DESIGN OF A HIGH-LOW UNIPOLAR PULSE WIDTH MODULATED INVERTER

By

Danny Li and

Christian Ramos

Senior Project

ELECTRICAL ENGINEERING DEPARTMENT

California Polytechnic State University

San Luis Obispo

Spring 2010

TABLE OF CONTENTS

Section *Page*

Appendices

LIST OF TABLES AND FIGURES

Tables

Figures

ACKNOWLEDGEMENTS

Danny Li:

I want to thank the entire Cal Poly faculty for all the knowledge that they have helped me obtain. Specifically, electrical engineering professors I am grateful for the support and information that you have presented to me. I realized that I have learned a whole lot from you while working through this senior project. I want to specially acknowledge our advisor Professor Taufik for all the wonderful classes he has taught me, and which has inspired me to do this project. His commitment to students and their needs cannot be appreciated enough. Lastly, my parents have a special place in my heart for providing me the opportunity to pursue my education at Cal Poly and to follow my dreams.

Christian Ramos:

First of all, I would like to thank Professor Taufik for being my senior project advisor. I took his EE 410 Introduction to Power Electronics course, which lead to my interest in this part of the electrical engineering field. He has been great source of input and support. Also, I would like to thank the rest of my EE colleagues. They have been great friends who have helped and supported me throughout my college career. Lastly, I would like to thank my parents who have blessed me with the opportunity to attend this great program at a wonderful university. They have invested everything within me and I am truly grateful for their unconditional love and support.

ABSTRACT

This senior project report describes the process of designing a unipolar inverter. This high low, unipolar inverter utilizes switches in which a pair operates in high frequency while a second pair operates in low frequency. The purpose of this project is to convert a range of high DC voltage and convert it to an AC output voltage of 120 Vrms with a power rating of 100 W. Nonetheless, this particular unipolar method of implementing pulse width modulation results in harmonics at multiples of the fundamental frequency leading to easier filtering. However, the switching loss increases because of the complex switching circuitry. Overall, this report focuses on the design, construction, and testing aspect of the inverter project.

I. INTRODUCTION

In today’s world, power electronics play a vital role in processing and controlling the flow of electrical energy for various applications. To efficiently meet the design requirements for a specific application, electrical engineers implement solid-state switches to achieve the desired voltage, current, or frequency. These converters vary from a low (mW) to a high (MW) power range. Nevertheless, energy consumption and production are very important issues concerning power electronics. As fossil fuels and their reliance dwindle down, new methods of achieving power are necessary, such as renewable energy sources, which include solar energy from photovoltaic cells and wind energy from wind turbines. However, these sources obtain energy by D.C. power, a direct current source. Because utility companies distribute power as A.C. power, an alternating current source, an interfacing device is needed to convert between to the different power sources. Therefore, an inverter is necessary to change the D.C. power to A.C. power.

This senior project consists of a voltage source, pulse width modulated, unipolar inverter using a full bridge topology. This inverter receives a D.C. voltage as an input and outputs an A.C. voltage. As the input voltage remains constant, the magnitude and the frequency of the output is controlled by the pulse width modulation of the switches. Pulse width modulation changes the width of the square wave to adjust the output voltage at high frequencies. As a result, the sinusoidal output is easier to filter because the undesired harmonics are located at higher frequencies even though it may have a higher total harmonic distortion value. In addition, the unipolar inverter has two pairs of switches operating at different frequencies. The high side pair of switches operates at the high switching frequency, while the low side pair operates at the low control frequency. Therefore, the electromagnetic noise is decreased and some of the harmonic components are again located in higher frequencies so relatively less filtering is required.

II. BACKGROUND

Power inverter converts DC power to AC. Figure 2-1 displays the basic inverter circuit using the full bridge topology. There are several methods to do this. The two main ways, which use switched mode inverters, are square wave and pulse width modulation. The DC input voltage is adjusted to control the AC output magnitude of a square wave inverter. The switching frequency is adjusted to control the frequency. However, the methods used for a square wave inverter differ for a pulse width modulated inverter. In PWM, the controller adjusts both the magnitude and frequency. The difference in switching causes the harmonics to appear at different locations. Although the square wave signals are simpler, the harmonics exist mostly in the lower frequencies, which are close to the output frequency. In a PWM inverter, the harmonics are at multiples of the fundamental frequency, and thus are easier to filter. As a result, the output sinusoid has less distortion. Unfortunately, a PWM hold a major disadvantage in requiring complex switching circuitry as well as increased switching loss.

The two types of pulse width modulation inverters are bipolar and unipolar switching. Each unique switching technique creates either a unipolar or bipolar output at the load. The control signals depend on comparing a reference signal and carrier signal. A PWM requires a sinusoidal reference signal and triangular carrier signal.

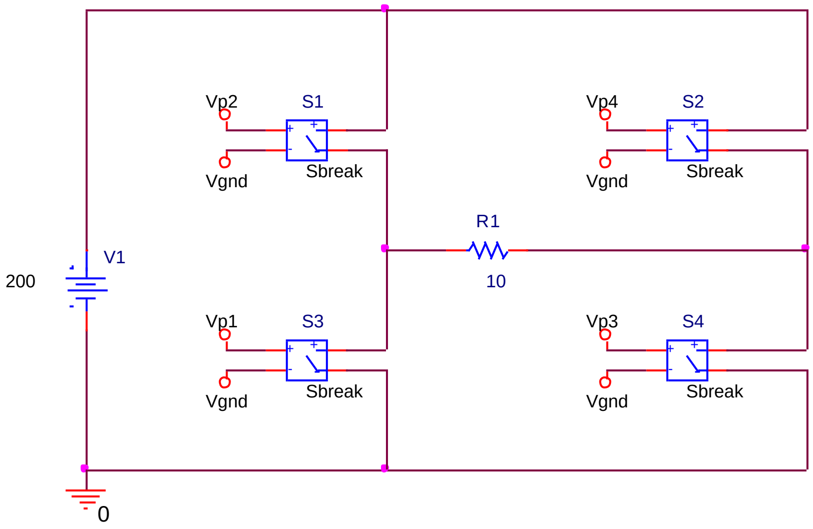


Figure 2-1: Full Bridge Inverter Topology

There are two types of unipolar switching. Type one operates both switch pairs at a high frequency. Type two uses a high frequency for one pair and a low frequency for the other pair of switches. We will be building a type two unipolar PWM inverter because it is simpler than type one. Figure 2-2 shows the output voltage switching characteristics for type two unipolar switching. The switching conditions are shown in tables I and II below where Vsine and Vtri are the sine wave and triangular wave control signals respectively [1].

Table I: UNIPOLAR SWITCHING - TYPE 1 LOGIC SIGNALS

Switch	On-State	Control Signals
S1	On	Vsine>Vtri
S2	On	–Vsine<Vtri
S3	On	–Vsine>Vtri
S4	On	Vsine<Vtri

Vo varies from Vdc to 0 or –Vdc to 0 and all four switches follow high frequency signals.

Table II: UNIPOLAR SWITCHING - TYPE 2 LOGIC SIGNALS

Switch	On-State	Control Signals
S1	On	Vsine>Vtri
S2	On	Vsine<Vtri
S3	On	Vsine>0
S4	On	Vsine<0

S1, S2 are high frequency and S3, S4 are low frequency at output frequency, such as 60 Hz.

Vo varies from Vdc to 0 or –Vdc to 0.

