Microcontroller Design of a Bidirectional Three-Level Pulse Width Modulation AC/DC Converter for Vehicle-to-Grid Application

A Thesis

Presented to the

Electrical Engineering Department

California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science in Electrical Engineering

By

Joshua Jonn Aquino

Summer 2012

© 2012

Joshua Jonn Aquino

ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: Microcontroller Design of a Bidirectional Three-Level Pulse Width Modulation
AC/DC Converter for Vehicle-to-Grid Application

AUTHOR: Joshua Jonn Aquino

DATE SUBMITTED: August 2012

COMMITTEE CHAIR: Dr. Dale Dolan, Assistant Professor

COMMITTEE MEMBER: Dr. William Ahlgren, Associate Professor

COMMITTEE MEMBER: Dr. Bridget Benson, Associate Professor

Abstract

Microcontroller Design of a Bidirectional Three-Level Pulse Width Modulation AC/DC Converter for Vehicle-to-Grid Application

Joshua Jonn Aquino

Bidirectional chargers provide plug-in electric vehicles (EV) the ability to not only transfer energy from the grid to the vehicle, but also transfer energy from the vehicle to the grid (V2G). The V2G mode allows power utility companies to offset peak power consumption thus allowing household consumers and industry corporations to save money on their electricity bills. This paper describes the design choices of bidirectional chargers and provides the implementation details for a Three-Level PWM AC-DC based charger. Although the three-level PWM AC-DC design requires a more complex controller than its counterparts, Matlab Simulink simulations show the design provides lower total harmonic distortion (THD) and reduced stress to the charger's components making it a viable charger design for the electric vehicle industry.

Acknowledgment

I would like to thank my committee chair and members for giving me the opportunity to present this thesis. I also would like to extend my gratitude to my fellow colleagues, Chris Grasberger, Matt King, Matt Bloom, Ryan White, Sandy Nguyen, Anuranga Gunaratne, Ruth Huang, and Gersewak Rai for all the help.

Table of Contents

List of Tables.....	ix
List of Figures.....	x
Chapter I: Introduction.....	1
A. Need for Bidirectional V2G Chargers.....	1
B. Load Demand and V2G Overview.....	2
C. V2G Advantages.....	3
D. Thesis Objectives.....	4
Chapter II: Background.....	5
A. Charger Type Classifications.....	5
1. Level 1 Charger.....	5
2. Level 2 Charger.....	9
3. Level 3 Charger.....	12
B. Overall Bidirectional Topology.....	14
1. Filter Stage.....	15
2. AC-DC Bidirectional Inverter/Rectifier Stage.....	16
a. Half-Bridge PWM.....	16
b. Full-Bridge PWM.....	18
c. Three-Level PWM AC-DC Bidirectional Converter.....	20
3. DC-DC Bidirectional Converter Stage.....	26
a. Dual Active Bridge.....	26
b. 2-Quadrant Buck-Boost.....	27

Chapter III: Simulations.....	29
A. MatLab PowerGUI Simulink Procedure.....	29
B. Simulations Results.....	33
1. Half-Bridge and Full-Bridge PWM Simulation Results.....	33
2. Three-Level PWM Simulations Results.....	36
Chapter IV: Closed Loop Microcontroller Hardware Prototype Design.....	41
A. Overview of the Hardware Design.....	41
1. Sense Stage [V_s , I_s , V_{c1} , V_{c2}].....	42
a. Grid Voltage Sense [V_s].....	42
b. Grid Current Sense [I_s].....	46
c. DC Bus Voltage Capacitors Voltage Sense Stages [V_{c1}/V_{c2}]...49	
2. Control Stage.....	52
B. Hardware Test.....	54
1. Eagle PCB Layout Design Procedure.....	54
a. Package Creation.....	54
b. Schematic and Layout.....	61
c. Gerber File Generation.....	65
2. PCB Test Board.....	68
3. Equipment List.....	72
4. Results.....	73
a. Grid Voltage Sense (V_s) Results.....	73
b. Grid Current Sense (I_s) Results.....	75
c. DC Bus Capacitors Voltage Sense (V_{C1}/V_{C2}) Results.....	76

d. Control stage (Arduino Microcontroller) Results.....	77
Chapter V: Conclusion.....	79
References.....	81
Appendix.....	83

List of Tables

Table 1: Three-Level PWM Switch Logic.....	22
Table 2: Closed Loop Control Logic.....	25
Table 3: Summary of Results.....	40
Table 4: Necessary Gerber and drill files for board fabrication.....	68
Table 5: Arduino Control Stage Results.....	78

List of Figures

Figure 1.1: Load Demand Curve.....	2
Figure 1: Tesla Level-1 Charger Connector.....	6
Figure 2: Nissan Level-1 Charger Connector.....	7
Figure 3: Chevrolet Level-1 Charger Connector.....	7
Figure 4: Tesla Level-2 Charger Connector.....	9
Figure 5: Nissan Level-2 Charger Connector.....	10
Figure 6: Chevrolet Level-2 Charger Connector.....	10
Figure 7: Nissan Leaf Charge Ports.....	12
Figure 8: Nissan Level 3 Charger.....	13
Figure 9: Overall Bidirectional Charger Topology.....	14
Figure 10: LCL Filter.....	15
Figure 11: Half-Bridge PWM Schematic.....	16
Figure 12: Half-Bridge PWM Charge Mode Waveform.....	17
Figure 13: Half-Bridge PWM V2G Mode Waveform.....	18
Figure 14: Full-Bridge PWM Schematic... ..	18
Figure 15: Full-Bridge PWM Charge Mode Waveform	19
Figure 16: Full-Bridge PWM V2G Mode Waveform	19
Figure 17: Three-Level PWM Schematic.....	21
Figure 18: Three-Level PWM Charge Mode Waveform	21
Figure 19: Three-Level PWM V2G Mode Waveform	22

Figure 20: Mode 1 (Left), Mode 2 (Right)	23
Figure 21: Mode 3 (Left), Mode 4 (Right).....	23
Figure 22: Mode 5 (Left), Mode 6 (Right).....	24
Figure 23: Closed Loop Control System.....	24
Figure 24: Dual Active Bridge DC-DC Converter.....	27
Figure 25: 2-Quadrant Buck-Boost Converter.....	28
Figure 26: Simulink Library Browser.....	30
Figure 27: PowerGUI options.....	31
Figure 28: PowerGUI parameter configurations.....	31
Figure 29: PowerGUI FFT Analysis Tool window.....	32
Figure 30: Matlab Simulink Half-Bridge Schematic.....	33
Figure 31: MatLab Simulink Full-Bridge Schematic.....	33
Figure 32: Half-Bridge Vin (Yellow) and Vout (Purple) Waveforms.....	34
Figure 33: Full-Bridge Vin (Yellow) and Vout (Purple) Waveforms.....	34
Figure 34: Half-Bridge FFT and THD Analysis.....	35
Figure 35: Full-Bridge FFT and THD Analysis.....	36
Figure 36: ESS MatLab Simulink Parameters.....	37
Figure 37: ESS Battery Discharge Characteristics.....	37
Figure 38: Three-Level PWM Closed Loop Control Schematic.....	38
Figure 39: Three-Level PWM Discharge Mode Schematic.....	38
Figure 40: Three-Level PWM V2G Mode Waveforms.....	39
Figure 41: Three-Level PWM FFT and THD Analysis.....	40
Figure 42: Arduino Microcontroller Board.....	41

Figure 43: Differential Amplifier Schematic.....	43
Figure 44: Vs ADC Mapping.....	44
Figure 45: Differential Amplifier for Grid Voltage Sense Spice Simulation Schematic..	45
Figure 46: Differential Amplifier for Grid Voltage Sense Spice Simulation Input (Blue) and Output (Green) Waveforms.....	45
Figure 47: AD623 Instrumentation Amplifier Schematic and Pin Assignment.....	47
Figure 48: Is ADC Mapping.....	47
Figure 49: Instrumentation Amplifier for Grid Current Sense Spice Simulation Schematic.....	48
Figure 50: Instrumentation Amplifier for Grid Current Sense Spice Simulation Input (Blue) and Output (Green) Waveforms.....	48
Figure 51: VC1/VC2 ADC Mapping.....	50
Figure 52: Differential Amplifier for VC1/VC2 Voltage Sense Spice Simulation Schematic.....	51
Figure 53: Differential Amplifier for VC1/VC2 Voltage Sense Spice Simulation Input (Blue) and Output (Green) Waveforms.....	51
Figure 54: Vs ADC Conversion.....	52
Figure 55: Is ADC Conversion.....	53
Figure 56: Vc1/Vc2 ADC Conversion.....	53
Figure 57: Cadsoft Eagle main window – starting a custom library.....	55
Figure 58: Cadsoft Eagle library window.....	56
Figure 59: Package creation work space.....	57
Figure 60: SMD and through-hole size/dimension options, respectively.....	57

Figure 61: Schematic symbol creation work space.....	59
Figure 62: Device creation work space.....	60
Figure 63: Schematic and package symbol connection window.....	60
Figure 64: Starting a new project in Eagle.....	61
Figure 65: Starting a new schematic in the project directory.....	61
Figure 66: Schematic workspace.....	63
Figure 67: Layout Workspace.....	64
Figure 68: CAM processor window.....	66
Figure 69: Gerber file configuration for a 2-layer board job.....	67
Figure 70: Eagle PCB Test Board Schematic.....	69
Figure 71: Eagle PCB Test Board Layout.....	70
Figure 72: Eagle PCB Test Board with the Arduino Microcontroller Control Stage.....	71
Figure 73: Digikey BOM.....	72
Figure 74: PCB Test Board Setup.....	73
Figure 75: Differential Amplifier for Grid Voltage Sense input waveforms where Channel 1 - V_a (orange), Channel 2 - V_b (green), and Math Channel ($V_a - V_b$) (purple).....	74
Figure 76: Differential Amplifier for grid voltage sense output waveform.....	75
Figure 77: Instrumentation Amplifier for grid current sense output waveform.....	76
Figure 78: Differential Amplifier for VC1/VC2 voltage sense input (orange) and output (green) waveforms where input PWM Duty Cycle is set at 50%.....	77

