



UNIVERSIDADE
ESTADUAL de LONDRINA

THIAGO BITTENCOURT DE MORAES MATEUS LEITE

**INVERSOR PUSH-PULL MICROCONTROLADO
MODULADO EM SPWM**

LONDRINA

2011

THIAGO BITTENCOURT DE MORAES MATEUS LEITE

**INVERSOR PUSH-PULL MICROCONTROLADO
MODULADO EM SPWM**

Trabalho de conclusão de curso submetido à
Universidade Estadual de Londrina como parte
dos requisitos para a obtenção do grau de
Engenheiro Eletricista.

Orientador: Prof. Aziz Elias Demian Junior

LONDRINA - PR

2011.

THIAGO BITTENCOURT DE MORAES MATEUS LEITE

**INVERSOR PUSH-PULL MICROCONTROLADO
MODULADO EM SPWM**

‘Este trabalho foi julgado adequado para a conclusão do curso de engenharia elétrica e aprovado em sua forma final pela Coordenação do Curso de Engenharia Elétrica da Universidade Estadual de Londrina’

BANCA EXAMINADORA

Orientador: Prof. Dr. Aziz Elias Demian Junior
Orientador

Prof. Msc. André Luiz Batista Ferreira

Prof. Dr. Marcio Roberto Covacic

Londrina, ____ de _____ de ____.

*A minha família pelo apoio incondicional durante toda a jornada.
Aos meus amigos de curso, com os quais esperei ansiosamente por este momento.*

.

AGRADECIMENTOS

Agradeço primeiramente a Deus pela força, proteção e companhia durante esta minha caminhada. Aos meus familiares que suportaram pacientemente meu humor em vésperas de prova e não pouparam esforços para que eu alcançasse meus objetivos e também a minha gatinha Laeti, que sempre me deu ânimo durante os momentos mais difíceis.

Ao professor Aziz, pelo companheirismo durante todo ano, proporcionando e pela forma de orientação, que de maneira descontraída incentivou-me a buscar soluções para os corriqueiros problemas durante meu projeto. Agradeço também ao professor André, por sua integral disposição em sanar dúvidas nebulosas e pelo acompanhamento do projeto em todas suas fases, contribuindo de forma essencial com sua experiência em circuitos de potência.

Agradeço a 3E-UEL que me fez vivenciar situações de contínuo crescimento e aprendizado, que mostrou, acima de tudo, quão grande é a força de pessoas unidas por um único ideal e o quanto isso pode mudar nós mesmos e o mundo ao nosso redor.

Sou grato pelos grandes amigos que fiz e pelas coisas que vivemos do decorrer destes seis anos. Amigos estes como Sr. Eng. Renan Sabão, Gaba, Taks, Mazzo e principalmente, os amigos do batidão Fabio CS e Cambs, com os quais foi aplicada uma metodologia de estudo pouco eficiente, mas totalmente eficaz que fez com que terminasse nosso último ano de matérias sem a presença em nenhum exame, quebrando uma sequência de quatro longos anos pendurados.

E finalmente agradeço a mim mesmo, por ter perseverado no curso, enquanto vi tantos outros desistirem para usufruir de “uma vida melhor”.

“Na teoria a prática é outra”.
(Autor Desconhecido)

LEITE, ThiagoBittencourt **Inversor Push-Pull Microcontrolado Modulado em SPWM**
2011. 90. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) –
Universidade Estadual de Londrina, Londrina, 2011.

RESUMO

O presente trabalho objetiva apresentar o projeto de um inversor utilizado para gerar tensão alternada 127VAC eficaz em 60Hz a partir de fonte de tensão contínua 12VDC. A configuração utilizada baseia-se em um circuito que forneça uma tensão de saída equivalente ao fornecimento de energia elétrica doméstico para cargas de até 216 Watts. Foi adotada a topologia *Push-Pull* atuando como inversor na qual o próprio conversor se encarrega também da conversão CC/CA. Para o sistema de controle do *Push-Pull* foi utilizado um microcontrolador da família 16F programado para gerar ondas SPWM em cada módulo PWM, chaveando em 60Hz entre si. O sinal SPWM foi projetado para que a largura do pulso seja incrementada a cada 3 graus da onda senoidal, sendo uma chave responsável pela parte positiva da senóide e outra pela parte negativa. Todo projeto foi simulado em softwares específicos para que o sistema fosse validado.

Palavras-chave: 1. Push-Pull 2. Inversor 3. Microcontrolador.

LEITE, Thiago Bittencourt **Microcontrolled Push-Pull Inverter with SPWM Modulation**
2011. 90. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) –
Universidade Estadual de Londrina, Londrina, 2011.

ABSTRACT

This study presents the design of an inverter used to generate AC voltage of 127 VAC to 60 Hz from a source of 12 VDC. The configuration used is based on a circuit that provides an output voltage equivalent to the domestic electricity supply for loads up to 216 Watts. It was adopted the Push-Pull topology acting as an inverter in which the converter itself also takes care of converting DC / AC. For the control of the push-pull system it was used a 16F family microcontroller programmed to generate SPWM waves in each PWM module, switching between them at 60 Hz. The SPWM signal is designed so that the pulse width is increased at steps of 3 degrees of the sine wave, a transistor being responsible for the positive part of sine wave and the other downside. All project has been simulated in software specific to the system were validated.

Key words: 1. Push-Pull2. Inverter 3. Microcontroller.

Lista de Ilustrações

Figura 1 - Conversor Push-Pull convencional.....	17
Figura 2 - Transistor T1 saturado e T2 em corte.	18
Figura 3 - Transistores T1 e T2 em corte.	19
Figura 4 - T1 em corte e T2 saturado.	19
Figura 5 - T1 e T2 em corte.....	20
Figura 6 - Curva B-H do conversor Push-Pull	20
Figura 7 - Formas de onda do conversor <i>Push-Pull</i>	21
Figura 8 - <i>Push-Pull</i> inversor.	22
Figura 9 - Formas de onda do <i>Push-Pull</i> inversor.....	23
Figura 10 – Correntes de magnetização e desmagnetização e tensão dos transistores.....	24
Figura 11 - Circuito de ilustração dos modos de operação do TBJ.....	27
Figura 12 - Regiões de operação do TBJ.....	28
Figura 13 - Variação de potência através de sinal PWM.	29
Figura 14 - Onda senoidal através de um sinal PWM filtrado.	30
Figura 15 - Geração de um sinal SPWM.....	31
Figura 16 - Potência transmitida por um sinal SPWM.....	31
Figura 17 - Agrupamento de funções de um microcontrolador.....	32
Figura 18 - Registradores de tempo.....	33
Figura 19 - Registradores de tempo com pré-escalas.	34
Figura 20 - Conceito da interrupção por tempo.....	34
Figura 21 - 1) Armazenamento do endereço atual do programa; 2) Acesso a rotinas de interrupção e 3) Retorno à execução do programa.	35
Figura 22 - Diagrama de entrada e saída de dados.	36
Figura 23 - Visão geral do PIC 16F874A.....	37
Figura 24 - Esquemático do módulo Timer 0.....	38
Figura 25 - Registrador OPTION_REG.	38
Figura 26 - Resistores <i>pull-ups</i>	40
Figura 27 - Lógica de funcionamento do módulo Timer 2.....	41
Figura 28 - Registrador T2CON.....	41
Figura 29 - Registrador CCPR1	43
Figura 30 - Lógica de funcionamento do módulo CCP1.....	44
Figura 31 - Parâmetros configuráveis para configuração de sinal de saída CCP.	44

Figura 32 - Registrador CCP1CON.....	45
Figura 33 - Configuração meia-ponte.....	46
Figura 34 - Registradores A/D. a) Justificação à direita; b) Justificação à esquerda.	47
Figura 35 - Registrador ADCON0	47
Figura 36 - Registrador ADCON1	48
Figura 37 - Diagrama de blocos do módulo A/D.	49
Figura 38 - Retificador e detector de pico	56
Figura 39 - Onda senoidal a 60 Hz.	57
Figura 40 - Etapas periódicas do sinal senoidal.	57
Figura 41 - Circuito ressonante.	59
Figura 42 - Pinos de saída dos módulos CCP.....	61
Figura 43–Diagrama de fluxo do código gerador de SPWM.	71
Figura 44 - Circuito de potência.....	73
Figura 45 - Tensão na base dos transistores.	74
Figura 46 - Tensão no transistor.....	74
Figura 47 - Tensão de saída quase-quadrada.....	75
Figura 48 - Corrente no primário com carga de indutância 10 μ H.	75
Figura 49 - Corrente no primário com carga de indutância 10mH.....	76
Figura 50 - Saída do divisor de tensão.	77
Figura 51 - Comparação entre a saída do divisor de tensão e tensão de pico retificada.	77
Figura 52 - Acionamento das saídas CCP1 e CCP2.....	78
Figura 53 - Variação senoidal da largura de pulso do sinal.	79
Figura 54 - Amostra do sinal de saída de CCP1, com escala de 10 μ s por divisão.....	80
Figura 55 - Tempo morto entre chaves.....	80
Figura 56 - Sinal de saída filtrado.	81
Figura 57 - Circuito de potência simulado no aplicativo OrCAD 16.3.....	93
Figura 58 - Circuito de controle simulado no aplicativo Proteus 7.8.....	94

Lista de Tabelas

Tabela 1 - Timers de referência do módulo CCP.	42
Tabela 2 - Interação entre modos do módulo CCP.....	43
Tabela 3- Parâmetros de projeto do circuito de potência do inversor.	51
Tabela 4 - Parâmetros do Transformador	53
Tabela 5 - Dados do período da onda.	57
Tabela 6 - Arquivo de cabeçalho (<i>header</i>).	60
Tabela 7 - Configuração do Timer 2.	63
Tabela 8 - Vetor de largura de pulso SPWM.....	66
Tabela 9 - Matriz de vetores SPWM.	70

Lista de Abreviaturas e Siglas

CC	Corrente contínua
CA	Corrente alternada
PWM	<i>Pulse Width Modulation</i> (Modulação por Largura de Pulso)
SPWM	<i>Sinusoidal Pulse Width Modulation</i> (Modulação por Largura de Pulso Senoidal)
RMS	<i>Root Mean Square</i> (Valor Quadrático Médio)
RAM	<i>Random Access Memory</i> (Memória de Acesso Aleatório)
ROM	<i>Read Only Memory</i> (Memória de Apenas Leitura)
EEPROM	<i>Electrically-Erasable Programmable Read-Only Memory</i> (Programável Eletricamente Apagável Memória de Apenas Leitura)
A/D	<i>Analog/Digital</i> (Analógico/Digital)
I/O	<i>In/Out</i> (Entrada/Saída)
CPU	<i>Central Processor Unit</i> (Unidade Central de Processamento)
I2C	<i>Inter Integrated Circuit</i> (Circuito Inter-Integrado)
SPI	<i>Serial Peripheral Interface Bus</i> (Barramento de Interface Periférica Serial)
AWG	<i>American Wire Gauge</i> (Bitola de Fio Americano)

Sumário

Sumário	xii
INTRODUÇÃO.....	14
FUNDAMENTAÇÃO TEÓRICA.....	16
2.1 <i>Topologia Push-Pull</i>	16
2.1.1 Topologia Convencional	17
2.1.2 Topologia Push-Pull Inversor.....	21
2.3 <i>Equacionamento do Push-Pull Inversor</i>	23
2.3.1 Transformador	24
2.3.2 Transistores	27
2.4 <i>Controle</i>	29
2.5 <i>Microcontroladores</i>	32
2.6 <i>Microcontrolador 16f874a</i>	36
2.6.1 Timers.....	37
2.6.2 Módulo PWM.....	42
2.6.3 Conversor A/D	46
2.6.4 Compilador.....	49
PROJETO.....	51
3.1 <i>Push-Pull</i>	51
3.1.1 Relação de Transformação	52
3.1.2 Transformador	52
3.1.3 Transistores	55
3.2 <i>Controle SPWM</i>	56
3.2.1 A Onda Senoidal 60Hz.....	56
3.2.2 Requisitos de Chaveamento	58
3.2.3 Programação do Pic.....	59
ANÁLISE DOS RESULTADOS	72
CONSIDERAÇÕES FINAIS.....	83
BIBLIOGRAFIA	84
ANEXO A.....	85

ANEXO B86

ANEXO C87

ANEXO D88

ANEXO E93

ANEXO F94

CAPÍTULO 1

INTRODUÇÃO

É notável o esforço aplicado em pesquisas científicas que buscam a criação e melhoria de sistemas energéticos renováveis e não poluentes visto que há um constante aumento na demanda por energia em todo o mundo. Energia esta que é normalmente obtida a custos de demasiada exploração de recursos naturais gerando um direto impacto negativo ao ambiente.

Neste cenário se faz presente a eletrônica de potência, que passa a atuar como aliada na busca por novas fontes de geração de energia que atendam plenamente as necessidades atuais, principalmente em locais onde existem deficiências na distribuição, como zonas rurais ou em locais onde a confiabilidade é fundamental, como é o caso de hospitais, bancos ou sistemas de banco de dados.

Este trabalho trata do projeto de uma fonte de tensão alternada de 216 *Watts* que pode ser utilizada em eletrodomésticos e nas demais aplicações que utilizem tensão fornecida pela rede padrão de distribuição de energia.

A fonte consiste de um inversor eletrônico que deve fornecer energia para carga na forma senoidal com uma amplitude controlada em 127 *Volts* de valor eficaz e com frequência fixa de 60 *Hertz*. Para tanto, foi levado em consideração os conversores *CC-CC* disponíveis e optou-se pela topologia *Push-Pull* configurado como inversor. Sendo assim abordou-se uma metodologia alternativa que elimina a necessidade de um circuito exclusivo para trabalhar como inversor propriamente dito.

Normalmente é utilizado um circuito conversor *CC-CC* para aumentar a tensão ao patamar utilizável e somente então aplicar esta tensão transformada em outro circuito meia-ponte ou ponte-completa para obter-se a tensão alternada. Para este projeto o próprio conversor *Push-Pull* realiza a elevação de tensão e suas duas chaves realizam o controle da inversão de tensão, devido à configuração de suas bobinas.

A escolha da tensão *CC* de entrada foi feita devido ao fato da facilidade de se obter baterias de automóveis, que trabalham em 12VDC. Já para o controle das chaves foi utilizado o microcontrolador 16F874A que realiza o controle dos transistores pelas suas duas saídas PWM. O código foi escrito a fim de gerar uma saída SPWM em 50kHz que tem sua largura de pulso incrementada de acordo com a função senoidal no tempo.

O controle da razão cíclica em cada chave é feito através do monitoramento da tensão retificada da saída pela porta A/D do microcontrolador. Desta forma, garante-se que a potência total entregue a carga permaneça constante independentemente da variação da tensão na bateria.

A validação do sistema foi feita a partir da simulação dos circuitos de controle e de potência nos aplicativos *Proteus 7.8* e *OrCAD 16.3* respectivamente. As formas de onda e comportamentos gerais dos circuitos serão apresentadas confirmando a aplicabilidade de suas funções no uso diário.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

2.1 TOPOLOGIA PUSH-PULL

Devido à baixa tensão disponível na entrada do circuito, visto que se considerou uma bateria comum de automóveis (fonte CC de 12V), faz-se necessária a adequação desta tensão para patamares utilizáveis. Portanto, esta tensão deverá ser elevada até o valor de pico da tensão da rede para que então, através da modulação SPWM seja fornecida a tensão alternada desejada à carga. Para realizar esta elevação optou-se pelo uso do conversor *CC-CC* elevador de tensão, adaptado para trabalhar também como inversor.

Os conversores *CC-CC* também são conhecidos como *choppers*, e são utilizados para obter tensões DC variáveis a partir de tensões fixas (AHMED, 2000). Para aplicações de potência, que envolvem alta tensão e corrente, são necessários componentes que suportem tais condições. Sendo assim, em sistemas que utilizam chaveamento, opta-se pelo uso de BJTs (*bipolar junction transistors* – transistores bipolares de junção) ou MOSFETs (*metal oxide semiconductor field effect transistors* – transistores de efeito de campo metal-óxido-semicondutores). O controle deste chaveamento é realizado pela técnica PWM (*pulse width modulation* – modulação por largura de pulso) que possui variações como o PWM senoidal que será apresentado posteriormente.

Segundo Ahmed (2000), os *choppers* são divididos em dois grupos primários: os *step-down* ou *buck* e *step-up* ou *boost*. Os conversores *step-down* são conhecidos como abaixadores, pois produzem uma tensão de saída menor ou igual à tensão de entrada, por complemento, os conversores *step-up* são conhecidos como elevadores por produzirem tensão maior ou igual à tensão de entrada.

Devido à necessidade de aumentar a tensão de entrada, utilizou-se neste trabalho um conversor elevador, mais precisamente um conversor *Push-Pull*. Este conversor pode ser definido como “um arranjo de dois conversores *Forward* operando complementarmente” (FILHO, 2005 apud BARBI, 2001). Por se tratar de um conversor isolado, devido ao seu acoplamento magnético, ele pode trabalhar tanto como elevador como abaixador.

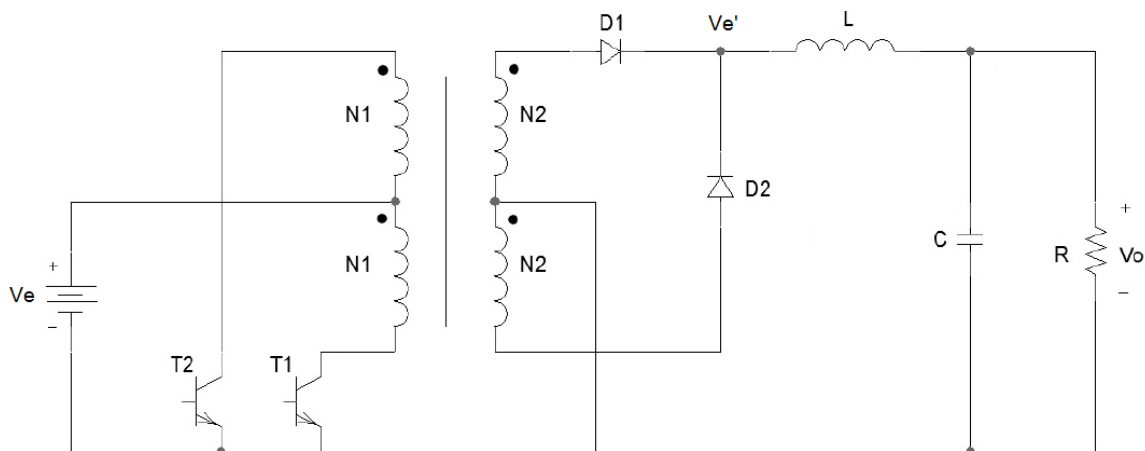
Este conversor é o mais indicado para médias potências, sendo amplamente utilizado em aplicações industriais, principalmente por sua robustez e simplicidade. Além de ser indicado para aplicações que utilizem baixa tensão de entrada e altos valores de corrente (MOHAN, 1994), já que as chaves do circuito ficam submetidas ao dobro da tensão de alimentação do sistema.

Em relação ao conversor *Forward*, esta topologia se sobressai, devido à sua curva B-H excursionar o primeiro e terceiro quadrante, enquanto que no conversor *Forward* a excursão ocorre apenas no primeiro quadrante, o que influencia diretamente do dimensionamento do transformador do circuito (TREVISIO, 2009). O *Push-Pull* trabalha com duas chaves, sendo cada uma relacionada com um sentido dos dipolos, evitando a necessidade de desmagnetização. Desta forma este conversor aproveita o tempo de chave desligada que ocorre no conversor *Forward* (FERREIRA, 2006). Este par de chaves, porém, não podem estar ligadas simultaneamente, limitando assim a razão cíclica máxima ($D_{\text{máx}}$) a 0,5 para cada uma delas. Assim para valores abaixo deste valor, ocorre o chamado tempo morto, que é o momento em que nenhuma das duas chaves conduz (TREVISIO, 2009; AHMED, 2000) e a variação do fluxo magnético é nulo.

2.1.1 Topologia Convencional

A topologia *Push-Pull* convencional consiste em um transformador com duas bobinas no primário, sendo cada bobina controlada por uma chave e estas trabalhando complementarmente entre si. A magnetização mútua entre as bobinas ocorre devido ao chaveamento do circuito pelos transistores no primário do transformador. Segundo a lei de Faraday, esta constante interrupção de fluxo magnético causada pelo chaveamento, leva ao surgimento de uma tensão grande o suficiente para que a corrente e o fluxo induzidos não se interrompam. Desta forma, os conversores possuem um arranjo de forma que a continuidade de fluxo é mantida pelos caminhos de corrente no secundário do transformador (POMILIO, 2005). O sistema básico do circuito *Push-Pull* convencional é mostrado na Figura 1.

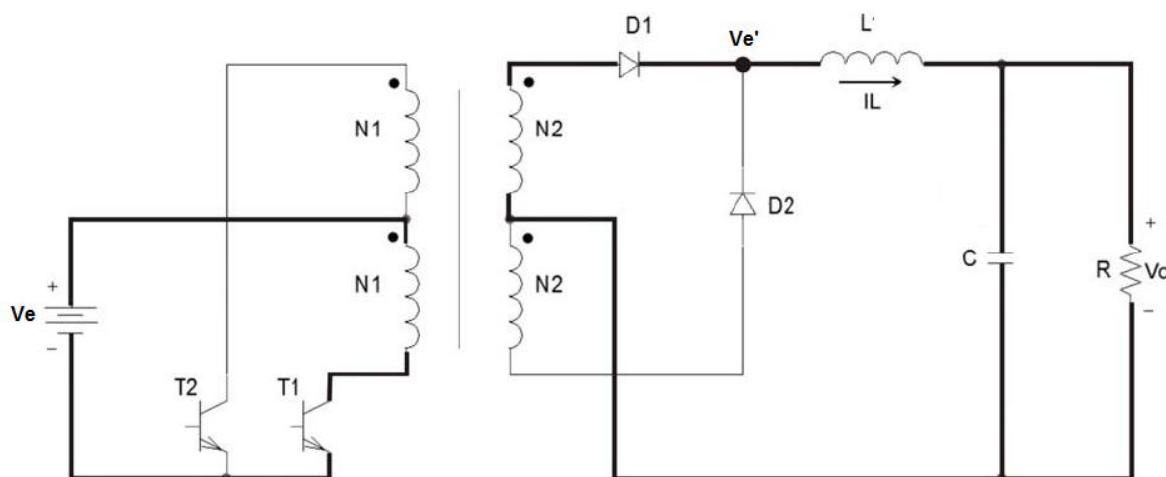
Figura 1 - Conversor Push-Pull convencional.



O corte e saturação das chaves $T1$ e $T2$ são feitos pelo controle do sistema que envia pulsos de nível lógico alto para chave que conduz e nível lógico baixo para a chave em corte.

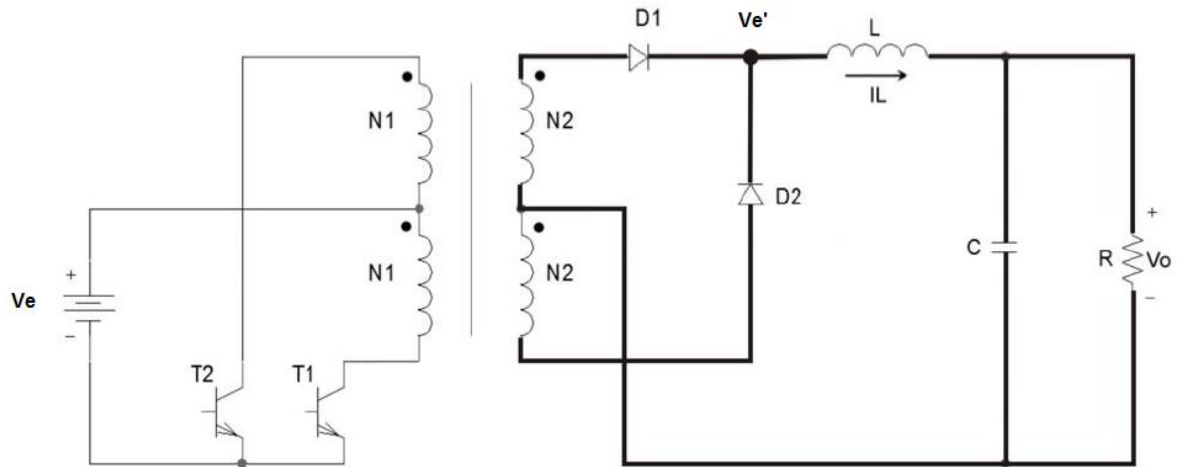
Na primeira etapa de funcionamento (Figura 2), $T1$ satura e passa a conduzir enquanto $T2$ permanece aberta, evitando a passagem de corrente. Neste instante a segunda chave encontra-se sob uma tensão de $2V_E$, considerando que os dois enrolamentos do primário possuem o mesmo número de espiras. A tensão de entrada V_E é aplicada em um dos enrolamentos primários e devido ao acoplamento magnético, uma tensão V_E' é induzida no secundário de acordo com a relação $N1/N2$ das espiras. Este pulso de tensão é retificado pelo diodo $D1$ e aplicado no indutor L enquanto o diodo $D2$ encontra-se bloqueado devido à polaridade da tensão no secundário do transformador (FERREIRA, 2006).

Figura 2 - Transistor $T1$ saturado e $T2$ em corte.