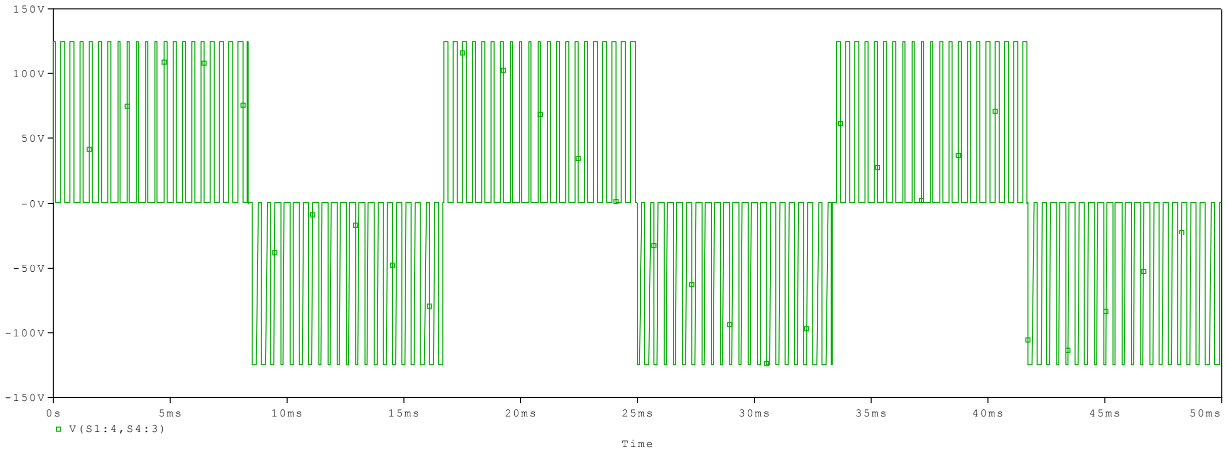


Figure 2-2: Unipolar Type 2 High-Low PWM Switching Characteristics

III. REQUIREMENTS

The requirements for our inverter design include the following:

- Output voltage equal to 120 Vrms
- Input DC voltage range of 170 V-220 V
- Maximum load current of 1 A
- Power rating of 100 Watts
- Full load efficiency greater than or equal to 85 %
- Operating frequency of 2.5 KHz and 60 Hz
- Digitally controlled by programmable microcontroller
- Circuit should fit in a 6" x 3" x 2" project enclosure
- Through hole soldering of components on a prototype board

These design requirements come from our senior project advisor Professor Taufik. For our inverter, the desired output voltage is the typical 120 Vrms used by various utilities. Therefore, our inverter can be connected to different input sources such as a solar cell array or wind turbine followed by a boost converter to step up the voltage to produce the necessary output. One challenging requirement is utilizing the microcontroller to provide the control signals to the overall circuit, which requires knowledge in embedded systems and programming in C.

IV. DESIGN

Our unipolar, full bridge, pulse width modulated voltage source inverter design is divided into two parts: hardware and software. The hardware portion of the design composes of the different components used to build the overall module. On the other hand, the software part of the design consists of the program necessary to implement the pulse width modulation to act as the controller of the inverter. The overall inverter design itself will follow the design done by Wibawa [2], see figure 4-1.

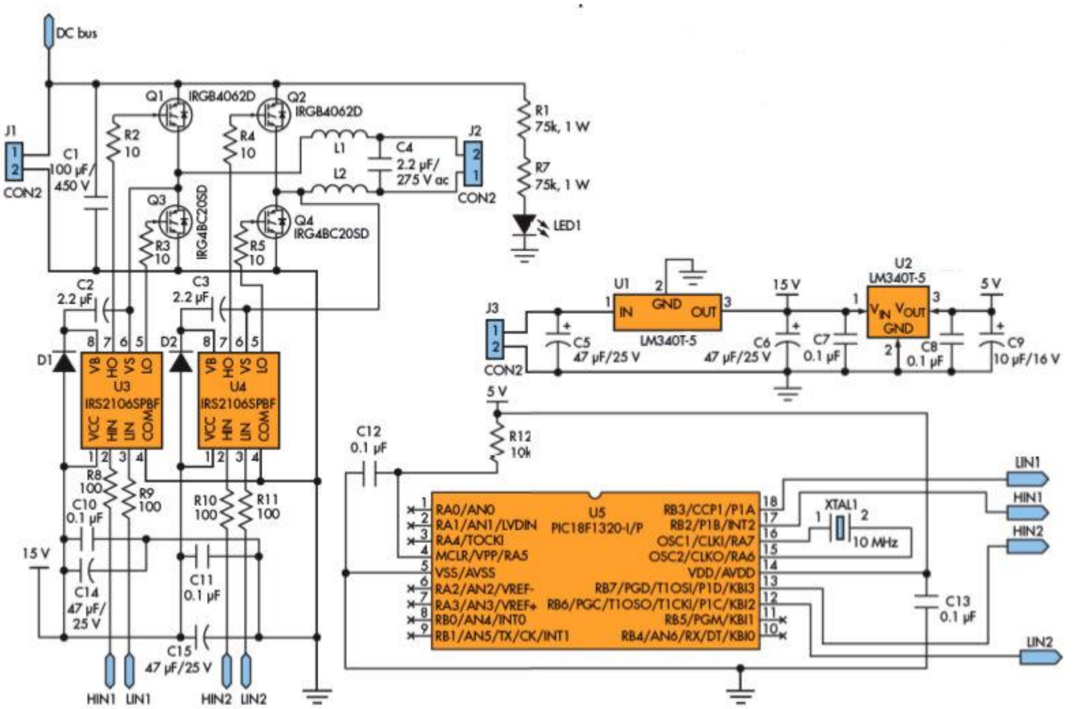


Figure 4-1: Unipolar, Full-Bridge, Voltage Source Inverter [2]

The hardware portion of the inverter design consists of the necessary components needed to build the circuit. For this inverter, insulated gate bipolar transistors (I.G.B.T.) are used as the switches because of their conduction characteristics. Insulated gate bipolar transistors are voltage-controlled devices with small turn on voltage for high voltage ratings. In our design, International Rectifier's high side IRGB4062D and low side IRG4BC20SD are used. These high voltage switches have a maximum voltage rating of 600 V with a turn on voltage of 1.65 V [3] and 1.4 V [4] respectively. For our unipolar, full bridge topology as shown in figure 1-1, the pair of high side IRGB4062D switches operates at a high frequency while the pair of low side IRG4BC20SD switches operates at a low frequency in order to minimize the harmonics and power dissipation. Furthermore, these insulated gate bipolar transistors are co-packaged with an ultra fast soft recovery diode to minimize switching losses, which can be seen in figure 4-2.

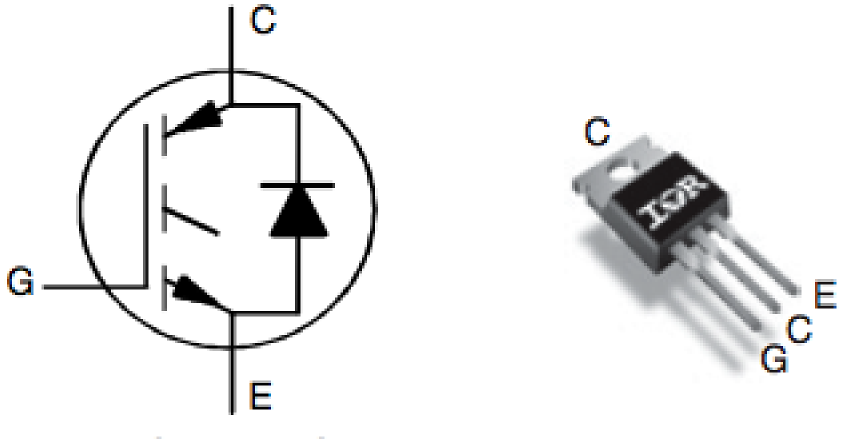


Figure 4-2: I.G.B.T. Circuit Diagram [3]

Nevertheless, these insulated gate bipolar transistors are connected and driven by a high side driver and a low side driver. Drivers are necessary components that turn on the switches while isolating the control signals from the switches' high voltage. For this inverter design, International Rectifier's IRS2106SPBF are used to drive the high and low side insulated gate bipolar transistors. Because of the driver's bootstrap power supply operation [5], the isolated power supply for the high side is removed, which reduces the amount of components. Also, power losses are minimized because of the freewheeling current on the low side of the switches eliminating freewheeling current on the high side. Therefore, overall amount of components are reduced while maintaining a high efficiency. Additionally, the drivers' output is connected to a current regulator because it reduces the cross conduction within the driver [5]. The current regulator utilizes internal current-limiting operations in order to act as a current buffer such that current cross conduction is minimized within the driver. Cross conduction of current can result in power loss, which reduces the overall efficiency and performance of the design. Therefore, this is a necessary issue that needs to be address when designing the inverter.

To power the microcontroller and the drivers, a voltage regulator power circuit is used to provide the necessary voltage. A 5 V and 15 V output is necessary to power the microcontroller and drivers respectively. For our design, we used two LM340T-5 chips connected to input and output capacitors, which provide fixed output voltages [6]. As a result, the voltage regulators are connected in series such that the output voltage produces 15 V and 5 V outputs with an input voltage range from 18 V to 23 V. This configuration is displayed in figure 4-3.