I. INTRODUCTION

A. *Need for Bidirectional V2G Chargers*

According to the international energy outlook report, by 2030, the transportation sector will increase its share in the world's total oil consumption by 55% [1]. This pushes the advancement of transportation technologies that reduce oil consumption. Such advancement is the development of the plug-in electric vehicles (EV). Recently, a number of these EVs have paved their way to the consumer market [2-4]. These EVs use their electric power train as their propulsion system unlike conventional hybrid electric vehicles (HEV) which rely on an internal combustion engine (ICE). EVs therefore carry a larger energy capacity battery pack (more than 4kWh) than HEVs [5]. Nowadays, to charge the vehicles' battery packs, a conventional unidirectional charger is used to convert the AC grid voltage to a specific DC voltage. The integration of EVs into the grid, especially at the residential distribution level, is significant considering the power ratings (more than 1kW) of the unidirectional chargers [5]. As these vehicles become more prominent, the power demand required to charge their batteries will also increase. This increase will contribute to transmission/distribution losses as well as grid stability issues. However, a bidirectional V2G charger could address these issues. It could support the grid by regulating the reactive power [5]. It could also send power back to the grid to alleviate grid consumption and promote peak shaving [5].

B. Load Demand and V2G Overview

Load demand is not constant throughout a typical day [6]. In a typical 24 hour load profile, peak energy consumption is observed around 1:00pm and 11:00pm. At 1:00pm, commercial loads consume around 3MW while household loads consume around 9MW. Around 11:00pm, 4MW and 11MW are consumed by commercial and household loads respectively. Energy consumption is at its lowest around 5:00am. Around this time, commercial loads consume 2MW while household loads consume 8MW. Figure 1.1 shows the aforementioned load demand curve.

Figure 1.1: Load Demand Curve [6]

V2G proposes to smoothen out stressful load demands within the grid, especially in time slots when grid power is at its peak. It also provides ancillary service to alleviate grid consumption. V2G utilizes EVs as an energy source during peak time, and when the grid is of peak, the batteries of the EVs are charged [7].

C. V2G Advantages

Bidirectional V2G chargers are not suited for providing base-load power [8]. However, they are suitable for peak shaving application [8]. For this application, the EVs act as energy storage units. Energy is stored in the EVs' batteries during night time when the cost of electricity is at its lowest. During peak hours, when the EVs are parked and unused, energy is drawn from them to put power back to the grid [8]. With this scheme, EV owners could produce revenue [9]. They charge their cars while electricity is cheap, and then use them to put power back to the grid during peak hours. Utilities could also benefit from this by having increased system flexibility. They could use this as energy storage for intermittent renewable energy sources such as wind and solar [9]. V2G chargers are also suitable for regulating frequency fluctuations caused by system imbalances [8]. Batteries used in EVs have very fast response capabilities. This capability makes EVs a strong candidate for the provision of frequency regulation services. Through bidirectional V2G chargers, regulation is made possible by allowing the grid to absorb/release small quantities of energy from/to the batteries [8].

D. Thesis Objectives/Organization

This thesis describes the design choices of bidirectional chargers and provides the implementation details for a Three-Level PWM AC-DC based charger. Although the three-level PWM AC-DC converter design requires a more complex controller than its counterparts, Matlab Simulink simulations show the design provides lower total harmonic distortion (THD) and reduced stress to the charger's components making it a viable charger design for the electric vehicle industry.

This paper is organized as follows. Section I is the introduction. Section II provides a description of different charger type classifications and design choices for the overall bidirectional charger topology. Section III provides simulation results comparing three different AC-DC bidirectional charger designs that illustrate the three-level PWM AC-DC based design provides superior harmonic content to its counterparts. Section IV describes a microcontroller based prototype implementation of the closed loop control system required for the Three-level PWM AC-DC bidirectional charger. Section V is the conclusion.

II. BACKGROUND

A. *Charger Type Classifications*

The following are brief descriptions of the three classifications of EV Chargers [10]. For the simplicity of designing a control system prototype, a level 1 charger will be implemented. Also, to maximize the cycle-life of lithium-ion batteries, it is not recommended to fully charge/discharge the batteries at a constant voltage and current [11]. For this thesis, simple calculations were made to determine the charge time of different charger level types where the batteries are charged from 20% to 85%.

1) Level 1 Charger

The 120VAC grounded outlet supplies energy to the vehicle's on-board charger. A separate connector links the grid connected supply to the on-board charger. The charger could supply a maximum of 1.92kW at 16A [12]. This is the most cost effective charger as it doesn't need any additional equipment to charge the vehicle. But the low current rating of this charger results the longest charge time. A level 1 charger connector, just the unit alone, costs around \$600.

For example, the Tesla Motors Roadster has a level-1 charger system that could output 1.5kW [2]. With this, charging the Tesla Roadster that has a 56kWh capacity battery pack (244 epa EV mile range) will take about 29 hours to charge from a 20% state-of-charge to 85% state-of-charge (SOC).

Figure 1: Tesla Level-1 Charger Connector

The Nissan Leaf, as another example, utilizes a 1.4kW level-1 charger system [3]. With a Nissan Leaf, that has 24kWh capacity battery (73 epa EV mile range), it will take about 13 hours to charge from a 20% to 85% SOC.

Figure 2: Nissan Level-1 Charger Connector

The last example is from Chevrolet with their 1.4kW level-1 charger system [4]. Charging the Chevrolet Volt, which has 16kWh capacity battery (35 epa EV mile range) will take around 9 hours to charge it from 20% to 85% SOC.

Figure 3: Chevrolet Level-1 Charger Connector

The following assumptions and calculations were made:

$$\text{Minimum Discharge Level} = 20\%$$

$$\text{Maximum Charge Level} = 85\%$$

$$\text{Net Capacity} = (85\% - 20\%) * \text{Capacity} = (65\%) * \text{Capacity}$$

$$\text{Assume Charger Loss of } 20\%$$

$$\begin{aligned}\text{Input Capacity to Charger} &= \text{Net Capacity} * (100\% + \text{Charger Loss}) \\ &= \text{Net Capacity} * (120\%) \end{aligned}$$

$$\text{Charge Time} = \frac{\text{Input Capacity to Charger}}{\text{Charger Output Power}}$$

For Tesla Roadster

$$\text{Charge Time} = \frac{43.68kWh}{1.5kW} = 29.12h$$

For Nissan Leaf

$$\text{Charge Time} = \frac{18.72kWh}{1.4kW} = 13.37h$$

For Chevrolet Volt

$$\text{Charge Time} = \frac{12.48kWh}{1.4kW} = 8.91h$$

2) *Level 2 Charger*

Energy is supplied by a 208-240VAC source to the onboard charger of the vehicle. Compared to level-1, the higher voltage source produces higher current supply and power charger output. The higher current and power requires a smarter connector that contains an external ground fault circuit interrupter (GFCI). At 240VAC, the charger could supply a maximum of 19.2kW at 80A [12]. It is a medium performing charger with regards to cost, current supply capability, and charge time. A level-2 charger connector costs around \$2000 just the unit alone.

The Tesla Roadster uses a level-2 16.8kW charger system [2]. With this, the Tesla Roadster will take about 3 hours to charge from a 20% to 85%.

Figure 4: Tesla Level-2 Charger Connector

The Nissan Leaf could be charged with a 3.3kW level-2 charger system [3]. This system will charge the Leaf from 20% to 85% SOC in about 6 hours.

Figure 5: Nissan Level-2 Charger Connector

The last example is from Chevrolet with their 3.3kW level-2 charger system [4]. Charging the Chevrolet Volt will take around 4 hours to charge it from 20% to 85% SOC.

Figure 6: Chevrolet Level-2 Charger Connector

The following assumptions and calculations were made:

$$\text{Minimum Discharge Level} = 20\%$$

$$\text{Maximum Charge Level} = 85\%$$

$$\text{Net Capacity} = (85\% - 20\%) * \text{Capacity} = (65\%) * \text{Capacity}$$

$$\text{Assume Charger Loss of } 20\%$$

$$\begin{aligned}\text{Input Capacity to Charger} &= \text{Net Capacity} * (100\% + \text{Charger Loss}) \\ &= \text{Net Capacity} * (120\%) \end{aligned}$$

$$\text{Charge Time} = \frac{\text{Input Capacity to Charger}}{\text{Charger Output Power}}$$

For Tesla Roadster

$$\text{Charge Time} = \frac{43.68kWh}{16.8kW} = 2.60h$$

For Nissan Leaf

$$\text{Charge Time} = \frac{18.72kWh}{3.3kW} = 5.67h$$

For Chevrolet Volt

$$\text{Charge Time} = \frac{12.48kWh}{3.3kW} = 3.78h$$

3) *Level 3 Charger*

Level-3 charger differs to the level-1 and level-2 chargers because it supplies a DC voltage and the charger is now an external component to the car. Here, energy is supplied by a 300-600VDC source from the off-board charger. A separate level-3 connector is required as seen in Figure 7. This type of charger is not suitable for households as it has high current supply capabilities. The high cost also makes it suitable only for business/commercial use. However, as it results in the shortest charge time, it has applications in fast charging stations for Electric Vehicles.

