



UNIVERSIDADE FEDERAL DE OURO PRETO



ESCOLA DE MINAS

DEPARTAMENTO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

FELIPE MARTINS FERRONI

**INVERSOR DE TENSÃO SENOIDAL MICROCONTROLADO POR
ARDUINO**

**MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E
AUTOMAÇÃO**

**OURO PRETO – MG
2018**

FELIPE MARTINS FERRONI

INVERSOR DE TENSÃO SENOIDAL MICROCONTROLADO POR ARDUINO

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Vinicius Marinho Silva

Co-orientador: Eduardo Francisco de Avelar

OURO PRETO – MG
2018

F396i

Ferroni, Felipe Martins.

Inversor de Tensão Senoidal Microcontrolado por Arduino [manuscrito] /
Felipe Martins Ferroni. - 2018.

77f.: il.: color; tabs.

Orientador: Prof. MSc. Vinicius Marinho Silva.

Coorientador: Prof. MSc. Eduardo Francisco Avelar.

Monografia (Graduação). Universidade Federal de Ouro Preto. Escola de
Minas. Departamento de Engenharia de Controle e Automação e Técnicas
Fundamentais.

1. Inversor de frequencia. 2. Ponte H. 3. Arduino. I. Silva, Vinicius
Marinho. II. Avelar, Eduardo Francisco. III. Universidade Federal de Ouro
Preto. IV. Título.

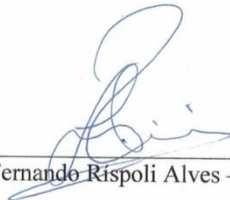
CDU: 681.5

Catálogo: ficha@sisbin.ufop.br

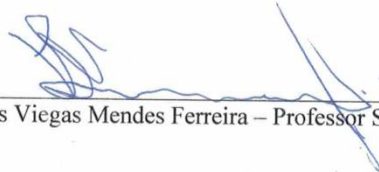
Monografia defendida e aprovada, em 21 de junho de 2018, pela comissão avaliadora constituída pelos professores:



Prof. Substituto M. Sc. Vinícius Marinho da Silva - Orientador



Prof. Dr. Luiz Fernando Rispoli Alves – Professor Convidado



M.Sc. Diógenes Viegas Mendes Ferreira – Professor Substituto Convidado

Dedico este trabalho a minha mãe que sempre me apoiou e estará ao meu lado independente do que aconteça. A meu pai por toda experiência. Minha irmã pelo carinho e meu filho pelo amor.

AGRADECIMENTOS

Agradeço ao professor Mestre Vinicius Marinho Silva por todo apoio, ensinamentos e conversas. Ao Eduardo Francisco de Avelar por todo conhecimento e disponibilidade, sem sua ajuda esta monografia não teria a mesma qualidade. A Universidade Federal de Ouro Preto por toda estrutura acadêmica e residencial. A Fundação Gorceix pelo suporte e capacitação. A minha família por todo apoio.

"O futuro vai mostrar os resultados e julgar cada um segundo as sua realizações"

Nikola Tesla

RESUMO

Inversores de frequência (ou conversores CC-CA) são dispositivos eletrônicos originados graças aos conceitos estudados na Eletrônica de Potência que visam, principalmente, converter a tensão contínua em tensão alternada através da devida comutação de suas chaves semicondutoras, usualmente comandadas por microcontroladores, para os mais diversos fins, desde o controle de máquinas CA à aplicações em projetos de conversão de energia ou geração de corrente alternada em projetos fotovoltaicos. Este trabalho propõe o estudo e implementação computacional e experimental do inversor ponte H utilizando modulação PWM em baixa frequência (1KHz) para aplicações em média potência.

Palavras chave: Inversor de frequência, Ponte H, sPWM, Arduino.

A B S T R A C T

Frequency Inverters are (or converters DC-AC) are electronics devices originated by concepts of power electronics that aim, mainly, to convert the direct current voltage in alternate current voltage by due switching from semiconductors keys usually micro-controllers modifying forms of voltage waves that are applied in your input through switching with micro-controllers, for the most diverse purposes, starts in controls of alternate voltage machines until applications of energy conversion or generation of alternating chain in photovoltaic project. This document proposes the study, computer and experimental implementation of inverter H bridge using PWM modulation in low frequency (1KHz) for applications in half power.

Key Words: Frequency Inverter, H bridge sPWM, Arduino.

TABELA DE SIGLAS

Sigla	Significado
CC	Corrente Contínua
CA	Corrente Alternada
SCR	Silicon Controlled Rectifier - Retificador Controlado de Silício
TRIAC	Triode for Alternating Current- Triodo para Corrente Alternada
MOSFET/TECMOS	Metal Oxide Semiconductor Field Effect Transistor, - Transistor de Efeito de Campo Metal - Óxido - Semicondutor
IGBT	Insulated Gate Bipolar Transistor - Transistor Bipolar de Porta Isolada
PWM	Pulse Width Modulation - Modulação por Largura de Pulso
TDHv	Taxa de Distorção Harmônica
TBJ	Transistor Junção Bipolar
KVA	Kilo Volt Ampere
Dc	Duty Cycle
kHz	Kilo Hertz
Ref	Referência
Port	Portadora
CI	Circuito Integrado
ON	Ligado

SUMÁRIO

CAPÍTULO 1.....	12
1 INTRODUÇÃO	12
1.1 Justificativa	13
1.2 Objetivos.....	13
1.2.1 Objetivo Geral	13
1.3 Metodologia	13
1.4 Estrutura do Trabalho	14
CAPÍTULO 2.....	15
2 REVISÃO BIBLIOGRÁFICA	15
2.1 PONTE H.....	16
2.2 MODULAÇÃO PWM.....	20
2.2.1 Sinal de Referência.....	20
2.2.2 Sinal de Portadora.....	21
2.2.3 Modulador	22
2.2.4 Geração de Sinal Complementar	25
2.2.5 Tempo Morto (dead time).....	26
2.3 MODULAÇÃO PWM ATRAVÉS DE UM ARDUINO MEGA	27
2.4 MODULAÇÃO sPWM	28
2.4.1 Etapas de Operação	30
2.5 MODULAÇÃO sPWM ATRAVÉS DE UM ARDUINO MEGA.....	32
CAPÍTULO 3.....	34
3 PROJETO	34
3.1 ETAPA DE POTÊNCIA	34
3.2 ETAPA QUE MONITORA A TENSÃO DE SAÍDA.....	35
3.3 ETAPA QUE MONITORA A TENSÃO DA BATERIA	36
3.4 ETAPA QUE MONITORA A CORRENTE CC DE ENTRADA	36
3.5 MICROCONTROLADOR	37
3.6 ALGORITMO CONTROLADOR	38
3.6.1 Variáveis Temporárias para o Filtro de Média Móvel	39
3.6.2 Variáveis de Leitura do Sensor de Corrente.....	40
3.6.3 Variáveis para Cálculo e Formação do <i>Locatable</i>	41
3.6.4 Leituras das Variáveis de Tensão CA/CC e Corrente	41
3.6.5 Declaração e Implementação dos Timers	42
3.6.6 Declaração dos Pinos de Entrada e Saída.....	43

3.6.7	Funções que Fazem a Leitura do Circuito	43
3.6.8	Função Média Móvel.....	44
3.6.9	Atuação nos Pinos do MOSFETs	45
3.7	SOFTWARE DE SIMULAÇÃO	47
3.8	MONTAGEM.....	48
CAPÍTULO 4.....		53
4	RESULTADOS E DISCUSSÕES	53
4.1	SIMULAÇÕES	53
4.2	RESULTADOS DO PROTÓTIPO.....	57
4.2.1	Aplicação da Modulação sPWM aos MOSFETs	57
4.2.2	O Sinal de Saída Alternado	58
4.3	TESTES DO INVERSOR	59
4.3.1	TESTES REALIZADOS NO INVERSOR	59
4.4	CIRCUITO BOOTSTRAP	60
4.5	O TRANSFORMADOR.....	60
4.6	CI ACS712.....	61
CAPÍTULO 5.....		62
5	CONCLUSÕES.....	62
CAPÍTULO 6.....		63
6	REFERÊNCIAS BIBLIOGRÁFICAS	63
ANEXO I – CÓDIGO FONTE UTILIZADO PELO MICROCONTROLADOR PARA OPERAR AS SAÍDAS DO MICONROLADOR COM O INTUITO DE SIMULAR E REALIZAR A MODULAÇÃO SPWM.....		64
ANEXO II - CÓDIGO FONTE UTILIZADO PELO MICROCONTROLADOR PARA OPERAR SUAS ENTRADAS E SAÍDAS COM O INTUITO DE REALIZAR LEITURAS E OPERARAÇÃO DO INVERSOR.		67

CAPÍTULO 1

1 INTRODUÇÃO

A eletrônica de potência pode ser entendida como ciência que visa estudar e desenvolver dispositivos eletrônicos como, por exemplo, os conversores estáticos de energia elétrica, inversores de frequência (CC(corrente contínua) - CA (corrente alternada)), retificadores e conversores CC-CC, de forma que estes dispositivos tenham sempre a máxima eficiência e qualidade nos processos de transformação da energia elétrica.

Para Silva (2013, p.1), a energia elétrica pode ter infinitas formas de onda que geralmente são relacionadas com sua fonte de criação. Essas formas de onda são a representação gráfica de uma onda eletromagnética ou sonora em relação ao tempo, podendo possuir períodos cíclicos (repetições e períodos idênticos ao longo do tempo).

Por volta de 1920, surge a Eletrônica de Potência que trouxe vantagens para o processamento de energia, se deve principalmente a pouca necessidade de manutenção das chaves semicondutoras e junto com baixa perda de energia no chaveamento. No princípio, as chaves eram caras, de grande porte e não muito confiáveis, foi a partir da invenção do tiristor que a eletrônica de Potência ganha expressividade, o que necessitou de bases teóricas para o estudo deste novo campo com inúmeros trabalhos de pesquisa e desenvolvimento.

O principal objetivo da eletrônica de potência é processar e controlar o fluxo da energia elétrica, com a utilização de dispositivos semicondutores (diodos e transistores) operando como chaveadores (ligar/desligar), realizado a conversão de formas de onda de tensão e correntes para fontes e cargas com o mínimo de perdas e interferências nos processos de conversão de energia. Como principais chaves nos dias de hoje temos diodos, transistores bipolares, SCR's, Triacs, GTO's, MOSFET's e IGBT's.

A eletrônica de potência esta presente em vários ambientes residenciais, comerciais e industriais em geral. Dentro de suas utilidades podemos destacar os inversores, objeto principal de estudo deste trabalho, mas também fontes chaveadas, controle de motores de corrente contínua e alternada, processamento de energias alternativas, amplificadores de potência, utilidades domésticas, robótica, sistemas de acionamento para veículos elétricos e híbridos e outra tantas aplicações.

1.1 JUSTIFICATIVA

A automação industrial (e das tecnologias em geral) está avançando em conjunto com os conceitos de eletrônica de potência e dispositivos integrados. Entender, simular o funcionamento de um conversor de tensão e montar um dispositivo, permite estudos avançados nesta área e ajuda entender como diversos equipamentos, que possuem um inversor em seu sistema, ou que dependam de um, funcionam.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

O objetivo geral é estudar e implementar um inversor ponte H de tensão controlado pela modulação sPWM através de um Arduino;

Os objetivos específicos são:

- Revisar os conceitos de circuitos e dispositivos eletrônicos;
- Estudar e projetar uma forma de transformar a corrente contínua, em corrente alternada;
- Simular o funcionamento do dispositivo;
- Construir um protótipo do inversor;
- Teste do protótipo “à vazio” e com cargas reais.
- Comparação de resultados do modelo computacional com o real

1.3 METODOLOGIA

No desenvolvimento do trabalho, realizou-se um estudo dos conceitos básicos do equipamento, justificando-se a possível aplicação do dispositivo e quais as vantagens na sua aplicação. Fez-se uma revisão de conceitos pertinentes à construção do dispositivo. Após pesquisa, realizou-se uma proposta para a construção do sistema, e então, desenvolveu-se uma simulação do funcionamento de um inversor e conseqüentemente uma montagem prática. Em suma, o estudo é dividido em:

- Definição de Inversores;

- Estudo dos conceitos pertinentes à construção do dispositivo;
- Estudo de modulações existentes para transformar a corrente;
- Definição de circuitos utilizados e suas funções no dispositivo conversor.
- Simulação de modulações;
- Montagem do sistema
- Coleta de resultados e testes com o protótipo montado.

1.4 ESTRUTURA DO TRABALHO

No segundo capítulo – Revisão Bibliográfica. Tem-se a revisão dos conceitos pertinentes para o entendimento do tema e para a compreensão do método proposto deste trabalho.

No terceiro capítulo – Metodologia utilizada para desenvolver o projeto

No quarto capítulo - Projeto. É estudado, desenvolvido, simulado e apresentado o trabalho proposto.

No quinto capítulo – Resultados e Discussões. São realizados comentários sobre os resultados e os objetivos atingidos além de possíveis melhorias que possam ser aplicadas ao projeto.

No sexto – Conclusão. Um apanhado geral de toda a experiência adquirida na execução deste trabalho e idias futuras para o projeto.

No sétimo capítulo – Referências Bibliográficas. É exposto todo o material consultado para o estudo do projeto.

Anexo I – Código fonte utilizado pelo microcontrolador para operar as saídas do microcontrolador com o intuito de simular e realizar a modulação sPWM.

Anexo II - Código fonte utilizado pelo microcontrolador para operar suas entradas e saídas com o intuito de realizar leituras e operação do inversor.

CAPÍTULO 2

2 REVISÃO BIBLIOGRÁFICA

Segundo Avelar (2018, p.4) inversores são equipamentos que permitem converter uma forma de energia elétrica contínua em alternada, esta pode ser entendida como a definição de inversores na eletrônica de potência. De acordo com Barbi (2007, p.9), inversores de tensão possuem uma vasta gama de estruturas (topologias), que são encontradas em diversas literaturas especializadas, porém não permitem desenvolver projetos completos. A escolha de uma destas estruturas depende principalmente da finalidade onde o mesmo irá atuar, além de custo, potência, frequência de operação dentre outros.

Os principais tipos de inversores encontrados atualmente são os inversores senoidais puros, senoidal modificada, onda totalmente quadrada e onda modificada. Para Silva et al. (2003, p.2), os inversores de onda quadrada apresentam alta taxa de distorção harmônica (TDHv), superior a 40%, e baixa eficiência, e torno de 60 a 80 %, porém são de mais fácil construção e mais baratos. Inversores de onda quadrada modificada apresentam TDHv de 15 a 35 % e eficiência de 80 a 95 %. Tanto os primeiros quanto estes têm o inconveniente de não conseguirem regular o pico de tensão, que varia com o estado da bateria.

Ainda para os mesmo autores, Os inversores de onda senoidal são vantajosos por oferecerem um melhor controle da frequência e da amplitude da tensão, com alta eficiência (85 a 95 %) e baixa TDHv (1 a 5 %), tornando-os tecnicamente desejáveis. No entanto, tornam-se inviáveis, em muitas aplicações, pelo alto preço.

Conversores são, geralmente, constituídos de três etapas: bloco retificador, filtros e o inversor propriamente dito. O bloco retificador, normalmente constituído por diodos, transforma o sinal de entrada em contínuo pulsante. Os filtros, em seguida, atuam sobre os pulsos contínuos tentando linearizar a forma de onda. Por último o inversor, constituído por um conversor estático e componentes semicondutores como IGBTs, MOSFETs, TBJ e SCR, capazes de fazerem o chaveamento em pequenas, médias e altas frequências, comandando uma etapa de potência através de um controle micro processado, que foi programado para obter a forma necessária de saída. (Galhardo e Pinho, 2003).

Para os mesmos autores, o chaveamento (ou modulação) aplicado aos conversores estáticos, geralmente em circuitos mais simples, é por onda quadrada, conhecida também como trem de pulsos. O *Pulse With Modulation* – PWM - é mais usual para circuitos de maior potência e que alimentam grandes diversidades de cargas, uma vez que consegue deixar a onda de saída do inversor igual ou muito próximo da onda necessária.

2.1 PONTE H

Para obter uma tensão alternada através de uma tensão em corrente contínua, aplica-se a técnica de montagem de um circuito conhecido como ponte H, que através de chaveamentos estratégicos, permite a carga “enxergar” uma tensão alternada.

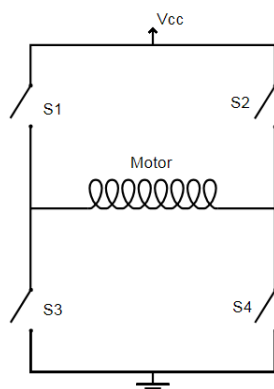


Figura 2.1 a

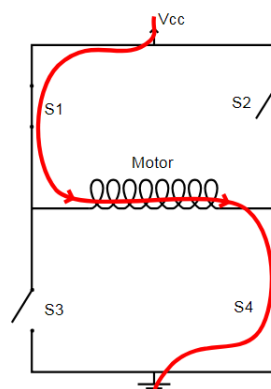


Figura 2.1 b

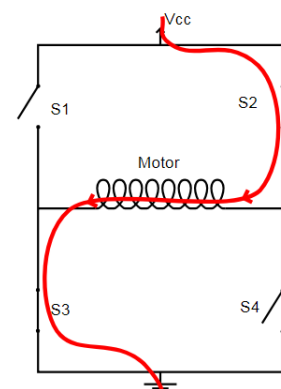


Figura 2.1 c

A sequência das figuras 2.1 acima exemplifica os 3 estados possíveis de uma ponte H

Na primeira imagem (2.1 a) temos todas as chaves do circuito abertas, não havendo corrente na carga (motor). A segunda imagem (2.1 b) consiste no chaveamento apenas das chaves S1 e S4 permitindo a passagem de corrente da esquerda para a direita através da carga constituindo assim o semiciclo positivo. Na terceira imagem temos o acionamento das chaves S2 e S3, o que permite a passagem da corrente no sentido inverso ao da segunda imagem. A corrente percorre a carga da direita para a esquerda, podendo ser considerado assim o semiciclo negativo.

Fica nítido que não se é possível fazer estas inversões de sentido da corrente através de chaves, sendo usado para isso transistores comandado por pulsos de algum tipo de controlador.

Porém devido às limitações físicas de componentes deve se levantar em consideração que a alteração no sentido da corrente não ocorre de forma discreta, passando de zero a um como se pode observar na figura 2.1.

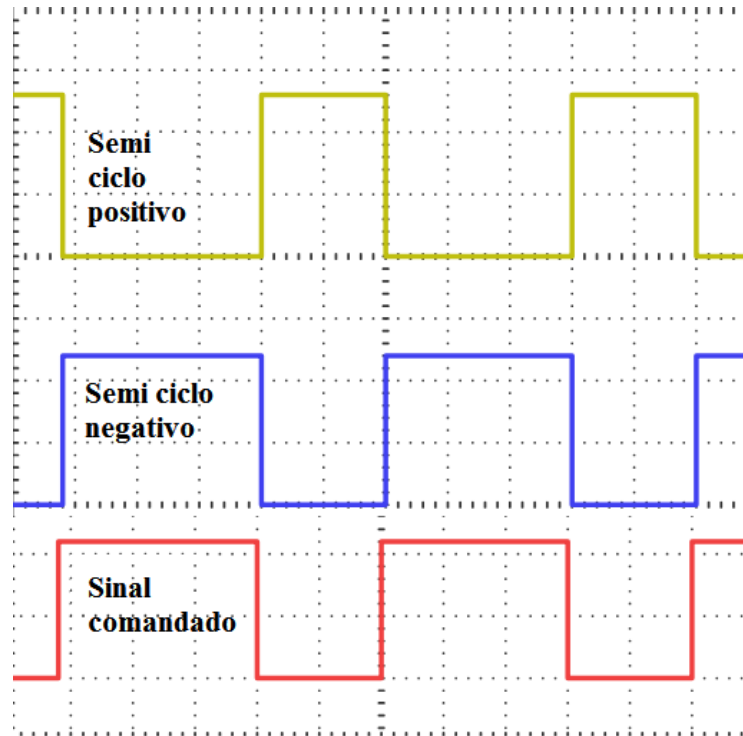


Figura 2.2 - Estados de transmissão dos transistores ideal

Ao se conectar pontas de provas nota-se um pequeno *delay* (atraso) após a mudança de estado que se deve as limitações físicas dos aparelhos.

