

UNIVERSIDADE SÃO FRANCISCO
CURSO DE ENGENHARIA ELÉTRICA

INVERSOR PWM

Área de Eletrônica Digital e Eletrônica de Potência

por

André Cristiano Aparecido

Prof. M.Sc. Marcus Vinicius Ataíde
Orientador

Itatiba (SP), Dezembro de 2009

UNIVERSIDADE SÃO FRANCISCO
CURSO DE ENGENHARIA ELÉTRICA

INVERSOR PWM

Área de Eletrônica Digital e Eletrônica de Potência

por

André Cristiano Aparecido

Monografia apresentada à Banca Examinadora
do Trabalho de Conclusão do Curso de
Engenharia Elétrica para análise e aprovação.
Orientador: Prof. M. Sc. Marcus Vinicius Ataíde

Itatiba (SP), Dezembro de 2009

AGRADECIMENTOS

Agradeço primeiramente a Deus por me dar saúde, força para terminar este trabalho, aos meus pais Cleide e Donizetti por me apoiarem desde o começo, a minha namorada Marília pela paciência e compreensão, aos meus colegas de faculdade Davilson, Júlio, André Gasparoti e Cícero pelas ajudas prestadas, ao meu professor e orientador Marcus Vinicius por me guiar neste trabalho e pelos ensinamentos transmitidos e aos meus familiares e amigos pelo incentivo.

*“Não te deixeis vencer pelo mal,
mas vence o mal pelo bem”. (Rm 12,21)*

SUMÁRIO

LISTA DE ABREVIATURAS.....	vi
LISTA DE FIGURAS.....	vii
RESUMO.....	viii
ABSTRACT.....	ix
1. INTRODUÇÃO.....	10
1.1. OBJETIVOS.....	10
1.1.1. Objetivo Geral.....	10
1.1.2. Objetivos Específicos.....	10
1.2. METODOLOGIA.....	11
2. FUNDAMENTAÇÃO TEÓRICA.....	11
2.1. MOTORES DE INDUÇÃO TRIFÁSICOS.....	11
2.1.1. Identificação das características técnicas de um motor.....	12
2.1.2. Rotação de um Motor.....	13
2.1.3. Motores de indução tipo gaiola de esquilo.....	14
2.1.4. Motores de anéis.....	15
2.1.5. Característica do torque em motores de indução.....	15
2.2. TEORIA DE ACIONAMENTO MODO ESCALAR V/F.....	16
3. PROJETO.....	18
3.1.1. Princípio de Funcionamento da Placa de Potência.....	18
3.1.2. Princípios de Funcionamento da Placa de Controle.....	22
3.1.3. O microcontrolador 18F2431.....	23
3.1.4. Princípios de funcionamento do Programa.....	24
3.1.5. Fluxograma do Programa.....	26
3.1.6. Detalhamento do Programa.....	27
3.1.7. Inicialização.....	27
3.1.8. Loop Principal.....	27
3.1.9. Interrupt Service Routine (ISR) (Rotina de serviço de interrupção).....	28
3.1.10. Inicialização dos registradores responsáveis pelo modo PWM.....	28
3.1.11. Inicialização dos registradores responsáveis pelo ADC.....	30
3.1.12. Controle V/F pelo software.....	31
3.1.13. Cálculo da frequência em malha aberta.....	32
3.1.14. Resultados Alcançados.....	33
4. CONCLUSÃO.....	38

5. SUGESTÃO PARA TRABALHOS FUTUROS	39
REFERÊNCIAS BIBLIOGRÁFICAS	40
APÊNDICE A – Software microcontrolador.....	41
APÊNDICE B –Circuito impresso placa Controle	57
APÊNDICE C –Circuito impresso placa Potência.....	57
ANEXO I – Estrutura interna do módulo de potência.....	58

LISTA DE ABREVIATURAS

PWM	(Pulse Width Modulation) – Modulação por largura de pulso
RMS	(Root Mean Square) Onda Média Quadrática
IGBT	Transistor Bipolar de Porta Isolada
CC	Corrente Contínua
CA	Corrente Alternada
CV	Cavalos Vapor (Unidade de Potência)
RPM	Rotação por minutos
Hz	Hertz
V/F	Tensão (V) dividido por Frequência (F) – Modo Escalar
PCPWM	Power Control PWM (Comando de Controle PWM)
ADC	Conversor Analógico Digital
FOSC	Frequência do Oscilador do Microcontrolador

LISTA DE FIGURAS

Figura 1. Típico rotor de gaiola.....	14
Figura 2. Velocidade X Torque em motores de indução.....	15
Figura 3. Relação Torque e Tensão X Frequência.....	17
Figura 4. Simbologia do módulo IGBT e o estator do motor de indução.....	19
Figura 5. Circuito da montagem da placa de Potência.....	19
Figura 6. Módulo de Potência IRAM 16UP60A.....	21
Figura 7. Protótipo da placa de Potência.....	21
Figura 8. Esquema de ligação placa de controle.....	22
Figura 9. Protótipo da Placa de controle.....	23
Figura 10. Microcontrolador PIC 18F2431.....	24
Figura 11. Diagrama em Blocos do controle V/F.....	25
Figura 12. Síntese das 3 formas de onda.....	25
Figura 13. Detalhe do programa que inicializa o módulo PWM.....	29
Figura 14. Inicialização dos registradores do ADC.....	31
Figura 15. Parte do programa que mostra o acesso a tabela por endereçamento indireto.....	32
Figura 16. Leitura do canal analógico.....	33
Figura 17. Formas de ondas obtidas por simulação.....	34
Figura 18. Formas de ondas defasadas em 120°.....	34
Figura 19. Tempo morto de segurança para o chaveamento entre as fases.....	35
Figura 20. Foto mostrando a forma de onda PWM retiradas do protótipo.....	36
Figura 21. Tempo morto entre os sinais PWM medido no protótipo.....	37

RESUMO

Aparecido, André Cristiano. **Inversor PWM**. Itatiba, 2009. 58f. Trabalho de Conclusão de Curso, Universidade São Francisco, Itatiba, 2009.

As máquinas de corrente alternada tem seu sistema construtivo muito mais simples e robusto do que uma máquina de corrente contínua, principalmente a máquina C.A. de indução. Para uma mesma potência, comparando-se o motor de corrente continua com o motor de indução, esse tem menor massa e, conseqüentemente, um custo menor na sua aquisição e na sua manutenção.

Para se fazer o acionamento de uma máquina C.A. de indução com velocidade variável, são necessários sistemas de controle mais sofisticados do que aqueles utilizados para o acionamento e variação de velocidade de máquinas C.C.. No entanto, com a evolução da microeletrônica, notadamente a área de microcontroladores e dos transistores de potência, aliado ao surgimento de novas técnicas de controle levou ao surgimento de conversores C.C. - C.A. de alta eficiência e de baixo custo, que são atualmente utilizados na indústria para o acionamento em velocidade variável de máquinas C.A. de indução.

O objetivo do projeto intitulado “INVERSOR PWM” é o de desenvolver um programa com o auxílio do microcontrolador capaz de gerar seis sinais PWM para acionar um módulo de potência. Com isso possibilita-se controlar a velocidade de motores de indução C.A. por intermédio da técnica de controle v/f , mais conhecida como controle escalar. Esta técnica se baseia na variação da tensão R.M.S. e da frequência aplicada a um motor C.A. de indução, o que ocasiona a variação da velocidade desse motor.

Palavras-chave: máquina C.A. de indução, microcontrolador, controle escalar.

ABSTRACT

Machinery AC has its constructive system much simpler and more robust than a dc machine, especially the AC induction machine. For a given power, comparing the current engine continues with the induction motor that has less mass and therefore a lower cost in its acquisition and maintenance.

To make the drive an AC induction machine with variable speed systems are needed to control more sophisticated than those used for the drive and speed variation of dc machines. However, with the evolution of microelectronics, especially the area of microcontrollers and power transistors, coupled with the emergence of new control techniques led to the emergence of DC converters - AC high efficiency and low cost that are currently used in the for the drive in variable speed AC induction machines.

The objective of the project entitled "PWM INVERTER" is to develop a program with the help of the microcontroller can generate six PWM signals to drive a power module. Thus enabling control the speed of AC induction motors through the technique of control v / f , best known as control scale. This technique is based on the variation of R.M.S. voltage and frequency applied to an AC induction motor, which causes the change of speed of a motor.

Keywords: AC induction machine, microcontroller, control scale.

1. INTRODUÇÃO

O desenvolvimento da eletrônica de potência vem com o passar do tempo, tornando mais fácil (e mais barato) o acionamento de motores. Com isto, sistemas que antes usavam motores C.C., pela facilidade de controle, hoje podem usar motores CA de indução graças ao desenvolvimento de Inversores de Frequência, também chamados de Conversores de Frequência (FRANCHI, 2008).

Em paralelo com o avanço da eletrônica de potência, a área de processadores digitais, mais conhecidos como microprocessadores e microcontroladores, também tem evoluído bastante, o que permitiu seu emprego nos Inversores de Frequência, com funções cada vez mais complexas (FRANCHI, 2008).

Os Inversores de Frequência podem substituir, com vantagens, os sistemas de controle de fluxo com válvulas (bombas) ou dampers (ventiladores) e acionamento em várias aplicações de motores, que exijam um controle preciso de velocidade, torque e consumo de energia (FRANCHI, 2008). Ao se acionar um motor de corrente alternada, por um inversor de frequência, obtém-se a mesma alta qualidade de controle obtida por um motor de corrente contínua, com a vantagem de que o motor de indução C.A. não precisa de tanta manutenção quanto o motor de corrente contínua.

1.1. OBJETIVOS

1.1.1. Objetivo Geral

Este trabalho tem por objetivo descrever um sistema de controle e comando digitais por intermédio de microcontrolador, para acionamento de motor de indução trifásico. O motor é acionado por um módulo integrado de potência, que recebe comandos vindos do microcontrolador empregando a técnica PWM.

1.1.2. Objetivos Específicos

O objetivo específico deste trabalho é desenvolver um programa para microcontrolador que seja capaz de gerar seis sinais PWM que possam acionar e controlar a velocidade de pequenos motores de indução C.A., conhecido comercialmente como inversor de frequência.

- Desenvolver o programa em linguagem assembly para o microcontrolador capaz de fazer o acionamento e controle de um motor de indução trifásico.

- Criação de um protótipo de uma placa de controle responsável por gerar os sinais PWM para a placa de potência e poder realizar testes no programa.
- Criação de um protótipo de uma placa de potência responsável por alimentar todo o inversor.

1.2. METODOLOGIA

A metodologia a ser empregada será a de estudos e pesquisas em livros e artigos técnicos, pesquisa em acervos técnicos por intermédio da rede mundial de computadores (Internet), montagem e testes de hardware e software necessários ao desenvolvimento do projeto.

O trabalho está estruturado seguindo os seguintes passos:

1. Desenvolvimento da parte teórica com estudos sobre máquinas C.A., técnica de modulação digital PWM e de técnicas de acionamentos.
2. Inicialização do projeto, princípios de funcionamento da placa de potência.
3. Princípios de funcionamento da placa de controle, e seus componentes.
4. Descrição do microcontrolador utilizado
5. Inicialização do programa, detalhes e funcionamentos
6. Resultados e conclusão.

2. FUNDAMENTAÇÃO TEÓRICA

Alguns conceitos sobre motores de indução, como características técnicas, tipos, características do torque e também conceitos sobre o controle V/F (modo escalar) devem ser abordados antes da elaboração do projeto. Assim sendo é apresentado a seguir um breve estudo sobre estes assuntos relacionados.

2.1. MOTORES DE INDUÇÃO TRIFÁSICOS

Os motores de indução trifásicos são os motores mais utilizados para as aplicações de controle industrial, e automação. Eles são robustos, confiáveis e duráveis. Quando a energia é

fornecida para um motor de indução nas especificações nominais, ele gira em sua velocidade nominal. No entanto, muitas aplicações necessitam de operações com velocidade variável (YEDAMALE, 2002). Por exemplo, uma máquina de lavar pode utilizar velocidades diferentes para cada ciclo de lavagem. Historicamente, os sistemas de engrenagens mecânicas foram utilizados para a obtenção de velocidade variável. Recentemente, energia e sistemas de controle eletrônicos têm condições suficientes de substituir engrenagens mecânicas. Estes aparelhos não só controlam a velocidade do motor, mas pode melhorar as características de desempenho dinâmico do motor. Além disso, o acionamento com uso de inversor de frequência pode reduzir o consumo de energia médio consumido pelo motor (FRANCHI, 2008).

O controle de um motor de indução é complexo devido às suas características não-lineares. Embora existam diferentes métodos de controle, o controle escalar V/F é o método mais comum de controle de velocidade. Este método é mais adequado para aplicações sem requisitos de controle de posição ou a necessidade de alta precisão de controle de velocidade. Exemplos destas aplicações incluem aquecimento, ar condicionado, ventiladores e sopradores. Controle V/F pode ser implementado usando microcontroladores de baixo custo como o modelo PIC18F2431 do fabricante Microchip®, em vez de utilizar processadores mais elaborados como os DSPs (digital signal processors- Processadores digitais de sinais) (FRANCHI, 2008).

2.1.1. Identificação das características técnicas de um motor

A placa de identificação de um motor de indução típico lista os seguintes parâmetros:

- Tensão nominal de alimentação do terminal em Volts
- Frequência nominal do fornecimento em Hz
- Corrente nominal em ampéres
- Base de velocidade em RPM
- Potência em Watts ou CV
- Torque em Newton metros ou Libra-Polegada
- Tipo de isolamento Classe A, B, F ou H
- Tipo de ligação do estator (para as 3-fase apenas), estrela (Y) ou delta (Δ).

Quando a tensão e a frequência nominais são aplicadas aos terminais de um motor de indução, ele consome a corrente nominal (ou a potência correspondente) e funciona na velocidade nominal e pode entregar o torque nominal (FRANCHI, 2008).

2.1.2. Rotação de um Motor

Quando a alimentação nominal de um motor CA é aplicada ao estator, ele gera um fluxo magnético de magnitude constante, girando na velocidade síncrona (FITZGERALD, 1978). O fluxo passa através da abertura de ar existente entre o estator e o rotor, varre a superfície através do rotor e os condutores do rotor parado. Uma força eletromotriz (FEM) é induzida nos condutores do rotor devido às diferenças de velocidade relativa entre o fluxo rotativo e os condutores estacionários (FITZGERALD, 1978).

A frequência da F.E.M. induzida é o mesmo que a frequência síncrona. Sua magnitude é proporcional à velocidade relativa entre o fluxo e os condutores. As barras do rotor estão curto-circuito nas extremidades, a F.E.M. induzida produz uma corrente nos condutores do rotor. A direção da corrente do rotor opõe-se a velocidade relativa entre o fluxo rotativo produzido pelo estator e rotor. Para reduzir a velocidade relativa, o rotor começa a girar na mesma direção do fluxo e tenta apanhar o fluxo rotativo. Mas, na prática, nunca o rotor consegue alcançar o campo do estator. Assim, a velocidade do rotor é mais lenta do que a velocidade do campo do estator. Esta diferença de velocidade é chamada de velocidade de escorregamento. Esta velocidade de escorregamento depende da carga mecânica no eixo do motor (FITZGERALD, 1978).

A frequência e a velocidade do motor são chamadas de frequência síncrona e velocidade síncrona. Velocidade síncrona é diretamente proporcional à taxa de frequência de fornecimento e número de pólos no motor . Velocidade síncrona de um motor de indução é dada por (FITZGERALD, 1978):

$$\text{Velocidade síncrona: } \omega_S = \frac{120 \times F}{P} \quad \text{onde:}$$

- F = Frequência nominal de um motor em Hz;
- P = Número de pólos de um motor

OBS: O número de pólos é o número de caminhos paralelos para o fluxo de corrente no estator. O número de pólos é sempre um número par para equilibrar o fluxo de corrente, sendo os motores de 4 pólos os mais utilizados (FITZGERALD, 1978).

Com base na construção do rotor, os motores de indução são classificados em duas categorias: motores de gaiola de esquilo ou escorregamento e motores de anéis. A construção do estator é a mesma em ambos os motores (YEDAMALE, 2002).

2.1.3. Motores de indução tipo gaiola de esquilo

Quase 90% dos motores utilizados atualmente são motores de indução do tipo gaiola de esquilo. Isso ocorre porque o motor gaiola de esquilo tem uma construção simples e robusta. O rotor consiste de um núcleo cilíndrico colocado axialmente e laminado, com ranhuras paralelas para a colocação dos condutores (YEDAMALE, 2002). Cada um carrega um slot de cobre, alumínio. Se os slots são semi-fechados, essas barras são inseridas a partir das extremidades (YEDAMALE, 2002). Essas barras do rotor são permanentemente em curto-circuito em ambas as extremidades por meio dos anéis colocados no final, como mostrado na figura 1. Esta montagem total assemelha-se a aparência de uma gaiola, o que dá o seu nome. As ranhuras do rotor não são exatamente paralelas ao eixo. Em vez disso, eles recebem uma inclinação por duas razões principais (YEDAMALE, 2002):

a) Para fazer o motor funcionar tranquilamente, reduzindo o ruído magnético (YEDAMALE, 2002).

b) Para ajudar a reduzir a tendência de travamento do rotor. Ranhuras do rotor tendem a permanecer fechadas com as ranhuras do estator devido à atração magnética direta entre os dois. Isso acontece se o número de ranhuras do estator é igual ao número de ranhuras do rotor (YEDAMALE, 2002).

