The Pennsylvania State University

The Graduate School

Department of Electrical Engineering

# ANALYSIS, DESIGN AND IMPLEMENTATION OF A CHARGE-EQUALIZATION CIRCUIT FOR USE IN AUTOMOTIVE BATTERY MANAGEMENT SYSTEMS

A Thesis in

Electrical Engineering

by

Nicholas James Brannen



Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2008

The thesis of Nicholas James Brannen was reviewed and approved* by the following:

Jeffrey S. Mayer
Associate Professor of Electrical Engineering
Thesis Advisor

John D. Mitchell
Professor of Electrical Engineering

Joel R. Anstrom
Research Associate of Mechanical Engineering

W. Kenneth Jenkins
Professor of Electrical Engineering
Head of the Department of Electrical Engineering

*Signatures are on file in the Graduate School

## ABSTRACT

A charge-equalization circuit for an automotive battery management system has been designed and implemented. The selection of the circuit topology is based on an analysis of several different topologies and criteria for utilizing the battery management system in two Penn State research vehicles. The design process included the selection of appropriate power electronic components, the adaptation of a current-mode controller to facilitate proportional control, and the design of the necessary analog electronics and communication network. Implementation included the fabrication of a printed circuit board, the population of the board, and the testing of the sub-circuits. The system was tested for its ability to transfer prescribed currents. Recommendations for improvements are discussed.

**TABLE OF CONTENTS**

LIST OF FIGURES ..... vi

LIST OF TABLES ..... x

ACKNOWLEDGEMENTS ..... xii

Chapter 1 Introduction ..... 1

1.1 Automobile Industry Transition toward Electric Vehicles ..... 1
1.2 Batteries and Battery Management Systems ..... 4
1.3 Penn State Research Vehicles ..... 6
1.4 Organization of Thesis ..... 8

Chapter 2 Candidate Solutions ..... 9

2.1 Evaluation Criteria ..... 9
2.2 General System Structures ..... 11
2.3 Series Systems ..... 13
2.3.1 Shunt-Resistor System ..... 13
2.3.2 Switched-Capacitor System ..... 15
2.3.3 Current-Diverter System ..... 18
2.3.4 Bidirectional Current-Diverter System ..... 23
2.4 Parallel Systems ..... 26
2.4.1 Flyback-Converter System ..... 26
2.4.2 Bidirectional Flyback-Converter System ..... 29
2.4.3 Centralized Forward-Converter System ..... 32
2.5 Hybrid Systems ..... 35
2.5.1 Hybrid System #1 ..... 36
2.5.2 Hybrid System #2 ..... 40
2.6 System Selection for the Penn State Research Vehicles ..... 43

Chapter 3 Analysis, Design, and Implementation of a Charge-Equalization Circuit ..... 46

3.1 Operating Conditions ..... 46
3.2 Analysis ..... 47
3.2.1 Discharging the Battery Module ..... 49
3.2.2 Charging the Battery Module ..... 53
3.3 Design ..... 57
3.3.1 Current, Voltage, and Power Levels ..... 59
3.3.2 Power Electronics ..... 63
3.3.3 Control ..... 67

3.3.4 Communication ..... 75
3.3.5 Microcontroller ..... 78
3.3.6 Power ..... 79
3.3.7 Software ..... 81
3.4 Implementation ..... 85
3.4.1 Circuit Layout ..... 85
3.4.2 On-Board DC-to-DC Converters ..... 86
3.4.3 Microcontroller Hello World ..... 86
3.4.4 Communication Validation ..... 88
3.4.5 Gate Drive Validation ..... 90
3.4.6 Feedback Signal Filter Circuits Validation ..... 93
3.4.7 Building the Transformer ..... 101

Chapter 4 Experimental Results ..... 106

4.1 Open-loop Peak-Current Test ..... 106
4.1.1 MOSFET Peak-Current Test ..... 107
4.1.2 IGBT Peak-Current Test ..... 111
4.2 Proportional Control Test ..... 115
4.3 Observations and Recommended Changes ..... 115

Chapter 5 Summary and Conclusions ..... 119

Bibliography ..... 122

Appendix A Communication Messages ..... 126

Appendix B Microcontroller Code ..... 130

Appendix C Schematic, Printed Circuit Board, and Parts List ..... 150

# LIST OF FIGURES

Figure 1.1: Penn State Electric-Lion vehicle and battery pack ..... 7

Figure 1.2: Penn State HyLion vehicle and battery pack ..... 7

Figure 2.1: General classification of BMS structures. ..... 12

Figure 2.2: Schematic diagram for the Shunt-Resistor system ..... 14

Figure 2.3: Schematic diagram for the Switched-Capacitor system ..... 16

Figure 2.4: Switched-Capacitor system transferring energy from battery cell k to battery cell k+1 ..... 17

Figure 2.5: Schematic diagram for the Current-Diverter system ..... 19

Figure 2.6: Current-Diverter system diverting charging current around the $k^{th}$ battery cell ..... 20

Figure 2.7: Current-Diverter system diverting charging current around the first battery cell ..... 21

Figure 2.8: Schematic diagram for the Bidirectional Current-Diverter system ..... 23

Figure 2.9: Bidirectional Current-Diverter system diverting discharging current around the $k^{th}$ battery cell ..... 24

Figure 2.10: Schematic diagram for the Flyback-Converter system ..... 27

Figure 2.11: Flyback-Converter system transferring energy from the $k^{th}$ battery cell to the high-voltage bus ..... 28

Figure 2.12: Schematic diagram for the Bidirectional Flyback-Converter system ..... 30

Figure 2.13: Bidirectional Flyback-Converter system transferring energy from the high-voltage bus to the $k^{th}$ battery cell ..... 31

Figure 2.14: Schematic diagram for the Centralized Forward-Converter system ..... 33

Figure 2.15: The Centralized Forward-Converter system transferring current from the high-voltage bus to the individual battery cells ..... 34

Figure 2.16: Schematic diagram of the Hybrid System #1 ..... 36

Figure 2.17: The Hybrid System #1 discharging current from the $k^{th}$ battery cell ..... 37

Figure 2.18: The Hybrid System #1 redistributing energy from the $k^{th}$ battery cell to the other battery cells ..... 38

Figure 2.19: Schematic diagram of the Hybrid System #2 ..... 40

Figure 2.20: The Hybrid System #2 discharging current from the $k^{th}$ battery cell ..... 41

Figure 2.21: The Hybrid System #2 discharging current from the $N^{th}$ battery cell ..... 42

Figure 3.1: Schematic Diagram for the analysis of the Bidirectional Flyback-Converter system ..... 48

Figure 3.2: Voltage and current waveforms of the magnetizing inductance during a discharge sequence ..... 50

Figure 3.3: Voltage and current waveforms of the magnetizing inductance during a charging sequence ..... 54

Figure 3.4: Top-level circuit schematic of the CEC ..... 58

Figure 3.5: Gate drive schematic ..... 69

Figure 3.6: Single-pole, low-pass RC filter of the battery module voltage ..... 70

Figure 3.7: Bode magnitude and phase plots of the battery module voltage RC filter ..... 71

Figure 3.8: Single-pole, low-pass RC filter of the high-voltage bus voltage ..... 72

Figure 3.9: Bode magnitude and phase plots of the high-voltage bus voltage RC filter ..... 72

Figure 3.10: Circuit schematics of the current-signal active filters ..... 74

Figure 3.11: Bode magnitude and phase plots of the discharging-current active filter ..... 74

Figure 3.12: Bode magnitude and phase plots of the charging-current active filter ..... 75

Figure 3.13: Schematic diagram of the RS485 communication network ..... 76

Figure 3.14: Main loop ..... 83

Figure 3.15: Adjust peak-current cutoff function ..... 84

Figure 3.16: Microcontroller hello world ..... 87

Figure 3.17: UART bit timing ..... 88

Figure 3.18: Transmission of a packet on the RS485 bus..... 89

Figure 3.19: Computer to CEC packet transmission and reply ..... 90

Figure 3.20: SPI communication to the DAC ..... 91

Figure 3.21: Relationship of compensation pin voltage to DAC voltage ..... 92

Figure 3.22: Response of the battery module voltage filter to input signals with 10 Hz, 100 Hz, and 1 kHz frequencies ..... 94

Figure 3.23: Response of the high-voltage bus voltage filter to input signals with 10 Hz, 100 Hz, and 1 kHz frequencies ..... 96

Figure 3.24: Response of the average charging current filter to input signals with 2.5 Hz, 25 Hz, and 250 Hz frequencies ..... 98

Figure 3.25: Response of the average discharging current filter to input signals with 2.5 Hz, 25 Hz, and 250 Hz frequencies ..... 100

Figure 3.26: Mounting clamp used to machine the transformer core ..... 102

Figure 3.27: Machining the transformer core ..... 102

Figure 3.28: Tinning the litz wire ..... 104

Figure 3.29: Image of the wound bobbin and machined transformer cores ..... 104

Figure 3.30: Completed transformer core wired to the CEC ..... 105

Figure 4.1: Peak-current test setup..... 107

Figure 4.2: MOSFET idle snapshot and 20-A peak-current test snapshot ..... 108

Figure 4.3: MOSFET peak-current test with jumper snapshots at a 25-µs window.... 109

Figure 4.4: MOSFET peak-current test with jumper snapshots at a 2.5 µs window ... 110

Figure 4.5: IGBT peak-current test snapshots at a 25-µs window..... 112

Figure 4.6: IGBT peak-current test with jumper snapshots at a 25-µs window ..... 113

Figure 4.7: IGBT 1-A peak-current test with jumper snapshot at a 2.5-µs window ... 114

Figure 4.8: IGBT peak-current test with jumper snapshots at a 5-µs window ..... 114

Figure C.1: Input and output headers schematic ..... 151
Figure C.2: Power schematic ..... 152
Figure C.3: Chassis ground schematic ..... 153
Figure C.4: Charge and discharge current filters ..... 154
Figure C.5: Battery module voltage RC filter and MOSFET gate drive ..... 155
Figure C.6: High-voltage bus voltage RC filter and IGBT gate drive ..... 156
Figure C.7: Top of the PCB ..... 157
Figure C.8: Bottom of the PCB ..... 158

# LIST OF TABLES

Table 2.1: Summary of component characteristics for the Shunt-Resistor system ..... 14

Table 2.2: Summary of component characteristics for the Switched-Capacitor system ..... 16

Table 2.3: Summary of component characteristics for the Current-Diverter system .. 19

Table 2.4: Summary of component characteristics for the Bidirectional Current-Diverter system ..... 23

Table 2.5: Summary of component characteristics for the Flyback-Converter system ..... 27

Table 2.6: Summary of component characteristics for the Bidirectional Flyback-Converter system ..... 30

Table 2.7: Summary of component characteristics for the Centralized Forward-Converter system ..... 33

Table 2.8: Summary of component characteristics of the Hybrid System #1 ..... 36

Table 2.9: Summary of component characteristics of the Hybrid System #2 ..... 40

Table 2.10: Criteria weight factors for the Penn State research vehicles ..... 44

Table 3.1: Maximum duty cycles for various transformer winding ratios ..... 60

Table 3.2: Magnetizing inductance values required to achieve 10 A of average battery module current ..... 61

Table 3.3: Calculated duty cycles to achieve 10 A of average battery module current ..... 61

Table 3.4: Calculated current and voltage levels for the power electronic components ..... 62

Table 3.5: Power levels when transferring 10 A of average battery module current .. 63

Table 3.6: Selected electronic switches and diodes characteristics ..... 63

Table 3.7: Recommended values for a pot-core transformer with square-wave excitation ..... 65

Table 3.8: General communication message structure ..... 77

Table 3.9: Voltage and power requirements of the CEC ..... 80

Table 3.10: Advanced Circuits printed circuit board specifications ..... 85

Table 3.11: Measured output voltages of the on-board dc-to-dc converters ..... 86

Table C.1: Capacitor list ..... 159

Table C.2: Diode list ..... 160

Table C.3: Resistor list ..... 161

Table C.4: Electronic switch list ..... 162

Table C.5: Integrated circuit list ..... 162

Table C.6: Header list ..... 162

## ACKNOWLEDGEMENTS

My research was financially supported by the Mid-Atlantic Universities Transportation Center (MAUTSC), the Department of Environmental Protection (DEP), and the Pennsylvania Department of Community and Economic Development (DCED).

I would like to thank the Thomas D. Larson Pennsylvania Transportation Institute for providing the equipment and space that was required to conduct my research. I would also like to thank Dr. Anstrom for giving me the inspiration, means, and guidance to complete this work. Dr. Mayer, thank you for providing me with direction in my academic career and guidance in completing this work. Dr. Mitchell, thank you for taking the time to review this work and to serve on my thesis committee.

Last but not least, I would like to express my sincerest gratitude to my family. Without their support, I would not have been able to complete this journey.

# Chapter 1

# Introduction

Electrically-powered vehicles are an active area of research in the automotive industry and academic community. One area of electric-vehicle technology that is receiving considerable attention is energy sources. Electrochemical batteries are currently the prevalent energy source. They require a robust battery management system (BMS) with a specialized charge-equalization circuit (CEC) to operate safely and maintain their performance over the lifetime of the vehicle.

## 1.1 Automobile Industry Transition toward Electric Vehicles

The internal combustion engine (ICE) has been the powerplant of choice in the automobile for the greater part of the past century. The cost of ownership of these vehicles has been low enough that the majority of Americans could afford one, and their effect on the environment was not considered to be substantial. However, most major automobile manufacturers are currently investigating the use of an electric motor as either a supplement to or direct replacement of the ICE. The Toyota Prius, Chevy Volt, and Tesla Roadster are examples of the recent interest in the integration of the electric motor into the automobile powertrain. The driving factors in the on-going transition from the ICE to the electric motor are economic and environmental.

The energy of one gallon of gasoline and one kWh of electricity can be converted to 130,800,000 J and 3,600,000 J, respectively [1]. Therefore, the energy in one gallon of gasoline is equivalent to approximately 36.3 kWh of electricity. Recently, the United States national average price of one gallon of gasoline has reached a record level at over $3.50 [2]. In comparison, the United States national average cost of one kWh of electricity in 2007 was $0.1064 [3]. Therefore, it would cost approximately $3.86 to get the energy equivalent of one gallon of gasoline from electricity. From an energy standpoint, electricity currently costs about the same amount as gasoline. However, studies have found that the maximum efficiency in converting latent chemical energy of fuel to propulsion for the ICE and its driveline was 15%, and the maximum efficiency for the electric motor with gear reduction was in excess of 90% [4][5]. The actual efficiencies that are achieved by current electric vehicles are substantially lower, but this is due to the limitations in energy storage. While energy storage is preventing the electric motor from being the current automotive powerplant of choice, it is clear that with energy source improvements, the efficiency of the electric motor has the potential to significantly exceed that of the ICE. Consequently, research is on-going in developing electric motor technology to increase the energy efficiency, and therefore lower the total cost of ownership, of the automobile.

