# Three-Level PWM DC/AC Inverter Using a Microcontroller

Oliver Rich

William Chapman

MQP Terms A-B-C 2011-2012

Advisor: Professor Stephen J. Bitar

Sponsor: NECAMSID

# Abstract

This project proposes a unique DC to AC inverter design to convert high voltage DC into pure sine wave 120VAC, 60Hz power. A microcontroller design was chosen to implement a 3-level pulse-width modulation technique for greater efficiency. Standard high voltage components were chosen for MOSFET drivers and H-bridge capable of handling a maximum of 1000 Watts. Initial tests show that the technique is viable.

# Table of Contents

# Table of Figures

# Introduction

The focus of this report is on the design and prototype testing of a DC to AC inverter which efficiently transforms a DC voltage source to a high voltage AC source similar to the power delivered through an electrical outlet (120$V_{rms}$, 60Hz) with a power rating of approximately 1000W. Electronic devices run on AC power, however, batteries and some forms of power generation produce a DC voltage so it is necessary to convert the voltage into a source that devices can use.

A low voltage DC source is inverted into a high voltage AC source in a two-step process. First the DC voltage is stepped up using a boost converter to a much higher voltage. This high voltage DC source is then transformed into an AC signal using pulse width modulation. Another method involves first transforming the DC source to AC at low voltage levels and then stepping up the AC signal using a transformer. A transformer however is less efficient and adds to the overall size and cost of a system.

This project builds upon the work of a previous group that was given the similar task of designing a DC to AC inverter. The previous group took a solely analog approach in the implementation of their system. While there are some advantages to this, it limits the flexibility of the system in that it can only be used for a specific purpose and if a design change is needed, the process is difficult and potentially labor intensive. In this report, we detail how the inverter's controls were implemented with a digital approach using a microprocessor for the control system and how effective and efficient a 3-level PWM inverter can be.

# Problem Statement

Unfortunately, many places in Africa lack a reliable power grid. This is a large problem for many reasons, especially in the medical field where reliable power is essential for doctors who need to be able to see and monitor their patients during operations. A company called Waste2Watts is attempting to alleviate this problem by providing a low cost device that serves as a backup power supply when the grid fails. While there are many systems already on the market that do this, Waste2Watts wants to provide a device that can be made cheaply with readily available parts from disposed computers components and car batteries.

An important piece of backup power supply is the DC to AC inverter which converts the DC voltage from a battery to an AC voltage that is necessary to operate electronic components, in this case medical equipment. Due to the delicate nature of this equipment, an inverter which is capable of producing a pure sine wave is necessary to avoid noise and wear on delicate and expensive gear. Many of these devices are very expensive so it is the goal of this project to design a DC/AC inverter capable of producing a pure sine wave for use with medical equipment.

# Background

Before going into the details of our implementation, it is important to first review some of the basics principles and components we will be using.

## Direct Versus Alternating Current

In the world today, there are currently two forms of electrical transmission, direct current (DC) and alternating current (AC) systems, each with their own advantages and disadvantages. DC power is simply the application of a constant voltage across a load resulting in a constant current. A battery is the most common power source for DC along with several forms of power generation. This is widely used in digital circuitry as it provides constant high and low values which represent the basic 1 and 0 bits used by computers.

Thomas Edison, inventor of the light bulb, was the first to transmit electricity commercially using DC power lines. It was not capable of transmission over long distances however, as the technology did not exist to step-up the voltage along the transmission path over which the power would dissipate. The equation below demonstrates how high voltage was necessary to decrease power loss.

$$V = IR$$

$$P = VI = I^2R$$

When the voltage is increased, the current decreases and concurrently the power loss decrease exponentially. Therefore, high voltage transmission decreases power loss. AC power was found to be much more efficient at transmitting power as it alternates between two voltages at a specific frequency, making it easier to either step up or down using a transformer. Today, electrical transmission is based mostly off of AC power, supplying American homes and businesses with 120V AC power at 60Hz.

While DC power is used in many digital applications, AC power also used in many other applications such as in power tools, televisions, radios, medical devices, and lighting. Therefore, it is necessary to have an efficient means of transforming DC to AC and vice versa. Without this ability, people would be restricted to using devices that only worked on the power that was supplied to them.

## Inverters

An inverter is defined as a device that converts direct current (DC) into alternating current (AC). Inverters can come in many different varieties, differing in price, power, efficiency and purpose. The most common purpose of a DC/AC power inverter is to take a battery or similar power storage, such as a 12V car battery, and convert it into a 120V AC power source operating at 60Hz, allowing it to be used as an ordinary household electrical outlet. Inverters have become more and more common over the past several years as support for self-sufficient solar power has increased. Because solar power is comes as a DC source, it requires an inverter before it can be used as general power.

There are two common methods for inverters on the market today, each of which has its own flaws.

## Modified Sine Wave

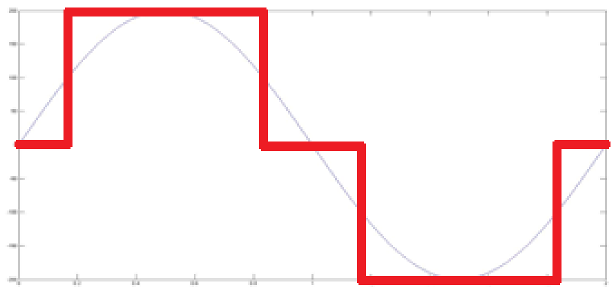


Figure 1: Modified Sine Wave

A modified sine wave is similar to a square wave, but sits at zero for a set time before powering high or low. For many simple devices, this is the easiest and cheapest solution, but it comes with a hidden cost: harmonics. Harmonics are integer multiples of the fundamental power frequency (in this case, 60 Hz) that appear when the sine wave is not completely pure. Many sensitive pieces of equipment, such as computers or the aforementioned medical equipment cannot run off of a modified sine wave, as the most common side effect of harmonics is increased current flow. When dealing with delicate circuits, this leads to burn out components and overall failure. Even if the device is not overly sensitive, many other devices such as motors and fluorescent lights output far more wear on themselves as they are not meant to deal with the increased current, resulting in significantly reduced lifespan.