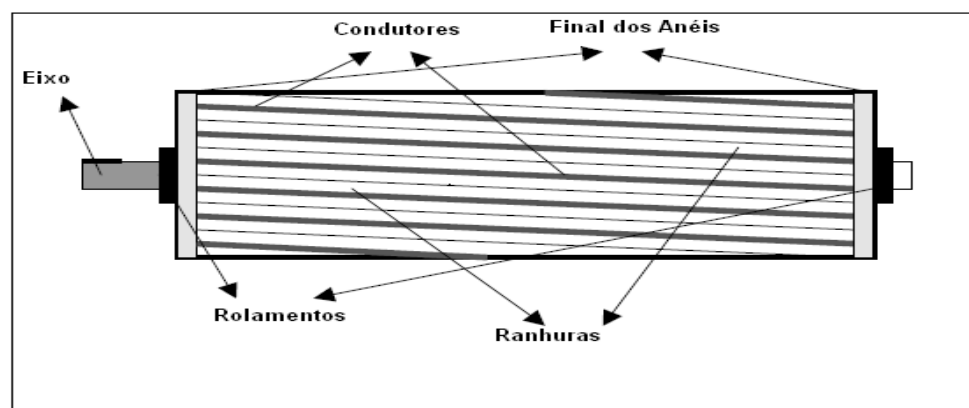


Figura 1. Típico rotor de gaiola

Fonte: Adaptado de Yedamale (2002)

2.1.4. Motores de anéis

Os enrolamentos no rotor são conectados a uma fonte CC externa através de três anéis de deslizamento isolado montado no eixo, com escovas repousando sobre eles. Isso permite a introdução de um resistor externo ao rotor. O resistor externo pode ser usado para aumentar o torque de partida do motor e mudar a velocidade característica de torque. Quando executando sob condições normais, os anéis são curto-circuitados, usando um colar de metal externo, que é empurrado ao longo do eixo para conectar os anéis. Assim, em condições normais, o deslizamento do anel funciona como um motor gaiola de esquilo (YEDAMALE, 2002). Este tipo de motor não é muito usado.

2.1.5. Característica do torque em motores de indução

A Figura 2 mostra a curva de velocidade características típica de torque de um motor de indução. O eixo X mostra a velocidade e o escorregamento. O eixo Y mostra o torque e a corrente. As curvas são desenhadas com tensão e frequência nominal fornecida ao estator. Durante o arranque, o motor normalmente consome sete vezes a corrente nominal. Na partida, o motor oferece 1,5 vezes o torque nominal do motor. Este torque de partida também é chamado de torque de rotor bloqueado. À medida que a velocidade aumenta, a corrente consumida pelo motor diminui ligeiramente (ver figura 2), (YEDAMALE, 2002).

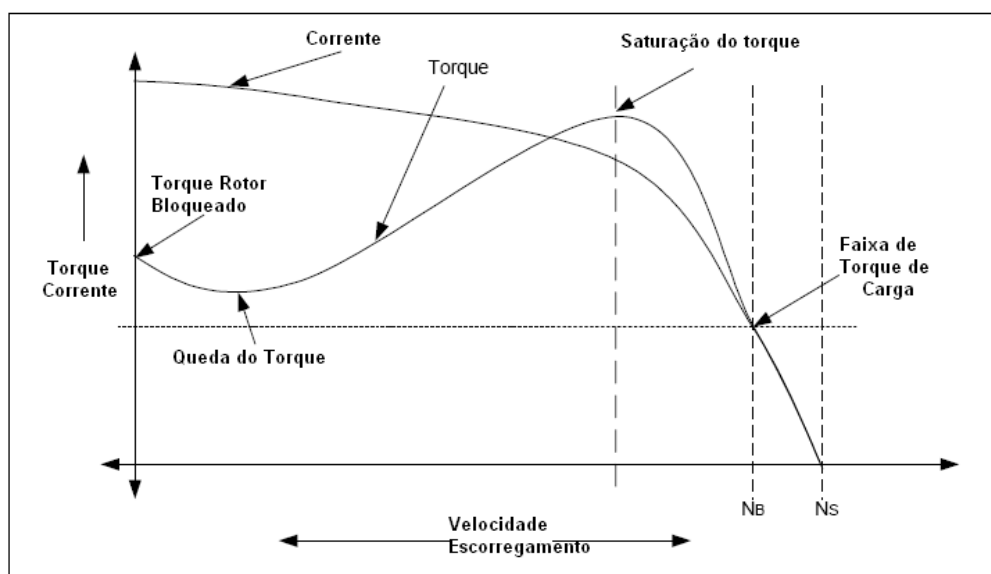


Figura 2. Velocidade X Torque em motores de indução

Fonte: Adaptado de Yedamale (2002)

Na baixa velocidade, se a carga no eixo do motor é aumentada, além de seu torque nominal, a velocidade começa a cair e aumenta o escorregamento. Quando o motor está funcionando em aproximadamente 80% da velocidade síncrona, a carga pode aumentar até 2,5 vezes o torque nominal. Este torque é chamado de torque máximo (YEDAMALE, 2002). Se a carga do motor é aumentada ainda mais, o motor não vai ser capaz de segurar qualquer carga adicional e irá parar. Além disso, quando a carga é aumentada, além da carga nominal, a corrente de carga aumenta. Devido a esse maior fluxo de corrente nos enrolamentos, as perdas inerentes à enrolamentos aumentam também. Isto leva a uma maior temperatura nos enrolamentos do motor. Enrolamentos do motor podem resistir a temperaturas diferentes, com base na classe de isolamento utilizada nos enrolamentos e sistema de arrefecimento utilizado no motor. Alguns fabricantes fornecem os dados sobre a capacidade de sobrecarga e de carga ao longo ciclo de trabalho. Se o motor está sobrecarregado por mais tempo do que o recomendado, o motor pode queimar. Como visto na velocidade acima da nominal, o torque é altamente não-linear com a variação da velocidade. Em muitas aplicações, a velocidade deve ser variada, o que torna o torque variável (YEDAMALE, 2002).

2.2. TEORIA DE ACIONAMENTO MODO ESCALAR V/F

Como pode ser visto na curva torque-velocidade da figura 2, o motor de indução extrai a corrente nominal e entrega o torque na velocidade base. Quando a carga é aumentada (excesso de carga nominal) a velocidade do motor cai aumentando o deslizamento. Como foi visto na seção anterior, o motor pode ter até 2,5 vezes o torque nominal, com cerca de 20% de queda na velocidade (FRANCHI, 2008). Qualquer aumento de carga no eixo pode parar o motor. O torque desenvolvido pelo motor é diretamente proporcional ao campo magnético produzido pelo estator (FRANCHI, 2008). Assim, a tensão aplicada ao estator é diretamente proporcional ao produto do fluxo do estator e velocidade angular (FRANCHI, 2008). Isso faz com que o fluxo produzido pelo estator seja proporcional ao quociente entre a tensão aplicada e a frequência do fornecimento. Variando a frequência, a velocidade do motor pode ser variada. Portanto, variando a voltagem e frequência pela mesma razão, o fluxo e, portanto, o torque, pode ser mantido constante em toda a gama de velocidades (FRANCHI, 2008).

O funcionamento de um motor de indução CA é regido por dois princípios (BURROUGHS,2004):

1. Base de velocidade é diretamente proporcional à frequência da corrente alternada aplicada ao estator e do número de pólos do motor (BURROUGHS, 2004).

2. O torque é diretamente proporcional ao quociente entre a tensão aplicada e a frequência da corrente alternada aplicada. Portanto, a velocidade pode ser controlada através da variação da frequência de entrada da corrente alternada aplicada e o torque pode ser mantido constante através da variação da amplitude em proporção direta a frequência (BURROUGHS, 2004). Estes são os dois objetivos básicos do controle em malha aberta V/F ou, como é mais conhecido, controle escalar.

$$\text{Tensão Estator (V)} = [\text{Fluxo Estator } \Phi] \times [\text{Velocidade Angular } \omega] \quad \text{Eq.(1)}$$

$$V = \Phi \times 2\pi f \quad \text{Eq.(2)}$$

$$\Phi = V/f \quad \text{Eq.(3)}$$

Isso torna constante a relação V / F , sendo este o modo de controle de velocidade mais comum de um motor de indução. A Figura 4 mostra a relação entre a tensão e torque versus frequência. Mostra também tensão e a frequência sendo aumentada até a velocidade nominal. Na base da velocidade, a tensão e a frequência atingem os valores avaliados como listado na placa de identificação. É possível conduzir o motor além da velocidade base, aumentando a frequência. No entanto, a tensão aplicada não pode ser aumentada para além da tensão nominal. Portanto, somente a frequência pode ser aumentada, o que resulta no enfraquecimento de campo e o torque disponível a ser reduzido (FRANCHI, 2008). Acima da velocidade nominal, os fatores que regem torque tornar-se complexo, pois o atrito e perdas aumentam significativamente em altas velocidades. Assim, a curva de torque se torna não-linear com relação a velocidade ou frequência (YEDAMALE, 2002).

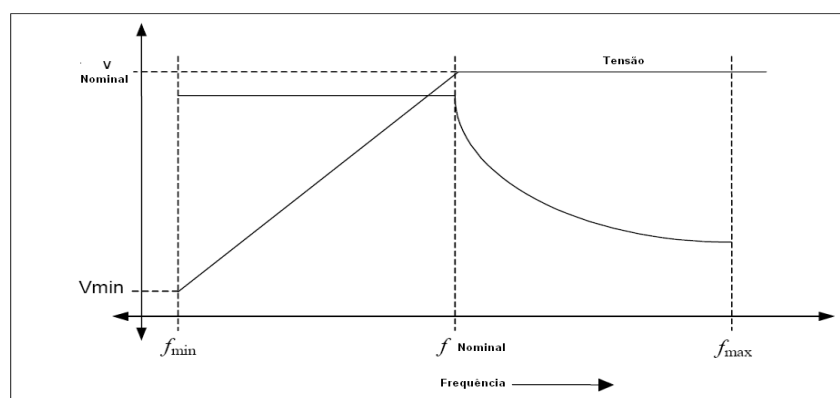


Figura 3. Relação Torque e Tensão X Frequência

Fonte: Adaptado de Yedamale (2002)

3. PROJETO

A seguir detalham-se todas as etapas desenvolvidas no projeto “INVERSOR PWM” e os resultados obtidos. Para o teste do programa foram criados dois protótipos, denominados de placa de potência e placa de controle. A placa de potência fará toda a alimentação do inversor utilizando 3 fontes sendo uma de 220 VCA/ 15VCC outra de 220VCA/5VCC e, por último 220VCA/ 315VCC e o módulo de acionamento dos IGBTs. A placa de controle contém entre outros elementos o microcontrolador que irá gerar a lógica de acionamento dos módulos IGBTs.

3.1.1. Princípio de Funcionamento da Placa de Potência

Para alimentar o motor, a tensão trifásica CA é convertida para uma tensão CC usando uma ponte retificadora trifásica de diodos. Um capacitor filtra a ondulação de tensão no barramento CC. Um módulo de IGBTs é usado para converter a tensão no barramento CC para a tensão com amplitude, frequência e formato variáveis . O motor é ligado ao módulo IGBTs, como mostrado na Figura 5. O inversor possui seis chaves eletrônicas que são controladas de tal forma a gerar uma saída CA com amplitude, frequência e forma desejadas.

Os sinais PWM são gerados a partir de um programa específico de controle presente no microcontrolador e aplicado nos 6 IGBTs. A tensão de fase é determinada pelo ciclo de trabalho dos sinais PWM. Quando os interruptores estão ligados, a corrente flui do barramento CC para o enrolamento do motor. Os enrolamentos do motor são altamente indutivos, e possuem energia elétrica armazenada. Esta energia precisa ser dissipada enquanto interruptores estão desligados. Diodos são conectados em paralelo com os interruptores para fornecer um caminho para a corrente que surge quando os interruptores são desligados. Estes diodos são também chamados de diodos de roda livre (FITZGERALD, 1978). Os IGBTs da parte inferior, não devem estar ligados ao mesmo tempo que a da parte superior, para se evitar um curto-circuito. Um tempo morto é dado no programa do microcontrolador entre desligar o IGBT de comutação superior e ligar o IGBT da parte inferior e vice-versa. Isso garante que dois IGBTs, de um mesmo ramo, não conduzam ao mesmo tempo, quando alterar os estados de ligado para desligado, ou vice-versa.

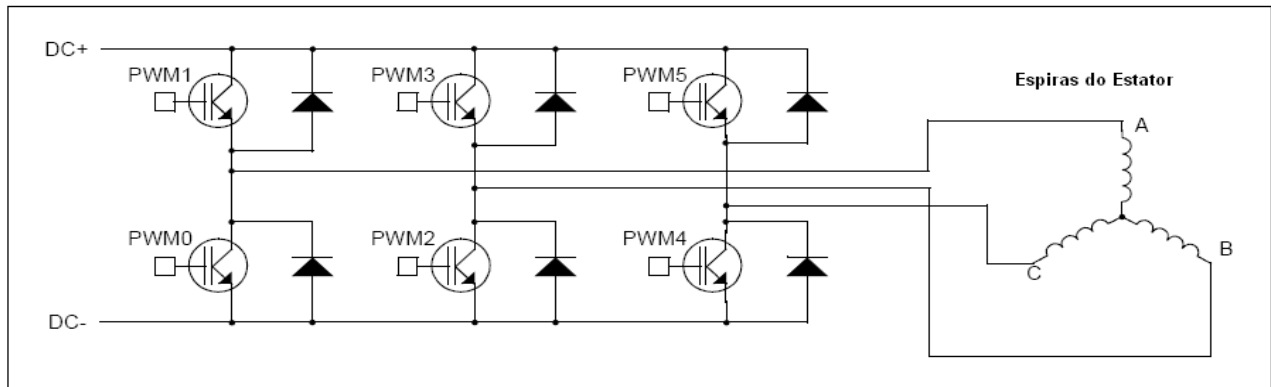


Figura 4. Simbologia do módulo IGBT e o estator do motor de indução
 Fonte: Adaptado de Yedamale (2002)

A seguir segue o esquema do protótipo montado da placa de potência:

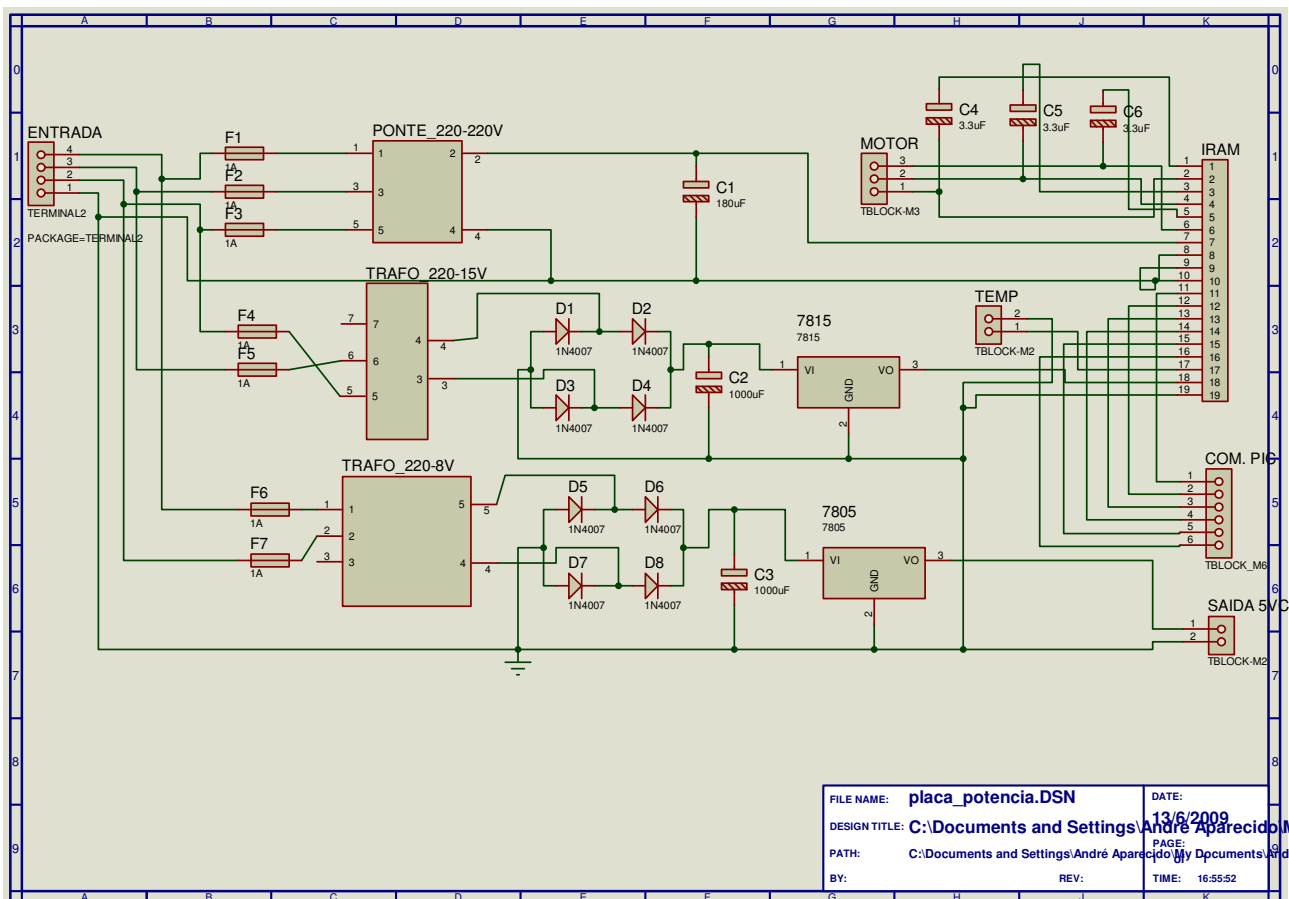


Figura 5. Circuito da montagem da placa de Potência

Para o protótipo da placa de potência foram desenvolvidas três fontes que alimentarão o inversor e a placa de controle digital. Existe uma única entrada que alimentará todos os circuitos,

sendo a mesma de $220V_{CA}$ trifásico. Antes de cada alimentação cada fonte tem um fusível para garantir a proteção em caso de curto-circuito, ou sobrecorrente conforme mostra a figura 6. Dentre estas fontes está uma de entrada $220V_{CA}$ monofásica saída $5V_{CC}$. Ela possui um transformador de encapsulamento fechado que recebe em sua entrada $220V_{CA}$ monofásico (fase e terra) e tem como saída $8V_{CA}$ e com potência de $5VA$ conforme a figura 6. Após ter abaixado para o valor de $8V_{CA}$ a tensão é retificada por uma ponte retificadora de onda completa com diodos modelo 1N4007 saindo uma tensão unidirecional pulsante. Para reduzir a ondulação de tensão, é utilizado um capacitor de $1000\mu F$ conforme mostra a figura 6. Para que a tensão fique fixa em $+5V_{CC}$, conforme necessidade, utiliza-se um regulador de tensão modelo 7805 que mantém a tensão de saída em $5V_{CC}$. Esta fonte servirá para alimentar a placa de controle digital.