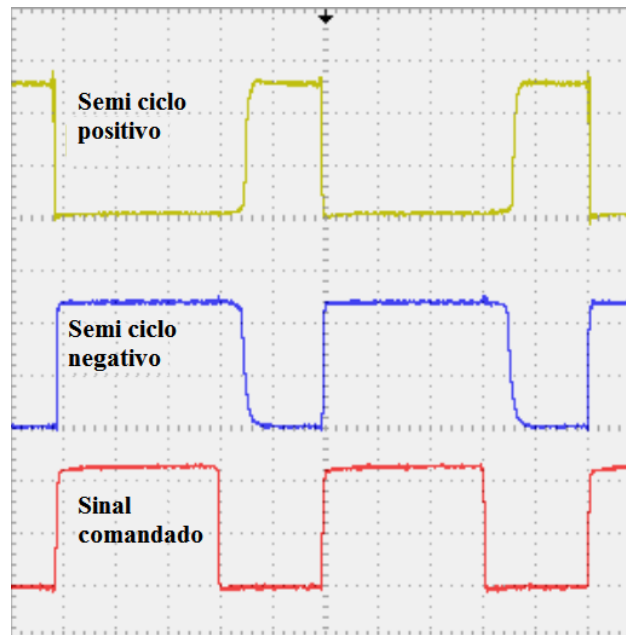


Figura 2.3 - Estados de transmissão dos transistores real

Para fins de estudos, podemos considerar que os períodos de transmissão e decaimentos são contínuos. Para uma melhor interpretação do que vem a ser o tempo morto, conforme mostrado na figura a seguir:

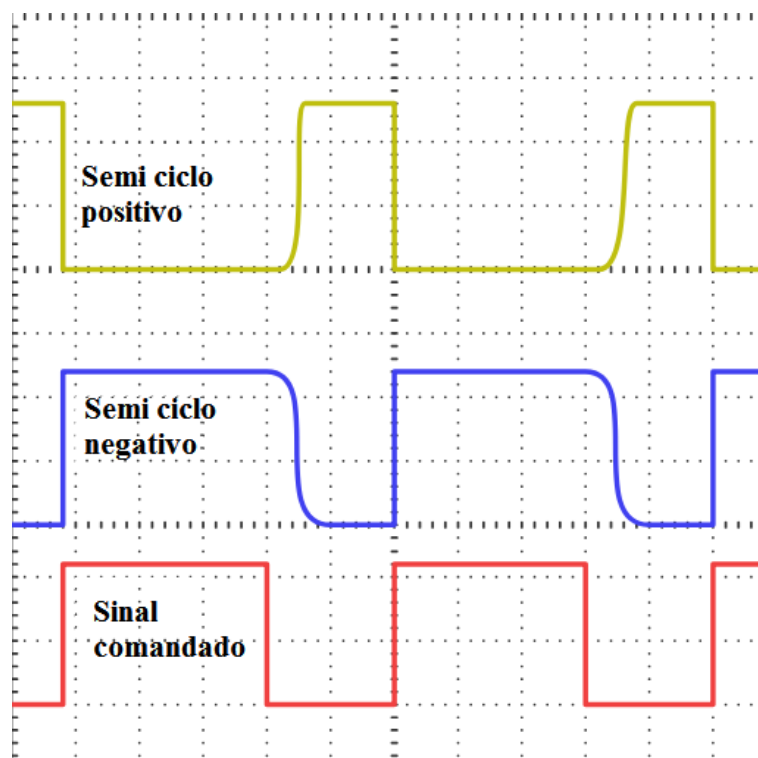


Figura 2.4 - Estados de transmissão dos transistores real contínuos

Para BARBI (2007, p.11), aplicações em potências superiores à 1KVA o inversor de tensão monofásico em ponte completa aparece como a topologia natural pela versatilidade e características tais como baixos esforços de tensão e corrente nos interruptores controlados e a necessidade de isolamento galvânica no comando desses interruptores aparecem como desvantagem dessa topologia

Silva, (2003 apud Patsko 2006, p2) afirma que, para o funcionamento dos conversores em ponte H (Figura 2.4), são necessários quatro dispositivos de chaveamento que trabalham 2 a 2 para garantir a inversão de tensão de alimentação de sua saída alternada, sem que haja curto circuito no barramento de alimentação de corrente contínua, previamente fornecido.

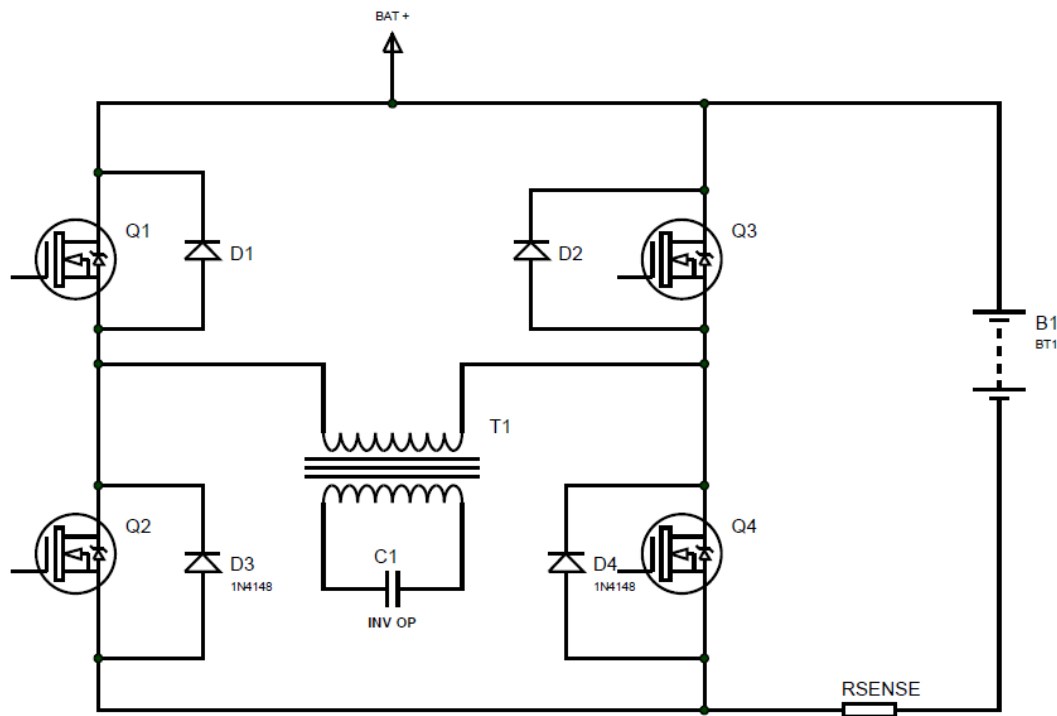


Figura 2.5 – Exemplo Ponte H do inversor de tensão

Segundo Avelar, quando os MOSFETs Q1 e Q4 estão ativos e Q2 e Q3 desligados, o transformador de saída recebe a corrente da bateria em um sentido no qual será reproduzido o semiciclo positivo. Já quando o contrário ocorre Q1 e Q4 desligados e Q3 e Q2 ligados o transformador receberá uma corrente que reproduzirá o semiciclo negativo da forma de onda alternada.

Trabalhando da forma descrita anteriormente, sem nenhum tipo de modulação, a carga “enxergará” uma forma de onda quadrada com apenas dois níveis. Para que se tenha uma forma de onda senoidal na saída, é preciso modular os pulsos fornecidos ao gate dos MOSFETs de

forma variável (AVELAR 2018). Para isso usamos a modulação já citada anteriormente e descrita a seguir conhecida como PWM

2.2 MODULAÇÃO PWM

Para Mezaroba, na maioria das aplicações industriais, necessita-se ter variação de velocidade de um motor a ser acionado, ou até mesmonos casos de produção de energia, onde se necessita transformar corrente contínua oriunda de painéis fotovoltaicos em corrente alternada para uso nos mais diversos dispositivos que utilizem CA. Isso é possível controlando-se a tensão e a frequência, no caso de motores CA.

Pode ser enunciado cinco parâmetros usados para aplicar as técnicas de PWM que consiste em modulação por largura de pulso, que nada mais é do que a comparação de dois sinais de tensão, um de baixa frequência (referência) e o outro de alta frequência (portadora), resultando em um sinal alternado com frequência fixa e largura de pulso variável.

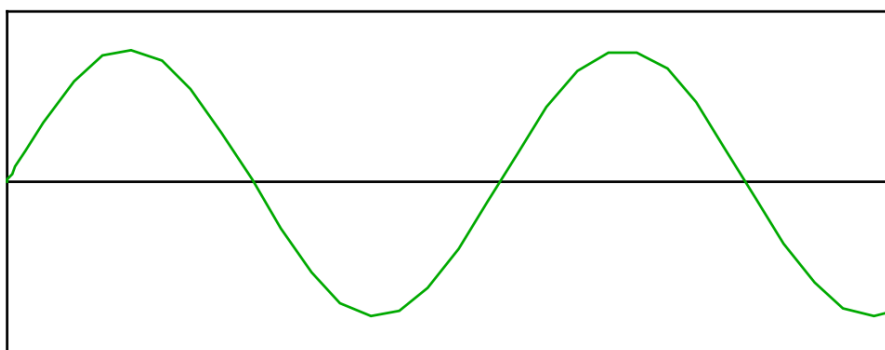
2.2.1 Sinal de Referência

Usada em conversores CC-CC, temos a referência em um sinal contínuo, pois precisamos de um sinal de saída também contínuo.



Figura 2.6 - Sinal de referência de um conversor CC-CC

Para os conversores CC-CA o sinal de referência deve ser senoidal pois o que se busca na saída se trata de uma tensão alternada. Para se determinar a frequência do sinal de saída, leva-se em consideração a mesma frequência para o sinal de referência.



2.2.2 Sinal de Portadora

Considerado um sinal de alta frequência (KHz), que é responsável pela definição da frequência de chaveamento e razão cíclica. Sua ordem deve ser no mínimo 2 vezes maior que o sinal de referência de acordo com o Teorema de Nyquist, mas na prática é necessário pelo menos 10 vezes maior para que se tenha uma boa reprodução do sinal na saída do conversor.

Para os conversores do tipo CC-CC é comum a prática da técnica do sinal dente-de-serra como portadora.

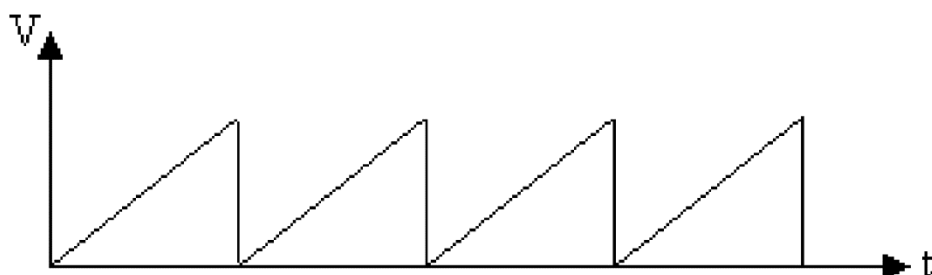


Figura 2.8 – Sinal de portadora dente-de-serra para CC-CC

No caso de conversores CC-CA é comum o uso da técnica do sinal triangular como a portadora. Esta técnica apresenta menor distorção harmônica total do que a portadora dente de serra segundo LIPO (2003, p.114)

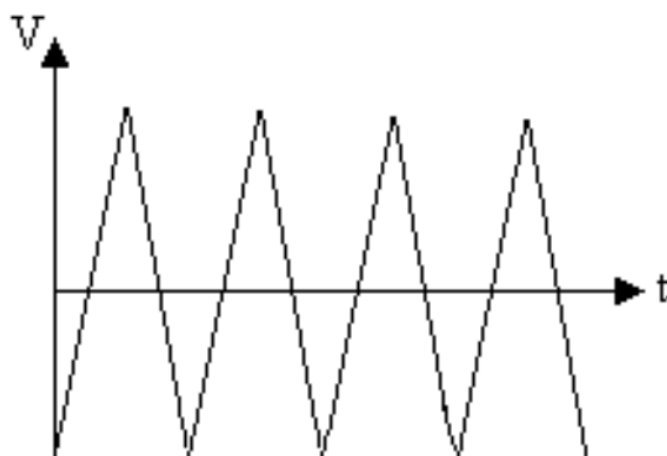


Figura 2.9 - Sinal de portadora triangular para CC-CA

2.2.3 Modulador

O modulador é circuito responsável por fazer a comparação entre o sinal de referência com o sinal da portadora, fazendo variar a largura do pulso na saída do modulador. A figura a seguir trás uma ideia de como o sinal é gerado

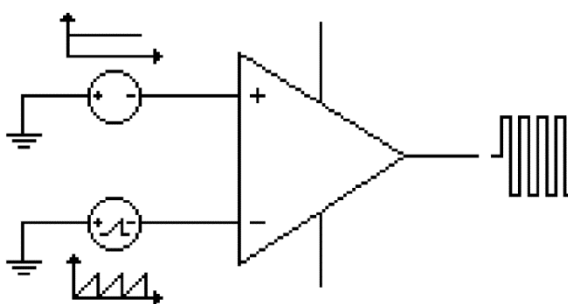


Figura 2.10 - Modulador de sinal PWM

A seguir podem-se observar de maneira mais coesa, as formas de onda de entrada e de saída para um conversor CC-CC

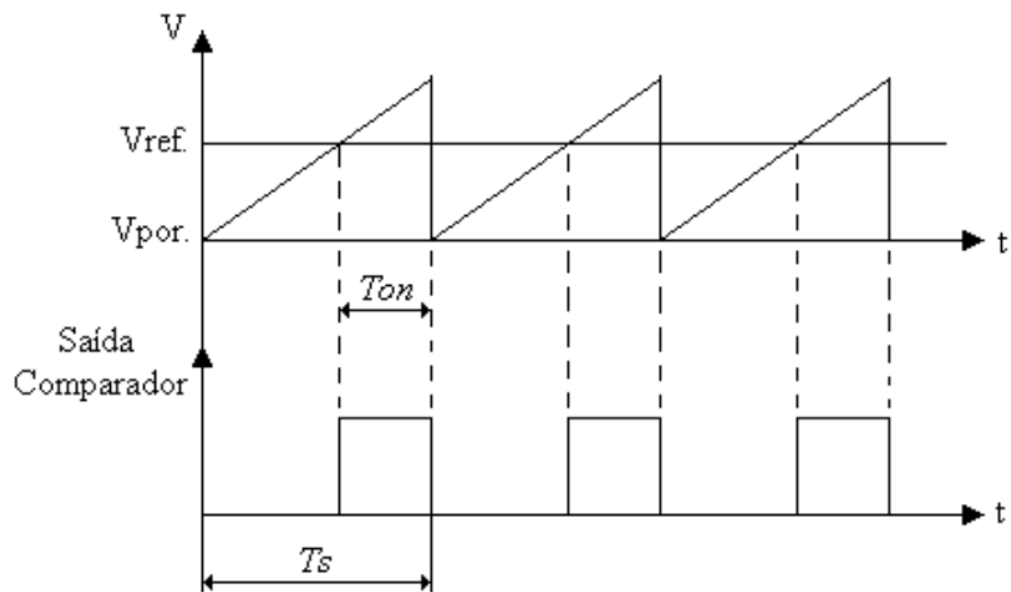


Figura 2.11 - Entrada e saída de um modelador CC-CC

Para um conversor CC-CA as formas de onda de entrada e saída são mostradas na figura 2.12:

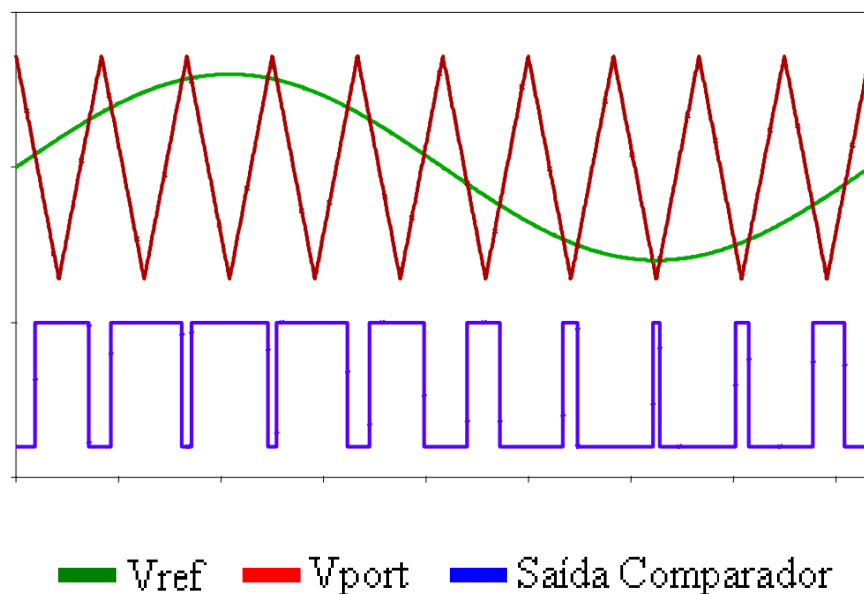


Figura 2.12 - Sinal de entrada e saída do modulador CC-CA

O princípio de funcionamento de um inversor de tensão, está intimamente ligado à estratégia de modulação escolhida para transformar a entrada na saída desejada. Definido por Barbi *et al.* (2007, p.12), as estratégias mais comuns e mais aplicadas são a modulação por pulso único, por largura de pulsos múltiplos e iguais entre si, por largura de pulsos otimizada

(PWM otimizada) e por largura de pulso senoidal (sPWM). Método este último que será estudado neste trabalho.

De acordo com Silva, (2003 apud Patsko 2006, p2) a modulação PWM varia a razão cíclica aplicada aos interruptores em uma alta frequência de comutação com intuito de suprir uma determinada tensão ou corrente na saída em baixa frequência. Deseja-se criar uma sequência de pulsos que devem ter o mesmo valor fundamental de uma referência (neste caso a função seno) definida. Ainda para o autor, nesta sequência de pulsos existem componentes harmônicos indesejados que devem ser minimizados.

O sinal PWM está ligado ao conceito que se chama de *Duty Cycle* (CC) ou ciclo de trabalho que tem por definição, descrever a fração de tempo em que um sistema que possui componentes, sejam eles relés, ou componentes eletrônicos estão em seu estado ativo. Para descobrir esta razão cíclica pode definir o CC conforme mostra a figura 2:

$$\text{duty cycle} = \frac{\tau}{T}$$

Figura 2.13 - Razão para encontrar o *Duty Cycle*

Onde

- τ é o intervalo de tempo no qual a função é não nula de cada período;
- T é o período da função.

Para se ter uma ideia melhor, segue abaixo uma figura que ilustra os CC trabalhando em zero, vinte e cinco, cinquenta, setenta e cinco e cem por cento para uma função constante de frequência (T) de 1 KHz além da tensão média (V_m) fornecida para na saída para uma entrada de 5 volts.

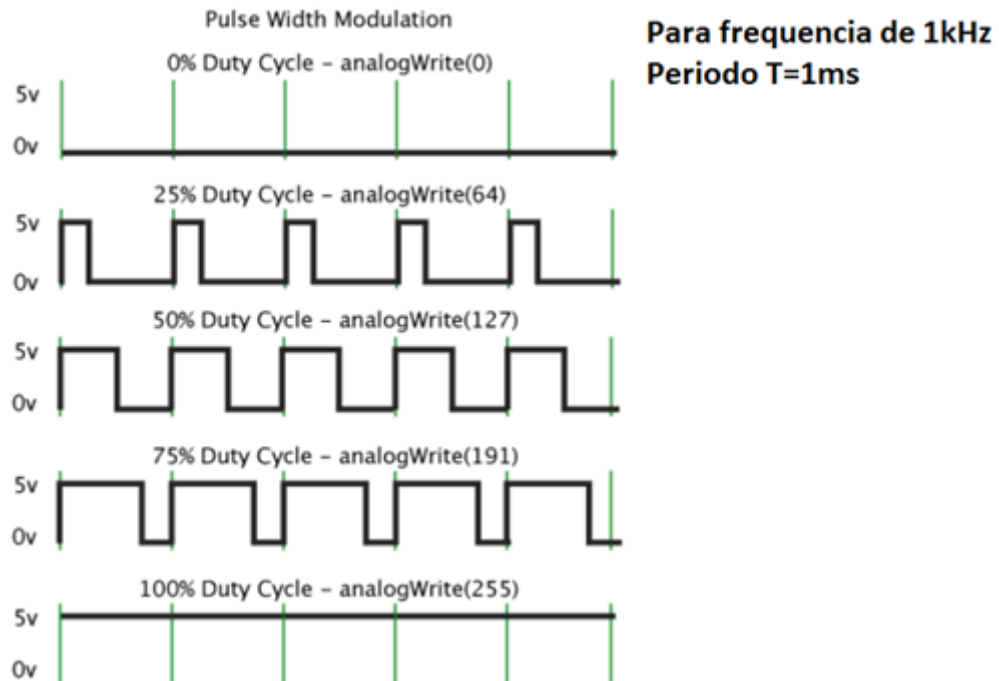


Figura 2.14 - Exemplos de duty cycle para as saídas do Arduino

2.2.4 Geração de sinal complementar

Quando coexistem dois interruptores configurados em um mesmo braço, Faz-se o acionamento das chaves de maneira inversa, ou seja, sempre terá um dispositivo que será complemento do outro. Por exemplo, na figura 2.15 tem-se que quando Q1 conduz, Q2 está cortado (não está em conduzindo), quando Q2 conduz, Q1 fica cortado e assim sucessivamente, o sinal complementar consiste em inverter o sinal modulador. Para isso, pode-se utilizar uma porta lógica NOT:

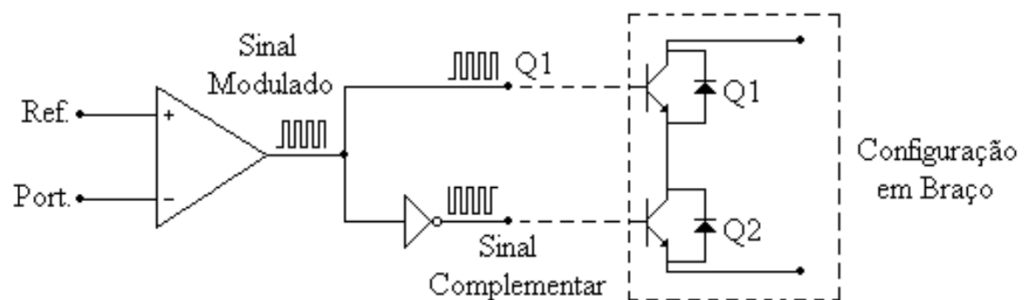


Figura 2.15 - Esquema de ligação do sinal complementar

2.2.5 Tempo Morto (dead time)

Conforme pode se ver nas figuras 2.2 e 2.4, o que idealizou-se na primeira figura e o que de fato foi mensurado na segunda, existe uma clara diferença.