## Pulse Width Modulation

The other common method of generating AC power in electronic power converters is pulse width modulation (PWM). PWM is used extensively as a means of powering AC devices with a DC power source. A DC voltage source can be made to look like an AC signal across a load by altering the duty cycle of the PWM signal. The pattern at which the duty cycle of the

PWM signal varies can be generated through simple analog components, a digital microcontroller, or specific PWM integrated circuits.

In analog circuitry, a PWM signal is generated by feeding a reference and a carrier signal through a comparator which creates the output signal based on the difference between the two inputs. The reference is a sinusoidal wave at the frequency of the desired output signal. The carrier wave is a triangle or 'sawtooth' wave which operates at a frequency significantly greater than the reference wave. When the carrier signal exceeds the reference the output is at one state, and when the reference exceeds the carrier the output is at the opposite state. The process is shown below in Figure A, with the carrier in blue, the reference in red, and the output in green.

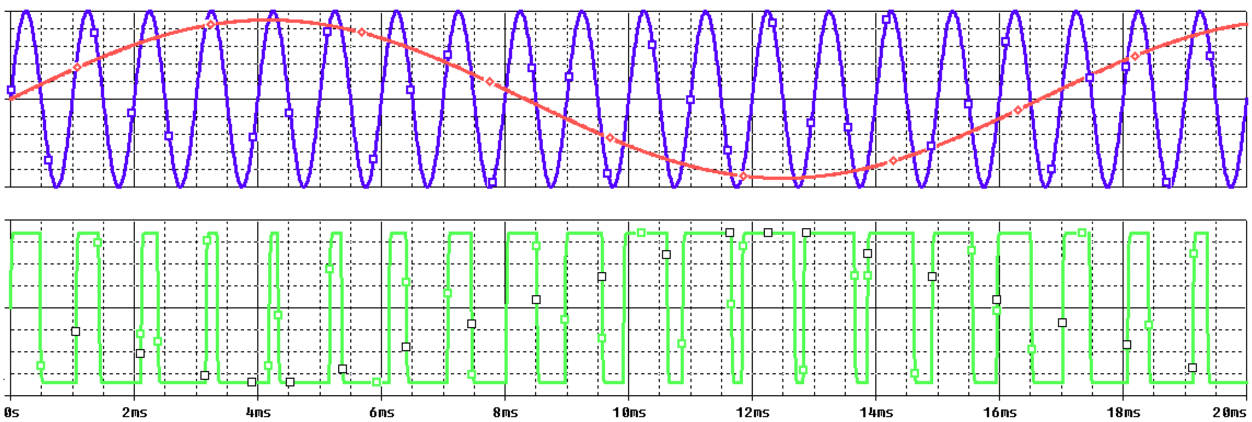



**Figure 2: 2 Level PWM Signal (analog control)**

The signal shown above is a simple 2 level PWM signal. In order for the signal to better resemble a sine wave, it is necessary to add in another level. This is usually accomplished using an H-bridge circuit which is discussed in the next section.

## H-Bridge

An H-bridge is a circuit which enables a voltage to be put across a load in either direction. It consists of four switches, typically MOSFETs, and load configured in the shape of an ‘H’. By controlling which switches are closed at any given moment, the voltage across the load can be either positive, negative, or zero. The table below outlines the positions. Note that other possible switch positions are omitted because they would cause a short between the power supply and ground, potentially damaging the devices or draining the power supply.

| High Side Left | High Side Right | Low Side Left | Low Side Right | Load Voltage |
|---|---|---|---|---|
| On | Off | Off | On | Positive |
| Off | On | On | Off | Negative |
| On | On | Off | Off | Zero |
| Off | Off | On | On | Zero |

In choosing which type of MOSFET switches to use, there are two options: P-channel devices and N-channel devices. The use of P-channel on the high sides and N-channel on the low sides is an easier route to go with as the high side switches will not require a driver. However, P-channel devices have a higher ‘on’ resistance so there is greater power loss. It is possible to use all N-channel devices for both the high and low sides of the device; however, the high side N-channel device will require a driver with a bootstrap capacitor to generate the higher voltage above the switching voltage of 170V to turn on the device. The MOSFET driver is discussed in the next section.

## MOSFET Driver

As stated in the previous section, it is beneficial to use N-channel MOSFETs as the high side switches as well as the low side switches because they have a lower ‘ON’ resistance and therefore less power loss. However, to do so, the drain of the high side device is connected to the 170V DC power which is to be inverted into the 120C AC power. This is a problem because the 170V is the highest voltage in the system and in order for the switch to be turned on the voltage at the gate terminal must be 10V higher than the drain terminal voltage. In order to achieve the extra voltage necessary to switch on the device, a MOSFET driver is used with a bootstrap capacitor.

The MOSFET driver operates from a signal input given from the microcontroller and takes its power from the battery voltage supply that the system uses. The driver is capable of operating both the high side and low side devices, but in order to get the extra 10V for the high side device, an external bootstrap capacitor is charged through a diode from the 12V power supply when the device is off. Because the power for the driver is supplied from the low voltage source, the power consumed to drive the gate is small. When the driver is given the signal to turn on the high side device, the gate of the MOSFET has an extra boost in charge from the bootstrap capacitor, surpassing the needed 10V to activate the device and turning the switch on.

## Microcontroller

In order to use the H-bridge properly, there are four MOSFETs that need to be controlled. This can be done either with analog circuits or a microcontroller. In this case, we chose the microcontroller over the analog system for several reasons. First, it would be simpler to adapt. With an analog system, it would be difficult to make changes for the desired output. In many cases, this is a desired trait, as it would be designed for a single purpose and therefore a single output. However, as this is something that is designed to be available all over the world, it needs to be adjustable to different standards of frequency and voltage. With an analog circuit, this would require a different circuit that it would have to switch over to, while with a micro-controller, it merely requires a change in the program's code.

The second advantage of using a microcontroller is that it can allow for easy feedback to control the power flowing through the load. One of the problems that can occur with systems like this is that the variances in load can cause variances in the supplied current and voltage. With a microcontroller, it is possible to have it “look” at the power output and change the duty cycle based on whether or not the load requires additional power or is being oversupplied.