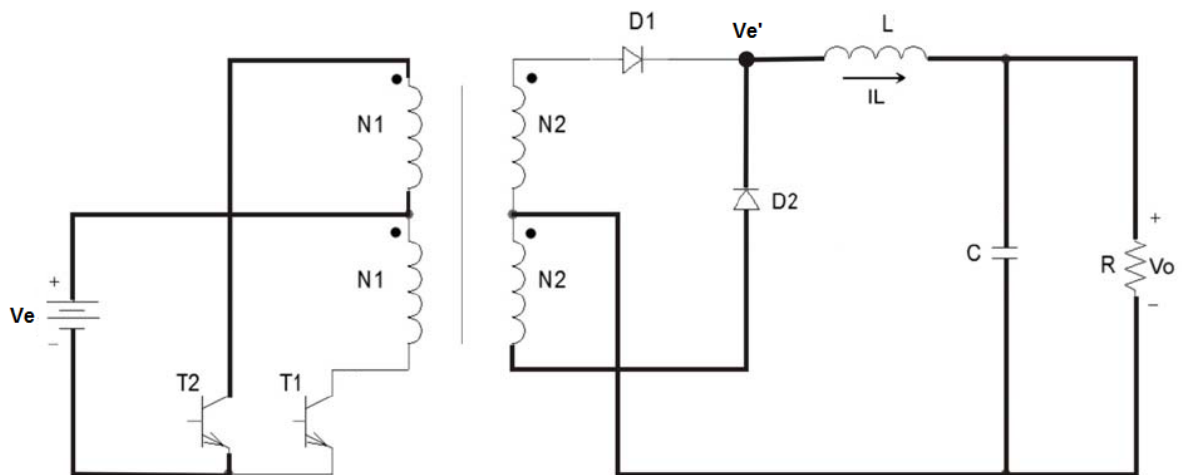
Assim que a chave $T1$ é desligada, para razões cíclicas menores que 0,5, ocorre o chamado tempo morto, em que nenhuma das chaves está ativa. Desta forma, toda corrente utilizada pela carga provém do indutor que faz com que os diodos $D1$ e $D2$ conduzam simultaneamente, operando assim, em roda livre (Figura 3). Nesta etapa o secundário do transformador entra em curto e uma tensão V_E aparece sobre cada chave, consequentemente a variação do fluxo do transformador será nula, permanecendo assim até que a próxima chave conduza (FERREIRA, 2006).

Figura 3 - Transistores T1 e T2 em corte.



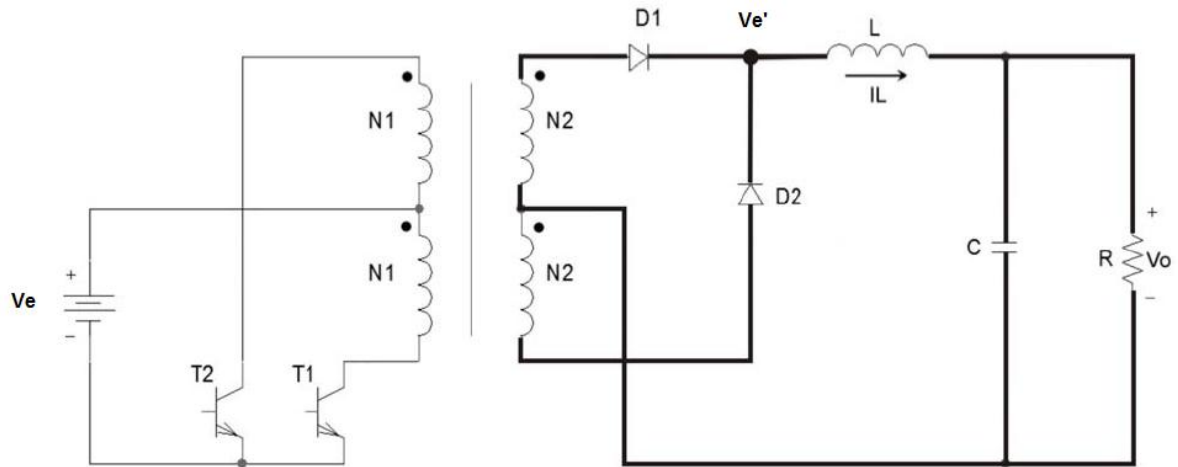
Após o tempo morto, a chave $T2$ passa a conduzir e a chave $T1$ fica em corte e sob tensão de $2V_E$ (Figura 4). A tensão V_E é aplicada agora no segundo enrolamento do primário e a tensão induzida no indutor é igual à V_E/N , sendo N a relação de espiras entre primário e secundário. O pulso de tensão é retificado pelo diodo $D2$, que passa a conduzir a corrente I_L que flui pelo indutor L . O diodo $D1$, por sua vez, permanece bloqueado devido à polaridade de tensão no secundário do transformador (FERREIRA, 2006).

Figura 4 - T1 em corte e T2 saturado.



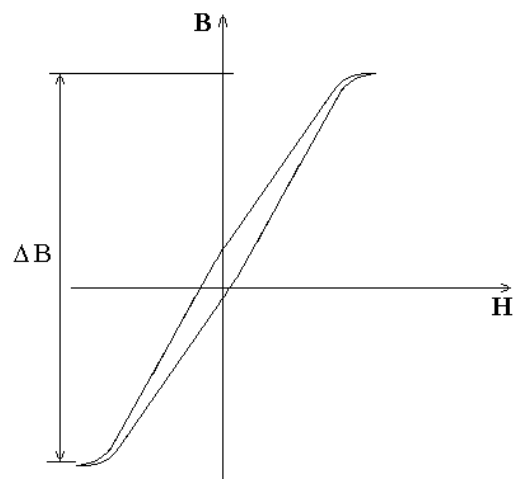
Por fim, novamente as duas chaves estão em corte, e outro tempo morto entre pulsos ocorre (Figura 5).

Figura 5 - T1 e T2 em corte.



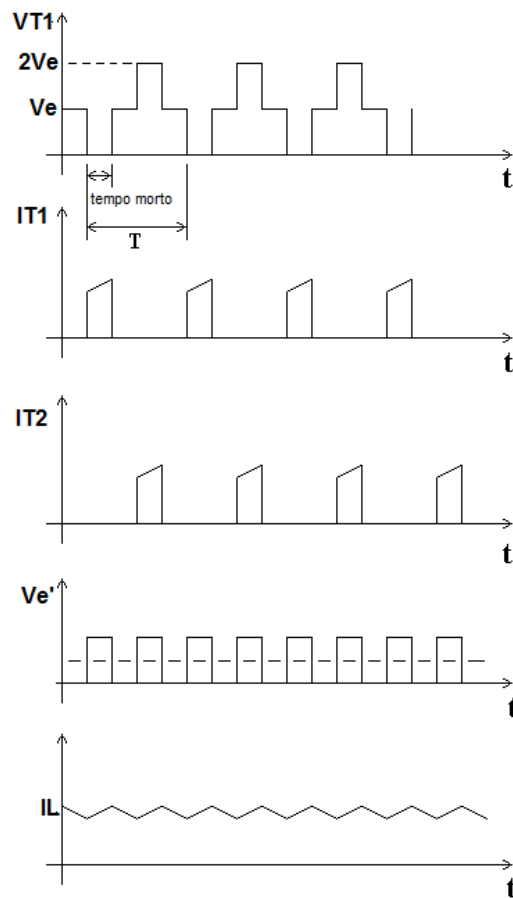
Pode-se então confirmar que na topologia *Push-Pull* a relação entre o campo magnetizante e o campo magnético (curva B-H), cuja representação gráfica se encontra na Figura 6, ocorre no primeiro e terceiro quadrantes, pois enquanto *T1* conduz o fluxo magnético é positivo, já quando *T2* conduz, o fluxo varia negativamente, por fim, durante o tempo morto a variação de fluxo magnético é nula (FERREIRA, 2006).

Figura 6 - Curva B-H do conversor Push-Pull



Por fim, deve-se notar que a frequência dos pulsos sobre indutor é duas vezes maior que a frequência no transformador, portanto a indutância necessária para manter uma corrente mínima será menor no conversor *Push-Pull* do que no *Forward*, resultando em indutores menores. (TREVISO, 2009).

Figura 7 - Formas de onda do conversor *Push-Pull*.



Fonte: Treviso (2009).

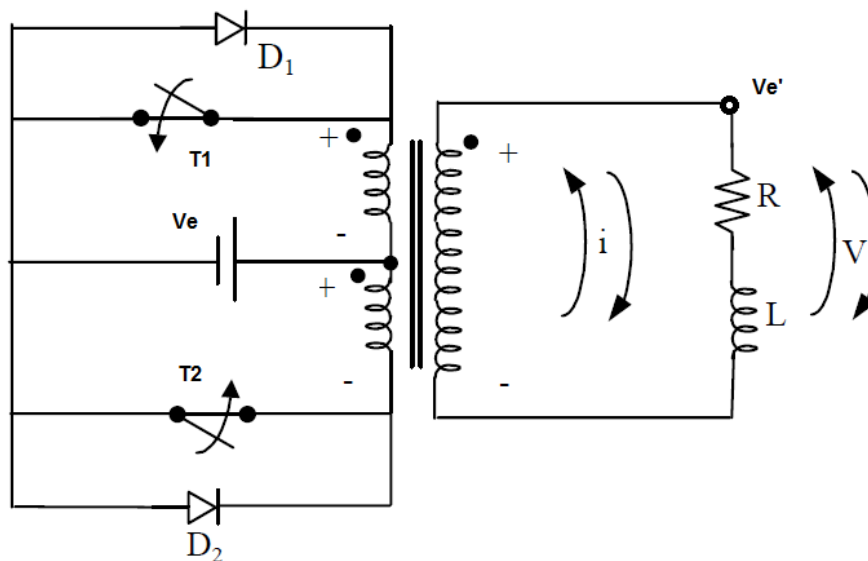
2.1.2 Topologia Push-Pull Inversor

A topologia proposta neste trabalho envolve a utilização do conversor *Push-Pull* inversor, que consiste na eliminação da retificação de tensão no secundário, como se pode verificar na Figura 8.

Estas alterações garantem que a tensão induzida no secundário do transformador e entregue a carga, possua polaridades alternadas para cada acionamento de chave. Assim, pela manipulação adequada dos pulsos em cada chave, o sistema passa da condição de estritamente realizar a conversão *CC-CC* para então realizar tanto esta conversão *CC-CC* quanto a conversão *CC-CA* que normalmente ficaria a cargo de um segundo circuito trabalhando em meia-ponte ou ponte-completa.

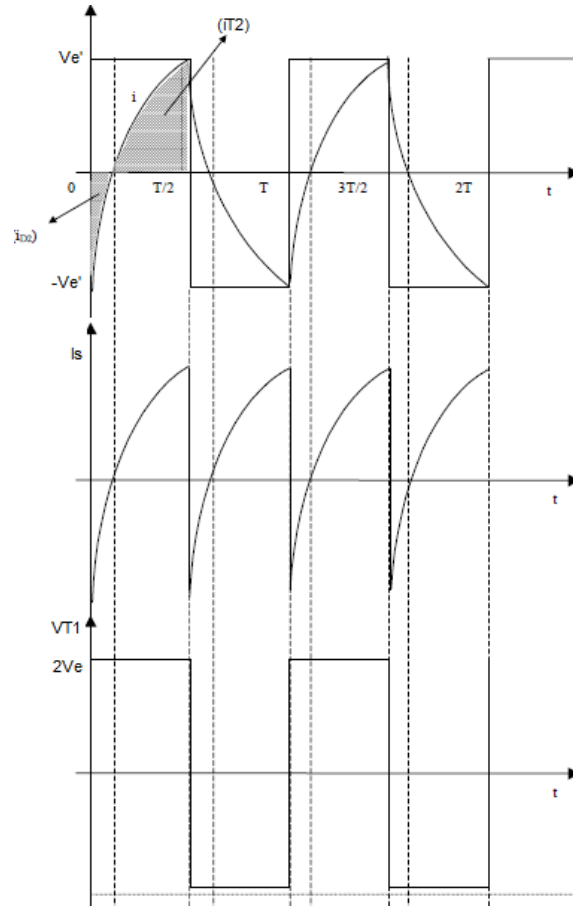
Uma desvantagem notável desta metodologia se deve ao fato de que qualquer assimetria nas duas bobinas do primário do transformador ou no tempo de chaveamento dos transistores de potência pode provocar um nível *CC* indesejado no fluxo magnético do transformador, que poderia então ter seu núcleo saturado ou então a perda de eficiência no sistema devido à circulação de correntes *CC* (RASHID, 2001).

Figura 8 - *Push-Pull* inversor.



O chaveamento do *Push-Pull* inversor ocorre da mesma maneira que na topologia convencional, porém neste sistema, o chaveamento gera uma corrente e tensão alternadas conforme a magnetização das bobinas. Como não há diodos retificadores no secundário do circuito a carga recebe tensões V_E' negativas e positivas conforme a relação de espiras. Na Figura 9 segue as formas de onda para esta topologia utilizando-se carga indutiva, nas quais ficam claras as diferenças com a topologia convencional. A carga *RL* gera uma saída de corrente curvilínea, resultado da magnetização e desmagnetização do componente indutivo o qual o circuito alimenta (SKVARENINA, 2002).

Convém notar que a Figura 9 não considera o tempo morto entre chaves, portanto a tensão nas chaves não apresenta o degrau V_E . No projeto deste trabalho será utilizado este tempo entre chaves, e a tensão sobre o transistor será semelhante ao formato encontrado no *Push-Pull* convencional.

Figura 9 - Formas de onda do *Push-Pull* inversor.

2.3 EQUACIONAMENTO DO *PUSH-PULL* INVERSOR

O projeto do conversor *Push-Pull* baseia-se em equações propostas na literatura (TREVISIO, 2009) e previamente validadas em publicações acadêmicas (FERREIRA, 2006).

A tensão de saída V_S do conversor é obtida calculando o valor médio da tensão V_E' (Figura 1).

$$V_S = \frac{2 \cdot D_{m\acute{a}x} \cdot (V_{E(m\acute{i}n)} - V_{CE(SAT)})}{N} - V_D \quad (1)$$

Considerando:

V_S – Tensão de saída (V);

$D_{m\acute{a}x}$ – Razão cíclica máxima;

$V_{E(m\acute{i}n)}$ – Tensão de entrada mínima (V);

$V_{CE(SAT)}$ – Tensão entre coletor e emissor do transistor quando saturado (V);

V_D – Tensão de queda no diodo (V);

N – Relação de espiras entre o primário e secundário;

Como se observa na Figura 8, esta topologia não realiza a retificação da tensão V_S' , portanto não se faz uso de diodos na parte secundário do circuito. Assim sendo, o valor da tensão do diodo na equação (1) é considerado nulo.

O valor de N é obtido através da relação de espiras do transformador:

$$N = \frac{N_1}{N_2} \quad (2)$$

A razão cíclica máxima é definida com o valor máximo de condução sem haver risco de que o transformador sature, definida em:

$$D_{m\acute{a}x} = 0,45 \quad (3)$$

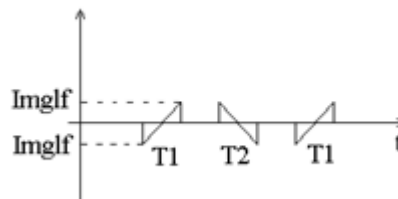
A variação da razão cíclica ocorrerá de acordo com a leitura da tensão de saída de forma que a tensão na carga permaneça constante.

2.3.1 Transformador

No *Push-Pull* a magnetização do transformador ocorre durante o tempo de condução de uma das chaves, quando a outra chave conduz, uma corrente de desmagnetização passa pelo transformador até chegar a zero e então este passa a ser magnetizado novamente.

Assim, há excursão no primeiro e terceiro quadrante da curva B-H, como é apresentado na Figura 6 que resulta na corrente de magnetização do trafo do *Push-Pull*, conforme Figura 10.

Figura 10 – Correntes de magnetização e desmagnetização e tensão dos transistores.



Fonte: Treviso (2009).

Deve-se observar, entretanto, que o tempo de condução das chaves deve ser exatamente o mesmo, caso contrário o transformador pode se saturar. Neste projeto, foi escolhido o controle através de um microcontrolador, que garante valores bem definidos para os tempos de condução das chaves (TREVISIO, 2009).

O valor do campo magnético presente neste conversor é dado por:

$$B = \frac{V_{E(min)}}{V_{E(max)}} \cdot B_{máx} \quad (4)$$

A potência aparente do transformador é obtida pela equação seguinte, na qual os fatores K_S e K_P dependem do modo como a conversão de potência é feita e η refere-se ao rendimento do transformador:

$$PA = P_S \cdot \left(K_S + \frac{K_P}{\eta} \right) \quad (5)$$

O núcleo do transformador é formado da área da janela A_J , que é o espaço disponível para o enrolamento das espiras e por uma área efetiva A_e , que é por onde flui o campo magnético (TREVISIO, 2009).

O produto das áreas descritas acima é chamado de A_P , e se relaciona com a energia do transformador por meio da equação a seguir:

$$A_P = \left(\frac{PA \cdot 10^4}{K \cdot K_U \cdot K_J \cdot B \cdot f} \right)^Z \quad (6)$$

Considerando:

PA – Potência aparente (VA);

K – Fator de forma da onda;

K_U – Fator de utilização da área da janela pelo fio;

K_J – Fator de densidade de corrente nos fios;

B – Densidade de fluxo magnético (Tesla);

f – Frequência de saída (Hz);

$Z = Z = 1 / (1 - x)$. Sendo x um valor tabelado que depende, entre outros fatores, do tipo do núcleo do transformador (ANEXO A).

Para compensar o duplo enrolamento do *Push-Pull*, K_U é considerado 0,2 (TREVISIO, 2009) e a equação (6) é simplificada para a forma abaixo:

$$A_P = \left(\frac{3,98 \cdot P_S \cdot 10^4}{K_J \cdot B \cdot f} \right)^Z \quad (7)$$

Após o cálculo do A_P , escolhe-se o melhor núcleo para o transformador conforme Anexo B.

O número de espiras no primário do transformador é obtido pela equação (8), da qual o resultado é utilizado para se obter o valor das espiras no secundário.

$$N_1 \geq \frac{V_{E(mín)} \cdot D_{máx}}{A_e \cdot B \cdot f} \quad (8)$$

O campo magnético B em (7) é obtida pela equação (4), enquanto o valor da área efetiva A_e é obtido pelo Anexo B.

A área de cobre a ser utilizada no secundário é obtida pela equação que se segue:

$$A_{CU} = \frac{I_{RMS(SEC)}}{J} \quad (9)$$

Sendo J a densidade de corrente medida em A/m² e obtida pela equação abaixo:

$$J = K_J \cdot A_P^{-x} \quad (10)$$

E a corrente no secundário $I_{RMS(SEC)}$ é obtida pela equação (11):

$$I_{RMS(SEC)} = I_S \cdot \sqrt{2 \cdot D_{máx}} \quad (11)$$

A área de cobre no primário é obtida da mesma forma que no secundário, como é mostrado a seguir:

$$A_{CU} = \frac{I_{RMS(PRI)}}{J} \quad (12)$$

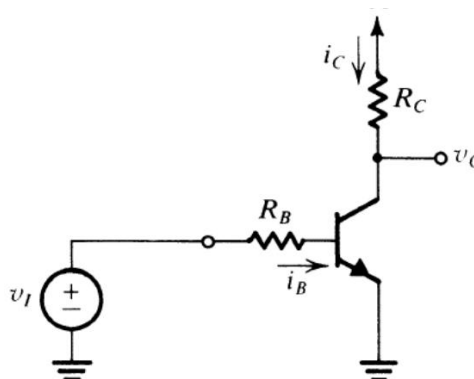
E por fim, a corrente RMS no primário do transformador é dada por:

$$I_{RMS(PRI)} = \frac{I_{RMS}}{2 \cdot N} \quad (13)$$

2.3.2 Transistores

Nos conversor *Push-Pull* existe o elemento que chaveia e regula a alimentação do sistema a cada ciclo. Os transistores são componentes que são amplamente utilizados com esta finalidade devido aos seus modos de trabalho. Neste projeto, foi utilizado um *TBJ* (Figura 11) ou transistores bipolares de junção que podem ser manipulados para atuarem em saturação ou em corte. Estes transistores combinam facilidade de controle com baixas perdas mesmo quando submetidos a altas tensões. Sua comutação é rápida o suficiente para que ele possa ser empregado em circuitos chaveados enquanto sua alta impedância de base requer apenas pequenas quantidades de energia para ativar o dispositivo (MOHAN, 2003).

Figura 11 - Circuito de ilustração dos modos de operação do TBJ.



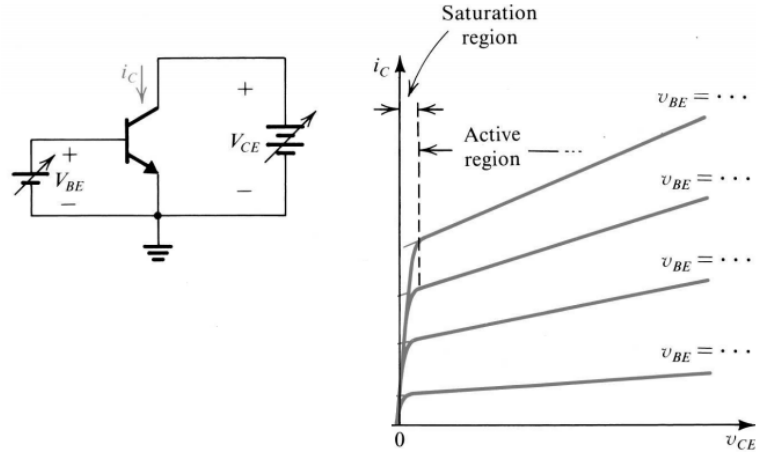
Fonte: Sedra, 1998.

O transistor bipolar é um dispositivo acionado por corrente, desta forma, ao se injetar uma corrente i_B em sua base, uma outra corrente i_C circulará pelo coletor (TREVISIO, 2009). Para tensões v_I menores que 0,5 V, o transistor se encontra em corte, pois a tensão entre base e emissor é menor do que a necessária para polarizar o transistor, assim nenhuma corrente i_C é gerada e o circuito não conduz.

Se uma corrente de base for gerada de modo que as junções base-emissor e base-coletor sejam polarizadas diretamente, uma corrente i_C , que depende de R_C , é gerada e o transistor passa a conduzir corrente (SEDRA, 1998).

Desta forma os transistores são ativados e desativados, alternando entre os modos de saturação e corte respectivamente, resultando assim em um estado de constante chaveamento. Estes dois modos de operação (Figura 12) são utilizados para controlar cada ciclo de trabalho do conversor, que são gerenciados pelo circuito de controle.

Figura 12 - Regiões de operação do TBJ.



Fonte: Sedra, 1998

Para o projeto das chaves no *Push-Pull* é preciso lembrar que as chaves ficam submetidos a uma tensão com valor igual ao dobro da tensão de entrada do circuito devido à polaridade dos enrolamentos do primário e suas espiras serem iguais.

Desta forma, a tensão máxima sobre o transistor quando esta em corte é:

$$V_{CE(máx)} = 2 \cdot V_{E(máx)} \quad (14)$$

A corrente eficaz que flui em cada transistor quando saturado é igual a:

$$I_{EFICAZ} = \frac{I_S \cdot \sqrt{2 \cdot D_{máx}}}{2 \cdot N} \quad (15)$$

A corrente média em cada chave no primário é obtida pela equação a seguir:

$$I_{MÉDIA} = \frac{I_S \cdot D_{máx}}{N} \quad (16)$$

O dimensionamento dos diodos depende da corrente máxima a qual eles serão submetidos, para tanto, primeiro calculamos a corrente de saída do circuito.

$$I_S = \frac{P_S}{V_S} \quad (17)$$

A corrente de pico no diodo é dada por:

$$I_{PICO} = I_S + I_{S(mín)} \quad (18)$$

Sendo $I_{S(min)}$ estipulado em um décimo da corrente I_S de saída.

A corrente média e eficaz são obtidas conforme apresentado a seguir:

$$I_{MÉDIA} = I_S \cdot D_{(máx)} \quad (19)$$

$$I_{EFICAZ} = \frac{I_S \cdot \sqrt{2 \cdot D_{máx}}}{N} \quad (20)$$

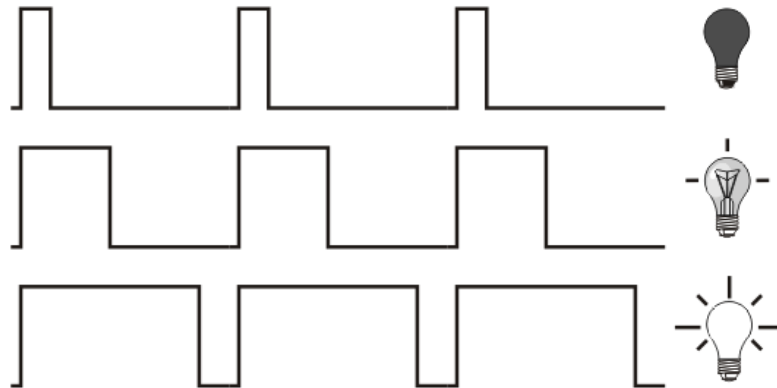
Este dimensionamento é necessário apenas na topologia conversora, visto que no projeto atual, o circuito não possui diodos no secundário do transformador.

2.4 CONTROLE

O controle das chaves do *Push-Pull* é feito a partir de uma variação da técnica PWM (*Pulse Width Modulation*), ou Modulação por Largura de Pulso.

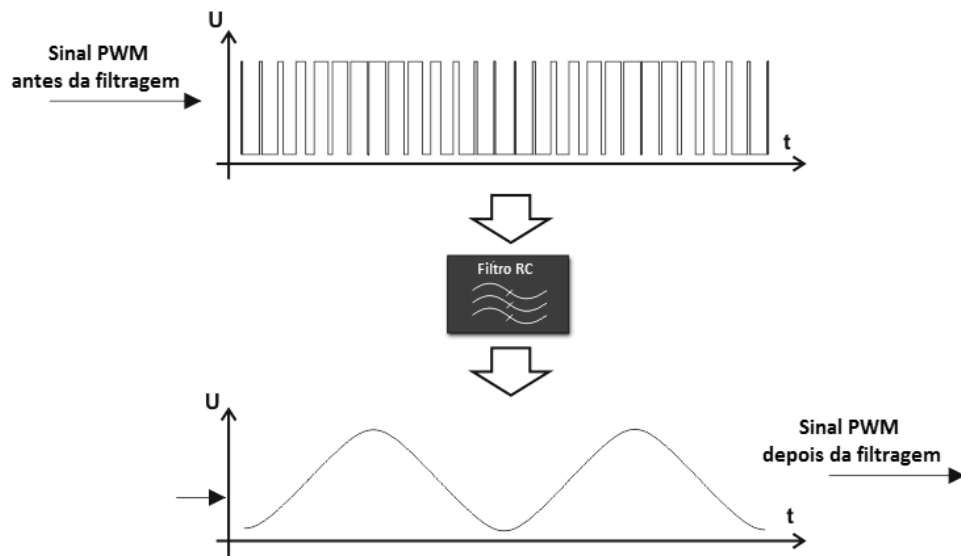
Sinais de frequência e razão cíclica variável são amplamente utilizados em automações de processos. Com o aumento do ciclo de trabalho, mais potência é aplicado à carga como exemplifica a Figura 13.

