtblau, 1989].

Os principais requisitos de um servoacionamento de qualidade são :

- comportamento tão linear quanto possível;
- grande faixa dinâmica de velocidades ($V_{min}:V_{max}$ da ordem de 1:1000);
- torque estável ao longo de toda faixa de velocidades;
- torque suficiente para superar as cargas estáticas e dinâmicas;
- capacidade de realizar pequenos deslocamentos angulares.

Basicamente um *driver*, uma fonte CC para alimentação do sistema e um sinal de entrada para comando (podendo ser implementado por um gerador de pulsos) são necessários para operar um motor de passo, conforme mostra a ilustração 2-1.

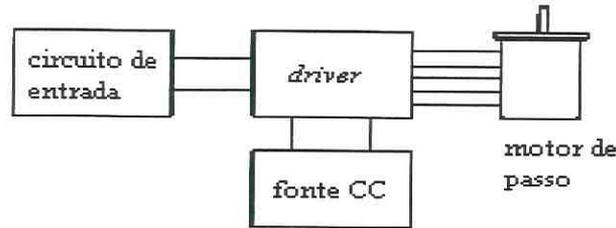


ilustração 2-1: Diagrama de blocos do controle de um motor de passo

Do ponto de vista construtivo, os acionamentos em geral podem ser divididos em três partes:

- servomotor (no caso, motor de passo)
- sistema eletrônico de controle/comando
- estágio amplificador de saída (o *driver* propriamente dito)

A escolha do servomecanismo depende muito das características dinâmicas exigidas pelo sistema, como a relação entre torque e velocidade, por exemplo. Para melhor entendimento dos detalhes envolvidos na escolha do sistema de acionamento, passa-se a descrição detalhada de cada um dos seus componentes.

2.2 Servomotores - os Motores de Passo

Servomotores são uma categoria de motores dos quais se pode controlar uma ou mais grandezas como: posição, velocidade, aceleração, torque; e que respondem aos comandos recebidos com elevada precisão e rapidez. O que distingue um servomotor de um motor comum é o fato de ter sido otimizado visando a obtenção de máxima performance dinâmica ao invés de mínimo custo de produção.

Independente de sua aplicação específica, os servomotores devem ter as seguintes características gerais:

- Alta dinâmica de torque, isto é, rápido controle de corrente;
- Alta dinâmica de velocidade, ou seja, baixo momento de inércia, baixa indutância e alto torque máximo;
- Alta capacidade de sobrecarga mecânica, isto é, alto torque máximo e boa rigidez

mecânica;

- Alta capacidade térmica, isto é, alto tempo de aplicação de sobrecargas sem sobreaquecimento;
- Característica torque X velocidade linear, para facilitar o controle;
- Larga faixa de controle de velocidade;
- Operação suave em baixas velocidades;
- Alta precisão no posicionamento;
- Projeto compacto, ou seja, elevada relação peso/volume;
- Alta confiabilidade, baixa manutenção e longa vida útil;
- Relação custo/performance aceitável.

Os motores de passo se enquadram na classificação de servomotores de movimento discreto, com a vantagem de serem diretamente controlados por sinal digital, ou seja, não há necessidade de controle realimentado, o que o torna de fácil utilização. Funcionalmente, os motores de passo são conversores de energia elétrica em energia mecânica e também transdutores de sinais elétricos digitais (pulsos) em sinais mecânicos discretos de movimento (posição, velocidade) [Lichtblau, 1989].

A rotação do eixo do motor de passo é caracterizada por um ângulo incremental específico para cada pulso de excitação. Este ângulo incremental é repetido precisamente em cada pulso, gerado por um circuito excitador adequado. Motores de passo apresentam elevada precisão de ângulo de passo, isto é, o erro que possa existir num determinado passo é geralmente menor que 5% de seu valor e não é acumulativo.

Este movimento fixo e preciso é o que possibilita a realização de controle de posição, velocidade, direção e distância, podendo em alguns casos até dispensar o controle em malha fechada, bastando para tal a segurança de que o motor tenha um torque suficiente para movimentar a carga acoplada [Bento, 1989].

O deslocamento do elemento de saída de um motor de passo é diretamente proporcional ao número de pulsos aplicados à entrada e a velocidade à frequência dos pulsos. O deslocamento elementar correspondente a um pulso aplicado é denominado passo fundamental do motor.

Por permitirem o posicionamento de seu eixo em malha aberta, podem prescindir de sensor para realimentação de posição, simplificando seu sistema de comando. Apresentam também elevada confiabilidade por exigirem pouca manutenção e não possuírem escovas e comutador.

As principais vantagens no uso de motores de passo foram resumidas a seguir:

- O ângulo de rotação do motor é diretamente proporcional ao número de pulsos do sinal de entrada;
- O erro de ângulo por passo é muito pequeno e não cumulativo;
- Possui resposta rápida para iniciar, parar e reverter o movimento;
- Sinais digitais como entrada em pulsos permitem controle em malha aberta, o que torna o sistema mais simples;
- Pode manter uma alta sincronização em condições de carga diretamente conectada;
- Apresenta capacidade de manter o rotor parado em determinada posição sem o uso de freios.
- Pode operar em grande faixa de velocidades de rotação, proporcionais à frequência dos pulsos de entrada;
- O custo do motor é relativamente baixo; em vista da simplicidade de construção;
- Pode-se alcançar uma boa suavidade de movimento, dependendo do tipo de motor e do acionamento utilizado;
- Provém uma alta confiabilidade, não sendo necessário utilizar componentes para realimentação;
- Pequena necessidade de manutenção por não possuir escovas e comutador;
- Pode trabalhar em qualquer tipo de ambiente;
- É menos sujeito a falhas do que outros servomotores.

Como desvantagens no uso de motores de passo, pode-se citar:

- Apresentam movimentos ásperos em baixa velocidade, a menos que não seja utilizado um acionamento com excitação por micropasso;
- Apresentam efeitos de ressonância na frequência natural;

- São incapazes de detectar perda de posicionamento, já que geralmente operam em malha aberta;
- Consomem alta corrente quando trabalhando com carga, o que faz com que aqueçam. Perdas de velocidade são relativamente altas e também podem causar excesso de aquecimento, e geram freqüentemente ruídos nas altas velocidades.
- Há limitação de tamanho dos motores e a precisão de posicionamento depende muito da mecânica (por exemplo, precisão do fuso). Estes efeitos podem ser superados com o uso de controle em malha fechada, mas com isto perde-se uma enorme vantagem, que é a operação em malha aberta.

Apresentadas estas características gerais, passa-se a discussão específica de cada tipo de motor de passo, de acordo com seu princípio de funcionamento. Outro tópico a ser tratado leva em consideração as diferentes formas de excitar as bobinas do motor a fim de se obter o passo, e com isto, o giro.

2.2.1 Classificação Estrutural dos Motores de Passo

Os motores de passo podem ser classificados quanto ao seu aspecto construtivo e princípio de funcionamento em três tipos: de relutância variável, de ímãs permanentes e híbrido.

2.2.1.1 Motor de Passo de Relutância Variável

O estator dos motores de passo de relutância variável consiste de multipolos salientes bobinados com dentes usinados nas sapatas polares. O rotor, constituído de núcleo cilíndrico de aço laminado, também é dentado. O rotor e o estator não possuem o mesmo número de dentes. Através dessa diferença produz-se um efeito de amplificação de resolução. O motor gira quando os dentes do rotor são atraídos pela energia eletromagnética dos dentes do estator.

Quando é aplicada corrente a um conjunto de bobinas do motor, produz-se um torque no sentido de levar o circuito magnético à condição de mínima relutância magnética (resistência à passagem de linhas de fluxo magnético). Enquanto for mantida a combinação de energização, o rotor permanece na posição de mínima relutância e se opõe a ser tirado

desta posição. Quando é modificada a combinação de bobinas energizadas muda o ponto de mínima relutância e o rotor busca atingi-lo novamente. Pela seqüência apropriada de chaveamento das bobinas, produz-se um campo girante discreto, que é acompanhado pelo rotor.

Os motores de relutância magnética atingem velocidades angulares mais elevadas se comparados aos demais tipos, por não possuírem ímãs que induzem força contra-eletromotriz nas bobinas. Com isto, o rotor gira livremente sem torque de retenção. Também não apresentam nenhum torque residual; quando as bobinas não estão energizados, o torque estático é zero. Não podem ser acionados de forma bipolar pois o fenômeno de relutância independe do sentido do fluxo [Vexta, 1987].

Este tipo de motor é geralmente utilizado em tamanhos reduzidos para aplicações como mesas de micro-posicionamento. Os motores de relutância variável são raramente utilizados em aplicações industriais. Eles não são sensíveis a polaridade de corrente e requer *drivers* de acionamento diferentes dos demais tipos de motores. A inércia do rotor é baixa e a resposta é bastante rápida, só que permite apenas cargas inerciais leves. O ângulo de passo do motor é de 15°.

2.2.1.2 Motor de Passo de Ímãs Permanentes

Estes motores possuem no estator uma série de pólos salientes bobinados. O funcionamento desse motor baseia-se na interação que ocorre entre o campo magnético produzido nos ímãs permanentes e o campo produzido pela corrente que circula nas bobinas do estator. Ao energizar o motor, o rotor se move para um ponto de equilíbrio mais próximo da posição onde ele se encontra, permanecendo nesta posição enquanto não for modificada a combinação de energização das bobinas. Ao se modificar a combinação de energização das bobinas em uma seqüência apropriada, o rotor segue em passos o campo girante discreto produzido no estator, permanecendo sempre num ponto de equilíbrio.

O torque dos motores de passo de ímã permanente é maior que os de relutância variável. Também o amortecimento é maior. A presença dos ímãs permanentes acarreta um torque residual que torna o rotor radialmente magnetizado em determinadas posições, mesmo

estando as bobinas desenergizadas. Contudo, este torque é baixo se comparado ao torque nominal do motor [Kenjo, 1993].

Este tipo de motor pode ser acionado de forma bipolar ou unipolar. No acionamento bipolar, a corrente pode circular nos dois sentidos possíveis das bobinas. No acionamento unipolar, uma bobina é utilizada para cada sentido de corrente.

Este tipo de motor é o mais utilizado em aplicações não industriais, como em periféricos de computador e aplicações envolvendo baixa inércia. A sua construção resulta em ângulos de passo relativamente grandes, mas a sua simplicidade faz com que a produção de altos volumes tenha um baixo custo. O motor de disco (*axialair gap*) é uma variação do ímã permanente, projetado para aumentar a performance, que acontece por causa da reduzida inércia do rotor. Entretanto, isto limita o seu campo de aplicação àquelas envolvendo pouca inércia (por exemplo, posicionamento do carro de impressão em impressoras matriciais).

Existem dois tipos de ímãs utilizados na construção do rotor: um é o *alunico-magnet* e o outro é o ferrite. Os motores de passo com rotor de *alunico-magnet* geralmente possuem ângulos de passo de 45° ou 90°, enquanto os de rotor de ferrite possuem ângulos de passo de 7,5°; 11,25°; 15° e 18°.

2.2.1.3 Motor de Passo Híbrido

Os motores de passo híbridos têm o estator semelhante aos de relutância variável, e o rotor combinando o princípio deste com o de ímãs permanentes. O rotor e o estator são dentados, sendo que o rotor apresenta concentração magnética ao longo de sua forma.

A ilustração 2-2 mostra esquematicamente um motor de passo híbrido cujo estator compõe-se de 4 bobinas enroladas em pólos salientes. O rotor compõe-se de 2 peças polares de aço doce dentadas, e entre elas um ímã permanente magnetizado de forma axial. O número de dentes do rotor e estator é diferente (em geral, 48 e 50 dentes respectivamente) e é a relação entre eles que determina o passo fundamental. Em geral, o ângulo de passo de 1,8° graus ou 200 passos por revolução.

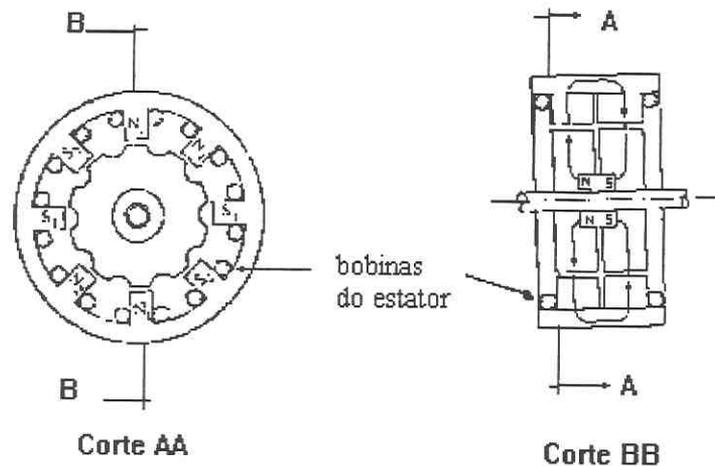


ilustração 2-2: Motor de passo híbrido

O motor de passo híbrido pode ser acionado de forma unipolar ou bipolar e apresenta a melhor resolução dos três. Este motor, geralmente de alta precisão e torque elevado, é o mais utilizado em aplicações industriais, pois combina muitas das vantagens dos dois anteriores.

O motor de passo híbrido apresenta um estator (constituído de vários enrolamentos independentes em torno de um material ferromagnético) e um rotor que pode girar livremente, coaxial ao estator. O estator pode ser descrito como um conjunto de eletroímãs distribuídos em círculo, cada um com uma bobina independente. À medida em que estas bobinas vão sendo energizadas, estabelece-se nelas um fluxo magnético cujo comando atrai o rotor para uma posição fixa predeterminada. Quando estas bobinas são alimentadas seqüencialmente; forma-se um campo magnético girante, discreto, que o rotor segue, causando deslocamentos igualmente discretos, chamados de passos.

A velocidade de chaveamento seqüencial das bobinas do estator e a ordem da seqüência deste chaveamento determina o sentido e a velocidade de rotação do rotor. O percurso angular realizado é proporcional ao número de chaveamentos realizados.

Um motor de passo híbrido padrão de 200 passos foi o utilizado neste projeto. Para melhor entendimento da operação do motor híbrido far-se-á a análise do funcionamento a partir de um modelo que pode produzir 12 passos por revolução, cuja seqüência pode ser acompanhada na ilustração 2-3.

O rotor desta máquina consiste em duas partes de pólos diferentes com três dentes em cada. Por entre as peças polares, existe um ímã permanente magnetizado ao longo do eixo do rotor, fazendo com que num extremo esteja o pólo norte e no outro o sul. Os dentes são montados nos dois lados do eixo.

O estator consiste numa carcaça contendo 4 dentes que ocupam todo o comprimento do rotor. Enrolamentos são montados nos dentes do estator e são conectados em pares.

Com o motor completamente desenergizado, sem corrente alguma passando pelos enrolamentos do motor, o rotor irá permanecer em alguma das posições (a), (b), (c) ou (d) mostradas no diagrama. Isto ocorre pelo fato de que os ímãs permanentes no rotor tentam minimizar a relutância magnética do fluxo magnético entre uma extremidade e outra. Isto irá ocorrer quando um par de dentes do rotor de pólos norte e sul são alinhados com dois pólos do estator. O torque que tende a segurar o rotor numa destas posições é geralmente pequeno, sendo chamado de *detent torque* ou torque de retenção. O motor mostrado possui 12 possíveis posições de retenção.

Se a corrente passa por um par de bobinas do estator, como mostra a ilustração 2-3(a), os pólos norte e sul resultantes no estator vão atrair os dentes de polaridade inversa em cada extremo do rotor. Agora existem apenas três posições estáveis para o rotor, o mesmo número que dentes do rotor. O torque requerido para tirar o rotor desta posição de equilíbrio é muito maior, e é conhecido por *holding torque*.

Pela mudança do fluxo de corrente do primeiro para o segundo conjunto de enrolamentos do estator, o campo magnético do mesmo irá girar 90° e atrairá um novo par de pólos. Isto resulta num movimento do rotor de 30° , correspondendo a um passo completo (ilustração 2-3(b)). Revertendo para o primeiro conjunto de bobinas, mas energizando-as com polaridade inversa, o campo do estator irá girar mais 90° e o do rotor mais 30° (ilustração 2-3(c)). Finalmente, o segundo arranjo de bobinas é energizado com polaridade inversa para dar uma terceira posição de passo (ilustração 2-3(d)). Pode-se então voltar para a primeira condição (ilustração 2-3(a)) e depois destes quatro passos, o rotor terá se deslocado por um dente completo. Este motor simples possui então um performance de 12 passos por revolução. Obviamente, se os enrolamentos são energizados na seqüência

inversa, o motor irá girar no sentido contrário.

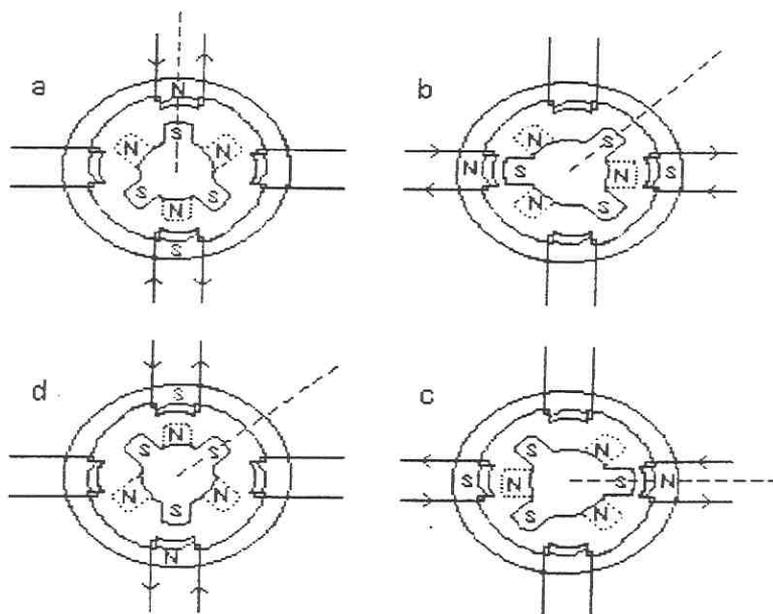


ilustração 2-3: Seqüência de acionamento das fases do motor de passo híbrido

Se dois enrolamentos são energizados simultaneamente, o rotor irá mover-se para uma posição intermediária desde que ele seja atraído igualmente pelos dois pólos do estator. Um torque maior é produzido nesta condição porque todos os pólos do estator estão influenciando no rotor. O rotor pode ser construído para apenas mover-se um passo completo simplesmente pela reversão da corrente em um conjunto de enrolamentos; o que causa uma rotação de 90° no campo do estator com anteriormente. De fato esta será a forma normal de acionar o motor no modo de passo completo, sempre mantendo dois enrolamentos energizados e revertendo a corrente em cada um alternadamente.

De acordo com o número e a forma com que as bobinas são alimentadas, consegue-se diminuir o tamanho do passo do motor (ângulo), aumentando a resolução no posicionamento. Pela energização alternada de um e dois enrolamentos, o rotor gira em passos de somente 15° em cada estágio e o número de passos por revolução será duplicado. Isto é chamado de operação em meio passo e a maioria das aplicações industriais utilizam este modo de operação. Embora algumas vezes há uma pequena perda de torque, este modo resulta numa melhor suavidade em baixas velocidades, menor sobre-sinal e pequena oscilação no fim de cada passo.

Segue uma melhor análise das diferentes formas de excitar as bobinas do motor a fim de se obter o movimento rotativo.

2.2.2 Modos de Excitação para Motores de Passo

Para possibilitar o movimento do motor, é necessário fornecer ao *driver* a seqüência correta de chaveamento de correntes nas bobinas. O tipo de excitação a ser selecionado depende da resolução de posição requerida. Existem 4 possíveis formas de excitar o motor [Vexta, 1987]. Demonstra neste as seqüências possíveis através de um motor de passo de ímãs permanentes de dois pólos, sendo que a mesma vale para os outros tipos de motores. As seqüências estão mostradas na ilustração 2-4.

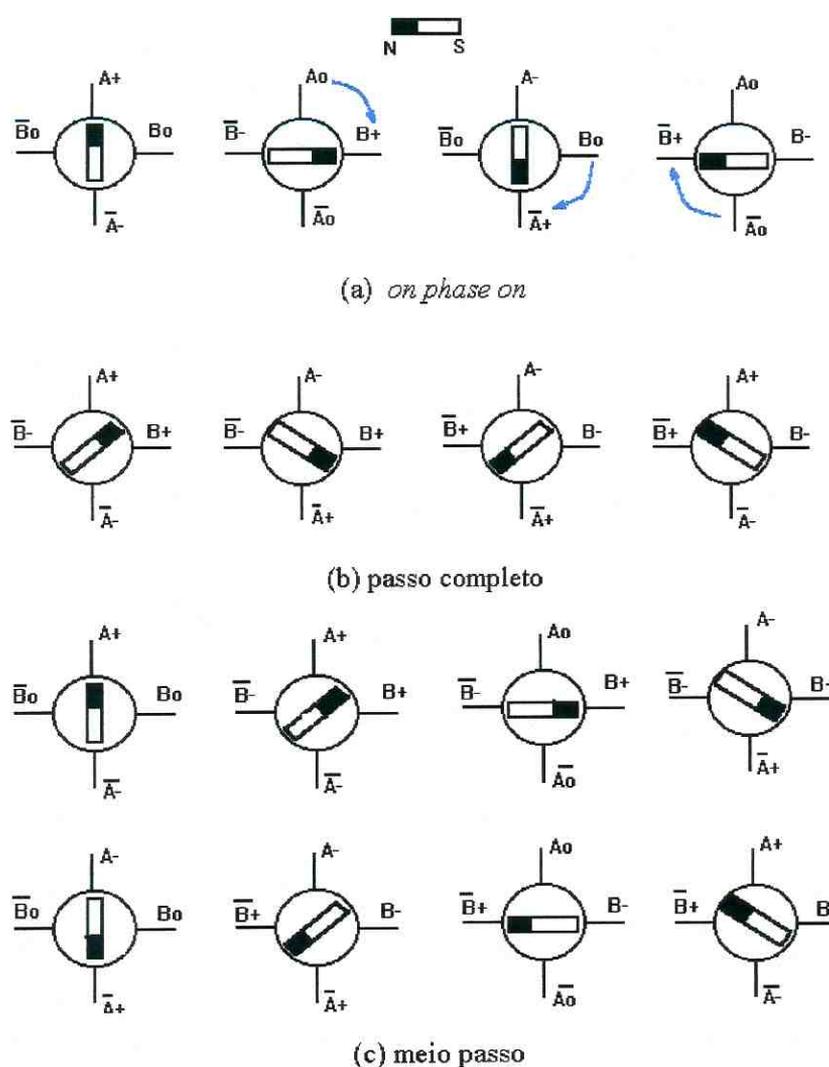


ilustração 2-4 : Seqüência de excitação das fases de um motor de passo

A primeira seqüência é conhecida como *one phase on* - “uma fase acionada”, ora a fase A ora a fase B. Tem a vantagem de consumir baixa potência e apresentar boa precisão de ângulo de passo. Entretanto, as oscilações são tão grandes que o motor tende a entrar em ressonância. Assim, este modo não pode ser usado onde vibrações são indesejáveis (ilustração 2-4 (a)). A mudança na condição de excitação de cada fase está mostrada na ilustração 2-5.

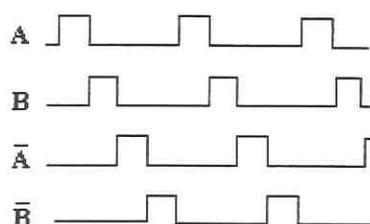


ilustração 2-5 : Excitação das fases na seqüência *one phase on*

A segunda possibilidade (ilustração 2-4(b)) é energizar ambas as bobinas a fim de manter o rotor sempre alinhado entre os dois pólos. As duas fases do motor estão sempre energizadas. Esta seqüência chama-se *two phase on*, mas é mais conhecida como **passo completo**, já que o deslocamento correspondente a um pulso é igual ao passo fundamental do motor. Apresenta menos oscilações do que o modo anterior e pode responder a uma faixa maior de frequências. Este é o modo mais comum de acionamento de motor de passo bipolar e dá ao motor o maior torque de saída. A mudança na condição de excitação e a forma de onda das correntes de fase é mostrada na ilustração 2-6.

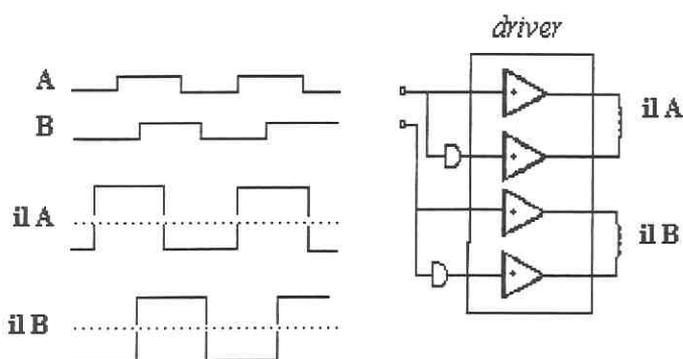


ilustração 2-6: Formas de onda da correntes nas bobinas para *driver* com excitação das fases A e B em passo completo

A terceira opção é energizar ora uma fase, ora duas fases, alternadamente; sendo o deslocamento correspondente a um pulso a metade do passo original. Esta seqüência - *one-two phase on* - mais conhecida como **meio passo**, apresenta um ângulo de passo efetivo de metade do ângulo construtivo do motor. Com isto, este modo é mais desejável quando se deseja ângulos de passo pequenos. Atinge um torque menor que no modo por passo completo (ilustração 2-4(c)). As curvas características da excitação das fases A e B no meio passo, bem como a forma de onda da corrente nas bobinas do motor (il. A e il. B) são mostradas na ilustração 2-7.

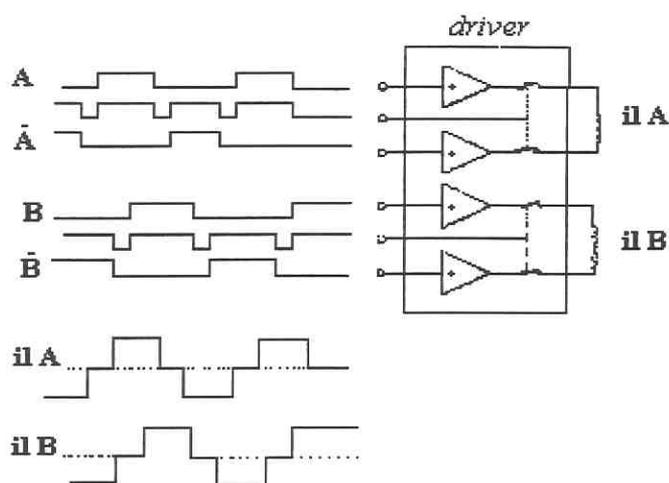


ilustração 2-7: Formas de onda de excitação das fases e correntes nas bobinas para *driver* com meio passo

Observa-se que para rotação em sentido oposto, utiliza-se a mesma seqüência só que na ordem reversa.

Como mostram as seqüências da ilustração 2-4, este motor apresenta um ângulo de passo fundamental de 90° . Motores reais tem múltiplos pólos para reduzir o ângulo de passo fundamental a poucos graus, mas o número de bobinas e a seqüência do *driver* não mudam devido a isto.

Quando o motor é acionado no modo de passo completo, energizando dois enrolamentos ou “fases” ao mesmo tempo (ilustração 2-6), o torque disponível em cada passo será o mesmo (sujeito a variações muito pequenas nas características do motor e acionamento). No modo de meio passo, está se energizando alternadamente duas fases e depois somente

uma como mostrado na ilustração 2-7. Assumindo que a corrente de cada enrolamento seja a mesma em cada caso, o maior torque é produzido quando as duas bobinas estão energizadas. Em outras palavras, passos alternados serão fortes e fracos. Isto não representa um grande problema para a performance do motor, o torque disponível é obviamente limitado pelo passo mais fraco, mas haverá uma melhora significativa na suavidade do movimento em baixas velocidades para o meio passo do que no modo de passo completo.

Viu-se também que energizando ambas as fases com correntes iguais produz-se uma posição de passo intermediária entre a posição de uma fase acionada. Se a corrente de duas fases são diferentes, a posição do rotor será mudada para mais perto do pólo mais forte. Este efeito é utilizado em acionamento de micropassos, o que subdivide o passo básico do motor pela proporção de corrente em dois enrolamentos. Deste modo, o tamanho de passo é reduzido e a suavidade em baixas velocidades é sensivelmente aumentada. *Drivers* de micropassos de alta resolução dividem o passo inteiro do motor em 200 micropassos, dando 40.000 passos por revolução. Nesta situação, o padrão de corrente nos enrolamentos se aproxima muito de duas ondas senoidais defasadas de 90° .

A ilustração 2-8 mostra a excitação das fases no modo passo completo e as formas de onda das tensões de referência do *driver* a fim de produzir as correntes senoidais nas bobinas do motor.

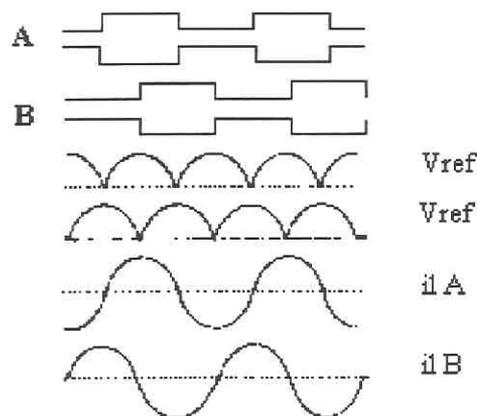


ilustração 2-8: Forma das tensões de referência para obter correntes de fase senoidais no motor de passo (*driver* por micropasso)

O modo de excitação por **micropasso** acontece pela aplicação de níveis de tensão variados nos bobinados do motor, podendo ser realizado através do controle digital (ou mesmo analógico) da corrente nas fases. O micropasso pode ser necessário em situações onde um passo completo (ou meio passo) do motor signifique um incremento de posição de baixa precisão, ou ainda, onde é preciso que o motor de passo praticamente deslize. Este tipo de operação permite obter o controle fino do passo do motor sem a necessidade de uso de motores mecanicamente mais complexos com ângulos de passo muito reduzidos.