# Methodology

A pure sine wave inverter is a complex devices as there are many steps along the process of turning the DC input voltage into a pure $120V_{rms}$ AC output. In order to better understand this process, we have broken it down in the follow sections where we detail the function of each part, how it was constructed, and its interaction of wither sections.

## Block Diagram

The block diagram shown below shows the various parts of the project what will be addressed. The control block is simply the microcontroller. It generates both the PWM and square wave signals needed in controlling the MOSFET drivers. The signals from the drivers are then used to drive the four N-channel MOSFETs in the H-bridge configuration. The output signal form the h-bridge is then sent through an low-pass LC filter so that the final output is a pure sine wave.

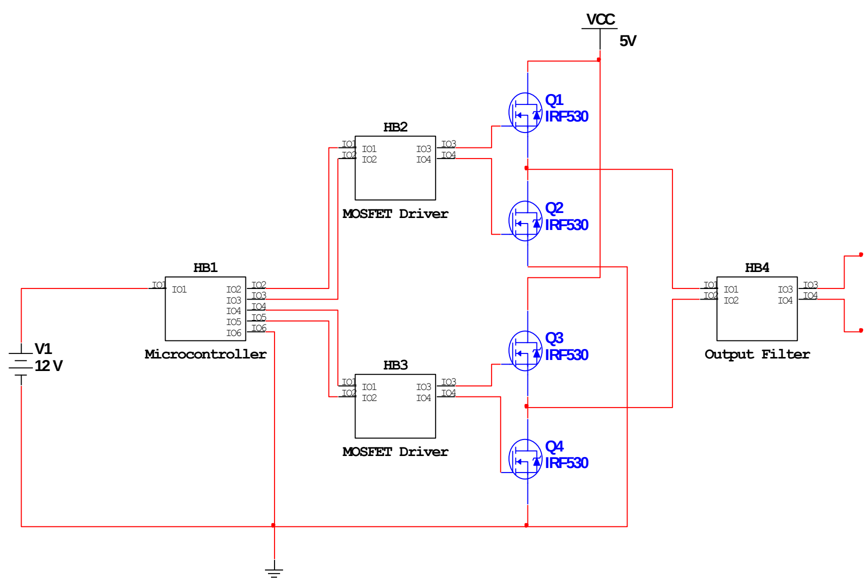



Figure 3: Basic Component Diagram

## Microcontroller Waveform Generation

The microcontroller is tasked with generating the four control signals that the MOSFET drivers use. They are two 60Hz square waves, each at 180 phase angels of each other, and two 60Hz 2-level pulse width modulation signals operating at a switching frequency of 50kHz also at 180 phase angels of each other. This was done by integrating the sine wave at each of the potential switch points, and generating a signal each time the overall integral increases by the value of a full pulse. The signal from the microcontroller operates at 3V as seen below in Figure 4.

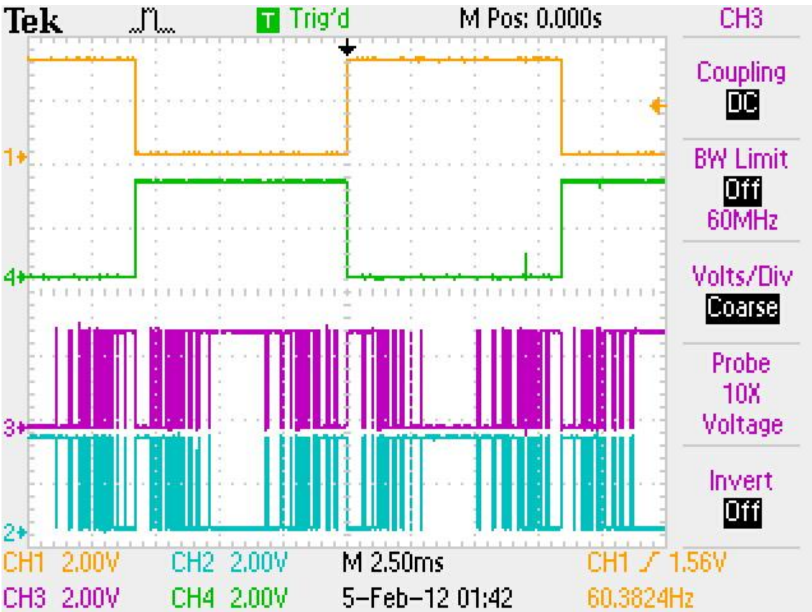



Figure 4: MSP430 Initial Output

## MOSFET Drivers Input

To test the operation of the system and to save on time, an MSP430 microcontroller was borrowed from the ECE Shop temporarily. The signals from the microcontroller operate at approximately 3V, which is below the minimum of 3.3V needed operate the MOSFET drivers. Instead of using a device such as a boost converter to increase the voltage, we implemented a simple logic switch which could operate using the 3V signal from the microcontroller. Using a simple BJT as a switch and sourcing the 12V batter power supply, we can create a 12V signal that has the same waveform shape needed to operate the MOSFETs. The 12V square wave and PWM signals can be seen below in Figure 5.

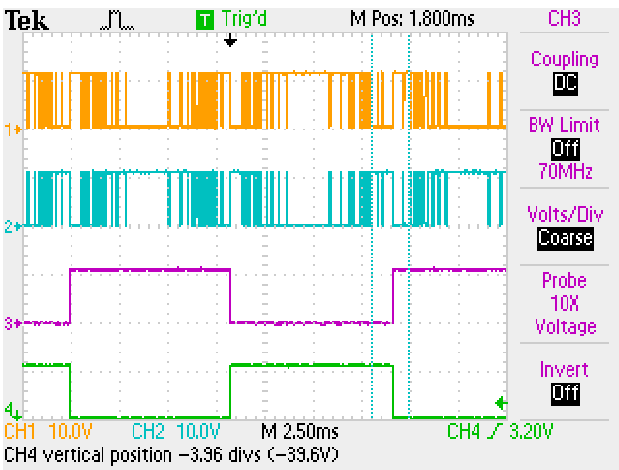



**Figure 5: MOSFET Drivers Input**

## MOSFET Drivers

The desired sine wave output has to be centered around zero so that means the power across the load has to been seen as both positive and negative. This is accomplished using four N-channel MOSFETs in an H-bridge configuration as seen below in Figure 6. For a driver, the

IR2110 was selected as it is more than capable of handling the high power requirements our project would be operating at.