Pode-se notar que os transistores respondem imediatamente a variação do sinal, porém vê-se uma diferença até que de fato o transistor consegue atingir seu próprio nível alto, alcançada na resolução de μs . Observa-se mais atentamente que para a mudança de baixo para alto, decorre em menos tempo para a mudança de alto para baixo. Este fator deve ser observado, pois ao realizar o chaveamento, esse tempo de mudança de estado, impacta diretamente no fato em que por alguns μs todos os transistores estarão ligados simultaneamente. Pode parecer que este efeito não precisa ser levado em consideração, porém ao longo do tempo estes pequenos pulsos de curto circuito podem queimar um regulador ou até mesmo superaquecer algum dos transistores, o que leva também a um desgaste precoce, e mesmo que isso não aconteça, não se é possível escapar dos ruídos que este efeito vá gerar no circuito o que provavelmente atrapalhará o funcionamento correto dos próprios microcontroladores, sensores ou qualquer dispositivo lógico.

Existem possivelmente duas maneiras para se corrigir este problema gerado pelo atraso de descarregamento do transistor. Uma delas é considerar o uso de transistores cada vez mais rápidos, capazes de passar da saturação para o corte em tempo muito curto, porém eles costumam ser mais caros quanto menor for seu tempo de desoperação. Outra alternativa é considerar que os dois conjuntos de transistores não operem de maneira simultânea, colocando um por vez em operação para garantir que o sistema se descarregue totalmente antes da próxima transmissão. A este tempo é denominado tempo morto.

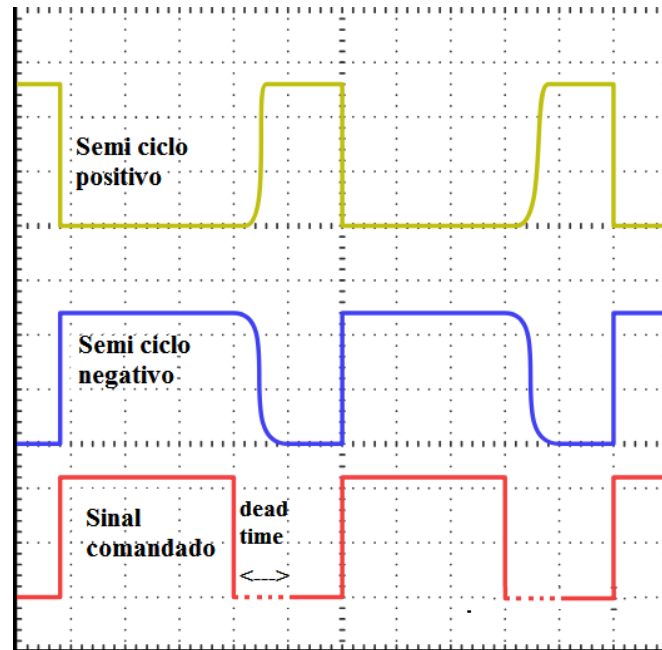


Figura 2.16 – estratégia dead time

2.3 MODULAÇÃO PWM ATRAVÉS DE UM ARDUINO MEGA

Tipicamente, microcontroladores utilizam timers, contadores e comparadores para implementar um sinal PWM (AVELAR, 2018). Para elucidar esta ideia, observe como a base a figura 4 para transcorrer um exemplo.

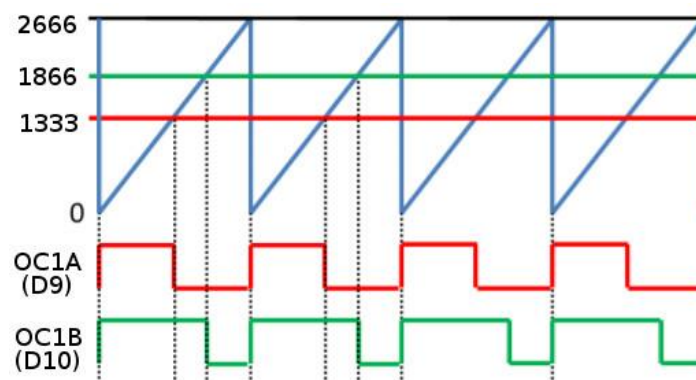


Figura 2.17 - Implementação de um PWM

Seja a onda tipo dente-de-serra na cor azul representando o período do sinal responsável por controlar a cada 2666 ticks de clock o *reset* do controlador quando o mesmo alcança o

overflow. No microcontrolador Arduino Mega1280, os timers correspondem aos registradores ICR1 e ICR3; OC1A e OC1B são comparadores que estão ligados aos pinos D9 e D10. Seja o registradores OCR1A e OCR1B contendo os valores 1333 (50% de CC) e 1866 (75% de CC) respectivamente. Quando OC1A alcança o valor de OCR1A, o pino D9 é “setado” em zero (ou nível baixo). O mesmo acontece com D10 para quando OC1B alcança o valor definido em OCR1B. Para que os pinos possam ser “setados” em um (ou nível alto), é necessário que se atinga o *overflow*, voltando o sinal azul em zero.

2.4 MODULAÇÃO sPWM

Para contornar o problema dos harmônicos, temos a modulação por sPWM, que visa deslocar as harmônicas para altas frequências.

Tendo visto como é possível se gerar um PWM teoricamente e em um microcontrolador e os conceitos que o envolve, o texto a seguir modula o sinal de PWM de acordo com a função seno. Que pode ser definido como SPWM de três níveis, ou sPWM unipolar

Segundo BARBI (2007), um sinal de referência (V_{ref}) é comparado com um sinal triangular V_{tri} na frequência de comutação de modo a se obter os pulsos de comando para os interruptores de um braço do inversor, S1 e S3 (vide figura abaixo) por exemplo. Já os pulsos de comando para o outro braço são obtidos através da comparação do sinal de referência com uma outra portadora triangular V_{tri2} , complementar a V_{tri} . Embora se utilize a portadora triangular, pode-se aplicar uma portadora tipo dente-de-serra, entretanto os resultados apresentados com relação ao espectro harmônico de tensão V_{ab} são inferiores.

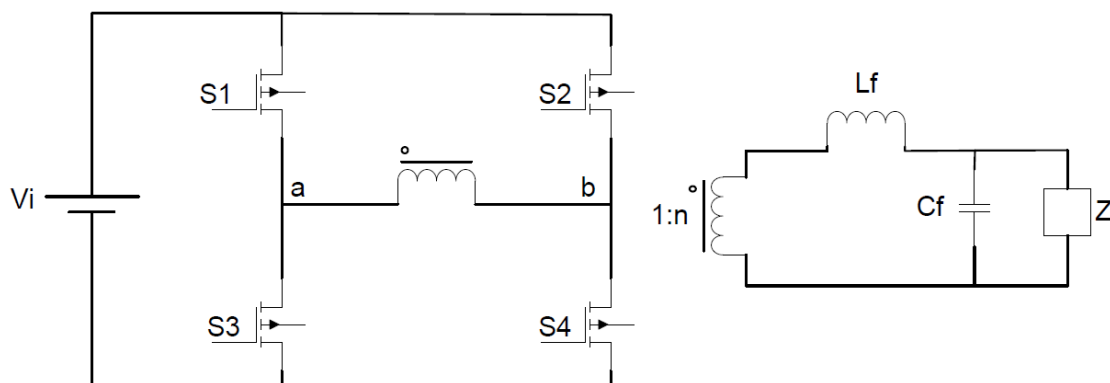


Figura 2.18 - Inversor em ponte completa – Disponível em Projeto de Inversores p.11

A seguir (figura 6), segue esboço detalhado da modulação descrita anteriormente para sPWM em 3 níveis.

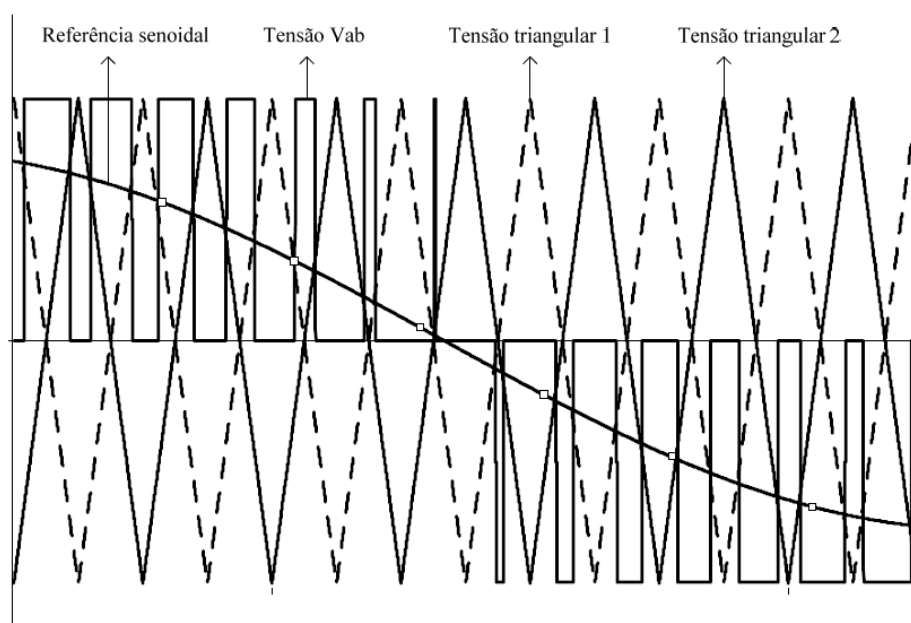


Figura 2.19 - Modulação sPWM em 3 níveis - Disponível em Projeto de Inversores p.13

Para se empregar esta modulação, o inversor apresenta 8 etapas de operação para um ciclo completo. Ou seja, 4 para o semiciclo positivo e 4 para o semiciclo negativo. A seguir serão apresentadas 4 etapas, onde não há inversão da corrente na carga.

A fim de simplificar as etapas de operação, algumas medidas serão adotadas como a retirada do transformador, e a substituição da carga por uma fonte de corrente.

2.4.1 Etapas de Operação

- Primeira etapa: Interruptores S1 e S4 conduzem simultaneamente enquanto S2 e S3 permanecem bloqueados, consequentemente ocorre transferência de energia da fonte V_i para a carga.

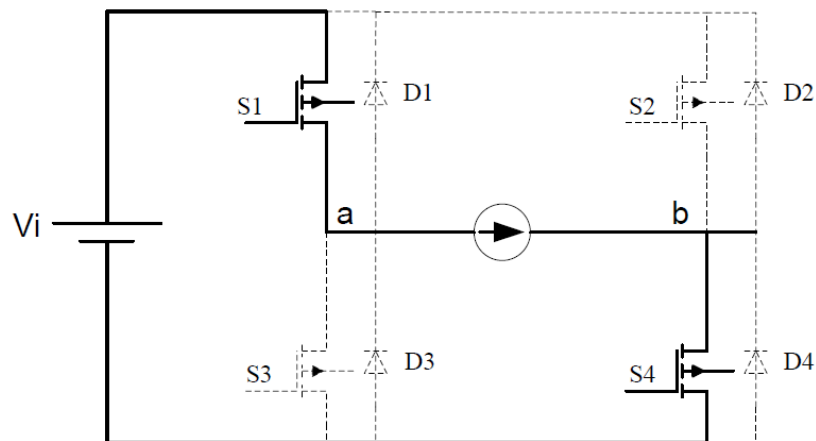


Figura 2.20 - Primeira etapa de operação sPWM - Disponível em Projeto de Inversores p.14

- Segunda etapa: Nesta etapa a fonte V_i deixa de transferir energia para a carga, acontecendo o fenômeno conhecido como roda livre através de S1 e o diodo D2.

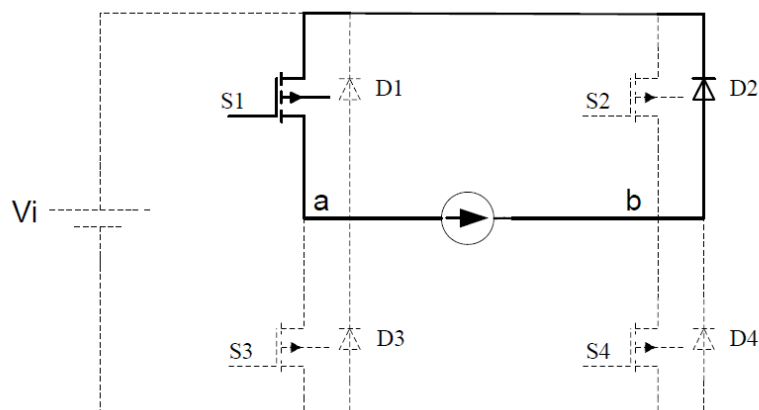


Figura 2.21 - Segunda etapa de operação sPWM - Disponível em Projeto de Inversores p.14

- Terceira etapa: De forma análoga a primeira etapa, porém desta vez, a entrada se dá no condutor S4. S2 e S3 permanecem bloqueados. A esquemática é apresentada na figura 9.

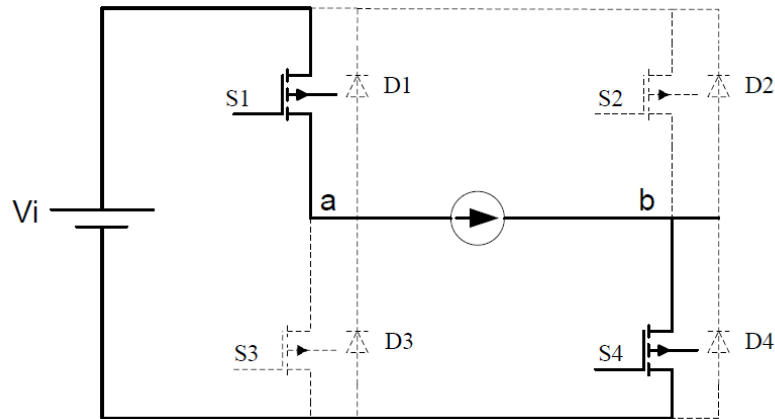


Figura 2.22 - Terceira etapa de operação sPWM - Disponível em Projeto de Inversores p.15

- Quarta etapa: Neste momento a chave S1 é bloqueada deixando o diodo D3 em condução, gerando outra “roda livre” no circuito em questão, S3 chega a ser acionado, porém não chega a conduzir devido ao sentido da corrente conforme nos mostra a figura 2.23.

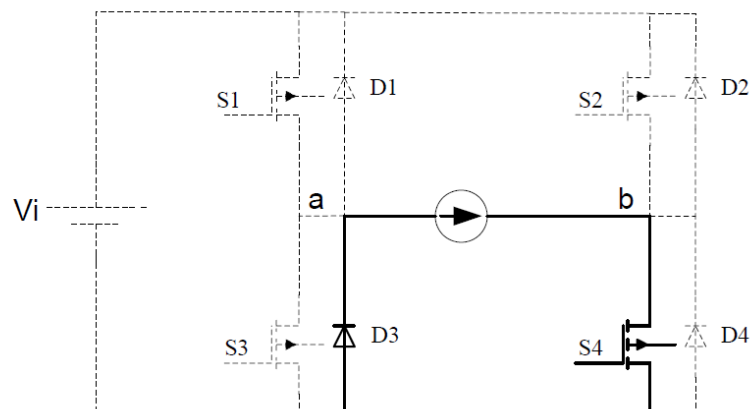


Figura 2.23 - Quarta etapa de operação sPWM - Disponível em Projeto de Inversores p.15

A seguir é apresentado um gráfico um esquemático de que mostra a tensão V_i e o comando dos interruptores S1, S2, S3 e S4.

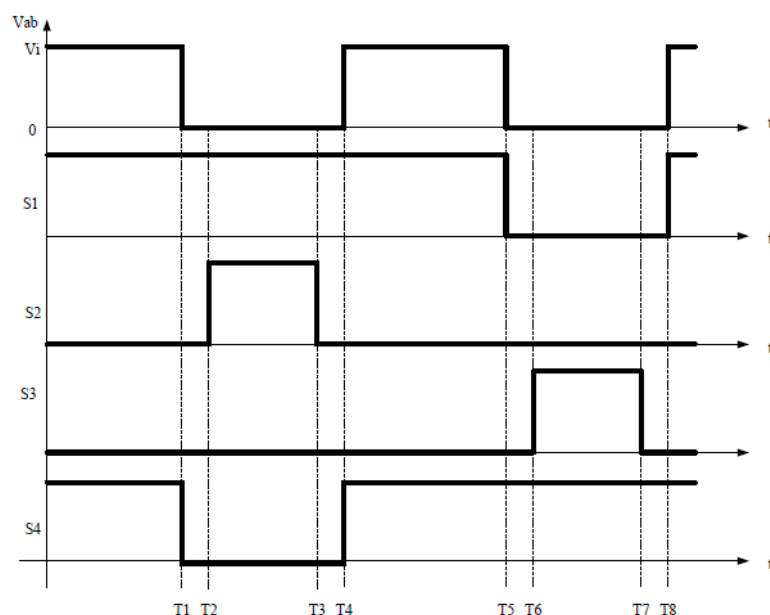


Figura 2.24 - Forma de onda da tensão V_{ab} e comando dos interruptores - Disponível em Projeto de Inversores p.16

Ainda para o autor, no inversor com modulação sPWM unipolar, a frequência de operação do filtro de saída é o dobro da frequência de comutação dos interruptores, portanto, o projeto do filtro de saída deve levar em consideração este fato. Logo se vê a importância da construção de um filtro LC para que o sinal de saída possa se aproximar ao máximo da senoide de referência do inversor. Esta podendo ser a mesma da rede das concessionárias de energia elétrica por exemplo.

2.5 MODULAÇÃO sPWM ATRAVÉS DE UM ARDUINO MEGA

Após explanação teórica vista, para se modular um sinal sPWM em um microcontrolador Arduino MEGA, é preciso “apenas mudar dinamicamente o valor dos registradores OCR1A e OCR1B o que mudará o duty cycle do sinal PWM.” (AVELAR, 2018). Existem duas formas de se obter a variação do valor dos registradores. Uma se dá através da implementação de uma tabela, contendo os valores fixos dos registradores se igualando a uma função seno dentro de uma estrutura de repetição, ou criar um vetor, com esses mesmos valores dentro de uma função *setup* e aloca-lo em memória dinâmica no microcontrolador, o que de fato, gera o mesmo resultado.

A ideia é ter pequenos pulsos nas pontas dos semiciclos (*Duty Cycle* com pequenos ciclos de trabalho) positivos e negativos, e grandes pulsos no meio dos semiciclos positivos e negativos.

A fim de elucidar melhor estas implementações, segue abaixo (Figura 2.25) um gráfico contendo os pulsos gerados nas saídas do microcontrolador e a senoide esperada como resultado no transformador.

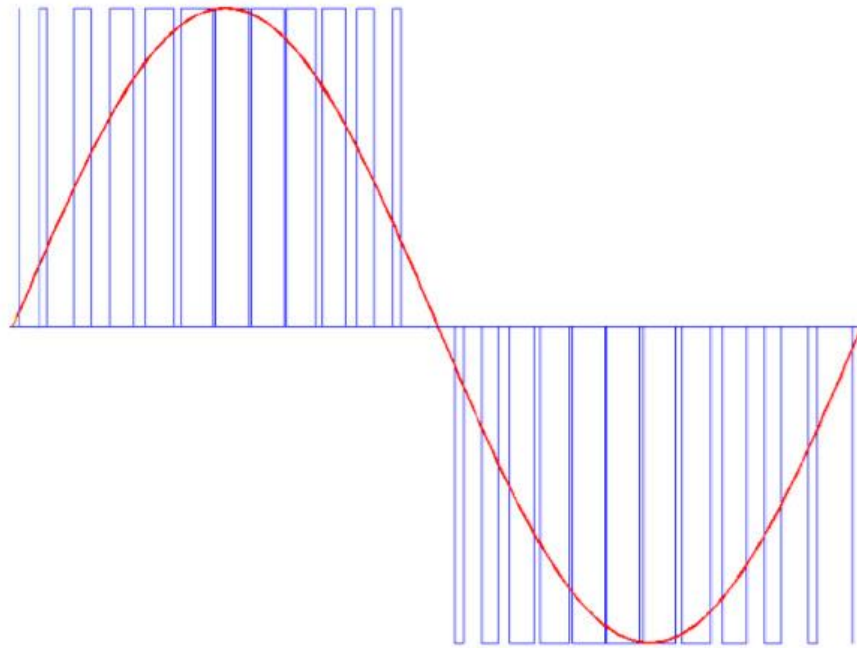


Figura 2.25 -Implementação estratégias sPWM -

ponte H necessita de um circuito que entrega mais que 12V, pois o *source* não está conectado a referência do circuito. Sendo necessária ser de 10 a 15 V maior do que a tensão de alimentação. De acordo com AVELAR (2018), esta tática é conhecida como *bootstrap*, e no circuito é representado pelo diodo de sinal (D1 e D2) (chaveamento rápido), e pelo capacitor de *Tantalum* (C2 e C3) conectado ao *Source* e alimentação do TLP250. Na seção resultados e discussões retomaremos o assunto.