Figura 13 - Variação de potência através de sinal PWM.



O uso de sinais PWM possibilita também a geração de sinais com formas de ondas arbitrárias. A Figura 14 apresenta a geração de uma onda senoidal a partir da modulação da largura de pulso do sinal.

Figura 14 - Onda senoidal através de um sinal PWM filtrado.



Fonte: Site mikroe.com (2011).

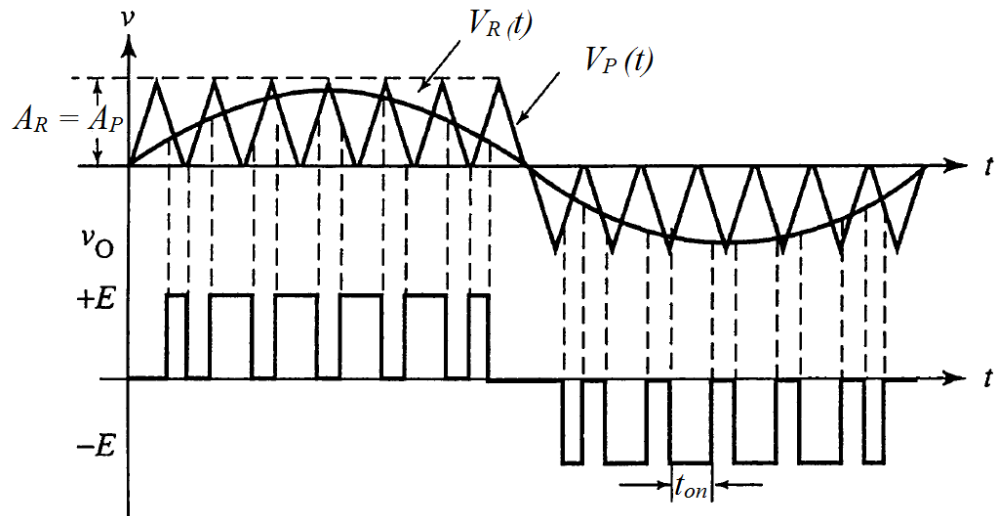
Este método consiste em um único pulso a cada semi-ciclo de tensão de saída, sendo que a largura de cada pulso D a fim de controlar a tensão CA de saída. Quanto maior a largura de pulso, mais potência é aplicada a carga. Assim, variando-se o ciclo ativo de cada pulso de maneira adequada, é possível gerar diferentes formas de onda na saída do circuito.

A técnica aplicada no inversor *Push-Pull* é chamada de SPWM (*Sinusoidal Pulse Width Modulation*), ou Modulação por Largura de Pulso Senoidal, na qual a tensão de saída é controlada pela variação do tempo de trabalho dos pulsos, de forma que estes tempos D sejam maiores no pico da onda.

Segundo (AHMED, 2000), o conceito do sinal SPWM consiste na comparação de uma onda senoidal de referência com uma onda portadora de alta frequência. Desta forma, a onda de referência, $V_R(t)$, é uma onda senoidal de amplitude A_R , e frequência f_R , que rege a frequência de saída do sistema. Já a onda portadora é um sinal triangular (t) de amplitude A_P e frequência f_p que é bem maior que a frequência de saída.

A largura de pulso é modulada pelo tempo em que a tensão referência é maior que a tensão da portadora ($V_R(t) > V_P(t)$) durante o semi-ciclo positivo. No semi-ciclo negativo a relação se inverte, e agora a modulação ocorre no tempo em que a tensão da portadora é maior que a tensão de referência ($V_R(t) > V_P(t)$). A comparação entre as ondas é feita conforme demonstrado na Figura 15.

Figura 15 - Geração de um sinal SPWM.

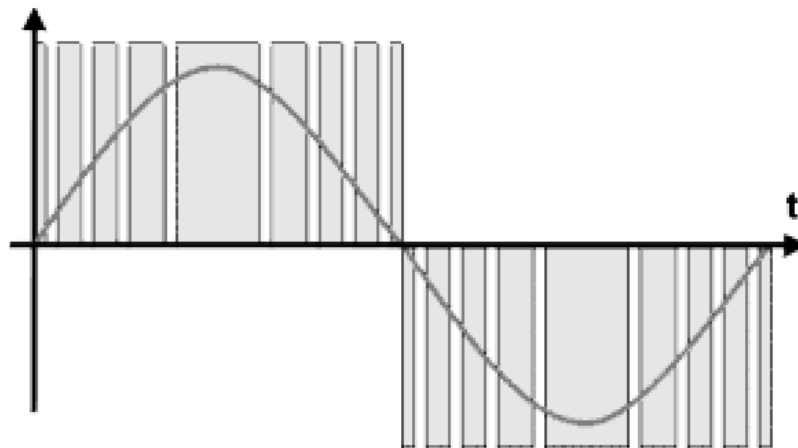


Fonte: Rashid (2001).

A tensão de saída é regulada pela relação de frequências f_P/f_R , que determina o número de pulsos de cada semi-ciclo e pela relação de pico das tensões A_R/A_P , que determina a largura dos pulsos e possui valor máximo igual a uma unidade e é denominada índice de modulação. O ajuste do índice de modulação geralmente é feito pela variação da amplitude do sinal de referência, enquanto a amplitude da portadora permanece constante. O resultado deste controle determina o valor da tensão RMS na saída do inversor.

Dito isto, o comportamento do SPWM pode ser observado na Figura 16.

Figura 16 - Potência transmitida por um sinal SPWM.

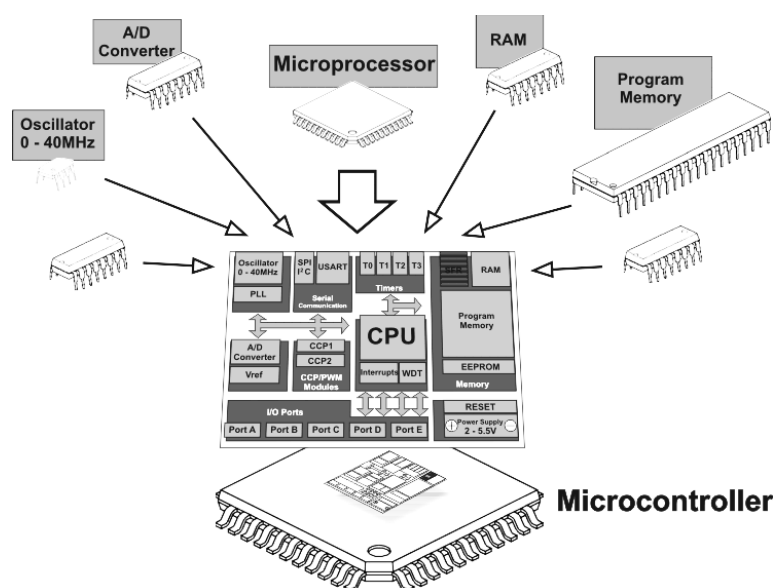


2.5 MICROCONTROLADORES

No projeto do *Push-Pull* inversor, optou-se por realizar o controle das chaves por meio de um microcontrolador, que gera os pulsos SPWM em cada transistor.

O microcontrolador é um circuito integrado programável que une diversos componentes como memória RAM, memória de armazenamento de programas, portas de entrada e saída, *timers*, oscilador interno, conversor analógico digital entre outros componentes.

Figura 17 - Agrupamento de funções de um microcontrolador.



Fonte: Site mikroe.com (2011).

Os microcontroladores possuem registradores internos que são células eletrônicas que podem memorizar o estado de um byte. Cada bit de um registrador está relacionado a circuitos internos do componente, como *timers*, conversor A/D entre outros. Assim sendo, eles estão diretamente no comando destas funções de acordo com a característica de cada microcontrolador.

A conexão do microcontrolador com componentes periféricos é feito a partir de pinos de entrada e saída (I/O). Programando os registradores de cada um desses pinos pode-se manipular o sinal de saída ou utilizar dados de entrada para gerenciar funções internamente no programa.

A memória é a parte do microcontrolador que é utilizada para armazenamento de dados. A memória ROM (*Read Only Memory*) é usada para salvar permanentemente o programa a ser executado. O tamanho desta memória depende do tipo do microcontrolador

utilizado. Atualmente é comum a utilização de endereçamento 16-bit, ou seja, cerca de 64 Kb de memória ROM disponível (DATASHEET PIC16F87XA, 2003).

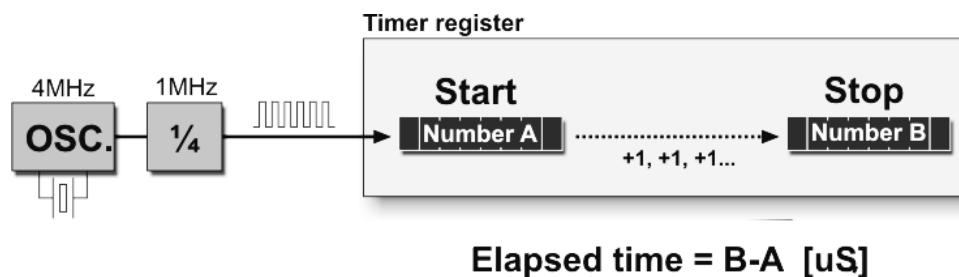
A memória RAM (*Random Access Memory*) mantém os dados apenas quando o microcontrolador está alimentado. Assim que a fonte de tensão é retirada os dados são perdidos. Esta memória é utilizada para armazenar dados temporários e gerenciar dados criados durante as operações do microcontrolador.

Assim como a memória RAM, a memória EEPROM pode ser alterada durante as operações, porém os dados são mantidos permanentemente salvos, mesmo se o microcontrolador deixar de ser alimentado.

O oscilador interno gera pulsos que habilita operações síncronas de todos os circuitos do microcontrolador. Geralmente o módulo do oscilador é configurado utilizando um cristal de quartzo ou um ressonador cerâmico. Porém, as instruções não são executadas na velocidade nominal do oscilador, mas sim em intervalos bem maiores que o período de *clock*. Isto acontece porque cada instrução é executada em vários passos e esta é a velocidade que deve ser considerada em cálculos que envolvem *timers*. Esta diferença de tempo de execução já está sendo considerada nas fórmulas apresentadas nos *datasheets* (DATASHEET PIC16F87XA, 2003).

Os cristais de quartzo são amplamente utilizados por serem muito estáveis e seus pulsos terem sempre a mesma largura, o que os tornam ideais para mensurar intervalos de tempo. Nos microcontroladores, os pulsos vindos do oscilador passam pela pré-escala, que diminui o valor do *clock*, como mencionado, e incrementam os registradores de tempo.

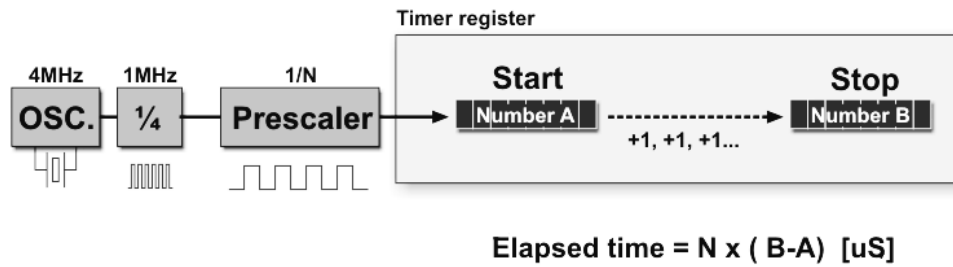
Figura 18 - Registradores de tempo.



Fonte: Site mikroe.com (2011).

Para medir tempos maiores, é comum a utilização de pré-escalas (*prescaler*) que reduz a frequência do contador a um fator pré-determinado. Assim, o *timer* demora 1, 2, 4 ou N vezes mais que o tempo padrão para realizar a contagem. A utilização deste recurso é demonstrada na Figura 19.

Figura 19 - Registradores de tempo com pré-escalas.

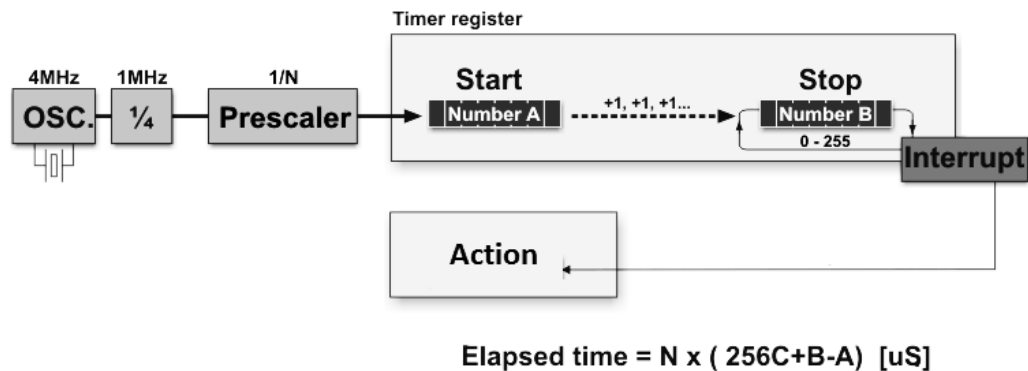


Fonte: Site mikroe.com (2011).

Para um registrador de tempo de 8 bits, o maior número que pode ser escrito nele é 255. Se este número se excede, o *timer* automaticamente zera o contador que então reinicia a contagem. Este acontecimento é conhecido como *overflow* e pode ser utilizado para gerar interrupções que podem ser utilizadas de várias maneiras.

A Figura 20 ilustra o uso da interrupção por tempo.

Figura 20 - Conceito da interrupção por tempo.



Fonte: Site mikroe.com (2011).

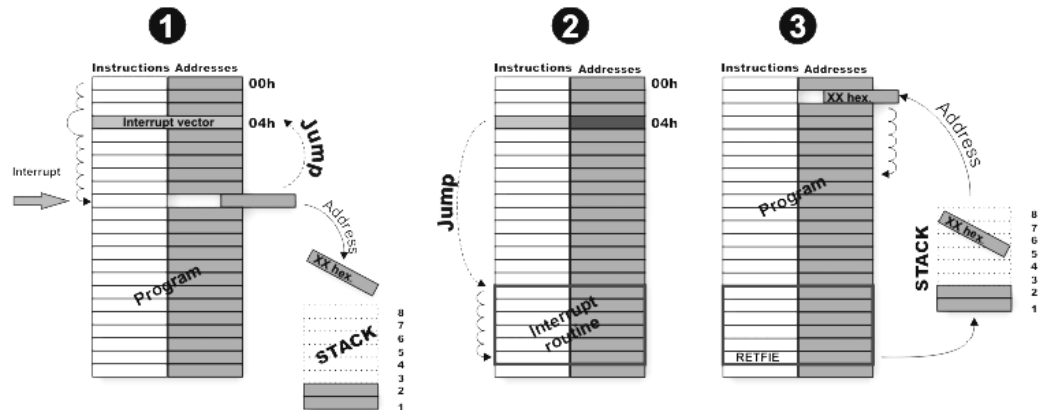
Quando a interrupção ocorre o microcontrolador executa a instrução referente a esta interrupção e para a execução normal do programa. Automaticamente o endereço do programa em execução é colocado na pilha e então o endereço que o programa irá continuar sua execução é chamado de vetor de interrupção.

A parte do programa que é chamada quando uma interrupção ocorre é chamada de rotina de interrupção. É a primeira instrução localizada no vetor de interrupção, sendo que cada interrupção disponível possui seu próprio vetor.

Por fim, quando a instrução programada para esta interrupção termina, o microcontrolador recupera o endereço do programa principal que está armazenado na pilha e continua a partir da última instrução realizada.

Estas etapas são representadas graficamente na Figura 21.

Figura 21 - 1) Armazenamento do endereço atual do programa; 2) Acesso a rotinas de interrupção e 3) Retorno à execução do programa.



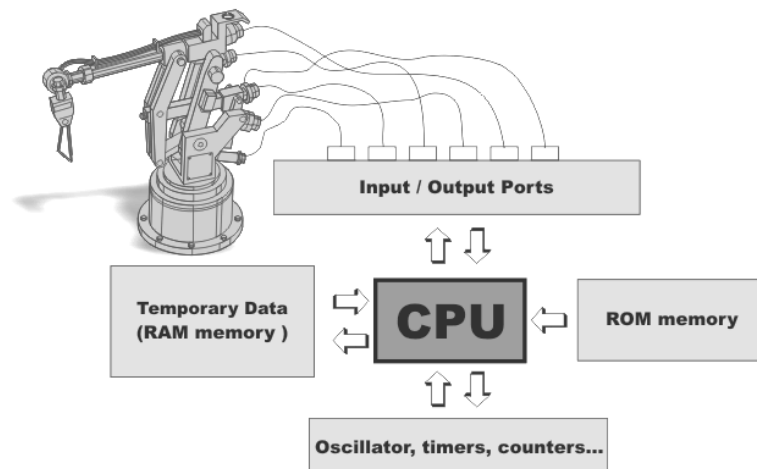
Fonte: Site mikroe.com (2011).

O conversor analógico/digital (*A/D Converter*) é utilizado para transformar sinais externos em um formato que seja compatível com o microcontrolador. O conversor é um circuito eletrônico que converte sinais contínuos em números discretos. Assim, sinais analógicos são transformados em valores binários que podem ser utilizados internamente em programas e manipulados pelo usuário.

Os microcontroladores também possuem uma Unidade de Processamento Central (*Central Processor Unit - CPU*) que monitora e controla todos os processos internos do componente, dos quais podemos ressaltar a decodificação de instruções, e as operações lógicas e matemáticas.

Por fim, pode-se configurar o microcontrolador para comunicação serial com componentes externos. Os dois tipos mais utilizados são o I2C (*Inter Integrated Circuit*) e SPI (*Serial Peripheral Interface Bus*) das quais se pode regular também o *Baud Rate*, que é a taxa de transmissão de bits que é fundamental neste tipo de comunicação.

Figura 22 - Diagrama de entrada e saída de dados.



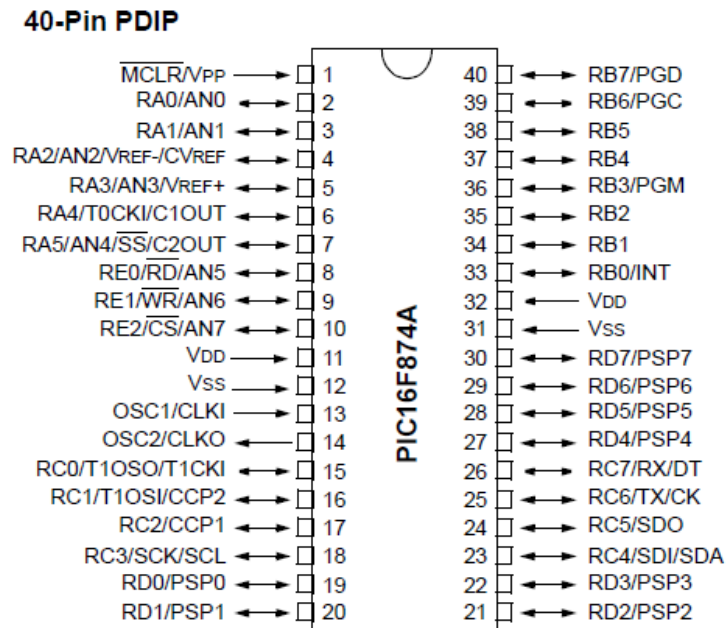
Fonte: Site mikroe.com (2011).

2.6 MICROCONTROLADOR 16F874A

O controle do inversor *Push-Pull* exige duas saídas para controlar o chaveamento dos transistores e uma entrada para leitura de tensão que realiza o controle do sistema de acordo com a tensão de saída. Portanto, optou-se pelo uso do PIC 16F874A, um microcontrolador da *Microchip* que possui duas saídas PWM e entradas analógicas, portanto atende aos requisitos de aplicação no sistema.

No PIC 16F874A existem três blocos de memória. A memória flash possui capacidade de 4K *words*, sendo cada *word* igual a 14 bits. A EEPROM armazena até 128 *bytes* e a Memória de Dados armazena 192 *bytes*. Seu componente de 40 pinos que faz parte deste projeto é apresentado na Figura 23.

Figura 23 - Visão geral do PIC 16F874A.



Fonte: Data Sheet 16F87XA (2003).

Em um aspecto geral, o microcontrolador da Figura 23, possui 15 interrupções programáveis, cinco portas tipo entrada/saída (*In/Out*), dois módulos PWM e oito canais multiplexados de conversor A/D (DATASHEET PIC16F87XA, 2003).

A geração de um sinal SPWM através do PIC depende da manipulação de alguns módulos do componente, como controle de *timers*, que gerencia a frequência do PWM, interrupções que são utilizadas para alterar a largura de pulso em períodos pré-determinados e conversores A/D responsáveis pelo controle de realimentação do inversor.

Estes blocos principais serão detalhados individualmente e sua compreensão é de fundamental importância para a programação do sinal SPWM.

2.6.1 Timers

O microcontrolador PIC16F874A é composto de três *timers* independentes, chamados de TMR0, TMR1 e TMR2 (DATASHEET PIC16F87XA, 2003).

O Timer TMR0 é muito utilizado para escrever programas ou sub-rotinas para gerar pulsos de duração arbitrária, mensurar tempo ou contar pulsos ou eventos externos de maneira praticamente ilimitada (SILVA, 2006).

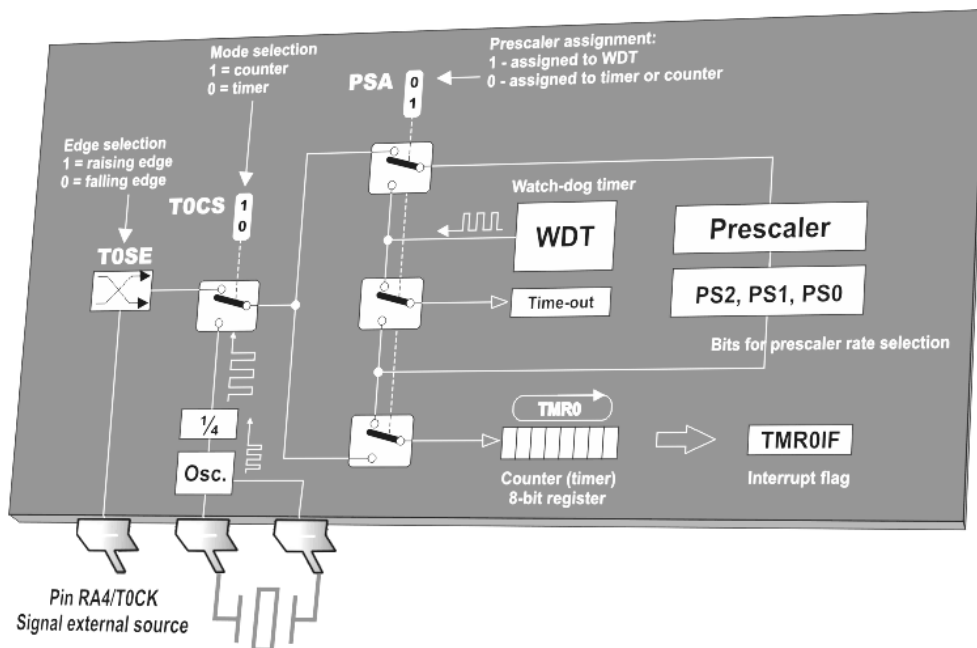
O módulo TMR0 conta com as seguintes características:

- 8-bit *timer*/contador;

- 8-bit pré-escala;
- Referência programável entre *clock* interno ou externo;
- Interrupção por estouro de tempo (*overflow*);
- Seleção programável de borda de *clock* externo.

A Figura 24 representa o esquemático do Timer TMR0 com todos os bits que determinam sua operação.

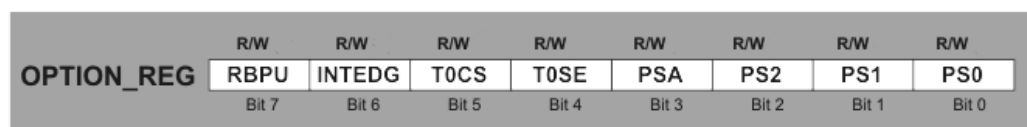
Figura 24 - Esquemático do módulo Timer 0.



Fonte: Site mikroe.com (2011).

O controle do TMR0 é feito através do registrador OPTION_REG que é estruturado conforme a Figura 25.

Figura 25 - Registrador OPTION_REG.



Fonte: Site mikroe.com (2011).

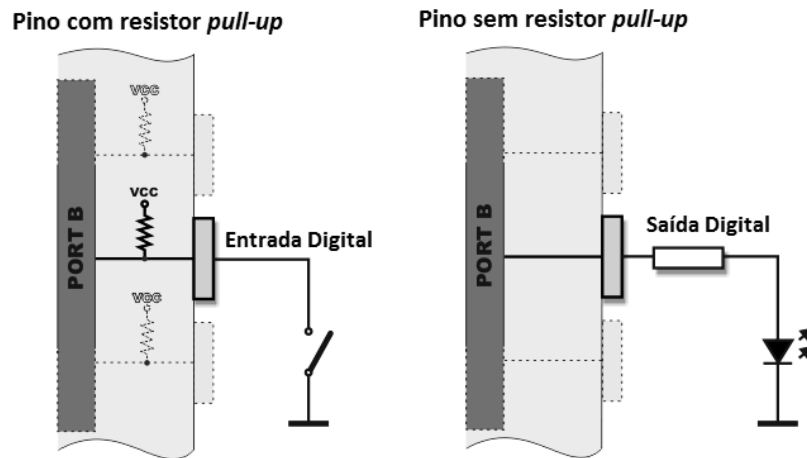
Todos os bits do registrador são passíveis de leitura e escrita (R/W) e quando o microcontrolador é reiniciado recebem o bit 1.