The second significant contributor to the reduction of interest in the ICE is its environmental impact. Automobile exhaust pollutants consist of carbon monoxide, hydrocarbons, and nitrogen oxide. It also is a major contributor of the carbon dioxide that is released into the environment [6]. Carbon monoxide reduces the flow of oxygen in the bloodstream and is particularly dangerous to people with heart disease. Some

hydrocarbons are toxic and have the potential to be carcinogenic. Nitrogen oxide reacts with hydrocarbons to cause ozone and also contributes to the formation of acid rain. Finally, carbon dioxide is considered a greenhouse gas that contributes to the potential for global warming.

The Environmental Protection Agency estimates that automobiles contribute about 60% of the carbon monoxide, 44% of the hydrocarbons, and 31% of the nitrogen oxides emitted in the United States [7]. While the federal government is slow in bringing change to its vehicle emission standards, individual states are beginning to take action to reduce it. In 2004 California had set its own emissions standards that are aimed at reducing air pollution from cars and light trucks by 25% and from sport utility vehicles by 18%. California has also filed suit against General Motors, Toyota, Ford, Honda, Chrysler and Nissan in an attempt to hold them accountable for the greenhouse gases that their vehicles produce [8]. There is a growing awareness of the impact of the ICE on the environment and a mounting push for cleaner alternatives.

To conform to the tougher emission regulations, automobile manufacturers are turning to the electric motor because it produces zero local emissions. One study has found that, in comparison to the ICE emissions, the generation of energy for automotive electric motors results in the emissions savings of 2% of carbon monoxide, 76% of carbon dioxide, 56% of nitrogen oxides, and 9% of hydrocarbons [9]. Therefore, even at the global level, the electric motor has shown it has the potential to make a much smaller impact on the environment then the ICE.

## 1.2 Batteries and Battery Management Systems

The main barrier that prevents the electric motor from becoming the automobile powerplant of choice is the lack of an acceptable energy source. To allow the motor to do a reasonable amount of work, the energy source is required to have a high energy density. Greater potential to do work results in a longer range of distance traveled for an electric vehicle or a series hybrid-electric vehicle, and it results in a higher fuel efficiency for a parallel hybrid-electric vehicle. It also must have a high power density to satisfy the peak power demand of the electric motor. The charge time has to be kept to a minimum. It is paramount that its operation be safe. Finally, it should be maintenance free and environmentally friendly [10]. The difficulty in finding an appropriate solution is that energy sources generally excel in one characteristic or another, but not all.

At the present time, electrochemical batteries are often the energy source that is utilized. However, an individual battery cell outputs a voltage level that is too low to reasonably meet the power demand of automotive electric motors. Automobiles must have a high-power powerplant to satisfy the consumers' demand for acceleration. Power ratings of currently used electric motors are in the range from 50 kW to 185 kW, as found in the efficiency-inspired 2008 Toyota Prius and the performance-inspired 2008 Tesla Roadster, respectively [11][12]. To increase the power output that can produced from batteries, a large number of battery cells are placed in series to create a high-voltage battery pack. The high voltage of the battery pack allows it to meet the power demand of the electric motor at a reasonable current level. Efficiencies are increased because the

power losses due to resistance are decreased. The battery pack voltage can be 360 V or even higher [13].

The state of charge (SOC) of a battery cell is defined as the ratio of the amount of remaining, usable energy to its completely-charged, usable energy. The charge efficiency of a battery cell is defined as the ratio of its discharged A·h to its charged A·h [10]. Due to the series configuration of the pack, all of the battery cells conduct the same current that is demanded from the electric motor or is supplied from the charging unit. However, manufacturer tolerances and uneven battery pack temperatures will result in varying charge efficiencies of the battery cells [14]. As the battery pack is used over time, the different charge efficiencies will result in battery cells with varying SOC.

Without a BMS, the individual battery cell voltages of the battery pack must be approximated by an average value that is calculated from the overall voltage. This results in errors when monitoring the SOC of the individual battery cells. During a discharge sequence these errors can lead to some battery cells that are over-discharged, and during a charge sequence it can lead to some battery cells that are overcharged. Cycling a battery cell beyond its rated limit may damage the cell. The damage may consist of capacity loss, a higher self-discharge rate, and lower charge efficiency [14]. Extreme cases may lead to a thermal runaway condition, which can result in a gas fire or explosion [15]. A BMS must monitor the state of charge of the individual battery cells and prevent an overcharge or over-discharge event from occurring.

The battery pack can only be discharged until the battery cell with the lowest SOC reaches its cutoff limit. Likewise, it can only be charged until the battery cell with the highest SOC reaches its cutoff limit. This lowers the useable amount of energy in the

battery pack as the variability in the battery pack increases. As a result, the contribution that the electric motor can make in powering the vehicle is reduced. One study had found that after 60 cycles of a battery pack, it could only produce 58% of its rated energy before its lowest SOC battery cell had reached its cutoff limit [14]. Consumers expect long lifetimes and consistent performance from their automobiles. Therefore, the BMS must also have a CEC that can effectively correct the SOC variability that develops within the battery pack.

### 1.3 Penn State Research Vehicles

Two hybrid electric vehicles are currently under development at Penn State. One vehicle, the Electric-Lion, is based on a 1992 Ford Escort chassis. It is considered a range-extending, series, hybrid-electric vehicle. The torque at the wheels is generated from the electric motors only. It employs two Solectria, 35-kW, ac induction motors to power the front wheels. A Kawaski, 12-kW, two-piston ICE and generator are used to charge the battery pack. The 144-V battery pack consists of 12 SAFT (SAFT is a manufacturer of batteries), 12-V nominal, 100-A·h nickel-metal hydride (NiMH) battery modules. A picture of the Electric-Lion and its battery pack are shown in Figure 1.1.


Figure 1.1: Penn State Electric-Lion vehicle and battery pack

The second vehicle, the HyLion, is based on a 1997 General Motors EV1 chassis. It is also considered a range-extending, series, hybrid-electric vehicle. Similarly to the Electric-Lion, the HyLion propulsion is produced from the electric motor only. It employs a 100-kW, ac induction motor to power the front wheels. A Ballard (Ballard is a manufacturer of Fuel Cells), 1.2-kW, hydrogen fuel cell is used to charge the battery pack. The 300-V battery pack consists of 25 Panasonic, 12-V nominal, 95-A·h, NiMH battery modules. A picture of the HyLion and its battery pack are shown in Figure 1.2.

Figure 1.2: Penn State HyLion vehicle and battery pack

Neither vehicle has a BMS; they both are currently operated based on an estimation of battery cell voltage. The current BMS under development has a centralized, master controller that is programmed using MATLAB's Simulink software. This simplifies the software development that is required, and it allows the BMS to be used as an instructional tool for students that span across multiple disciplines. The master controller monitors the SOC of the battery modules, instructs the CEC in how to manage the energy variability in the battery pack, and communicates to the vehicle controller. The scope of the research presented here is the analysis, design, and implementation of the CEC for the developing BMS.

### 1.4 Organization of Thesis

The thesis is organized into five chapters. The next chapter presents evaluation criteria and general system structures of CECs. Systems presented in the literature are investigated in how well they perform against the evaluation criteria, and a system is chosen for the Penn State research vehicles. The focus of Chapter 3 will be the analysis, design, and implementation of the selected CEC. In Chapter 4, the implemented CEC will be tested, the results and observations presented, and recommendations for future work will be made. Finally, in Chapter 5, the thesis will be summarized.

# Chapter 2

# Candidate Solutions

A Charge-Equalization Circuit (CEC) allows a battery management system (BMS) to maintain an appropriate distribution of energy throughout a battery pack. Many different CECs have been presented in the literature, and the particular application must be analyzed to determine which one is optimal. Several different CEC systems are examined in this chapter, and one is then selected for the Penn State research vehicles.

## 2.1 Evaluation Criteria

Several criteria have been determined to evaluate the various CECs. Applications must weight the different criteria to determine which CEC is optimal.

- **Equalization Power**: The charging and discharging sequences of the battery pack can be large in magnitude and short in duration, which can make the equalization process difficult. The CEC must have the power capability to be able to eliminate the state of charge (SOC) variability of the battery pack. This will allow a vehicle controller to utilize the full energy potential of the battery pack continuously. All systems are evaluated in the context of equalizing battery cells with a common, quasi-static voltage value. Therefore, power capability of each CEC is directly proportional to the maximum average value of current that it can transfer.

- **Modular Setup**: A modular setup facilitates assembly and repair, as well as adaptability to battery packs of different size.
- **Simple/Robust Control Strategy**: The system should not be overly complex and should have a low probability of failure.
- **Expense**: Relative to the battery cells it equalizes and the cost of the automobile, the implementation cost should be kept to a minimum.
- **Equalization at the Battery Cell Level**: Charge efficiencies can vary from battery cell to battery cell. This leads to SOC variation between neighboring battery cells. The system should have the resolution to remove this variability.
- **High SOC and Low SOC Equalization**: Battery cells can develop SOC variation that is higher or lower than average. To achieve maximum efficiency, a system should have the ability to equalize both variations in SOC. For example, consider a system that can only equalize battery cells that have higher-than-average SOC. When one cell has a lower-than-average SOC, the system will have to decrease the energy in all other battery cells to equalize the battery pack. This would result in a substantial amount of energy that is lost in the equalization process.

- **Even Energy Redistribution**: When equalizing a battery cell, the energy should be redistributed evenly throughout the battery pack. The equalization process should not induce SOC variability in the other battery cells.

## 2.2 General System Structures

Several different systems have been presented in the literature. They can be separated into series, parallel and hybrid structures. The connectivity of the respective series, parallel and hybrid structures is shown in Figure 2.1. The series systems have the circuits arranged in series with each other, and they distribute energy to (from) a battery cell along the natural flow of energy within the battery pack. The parallel systems have the circuits arranged in parallel to each other, and they distribute energy to (from) a battery cell in a path that is perpendicular to the natural flow of the battery pack energy. Hybrid configurations utilize circuit characteristics from both series and parallel systems in an attempt to capture the benefits of both.

Systems that incorporate dc-to-dc converters are typically implemented in the discontinuous conduction mode, which helps to simplify the control schemes. Consequently, the analyses presented here will also be for the discontinuous conduction mode.

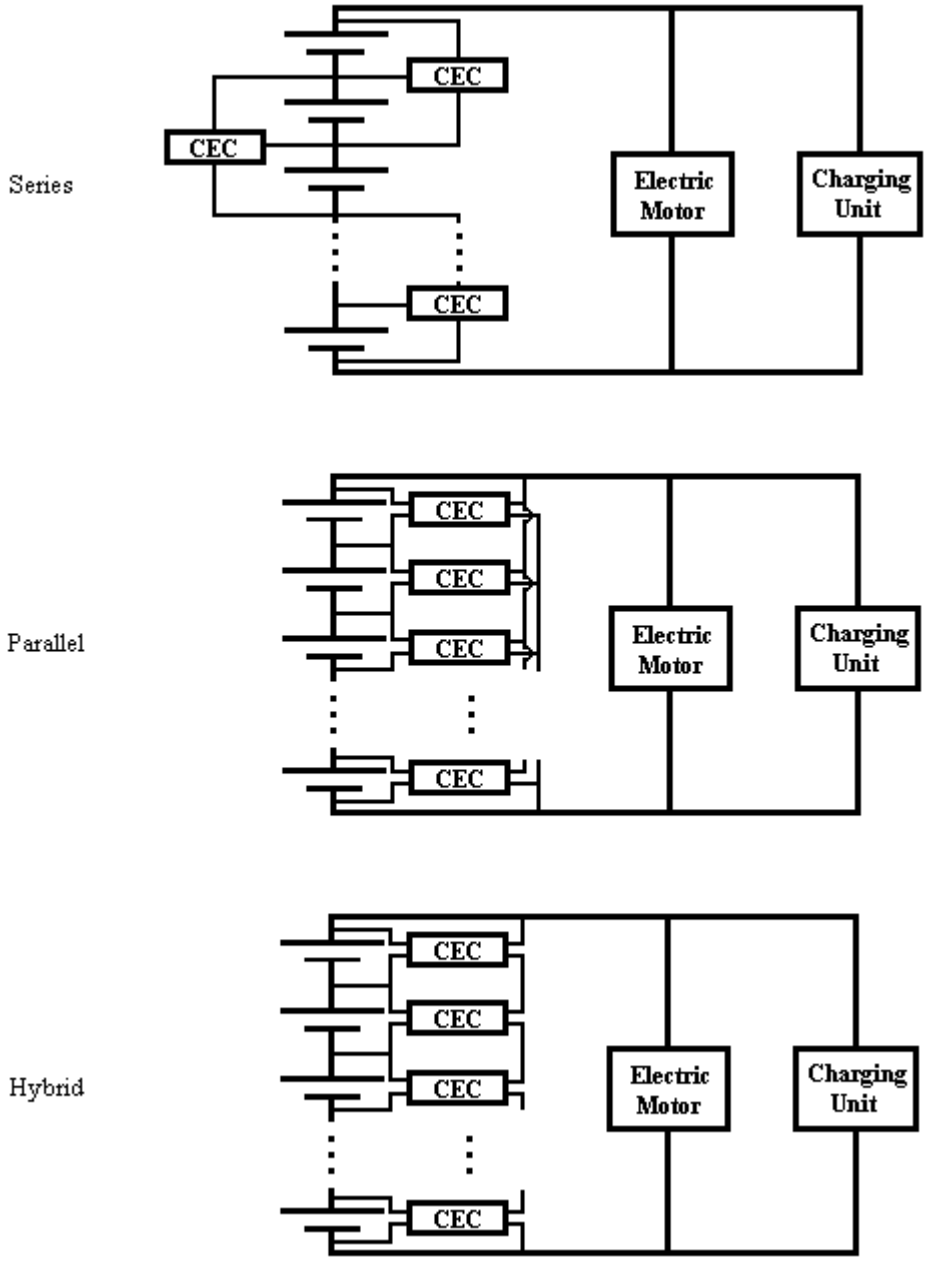



Figure 2.1: General classification of BMS structures.

## 2.3 Series Systems

Series systems often utilize simpler circuit structures and require less wiring than the parallel and hybrid systems. The tradeoff is the distribution of energy to (from) a battery cell is redistributed unevenly (if redistributed at all) among the remaining battery cells. These systems are advantageous in terms of cost and are well suited for space-sensitive applications that do not require large amounts of energy to be redistributed.

### 2.3.1 Shunt-Resistor System

The only dissipative system, the Shunt-Resistor system equalizes SOC imbalance by dissipating charging energy through a switched, power resistor [16][17]. During a charging cycle, the system can shunt current around battery cells with higher-than-average SOC and allow the other battery cells to continue to charge. For $N$ battery cells, the system consists of $N$ electronic switches (typically MOSFETs) and $N$ power resistors. A schematic diagram is shown in Figure 2.2, and a summary of component characteristics is given in Table 2.1.