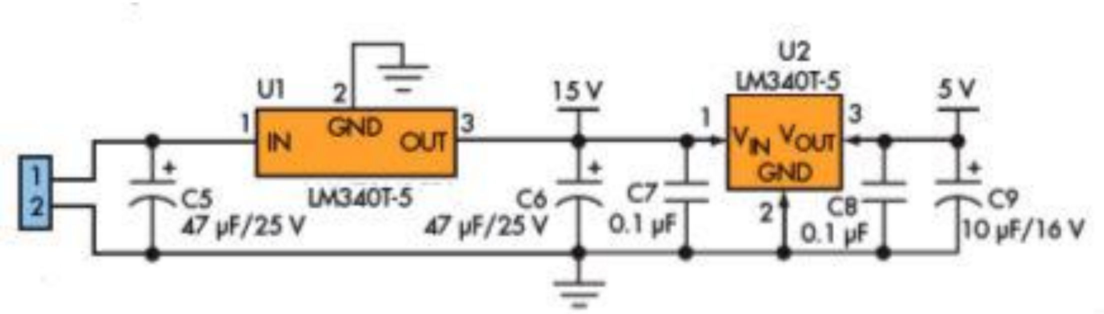


Figure 4-3: Voltage Regulator Power Circuit

The output of the unipolar inverter needs to be filtered once it is operational and producing an output. The unfiltered output is a DC wave as seen in figure 2-2. However, the filtered output needs to be an AC sinusoidal wave to produce the required RMS voltage. Therefore, a resonant LC filter is needed to produce such a waveform. The inductor maintains a constant current so it charges and discharges the capacitor resulting in a cycle of resonance. Figure 4-4 displays the required filter for our inverter design.

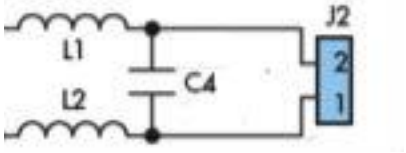


Figure 4-4: Resonant AC Output Filter

For the software portion of the design, a microcontroller is programmed such that the control signals are generated for the drivers and the insulated gate bipolar transistors. For our design, the PIC18F1320 8bit microcontroller is used in order to control the overall circuit. The microcontroller is coded in C utilizing the pulse width modulation functions. The pulse width modulated wave increases and decreases in width when the sine wave reaches a high, maximum value or a low, minimum value.

Therefore, the duty cycle increases or decreases according to the magnitude of the sine wave. Nonetheless, these control signals are programmed in order to manipulate the hardware to function as unipolar voltage source inverter.

V. CONSTRUCTION

We first constructed our design by ordering the parts and breaking it down to several components. First, we started with a blank prototype board (4” x 3”) with copper traces so common nodes can be used to make easier connections, such as a common ground node. Also, one inch standoffs are screwed in Figure 5-1 displays the blank board before we started soldering on components.

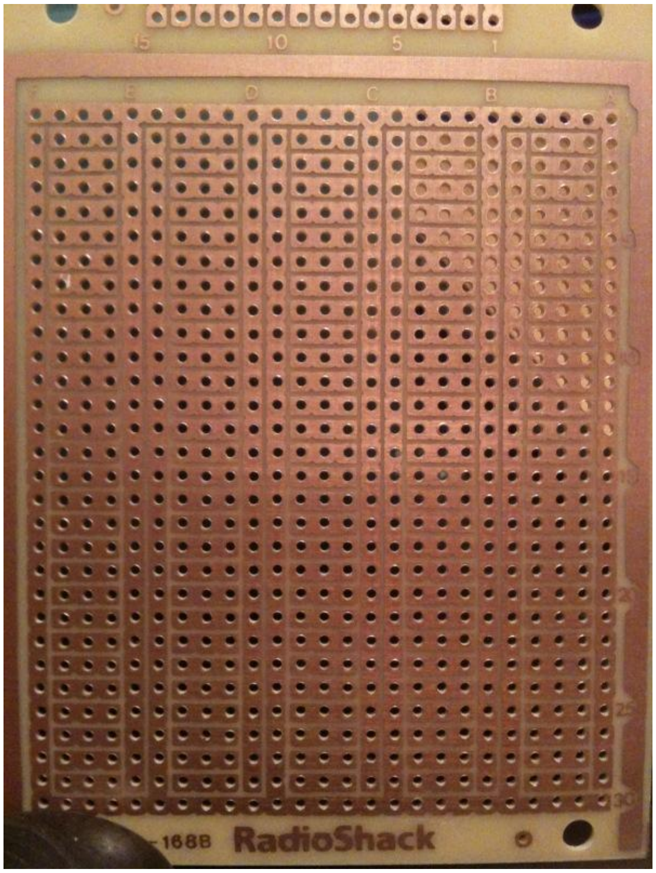


Figure 5-1: Blank Prototype Board with Copper Traces

We then began soldering on each part by component. First, the voltage regulator power circuit is soldered onto a portion of the board. According to the completed circuit displayed in figure 5-2, this power circuit is located on the bottom right side.

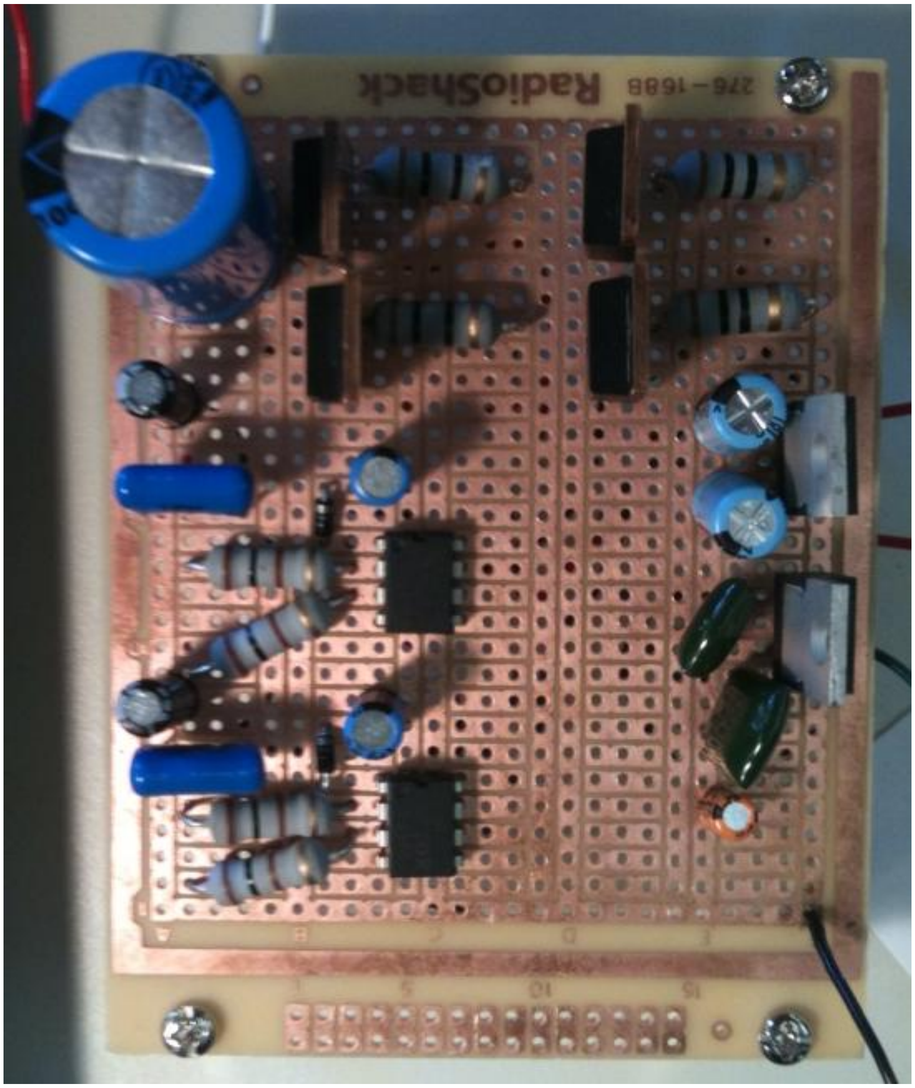


Figure 5-2: Prototype Board with Soldered on Components

Next, the I.G.B.T. switches with its respective components are soldered on with the high frequency side switches and low frequency side switches located on the top left and right sides according to figure 5-2. Afterwards, the high and low side drivers with its respective components are soldered onto the board. In figure 5-2, the high and low side drivers are located on the lower left side of the prototype board. Then, the input filter capacitor of the circuit is soldered on the top left side of the board. Once each component is soldered onto the board, several wires are used to interconnect between the components. Figure 5-3 displays the wire connections between the components.