A segunda fonte é a de entrada $220 V_{CA}$ e saída $15V_{CC}$. Ela possui um transformador de entrada de $220V_{CA}$ e saída $15V_{CA}$ com potência de $15VA$ conforme a figura 6. Após ter abaixado o valor de tensão para $15V_{CA}$ esta tensão também é retificada por uma ponte de onda completa formada por diodos modelo 1N4007. Na saída da ponte é colocado um capacitor para filtrar o ripple de valor $1000\mu F$ conforme mostra a figura 6. Para fixar a tensão em $+15V_{CC}$, conforme especificado foi utilizado um regulador de tensão modelo 7815. Esta fonte servirá para alimentar o módulo IGBTs.

A terceira e última fonte é a de entrada $220 V_{CA}$ e saída $315V_{CC}$. Ela possui como alimentação a rede CA trifásica que é retificada e filtrada, gerando um barramento CC. Para esta retificação, da rede trifásica, foi utilizada uma ponte de diodos de potência, trifásica marca SEMIKROM®, e em sua saída foi colocado um capacitor para filtragem de $180\mu F$.

Para que o motor possa ser acionado, foi utilizado um módulo de potência com 6 IGBTs, que são responsáveis pelo chaveamento e conversão da tensão CC em tensão CA na saída do inversor. Este módulo consiste em seis transistores (IGBTs) ligados em uma configuração de ponte inversora trifásica. (Ver diagrama no ANEXO I).

Este módulo foi escolhido por atender aos requisitos de tensão de até $600V$ e corrente eficaz de até $5A_{rms}$ (na pior condição de temperatura $T=100^{\circ}C$) suficientes para acionar um motor trifásico de até $750W$ (~1HP) .

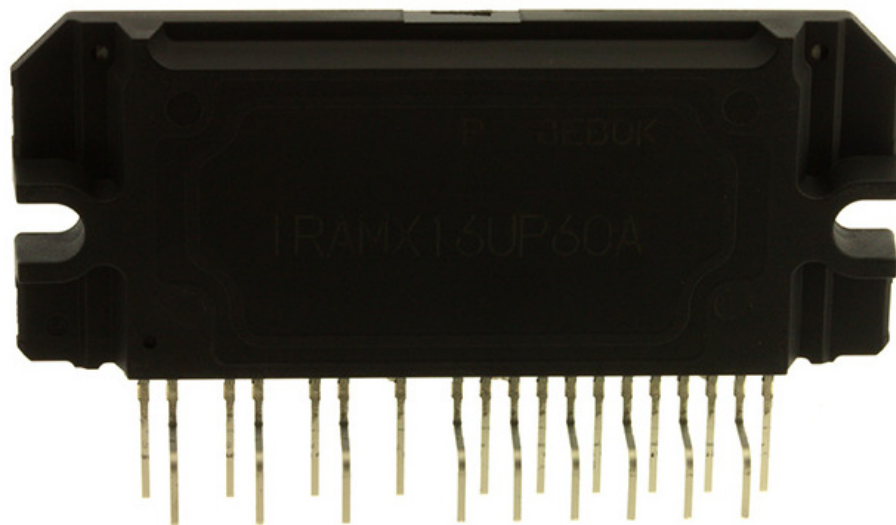


Figura 6. Módulo de Potência IRAM 16UP60A

Fonte: Adaptado Datasheet do componente IRAMS16UP60A.

Na saída deste módulo foram instalados 3 capacitores de desacoplamento de 3,3uF, de acordo com manual do fabricante conforme mostrado na figura 6.

Na figura 8 segue a foto do protótipo da placa de potência montada e seus respectivos componentes.

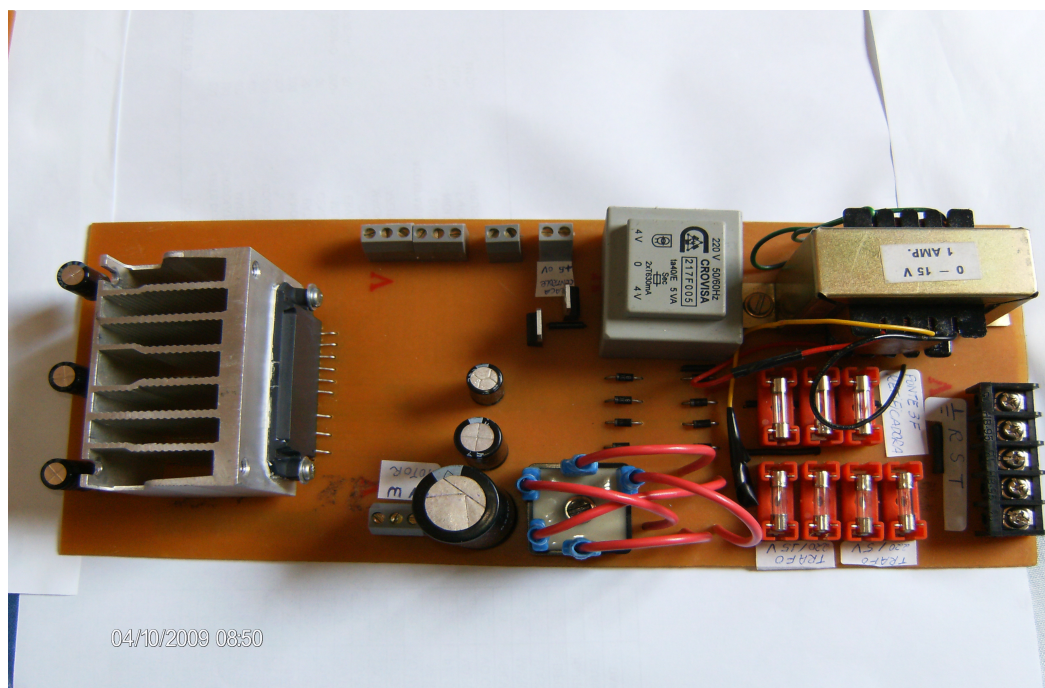


Figura 7. Protótipo da placa de Potência

3.1.2. Princípios de Funcionamento da Placa de Controle

Para se controlar o módulo de IGBT é necessária uma lógica de funcionamento para que se consiga na saída deste módulo uma tensão CA que tenha os requisitos necessários para se acionar um motor de indução. Para isto foi criado um protótipo de uma placa de controle.

A placa de controle possui vários componentes responsáveis pelo acionamento do módulo IRAM, como demonstrado na figura 9, dentre eles se destacam:

- Chave on-off responsável pela ligação da placa
- LED de indicação que a placa foi ligada
- 2 botões de acionamento que serão responsáveis pelo acionamento do inversor.
- 6 opto-acopladores que farão a isolação das saídas do microcontrolador, para segurança da placa de controle.
- Microcontrolador modelo 18F2431 do fabricante Microchip®
- Display de LCD 2 linhas por 16 colunas
- Oscilador a cristal de 20MHz para o clock do microcontrolador.
- Potenciômetro responsável por variar a velocidade do motor a ser acionado.
- Bornes para a interligação entre a placa de controle e a placa de potência.
- Resistores diversos para limitação de corrente

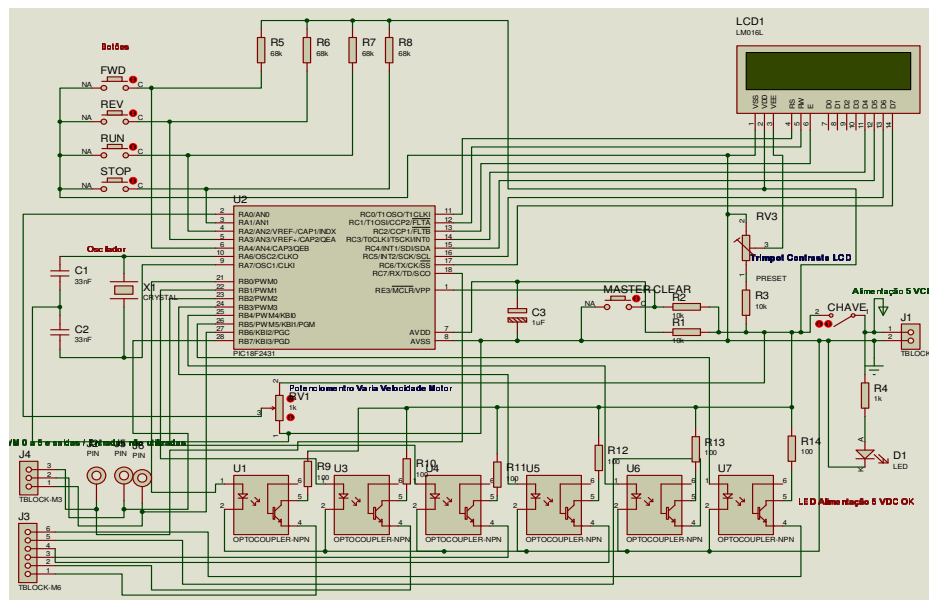


Figura 8. Esquema de ligação placa de controle

Esta placa será alimentada pelos 5 volts vindos da placa de potência o qual foi explicado no item anterior. Estes 5 volts alimentarão o microcontrolador, o display de LCD, e os opto-acopladores. Segue abaixo o protótipo da placa montada com a distribuição dos componentes.

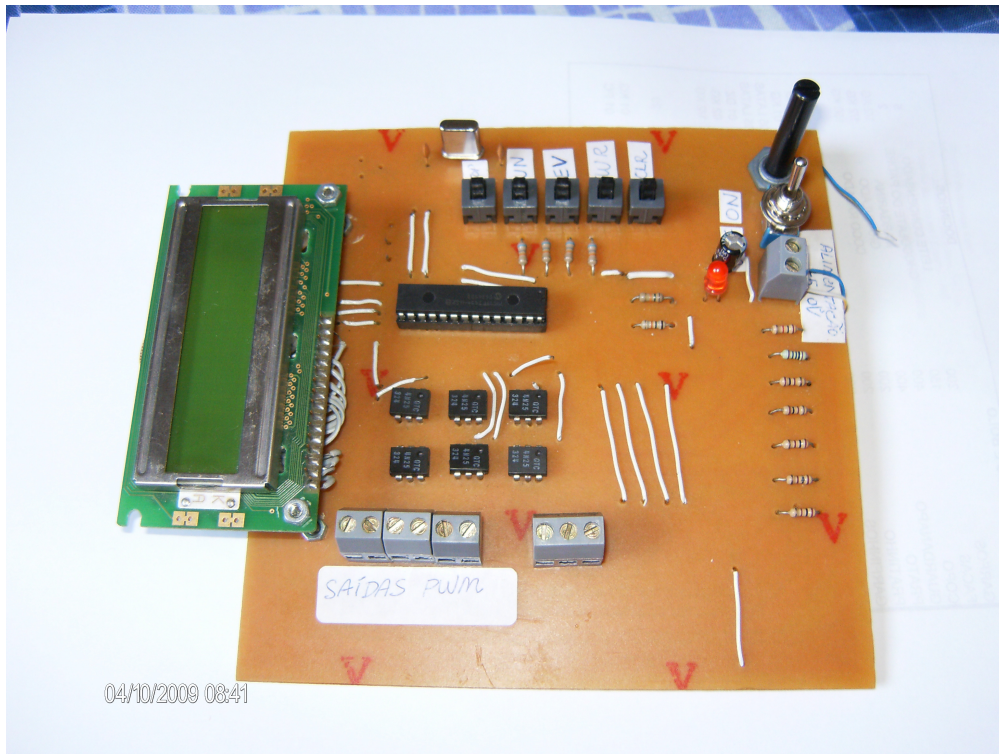


Figura 9. Protótipo da Placa de controle

3.1.3. O microcontrolador 18F2431

O microcontrolador escolhido para este projeto foi o PIC18F2431 que possui as seguintes características:

- Seis saídas PWM, para controlar os seis transistores de potência do inversor trifásico. Poderíamos também utilizar apenas três saídas do microcontrolador e usar um inversor lógico para obter as outras três, porém desta maneira temos uma operação mais flexível (podemos, por exemplo, ajustar o tempo morto por software) e segura, pois podemos manter todos os transistores desligados.
- Possui uma entrada para capturar o sinal de um sensor de velocidade (“Motion Feedback Module”) que facilita a aquisição de informação de velocidade, caso se queira implementar.

- Memória flash e possibilidade de programação In-Circuit (isto é, sem tirar o microcontrolador do circuito) que aceleram o processo de desenvolvimento.
- Módulo de comunicação serial RS232.
- Pinos de I/O suficientes para a interface com o usuário (LCD e botões).
- Disponibilidade de conversores A/D para o potenciômetro usado como referência e para monitoração e realimentação de variáveis de interesse.
- Operação em 8 bits e até 40MHz ,utilizados para atualização do PWM, rotinas de controle, comunicação e interface.

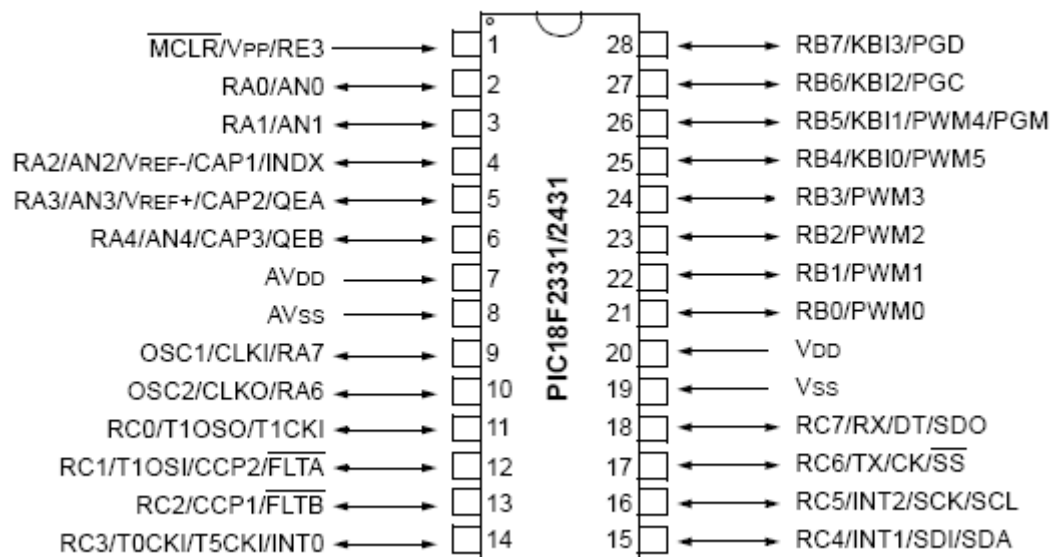


Figura 10. Microcontrolador PIC 18F2431

Fonte: Adaptado do Datasheet do microcontrolador

3.1.4. Princípios de funcionamento do Programa

Controle escalar (V/F) para motores de indução CA é frequentemente operado em malha aberta, sem realimentação da posição do eixo e da velocidade do motor. A relação tensão X frequência (V/F) se mantém constante para proporcionar um torque constante em toda a gama de funcionamento. Esta forma de controle é relativamente barato e fácil de implementar. Não é utilizado realimentação de variáveis elétricas do rotor, sendo o mesmo forçado a seguir a rotação do campo girante estabelecido no estator, com certa quantidade de deslizamento, dependendo da carga.

Para acionar o motor de indução CA, as 3 fases necessárias são criadas na saída do inversor por modulação PWM, conforme se vê pela figura 12. Ao ser alterado os ciclos de trabalho do sinal PWM de forma regular, as saídas PWM são moduladas para sintetizar formas de ondas senoidais (trifásico CA) entre os três enrolamentos do motor conforme a figura 13. Uma tensão CA é aplicada aos três enrolamentos do estator como três correntes senoidais, de mesma amplitude e frequência, mas defasados entre si de 120 graus. Como resultado, a corrente no estator gera um campo magnético girante. Este campo rotativo induz a força eletromotriz no rotor, que por sua vez produz um campo magnético no rotor que tentará se alinhar com o campo magnético girante no estator. Isso faz com que o rotor gire.