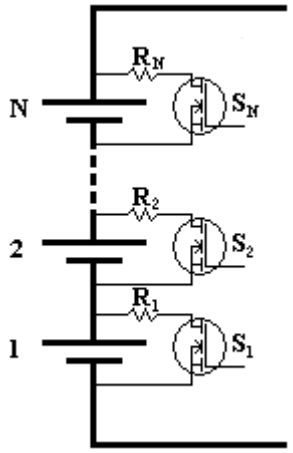



Figure 2.2: Schematic diagram for the Shunt-Resistor system

Table 2.1: Summary of component characteristics for the Shunt-Resistor system

| Description | Number | Voltage Rating |
|---|---|---|
| MOSFET | N | $V_{cell}$ |
| Power Resistor | N | $V_{cell}$ |

To examine how the Shunt-Resistor system equalizes SOC imbalance, consider the situation when the $k^{th}$ battery cell has a higher-than-average SOC. During a charging sequence, the BMS acts to shunt the charging current around that battery cell. The other battery cells continue to receive the charging current. The average current $I_k$ that is shunted around the $k^{th}$ battery cell is approximated by Equation 2.1, where $v_k$ is the voltage of the $k^{th}$ battery cell, $D_k$ is the duty cycle of $S_k$, and $R_k$ is the resistance of the resistor.

$$I_k = \frac{v_k D_k}{R_k} \quad k = 1, 2, \ldots, N \tag{2.1}$$

The Shunt-Resistor system has a completely modular structure. In addition, it uses only a few, off-the-shelf components. This results in a system with a low implementation cost, and is applicable for equalization at the battery cell level. Finally, it does not induce SOC variability in the other battery cells because all excess energy is dissipated in the power resistors.

However, a primary motivator in the use of the electric motor is to increase the energy efficiency of the automobile. Since the system dissipates all of the excess energy as heat in power resistors, a shunt-resistor system is typically designed with a low power capacity. The control scheme is moderately complex, with gate drive and voltage feedback signals required for each battery cell. Finally, it only has the ability to equalize battery cells that have higher-than-average SOC.

### 2.3.2 Switched-Capacitor System

The Switched-Capacitor system equalizes SOC imbalance by charging a capacitor up to the voltage of a battery cell and discharging the energy into a neighboring battery cell with a lower voltage [18][19][20]. For $N$ battery cells, the system consists of $N$ single-pole, double-throw (SPDT) switches and $N$-1 capacitors. A schematic diagram is shown in Figure 2.3, and a summary of component characteristics is given in Table 2.2.

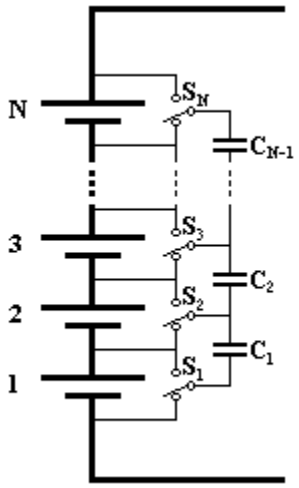



Figure 2.3: Schematic diagram for the Switched-Capacitor system

Table 2.2: Summary of component characteristics for the Switched-Capacitor system

| Description | Number | Voltage Rating |
|---|---|---|
| SPDT Switch | N | $V_{cell}$ |
| Capacitor | N-1 | $V_{cell}$ |

To evaluate how the Switched-Capacitor system equalizes SOC imbalance, consider the situation when battery cell $k$ has a higher voltage than the battery cell $k$+1. When the switches are in the lower position, battery cell $k$ charges capacitor $C_k$ up to its voltage $v_k$, with an average current $I_k$. When the switches are in the upper position, battery cell $k$+1 discharges capacitor $C_k$ down to its voltage $v_{k+1}$, with an average current $I_{k+1}$. The reference directions of the average currents are shown in Figure 2.4, where $R_{eq}$ is the equivalent resistance of the wire, switches and capacitor. The net average current between the two adjacent cells, $I_{k,k+1}$ is approximated by Equation 2.2 [20], where $f_s$ is the switching frequency. Without external control or a feedback loop, this current decreases

with a decreasing voltage difference. It is zero at the limit when the battery cells achieve the same voltage.

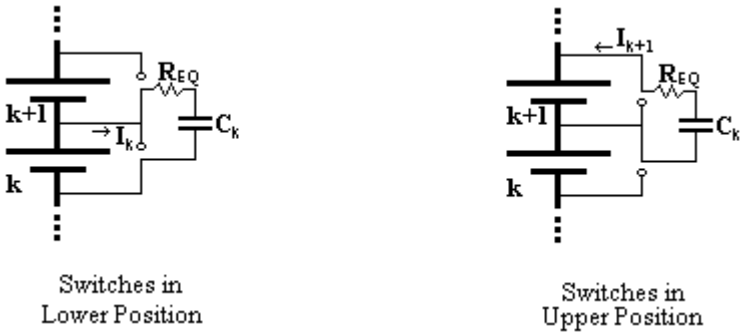



Figure 2.4: Switched-Capacitor system transferring energy from battery cell k to battery cell k+1.

$$I_{k,k+1} = \frac{v_k - v_{k+1}}{\frac{1}{f_s C_k} + R_{eq}} \quad k = 1, 2, \ldots, N-1 \tag{2.2}$$

The Switched-Capacitor system has a completely modular setup. In addition, it utilizes only a few, off-the-shelf components. Also, it is intended to be operated continuously, thereby removing the need for a supervisory control system. This results in a system with low implementation cost that is applicable for equalization at the battery-cell level. Finally, it has the ability to equalize battery cells with both higher-than-average and lower-than-average voltages.

However, the equalization power is limited. Practical current rates of only Capacity/(500 h) and Capacity/(2000 h) (Capacity is the nominal capacity of the battery cell in A·h) are achievable at a voltage differential of 10 mV per cell [20]. In addition, the system only distributes recovered energy to neighboring battery cells. Because of its

continuous operation the energy is eventually distributed across the pack, but at the cost of compounding efficiencies. Finally, equalization is only performed when neighboring battery cells have different voltages. Some battery chemistries exhibit very small changes in voltage across large spans of SOC. When these batteries do exhibit a significant voltage difference, there could be a large variation in energy, and this system might not have the power capacity to eliminate it in a timely manner.

### 2.3.3 Current-Diverter System

The Current-Diverter system equalizes SOC imbalance by diverting charging current around an unbalanced battery cell [21]. For *N* battery cells, the system is composed of *N*-1 buck-boost converters and one flyback converter. Its components consist of *N* electronic switches (typically MOSFETs), *N*-1 inductors, *N* diodes, and one transformer. A schematic diagram is shown in Figure 2.5, and a summary of component characteristics is given in Table 2.3.

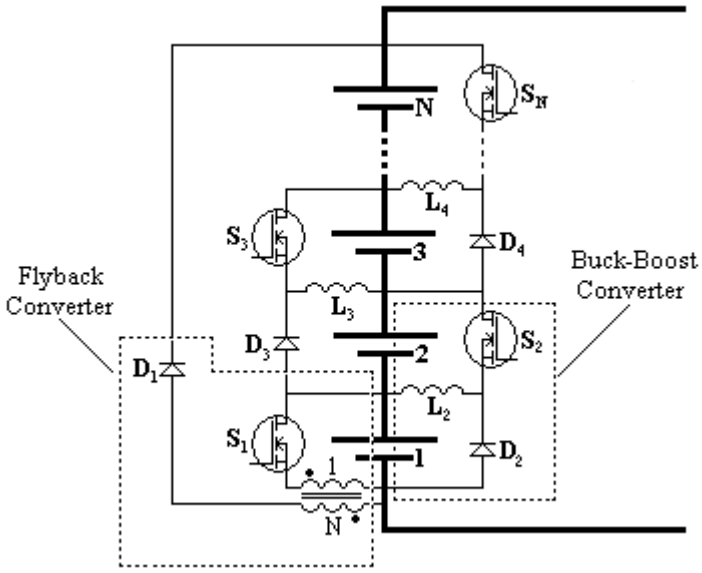



Figure 2.5: Schematic diagram for the Current-Diverter system

Table 2.3: Summary of component characteristics for the Current-Diverter system

| Description | Number | Voltage Rating |
|---|---|---|
| MOSFET | N-1 | $2*V_{cell}$ |
| Diode | N-1 | $2*V_{cell}$ |
| Diode | 1 | $2*N*V_{cell}$ |
| Transformer | 1 | 1:N turns ratio |
| Inductor | N-1 | n/a |

To evaluate how the Current-Diverter system equalizes SOC imbalance, consider the circuit diagram and current waveforms in Figure 2.6. In this case, the $k^{th}$ battery cell has a higher-than-average SOC. To compensate for this, the BMS acts to divert the average charging current $I_{ch}$ around that cell. This prevents the battery cell from being charged while the remaining cells continue to be charged. Note that any portion of $I_{ch}$

could be diverted; for this example it has been chosen to be equal to the charging current. Therefore, the $k^{th}$ battery cell will not receive any charging current, while the other battery cells will continue to charge as normal.

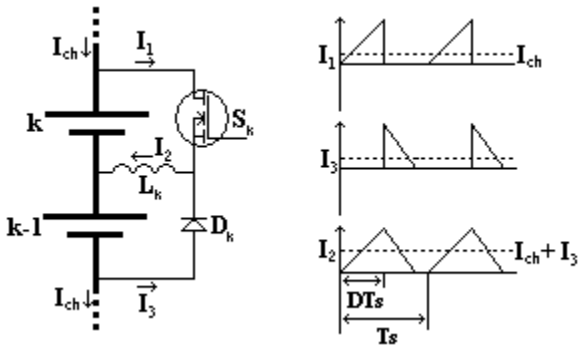



Figure 2.6: Current-Diverter system diverting charging current around the k$^{th}$ battery cell

Initially, $S_k$ is switched on until its average current $I_1$ is equal to $I_{ch}$. The average current $I_1$ is defined by Equation 2.3, where $v_k$ is the voltage of the $k^{th}$ battery cell, $L_k$ is the inductance of the inductor, and $D_k$ is the duty cycle of $S_k$. $S_k$ is then switched off, and the freewheeling diode conducts current until the inductor is demagnetized. The average charging current of the $k^{th}$ battery cell is the difference between $I_{ch}$ and $I_1$, and in this case results in zero amps.

$$I_1 = \frac{v_k D_k}{2L_k} \qquad \text{k} = 2, 3, \ldots \text{N} \qquad 2.3$$

The average current of the inductor $I_2$ is the sum of currents $I_1$ and the average current of the diode $I_3$. The average charging current of the $k$-1 battery cell is the sum of $I_{ch}$ and $I_3$, and the charging currents of the remaining battery cells are unchanged at $I_{ch}$.

This analysis applies to all battery cells except the first battery cell of the battery pack. To evaluate how the Current-Diverter system equalizes SOC imbalance in the first battery cell, consider the circuit diagram and waveforms in Figure 2.7. In this case, the first battery cell has a higher-than-average SOC. The BMS acts to correct this by diverting the charging current $I_{ch}$ around it. Note that any portion of $I_{ch}$ can be diverted; in this example it has been chosen to be all of the charging current. Therefore, the first battery cell will not receive any current from $I_{ch}$, while the remaining battery cells will continue to receive $I_{ch}$.

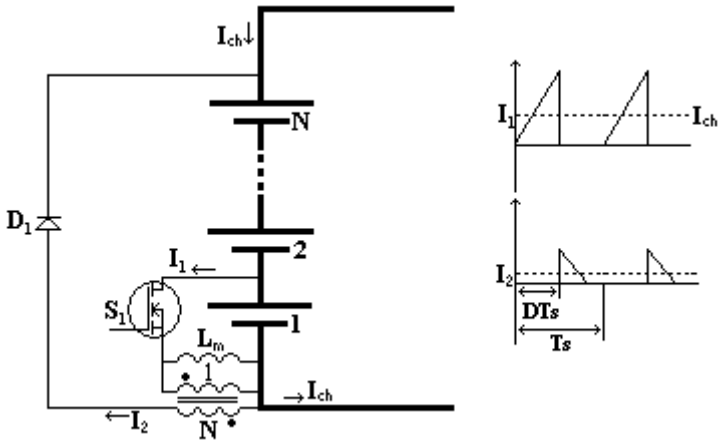



Figure 2.7: Current-Diverter system diverting charging current around the first battery cell

Initially, $S_1$ is switched on until its current reaches an average current level $I_1$, which is set to be equal to $I_{ch}$. The average current $I_1$ is approximated by Equation 2.4, where $v_1$ is the voltage of the first battery cell, $D_1$ is the duty cycle of $S_1$, and $L_m$ is the magnetizing inductance of the transformer. $S_1$ is then switched off, and the freewheeling diode will conduct current until the transformer is demagnetized. The average current through the diode $I_2$, where $|I_{ch}| > |I_2|$, is then redirected back into the battery pack. The

net average charging current of the first battery cell is equal to $I_2$, and the net average charging currents for the other battery cells are the sum of $I_{ch}$ and $I_2$.

$$I_1 = \frac{v_1 D_1}{2L_m} \tag{2.4}$$

The power capability of the Current-Diverter system is greater then the previous two systems discussed, and through the proper selection of components it can be tailored to most applications. In addition, the system has a relatively modular setup; only the flyback converter must be designed for a specific battery pack. Finally, the system uses a moderate number of off-the-shelf components, with only the transformer being a custom component.

However, the control scheme is moderately complex. It requires a gate drive signal and a current feedback signal for each battery cell. The complexity in the control scheme and the moderate cost of the power electronic components make this system applicable for equalization down to groups of battery cells only (battery-module level). In addition, equalization can only be performed on battery cells with higher-than-average SOC. Finally, the redistribution of recovered energy is poor. For the buck-boost converters, the energy is redistributed to the neighboring battery cell only. This causes the neighboring battery cell to charge faster then the other battery cells in the pack, which may exacerbate the SOC variation.

### 2.3.4 Bidirectional Current-Diverter System

An adaptation to the Current-Diverter system, the Bidirectional Current-Diverter system also equalizes SOC imbalance by diverting current around a battery cell. Both systems can divert current around a battery cell during a charging sequence; however, the bidirectional system can also divert current around a battery cell during a discharging sequence [21]. For *N* battery cells, the system is composed of *N*-2 bidirectional, buck-boost converters. Its components consist of 2*N*-2 electronic switches (typically MOSFETs), 2*N*-2 diodes, and *N*-1 inductors. A schematic diagram is shown in Figure 2.8, and a summary of component characteristics is given in Table 2.4.