As vantagens de se utilizar o modo de excitação através do micropasso estão resumidas a seguir :

- Eliminação do fenômeno de ressonância;
- Redução drástica do ruído do motor;
- Menor desgaste das engrenagens e cargas conectadas ao motor;
- Resolução de posição é muito melhor (tamanho do passo relativamente pequeno);
- Fornecimento de torque praticamente constante.

Observa-se que a característica das correntes das fases do motor tende a ser do tipo sinusoidal à medida que se reduz a largura do passo utilizado. Assim, a geração do micropasso está baseada na forma senoidal das correntes das bobinas do motor. Esta função de gerar o perfil de onda desejado cabe ao circuito *driver*. Para se obter uma corrente de fase sinusoidal, a tensão de referência do circuito *driver* deve ser modulada por meia onda senoidal, conforme mostrado na ilustração 2-8 [Smith, 1989].

Para melhor entendimento da forma como isto é gerado, faz-se necessário primeiramente explicar as características desta etapa de amplificação dos sinais de acionamento antes de atingirem o motor.

2.3 Estágio Amplificador de Saída - *Driver*

Para que o motor possa ser acionado pelos pulsos digitais na seqüência e na velocidade desejadas, faz-se necessário o uso de um circuito *driver*. O *driver* é um circuito que engloba um estágio amplificador que permite excitar o motor através do fornecimento de

energia elétrica na forma e quantidade adequadas para que este execute com precisão e rapidez os comandos recebidos. O *driver* deve amplificar o sinal de comando à sua entrada entregando-o ao motor. A definição da corrente é de grande importância no funcionamento do motor, visto que é dela que depende o torque do mesmo.

As funções principais do *driver* são de sequenciar, distribuir e amplificar adequadamente os pulsos do sinal de comando, excitando cada bobina do motor de passo em valores específicos. O sequenciamento é realizado de acordo com a informação do sentido de deslocamento. As saídas deste circuito são tensão e corrente para cada uma das fases do motor.

O termo *driver* comumente é confundido com o sistema completo de acionamento do motor, em vista de sua importância fundamental no funcionamento do circuito.

2.3.1 Tipos de Acionamento de Motores de Passo

Existem 2 tipos de acionamentos (*drivers*) determinados pelo sentido da corrente através das bobinas do estator: o unipolar e o bipolar.

2.3.1.1 Driver Unipolar

Quando a corrente circula somente em um sentido nas fases do motor. Este tipo de acionamento é normalmente utilizado onde a estrutura de circuito é simples devido à direção fixa de excitação da corrente. No caso de modo de excitação por passo completo, entretanto, este modo de acionamento não mostra as principais características do motor porque se utiliza de apenas metade das bobinas do motor.

Este tipo de acionamento é mais empregado para excitar motores de passo de relutância variável [Kenjo, 1993]. A estrutura do *driver* unipolar pode ser vista na ilustração 2-9.

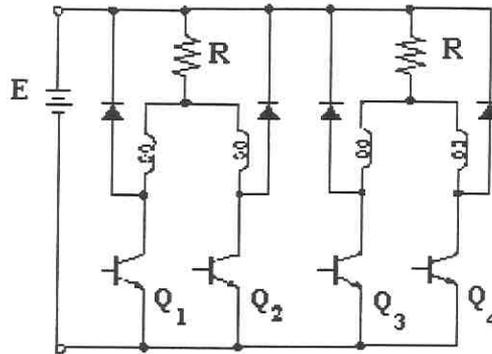


ilustração 2-9: Diagrama do circuito para *driver* unipolar

2.3.1.2 *Driver* Bipolar

Quando a corrente circula em ambos os sentidos nas fases do motor, está se utilizando o acionamento bipolar. Neste *driver* faz-se uso de um circuito em cascata conhecido como **ponte-H**, composto por um esquema de 4 transistores e 4 diodos aliados a uma fonte de tensão contínua. Os transistores superiores do esquema são chamados de fonte e os inferiores de dreno do *driver*. O *driver* bipolar representado na sua configuração mais usada, a emissor comum, com transistores tipo NPN é mostrada na ilustração 2-10 abaixo.

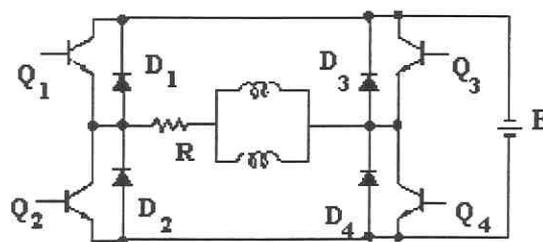


ilustração 2-10: Diagrama do circuito para *driver* bipolar

A configuração em ponte-H funciona pelo chaveamento ordenado dos transistores pertencentes ao circuito amplificador. O *driver* bipolar se caracteriza por chavear os transistores aos pares. Quando se quer corrente positiva no motor, liga-se os transistores Q1 e Q4. No sentido negativo, liga-se Q2 e Q3. Pelo chaveamento alternado destes dois pares, pode-se controlar a corrente.

A combinação com o chaveamento de um transistor do par num tempo variável faz com que a tensão do motor varie entre $-E$ e $+E$, cujo valor médio é dependente do ciclo de trabalho, ou seja, da frequência com que ocorre este chaveamento. Este método é conhecido como amplificação com modulação por largura de pulsos, ou simplesmente, *driver* com PWM ou *chopper*, que será discutido mais a frente.

Quando um par de transistores é desligado, existe um tempo finito para que estes atinjam o estado OFF. Assim, deve-se permitir um certo tempo de atraso (*blanking time*) entre a ação de desligar um par de transistores e ligar o outro par. Este tempo de atraso, que é aproximadamente de $5\mu\text{s}$, é o que limita a frequência de chaveamento em torno de 20 kHz [Stute et al., 1983].

No circuito mostrado na ilustração 2-11, verifica-se também a utilização de diodos colocados entre o coletor e o emissor de cada transistor. Estes diodos são necessários em muitas situações para proteger os dispositivos que chaveiam a corrente fluindo na carga, que pode ser indutiva, como no caso de um motor. Considere-se a função destes diodos na ilustração 2-11(a). Os transistores Q1 e Q4 estão ligados (ON), e Q2 e Q3 desligados (OFF), e a corrente está fluindo através da carga como indicado. Quando os transistores Q1 e Q4 são abertos, a corrente no motor, que tende a permanecer fluindo na mesma direção, irá circular através de D2 e D3 e assim a energia indutiva retornará a fonte.

Com isto, não há problema se os transistores Q2 e Q3 foram fechados ou não no mesmo instante de desligamento de Q1 e Q4. Se estes diodos não existissem no circuito, os transistores OFF certamente seriam destruídos pela alta tensão produzida em seus terminais pela mudança abrupta de corrente na carga. Quando Q1 e Q4 passam do estado ON para OFF, os diodos D2 e D3 os protegem deste problema. Estes diodos são conhecidos como diodos de recirculação em vista da forma como atuam, retornando a energia à fonte.

Agora demonstra-se o que ocorre quando os transistores Q2 e Q3 estão abertos e Q4 é mantido fechado. O transistor Q1 é usado como uma chave. Conforme mostra a ilustração 2-11(b), a corrente flui no sentido direto quando o transistor Q1 está ON. Quando a chave é aberta, a corrente de carga irá continuar fluindo através do diodo. Deve-se notar que a

corrente não retorna a fonte pela recirculação como no caso anterior, mas que circula em “roda livre” através da carga e dos elementos do circuito, que no caso são D2 e Q4. Por esta razão, o diodo em paralelo com o elemento que faz a função de chave é chamado de diodo de roda livre. Assim, um diodo pode ter função de recirculação ou roda livre, depende de como o circuito está operando.

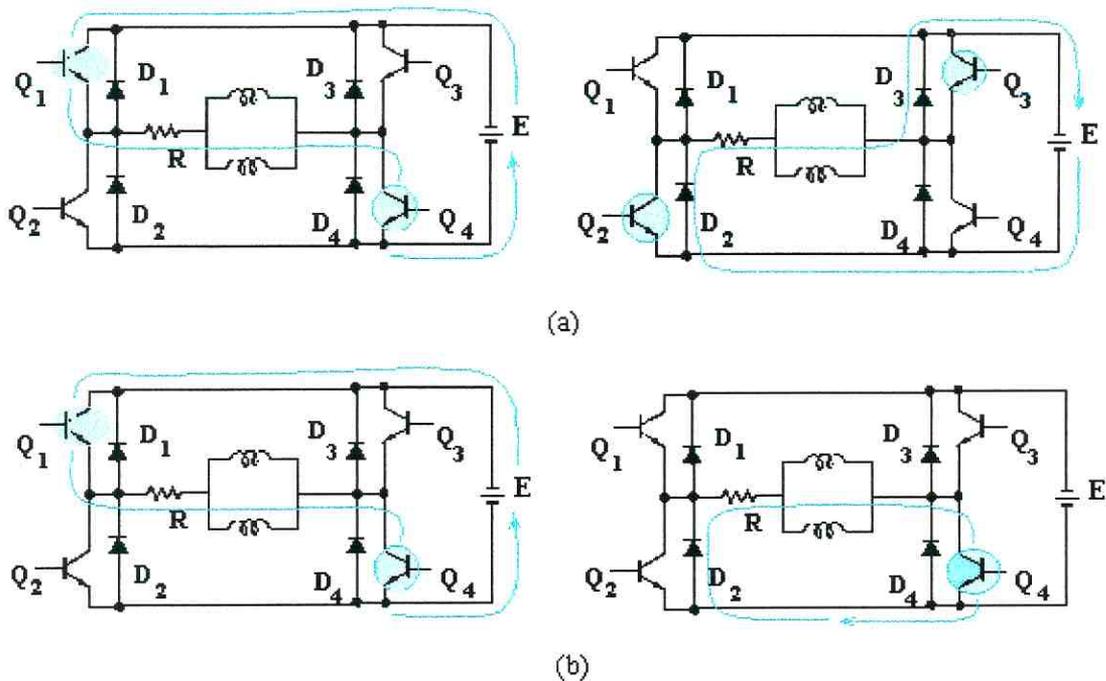


ilustração 2-11: Esquema de funcionamento da ponte-H

No caso do circuito para *driver* unipolar (ilustração 2-9), quando ocorre o desligamento de um transistor, o diodo em paralelo com a carga tem sempre a função de roda livre, permitindo que a corrente circule entre a carga, o diodo e a resistência R em série.

2.3.1.3 Análise Comparativa

Como os enrolamentos do motor são efetivamente usados porque a corrente de excitação é bidirecional, o *driver* bipolar pode gerar torque de saída mais alto a velocidades baixas se comparado ao unipolar. A constante de tempo entretanto é aumentada e o torque a alta velocidade é decrementado, porque as bobinas estão acionadas juntas. O número de bobinas de fase é a metade do *driver* unipolar.

O custo do *driver* bipolar é maior que o do unipolar. Quando o principal requisito for alta velocidade, o acionamento unipolar é mais recomendado, por apresentar menor constante de tempo, fazendo a corrente se estabilizar mais rapidamente. Quando se deseja maior torque a velocidades reduzidas, melhor utilizar o bipolar [Vexta, 1987]. Antigamente, os *drivers* unipolares eram mais usados, devido a sua simplicidade construtiva. No entanto, hoje em dia, com a disponibilidade de circuitos integrados em ponte-H, e até mesmo de *drivers* completos em *chips*, o acionamento bipolar conquistou a preferência dos usuários de motores de passo.

Neste projeto, o *driver* utilizado será um bipolar implementado através de um circuito integrado, o 3952.

2.3.2 Técnica de controle de corrente nas fases

No acionamento de um motor de passo, o tempo de atraso da corrente em relação a tensão é um sério problema, especialmente em elevadas velocidades. Quando operando em alta velocidade, o torque de saída de um motor de passo decresce quando a taxa de pulsos de entrada aumenta. Por esta razão, alguns métodos são empregados para melhorar a característica de corrente, que é responsável pelo torque, em altas frequências de entrada.

2.3.2.1 Uso de Diodos Zener e Resistores

Com auxílio de um resistor em série com a conexão em paralelo das duas fases, pode-se reduzir a constante de tempo elétrica. Aumentando-se o valor da resistência em série pode ser aumentada a tensão da fonte CC e, por conseguinte, obtida uma constante de tempo menor (crescimento mais rápido da corrente), com melhor dinâmica. Esta técnica tem a desvantagem de dissipar energia no resistor em série, mas é bastante usada por ser de baixo custo e simples de realizar [Lichtblau, 1989]. Um diodo zener pode ser usado em série com o diodo de recirculação para suprir o transiente de corrente após o desligamento das bobinas. Os circuitos apresentados nas ilustrações 2-9 e 2-10 já incluem estes resistores.

2.3.2.2 Uso de um *Driver Chopper*

Esta técnica é implementada através do chaveamento em alta frequência dos transistores (*chopper*) da ponte-H. O princípio de funcionamento de um *driver chopper* ou PWM é ilustrado pela ilustração 2-12, onde a carga é representada pela combinação de uma resistência e uma reatância.

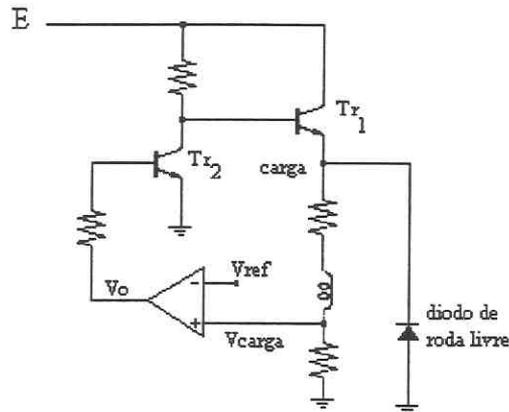


ilustração 2-12: *Driver chopper*

O transistor Tr_1 é repetidamente chaveado a uma alta frequência para ajustar automaticamente a corrente de carga a um nível determinado como a tensão de referência (V_{ref}) aplicada ao terminal negativo de um comparador. No terminal de referência, é aplicada uma onda triangular com um dado nível CC superposto. Quando a tensão do terminal positivo (corrente da carga) é maior que a de referência, o tempo em que o Tr_2 está conduzindo (T_{on}) decresce; por outro lado, quando a tensão da carga é menor que o nível de referência, T_{on} de Tr_2 aumenta a fim de ajustar a corrente até o nível desejado. Com este esquema, a frequência PWM é a mesma da onda triangular superposta à tensão de referência CC [Kenjo, 1993].

O *driver* PWM permite monitorar a corrente nas fases, através de um circuito de controle que compara a corrente medida com um valor de referência previamente ajustado e chaveia os transistores de forma a manter a corrente muito próxima do valor requerido. Esta técnica permite obter melhor performance dinâmica de servomotores. Geralmente, é a utilizada com acionamentos bipolares.

2.4 Sistema de Comando/Controle

Se motores de corrente contínua giram quando alimentados com tensão, para se acionar os motores de passo necessita-se de um sinal de comutação para realizar o controle dos mais diversos comandos que permitem sua movimentação. Pelo fato de não possuir escovas e comutador para operação, o seu controle pode ser interfaceado diretamente com um microprocessador ou computador. Em 95% das aplicações utilizando motores de passo, a origem destes comandos digitais é um sistema microprocessado [Smith, 1987].

Na forma mais simples de acionar o motor, o controle por passo completo, necessita apenas de dois sinais retangulares em quadratura. De acordo com cada seqüência de fase ativa, o eixo do motor gira no sentido horário ou anti-horário, numa velocidade proporcional à frequência dos pulsos de entrada.

Desde que se tenha todos os sinais desejáveis nos circuito e se conheça a forma como interagem entre si, pode-se desenvolver uma lógica para acionar o motor da maneira que se quiser. Geralmente, quatro sinais estão presentes no controle de um motor em todas as operações:

- **Clock:** o sinal de *clock* dá ao motor o comando do passo. É a frequência com que o motor vai responder aos comandos.
- **Reset:** coloca todos os sinais do sistema em uma posição de partida pré definida.
- **Direção:** determina o sentido do movimento de rotação do eixo do motor.
- **Modo de Excitação:** decide se o sistema opera por passo completo, meio passo ou ainda micropasso.

Para gerar a interface entre um motor de passo e um computador é necessário um circuito lógico digital que realize o comando do chaveamento da corrente contínua de uma fonte CC para as bobinas do motor. Utiliza-se um bit de uma porta de saída do processador para enviar os pulsos, outro bit para enviar a informação de sentido de movimento ao circuito lógico e mais dois com a forma de onda da seqüência de excitação das fases. Para gerar um passo do motor por *software* é preciso primeiro determinar a taxa de ocorrência do passo desejada e calcular o período de tempo necessário entre os passos. O

microcontrolador é então programado para contar este período antes de enviar os sinais ao motor para mover mais um passo.

No caso de acionamento através de micropasso, o microcontrolador pode ser utilizado para controlar também o nível da tensão de referência do *driver*, que é responsável pela forma de corrente circulando no motor.

O microcontrolador, PIC16C65, responsável pelo sistema de comando para o acionamento realizado neste projeto será detalhado em um capítulo próprio.

2.5 Conclusão

Este capítulo permitiu a formação do embasamento teórico necessário para o entendimento de um circuito de acionamento. Com esta revisão bibliográfica apresentada, tem-se subsídios para fazer uma análise da temática do projeto e compreender a metodologia empregada.

Capítulo III



Metodologia do Projeto

3.1 Introdução

O projeto em questão propõe-se a realização de um circuito de acionamento de motores de passo através do modo de excitação por micropasso. Com base no estudo bibliográfico, reunido no capítulo II, fez-se uma seleção dos principais componentes que fariam parte deste circuito, bem como uma análise teórica da forma como este projeto poderia ser implementado. Este capítulo trata desta discussão com base nos componentes selecionados para o desenvolvimento do projeto.

3.2 Etapas do Projeto

O projeto foi desenvolvido para acionar um **motor de passo híbrido**, em vista de seu elevado torque e alta exatidão de posicionamento, o que o destaca como o mais utilizado a nível industrial. Esta seleção foi importante para o projeto, pois dela é dependente a forma de excitação a ser empregada, o tipo de *driver* a ser escolhido e a resolução possível de se obter pelo emprego do micropasso. Conforme já citado, o motor de passo híbrido tem ângulo de passo fundamental de 1.8° , ou seja, 200 passos por resolução. Para o acionamento do motor foi selecionado um ***driver* bipolar**, considerando suas características quanto à velocidade e ao torque. Este *driver* foi implementado com um circuito integrado dedicado, baseado em ponte-H. O sistema de comando para gerar o micropasso, pelo controle das corrente de fase do motor, foi realizado através de um **microcontrolador** específico a fim de se obter uma resolução de 32 micropassos por passo, em tempo real.

A realização de projeto se desenvolveu em três etapas principais :

Primeira Etapa : dedicou-se ao estudo do comportamento de cada subsistema que integra o acionamento, como análise do motor, do *driver* e da geração do sinal de comando.

Segunda Etapa : dedicou-se ao estudo e a implementação de programas usando o PIC16C65, da *Microchip Inc.*, a fim de gerar as funções individuais requeridas no projeto através deste microcontrolador.

Terceira Etapa: dedicou-se à montagem de um protótipo do circuito de acionamento, à análise dos resultados obtidos no acionamento do motor de passo por meio de diferentes programações de controle e a comparação destes resultados a fim de se buscar a melhor resposta para o projeto.

Conforme mostrado no capítulo II, a excitação do motor de passo através de passos menores que o fundamental do motor passa pela geração de um sinal de controle que leve as correntes das fases do motor a ser do tipo senoidal à medida que se reduz a largura do passo de comando utilizado até atingir o micropasso. A geração do micropasso está baseada na forma senoidal das correntes das bobinas do motor, o que é obtido se a tensão de referência do *driver* bipolar for comandada por sinais em meia onda senoidais retificados.

O microcontrolador PIC16C65 foi utilizado para controlar o nível da tensão de referência de um *driver* através da geração de um seno discretizado, permitindo que as correntes de fase atinjam a forma desejada, mostrada na ilustração 2-8.

Existem diferentes formas de enviar os níveis de corrente necessários para acionar o motor pelo micropasso. Neste projeto, realizou-se a implementação por meio de conversores digital/analógico (D/A) e de circuitos PWM. O uso de um conversor D/A, que transforma valores capturados numa tabela em níveis de tensão correspondentes a uma senóide discretizada no tempo pode ser adequado em muitos casos, pois é de simples programação. Mas uma implementação mais compacta, de menor custo, sem a necessidade de componentes externos adicionais, pode ser interessante. Esta alternativa para a geração dos sinais senoidais é a sintetização de uma onda modulada por pulsos (PWM) pelo próprio

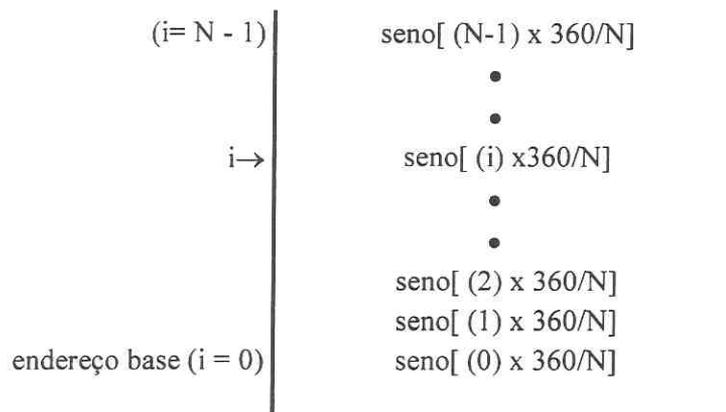
microcontrolador responsável pelo comando dos sinais de acionamento. Neste caso, uma tabela contendo os valores digitalizados do seno baseados no período do PWM dá ao sistema o valor de ciclo de trabalho pertinente a cada novo pulso do sinal de entrada. Este sinal gerado é filtrado e aplicado às tensões de referência do *driver* [Carlson et al., 1988].

Será feita uma breve explanação da forma como estes sinais senoidais podem ser implementados através de um processador digital. A modulação destes sinais através de um PWM também será tratada neste capítulo.

3.3 Geração de Sinais Senoidais

A geração de ondas senoidais é muito usada em aplicações de controle. Com a introdução de processadores de alta velocidade e precisão, ondas senoidais estáveis, de baixa distorção e com frequência qualquer, podem ser produzidas digitalmente usando-se apenas uma tabela (*table look-up*) com os valores dos senos, baseada num certo número de acessos a serem executados [Chrysafis, 1988].

A rotina para sintetizar sinais senoidais está fundamentada no seguinte procedimento: os valores usados para aproximar uma onda senoidal são armazenados numa tabela da memória como mostra a ilustração 3-1.



onde: N = comprimento da tabela

i = índice da tabela; $0 \leq i \leq n$

N(i) = valor do seno armazenado no i-ésimo endereço da tabela

ilustração 3-1: Tabela *look-up* para geração do seno discretizado

Observa-se que para gerar uma senóide só é necessário armazenar os valores referentes ao 1º quadrante, pois estes se repetem segundo funções trigonométricas conhecidas, sendo preciso apenas reduzir ao 1º quadrante ângulos pertencentes ao 2º, 3º ou 4º. A frequência da onda gerada é função do intervalo de tempo e do incremento entre ângulos (Δ , Δ) entre acessos sucessivos à tabela.

Se o delta for unitário, isto é, entradas lidas seqüencialmente, e a tabela for acessada a cada T segundos (T é considerado um intervalo de amostragem), então a frequência fundamental da tabela - FTF, devido ao nome original *Fundamental Table Frequency* - para a onda senoidal sintetizada será:

$$FTF = 1/NT \text{ Hz}$$

Por outro lado, se delta é maior do que a unidade, ou seja, $\Delta=2$, $\Delta=3$ e assim por diante, e a tabela continua sendo acessada a cada T segundos, então a frequência do seno discretizado passa a ser:

$$f = \Delta \cdot FTF \text{ Hz}; \Delta \leq N/2$$

Uma onda senoidal sintetizada depende do comprimento da tabela, N, da acurácia (número de bits de precisão) dos dados armazenados na tabela e do valor de delta, Δ .

Esta implementação é o método *look-up* direto. Por causa do valor de delta estar limitado a números inteiros, todos os acessos terão seus valores contidos diretamente na tabela, não sendo necessário fazer aproximações.

A priori, estipulou-se que seriam utilizados 64 valores para representar a meia onda senoidal desejada. Como é preciso armazenar apenas um quadrante da função seno, foi realizado o cálculo dos 32 valores integrantes da tabela *look-up*, baseado nos seguintes conhecimentos:

- Como 90º é o décimo sexto valor da tabela, isto implica que o incremento entre ângulos, partindo-se de zero é de $\Delta= 2.813^\circ$. Assim, o 1ºquadrante se verifica por ter ângulos representados por : $\alpha(i) = (i) \times 90/32$, onde $i = \text{índice da tabela}$, $0 \leq i \leq 31$.

- Para completar a tabela, a cada ângulo $\alpha(i)$, obtém-se o valor do seno e realiza-se a transformação para o hexadecimal de 2 bytes correspondente, através de uma simples regra de três. Para obter esta relação, basta considerar que o seno de 90° é o máximo valor da tabela e que corresponde a um número desejado em hexa, que pode ser FFh, 3Fh, 1Fh dependendo do tipo de programação que se está utilizando. Por exemplo, com conversores D/A, usa-se FFh, que é o maior valor de conversão do circuito. Já com o uso de PWM, cada um dos valores citados representa uma frequência diferente para o PWM, que deve ser selecionada durante a programação.

Através deste procedimento, obteve-se diferentes tabelas *look-up* a serem utilizadas no programa de acionamento por micropasso. Por exemplo, a tabela que implementa o 1º quadrante da função seno digitalizada, calculada para $\text{seno}(90^\circ)=1=\text{FFh}$; é a representada abaixo:

Ordem	Ângulo	Seno	Valor
00	0	0	00h
01	2.81	0.049	0Ch
02	5.63	0.098	19h
03	8.44	0.147	25h
04	11.26	0.195	32h
05	14.07	0.243	3Eh
06	16.88	0.290	4Ah
07	19.69	0.337	56h
08	22.50	0.383	62h
09	25.82	0.427	6Dh
10	28.13	0.471	78h
11	30.94	0.514	83h
12	33.76	0.555	8Eh
13	36.57	0.596	98h
14	39.38	0.634	A2h
15	42.20	0.672	ACh
16	45.01	0.707	B5h

Ordem	Ângulo	Seno	Valor
17	47.82	0.741	BEh
18	50.63	0.773	C6h
19	53.45	0.803	CDh
20	56.26	0.831	D5h
21	59.07	0.858	DBh
22	61.89	0.882	E2h
23	64.70	0.904	E7h
24	67.51	0.924	ECh
25	70.33	0.941	F1h
26	73.14	0.957	F5h
27	75.95	0.970	F8h
28	78.76	0.981	FBh
29	81.58	0.989	FDh
30	84.39	0.995	FEh
31	87.20	0.998	FFh
32	90.00	1.000	FFh

ilustração 3-2: Tabela para geração do 1º quadrante da função seno (32 valores)

Considerando que o objetivo desta rotina é a geração de meia-onda senoidal, é preciso acrescentar na subrotina do programa uma função que permita reduzir ângulos de 2º quadrante ao 1º, o que pode ser feita pela utilização da seguinte relação trigonométrica:

- $\text{seno}[x] = \text{seno}[180^\circ - y]$, onde: x, y = argumentos do 1º e 2º quadrante, respectivamente.

Basta reduzir o argumento antes da chamada da tabela para se obter o seno correspondente.

Estes 32 valores de seno serão percorridos enquanto o motor estiver realizando um passo fundamental, ou seja, serão realizados 32 micropassos da corrente de fases por passo de tensão do motor.

Inicialmente, pretendia-se ampliar esta tabela para 128 valores a fim de gerar a meia onda senoidal com maior precisão. No entanto, isto se tornou inviável para este projeto em vista da limitação do próprio controlador, que operando a 20 MHz de frequência de *clock*, não conseguiria acessar e calcular em tempo hábil todas as rotinas necessárias ao controle do motor, o que implicaria num funcionamento apenas a baixas velocidades (menores que 500 rpm).

Os valores desta meia onda senoidal precisam ser convertidos analogicamente para patamares de 0 a 5 V de tensão, a fim de serem utilizados como sinal de entrada do circuito *driver*, permitindo que a corrente do motor tenha a característica desejada.

Uma forma de realizar isto seria simplesmente carregar estes valores em um conversor D/A que estivesse programado para uma tensão de referência máxima de 5 V. A saída do conversor D/A seria diretamente acoplada ao *driver*. Informações sobre o funcionamento e a programação do conversor D/A estão em anexo 1. No entanto, a forma mais compacta de gerar este sinal seria atribuindo aos valores da tabela *look-up* o correspondente ciclo de trabalho de um modulador por largura de pulsos (PWM - *Pulse Width Modulation*) para a confecção de uma senóide. Com este sinal, devidamente filtrado e tratado, obtém uma meia onda perfeita.

A escolha do uso do PWM baseia-se primeiramente nos princípios gerais: simplicidade, confiabilidade, custo e resposta dinâmica de controle. A resposta dinâmica pode ser um requisito de controle mais exigido a fim de satisfazer as especificações de projeto. Há um compromisso entre a boa resposta dinâmica e estabilidade. No caso deste projeto, o funcionamento do sistema depende fortemente da velocidade de execução do PWM, interno ao microcontrolador. Isto será tratado no capítulo sobre o desenvolvimento do projeto de *software*. A explicação sobre o que é e como pode ser utilizado neste projeto um circuito PWM será dada a seguir.

3.4 Modulação por Largura de Pulsos

Conforme citado anteriormente, para se obter a forma de onda senoidal a ser atribuída ao *driver* bipolar como referência de tensão para sequenciar os níveis de corrente a serem distribuídos às bobinas do motor; é preciso que, partindo das formas de onda digitalizadas por uma cadeia de pulsos de amplitudes referentes a um seno, se obtenha uma forma de onda senoidal suave e de amplitude coerentes gerada pelo circuito de acionamento microprocessado. Uma forma de conformar uma onda senoidal sintetizada é através de um modulador por largura de pulsos (PWM) que é um circuito usado para produzir diferentes níveis de tensão CC dependendo do ciclo de trabalho dos pulsos gerados.