Figure 7: Nissan Leaf Charge Ports

The Tesla Roadster and Chevrolet Volt do not support level 3 charging. The Nissan leaf on the other hand will take around 25 minutes to charge from 20% to 85% SOC. The Nissan level 3 charger runs at 480VDC, 45kW, 125A max [3]. Level 3 chargers vary on cost depending on the manufacturer. Business owners could buy the Nissan Leaf Level 3

charger (Figure 7) for around \$10,000 to 16,000. Consumers could opt for the optional \$700 level-3 charge port and connector.

Figure 8: Nissan Level 3 Charger

The following assumptions and calculations were made:

$$\text{Minimum Discharge Level} = 20\%$$

$$\text{Maximum Charge Level} = 85\%$$

$$\text{Net Capacity} = (85\% - 20\%) * \text{Capacity} = (65\%) * \text{Capacity}$$

$$\text{Assume Charger Loss of } 20\%$$

$$\begin{aligned} \text{Input Capacity to Charger} &= \text{Net Capacity} * (100\% + \text{Charger Loss}) \\ &= \text{Net Capacity} * (120\%) \end{aligned}$$

$$\text{Charge Time} = \frac{\text{Input Capacity to Charger}}{\text{Charger Output Power}}$$

For Nissan Leaf

$$\text{Charge Time} = \frac{18.72\text{kWh}}{3.3\text{kW}} = 0.416\text{h}$$

B. Overall Bidirectional Topology

A possible implementation of the bidirectional charging system is using two independent unidirectional converters, one for charging mode, and another for V2G mode [13]. But this system has some limitations. Unidirectional V2G regulation and reserves capacities are significantly less than those of a bidirectional V2G [14]. Also, this system is not suitable to be an internal charger in an EV because it requires more components. It is more appropriate to use a bidirectional on-board charger where the power electronics devices are integrated into one unit. This is the most efficient approach because it minimizes manufacturing costs, maintenance costs, and weight of the EV [15]. Figure 9 shows a generic bidirectional EV charger. It consists of a filter, an AC-DC inverter/rectifier stage, and a DC-DC converter stage [16]. In charge mode, the AC grid acts as the source and the Energy Storage System (ESS) acts as the load. In V2G mode, the AC grid acts as a sink while the ESS acts as the source.

Figure 9: Overall Bidirectional Charger Topology

1) Filter Stage

When connecting renewable energy sources to the grid, a grid-friendly interface between the energy source and the utility network is needed. It should also meet the IEEE 1547-2003 standards as cited in the Federal Energy Policy Act of 2005 [17]. The objective of the filter is to attain this grid-friendly interface by filtering out the non-sinusoidal currents delivered to the grid. Filtering out these non-sinusoidal currents is crucial to maintain power quality because non-sinusoidal currents delivered to the grid can introduce an additional non-sinusoidal voltage drop across the line impedance and thereby increase the voltage distortion supplied to the load [18]. For this stage, an LCL filter could be used because it can achieve good attenuation with relatively small component values and reasonable filter costs [18]. It consists of lower inductance values than its counterparts, the L and the LC filters. When the third-order LCL filter is connected between the inverter and the grid, it can effectively reduce the switching frequency ripple created by high switching frequency switches [18]. Figure 10 shows a generic LCL filter.

Figure 10: LCL Filter

2) AC-DC Bidirectional Inverter/Rectifier Stage Purpose and Topology Types

The bidirectional AC-DC stage converter is the main link between the grid and the ESS. In charge mode, this stage acts as a rectifier converting the AC voltage from the grid into a DC voltage for the ESS. In V2G mode, it acts as inverter, converting the DC voltage to an AC voltage to provide power back to the grid. This stage could be implemented with the following topologies: 1) Bidirectional Half-Bridge PWM, 2) Bidirectional Full-Bridge PWM, and 3) Three-Level Bidirectional AC/DC Converter. These converters are shown in Figures 11, 14, and 17 respectively.

a) Half-Bridge PWM

Figure 11: Half-Bridge PWM Schematic

Half-Bridge PWMs are the simplest among the three AC-DC converters since they require the fewest components. It also only needs two switches therefore it has the

simplest control system. However, these converters introduce high stress on the components that could eventually cause failure to the system [16].

In charge mode, the control system switches T1 and T2 open and then the intrinsic diodes of both switches act as a rectifier circuit to convert the AC grid voltage to DC. The half-bridge rectifier waveform is shown in Figure 12.

Figure 12: Half-Bridge PWM Charge Mode Waveform

In V2G mode, the control system complimentary turns T1 and T2 on and off to produce a PWM signal. The controller sets the duty cycle of the inverter output depending on the value of input voltage. If the controller senses a maximum peak voltage from the input, the PWM duty cycle is set to 100%. If the input is sensed at ground or 0V, the duty cycle is set to 50%. If the input is sensed at minimum peak voltage, the duty cycle is set to 0%. The output PWM signal goes from $-V_{pk}/2$ to $V_{pk}/2$ (where V_{pk} is the input peak voltage). This PWM signal provides power back to the grid. The half-bridge inverter PWM waveform is shown in Figure 13.

Figure 13: Half-Bridge PWM V2G Mode Waveform

b) *Full-Bridge PWM*

Figure 14: Full-Bridge PWM Schematic

The Full-Bridge PWM is similar to the Half-Bridge PWM. The difference is that Full-Bridge PWMs require two more switches and one less capacitor. Compared to the half-bridge PWM, the added switches require a more complicated control system but this lowers the stress on the components [16].

In charge mode, just like the half-bridge PWM, all four switches are left open and then the intrinsic diodes of each switch act as a rectifier circuit. The full-bridge rectifier waveform is shown in Figure 15.

Figure 15: Full-Bridge PWM Charge Mode Waveform

In discharge mode, T1&T4 or T2&T3 are alternately turned on and off to produce a PWM signal to simulate a 60Hz sinusoidal wave. Unlike the unipolar PWM output signal produced in the half-bridge PWM, the full-bridge output signal produces a bipolar PWM signal. It produces two separate PWM signals, one in the positive cycle, the other at the negative cycle.

If the controller senses a maximum peak voltage from the input, the PWM duty cycle is set to 100% at the positive cycle. If the input is sensed at minimum peak voltage, the duty cycle is also set to 100% but in the negative cycle. If the input is sensed at ground or 0V, the duty cycle is set to 0%. The output PWM signal goes from 0 to $V_{pk}/2$ at the positive cycle. On the other hand, the PWM signal goes to 0 to $-V_{pk}/2$ at the negative cycle. The full-bridge inverter PWM waveform is shown in Figure 16.

Figure 16: Full-Bridge PWM V2G Mode Waveform

c) *Three-Level PWM AC-DC Bidirectional Converter*

This configuration has the most complex control system of all three converters but has the lowest stress on the components. This also has reduced EMI and acoustic noise. This topology requires a very complex control system but the added complexity significantly lowers the stress level on the components. Compared to the half-bridge PWM, this topology lessens the stress to the switches by a factor of four [19]. Figure 17 shows the schematic of the proposed topology.

Figure 17: Three-Level PWM Schematic

In charge mode, all the switches are left open. Then, the intrinsic diodes act as a full-bridge rectifier circuit. During the positive cycle of the AC grid, the intrinsic diodes of T1, T3, and T6 conduct. While at the negative cycle, the intrinsic diodes of T2, T4, and T5 conduct [19]. The full-bridge rectifier waveform is shown in Figure 18.

Figure 18: Three-Level PWM Charge Mode Waveform

In V2G mode, the topology follows the switch logic shown in Table 1. The PWM voltage signal goes between ($V_{dc-dc}/2$ and V_{dc-dc}) when switching between modes (3 and 1), ($0V$ and $V_{dc-dc}/2$) when switching between modes (6 and 3), ($0V$ and $-V_{dc-dc}/2$) when switching between modes (5 and 4), or ($-V_{dc-dc}/2$ and $-V_{dc-dc}$) when switching between modes (4 and 2) [19]. The inverter output waveform is shown in Figure 19. Figures 20, 21, and 22 illustrate the current flow through the schematic at each operation modes.

Table 1: Three-Level PWM Switch Logic

Operation Mode	T1	T2	T3	T4	T5	T6	Voltage Across the Grid
1	on	off	on	off	off	on	V_{dc-dc}
2	off	on	off	on	on	off	$-V_{dc-dc}$
3	off	off	on	on	off	on	$V_{dc-dc}/2$
4	off	on	on	on	off	off	$-V_{dc-dc}/2$
5	on	on	on	off	off	off	0
6	off	off	off	on	on	on	0

Figure 19: Three-Level PWM V2G Mode Waveform

In operation mode 1, T1, T3, and T6 are switched on to apply [$V_{C1}+V_{C2}$] or [V_{dc-dc}] across the grid. In operation mode 2, T2, T4, and T5 are switched on to apply [$-V_{C1}-V_{C2}$] or [$-V_{dc-dc}$] across the grid.

Figure 20: Operation Mode 1 (Left), Operation Mode 2 (Right) [19]

In operation mode 3, T3, T4, and T6 are switched on to apply $[+V_{C2}]$ or $[V_{dc-dc}/2]$ across the grid. In operation mode 4, T2, T4, and T5 are switched on to apply $[-V_{C1}]$ or $[-V_{dc-dc}/2]$ across the grid.