Cada bit controla uma função do TMR0, como é descrito abaixo:

- RBPU - *PORTB Pull-up enable bit*
 - 1 – Resistores *pull-up* da porta PORTB desabilitados.
 - 0 – Pinos da porta PORTB podem ser conectados em resistores *pull-up*.
- INTEDG - *Interrupt Edge Select bit*
 - 1 – Interrupção por borda de subida no pino 33 INT (0-1);
 - 0 - Interrupção por borda de descida no pino 33 INT(1-0).
- T0CS - *TMR0 Clock Select bit*
 - 1 – Pulsos são trazidos ao *timer* TMR0 através do pino RA4 (oscilador externo);
 - 0 – Ciclo de *clock* interno ($F_{osc}/4$).
- T0SE - *TMR0 Source Edge Select bit*
 - 1 – Incrementa transição lógica de alto-para-baixo no pino TMR0;
 - 0 - Incrementa transição lógica de baixo-para-alto no pino TMR0.
- PSA - *Prescaler Assignment bit*
 - 1 – A pré-escala é atribuída pelo WDT (*Watch-dog timer*);
 - 0 – A pré-escala é atribuída pelo *timer* TMR0.
- PS2, PS1, PS0 - *Prescaler Rate Select bit*
 - A taxa da Pré-escala é ajustada pela combinação destes três bits. Esta combinação resulta em diferentes taxas de pré-escalas para o *timer* TMR0 e *watch-dog timer*.

O Bit7 do registrador OPTION_REG habilita os resistores *pull-up* na porta PORTB. Todas as portas possuem resistores *pull-up* internamente, que os fazem ideais para algumas aplicações como ligações em *push-buttons*. Estes resistores virtuais possuem vários kilo ohms e são conectados na lógica CMOS do circuito interno do PIC, por isto possuem esta nomenclatura, visto que não precisam ser implementados externamente caso seu uso seja necessário. Apesar disto, não afetam os pinos configurados como saídas (*outputs*), mas são úteis complementos para entradas (*inputs*).

A representação gráfica destes resistores *pull-up* é apresentada na Figura 26.

Figura 26 - Resistores *pull-ups*.

Fonte: Site mikroe.com (2011).

Neste projeto, o Timer TMR0 é utilizado para gerar rotinas de interrupção a cada intervalo de tempo pré-determinado. A cada interrupção a largura de pulso do sinal PWM é alterada de maneira específica, a fim de gerar um pulso SPWM.

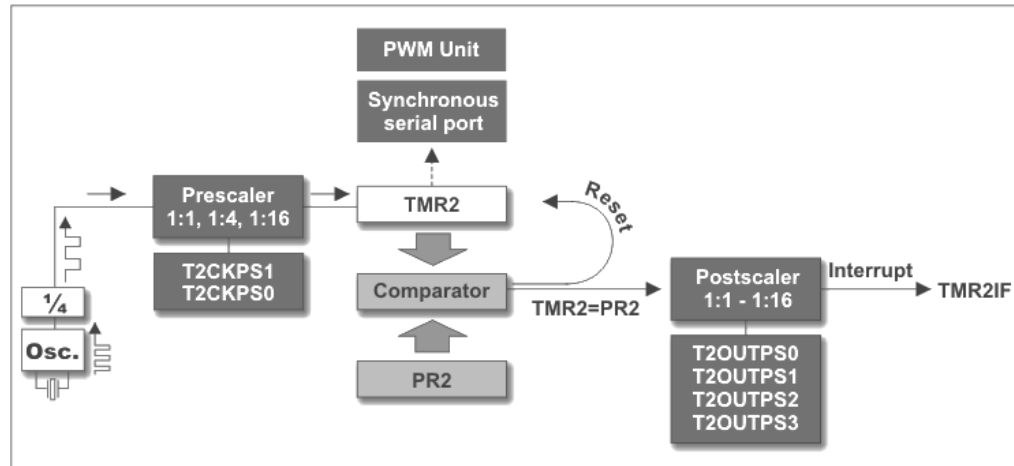
O módulo Timer TMR1 é composto por um contador de 16 bits, ou seja, é composto por dois registradores (TMR1L e TMR1H) e pode contar até 65.535 pulsos. O controle deste *timer* é feito pelo registrador T1CON, porém não será detalhado, pois este módulo não foi utilizado no controle do inversor *Push-Pull*.

Por fim, o último módulo temporizador é chamado de Timer TMR2, que é composto por um *timer* de 8 bits que opera de maneira bem específica.

Os pulsos do oscilador de quartzo passam primeiramente pela pré-escala, cujos valores podem ser alterados pela combinação dos bits T2CKPS1 e T2CKPS0. O sinal de saída da pré-escala é utilizada para incrementar o registrador TMR2, que inicia em 0x00. Os valores de TMR2 e PR2 são constantemente comparados e o registrador TMR2 continua sendo incrementado até atingir o valor de PR2. Quando isto ocorre, o registrador TMR2 é automaticamente zerado e assume novamente o valor 0x00. O *timer* da pós-escala de TMR2 é incrementado e sua saída é utilizada para gerar uma interrupção, caso esteja habilitada (SILVA, 2006).

Este mecanismo é representado graficamente na Figura 27.

Figura 27 - Lógica de funcionamento do módulo Timer 2.



Fonte: Site mikroe.com (2011).

Os registradores TMR2 e PR2 são de leitura e escrita e a contagem do temporizador pode ser desativada ao zerar-se o bit TMR2ON, o que contribui para economia de energia, visto que assim o microcontrolador não precisa gerenciar este *timer*. O controle do Timer TMR2 é feito pelo registrador T2CON, cuja estrutura é apresentada na Figura 28. Os bits deste registrador também são de leitura e escrita e seus bits são zerados a cada reset do microcontrolador.

Figura 28 - Registrador T2CON

		R/W	R/W	R/W	R/W	R/W	R/W	R/W
T2CON	-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Fonte: Site mikroe.com (2011).

- TOUTPS3 – TOUTPS0 – *Timer2 Output Postscaler*
 - Os bits selecionados determinam o valor da pós-escala.
- TMR2ON – *Timer2 On bit*
 - 1 – Liga Timer TMR2;
 - 0 – Desliga Timer TMR2.
- T2CKPS1, T2CKPS0 – *Timer2 Clock Prescale*
 - Determina o valor da pré-escala.

Alguns pontos devem ser levados em conta ao se utilizar o Timer TMR2. Assim que ligado, o registrador PR2 assume o valor 0xFF ou 255 em valores decimais. Tanto a pré

quanto a pós-escala são zeradas quando se escreve nos registradores TMR2 e T2CON e quando o microcontrolador é reiniciado.

O papel do Timer TMR2 é gerenciar a temporização do bloco PWM, que por sua vez controla cada uma das chaves do *Push-Pull* inversor.

2.6.2 Módulo PWM

O PWM do microcontrolador é controlado pelo módulo CCP, acrônimo de *Capture/Compare/PWM*, ou Captura/Comparação/PWM. Este é um módulo periférico que habilita a temporização e controle de diferentes eventos.

O *Capture Mode* habilita a contagem de tempo de um evento, que pode ser uma borda de subida ou descida ou a cada quatro ou 16 transições de borda de subida.

O *Compare Mode* compara os valores contidos nos registradores CCPR1 e TMR1. Quando os valores são iguais é possível disparar um evento externo, possibilitando gerar um sinal lógico alto, baixo ou manter o valor anterior assim que o tempo pré-determinado expira.

Finalmente, com o modo PWM é possível gerar sinais de várias frequências e ciclos de trabalho diferentes como será mais detalhadamente explicado.

Cada um destes modos usa uma fonte de *timer* para realizar sua operação. Como mostrado na Tabela 1 o modo PWM utiliza o Timer2, e os outros modos utilizam o Timer1 (DATASHEET PIC16F87XA, 2003).

Tabela 1 - Timers de referência do módulo CCP.

Modo CCP	Timer de Referência
Captura	Timer 1
Comparação	Timer 1
PWM	Timer 2

O PIC16F874A possui dois módulos CCP, chamados de CCP1 e CCP2 que são idênticos, exceto por algumas configurações apresentadas na Tabela 2, que não influenciam no projeto de controle.

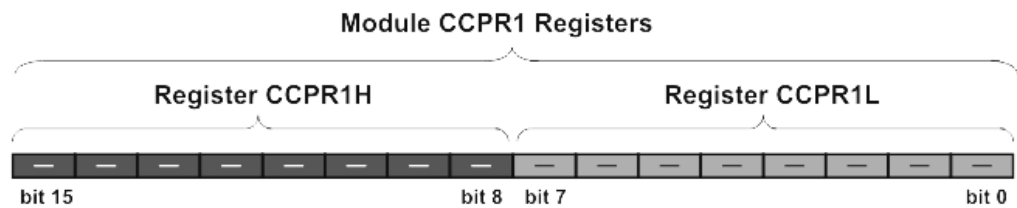
Tabela 2 - Interação entre modos do módulo CCP.

Modo CCPx	Modo CCPy	Interação
Captura	Captura	Mesma base de TMR1
Comparação	Comparação	Comparação configurada para eventos que reiniciam TMR1
PWM	PWM	PWM possui a mesma frequência de atualização que TMR2
PWM	Captura	Nenhuma
PWM	Comparação	Nenhuma

Nesta etapa será descrito o funcionamento do módulo CCP1 que também se aplica ao CCP2 com as devidas considerações de nomes de registradores para cada um.

Este módulo é composto por um registrador de 16 bits, conforme Figura 29, chamado CCPR1 que é composto por dois registradores de 8 bits chamados de CCPR1L e CCPR1H. Ele é utilizado para capturar ou comparar os números binários de TMR1 nos modos *Capture* e *Compare*, respectivamente.

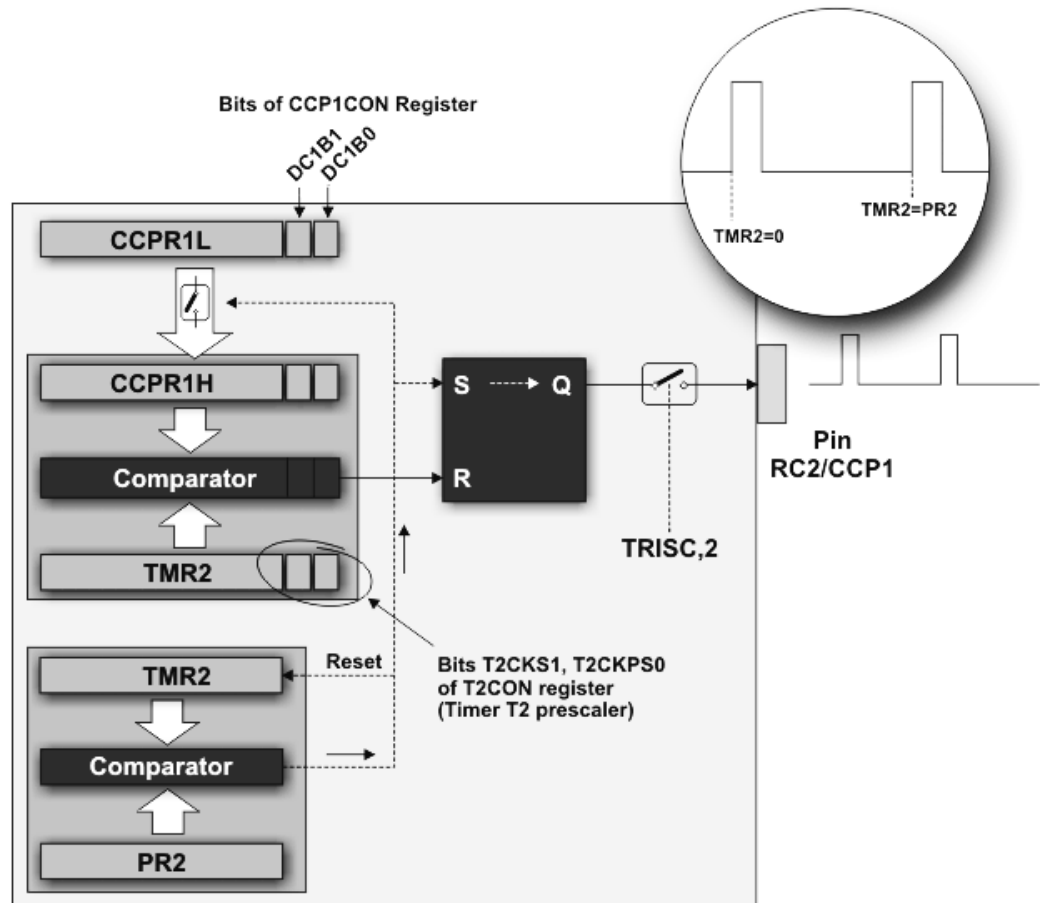
Figura 29 - Registrador CCPR1



Fonte: Site mikroe.com (2011).

A Figura 30 mostra o diagrama de bloco do módulo CCP1 que será detalhado a seguir.

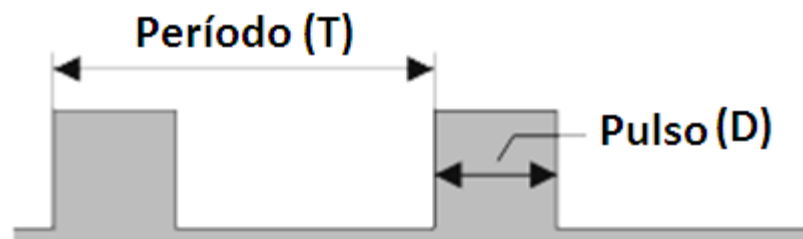
Figura 30 - Lógica de funcionamento do módulo CCP1.



Fonte: Site mikroe.com (2011).

Para geração de um pulso arbitrário no pino de saída é necessário determinar apenas o período e a duração do pulso em estado ativo, parâmetros representados na Figura 31.

Figura 31 - Parâmetros configuráveis para configuração de sinal de saída CCP.



O período do PWM é especificado pelo registrador PR2 do Timer TMR2 e é calculado pela equação (21) (DATASHEET PIC16F87XA, 2003).

$$\text{Período PWM } (T) = [(PR2) + 1] \cdot 4 \cdot T_{osc} \cdot (\text{Valor da pré-escala de TMR2}) \quad (21)$$

O ciclo de trabalho D é especificado pelo uso de 10 bits, sendo 8 provenientes do registrador CCPR1L, que são os 8 bits menos significativos do registrador CCPR1 da Figura 29 e dois bits do registrador CCP1CON que são os bits 5 e 4 da Figura 32. Desta forma, o ciclo de trabalho é obtido pela equação (22) (DATASHEET PIC16F87XA, 2003).

$$D = (CCPR1L, CCP1X, CCP1Y) \cdot T_{osc} \cdot \text{Valor da pré-escala de TMR2} \quad (22)$$

Deve-se observar, entretanto, que se a largura de pulso (D) for configurada com um valor maior que o período do PWM, o pulso será alto durante todo o tempo.

Para uma determinada frequência de pulso, há um numero limitado de combinações de largura de pulsos. Este número é chamado de resolução e é medido em bits. No projeto atual, este valor será determinado pelo compilador.

O registrador CCP1CON controla algumas configurações do módulo CCP.

Figura 32 - Registrador CCP1CON

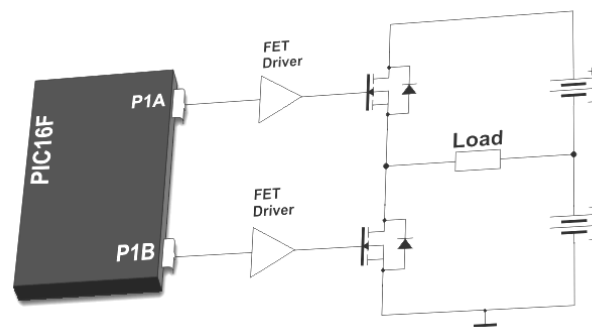
	R	R	R/W	R/W	R/W	R/W	R/W	R/W
CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Fonte: Site mikroe.com (2011).

- CCP1M3 – CCP1M0: *CCP1 Mode Select bits*
 - Define qual modo de operação do CCP1. O modo PWM é ativado com o valor binário 11xx.
- CCP1X, CCP1Y: Bits menos significantes do PWM do ciclo de trabalho.
- Bit 7,6 – Não implementados.

Assim através do gerenciamento do Timer TMR2 é feito o controle de circuitos como os de meia-ponte, no qual existem duas chaves que conduzem alternadamente e que nunca podem conduzir ao mesmo tempo.

Figura 33 - Configuração meia-ponte.



Fonte: Site mikroe.com (2011).

Para isto é necessário determinar o modo de operação do módulo CCP como PWM e os valores de frequência do sinal e tempo de ciclo ativo, cujos valores são inseridos em registradores específicos.

2.6.3 Conversor A/D

O controle da razão cíclica do circuito é feito via realimentação do valor de tensão de saída. Para a leitura desta tensão fez-se uso do conversor A/D que converte sinais analógicos dos pinos de entrada em valores digitais de 10 bits. Há oito entradas A/D no PIC16F874A de quarenta pinos que são gerenciadas por três bits do registrador ADCON0 (DATASHEET PIC16F87XA, 2003).

As tensões de entrada nas entradas analógicas devem ser de até 5 VCC, assim a conversão será proporcional a este valor, ou seja, uma tensão de 5 Volts resulta em 1024, que é o valor máximo dos 10 bits, e uma tensão de 2,5 Volts resulta em um valor igual a 512 armazenado em forma binária nos registradores de resultado da conversão.

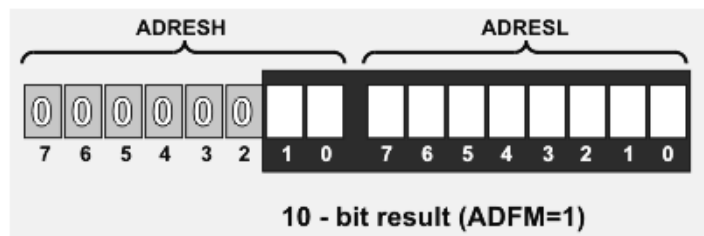
Portanto, para conversão de sinais maiores de 5 Volts é utilizado um divisor de tensão, que deve ser calculado de forma que o valor máximo da tensão a ser medida seja convertido em 5 Volts.

O módulo do conversor A/D é composto de quatro registradores, genericamente descritos a seguir:

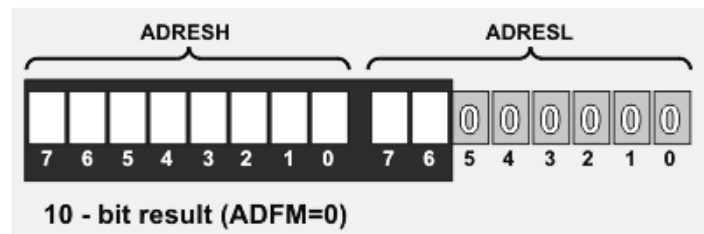
- ADRESH – Contém o byte mais alto do resultado da conversão A/D;
- ADRESL – Contém o byte mais baixo do resultado da conversão A/D;
- ADCON0 – Registrador de controle 0;
- ADCON1 – Registrador de controle 1.

Quando há a conversão de um sinal analógico para o formato digital, o resultado da conversão A/D é um número de 10 bits que é armazenado nos registradores ADRESH e ADRESL conforme Figura 34. Os dois registradores possuem um espaço disponível de 16 bits, portanto 6 bits não são utilizados. Para a manipulação deste resultado há a opção de justificar o resultado à direita ou à esquerda através do valor guardado em ADFM, o bit 7 do registrador ADCON1.

Figura 34 - Registradores A/D. a) Justificação à direita; b) Justificação à esquerda.



a)

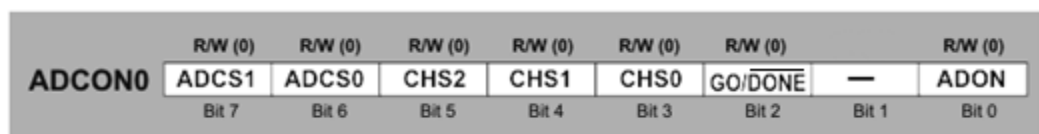


b)

Fonte: Site mikroe.com (2011).

Os registradores de controle são utilizados para gerenciar toda lógica e configuração da conversão analógico-digital e são mais apresentados de forma mais detalhada abaixo.

Figura 35 - Registrador ADCON0



Fonte: Site mikroe.com (2011).

ADSC0, ADSC1 – A/D Conversion Clock Select bits

▪ Seleciona a frequência do *clock* utilizado para sincronização interna do conversor A/D que influencia na duração da conversão.

· CHS2 – CHS0 – *Analog Channel Selected bits*

▪ Seleciona um pino ou um canal analógico para realizar a conversão.

· GO/ \overline{DONE} – *A/D Conversion Status bit*

▪ Determina o estado atual da conversão

▪ 1 – Conversão A/D em progresso.

▪ 0 – Conversão A/D está completa.

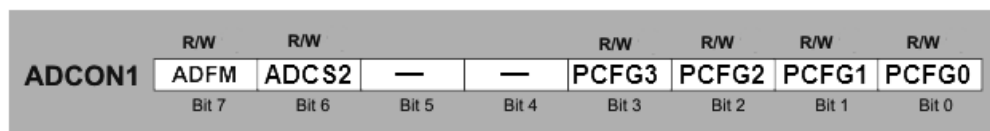
· ADON – *A/D On bit*

▪ Habilita o conversor A/D

▪ 1 – Conversão A/D habilitada.

▪ 0 – Conversão A/D desabilitada.

Figura 36 - Registrador ADCON1



Fonte: Site mikroe.com (2011).

· ADFM – *A/D Result Format Select bit*

· 1 – Justificado a direita.

· 0 – Justificado a esquerda.

· ADCS2 – *A/D Conversion Clock Select bit*

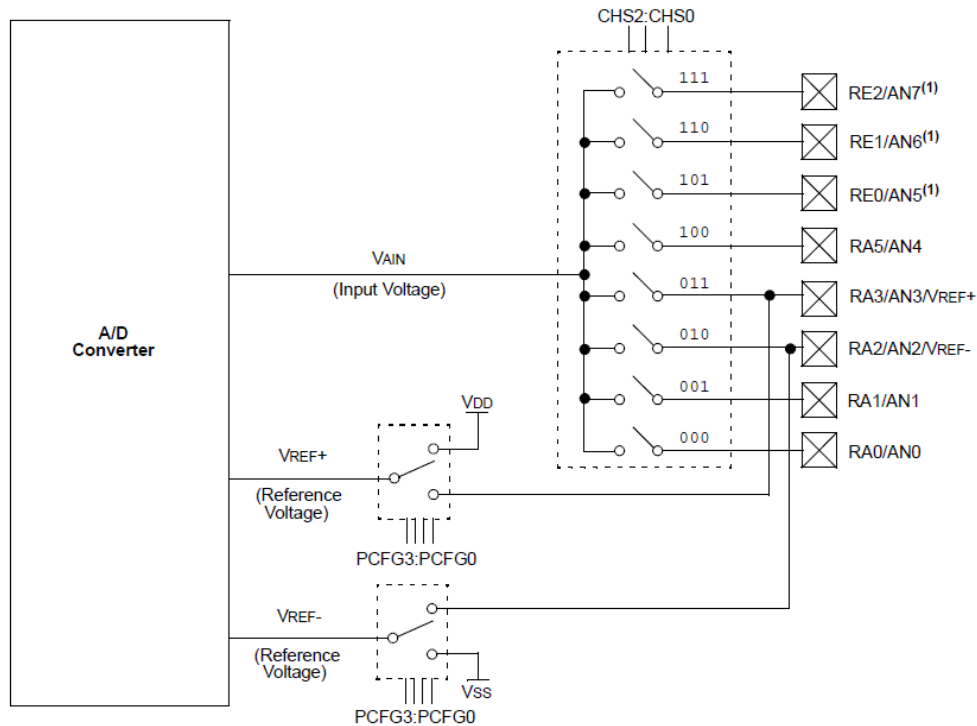
· Juntamente com o ADCON0 determina a frequência do *clock*.

· PCFG0 – PCFG3 – *A/D Port Configuration Control bits*

· Configura as portas para ler tensão com tensão de referência.

Todo este conjunto de funções e configurações é mais facilmente visualizado através do diagrama de blocos da Figura 37 que representa todas as interações do módulo A/D.

Figura 37 - Diagrama de blocos do módulo A/D.



Fonte: Data Sheet 16F87XA (2003).

Para obter valores mais precisos do módulo A/D, é necessário certo tempo de *delay* entre a seleção de uma entrada analógica específica e a conversão em si. Este tempo é chamado de tempo de aquisição, portanto é recomendado esperar um intervalo de 10 μ s antes de ler o resultado da conversão (PEREIRA, 2003).

2.6.4 Compilador

O compilador é um programa capaz de traduzir um programa escrito em alto nível para um programa equivalente em código de máquina próprio para um processador. Neste trabalho foi utilizada a linguagem C, uma linguagem estruturada que não é considerada linguagem alto nível, porém trabalha em nível mais alto do que a linguagem *assembly*, ou linguagem de máquina.

O software utilizado para escrever e compilar o código é chamado de *PCWH Compiler* em sua versão 3.236. Ele possui funcionalidades muito práticas e úteis que facilitam a criação do programa. Uma delas é o *Setup Wizard* que dá a possibilidade de configurar inúmeros parâmetros antes mesmo do início da programação e mostra uma previsão do código a cada opção selecionada.

Nesta etapa inicial é possível escolher o microcontrolador a ser trabalhado, o *clock* do oscilador externo, configuração de *timers* e suas interrupções e definição de parâmetros de PWM e do conversor A/D. Entretanto, estas funcionalidades não eliminam a necessidade do conhecimento do PIC e suas características conforme apresentadas neste trabalho.

O compilador conta também com uma biblioteca de funções que deixam a programação em um nível bem mais alto do que se fosse optado pela programação em *assembly*. Estas funções foram utilizadas sempre que possível, deixando o código muito mais limpo, organizado e de fácil atualização para aplicações futuras ou incremento de elementos.

Abaixo seguem alguns exemplos de funções utilizadas neste projeto, que serão mais profundamente detalhadas no projeto do controle.

- `set_pwm_duty ();`
- `delay_us ();`
- `enable_interrupts ();`
- `setup_timer_2 ();`

Estas funções estão diretamente ligadas aos registradores de cada módulo a que as funções se referem e realizam as operações em alto nível.

CAPÍTULO 3

PROJETO

O projeto do *Push-Pull* inversor foi realizado em duas etapas. Primeiro será apresentado o projeto do circuito de potência, que trata do *Push-Pull* em sua forma inversora, conforme proposto na parte teórica deste trabalho. A segunda parte contém o projeto de controle das chaves do inversor, que foi baseado nas características disponíveis no PIC16F874A para geração de um sinal SPWM e monitoramento do nível de tensão de saída.