Na saída da ponte, encontra-se um TRAFO (TR1) responsável por elevar a tensão de 12 para 220Vac e um capacitor de 1uf para filtrar a tensão que esta sendo entregue a carga.

3.2 ETAPA QUE MONITORA A TENSÃO DE SAÍDA

No lado secundário de TR1, é conectado um novo TRAFO (TR2) abaixador de tensão, um esquema conhecido como filtro passa baixas, composto por um resistor (R9) e um capacitor (C6) é colocado no secundário de TR2 e o sinal é aplicado a uma ponte de diodos (BR1) afim de obter uma retificação completa do sinal. O capacitor C7 tem a função de eliminar os *ripples* e os resistores R10 e R11 montam o esquema conhecido como circuito divisor de tensão. O capacitor C8 do tipo cerâmico é usado para fazer uma nova filtragem dos ruídos em alta frequência. O esquema é mostrado a seguir:

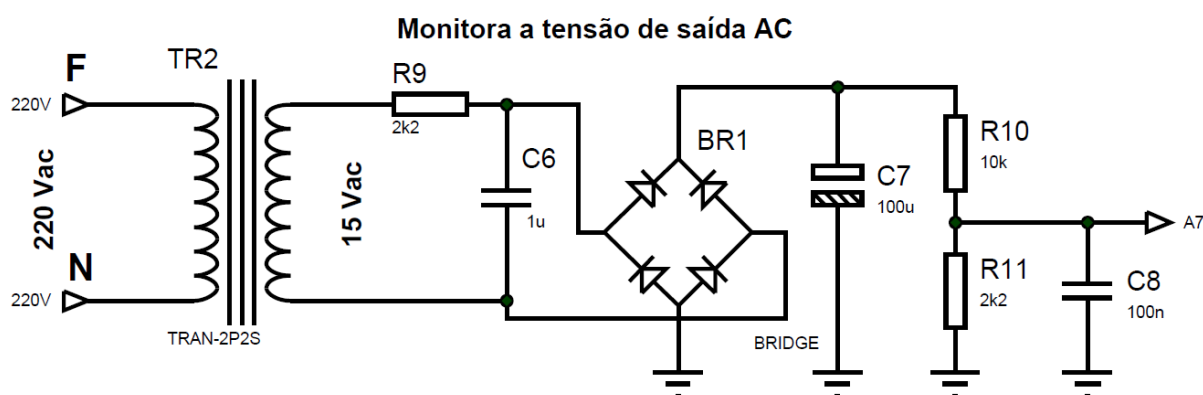


Figura 3.2 - Etapa que monitora a tensão de saída

3.3 ETAPA QUE MONITORA A TENSÃO DA BATERIA

Etapa de simples construção, possui um capacitor (C10) para estabilizar a amostra da tensão, resistores R12 e R13 montando um esquema de divisor de tensão e um capacitor C11 do tipo cerâmico com a finalidade de eliminar os ruídos de alta frequência, segue o esquema da montagem abaixo conforme descrito acima.

Monitora a tensão da bateria DC

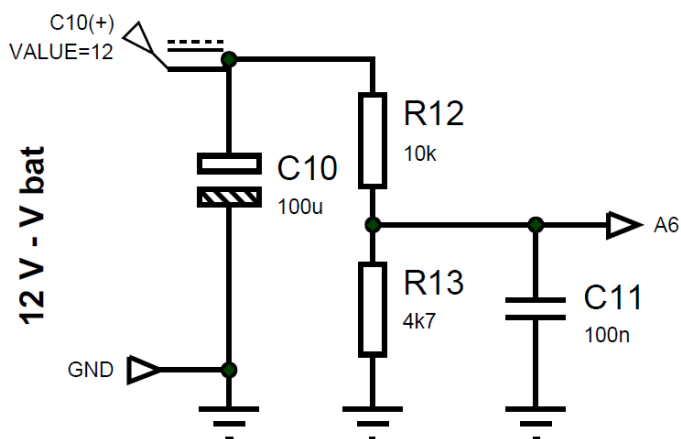


Figura 3.3 – Circuito que monitora a tensão da bateria

3.4 ETAPA QUE MONITORA A CORRENTE CC DE ENTRADA

Para a montagem desta etapa foi usado um CI ACS712, conhecido como um sensor de corrente. Toda corrente contínua que passa pelo circuito é monitorada por ele. O capacitor C12 possui função de desacoplamento e o C9 compõe um filtro para altas frequências

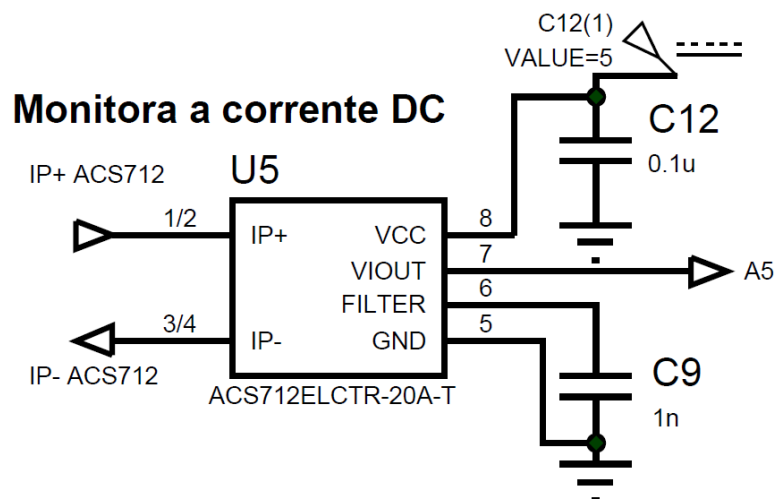


Figura 3.4 - Circuito de monitoração da corrente CC

3.5 MICROCONTROLADOR

Silva (2013, p.4) define brevemente o que seria um microcontrolador e suas vantagens:

Um micro controlador é um componente eletrônico de constituição interna arrojada, o qual possui vários outros componentes interligados de forma a receber instruções, trabalhá-las e responder o processo. A sua utilização reduz o número de componentes externos do circuito, minimizando, assim, o custo do sistema, melhorando a confiabilidade e reduzindo o consumo de energia. Dessa forma, esse componente consegue efetuar com rapidez várias funções, operações lógicas e aritméticas, sob controle de um programa, que dita para a máquina a sequência das funções e operações a serem utilizadas.

O microcontrolador usado neste trabalho é o ARDUINO ATMEGA1280® devido ao número de *timers* disponíveis neste trabalho foram utilizados os *timer* 1 e 3, ambos de 16 bits.



Figura 3.5 - ARDUINO ATMEGA1280®

3.6 ALGORITMO CONTROLADOR

O algoritmo usado para controlar o dispositivo encontra-se em anexo ao final deste trabalho. Vale ressaltar que a declaração dos pinos de saída feita com o uso do bitwise na função OR é apenas para o Arduino Mega.

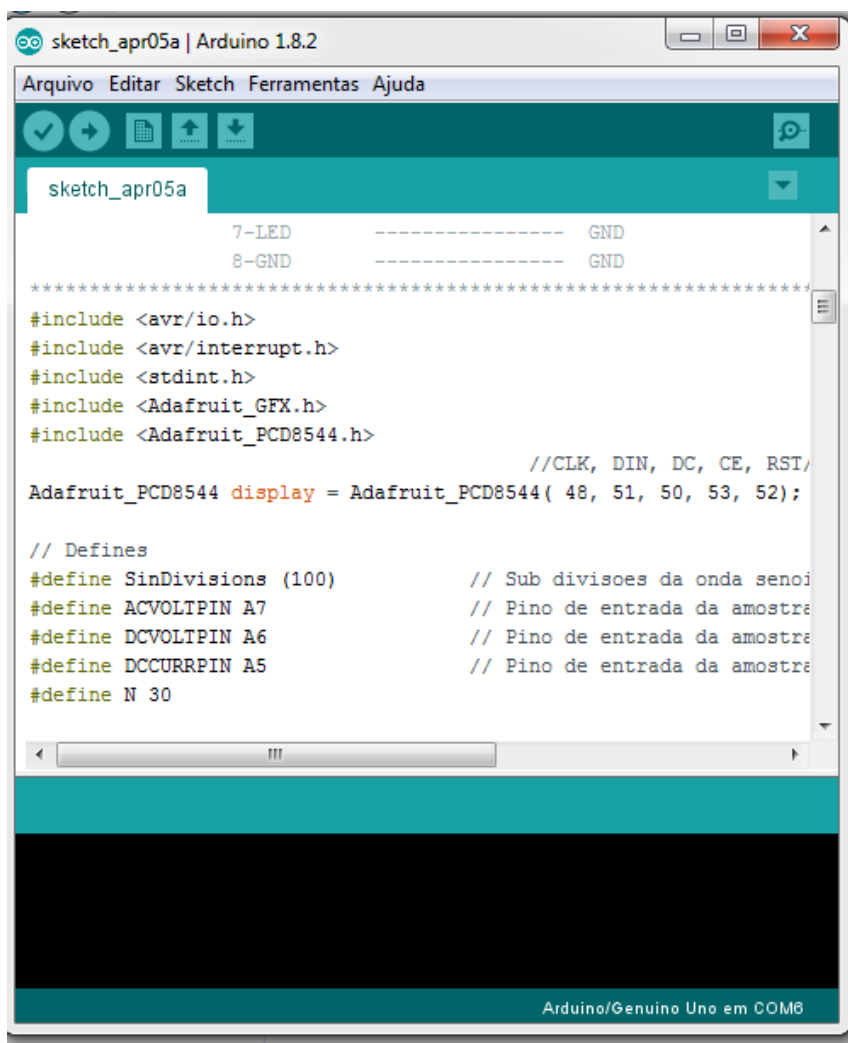


Figura 3.6 - Plataforma de programação

A seguir, uma breve explicação sobre partes relevantes do código, para um melhor entendimento, o software contém variáveis temporárias para o filtro de média móvel, variáveis de leitura do sensor de corrente, declaração dos pinos de entrada analógica do microcontrolador, leituras das variáveis de corrente, cálculo para a formação do *Locatable*, variáveis que são responsáveis pela leitura da tensão CC/CA do circuito, declaração e implementação dos timers utilizados no programa, declaração dos pinos CEIS e por fim a implementação dos pulsos que serão aplicados aos gates de cada mosfet do circuito.

3.6.1 Variáveis Temporárias para o Filtro de Média Móvel

Na plataforma de desenvolvimento para Arduino, para trabalhar com manipulação de dados, é necessário se criar variáveis que recebam valores inteiros e longos, a figura 3.7 evidencia o local de declaração destas, logo no início do programa, após a definição das bibliotecas que possuem funções que serão realizadas ao longo do programa.

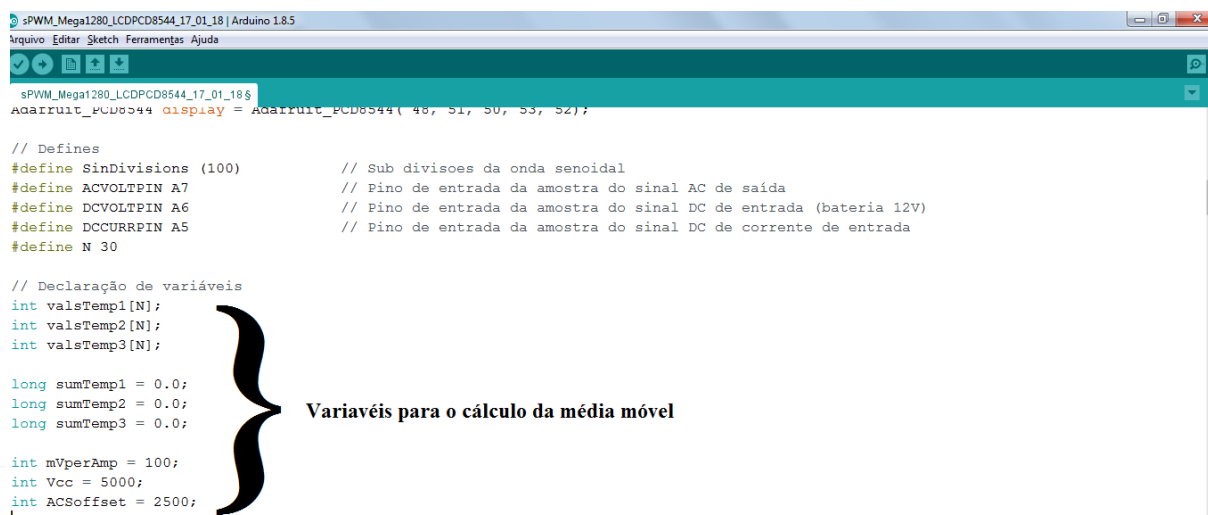


Figura 1.7 - Declaração de Variáveis da média móvel

3.6.2 Variáveis de Leitura do Sensor de Corrente

Para que o microcontrolador possa registrar e apresentar a corrente registrada no protótipo, faz necessária a criação de algumas variáveis que serão necessárias para armazenar os valores transmitidos pelo sensor real CI ACS712

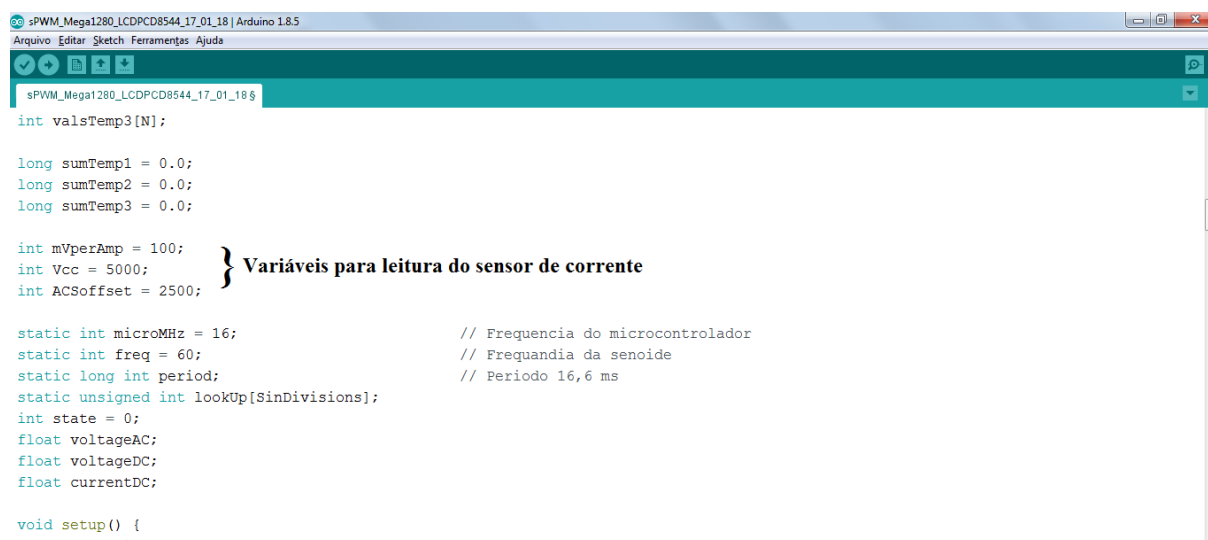


Figura 3.8 – Variáveis que recebem a saída de CI ACS712

3.6.3 Variáveis para Cálculo e Formação do *Locatable*

Para se consolidar a formação da senoide, é necessário declarar variáveis que serão responsáveis por fazer o cálculo e a formação de uma tabela que conterá os valores específicos para correto acionamento das saídas do microcontrolador para a geração da senoide através do chaveamento correto dos MOSFETs

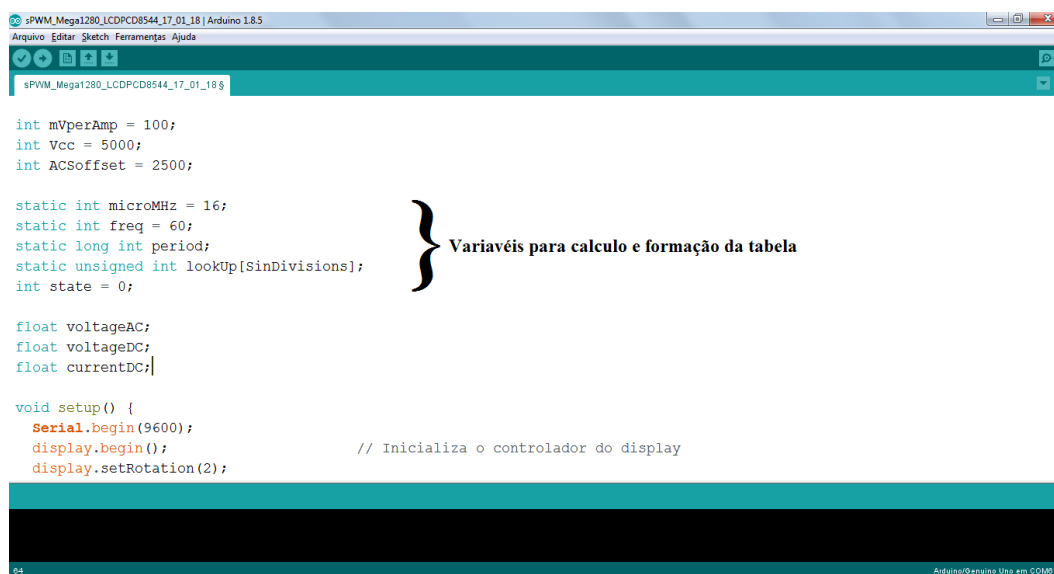


Figura 3.9- Variáveis de cálculo e formação da tabela senoidal

3.6.4 Leituras das Variáveis de Tensão CA/CC e Corrente

As tensões e correntes que se aplicam sobre o sistema também são medidas e apresentada à interface também precisam ser manipuladas pelo software.

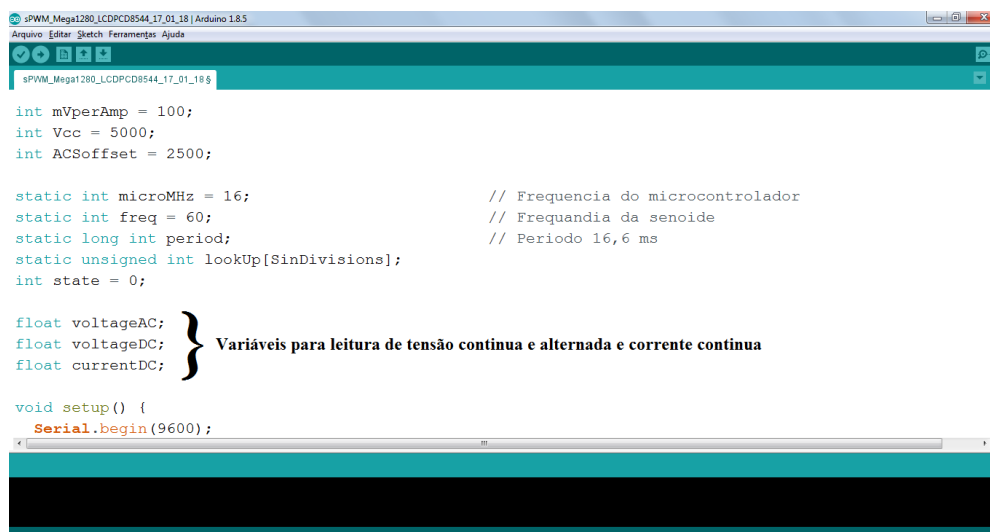


Figura 3.10 - Variáveis que recebem tensão CA/CC e corrente

3.6.5 Declaração e Implementação dos Timers

Os timers do sistema são responsáveis por manter as portas acionadas uma determinada quantidade de tempo para perfazer uma forma senoidal vista pelas saídas, por isso torna-se necessárias a implementação deles, conforme mostra trecho abaixo na figura 3.11