Figure 21: Operation Mode 3 (Left), Operation Mode 4 (Right) [19]

In operation mode 5, T1, T2, and T3 are switched on to apply 0V across the grid. In operation mode 6, T4, T5, and T6 are switched on to apply 0V across the grid.

The closed loop control system block diagram is shown in Figure 23. Boolean X ensures the inverter's output current is in-phase with the 60Hz sinusoidal current. Boolean Y determines which voltage level the PWM output should operate. Finally, boolean Z determines whether the system should operate at the positive cycle or the negative cycle [20]. With all the three logic value of X, Y, and Z, the operation mode can be determined. Table 2 summarizes the operation modes based on XYZ boolean.

Table 2: Closed Loop Control Logic

X	Y	Z	Operation Mode	X	Y	Z	Operation Mode
0	0	0	6	1	0	0	4
0	0	1	3	1	0	1	5
0	1	0	3	1	1	0	2
0	1	1	1	1	1	1	4

The following formulas demonstrate the boolean XYZ logic.

$$X = 1$$

$$if \left\{ - \left[(VC1 + VC2) + \frac{V_{pk-pk}}{2} \right] * \frac{Vs}{\frac{V_{pk-pk}}{2}} + (VC2 - VC1) \right\} - Is \leq 0$$

$$Y = 1$$

$$if \frac{V_{pk-pk}}{4} \leq |Vs|$$

$$Z = 1$$

$$if Vs < 0$$

Otherwise: X = 0, Y = 0, Z = 0

Where:

V_{pk-pk} = Maximum Grid peak to peak votage

Vs = Grid Voltage Sense

Is = Grid Current Sense

$VC1$ and $VC2$ = DC Voltage Bus Capacitor Voltage Sense Sense

3) DC-DC Bidirectional Converter Stage

In discharge mode, the AC-DC converter stage alone would have power factor issues. To address this, a secondary stage DC-DC converter is required. This also helps the AC-DC stage's DC-bus voltage match with the battery voltage [16]. The second-stage could be implemented with the following: 1) Dual Active Bridge, and 2) Two-Quadrant Buck-Boost.

a) Dual Active Bridge

This converter allows fast control and high power density but requires a high number of components. It consists of two H-bridge inverters linked by a transformer. This transformer is set to a ratio suitable for the application. It also needs eight switches with internal diodes. Since it has eight switches, this converter requires additional ZVS and ZCS to reduce switching losses [16]. This topology is shown in Figure 24.

In charge mode, the left H-bridge converter acts as an inverter. This converts the DC voltage from the AC-DC stage to an AC voltage. This AC voltage is passed through the transformer with a fixed ratio before it goes to the right H-bridge converter. The internal diodes of the right H-bridge then act as a rectifier circuit to convert the transformed AC voltage to a DC voltage for the battery [16].

In discharge mode, the right H-bridge converter acts as an inverter. This converts the DC voltage from the battery to an AC voltage. This AC voltage is passed through the transformer with a fixed ratio before it goes to the left H-bridge converter. The internal diodes of the left H-bridge then act as a rectifier circuit to convert the transformed AC voltage to a DC voltage for the AC-DC Converter stage [16].

Figure 24: Dual Active Bridge DC-DC Converter

b) 2-Quadrant Buck-Boost

This converter, as seen in Figure 25, has a simpler design and requires fewer components than the dual active bridge configuration. It needs a capacitor, and two switches with internal diodes. However it needs two bulky power inductors which tend to be expensive [16].

This converter acts as a buck converter in one direction and a boost converter in the other. In charge mode, the circuit acts as a buck converter where the DC voltage from the

AC-DC Converter stage is stepped down for the battery. T2 is left open, while T1 is controlled with a PWM signal. In discharge mode, the circuit acts a boost converter where the battery DC voltage is stepped up for the AC-DC Converter stage. T1 is left open and T2 is controlled with a PWM signal. The 2-Quadrant DC-DC converter is preferable than the Dual-Active Bridge DC-DC converter because it requires less components. One limitation of the 2-Quadrant converter is that it could only act as a boost converter in one direction and a buck converter in the other. But for this application, this limitation is not a problem [16].

Figure 25: 2-Quadrant Buck-Boost Converter

III. SIMULATIONS

This paper used MatLab Simulink to simulate the V2G mode inverter output waveforms. It also provides Fourier transform (FFT) and total harmonic distortion THD analysis. These simulations are used to compare the performance of the three different bidirectional AC-DC stage topologies.

A. *MatLab PowerGUI Simulink Procedure*

MatLab Simulink contains a script called PowerGUI that does FFT analysis of circuits. This thesis uses this script to analyze the THD of the output signals of different inverter topologies. The following describes the steps to use the script.

First, run the Simulink schematic. In the “Simulink Library Browser” window (Figure 26), search for “PowerGUI”. Drag the PowerGUI icon to the Simulink schematic.

Figure 26: Simulink Library Browser

In the Simulink schematic double click on the PowerGUI icon. In the newly opened PowerGUI options window, click “Configure parameters” in the “Simulation and Configuration options” (Figure 27). Under the “solver” tab, select “Continuous” for “Simulation type” (Figure 28). Click “OK”.

Figure 27: PowerGUI options

Figure 28: PowerGUI parameter configurations

Then, under “Analysis Tools”, click “FFT Analysis” (Figure 27). In the “PowerGUI FFT Analysis Tool” window (Figure 29), select the signal to be analyzed. Under the “FFT window”, set the “start time” to 0.0s and “fundamental frequency” to 60 Hertz. Select the “number of cycles” of your own choice. Under “FFT settings”, select the “Display style” and “Max frequency” to your liking. Select “Frequency axis” to “Harmonic order”. Click “Display” to view the THD and FFT results.

Figure 29: PowerGUI FFT Analysis Tool window

B. Simulation Results

1) Half-Bridge and Full-Bridge PWM Simulation Results

Using MatLab Simulink, the half-bridge and full-bridge schematic were drawn as shown in Figures 30 and 31 respectively. The half-bridge used a 2-pulse PWM generator since it requires two switches while the full-bridge used a 4-pulse PWM generator since it utilizes 4 switches.

Figure 30: Matlab Simulink Half-Bridge Schematic

Figure 31: MatLab Simulink Full-Bridge Schematic

The inverter output waveforms are simulated for both topologies as shown in Figures 32 and 33. The 60Hz input sinusoidal wave is also included in both waveforms as reference.

Figure 32: Half-Bridge V_{in} (Yellow) and V_{out} (Purple) Waveforms

Figure 33: Full-Bridge V_{in} (Yellow) and V_{out} (Purple) Waveforms

For the half-bridge inverter output waveform, a unipolar PWM was observed. When the grid voltage is at $+V_{pk}$ (max peak input voltage), the PWM duty cycle is set at 100%. On the other hand, when the grid voltage is at $-V_{pk}$ (min peak input voltage), the duty cycle is set at 0%. Also, the output PWM signal goes between $(-V_{pk}$ and $+V_{pk})$. The switches are put in a lot of stress by the PWM signal since they need to withstand $2V_{pk}$.

For the full-bridge inverter output waveform, a bipolar PWM was observed. At $+V_{pk}$, the duty cycle is set at 100% on the positive cycle. At $-V_{pk}$, the duty cycle is also set at 100 percent but is now located at the negative cycle. The output PWM signal goes between (0V and $+V_{pk}$) or (0V and $-V_{pk}$). Compared to the half-bridge switches, this PWM signal puts half the stress on the full-bridge switches. The switches now only need to be rated at V_{pk} .

Using the Simulink's PowerGUI FFT analysis script, the harmonics for both topologies were simulated. From the half-bridge FFT analysis, as shown in Figure 34, there were odd and even harmonics with a THD of 146%. The largest harmonic is significantly large with a value of a little bit over a 100% with respect to the fundamental. This is an undesirable design since it may be difficult to filter the harmonic content since its magnitude is significantly large.

Figure 34: Half-Bridge FFT and THD Analysis

In Figure 35, the full-bridge topology offers improved harmonics since only odd harmonics were observed. Also the THD is lower at 77% and the harmonic content magnitude is less compared to the half-bridge. Its largest harmonic, with a value of 40% with respect to the fundamental, is easier to filter out compared to the half-bridge.

Figure 35: Full-Bridge FFT and THD Analysis (Note: 60Hz Fundamental is off the chart at 100%)

2) Three-Level PWM Simulations Results

To test the proposed design, a battery simulator script from MatLab Simulink is used to simulate the ESS. This simulated ESS will be used in the three-level-PWM topology. The parameters of the ESS are shown in Figure 36. The DC-bus voltage is set to 300VDC, and the chosen battery type is a 100Ah Lithium-Ion. Figure 37 shows the simulated battery's discharge characteristics. If the ESS drives a continuous 32.5A load during discharge mode, it will take the battery to fully discharge from 100% charge in around 3 hours.

Parameters	View Discharge Characteristics	Battery Dynamics
Battery type	Lithium-Ion	
Nominal Voltage (V)	300	
Rated Capacity (Ah)	100	
Initial State-Of-Charge (%)	100	
☒ Use parameters based on Battery type and nominal values		
Maximum Capacity (Ah)	100	
Fully Charged Voltage (V)	349.1961	
Nominal Discharge Current (A)	43.4783	
Internal Resistance (Ohms)	0.03	
Capacity (Ah) @ Nominal Voltage	90.4348	
Exponential zone [Voltage (V), Capacity (Ah)]	[324.1158 4.913043]	

Figure 36: ESS MatLab Simulink Parameters

Figure 37: ESS Battery Discharge Characteristics

Figure 38 shows the MatLab logic of the three-level PWM control system. This represents the closed-loop control system of the proposed design. It senses the voltage and the current of the grid. It also senses the DC bus voltages of the topology's two capacitors. With all the input signals sensed, it then generates the XYZ Boolean output logic that will control the 6 switches of the 3-level PWM circuit shown in Figure 39.