3.1 PUSH-PULL

O projeto se baseia na utilização de uma bateria de 12 *Volts* como fonte de alimentação *CC* do circuito, portanto as tensões de entrada máxima e mínimas foram definidas de forma que preservem a bateria. A potência foi baseada no consumo de um motor elétrico *CA* comum, como os presentes em ventiladores domésticos e que consome cerca de 1A e a tensão de saída foi projetada para atender seu funcionamento, ou seja, uma tensão 127VAC RMS em 60Hz.

A sequência deste projeto será baseada no formato apresentado por Ferreira (2006).

Dito isto, os parâmetros base que norteiam o projeto são os que se encontram na Tabela 3.

Tabela 3- Parâmetros de projeto do circuito de potência do inversor.

Parâmetros de Projeto	Valores
Tensão máxima de entrada ($V_{E(máx)}$)	12,75V
Tensão mínima de entrada ($V_{E(mín)}$)	11,5V
Tensão de saída (V_S)	180VAC
Tensão sobre diodo (V_D)	0,78V
Tensão coletor-emissor em transistores saturados ($V_{CE(SAT)}$)	0V
Frequência de chaveamento (f_S)	50kHz
Razão cíclica máxima ($D_{máx}$)	0,45
Densidade de campo magnético máxima ($B_{MÁX}$)	0,3T

3.1.1 Relação de Transformação

A relação de transformação entre as bobinas do transformador é obtida pela equação (1):

$$V_S = \frac{2 \cdot D_{\max} \cdot (V_{E(\min)} - V_{CE(SAT)})}{N} - V_D = 180 = \frac{2 \cdot 0,45 \cdot (11,5 - 0)}{N} - 0,78 \quad (23)$$

$$N = 0,0577 \cong 0,06 \quad (24)$$

Assim, pelo resultado em (24) a bobina no secundário precisa ter cerca de 17 vezes mais espiras que no primário para que a elevação da tensão de 12 Volts da entrada seja convertida em 180 Volts na saída do transformador.

3.1.2 Transformador

Após a relação de transformação do transformador é necessário calcular o campo magnético no núcleo, cujo valor é o resultado da equação (3).

$$B = \frac{V_{E(\min)}}{V_{E(\max)}} \cdot B_{\max} = \frac{11,5}{12,75} \cdot (0,3 \cdot 2) \cong 0,54T \quad (25)$$

O parâmetro B_{\max} é multiplicado por dois, pois segundo Treviso (2009, p. 42) “[...] a variação total permissível da intensidade do campo magnético B pode ser o dobro da variação total permissível do conversor *Forward*.” conforme visto anteriormente na Figura 6.

O tipo de núcleo é escolhido através do cálculo de A_P que é obtido pela equação (7).

$$A_P = \left(\frac{3,98 \cdot P_S \cdot 10^4}{K_f \cdot B \cdot f} \right)^Z = \left(\frac{3,98 \cdot 216 \cdot 10^4}{397 \cdot 0,54 \cdot 50k} \right)^{1,136} = 0,7783cm^4 \quad (26)$$

O núcleo recomendado para este projeto é o EE 30/15/14, que é obtido pelo Anexo B através do resultado de (26). Suas características são apresentadas na Tabela 4.

Tabela 4 - Parâmetros do Transformador

Parâmetros	Valores
A_P	1,43 cm ⁴
A_E	1,2 . 10 ⁻⁴ m
L_E	6,79 . 10 ⁻² m

Os valores da Tabela 4 são utilizados para o cálculo do número de espiras nas bobinas do transformador através da equação (8).

$$N_1 \geq \frac{V_{E(mín)} \cdot D_{máx}}{A_e \cdot B \cdot f} = \frac{11,5 \cdot 0,45}{1,2 \cdot 10^{-4} \cdot 0,54 \cdot 50 \cdot 10^3} \cong 1,597 \quad (27)$$

Pelo resultado em (27) escolhe-se o valor de três espiras para o enrolamento primário do transformador, valor que é utilizado para encontrar o número de espiras no secundário através da relação de transformação (2).

$$N_2 = \frac{N_1}{N} = \frac{3}{0,06} \cong 50 \text{ espiras} \quad (28)$$

A seguir é calculado o valor da densidade de corrente J e a corrente RMS $I_{RMS(SEC)}$ no secundário do transformador que são obtidas através das equações (10) e (11), respectivamente.

$$J = K_J \cdot A_P^{-x} = 397 \cdot 1,43^{-0,12} = 380,05 \text{ A/cm}^2 \quad (29)$$

Para o cálculo da corrente RMS, além da densidade de corrente obtida em (30) é preciso determinar a corrente de saída do circuito I_S que é obtida por (17).

$$I_S = \frac{P_S}{V_S} = \frac{216}{180} = 1,2 \text{ A} \quad (30)$$

O valor de $I_{RMS(SEC)}$ é calculado a seguir.

$$I_{RMS(SEC)} = I_S \cdot \sqrt{2 \cdot D_{máx}} = 1,2 \cdot \sqrt{2 \cdot 0,45} \cong 1,14 \text{ A} \quad (31)$$

Com a relação da corrente RMS do secundário e da densidade de corrente calcula-se a área de cobre necessária para a condução desta corrente, como apresentado em (9).

$$A_{CU} = \frac{I_{RMS(SEC)}}{J} = \frac{1,14}{380,05} = 0,002996 \text{ cm}^2 \quad (32)$$

O cálculo da área de cobre do primário é calculado pela mesma relação do enrolamento secundário. Para o cálculo da corrente RMS no primário basta considerar a relação de espiras N entre os enrolamentos, a partir da equação (13).

$$I_{RMS(PRI)} = \frac{I_{RMS}}{2 \cdot N} = \frac{1,14}{2 \cdot 0,45} = 9,5 \text{ A} \quad (33)$$

E por fim, a corrente RMS no primário do transformador é obtida por (12):

$$A_{CU} = \frac{I_{RMS(PRI)}}{J} = \frac{9,5}{380,05} = 0,024997 \text{ cm}^2 \quad (34)$$

Com os valores da área de cobre mínima em cada um dos enrolamentos obtidos em (32) e (34) é escolhido o melhor condutor para o circuito conforme Anexo C. Desta forma para o enrolamento primário utiliza-se o fio # 13 AWG que possui área de cobre igual a $0,0262243 \text{ cm}^2$ e para o enrolamento secundário a melhor opção é o fio # 22 AWG que tem $0,003255 \text{ cm}^2$ em área de cobre.

O fio do enrolamento primário tem quase o triplo do diâmetro do secundário devido à diferença de corrente entre as duas malhas do circuito. Para evitar o trabalho com uma bitola muito grossa, que dificulta o manuseio do fio, é comum a escolha de fios de diâmetros menores que correspondam à bitola original. Neste caso podem ser utilizados 3 fios # 22 AWG, que facilita a montagem do circuito e auxilia contra o efeito Skin (TREVISIO, 2009). Como verificado, a tabela AWG com os parâmetros dos fios de diversas espessuras possibilita escolher o melhor condutor para o circuito assegurando que ele será suficiente para conduzir a corrente sem que o fio seja danificado e ao mesmo tempo evita que sejam escolhidas bitolas além do necessário na implementação de circuitos, que resulta em uma economia considerável para trabalhos em alta potência.

3.1.3 Transistores

Na topologia *Push-Pull* os transistores ficam submetidos ao dobro da tensão de alimentação do circuito e a corrente eficaz durante a saturação é obtida conforme equação (15).

$$I_{EFICAZ} = \frac{I_S \cdot \sqrt{2 \cdot D_{máx}}}{2 \cdot N} = \frac{1,2 \cdot \sqrt{2 \cdot 0,45}}{2 \cdot 0,06} = 9,49 \text{ A} \quad (35)$$

E a corrente média nas chaves do circuito é o resultado de (16).

$$I_{MÉDIA} = \frac{I_S \cdot D_{máx}}{N} = \frac{1,2 \cdot 0,45}{0,06} = 9 \text{ A} \quad (36)$$

E finalmente, a tensão máxima sobre os transistores é obtida pela equação (14).

$$V_{CE(máx)} = 2 \cdot V_{E(máx)} = 2 \cdot 12,75 = 25,5 \text{ V} \quad (37)$$

A realimentação do circuito é feito através de um divisor de tensão, visto que a entrada conversora A/D do microcontrolador aceita tensões máximas de 5V.

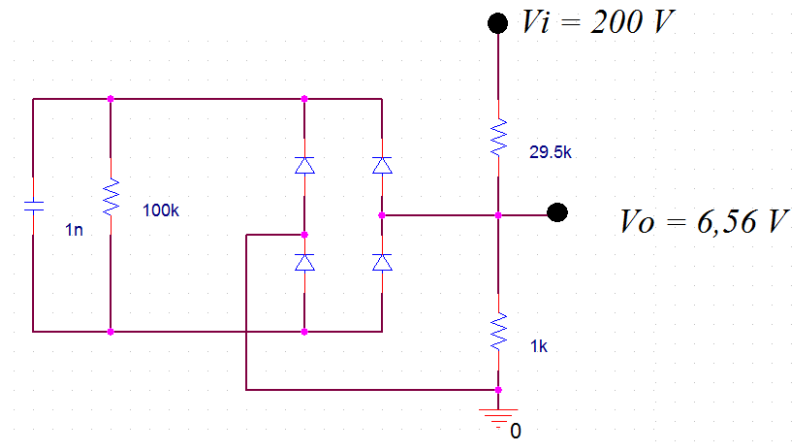
O circuito de realimentação prevê também um retificador de onda completa, utilizando uma ponte de quatro diodos, portanto o divisor de tensão precisa prever a queda de tensão em cada diodo durante a retificação. Sabe-se que a cada semi-ciclo da onda, dois diodos são utilizados para retificá-la, portanto há uma queda de tensão igual a 1,56 V durante todo o processo. Portanto esta tensão será somada ao valor desejado de saída, 5V, para que seja compensado posteriormente.

A seguir o equacionamento do divisor de tensão:

$$V_0 = \frac{V_i \cdot R_0}{R_0 + R_i} = 6,56 = \frac{200 \cdot R_0}{R_0 + R_i} \quad (38)$$

Adotando-se $R_0 = 1 \text{ K}\Omega$, obtemos $R_i = 29,5 \text{ K}\Omega$, conforme mostrado na Figura 38.

Figura 38 - Retificador e detector de pico



O circuito ainda conta com um capacitor e resistor em série para detecção de pico da tensão de entrada no retificador. Desta forma há uma conversão na tensão da carga que a torna passível de utilização pelo microcontrolador.

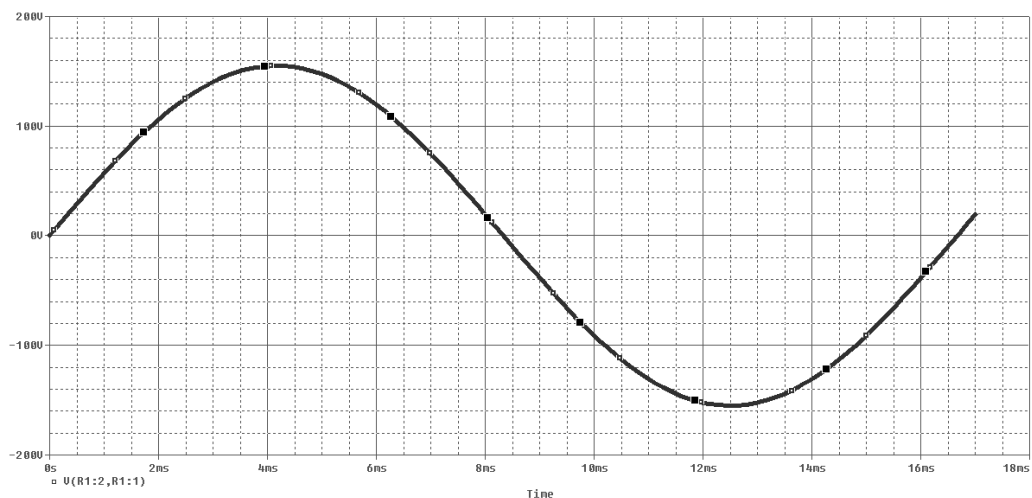
3.2 CONTROLE SPWM

O objetivo deste projeto é fornecer uma tensão alternada na saída que seja equivalente a tensão doméstica fornecida por concessionárias de energia, que são 127 Volts RMS alternados senoidalmente em frequência de 60 Hz. A amplificação da tensão de entrada fica por conta do circuito de potência do *Push-Pull* como apresentado anteriormente. A formatação da onda de saída e sua frequência são feitas pelo microcontrolador PIC16F874 através dos módulos do dispositivo e do compilador utilizado.

3.2.1 A Onda Senoidal 60Hz

A programação do microcontrolador norteou-se nas características da onda senoidal de 60Hz (Figura 39), visando gerar um trem de pulsos de largura variável cuja potência média de tensão entregue à carga seja a mais próxima possível do padrão senoidal.

Figura 39 - Onda senoidal a 60 Hz.



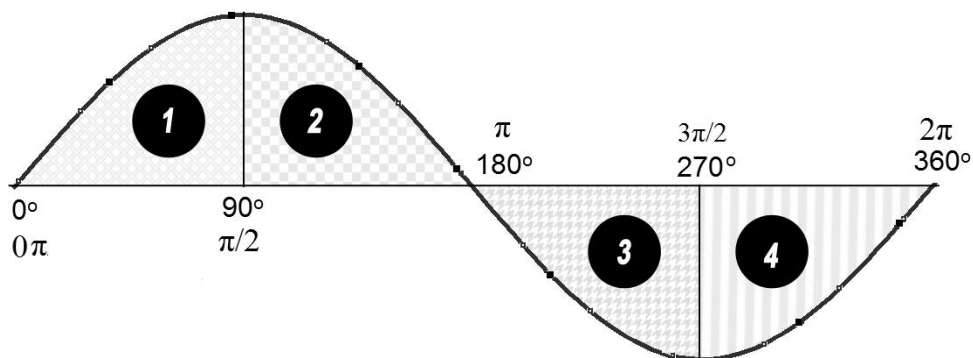
O projeto de geração de um sinal PWM é feito a partir da análise do período da senóide em 60 Hertz conforme Tabela 5.

Tabela 5 - Dados do período da onda.

Período T da Senóide 60 Hz	Tempo
T	16,67 ms
$T/2$	8,33 ms
$T/4$	4,16 μ s

A forma de onda senoidal tem um padrão que se repete quatro vezes durante seu período, conforme mostrado na Figura 40.

Figura 40 - Etapas periódicas do sinal senoidal.



Nestas quatro etapas, desconsiderando o sentido da amplitude, a mesma potência é entregue a carga visto que a mesma forma de onda se repete a cada 90° ($\pi/2$) do período total. Portanto o projeto buscou recriar os pontos de apenas uma destas etapas, bastando apenas replicar o mesmo comportamento para as demais fases. Isto reduz o número de dados introduzidos na memória do microcontrolador, gerando assim um melhor rendimento e fluidez para o código. A inversão da tensão no semi-ciclo negativo da onda senoidal $[\pi - 2\pi]$ fica a cargo do circuito do transformador do *Push-Pull* que possui uma das bobinas do primário montada inversamente.

A onda SPWM é gerada a partir do intervalo de 0° a 90° , ou seja, o sinal parte da menor largura de pulso e cresce em escala senoidal até sua maior razão cíclica t_{on} que coincide com o pico da onda.

3.2.2 Requisitos de Chaveamento

O sinal gerado pelo controle funciona como comando liga e desliga sobre os transistores *TBJ*, que passam a atuar em saturação e corte, respectivamente. Isto habilita a amplificação da tensão em cada bobina no primário do transformador do *Push-Pull*.

Para garantir o ganho de corrente nos transistores bipolares, é utilizada a configuração *darlington*, que consiste no acoplamento de dois transistores *TBJ* do mesmo tipo, de forma que o ganho de corrente final é o produto dos ganhos individuais β_1 e β_2 dos transistores. Esta configuração possibilita ganho suficiente para a condução da corrente de saída especificada no projeto do *Push-Pull*.

Como é comum durante desenvolvimento de *software*, é necessário levantar os requisitos que o código deve cobrir. Estes requisitos são propriedades ou restrições que o programa deve seguir para que o sistema funcione conforme o esperado.

Duas restrições, provenientes da topologia do circuito de potência, são impostas no chaveamento do circuito. A primeira delas refere-se à proibição de condução simultânea das chaves, que pode causar curto-circuito na fonte de alimentação do inversor.

Este requisito é coberto pelo gerenciamento dos módulos PWM do microcontrolador. Sempre que um módulo estiver ativo o outro deve ser desligado e para aumentar esta segurança, um tempo morto é previsto, ou seja, deve haver um pequeno tempo entre chaves em que nenhuma delas conduz, o que elimina o risco de condução simultânea.

O segundo requisito baseia-se no tempo de magnetização do transformador no circuito de potência e determinam que os pulsos em alta frequência gerados pelo microcontrolador não podem ter largura maior que 50% do comprimento total do pulso. Um ciclo de trabalho maior do que este faz com o que o transformador não tenha tempo hábil de desmagnetização resultando em sua saturação.

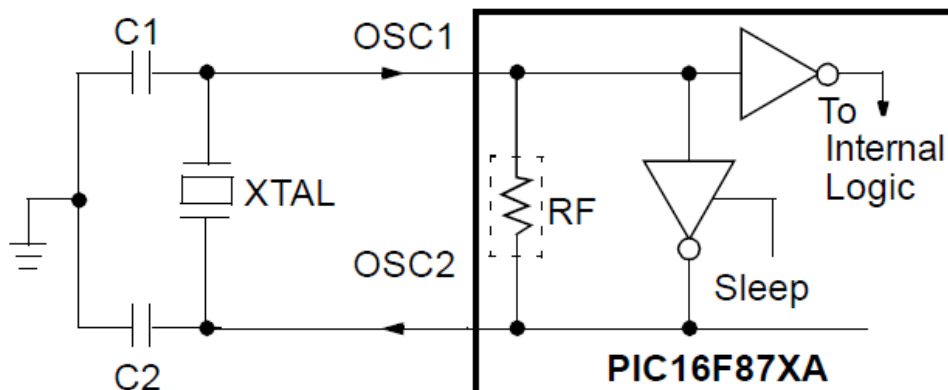
3.2.3 Programação do Pic

A programação do PIC16F874A foi realizada no programa PCWH da CCS, através do PIC Wizard, que funciona como um gerenciador de módulos para a inicialização do projeto.

Os primeiros parâmetros a serem definidos são o componente a ser utilizado, no caso, o microprocessador do projeto é o PIC17F874A, e a frequência de oscilação. Esta frequência é referente ao oscilador externo do circuito que é composto por um cristal e dois capacitores configurados para oscilarem na frequência do cristal.

O circuito oscilador é montado conforme especificado na documentação técnica do microcontrolador (DATASHEET PIC16F87XA, 2003) e mostrado na Figura 41, no qual a resistência RF varia conforme o tipo do cristal utilizado. Para este projeto foram utilizados um cristal *HS* (*High Speed Crystal Resonator*) que opera a 24 MHz e dois capacitores cerâmicos de 33 pF cada um.

Figura 41 - Circuito ressonante.



Fonte: Data Sheet 16F87XA (2003).

Para frequências de oscilação maiores que 4MHz é preciso habilitar as configurações de alta velocidade de oscilação. É possível ainda habilitar funções de proteção de leitura de código, temporização para inicialização, e proteção de dados da memória EEPROM.

Estas configurações são escritas no arquivo de cabeçalho (*header*) criado pelo compilador, que contem os protótipos de funções utilizados no código fonte do programa. Estes arquivos são comumente nomeados com a extensão *.h* e são incluídos nos programas pela diretiva *#include* seguido pelo nome do arquivo cabeçalho. As configurações iniciais citadas são inseridas nos arquivos conforme apresentado e comentado na Tabela 6 abaixo:

Tabela 6 - Arquivo de cabeçalho (*header*).

Configurações Inseridas no arquivo de cabeçalho (<i>header .h</i>)	
<i>Linhas de código</i>	<i>Comentário</i>
<i>#include <16F874A.h></i>	Adicional biblioteca do dispositivo escolhido
<i>#device adc=8</i>	Número de conversores A/D do micro
<i>#FUSES NOWDT</i>	Não utiliza o <i>Watch Dog Timer</i> (WDT)
<i>#FUSES HS</i>	Oscilador de alta velocidade (> 4 <i>MHz</i>)
<i>#FUSES NOPUT</i>	Sem tempo de inicialização
<i>#FUSES NOPROTECT</i>	Código não protegido de leitura
<i>#FUSES NODEBUG</i>	Modo debug desabilitado
<i>#FUSES LVP</i>	Programação em baixa voltagem (PIC16)
<i>#FUSES NOCPD</i>	Desabilita proteção de código
<i>#FUSES NOWRT</i>	Sem proteção de escrita na memória
<i>#FUSES BROWNOUT</i>	Proteção de sobrecarga do micro habilitada
<i>#use delay(clock=24000000)</i>	Frequência do oscilador externo.

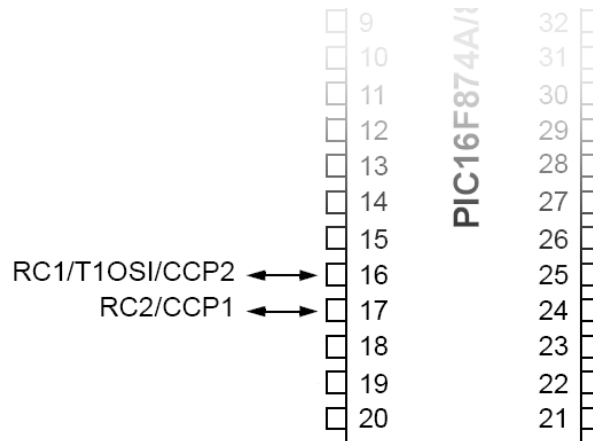
Os parâmetros *FUSES* configurados no cabeçalho são registradores de 3bytes que ditam o comportamento do chip, no que diz respeito ao sistema de *hardware* e apesar da tradução para o português significar fusíveis, eles nada têm a ver com proteção de sobrecorrente no circuito.

Conforme estipulado em projeto, o controle dos transistores é realizado em uma frequência igual a 50 *kHz*, portanto o microcontrolador deve gerar um sinal com período de 20 μs em cada uma das saídas que controlam a tensão nas bobinas do primário do transformador.

Porém não basta gerar um sinal na frequência determinada, é preciso que a razão cíclica seja variável, portanto não basta criar uma rotina em um pino de saída que gere a frequência desejada. Para suprir esta necessidade foram utilizados os módulos CCP1 e CCP2 do microcontrolador, responsáveis pelos modos Comparação, Captura e PWM.

Estes módulos estão localizados nos pinos 17 e 16 do microcontrolador conforme Figura 42. São estas duas saídas que são conectadas na base de cada transistor do inversor.

Figura 42 - Pinos de saída dos módulos CCP.



Fonte: Data Sheet 16F87XA (2003).

Para gerar a frequência desejada é preciso fazer uso da equação (21) reapresentada abaixo (DATASHEET PIC16F87XA, 2003)..

$$\text{Período PWM } (T) = [(PR2) + 1] \cdot 4 \cdot T_{osc} \cdot (\text{Valor da pré-escala de TMR2}) \quad (21)$$

Para o equacionamento do registrador PR2 que compõe os módulos CCP foram considerados os parâmetros abaixo listados:

- Frequência do PWM (f_s) = 50 kHz;
- Período do PWM (T_{PWM}) = 20 μ s;
- Frequência de oscilação (f_{osc}) = 24 MHz (oscilador externo – cristal HS);
- Período de oscilação (T_{osc}) \cong 41,67 ns;
- Valor da pré-escala de TMR2 = 1.

As opções de pré-escala para o Timer TMR2 são 1, 4 e 16. A escolha destes parâmetros depende da faixa de operação que o projetista precisa.

Portanto, atribuindo os valores listados acima na equação (21) encontra-se o valor a ser escrito no registrador PR2.

$$20 \cdot 10^{-6} = [(PR2) + 1] \cdot 4 \cdot 41,67 \cdot 10^{-9} \cdot 1 \quad (39)$$

Utilizando o valor do período de oscilação (T_{osc}) com a dízima periódica obtida pelo inverso da frequência de oscilação (f_{osc}), o registrador PR2 deve assumir o resultado de (39) mostrado abaixo.

$$PR2 = 120 - 1 = 119 \quad (40)$$

Portanto, para a configuração proposta até então, é preciso armazenar o valor 119 ou o *byte* 01110111_b no registrador descrito.

Normalmente esta operação é feita em linguagem de baixo nível e é necessário realizar a operação passo a passo em cada registrador. Mesmo se feita em linguagem C é necessário configurar primeiro o registrador T2CON e depois atribuir valores para PR2 e TMR2 que definem o final e início da temporização, respectivamente.

Porém, o compilador PCWH possui diversos comandos de alto nível que facilitam à escrita e atualização do código. A configuração do Timer TMR2 é feita através de uma função específica, já implementada pelo compilador e que possui a seguinte sintaxe:

```
setup_timer_2 (mode, period, postscale);
```

Nesta função os parâmetros podem assumir os seguintes estados:

- *mode*: Refere-se ao valor da pré-escala que divide o período do oscilador e pode assumir os valores seguintes: T2_DISABLED, T2_DIV_BY_1, T2_DIV_BY_4, T2_DIV_BY_16;
- *period*: pode assumir um valor de 0-255 e determina quando o valor do *timer* é resetado. Este é o valor que será inserido em PR2;
- *postscaler*: é um número de 1-16 que determina quantos resets devem ocorrer antes de apontar um evento de interrupção.

Portanto, com os valores calculados anteriormente, esta configuração é feita de maneira direta como mostrado a seguir:

```
setup_timer_2 (T2_DIV_BY_1,119,1);
```

A mesma configuração feita em linguagem C sem o auxílio desta função ficaria conforme o código seguinte:

Tabela 7 - Configuração do Timer 2.