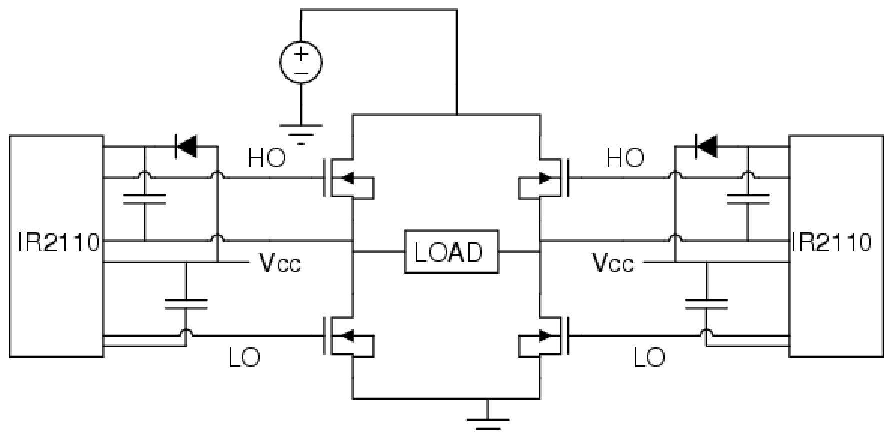



**Figure 6: H-Bridge Configuration**

The difficulty in this system however is that in the high side MOSFETs, the gate voltage (12V) needs to be at least 10V higher than the drain to source voltage (170V). To meet this requirement, a bootstrap capacitor is needed to maintain the voltage difference approximately 10V above the drain to source voltage. In the full bridge configuration, two of these devices are needed, one for each high side MOSFET. The figure below details the configuration of the bootstrap capacitors.

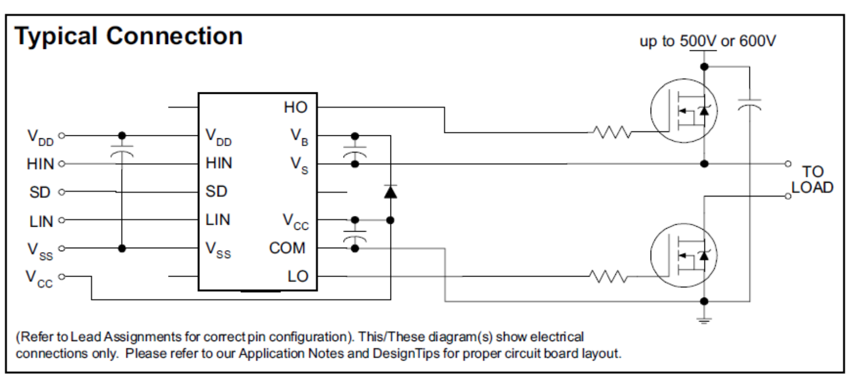



**Figure 7: MOSFET driver circuit**

As stated earlier, operation of the IR2110 device will be controlled through the signals genearted by the microcontroller. Each driver will input either both of the PWM signals or the square wave signals, being fed into the HIN and LIN pins simultaneously. If the internal logic a logic high, the HO pin will be driven and if a logic low is detected, the LO pin is driven. As they are not needed in the operation of the circuit $V_{SS}$, $V_{DD}$, and SD will be grounded.

The compents for the bootstrap capacitors and diode are connected as shown above and their values are calculated using International Rectifiers AN-978 application notes. The formula for the minimum bootstrap capacitor values obtained from this document is:

$$C \geq \frac{2[2Q_g + \frac{I_{qbs}}{f} + Q_{Is} + \frac{I_{Cbs}}{f}]}{V_{CC} - V_f - V_{LS} - V_{Min}}$$

Using the values determined from the data sheets of the components used as seen below, we calculated the square wave side of the bridge to be 2μF and the PWM side of the bridge to be 56nF. The diode used is an IR 8ETu04-ND 8A 400V Ultrafast Rectifier.

$Q_g$ = Gate Charge of High Side FET = 110nC
$I_{qbs}$ = Quiescent current for high side driver circuitry = 230uA
$Q_{ls}$ = Level shift charge required per cycle = 5nC (given in application note)
$I_{cbs}$ = Bootstrap capacitor leakage current = 250uA
$f$ = Frequency = 60Hz for left side of bridge, 50kHz for right side of bridge
$V_{cc}$ = Supply Voltage = 12V
$V_f$ = Forward voltage drop across bootstrap diode = 1.3V
$V_{ls}$ = Voltage drop across low side FET = 1.5V
$V_{min}$ = Minimum voltage between $V_B$ and $V_S$

Driving the four MOSFETs in an H-Bridge configuration allows +170, -170, or 0 volts across the load at any time. To achieve this, the left side of the bridge is driven by the PWM signals to determine whether the output voltage is non-zero or zero while the right side of the bridge is driven by the square wave signals to determine the polarity, either positive or negative. The MOSFETs used in the design are IRFB20N50KPbF Power MOSFETs rated for 500V at 20A with an $R_{ds}$ of 0.21Ω.

## Output Filter

The final component necessary to output a pure sine wave signal is an output filter. For our circuit we need a basic LC lowpass filter with the following setup below in Figure 8.

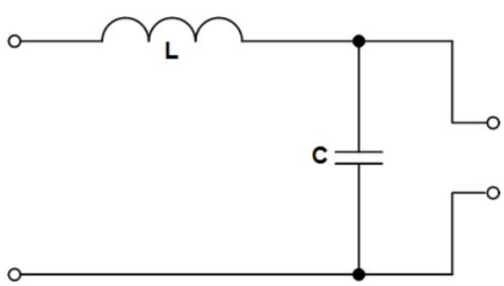



**Figure 8: Basic LC Filter**

This will filter out all the excess noise above the critical frequency $f = \frac{1}{2\pi LC}$. The goal for this was to bring the critical frequency as close as possible to the desired frequency of 60 Hz, removing other harmonics that crop up within the system. The issue with the filter is one of component size and availability. The slower the cutoff frequency, the greater the capacitance and inductance required to properly create the filter. Therefore, filter design becomes a tradeoff between the effectiveness of the filter and the cost and size of the components.