O princípio básico do PWM é a variação da tensão de saída (V_o) em um circuito baseado numa chave S (vide ilustração 3-3), de acordo com uma onda quadrada, apresentando um valor médio dentro de um ciclo repetitivo que está relacionado à tensão de entrada (V_i) e ao tempo que se mantém a chave aberta ou fechada, baseada na seguinte fórmula :

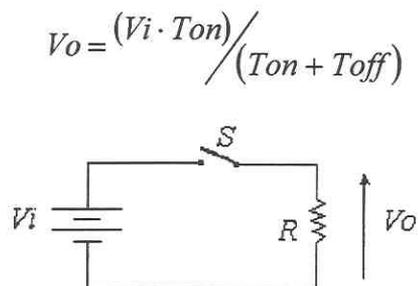


ilustração 3-3: Circuito básico para demonstração do PWM

Se utilizar o conceito de *duty cycle* (ciclo de trabalho), tem-se:

$$DT = \frac{T_{on}}{(T_{on} + T_{off})}$$

Tem-se assim:

$$V_o = DT \cdot V_i$$

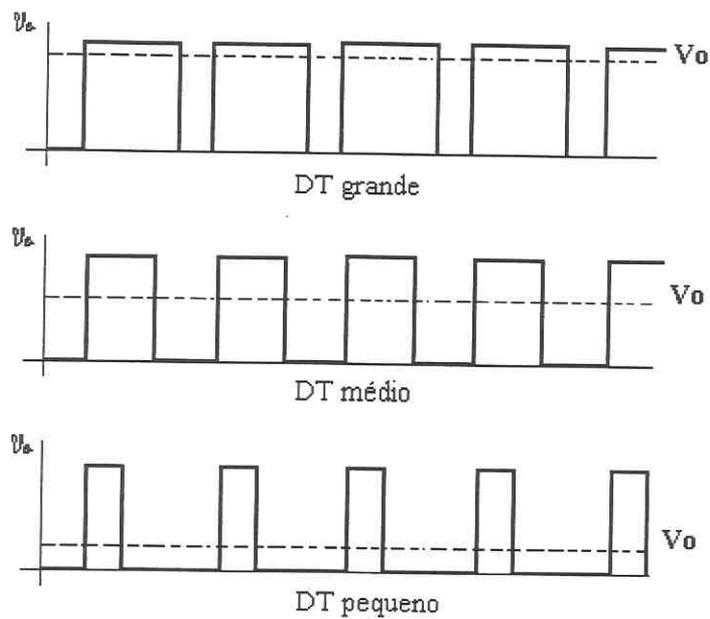


ilustração 3-4 : Saída V_o para diferentes valores de *duty cycle* (DT) com frequência de chaveamento fixa

Pela ilustração 3-4, observa-se como a onda de saída muda com a variação da razão do *duty cycle*, para um período de chaveamento constante.

Pode-se dizer que PWM é um método de expressar um sinal analógico em uma forma de sinal de um bit variável no tempo. Existem diferentes métodos de gerar ou demodular os sinais PWM.

Uma onda triangular pode ser aplicada a um dos terminais de um comparador para modular um sinal que está aplicado a outro terminal. A onda PWM é obtida no terminal de saída. Pode-se definir o sinal da fonte como sendo o sinal a ser modulado e a onda triangular como onda de modulação. Isto é similar a forma usada em *drivers chopper* para implementar um PWM.

Ao invés de utilizar um comparador, que é um componente analógico, um conjunto de componentes digitais ou microprocessador pode ser usado. O sinal da fonte é convertido em dados digitais para serem enviados ao microprocessador e são lidos por um programa cuja função é similar a do comparador.

Outra possibilidade é não utilizar nenhum componente analógico. O sinal da fonte é gerado por um programador e a função de produzir o PWM é atribuída a alguma região de memória do computador. Este método, completamente digital, é o utilizado para desenvolvimento deste projeto, sendo que o sinal de entrada está disponível na forma de um seno digitalizado.

3.4.1 A Demodulação dos Sinais PWM

O significado de demodulação é recuperar o sinal analógico a partir de um sinal em pulsos. Existem diversas formas de demodular um sinal. Isto pode ser obtido através de filtros passivos (apenas com componentes discretos - resistores, capacitores e indutâncias) ou por ativos, que se utilizam de amplificadores operacionais.

O sinal de magnitude do PWM deve ser filtrado para converter a onda de ciclo de trabalho variável em um sinal analógico. Uma forma de se implementar um filtro para PWM é estipular uma frequência de corte por volta de 10 vezes a da onda modulada. A maneira mais simples certamente é através de uma rede RC, funcionando como um filtro passa-baixa (PB), podendo ser um filtro passivo de 1^o ou 2^o ordem. Um exemplo de filtro PB e dos sinais modulado e demodulado pode ser visto nas ilustrações 3-5 e 3-6.

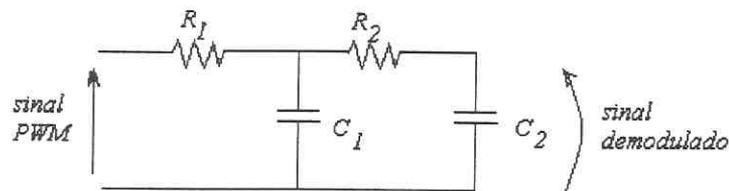


ilustração 3-5: Filtro passa-baixa de 2^o ordem

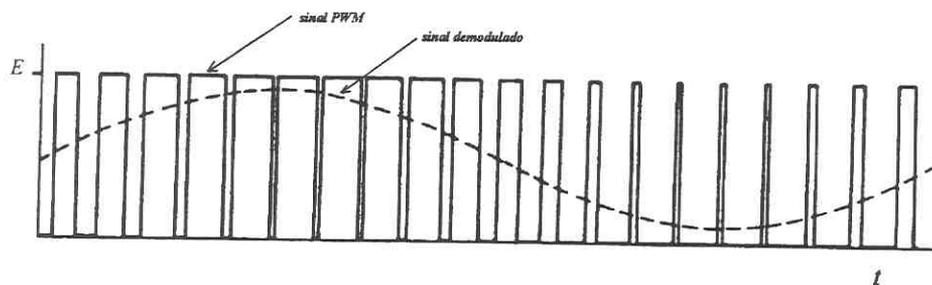


ilustração 3-6: Formas de onda do sinal PWM e demodulado

A resposta do sistema com sinal modulado (PWM) após a filtragem é uma onda tipo senoidal com amplitudes relativas a largura do pulso de modulação. Antes da filtragem do sinal, este contém uma representação da tensão analógica desejada. Para um ciclo PWM variando de 00h a FFh em 32 valores; o ponto inicial 00h é a tensão mínima do sinal analógico obtido, o pulso de saída 16 do PWM (50%) representará a metade da escala total do sinal e o último valor representa a tensão máxima do sinal analógico de referência (5V). O sinal demodulado isolado está mostrado na ilustração 3-7.

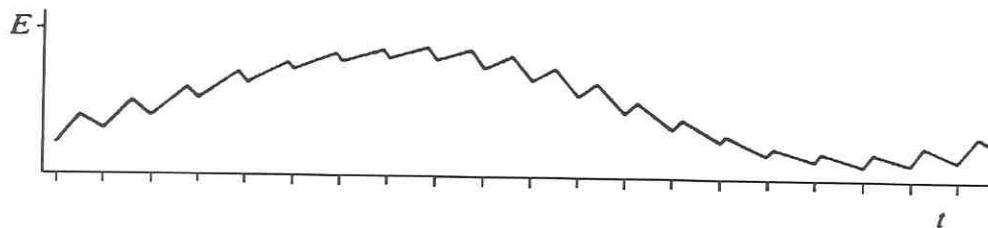


ilustração 3-7: Sinal filtrado de saída do PWM

Com base nesta teoria, foi realizado digitalmente o circuito PWM das tensões de referência do *driver* e com isto obtido um sinal em meia onda senoidal, que permite ao motor girar com curvas de corrente suaves, implementando o micropasso.

3.5 Conclusão

Este capítulo tratou da forma de implementação do micropasso, através do controle de corrente das bobinas de fase do motor baseado na geração de um sinal senoidal de referência para o *driver*. Acrescentou-se duas maneiras possíveis de gerar este sinal: ou pela conversão D/A ou por modulação PWM. A modulação PWM foi detalhada em vista da complexidade teórica. A partir da conclusão deste estudo teórico sobre a execução do projeto, passa-se a discussão dos componentes que são utilizados para a composição do sistema.

Capítulo IV

O Circuito *Driver*

4.1 Introdução

Conforme tratado nos capítulos anteriores, o acionamento do motor de passo foi projetado para utilizar um *driver* bipolar, sendo que este foi implementado através de um circuito integrado dedicado, o 3952, que é um componente projetado para controle de corrente bidirecional PWM de cargas indutivas.

O 3952 é capaz de manter uma corrente contínua de saída de 2A e uma tensão de operação de 50V. Apresenta também um circuito interno de controle baseado no tempo fixo Toff de um PWM que pode ser utilizado para regular a máxima corrente de carga num valor desejado (*driver chopper* – veja seção 2.3.2.2). O limite da corrente de pico da carga é selecionado pelos valores de uma entrada de tensão de referência e de um resistor externo, de acordo com a necessidade do usuário. O tempo Toff do pulso é selecionado através de uma constante RC externa. Circuitos internos de proteção incluem *shutdown* térmico com histerese, diodos de supressão de transiente e proteção contra alta corrente. Não são necessárias seqüências especiais de ligação.

As principais características do 3952 podem ser resumidas em :

- Taxa de corrente de saída de $\pm 2A$ contínuos
- Taxa de tensão de saída de 50V
- Controle interno de corrente por PWM
- Modos de decaimento de corrente rápido e lento
- Modo *sleep* (baixo consumo de corrente)
- Diodos internos de supressão de transiente

- Circuito interno de *shutdown* térmico
- Proteção à sobrecorrente

Este capítulo está reservado a discussão das funcionalidades do 3952 e a análise de seus circuitos internos que podem ser utilizados para melhorar a qualidade do sinal de micropasso gerado.

4.2 Aspectos Funcionais do 3952

A descrição funcional das propriedades do 3952 descritas neste capítulo estão baseadas no diagrama de blocos funcional do circuito integrado, que se encontra na ilustração 4-1.

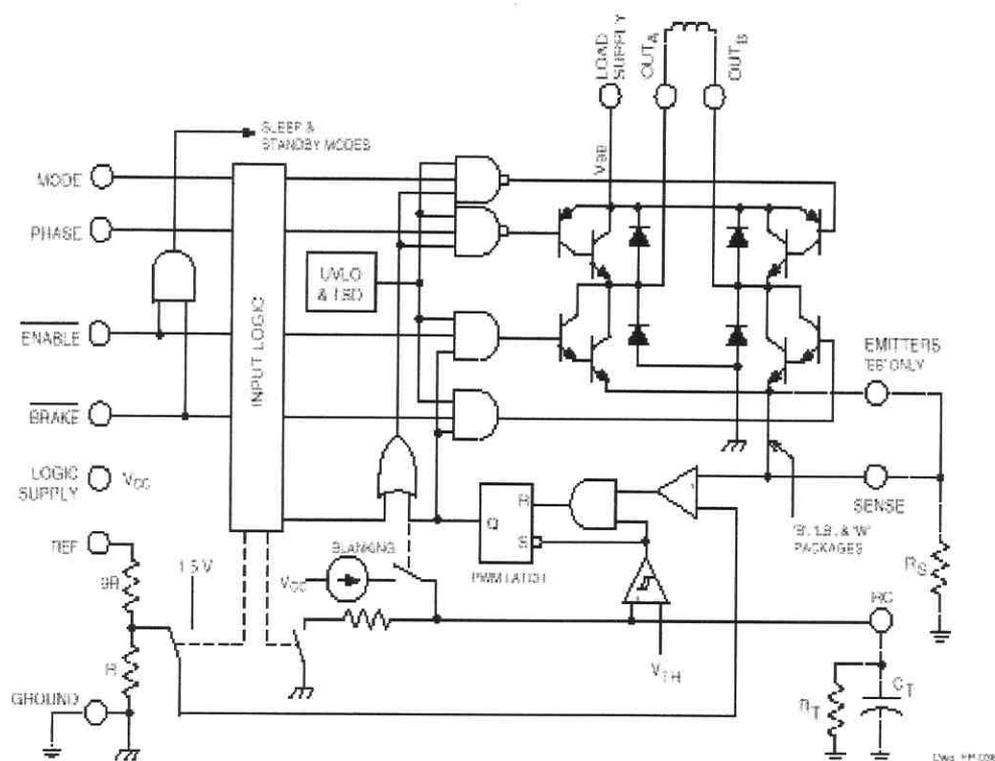


ilustração 4-1 : Diagrama funcional do 3952

Como se pode ver pelo diagrama funcional, o *driver* 3952 conta com importantes entradas, a saber: ENABLE, PHASE, BRAKE e MODE que permitem gerar nas saídas para a conexão da bobina do motor (OUTA e OUTB) diferentes situações de comando de uma

fase do motor. Para acionar o motor de passo híbrido são necessários dois 3952, um para cada fase.

Com a entrada ENABLE em nível baixo, a entrada PHASE controla a polaridade da corrente da carga associando um par apropriado de transistores fonte e dreno do *driver*. A entrada MODE determina quando o circuito PWM de controle de corrente opera no modo de operação de baixa taxa de decaimento da corrente (somente chaveando o dreno selecionado) ou no modo de operação de alta taxa de decaimento da corrente (chaveando fonte e dreno). Com a entrada ENABLE em nível alto, todas as saídas do *driver* são desabilitadas. Quando se aplica lógica alta às entradas ENABLE e MODE, se coloca o componente no modo *sleep* para minimizar o consumo de corrente quando não está em uso.

Quando um nível baixo é aplicado na entrada BRAKE, a função de parada é habilitada. Isto faz com que o ENABLE e PHASE desliguem as fontes do *driver* e liguem os drenos. Em aplicações usando motores CC, este efeito resulta num fluxo de corrente que pára o motor dinamicamente.

4.3 Controle de Corrente através do PWM

Conforme citado, o 3952 contém um circuito de controle de corrente baseado no tempo T_{off} de um PWM usado para limitar a corrente de carga em um valor desejado. O valor da limitação da corrente (I_{trip}) é obtido através de um resistor externo (R_s) e do nível de tensão da entrada de referência (V_{ref}). O circuito interno compara a tensão através do resistor externo com um décimo da tensão no pino de entrada REF, resultando na função

$$\text{aproximada de : } I_{trip} = \frac{V_{ref}}{(10 \cdot R_s)}.$$

No movimento horário ou anti-horário, o circuito de controle de corrente limita a corrente de carga. Quando a corrente atinge I_{trip} , o comparador apaga o valor do PWM *latch* para zerar o dreno do *driver* selecionado (no modo de decaimento lento) ou o par fonte/dreno (no modo de decaimento rápido). No modo de decaimento lento, o dreno do *driver* selecionado está desabilitado, a indutância de carga faz a corrente recircular através do

transistor fonte e do diodo de roda livre. No modo de decaimento rápido, o par fonte/dreno selecionado está desabilitado; a indutância de carga faz a corrente fluir através do terra e dos diodos de roda livre até a fonte V_{BB} . Isto pode ser melhor acompanhado pelo diagrama dos caminhos da corrente na ilustração 4-2.

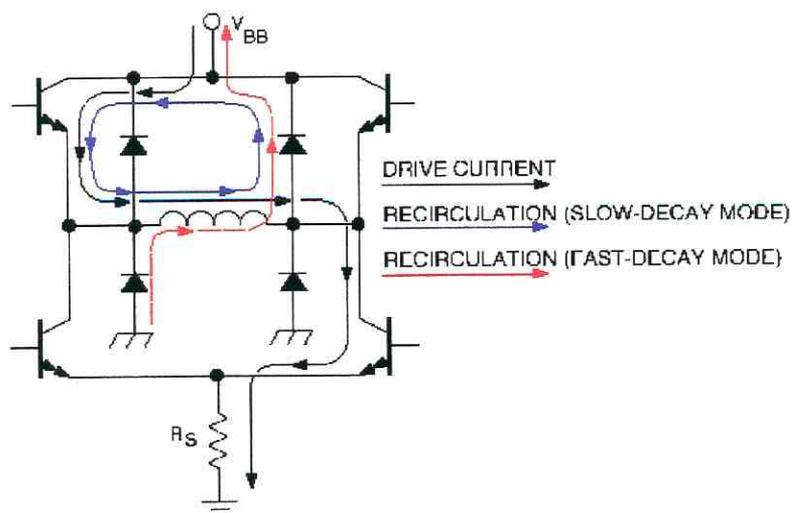


ilustração 4-2: Correntes circulando no *driver* do 3952 para os diferentes modos de decaimento

O usuário deve selecionar o resistor externo (R_t) e o capacitor (C_t) para determinar o período de tempo ($T_{off} = R_t.C_t$) durante o qual os *drivers* permanecem desabilitados. No fim de cada intervalo $R_t.C_t$, os *drivers* são reabilitados permitindo a corrente de carga crescer novamente. O ciclo PWM se repete, mantendo a corrente no valor desejado (ver a ilustração 4-3).

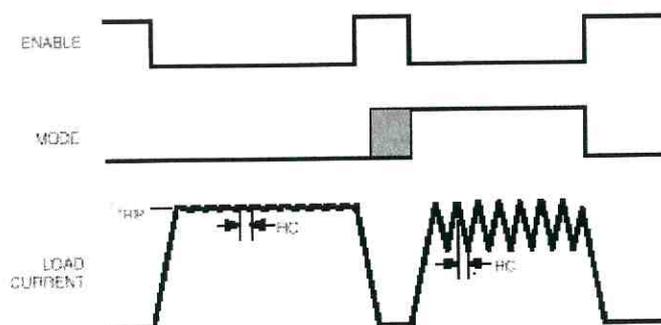


ilustração 4-3: Formas de onda da corrente de carga para diferentes modos de decaimento

Além de determinar o período do tempo T_{off} do circuito de controle PWM, o componente C_t seleciona o tempo sem ocorrências de eventos (*blanking time*) do comparador. Esta função mantém a saída do comparador em vazio quando as saídas são chaveadas por um circuito de controle de corrente (ou pelas entradas PHASE, BRAKE ou ENABLE). A saída do comparador permanece em vazio para prevenir detecções falsas de sobrecorrente devido a correntes reversas circulando pelos diodos ou/e transientes relacionados a capacitância distribuída na carga.

Em aplicações que utilizam o modo de decaimento rápido, o valor mínimo recomendado para o capacitor C_t é de $1200\text{pF} \pm 5\%$. Para outras aplicações, o mínimo deve ser de $820\text{pF} \pm 5\%$. Estes valores certificam que o tempo sem eventos é suficiente para proibir valores falsos do comparador em condições normais de operação. Para uma regulação ótima da corrente de carga, os valores de C_t devem estar entre os recomendados e o valor de R_t escolhido para determinar o T_{off} desejado.

Importante observar que para valores crescentes de T_{off} , as perdas devido ao chaveamento decrescem, a regulação da corrente de carga melhora, a frequência do PWM diminui e a oscilação da corrente aumenta. O tempo T_{off} do PWM pode ser selecionado para otimizar estes parâmetros. Os valores típicos de T_{off} se encontram entre 15 e 35 μs .

4.4 Aplicação do 3952 com Motor de Passo

O terminal MODE pode ser usado para otimizar a performance do dispositivo em aplicações de *drivers* para micropassos. Quando a corrente média da carga está aumentando, o modo de decaimento lento é utilizado para limitar as perdas de chaveamento no dispositivo e as perdas no metal do motor. Isto também melhora a máxima taxa na qual a corrente de carga pode aumentar (quando comparado ao decaimento rápido) devido a baixa taxa de decaimento durante T_{off} (determinado pela constante externa RC). Quando a corrente média da carga está diminuindo, o modo de decaimento rápido é utilizado para regular a corrente da carga até o nível desejado. Isto previne que o perfil da corrente causado pelas forças contra-eletromotriz do motor de passo seja seguido.

Em aplicações com motor de passo onde se deseja uma corrente constante na carga (acionamento por passo completo ou meio passo), o modo PWM de decaimento lento é usado tipicamente para limitar as perdas de chaveamento no dispositivo e as perdas no entreferro do motor.

4.5 Conclusão

A discussão sobre a operação com o *driver* 3952 é crucial para a correta programação do microcontrolador. Isto ocorre porque o funcionamento esperado do sistema de controle será dependente da seleção e atualização do valores de entrada dos *drivers*. O controle de corrente do motor está baseado na programação das entradas REF, PHASE, ENABLE e MODE. O perfil de corrente sinusoidal desejado para o micropasso é obtido pelo controle da entrada REF associado ao valor da resistência R_s , segundo a ilustração 4-1.

De posse destes dados, passa-se a descrição da forma como este controle pode ser realizado por meio de um microcontrolador.

Capítulo V



O Microcontrolador

5.1 Introdução

Este capítulo apresenta uma descrição do principal componente para a implementação do sistema de comando e controle do *driver* por micropasso. É através da programação deste microcontrolador que se obtém todos os sinais a serem entregues ao *driver*, apresentado no capítulo anterior, a fim de acionar o motor.

5.2 Justificativa de Escolha

A escolha de realizar a programação do sistema de acionamento através dos microcontroladores da *Microchip* se deve a evolução tecnológica que estes componentes representam. Os microcontroladores PIC16/17 provêm uma combinação única da alta performance dos processadores RISC com a tecnologia a baixo custo *One-Time-Programmable* (OTP). A OTP traz benefícios quanto ao custo de implementação quando comparado a soluções encontradas em ROM. Estes benefícios incluem: rápido tempo para mercado, facilidades de alterações em código, habilidade para gerar soluções adaptáveis às necessidades do consumidor final, redução de estoque, redução de trabalho em processo.

No estágio atual da manufatura, a redução do ciclo de vida do produto pode levar a grandes estoques e aumento do tempo de produção. Usando a tecnologia tradicional de microcontroladores baseada em ROM limita-se a habilidade de responder ao mercado com soluções gerais para clientes específicos. Usando a tecnologia OTP, resolve-se este

problema: pode-se ter um estoque mínimo, efetivamente mantido por utilizar o mesmo componente para diversas aplicações. Assim, os custos podem ser reduzidos pelo volume utilizado (*purchasing*). Pode-se reduzir o estoque em processo e manter um estoque de segurança para produtos usando um componente padrão.

A OTP é a tecnologia de manufatura flexível no mundo dos microcontroladores. Com a competição mundial, a procura por produtos específicos aumenta. Ter habilidade para se adaptar a mudanças pode ser uma grande vantagem competitiva. Hoje em dia, um microcontrolador programável apresenta muitas facilidades implementáveis por *software* que podem ser mais rapidamente adaptáveis a alterações do que soluções feitas em *hardware*.

A família PIC16C6X é principalmente utilizada em aplicações de alta velocidade (tempo real requerido) e de controle, que pode ser realizado por meio de sensores, teclado ou interrupções. Baixo custo e potência, alta performance, facilidade de uso e flexibilidade de recursos I/O fazem do PIC16C6X ideal para a aplicação no acionamento microcontrolado do motor de passo. Neste projeto, utilizou-se o microcontrolador PIC16C65 desta família, que será tratado a seguir.

5.3 O Microcontrolador PIC16C65

Neste, far-se-á uma descrição detalhada do microcontrolador PIC16C65, analisando suas principais características, a arquitetura interna bem como os principais módulos que foram utilizados para execução deste projeto.

5.3.1 Características do PIC16C65

A família de microcontroladores de 8 bits - PIC16C6X - tem como vantagens o baixo custo, alta performance e estabilidade, além da utilização de tecnologia CMOS. Os microcontroladores desta família são baseados na arquitetura RISC (*Reduced Instructions Set Computer*), possuindo apenas 35 instruções para programação. Dois estágios de instruções em *pipeline* permitem executar todas as instruções em um único ciclo, exceto em caso de saltos (que requerem dois ciclos). Os barramentos de instruções e dados são

separados numa arquitetura que compreende instruções de 14 bits e dados de 8 bits. O PIC16C65 possui uma RAM interna de 192 bytes e 33 pinos de I/O. Contém ainda um *stack* com oito níveis de profundidade, além de múltiplas fontes de interrupções internas e externas. Complementarmente possui um grande número de registradores que contribuem com as inovações da arquitetura e que são utilizados para a obtenção de alta performance. Comparando com os controladores de 8 bits desta classe, os PIC16C6X provém uma compressão do código de 2:1 e um aumento de velocidade de 4:1 [Microchip, 1993].

Este microcontrolador dispõe de vários dispositivos periféricos como contadores/temporizadores, módulos PWM e portas seriais. A porta serial síncrona pode ser configurada como Interface Serial Periférica (SPI) de 3 fios ou como uma interface I²C (*Inter-Integrated Circuit*) de 2 fios. A porta serial de interface de comunicação (SCI) pode ser configurada como síncrona ou assíncrona (USART). Esta família pode ter ainda canais de alta velocidade para conversores A/D com 8 bits de resolução, sendo estes suficientes para aplicações que requeiram uma interface analógica de baixo custo, como em controle de temperatura, sensoriamento de pressão, etc.

A família PIC16C6X possui ainda características especiais que reduzem componentes externos, o que diminui custos, aumenta a confiabilidade do sistema e reduz o consumo de energia. O modo SLEEP oferece economia de energia e o usuário pode “acordar” o microcontrolador através do *reset* e de diversas interrupções internas ou externas. Um *Watchdog Timer* com o seu próprio oscilador RC é altamente confiável, garantindo ao *software* proteção contra travamento.

A versão de encapsulamento *UV-erasable CERDIP* – reutilizável após a exposição a raios ultra-violeta - é ideal para o desenvolvimento de código, enquanto que o custo efetivo da versão OTP torna viável a produção em qualquer volume.

Esta família de microcontroladores se adapta perfeitamente em aplicações que visam segurança, sensoriamento remoto e aplicações para controle de movimento. A tecnologia EPROM torna as aplicações extremamente rápidas e conveniente (transmissão de códigos, controle de velocidade de motores, receptor de frequências, etc.). O reduzido tamanho do seu encapsulamento torna-o perfeito para todas as aplicações de espaço.

Todas estas características aliadas a facilidade de utilização e a flexibilidade de I/O tornam o PIC16C65 muito versátil até mesmo em áreas onde microcontroladores não são convencionalmente utilizados (por exemplo, funções de temporização, comunicação serial, funções de captura, comparação, PWM e aplicações com coprocessadores).

Algumas especificações do PIC16C65 serão rapidamente tratadas a seguir.

5.3.1.1 Frequência do *Clock*

O PIC16C6X apresenta quatro opções de osciladores; o usuário pode programar 2 bits de configuração (FOSC1 e FOSC2) para selecionar um destes modos:

- LP - *low power crystal* – oscilador que minimiza o consumo de energia;
- XT- *crystal/resonator* - um cristal padrão até 4 MHz;
- HS - *high speed crystal/resonator* - para cristais de alta velocidade (variando de 8MHz a 20 MHz)
- RC - *resistor/capacitor* - provê uma solução de baixo custo com um arranjo de resistores e capacitores.

Recomenda-se o uso de dois capacitores em paralelo ao oscilador, cujos valores dependem das frequências de *clock* utilizadas e se encontram tabelados na folha de dados dos componentes da *Microchip*.

5.3.1.2 Esquema do *Clock* e Ciclo de Instruções

A entrada do *clock* (do OSC1) é internamente dividida por quatro para gerar 4 sinais de relógio quadrados Q1, Q2, Q3 e Q4. Um ciclo de instruções consiste de 4 ciclos de sinais (Q1 a Q4). Um ciclo de busca começa com o PC incrementando em Q1, sendo a instrução trazida da memória de programa e armazenada no registro de instruções (IR) em Q4. No ciclo de execução, em Q1 a instrução está guardada em IR. Esta instrução é então decodificada e executada durante Q2, Q3 e Q4. A memória de dados é lida durante Q2 (leitura do operando) e escrita em Q4 (escrita do destino).

Isto implicaria em gastar dois ciclos para execução de uma instrução. Entretanto, devido ao

pipeline, cada instrução é efetivamente executada em um único ciclo. Se uma instrução faz com que o contador de programa tenha que ser mudado (por exemplo, GOTO) então dois ciclos são necessários para completar a instrução.

5.3.1.3 Organização da Memória

Memória de Dados

A memória de dados é particionada em dois bancos que contém os registros de propósito geral e de funções especiais. Cada banco se estende até 7Fh (128 bytes). A seleção entre o banco 0 ou 1 depende do estado do bit RP0 no registro de STATUS: banco 0 quando o bit RP0 está zerado e banco 1 quando está em um. As posições mais baixas de cada banco são reservadas a registros de funções especiais. Os registros de funções especiais são registros de propósito geral implementados como RAM estática. Por não ser escopo deste projeto o detalhamento do microcontrolador utilizado, cada registro que aparecer no contexto deste trabalho será tratado particularmente.

Memória de Programa

A família PIC16C6X tem um contador de programa de 13 bits capaz de endereçar uma memória de programa de 8KX14. Para o PIC16C65, apenas os primeiros 4KX14 estão fisicamente implementados. O vetor de RESET está na posição 0000h e o vetor de interrupção na posição 0004h .

Stack

O PIC16C6X tem um *hardware* de pilha (*stack*) de 8 de profundidade X 13 de largura. A área do *stack* não faz parte da memória de programa ou dados e o *stack pointer* (apontador da pilha) não pode ser lido ou escrito. O PC é colocado na pilha quando uma instrução CALL é executada ou uma interrupção é aceita pelo microcontrolador. O PC é recuperado em um evento de RETURN, RETLW ou um RETFIE, retorno de interrupção. Na verdade, no PIC16/17 não existem instruções de PUSH e POP - armazenamento e recuperação de dados na pilha. Estas ações ocorrem automaticamente na execução de CALL, RETURN, RETLW, ou RETFIE, ou no atendimento de interrupção. Em outros casos, registros que se deseja manter inalterados devem ser salvos em outros registradores e restabelecidos após a o atendimento da subrotina ou interrupção.

O *stack* opera em um buffer circular. Isto significa que após se ter armazenado oito valores na pilha em chamadas a subrotinas, a nona chamada irá sobrescrever o valor armazenado no primeiro nível do *stack*, a décima no segundo e assim por diante. É importante lembrar que o PIC16/17 não apresenta bits de status para indicar que o *stack* está cheio (situação de *overflow*) ou vazio (*underflow*).

Veja o mapa de memória de programa e a área de pilha na ilustração 5-1.