Configuração do Timer TMR2	
<i>Linhas de código</i>	<i>Comentário</i>
<i>T2CON=0b00001101;</i>	Configura registrador T2CON
<i>TMR2=0;</i>	Zera o registrador TMR2
<i>PR2=0x77</i>	Atribui o valor calculado em PR2 (119 _d ou 0x77 _h)
<i>while(1)</i>	Gera um loop.
<i>{</i>	
<i>if (TMR2IF !=0);</i>	Verifica se <i>flag</i> da interrupção foi ativada
<i>{</i>	
<i>Função desejada</i>	Realiza tarefa
<i>TMR2IF = 0;</i>	Zera a <i>flag</i> e espera por outro evento
<i>}</i>	
<i>}</i>	

Portanto fica clara a funcionalidade das funções do compilador. Neste caso, esta operação de espera e verificação dos registradores é feita de maneira automática e paralela com o código principal.

Com o Timer 2 configurado, um sinal da frequência será gerado nos módulos CCP assim que os habilitarmos. A configuração adotada será para o modo PWM, portanto para ativar os pinos 17 e 16 no modo CCP1 PWM e CCP2 PWM, respectivamente, utiliza-se da função para este propósito:

```
setup_ccp1(CCP_PWM);
setup_ccp2(CCP_PWM);
```

Mais uma vez, as funcionalidades das funções implementadas no compilador se mostram muito úteis e diretas. Feito isto, as duas saídas irão gerar um pulso de 50 *kHz* de modo contínuo.

Agora é necessário ajustar a largura deste pulso para que seja possível gerar a onda SPWM. Este ajuste é feito por uma função específica que escreve no registrador dos dois módulos CCP, CCPR1L e CCPR2L, o valor de 10 bits que gera o ciclo de trabalho do PWM. Caso os dois bits menos significantes do registrador CCP1CON não sejam inseridos, é feito um deslocamento do byte, e os bits 4 e 5 do registrador CCP1CON são preenchidos com zero, preservando assim o valor de 10 bits que então é utilizado para determinar a quantidade de tempo que dura o sinal nível lógico alto durante o período da onda.

A sintaxe desta função para os módulos CCP1 e CCP2 respectivamente, são:

```
set_pwm1_duty (value);
set_pwm2_duty (value);
```

Sendo que o dado de entrada (*value*) da função é obtido através da equação a seguir (PCWH, 2003):

$$Duty (value) = \frac{T_{on}}{pré - escala TMR2 \cdot T_{osc}} \quad (41)$$

Portando, por (41) a razão cíclica máxima para a frequência de 50 kHz é obtida considerando-se os seguintes parâmetros:

- $T_{on} = 20 \mu s$, para $D_{on} = 100\%$;
- $T_{OSC} = 1/24.000.000$;
- $pré-escala TMR2 = 1$.

$$Duty (value) = \frac{20 \cdot 10^{-6}}{1 \cdot \frac{1}{24.000.000}} = 480 \quad (42)$$

O valor obtido em (42) fornece a razão cíclica máxima na saída dos módulos CCPx. Este valor é diretamente proporcional ao ciclo de trabalho, portanto diferentes razões cíclicas podem ser obtidas através de regra de três simples, tomando este valor como referência. Portanto caso seja necessário manter o pino 17 do circuito sempre em nível lógico alto e nível lógico baixo no pino 16, basta utilizarmos esta função da seguinte forma:

```
set_pwm1_duty (480);
set_pwm2_duty (0);
```

Antes, porém de programar o código que gera o sinal SPWM, é preciso considerar os requisitos de projeto. Como dito anteriormente, o tempo em nível alto do sinal PWM não pode ultrapassar a metade do período da onda, para evitar saturação do transformador. Foi definido em projeto que para garantir que isto não ocorra, a razão cíclica máxima deve ser igual a 45%, ou em termos numéricos, ter no máximo 9μs de duração.

Portanto, por matemática simples, o valor máximo que pode ser utilizado na função *set_pwm_x_duty()* é:

$$D_{m\acute{a}x} = 45\% = 480 \cdot 0,45 = 216 \quad (43)$$

Desta forma, este é o valor máximo que será utilizado neste projeto e equivale ao momento em que a senóide transmite sua maior potência, que corresponde a 90° na escala trigonométrica. Os demais pulsos deverão possuir valores menores que o encontrado em (43), o que garante que o requisito do projeto será totalmente atendido.

Esta variação da largura de pulso até $\frac{1}{4}$ da senóide não pode ser feita de maneira linear, pois isto geraria uma onda triangular, na qual a potência aumenta linearmente até seu pico. Este aumento deve ser baseado no comportamento da função seno, para que após a filtragem do sinal, o trem de pulsos gerados se aproxime o máximo possível do formato de onda esperado.

A precisão na forma de onda depende da frequência em que a largura de pulso é variada. Quanto mais frequente for o incremento da largura de pulso, mais próximo à função seno será a onda. Neste projeto, o incremento acontece a cada 3° dentro de um intervalo igual a $4,16 \mu s$, equivalente a $\frac{1}{4}$ do período da senóide em 60 Hz. Portanto há 31 estágios de razão cíclica diferentes de 0° a 90° .

Desta forma, os valores de razão cíclica no primeiro quarto de onda são feitos a partir dos valores obtidos na equação abaixo:

$$D_n = \sum_{n=0}^{n=30} [\text{sen}(3 \cdot n)] \cdot 216 \quad (44)$$

A equação (44) calcula a razão cíclica a partir do seno do ângulo n , em graus, e multiplica pelo valor máximo do ciclo de trabalho, que acontece quando o ângulo é 90° , ou seja, quando $n = 30$.

Este procedimento vale para a maior razão cíclica possível neste projeto, ou seja, um sinal SPWM com valor máximo de ciclo de trabalho $D_{on} = 0,45$ ou $t_{on} = 9 \mu s$.

Os valores obtidos pela equação (44) não geram necessariamente valores inteiros, portanto é realizado um arredondamento através da função de arredondamento (ARRED(numero, num_dígitos)) do aplicativo *Microsoft Excel*. O resultado final utilizado na função *set_pwm_x_duty(valor)* é o valor D_n da última coluna na Tabela 8.

Tabela 8 - Vetor de largura de pulso SPWM.

<i>n</i>	Angulo θ	Seno (θ)	[Seno (θ)] . 216	Dmáx 45%
				Dn
0	0	0	0	0
1	3	0,052336	11,30456655	11
2	6	0,104528	22,57814807	23
3	9	0,156434	33,78984445	34
4	12	0,207912	44,90892522	45
5	15	0,258819	55,90491374	56
6	18	0,309017	66,74767078	67
7	21	0,358368	77,4074771	77
8	24	0,406737	87,8551149	88
9	27	0,45399	98,06194794	98
10	30	0,5	108	108
11	33	0,544639	117,6420316	118
12	36	0,587785	126,9616145	127
13	39	0,62932	135,9332045	136
14	42	0,669131	144,532211	145
15	45	0,707107	152,7350647	153
16	48	0,743145	160,5192823	161
17	51	0,777146	167,8635277	168
18	54	0,809017	174,7476708	175
19	57	0,838671	181,1528427	181
20	60	0,866025	187,0614872	187
21	63	0,891007	192,4574092	192
22	66	0,913545	197,3258189	197
23	69	0,93358	201,6533721	202
24	72	0,951057	205,4282075	205
25	75	0,965926	208,6399785	209
26	78	0,978148	211,2798818	211
27	81	0,987688	213,3406816	213
28	84	0,994522	214,8167294	215
29	87	0,99863	215,7039795	216
30	90	1	216	216

Com os valores da razão cíclica definidos, é preciso definir o intervalo em que o incremento destes valores ocorrerá. Isto é feito através da interrupção do Timer 0, que precisa ser configurado conforme a necessidade do projeto.

Primeiramente divide-se o intervalo da onda utilizada pelo número de pontos em que ocorrerá a variação do ciclo de trabalho. Como dito anteriormente, o projeto esta sendo realizado sobre a função $\text{seno}(x)$ para $x \in [0^\circ, 90^\circ]$ na frequência de 60 Hz. Este intervalo corresponde a $4,16 \mu\text{s}$, conforme Tabela 8.

Portanto temos o seguinte intervalo T_D a ser considerado:

$$T_D = \frac{T_{\text{seno } 60\text{Hz}} \cdot \frac{1}{4}}{31} = \frac{16,67 \cdot 10^{-3} \cdot 0,25}{31} \cong 134,4 \cdot 10^{-6} \text{ s} \quad (45)$$

A configuração do Timer 0 é feita da seguinte forma:

```
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
```

O primeiro parâmetro (RTCC_INTERNAL) faz com que o Timer 0 esteja configurado para atuar com o *clock* interno do microprocessador que atua em 6 MHz. O segundo (RTCC_DIV_1) ajusta a pré-escala para 1.

Segundo (SILVA, 2006), a tempo t_{TMR0} que Timer 0 demora para chegar ao seu valor máximo é obtido pela equação (46).

$$t_{TMR0} = \frac{1}{\text{clock}} \cdot 256 \cdot \text{pré-escala} \quad (46)$$

Considerando os valores descritos acima e a equação (46).

$$t_{TMR0} = \frac{1}{6 \cdot 10^6} \cdot 256 \cdot 1 \cong 42,6 \cdot 10^{-6} \text{ s} \quad (47)$$

O compilador PCWH possui uma configuração específica para interrupções de Timer 0. Primeiramente é preciso ativa-las conforme a função abaixo:

```
enable_interrupts(INT_TIMER0);
```

Feito isto, ativa-se a utilização da função de interrupção de Timer 0 que gera o código seguinte:

```
#int_TIMER0
TIMER0_isr()
{
}
```

Esta configuração faz com que a cada *overflow* do Timer 0 o microcontrolador gere uma interrupção e entre na função *TIMER0_isr()*. Isto ocorre porque esta função faz automaticamente a verificação da *flag* TMR0F de forma paralela a execução do código.

Com esta configuração, nota-se que o valor de do Timer 0 obtido em (47) é bem menor do que o valor encontrado em (45), então se utiliza um contador que faz com que esta interrupção ocorra três vezes para então a razão cíclica ser incrementada. Esta opção resulta em um novo intervalo T_D igual a 127,8 μ s, porém ainda não atinge o valor de (45).

Esta diferença entre o período T_D calculado anteriormente e o novo T_D descrito acima acontece durante os 31 intervalos definidos resultando em um tempo total de:

$$T_M = (134,4 \cdot 10^{-6} - 127,8 \cdot 10^{-6}) \cdot 31 = 204,6 \cdot 10^{-6} \text{ s} \quad (48)$$

Este tempo excedente acaba sendo totalmente funcional para o projeto, já que um requisito de chaveamento é existir um tempo em que nenhuma das chaves conduz, portanto este tempo será manipulado para que seja contabilizado como tempo morto (T_M) do sistema. Este tempo ainda será utilizado para compensar o tempo de processamento do código e da leitura da entrada A/D que requer um tempo de leitura, como será mostrado a seguir.

O conversor A/D realiza a leitura do divisor de tensão do circuito para ajustar a largura de pulso máxima conforme a tensão de saída. Esta etapa equivale ao sinal de tensão referência da geração do SPWM, como explicitado nos apontamentos teóricos. Assim, o circuito é realimentado e faz uma medição constante da tensão de saída, alterando para mais ou para menos a largura referência do pulso, a fim de manter a tensão na saída constante.

O módulo A/D é ativado por uma função específica que ajusta as portas indicadas como portas conversoras e por outra que indica qual será a referência de frequência do conversor. As funções abaixo fazem com que as portas AN0 a AN4 sejam habilitadas como conversores A/D e ajusta a frequência de funcionamento de acordo com o *clock* interno do microcontrolador.

```
setup_adc_ports(AN0_AN1_AN2_AN3_AN4);
setup_adc(ADC_CLOCK_INTERNAL);
```

Existe uma metodologia específica para a leitura dos canais A/D, pois a leitura não é feita instantaneamente, já que ela depende de carregamento de capacitores. Portanto antes de fazer a atribuição de variáveis após a leitura do canal, é preciso esperar um período de tempo para assegurar que a leitura será realizada corretamente. Segundo dados fornecidos pelo compilador, para este projeto o tempo de leitura varia entre 2 μ s e 6 μ s, porém para assegurar a

precisão utilizou-se de um tempo de 10 μs durante a leitura, que é feita da seguinte maneira (PEREIRA, 2003):

```
set_adc_channel();  
delay_us(10);  
variável = read_adc();
```

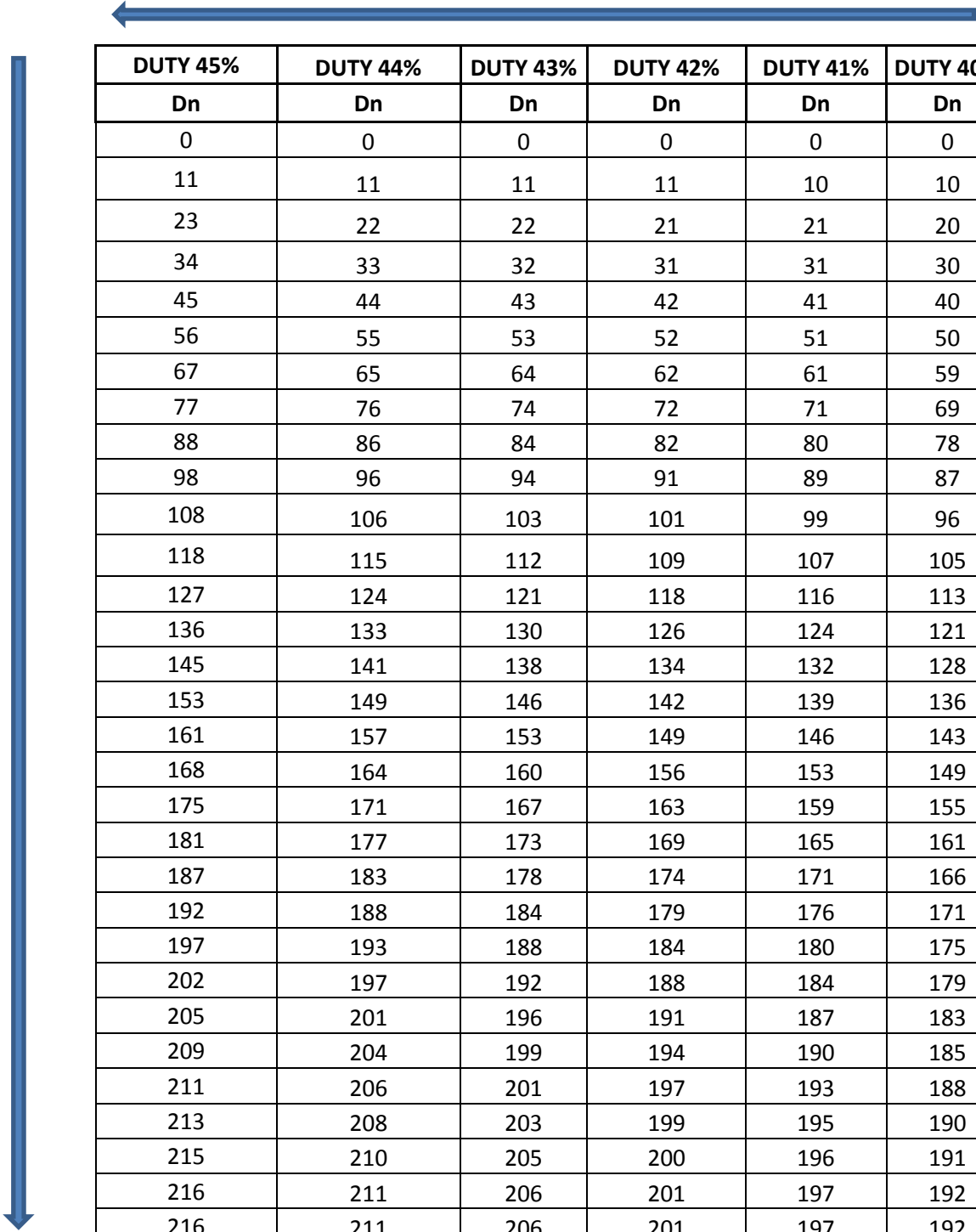
Portanto, primeiro é feito a seleção do canal, que para este projeto será realizado no canal 1, então é gerado um tempo de espera de 10 μs para enfim atribuir a uma variável o valor da leitura.

O projeto do divisor de tensão faz com que a tensão de 200 V seja medida em um intervalo de 0 V a 5 V, portanto a tensão de referência que será utilizada no código é igual a 4,5 V, que corresponde a uma tensão de saída de 180 V, conforme estipulada em projeto.

Como foi utilizada uma escala de 8 bits para realizar a leitura, a variável receberá 255 quando a saída do conversor estiver em 200 V e utilizando a proporção direta, 229 quando estiver a 180 V. Portanto este será o valor referência no código que gera o aumento ou diminuição da referencia da onda SPWM.

A Tabela 9 apresenta os valores de D_n para várias referências de razão cíclica máximas. Neste projeto utilizaram-se seis diferentes valores de razão cíclica, variando de 0,4 a 0,45, cujos valores foram obtidos conforme o método para $D = 0,45$ já apresentado previamente. Estes valores foram inseridos em forma de matriz no código, sendo que o índice de linhas é responsável pelo incremento da largura de pulso no tempo, referente aos ângulos da função seno, e o índice de coluna varia conforme a leitura do conversor A/D. Caso seja necessária uma potência maior, o índice diminui até atingir o ponto de equilíbrio ou até o valor máximo $D = 0,45$, que ocorre quando este índice é zero. O mesmo raciocínio é feito quando a tensão de saída está acima do esperado, fazendo com que o índice aumente e valores de razão cíclica máxima diminuam.

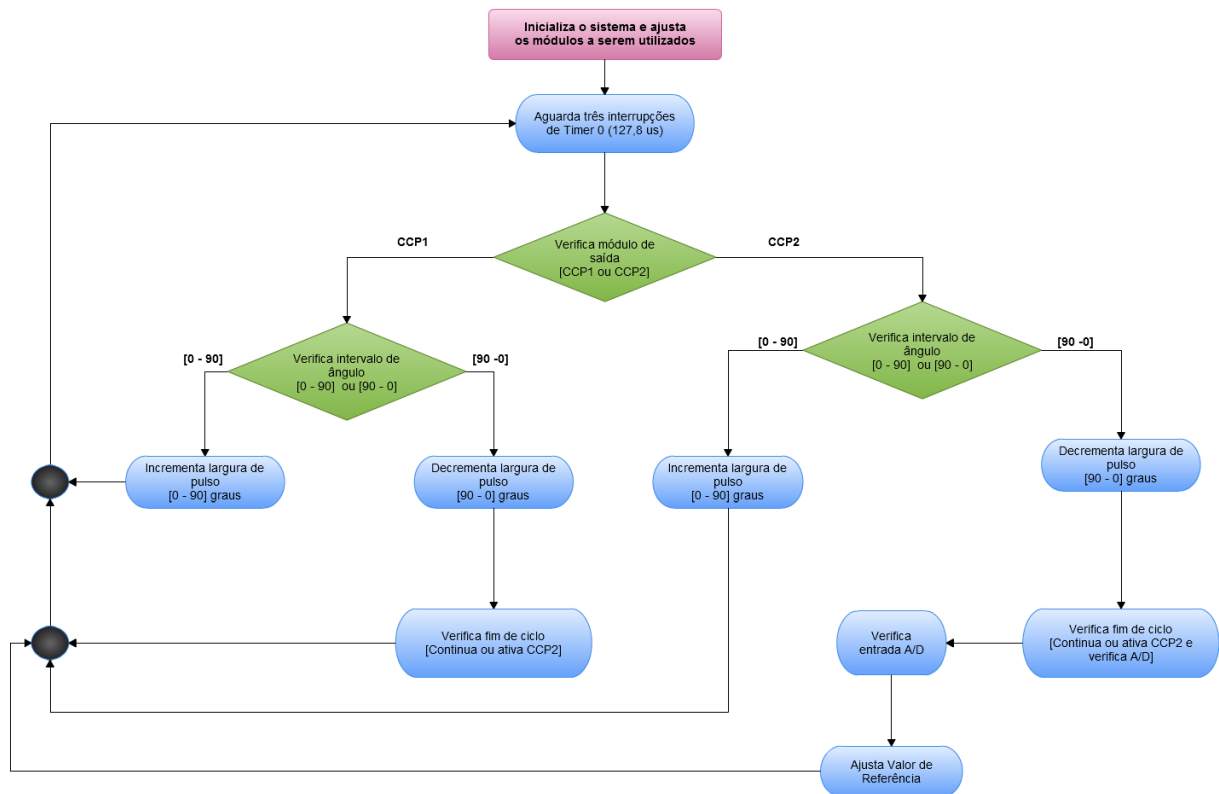
Tabela 9 - Matriz de vetores SPWM.



DUTY 45%	DUTY 44%	DUTY 43%	DUTY 42%	DUTY 41%	DUTY 40%
Dn	Dn	Dn	Dn	Dn	Dn
0	0	0	0	0	0
11	11	11	11	10	10
23	22	22	21	21	20
34	33	32	31	31	30
45	44	43	42	41	40
56	55	53	52	51	50
67	65	64	62	61	59
77	76	74	72	71	69
88	86	84	82	80	78
98	96	94	91	89	87
108	106	103	101	99	96
118	115	112	109	107	105
127	124	121	118	116	113
136	133	130	126	124	121
145	141	138	134	132	128
153	149	146	142	139	136
161	157	153	149	146	143
168	164	160	156	153	149
175	171	167	163	159	155
181	177	173	169	165	161
187	183	178	174	171	166
192	188	184	179	176	171
197	193	188	184	180	175
202	197	192	188	184	179
205	201	196	191	187	183
209	204	199	194	190	185
211	206	201	197	193	188
213	208	203	199	195	190
215	210	205	200	196	191
216	211	206	201	197	192
216	211	206	201	197	192

O código do projeto foi feito conforme o diagrama de fluxo apresentado na Figura 43.

Figura 43—Diagrama de fluxo do código gerador de SPWM.



A programação foi feita em linguagem C e utilizando a biblioteca de funções do aplicativo PCHW seguindo o fluxograma descrito, através da utilização de funções específicas para cada tarefa. O código programado encontra-se no Anexo D.

CAPÍTULO 4

ANÁLISE DOS RESULTADOS

A validação do projeto será feito através de duas etapas devido à limitação dos aplicativos de simulação. A primeira será a simulação do circuito de potência do inversor, que será realizado no aplicativo OrCAD 16.3, e não possui suporte a sistemas microprocessados, portanto será realizada a simulação em malha aberta do circuito. O objetivo desta etapa é confirmar a validade do circuito *Push-Pull* configurado como inversor, apresentando as formas de onda e metodologia de simulação.

Como se trata da versão malha aberta do circuito, os transistores são ativados por componentes *VPULSE* que geram pulsos na frequência de 50 kHz conforme definido em projeto. Foram utilizados dois transistores em cada chave, configurados como *darlington* para gerar o ganho de corrente necessário para operação e o transformador foi construído através de indutores acoplados pelo componente *K_Linear* que habilita a magnetização mútua entre as bobinas. Para simulação de transformadores é preciso adicionar dois tipos de resistências no circuito. A primeira é referente à resistência do primário e possui valores muito baixos a segunda refere-se à resistência entre o primário e secundário do transformador que possui valores muito altos. Sem estas resistências o OrCAD não consegue simular projetos que envolvam transformadores e aponta erro no *layout* do projeto.

Neste projeto foi utilizado 0,5 Ω como resistência de primário e 1000000 $M\Omega$ como resistência entre primário e secundário. Este procedimento garantiu o sucesso da simulação do circuito inversor.

Os indutores do transformador precisam ser inseridos em grandeza de indutância, porém no projeto foram calculados em número de espiras. Essa transformação foi feita através da equação abaixo:

$$N = \sqrt{\frac{L_P}{L_S}} = 0,06 \quad (49)$$

Adotando $L_S = 100 \text{ mH}$ se obtém $L_P = 445 \text{ } \mu\text{H}$, desta forma a relação de espiras N é convertida em unidades de *Henry*.

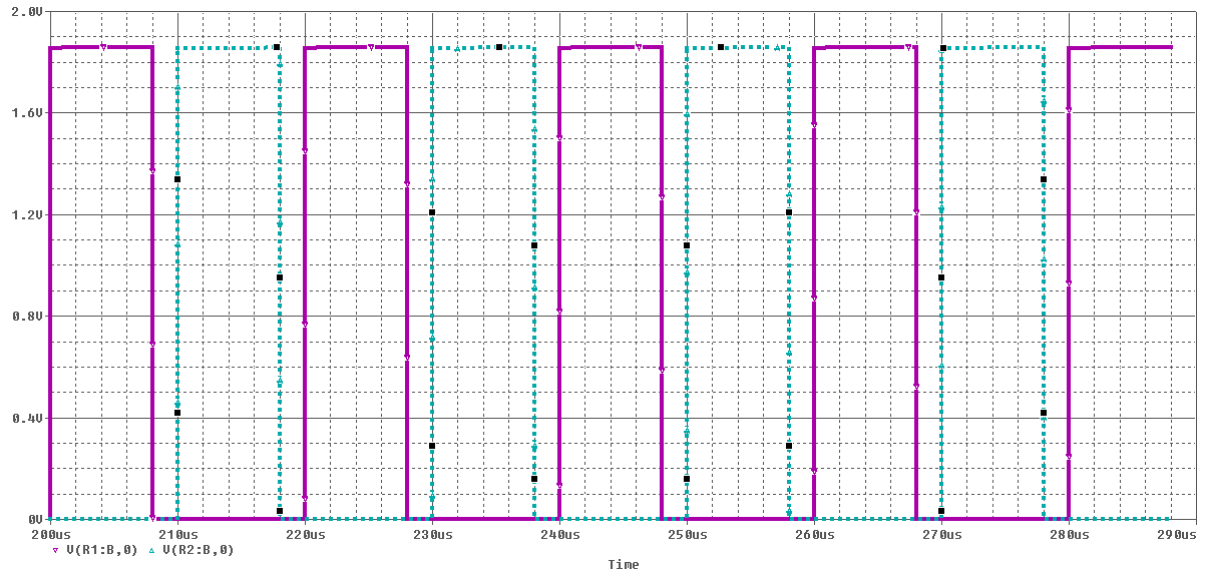
Por fim foi implementado o circuito divisor de tensão cuja tensão de saída foi aplicada em um retificador de onda completa com detector de pico, fazendo com que esta tensão final seja aplicada na entrada A/D do microcontrolador para controle de realimentação.