# Implementation

While designing an inverter can be complex, it does become easier when broken down into its component steps. The following sections detail each component within the project, as well as how each section is constructed and interacts with other blocks to result in the production of a 120V pure sine wave power inverter.

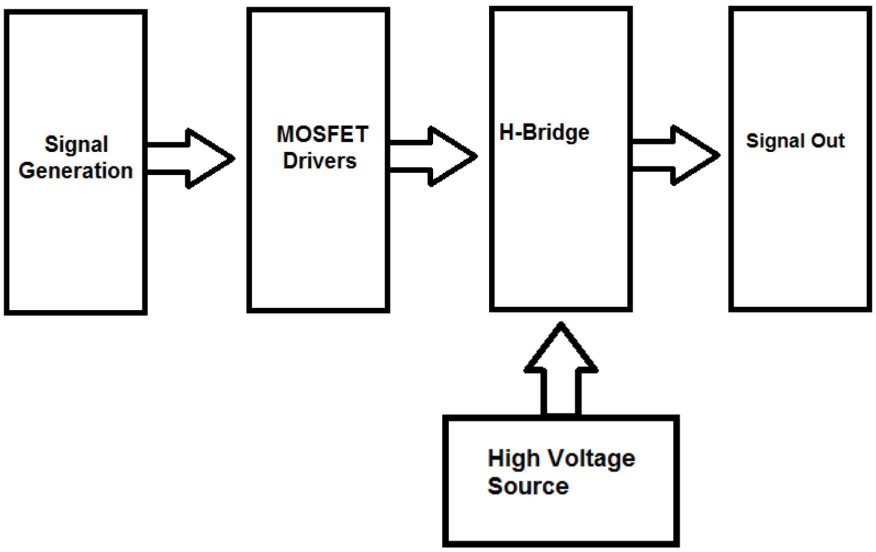



Figure 9: Overall Block Diagram

With this, the assumption is made that that the high-voltage DC source already exists, whether it is from a transformer or a boost converter. There are many examples of those already on the market, and they are outside the scope this project. The block diagram in this figure shows the three steps we need to generate the output signal. The signal is generated within the microcontroller. Then, it is input to the MOSFET drivers to provide a safety catch as well as the ability to keep the MOSFETS active when they are high. Finally, it passes into the MOSFETS of the H-bridge and draws power from the high voltage supply through the filter to generate the appropriate output signal.

## Design Process

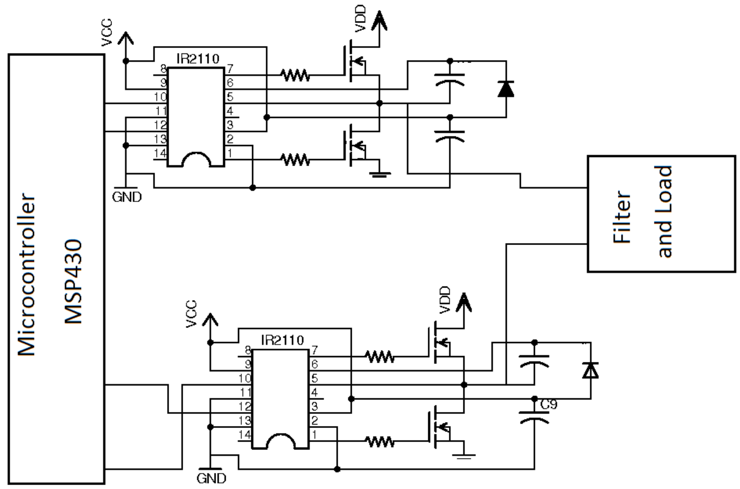



Figure 10: Inital Testing Circuit Design

This was the initial design of the circuit. The first task accomplished was an initial design for the output waveforms for the MSP430. In order to design these, the waveforms were simulated in MATLAB, to predict an accurate output.

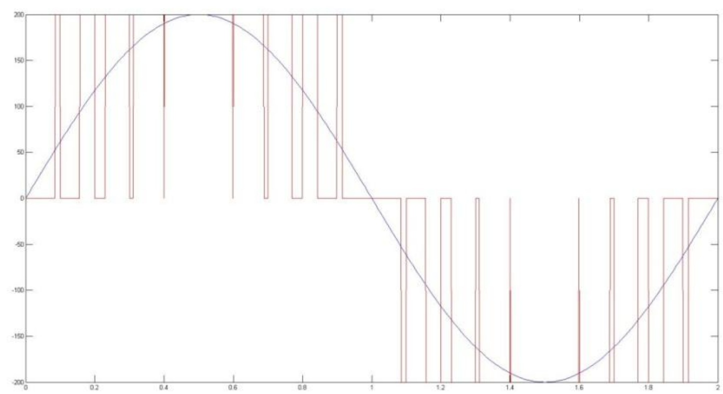


Figure 11: Simulated 3-Level PWM

This first signal design was generated by integrating the sine wave at each potential switch point for the signal, and then adding a pulse of either positive or negative voltage whenever the integral had increased by more than a pulse's power. The first result of this design was the following microcontroller output:

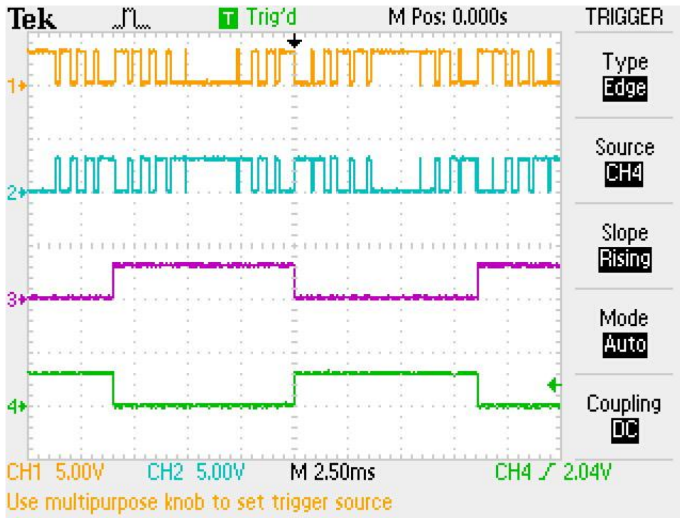



Figure 12: Initial MSP430 Output

The bottom two signals were the 60Hz base signal, while the upper two were the PWM activation. There were two sets of inverted signals to act as controls to both sides of the pillars of the "H"; when one MOSFET was on, the other on that side would be off, and vice versa.