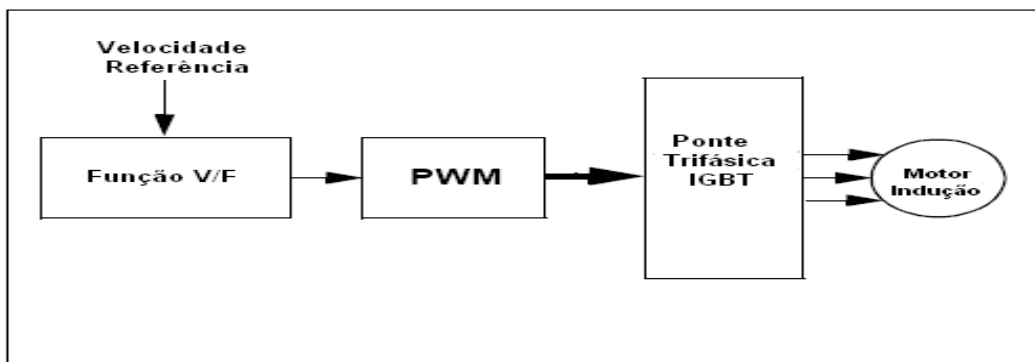


Figura 11. Diagrama em Blocos do controle V/F

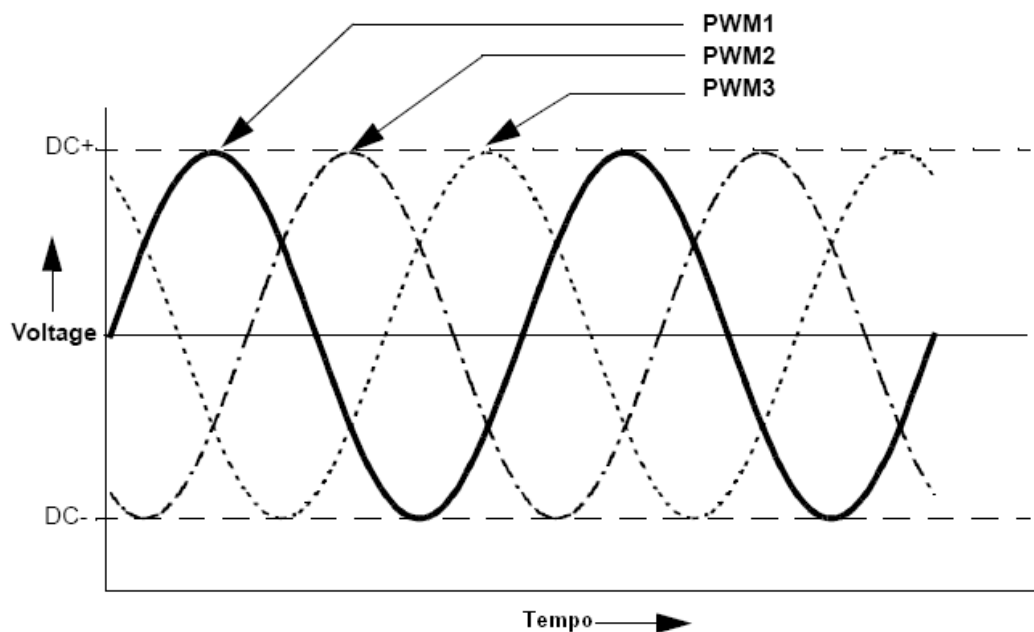
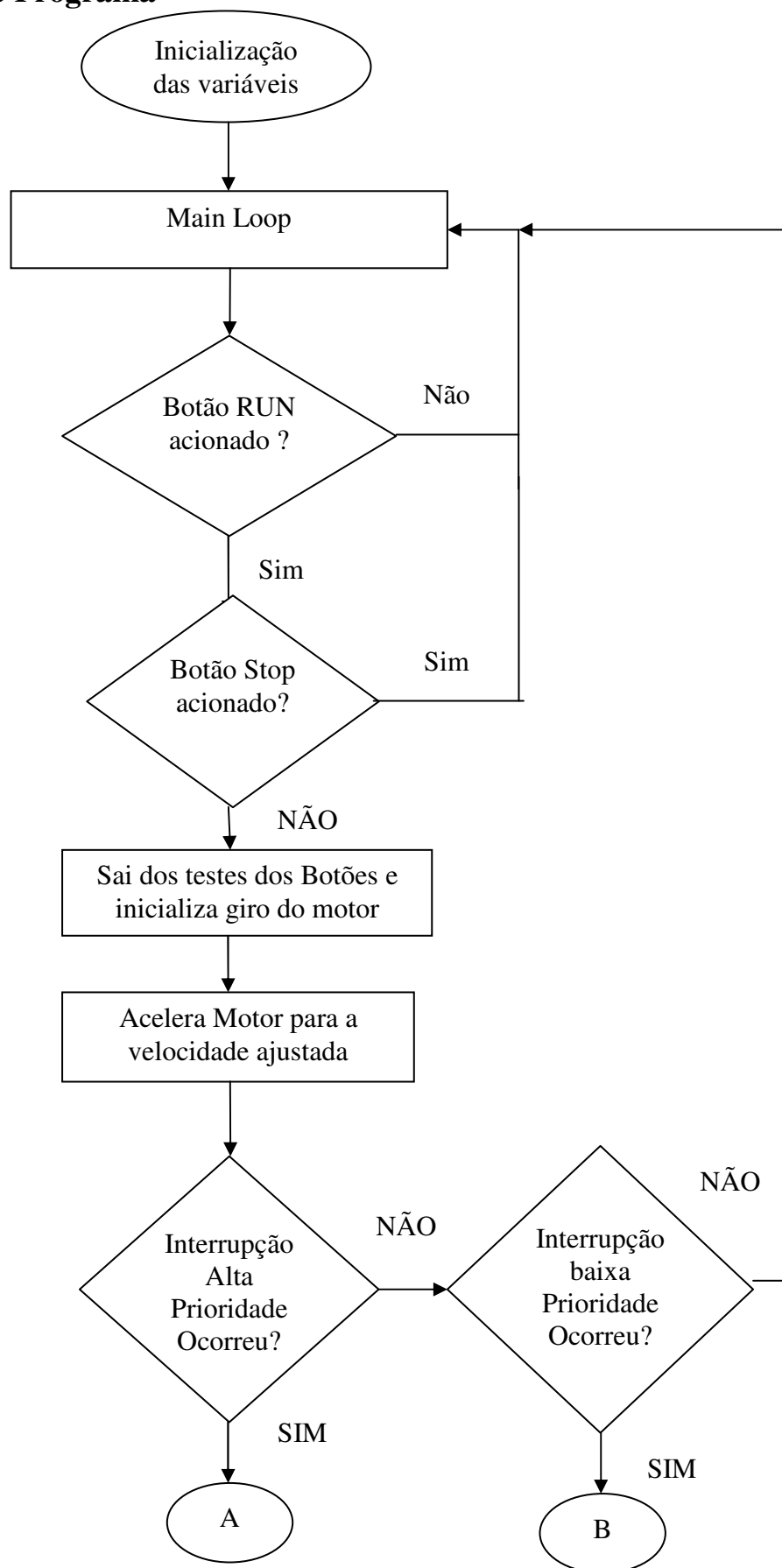
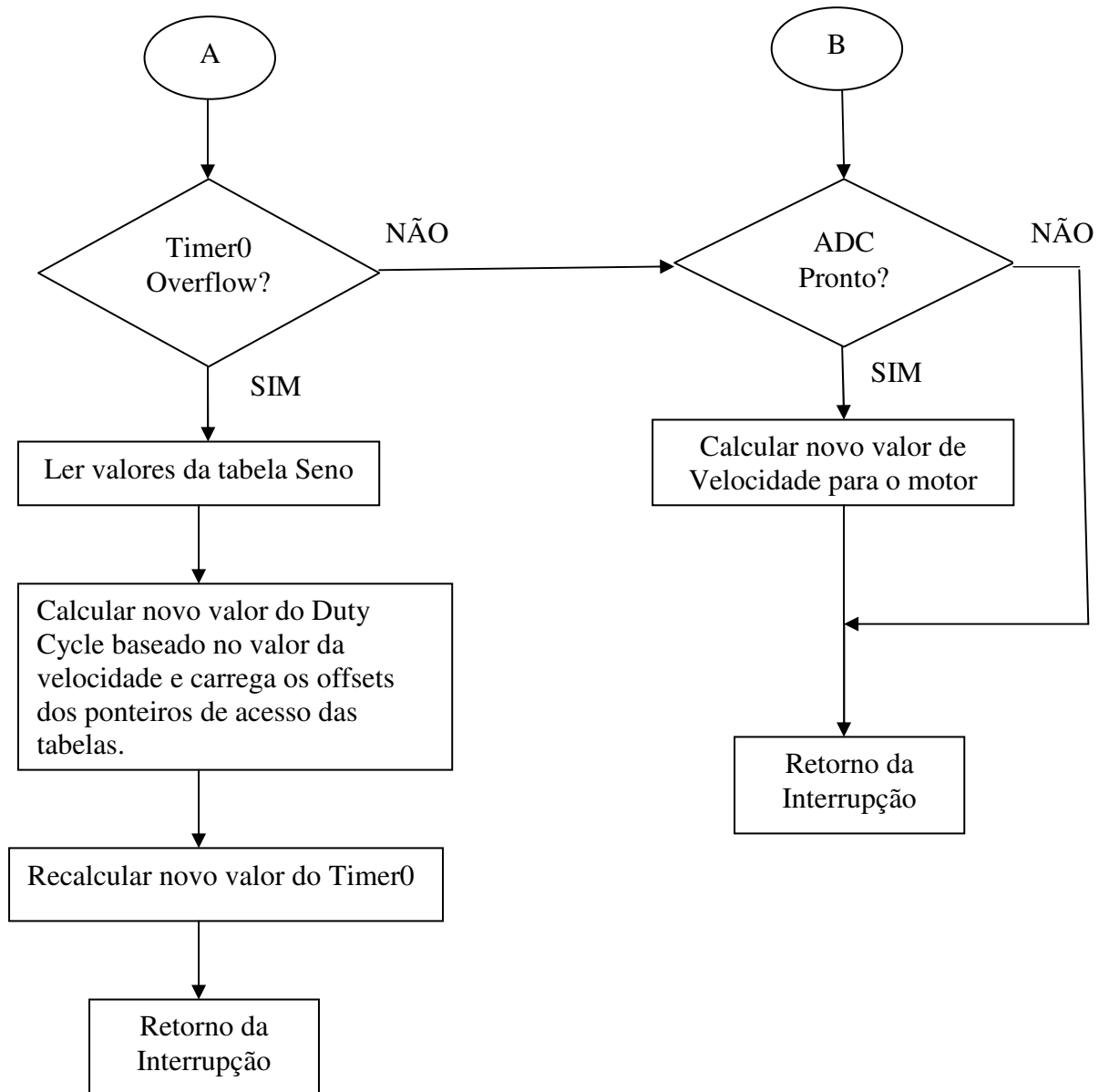


Figura 12. Síntese das 3 formas de onda

Fonte: Adaptado de Burroughs (2004)

3.1.5. Fluxograma do Programa





3.1.6. Detalhamento do Programa

3.1.7. Inicialização

A rotina de inicialização define os pinos das portas para os estados desejados e inicializa os periféricos. O High-Speed ADC, Power Control PWM (PCPWM), inicialização do LCD e entradas de hardware são inicializados definindo seus registradores de acordo com o hardware implementado. Inicialização desses periféricos é discutida em detalhes abaixo

3.1.8. Loop Principal

O laço principal verifica continuamente status dos botões. Dois push Button RUN e STOP são ligados no microcontrolador que ficam sendo monitorados a todo instante havendo um pedido de parada ele zera todos os valores de duty cycle utilizados fazendo o motor desacelerar e parar. Nos dois botões foram implantadas rotinas para filtrar os acionamentos indesejáveis (DEBOUNCE).

3.1.9. Interrupt Service Routine (ISR) (Rotina de serviço de interrupção)

ISR trata das interrupções do ADC (Conversor analógico Digital), rotina de baixa prioridade e do estouro do temporizador TMR0, rotina de alta prioridade. Para o conversor A / D é utilizado o canal AN0. AN0 é usado como entrada para a referência de velocidade. O estouro do TMRO será detalhado logo abaixo.

3.1.10. Inicialização dos registradores responsáveis pelo modo PWM

O Controle do módulo PWM simplifica a tarefa de acionar uma ponte trifásica no barramento CC, prevendo três pares de saídas complementares PWM, com o tempo morto inserido entre os canais. Ela também fornece hardware baseado em entradas de falhas que são capazes de desligar o sinal PWM nas saídas completamente em uma situação de falha. Foi inicializado o módulo PCPWM seguindo as instruções fornecidas pelo datasheet do microcontrolador ;

1. Configurado a base de tempo PCPWM:

a) Selecionado o valor base de tempo PWM postscale de 1:1.

b) Selecionado o PWM base de entrada de 1:1 (FOSC / 4).

c) Configurando a base de tempo para Free-PWM Running (modo de operação de ponta-alinhada).

2. Colocado os registradores PTPERH: PTPERL (Registradores que controlam a frequência de PWM) para obter uma frequência PWM de 20 kHz.

3. Configurado as saída PCPWM: de PWM0 até PWM5 como saídas.

a) Definido as saídas PWM pares I / O (PWM 0 / 1, 2 / 3 e 4 / 5) como pares complementares.

4. Configurado o disparador de eventos especiais:

- a) Definido o postscaler gatilho especial de eventos para 1:1.
 - b) Configurado o evento especial de gatilho para ocorrer quando a base de tempo estiver para contar para cima.
 - c) Permitido atualizações do ciclo de trabalho e registros de buffer por período.
 - d) Configurado para saídas assíncrona do registrador OVDCON.
5. Configurado o tempo morto do PCPWM:
- a) Selecionado FOSC / 2 como o prescaler do tempo morto.
 - b) Definido valores para conseguir 1,0 μs (micro segundos) de tempo morto.
6. Desabilitado e substituído as saídas nos pinos PWM definindo os bits do registrador Povd.
7. Limpado os especiais eventos pares para registrar o novo valor de ciclo (SEVTCMPH: SEVTCMPL).
8. Limpado todos os valores dos registradores (PDCxH: PDCxL) para definir os ciclos a 0.

Abaixo segue o trecho do programa que inicializa o módulo PCPWM :

```

INIT_PCPWM
movlw b'00000000' ; O registrador PTCON0 foi configurado da seguinte forma:
movwf PTCON0 ; a) Postscale valor setado 1:1
; b) Tempo base do PWM será Fosc/4
; c) Tempo base do PWM modo é free-running para operação em
edge-aligned

movlw 0xF9 ; PTPERL e PTPERH são setados para 20KHz frequência PWM.
movwf PTPERL ; PTPERH:PTPERL = ((Fosc/4)/(PWMfreq x PTMRps))-1
(datashet)
movlw 0x00 ; PTPERH:PTPERL = ((20MHz/4)/(20KHz x 1))-1 = 249d = F9h
movwf PTPERH

movlw b'01000000' ; PWMCON0 foi configurado da seguinte forma:
movwf PWMCON0 ; a) PWM0, PWM1, PWM2, PWM3, PWM4, e PWM5 são ligados
para saída. ; b) Todos PWM I/O pares estão setados em complementar modo

movlw b'00000001' ; PWMCON1 foi configurado da seguinte forma:
movwf PWMCON1 ; a) Eventos Especiais trigger post-scaler são setados para 1:1
; b) Eventos especiais ocorridos quando tempo-base está
contando para cima.

```

Figura 13. Detalhe do programa que inicializa o módulo PWM

3.1.11. Inicialização dos registradores responsáveis pelo ADC

O módulo de registradores do ADC incorpora várias características, tais como o Auto-Modo de conversão e um buffer de resultado FIFO, que reduzem a sobrecarga do firmware associadas à monitoração múltiplos canais analógicos e aumenta a taxa de transferência ADC . No projeto ele é responsável por transformar o sinal analógico do potenciômetro em um sinal conhecido pelo microcontrolador.

Para inicializar o módulo HSADC foram realizados os seguintes passos:

1. Configurado a operação ADC:

a) Ativado Modo Loop Contínuo.

b) Ligado Multi-Modo Channel.

c) Configurado auto-seqüência de conversão de exemplo para seqüência do Grupo A e Grupo B.

d) Atribuído VREF + e VREF para AVDD e AVSS, respectivamente.

e) Ativado o buffer FIFO.

f) Selecionado o formato da esquerda para justificar o resultado do A / D.

g) Definido o A / D e o tempo de aquisição de 12 TAD (obrigatório para a conversão seqüencial).

h) Definido o relógio de conversão do A / D para FOSC/32. Ligado o ADC.

2. Configurado os eventos de disparo das interrupções:

a) Definido as interrupção do A / D a ser gerada em cada 2^a e 4^a escrita do buffer FIFO.

b) Desativado disparo do ADC externo.