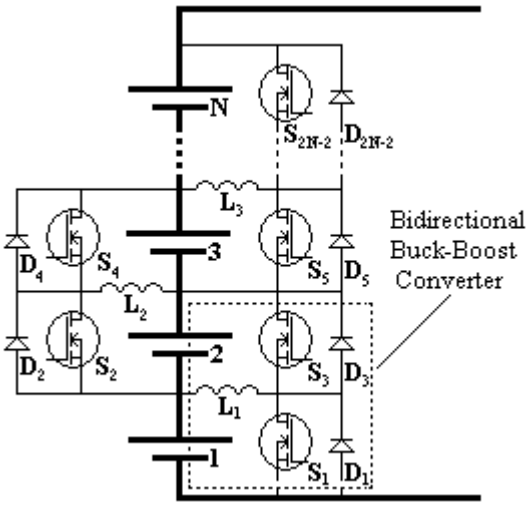



Figure 2.8: Schematic diagram for the Bidirectional Current-Diverter system

Table 2.4: Summary of component characteristics for the Bidirectional Current-Diverter system

| Description | Number | Voltage Rating |
|---|---|---|
| MOSFET | 2N-2 | 2* $V_{cell}$ |
| Diode | 2N-2 | 2* $V_{cell}$ |
| Inductor | N-1 | n/a |

When diverting current around an inner battery cell with a higher-than-average SOC, the circuit analysis is the same as the Current-Diverter system. The Bidirectional Current-Diverter system has the added feature that it can divert discharging current around an inner battery cell that has a lower-than-average SOC. To analyze the ability of the system to divert discharging current, consider the circuit diagram and current waveforms in Figure 2.9.

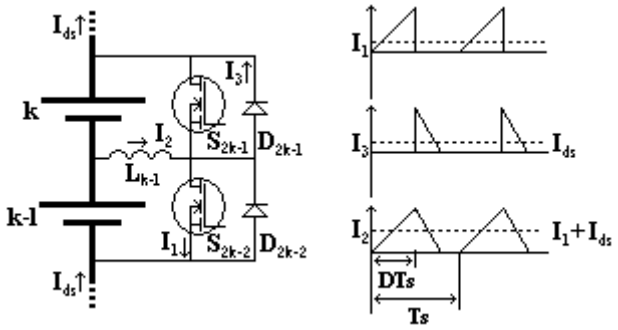



Figure 2.9: Bidirectional Current-Diverter system diverting discharging current around the $k^{th}$ battery cell

In this case, the $k^{th}$ battery cell has a lower-than-average SOC. The BMS will act to correct this by diverting the discharging current $I_{ds}$ around it. This allows the battery cell to maintain its energy while the remaining battery cells continue to discharge. Note that any portion of $I_{ds}$ could be diverted around the battery cell; for this example, it will be the entire discharging current. This will allow the battery cell to maintain all of its energy while the remaining battery cells continue to discharge.

Initially, $S_{2k-2}$ is switched on, which generates the average current $I_1$. It is then switched off, and the freewheeling diode conducts current until the inductor is demagnetized. The net average discharge current on the $k$-1 battery cell is the sum of $I_1$ and $I_{ds}$.

The average current of the diode $I_3$ is set to $I_{ds}$. The average current $I_3$ is approximated by Equation 2.5, where $v_k$ is the voltage of the $k^{th}$ battery cell, $v_{k\text{-}1}$ is the voltage of the $k$-1 battery cell, $D_{2k\text{-}2}$ is the duty cycle $S_{2k\text{-}2}$, $L_{k\text{-}1}$ is the inductance of the inductor, and $T_s$ is the switching frequency. The net average current being discharged from the $k^{th}$ battery cell is equal to the difference between $I_3$ and $I_{ds}$, which results in zero amps.

$$I_3 = \frac{v_{k-1}^2 D_{2k-2}^2 T_s}{2 v_k L_{k-1}} \quad k = 1, 2, \ldots, N-1 \qquad 2.5$$

The power capability of the Bidirectional Current-Diverter system matches the power capability of the Current-Diverter system, and likewise can be tailored to most applications. In addition, the system uses off-the-shelf components in a modular setup.

However, the control scheme is complex. It requires two gate drive signals and two current feedback signals per battery cell. The complexity in the control scheme and the cost of the components make this system applicable for equalization at only the battery-module level. In addition, the flexibility in current diversion is not completely bidirectional. The first battery cell can only be equalized when it has a higher-than-average SOC, not lower. Likewise, the last battery cell can only be equalized when it has a lower-than-average SOC. Finally, the recovered energy is only distributed to or from the neighboring battery cells, which may exacerbate the SOC variation.

## 2.4 Parallel Systems

Parallel systems provide the most even distribution of the recovered energy. The tradeoff is that they require custom transformers to provide galvanic isolation between the respective battery-cell grounds and the high-voltage bus ground. They also require more wiring than the series systems. These systems are advantageous in applications that have high-energy battery packs and require significant amounts of energy redistribution.

### 2.4.1 Flyback-Converter System

The Flyback-Converter system equalizes SOC imbalance by taking energy from the individual battery cells and redistributing it to the high voltage bus [22]. For $N$ battery cells, the system consists of $N$ transformers, $N$ electrical switches (typically MOSFETs), and $N$ diodes. The schematic diagram is shown in Figure 2.10, and a summary of component characteristics is given in Table 2.5.

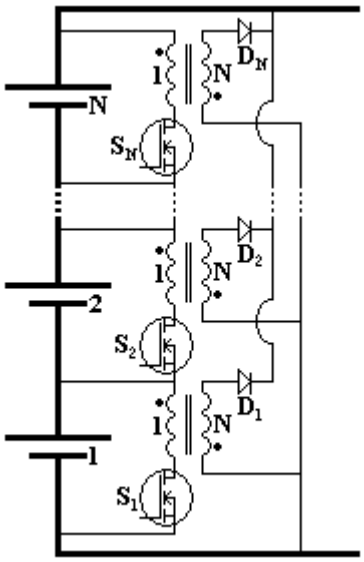



Figure 2.10: Schematic diagram for the Flyback-Converter system

Table 2.5: Summary of component characteristics for the Flyback-Converter system

| Description | Number | Voltage Rating |
|---|---|---|
| MOSFET | N | $2*V_{cell}$ |
| Diode | N | $2*N*V_{cell}$ |
| Transformer | N | 1:N turns ratio |

The Flyback-Converter system has the ability to transfer energy from a battery cell with a higher-than-average SOC to the high-voltage bus. The recovered energy can be used to charge the entire battery pack or applied to the electric motor, depending on the current state of the vehicle. To analyze the ability of the system to transfer energy from a battery cell to the high-voltage bus, consider the circuit diagram and current waveforms in Figure 2.11.

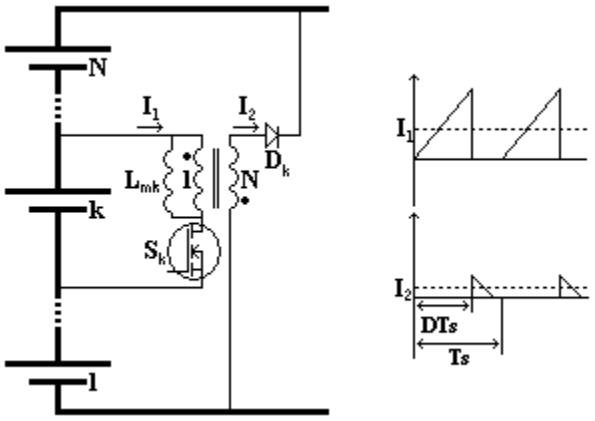



Figure 2.11: Flyback-Converter system transferring energy
from the k$^{th}$ battery cell to the high-voltage bus

In this case, the $k^{th}$ battery cell has a higher-than-average SOC. The BMS acts to correct this by transferring energy from it to the high-voltage bus. It has determined that an average discharge current $I_1$ from the $k^{th}$ battery will equalize the battery pack in a reasonable amount of time. Initially, $S_k$ is switched on until its average current reaches $I_1$. $I_1$ is approximated by Equation 2.6, where $v_k$ is the voltage of the $k^{th}$ battery cell, $D_k$ is the duty cycle of $S_k$, and $L_k$ is the magnetizing inductance of the transformer. $S_k$ is then switched off, and the freewheeling diode conducts current until the transformer is demagnetized. The average current of the diode $I_2$, where $|I_1| > |I_2|$, is transferred to the high-voltage bus.

$$I_1 = \frac{v_k D_k}{2L_k} \quad k = 1, 2, \ldots, N \tag{2.6}$$

During a discharging sequence, $I_2$ supplements the discharging current of the battery pack in powering the electric motor. In an idle or charging sequence, $I_2$ is redirected back into the battery pack. An idle battery pack results in the $k^{th}$ battery being

charged with the difference between $I_1$ and $I_2$, while the other battery cells are charged with $I_2$. A charging sequence results in the $k^{th}$ battery cell being charged with $I_{ch} + I_2 - I_1$, while the other battery cells are charged with the sum of $I_{ch}$ and $I_2$.

The Flyback-Converter system has the power capability that can be tailored to most applications. In addition, the setup is completely modular. Finally, it evenly redistributes the recovered energy by transferring it directly to the high-voltage bus.

However, the control scheme is moderately complex. It requires a gate drive signal and current feedback signal for each battery cell. In addition, it uses a custom designed transformer and a moderate amount of power electronics. The moderate control complexity and component cost make this system applicable for equalization at the battery module level. Finally, equalization can only be performed on battery modules with a higher-than-average SOC.

### 2.4.2 Bidirectional Flyback-Converter System

The Bidirectional Flyback-Converter system equalizes SOC imbalance by taking energy from (to) a battery cell and transferring it to (from) the high-voltage bus [23]. For $N$ battery cells, the system consists of $2N$ electronic switches (typically $N$ MOSFETs and $N$ IGBTs), $2N$ diodes, and $N$ transformers. A schematic diagram is shown in Figure 2.12, and a summary of component characteristics is given in Table 2.6.

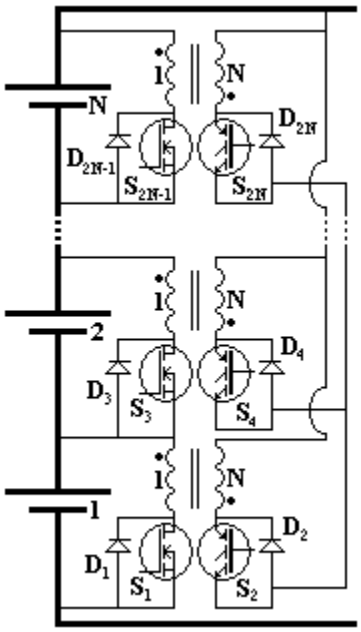



Figure 2.12: Schematic diagram for the Bidirectional Flyback-Converter system

Table 2.6: Summary of component characteristics for the Bidirectional Flyback-Converter system

| Description | Number | Voltage Rating |
|---|---|---|
| MOSFET | N | $2*V_{cell}$ |
| Diode | N | $2*V_{cell}$ |
| IGBT | N | $2*N*V_{cell}$ |
| Diode | N | $2*N*V_{cell}$ |
| Transformer | N | 1:N turns ratio |

The system has the ability to transfer energy from the high-voltage bus to a battery cell a lower-than-average SOC, and to transfer energy from a battery cell with a higher-than-average SOC to the high-voltage bus. When equalizing a battery cell with a higher-than-average SOC the circuit analysis is the same as the Flyback-Converter system. To

analyze its ability to equalize a battery cell with a lower-than-average SOC, consider the circuit diagram and current waveforms in Figure 2.13.

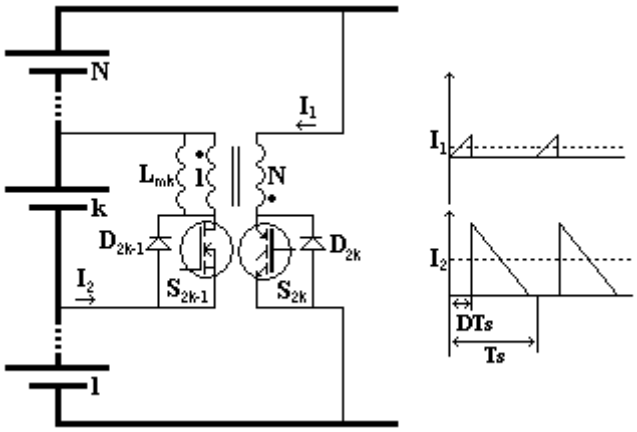



Figure 2.13: Bidirectional Flyback-Converter system transferring energy from the high-voltage bus to the $k^{th}$ battery cell

In this case, the $k^{th}$ battery cell has a lower-than-average SOC. The BMS acts to correct this imbalance by transferring energy from the high-voltage bus to this battery cell. It has determined that the average charging current $I_2$ will equalize the imbalance of the battery pack in a reasonable amount of time. First, $S_{2k}$ is switched on until its average current reaches $I_1$. It is then switched off, and the battery-cell side freewheeling diode conducts current until the transformer is demagnetized. $I_2$ is approximated by Equation 2.7, where $v_{pk}$ is the voltage of the battery pack, $v_{cell}$ is the voltage of the battery cell, $N$ is number of cells in the battery pack, $L_k$ is the magnetizing inductance of the transformer, and $D_k$ is the duty cycle of the IGBT.

$$I_2 = \frac{v_{pk}^2 D_k^2 T_s}{2N^2 v_{cell} L_k} \quad k = 1, 2, \ldots, N \qquad 2.7$$

When the battery pack is in a discharging state with an average current $I_{ds}$, where $|I_{ds}| > |I_2| > |I_1|$, the net average discharging current of the $k^{th}$ battery cell is $I_{ds} + I_1 - I_2$. The net average discharging currents of the other battery cells are the sum of $I_{ds}$ and $I_1$. When the battery pack is idle, the net average charging current of the $k^{th}$ battery cell is the difference between $I_2$ and $I_1$. The average discharging currents of the other battery cells are $I_1$. When the battery pack is charging with an average charging current $I_{ch}$, the net average charging current of the $k^{th}$ battery cell is $I_{ch} + I_2 - I_1$. The net average charging currents of the other battery cells are the difference between $I_{ch}$ and $I_1$.

The power capability of the Bidirectional Flyback-Converter system can be tailored to most applications. In addition, the setup is completely modular. Also, battery cells with higher-than-average SOC and lower-than-average SOC can be equalized. Finally, the recovered energy is evenly redistributed to (from) the high-voltage bus.

However, the system requires custom-designed transformers, and a significant amount of power electronics. In addition, the control scheme is complex. It requires two gate drive signals and two current feedback signals for each battery cell. The complexity in the control scheme and the high cost of its components results in the system being applicable for equalization at the battery-module level.

### 2.4.3 Centralized Forward-Converter System

The Centralized Forward-Converter system equalizes SOC imbalance by transferring energy from the high-voltage bus to the battery cells [24][25]. For $N$ battery cells, the system consists of 2 electronic switches (typically MOSFETs), $N$+2 diodes, and

one coaxial-winding transformer. A schematic diagram is shown in Figure 2.14, and a summary of component characteristics is given in Table 2.7.