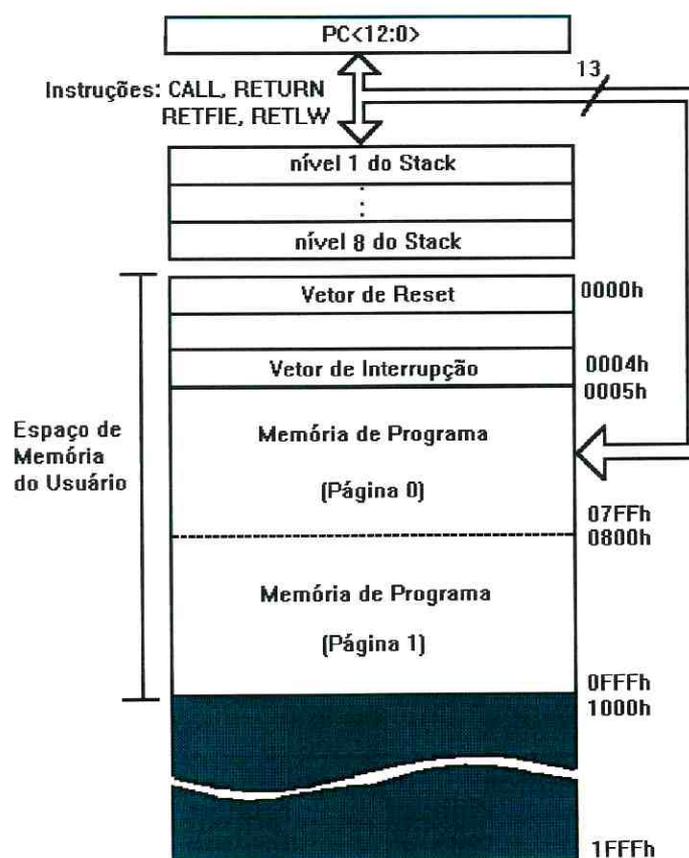


ilustração 5-1: Mapa de memória de programa e *stack* do PIC16C65

5.3.2 Arquitetura

A alta performance do PIC16C65 pode ser atribuída às características da arquitetura RISC por ela adotada. Para começar, esta família utiliza uma arquitetura *Harvard*, na qual, programas e dados são acessados de memórias separadas utilizando barramentos

separados, ao contrário da tradicional arquitetura de *von Neumann* onde os programas e os dados são capturados de uma mesma memória usando um único barramento. Separando o barramento de dados e instruções, pode-se então ter instruções de tamanho diferente da palavra de 8 bits. Os códigos de instrução têm largura de 14 bits, possibilitando que todas as instruções sejam formadas por uma única palavra. As instruções são trazidas da memória em um único ciclo. Um *pipeline* de dois estágios sobrepõem a busca e execução das instruções e, conseqüentemente, todas as 35 instruções são executadas em ciclo único, exceto para instruções que levam a saltos do programa, com alteração do PC.

O PIC16C65 endereça 4k de memória interna de programa com largura de 14 bits. Ele pode endereçar direta ou indiretamente seus arquivos de registradores ou a memória de dados. Todos os registradores de função especial, incluindo o PC são mapeados na memória de dados. O PIC16C65 possui instruções selecionadas de forma ortogonal (simétrica) o que torna possível carregar qualquer operação ou registro usando qualquer modo de endereçamento. Esta natureza simétrica e a falta de situações de otimização especial fazem da programação do PIC16C65 muito simples e eficiente.

Este dispositivo contém uma ULA de 8 bits e um registrador de trabalho. A ULA é uma unidade aritmética de propósito geral; ela realiza funções aritméticas e booleanas entre dados no registrador de trabalho e qualquer registro de arquivo. Pode realizar operações de adição, subtração, deslocamento e operações lógicas (as operações aritméticas são realizadas em complemento dois). Nas operações de dois operandos, tipicamente um operando é o registro de trabalho (*work register* ou W), o outro operando é um registrador de arquivo ou uma constante imediata. Em instruções de um único operando, este poderá ser o registro de trabalho ou um registro de arquivo. O registro W é um registro de 8 bits de trabalho usado nas operações da ULA, e não é endereçável. Dependendo da instrução executada, a ULA poderá afetar o valor dos bits de *Carry* (C), *Digit Carry* (DC) e *Zero* (Z) no registrador de Status.

A ilustração 5-2 mostra o diagrama de blocos da arquitetura do PIC16C65.

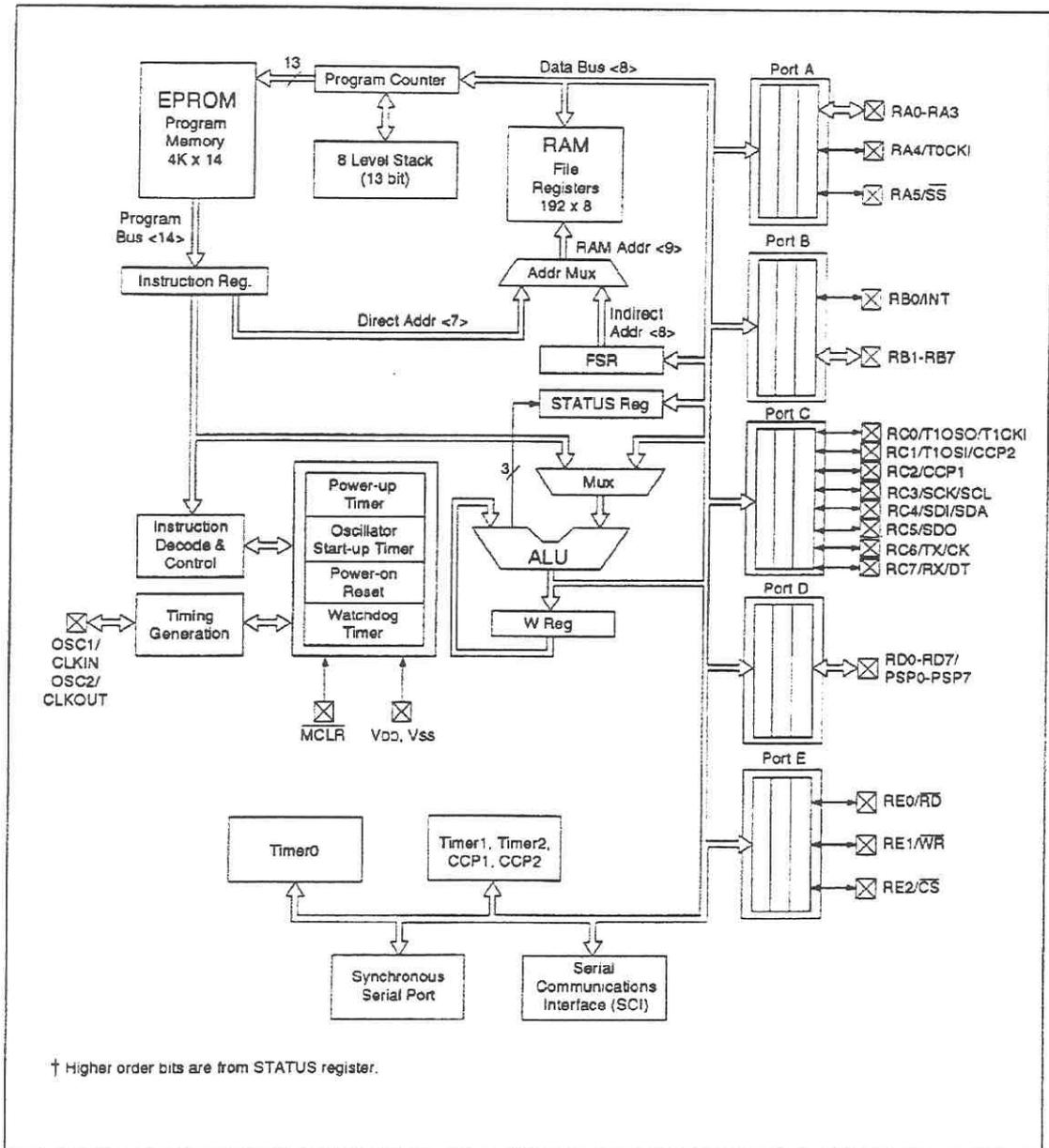


ilustração 5-2: Diagrama de blocos da arquitetura do PIC16C65

5.3.3 Principais Módulos Utilizados

Os módulos estudados e utilizados no desenvolvimento deste projeto serão brevemente abordados a seguir. Será detalhado o modo PWM em vista de sua importância para a execução do sistema de controle do *driver*.

5.3.3.1 Portas de Entrada e Saída

O PIC16C65 apresenta 5 portas de I/O, de A a E, cujos pinos podem ser multiplexados com outras funções periféricas. Serão tratadas aqui as características básicas de cada porta, se detendo principalmente nos aspectos especiais das portas PORT_B e PORT_C, utilizados no desenvolvimento do acionamento por micropasso.

PORT_A

É uma porta de 6 bits, onde estes podem ser configurados como entrada ou saída pelo registro especial TRISA, sendo o valor binário um correspondente a função de entrada e zero à saída.

O pino RA4 (bit 4) pode ser uma entrada tipo *Schmitt Trigger* ou uma saída coletor aberto e está multiplexada à entrada de *clock* do temporizador TMRO.

Todos os outros bits tem entrada TTL e *drivers* de saída CMOS.

PORT_B

É uma porta bidirecional de 8 bits, podendo ser selecionados como entrada ou saída através do registro TRISB.

O pino RB0 pode ser utilizado como um pino de entrada de interrupção externa.

PORT_C

É uma porta bidirecional de 8 bits, onde cada bit pode ser configurado individualmente como entrada ou saída através do registro TRISC.

O PORT_C está multiplexado com diversas funções periféricas, como os módulos PWM, apresentando *buffers* de entrada *Schmitt Trigger* em todos os pinos.

PORT_D e PORT_E

São portas com *buffers* de entrada *Schmitt Trigger*, que podem ser selecionados como entrada ou saída através do registro TRISD e TRISE, respectivamente. A PORT_D apresenta 8 bits e a PORT_E somente 3. Ambas podem ser utilizadas para implementar uma porta paralela (*parallel slave port*).

5.3.3.2 Módulos de Temporização

O PIC16C65 tem três módulos de temporização, o TMR0, o TMR1 e o TMR2. Cada módulo pode gerar uma interrupção para indicar que um evento ocorreu (i.e. *timer overflow*). Ainda existem dois outros módulos que podem ser usados em conjunto com o TMR1 ou TMR2, que são os módulos de Captura/Comparação/PWMx, ou simplesmente, os CCPx. (x pode ser interpretado como 1 ou 2, dependendo de qual dos dois módulos se está fazendo uso).

A seguir, far-se-á uma breve descrição dos temporizadores, detendo-se principalmente no módulos CCP1 e CCP2 que foram empregados neste projeto.

O Timer TMR0

O TMR0 é um simples contador de 8 bits, acionado por um estouro na contagem de tempo (*overflow*). O relógio de contagem pode ser o próprio *clock* do sistema (OSC/4) ou um externo, sendo a máxima frequência permitida de 50 MHz. A interrupção é gerada quando o contador/temporizador ultrapassa de FFh para 00h, reiniciando a contagem de 8 bits.

O Timer TMR1

O TMR1 é um contador/temporizador de 16 bits. O relógio pode ser o *clock* interno do sistema (OSC/4), um *clock* externo ou um cristal externo. Quando operando como contador, o TMR1 pode operar sincronamente ou assincronamente ao controlador (*clock* é externo). A operação assíncrona permite o TMR1 operar durante o modo de redução de potência (*sleep*) do PIC16C65, o que é útil em aplicações que necessitem de tempo real bem como economia de energia.

Quando usado em conjunto com os módulos de captura/comparação/PWM, o TMR1 deve estar sincronizado ao PIC16C65. Este caso, o TMR1 serve como base de tempo de 16 bits para realizar a captura ou comparação.

O Timer TMR2

O TMR2 é um temporizador de 8 bits. Quando usado em conjunto com os módulos de

captura/comparação/PWM, este temporizador serve de base de tempo para a operação com o modo PWM.

Os Módulos CCPx

Os módulos CCPx podem operar de três modos: captura de 16 bits, comparação de 16 bits e PWM (*pulse width modulation*) com até 10 bits.

O modo captura coloca o valor de 16 bits do TMR1 no par de registros CCPxH: CCPxL a cada vez que ocorre um evento no pino CCPx.

No modo de comparação, o valor do par de registros TMR1H:TMR1L (TMR1) é constantemente comparado com o valor do par CCPxH: CCPxL (CCPx). Quando uma coincidência ocorre, uma interrupção é gerada e a saída do modo CCPx é forçada a um estado (alto ou baixo, dependendo da programação) e o TMR1 é resetado.

O modo PWM compara o TMR2 a um registro de ciclo de trabalho (conhecido como *duty cycle* DC) de 10 bits bem como a um registro de período de 8 bits (PR2). Quando o valor do temporizador TMR2 atinge o do período, zera-se o TMR2, uma interrupção pode ser gerada e o pino CCPx pode ser forçado a um. Quando o temporizador TMR2 atinge o valor do registro DC, o pino CCPx é forçado a zero. O modo PWM será particularmente tratado em vista de sua utilização no projeto.

5.4 Modo PWM

Uma saída modulada por largura de pulsos PWM, mostrada na ilustração 5-3, é um sinal que apresenta uma base de tempo (período) e um tempo em que a saída permanece em nível lógico alto (ciclo de trabalho ou *duty cycle*) de 10 bits bem como a um registro de período de 8 bits (PR2). O período é um tempo fixo, repetitivo. A resolução da saída do PWM é baseada na menor partição do período na qual o ciclo de trabalho pode variar. A frequência de um PWM é simplesmente o inverso do período.

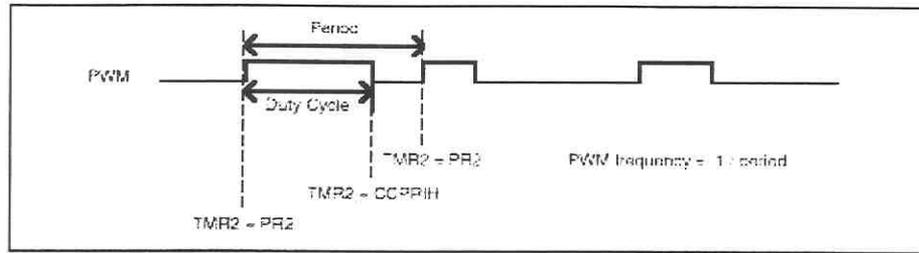


ilustração 5-3: Saída do sinal em PWM

Cada módulo CCP pode suportar um sinal de saída PWM. Este sinal PWM pode ser realizado com uma resolução de até 10 bits, sendo 8 bits provenientes do módulo de temporizador TMR2. Com isto, pode-se alcançar a 1024 passos de variação de um contador de 8 bits. Isto resulta em uma precisão máxima de T_{osc} (50 ns para um dispositivo operando a 20 MHz).

No modo PWM, o usuário escreve o valor de ciclo de trabalho de 8 bits no byte menos significativo do registro CCPRx, chamado CCPRxL. O byte mais significativo, CCPRxH funciona como um *buffer* escravo do byte CCPRxL. No modo PWM, o CCPRxH é apenas para leitura.

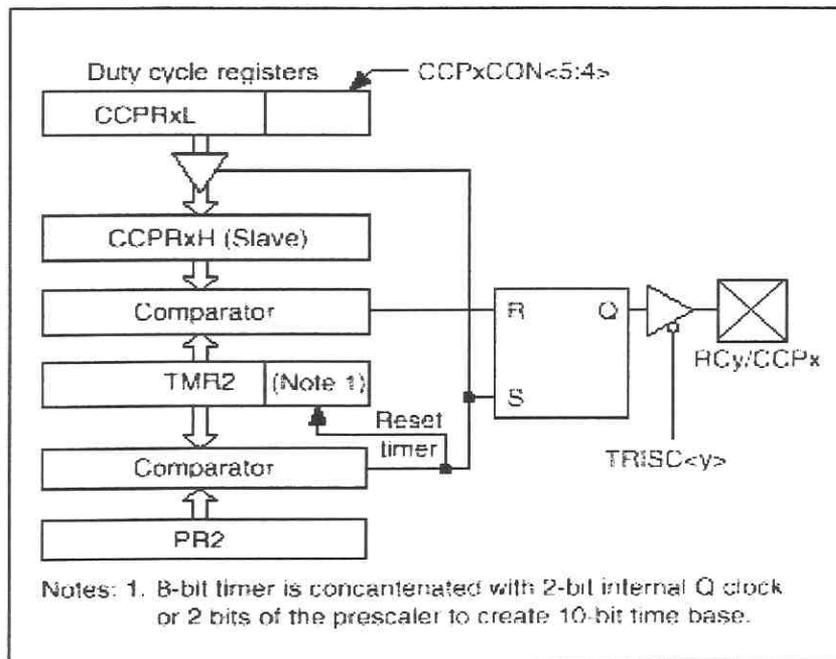


ilustração 5-4: Diagrama de blocos do módulo CCP operando no modo PWM

A ilustração 5-4 mostra o diagrama de blocos de um módulo CCP selecionado para o modo PWM. Quando o temporizador TMR2 satura, isto é, o valor do *timer* é o mesmo do registro de período (PR2), o valor de 10 bits armazenado no registro de ciclo de trabalho (CCPRxL: CCPRxCON<5:4>) é colocado no *buffer* escravo de 10 bits. Um novo valor de *duty cycle* pode ser carregado no registro DC a qualquer momento, mas só será jogado no *buffer* quando o valor do temporizador TMR2 for igual ao do registro de período.

O período do temporizador TMR2 (e do PWM) é determinado pela frequência de *clock* do dispositivo, o valor do *prescaler* do TMR2 (1, 4 ou 16) e o registro de período TMR2. As equações abaixo mostram o cálculo do período, do ciclo de trabalho e das frequências máxima e mínima do PWM.

Equação do Período:

$$\text{Período_PWM} = [(PR2)+1] \cdot 4T_{osc} \cdot (\text{Prescaler_TMR2})$$

Equação do Ciclo de Trabalho:

$$\text{Duty_Cicle_PWM} = [CCPxL : CCPRxCON < 5 : 4 >] \cdot 4T_{osc} \cdot (\text{Prescaler_TMR2})$$

Equação da Máxima Frequência:

$$\text{Frequência}(\text{Modo_Resolução_Padrão}) = \frac{1}{(PR2 \cdot 4T_{osc})}$$

$$\text{Frequência}(\text{Modo_Alta_Resolução}) = \frac{1}{(PR2 \cdot T_{osc})}$$

Equação da Mínima Frequência:

$$\text{Frequência}(\text{Modo_Resolução_Padrão}) = \frac{1}{(16 \cdot PR2 \cdot 4T_{osc})}$$

$$\text{Frequência}(\text{Modo_Alta_Resolução}) = \frac{1}{(16 \cdot PR2 \cdot T_{osc})}$$

A tabela mostrada na ilustração 5-5 mostra algumas frequências de PWM alcançadas para um microcontrolador PIC16C65, alimentado por um oscilador de 20 MHz.

Frequência do PWM	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz
Prescaler do TMR 2 (1, 4, 16)	16	4	1	1	1
Valor do PR2	0xFF	0xFF	0xFF	0x3F	0x5F
Modo de alta resolução	10 bits	10 bits	10 bits	8 bits	7 bits
Modo de resolução padrão	8 bits	8 bits	8 bits	6 bits	5 bits

ilustração 5-5: Tabela de frequências para PWM (com *clock* de 20MHz)

5.5 Interrupções

O PIC16C65 apresenta doze fontes para interrupção, sendo estas do tipo:

- Interrupção externa do pino RB0/INT
- Interrupção na mudança de nível dos pinos (RB<7:4>) da PORT_B
- Interrupção por *overflow* do temporizador TMR1
- Interrupção do temporizador TMR2 quando atinge um dado período
- Interrupção dos módulos CCP1 e CCP2
- Transmissão e recepção assíncrona SCI
- Interrupção da porta serial síncrona
- Interrupção para leitura/escrita na porta paralela

O registro de controle de interrupções (INTCON) armazena individualmente os pedidos de interrupção nos bits de *flag*. Também pode habilitar interrupções globais e individuais.

Um bit (*global interrupt enable* – GIE) habilita todas as interrupções não mascaradas ou desabilita todas as interrupções se estiver em nível lógico alto ou baixo, respectivamente. Interrupções podem também ser desabilitadas individualmente através de seus bits correspondentes no registro INTCON. O retorno de interrupção (*return from interrupt* – RETFIE) sai da rotina de interrupção e seta o GIE, reabilitando as interrupções. No caso de RESET, o GIE é zerado.

Os *flags* de interrupção do pino INT, da porta PORT_B, do TMR0 estão contidos no registro INTCON. Os *flags* de interrupções periféricas estão em um registro especial, o PIR1. Os bits correspondentes para habilitar as interrupções estão no registro PIE1 e o bit

que habilita interrupções periféricas está no INTCON.

Quando uma interrupção é atendida, o GIE é zerado para desabilitar qualquer outra interrupção. O endereço de retorno é armazenado no *stack* e o PC é carregado com 0004h, onde deve estar a rotina de atendimento de interrupção, ou um endereço de salto para a mesma. Uma vez no endereço do vetor de atendimento, a fonte da interrupção pode ser determinada pela análise dos bits de *flags* de interrupção. O bit do *flag* da interrupção deve ser limpo via *software* antes de se reabilitar as interrupções para evitar chamadas recursivas.

Durante o atendimento de uma interrupção, apenas o valor de PC é salvo no *stack*. Tipicamente, os usuários podem desejar salvar o contexto do programa durante as interrupções, por exemplo o registro W e o STATUS. Isto precisa ser implementado via *software*.

Neste projeto, a forma de interrupção utilizada foi a interrupção externa através do RB0/INT, a fim de se acionar o *driver* na frequência de sinais de entrada, enviados por um gerador de funções. Devido a isto, esta forma de interrupção será melhor detalhada.

5.5.1 Interrupção Externa através do RB0/INT

A interrupção através do pino RB0/INT é acionada na mudança do sinal externo, podendo ser selecionada como rampa de subida se o bit INTEDG (registro OPTION<6>) for um ou de descida do sinal se o bit for zero. Quando o valor do sinal no pino INT muda de estado, o bit INTF (registro INTCON<1>) vai a nível lógico um, o que é interpretado como uma chamada de interrupção. Esta interrupção pode ser desabilitada se o bit de controle INTE (registro INTCON<4>) for colocado em zero. O bit INTF deve ser limpo via *software* na rotina de atendimento de interrupção, antes de se reabilitar a ocorrência de interrupções no programa.

A interrupção através do INT também pode ser usada para “acordar” o microcontrolador do modo SLEEP, se o bit INTE for setado antes de se entrar neste modo. O estado do bit GIE decide se o processador vai ou não para o vetor de interrupções após “acordar”.

5.6 Conclusão

Este capítulo apresentou o microcontrolador usado para o gerenciamento de todas as funções e atividades a serem executadas no circuito de acionamento. O modo PWM apresentado é indispensável quando se fala de programação simples, de baixo custo (sem componentes externos ao sistema), mas com suavidade de resposta e velocidade. O entendimento da execução da rotina de interrupção é a chave para o desenvolvimento do programa pois cada micropasso só é executado se uma interrupção de entrada (sinal em pulsos numa dada frequência) for sentida pelo microcontrolador. O capítulo seguinte explica como o programa é desenvolvido.

O Desenvolvimento do Projeto

6.1 Introdução

Este capítulo tem por objetivo o desenvolvimento do circuito de acionamento para motores de passo usando o micropasso, sendo que será descrito tanto o projeto de *hardware* para acionamento bem como o projeto do *software* para atender as especificações do sistema. Esta descrição se inicia pela caracterização do ambiente de trabalho disponível para a execução do projeto. A seguir, são detalhadas as especificações requeridas durante o desenvolvimento do projeto. Na descrição de projeto de *software*, é apresentado um fluxograma da programação realizada com o microcontrolador e quanto ao projeto do *hardware* são mostrados alguns dos circuitos implementados no protótipo do sistema.

6.2 Ambiente de Trabalho

Os microcontroladores PIC16C6X são suportados por uma série de ferramentas de desenvolvimento de *hardware* e *software*. As mais utilizadas neste projeto serão descritas abaixo.

6.2.1 Placa de Demonstração de Baixo Custo PICDEM-2

É uma placa que possui dispositivos e ligações periféricas que permitem a realização de testes com os microcontroladores PIC16C63/64 e PIC16C73/74. Pode ser utilizada

isoladamente com a inserção de um *chip* já programado ou também pode ser conectada ao sistema de emulação, permitindo que os testes sejam realizados com a utilização do mesmo. Possui também uma área extra para montagem de protótipo próprio para a aplicação.

As características do *hardware* são:

- Possui soquetes de 40 e 28 pinos para cada microcontrolador citado anteriormente, podendo apenas um ser usado por vez.
- Pode ser alimentada diretamente por uma bateria de 9V ou pela fonte AC/DC de 9V, parte integrante do seu *kit*, já que a placa dispõe de um regulador de +5V. Entretanto, também permite ser ligado a uma fonte regulada de +5V externa ao circuito.
- Possui soquete e *hardware* associado para conexão direta a uma interface RS232C.
- Apresenta um potenciômetro que está conectado a um pino de entrada de um canal de conversão analógica/digital, para que testes e calibrações com o conversor possam ser realizados.
- Um oscilador RC de 2 MHz está disponível na placa. Para que um *clock* mais alto seja gerado, um jumper permite que o oscilador seja desabilitado e habilite a entrada para um cristal (até 20 MHz). Há espaço reservado a conexões dos capacitores.
- Possui chaves ligadas a pinos das portas de I/O para testes com módulos como: *timer* (CCP1/RC2), *clock* (RA4). Além disto, apresenta chave de RESET ligada ao MCLR do microcontrolador.
- Possui 8 *leds* conectados a porta PORT_B para que se possa visualizar os dados armazenados nesta.
- Um soquete para EEPROM de 128x8 também está disponível.
- Possui interface para *display* de cristal líquido.
- Apresenta possíveis conexões para acessar os barramentos.
- Como já citado, apresenta uma área para protótipo de *hardware*.

Os testes iniciais da programação a fim de se adaptar com o funcionamento do microcontrolador foram realizados utilizando a PICDEM-2, sendo posteriormente desenvolvido um protótipo específico para a execução do projeto.

6.2.2 Assembler Universal MPASM

Este programa de Assembler suporta todas as séries de microcontroladores PIC. O MPASM possui as seguintes características que assistem o desenvolvimento de *softwares* para as aplicações do usuário:

- Fazem a tradução do código fonte em Assembler para o código objeto para todos os microcontroladores PIC.
- Geram arquivos de objeto, simbólico, de listagem geral de alocação de memória e arquivos especiais para o uso em *debugger* com algum sistema de emulação.
- Suporta fontes em base hexadecimal (*default*), decimal ou octal e formatos de listagem.

O MPASM é uma linguagem de diretivas completa representada por quatro classes básicas de diretivas:

1. **Diretivas de dados:** são aquelas que controlam a alocação de memória e provém um caminho para referenciar os itens de dados simbolicamente.
2. **Diretivas de listagem:** controlam as listagens mostradas pelo MPASM. Elas permitem a especificação de títulos e subtítulos, páginas e outros controles de listagem.
3. **Diretivas de controle:** permitem a divisão condicional de códigos “assemblados”.
4. **Diretivas de macros:** controlam a execução e a alocação de dados nas macros definidas. Macro é um conjunto de instruções em *assembler* que são incluídas no código assemblado no momento da chamada desta pelo programa fonte, são diretivas definidas pelo usuário que são anexadas ao *assembler* do programa no momento em que são chamadas por este.

O programa de acionamento por micropasso foi gerado utilizando-se desde recurso de *software*, o que facilitou bastante a execução do mesmo.

6.2.3 O Programador PICSTART Plus

O PICSTART é um dispositivo programador de componentes da *Microchip* que vem acompanhado dos seguintes acessórios:

- cabo de interface RS-232 para conexão ao PC
- fonte de tensão universal
- um ambiente de desenvolvimento integrado, o MPLAB, que inclui um editor de texto

que possibilita escrever e editar os arquivos para os microcontroladores, um gerenciador de projetos para criar e trabalhar em arquivos específicos do projeto em execução, um assembler MPASM que permite montar o código sem a necessidade de sair do MPLAB e um simulador/debugador de programas, o MPLAB_SIM que simula a execução das instruções e dispositivos de I/O para microcontroladores PIC 16/17.

O programador PICSTART tem como principais características:

- programa microcontroladores PIC16/17;
- roda em Windows em conjunto com o MPLAB. No MPLAB, o usuário pode criar, editar, e mostrar os programas com os microcontroladores PIC16/17 a serem gravados com o PICSTART;
- comunica-se com o PC através de um cabo com interface RS-232.

O MPLAB é um *software* para o PICSTART Plus. Ele possibilita ao usuário um ambiente de trabalho integrado onde pode escrever seus programas no editor do MPLAB, criar projetos e assemblá-los através do MPASM. Além disto, possibilita realizar testes de execução do código e avaliar os erros de programação (*debugger*) sem a necessidade de um emulador, só com o simulador. O MPLAB também suporta o funcionamento com um sistema de emulação para testes de aplicações em tempo real.

Para a execução dos testes de programação em tempo real, utilizou-se de um microcontrolador PIC16C65 com encapsulamento tipo *UV-erasable CERDIP*, do gravador PICSTART e de um apagador de EPROM. Assim, os programas eram implementados, gravados, testados, apagados e regravados, até a obtenção da versão final.

6.2.4 Circuito Emulador de Alta Performance ICEPIC

O ICEPIC é um circuito emulador modular para os microcontroladores PIC16/17 da *Microchip*, e foi projetado para ser um produto simples para que seu funcionamento seja entendido facilmente pelos usuários e as aplicações possam se dar da forma mais rápida possível. O sistema do emulador é desenvolvido para operar em qualquer PC compatível com 386 ou melhores máquinas. Roda em Windows, permitindo ao usuário acesso a uma

grande escala de *softwares* e acessórios.

O emulador possui as seguintes características:

- Possibilita a execução dos códigos dos programas em tempo real.
- Possui memória de emulação de 8K (expansível para 16K).
- Pode realizar a emulação no modo passo a passo.
- Possui modo automático de execução passo a passo (animação).
- Possui a mesma interface para o usuário para todos os tipos de microcontroladores.
- Possibilidade de se gerar 8K de *break points* (expansível para 16K).
- Possibilidade de se gerar *watch points*.
- Pode-se alterar os valores de registros durante a execução.