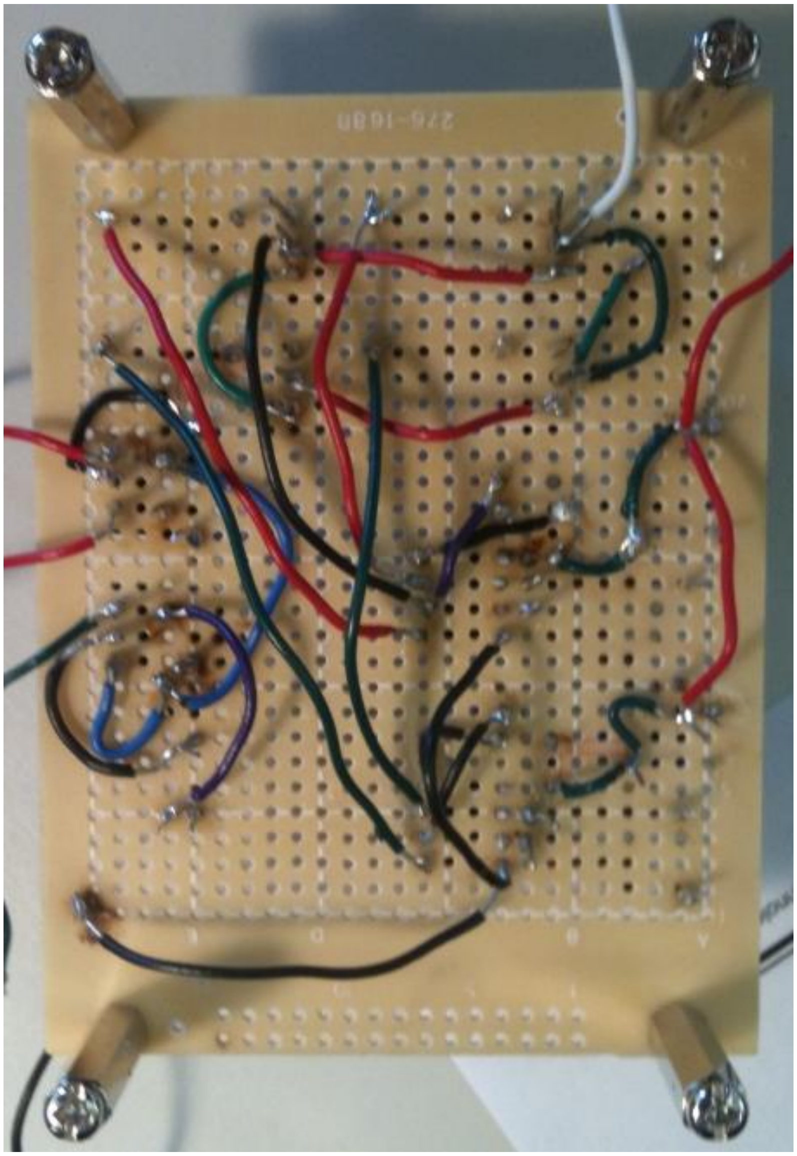


Figure 5-3: Prototype Board Underside

The control signal generation was a massive journey. In an attempt to stick with our original design, we wanted to use the PIC18F1320 but found that the programmer for the chip was not economical. This led to a search of alternative microcontrollers to make the project easier to pick up by others. The main goal was to find some way to generate PWM based control signals. In the search, ATMEL microcontroller turned out to be the only other viable option. The functionality and output needed eventually brought us back to the original design. It took a long time to find a cheap, stateside module that can program a microchip. However, we finally found a retailer in California for a good price. We purchased a USB compatible programmer making it easier to interface between the computer and programmer. Figure 5-4 displays the microcontroller programmer. The microcontroller is soldered onto another board because there is not enough space for an 18-pin microchip. Before soldering it onto another board, it needs to be programmed so that it can perform its function.

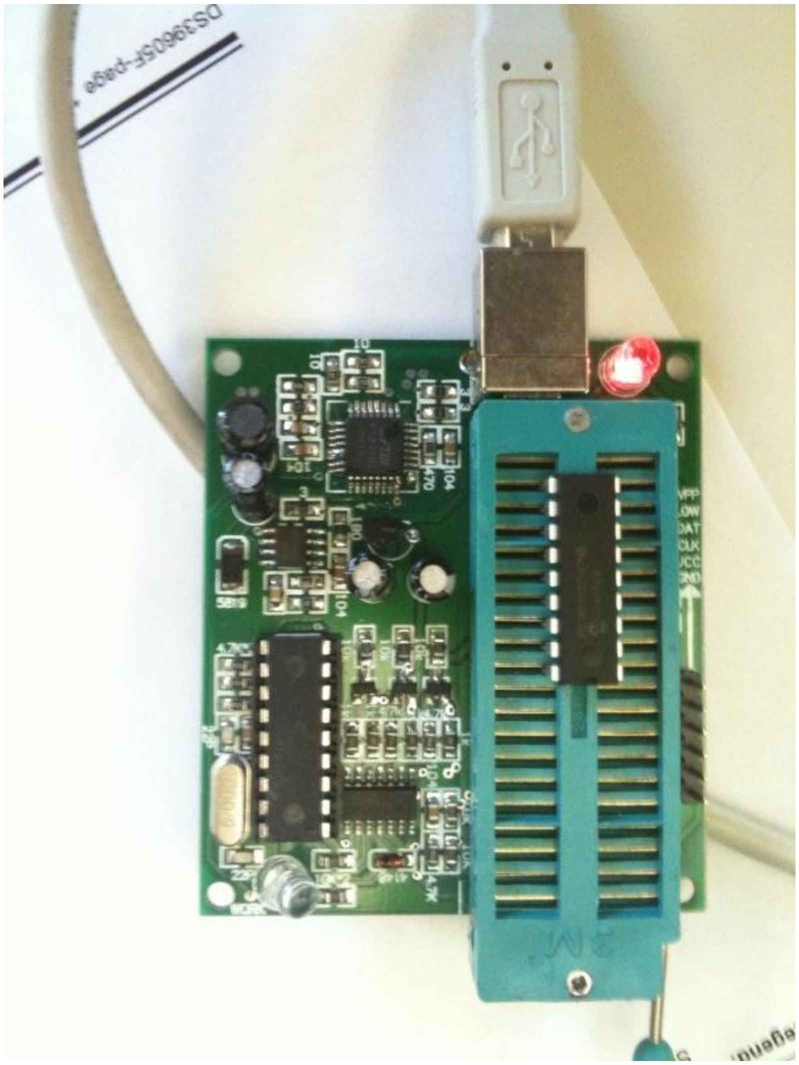
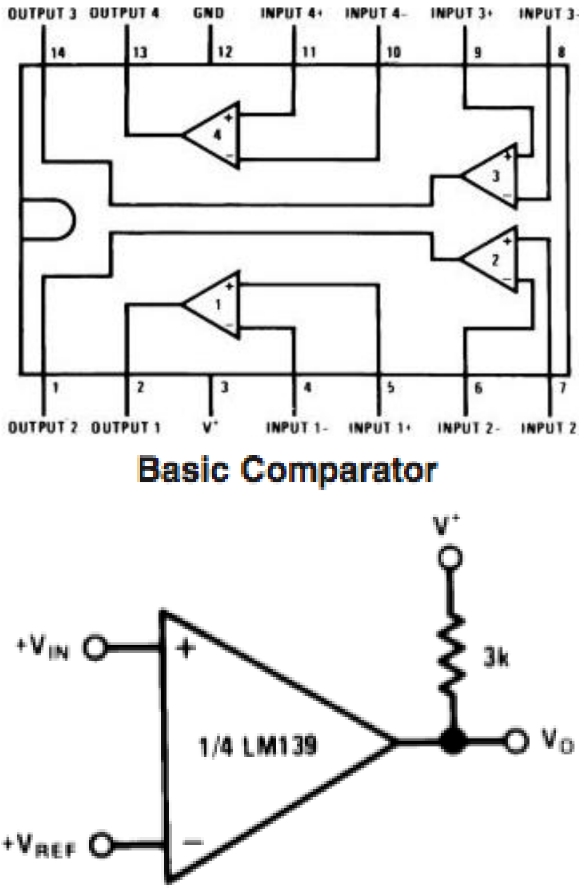


Figure 5-4: USB Microcontroller Programmer

However, after many hours debugging and reprogramming, we were unsuccessful at achieving an operational program that provided our specific requirements.