Figure 38: Three-Level PWM Closed Loop Control Schematic

Figure 39: Three-Level PWM Discharge Mode Schematic

Figure 40 shows the simulated three-level PWM inverter output waveform. The purple waveform is the inverter 3-level PWM output which tracks the yellow 60Hz sinusoidal input voltage. Since the topology is operating in discharge mode, the sinusoidal input current, in green, is 180° out of phase with the sinusoidal input voltage.

Figure 40: Three-Level PWM V2G Mode Waveforms

Compared to the inverter outputs of the half-bridge and full-bridge, shown in Figures 32 and 33 respectively, the three-level PWM has the least stress on the switches. The PWM signal either goes between $(V_{pk}/2 \text{ and } V_{pk})$, $(0V \text{ and } V_{pk}/2)$, $(0V \text{ and } -V_{pk}/2)$, or $(-V_{pk}/2 \text{ and } -V_{pk})$. The switch components only need to be rated at a magnitude of $V_{pk}/2$, which is half the required ratings of the full-bridge and a quarter of the half-bridge.

As seen in Figure 41, compared to the other topologies, there is a drastic improvement with the harmonic content. The THD is 21%. Its largest harmonic is less than 10% with respect to the fundamental. At this magnitude, the harmonic content is fairly easy to filter out compared to the other topologies.

Figure 41: Three-Level PWM FFT and THD Analysis (Note: 60Hz Fundamental is off the chart at 100%)

Table 3 shows a quick summary of the data gathered from Figure 32, 33, 34, 35, 40, and 41.

Table 3: Summary of Results

Topology	Switch Max Voltage	THD	Highest Magnitude Harmonic
Half-Bridge	$2V_{pk}$	146%	101%
Full-Bridge	V_{pk}	77%	39%
3-Level PWM	$V_{pk}/2$	21%	9%

IV. CLOSED LOOP MICROCONTROLLER HARDWARE PROTOTYPE DESIGN

A. *Overview of the Microcontroller Design*

For the purpose of this prototype, an Arduino microcontroller (Figure 42) was used because it is cheap and easy to implement the proposed design. Also the microcontroller could easily read the 60Hz input and output a switching maximum frequency of around 7kHz.

Figure 42: Arduino Microcontroller Board

Since this is a prototype, instead of using a 170VAC peak-to-peak (120V_{rms}) grid source, a scaled down source of 20VAC peak-to-peak (28.28V_{rms}, -10V_{pk-min}, +10V_{pk-max}) was used. The scaled-down source was still set at 60Hz to match the frequency found in typical North America power outlets.

The closed-loop design consists of multiple stages. The system requires the following stages: 1) Sense Stages [V_s , I_s , V_{c1} , V_{c2}], 2) Control Stage.

1) Sense Stages [V_s , I_s , V_{c1} , V_{c2}]

The sense stages handle all the inputs going into the closed-loop system. Such inputs are the grid voltage, grid current, and DC bus voltages for C1 and C2. These input signals are differential (non-grounded) signals. Unfortunately, the system could not handle differential signals as inputs. The sense stages convert these differential input signals to grounded voltage signals which the system could operate with. After that, the grounded voltage signals are mapped from 0V to 5V to match with Arduino's analog-to-digital converter (ADC). The sense stages are the following, a) Grid Voltage Sense Stage [V_s], b) Grid Current Sense Stage [I_s], and c) DC Bus Capacitors Voltage Sense Stages [V_{c1}/V_{c2}].

a) Grid Voltage Sense Stage [V_s]

For the closed-loop system to be able to determine the appropriate operation mode, it needs to sense the grid's voltage magnitude. This enables the control stage to determine which voltage level the inverter output should operate.

A differential amplifier circuit, shown in Figure 43, is used to implement the ADC mapping shown in Figure 44. The circuit uses the following parameters:

$$V_{out} = \left[(V_2 - V_1) * \frac{R_f}{R_{in}} \right] + V_{offset}$$

Where:

$(V_2 - V_1) = \text{differential voltage sense input } (V_s)$

$$R_f = 3k\Omega$$

$$R_{in} = 12k\Omega$$

$$V_{offset} = 2.5V$$

Figure 43: Differential Amplifier Schematic

As seen in Figure 44, the scaled-down analog voltage sense input goes +10Vmax to -10Vmin. For the Arduino board to understand this analog voltage input, the signal should be mapped to match the built-in ADC. The ADC of the Arduino has a maximum range of 5V, so the +10Vmax input signal is mapped to 5V. The minimum ADC range value is at 0V therefore the -10Vmin input signal is mapped to 0V.

Figure 44: Vs ADC Mapping

The differential amplifier is simulated in spice. The differential amplifier spice circuit shown in Figure 45 is simulated. R_f is set to $3k\Omega$, R_{in} to $12k\Omega$. The voltage offset is set to 2.5V. The simulated spice waveforms are shown in Figure 46. The green waveform is the scaled down Grid voltage and the blue waveform is the output of the differential amplifier.

Figure 45: Differential Amplifier for Grid Voltage Sense Spice Simulation Schematic

Figure 46: Differential Amplifier for Grid Voltage Sense Spice Simulation Input (Blue) and Output (Green) Waveforms

b) Grid Current Sense Stage (I_s)

Not only does the closed-loop system need to sense the grid's voltage but it also needs to determine the grid's current magnitude. This ensures the inverter's output current is in-phase with the grid's 60Hz sinusoidal current during discharge mode.

An instrumentation amplifier shown in Figure 47 is used to implement the ADC mapping presented in Figure 48. This type of circuit was chosen instead of the differential amplifier because it has input buffers to eliminate impedance mismatching [21]. This circuitry is suitable for this type of application because it needs to measure the voltage across a shunt resistor to determine the grid's current. The AD 623 chip shown in Figure 47 was specifically used with the following parameters:

$$V_{out} = [(+IN) - (-IN)] * \left(1 + \frac{100k\Omega}{R_g}\right)$$

Where:

$$[(+IN) - (-IN)] = \text{differential current sense input } (I_s)$$

$$R_g = 15k\Omega$$

The shunt resistor used is set at $R_{sense} = 100m\Omega$

Figure 47: AD623 Instrumentation Amplifier Schematic and Pin Assignment [22]

Just like the V_s voltage sense mapping, the same principle goes to the I_s current sense. In Figure 48, the maximum +270A input signal is mapped to 5V and the minimum -270mA signal is mapped to 0V.

Figure 48: I_s ADC Mapping

The instrumentation amplifier is simulated in spice. The spice circuit shown in Figure 49 is simulated. The simulated spice waveforms are shown in Figure 50. The green waveform is the grid current and the blue waveform is the output of the instrumentation amplifier.

Figure 49: Instrumentation Amplifier for Grid Current Sense Spice Simulation Schematic

Figure 50: Instrumentation Amplifier for Grid Current Sense Spice Simulation

Input (Blue) and Output (Green) Waveforms

c) DC Bus Voltage Capacitors Voltage Sense Stages [VC1 and VC2]

The closed loop-system also needs to sense the voltage level of both the DC bus capacitor to ensure that the inverter's output current is in-phase with the grid's 60Hz sinusoidal current during discharge mode.

Just like the V_s sense stage, differential amplifier circuits were used to implement the ADC mapping presented in Figure 51. The following parameters were used:

$$V_{out} = \left[(V_2 - V_1) * \frac{R_f}{R_{in}} \right] + V_{offset}$$

Where:

$$(V_2 - V_1) = \text{differential voltage sense input (VC1/VC2)}$$

$$R_f = 10k\Omega$$

$$R_{in} = 20k\Omega$$

$$V_{offset} = 0V$$

The VC1/VC2 voltage signals goes from 0V to 10V. Figure 51 shows that the maximum +10V input signal is mapped at 5V and the minimum 0V signal is mapped at 0V.

Figure 51: Vc1 Vc2 ADC Mapping

The differential amplifier is simulated in spice. The differential amplifier spice circuit is shown in Figure 52. R_f is set to $3k\Omega$, R_{in} to $12k\Omega$. The voltage offset is set to 2.5V. The simulated spice waveforms are shown in Figure 53. The green waveform is the voltage across C1 or C2 and the blue waveform is the output of the differential amplifier.

Figure 52: Differential Amplifier for VC1/VC2 Voltage Sense Spice Simulation Schematic

Figure 53: Differential Amplifier for VC1/VC2 Voltage Sense Spice Simulation

2) Control Stage

The prototype closed loop controller design was written in C++. The code was used to program the control stage or specifically, the Arduino microcontroller. The Arduino monitors the mapped signals generated by V_s , I_s , V_{c1} and V_{c2} . Then, it converts the mapped signals to a digital value through its built-in ADC. The converted digital value for 0V, the minimum mapped voltage level, is 0. The digital representation for 5V, the maximum mapped voltage level, is 1023. Figures 54, 55, and 56 illustrate the ADC value conversion for the mapped signals V_s , I_s , V_{c1} , and V_{c2} respectively. After the ADC conversion, it then generates the XYZ Boolean logic. The logic then determines the appropriate output signals to control the 6 switches of the three-level PWM topology.