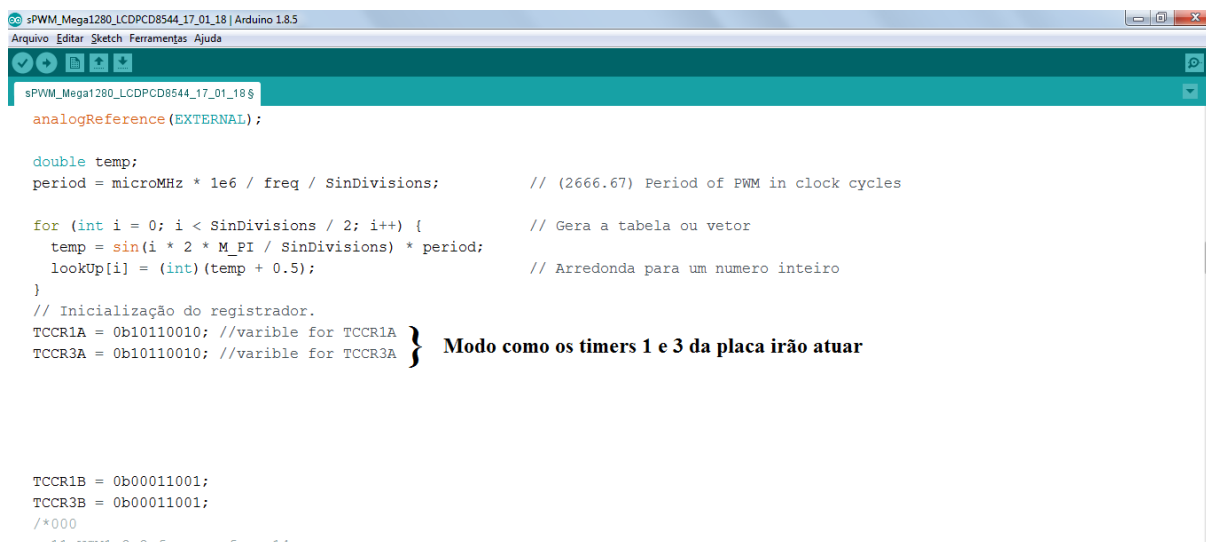
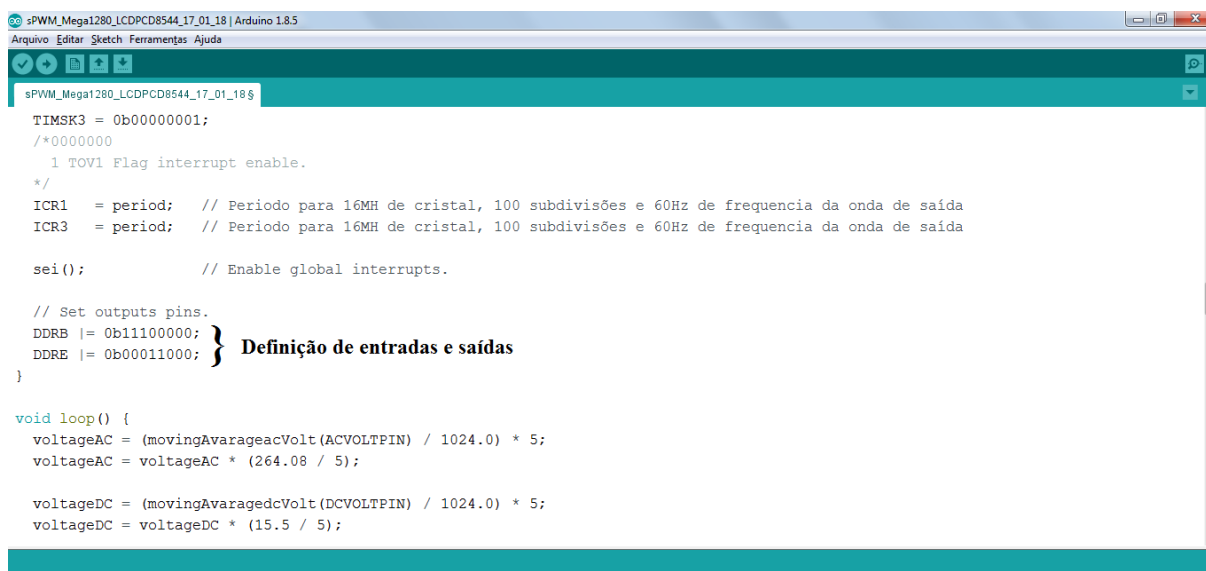


Figura 3.11 - Inserção dos timers do sistema

3.6.6 Declaração dos Pinos de Entrada e Saída

No software, é necessário dizer quais pinos serão considerados entradas e quais pinos deverão ser considerados saída para a referência vindos através do esquema de montagem do protótipo. A figura 3.12 elucida sua criação:



```
sPWM_Mega1280_LCDPCD8544_17_01_18 $
TIMSK3 = 0b00000001;
/*00000000
 1 TOV1 Flag interrupt enable.
*/
ICR1 = period; // Período para 16MH de cristal, 100 subdivisões e 60Hz de frequência da onda de saída
ICR3 = period; // Período para 16MH de cristal, 100 subdivisões e 60Hz de frequência da onda de saída

sei(); // Enable global interrupts.

// Set outputs pins.
DDRB |= 0b11100000;
DDRE |= 0b00011000; } Definição de entradas e saídas
}

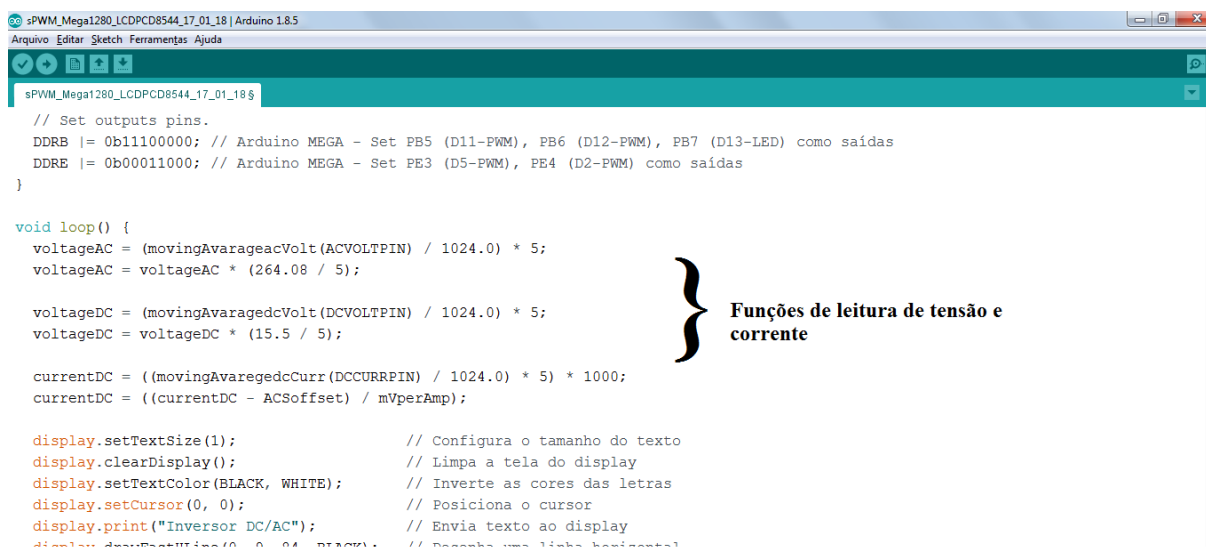
void loop() {
  voltageAC = (movingAverageacVolt(ACVOLTYPIN) / 1024.0) * 5;
  voltageAC = voltageAC * (264.08 / 5);

  voltageDC = (movingAveragedcVolt(DCVOLTYPIN) / 1024.0) * 5;
  voltageDC = voltageDC * (15.5 / 5);
```

Figura 3.12 - Entradas e saídas dos pinos do Arduino

3.6.7 Funções que Fazem a Leitura do Circuito

Na figura abaixo pode-se observar o trecho de código responsável por receber os valores que serão exibidos no LCD para a tensão e a corrente do circuito.



```
sPWM_Mega1280_LCDPCD8544_17_01_18 $
// Set outputs pins.
DDRB |= 0b11100000; // Arduino MEGA - Set PB5 (D11-PWM), PB6 (D12-PWM), PB7 (D13-LED) como saídas
DDRE |= 0b00011000; // Arduino MEGA - Set PE3 (D5-PWM), PE4 (D2-PWM) como saídas
}

void loop() {
  voltageAC = (movingAverageacVolt(ACVOLTYPIN) / 1024.0) * 5;
  voltageAC = voltageAC * (264.08 / 5);

  voltageDC = (movingAveragedcVolt(DCVOLTYPIN) / 1024.0) * 5;
  voltageDC = voltageDC * (15.5 / 5);

  currentDC = ((movingAveragedcCurr(DCCURRYPIN) / 1024.0) * 5) * 1000;
  currentDC = ((currentDC - ACSoffset) / mVperAmp);

  display.setTextSize(1); // Configura o tamanho do texto
  display.clearDisplay(); // Limpa a tela do display
  display.setTextColor(BLACK, WHITE); // Inverte as cores das letras
  display.setCursor(0, 0); // Posiciona o cursor
  display.print("Inversor DC/AC"); // Envia texto ao display
  display.setCursor(0, 1); // Desenha uma linha horizontal
```

Figura 3.13 - Funções que fazem a leitura do circuito

3.6.8 Função Média Móvel

O filtro de média móvel programado no sistema é propriamente para a tensão alternada, contínua e para a corrente do circuito. Devido ao ruído da forma da corrente, decorrente das limitações de implementação, define-se uma constante N de 30 unidades de leitura para cada ciclo do sistema, a média apresentada é a soma de todos eles dividido pelo número de amostras coletadas naquele ciclo, esta constante tem por função atuar na estabilidade da leitura encontrada. Passe-se como parâmetro o pino de entrada de onde se vai fazer a leitura e armazena os N valores encontrados em um vetor que são somados em uma variável e depois é dividido pelo número de amostras. Segue abaixo a forma de implementação utilizada para o cálculo e armazenamento das leituras ao longo de um dos ciclos:

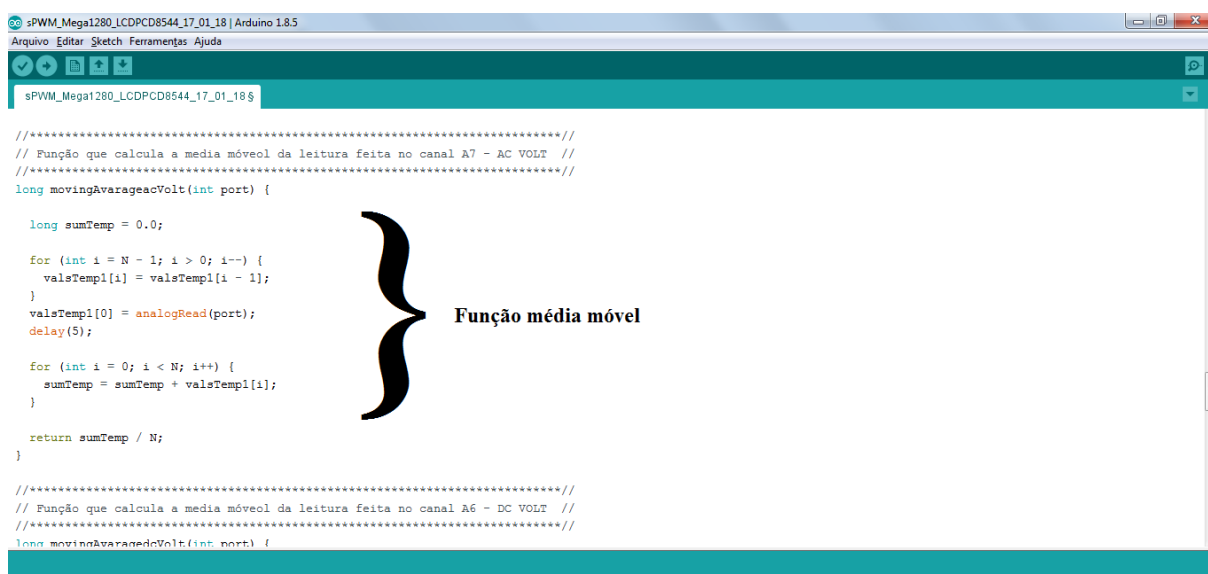


Figura 3.14 – Implementação da média móvel para CC

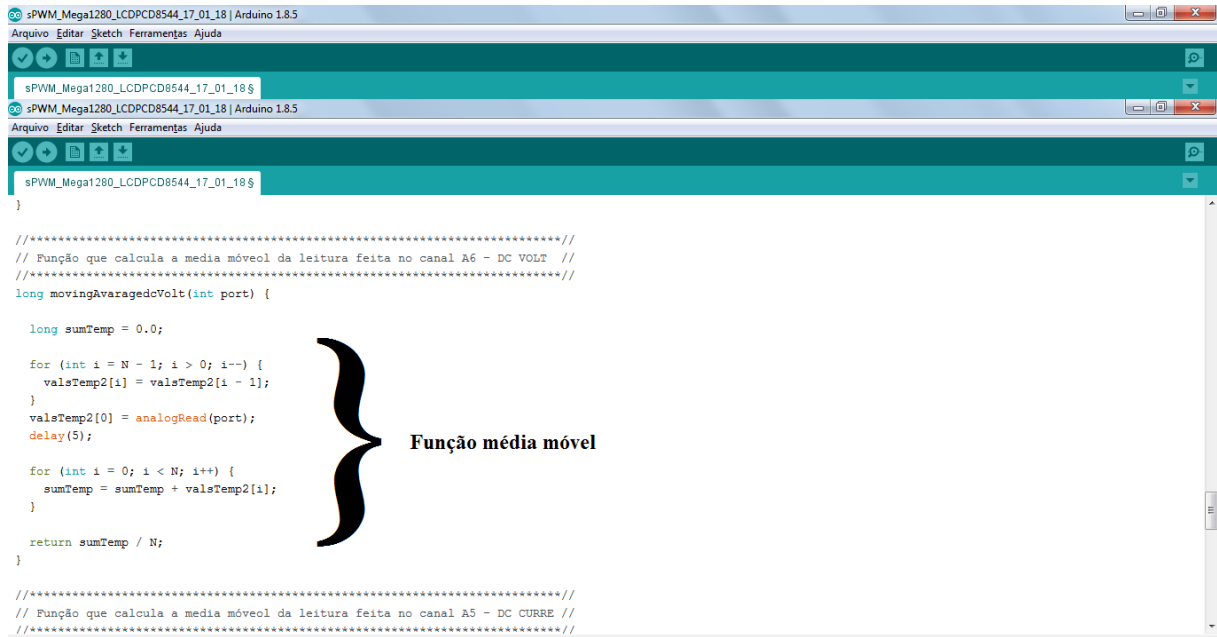


Figura 3.15 - Implementação da média móvel para CA

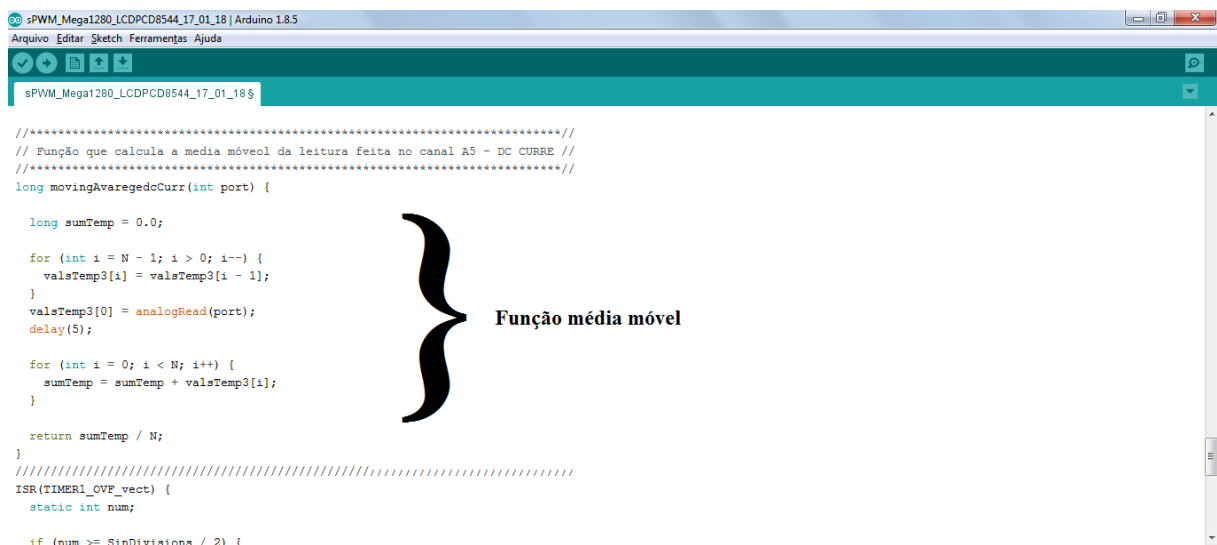


Figura 3.16 - Média móvel para CC

3.6.9 Atuação nos Pinos do MOSFETs

O Programa é responsável também por acionar as saídas do microcontrolador em um tempo pré determinado pela formação da tabela que guarda os valores da senoide e é definido o estado lógico para cada semiciclo.

No estado positivo, $state = 0$, o MOSFET Q3 se ativa através da estratégia de modulação e Q4 gerando os complementos de Q3, enquanto o MOSFET Q1 está desligado e seu complemento Q2 esta ativado, quando $state = 1$, entra-se no estado negativo, entrando em atuação o MOSFET Q1 realizando a estratégia de modulação enquanto Q2 gera sinais de complemento, Q3 permanece desligado e seu complemento Q4 ativado. Pode-se observar o circuito da figura 3.1.

```

sPWM_Mega1280_LCDPCD8544_17_01_18$
return sumTemp / N;
}
///////////////////////////////////////////////////////////////////
ISR(TIMER1_OVF_vect) { -> Início da função timer
    static int num;

    if (num >= SinDivisions / 2) {
        num = 0;          // Reset num
        state ^= 1;
    }

    PORTB ^= 0b10000000; // Arduino MEGA - Faz um toggle no pino PB0 - D8 (LED)
                        //sem afetar os outros pinos utilizando uma XOR
    }

    if (state == 0) {
        OCR1A = 0; // D11 - Q1 -> OFF
        OCR1B = 0; // D12 - Q2 -> ON (Complemento do D11)

        OCR3A = lookUp[num]; // D5 - Q3 -> PWM
        OCR3B = lookUp[num]; // D2 - Q4 -> PWM Complemento
        num++;
    }

    if (state == 1) {
        OCR1A = lookUp[num]; // D11 - Q1 -> PWM
    }
}

```

Figura 3.17 - Função TIMER de atuação dos MOSFETs



Figura 3.18 - Técnica utilizada para chaveamento dos MOSFET

3.7 SOFTWARE DE SIMULAÇÃO

O software usado para realizar as etapas de simulação, pertence a empresa LabCenter Eletronics Ltd chamado *Proteus Design Suite®*, Possui 2 IDE (Ambiente de Desenvolvimento Integrado) uma que permite criação de circuitos (ISIS Professional) e outra para elaborar a placa de circuito impresso (ARES Professional), totalmente integrado, além de permitir simular diversos microcontroladores, podendo executar o firmware, acessando os registradores, além de opções de Debug, amplamente utilizado para testar antes de montar o projeto real em uma placa Protoboard.

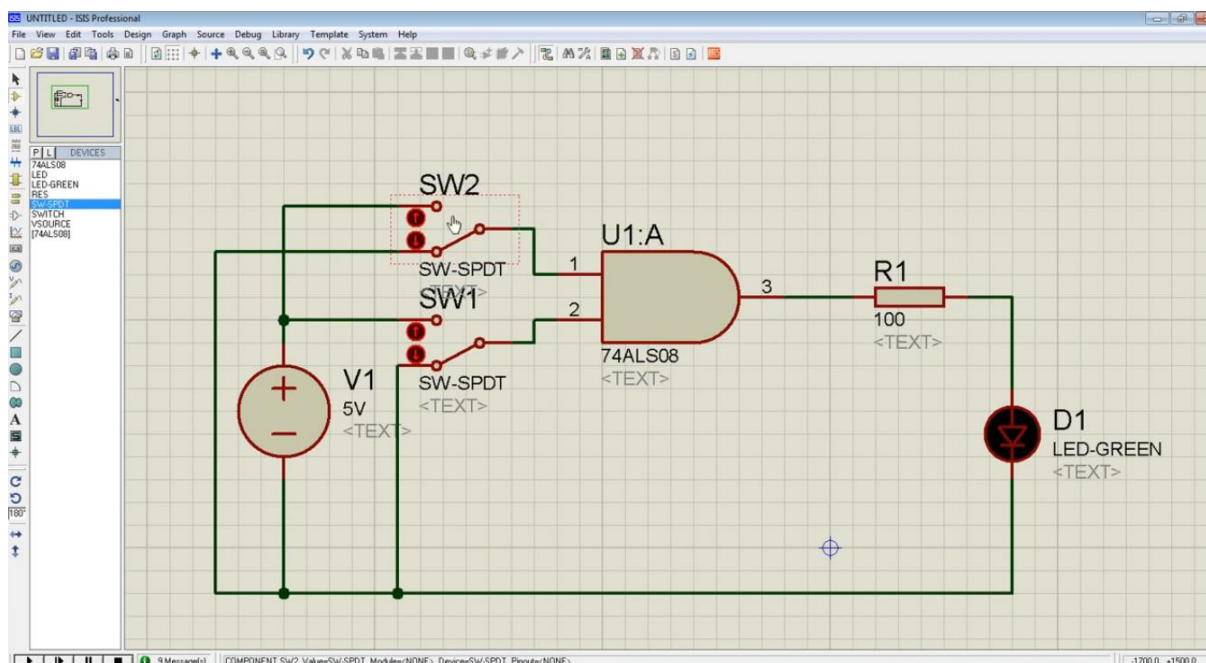


Figura 3.19 - Exemplo de circuito básico montado no Proteus

Com o intuito de se ter uma visão geral do esquema de montagem do inversor, tão como a possibilidade de se aplicar simulações nos circuitos mostrados anteriormente nas figuras 3.1, a 3.4, foi utilizado um software de eletrônico chamado *Proteus Design Suite®*, desenvolvido por Labcenter Electronics Ltd.