Because we could not get our software to function, we decided to build an analog controller as a last ditch effort by using comparators. According to figure 5-5, LM339 integrated circuit’s operational amplifiers are configured to operate as comparators receiving signals from the function generators as references [7].



Basic Comparator

Figure 5-5: LM339 Voltage Comparator [7]

Since this is a last minute solution, the circuit had to be built on a breadboard for quick and easy construction and testing. Figure 5-6 shows our built analog control circuit.

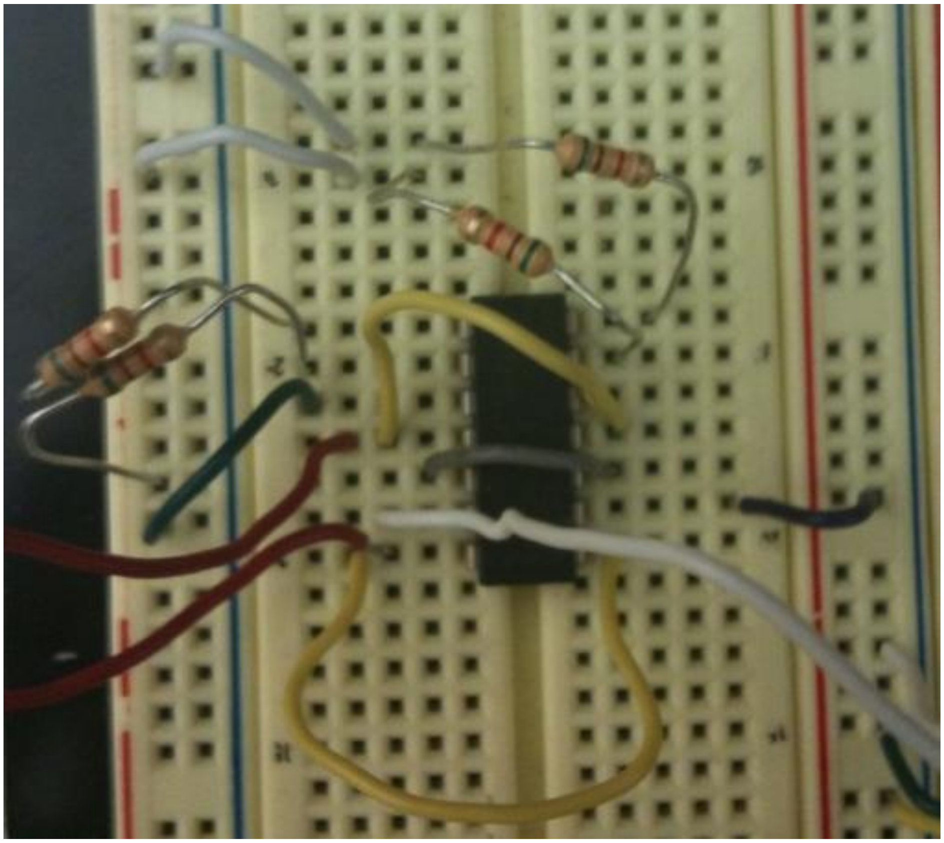


Figure 5-6: LM339 Analog Controller

Also, no output filter was constructed because we needed to produce a waveform before we decide to filter anything. Otherwise, the filter may cause some unnecessary leakage current spikes that may damage our whole project.

VI. TESTING

The overall circuit is tested by testing the control module. However, we did test the voltage regulator power circuit first because it is needed to power the other components in the circuit. We varied the input voltage and recorded the two output voltages in which we expect an output of 15 V and 5 V. Table III displays the results.

Table III: VOLTAGE REGULATOR POWER CIRCUIT VOLTAGES

Vin [V]	Vout1 [V]	Vout2 [V]
13.112	12.388	3.206
14.105	13.375	3.606
15.102	14.364	4.058
16.093	15.346	4.491
17.065	16.311	4.91
18.002	17.031	5.044
19.028	17.545	5.0443
20.034	18.046	5.0445

The PIC18F1320 presented a challenging learning curve. Time was spent on becoming familiar with the MPLAB IDE, followed by the C Complier associated with the software. After implementing some sample code and testing a sample LED circuit, the PWM output was the next objective.

Using the PWM.h library, the process was simplified for obtaining an output, but only after getting familiar with the compliers. Next, random values were entered into the period and duty cycle registers to observe an output of any PWM waveform. With success, the next step involved figuring out the appropriate values by applying given formulas from the datasheet. The PIC equations were confusing, which made it difficult to determine PWM values. In order to obtain the appropriate PWM signal, the periods and duty cycle must vary according to the sinusoidal and triangle waveform comparisons. In response, the only way to determine these values was through manually reading the values off a waveform simulation. Finally, with the duty cycles and period in hand, the settings were made and the values were entered through an array of duty cycles.

Using an excel spreadsheet, all the duty cycle and periods were mass produced and converted from time in seconds, to decimal, and lastly hex for entering in the program. This generated all the signals we needed at one output but not at all four outputs. The desired switching called for four different signals, with the closest method to generate them was by applying a reverse mode on the PIC18F1320 [8]. Switching between forward and reverse mode satisfied both high side signals but failed in producing the signals with accurate timing. The correct timing called for the flipping to occur after the positive part of a duty cycle, but the microcontroller will only flip after the end of each period. The low side requires a much slower switching signal, but it can still work by reversing the output modes from output high to output low. The microcontroller’s operation limited the ability to generate all four necessary signals. An attempt to work around the timing of the flipping in the circuit created problems in the memory and output waveform. At a lost to create a working signal with the microcontroller, the next step called for using a couple of function generators and comparator to create a control signal in order to test the hardware.

The LM339 comparator was used along with three function generators. An attempt to obtain two outputs on a single function generator to minimize the required hardware failed. As a result, two function generators were used. The first attempt to compare the triangle and sinusoidal signals produced no output because the output did not have a pull up resistor. After discovering the missing resistor, the signals' conditions to produce high and low side outputs were tested one by one. The only problem witnessed was a doubling of pulses at some unexpected points and gaps in the pulse width modulated signal. The double pulsing could be from the noise in the circuit due to the grounded negative rail and ground reference. In order to remedy this problem, a negative rail of -15Vdc was used instead of ground. Fortunately, changing the negative rail fixed the double pulsing problem. For the other problem, the gaps in the PWM were due to the sinusoid waveform having larger amplitude than the triangle and both signals having negative components. Lowering the amplitude and using a negative rail fixed those problems as well. A third function generator is used for the low side control signals to generate a sinusoidal without an offset that the other sine wave has, so it could be compared with zero volts.

The analog controller is tested by monitoring the sinusoidal and triangular waves produced by the function generators. The waveforms are captured on the oscilloscope and are displayed in figure 6-1.

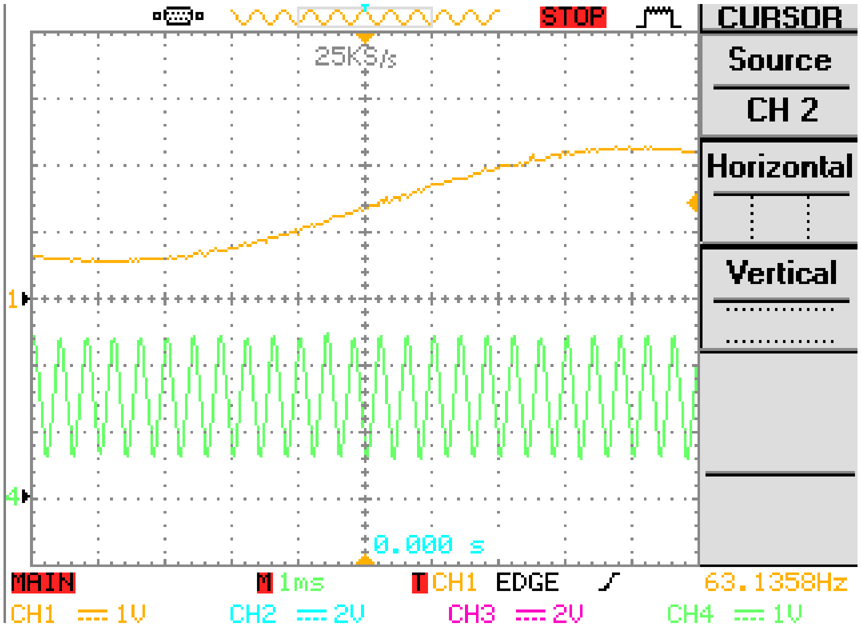


Figure 6-1: Vsine (*f*=60 Hz) and Vtri (*f*=2500 Hz) Waveforms

The waveforms are connected to the comparators to produce the high frequency PWM switching signals, which will drive the high side pair of switches. The resulting waveforms are shown in figure 6-2 below.