Almost immediately, an issue appeared: despite the specs of the MOSFET driver listing 3V as the minimum input voltage and the MSP430 outputting a high of 3.3V, the signals were not triggering the driver successfully. After testing to ensure that the drivers did, in fact, work and that they had been installed correctly, it appeared that there was an intermediate step necessary to amplify the signal to the point where the drivers would receive it. To accomplish this, a pull-up resistor and a BJT were connected to the output of the microcontroller, pulling the on signal up to the 12V rail that also powered the MOSFET drivers, resulting in enough signal to pass through the drivers and activate the MOSFETs.

# Results

After the various iterations of the circuit and the minor bug fixes that had to be made, the final product looked like this:

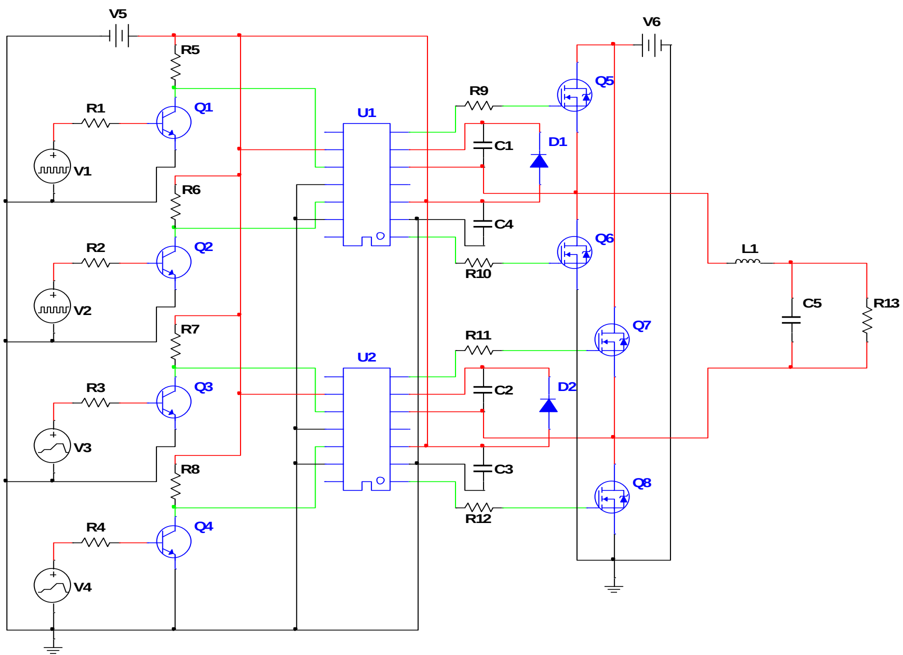



Figure 13: Final Circuit Diagram

The first tests performed with this version of the circuit were low voltage tests with varying filters and loads, in order to see how the circuit responded. With no load and filter componants at 1 mH and 100 µF, the following output was taken across the open “load”.

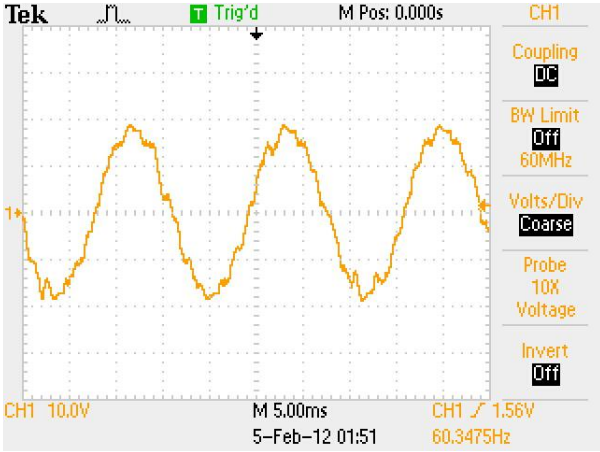



**Figure 14: Filtered Output**

When a standard 12V, 25W light bulb was used as the load the waveform smoothed to the following:

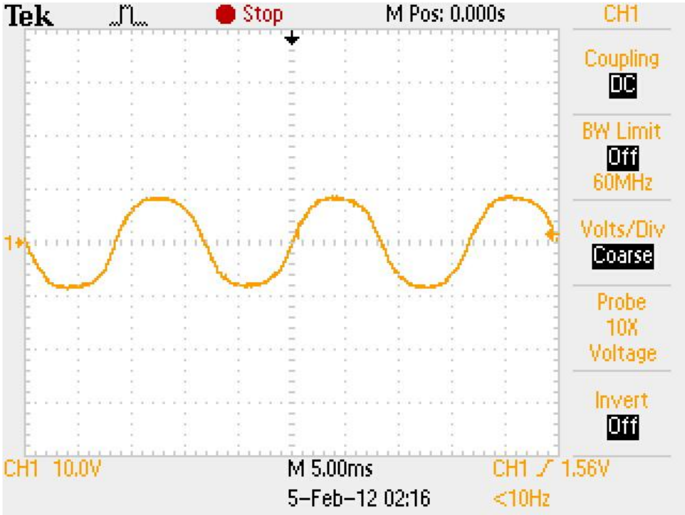



**Figure 15: Filtered Output with Load**

This led to a significant voltage drop across the filter due to the very low resistance of the light bulb, as compared to the voltage source in use. The voltage across the H-bridge also dropped when this path was added. Therefore, to calculate the low voltage efficiency of this circuit, the power entering the H-Bridge was compared to the power exiting the H-bridge, which resulted in an efficiency of 62.7%. This seemed low; however, as the circuit was tested with higher voltages, the efficiency increased as the static losses became less relevant.