The circuit diagram illustrates a two-stage CMOS differential amplifier. The first stage is a differential pair consisting of two NMOS transistors (M1, M2) and two PMOS transistors (M3, M4). The PMOS transistors are connected to a Wilson current source, which is implemented using three NMOS transistors (M5, M6, M7) and a PMOS transistor (M8). The Wilson current source is designed to provide a high output impedance and a precise current. The differential pair is biased by a common-mode feedback (CMFB) circuit, which is implemented using a PMOS transistor (M9) and a resistor (R1). The CMFB circuit is designed to maintain the common-mode voltage of the differential pair at a constant level. The output of the differential pair is taken from the drains of M1 and M2, which are connected to a load resistor (R2). The circuit is simulated using a SPICE-like simulator, and the simulation results are shown in the form of a plot of the differential-mode voltage gain versus frequency. The plot shows a peak gain of approximately 100 dB at a frequency of 100 kHz, with a bandwidth of about 1 MHz. The circuit parameters are as follows:

- Transistors: M1, M2, M3, M4, M5, M6, M7, M8, M9
- Resistors: R1 (100k), R2 (100k)
- Capacitors: C1 (1nF), C2 (100pF), C3 (100pF), C4 (100pF), C5 (100pF), C6 (100pF), C7 (100pF), C8 (100pF), C9 (100pF), C10 (100pF), C11 (100pF), C12 (100pF), C13 (100pF), C14 (100pF), C15 (100pF), C16 (100pF), C17 (100pF), C18 (100pF), C19 (100pF), C20 (100pF), C21 (100pF), C22 (100pF), C23 (100pF), C24 (100pF), C25 (100pF), C26 (100pF), C27 (100pF), C28 (100pF), C29 (100pF), C30 (100pF), C31 (100pF), C32 (100pF), C33 (100pF), C34 (100pF), C35 (100pF), C36 (100pF), C37 (100pF), C38 (100pF), C39 (100pF), C40 (100pF), C41 (100pF), C42 (100pF), C43 (100pF), C44 (100pF), C45 (100pF), C46 (100pF), C47 (100pF), C48 (100pF), C49 (100pF), C50 (100pF), C51 (100pF), C52 (100pF), C53 (100pF), C54 (100pF), C55 (100pF), C56 (100pF), C57 (100pF), C58 (100pF), C59 (100pF), C60 (100pF), C61 (100pF), C62 (100pF), C63 (100pF), C64 (100pF), C65 (100pF), C66 (100pF), C67 (100pF), C68 (100pF), C69 (100pF), C70 (100pF), C71 (100pF), C72 (100pF), C73 (100pF), C74 (100pF), C75 (100pF), C76 (100pF), C77 (100pF), C78 (100pF), C79 (100pF), C80 (100pF), C81 (100pF), C82 (100pF), C83 (100pF), C84 (100pF), C85 (100pF), C86 (100pF), C87 (100pF), C88 (100pF), C89 (100pF), C90 (100pF), C91 (100pF), C92 (100pF), C93 (100pF), C94 (100pF), C95 (100pF), C96 (100pF), C97 (100pF), C98 (100pF), C99 (100pF), C100 (100pF)

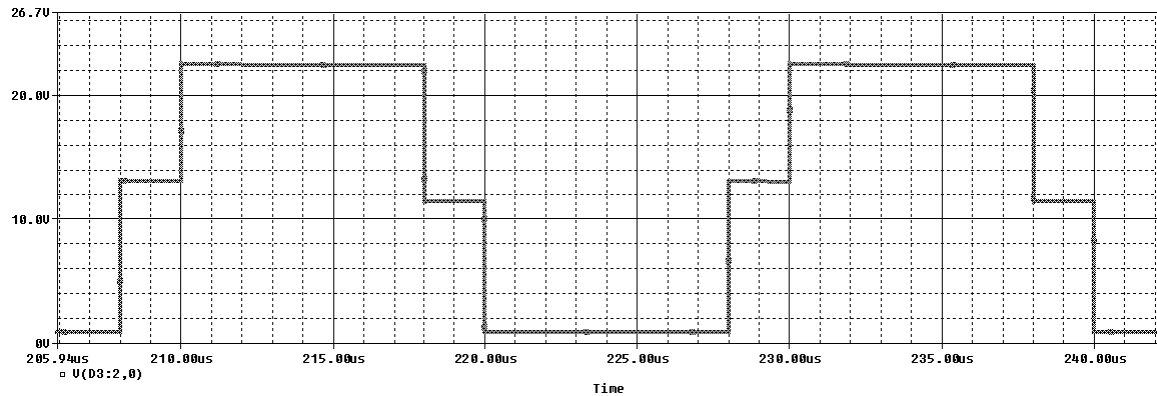
Os componentes *VPULSE* geram uma tensão de amplitude de 12 V e 0V em seus estados *on* e *off* que ligam e desligam os transistores a cada 20 μ s. A forma de onda na base dos transistores, após o resistor de 100 Ω pode ser vista na Figura 45.

Figura 45 - Tensão na base dos transistores.



Como descrito na revisão teórica, os transistores da topologia *Push-Pull* ficam submetidos a uma tensão igual ao dobro da tensão de entrada do circuito. A Figura 46 mostra a tensão a que um dos transistores fica submetido quando em corte.

Figura 46 - Tensão no transistor.



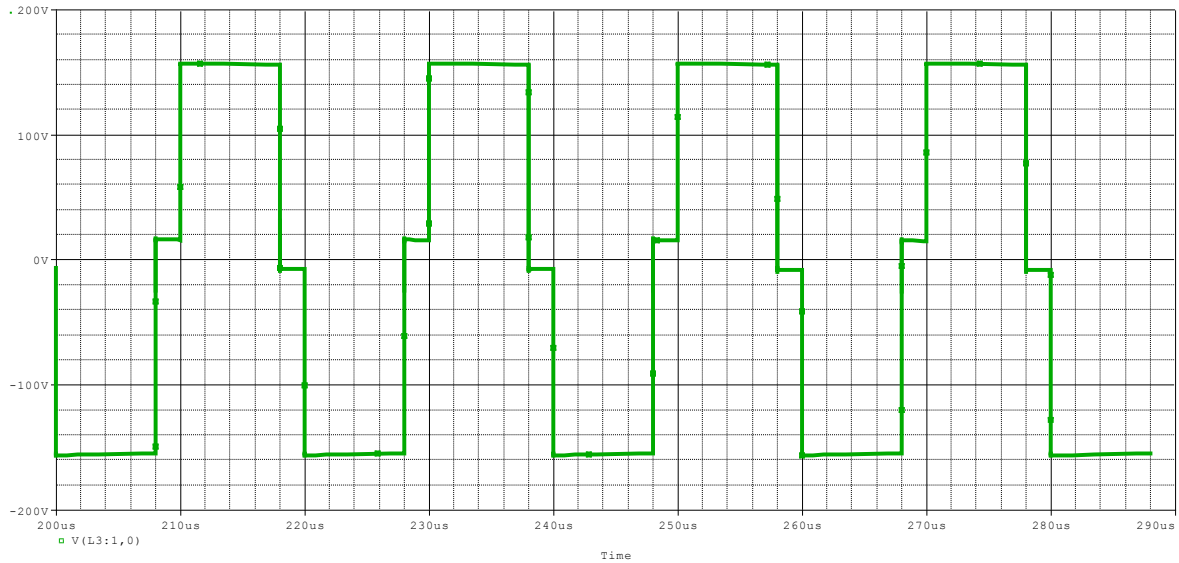
Conforme dados coletados na simulação, os transistores ficam sob uma tensão igual a 22,6V quando a outra chave está em condução e cerca de 11V durante o tempo morto, em que nenhum dos transistores conduz.

A simulação em malha aberta restringe os testes de carga no circuito, devido à ausência da realimentação do sistema, portanto foram utilizadas cargas baixas para validação da topologia inversora em si. A Figura 47 mostra a tensão na saída do circuito para uma razão cíclica fixa, conforme o sinal *VPULSE* enviado aos transistores. O circuito consegue converter a tensão de entrada de 12 VCC para uma tensão aproximada de 158 VCA a 60 Hz através de

uma onda quadrada, embora tenha havido uma pequena perda na amplitude projetada. Este fator poderia ser corrigido através da realimentação do circuito aumentando a potência na carga.

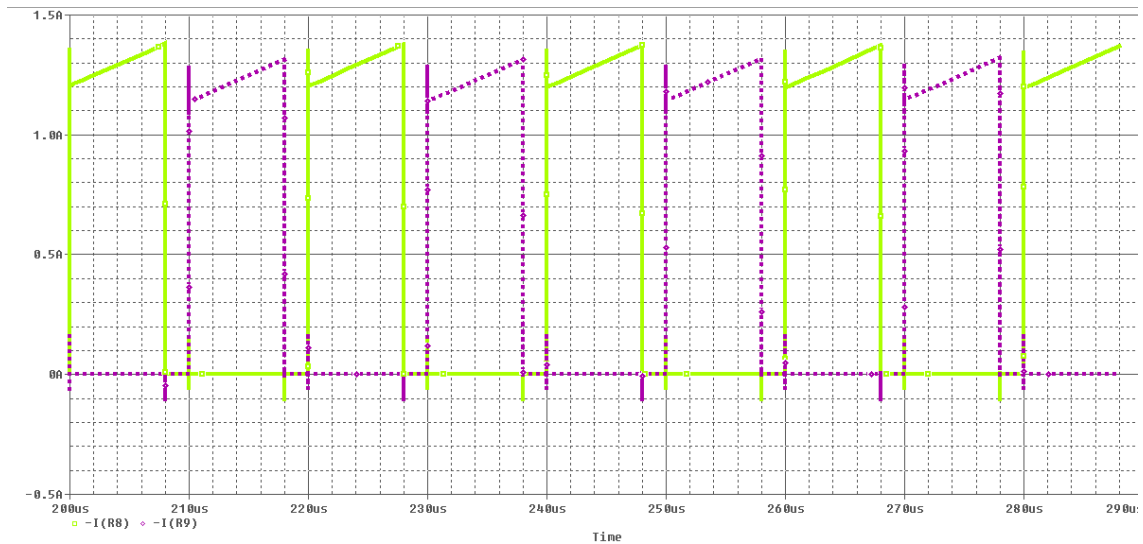
Mas mesmo funcionando em malha aberta os resultados foram satisfatórios, pois comprovam a versatilidade da topologia *Push-Pull* atuando como conversor e inversor.

Figura 47 - Tensão de saída quase-quadrada.



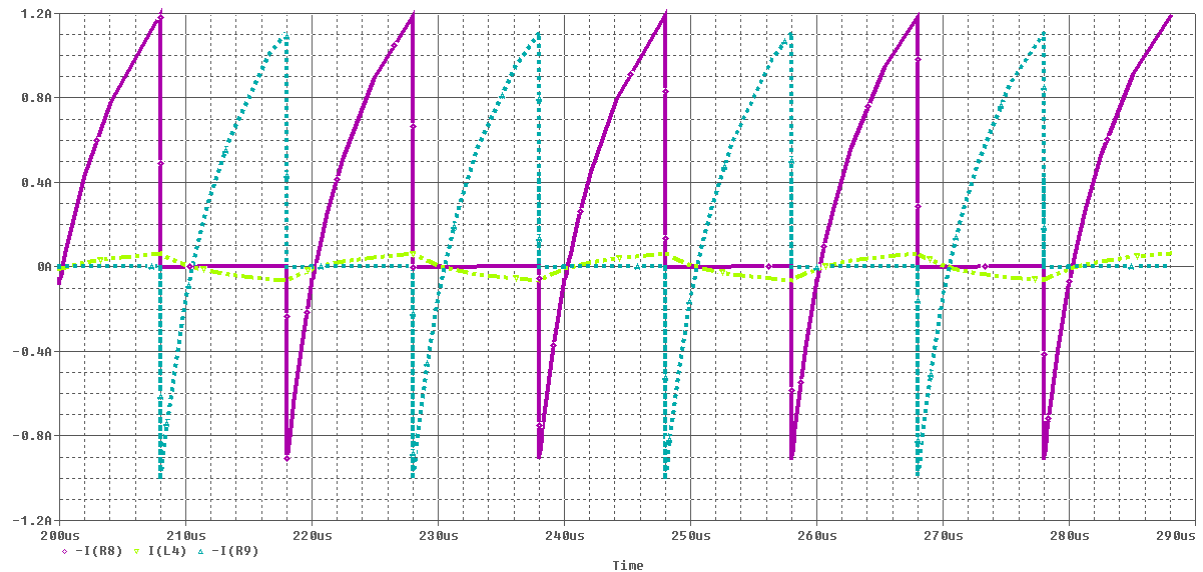
A corrente nas bobinas primárias do circuito varia conforme a indutância da carga. Para cargas com baixos valores de indutância existe baixa resistência ao aumento de corrente gerando um aumento linear no nível de corrente em cada bobina conforme Figura 48.

Figura 48 - Corrente no primário com carga de indutância 10 μ H.



Para cargas com alto fator indutivo, a corrente muda de sentido, mas ainda precisa desmagnetizar o sentido atual da bobina, para então inverter seu sentido e realizar a magnetização na direção oposta. Isto ocorre de maneira gradual e não-linear como mostra a Figura 49.

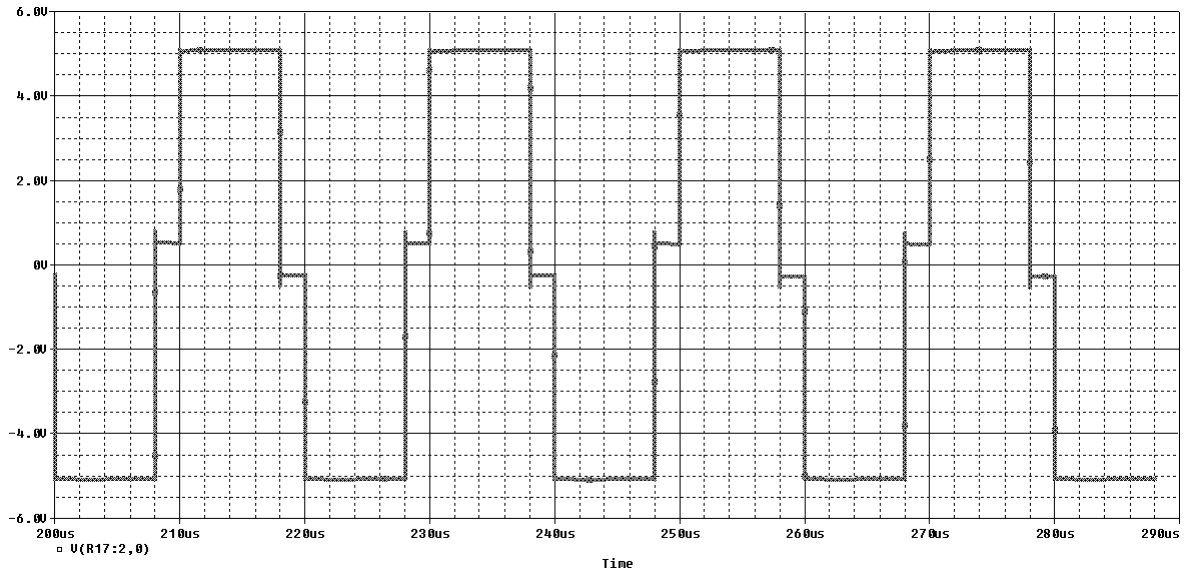
Figura 49 - Corrente no primário com carga de indutância 10mH.



Conforme os resultados apresentados até então, confirma-se a funcionalidade da topologia proposta, visto que o circuito de potência trabalha naturalmente como um inversor. A tensão de saída deste circuito deve prover dados necessários para a realimentação do sistema. Para isto foi simulado a interface entre o circuito de controle e de potência que é apresentado aqui como o divisor de tensão e retificador de onda completa com detector de pico.

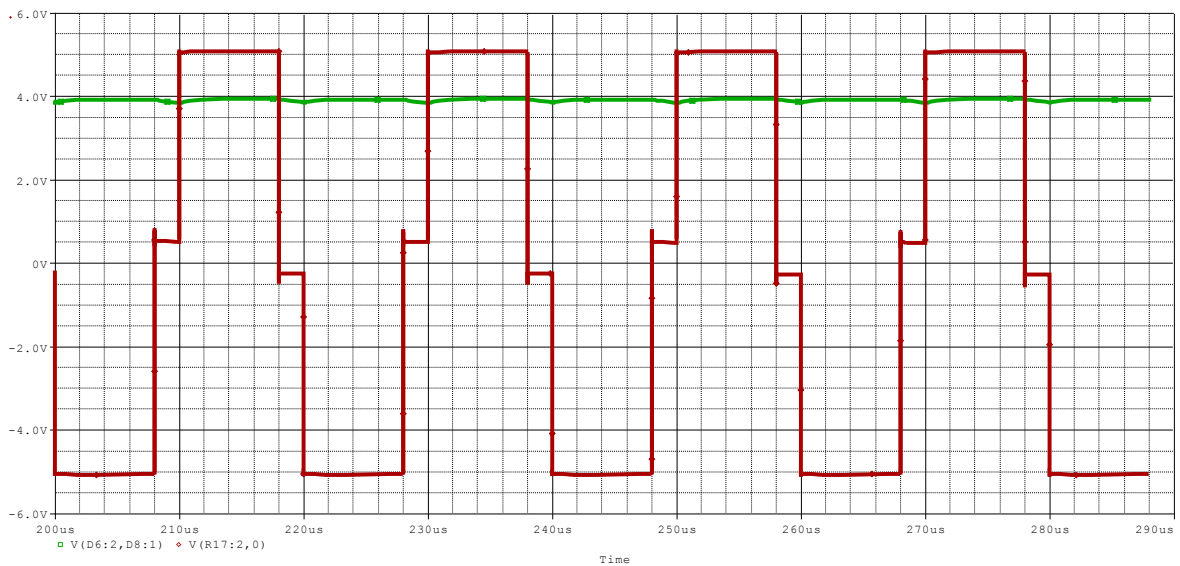
A Figura 50 mostra a amplitude do sinal de saída do divisor de tensão para a tensão de saída simulada anteriormente. Conforme projetado a tensão apresenta um valor de pico um pouco maior que 5V, embora a tensão na carga seja menor que 200 V. Esta diferença ocorre pois foi considerado as quedas nos diodos após a retificação completa da onda.

Figura 50 - Saída do divisor de tensão.



A Figura 51 mostra a tensão de saída do divisor de tensão e o sinal retificado já com seu valor de pico. Com as considerações descritas anteriormente, o valor final da tensão a ser enviada para o microprocessador é equivalente à tensão enviada à carga. Para este caso, os 158 V de pico da saída são convertidos corretamente para 3,95 V, valor que corresponde proporcionalmente à tensão na carga.

Figura 51 - Comparação entre a saída do divisor de tensão e tensão de pico retificada.



Com as limitações do aplicativo OrCAD, nesta seção foi simulado apenas o circuito de potência em malha aberta, e embora não haja realimentação, o circuito realizou a

conversão CC-CA da tensão de entrada. Também foi validado o circuito que gera o valor referência de tensão para a realimentação através do microcontrolador.

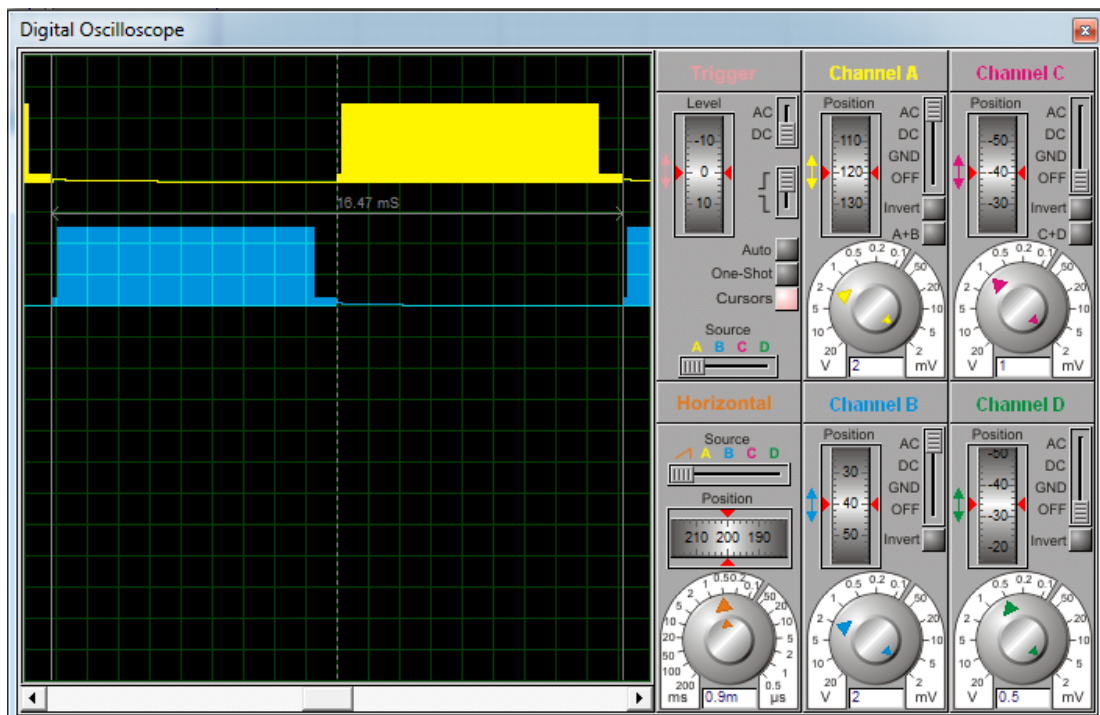
A segunda parte da análise dos resultados foi realizada através de simulação no aplicativo *Proteus 7.8*, no qual foi possível adicionar o código de controle no microcontrolador e analisar seu comportamento. A montagem do circuito é feita pela adição do PIC16F8764A e a entrada de parâmetros deste componente. Primeiramente foi adicionado o código em linguagem hexadecimal (.HEX) gerado pelo compilador no microcontrolador e então inserido o valor de *clock* externo, que torna desnecessário a montagem do circuito oscilador externo, composto pelo cristal de 24 MHz e os dois capacitores cerâmicos de 33 pF.

No circuito simulado (Anexo F), as saídas PWM foram monitoradas através de um osciloscópio pelo qual foram obtidas as formas de ondas apresentadas a seguir.

A Figura 52 mostra o comportamento das chaves CCP1 e CCP2 do microcontrolador, nos pinos 17 e 16 respectivamente. Como projetado, as chaves não devem conduzir ao mesmo tempo e o tempo de condução conjunto devem ser suficientemente próximos do período de um sinal de 60 Hz.

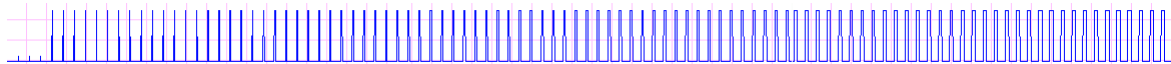
Devido à resolução do osciloscópio digital do aplicativo e da escala de tempo escolhida, não é possível enxergar os pulsos com suas larguras de pulso variando, porém o período de chaveamento conjunto dos transistores está no nível adequado para controle do sistema.

Figura 52 - Acionamento das saídas CCP1 e CCP2.



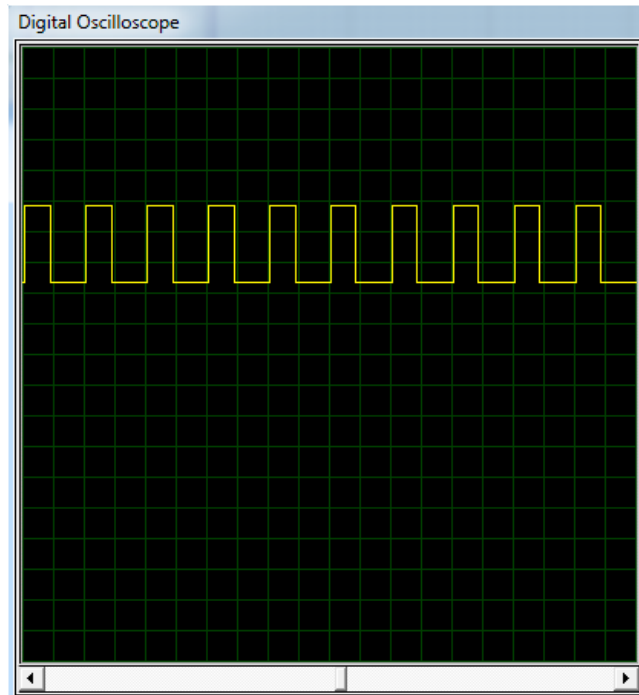
O osciloscópio do Proteus é muito funcional, pois é possível realizar a análise dos sinais de saída em tempo real, porém compromete captura de telas estáticas para apresentação. Portanto foi realizada uma montagem de imagens retiradas deste osciloscópio para que seja possível visualizar o incremento gradual da largura de pulso. Este incremento ocorre a um período aproximado de $128\ \mu s$, tempo suficiente para que seis pulsos com mesmo ciclo de trabalho sejam gerados antes de um novo incremento. Isto ocorre pois a frequência do sinal de saída do módulo PWM é bem maior do que a frequência com que a largura dos pulsos é incrementada. A Figura 53 mostra um trecho do sinal gerado em um dos módulos PWM, no qual é possível ver o aumento da largura do pulso no decorrer do tempo.

Figura 53 - Variação senoidal da largura de pulso do sinal.



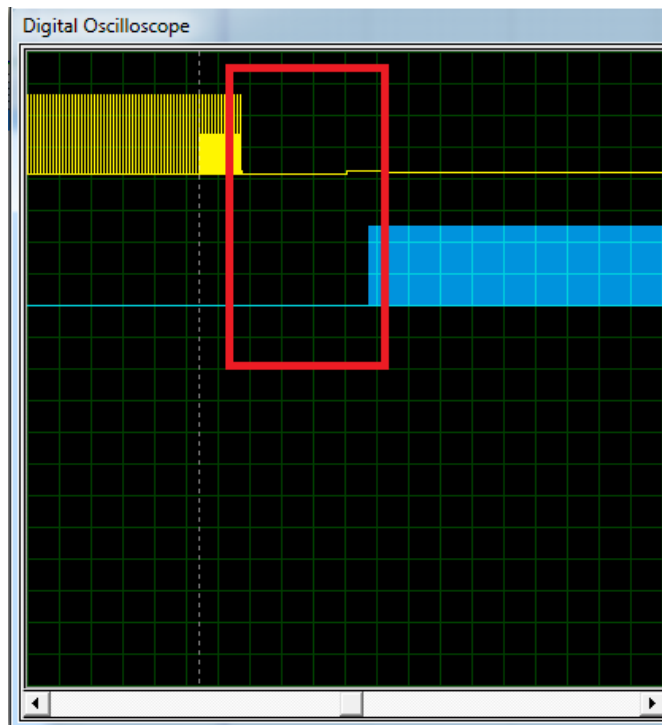
A Figura 54 mostra como ocorre o processo de mudança de razão cíclica. São gerados cinco pulsos de largura exatamente iguais, após estes os pulsos seguintes apresentam uma largura de pulso ligeiramente menor, que pode ser vista comparando-se as bordas de descida com a grade de fundo. Embora haja esta variação, a frequência de saída permanece fixa em $50\ kHz$, mantendo os pulsos com período de $20\ \mu s$.