Figure 54: V_s ADC Conversion

Figure 55: I_s ADC Conversion

Figure 56: V_{c1}/V_{c2} ADC Conversion

B. Hardware Test

1. Eagle PCB Layout Design Procedure

To test if the prototype design works, a test PCB layout board was designed and built. The following illustrates the steps to design a PCB layout.

a. Package Creation

Using the link <http://www.cadsoftusa.com/download-eagle/?language=en>, download the free version Cadsoft Eagle from There are usually three steps involved in the entire PCB process: package creation, schematic entry, and PCB layout. The package creation step, in some cases, can be skipped if a package/footprint for a specific part is already given.

Start by creating a new library in Eagle (refer to Figure 57). After a new library window appears, name the library file and save it to the “lbr” directory where Eagle was installed to (example: JohnDoeCustom.lib).

Figure 57: Cadsoft Eagle main window – starting a custom library

During the component creation process, there are three steps the user must proceed through to make the final part. The three steps are Package Creation, Symbol Creation, and Device Creation. Package Creation involves creating a footprint for the component, which will consist of laying out land patterns (surface mount, through hole, etc.), silk screens, and so forth. Symbol Creation involves creating the schematic symbol for the component. Component Creation links the package to the schematic symbol. Refer to Figure 58 for the library window.

Figure 58: Cadsoft Eagle library window

Begin component creation by selecting the package button. When the “Edit” window appears, name the package after the component’s package type, and then save the name. A black background with a grid should appear; the window has now changed to the package creation window. Before setting up the package in the workplace, the user should have consulted the component’s datasheet for recommended land patterns. Once the land pattern is found, start the process by setting up the work space’s grid (Figure 59). After setting a desired grid area, start placing either surface mount pads (SMDs) or through-hole pads onto the workspace using the recommended land patterns from the datasheet. The size and shape of each pad can be altered by going to the top of the work space and changing the dimensions (Figure 60). To make a circular SMD, set the roundness to 100%.

Figure 59: Package creation work space

Figure 60: SMD and through-hole size/dimension options, respectively

Note that the SMDs should be on the “Top” (copper) layer, and the through-hole pads should be on the “Pads” layer. To check if certain aspects of the design are on the right layers, select the visible layers drop down and select/deselect layers. After placing pads,

silkscreens can be placed to outline the component's package and display the component's schematic reference number and manufacturer number. Component outline silkscreens can be created using the "Wire" and "Circle" tools on the left toolbar, while the reference number (Name) and manufacturer's part number (Value) silkscreens can be created using the "Text" tool. Note that during component creation, the reference number and part number should not be input, but rather ">Name" and ">Value." The actual designators will be placed onto the layout workspace once the schematic is created. The component outline silkscreens should be created on the "tPlace" layer while the Name and Value silkscreens should be placed on the "tName" and "tPlace" layers, respectively.

Next, create a schematic symbol by selecting the associated icon. An "Edit" prompt will appear much like during the package process. Name the schematic symbol after the manufacturer's part name and proceed. A window similar to the package creation workspace will appear, but the background will be white (Figure 61). The default grid configuration should be acceptable for this process. The entire schematic process is similar to that of the package process, where each pin will receive a name/designator and the symbol will be labeled with the >Name and >Value conventions.

Figure 61: Schematic symbol creation work space

Finally, select the “Device” icon to complete the component creation process. A new window should be as shown in Figure 62. First, click the “Add” button and select the schematic symbol. Next, click the “New” button and add the package symbol. Finally, press the “Connect” button to link the schematic and package symbols together. The connection window should look like Figure 63. If the pads and pins from the previous processes were given their respective names, they should line up directly with one another as shown in Figure 63. If not, the user will have to search for and select the connections between each associated pad and pin. By the end of the connection process, the “Connection” section in the last-third of the window should be full. Press “OK” to complete the component process.

Figure 62: Device creation work space

Figure 63: Schematic and package symbol connection window

b. Schematic and Layout

Start a new project in Eagle by referring to Figure 64, and then starting a new schematic as shown in Figure 65.

Figure 64: Starting a new project in Eagle

Figure 65: Starting a new schematic in the project directory

A new schematic work space should appear, as shown in Figure 66. All parts should be accessible via the “Add” button (including parts from the custom-made library). Place parts onto the schematic work space and wire them together using the “Wire” tool. All components can have their reference names and values changed by selecting the “Name” and “Value” tools. One useful tool to use is the “Group” tool when having to move or copy multiple components at any given time. Select either the move or copy option. Next, group all desired components together, and then right-click the work space. Finally, when the tool-tip appears, select “Group: “<Action>,” where <Action> is either move, copy, or another command. For an example of a finished schematic, refer to Appendix C. After completing the schematic entry, press the “Board” button at the top of the work space to generate a board layout from the schematic.

Figure 66: Schematic workspace

When Eagle generates the board layout, the program will create a component pile at the bottom left corner of the layout work space (Figure 67). Before routing the board, it is usually wise to specify the design rules for the board, which follow the guidelines established by the board fabricator. To access the design rules, go to **Tools > DRC...** and input the necessary fabrication limitations.

Figure 67: Layout Workspace

As shown in Figure 67, the white rectangular boundaries within the work space are the board dimensions for the PCB. The boundaries can be adjusted by clicking and dragging. Each of the components on the screen can also be clicked and dragged onto the PCB area. Note how each connected component from the schematic is now connected by a yellow line known as a “netlist.” All components can either be manually routed by following each netlist, or an auto-route feature can be used by selecting the “Auto” button on the left toolbox. When routing components, a majority of the routing will take place on the “Top” layer (rustic-colored). However, routing can also take place on the “Bottom” layer (blue-colored). For an example, refer to Appendix D. Because the board

is a 2-layer design, the top layer is usually the component layer whereas the bottom layer is specified as ground.

To set the bottom layer as ground, select the polygon tool in the left toolbox, and then select the layer to be “Bottom” from the layer pulldown menu at the top of the work space. Create a rectangular polygon around the inner sides of the PCB boundary. After the polygon is created, right-click on the polygon, select “Properties,” and rename the signal name to “GND.” Once connections are made to the GND netlist, the bottom layer should automatically fill up from the blue polygon GND layer. If not, press the “Ratsnest” button. Use the steps above to complete the PCB layout.

c. Gerber File Generation

To generate the Gerber and drill files, select the “CAM” button at the top of the work space (Figure 67). When the CAM processor window opens, a job must be selected to generate the Gerber files for a 2-layer board (Figure 68). Next, select the “gerb274x.cam” job file, which is a default job created specifically for 2-layer boards.

Figure 68: CAM processor window

Once the job is selected, the Cam processor window should now look like Figure 69. Take notice of the tabs that appear at the top of the window. Gerber files should be generated for the component layer, ground layer, component silkscreens, component layer solder-mask stop, and ground layer solder-mask stop. For each layer, the affected board attributes are highlighted in blue. Click “Process Job” when all parameters are configured.

Figure 69: Gerber file configuration for a 2-layer board job

After the Gerber Files are saved, select File > Open > Job... and select the “excellon.cam” job to create the drill files for the board. A similar screen like Figure 69 should appear. Again, once all parameters are set, click “Process Job.” When sending the Gerber and drill files to a board fabrication house, certain files are necessary in the board fabrication process. For this project, the company Advanced Circuits (www.4pcb.com) was used as a board fabrication vendor. Table 4 lists the files necessary to have Advanced Circuits fabricate a board.

Table 4: Necessary Gerber and drill files for board fabrication

File	Description
.cmp	Component side data (Top Copper)
.sol	Solder side data (Bottom Copper)
.plc	Component side silk screen data (Top Silk Screen)
.stc	Component side solder stop mask data (Top Solder Mask)
.sts	Solder side solder stop mask data (Bottom Solder Mask)
.drd	Excellon drill description (NC Drill)

2. PCB Test Board

Figure 70 shows the Eagle schematic design. It includes the three-level PWM inverter/rectifier circuit with the Vs/Is/VC1/VC2 sense stages. Figure 71 shows the two-sided PCB layout Eagle design. The populated PCB board design with the Arduino Microcontroller is shown in Figure 72. Figure 73 lists the Digikey build of materials (BOM) used on the board design build.

Figure 70: Eagle PCB Test Board Schematic

Figure 72: Eagle PCB Test Board with the Arduino Microcontroller Control Stage

Index	Quantity	Part Number	Description	Customer Reference	Backorder Quantity	Unit Price	Extended Price
1	2	P6901-ND	CAP ALUM 47000UF 25V 20% SNAP		0	6.80000	\$13.60
2	8	MUR105GOS-ND	DIODE ULTRA FAST 1A 50V DO-41		0	0.44000	\$3.52
3	30	1040K-ND	TERM TEST POINT SLOTTED .118"		0	0.17700	\$5.31
4	6	IRFZ20PBF-ND	MOSFET N-CH 50V 15A TO-220AB		0	1.57000	\$9.42
5	10	RMCF1206FT10K0CT-ND	RES 10K OHM 1/4W 1% 1206 SMD		0	0.04400	\$0.44
6	4	RMCF1206FT12K0CT-ND	RES TF 12K OHM 1% 0.25W 1206		0	0.06000	\$0.24
7	2	CRA2512-FZ-R100ELFCT-ND	RES .10 OHM 3W 1% 2512 SMD		0	0.49000	\$0.98
8	2	311-0.001AGCT-ND	RES 0.001 OHM 1W 1% 2512 SMD		0	1.01000	\$2.02
9	2	RMCF2512ZT0R00CT-ND	RES 0.0 OHM 1W 2512 SMD		0	0.44000	\$0.88
10	2	P13.0KFCT-ND	RES 13.0K OHM 1/4W 1% 1206 SMD		0	0.10000	\$0.20
11	4	311-3.00KFCT-ND	RES 3.00K OHM 1/4W 1% 1206 SMD		0	0.10000	\$0.40
12	10	RMCF1206FT20K0CT-ND	RES 20K OHM 1/4W 1% 1206 SMD		0	0.04400	\$0.44
13	1	AD623ANZ-ND	IC AMP INST R-R LP 8DIP		0	4.90000	\$4.90
14	3	1014-1091-ND	IC OPAMP GP R-R CMOS 8PDIP		0	2.83000	\$8.49
15	2	RMCF1206FT15K0CT-ND	RES 15K OHM 1/4W 1% 1206 SMD		0	0.06000	\$0.12
16	50	P0.0FCT-ND	RES 0.0 OHM 1/4W 1206 SMD		0	0.02420	\$1.21
Subtotal							\$52.17

Figure 73: Digikey BOM

3. Equipment List

Figure 74 shows the setup used to test the PCB test board. Below shows a list of equipment used to test the hardware.