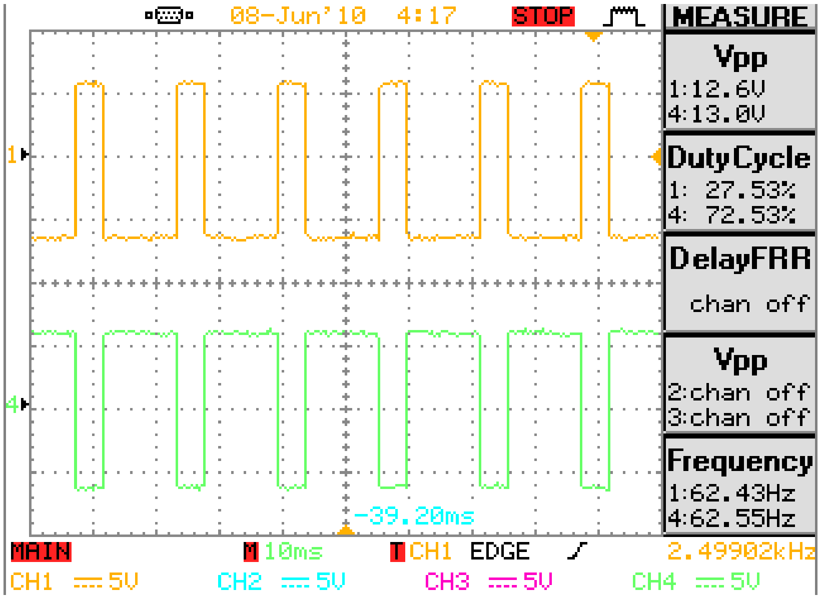


Figure 6-2: High Frequency Switching Signals

Afterwards, the low frequency signals are produced by using a zero crossing method of comparing the sine wave traveling at 60 Hz with ground as the voltage reference. The resulting pulses are captured on the oscilloscope and displayed in figure 6-3.

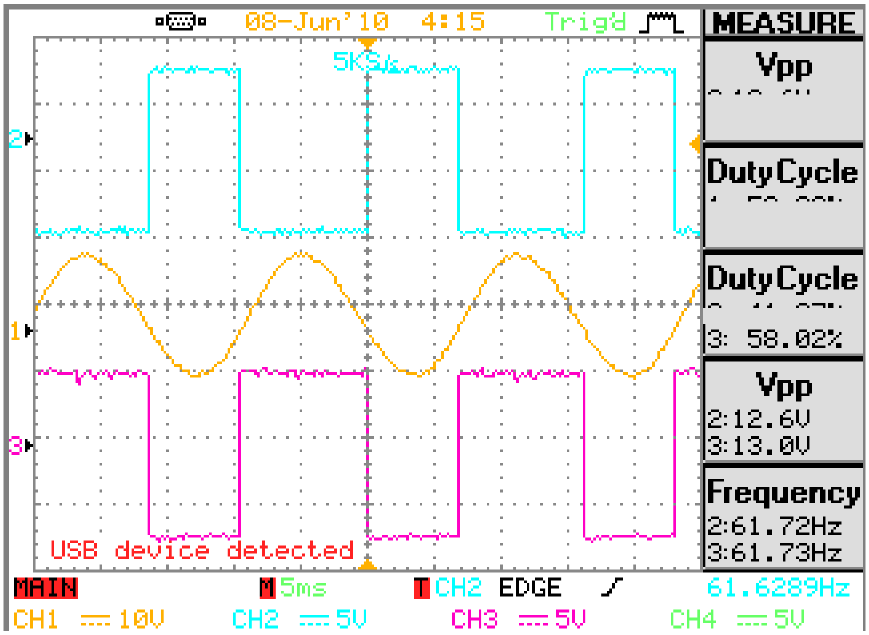
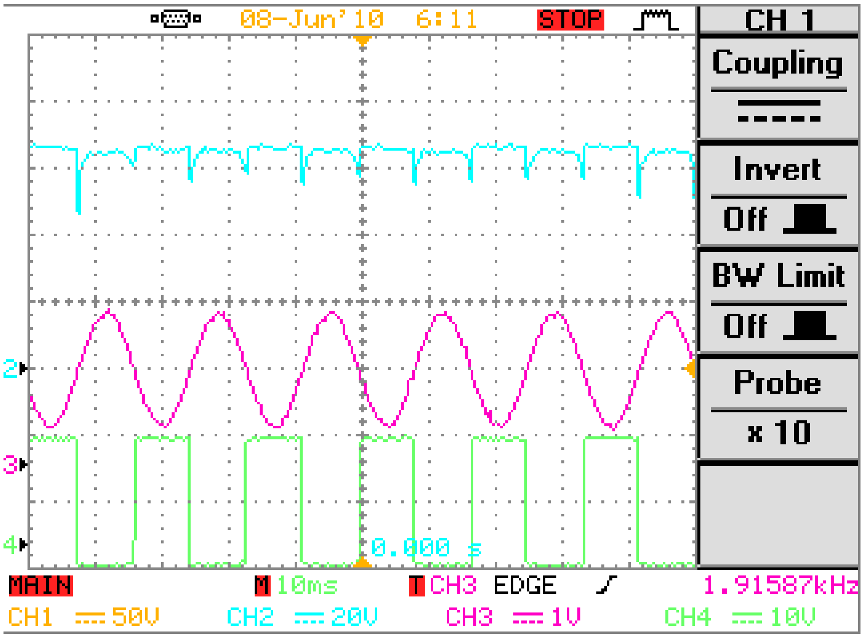


Figure 6-3: Low Frequency Switching Signals

Once all the switching signals have been obtained, the controller is then connected to the main part of the hardware which are the high and low side drivers that drive the pairs of switches. The overall circuit is connected to the MPJA DC Regulated Power Supply and a power resistor decade box as a load. The power source is turned on to provide an input to see if the inverter is operational. Although everything is connected together, the output waveform is not as desired. Figure 6-4 shows the resulting waveform.



CH 1
Coupling
Invert
Off
BW Limit
Off
Probe
x 10

MAIN M 10ms T CH3 EDGE 1.91587kHz
CH1 50V CH2 20V CH3 1V CH4 10V

Figure 6-4: Output Waveform (Ch. 2) with Respect to Switching Signals

The resulting waveform produces a 67.2 V output with an input of 90.1 V, which is not expected. Also, the output voltage stays at a high state without switching to a low state as displayed in the simulation plot of figure 2-2. Therefore, our inverter fails to operate as designed.

VII. CONCLUSIONS

This report explores the design and operation of a high low unipolar inverter. The inverter was first simulated for circuit functionality for our necessary application. This design is based from Wibawa's 500 W solar power inverter design [2]. However, our inverter is designed for the 100 W range. The high low unipolar topology allows high and low frequency switches to be used, which enables easier filtering since the harmonics are located in the higher frequency range.

After confirming its operation through SPICE simulation, the overall hardware was built, followed by the control software being programmed. Each component was soldered and connected onto a prototype board. However, the control program was a more challenging task. A microchip programmer with its respective software was needed. Also, the program had to be coded such that our complex switching scheme is achieved. By utilizing the PIC datasheet, some form of PWM was produced. However, our particular scheme of high frequency and low frequency control signals was much more difficult. After many attempts and hours spent debugging and troubleshooting, we decided to dismiss the digital control last minute and switch to an analog control approach in order achieve some kind of output.

Nevertheless, function generators and comparators were used to create the control signals needed for the drivers and switches. It was not easy achieving the necessary signals, but after some troubleshooting, we finally made the analog control scheme operational. After verifying the control signals, the controller is connected to the overall circuit via the drivers. Then, the whole test setup is connected using a DC power supply as the input and connecting the circuit to the power resistor decade box as the load. The inverter is fired up but we were not outputting the correct waveform. After some troubleshooting and some burned out ICs, we concluded that the drivers were at fault. The low side signals were being received at the input of the drivers, but the low side output was not correct. It remained at a high output so the switches were not operating as expected. As seen in figure 6-4, the output waveform tried to switch to low, but the output remained high. Therefore, our inverter failed to operate correctly.