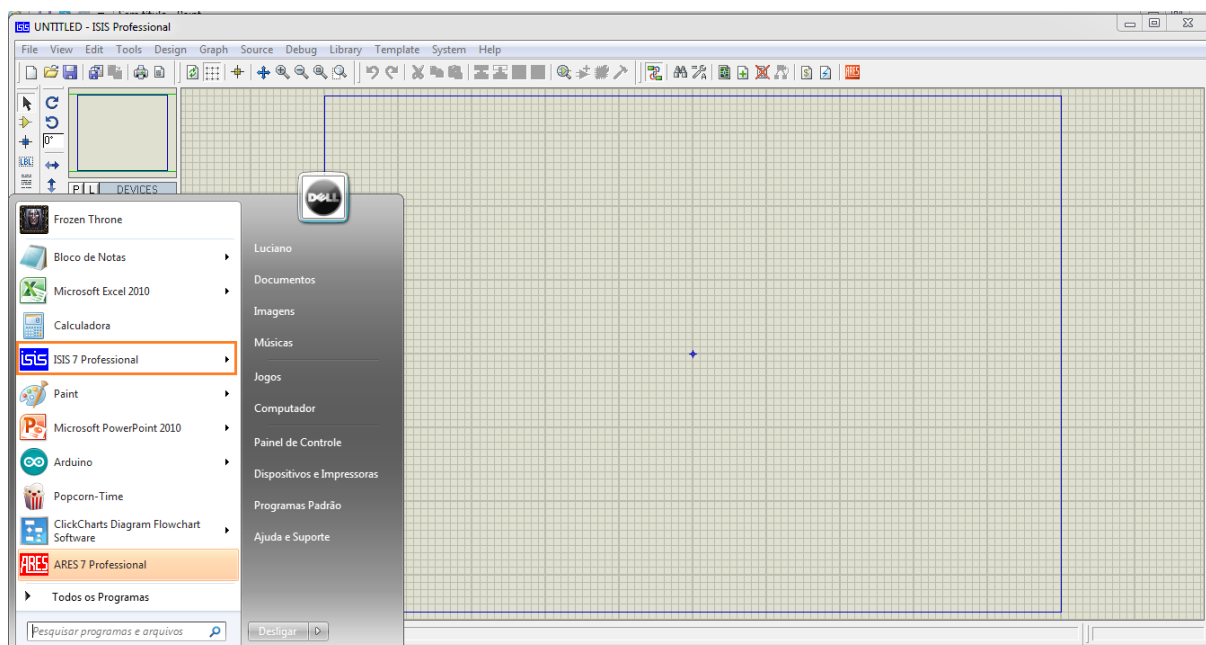


Figura 3.20 - Plataforma para simulação do circuito

3.8 MONTAGEM

Após apresentado todos os componentes, circuitos, códigos, feita as devidas simulações, é hora de montar o projeto na prática, fazendo-se uso de recursos reais, com todas as limitações físicas presentes em cada componente a fim de por a prova, todo conteúdo estudado. Os resultados serão apresentados no capítulo seguinte. A imagem a seguir mostra a montagem de uma forma geral. Consegue observar todos os conjuntos em um mesmo plano, identifica-se em [1] a carga resistiva na saída alternada do inversor, em [2] tem-se a etapa de elevação de potência e leitura da tensão CA, em 3 temos a etapa de potência do protótipo onde ocorre o sequenciamento da ativação dos MOSFETs, em [4] se tem o arduíno MEGA1280, onde esta inserido o código responsável por tratar todos os dados colhidos e comandar todos os chaveamentos responsáveis para o funcionamento objetivado, em [5] temos o visor LCD para acompanhamento das tensões de entrada continua, saída alternada e corrente interpretadas pelo microcontrolador, em [6] temos a etapa de leitura de corrente, em [7] a etapa de leitura de tensão CC, em [8] a etapa de leitura da tensão CA, todos os três últimos referenciados são responsáveis por enviar sinais ao microcontrolador para ser tratado pelo algoritmo, e em [9], a fonte utilizada para alimentar o inversor em tensão contínua

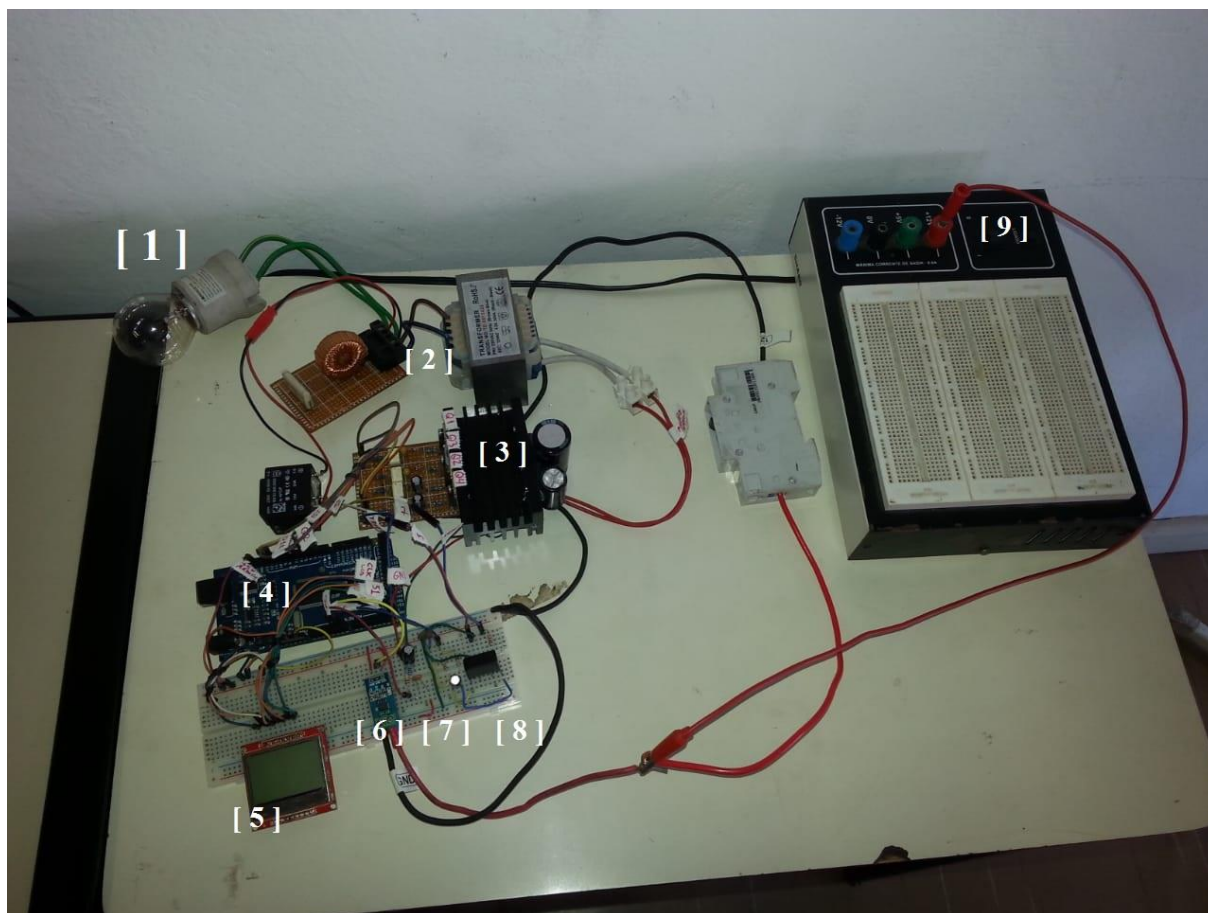


Figura 3.21 - Montagem do Projeto em Laboratório de Eletrônica Analógica Digital-UFOP

De uma forma mais detalhada é possível observar na figura 3.22 o local em específico onde temos a etapa de leituras dos sinais colhidos no protótipo, em [1] a etapa de que monitora a corrente através de um CI ACS712, em [2] o circuito responsável por fazer a leitura CC vinda da fonte de tensão em corrente contínua, e em [3] o circuito responsável por fazer a leitura da tensão de saída CA. Todos estes dados obtidos são enviados para o microcontrolador que será o responsável por processar e enviar para a interface LCD.

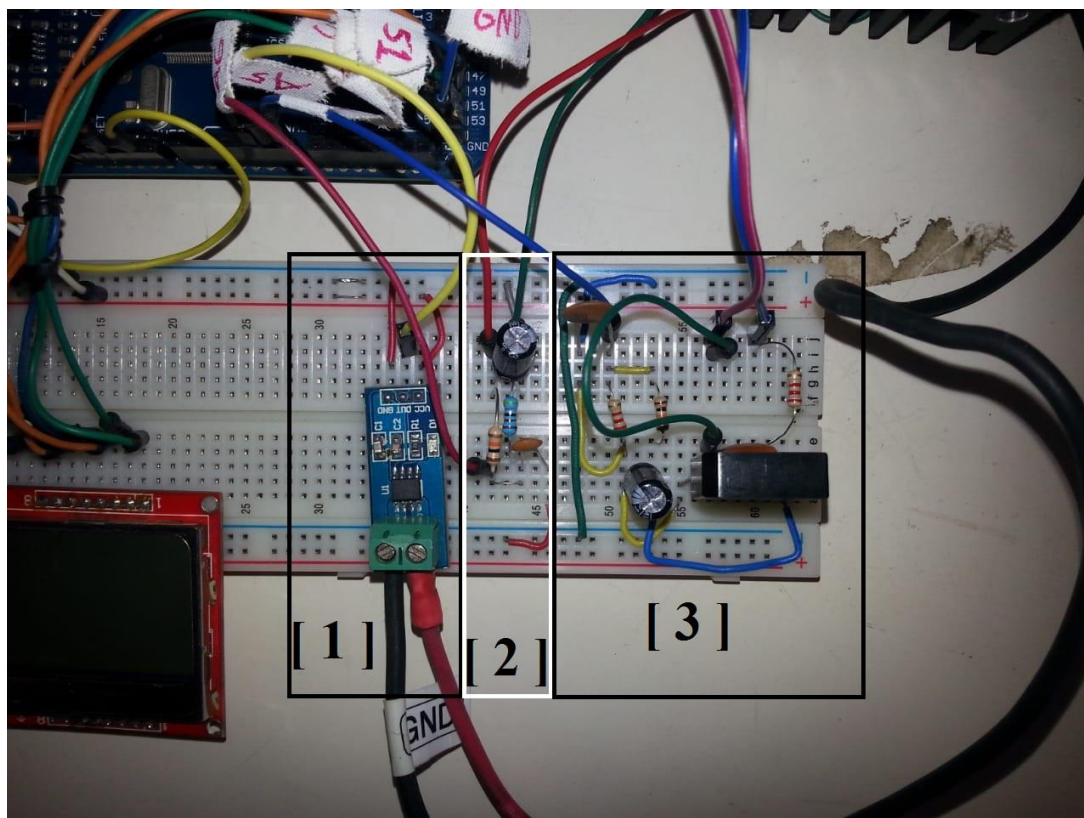


Figura 3.22 – Etapa de leitura dos sinais

Observando de forma detalhada também a etapa de potência, nela chaga-se a tensão continua e é possível observar na saída uma tensão alternada em baixa tensão. O microcontrolador envia sinais aos drivers TLP250 que permite o envio de um pulso ao pino central de um MOSFET, permitindo assim a passagem da corrente. Existe um TLP250 para cada MOSFET utilizado no circuito conforme pode-se observar na figura 3.1

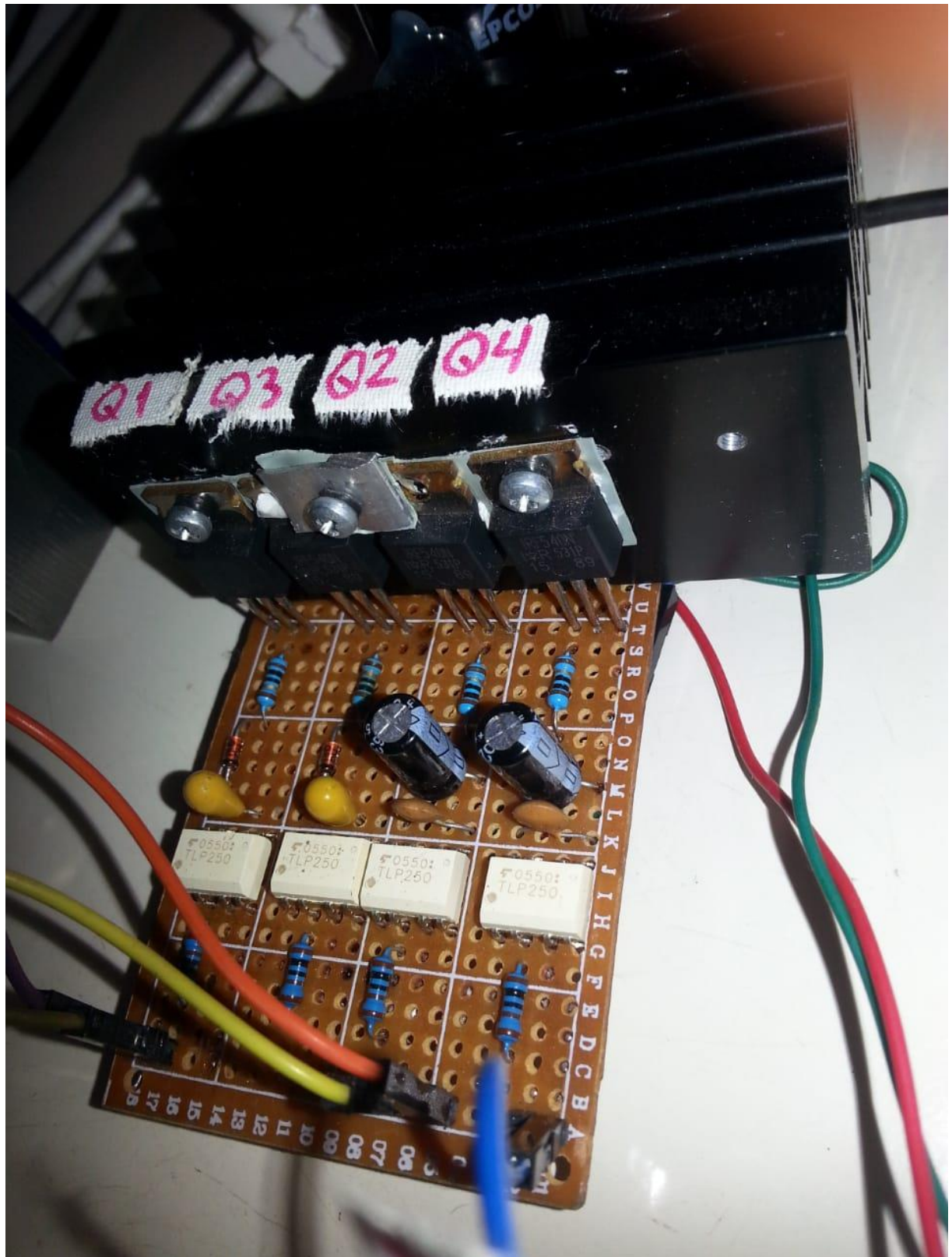


Figura 3.23 - Etapa de potência

Com o sinal de saída alternado oriundo da etapa de potência, um transformador que eleva a tensão de 12 V para 220 V e entrega a saída do circuito onde se pode conectar cargas alternadas e eletrônicas.

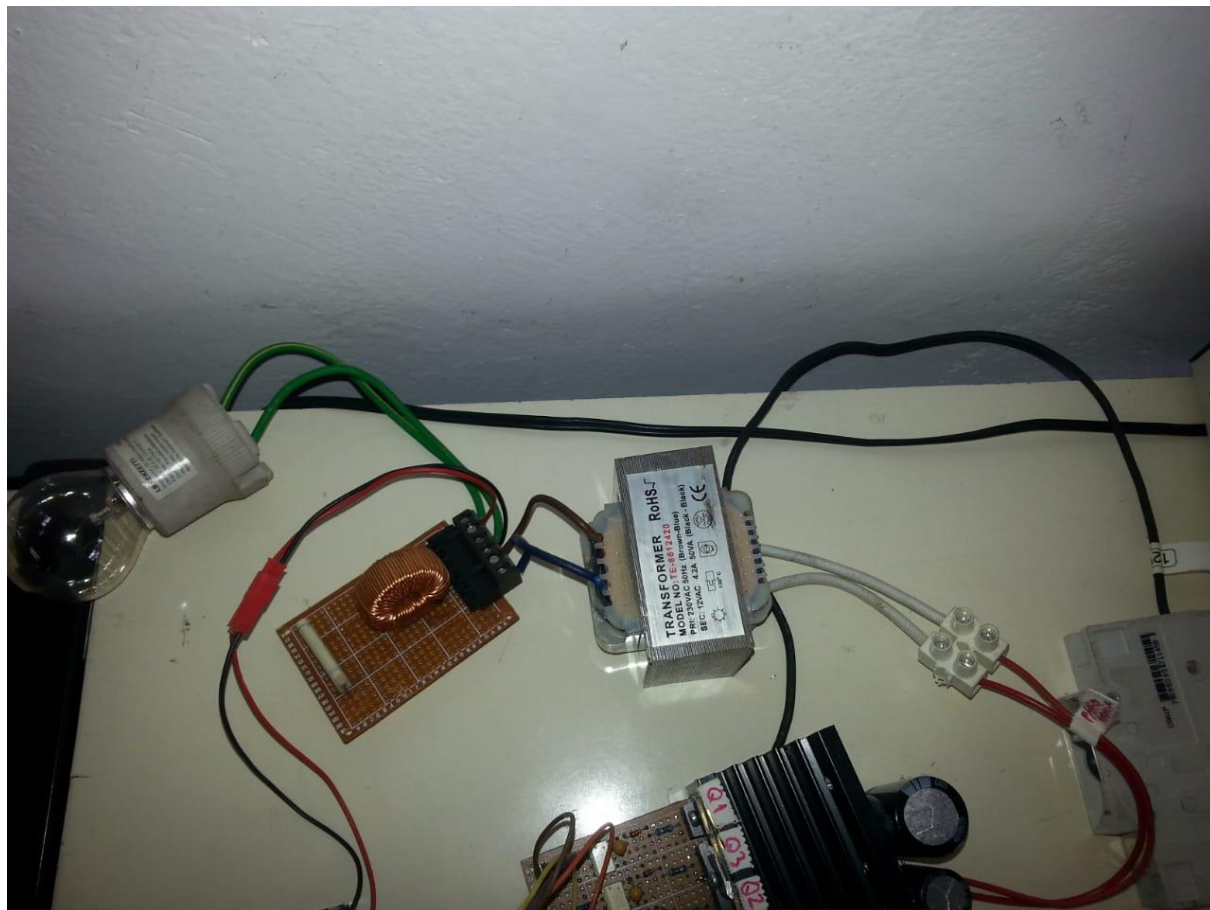


Figura 3.24 - Etapa de elevação da tensão e entrega da saída alternada

CAPÍTULO 4

4 RESULTADOS E DISCUSSÕES

Neste capítulo é apresentado todos os resultados obtidos nas simulações via software e na montagem previamente apresentada no item 4.8 do capítulo anterior

4.1 SIMULAÇÕES

Antes do início da montagem de qualquer dispositivo eletrônico, é importante que se faça uma simulação do estudo para que não se perca tempo com uma montagem que não apresente os resultados esperados. Para isso, realizou-se uma montagem virtual apenas do circuito de potência (figura 23) com o intuito de simular parte do código que garante a modulação tipo sPWM (Anexo I).