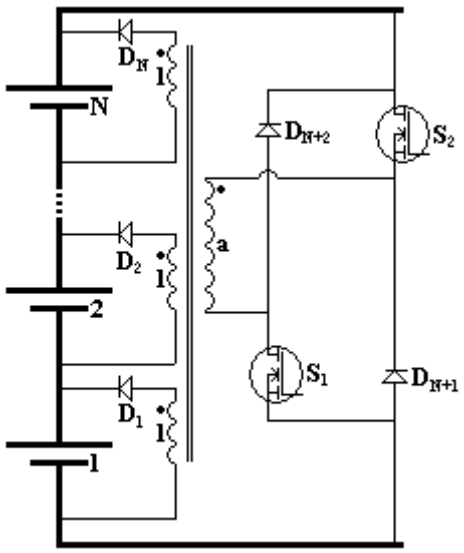



Figure 2.14: Schematic diagram for the Centralized Forward-Converter system

Table 2.7: Summary of component characteristics for the Centralized Forward-Converter system

| Description | Number | Voltage Rating |
|---|---|---|
| MOSFET | 2 | N* $V_{cell}$ |
| Diode | 2 | N* $V_{cell}$ |
| Diode | N | $V_{cell}$ |
| Coaxial-Winding Transformer | 1 | 1:a turns ratio |

The variation of SOC is equalized by providing different charging currents to the individual battery cells from the high-voltage bus. The charging currents to each battery cell are automatically set by its voltage. To illustrate the equalization process, consider the diagram and current waveforms shown in Figure 2.15.

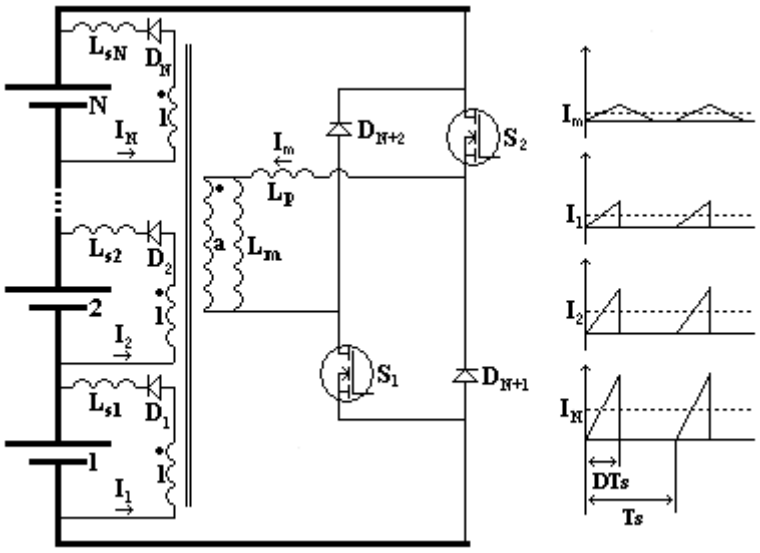



Figure 2.15: The Centralized Forward-Converter system transferring current from the high-voltage bus to the individual battery cells

In this example, the first battery has the highest voltage and the $N^{th}$ battery has the lowest voltage. First, both $S_1$ and $S_2$ are switched on. This applies the voltage of the battery pack $v_{pk}$ to the primary side of the transformer. The driving impedance of the current in the primary winding is its magnetizing inductance $L_m$ and primary leakage inductance $L_p$. Likewise, the driving impedances of the currents in the secondary windings are their corresponding leakage inductances $L_k$. The average charging currents of the battery cells are approximated by Equation 2.8, where $v_k$ is the voltage of the $k^{th}$ battery cell, and $D$ is the duty cycle of the MOSFETs.

$$I_k = \frac{(\frac{v_{pk}}{a} - v_k)D}{2(L_k)} \quad k = 1, 2, \ldots, N \qquad 2.8$$

Then both $S_1$ and $S_2$ are switched off, the freewheeling diodes conduct current until the transformer is demagnetized. In this state, the secondary-side diodes are reverse biased and do not conduct current. The freewheeling diodes transition to open circuits when transformer is demagnetized.

The power capability of the Centralized Forward Converter system can be tailored to most applications. In addition, the control strategy is simple, as it only requires one gate drive signal. Finally, the energy used to charge the battery cells is sourced evenly from the high-voltage bus.

However, the system requires a custom transformer core and an *N*-signal, co-axial secondary wire, which are costly to implement. In addition, the system is not modular. The cost of the components and non-modular setup make this system applicable for equalization at the battery module level. Also, equalization can only be performed to correct lower-than-average SOC. Finally, the system is only effective when the battery chemistry has a voltage profile that is proportional to SOC.

## 2.5 Hybrid Systems

Hybrid systems utilize off-the-shelf circuit components and attempt to redistribute energy evenly to the other battery cells. They do not use electrical transformers as found in the parallel systems, but do require more wiring than the series systems. These systems are advantageous to space sensitive applications that require large amounts of energy to be redistributed.

### 2.5.1 Hybrid System #1

The Hybrid System #1 equalizes SOC imbalance by transferring energy from a battery cell with a higher-than-average SOC and distributing it to the other battery cells [26]. For *N* battery cells, the system consists of *N* electronic switches (typically MOSFETs), *N* diodes, and *N*-1 inductors. A schematic diagram is shown in Figure 2.16, and a summary of component characteristics is given in Table 2.8 .

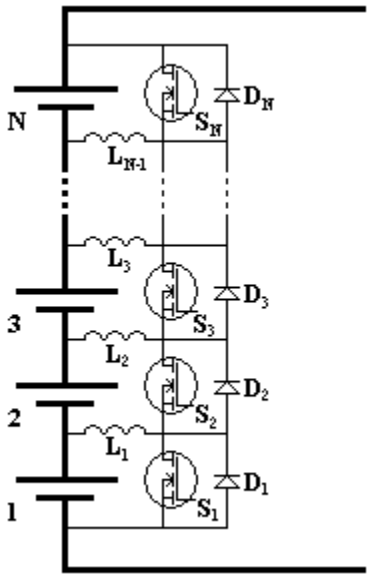



Figure 2.16: Schematic diagram of the Hybrid System #1

Table 2.8: Summary of component characteristics of the Hybrid System #1

| Description | Number | Voltage Rating |
|---|---|---|
| MOSFET | N | $V_{cell}$ |
| Diode | N | $V_{cell}$ |
| Inductor | N-1 | n/a |

To analyze the ability of the Hybrid System #1 to equalize SOC imbalance, consider the case when the $k^{th}$ battery cell has a higher-than-average SOC. To correct this, the BMS switches $S_k$ on, which generates the average discharge current $I_d$ that is shown in Figure 2.17. Equation 2.9 approximates $I_d$, where $v_k$ is the voltage of the $k^{th}$ battery cell, $D_k$ is the duty cycle of $S_k$, $T_s$ is the switching period, $L_k$ is the inductance of the $k^{th}$ inductor, and $L_{k\text{-}1}$ is the inductance of the $k$-1 inductor.

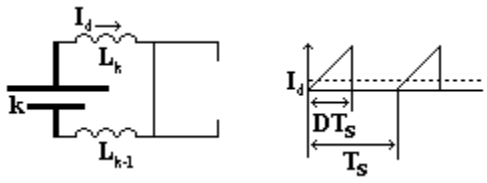



Figure 2.17: The Hybrid System #1 discharging current from the $k^{th}$ battery cell

$$I_d = \frac{v_k D_k^2 T_s}{4(L_k + L_{k-1})} \quad k = 2, 3, \ldots, N-1 \qquad 2.9$$

When $I_d$ reaches a desired level that has been determined to equalize the SOC imbalance, $S_k$ is switched off. The energy is then redistributed to the other battery cells, as shown in Figure 2.18. The average inductor currents $I_k$ and $I_{k\text{-}1}$ are approximated by Equation 2.10 and Equation 2.11, respectively.

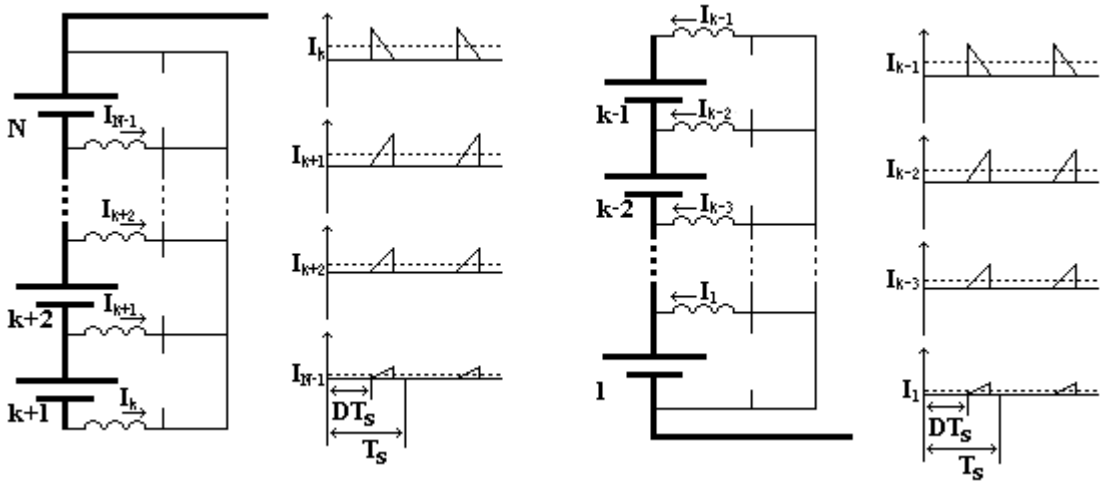



Figure 2.18: The Hybrid System #1 redistributing energy
from the k[th] battery cell to the other battery cells

$$I_k = \frac{v_k^2 D_k^2 T_s}{8L_k \sum_{i=k+1}^{N} v_i} \quad k = 2, 3, \ldots, N-1 \tag{2.10}$$

$$I_{k-1} = \frac{v_k^2 D_k^2 T_s}{8L_{k-1} \sum_{i=1}^{k-1} v_i} \quad k = 2, 3, \ldots, N-1 \tag{2.11}$$

The voltages present in the system induce currents to flow in the other inductors as well. The average inductor currents above the $k^{th}$ battery cell are approximated by Equation 2.12, and the average inductor currents below the $k^{th}$ battery cell are approximated by Equation 2.13.

$$I_x = \frac{v_k^2 D_k^2 T_s \sum_{j=x+1}^{N} v_j}{8L_x (\sum_{i=k+1}^{N} v_i)^2} \quad x = k+1, k+2, \ldots, N-1 \qquad 2.12$$

$$I_x = \frac{v_k^2 D_k^2 T_s \sum_{j=1}^{x} v_j}{8L_x (\sum_{i=1}^{k-1} v_i)^2} \quad x = 1, 2, \ldots, k-2 \qquad 2.13$$

The average battery cell currents above the *k*th battery cell are approximated by Equation 2.14, and the average battery cell currents below the *k*th battery cell are approximated by Equation 2.15.

$$I_{cellx} = \sum_{i=k+1}^{x} I_{i-1} \quad x = k+2, k+3, \ldots, N \qquad 2.14$$

$$I_{cellx} = \sum_{i=k-1}^{x} I_i \quad x = 1, 2, \ldots, k-1 \qquad 2.15$$

The equalization power of the Hybrid System #1 can be tailored to most applications. In addition, the use of off-the-shelf parts permits a moderate implementation cost. Finally, it utilizes a completely modular setup.

However, the control scheme is moderately complex. It requires a gate drive signal and current feedback signal for each battery. The moderate control complexity and implementation cost makes this system applicable for equalization at the battery module level. In addition, equalization can only be performed on a battery cell with a higher-than-average SOC. Finally, the redistribution of energy is uneven. The closer a battery cell is to the battery cell being equalized, the less charging current it receives.

### 2.5.2 Hybrid System #2

The Hybrid System #2 equalizes SOC imbalance by discharging energy from battery cells with higher-than-average SOC. For the top battery cell in the battery pack, the energy is redistributed to the battery cells below it. For all other battery cells, the energy is redistributed to the battery cells that are above them [27][28][29][30]. For *N* battery cells, the system consists of *N* electrical switches (typically MOSFETs), *N* diodes, and *N* inductors. A schematic diagram is shown in Figure 2.19, and a summary of component characteristics is given in Table 2.9.

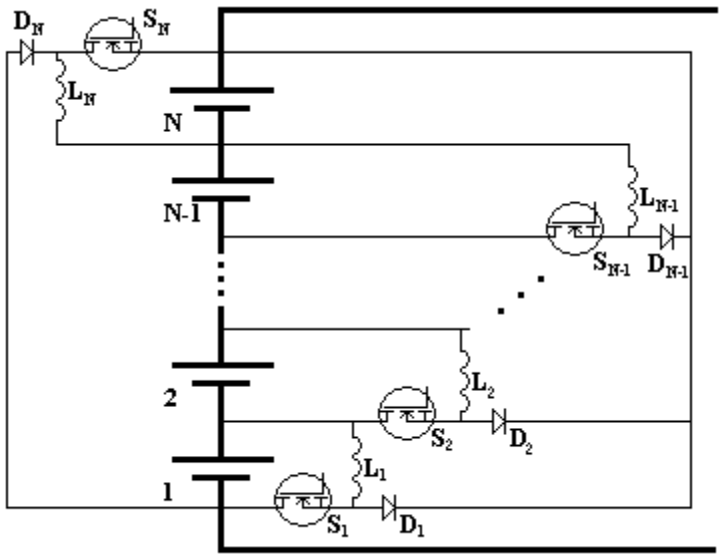



Figure 2.19: Schematic diagram of the Hybrid System #2

Table 2.9: Summary of component characteristics of the Hybrid System #2

| Description | Number | Voltage Rating |
|---|---|---|
| MOSFET | N | $V_{cell}$<br>2: N*$V_{cell}$ |
| Diode | N | 1: (N-1)*$V_{cell}$, (N-2)*$V_{cell}$, …, $V_{cell}$ |
| Inductor | N | n/a |

To analyze how the Hybrid System #2 equalizes SOC imbalance, consider the case when the $k^{th}$ battery cell a higher-than-average SOC, where $1 \geq k \geq N\text{-}1$. The BMS acts to correct this variation by discharging energy from the battery cell and redistributing it to the battery cells that are higher up on the battery pack. The circuit diagram and typical current waveforms are shown in Figure 2.20.