Figura 54 - Amostra do sinal de saída de CCP1, com escala de 10 μ s por divisão.



Obedecendo ao requisito de chaveamento, existe um tempo entre chaves em que nenhuma das duas conduz, eliminando o risco de curto-circuito na fonte de alimentação.

Figura 55 - Tempo morto entre chaves.



Por fim, foi projetado um filtro RC passa-baixa com frequência de corte de 100 Hz para validar o sinal SPWM gerado que controla o chaveamento dos transistores.

O filtro foi obtido através da equação abaixo (SEDRA, 1998) na qual f_c é a frequência de corte do circuito, estipulada em 100Hz .

$$f_c = \frac{1}{2 \cdot \pi \cdot R \cdot C} \quad (50)$$

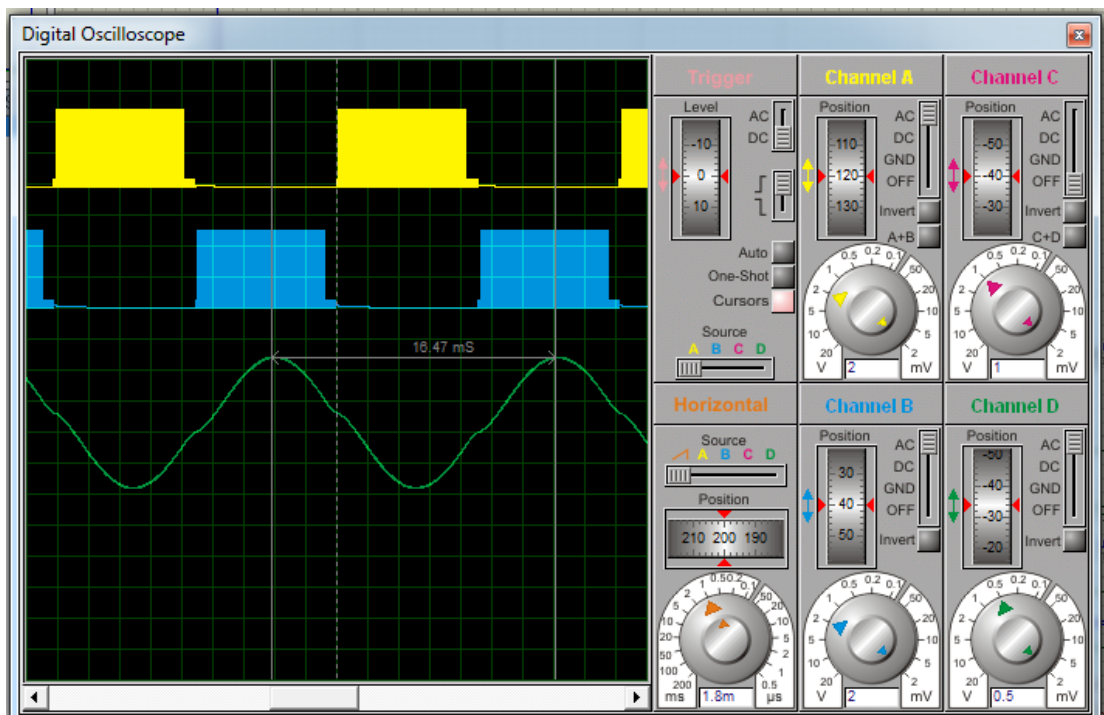
O valor de R foi estipulado em $10\text{k}\Omega$, portanto se valendo da equação (50):

$$C = \frac{1}{2 \cdot \pi \cdot 10k \cdot 100} \cong 160\text{ nF} \quad (51)$$

Cada saída CCP foi seguida de um filtro RC com os parâmetros acima calculados, sendo que uma das saídas ainda foi aplicada a um inversor para simular a tensão negativa induzida no secundário do circuito.

Um das saídas CCP foi invertida através de um amplificador operacional saturado negativamente, para representar a tensão invertida em uma das bobinas do transformador (Anexo F). A onda de saída do filtro é mostrada na Figura 56, na qual é possível ver a variação da tensão formando uma onda com formato senoidal com período de $16,47\text{ ms}$, gerando assim uma frequência aproximada de $60,7\text{ Hz}$, um valor muito próximo do projetado.

Figura 56 - Sinal de saída filtrado.



Através das simulações realizadas no *Proteus* 7.8, foi possível validar a funcionalidade do módulo de controle, cujas saídas PWM gerenciam as chaves do circuito de potência da maneira projetada, gerando um sinal PWM que varia senoidalmente a uma frequência aproximada de 60 *Hz*.

CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo abordar uma topologia alternativa para a implementação de circuitos inversores. Utilizando o conversor *Push-Pull* de maneira modificada, foi possível fazer com que o mesmo circuito realizasse a conversão CC-CC de tensão, elevando o nível do sinal de entrada a patamares desejáveis no projeto e ao mesmo tempo realizar a conversão CC-CA na frequência desejada.

Apesar de o circuito ter sido analisado em malha aberta, devido às limitações nos aplicativos de simulação, foi possível comprovar a eficácia do circuito que realmente atua como um inversor, tornando esta topologia muito interessante e vantajosa, visto que utiliza menos componentes que o conversor *Push-Pull* convencional e seu controle é realizado apenas por duas chaves, simplificando toda a lógica de gerenciamento do sistema.

Para melhor aproveitamento do sinal convertido, optou-se pela modulação SPWM microcontrolada, pela qual foi possível explorar diversas funcionalidades no PIC 16F874A que certamente serão úteis para as mais diversas situações envolvendo sistemas embarcados. Os resultados obtidos pela simulação do microcontrolador foram muito satisfatórios, gerando um sinal com largura de pulso incrementada conforme a função seno. Esta precisão gerou uma senóide com formato bastante próximo do sinal desejado, alternando numa frequência de praticamente 60 Hz conforme o desejado na saída do circuito.

Quanto ao código escrito para o microcontrolador, seria um avanço desenvolver internamente o algoritmo que gere os valores de largura de pulso, pois isto reduziria grandes recursos de alocação de memória do dispositivo além de possibilitar maior número de possibilidades de controle. Para tanto, bastaria realizar as operações feitas para gerar as tabelas dentro do microcontrolador, juntamente com a sua biblioteca de funções trigonométricas internas.

Portanto, o resultado final deste projeto cumpriu com as expectativas ao validar cada parte do projeto com seus respectivos aplicativos de simulação, além de trazer um grande conhecimento multidisciplinar da Engenharia Elétrica através do estudo de sistemas de potência e circuitos microcontrolados diretamente aplicado na solução de problemas comuns de nossa vivência.

BIBLIOGRAFIA

AHMED, A, **Eletrônica de Potência**; tradução Bazán Tecnologia e Linguística; revisão técnica João Antônio Martino. São Paulo: Prentice Hall, 2000.

BARBI, Ivo; MARTINS, Denizar C. **Teoria Fundamental da Eletrônica de Potência**. Documento Interno (INEP - 2001)

FERREIRA, André L. B. **Inversor de tensão 12VDC para 110VAC com potência de 600W**. 2006. 98P. Monografia (graduação em Engenharia Elétrica) – Universidade Estadual de Londrina, Londrina.

FILHO, Ricardo F. P. **Estudo e implementação de uma fonte de tensão alternada de 220V/1kW alimentada por fontes cc de 24V**. 2005. 162P. Monografia (pós-graduação em Engenharia Elétrica) – Universidade Federal de Santa Catarina, Florianópolis.

MICROCHIP. PIC16F87XA Data Sheet: 28/40/44-Pin Enhanced Flash Microcontrollers. Microchip, 2003.

MOHAN, Ned. **First Course on Power Electronics and Drives**. Year 2003 ed. Mineapolis: MNPERE, 2003. p. 30 – 39.

MOHAN, Ned. **Power Electronics – Converters, Applications and Design**. 2 ed. New York: Wiley, John & Sons, Incorporated, 1994.

PCWH, Compiler. Custom Computer Services Inc. help Version 3.236, 2003. Documento eletrônico disponibilizado com o PCW Compiler.

POMILIO, José A. **Fontes Chaveadas**; Campinas: Universidade Estadual de Campinas, 2005.

RASHID, Muhammad H. **Power Electronics Handbook**. San Diego: Academic Press, 2001.

SEDRA, Adel S. e SMITH, Kenneth S. **Microeletrônica**. 4. ed. Makron Books Ltda, São Paulo, 1998.

SILVA, Renato A. **Programando microcontroladores PIC: Linguagem “C”**. São Paulo: Ensino Profissional, 2006.

SKVARENINA, Timothy L. **Power Electronics Handbook** – Industrial Electronics Series. 1. ed. West Lafayette: CRC Press, 2002. p. 201 – 205.

TREVISIO, C.H., **Eletrônica de Potência**; Londrina: Universidade Estadual de Londrina, 2009.

ANEXO A

VALORES DE K_j E X PARA ALGUNS TIPOS DE NÚCLEO.

NÚCLEO	K_j ($20\text{ }^{\circ}\text{C} \leq \Delta t \leq 60\text{ }^{\circ}\text{C}$)	X
POTE	$74,78 \cdot \Delta T^{0,54}$	+0,17
EE	$63,35 \cdot \Delta T^{0,54}$	+0,12
X	$56,72 \cdot \Delta T^{0,54}$	+0,14
RM	$71,7 \cdot \Delta T^{0,54}$	+0,13
EC	$71,7 \cdot \Delta T^{0,54}$	+0,13
PQ	$71,7 \cdot \Delta T^{0,54}$	+0,13

Fonte: Treviso, 2009

ANEXO B

CATÁLOGO DE NÚCLEOS DE FERRITE

Dados de núcleos de Ferrite.

TIPO	DESIGNAÇÃO	A_p (cm ⁴)	CEM (cm)	l_g (cm)	A_g (cm ²)	A_m (cm ²)
POTE	9x5	0,0065	1,85	1,25	0,101	3,18
	11x7	0,0152	2,20	1,55	0,162	4,20
	14x8	0,0393	2,80	1,98	0,251	6,77
	18x11	0,114	3,56	2,58	0,433	10,99
	22x13	0,246	4,40	3,15	0,634	16,31
	26x16	0,498	5,20	3,76	0,939	23,0
	30x19	1,01	6,00	4,52	1,37	31,9
	36x22	2,01	7,30	5,32	2,02	44,2
	42x29	4,81	8,60	6,86	2,65	67,4
	47x28	5,62	9,30	7,25	3,12	76,0
RM	6	0,047	3,0	2,67	0,313	7,3
	7	0,086	3,5	2,98	0,400	9,9
	8	0,156	4,2	3,51	0,520	13,3
	10	0,340	5,2	4,20	0,830	19,7
	12	1,02	6,1	5,69	1,40	34,1
	14	1,90	7,1	7,10	1,78	46,6
X	22	0,330	4,9	3,80	0,66	19,2
	25	0,480	5,5	4,15	0,73	24,5
	30	1,43	6,4	5,50	1,12	40,9
	35	2,85	7,7	6,73	1,64	56,3
PQ	20/16	0,27	4,4	3,74	0,61	14,2
	26/20	0,65	5,6	4,63	1,13	23,6
	26/25	0,91	5,6	5,55	1,13	27,8
	32/20	1,10	6,71	5,55	1,42	29,5
	32/30	2,06	6,71	7,46	1,42	40,1
	35/35	3,48	7,52	8,79	1,62	51,8
	40/40	6,40	8,39	10,20	2,01	65,6
EE	20/10/5	0,48	3,8	4,28	0,312	28,6
	30/15/7	0,71	5,6	6,69	0,597	34,8
	30/15/14	1,43	6,7	6,69	1,20	43,2
	42/21/15	4,66	9,3	9,7	1,82	89,1
	42/21/20	6,14	10,5	9,7	2,40	97,5
	55/28/21	14,91	11,6	12,3	3,54	150
	65/33/26	36,28	15,0	14,7	5,25	312
EC	35	1,36	5,0	7,74	0,843	43,5
	41	2,68	6,0	8,80	1,25	59,0
	52	5,71	7,3	10,3	1,83	91,0
	70	19,2	9,5	14,1	2,83	170

CEM = Comprimento da Espira Média

ANEXO C

CATÁLOGO DE FIOS DE COBRE

Tabela de Fios

AWG	DIÂMETRO cobre cm	ÁREA cobre cm ²	DIÂMETRO c/isol. cm	ÁREA c/isol. cm ²	OHMS/CM 20°C	OHMS/CM 100°C
10	.259	.052620	.273	.058572	.000033	.000044
11	.231	.041729	.244	.046738	.000041	.000055
12	.205	.033092	.218	.037309	.000052	.000070
13	.183	.026243	.195	.029793	.000066	.000088
14	.163	.020811	.174	.023800	.000083	.000111
15	.145	.016504	.156	.019021	.000104	.000140
16	.129	.013088	.139	.015207	.000132	.000176
17	.115	.010379	.124	.012164	.000166	.000222
18	.102	.008231	.111	.009735	.000209	.000280
19	.091	.006527	.100	.007794	.000264	.000353
20	.081	.005176	.089	.006244	.000333	.000445
21	.072	.004105	.080	.005004	.000420	.000561
22	.064	.003255	.071	.004013	.000530	.000708
23	.057	.002582	.064	.003221	.000668	.000892
24	.051	.002047	.057	.002586	.000842	.001125
25	.045	.001624	.051	.002078	.001062	.001419
26	.040	.001287	.046	.001671	.001339	.001789
27	.036	.001021	.041	.001344	.001689	.002256
28	.032	.000810	.037	.001083	.002129	.002845
29	.029	.000642	.033	.000872	.002685	.003587
30	.025	.000509	.030	.000704	.003386	.004523
31	.023	.000404	.027	.000568	.004269	.005704
32	.020	.000320	.024	.000459	.005384	.007192
33	.018	.000254	.022	.000371	.006789	.009070
34	.016	.000201	.020	.000300	.008560	.011437
35	.014	.000160	.018	.000243	.010795	.014422
36	.013	.000127	.016	.000197	.013612	.018186
37	.011	.000100	.014	.000160	.017165	.022932
38	.010	.000080	.013	.000130	.021644	.028917
39	.009	.000063	.012	.000106	.027293	.036464
40	.008	.000050	.010	.000086	.034417	.045981
41	.007	.000040	.009	.000070	.043399	.057982

ANEXO D

ALGORITIMO EM LINGUAGEM C

```
/******  
//Função:Funções  
//Descrição: Inicilização das funções.  
*****/  
voidvCCP_Select();  
void vCCP1_Seno_Up();  
void vCCP1_Seno_Down();  
void vCCP2_Seno_Up();  
void vCCP2_Seno_Down();  
voidvA_D_Select_Duty();  
  
/******  
//Função:Variaveis  
//Descrição: Declaração das variaveis.  
*****/  
  
inti_SEN;  
intj_SEN;  
int i_TIMER0;  
  
intflag_SEN;  
int duty;  
intflag_ccpx;  
  
int16 seno [6] [31] =  
{0,11,23,34,45,56,67,77,88,98,108,118,127,136,145,153,161,168,175,181,187,192,197,  
202,205,209,211,213,215,216,216,  
  
0,11,22,33,44,55,65,76,86,96,106,115,124,133,141,149,157,164,171,177,183,188,193,1  
97,201,204,206,208,210,211,211,  
  
0,11,22,32,43,53,64,74,84,94,103,112,121,130,138,146,153,160,167,173,178,184,188,1  
92,196,199,201,203,205,206,206,  
  
0,11,21,31,42,52,62,72,82,91,101,109,118,126,134,142,149,156,163,169,174,179,184,1  
88,191,194,197,199,200,201,201,  
  
0,10,21,31,41,51,61,71,80,89,99,107,116,124,132,139,146,153,159,165,171,176,180,18  
4,187,190,193,195,196,197,197,  
  
0,10,20,30,40,50,59,69,78,87,96,105,113,121,128,136,143,149,155,161,166,171,175,17  
9,183,185,188,190,191,192,192};
```

```
#int_TIMER0
TIMER0_isr()
{
/*****
//Função:v3xTimer0
//Descrição: Conta 3 interrupções do Timer_0. A interrupção ocorre
//          a cada 42,6us. As demais funções são chamadas após
//          127,8us.
*****/

if (i_TIMER0 > 1)
{
vCCP_Select();
}
else
{
i_TIMER0 = i_TIMER0 + 1;
}

}

/*****
//Função: vMain
//Descrição: Função main. Inicia variaveis e configura os setups
//          do Pic
*****/

void main()
{
int loop = 1;
setup_adc_ports(AN0_AN1_AN2_AN3_AN4);
setup_adc(ADC_CLOCK_INTERNAL);
setup_psp(PSP_DISABLED);
setup_spi(FALSE);
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
setup_timer_1(T1_DISABLED);
setup_timer_2(T2_DIV_BY_1,119,1);
setup_ccp1(CCP_PWM);
setup_ccp2(CCP_PWM);
setup_comparator(NC_NC_NC_NC);
setup_vref(FALSE);
enable_interrupts(INT_TIMER0);
enable_interrupts(GLOBAL);

while (loop = 1)
{
loop = 1;

}
```

```
}

/*****
//Função:    vCCP_Select
//Descrição: Seleciona qual saída será selecionada: CCP1 ou CCP2
//           CCP1 e CCP2 são seletores das chaves do push-pull
*****/
void vCCP_Select()
{
    if (flag_ccpx == 1)
    {
        vCCP1_Seno_Up();
    }

    else
    {
        vCCP2_Seno_Up();
    }
}

/*****
//Função:    vCCP1_Seno_Up
//Descrição: Incrementa índice do vetor seno a cada 127,8us
//           em um intervalo de 0graus - 90graus (de 3 em 3 graus).
*****/
void vCCP1_Seno_Up()
{
    if ((j_SEN < 30) && (flag_SEN == 1))
    {
        setup_ccp1(CCP_PWM);
        setup_ccp2(CCP_OFF);
        duty = seno[i_SEN][j_SEN];
        set_pwm1_duty(duty);
        j_SEN = j_SEN + 1;
        i_TIMER0 = 0;
    }

    else
    {
        vCCP1_Seno_Down();
    }
}

/*****
//Função:    vCCP1_Seno_Down
//Descrição: Decrementa índice do vetor seno a cada 127,8us
//           em um intervalo de 90graus - 0graus (de 3 em 3 graus).
*****/

void vCCP1_Seno_Down()
{
    flag_SEN = 0;

    if ((j_SEN > 0) && (flag_SEN == 0))
    {

```

```
j_SEN = j_SEN - 1;
duty = seno[i_SEN][j_SEN];
        set_pwm1_duty(duty);
        i_TIMER0 = 0;
    }

else
    {
flag_ccpx = 2;
flag_SEN = 1;
disable_interrupts(INT_TIMER0);
delay_us (32);
enable_interrupts(INT_TIMER0);

        vCCP2_Seno_Up();

    }

}

/*****
//Função:    vCCP2_Seno_Up
//Descrição: Incrementa índice do vetor seno a cada 127,8us
//            em um intervalo de 0graus - 90graus (de 3 em 3 graus).
*****/
void vCCP2_Seno_Up()
{
if ((j_SEN< 30) && (flag_SEN == 1))
    {
        setup_ccp2(CCP_PWM);
        setup_ccp1(CCP_OFF);
        duty = seno[i_SEN][j_SEN];
        set_pwm2_duty(duty);
        j_SEN = j_SEN + 1;
        i_TIMER0 = 0;
    }

else
    {
        vCCP2_Seno_Down();
    }
}

/*****
//Função:    vCCP2_Seno_Down
//Descrição: Decrementa índice do vetor seno a cada 127,8us
//            em um intervalo de 90graus - 0graus (de 3 em 3 graus).
*****/
void vCCP2_Seno_Down()
{
flag_SEN = 0;

if ((j_SEN> 0) && (flag_SEN == 0))
    {
        j_SEN = j_SEN - 1;
        duty = seno[i_SEN][j_SEN];
    }
}
```

```
        set_pwm2_duty(duty);
        i_TIMER0 = 0;
    }

else
    {
flag_ccpx = 1;
flag_SEN = 1;
disable_interrupts(INT_TIMER0);
delay_us (32);
enable_interrupts(INT_TIMER0);

vA_D_Select_Duty();

    }

}

/*****
//Função:    vA_D_Select_Duty
//Descrição: Realiza a leitura da porta A/D 1 e ajusta a referência
//            do ciclo de trabalho.
*****/

voidvA_D_Select_Duty()

{
intDn;

set_adc_channel(1);
delay_us(10);
Dn = read_adc();

    if (Dn> 229)
    {
        if (i_SEN< 5)
        {
            i_SEN = i_SEN++;
        }
    }

    else
    {
        if (i_SEN> 0)
        {
            i_SEN = i_SEN--;
        }
    }

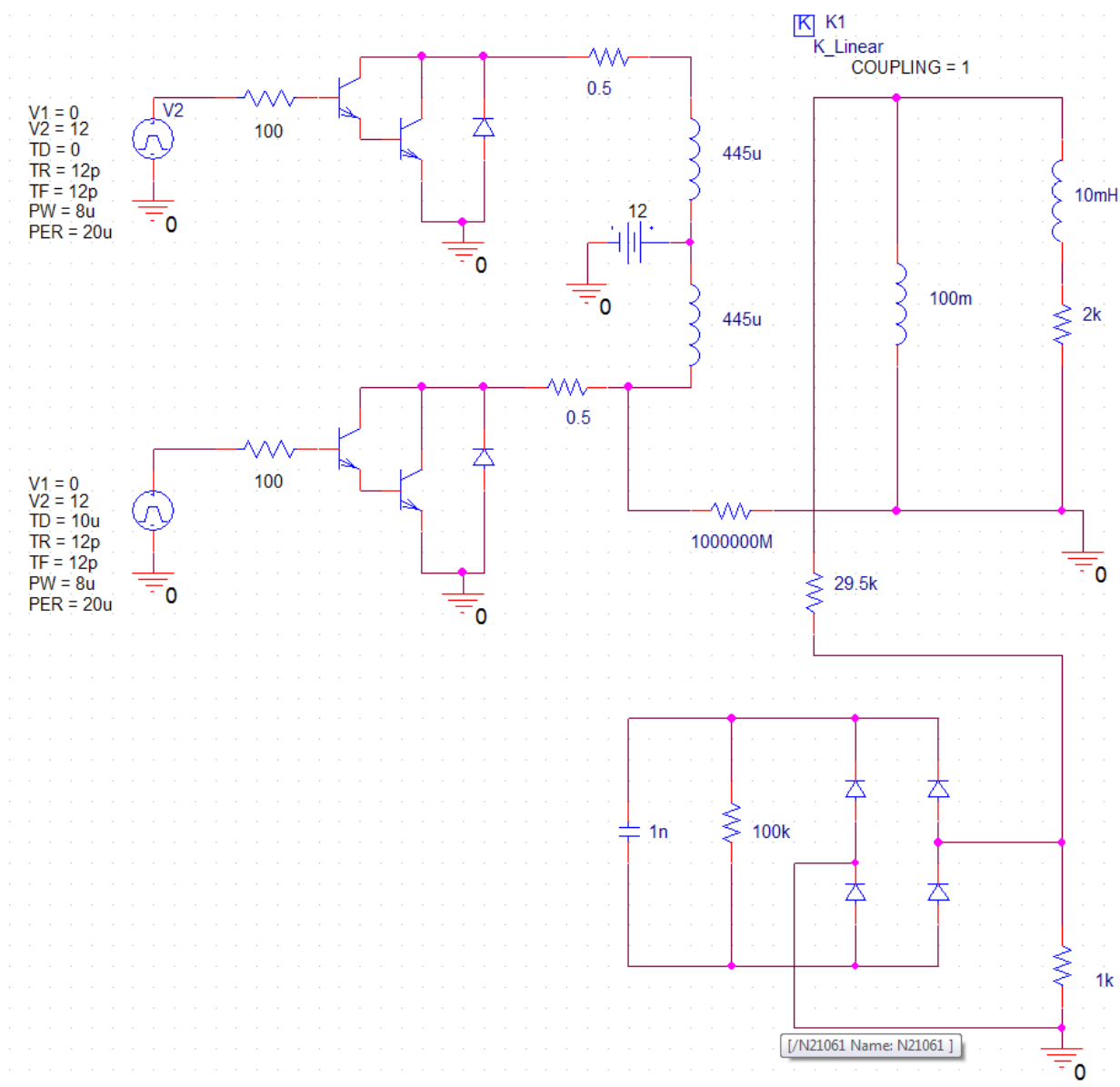
}
```

ANEXO E

CIRCUITO DE POTÊNCIA SIMULADO NO APLICATIVO ORCAD 16.3.

Push-Pull inversor simulado em malha aberta, no qual cada *darlington* representa uma chave do conversor. O Trafo foi configurado através de acoplamento de indutores através do componente K_Linear com fator de acoplamento unitário.

Figura 57 - Circuito de potência simulado no aplicativo OrCAD 16.3.



ANEXO F

CIRCUITO DE CONTROLE SIMULADO NO APLICATIVO PROTEUS 7.8

No Proteus não se faz necessária a montagem do circuito oscilador externo, bastando apenas apontar a frequência de oscilação na configuração do microcontrolador. As duas saídas CCP do microcontrolador foram seguidas de um filtro RC passa-baixa, sendo que o módulo CCP1 no pino 17 passa ainda por um inversor, que representa a magnetização negativa do secundário do transformador. O osciloscópio faz a leitura de cada saída dos módulos PWM e da saída de um somador, que mostra os sinais agindo conjuntamente.

Figura 58 - Circuito de controle simulado no aplicativo Proteus 7.8.