When testing the higher voltages, due to the cap on the switching speed, we were unable to find filter components large enough to properly test a filtered high-voltage circuit. However, we did obtain the following output across a 100W, 120V standard light bulb:

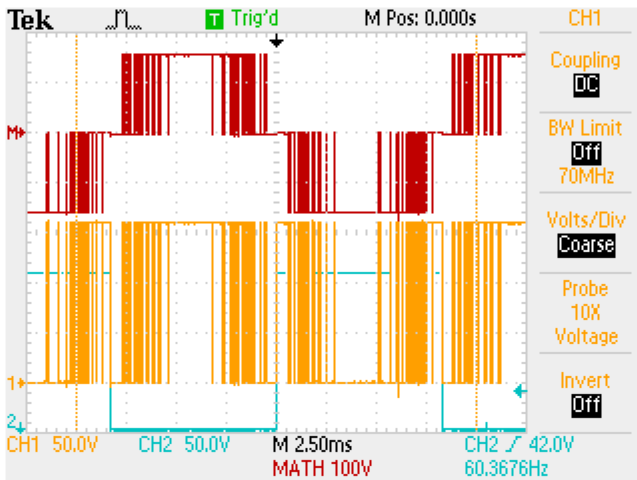



Figure 16: Unfiltered High Voltage Outputs

When we measured the power loss between the DC source and the load, we obtained a much better efficiency than in the low voltage tests: 82.6%.

# Recommendations

The overall goals of this project were met; however, there is still much room for improvement that can be made. The primary recommendation would be to find and use a microcontroller which is capable of outputting to the MOSFET drivers directly without the need for an intermediate step to increase the power of the input signal. The MSP430 could only generate signals up to a certain frequency before they became unstable and the BJT's used to increase the signal power further lowered the limit of the switching speed. With a more powerful microprocessor outputting a faster signal, it would also be easier to implement the filter design of the system. With a faster switching speed, the cutoff frequency of the filter could also be higher and not require such bulky or hard to find components.

Another improvement that can be made is a feedback system which would give the microcontroller a view of the output across the load so that the signals controlling the system could be adjusted according to certain parameters in the programming. As different loads are connected and disconnected, the efficiency and output of the system will change. In order to keep the system running at $120V_{rms}$ and 60Hz, it has to be able to adapt to changes in its load. Implementing an opto-coupler is one recommended method as it would have minimal effect on the output and project the microcontroller from the high power of the system.

# Conclusion

The goals of this project were to create a working DC-AC sine wave inverter that could efficiently provide a kilowatt of power using 3-level PWM and comparing it to other methods currently on the market. Looking at this goal, we accomplished it, though there is room for improvement. Our inverter had an efficiency of over 80%, a significant improvement over similar 2-level PWM systems. By comparison to the modified sine systems, the harmonics were significantly smaller, allowing for more effective use with delicate circuitry.

While we did not test the maximum power provided by this circuit, we are confident that the inverter we built can withstand this power. The parts are all well within specs limits with an additional 30% safety margin, therefore we can safely say that it will provide the necessary power.

This project provides a good building block that can be added in to many general-use high-power applications, as well as a base to work off for a self-regulating power supply, especially if the microcontroller controls both the DC/DC converter and the DC/AC inverter.

# Bibliography


Doucet, J., Eggleston, D., & Shaw, J. (2007). *DC/AC Pure Sine Wave.* Worcester: WPI.

Fairchild Semiconductor. (2011, October). 2N3904 NPN General Purpose Amplifier. Fairchild Semiconductor Corporation.

International Rectifier. (2005, March 23). R2110(-1-2)(S)PbF/IR2113(-1-2)(S)PbF High and Low Side Driver. El Segundo, California, USA: IR World Headquarters.

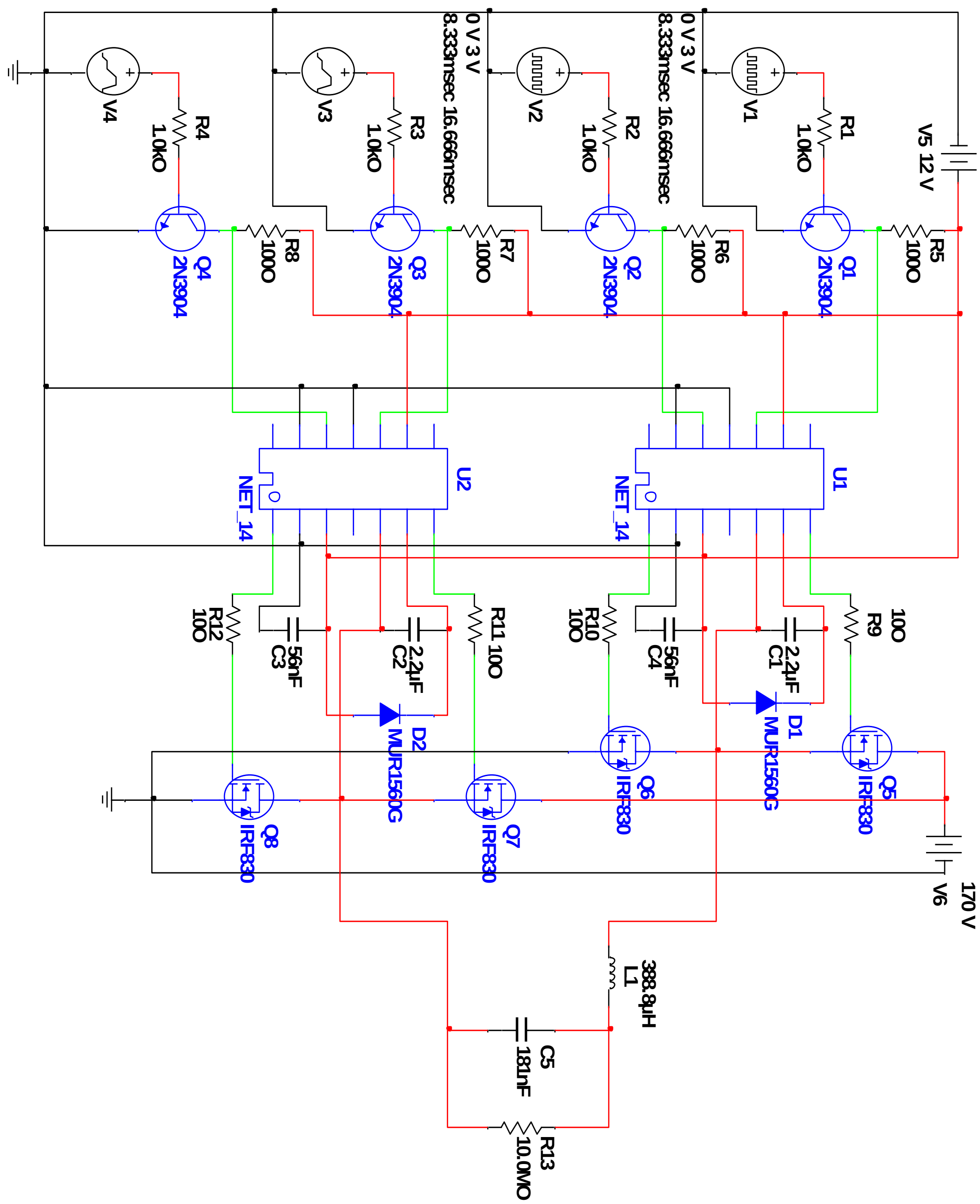
International Rectifier. (2007, March 23). Application Note AN-978. El Segundo, California, USA: IR World Headquarters.