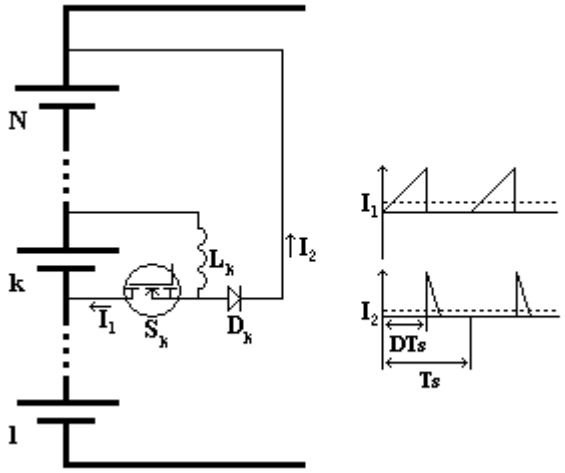



Figure 2.20: The Hybrid System #2 discharging current from the $k^{th}$ battery cell

The BMS has determined that the average discharging current $I_1$ will equalize the SOC variability of the battery pack. $S_k$ is switched on, until its average current reaches $I_1$. This current is approximated by Equation 2.16, where $v_k$ is the voltage of the battery cell, $D_k$ is the duty cycle of $S_k$, and $L_k$ is the inductance of the inductor. $S_k$ is then switched off, and the diode conducts current until the inductor is demagnetized. The average current $I_2$ charges the battery cells that are above the $k^{th}$ battery cell in the battery pack.

$$I_1 = \frac{v_k D_k}{2L_k} \quad k = 1, 2, \ldots, N-1 \qquad 2.16$$

To analyze how the Hybrid System #2 equalizes SOC imbalance in the $N^{th}$ battery cell, consider the case when it has a higher-than-average SOC. The BMS acts to correct this variation by discharging energy from it and redistributing the energy to the battery cells that are below it. The circuit diagram and typical current waveforms are shown in Figure 2.21.

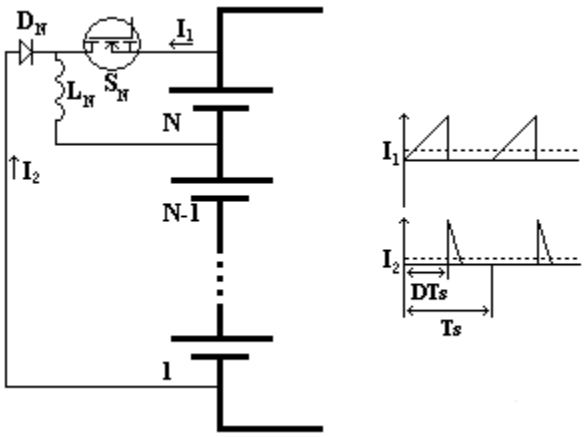



Figure 2.21: The Hybrid System #2 discharging current from the $N^{th}$ battery cell

The BMS has determined that an average discharge current $I_1$ will equalize the SOC variation. $S_N$ is switched on until its average current reaches $I_1$. $I_1$ is approximated by Equation 2.17, where $v_N$ is the voltage of the $N^{th}$ battery cell, $D_N$ is the duty cycle of $S_N$, and $L_N$ is the inductance of the inductor. $S_N$ is then switched off, and the freewheeling diode conducts current until the inductor is demagnetized. The average current $I_2$ charges the other battery cells in the battery pack.

$$I_1 = \frac{v_N D_N}{2L_N} \qquad 2.17$$

The equalization power of the Hybrid System #2 can be tailored to most applications. In addition, the use of off-the-shelf components keeps the implementation cost moderate. Finally, it utilizes a relatively modular setup, with only the $N^{th}$ circuit requiring its design to be specific to the battery pack.

However, the control system is moderately complex. It requires a gate drive signal and current feedback signal for each battery cell. The moderate control complexity implementation cost make the system applicable for equalization at the battery module level. In addition, equalization can only be performed on battery modules with higher-than-average SOC. Finally, the redistribution of energy is uneven. Except for the $N^{th}$ battery cell, all other battery cells redistribute the recovered energy unevenly.

## 2.6 System Selection for the Penn State Research Vehicles

To determine which system to implement for the Penn State research vehicles, the evaluation criteria are weighted based on importance to the project. Each criterion is compared to the other criteria. A score of one indicates the criterion is more important than the other criterion, and a score of zero indicates that it is less important than the other criterion. The scores of each criterion are totaled, and then a weight factor is determined. The weight factor is the ratio of the score of the criterion to the total of all criteria scores. The higher the weight factor, the more important it is to the research vehicles. The results are shown in Table 2.10.

Table 2.10: Criteria weight factors for the Penn State research vehicles

| | Equalization Power | Modular Setup | Simple / Robust Control Strategy | Expense | Equalization at the Battery Cell Level | High and Low SOC Equalization | Even Energy Redistribution | Total Score | Weight Factor |
|---|---|---|---|---|---|---|---|---|---|
| Equalization Power | x | 1 | 1 | 1 | 1 | 1 | 0 | 5 | 0.25 |
| Modular Setup | 0 | x | 1 | 1 | 1 | 0 | 0 | 3 | 0.15 |
| Simple / Robust Control Strategy | 0 | 0 | x | 1 | 0 | 0 | 0 | 1 | 0.05 |
| Expense | 0 | 0 | 0 | x | 0 | 0 | 0 | 0 | 0.00 |
| Equalization at the Battery Cell Level | 0 | 0 | 1 | 0 | x | 0 | 0 | 1 | 0.05 |
| High and Low SOC Equalization | 0 | 1 | 1 | 1 | 1 | x | 0 | 4 | 0.20 |
| Even Energy Redistribution | 1 | 1 | 1 | 1 | 1 | 1 | x | 6 | 0.30 |

The ability to evenly redistribute energy to (from) the battery pack is the most important criterion. Intuitively this makes sense, because a system that induces SOC variability in other battery cells is counter-productive. The second most important criterion is the ability to generate substantial equalization power. A design goal of 160 W (10 A) was established to provide enough equalization power to eliminate the variability in the 100-A·h and 95-A·h battery packs. The third most important criterion is the ability to equalize both higher-than-average and lower-than-average SOC. With significant equalization expected, the flexibility to equalize both situations is desired to reduce efficiency losses. The fourth most important criterion is to have a modular setup. The system must be able to be adapted to two differently sized battery packs.

The criteria that were considered least important were a simple/robust control strategy, equalization at the battery cell level, and expense. It is acceptable to use the amount of control that is required to achieve the desired criteria, as long as a fail-safe design is followed. Equalization at the battery cell level is not tremendously important, because the batteries are packaged in ten-cell battery modules. When a failure is acknowledged, the battery module will be replaced. Finally, low expense is not important because it is being developed as a research vehicle, not a commercial vehicle.

After careful review of the systems presented and the weights of the criteria, it was determined to implement the Bidirectional Flyback-Converter system. It excels in the important areas; it has the ability to evenly redistribute energy, it can meet our power requirements, it is modular by design, and it can equalize both types of SOC variation. It suffers in the non-important areas; it has high implementation cost, and its control scheme is complex.

# Chapter 3

## Analysis, Design, and Implementation of a Charge-Equalization Circuit

A Bidirectional Flyback-Converter system has the characteristics that match the design goals for the Penn State research vehicles. It has even re-distribution of energy, high equalization power, the flexibility to equalize battery cells with both higher-than-average and lower-than-average state of charge (SOC), and it utilizes a modular design. To handle its complexity and to reduce its cost, it will be implemented at the battery-module level (groups of 10 battery cells) and will be operated in the discontinuous conduction mode.

### 3.1 Operating Conditions

Both Penn State research vehicles use nickel-metal hydride (NiMH) battery cells. The output voltage of a NiMH battery cell can range from 1 V to 1.5 V [31]. Each battery module consists of 10 battery cells. To be cautious, a battery-module voltage range of 8 V to 16 V is considered. The Electric-Lion research vehicle has 12 battery modules, resulting in a high-voltage bus voltage range of 96 V to 192 V. The HyLion research vehicle has 25 battery modules, resulting in a high-voltage bus voltage range of 200 V to 400 V. Therefore, the system must work with a voltage range of 8 V to 16 V at the battery module, and a voltage range of 96 V to 400 V at the high-voltage bus.

The Electric-Lion has battery modules with capacities of 100 A·h, while the HyLion has battery modules with capacities of 95 A·h. An average current of ± 10 A was determined as a design goal.

### 3.2 Analysis

The input and output side of the converter changes depending on whether the battery module is being charged or discharged. Therefore, these terms will not be used. Instead, the term module side will be used when discussing components on the side of the transformer with the battery module. Also, the term bus side will be used when discussing components on the side of the transformer with the high-voltage bus. The transformer will be modeled by an ideal transformer and magnetizing inductance $L_m$, with the magnetizing inductance referenced to the module side of the transformer. All other power electronic components will be modeled as ideal. A schematic diagram of the circuit to be analyzed is shown in Figure 3.1.

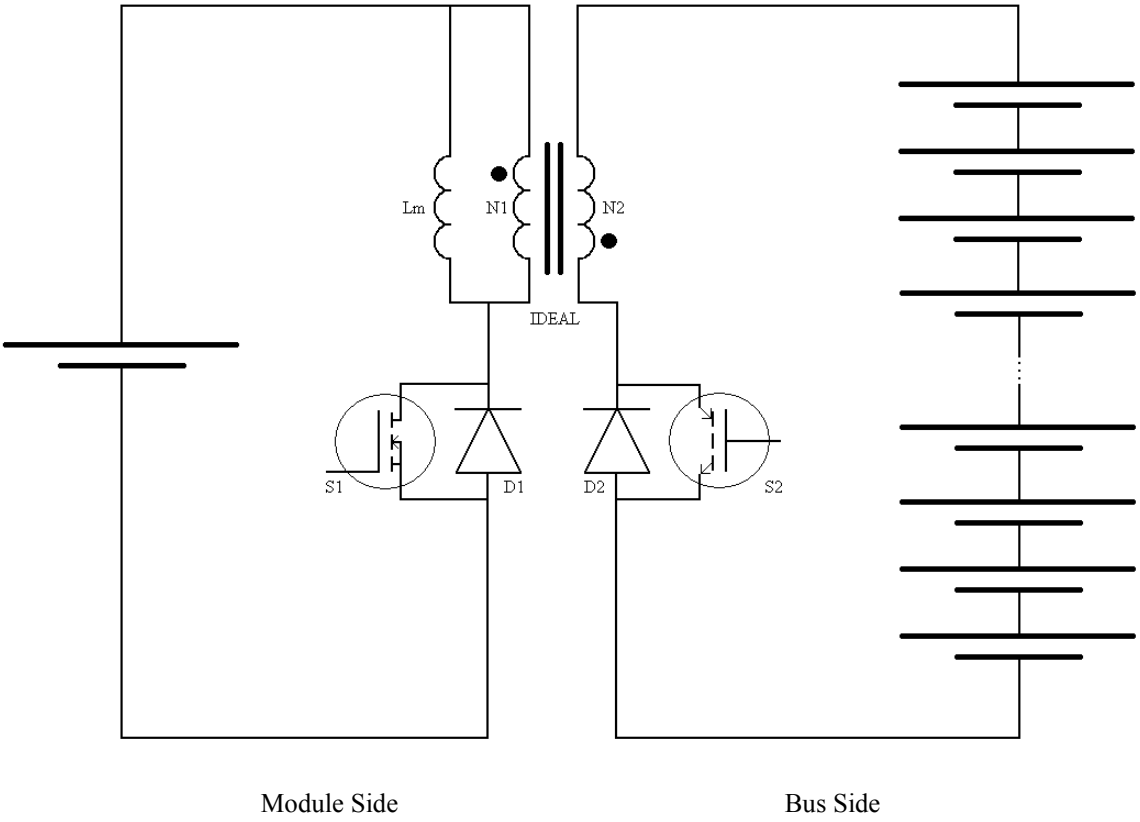



Figure 3.1: Schematic diagram for the analysis of the
Bidirectional Flyback-Converter system

When in use, the circuit operation can be in one of two states. The first state is when the battery module has a higher-than-average SOC, and the circuit acts to discharge current from it and redistribute it to the high-voltage bus. The second state is when the battery module has a lower-than-average SOC, and the circuit acts to draw current from the high-voltage bus and uses it to charge the battery module. The discharge state will be described next.

### 3.2.1 Discharging the Battery Module

When the master controller of the battery management system (BMS) determines that a battery module has a higher-than-average SOC, it directs the charge-equalization circuit (CEC) to discharge current from that battery module. The master controller calculates an average discharge current and duration of discharge, and sends this information to the CEC. The gate drive of the MOSFET $S_1$ is then pulse-width modulated to generate this desired average discharge current.

Consider the circuit diagram and waveforms shown in Figure 3.2. The voltages of the battery module $V_{bm}$ and high-voltage bus $V_{hv}$ are considered constant because the switching period $T_s$ is much smaller then time required for these voltages to change. Let $d_1$ be the duty cycle of $S_1$, $L_m$ be the magnetizing inductance of the transformer, $i_{Lm}(t)$ be the current of the magnetizing inductance, and $v_{Lm}(t)$ be the voltage that is applied to the magnetizing inductance. When $S_1$ is switched on, $v_{Lm}(t) = V_{bm}$, and $i_{Lm}(t)$ ramps up. The current $i_{Lm}(t)$ is approximated by Equation 3.1, and its peak value $\hat{i}_{Lm}$ occurs at the time $t = d_1 T_s$, and is approximated by Equation 3.2.