O equipamento é formado dos seguintes componentes:

1. Gabinete: Carrega todos os circuitos do emulador que consistem em duas placas de circuito impresso, a placa mãe do emulador que contém todos os circuitos lógicos e a placa filha, que define o tipo de microcontrolador que está sendo emulado.
2. Placa filha personalizada: dentro do gabinete, placa filha é conectada à placa mãe. Esta placa contém o núcleo do processador de emulação que permite que as funções periféricas possam ser habilitadas e manipuladas para cada tipo de microcontrolador. Estas placas são ligadas ao computador através de uma interface serial.
3. *Software ICEPIC*: é o *software* que provém ao operador uma interface para o acionamento do microcontrolador.

Alterações em jumpers internos ao sistema permitem que este seja configurado para emular diferentes processadores. A arquitetura universal do ICEPIC permite suportar todos os microcontroladores PIC16C5X, PIC16CXX e PIC17CXX.

O principal problema encontrado na utilização do emulador foi a falta de confiabilidade na execução de programas emulados a 20 MHz, frequência necessária para gerar o micropasso. Devido a este inconveniente, os testes de execução do programa em tempo real tiveram que ser realizados através da gravação do componente no PICSTART, e não pelo sistema de emulação.

6.3 Especificações do Projeto

O projeto foi desenvolvido com base em uma série de requisitos de funcionamento listados a seguir.

- Suportar cargas de motores de passo bipolares que operem a tensões máximas de 50V e corrente de até 2A.
- Permitir alcançar velocidades de rotação de até 1000 rpm.
- Constar do equipamento as seguintes entradas:
 - chave liga/desliga para alimentação do circuito;
 - chave para habilitação dos *drivers*;
 - botão para *reset* do sistema;
 - chave para seleção de direção (sentido horário/anti-horário)
 - seleção de modo de excitação: passo completo, meio passo e micropasso;
 - entrada para o sinal de frequência do micropasso.
- O sistema deve oferecer como saídas:
 - bornes de conexão às duas fases do motor;
 - painel com *leds* indicativos para alimentação acionada, para o sinal de entrada em funcionamento, demonstrando a direção e o tipo de excitação selecionados.

Conhecidas as especificações do projeto, passou-se a execução do programa que gerencia todas as funções embutidas no sistema de acionamento para motores de passo.

6.4 Projeto do *Software*

Com base nas especificações do projeto, detalhou-se a seqüência de acionamento do sistema em fluxograma que representa o funcionamento geral do circuito e das funções detalhadas que dele fazem parte.

O *software* foi desenvolvido modularmente, ou seja, as tarefas foram resumidas a etapas bem definidas e executadas separadamente até seu pleno funcionamento, para posterior concatenação de resultados num programa único.

As etapas iniciais de programação conforme já citado foram a geração de sinais senoidais e a modulação do sinal através de PWM. Estas tarefas fazem parte de um desenvolvimento mais amplo que é a geração do passo do motor em micropassos.

Quando se opera no modo em excitação por micropasso, as tensões do motor são acionadas na seqüência de passo completo, enquanto as correntes do motor devem percorrer sinais em meia onda defasados de 90° nas duas bobinas do motor.

O fluxograma do programa de gerenciamento do circuito de acionamento de motor de passo pode ser acompanhado na ilustração 6-1.

Para a programação do PIC16C65, deve-se começar carregando os registros com a condição inicial, ou seja, ângulo zero, corrente e tensões nulas. O microcontrolador deve iniciar a execução do programa se a chave liga/desliga estiver acionada, isto significa que ele fica varrendo um bit de entrada até que seu estado varie, implicando em início de operação.

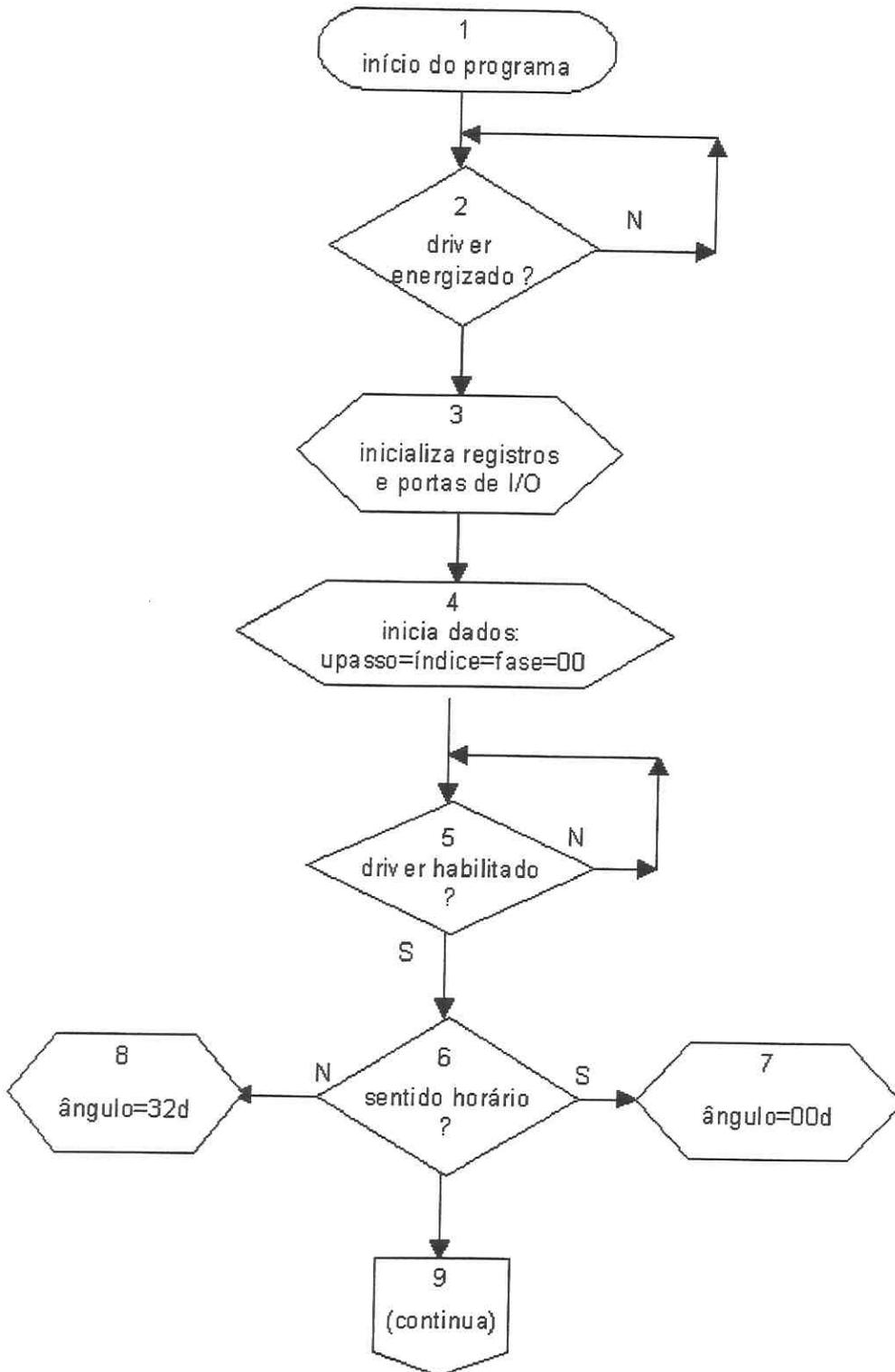
Quando o sistema já está energizado, o microcontrolador aguarda um sinal de habilitação do *driver* para começar a acionar o motor. Este sinal é reconhecido pelo fechamento de uma chave ligada ao ENABLE dos *drivers*.

O sistema apresenta ainda uma chave que permite alterar de sentido anti-horário para horário e vice-versa. O sentido de rotação é avaliado pelo estado em que se encontra um bit de entrada do microcontrolador ligado a esta chave.

Outra entrada a ser considerada é o *reset*. Este está diretamente ligado ao pino *Master Clear*, ou MCLR do PIC16C65, que produz uma reinicialização do programa em memória. O acionamento deste pino através de uma chave *push-bottom* é prioritário na execução do programa.

Com o sistema devidamente ajustado para operar, o programa inicialmente deve verificar o sentido de rotação escolhida, pois a ordem das tensões de fase é diferente. As ondas de tensão de fase do motor são implementadas conforme o modo de excitação por passo

completo, com duas fases energizadas; e estão ligados aos pinos de PHASE dos *drivers* utilizados no sistema. Conhecido o sentido de rotação, carrega-se o valor inicial das tensões de fase e das correntes do motor.



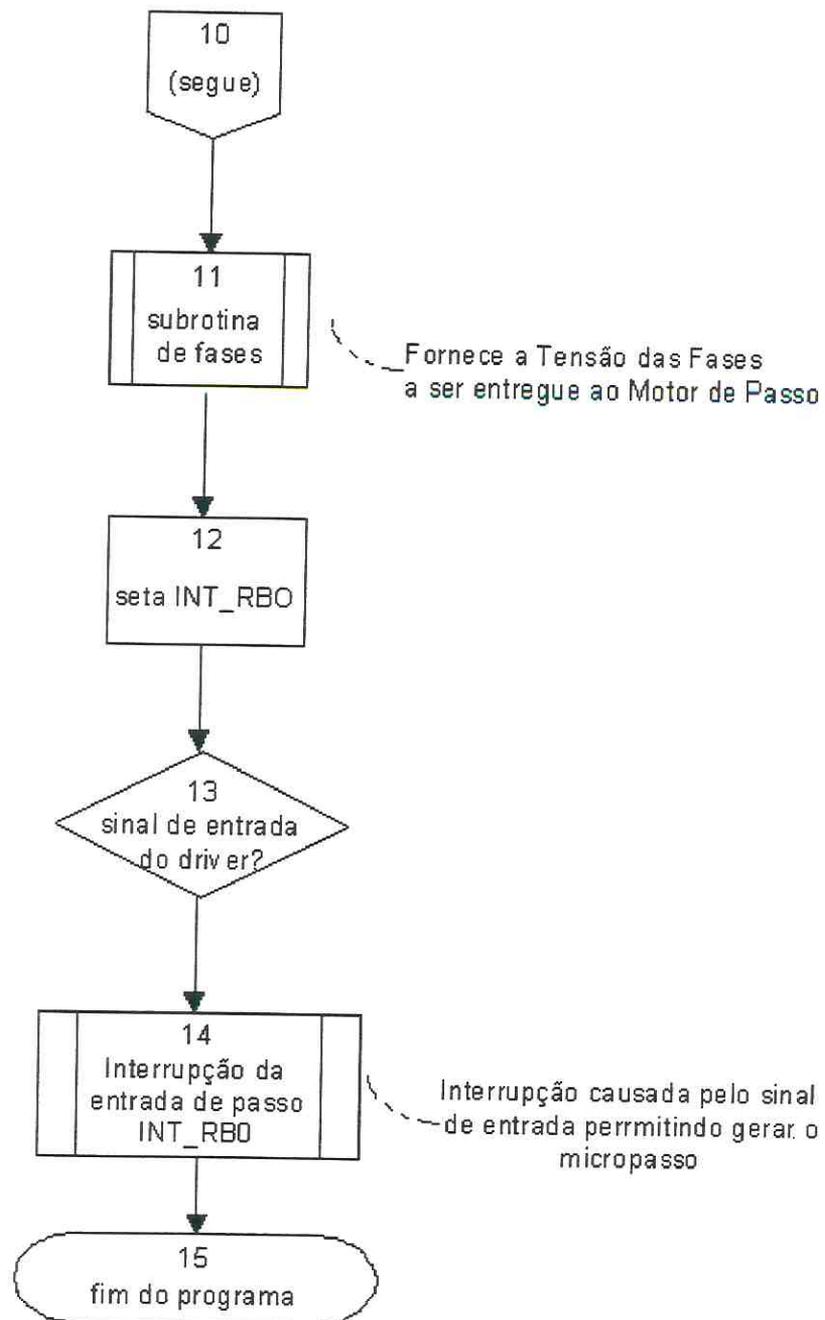


ilustração 6-1: Fluxograma do gerenciamento do *driver* de micropasso

A rotina que avalia os níveis de tensão a serem acoplados ao *driver* está baseada na seqüência de passo completo mostrada na ilustração 2-4(b). O programa segue uma tabela de valores a ser carregada nas duas fases, dependente do sentido de rotação selecionado. Esta tabela é apresentada na ilustração 6-2.

Tensões de Fase (a)	Tensões de Fase (b)
00h	00h
01h	10h
11h	11h
10h	11h

ilustração 6-2: Tabela de valores de tensões de fase nos sentidos: (a) horário, (b) anti-horário

A subrotina de fases verifica o sentido e percorre uma das tabelas com base na implementação de um ponteiro circular (índice). O fluxograma desta rotina é mostrado na ilustração 6-3.

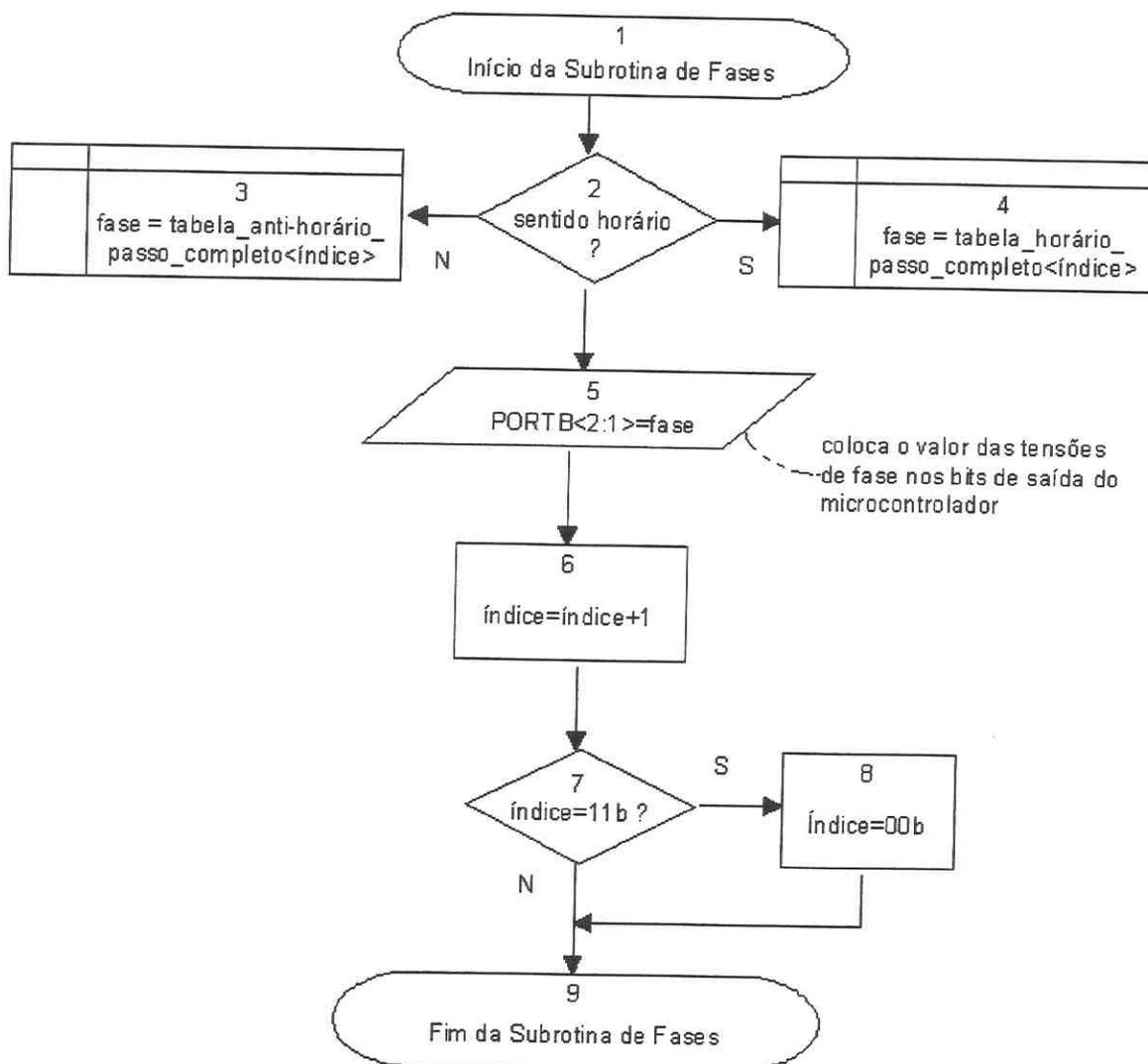


ilustração 6-3: fluxograma da subrotina de fases

Em operações com passo completo(ou meio passo), basicamente só é verificada a tabela de tensões de fase no sentido do movimento escolhido e aplicado um valor de corrente constante desejada no motor (dependente da escolha de valores de R_s e V_{ref} do *driver*). Para a operação com micropasso, a corrente do motor é variada pela aplicação de diferentes níveis de tensão de V_{ref} aplicados ao *driver* do sistema. A frequência com que o micropasso é gerado é dependente da velocidade com que a entrada de pulsos varia. A cada subida do sinal de entrada, uma interrupção é gerada no pino RB0/TNT e o microcontrolador inicia a geração dos micropassos para o sinal de corrente, internamente a execução de um passo completo de tensão de fase do motor.

A subrotina de atendimento a interrupção segue a seguinte rota: o valor do seno discretizado é pego numa tabela conforme mostrado na ilustração 3-1, e de posse deste valor passa-se a rotina de implementação do PWM ou conversão D/A. Com o sinal senoidal gerado, o ângulo é incrementado, o que implica que ocorreu um micropasso. É feito um teste para verificar se o número de micropassos já atingiu o limite (no caso, 32 micropassos por passo) para a próxima troca de fases. Se não, o programa fica aguardando nova interrupção; se a resposta é afirmativa as tensões de fase são atualizadas pela tabela de seqüência de passo completo, os valores de ângulos são zerados, até a ocorrência de uma nova interrupção. O fluxograma que implementa a subrotina de atendimento a interrupção pode ser acompanhado pela ilustração 6-4.

Pode-se verificar pelo fluxograma geral do gerenciamento do sistema de acionamento que o programa praticamente só executa uma rotina de interrupção, após a inicialização das variáveis, registros e portas de entrada/saída utilizadas pelo microcontrolador. E a limitação de máxima frequência do sinal de entrada está vinculada ao tempo de execução desta subrotina. Se o sinal de entrada variar numa frequência muito alta, corre-se o risco do microcontrolador estar atendendo a uma chamada de interrupção e com isto, ignorar a ocorrência de interrupções subsequentes, resultando na perda de passos do movimento. No próximo capítulo, será melhor descrito com este problema foi solucionado a fim de permitir o giro de 1000 rpm do motor.

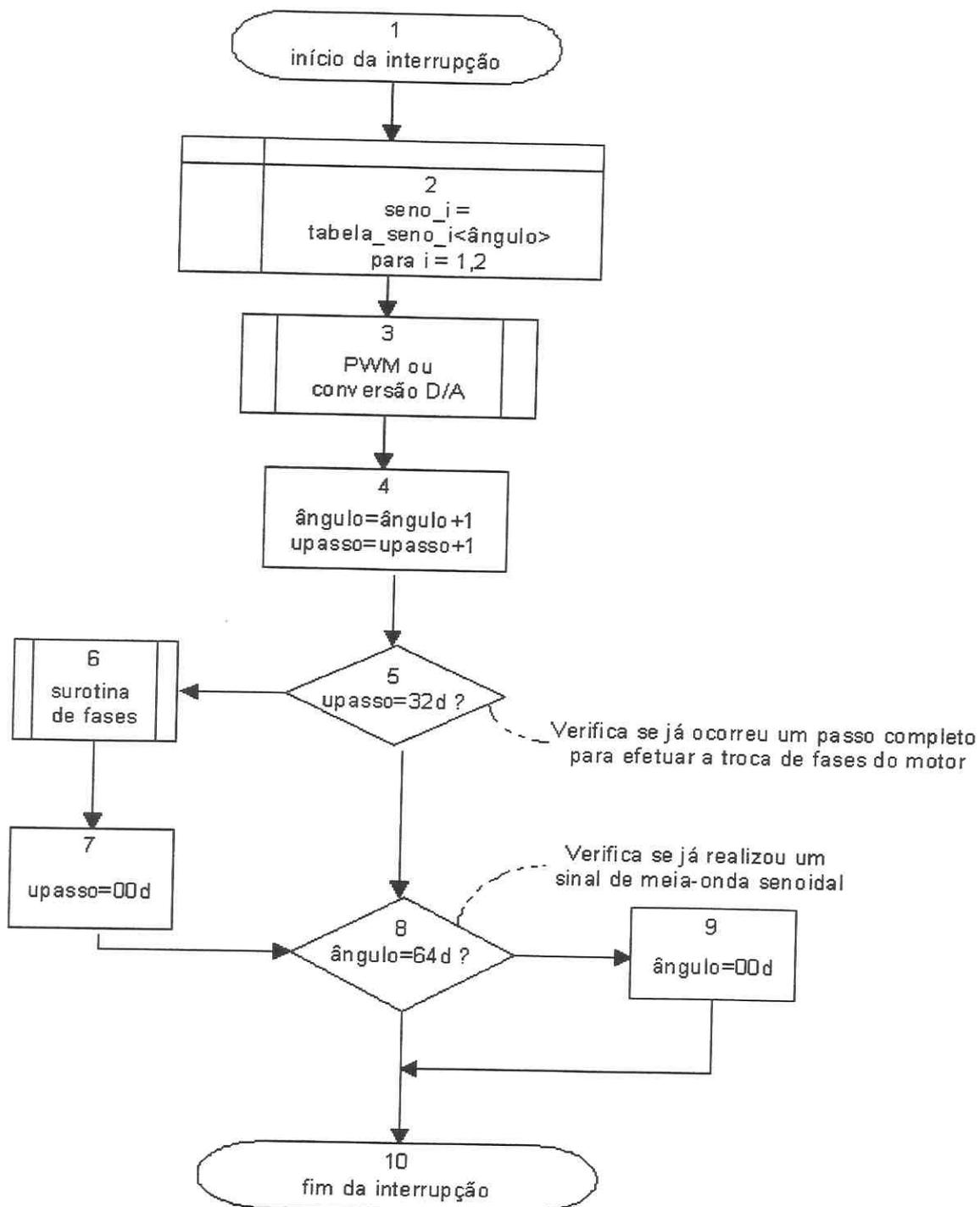


ilustração 6-4: Fluxograma da subrotina de atendimento a interrupção

Conforme analisado nos capítulos anteriores, a programação da rotina de interrupção para a geração do sinal senoidal foi implementada nas duas possibilidades estudadas para fins comparativos de resultados: por PWM e por conversão D/A.

A programação com PWM, mais compacta, pois se utiliza do próprio microcontrolador, segue o fluxograma da ilustração 6-5, baseado na programação do módulo CCP do PIC16C65 no modo PWM, descrita no item 5-4, capítulo 5.

A implementação com PWM usando o microcontrolador PIC16C65 necessita de 2 bits de saída para gerar as referências de corrente do *driver* para cada fase do motor, 2 bits para as ondas de tensão e outros dois para controlar o modo de decaimento das correntes. Cada bit que produz um sinal PWM é demodulado através de um filtro passa-baixa antes de ser utilizado para controlar a corrente percorrendo as bobinas do motor. Os sinais de tensão de referência demodulados são aplicados nas entradas de REF do circuito *driver* a fim de gerar as formas de onda de corrente senoidais.

Na programação com o conversor D/A, a diferença básica está na alteração da chamada de rotina de geração do seno. Na implementação com conversor D/A (ver fluxograma da ilustração 6-6), o valor da tensão de referência senoidal para o *driver* é enviado através das portas de I/O (PORTD e PORTE) ao barramento de dados do conversor D/A. O microcontrolador gerencia os bits de comando do conversor, programando-os para que, após o recebimento do dado capturado na tabela de senos (8 bits) efetue a conversão D/A. Baseado no valor de referência aplicado ao terminal Vref (vide anexo A-1), selecionado como 5 volts; o conversor D/A coloca na saída o valor analógico correspondente àquele valor digital. Os terminais de saída Vout dos conversores D/A estão acoplados aos pinos REF dos *drivers*, permitindo a geração da corrente senoidal para o motor.

Quanto ao modo de decaimento do *driver*, este é utilizado para melhorar a performance dos dispositivos conforme explicado no capítulo 4. A programação foi feita para que em caso de crescimento da corrente, o modo lento fosse acionado (colocando-se o pino MODE em nível lógico baixo) e o modo rápido o fosse, no caso da curva das correntes estar decrescendo (com MODE no nível alto). Com esta característica, a resposta das correntes do sistema se apresentou bem mais suave.

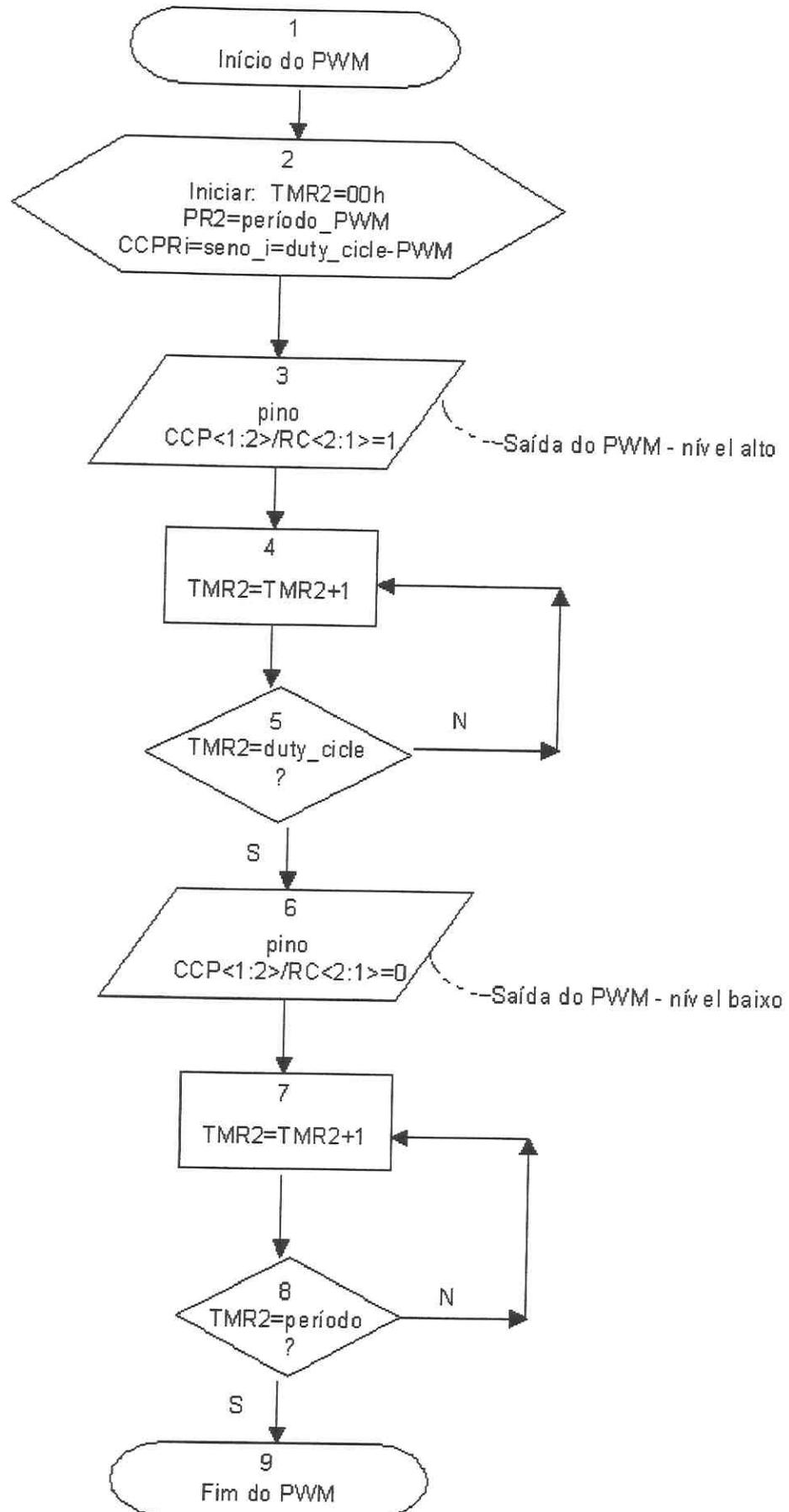


ilustração 6-5: Fluxograma da rotina de PWM

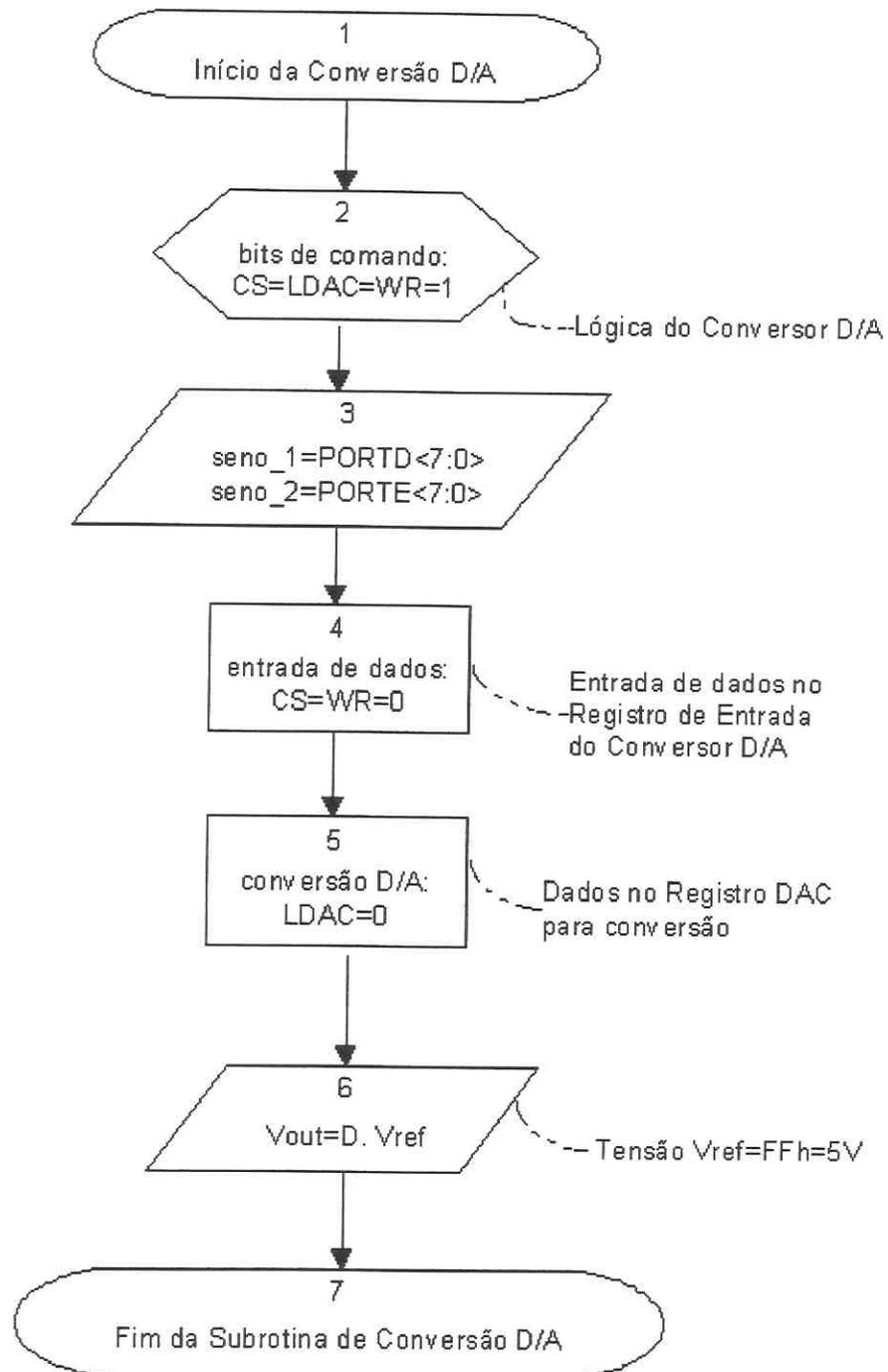


ilustração 6-6: Fluxograma de conversão D/A

6.5 Projeto do *Hardware*

Para a realização de testes em tempo real com o sistema projetado, foi desenvolvido um protótipo do circuito de acionamento por micropasso. Para confecção deste, precisou-se inicialmente projetar uma fonte de tensão contínua para alimentar o sistema. Esta fonte foi projetada para funcionar em 110/220 V e fornecer 5 V de tensão de saída para a alimentação do PIC16C65 e principais componentes, 12 V para alimentação dos conversores D/A e 24 V para alimentar os *drivers* 3952 (V_{BB}). O circuito projetado está disponível na ilustração A-2.1, anexo A-2.