Although this project did not function as expected, we should have taken a different approach and planned more carefully. We knew we were not very well verse with our programming skills, so we should have had a back up plan with an analog controller. We wasted a lot of time trying to debug because we thought we could get the digital controller working since we wanted to challenge ourselves and provide a compact module. However, we underestimated the challenge and did not have enough time in the end. Also, even though the control signals for the drivers are necessary, we should have tested the drivers earlier as well. Because of these shortcomings, we failed to get our inverter design to function. Nevertheless, we will learn from these mistakes and apply them to our future endeavors.

VIII. BIBLIOGRAPHY

[1]. Dr. Taufik. *Introduction to Power Electronics 7th Revision*. San Luis Obispo, CA: Taufik, 2009. Print.

[2] Chou, Wibawa T. “Build an Efficient 500-W Solar-Power Inverter Using IGBTs.” *Electronic Design*. 13 Aug. 2009. Web. 3 Feb. 26. <http://electronicdesign.com/article/design-solutions/build-an-efficient-500-w-solar-power-inverter-usin.aspx>.

[3] International Rectifier. "Insulated Gate Bipolar Transistor with Ultrafast Soft Recovery Diode IRGP4062DPbF." 2009. Web. <http://www.irf.com/product-info/datasheets/data/irgb4062dpbf.pdf>.

[4] International Rectifier. "Insulated Gate Bipolar Transistor with Ultrafast Soft Recovery Diode IRGB4062DPbF." 2009. Web. <http://www.irf.com/product-info/datasheets/data/irgb4062dpbf.pdf>.

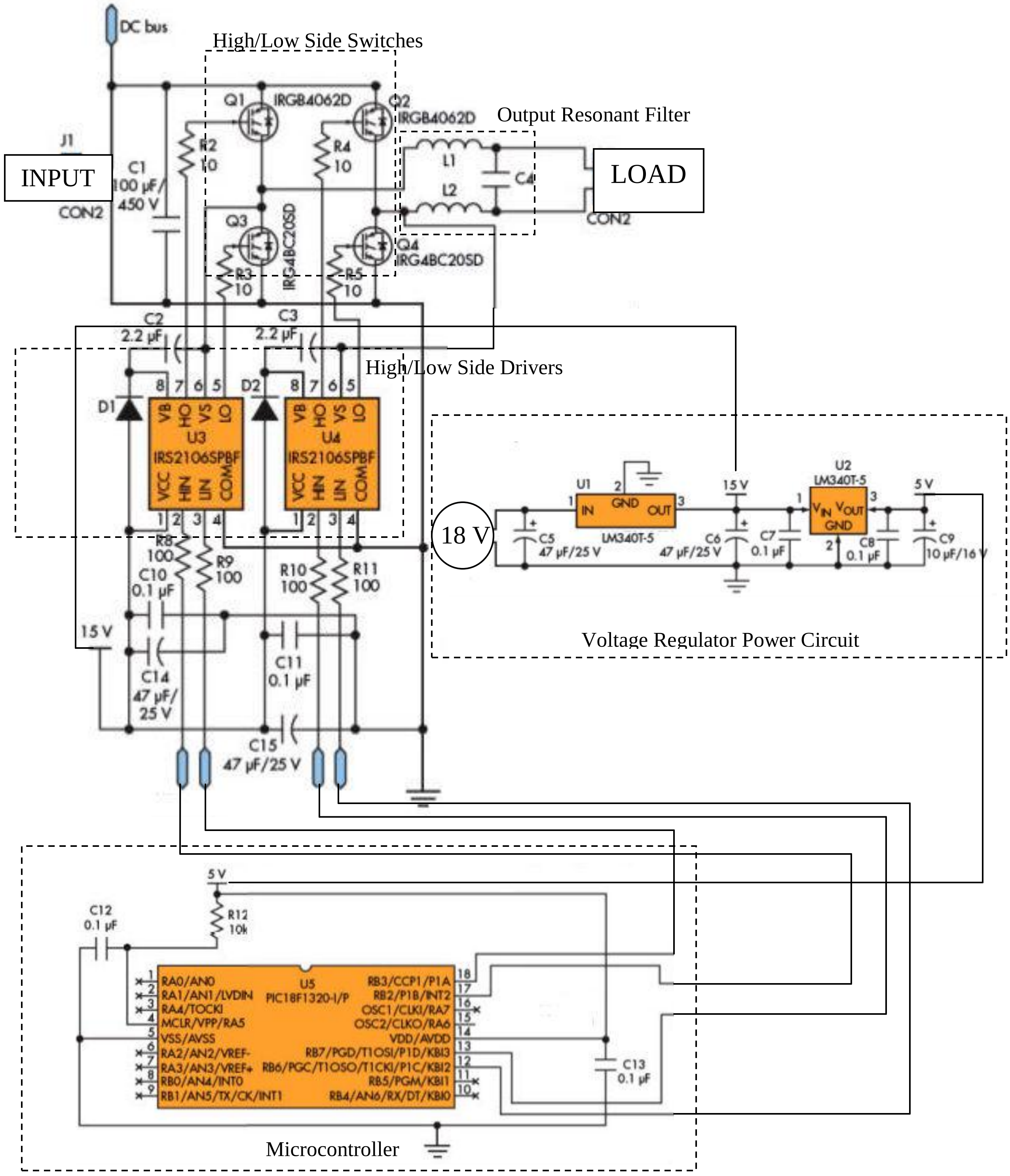
[5] International Rectifier. "High and Low Side Driver IRS2106/ IRS21064(S)PbF." 2009. Web. <http://www.irf.com/product-info/datasheets/data/irs2106.pdf>.

[6] National Semiconductor. “LM340/ LM78XX Series 3-Terminal Positive Regulators.” 2009. Web. <http://www.national.com/ds/LM/LM340.pdf>.

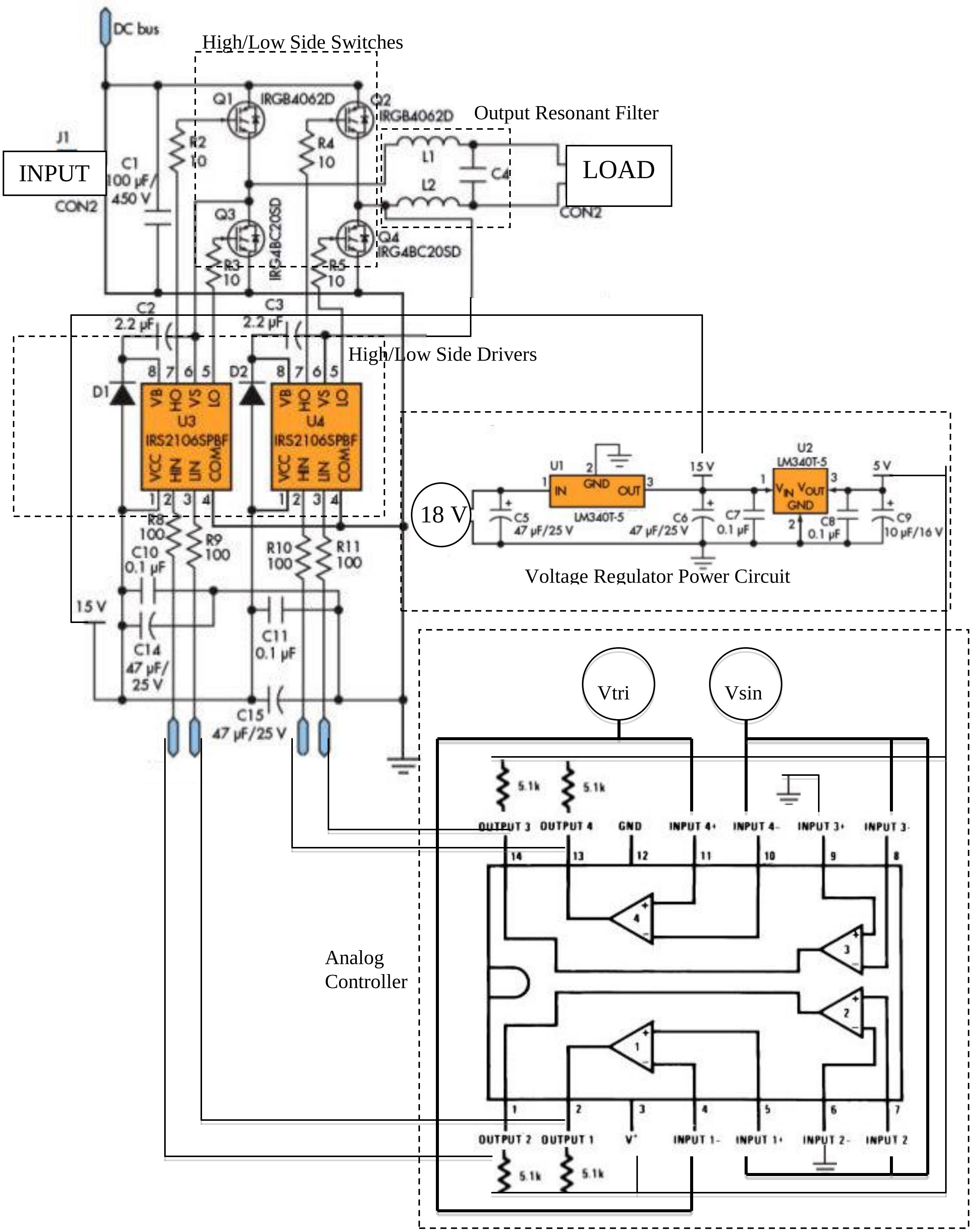
[7] National Semiconductor. “LM139/ LM239/ LM339/ LM2901/ LM3302 Low Power Low Offset Voltage Quad Comparators.” 2010. Web. <http:// *www.national.com/ds/LM/**LM339**.pdf*>