3. Configurado RA0 como entradas analógicas:

Abaixo segue um o trecho do programa que faz a inicialização dos registradores do conversor analógico digital ADC:

INIT_HSADC		
movlw	b'00000000'	; ADCON1 é configurado da seguinte forma:
movwf	ADCON1	; a) Vref+ e Vref- serão Avdd e Avss, respectivamente
		; b) O FIFO buffer está desabilitado
movlw	b'00110010'	; ADCON2 é configurado da seguinte forma:
movwf	ADCON2	; a) O resultado do A/D está justificado na esquerda
		; b) O tempo de aquisição do A/D está setado para 12Tad
		; c) O relógio de conversão do A/D está setado para Fosc/32.
movlw	b'01000000'	; ADCON3 é configurado da seguinte forma:
movwf	ADCON3	; a) As interrupções são geradas na 2 e 3 interrupção do buffer
FIFO		; b) Não é ligado nenhum externo triggers ADC.
movlw	b'00100000'	; ADCHS é configurado da seguinte forma:
movwf	ADCHS	; a) Grupo A é sinal da entrada AN0, referencia de velocidade
		; b) Group B desativado entrada analógica.
		; c) Group C desativado entrada analógica
movlw	b'00000001'	; ANSEL0 é configurado da seguinte forma:
movwf	ANSEL0	; a) AN0 Selecionado como entrada analógica.
movlw	b'11111111'	; b) Correspondentes bits do TRISA selecionados como
entradas.		
movwf	TRISA	
movlw	b'00000101'	; ADCON0 é configurado da seguinte forma:
movwf	ADCON0	; a) Single shot modo é ligado
		; b) Single-channel modo é ligado
		; c) Group A sinal amostrado.

Figura 14. Inicialização dos registradores do ADC

3.1.12. Controle V/F pelo software

O coração do controle do ACIM (AC Induction Motor- Motor de indução CA) é realizado com a PCPWM. O ciclo dos três canais de PWM são alteradas de forma regular utilizando a interrupção do Timer0 para sintetizar as três fases da onda senoidal de saída que vão para o motor. Uma tabela é armazenada na memória de programa. É transferido para a memória de dados durante a inicialização dos dados para acesso rápido. Três registros são usados como deslocamentos para a tabela através de endereçamento indireto. Cada um dos valores de deslocamento aponta para um dos valores na tabela, de modo que sempre há uma mudança de 120 graus entre as fases. (Cada uma das formas de onda na Figura 13 é criado por um deslocamento de registro associadas com o seu PWM.) O exemplo de código abaixo foi compilado no software MPLAB® e a linguagem utilizada foi Assembly. O código mostra como a tabela é lida usando os registros de endereçamento indireto. O potenciômetro determina o quanto será o sinal de referência da velocidade do motor.

UPDATE_PWM_DUTYCYCLES

```
movf  TABLE_OFFSET1,W      ;Carregar a tabela de deslocamento para a Fase 1
movf  PLUSW0,W              ;Usar endereçamento indireto para acessar valores da tabela
mulwf FREQUENCY, W          ;Valor da tabela X Frequência
movff PRODH,PDC0H_TEMP      ;Copiar resultado da multiplicação parte alta para PDC0H
movff PRODL,PDC0L_TEMP      ;Copiar resultado da multiplicação parte baixa para PDC0L
```

UPDATE_PWM2

```
movf  TABLE_OFFSET2,W      ; Carregar a tabela de deslocamento para a Fase 2
movf  PLUSW0,W              ; Usar endereçamento indireto para acessar valores da tabela
mulwf FREQUENCY, W          ; Valor da tabela X Frequência
movff PRODH,PDC1H_TEMP      ; Copiar resultado da multiplicação parte alta para PDC1H
movff PRODL,PDC1L_TEMP      ; Copiar resultado da multiplicação parte baixa para PDC1L
```

UPDATE_PWM3

```
movf  TABLE_OFFSET3,W      ; Carregar a tabela de deslocamento para a Fase 3
movf  PLUSW0,W              ; Usar endereçamento indireto para Acessar valores da tabela
mulwf FREQUENCY, W          ; Valor da tabela X Frequência
movff PRODH,PDC2H_TEMP      ; Copiar resultado da multiplicação parte alta para PDC2H
movff PRODL,PDC2L_TEMP      ; Copiar resultado da multiplicação parte baixa para PDC2L
```

Figura 15. Parte do programa que mostra o acesso a tabela por endereçamento indireto

3.1.13. Cálculo da frequência em malha aberta

No controle V/F, a frequência de acionamento do motor é diretamente calculada a partir da entrada do potenciômetro. Especificamente, o byte superior resultado do registrador do A / D é dividido por quatro para dar a unidade de saída da frequência em Hertz. No projeto, a velocidade tem um limite inferior de 12 Hz e um limite superior de 60 hertz. A função V / F determina a amplitude da unidade correspondente a essa frequência. Dado que a velocidade síncrona (em RPM) para um motor de indução é de $120 f / p$, onde p é o número de pólos do estator, a velocidade da unidade de destino (em RPM) pode ser diretamente calculada como 30 vezes o valor da ADRESH (120 vezes ADRESH dividido por 4), e dividido por p .

Cálculo da frequência:

Frequência pretendida = (ADRESH/4)

(12 < Frequência Pretendida > 60)

Cálculo da Velocidade:

$$N_s = \frac{120 \times F}{P} \times (\text{ADRESH}/4) = \mathbf{30. \text{ADRESH} / P}$$

Abaixo trecho do programa que trabalha com o valor transformado pelo ADC e move para uma variável chamada frequência:

```
READ_ADC_RESULTS

    movff  ADRESH,FREQUENCY ;Primeiro valor é grupo A assinalada para AN0
    movlw  0x30              ;Mínima frequência set para 12Hz
    cpfsgt FREQUENCY        ;Se frequência for inferior ou igual a 12Hz..
    movwf  FREQUENCY        ;seta ela para 12Hz
    movlw  0xF0              ;Limitando V/F para F= 60Hz (fator escalonado X4)
    cpfls  FREQUENCY        ;Se frequência for igual ou superior a 60Hz
    movwf  FREQUENCY        ;seta para 60Hz

    bsf    FLAGS1, FREQ_UPDAT ;seta flag para indicar FREQUENCY está sendo
carregada;
    bcf    PIR1,ADIF        ;ADIF flag é limpo para próxima interrupção

RETFIE FAST
```

Figura 16. Leitura do canal analógico

3.1.14. Resultados Alcançados

Como o objetivo do projeto é criar um programa com auxílio de um microcontrolador que possa controlar motores de indução trifásicos foram realizados por intermédio do software Proteus VSM uma simulação do modelo montado com o programa elaborado, conforme mostra a figura 9, e foram obtidas as seguintes formas de onda nas saídas do canal PWM do microcontrolador pegando as leituras nos pinos RB0, RB1, RB2, RB3, representadas no osciloscópio, respectivamente, pelos sinais PWM nas cores amarelo, azul, vermelho e verde. A frequência de chaveamento do sinal PWM pretendida é de 20 kHz, sendo a mesma alcançada nas simulações. O sinal PWM encontrado tem $5,0V_{CC}$ de pico a pico, valor este ideal para o acionamento do módulo IRAM.

Como podem ser observados, os sinais PWM produzidos são complementares. Portanto as saídas RBO, RB2 e RB4 são definidas como saídas defasadas em 120° e controladas pelo programa para conseguir os seis sinais PWM correspondentes para formação da onda senoidal que irá controlar o motor, e os seus respectivos pares RB1, RB3 e RB5 também são complementares. A figura 20 mostra esta defasagem de 120° alcançado na programação entre os sinais, o sinal amarelo retirado do pino RB0, o sinal azul retirado do pino RB2 e o sinal vermelho retirado do pino RB4.

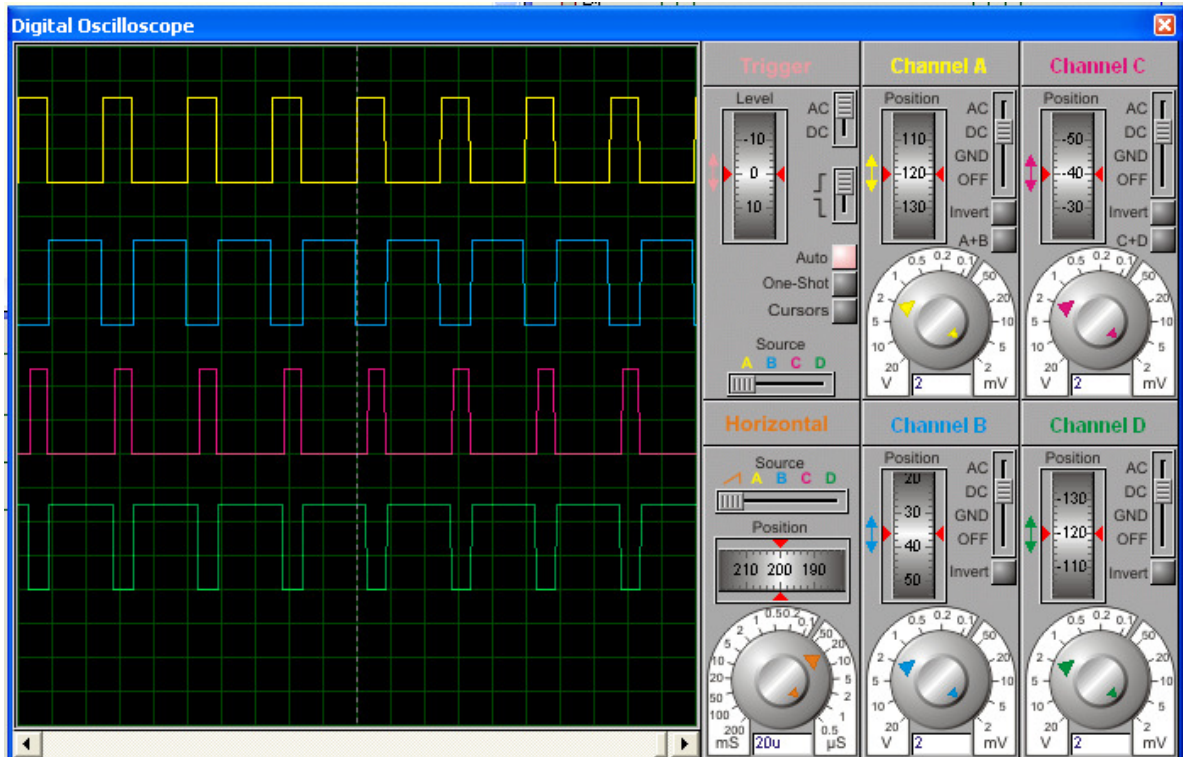


Figura 17. Formas de ondas obtidas por simulação.

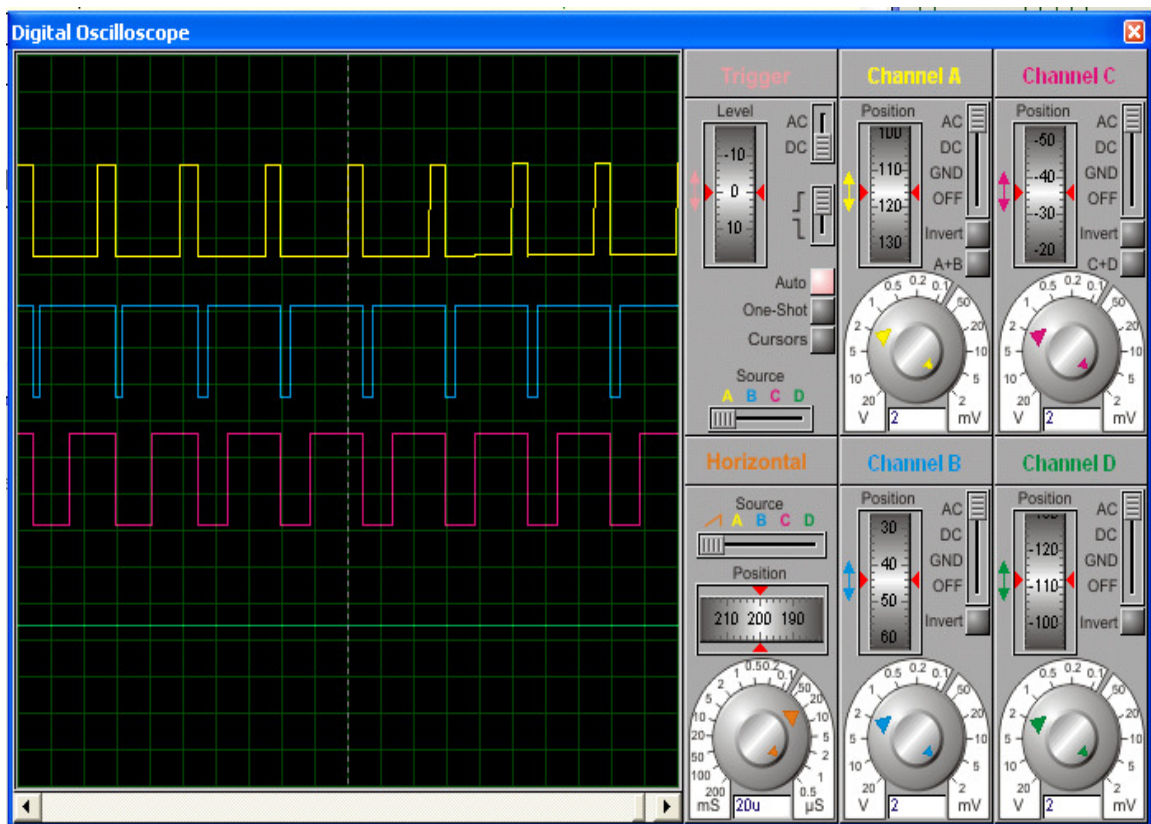


Figura 18. Formas de ondas defasadas em 120°

Aumentando a resolução do osciloscópio pode-se notar o tempo morto de proteção inserido no programa para proteger o chaveamento dos IGBTs de se cruzarem e com isso ocorrer um curto-circuito danificando o módulo de potência. O tempo conseguido foi de $1\mu\text{s}$ (micro-segundo), conforme o resultado esperado da programação e é mostrado na figura 20, as ondas foram sobrepostas para conseguir demonstrar melhor este tempo.

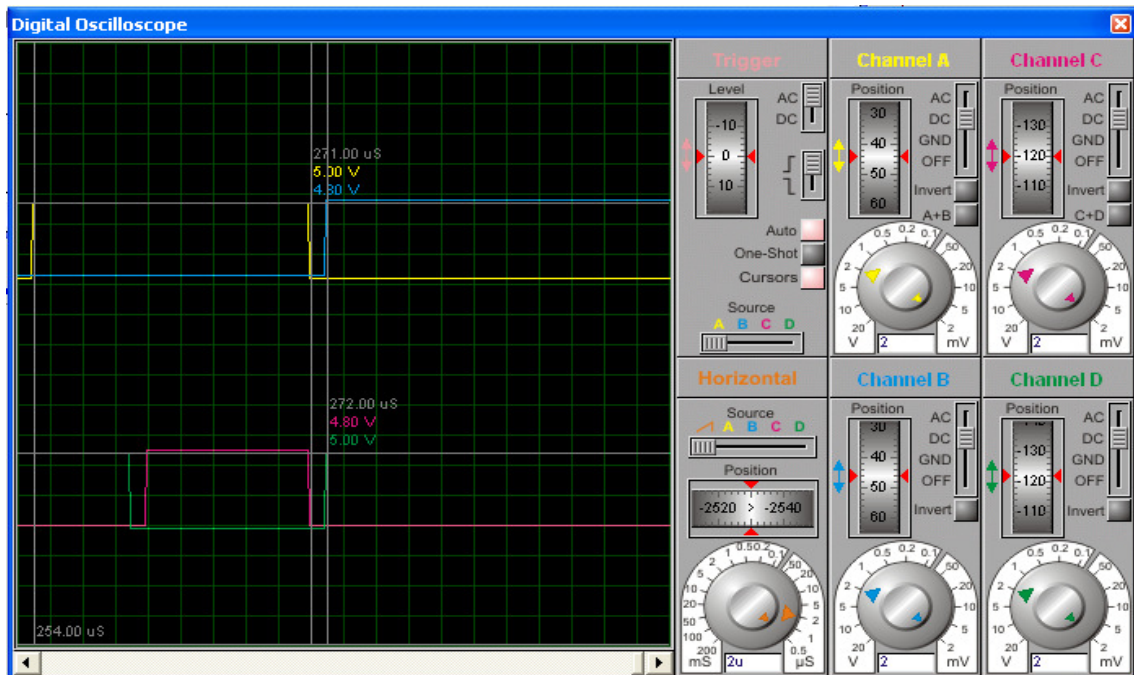


Figura 19. Tempo morto de segurança para o chaveamento entre as fases

Para melhor análise dos resultados foram utilizados os laboratórios da universidade no intuito de testar as saídas PWM diretamente no microcontrolador e, para isso, foi utilizado um osciloscópio digital Marca Tektronix modelo TDS 210. O primeiro teste realizado foi a forma dos sinais de saída, tensão pico a pico e frequência de chaveamento e os sinais apresentaram uma tensão de pico a pico entre $5,3V_{CC}$ e $6,16V_{CC}$, diferença esta vinda das pontas de provas, para conseguir chegar a esta conclusão foi feito a troca das pontas e em cada ponta utilizada um valor novo dentro destes limites apareciam. A frequência do sinal PWM medida foi de 20 kHz, e as formas de onda obtidas foram as mesmas conseguidas na simulação conforme mostra a figura 22.