Outra característica de *hardware* a ser considerada é a do estágio de amplificação cujo funcionamento depende dos valores de componentes agregados ao *driver*. A corrente de carga foi limitada a 0.5A, pela utilização de um resistor R_s de 1 ohm. O circuito RC responsável em determinar o período em que os *drivers* ficam desligados (*blanking time*) foram ajustados para um tempo de 20 μ s, pelo uso de um resistor de 25k Ω e um capacitor de 820pF. O circuito de amplificação para acionar o motor de passo é composto por dois *drivers* 3952 e demais componentes passivos especificados, podendo ser visto na ilustração A-2.2, anexo A-2.

O filtro passa-baixa foi projetado para cortar frequências 10 vezes maiores que a frequência da onda portadora do PWM. Um exemplo de filtro utilizado foi o implementado através de dois resistores de 3k3 e 330 ohms e dois capacitores de 10nF.

Além disto, o protótipo conta com um microcontrolador PIC16C65, um oscilador de 20MHz e dois conversores D/A. A seleção entre o uso do sistema com PWM e com conversores D/A é feita por meio de *jumpers* que alteram o circuito para a forma desejada. Todo o funcionamento do sistema é sinalizado por meio de *leds* indicativos que permitem visualizar as entradas de liga/desliga, habilita, direção e *reset*; bem como acompanhar a frequência de entrada de pulsos e das trocas de fase.

Por questões de sigilo industrial, o circuito completo do sistema de acionamento para motores de passo não será apresentado neste trabalho. Uma foto do protótipo se encontra na ilustração 6-6 a fim de propiciar uma melhor visualização do *hardware* do sistema.

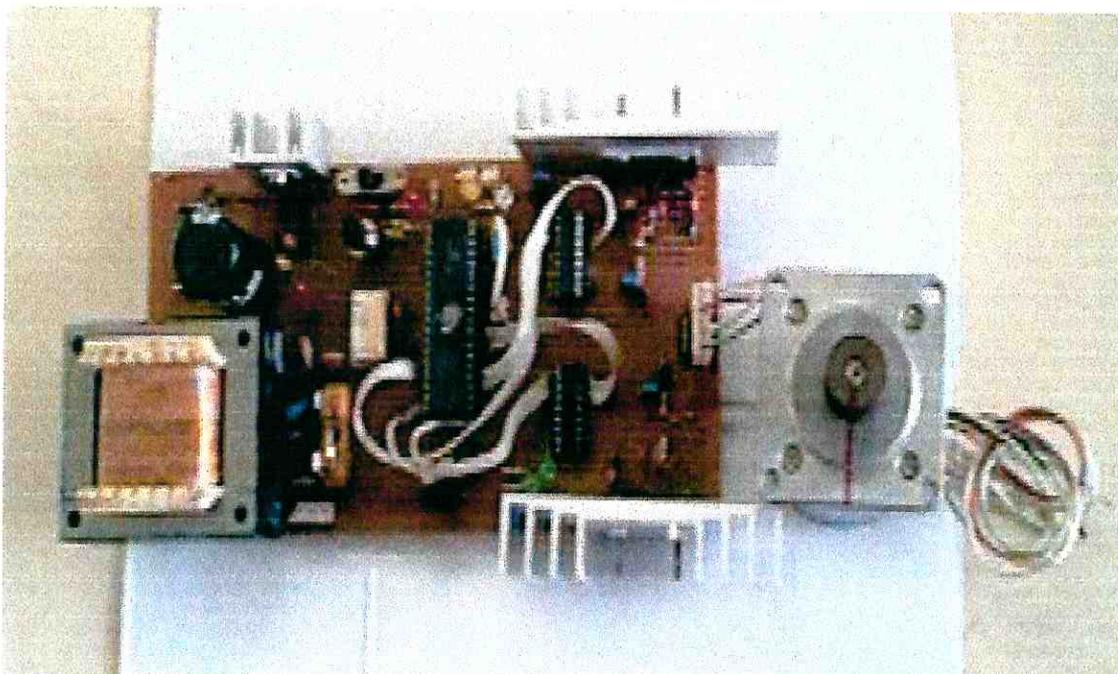


ilustração 6-6: foto do protótipo do *driver* de micropasso

6.6 Conclusão

O uso do microcontrolador PIC16C65 facilitou a realização do projeto devido às suas características de *hardware* em conter em uma única pastilha quase todos os componentes necessários para a construção do sistema, com a facilidade de interligação com dispositivos externos. A interface de programação facilitou muito o desenvolvimento do projeto, visto que a simulação do *software* e o monitoramento dos registros puderam ser realizados, auxiliando a execução e a correção de falhas de programação.

A implementação do protótipo possibilitou a realização de ensaios experimentais com diferentes tipos de programação, tanto para o PWM quanto para o conversor D/A. Estes testes estão apresentados no capítulo seguinte, bem como a análise dos resultados obtidos.

Capítulo VII



Resultados Obtidos

7.1 Introdução

Neste capítulo são apresentados os resultados experimentais obtidos a partir da implementação de um protótipo cujo projeto foi tratado nos capítulos anteriores. São apresentadas formas de onda que verificam o desempenho do circuito operando a diferentes faixas de velocidade do motor.

As formas de onda foram documentadas através de um osciloscópio digital de frequência de 200MHz, pois a maioria dos sinais do circuito de acionamento são de difícil sincronismo, raramente capturados em osciloscópios de menores frequências. Esta dificuldade está relacionada principalmente ao uso de PWM, que envolve frequências distintas na entrada, na modulação e na saída; além do próprio ruído causado pela operação do circuito a 20MHz.

7.2 Ensaio Realizados

Os ensaios foram realizados em duas frentes de trabalho: uma para testar a programação com PWM e outra com a conversão D/A. Para fins comparativos de resultados possíveis com a programação do PWM do microcontrolador PIC16C65, este modo foi preparado de três diferentes maneiras a fim de gerar frequências de onda portadora distintas. De acordo

com o que foi tratado no modo PWM e mostrado pela ilustração 5-6, procurou-se gerar ondas de 156.3 kHz, 19.53 kHz programadas com resolução padrão (8 bits) do PWM e 78.12 kHz com alta resolução (10 bits) do PWM.

Os testes de programação foram analisados quanto à veracidade e qualidade das respostas obtidas nos sinais de referência (corrente) e fase (tensão) a serem transmitidos aos *drivers*. Foram observadas as seqüências corretas de acionamento das tensões de fase, além da defasagem de 90° existente entre os sinais referentes às duas bobinas do motor. Pode-se comparar o nível de distorção do seno em função do tipo de programa utilizado (PWM ou D/A), e da frequência usada para gerar o PWM.

Sabe-se que a quantidade de micropassos que o motor pode realizar a cada passo está intimamente ligada ao tempo de execução da subrotina de atendimento de interrupção, pois este tempo limita a frequência com que cada micropasso pode ser realizado. Como o motor para dar uma volta completa realiza 200 passos, então para uma velocidade média de 1000 rpm, o tempo para realizar cada rotação é de 60ms, o que significa dizer um passo a cada 300µs. Devido a isto, para realizar 32 micropassos por passo implica em executar uma rotina de interrupção no máximo em 9µs. Com o microcontrolador operando a 20MHz (máxima frequência suportável), isto leva a concluir que a rotina pode conter no máximo 45 ciclos de instrução!

A par desta característica do processo, o programa, inicialmente modular conforme detalhado no fluxograma apresentado na ilustração 6-1, passou por um procedimento de compressão de código, a fim de se ganhar em tempo de execução, o que resulta em ganho de velocidade na saída do sistema. O código atual para a programação com PWM de 8 bits conta com apenas 29 ciclos, com tempo de atendimento de interrupção de 5.6µs na maior parte da execução do programa, a menos da necessidade de realizar a troca de fases a cada 32 micropassos, o que representa mais 13 ciclos de instrução, limitando a velocidade a pouco mais do que 1000 rpm. Já o código para PWM de alta resolução, por apresentar melhor qualidade da resposta na saída exige um tempo de execução maior, com 54 ciclos de instrução no pior caso (durante a troca de fases), atingindo velocidades de 900 rpm. O

programa com conversores D/A não representa melhoria em relação ao PWM, contendo 50 ciclos de instrução, podendo atingir os 1000 rpm desejados.

Isto significa que ao se desejar dispor de uma melhor resolução de passo, isto é, maior número de micropassos por passo, há a possibilidade de sua implementação através do microcontrolador PIC16C65, mas isto certamente representaria perda na velocidade final de rotação do motor. Por exemplo, para gerar 64 micropassos por passo, a velocidade do motor cairia a metade, 500 rpm para o microcontrolador operando a 20MHz.

Esta seleção entre resolução de passo/velocidade portanto depende somente da aplicação para qual será utilizado o sistema de acionamento e é facilmente implementável por *software*.

O funcionamento do circuito operando nos diferentes modos testados foi analisado para as mesmas frequências de entrada, sob as mesmas condições do sistema. Para enviar o sinal de entrada ao *driver*, fez-se uso de um gerador de funções. As ondas que melhor propiciaram a visualização do sinal em micropasso, da resolução do PWM e dos sinais de fase e corrente defasados de 90° foram coletadas através do osciloscópio digital, quando o sistema estava operando a 285Hz de entrada (frequência do micropasso), implicando num sinal senoidal de 4.43Hz (para os 64 valores de um seno), o que significa uma velocidade de rotação de aproximadamente 3 rpm.

As respostas obtidas para o sinal de micropasso na programação com PWM estão apresentadas nas ilustrações 7-1 a 7-4, para 156.3kHz e 7-5 a 7-8, para o 78.12kHz. O sinal para 19.53kHz apresentou-se bastante distorcido em vista da baixa frequência da moduladora, sendo descartado como solução do projeto. As formas de onda estão apresentadas na seguinte ordem: visualização do sinal PWM e do micropasso (sinal demodulado), ondas de seno (referência) para as duas fases do motor, seqüência de fase do motor (sentido horário), sobreposição do sinal de corrente (seno) e da tensão de fase.

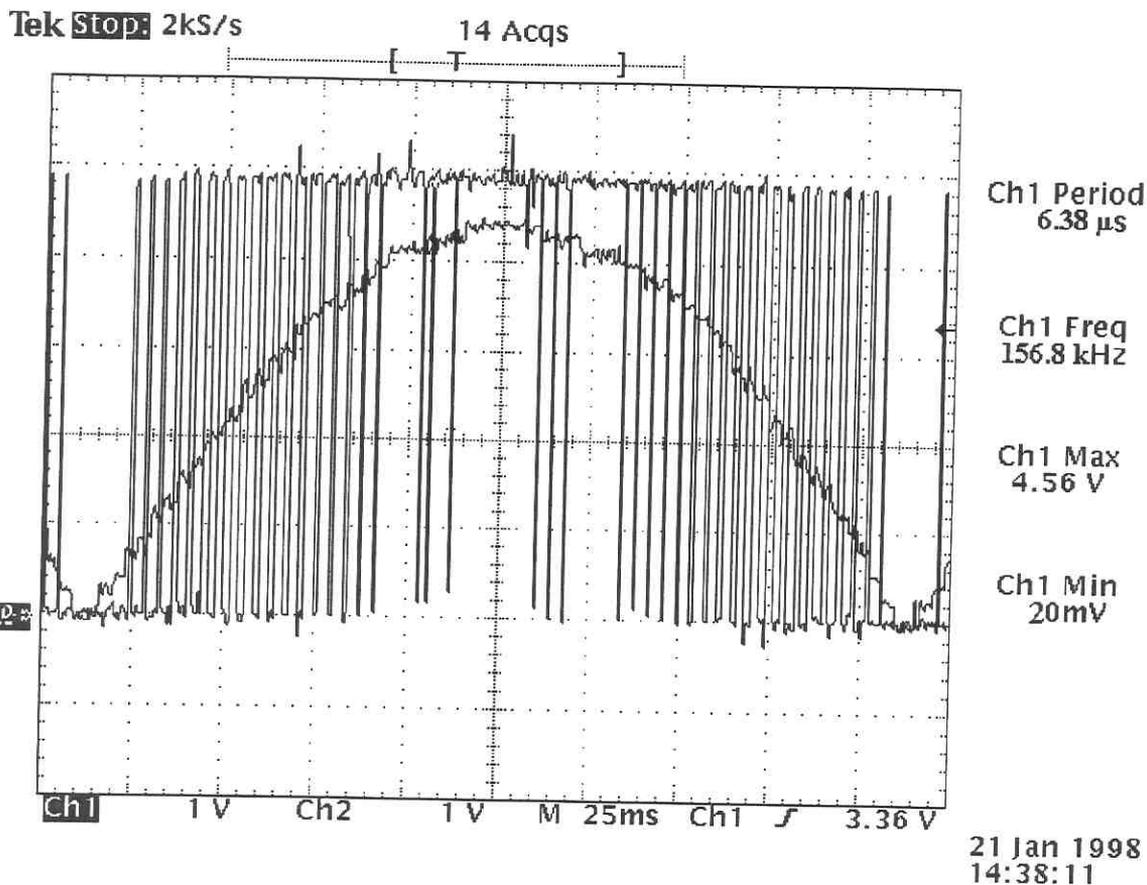


ilustração 7-1: Formas de onda do sinal PWM e micropasso para o *driver* com PWM a 156.3kHz

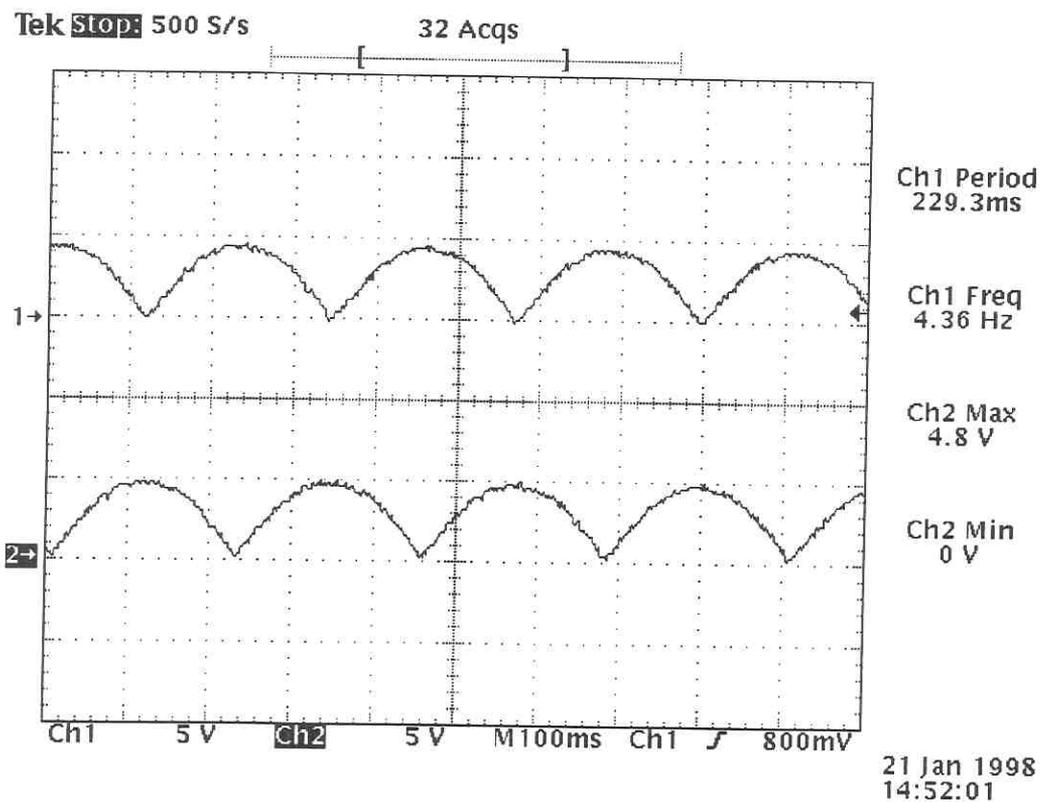


ilustração 7-2: Formas de onda do sinal de referência (seno) para o *driver* com PWM a 156.3kHz

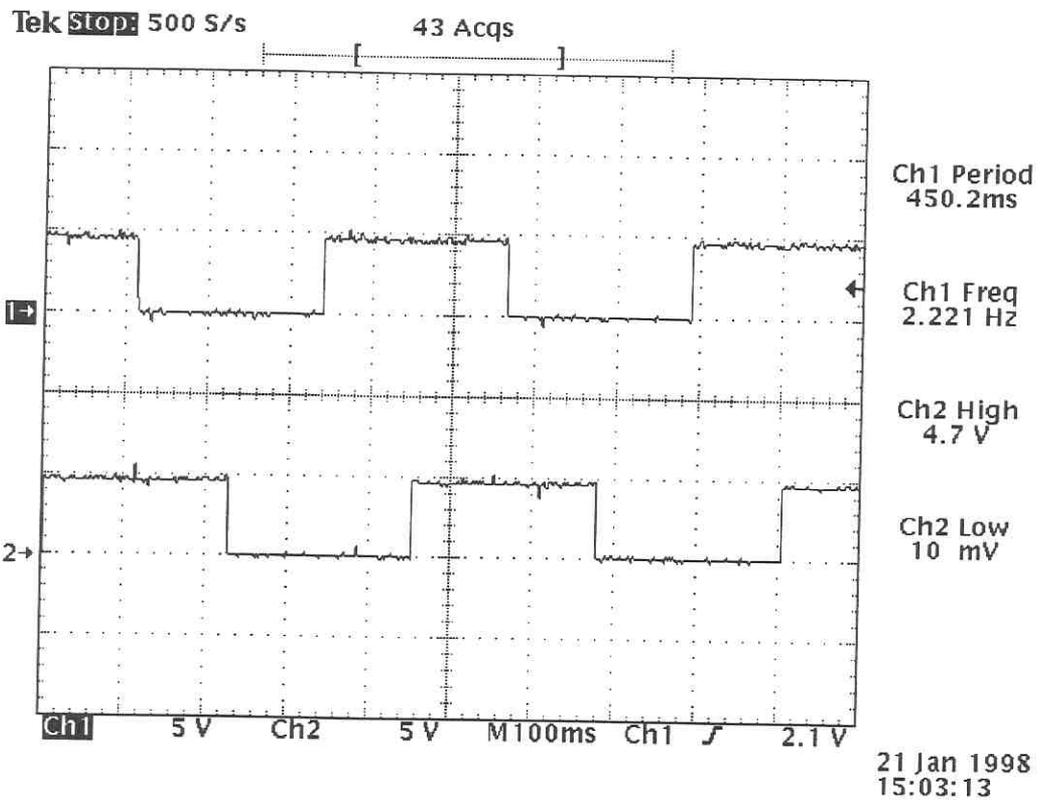


ilustração 7-3: Formas de onda da sequência de fases para o *driver* com PWM a 156.3kHz

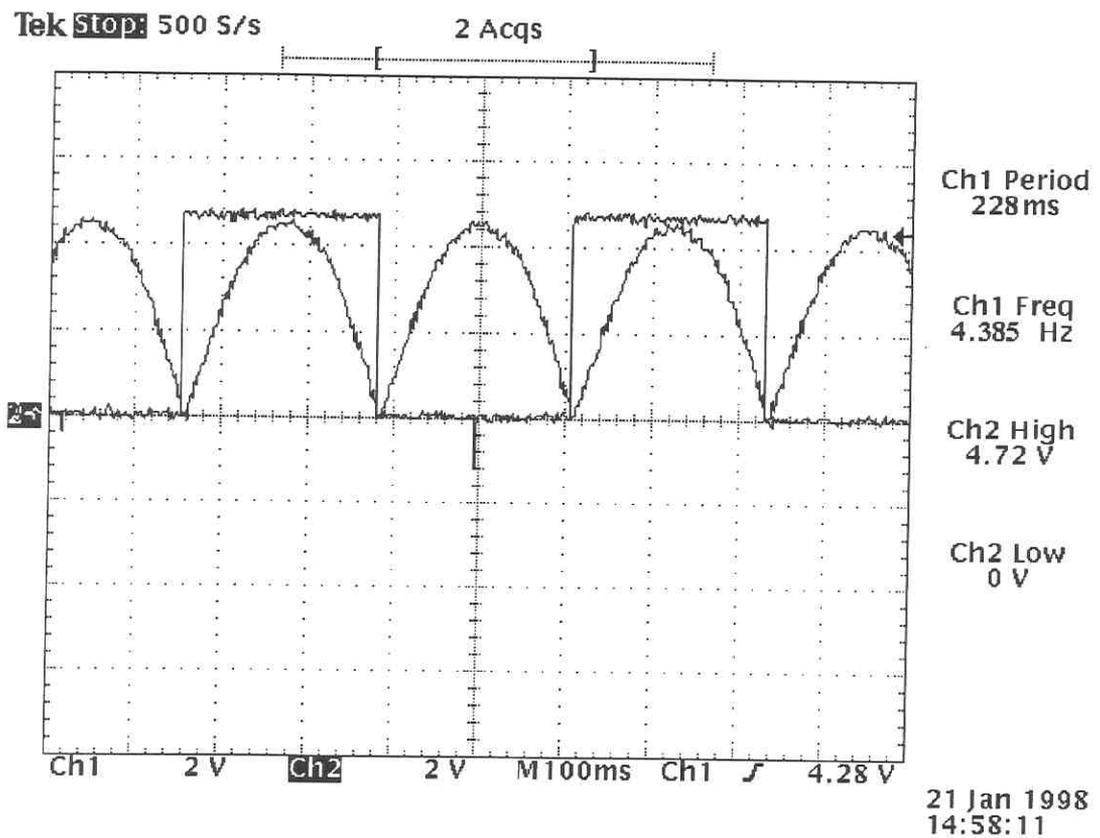


ilustração 7-4: Sobreposição do sinal de corrente e fase para o *driver* com PWM a 156.3kHz

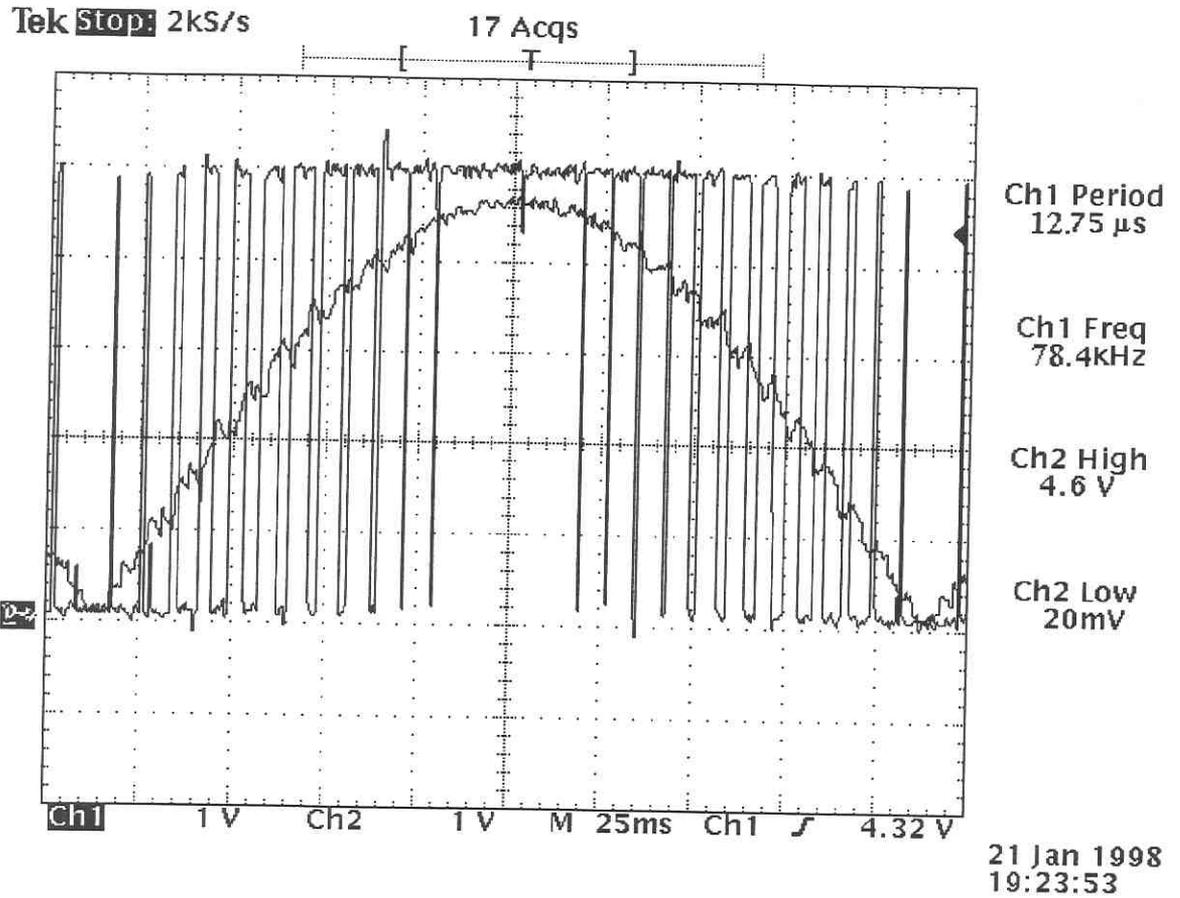


ilustração 7-5: Formas de onda do sinal PWM e micropasso para o *driver* com PWM a 78.12kHz

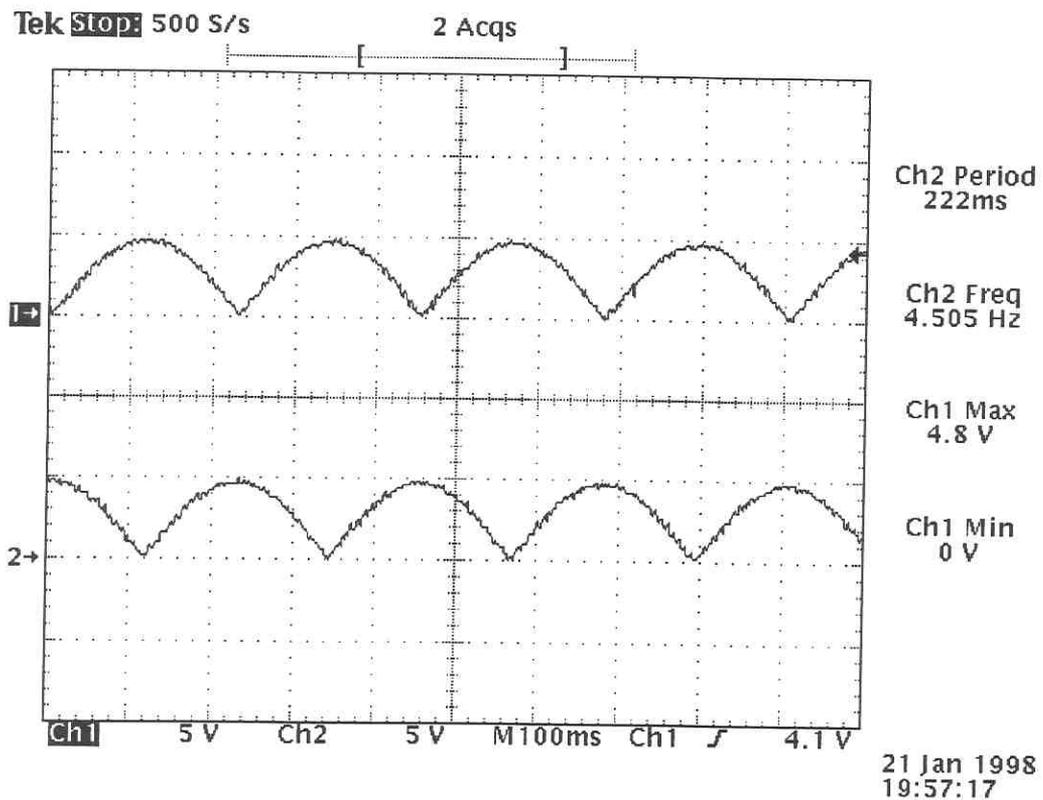


ilustração 7-6: Formas de onda do sinal de referência (seno) para o *driver* com PWM a 78.12kHz