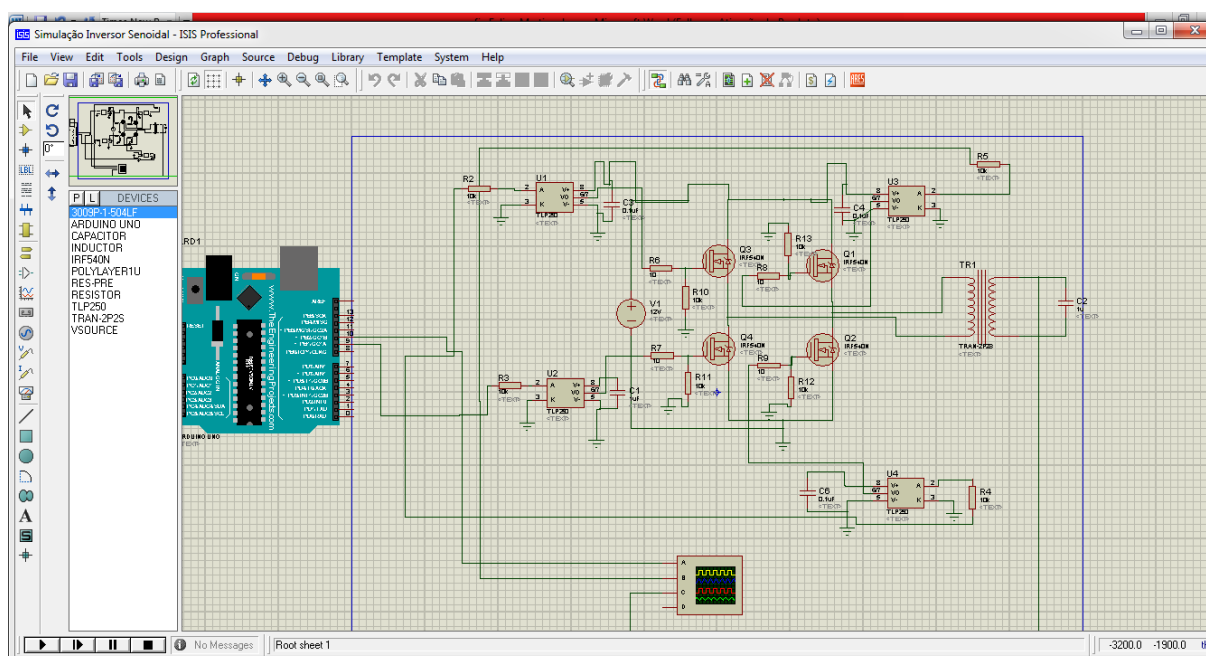


Figura 4.1 - Montagem para simulação da modulação sPWM

A seguir são apresentados os resultados obtidos através do osciloscópio digital disponível dentro do software, observe que, conforme estudado, os sinais se alternam entre si,

formando os ciclos positivos e negativos, e possuem curta duração nas extremidades ($CC = 0\%$) e longa duração no centro dos semiciclos ($CC = 100\%$)

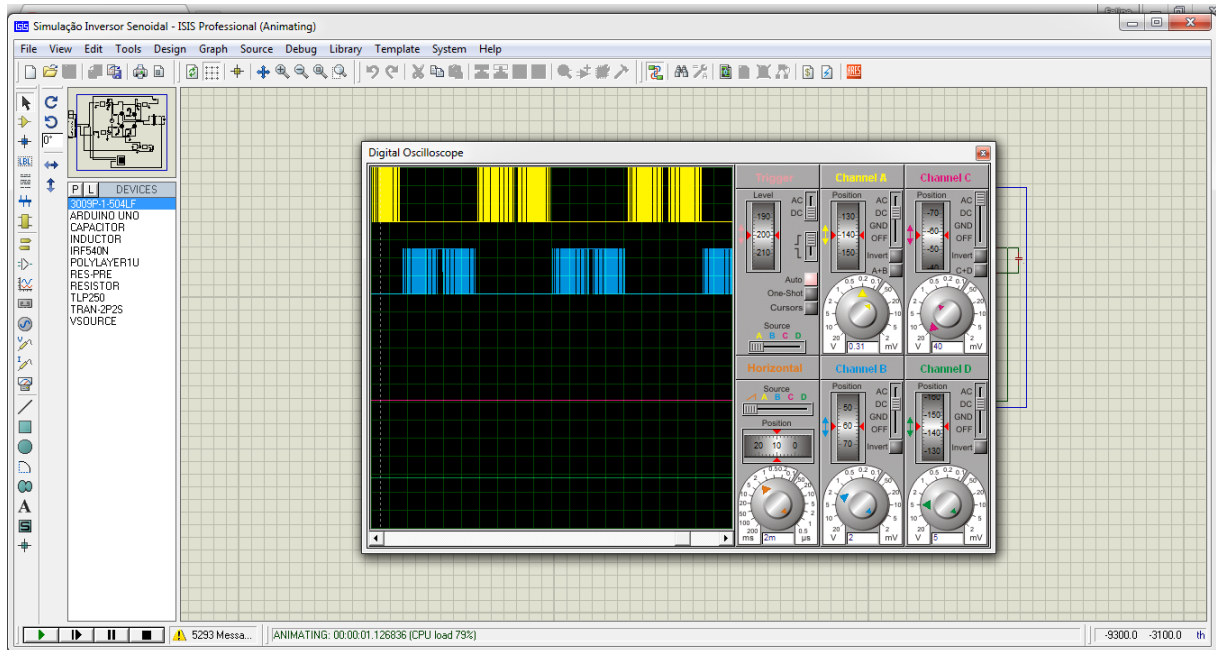


Figura 4.2 - Simulação modulação sPWM

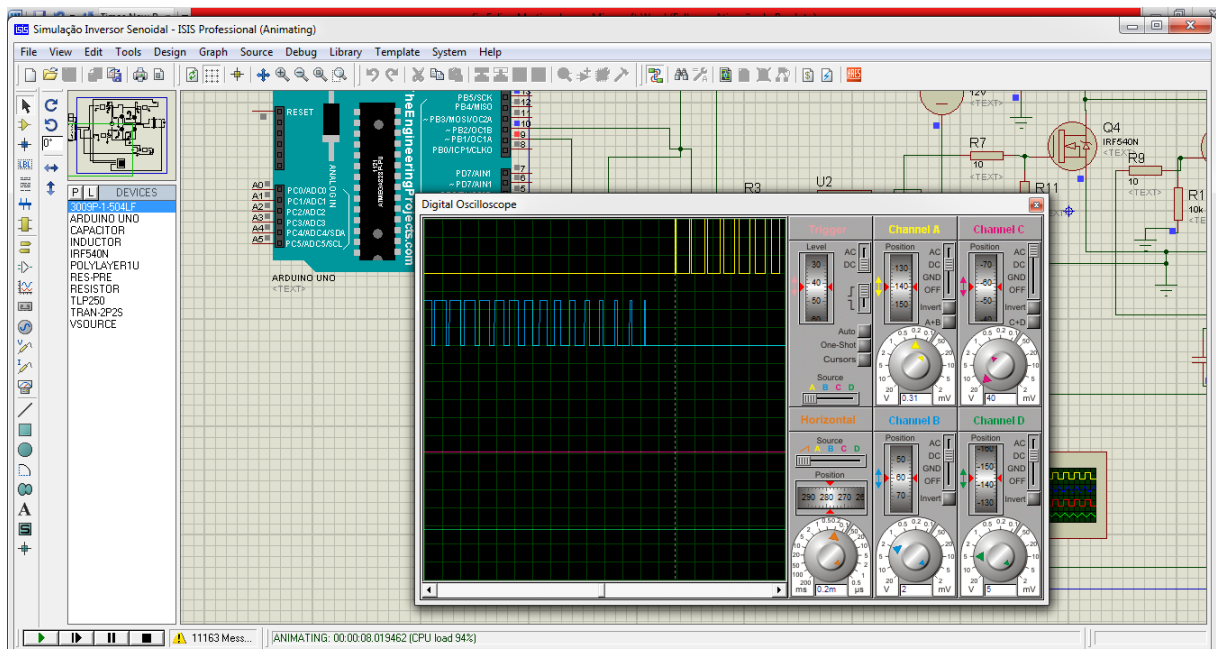


Figura 4.3 - Evidenciação da diferença do tamanho dos pulsos

Feita a simulação para a modulação, tem-se agora a necessidade de simular o inversor por completo, unindo todos os componentes e circuitos apresentados ao longo do capítulo 4, e observar se a montagem fornece uma saída desejada. O código que carrega o microcontrolador foi trocado, pelo que se encontra no Anexo II, e todas as partes foram unidas em um só projeto no *Proteus Design Suite®*, conforme a imagem abaixo:

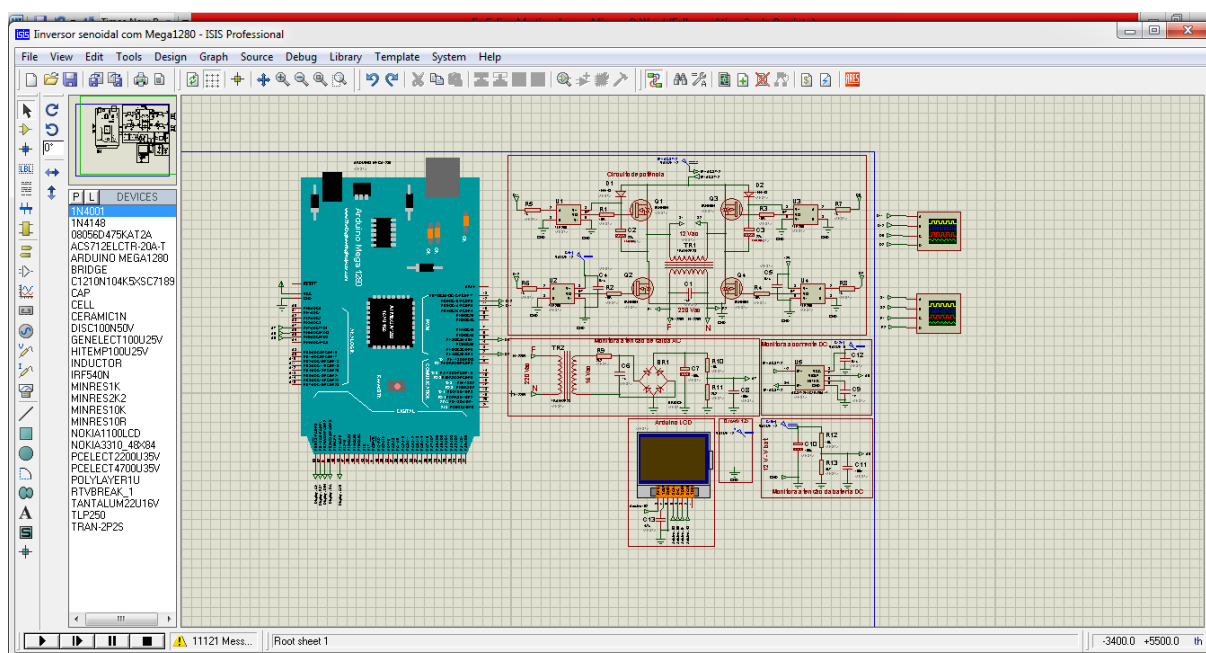


Figura 4.4 - Montagem completa do inversor

Instalando as bibliotecas do LCD e colocando o protótipo para funcionar, temos na saída dos osciloscópios digitais as seguintes formas de onda:

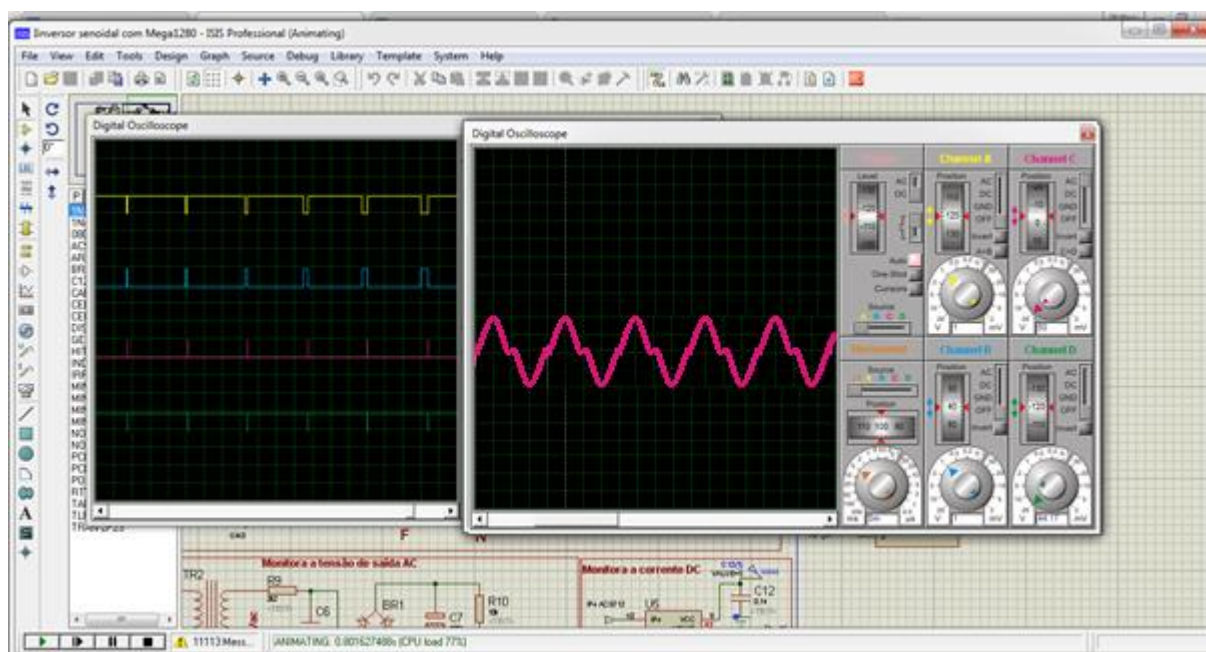


Figura 4.5 - Formas de onda de entrada e saída do inversor

No osciloscópio do lado esquerdo, nas linhas amarela e azul temos os pulsos dados pelo pinos do arduíno para ativação dos braço esquerdo do inversor.

No osciloscópio do lado direito, temos a forma de onda da saída do TRAFO, note que se percebe uma onda senoidal, não perfeita devido à estratégia utilizada para chaveamento dos MOSFETs e a simulação não levar em consideração as limitações físicas do sistema, mas consegue-se ter uma ideia que a tensão da bateria continua, foi transformada em tensão alternada, alcançando de acordo com a leitura uma onda de saída que possui amplitude bem definida comprovando que o chaveamento dos MOSFET está ocorrendo de forma estratégica modulada , o que faz alcançar os objetivos propostos deste trabalho existem outras fórmulas e técnicas para obtenção de uma tensão senoidal mais pura com até menos de um por cento de ruídos ou sinais de distorção harmônicos que são responsáveis por reduzir a qualidade de operação do sistema como um todo, o que leva a uma perda de qualidade do sinal de saída. Porém estes não são objetos de estudo deste trabalho.

4.2 RESULTADOS DO PROTÓTIPO

4.2.1 Aplicação da Modulação sPWM aos MOSFETs

Observando um dispositivo que já havia sido montado e tendo acesso a suas respostas, com a ajuda de um osciloscópio real, realizou-se o teste da modulação sPWM através dos MOSFETs, observa-se, assim como registrado na figura 4.6, pulsos de pequena largura no início e final do ciclo e de grande largura no meio do ciclo, possibilitando assim uma variação de transferência de potência para a o transformador, podemos perceber também que nos períodos próximos a amplitude máxima de uma senoide, a condução através pinos torna-se constante, muito bem definido em seu estado de transmissão, eliminando harmônicas e ruídos que possam vir a prejudicar o sinal.

Combinaram-se os pinos Q1 E Q4 dos MOSFETs a fim de se obter um sinal alternado, que correspondem aos semiciclos positivo e negativo, formando um período completo.

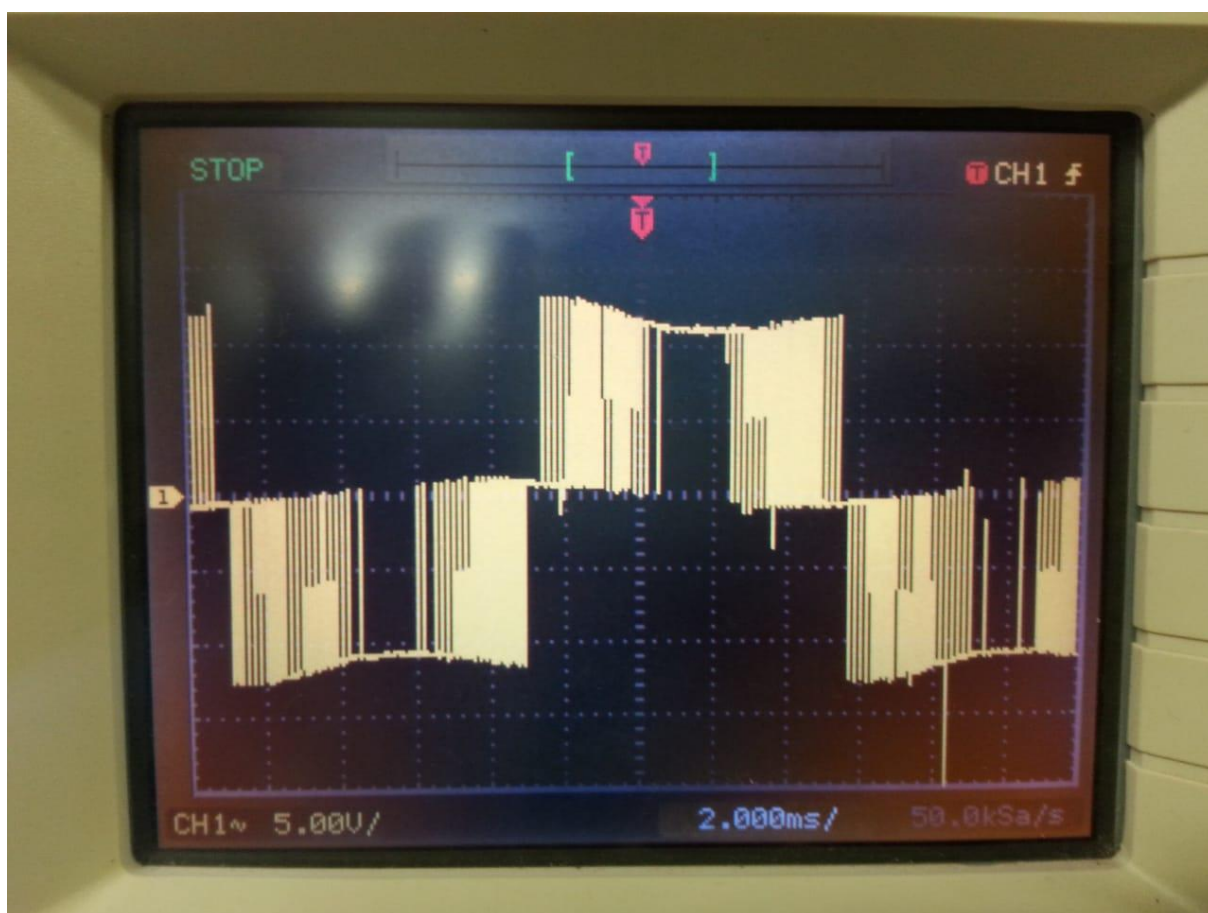


Figura 4.6 – Sinal de chaveamento alternado osciloscópio

4.2.2 O Sinal de Saída Alternado

Visto o correto funcionamento do chaveamento dos pinos dos MOSFET, faz-se necessário observar como esta a forma de onda da saída, se ela de fato esta se comportando de forma senoidal. encontrou-se uma onda perto da esperada conforme apresentada pelo osciloscópio, a figura 4.7 apresenta a resposta do inversor a um ensaio a vaz io:

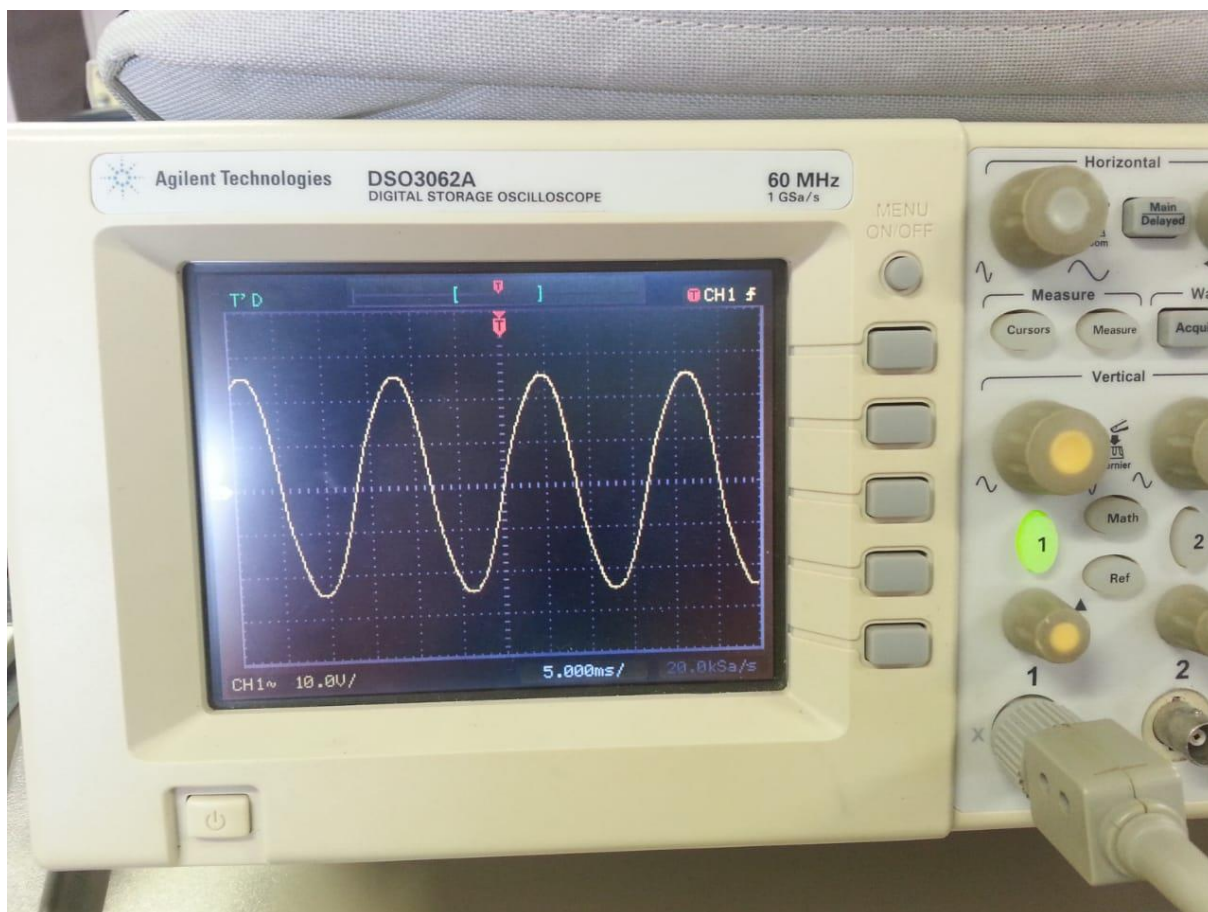


Figura 4.7 Sinal de saída do inversor sem carga

Ao se conectar uma carga no inversor, devido a ausência de circuito para resposta a carga, a onda sofre uma pequena defasagem, que pode ser facilmente corrigida em trabalhos futuros. A seguir a resposta do inversor a um ensaio com carga eletônica:

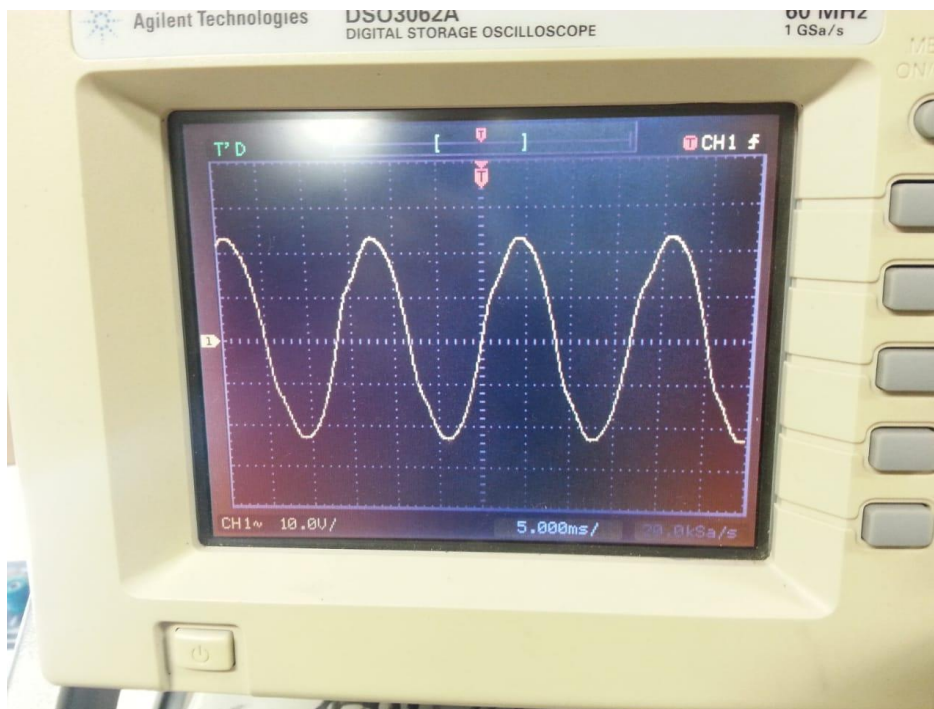


Figura 4.8 Sinal de saída do inversor com carga

4.3 TESTES DO INVERSOR

Os testes foram realizados no Laboratório de Eletrônica Analógica e Digital da Universidade Federal de Ouro Preto.