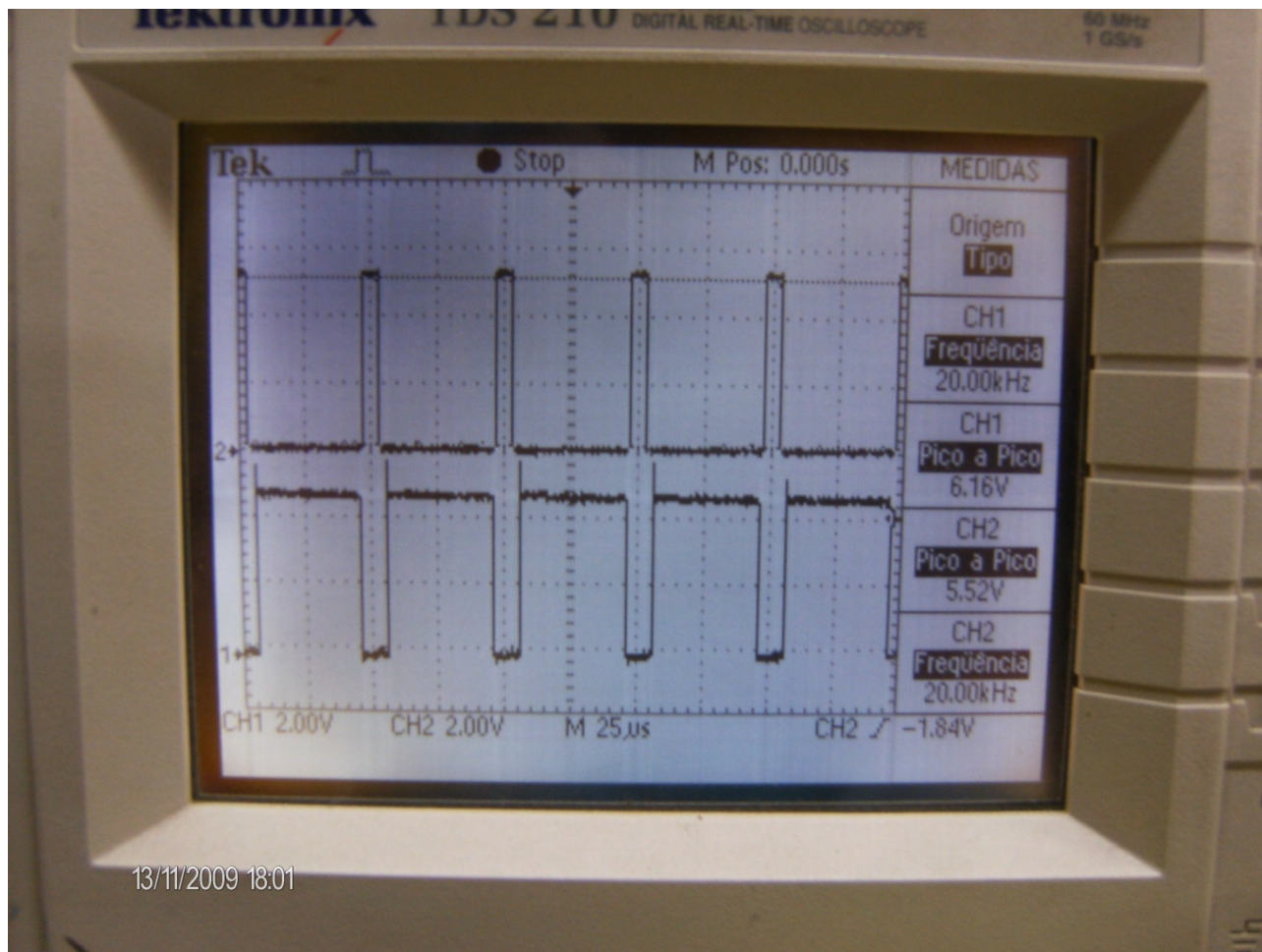


Figura 20. Foto mostrando a forma de onda PWM retiradas do protótipo.

Para esta foto os pinos analisados foram RB0 e seu complementar RB1, ela mostra também que as saídas estão trabalhando de forma complementar como no resultado com o simulador.

O outro teste realizado foi o de tempo morto entre os sinais PWM de saída, item esse extremamente importante para se evitar um curto-circuito no chaveamento dos IGBTs, e para isso foi diminuído o tempo de varredura para houvesse maior facilidade para a realização da medida do tempo morto no osciloscópio, alcançando-se $1\mu\text{s}$ (micro-segundo) de tempo morto entre os sinais, resultado este esperado pela programação. Esta medida é mostrada na figura23.

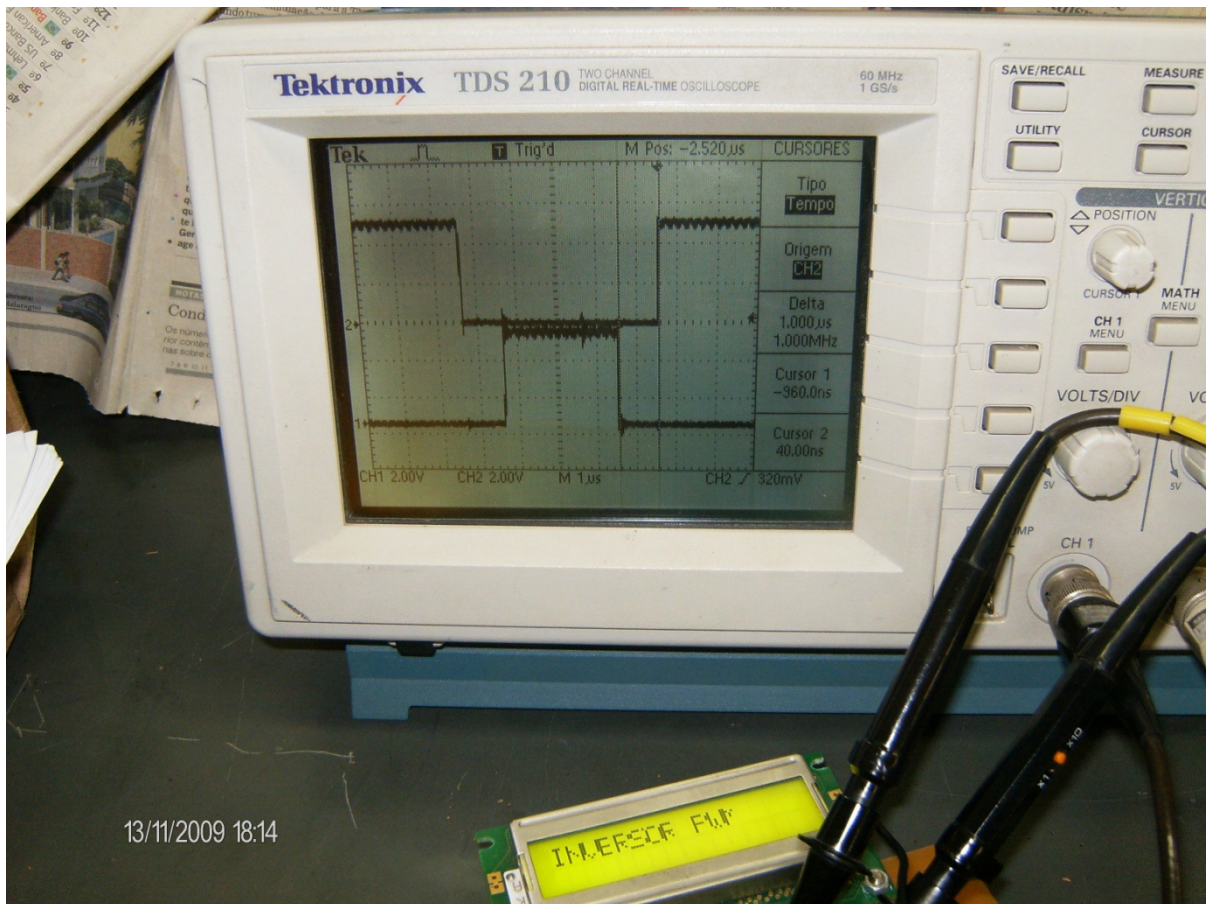


Figura 21. Tempo morto entre os sinais PWM medido no protótipo.

Estes testes foram realizados na placa de controle, para testar o programa comprovando a eficácia da lógica do programa. No protótipo da placa de potência foram realizados testes nas 3 fontes criadas medindo com o multímetro do laboratório de máquinas elétricas marca MINIPA® obtendo uma tensão de 5,2 VCC para fonte de 220 VAC/5VCC , uma tensão de 15,1 para a fonte de 220 VAC/ 15VCC. Para a fonte de 220VCA trifásico / 315 VCC foi utilizado um transformador variável devido a alta tensão e iniciado do 0 (zero) volts e aumentando gradativamente até os 220 VCA funcionando corretamente alcançando 316,8 VCC de saída.

4. CONCLUSÃO

Foi criado um programa por intermédio de microcontrolador utilizando como linguagem de programação assembly que gera seis formas de ondas senoidais moduláveis capaz de acionar um módulo de potência para controle de motor C.A..

Neste programa foi criado rotinas de inicialização para os registradores responsáveis pela geração do PWM e os registradores do canal analógico, definindo bem as formas de trabalho dos mesmos. Dentre estes registradores foi setado o tempo morto para a segurança de chaveamento entre as formas de onda.

Foram demonstrados os princípios de acionamentos do controle escalar utilizando a técnica PWM, e realizado testes no protótipo da placa de controle, verificando as formas de onda de saída. Verificou-se também a segurança deste acionamento medindo-se o tempo morto entre as ondas quadradas, dando segurança para o correto acionamento do módulo de potência.

Como o desenvolvimento do software foi complexo delegando muito tempo não foi possível realizar testes na placa de potência com o acionamento do motor, mas foram testadas as fontes criadas no projeto para alimentação de todo o inversor e funcionaram corretamente, obtendo os valores de tensão estabelecidos no projeto teórico.

Todos os testes executados no programa apresentaram valores que se aproximam dos valores teóricos validando o programa criado para acionamento do módulo I.G.B.T., e cabível de implementação.

Apesar da ênfase deste projeto ser no controle de motores de indução trifásico, o sistema é organizado em dois blocos construtivos (controle e potência) flexíveis, que poderão ser utilizados (em conjunto ou separados) na implementação de outros sistemas de condicionamento de energia elétrica.

Cabe ressaltar ainda que esse projeto se implementado na íntegra terá preço final em torno de 150 reais que é muito inferior aos similares vendidos no comércio em função da simplicidade dos circuitos desenvolvidos e também da pequena quantidade de componentes requeridos para sua montagem final.

5. SUGESTÃO PARA TRABALHOS FUTUROS

Para um próximo candidato que se interessar pelo assunto segue algumas sugestões para trabalhos futuros:

- Desenvolver o programa em linguagem C facilitando o trabalho de escrita no LCD.
- Desenvolver um hardware e conseqüentemente mudança no programa com capacidade de se trabalhar com realimentação no microcontrolador, ficando uma malha fechada, já que este modelo de microcontrolador oferece entradas de contagem rápida, é possível o acoplamento de um encoder no motor e com isso melhorar a dinâmica motor-inversor.
- Criar um circuito de medição de corrente e temperatura no módulo I.G.B.T. capaz de detectar uma sobrecorrente ou um aquecimento no módulo e interligar isto no microcontrolador para que ocorra o desligamento em caso de falha.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] YEDAMALE, Padmaraja. **Speed Control of 3-Phase Induction Motor Using PIC18 Microcontrollers**, Application Note , 16 jul. 2002. Disponível em: < <http://www.microchip.com> >. Acesso em 10 set. 2009.

[2] FRANCHI, Claiton M. **Inversores de frequência: Teoria e Aplicações**. 1. ed. São Paulo: Érica, 2008. 192p.

[3]. FITZGERALD, A. E.; KINGSLEY JR., Charles; KUSKO, Alexander. **Máquinas elétricas**. São Paulo: McGraw-Hill, 1975-1978. 623 p.

[4] IRF: **Datasheet IRAMS16UP60A**. Disponível em: < <http://www.irf.com/product-info/datasheets/data/irams16up60a.pdf> >. Acesso em 16 set. 2009.

[5] Microchip: **Datasheet microcontrolador PIC18F2431**. Disponível em: < <http://www.microchip.com/downloads/en/DeviceDoc/39616b.pdf> > Acesso em 16 set. 2009.

[6] BURROUGHS, Jon. **Controlling 3-Phase AC Induction Motors Using the PIC18F4431**. Application Note , 28 jul. 2004. Disponível em: < <http://www.microchip.com> >. Acesso em 10 set. 2009.

[7] MOHAN, UNDELAND, ROBBINS. **Power Electronics: Converters, Applications, and Design**. 3.ed. John Wiley & sons, 2002. 824p.

[8] MUHAMMAD. , H. Rashid. **Eletrônica de Potência**. 1. ed. São Paulo: Makron Books do Brasil, 1999.

APÊNDICE A – SOFTWARE MICROCONTROLADOR

```
#define OSCILLATOR    d'2000000'
;-----
#define TIMER0_PRESCALE    d'16'
;-----
#define SINE_TABLE_ENTRIESd'19'
;-----
SAMPLES_PER_CYCLE = (SINE_TABLE_ENTRIES-1)*d'2'
INSTRUCTION_CYCLE = (OSCILLATOR)/d'4'
FREQUENCY_SCALE = (INSTRUCTION_CYCLE/SAMPLES_PER_CYCLE)/(TIMER0_PRESCALE/4)
;-----
#define TIMER2_PRESCALE    d'01'
#define PWM_FREQUENCY    d'20000'
PR2_VALUE = (OSCILLATOR/(4*PWM_FREQUENCY*TIMER2_PRESCALE))-1
;-----
#define Fosc_by_2    b'000'
#define Fosc_by_8    b'001'
#define Fosc_by_32   b'010'
#define FRC          b'011'
#define Fosc_by_4    b'100'
#define Fosc_by_16   b'101'
#define Fosc_by_64   b'110'
#define ADC_CLOCK    Fosc_by_32
#define ADC_CHANNEL  d'0'
#define ADC_ON_BIT   b'1'
#define LEFT_JUSTIFIED
#define ADC_PORT_CONFIG    b'1110' ;
ADCON0_VALUE = ((ADC_CLOCK<<6)|(ADC_CHANNEL<<4)|(ADC_ON_BIT))
if((ADC_CLOCK==Fosc_by_2)||(ADC_CLOCK==Fosc_by_8)||(ADC_CLOCK==Fosc_by_32)||(ADC_CLOCK==FRC))
ifndef  ADCON1_VALUE = ((1<<7)|(ADC_PORT_CONFIG))
else  ADCON1_VALUE = ADC_PORT_CONFIG
endif
endif
if((ADC_CLOCK==Fosc_by_4)||(ADC_CLOCK==Fosc_by_16)||(ADC_CLOCK==Fosc_by_64))
ifndef  LEFT_JUSTIFIED ADCON1_VALUE = ((1<<7)|(1<<6)|(ADC_PORT_CONFIG))
else  ADCON1_VALUE = ( (1<<6)|(ADC_PORT_CONFIG))
endif
endif

;*****
;
;                               Projeto Inversor de Frequência
;*****
;
;                               Definições do Microcontrolador
;*****
;
;                               include    <p18f2431.inc>
;                               include    <3im_vf.inc>
;*****

__CONFIG__CONFIG1H, 0x02 ;_OSC_HS_1H & _FCMEN_OFF_1H& _IESO_OFF_1H
__CONFIG__CONFIG2L, 0x02 ;_PWRTEN_ON_2L & _BOREN_ON_2L & _BORV_20_2L
__CONFIG__CONFIG2H, 0x3E ;_WDTEN_OFF_2H
__CONFIG__CONFIG3L,0x3C ;;_PWMPIN_OFF_3L & _LPOL_LOW_3L & _HPOL_LOW_3L &
__CONFIG__CONFIG3H, 0x80 ;MCLEAR_ON
__CONFIG__CONFIG4L, 0xBA ;LVP_OFF, DEBUG_OFF
__CONFIG__CONFIG5L, 0x0F
```

```

__CONFIG__CONFIG5H, 0xC0
__CONFIG__CONFIG6L, 0x0F
__CONFIG__CONFIG6H, 0xE0
__CONFIG__CONFIG7L, 0x0F
__CONFIG__CONFIG7H, 0x40

;FLAGS bits
#define TIMER0_OV_FLAG      0
#define FLAG_FAULT         1
#define PARAM_DISPLAY      2
#define POSITION_BIT        3
#define OFFSET1_FLAG       4
#define OFFSET2_FLAG       5
#define OFFSET3_FLAG       6
#define MOTOR_DIRECTION    7

;FLAGS1 bits
#define DEBOUNCE           0
#define KEY_RS             1
#define KEY_FR             2
#define KEY_PRESSED       3
#define RUN_STOP          4
#define FWD_REV           5
#define FREQ_UPDATE       6
#define FEEDBACK_UPDATE   7

;Keys parameters
#define KEY_PORT           PORTA
#define RUN_STOP_KEY      3
#define FWD_REV_KEY      1
#define Desliga           4
#define Reverso           2
#define DEBOUNCE_COUNT    0xFF

;Delay parameters
#define DELAY_COUNT1      0xFF
#define DELAY_COUNT2      0xFF

;Parâmetros LCD
#define LCD               PORTC
#define RS                 0
#define RW                 1
#define E                  2

;LED parameters
#define LED_PORT          PORTB
#define RUN_STOP_LED      6
#define FWD_REV_LED       7

;Duty cycle limit definition, for 20KHz @20MHz, 2uS dead dtme
#define MINH_DUTY_CYCLE   0x00 ;
#define MINL_DUTY_CYCLE   0x3C ;
#define MAXL_DUTY_CYCLE   0xE0 ;
#define MAXH_DUTY_CYCLE   0x03 ;

;*****
;RAM Localização no Acesso bank, inicialização (Instrução 'res' reserva um espaço de memória para o nome)
;*****