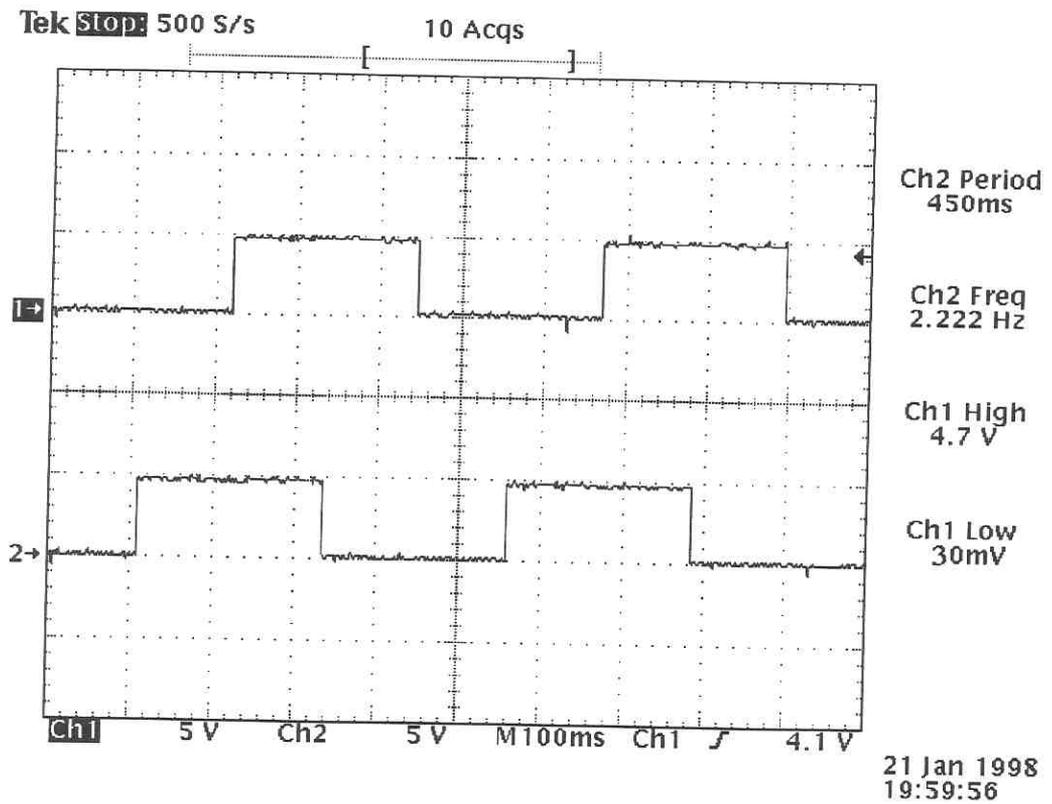


ilustração 7-7: Formas de onda da sequência de fases para o *driver* com PWM a 78.12kHz

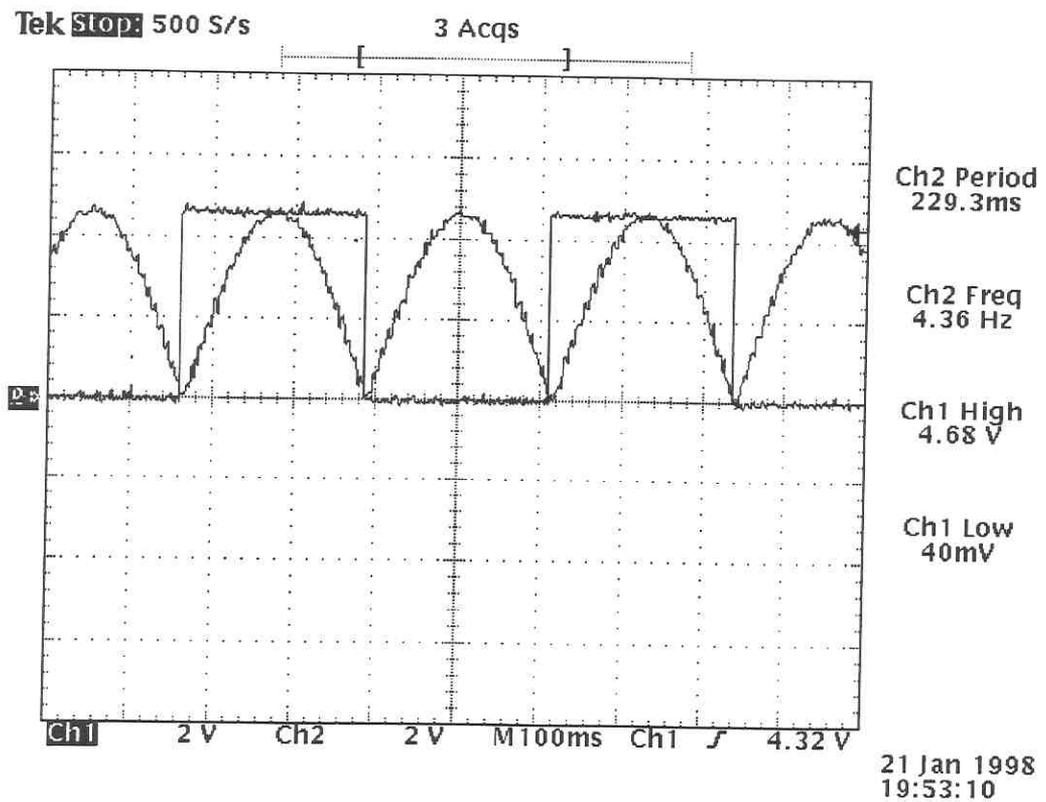


ilustração 7-8: Sobreposição do sinal de corrente e fase para o *driver* com PWM a 78.12kHz

Analisando as respostas obtidas, pode-se dizer que para o modo PWM utilizando apenas 5 bits de resolução para gerar a frequência de 156,3 kHz; detectou-se a “perda” de micropassos devido a alta velocidade de geração da portadora. Pode-se perceber a existência de algo em torno de 29 valores do seno ao invés dos 32 esperados. Além disto, a sensibilidade na troca de fase é maior que no caso de PWM’s operando a menores frequências.

Já a resposta com PWM de 8 bits (tabela de valores até FFh) na resolução padrão, apresentou-se melhor quanto ao número de micropassos, mas a resposta do seno ficou mais distorcida pela baixa frequência do sinal de PWM gerado (19.53kHz).

A solução intermediária encontrada em operar com valores de 8 bits (00h a FFh) numa frequência mais elevada, através da programação com alta resolução do PWM, mostrou-se mais confiável. Neste caso, consegue os 32 micropassos desejados e pode-se perceber que o sistema fica menos sensível a mudança de fases do sistema (a cada 32 micropassos, quando a rotina de atendimento de interrupção atinge o maior tempo de execução). O problema nesta implementação está no tempo de atendimento a interrupção, que é maior, implicando numa operação a velocidades um pouco reduzidas.

As respostas obtidas para os mesmos sinais na programação com conversor D/A estão apresentadas nas ilustrações 7-9 a 7-12; onde se pode observar a similaridade com a resposta obtida pelo PWM. Em termos de programação, o uso do conversor D/A, além de necessitar de dois componentes adicionais e uma série de conexões, mostrou-se pouco vantajoso em relação à programação do PWM; pois a necessidade de programar os bits de comando em seus corretos instantes representa significativa perda de tempo de processamento, tornando o tempo de execução praticamente o mesmo.

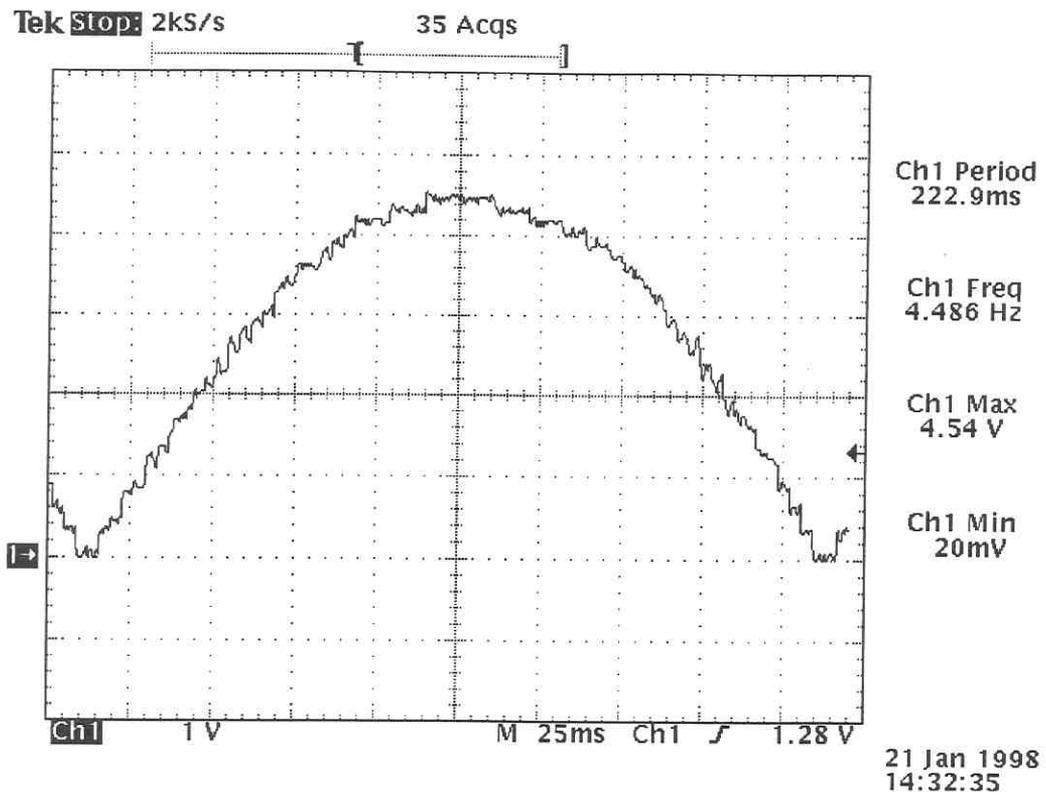


ilustração 7-9: Formas de onda de micropasso para o *driver* com conversores D/A

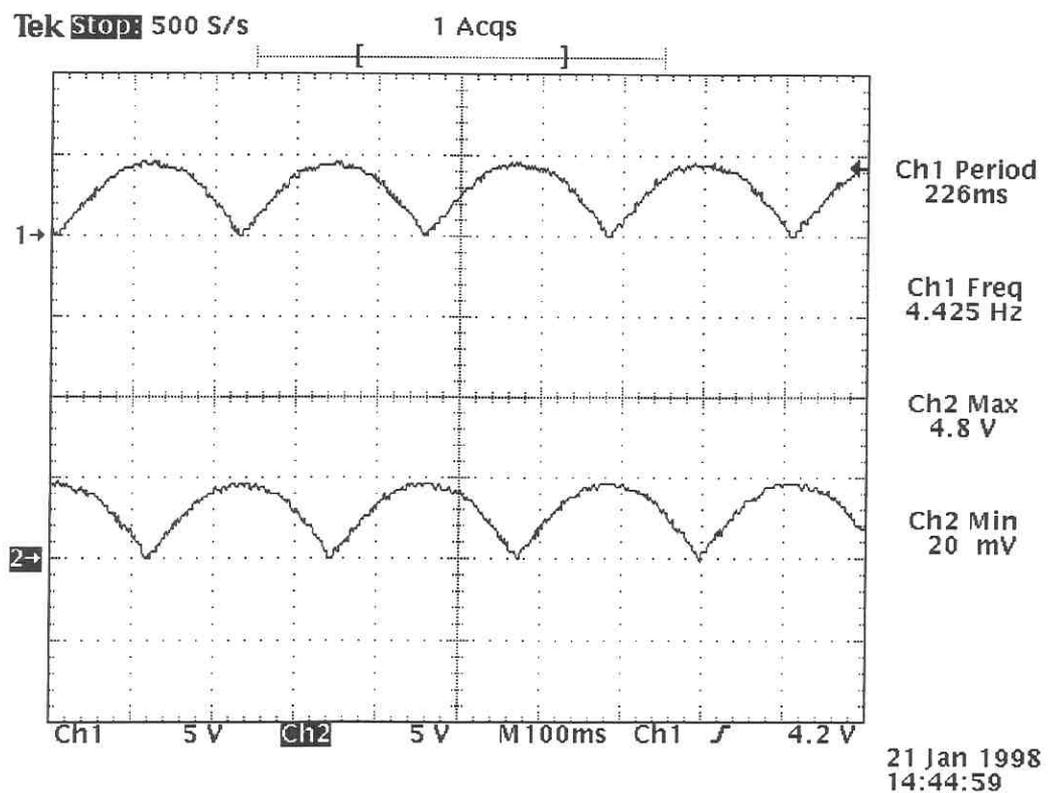


ilustração 7-10: Formas de onda do sinal de referência (seno) para o *driver* com conversores D/A

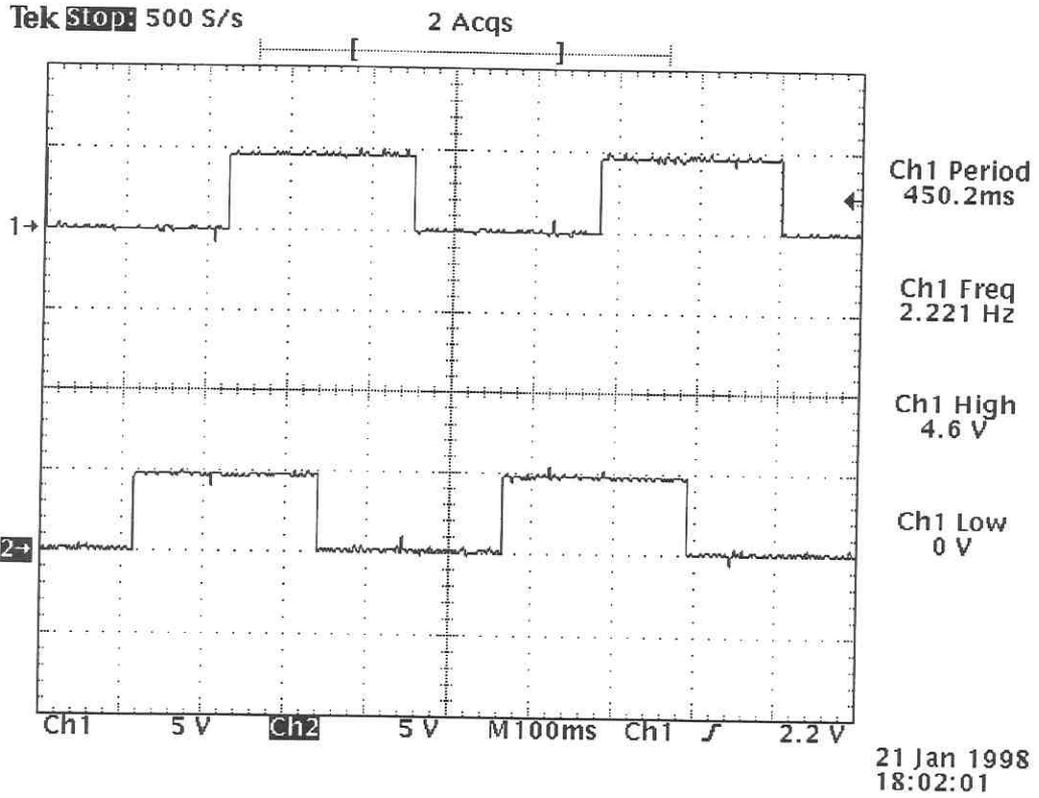


ilustração 7-11: Formas de onda da sequência de fases para o *driver* com conversores D/A

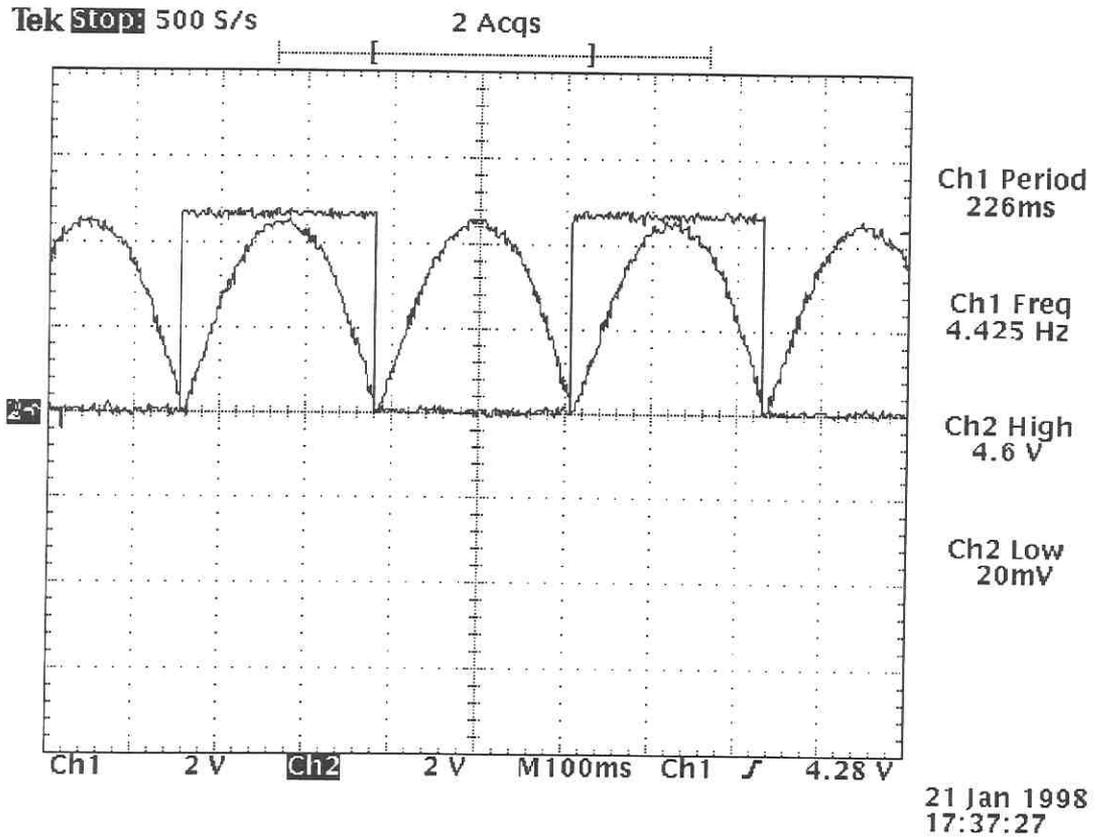


ilustração 7-12: Sobreposição do sinal de corrente e fase para o *driver* com conversores D/A

Para demonstrar o funcionamento do circuito de acionamento em toda a faixa de operação, foram realizados ensaios para os diversos programas (PWM de 156.3kHz, PWM de 78.12kHz e conversor D/A) com frequências de entrada de micropasso variando de 1 Hz a 100 kHz; permitindo ao motor girar de 0 a 1000 rpm; aproximadamente. Estes dados estão apresentados pelos sinais de referência senoidais a serem aplicados ao *driver*, dispostos de forma a facilitar a comparação dos resultados obtidos. Para todos os programas, os ensaios foram coletados na mesma frequência dentro de cada faixa de operação. As faixas foram divididas em múltiplos de 10: de 1 a 10 Hz, de 10 a 100Hz, e assim por diante; em vista da resolução do equipamento utilizado para gerar o sinal de entrada. Estes resultados podem ser analisados em anexo A-3.

7.3 Conclusão

Pelos resultados apresentados neste capítulo, pôde-se verificar o correto funcionamento do circuito de acionamento para motores de passo utilizando um microcontrolador como elemento responsável pelo comando dos sinais a serem entregues ao *driver*. Prova-se com isto a viabilidade do projeto, salvo as restrições de velocidade encontradas em vista da frequência máxima de operação do PIC16C65.

Comparando as respostas obtidas, pode-se concluir que a implementação do *driver* com PWM torna o projeto mais interessante, à medida que reduz o uso de componentes externos ao microcontrolador, o que implica em diminuição de custos e de tamanho do circuito; pois o uso de conversores D/A não representou melhorias significativas nos resultados obtidos com o protótipo.

Capítulo VIII



Conclusões e Perspectivas

Neste trabalho foi apresentado o estudo, desenvolvimento e implementação de um circuito de acionamento baseado em um microcontrolador. A idéia básica consistia em verificar a viabilidade do sistema, fundamentado na imposição de um sinal de tensão em meia onda num *driver* a fim de permitir gerar uma corrente de saída senoidal que garantisse um movimento de rotação mais suave ao motor de passo, sem a necessidade de controle realimentado no circuito.

O comando desenvolvido baseou-se no uso de um microcontrolador pela facilidade de implementação das leis de controle do servoacionamento e pela flexibilidade proporcionada por ele. O armazenamento dos ângulos de micropasso em memória de dados proporcionou um ganho significativo em termos de tempo de execução do programa. O resultado disto é um circuito bastante eficiente, em que os pulsos de entrada de passo são lidos com bastante rapidez, a menos da limitação do próprio componente.

Baseando-se nos resultados confirmados na prática através do protótipo implementado, pode-se afirmar que o projeto do circuito de acionamento microcontrolado para motores de passo é viável quando aplicado a cargas que não exijam velocidades muito elevadas (acima de 1000rpm), para a resolução desejada de 32 micropassos por passo.

As aplicações deste circuito se justificam em operações específicas de projeto, onde conhecendo-se a velocidade média de funcionamento de um sistema, permite-se a execução de um programa dedicado ao problema em questão. Pode-se inclusive aumentar/reduzir o número de micropassos requeridos, analisando o compromisso existente entre velocidade de rotação/resolução de passo do motor.

Como sugestão para trabalhos futuros, fica a proposta de desenvolvimento deste circuito de acionamento a partir de um componente que suporte operações a frequências mais elevadas; como os circuitos integrados da família ALTERA, que podem operar em até 50MHz. Com isto, pode-se reduzir o tempo de execução do programa, permitindo gerar maior número de micropassos por passo, além de se poder operar a maiores velocidades que as alcançadas com esta implementação com o microcontrolador da *Microchip*.

Sendo este um projeto de conclusão de curso, deve-se salientar que o projeto abrangeu uma grande quantidade de temas relacionados ao conteúdo ministrado durante o período acadêmico. Analisando este projeto como parte do aprendizado acadêmico, pode-se salientar que o contato com a empresa proporcionou uma boa experiência profissional, tanto pela convivência no ambiente de trabalho com profissionais mais experientes e de diferentes áreas quanto pela responsabilidade gerada pelo compromisso do desenvolvimento do projeto. Cabe ainda uma ressalva: o projeto concluído está dentro do que foi proposto inicialmente pelo anteprojeto apresentado em agosto de 1997.



Acionamento Microcontrolado para Motores de Passo

A-1 Conversão D/A

A-2 Circuitos do Projeto de *Hardware*

A-3 Resultados Complementares



Conversão D/A

A-1.1 Características Gerais

O conversor D/A utilizado no projeto realiza a conversão digital para analógica com resolução de saída em tensão de 8 bits. A interface lógica consiste de um registro de entrada de dados e um de conversão D/A (DAC). O que determina o valor analógico da saída do conversor D/A são os dados armazenados no registro de entrada. A lógica de controle do componente é feito através de linhas externas: CS (*chip select*), WR (*write*) e LDAC (*load digital/analogic converter*); sendo sua velocidade de atuação compatível com a maioria dos microcontroladores de 8 bits. Apresenta ainda uma entrada de RESET. É compatível com TTL e CMOS, podendo ser alimentado com tensão de 5 a 16.5V. A saída de tensão V_{out} do conversor D/A depende do valor de tensão aplicada ao pino de referência V_{ref} . Esta tensão pode variar de 2V a 12.5V [AD7224].

O DAC consiste de uma rede R-2R, ligada por 8 chaves bipolares de alta velocidade, cuja posição depende do valor armazenado no registro de entrada de dados: nível alto implica em conexão com a referência de tensão (V_{ref}) e nível baixo, com o sinal de terra (AGND).

O pino de saída V_{out} pode ser considerado como uma fonte de tensão programável digitalmente cujo valor é dado pela equação:

$$V_{out} = D.V_{ref}$$

onde D é a representação fracional do valor numérico do código de entrada de dados e pode variar de 0 a 255/256 (8 bits).

A-1.2 Informação da Interface Lógica

Os pinos CS e WR controlam o carregamento dos dados no registro de entrada do conversor D/A, enquanto o LDAC controla a transferência de informação do registro de entrada para o DAC. Os dados do DAC são convertidos no valor analógico de saída V_{out} .

O conteúdo dos registros pode ser apagado através de um sinal baixo aplicado ao pino de RESET. Um sinal de RESET coloca a saída do conversor D/A em 0V e pode ser útil para sincronização e calibração do sistema.

Para manter a precisão de funcionamento do registro DAC, a tensão de referência deve ser pelo menos 4V abaixo da tensão de alimentação do conversor D/A. Este diferencial de tensão é necessário para a geração correta da tensão média sobre as chaves da rede R-2R.

O diagrama de blocos do modo básico de operação do conversor D/A pode ser visto na figura A-1.

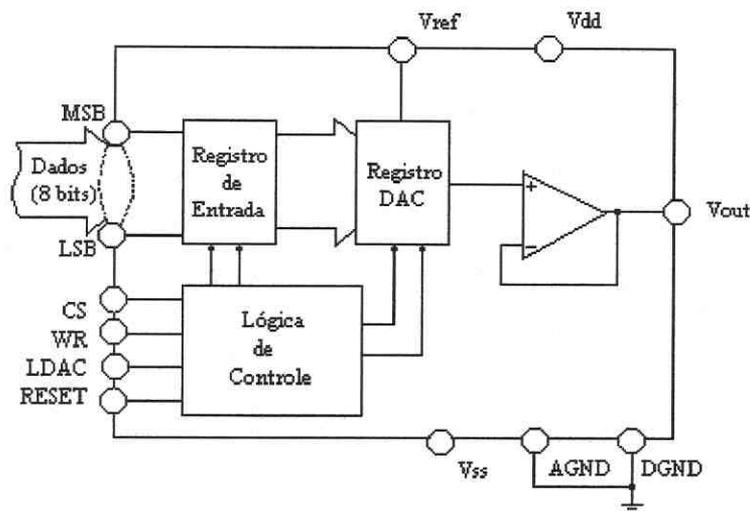


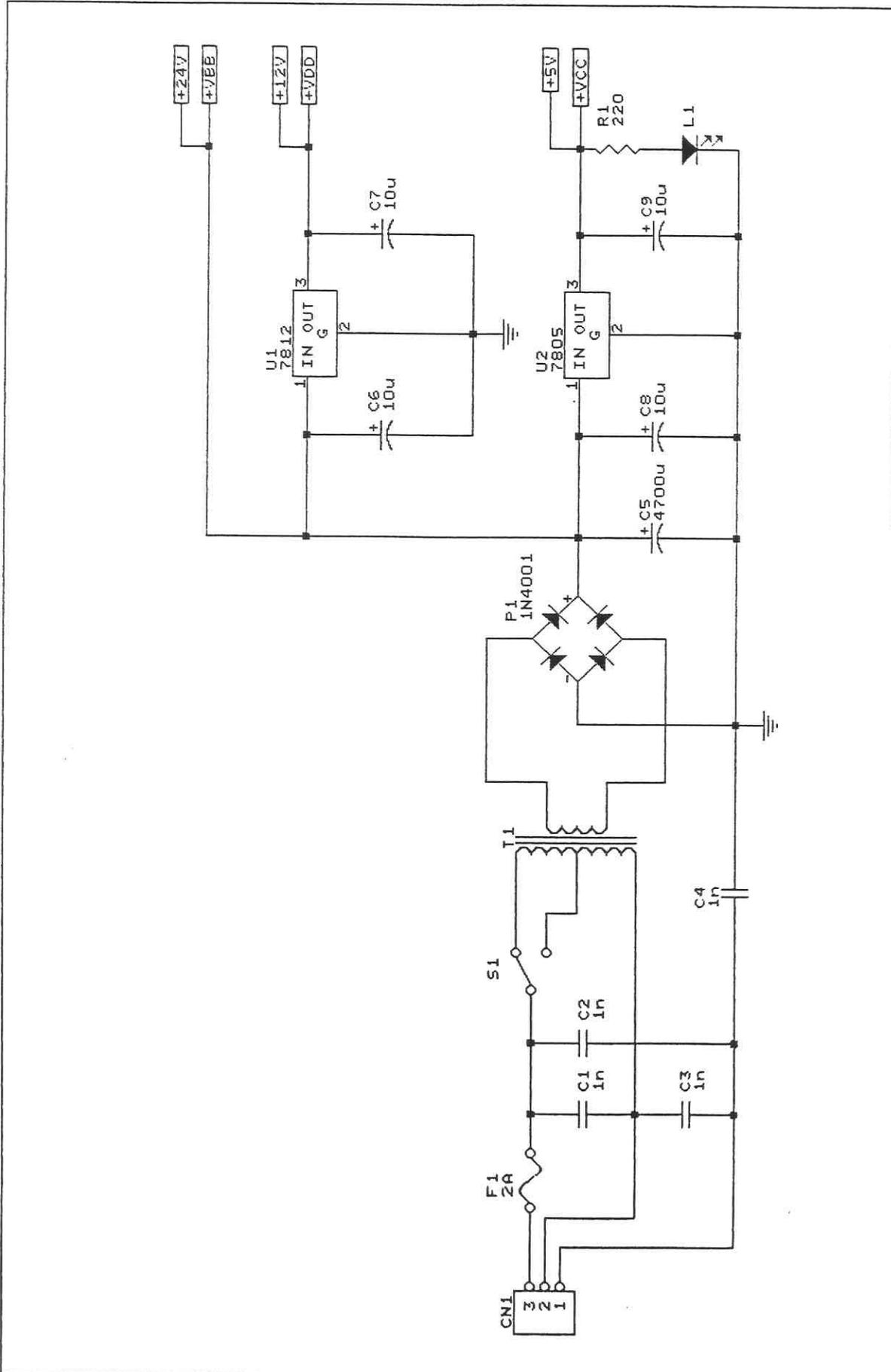
ilustração A-1: Diagrama de blocos do conversor D/A