$$i_{Lm}(t) = \frac{v_{bm} t}{L_m} \quad 0 \le t \le d_1 T_s \tag{3.1}$$

$$\hat{i}_{Lm} = \frac{v_{bm} d_1 T_s}{L_m} \tag{3.2}$$

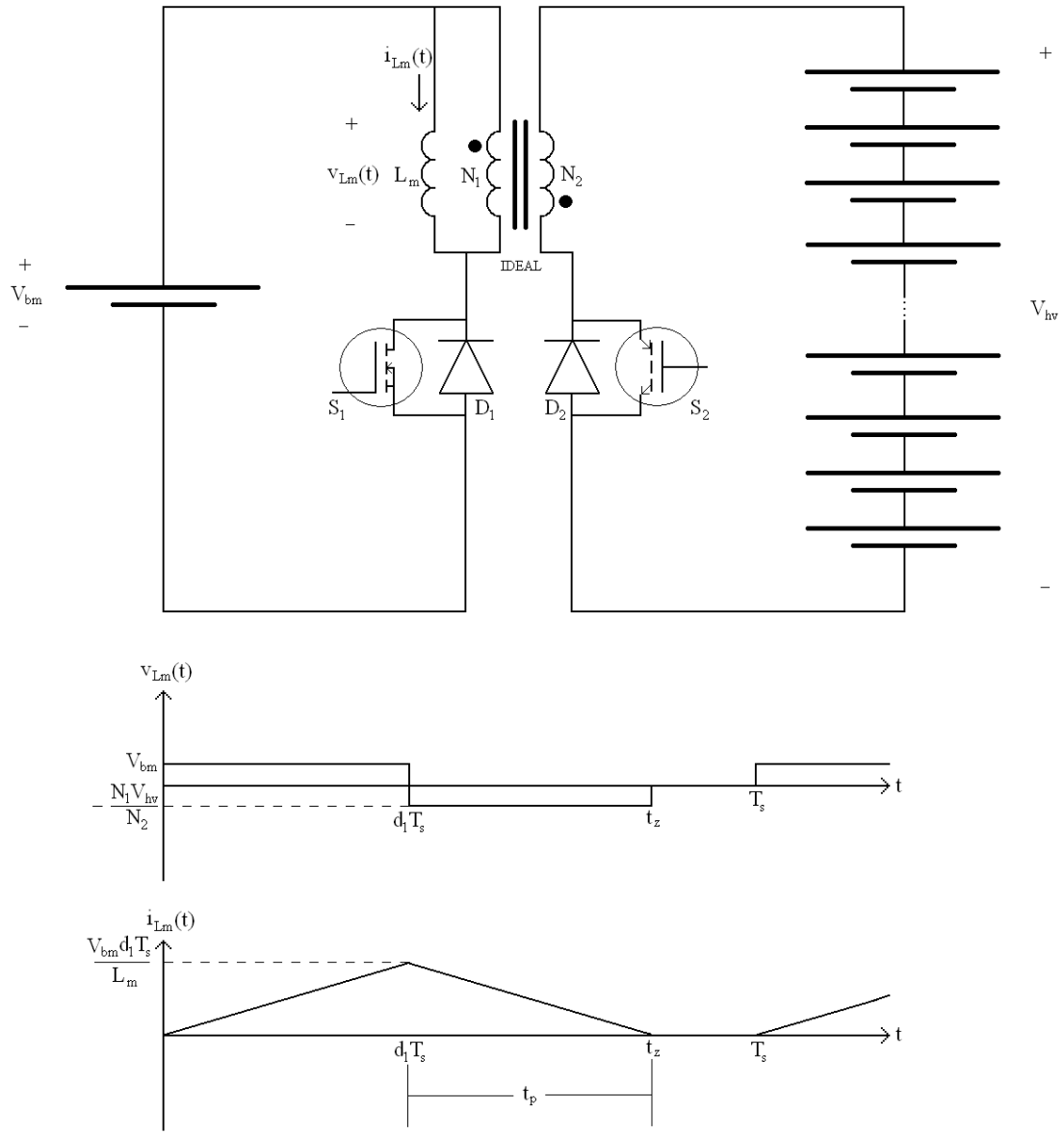



Figure 3.2: Voltage and current waveforms of the magnetizing inductance during a discharge sequence

When $S_1$ is switched off, the freewheeling diode $D_2$ transitions from an open circuit to a short circuit and conducts current, which clamps $v_{Lm}(t) = -N_1 V_{hv} / N_2$. The current $i_{Lm}(t)$ ramps down, and is approximated by Equation 3.3, where $t_z$ is the time when it reaches zero amps. The freewheeling diode $D_2$ transitions from a short circuit to an open circuit at the time $t = t_z$.

$$i_{Lm}(t) = \frac{V_{bm} d_1 T_s}{L_m} - \frac{N_1 V_{hv} (t - d_1 T_s)}{N_2 L_m} \quad d_1 T_s \le t \le t_z \tag{3.3}$$

To maintain periodic steady state, the average voltage applied to the magnetizing inductance must be zero. From this restriction, a relationship for the maximum discharging duty cycle $d_{1,MAX}$ is established. This relationship is approximated by Equation 3.4.

$$d_{1,MAX} = \frac{1}{\frac{N_2 V_{bm}}{N_1 V_{hv}} + 1} \tag{3.4}$$

The discharge current of the battery module is equal to $i_{Lm}(t)$ when $S_1$ is switched on. Due to the triangular shape of the current waveform, the dc component of the battery module discharge current $I_{bm,DC}$ is approximated by Equation 3.5, the rms battery module discharge current $I_{bm,RMS}$ is approximated by Equation 3.6, and the power of the CEC during a discharge sequence $P_{DIS}$ is approximated by Equation 3.7.

$$I_{bm,DC} = \frac{V_{bm} d_1^2 T_s}{2 L_m} \tag{3.5}$$

$$I_{bm,RMS} = \frac{V_{bm} d_1 T_s}{L_m} \sqrt{\frac{d_1}{3}} \tag{3.6}$$

$$P_{DIS} = \frac{V_{bm}^2 d_1^2 T_s}{2 L_m} \tag{3.7}$$

The current that is transferred to the high-voltage bus is equal to $N_1 i_{Lm}(t) / N_2$ when $S_1$ is switched off. The time $t_p$ that the freewheeling diode $D_2$ is conducting current is approximated by Equation 3.8, the dc component of the high-voltage bus charging current $I_{hv,DC}$ is approximated by Equation 3.9, and the rms high-voltage bus charging current $I_{hv,RMS}$ is approximated by Equation 3.10.

$$t_p = \frac{N_2 V_{bm} d_1 T_s}{N_1 V_{hv}} \tag{3.8}$$

$$I_{hv,DC} = \frac{V_{bm}^2 d_1^2 T_s}{2 V_{hv} L_m} \tag{3.9}$$

$$I_{hv,RMS} = \frac{N_1 V_{bm} d_1 T_s}{N_2 L_m} \sqrt{\frac{N_2 V_{bm} d_1}{3 N_1 V_{hv}}} \tag{3.10}$$

### 3.2.2 Charging the Battery Module

When the BMS determines that a battery module has a lower-than-average SOC, it directs the CEC to transfer charging current to that battery module. The BMS calculates an average current and duration of charge, and sends this information to the CEC. The gate drive of the IGBT $S_2$ is pulse-width modulated to generate this desired average charging current.

Consider the circuit diagram and waveforms shown in Figure 3.3. Let $d_2$ be the duty cycle of $S_2$. When $S_2$ is switched on, $v_{Lm}(t) = -N_1 V_{hv} / N_2$, and $i_{Lm}(t)$ begins to ramp down. The current $i_{Lm}(t)$ is approximated by Equation 3.11. Its peak value occurs at the time $t = d_2 T_s$, and is approximated by Equation 3.12.

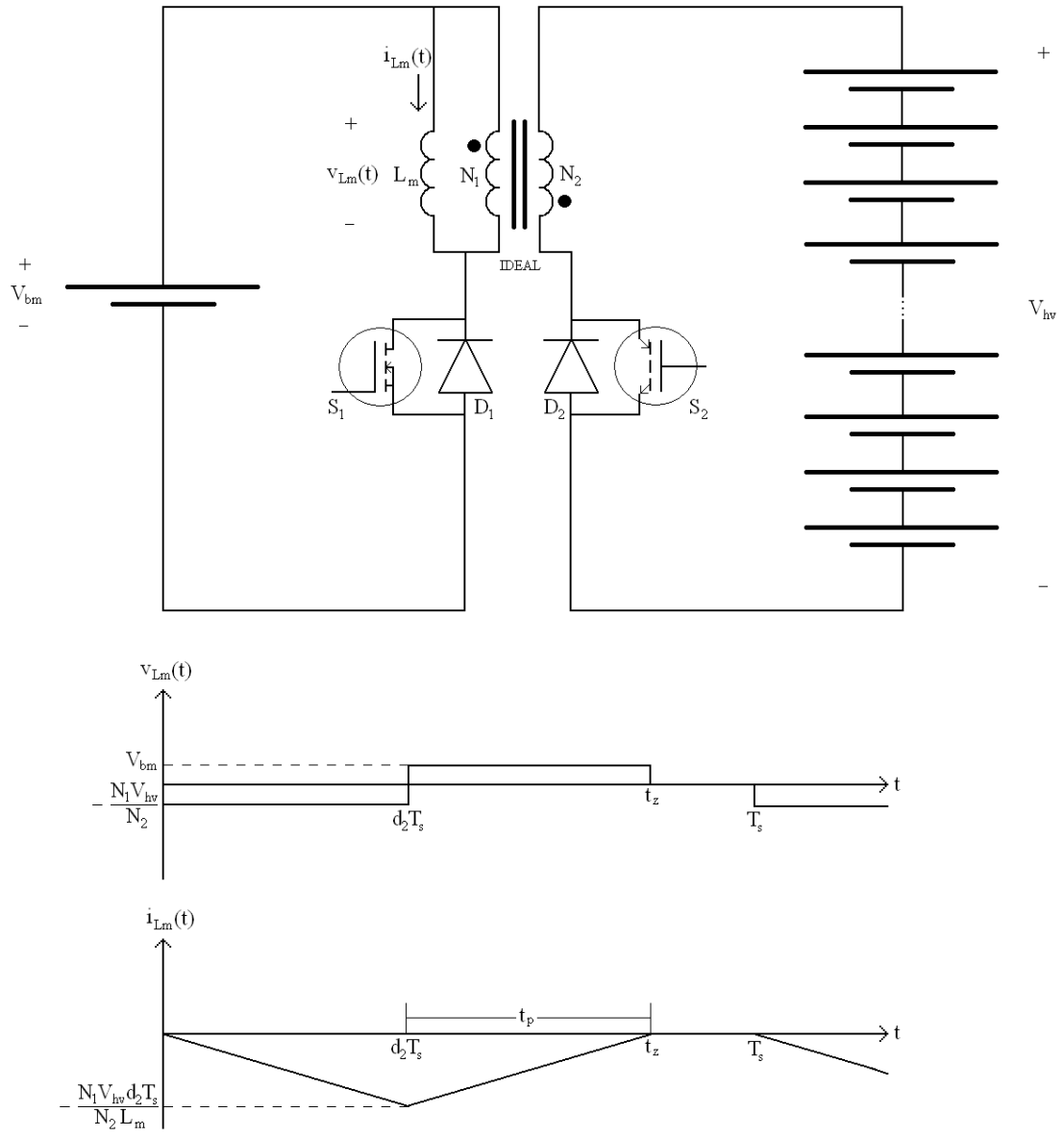



Figure 3.3: Voltage and current waveforms of the magnetizing inductance during a charging sequence

$$i_{Lm}(t) = -\frac{N_1 V_{hv} t}{N_2 L_m} \quad 0 \le t \le d_2 T_s \tag{3.11}$$

$$\hat{i}_{Lm} = -\frac{N_1 V_{hv} d_2 T_s}{N_2 L_m} \tag{3.12}$$

When $S_2$ is switched off, the freewheeling diode $D_1$ transitions from an open circuit to a short circuit and conducts current, which clamps $v_{Lm}(t) = V_{bm}$. The current $i_{Lm}(t)$ ramps up, and it is approximated by Equation 3.13, where $t_z$ is the time when it reaches zero amps. The freewheeling diode transitions from a short circuit to an open circuit at the time $t = t_z$.

$$i_{Lm}(t) = -\frac{N_1 V_{hv} d_2 T_s}{N_2 L_m} + \frac{V_{bm}(t - d_2 T_s)}{L_m} \quad d_2 T_s \le t \le t_z \tag{3.13}$$

To maintain periodic steady state, the average voltage applied to the magnetizing inductance must be zero. From this restriction, a relationship for the maximum charging duty cycle $d_{2,MAX}$ is established. This relationship is approximated by Equation 3.14.

$$d_{2,MAX} = \frac{1}{\frac{N_1 V_{hv}}{N_2 V_{bm}} + 1} \tag{3.14}$$

The charging current of the battery module is equal to $i_{Lm}(t)$ when $S_2$ is switched off. The time that the freewheeling diode $D_1$ is conducting current $t_p$ is approximated by Equation 3.15, the dc component of the battery module charging current $I_{bm,DC}$ is approximated by Equation 3.16, and the rms battery module charging current $I_{bm,RMS}$ is approximated by Equation 3.17.

$$t_p = \frac{N_1 V_{hv} d_2 T_s}{N_2 V_{bm}} \tag{3.15}$$

$$I_{bm,DC} = \frac{N_1^2 V_{hv}^2 d_2^2 T_s}{2 N_2^2 V_{bm} L_m} \tag{3.16}$$

$$I_{bm,RMS} = \frac{N_1 V_{hv} d_2 T_s}{N_2 L_m} \sqrt{\frac{N_1 V_{hv} d_2}{3 N_2 V_{bm}}} \tag{3.17}$$

The current that is discharged from the high-voltage bus is equal to $N_1 i_{Lm}(t) / N_2$ when $S_2$ is switched on. The dc component of the high-voltage-bus discharging current $I_{hv,DC}$ is approximated by Equation 3.18, the rms high-voltage-bus discharging current $I_{hv,RMS}$ is approximated by Equation 3.19, and the power of the CEC $P_{CH}$ during a charging sequence is approximated by Equation 3.20.

$$I_{hv,DC} = \frac{N_1^2 V_{hv} d_2^2 T_s}{2 N_2^2 L_m} \tag{3.18}$$

$$I_{hv,RMS} = \frac{N_1^2 V_{hv} d_2 T_s}{N_2^2 L_m} \sqrt{\frac{d_2}{3}} \tag{3.19}$$

$$P_{CH} = \frac{N_1^2 V_{hv}^2 d_2^2 T_s}{2 N_2^2 L_m} \tag{3.20}$$

### 3.3 Design

The BMS will consist of one master controller and either 12 or 25 slave CECs, depending on the vehicle. The master controller will monitor the SOC of each battery module. When equalization is required on a particular battery module, it will instruct the appropriate CEC to transfer a certain magnitude of current, in a certain direction (charge or discharge), for a certain amount of time. The CEC will act to perform the instruction, and report the results back to the master controller.

Control within the CEC will be performed by a microcontroller. When it receives an equalization instruction, it will control the power electronics to perform the instruction. The CEC must operate over a wide module voltage range and bus voltage range. In addition, it must be robust to parameter variations such as the tolerance in the inductance of the transformer and parasitic resistance of the circuit. Therefore, it will utilize feedback control to ensure accurate results.

Power for the master controller and CEC will be from a 12-V, chassis-ground power rail. The CEC microcontroller and communication electronics will be powered at chassis ground potential. The CEC gate drive and signal conditioning electronics will be powered by isolated dc-to-dc converters and referenced to the module-side ground or bus-side ground. A top-level circuit schematic of the CEC is shown in Figure 3.4.