```

UDATA_ACS		
Tempo_botao	res	1
Terminou	res	1
Rotaciona	res	1
loop	res	1
Contador	res	1
HIGH_NIBLE	res	1
LOW_NIBLE	res	1
RASCUNHO	res	1
Controle	res	1
APONTADOR	res	1
TABLE_OFFSET1	res	1
TABLE_OFFSET2	res	1
TABLE_OFFSET3	res	1
COUNTER	res	1
COUNTER1	res	1
COUNTER_SP	res	1
COUNTER_SP1	res	1
FLAGS	res	1
FLAGS1	res	1
FREQ_REF_H	res	1
FREQ_REF_L	res	1
FREQUENCY	res	1
TEMP	res	1
TEMP1	res	1
TEMP_LOCATION	res	2
DEBOUNCE_COUNTER	res	1
PDC0L_TEMP	res	1
PDC0H_TEMP	res	1
PDC1L_TEMP	res	1
PDC1H_TEMP	res	1
PDC2L_TEMP	res	1
PDC2H_TEMP	res	1
CURRENT	res	1
TEMPERATURE	res	1
VELOCITY_HIGH	res	1
VELOCITY_LOW	res	1
temp	res	1
temp1	res	1
botao	res	1
SINE_TABLE	res	0x14

```

;*****
;
;                               RESET E VETORES DE INTERRUPÇÃO
;*****
STARTUP code 0x00
    goto    Start                ;Reset Vector address

    CODE 0x08
    goto    ISR_HIGH            ;High priority ISR at 0x0008

PROG_LOW    CODE 0x018
    nop
    goto    ISR_LOW            ;Low priority ISR at 0x0018

;*****
;
;                               INICIALIZAÇÃO
;*****
PROG1 code

```

```

Start
    call pula
    GOTO MAIN_LOOP
Inversor
    BTFSC Terminou, 0x00
    RETURN
    MOVF APONTADOR,W
    ADDWF PCL
    RETLW A''
    RETLW A'I'
    RETLW A'N'
    RETLW A'V'
    RETLW A'E'
    RETLW A'R'
    RETLW A'S'
    RETLW A'O'
    RETLW A'R'
    RETLW A''
    RETLW A'P'
    RETLW A'W'
    RETLW A'M'
    RETLW 0x00

pula
    clrf botao
    clrf FREQUENCY
    clrf FLAGS
    clrf FLAGS1
    clrf APONTADOR
    clrf Contador

    call INIT_HSADC
    call INIT_PCPWM
    call INIT_TMR0
    call INIT_PORTC
    call COPY_TABLE_TO_RAM
    call INIT_MOTOR_START
    call INICIALIZA_LCD
    call MSG1
WAIT_HERE
    CALL KEY_CHECK ;aguarde até tecla seja pressionada antes de inicializar a motor
    BTFSS FLAGS1,KEY_PRESSED ;Se 1 bit 3 (KEY_PRESSED) do registro flags1 pula a próxima
linha.
    BRA WAIT_HERE
continua
    bsf PORTC,7
    CALL INIT_INTERRUPTS

;*****
;*****
;
; Loop Principal (Main Loop)
;*****
;*****
;*****
MAIN_LOOP

    btfss FLAGS,TIMER0_OV_FLAG ;Volta do Timer0 overflow? Se 1 pula próxima linha
    bra bypass ;Não
    call UPDATE_PWM_DUTYCYCLES ;SIM, carrega o PWM duty cycle com um novo valor.
    call UPDATE_TABLE_OFFSET ;Carrega 3 offsets

```

```

    bcf     FLAGS,TIMER0_OV_FLAG      ;Limpa o flag bypass
bypass
    btfsc  FLAGS1,FREQ_UPDATE        ;volta de A / D overflow? Se 1 não pula próxima linha
    call   CALCULATE_TIMER0_RELOAD ;
    btfss  ADCON0, GO                ;Se AD Conversion está completa
    bsf    ADCON0, GO                ;Então inicia uma nova conversão
    btfss  FLAGS1, FEEDBACK_UPDATE   ;Se 1 pula próxima linha
    call   CALCULATE_ACTUAL_SPEED
    call   KEY_CHECK                  ;(Checa mudança dos botões)

continua2
    call   PROCESS_KEY_PRESSED
    btfsc  FLTCONFIG,FLTAS
    bra    MAIN_LOOP

;*****
;
;                               ROTINAS DE SERVIÇOS DAS INTERRUPÇÕES
;*****
;
;Alta prioridade rotinas de serviços de interrupções
;Timer0 overflow são checados
;*****
ISR_HIGH
    btfsc  INTCON,TMR0IF              ;Interrupção Timer0 overflow ocorreu?
    bra    TIMER0_OVERFLOW            ;sim
    RETFIE FAST

TIMER0_OVERFLOW                    ;TMR0 overflow ISR (interrupção do TMR0)
    movff  FREQ_REF_H,TMR0H          ;Coloque o byte superior do comando de velocidade para TMR0H
    movff  FREQ_REF_L,TMR0L          ;Coloque o byte inferior do comando de velocidade para TMR0L
    bsf    FLAGS,TIMER0_OV_FLAG
    bcf    INTCON,TMR0IF            ;Limpa TMR0IF
    RETFIE FAST

ISR_LOW
    btfsc  PIR3,IC1IF                ;IC1 (QEI Modo de velocidade) interrupção ocorreu? Se 1 não pula linha
    bra    READ_SPEED_FEEDBACK        ;Sim

    btfsc  PIR1,ADIF                 ;HSADC interrupção ocorreu? Se 1 não pula proxima linha.
    bra    READ_ADC_RESULTS            ;Sim
    RETFIE FAST

READ_SPEED_FEEDBACK
    movff  VELRH, VELOCITY_HIGH ;copia VELRH:VELRL valor para cálculo da velocidade
    movff  VELRL, VELOCITY_LOW
    bsf    FLAGS1, FEEDBACK_UPDATE ;set flag para indicar velocidade
    nop
    bcf    PIR3,IC1IF                ;IC1IF flag é limpa para próxima interrupção
    RETFIE FAST

READ_ADC_RESULTS
    movff  ADRESH,FREQUENCY ;Primeiro valor é grupo A assinalada para AN0
    movlw  0x30                  ;Mínima frequência set para 5Hz (fator escalonado X4)
    cpfsgt FREQUENCY             ;Se frequência for inferior ou igual a 5Hz..
    movwf  FREQUENCY             ;set ela para 5Hz
    movlw  0xF0                  ;Limitando V/F para F= 60Hz (fator escalonado X4)
    cpfslt FREQUENCY             ;Se frequência for igual ou superior a 60Hz
    movwf  FREQUENCY             ;set para 60Hz
    bsf    FLAGS1, FREQ_UPDATE     ;set flag para indicar FREQUENCY está sendo carregado

```

```

    bcf    PIR1,ADIF          ;ADIF flag é limpo para próxima interrupção
    RETFIE FAST
;*****
; Rotinas do Display de LCD
;*****
MSG1
    BTFSC Terminou, 0x00
    RETURN
    CALL  Inversor
    CALL  VE_SE_FIM_MENSAGEM      ;VAI ENVIAR AO LCD EVER SE MSG TERMINOU
    BTFSS Controle,0x00          ;SE MSG TERMINOU (controle,0 =1)
    GOTO  MSG1
    CALL  LINHA

INICIALIZA_LCD
    BTFSC Terminou, 0x00
    RETURN
    MOVLW B'00011000'
    CALL  ENVIA_CMD
    CALL  Delay_LCD
    CALL  Delay_LCD
    CALL  Delay_LCD
    CALL  Delay_LCD
    MOVLW B'00011000'
    CALL  ENVIA_CMD
    CALL  Delay_LCD
    CALL  Delay_LCD
    CALL  Delay_LCD
    CALL  Delay_LCD
    MOVLW B'00011000'
    CALL  ENVIA_CMD
    CALL  Delay_LCD
    CALL  Delay_LCD
    MOVLW B'00010000'
    CALL  ENVIA_CMD
    CALL  Delay_LCD
    MOVLW B'00010000'
    CALL  ENVIA_CMD
    CALL  Delay_LCD
    MOVLW B'01000000'
    CALL  ENVIA_CMD
    CALL  Delay_LCD
    MOVLW B'00000000'
    CALL  ENVIA_CMD
    CALL  Delay_LCD
    MOVLW B'01100000'
    CALL  ENVIA_CMD
    CALL  Delay_LCD
    MOVLW B'00000000'
    CALL  ENVIA_CMD
    CALL  Delay_LCD
    MOVLW B'00001000'
    CALL  ENVIA_CMD
    CALL  Delay_LCD
    MOVLW B'00000000'
    CALL  ENVIA_CMD
    CALL  Delay_LCD
    MOVLW B'00110000'
    CALL  ENVIA_CMD

```

```

CALL Delay_LCD
RETURN

VE_SE_FIM_MENSAGEM    ;VE SE MSG TERMINOU (W=0) SENÃO, ENVIA O CARACTER P/ lcd
    BCF      Controle,0x00
    MOVWF   RASCUNHO
    MOVLW   0x00
    XORWF   RASCUNHO,W
    BTFSC   STATUS,Z
    GOTO    MSG_TERMINOU
    MOVF    RASCUNHO,W
    CALL    ENVIA_CAR
    INCF    APONTADOR,F
    INCF    APONTADOR,F
    RETURN

ENVIA_CAR
    MOVWF   LOW_NIBLE
    SWAPF   LOW_NIBLE,W
    MOVWF   HIGH_NIBLE
    MOVLW   H'0F'
    ANDWF   LOW_NIBLE,F
    ANDWF   HIGH_NIBLE,F
    MOVF    HIGH_NIBLE,W
    CALL    ENVIA_CAR2
    MOVF    LOW_NIBLE,W
    CALL    ENVIA_CAR2
    RETURN

ENVIA_CAR2
    CLRF    Rotaciona
    MOVWF   Rotaciona
    RLNCF   Rotaciona
    RLNCF   Rotaciona
    RLNCF   Rotaciona
    MOVF    Rotaciona,W
    MOVWF   PORTC
    BSF     LCD,RS
    BCF     LCD,RW
    BSF     LCD,E
    CALL    Delay_LCD
    BCF     LCD,E
    BCF     LCD,RS
    CALL    Delay_LCD
    RETURN

MSG_TERMINOU
    BSF     Controle,0x00
    CLRF    APONTADOR
    RETURN

ENVIA_CMD
    MOVWF   PORTC
    BCF     LCD,RS
    BCF     LCD,RW
    BSF     LCD,E
    CALL    Delay_LCD
    BCF     LCD,E
    RETURN

```

```

LINHA2
    MOVLW B'01100000' ;COLOCA ENDEREÇO DD RAM
    CALL ENVIA_CMD ;da SEGUNDA LINHA
    MOVLW B'00000000' ;COLOCA ENDEREÇO DD RAM ERA00000000
    CALL ENVIA_CMD
    BSF Terminou, 0X00
    RETURN

Delay_LCD
    MOVLW .1
    MOVWF loop

teste
    MOVLW .255
    MOVWF Contador

dec_contador
    decfsz Contador,F
    bra dec_contador
    decfsz loop,F
    bra teste
    clrf Contador
    clrf loop
    return
;*****
;*****
;
;               SUBROTINAS DRIVE MOTOR
;*****
;*****
;*****
;*****
;UPDATE_PWM_DUTYCYCLES
;Esta rotina irá atualizar o PWM duty no CCPx de acordo com a offset da tabela com 0-120-240 graus.
;Esta rotina escala o valor do PWM com base na tabela e na frequência para manter V / F
; constante.
;*****
UPDATE_PWM_DUTYCYCLES

    movf TABLE_OFFSET1,W ;Carregar o offset valor da tabela para Phase 1
    movf PLUSW0,W ;Usar offset para acessar o valor na tabela seno via indireto endereçamento
    mulwf FREQUENCY, W ;Table_value X Frequency
    movff PRODH,PDC0H_TEMP ;Copie os bits altos (high) do produto variável temporário PDC0H
    movff PRODL,PDC0L_TEMP ;Copie os bits baixos (Low) do produto variável temporário PDC0L

UPDATE_PWM2
    movf TABLE_OFFSET2,W ;Carregar o offset valor da tabela para Phase 2
    movf PLUSW0,W ;Usar offset para acessar o valor na tabela seno via indireto endereçamento
    mulwf FREQUENCY, W ;Table_value X Frequency
    movff PRODH,PDC1H_TEMP ;Copie os bits altos (high) do produto variável temporário PDC1H
    movff PRODL,PDC1L_TEMP ;Copie os bits baixos (Low) do produto variável temporário PDC1L

UPDATE_PWM3
    movf TABLE_OFFSET3,W ;Carregar o offset valor da tabela para Phase 3
    movf PLUSW0,W ;Usar offset para acessar o valor na tabela seno via indireto endereçamento
    mulwf FREQUENCY, W ;Table_value X Frequency
    movff PRODH,PDC2H_TEMP ;Copie os bits altos (high) do produto variável temporário PDC2H
    movff PRODL,PDC2L_TEMP ;Copie os bits baixos (Low) do produto variável temporário PDC2L

```



```

TRUNCATE_PWM123                                ;Truncar resultados de multiplicar superior a 10 bits
        bcf     STATUS,C                        ;devoluções inferior direita dois bits e justificando
        rlcfc  PDC0L_TEMP,F
        rlcfc  PDC0H_TEMP,F
        rlcfc  PDC0L_TEMP,F
        rlcfc  PDC0H_TEMP,F
        rlcfc  PDC0L_TEMP,W
        andlw  0x3
        movff  PDC0H_TEMP,PDC0L_TEMP
        movwf  PDC0H_TEMP
        bcf     STATUS,C
        rlcfc  PDC1L_TEMP,F
        rlcfc  PDC1H_TEMP,F
        rlcfc  PDC1L_TEMP,F
        rlcfc  PDC1H_TEMP,F
        rlcfc  PDC1L_TEMP,W
        andlw  0x3
        movff  PDC1H_TEMP,PDC1L_TEMP
        movwf  PDC1H_TEMP
        bcf     STATUS,C
        rlcfc  PDC2L_TEMP,F
        rlcfc  PDC2H_TEMP,F
        rlcfc  PDC2L_TEMP,F
        rlcfc  PDC2H_TEMP,F
        rlcfc  PDC2L_TEMP,W
        andlw  0x3
        movff  PDC2H_TEMP,PDC2L_TEMP
        movwf  PDC2H_TEMP

        call   CHECK_LIMITS

        bsf    PWMCON1,UDIS                    ;Desabilitar carregamentos para duty cycle e periodo
        movff  PDC0L_TEMP,PDC0L                ;Transferir temporários valores dentro registradores duty cycle
        movff  PDC0H_TEMP,PDC0H
        movff  PDC1L_TEMP,PDC1L
        movff  PDC1H_TEMP,PDC1H
        movff  PDC2L_TEMP,PDC2L
        movff  PDC2H_TEMP,PDC2H
        bcf    PWMCON1,UDIS                    ;Habilitar carregamentos para duty cycle períodos
        return

;*****
;
;          UPDATE_TABLE_OFFSET
;
;Está rotina Carrega os ponteiros offset para a tabela depois cada acesso.
;*****
UPDATE_TABLE_OFFSET
        btfss  FLAGS,OFFSET1_FLAG              ;Se set incremento na tabela
        bra    DECREMENT_OFFSET1
        movlw  (SINE_TABLE_ENTRIES-1)         ;Checar o último valor na tabela.
        cpfslt TABLE_OFFSET1
        bra    CLEAR_OFFSET1_FLAG
        incf  TABLE_OFFSET1,F                ;Incrementar offset1
        bra    UPDATE_OFFSET2

CLEAR_OFFSET1_FLAG
        bcf    FLAGS,OFFSET1_FLAG

DECREMENT_OFFSET1

```

```

    dcfsnz TABLE_OFFSET1,F           ;Decrementar offset1
    bsf    FLAGS,OFFSET1_FLAG

UPDATE_OFFSET2
    btfss  FLAGS,OFFSET2_FLAG         ;Se set incremento na tabela
    bra    DECREMENT_OFFSET2
    movlw  (SINE_TABLE_ENTRIES-1)     ;Checar o último valor na tabela
    cpfslt TABLE_OFFSET2
    bra    CLEAR_OFFSET2_FLAG
    incf   TABLE_OFFSET2,F           ;Incrementar offset2
    bra    UPDATE_OFFSET3

CLEAR_OFFSET2_FLAG
    bcf    FLAGS,OFFSET2_FLAG

DECREMENT_OFFSET2
    dcfsnz TABLE_OFFSET2,F           ;Decrementar offset2
    bsf    FLAGS,OFFSET2_FLAG

UPDATE_OFFSET3
    btfss  FLAGS,OFFSET3_FLAG         ;Se set incremento na tabela
    bra    DECREMENT_OFFSET3
    movlw  (SINE_TABLE_ENTRIES-1)     ;Checar o último valor na tabela
    cpfslt TABLE_OFFSET3
    bra    CLEAR_OFFSET3_FLAG
    incf   TABLE_OFFSET3,F           ;Incrementar offset3
    return

CLEAR_OFFSET3_FLAG
    bcf    FLAGS,OFFSET3_FLAG

DECREMENT_OFFSET3
    dcfsnz TABLE_OFFSET3,F           ;Decrementar offset3
    bsf    FLAGS,OFFSET3_FLAG
    return

;*****
; CALCULATE_TIMER0_RELOAD
;
;Esta rotina calcula o valor recarregamento do Timer0 baseado valor lido do ADC e o
;fator escalonamento calculado baseado no clock e números de entradas da tabela seno.
;Timer0 value = FFFF - (FREQUENCY_SCALE/Frequency) Frequ = (adc result)
;*****
CALCULATE_TIMER0_RELOAD

    bcf    FLAGS1,FREQ_UPDATE
    clrf   TEMP
    clrf   TEMP1
    movlw  HIGH(FREQUENCY_SCALE)      ;FREQUENCY_SCALE/Frequency
    movwf  TEMP_LOCATION              ;16 bit por 8 bit divisão
    movlw  LOW(FREQUENCY_SCALE)       ;
    movwf  TEMP_LOCATION+1

continue_subtraction

    movf   FREQUENCY,W
    btfsc  STATUS,Z ; Resultado da última operação aritmética igual a zero bit Z=1
    return
    bsf    STATUS,C                   ; Seta 1 no estouro do Carry.