[8] Microchip. “PIC18F1220/1320 Data Sheet 18/20/28-Pin High Performance, Enhanced Flash Microcontrollers with 10-bit A/D and nanoWatt technology” 2010. Web. <*ww1.microchip.com/downloads/en/devicedoc/39605c.pdf*>

APPENDIX A - Schematic and Component Location with Microcontroller



APPENDIX B - Schematic and Component Location with Analog Controller



APPENDIX C – Parts List and Costs

Parts List	Quantity	Cost [$]
100 Ω Resistors	10	0.21
10 Ω Resistors	10	0.41
75 KΩ Resistors	4	0.71
10 KΩ Resistors	3	0.53
47 uF Electrolytic Capacitor	8	1.36
0.1 uF Metal Film Capacitor	10	0.83
10 uF Electrolytic Capacitor	2	0.17
100 uF Electrolytic Capacitor	2	0.63
2.2 uF Electrolytic Capacitor	5	0.07
2.2 uF Metal Film Capacitor	2	1.82
IRS2106 High and Low Side Drivers	4	7.76
IRG4BC20SD IGBTs	4	8.64
IRGB4062DPbF IGBTs	4	15.12
LM340T-5 Voltage Regulator	4	3.16
LM339N Comparator	3	1.5
Microcontroller Programmer	1	39.95
Diodes	4	0.4
Prototype Board	2	3.98
1" Standoffs	1	1.99
Total Price		89.24

APPENDIX C - C code for PIC18F1320 Microcontroller

Closest attempt at working code

```
#define USE_OR_MASKS
#include <p18cxxx.h>
#include "pwm.h"
#include "timers.h"



//------------------------------Configuration setting -------------------------------------------
                -
/**
                * Oscillator is configured as HS
                * Fail safe monitor is enabled
                * watch dog timer is disabled
                * Extended instruction mode is disabled
                * oscillator switch over is enabled
*/
#if defined(__18F1320)                  //If the selected device if PIC18F4685, then
                apply below settings else user will have to set
#pragma config OSC=HS, FSCM=ON, WDT=OFF, IESO=ON, LVP=OFF
#endif

void main(void)
{
  char period=0x00;
  unsigned char outputconfig=0,outputmode=0,config=0;
  unsigned int duty_cycle=0;
  unsigned int timer=0;
  unsigned int i=0;

  unsigned int dutyCycle_Array[40]={0x005B, 0x0053, 0x004E, 0x0048, 0x0044,
                0x0040, 0x003D, 0x003C,  0x003D, 0X0040, 0x0044, 0x0048, 0x004E,
                0x0055, 0x005D, 0x0065,  0x006E, 0x0078, 0x0080, 0x0089, 0x0093,
                0x009B, 0x00A3, 0x00A7, 0X00AF, 0X00B2, 0X00B5, 0X00B7, 0X00B7,
                0X00B5, 0X00B4, 0X00B1,  0X00AB, 0X00A6, 0X009E, 0X0097, 0X008E,
                0X0086, 0X007E};

                OpenTimer2(T2_PS_1_4);


//-----------------Configure pwm --------------------------------------
```

```
                period = 0x3C;
                OpenPWM1(period);                         //Configure PWM module and
                initialize PWM period

while(1)                                                  //observe output on CCP1 pin
{
//---------------------set duty cycle-------------------------------------------------------
                for(i=0;i<40;i++)
                {
                     SetDCPWM1(dutyCycle_Array[i]);    //set the duty cycle

//------------------set pwm output---------------------------------------------------------.
                     outputconfig = FULL_OUT_FWD;
                     outputmode = PWM_MODE_1;
                     SetOutputPWM1( outputconfig, outputmode);        //output PWM in
                respective modes

                     outputconfig = FULL_OUT_REV;
                     outputmode = PWM_MODE_3;
                     SetOutputPWM1( outputconfig, outputmode);

                }
}

//-------------------------close pwm---------------------------------------
  ClosePWM1();

}
```

Final attempt using 100% duty cycles

```
#define USE_OR_MASKS
#include <p18cxxx.h>
#include "pwm.h"
#include "timers.h"


//------------------------------Configuration setting -------------------------------------------
                -
/**
                * Oscillator is configured as HS
                * Fail safe monitor is enabled
```

```
            * watch dog timer is disabled
            * Extended instruction mode is disabled
            * oscillator switch over is enabled
*/
#if defined(__18F1320)              //If the selected device if PIC18F4685, then
            apply below settings else user will have to set
#pragma config OSC=HS, FSCM=ON, WDT=OFF, IESO=ON, LVP=OFF
#endif

void main(void)
{
 char period=0x00;
 unsigned char outputconfig=0,outputmode=0,config=0;
 unsigned int duty_cycle=0;
 unsigned int timer=0;
 unsigned int i=0;

 unsigned int dutyCycle1_Array[40]={0x005B, 0x0053, 0x004E, 0x0048, 0x0044,
            0x0040, 0x003D, 0x003C, 0x003D, 0X0040, 0x0044, 0x0048, 0x004E,
            0x0055, 0x005D, 0x0065, 0x006E, 0x0078, 0x0080, 0x0089, 0x0093,
            0x009B, 0x00A3, 0x00A7, 0X00AF, 0X00B2, 0X00B5, 0X00B7, 0X00B7,
            0X00B5, 0X00B4, 0X00B1, 0X00AB, 0X00A6, 0X009E, 0X0097, 0X008E,
            0X0086, 0X007E};

 unsigned int
            dutyCycle2_Array[15]={0x009D,0x00A4,0x00A9,0x00AF,0x00B2,0x00B6
            ,0x00B7,0x00B7,0x00B5, 0x00B3,0x00AE};

 overlay unsigned char
            period1_Array[15]={0x15,0x13,0x12,0x11,0x10,0x0F,0x0E,0x0E,0X0E,0
            X0F,0x10,0x11,0x12};

 overlay unsigned char
            period2_Array[15]={0x26,0x28,0x29,0x2A,0x2B,0x2C,0x2C,0x2C,0x2C,0x
            2B,0x2A,0x29,0x27};

            OpenTimer2(T2_PS_1_4);


//------------------Configure pwm ----------------------------------------
            period = 0x3C;
            OpenPWM1(period);                   //Configure PWM module and
            initialize PWM period
while(1)                                          //observe output on CCP1 pin
{
//---------------------set duty cycle---------------------------------------------------------
            for(i=0;i<40;i++)
            {
                period = period1_Array[i];
                OpenPWM1(period);

                SetDCPWM1(dutyCycle1_Array[i]);  //set the duty cycle

//------------------set pwm output---------------------------------------------------------.
                outputconfig = FULL_OUT_FWD;
                outputmode = PWM_MODE_1;
                SetOutputPWM1( outputconfig, outputmode);        //output PWM in
            respective modes

                period = period2_Array[i];
                OpenPWM1(period);

                SetDCPWM1(dutyCycle2_Array[i]);

                outputconfig = FULL_OUT_REV;
                outputmode = PWM_MODE_3;
                SetOutputPWM1( outputconfig, outputmode);

            }
}

//-------------------------close pwm---------------------------------------
 ClosePWM1();

}
```