Circuitos do Projeto de Hardware

A-2.1 Circuito da Fonte de Alimentação do *Driver* de Micropasso

A-2.2 Circuito de Amplificação do *Driver* de Micropasso



Universidade Federal de Santa Catarina	
Title	
Fonte de Alimentacao do Driver de Microprocesso	
Size	Document Number
A	Ilustracao A-2.1
REV	CAD
Date:	February 16, 1998
Sheet	1 of 1

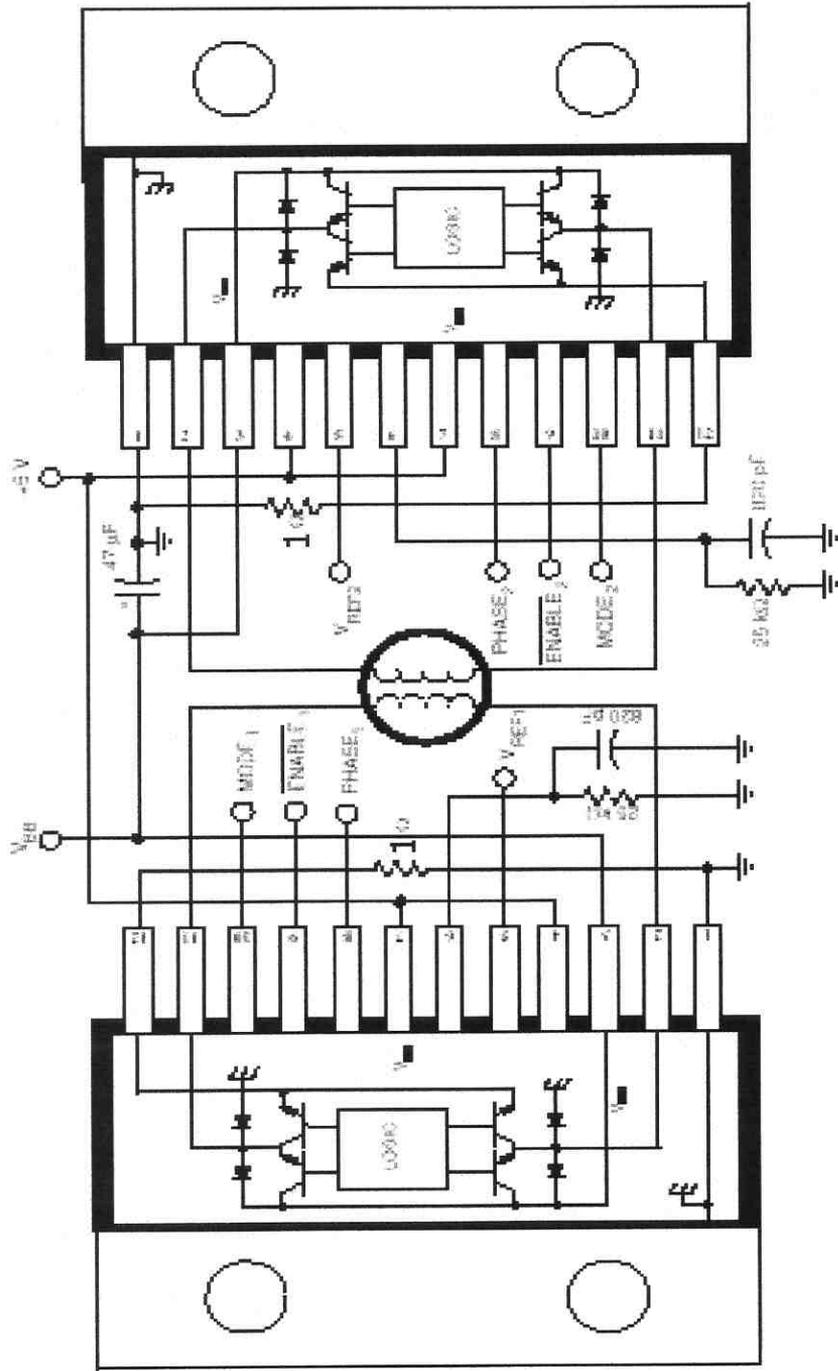


ilustração A-2.2: Circuito de amplificação do *driver* de micropasso

Resultados Complementares

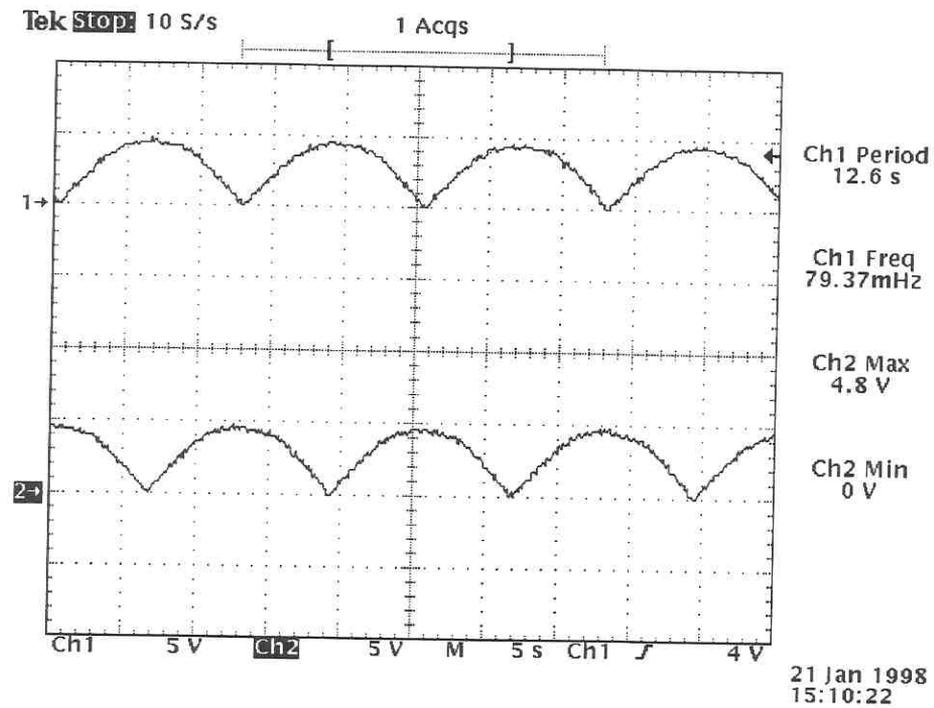
A-3.1 Introdução

Os resultados obtidos para a operação do *driver* de micropasso tanto para a programação com PWM como para com a conversão D/A estão apresentados neste anexo. As frequências de entrada de passo foram atribuídas ao sistema através de um gerador de funções, sendo documentado o funcionamento do circuito a uma frequência média dentro de cada faixa de operação estipulada. Foram divididas 5 faixas de velocidade de entrada: de 1 a 10 Hz, de 10 a 100Hz, de 100 a 1kHz, de 1k a 10kHz e de 10k a 100kHz.

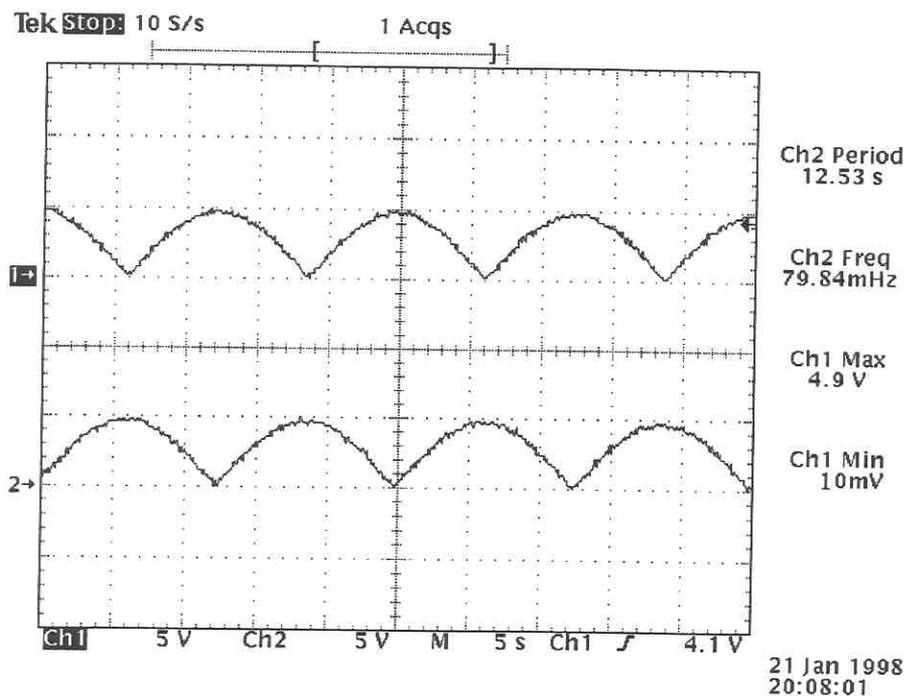
A fim de não sobrecarregar o conteúdo deste trabalho, só serão apresentados os resultados para a programação com PWM de 78.12kHz (figuras (a)) e com o conversor D/A (figuras (b)). A frequência de entrada do passo está descrita em cada ilustração. Para se obter a velocidade do motor, basta calcular o tempo que ele leva para realizar uma volta completa: girando 32 micropassos (na frequência da entrada) para cada um dos 200 passos do motor, isto resulta (em rpm) na fórmula:

$$f_{motor} = 60 \cdot \frac{f_{passo}}{(200 \cdot 32)}$$

A-3.2 Faixa de Operação de 1 a 10Hz



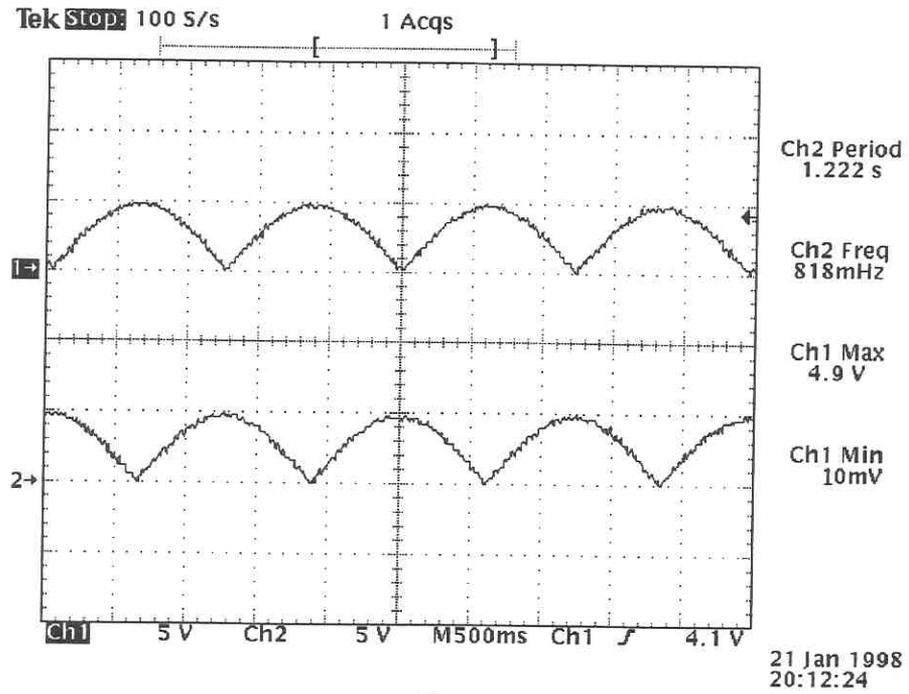
(a)



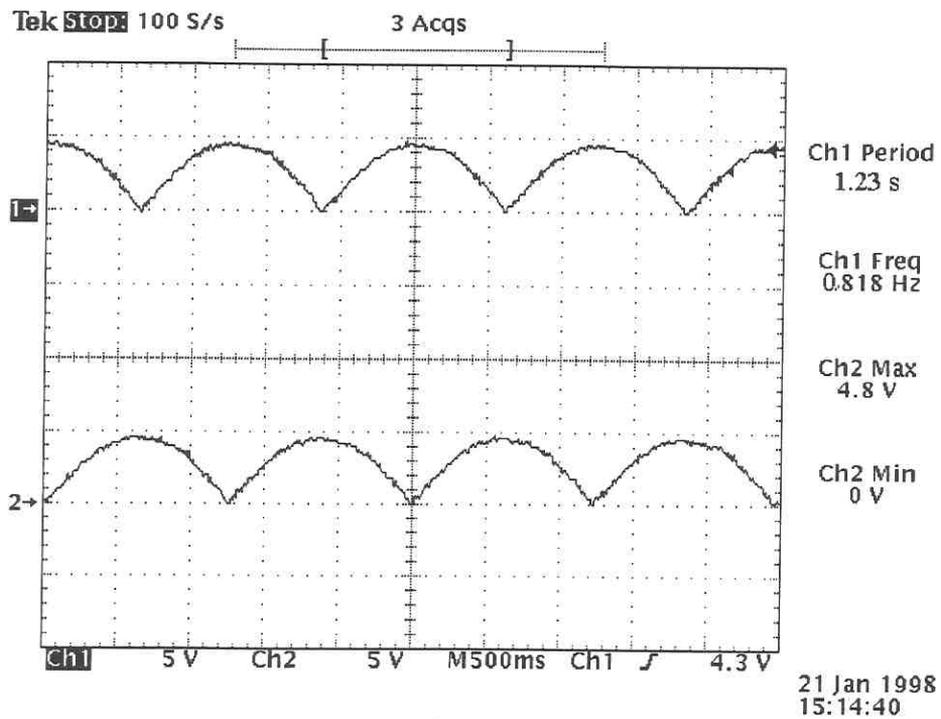
(b)

ilustração A-3.1 : Entrada de passo a 5.3 Hz – velocidade do motor: 0.05 rpm

A-3.3 Faixa de Operação de 10 a 100Hz



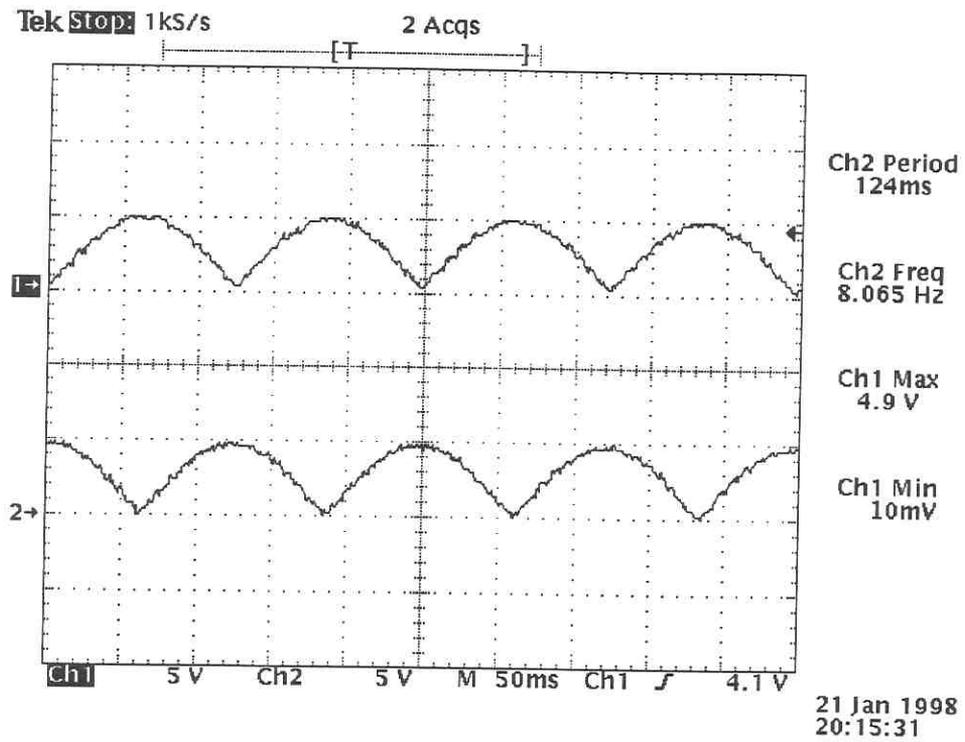
(a)



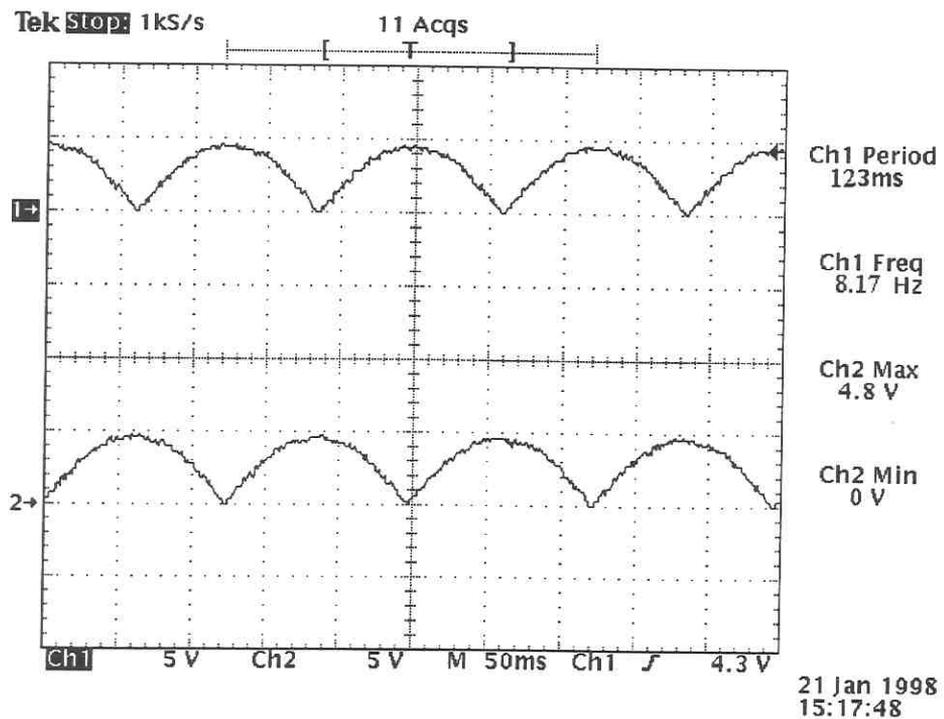
(b)

ilustração A-3.2 : Entrada de passo a 53 Hz – velocidade do motor: 0.5 rpm

A-3.4 Faixa de Operação de 100 a 1kHz



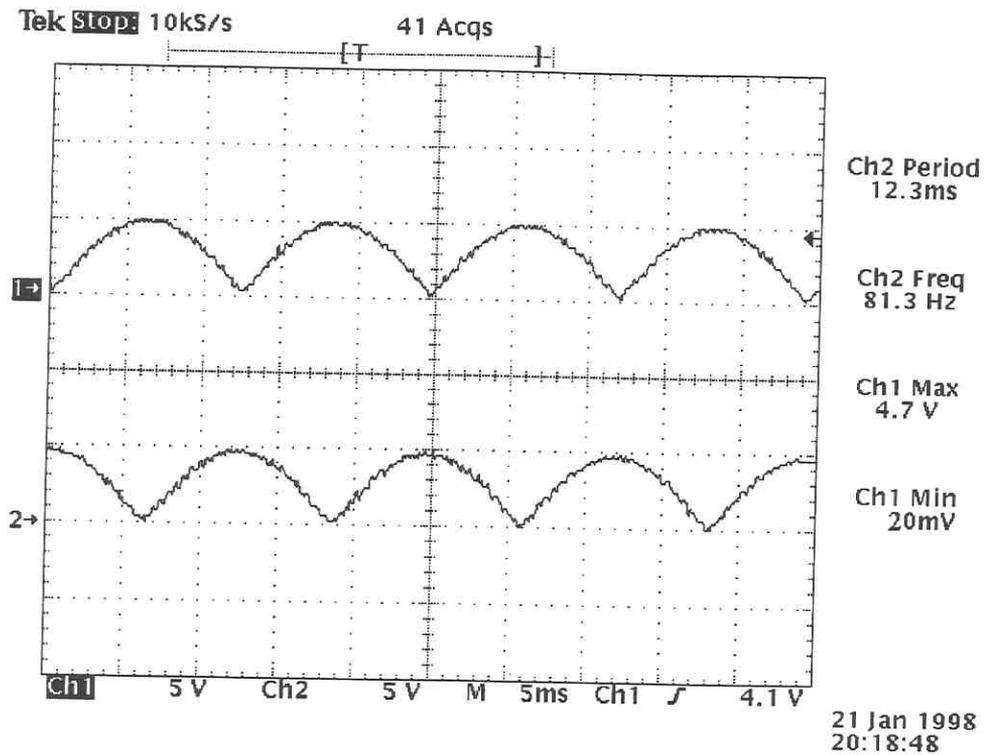
(a)



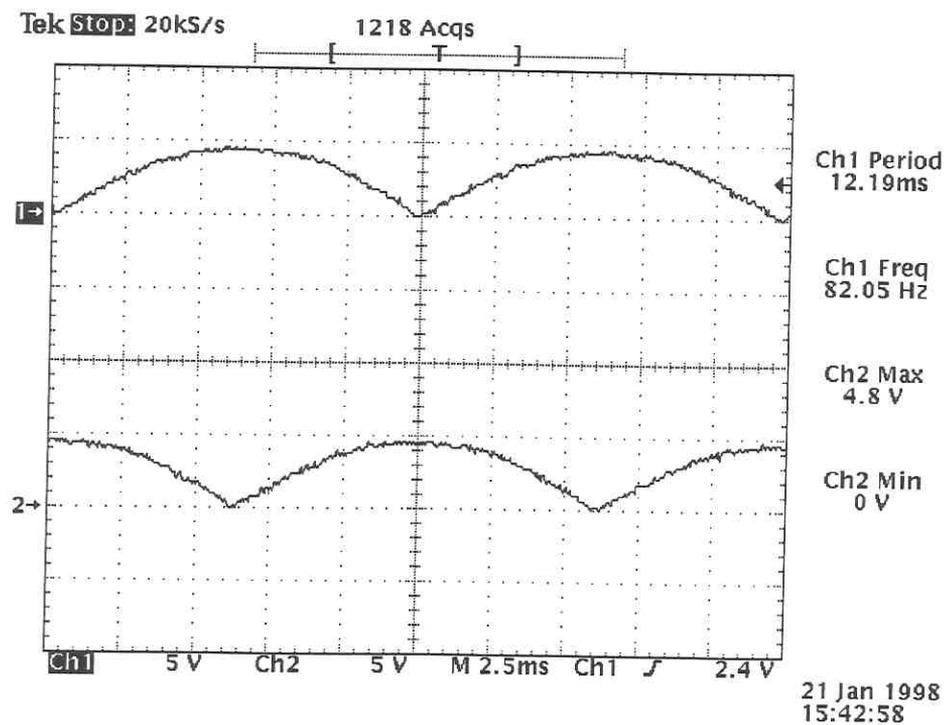
(b)

ilustração A-3.3 : Entrada de passo a 530 Hz – velocidade do motor: 5 rpm

A-3.5 Faixa de Operação de 1k a 10kHz



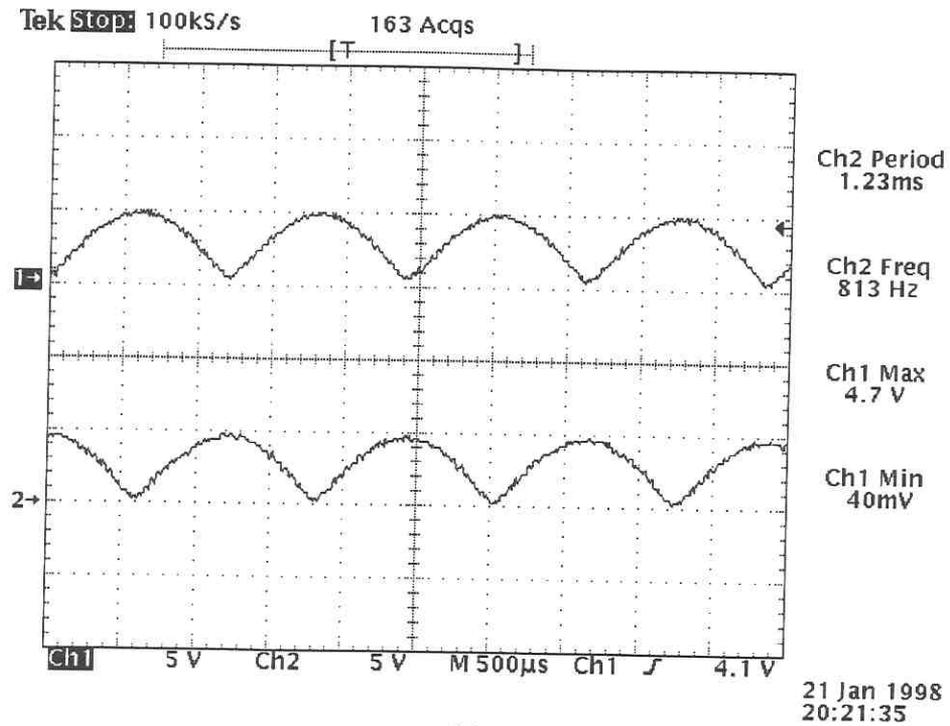
(a)



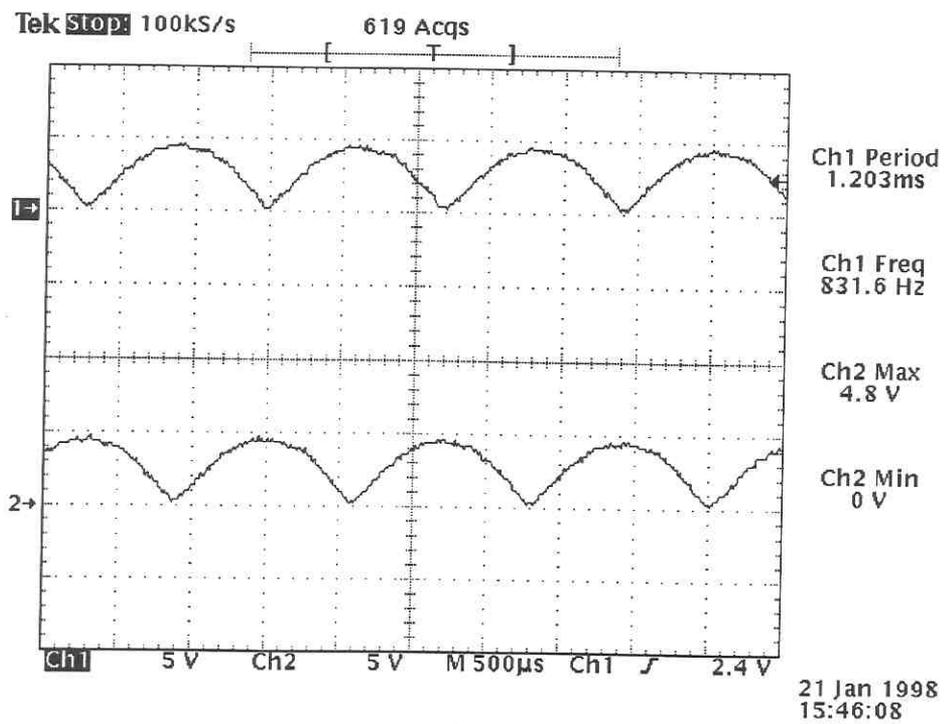
(b)

ilustração A-3.4 : Entrada de passo a 5.3 kHz – velocidade do motor: 50 rpm

A-3.6 Faixa de Operação de 10k a 100kHz



(a)



(b)

ilustração A-3.5 : Entrada de passo a 53 kHz – velocidade do motor: 500 rpm

Referências Bibliográficas

[Acarnley, 1992]

ACARNLEY, P. P.. **Stepping Motors: a guide to modern theory and practice**. IEE Control Engineering. 1992.

[AD7224]

ANALOG DEVICES. Data Book: **AD7224, Digital to Analog Converts**.

[Allegro, 1993]

ALLEGRO Microsystems, Inc. **Integrated and Discrete Semiconductores**. Data Book - A3952S. 1993.

[Barbi et al., 1995]

BARBI, Ivo; MARTINS; Denizar C.. **Princípios de Acionamento em Corrente Alternada**. LAMEP. UFSC. 1995.

[Bento, 1989]

BENTO, Celso R.. **Sistemas de Controle - Teoria e Projetos**. Editora Érica. São Paulo, 1989.

[Carlson et al., 1988]

CARLSON, Gordon; COLLINS, Michael. Precision Stepping With Software. **Machine Design**, p. 79-83, jul. 1988.

[Chrysafis, 1988]

CHRYSAFIS, Andreas. **Digital Sine-Waves Synthesis Using the DSP56001**. Strategic Applications with Digital Signal Processing Operation. Motorola Inc. 1988.

[Compumotor, 1993]

COMPUMOTOR Parker Motion Control. **Step Motor & Servo Motor Systems and Control**. Catálogo OEM350/OEM650 Drive. 1993.

[Kenjo, 1993]

KENJO, Takashi. **Power Electronic for the Microprocessor Age**. Oxford University Press. Great Britain, 1990.

[Lichtblau, 1989]

LICHTBLAU, Marcos. **Uma contribuição ao estudo da dinâmica de posicionadores**. Dissertação de mestrado. UFSC, 1989.

[Microchip, 1994]

MICROCHIP Technology Inc.. **User's Guide**. Handbook : Using the PIC16CXX. 1994.

[Smith, 1987]

SMITH, S. G. **Motion Control Application Manual**. 1987.

[Stute et al., 1983]

STUTE, Gottfried; BÖBEL, Karl-Heinz; HESSELBACH, Jürgen; HODEL, Ulf; STOF, Peter. **Electrical Feed Drives for Machine Tools**. Siemens - Aktiengesellschaft. Chichester: Wiley, 1983.

[Vexta, 1987]

TECHNICAL INFORMATION ON STEPPING MOTORS. **Catálogo Oficial. VEXTA™**
Oriental Motor Corporation. Califórnia, 1987.