- Agilent E3630A Power Supply
 - This powers the instrumentation amplifier and differential amplifier op-amps.
- Agilent MSO-X 2012A Mixed Signal Oscilloscope
 - This measures and capture input/output signals from the PCB test board.
- Rigol DG1022 2 channel 20Mhz Function Generator
 - This generates the scaled-down grid input signal (Vs)

Figure 74: PCB Test Board Setup

4. *Results*

a. *Grid Voltage Sense (V_s) Results*

Both channels of the Rigol DG1022 function generator were used to generate the scaled-down differential input ($V_a - V_b$). The input waveforms are shown in Figure 75, where channel 1 is V_a (yellow waveform) and channel 2 is V_b (green waveform).

Subtracting channel 2 from channel 1, the differential input $V_a - V_b$ (purple waveform) was generated. This input signal is a 60Hz sinusoidal waveform that goes to a minimum peak voltage of -10V to a maximum peak of +10V. This input was fed to the differential amplifier circuit and the output (V_s) was captured and shown in Figure 76. The output (V_s) corresponds to the simulated output presented in Figure 46.

Figure 75: Differential Amplifier for Grid Voltage Sense input waveforms

where Channel 1 - V_a (orange), Channel 2 - V_b (green), and Math Channel ($V_a - V_b$) (purple)

Figure 76: Differential Amplifier for grid voltage sense output waveform

b. Grid Current Sense (I_s) Results

Using the same input ($V_a - V_b$) shown in Figure 75, the instrumentation amplifier output (I_s) was captured and is shown in Figure 77. The output simulation presented in Figure 50 did not match with the experimental data. Harmonic distortion was observed as well as some clipping.

Figure 77: Instrumentation Amplifier for grid current sense output waveform

c. DC Bus Capacitors Voltage Sense (VC1/VC2) Results

To test the differential amplifier for the DC bus voltage sense stages, they were isolated in the circuit by removing the 0Ω jumpers surrounding it. A 50% duty cycle PWM signal was used as the differential input. The frequency was set to 3.9kHz to simulate the Arduino's PWM output signal. The input and output waveforms were captured and are shown in Figure 78. The experimental output (Vc1/Vc2) matched the simulated output presented in Figure 53.

Figure 78: Differential Amplifier for VC1/VC2 voltage sense input (orange) and output (green) waveforms
where input PWM Duty Cycle is set at 50%

d. Control stage (Arduino Microcontroller) Results

To test the control stage, it was isolated from the overall circuit. All the sense stages (V_s , I_s , V_{c1} , and V_{c2}) were disconnected by removing the 0Ω jumpers. Then, test signals that replicate V_s , I_s , V_{c1} , and V_{c2} were fed into the Arduino. The replicated values used for the test signals were based on the mapped values observed from the simulations. In charge mode, a 0V-5V sinusoidal wave was used to replicate V_s and I_s . A constant 3.97V was used for V_{c1} and V_{c2} . In V2G mode, a 0V-5V sinusoidal wave was also used for V_s . The I_s signal also used the same magnitude of

the test signal but it was 180° out of phase. A constant 5V was used for Vc1 and Vc2. These test signals are then processed into the Arduino. The ADC converts those mapped analog signals into digital signals. The microcontroller then processes the XYZ boolean logic to determine the operation mode. From that particular mode, the microcontroller outputs the appropriate gate logic to switches T1, T2, T3, T4, T5, and T6. Data was gathered and recorded when the test signal (Vs) was at 5V, 3.75V, 2.5V, 1.25V and 0V. The experimental gate logic matched with the logic presented in Tables 1 and 2. The summarized data is shown in Table 5.

Table 5: Arduino Control Stage Results

	Test Input Signals				ADC Digital Value				Boolean Logic			Operation Mode	Gate Logic					
	Vs	Is	Vc1	Vc2	Vs	Is	Vc1	Vc2	X	Y	Z	Mode	T1	T2	T3	T4	T5	T6
Charge Mode	5	5	3.97	3.97	1023	1023	812	812	1	1	0	2	off	on	off	on	on	off
	3.75	3.75	3.97	3.97	767	767	812	812	1	1	0	2	off	on	off	on	on	off
	2.5	2.5	3.97	3.97	511	511	812	812	0	0	0	6	off	off	off	on	on	on
	1.25	1.25	3.97	3.97	255	255	812	812	0	1	1	1	on	off	on	off	off	on
	0	0	3.97	3.97	0	0	812	812	0	1	1	1	on	off	on	off	off	on
V2G Mode	Vs	Is	Vc1	Vc2	Vs	Is	Vc1	Vc2	X	Y	Z	Mode	T1	T2	T3	T4	T5	T6
	5	0	5	5	1023	0	1023	1023	1	1	0	2	off	on	off	on	on	off
	3.75	1.25	5	5	767	255	1023	1023	1	1	0	2	off	on	off	on	on	off
	2.5	2.5	5	5	511	511	1023	1023	0	0	0	6	off	off	off	on	on	on
	1.25	3.75	5	5	255	767	1023	1023	0	1	1	1	on	off	on	off	off	on
	0	5	5	5	0	1023	1023	1023	0	1	1	1	on	off	on	off	off	on

V. CONCLUSION

Most of the stages, such as the V_s sense, V_{c1} , and V_{c2} sense, and control stage, were all working as expected. The differential amplifier experimental data for V_s in Figure 73, matches with the data gathered from the simulations presented in Figure 48. Same goes for the differential amplifier experimental outputs for V_{c1}/V_{c2} shown in Figure 75, as they match the simulated data from Figure 53. Also, the data gathered from the Arduino microcontroller presented in Table 4 outputs the appropriate gate logic signals as expected.

Unfortunately the over-all close-loop system fails to output the three-level PWM that was expected from the simulations shown in Figure 40. The main culprit for this failure is due to the grid current sense (I_s) stage. The experimental data displayed in Figure 74 did not match the simulated data exhibited in Figure 51. The experimental data contained a lot of harmonic distortion probably caused from the microcontroller $\sim 7\text{kHz}$ switching output signals. This makes the shunt resistor of the instrumentation amplifier sensitive and susceptible to these noises. A probable fix to this, is using a hall-effect sensor to measure the grid current. This sensor does not require a shunt resistor and is totally isolated from the grid which makes it more robust for this particular application. Unfortunately for the purpose of this prototype design, this sensor type is not realizable. Most hall-effect sensors require high voltage for it to detect current and the scaled down $20\text{V}_{\text{pk-pk}}$ ($-10\text{V}_{\text{pk-min}}$ to $+10\text{V}_{\text{pk-max}}$) grid voltage used in the prototype is relatively small.

Another improvement noted for this prototype design is using a rail-to-rail ADC instead of the Arduino's ground-to-rail ADC. This prevents the use of voltage offsets. This makes it easier for the microcontroller to determine the ground since with rail-to-rail ADC's, the ground could be set to 0V.

REFERENCES

- [1] "International Energy Outlook 2009," Energy Information Administration (EIA), Office of Integrated Analysis and Forecasting, U.S. Department of Energy, OE/EIA-0484(2009), May 2009.
- [2] (2012, August). *Tesla Motors*.
Available: <http://www.teslamotors.com/goelectric/charging>
- [3] (2012, August). *Nissan Leaf Electric Car*.
Available: <http://www.nissanusa.com/leaf-electric-car/index#/leaf-electriccar/index>
- [4] (2012, August). *2011 Chevrolet Volt Plug-in Hybrid Electric Car*.
Available: <http://www.chevrolet.com/pages/open/default/future/volt.do>
- [5] M.C. Kisacikoglu, B. Ozpineci, T.M. Tolbert, "Reactive power operation analysis of a single-phase EV/PHEV bidirectional battery charger," *Power Electronics and ECCE Asia (ICPE & ECCE), 2011 IEEE 8th International Conference on*, vol., no., pp.585-592, May 30 2011-June 3 2011.
- [6] J. A. P. Lopes, F. J. Soares, P. M. R. Almeida, "Integration of electric vehicles in the electric power system," in *Proc. of the IEEE*, vol. 99, pp. 168-183, Jan. 2011.
- [7] E.S. Dehaghani, S.S. Williamson, "On the inefficiency of vehicle-to-grid (V2G) power flow: Potential barriers and possible research directions," *Transportation Electrification Conference and Expo (ITEC), 2012 IEEE*, vol., no., pp.1-5, 18-20 June 2012.
- [8] C. Guille, G. Gross, "Design of a Conceptual Framework for the V2G Implementation," *Energy 2030 Conference, 2008. ENERGY 2008. IEEE*, vol., no., pp.1-3, 17-18 Nov. 2008.
- [9] E. Sortomme, M.A. El-Sharkawi, "Optimal Scheduling of Vehicle-to-Grid Energy and Ancillary Services," *Smart Grid, IEEE Transactions on*, vol.3, no.1, pp.351-359, March 2012.
- [10] (2012, August). *Plug-in America, Plug-in EV Charger Level Classification*,
Available: <http://www.pluginamerica.org/drivers-seat/understanding-electric-vehicle-charging>
- [11] Y.H. Liu, C.H. Hsieh, Y.F. Luo, "Search for an Optimal Five-Step Charging Pattern for Li-Ion Batteries Using Consecutive Orthogonal Arrays," *Energy Conversion, IEEE Transactions on*, vol.26, no.2, pp.654-661, June 2011.
- [12] SAE International, "SAE J1772 Electric Vehicle and Plug in Hybrid Electric Vehicle Conductive Charge Coupler", 21 Feb 2012.