Para se realizar os testes no inversor, primeiramente o mesmo foi ligado sem nenhuma carga, e depois foi ligado com dois tipos diferentes de carga, uma lâmpada LED 8W bivolt e uma lâmpada incandescente de 100 A / 220V.

As leituras da corrente alternada foram apresentadas em RMS (Root Mean Square).

4.3.1 Testes Realizados no Inversor

A tabela abaixo mostra os testes obtidos pelo inversor aplicando-se os ensaios à vazio (sem carga) e com carga:

Tabela 4.1 - Ensaios realizados no protótipo

	INVERSOR SEM CARGA (0 W)	INVERSOR COM CARGA (8 W)
TENSÃO DA BATERIA [V]	11.6	11.9
TENSÃO DE SAÍDA PONTE H [V]	8.91	8.19
	8.54 Vrms	8.24 Vrms
TENSÃO DE SAÍDA NO TRAFO [Vac]	165.5	150
	238.32 Vrms	216.72 Vrms
CORRENTE DE ENTRADA NA PONTE H [A]	0.29	0.99
CORRENTE AC SAÍDA TRAFO [A]	0	0.06

Uma maneira mais recomendável de se fazer a leitura da corrente alternada é utilizando um algoritmo *RMS*.

4.4 CIRCUITO BOOTSTRAP

A falta deste circuito impede que a tensão V_{gs} seja alcançada, fazendo com que o MOSFET trabalhe na sua região linear, podendo então ser classificado como um amplificador, causando um aquecimento com a mínima corrente aplicada ao mesmo, o que pode ocasionar a ruptura da sua junção. No começo dos testes não atentamos para o uso do *bootstrap* e 4 MOSFETs queimaram-se. Felizmente, alguns drivers para MOSFET já apresentam o circuito de *bootstrap* embutidos no encapsulamento do chip, eliminando a preocupação com esse tipo de cuidado que preciso foi ser utilizado neste projeto.

A utilização deste circuito, ocasionou um erro na amplitude da tensão de saída do inversor, para corrigir este problema, deve-se usar outro tipo de *driver* para o chaveamento da ponte H.

4.5 O TRANSFORMADOR

O Transformador elevador de tensão (no projeto chamado de TR1) utilizado apresentou uma corrente máxima de operação muito aquém no lado secundário (0,229A). Desta forma, recomenda-se a utilização de outro TRAFO para realização da montagem.

4.6 CI ACS712

Por se tratar de um sensor de leitura para correntes para diversos níveis de amperagem, o sensor apresentou alto ruído na saída, pois atuando a vazio, o inversor apresentou um consumo de apenas 390mA. O mais recomendável, é que se utilize um resistor do tipo *shunt* para realizar a leitura da corrente CC que foi drenada da bateria.

CAPÍTULO 5

5 CONCLUSÕES

Ao chegar ao fim deste projeto, observa-se que a implementação de um inversor de tensão controlado por um Arduino, com relação da teoria para a prática apresenta grandes diferenças, pois as implementações das teorias estudadas em eletrônica de potência por alguns momentos não se dão da forma esperada no mundo real, cabendo fazer adaptações para que os resultados esperados possam ser encontrados. O *Proteus Design Suite*® se mostrou uma excelente ferramenta para montagem dos circuitos de forma digital além de possibilitar simulações do dispositivo para inferência de resultados. As simulações foram feitas na versão *7 professional service pack 2*, por melhor se adaptar a integração da biblioteca de um microcontrolador. Para lidar com dispositivos reais, deve-se sempre levar em conta que os mesmos possuem limitações físicas e o sobreaquecimento impede, na maioria das vezes que os dispositivos cumpram seu papel em um circuito eletrônico, ao se fazer uma montagem de dispositivos eletrônicos deve – se também ter extrema atenção ao manuseio pois além de frágeis, todos os componentes precisam estar perfeitamente alinhados e bem posicionados para uma perfeita transmissão e fácil interpretação respectivamente.

Pode-se concluir também que o dispositivo construído é de certa forma eficiente em termos de consumo de energia e baixo investimento (valores próximos a R\$250,00). Além de ser economicamente viável, sua característica construtiva foi capaz de atender projetos de autossuficiência de energia.

Para trabalhos futuros fica-se a possibilidade de utilizar um TRAFO de alta frequência, o que tornaria o protótipo menor e mais interessante, pois a partir desse ponto se consegue dois pontos de controle em alta frequência (Chaveamento da entrada do transformador e da ponte H), além de implementação de técnicas de modulação com a injeção de terceiro harmônico e estudo aprofundado sobre a distorção harmônica total e seus efeitos sobre a corrente e tensão da saída do conversor.

CAPÍTULO 6

6 REFERÊNCIAS BIBLIOGRÁFICAS

AVELAR, E. F ARDUINO LAB 17 – INVERSOR DE TENSÃO SENOIDAL MICROCONTROLADO – VERSÃO 1. Belo Horizonte, Fev.2018

SILVA, F. L. et al. CONVERSOR DE FREQUÊNCIA CC-CA. E-xacta. Belo Horizonte - Minas Gerais, Mar.2013.

BARBI, I. et al. PROJETO DE INVERSORES. Florianópolis - Santa Catarina, Mai.2007.

SEGUNDO, A. K. R.; RODRIGUES, C. L. C. ELETÔNICA DE POTÊNCIA E ACIONAMENTOS ELÉTRICOS. Ouro Preto - Minas Gerais, 2015.

MEZARоба, M; MODULAÇÃO PWM. Florianópolis – Santa Catarina, Jun. 2008

HOLMES, D. G.; LIPO T. A. PULSE WIDTH MODULATION FOR POWER CONVERTERS. Estados Unidos, 2003.

Eletrônica Digital 28 - SINAL PWM - MODULAÇÃO POR LARGURA DE PULSO. 2015. Disponível em: < <https://www.youtube.com/watch?v=z4Rm6BrruxU>>. Acesso em Abr.2018.

Eletrônica de Potência, o que é?. Disponível em: < <https://www.mundodaeletrica.com.br/eletronica-de-potencia-o-que-e/>>. Acesso em Abr.2018,

DATASHEET, Sensor de corrente ACS712. Allegro® MicroSystems, Inc. Disponível em: <<https://www.sparkfun.com/datasheets/BreakoutBoards/0712.pdf>>. Acesso em Abr.2018.

DATASHEET, GaAlAs Ired & Photo-IC TLP250. TOSHIBA®. Disponível em: <<https://toshiba.semicon-storage.com/info/docget.jsp?did=16821>>. Acesso em Abr.2018.

DATASHEET, MOSFET de potência IRF540N HEXFET®. Disponível em:<<https://www.infineon.com/dgdl/irf540n.pdf?fileId=5546d462533600a4015355e396cb199f>>. Acesso em Abr.2018.

ANEXO I – CÓDIGO FONTE UTILIZADO PELO MICROCONTROLADOR PARA OPERAR AS SAÍDAS DO MICONROLADOR COM O INTUITO DE SIMULAR E REALIZAR A MODULAÇÃO SPWM.

```
#include <avr/interrupt.h>

#define SinDivisions (100)// Sub divisions of sisusoidal wave.

static int microMHz = 16; // Micro clock frequency

static int freq = 60;    // Sinusoidal frequency

static long int period; // Period of PWM in clock cycles.

static unsigned int lookUp[SinDivisions];

static char theTCCR1A = 0b10000010; //variable for TCCR1A

void setup(){

    double temp; //Double variable for <math.h> functions.

    period = microMHz*1e6/freq/SinDivisions;// Period of PWM in clock cycles

    for(int i = 0; i < SinDivisions/2; i++){ // Generating the look up table.

        temp = sin(i*2*M_PI/SinDivisions)*period;

        lookUp[i] = (int)(temp+0.5);    // Round to integer.

    }

    // R

    egister initilisation, see datasheet for more detail.

    TCCR1A = theTCCR1A; // 0b10000010;

    /*10 clear on match, set at BOTTOM for compA.

    00 compB disconnected initially, toggled later to clear on match, set at BOTTOM.

    00

    10 WGM1 1:0 for waveform 15.

    */

    TCCR1B = 0b00011001;

    /*000
```

```

    11 WGM1 3:2 for waveform 15.

    001 no prescale on the counter.

    */

TIMSK1 = 0b00000001;

    /*00000000

    1 TOV1 Flag interrupt enable.

    */

ICR1 = period; /* Period for 16MHz crystal, for a switching frequency of 100KHz for 200
subdivisions per 50Hz sin wave cycle. */

sei();          // Enable global interrupts.

// Set outputs pins.

DDRB = 0b00000110; // Set PB1 and PB2 as outputs.

pinMode(13, OUTPUT);
}

void loop(){/*Do nothing . . . forever*/}

ISR(TIMER1_OVF_vect){

    static int num;

    static int delay1;

    static char trig;

    int sensorValue = analogRead(A0);

    if(delay1 == 1){/*delay by one period because the high time loaded into OCR1A:B values
are buffered but can be disconnected immediately by TCCR1A. */

        theTCCR1A ^= 0b10100000; // Toggle connect and disconnect of compare output A and B.

        TCCR1A = theTCCR1A;

        delay1 = 0;          // Reset delay1

    } else if(num >= SinDivisions/2){

        num = 0;             // Reset num

```

```
    delay1++;  
    trig ^=0b000000001;  
    digitalWrite(13,trig);  
}  
// change duty-cycle every period.  
OCR1A = OCR1B = lookUp[num]+sensorValue*2;  
num++;  
}
```

ANEXO II - CÓDIGO FONTE UTILIZADO PELO MICROCONTROLADOR PARA OPERAR SUAS ENTRADAS E SAÍDAS COM O INTUITO DE REALIZAR LEITURAS E OPERARAÇÃO DO INVERSOR.

```

/*****
****

```

Inversor de tensão CC / AC controlado pelo Arduino

```

****
*

```

-- IDE do Arduino Versão 1.8.3

-- Autor do código base do display: Limor Fried/Ladyada, Adafruit Industries.

<https://github.com/adafruit/Adafruit-GFX-Library>

<https://github.com/adafruit/Adafruit-PCD8544-Nokia-5110-LCD-library>

-- Autor código inicial sPWM: Kurt Hutten

<https://github.com/Terbytes/Arduino-Atmel-sPWM>

-- Autor versão com o Mega e display, modulando os 4 MOSFETs: Eduardo Avelar

-- Blog: easytromlabs.com

-- email: contato@easytromlabs.com

-- Fevereiro, 2018

```

****
*

```

```

****
**

```

Conexões entre Hardwares:

Graphic LCD Pin ----- Arduino Pin

1-RST ----- 52

2-CE ----- 53

3-CC ----- 50

4-DIN(MOSI) ----- 51

5-CLK ----- 48

6-VCC ----- 3,3V
 7-LED ----- GND
 8-GND ----- GND

 */

#include <avr/io.h>

#include <avr/interrupt.h>

#include <stdint.h>

#include <Adafruit_GFX.h>

#include <Adafruit_PCD8544.h>

//CLK, DIN, CC, CE, RST//

Adafruit_PCD8544 display = Adafruit_PCD8544(48, 51, 50, 53, 52);

// Defines

#define SinDivisions (100) // Sub divisoes da onda senoidal

#define ACVOLTPIN A7 // Pino de entrada da amostra do sinal AC de saída

#define CCVOLTPIN A6 // Pino de entrada da amostra do sinal CC de entrada (bateria 12V)

#define CCCURRPIN A5 // Pino de entrada da amostra do sinal CC de corrente de entrada

#define N 30

// Declaração de variáveis

int valsTemp1[N];

int valsTemp2[N];

int valsTemp3[N];

long sumTemp1 = 0.0;

long sumTemp2 = 0.0;

long sumTemp3 = 0.0;

int mVperAmp = 100;

```

int Vcc = 5000;

int ACSoffset = 2500;

static int microMHz = 16;           // Frequencia do microcontrolador

static int freq = 60;               // Frequencia da senoide

static long int period;             // Período 16,6 ms

static unsigned int lookUp[SinDivisions];

int state = 0;

float voltageAC;

float voltageCC;

float currentCC;

void setup() {

    Serial.begin(9600);

    display.begin();                // Inicializa o controlador do display

    display.setRotation(2);

    display.setContrast(60);        // Configura contraste para o valor de 55

    display.clearDisplay();         // Apaga a tela do display

    display.display();              // Manda atualizar o display

    analogReference(EXTERNAL);

    double temp;

    period = microMHz * 1e6 / freq / SinDivisions;    // (2666.67) Period of PWM in clock
cycles

    for (int i = 0; i < SinDivisions / 2; i++) {      // Gera a tabela ou vetor

        temp = sin(i * 2 * M_PI / SinDivisions) * period;

        lookUp[i] = (int)(temp + 0.5);                // Arredonda para um número inteiro

    }

    // Inicialização do registrador.

    TCCR1A = 0b10110010; //variable for TCCR1A

    TCCR3A = 0b10110010; //variable for TCCR3A

```

```

/*10 Comp A, apaga no valor de comparação, (Modo não invertido
  11 Comp B, ativa no valor de comparação, (Modo invertido)
  00
  10 WGM1 1:0 for waveform 14.
  */
  TCCR1B = 0b00011001;
  TCCR3B = 0b00011001;
  /*000
    11 WGM1 3:2 for waveform 14.
    001 no prescale on the counter.
    */
    TIMSK1 = 0b00000001;
    TIMSK3 = 0b00000001;
    /*0000000
      1 TOV1 Flag interrupt enable.
      */
      ICR1 = period; // Período para 16MHz de cristal, 100 subdivisões e 60Hz de frequência da
onda de saída

      ICR3 = period; // Período para 16MHz de cristal, 100 subdivisões e 60Hz de frequência da
onda de saída

      sei(); // Enable global interrupts.

      // Set outputs pins.

      DDRB |= 0b11100000; // Arduino MEGA - Set PB5 (D11-PWM), PB6 (D12-PWM), PB7
(D13-LED) como saídas

      DDRE |= 0b00011000; // Arduino MEGA - Set PE3 (D5-PWM), PE4 (D2-PWM) como saídas
    }

void loop() {

  voltageAC = (movingAverageacVolt(ACVOLT_PIN) / 1024.0) * 5;

```

```

voltageAC = voltageAC * (264.08 / 5);
voltageCC = (movingAverageCCVolt(CCVOLTPIN) / 1024.0) * 5;
voltageCC = voltageCC * (15.5 / 5);
currentCC = ((movingAverageCCCurr(CCCURRPIN) / 1024.0) * 5) * 1000;
currentCC = ((currentCC - ACSoffset) / mVperAmp);

display.setTextSize(1);          // Configura o tamanho do texto
display.clearDisplay();          // Limpa a tela do display
display.setTextColor(BLACK, WHITE); // Inverte as cores das letras
display.setCursor(0, 0);         // Posiciona o cursor
display.print("Inversor CC/AC"); // Envia texto ao display
display.drawFastHLine(0, 9, 84, BLACK); // Desenha uma linha horizontal
display.setCursor(0, 10);
display.setTextColor(WHITE, BLACK);
display.print("  AC OUTPUT  ");
display.setTextColor(BLACK, WHITE);
display.setCursor(15, 20);
display.print(voltageAC, 1);
display.print(" Vac ");
display.drawFastHLine(0, 29, 84, BLACK);
display.setCursor(0, 30);
display.setTextColor(WHITE, BLACK);
display.print("  CC INPUT  ");
display.setCursor(2, 40);
display.setTextColor(BLACK, WHITE);
display.print(voltageCC, 1);
display.setCursor(29, 40);
display.print("V");

```



```

display.setCursor(45, 40);

display.print(currentCC, 2);

display.print(" A");

display.display();

delay(100);

}

//*****
*//

// Função que calcula a media móveol da leitura feita no canal A7 - AC VOLT //

//*****
*//

long movingAvarageacVolt(int port) {

    long sumTemp = 0.0;

    for (int i = N - 1; i > 0; i--) {

        valsTemp1[i] = valsTemp1[i - 1];

    }

    valsTemp1[0] = analogRead(port);

    delay(5);

    for (int i = 0; i < N; i++) {

        sumTemp = sumTemp + valsTemp1[i];

    }

    return sumTemp / N;

}

//*****
*//

// Função que calcula a media móveol da leitura feita no canal A6 - CC VOLT //

//*****
*//

long movingAvarageCCVolt(int port) {

```

```

long sumTemp = 0.0;
for (int i = N - 1; i > 0; i--) {
    valsTemp2[i] = valsTemp2[i - 1];
}
valsTemp2[0] = analogRead(port);
delay(5);
for (int i = 0; i < N; i++) {
    sumTemp = sumTemp + valsTemp2[i];
}
return sumTemp / N;
}

//*****
*//

// Função que calcula a media móveol da leitura feita no canal A5 - CC CURRE //

//*****
*//

long movingAvaregeCCCurr(int port) {
    long sumTemp = 0.0;
    for (int i = N - 1; i > 0; i--) {
        valsTemp3[i] = valsTemp3[i - 1];
    }
    valsTemp3[0] = analogRead(port);
    delay(5);
    for (int i = 0; i < N; i++) {
        sumTemp = sumTemp + valsTemp3[i];
    }
    return sumTemp / N;
}

```

```

////////////////////////////////////
ISR(TIMER1_OVF_vect) {

    static int num;

    if (num >= SinDivisions / 2) {

        num = 0;          // Reset num

        state ^= 1;

        PORTB ^= 0b10000000; // Arduino MEGA - Faz um toggle no pino PB0 - D8 (LED)

                               //sem afetar os outros pinos utilizando uma XOR

    }

    if (state == 0) {

        OCR1A = 0; // D11 - Q1 -> OFF

        OCR1B = 0; // D12 - Q2 -> ON (Complemento do D11)

        OCR3A = lookUp[num]; // D5 - Q3 -> PWM

        OCR3B = lookUp[num]; // D2 - Q4 -> PWM Complemento

        num++;

    }

    if (state == 1) {

        OCR1A = lookUp[num]; // D11 - Q1 -> PWM

        OCR1B = lookUp[num]; // D12 - Q2 -> PWM Complemento

        OCR3A = 0; // D5 - Q3 -> OFF

        OCR3B = 0; // D2 -> ON

        num++;

    }

}

ISR(TIMER3_OVF_vect) {

}

//-- Apenas como entendimento da matemática nos pinos ----//

```

```
//PORTB ^= 0b00000001; // Faz um toggle no pino PB0 - (D8) sem afetar os outros bits  
utilizando uma XOR
```

```
//PORTB |= 0b00000001; // Escreve no pino PB0 - (D8) sem afetar os outros pinos
```

```
//PORTB &= 0b11111110; // Apaga o bit PB0 - (D8) sem afetar os outros pinos
```