STMicroelectronics. (2007, March 11). STTH8R04 Ultrafast Recovery Diode. STMicroelectronics.

Vishay. (2010, April 5). IRFB20N50K, SiHFB20N50K Power MOSFET. Vishay Siliconix.

# Appendices

## Appendix A: Circuit Diagram



# Appendix B: Code for Microcontoller

```
/************* Include Headers **********************/
#include "msp430x44x.h"      // Definitions, constants, etc for msp430F449
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <in430.h>
/****************************************************/

unsigned int timer = 0;
unsigned int leap_cnt = 0;



#pragma vector = TIMERB0_VECTOR
__interrupt void Timer_B0(void)
{
  if (timer==243) timer = 0;
    timer++;
    leap_cnt++;
  }
  else{
    leap_cnt = 0;
  }
}
void LEDOff(void);   // sets up green LED port and turns LEDs off
void LEDdisplayHex(unsigned char num);  // displays hex code for digit num

void init_sys(void);      // MSP430 Initialization routine

void main(void)
{
  int         pos = 1;
  unsigned char   lastKey=0;

  WDTCTL = WDTPW + WDTHOLD;   // Stop watchdog timer
  init_sys();                 // Initialize the MSP430

  int output[4] = {0x33, 0xC3, 0xCC, 0x3C};
  short int pointers[244] =

{0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,1,0,0,1,0,1,0,1,0,1,1,0,1,0,1,1,1,0,1,1,1,0,1,
1,

1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1
,

1,1,1,1,1,1,0,1,1,1,1,0,1,1,0,1,1,0,1,1,0,1,0,1,0,1,0,1,0,0,1,0,0,1,0,0,0,1,0
,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,1,0,0,1,0,1,0,1,0,1,1,0,1,0,1,1,1,0,1,1,1,0,1,1
,
```

```
1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1
,

1,1,1,1,1,1,0,1,1,1,1,0,1,1,0,1,1,0,1,1,0,1,0,1,0,1,0,1,0,0,1,0,0,1,0,0,0,1,0
,0,0,0,0,0};
  for (int i=0;i<122;i++){
    pointers[122+i] = pointers[i]+2;
  }
  TBCTL = 0x0200 + CNTL_0 + ID_0 + MC_1;
  TBCCR0 = (70);
  TBCCTL0 = CCIE;
  _BIS_SR(GIE);

  while (1){
    P2DIR |= 0xFF;      // Set P2.7-2.4 to output direction
    P2SEL &= 0x00;
    //if (timer%2 < 1)LEDdisplayHex(0xCC);
    //else if (timer %2 >=1)LEDdisplayHex(0x33);
    LEDdisplayHex(output[pointers[timer]]);

  }


}

void LEDOff(void)
{
  P2DIR |= (BIT7|BIT6|BIT5|BIT4);     // Set P2.7-2.4 to output direction
  P2SEL &= ~(BIT7|BIT6|BIT5|BIT4);    // P2.7-2.4 I/O option
  P2OUT |= (BIT7|BIT6|BIT5|BIT4);     // P2.7-2.4 output = 1 (LEDs off)
}

void LEDdisplayHex(unsigned char num)
{
  unsigned char   tmp_num;

  tmp_num = (~num);
  P2OUT = tmp_num & 0xFF;
}

void init_sys(void)
{
  //initLCD();                  // Setup LCD for work
  //clearLCD();                 // Clear LCD display
  //setupKeypad();              // Setup Keypad ports
  LEDOff();
}
```

# Appendix C: Parts List

| Quantity | Componant | Cost (1) | Cost (10000) | Total Cost (single) | Total Cost (bulk) |
|---|---|---|---|---|---|
| 4 | IRFB20N50k MOSFET | 5.26 | 2.33 | 21.04 | 9.32 |
| 2 | STTH8R04D Diode | 1.73 | 0.657 | 3.46 | 1.314 |
| 4 | 100 Ω Resistor | 0.1 | 0.005 | 0.4 | 0.02 |
| 4 | 1000 Ω Resistor | 0.1 | 0.0005 | 0.4 | 0.002 |
| 4 | 10 Ω Resistor | 0.1 | 0.005 | 0.4 | 0.02 |
| 2 | 2.2 µF Capacitor | 0.4 | 0.04 | 0.8 | 0.08 |
| 2 | 56 nF Capacitor | 0.4 | 0.04 | 0.8 | 0.08 |
| 2 | IR2110 MOSFET Driver | 4.68 | 2.016 | 9.36 | 4.032 |
| 4 | 2N3904 PNP BJT | 0.19 | 0.0328 | 0.76 | 0.1312 |
| 1 | MSP430F449 | 10.1 | 4.938 | 10.1 | 4.938 |
| | | | | Total Cost | |
| | | | | $47.52 | $19.94 |