- [13] E. Sortomme, M.A. El-Sharkawi, "Optimal charging strategies for unidirectional vehicle-to-grid," *IEEE Trans. Smart Grid*, vol. 2, no. 1, pp. 131–138, 2011.
- [14] J. Tomic, W. Kempton, "Using fleets of electric-drive vehicles for grid support," *J. Power Sources*, vol. 168, no. 2, pp. 459–468, 2007.
- [15] G. Glanzer, T. Sivaraman, J.I. Buffalo, M. Kohl, H. Berger, "Cost-efficient integration of electric vehicles with the power grid by means of smart charging strategies and integrated on-board chargers," *Environment and Electrical Engineering (EEEIC), 2011 10th International Conference on*, vol., no., pp.1-4, 8-11 May 2011.
- [16] D.C. Erb, O.C. Onar, A. Khaligh, "Bi-directional charging topologies for plug-in hybrid electric vehicles," *Applied Power Electronics Conference and Exposition (APEC), 2010 Twenty-Fifth Annual IEEE*, vol., no., pp.2066-2072, 21-25 Feb. 2010.
- [17] IEEE; "IEEE Application Guide for IEEE Std. 1547", *IEEE Standard for Interconnecting Distributed Resources With Electric Power Systems, IEEE 1547.2-2008*, 2008.
- [18] P. Peltoniemi, R. Pollanen, M. Miemela, J. Pyrhonen, "Comparison of the Effect of Output Filters on the Total Harmonic Distortion of Line Current in Voltage Source Line Converter – Simulation Study", Lappeenranta University of Technology. Available: <http://www.icrepq.com/icrepq06/297-peltoniemi.pdf>
- [19] B.R. Lin, D.J. Chen, H.R. Tsay, "Bi-directional AC/DC converter based on neutral point clamped," *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, vol.1, no., pp.619-624 vol.1, 2001.
- [20] D.C. Erb, O.C. Onar, A. Khaligh, "An integrated bi-directional power electronic converter with multi-level AC-DC/DC-AC converter and non-inverted buck-boost converter for PHEVs with minimal grid level disruptions," *Vehicle Power and Propulsion Conference (VPPC), 2010 IEEE*, vol., no., pp.1-6, 1-3 Sept. 2010.
- [21] R.F. Coughlin, F.F. Driscoll, *Operational Amplifiers and Linear Integrated Circuits* (2nd Ed.1982. ISBN 0-13-637785-8) p.161.
- [22] Analog Devices, "AD623 Single-Supply, Rail-to-Rail, Low Cost Instrumentation Amplifier datasheet", 2008.

Appendix

Arduino Code:

```
/* 3-Lvl PWM Closed Loop Controller */

int vsPin = A5; //Vs input
int isPin = A4; //Is input
int vc1Pin = A3; //VC1 input
int vc2Pin = A2; //VC2 input

int quarterVpkpk = 255; //[(Vpk-pk)/4 target value. Where:[Vpk-pk]/4 = [(1024)/4]-1 = 255 ==> [170Vpk]/4 = 42.5V
int halfVpkpk = 511; //[(Vpk-pk)/2 target value. Where:[Vpk-pk]/2 = [(1024)/2]-1 = 511 ==> [170Vpk]/2 = 85.0V

int t1Pin = 11; //switch t1 pin
int t2Pin = 10; //switch t2 pin
int t3Pin = 9; //switch t3 pin
int t4Pin = 6; //switch t4 pin
int t5Pin = 5; //switch t5 pin
int t6Pin = 3; //switch t6 pin

boolean t1 = LOW; //t1 gate
boolean t2 = LOW; //t2 gate
boolean t3 = LOW; //t3 gate
boolean t4 = LOW; //t4 gate
boolean t5 = LOW; //t5 gate
boolean t6 = LOW; //t6 gate

int xPin = 2; //test pin for boolean X
int yPin = 4; //test pin for boolean Y
int zPin = 7; //test pin for boolean Z

boolean x = LOW; //boolean X
boolean y = LOW; //boolean Y
boolean z = LOW; //boolean Z

int vsVALUE = 0; //convert analog vsPin input to digital value.
//Where Maximum: 170V (scaled to 5.0V) = 1023
// Zero/Gnd: 0V (scaled to 2.5V) = 0511
// Minimum: -170V (scaled to 0.0V) = 0000

int isVALUE = 0; //convert analog isPin input to digital value.
//Where Maximum: 10A (scaled to 5.0V) = 1023
// Zero/Gnd: 0A (scaled to 2.5V) = 0511
// Minimum: -10A (scaled to 0.0V) = 0000

int vc1VALUE = 0; //convert analog vc1Pin input to digital value
//Where Maximum: 170V (scaled to 5.0V) = 1023
// Minimum: 0V (scaled to 0.0V) = 0000
//Note: [Vbatt]/2 = [340V]/2 = Vc1 = 170V

int vc2VALUE = 0; //convert analog vc2Pin input to digital value
//Where Maximum: 170V (scaled to 5.0V) = 1023
// Minimum: 0V (scaled to 0.0V) = 0000
//Note: [Vbatt]/2 = [340V]/2 = Vc2 = 170V

int ipVALUE = 0; //used for boolean X logic

void setup()
{
  Serial.begin(9600);

  pinMode(t1Pin, OUTPUT);
  pinMode(t2Pin, OUTPUT);
  pinMode(t3Pin, OUTPUT);
  pinMode(t4Pin, OUTPUT);
  pinMode(t5Pin, OUTPUT);
  pinMode(t6Pin, OUTPUT);
  pinMode(xPin, OUTPUT);
  pinMode(yPin, OUTPUT);
  pinMode(zPin, OUTPUT);

  TCCR0B = TCCR0B & 0b11111000 | 0x02; //set PWM pin 5 and 6 freq
  TCCR1B = TCCR1B & 0b11111000 | 0x02; //set PWM pin 9 and 10 freq
  TCCR2B = TCCR2B & 0b11111000 | 0x02; //set PWM pin 3 and 11 freq
}
```



```

void loop()
{
    //Arduino inputs
    vsVALUE = analogRead(vsPin);
    isVALUE = analogRead(isPin);
    vc1VALUE = analogRead(vc1Pin);
    vc2VALUE = analogRead(vc2Pin);

    //Arduino process

    // Boolean X logic
    ipVALUE = ((halfVpkpk)-((vc1VALUE-511)+(vc1VALUE-511)))*(vsVALUE-511);

    if(ipVALUE <= 0 )
    {
        x = HIGH; digitalWrite(xPin, HIGH);
    }

    else
    {
        x = LOW; digitalWrite(xPin, LOW);
    }

    // Boolean Y logic
    if(vsVALUE <= 255 || vsVALUE >= 767)
    {
        y = HIGH;
        digitalWrite(yPin, HIGH);
    }

    else { y = LOW; digitalWrite(yPin, LOW);
    }

    // Boolean Z logic if(vsVALUE < 511)
    {
        z = HIGH; digitalWrite(zPin, HIGH);
    }

    else
    {
        z = LOW; digitalWrite(zPin, LOW);
    }

    //Switches Logic
    if(x==LOW && y==LOW && z==LOW) //OV
    {
        t1=0; t2=0; t3=0; t4=1; t5=1; t6=1;
    }

    else if(x==LOW && y==LOW && z==HIGH) //(Vdc-dc)/2
    {
        t1=0; t2=0; t3=1; t4=1; t5=0; t6=1;
    }

    else if(x==LOW && y==HIGH && z==LOW) //(Vdc-dc)/2
    {
        t1=0; t2=0; t3=1;
        t4=1; t5=0; t6=1;
    }

    else if(x==LOW && y==HIGH && z==HIGH) //(Vdc-dc)
    {
        t1=1; t2=0; t3=1; t4=0; t5=0; t6=1;
    }

    else if(x==HIGH && y==LOW && z==LOW) //- (Vdc-dc)/2
    {
        t1=0; t2=1; t3=1; t4=1; t5=0; t6=0;
    }

    else if(x==HIGH && y==LOW && z==HIGH) //0
    {
        t1=1; t2=1; t3=1; t4=0; t5=0; t6=0;
    }

    else if(x==HIGH && y==HIGH && z==LOW) //- (Vdc-dc)
    {
        t1=0; t2=1; t3=0; t4=1; t5=1;
        t6=0;
    }
}

```