```

```

    movf   FREQUENCY,W
    subwfb TEMP_LOCATION+1,F
    clrf   WREG
    subwfb TEMP_LOCATION,F
    btfss  STATUS,C
    goto   keep_result_in_rpm
    incf   TEMP,F
    btfsc  STATUS,C      ;Resultado da divisão está armazenado no TEMP&TEMP1
    incf   TEMP1,F
    goto   continue_subtraction

keep_result_in_rpm      ;Timer0 value = FFFF-Timer0
    bsf    STATUS,C
    movlw  0xFF
    subwfb TEMP,F
    subwfb TEMP1,F      ;O recarregamento do valor Timer0 está armazenado na
    movff  TEMP1,FREQ_REF_H ;FREQ_REF_H & FREQ_REF_L
    movff  TEMP,FREQ_REF_L  ;Estes valores serão carregados para
    return

;*****
;
;           CHECK_LIMIT ROUTINE
;
;   Para frequência < 60Hz, duty cycle será inferior a MAX_DUTY_CYCLE (4 x PTPER) devido a
;   seleção dos valores tabela seno
;
;   ainda é necessária para garantir que a PDC é maior ou igual a MINL_DUTY_CYCLE (3 x deadtime)
;*****
CHECK_LIMITS

CHK_PWM0_MIN      ;Testa para ver se PDC0H:PDC0L < 0:MINL_DUTY_CYCLE
    movf   PDC0H_TEMP, F ;Primeiro, PDC0H = 0 ?
    bnz    CHK_PWM1_MIN ;Se não, então PDC não pode ser menor que minimo, checar proximo
    movlw  MINL_DUTY_CYCLE ;Segundo, PDC0L > MINL_DUTY_CYCLE?
    cpfsgt PDC0L_TEMP ;Se sim, então PDC não pode ser menor que minimo, checar proximo
    movwf  PDC0L_TEMP ;Se não, torná-lo o valor mínimo

CHK_PWM1_MIN      ;Testa para ver se PDC1H:PDC1L < 0:MINL_DUTY_CYCLE
    movf   PDC1H_TEMP, F ;Primeiro, PDC1H = 0 ?
    bnz    CHK_PWM2_MIN ;Se não, então PDC não pode ser menor que minimo, checar proximo
    movlw  MINL_DUTY_CYCLE ;Segundo, PDC1L > MINL_DUTY_CYCLE?
    cpfsgt PDC1L_TEMP ;Se sim, então PDC não pode ser menor que minimo, checar proximo valor
    movwf  PDC1L_TEMP ;Se não, torná-lo o valor mínimo

CHK_PWM2_MIN      ;Testa para ver se PDC2H:PDC2L < 0:MINL_DUTY_CYCLE
    movf   PDC2H_TEMP, F ;Primeiro, PDC2H = 0 ?
    bnz    DONE_CHECK_LIMITS ;Se não, então PDC não pode ser menor que minimo,
    movlw  MINL_DUTY_CYCLE ;Segundo, PDC2L > MINL_DUTY_CYCLE?
    cpfsgt PDC2L_TEMP ;Se sim, então PDC não pode ser menor que minimo, checar proximo valor
    movwf  PDC2L_TEMP ;Se não, torná-lo o valor mínimo

DONE_CHECK_LIMITS
    return

;*****
;Esta rotina para o motor pela condução do PWMs para 0% duty cycle
;*****
STOP_MOTOR
    bcf    PIE1,ADIE
    bcf    INTCON,TMR0IE
    clrf   OVDCOND ;Ligar saídas overrides antes setar duty cycle to zero

```

```

        BCF    PORTC,7
        clrf  TABLE_OFFSET1
        clrf  TABLE_OFFSET2
        clrf  TABLE_OFFSET3
        bcf   FLAGS, TIMER0_OV_FLAG
        clrf  Terminou
        return

;*****
;Esta rotina starta motor de anteriores paradas com parâmetros do motor inicializados
;*****
RUN_MOTOR_AGAIN
        bsf   FLAGS1,RUN_STOP
        bcf   FLAGS,FLAG_FAULT
        bsf   PIE1,ADIE
        call  INIT_MOTOR_START
        call  UPDATE_PWM_DUTYCYCLES ;Sim, carrega o PWM duty cycle com novos valores
        call  UPDATE_TABLE_OFFSET   ;Carrega 3 offsets
        bsf   INTCON,TMR0IE
        movlw b'11111111'           ;Depois duty cycles contêm valores não nulos
        movwf OVDCOND               ;OVDCOND esta configurado tal que não há saída override
        return
        FLAGS1,FEEDBACK_UPDATE
        return

;*****
;
;                               SUBROTINAS DE TESTE DOS BOTÕES
;*****
KEY_CHECK
        btfss KEY_PORT,RUN_STOP_KEY
        goto  CHECK_FWD_REV_KEY
        btfsc FLAGS1,DEBOUNCE
        return
        call  KEY_DEBOUNCE
        btfss FLAGS1,DEBOUNCE
        return
        bsf   FLAGS1,KEY_RS
        return

CHECK_FWD_REV_KEY
        goto  SET_KEYS
        btfsc FLAGS1,DEBOUNCE
        return
        call  KEY_DEBOUNCE
        btfss FLAGS1,DEBOUNCE
        return
        bsf   FLAGS1,KEY_FR
        return

SET_KEYS
        btfss FLAGS1,DEBOUNCE
        return
        bcf   FLAGS1,DEBOUNCE
        bsf   FLAGS1,KEY_PRESSED
        btfss FLAGS1,KEY_RS
        bra   ITS_FWD_REV
        btg   FLAGS1,RUN_STOP
        return
ITS_FWD_REV

```

```

        btg          FLAGS1,FWD_REV
        return

;*****
KEY_DEBOUNCE
        decfsz     DEBOUNCE_COUNTER,F
        return
        bsf        FLAGS1,DEBOUNCE
        movlw     DEBOUNCE_COUNT
        movwf     DEBOUNCE_COUNTER
        return
;*****
PROCESS_KEY_PRESSED
        btfss     FLAGS1,KEY_PRESSED
        return
        btfss     FLAGS1,KEY_RS
        goto      CHECK_FWD_REV
        btfss     FLAGS1,RUN_STOP
        goto      STOP_MOTOR_NOW
        call      RUN_MOTOR_AGAIN
        bcf        FLAGS1,KEY_PRESSED
        bcf        FLAGS1,KEY_RS

        bsf        LED_PORT,RUN_STOP_LED
        return

STOP_MOTOR_NOW
        call      STOP_MOTOR
        bcf        FLAGS1,KEY_PRESSED
        bcf        FLAGS1,KEY_RS
        bcf        LED_PORT,RUN_STOP_LED
        return

;*****
; Delay routine.
;*****
DELAY
        movlw     DELAY_COUNT1
        movwf     COUNTER
dec_count
        movlw     DELAY_COUNT2
        movwf     COUNTER1
dec_count1
        decfsz     COUNTER1,F
        bra        dec_count1
        decfsz     COUNTER,F
        bra        dec_count
        clrf      COUNTER
        clrf      COUNTER1
        return

;*****
;
;                               INICIALIZAÇÃO DAS SUBROUTINES
;*****
INIT_HSADC
        movlw     b'00000000'
        movwf     ADCON1

```

```

movlw b'00110010'
movwf ADCON2

movlw b'01000000'
movwf ADCON3

movlw b'00100000'
movwf ADCHS

movlw b'00000001'
movwf ANSEL0

movlw b'11111111'
movwf TRISA

movlw b'00000101'
movwf ADCON0
return

```

INIT_PCPWM

```

movlw b'00000000'
movwf PTCON0

movlw 0xF9
movwf PTPERL

movlw 0x00
movwf PTPERH

movlw b'01000000'
movwf PWMCON0

movlw b'00000001'
movwf PWMCON1

movlw b'00001010'
movwf DTCON
;1us de tempo morto

movlw b'11111111'
movwf OVDCOND

movlw b'00000000'
movwf OVDCONS

movlw b'00000000'
movwf FLTCONFIG

;movlw 0x00
;movwf SEVTCMPL
;movlw 0x00
;movwf SEVTCMPH

bsf PTCON1, PTEN
return

```

INIT_PORTC

```

movlw b'00000000'

```

```

        movwf TRISC
        return

INIT_TMR0
    movlw b'10000100'
    movwf T0CON
    movlw 0xF8
    movwf TMR0H
    movlw 0x5E
    movwf TMR0L
    return

INIT_INTERRUPTS
    bsf    INTCON,TMR0IE
    bsf    INTCON2,TMR0IP
    bsf    PIE1,ADIE
    bcf    IPR1,ADIP
    bsf    PIE3,IC1IE
    bcf    IPR3,IC1IP
    movlw b'10010011'
    movwf RCON
    bsf    INTCON,GIEL
    bsf    INTCON,GIEH
    return

INIT_MOTOR_START
    movlw 0x09
    movwf TABLE_OFFSET1
    bsf    FLAGS,OFFSET1_FLAG
    btfss  FLAGS,MOTOR_DIRECTION
    bra   INIT_MOTOR_START_REV
    movlw 0x03
    movwf TABLE_OFFSET2
    bcf    FLAGS,OFFSET2_FLAG
    movlw 0x0F
    movwf TABLE_OFFSET3
    bcf    FLAGS,OFFSET3_FLAG
    bsf    PORTC,0
    bra   CONT_INIT_MOT

INIT_MOTOR_START_REV
    movlw 0x0F
    movwf TABLE_OFFSET2
    bcf    FLAGS,OFFSET2_FLAG
    movlw 0x03
    movwf TABLE_OFFSET3
    bcf    FLAGS,OFFSET3_FLAG

CONT_INIT_MOT
    movlw 0x30
    movwf FREQUENCY
    movlw 0xFD
    movwf  FREQ_REF_H
    movwf  TMR0H
    movlw 0x2C
    movwf  TMR0L
    movwf  FREQ_REF_L
    bsf    FLAGS,TIMER0_OV_FLAG
    return

```

```

COPY_TABLE_TO_RAM
    movlw UPPER sine_table
    movwf TBLPTRU
    movlw HIGH sine_table
    movwf TBLPTRH
    movlw LOW sine_table
    movwf TBLPTRL
    movlw LOW(SINE_TABLE)
    movwf FSR0L
    movlw HIGH(SINE_TABLE)
    movwf FSR0H
    movlw 0x14
    movwf TEMP
COPY_AGAIN
    TBLRD*+
    movff TABLAT,POSTINC0
    decfsz TEMP,F
    bra COPY_AGAIN
    movlw LOW(SINE_TABLE)
    movwf FSR0L
    movlw HIGH(SINE_TABLE)
    movwf FSR0H
    return

```

```

;*****
;
;                                TABELA SENO
;*****

```

```

TABLE code 0x0600
db0x00,0x02,0x08,0x11,0x1E,0x2E,0x40,0x54,0x69,0x80,0x96,0xAB,0xBF,0xD1,0xE1,0xEE,0xF7,0xFD,
0xFF

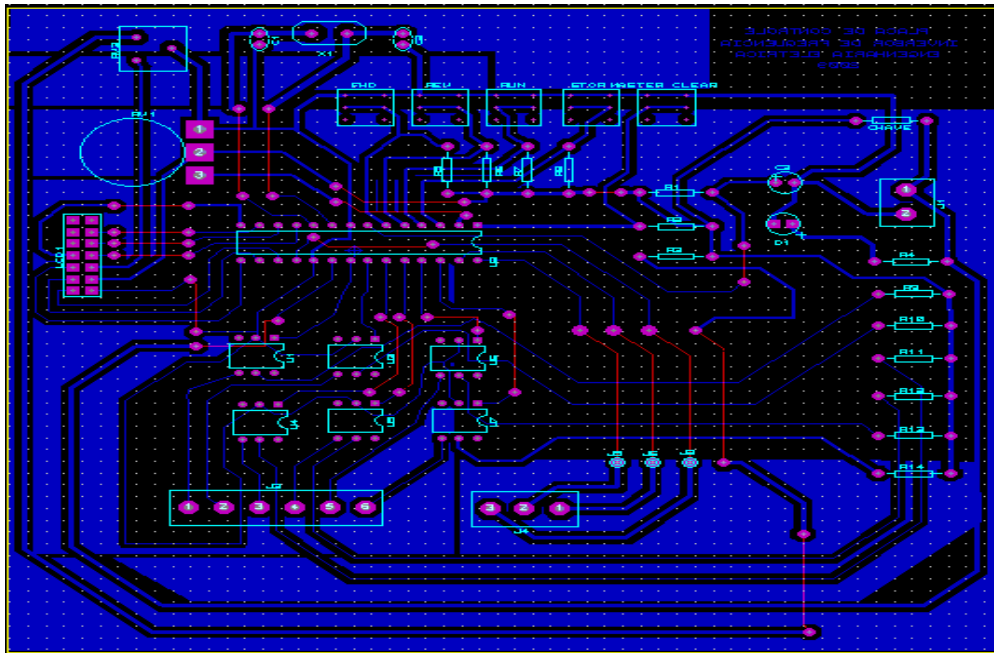
```

```

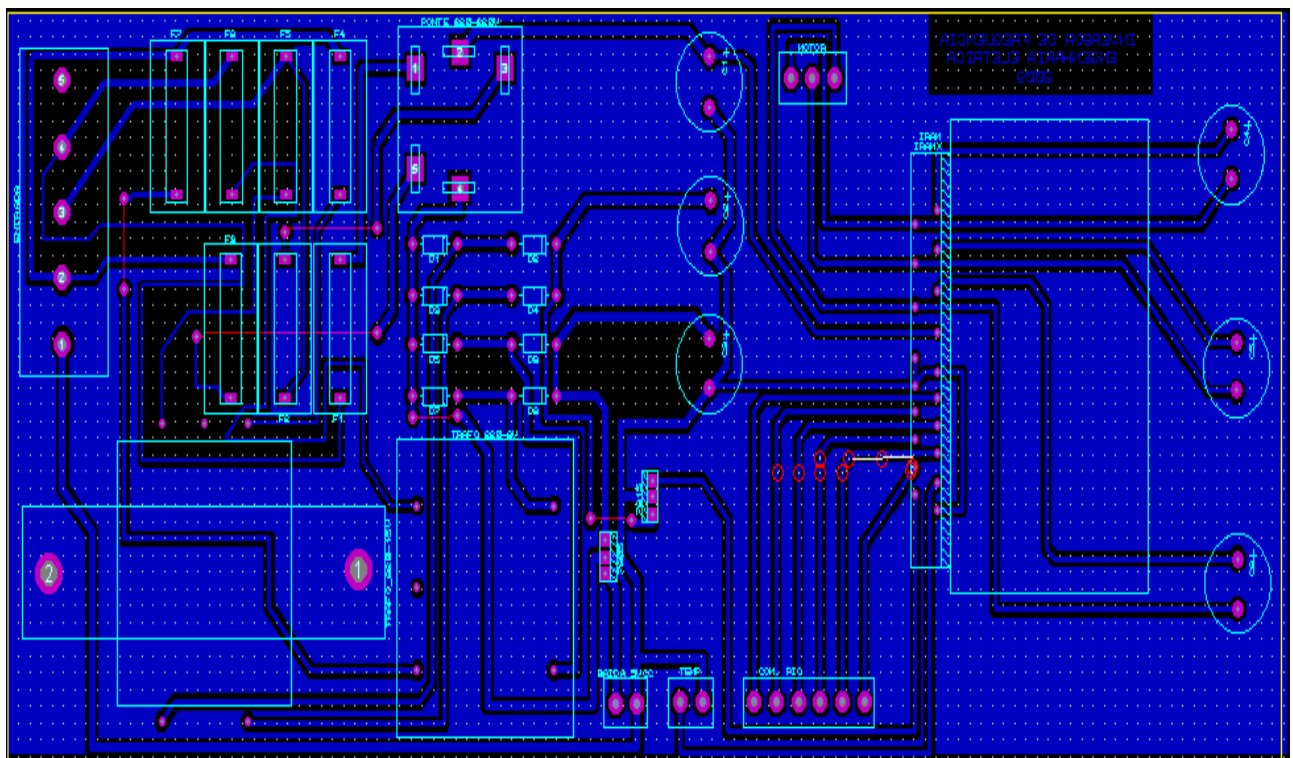
END

```


APÊNDICE B –CIRCUITO IMPRESSO PLACA CONTROLE



APÊNDICE C –CIRCUITO IMPRESSO PLACA POTÊNCIA



ANEXO I – ESTRUTURA INTERNA DO MÓDULO DE POTÊNCIA

IRAMX16UP60A

International
IGR Rectifier

Internal Electrical Schematic - IRAMX16UP60A