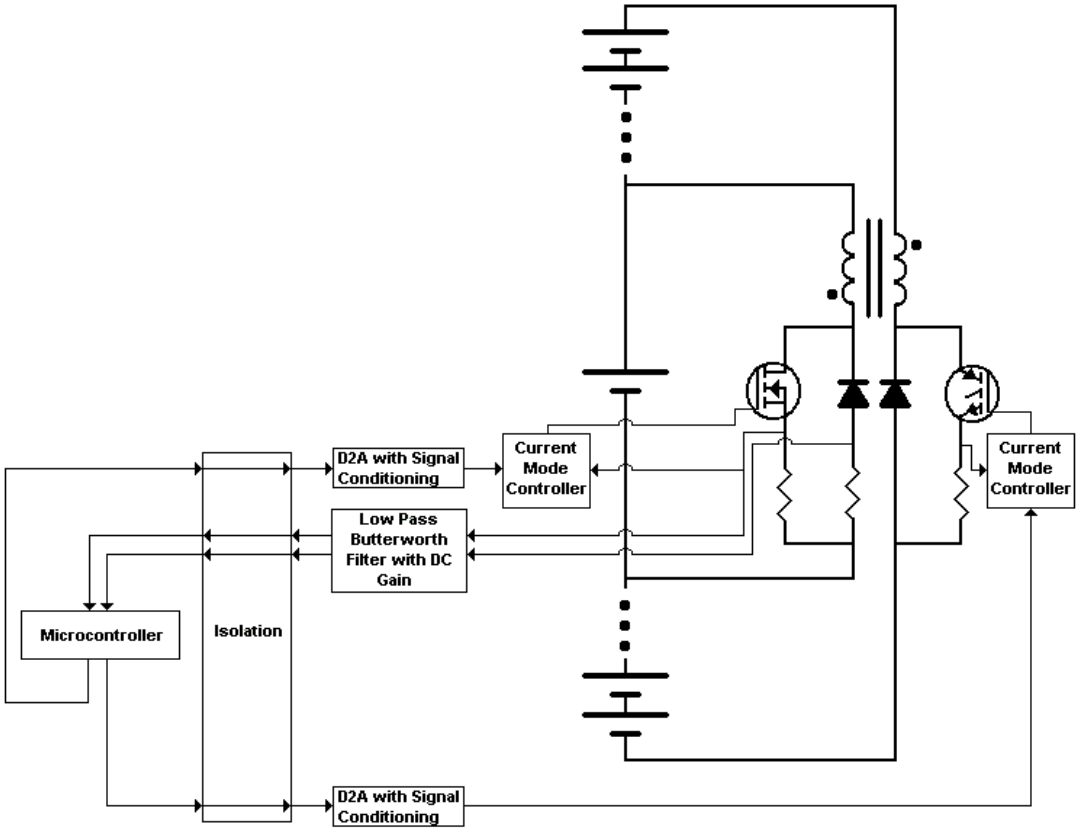



Figure 3.4: Top-level circuit schematic of the CEC

The first step in the design of the CEC is to apply the analysis of the previous section to the range of operating conditions to determine the current, voltage, and power levels in the different the sub-circuits. Next, the power electronic components will be selected. From there, the control circuitry will be designed. Following that, the communications network and protocol will be developed. With all of the circuit components selected, the microcontroller and on-board dc-to-dc converters will be specified. Finally, the software will be developed for the microcontroller.

### 3.3.1 Current, Voltage, and Power Levels

The first step in the design of the Bidirectional Flyback-Converter system for the Penn State research vehicles is to determine the current, voltage, and power levels from the relationships established in the previous section. To begin, the switching frequency for both the MOSFET and IGBT is chosen. It is desired to keep the switching frequency low to simplify implementation, and to reduce the peak current that is required to reach the average current design goal. However, it is not desired to allow humans to hear the system in operation. Therefore, the CEC will operate with a switching frequency of 25 kHz.

The next step is to determine the transformer winding ratio $N_2 / N_1$. The winding ratio sets the maximum duty cycle that can be used while maintaining periodic steady state. A large maximum duty cycle is desired to increase the resolution of the current that is transferred by the CEC. The optimal winding ratio would typically be set to $N_2 / N_1 = V_{hv} / V_{bm} = N$, where $N$ is the number of battery modules in the battery pack. However, the system is being designed for two different vehicles. Therefore, Equation 3.4 and Equation 3.14 are evaluated to determine the maximum duty cycles for all operating conditions and various winding ratios; the results are shown in Table 3.1.

Table 3.1: Maximum duty cycles for various transformer winding ratios

| | | Electric-Lion | | | | HyLion | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $V_{bm}$ | 8 | 8 | 16 | 16 | 8 | 8 | 16 | 16 |
| | $V_{hv}$ | 96 | 192 | 96 | 192 | 200 | 400 | 200 | 400 |
| | $d_{1,MAX}$ | 0.50 | 0.67 | 0.33 | 0.50 | 0.68 | 0.81 | 0.51 | 0.68 |
| Winding Ratio = 12 | $d_{2,MAX}$ | 0.50 | 0.33 | 0.67 | 0.50 | 0.32 | 0.19 | 0.49 | 0.32 |
| | **Range** | **0.33** | **to** | | **0.67** | **0.19** | **to** | | **0.81** |
| | $d_{1,MAX}$ | 0.32 | 0.49 | 0.19 | 0.32 | 0.50 | 0.67 | 0.33 | 0.50 |
| Winding Ratio = 25 | $d_{2,MAX}$ | 0.68 | 0.51 | 0.81 | 0.68 | 0.50 | 0.33 | 0.67 | 0.50 |
| | **Range** | **0.19** | **to** | | **0.81** | **0.33** | **to** | | **0.67** |
| | $d_{1,MAX}$ | 0.41 | 0.59 | 0.26 | 0.41 | 0.60 | 0.75 | 0.42 | 0.60 |
| Winding Ratio = 17 | $d_{2,MAX}$ | 0.59 | 0.41 | 0.74 | 0.59 | 0.40 | 0.25 | 0.58 | 0.40 |
| | **Range** | **0.26** | **to** | | **0.74** | **0.25** | **to** | | **0.75** |

From the data in Table 3.1, the Electric-Lion has the greatest lower bound of its maximum duty cycle when the winding ratio is 12. However, the HyLion has the greatest lower bound of its maximum duty cycle when the winding ratio is 25. To accommodate both vehicles, a winding ratio of 17 is selected.

With the switching frequency and winding ratio determined, the next step is to determine the magnetizing inductance of the transformer. This inductance must be selected such that 10 A of average current can be transferred to/from the battery module, during all operating conditions. Equation 3.5 and Equation 3.16 are evaluated with the maximum values of duty cycle that were calculated earlier, and 10 A of average current, to determine the inductance values that are required. These values are shown in Table 3.2.

Table 3.2: Magnetizing inductance values required to achieve 10 A of average battery module current

| | Electric-Lion | | | | HyLion | | | |
|---|---|---|---|---|---|---|---|---|
| $V_{bm}$ | 8 | 8 | 16 | 16 | 8 | 8 | 16 | 16 |
| $V_{hv}$ | 96 | 192 | 96 | 192 | 200 | 400 | 200 | 400 |
| $L_{m,DIS}$ | 2.74E-06 | 5.48E-06 | 2.18E-06 | 5.48E-06 | 5.67E-06 | 8.91E-06 | 5.75E-06 | 1.13E-05 |
| $L_{m,CH}$ | 2.74E-06 | 5.48E-06 | 2.18E-06 | 5.48E-06 | 5.67E-06 | 8.91E-06 | 5.75E-06 | 1.13E-05 |
| **$L_{m,MIN}$** | **2.18E-06** | | | | | | | |

To ensure that all operating conditions can achieve the design goal of 10 A of average battery module current, the magnetizing inductance must be equal to or less then the $L_{m,MIN}$ shown in Table 3.2. Therefore, a magnetizing inductance of 1.75 μH is chosen. This value is lower then the value calculated to accommodate non-ideal voltage drops present in the system wires and electronic switches, and the tolerance in the inductance value of the transformer.

With the magnetizing inductance value of the transformer established, the duty cycles required to achieve 10 A of average battery module current can be calculated. These values must be lower then the maximum duty cycles that were calculated earlier. Equation 3.5 and Equation 3.16 are used to calculate the duty cycle values for the various operating conditions. The results are shown in Table 3.3.

Table 3.3: Calculated duty cycles to achieve 10 A of average battery module current

| | Electric-Lion | | | | HyLion | | | |
|---|---|---|---|---|---|---|---|---|
| $V_{bm}$ | 8 | 8 | 16 | 16 | 8 | 8 | 16 | 16 |
| $V_{hv}$ | 96 | 192 | 96 | 192 | 200 | 400 | 200 | 400 |
| $d_1$ | 0.33 | 0.33 | 0.23 | 0.23 | 0.33 | 0.33 | 0.23 | 0.23 |
| $d_2$ | 0.47 | 0.23 | 0.66 | 0.33 | 0.22 | 0.11 | 0.32 | 0.16 |

Next, the voltage blocking and current requirements for the MOSFET, IGBT, and freewheeling diodes are calculated. Equation 3.2, Equation 3.5 and Equation 3.6 are used to calculated the values for $S_1$. Equation 3.12, Equation 3.16, and Equation 3.17 are used to calculated the values for $D_1$. Equation 3.12* $N_1 / N_2$, Equation 3.18, and Equation 3.19 are used to calculate the values for $S_2$. Finally, Equation 3.2* $N_1 / N_2$, Equation 3.9, and Equation 3.10 are used to calculated the values for $D_2$. These values are shown in Table 3.4.

Table 3.4: Calculated current and voltage levels for the power electronic components

| | | Electric-Lion | | | | HyLion | | | | Max |
|---|---|---|---|---|---|---|---|---|---|---|
| | $V_{bm}$ | 8 | 8 | 16 | 16 | 8 | 8 | 16 | 16 | |
| | $V_{hv}$ | 96 | 192 | 96 | 192 | 200 | 400 | 200 | 400 | |
| | $V_{bl}$ | 13.6 | 19.3 | 21.6 | 27.3 | 19.8 | 31.5 | 27.8 | 39.5 | 39.5 |
| | $i_{pk}$ | 60.5 | 60.5 | 85.5 | 85.5 | 60.5 | 60.5 | 85.5 | 85.5 | 85.5 |
| $S_1$ | $I_{DC}$ | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 |
| | $I_{RMS}$ | 20.1 | 20.1 | 23.9 | 23.9 | 20.1 | 20.1 | 23.9 | 23.9 | 23.9 |
| | $V_{bl}$ | 13.6 | 19.3 | 21.6 | 27.3 | 19.8 | 31.5 | 27.8 | 39.5 | 39.5 |
| | $i_{pk}$ | 60.5 | 60.5 | 85.5 | 85.5 | 60.5 | 60.5 | 85.5 | 85.5 | 85.5 |
| $D_1$ | $I_{DC}$ | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 | 10.0 |
| | $I_{RMS}$ | 20.1 | 20.1 | 23.9 | 23.9 | 20.1 | 20.1 | 23.9 | 23.9 | 23.9 |
| | $V_{bl}$ | 232.0 | 328.0 | 368.0 | 464.0 | 336.0 | 536.0 | 472.0 | 672.0 | 672.0 |
| | $i_{pk}$ | 3.6 | 3.6 | 5.0 | 5.0 | 3.6 | 3.6 | 5.0 | 5.0 | 5.0 |
| $S_2$ | $I_{DC}$ | 0.8 | 0.4 | 1.7 | 0.8 | 0.4 | 0.2 | 0.8 | 0.4 | 1.7 |
| | $I_{RMS}$ | 1.4 | 1.0 | 2.4 | 1.7 | 1.0 | 0.7 | 1.6 | 1.2 | 2.4 |
| | $V_{bl}$ | 232.0 | 328.0 | 368.0 | 464.0 | 336.0 | 536.0 | 472.0 | 672.0 | 672.0 |
| | $i_{pk}$ | 3.6 | 3.6 | 5.0 | 5.0 | 3.6 | 3.6 | 5.0 | 5.0 | 5.0 |
| $D_2$ | $I_{DC}$ | 0.8 | 0.4 | 1.7 | 0.8 | 0.4 | 0.2 | 0.8 | 0.4 | 1.7 |
| | $I_{RMS}$ | 1.4 | 1.0 | 2.4 | 1.7 | 1.0 | 0.7 | 1.6 | 1.2 | 2.4 |

Finally, the power levels of the CEC are calculated when transferring 10 A of average battery module current. Because the components have been modeled as ideal, the input and output power levels are equal. Equation 3.7 and Equation 3.20 are used to

calculate the power levels during a discharge and charge sequence of 10 A of average battery module current. The results are shown in Table 3.5.

Table 3.5: Power levels when transferring 10 A of average battery module current

| | Electric-Lion | | | | HyLion | | | |
|---|---|---|---|---|---|---|---|---|
| $V_{bm}$ | 8 | 8 | 16 | 16 | 8 | 8 | 16 | 16 |
| $V_{hv}$ | 96 | 192 | 96 | 192 | 200 | 400 | 200 | 400 |
| $P_{DIS}$ | 80.0 | 80.0 | 160.0 | 160.0 | 80.0 | 80.0 | 160.0 | 160.0 |
| $P_{CH}$ | 80.0 | 80.0 | 160.0 | 160.0 | 80.0 | 80.0 | 160.0 | 160.0 |

### 3.3.2 Power Electronics

With the voltage, current, and power levels of the converter calculated, the power electronic components can be specified. From the data in Table 3.4, the module side MOSFET and diode must be able to block 39.5 V, and be able to conduct an average current of 10 A and peak current of 85.5 A. The bus side IGBT and diode must be able to block 672 V, and be able to conduct an average current of 1.7 A and peak current of 5 A. The selected components and their characteristics are shown in Table 3.6.

Table 3.6: Selected electronic switches and diodes characteristics

| Abbreviation | Device | Blocking Voltage (V) | Continuous Current (A) | Peak Current (A) | Other |
|---|---|---|---|---|---|
| $S_1$ | IRFB4110PbF | 100 | 180 | 670 | $R_{DS}(ON)$ = 3.7 Ωm |
| $D_1$ | 30CTQ100PbF | 100 | 30 | 850 | $V_F$ = 0.67 V |
| $S_2$ | IRG4PH40UPbF | 1200 | 21 | 82 | $V_{CE}(ON)$ = 2.43 V |
| $D_2$ | 10ETF12 | 1200 | 10 | 160 | $V_F$ = 1.33 V |

All of the devices selected meet the current and voltage specifications of the circuit. These particular devices have low on-state voltages in comparison to other

devices of similar performance and, consequently, help to minimize circuit loses. With these components selected, the final power electronic device to be specified is the transformer.

The MAGNETICS Corporation will be used to supply the transformer. They offer three difference core types, and the ferrite core composition will be used. Ferrite cores are economical, have a large selection of shapes, and are well suited for switching frequencies of 10 kHz to 50 MHz [32]. The ferrite cores can be composed of F, P or R materials. The various compositions have different core losses at specific temperatures. The material P will be selected because it has the lowest core loss at operating temperatures from 30 °C to 80 °C [33]. With 25 kHz excitation, the P material will maintain a maximum core loss of 100 mW/cm$^3$ when flux levels are below 2000 G at 25 °C and 3000 G at 60 °C to 80 °C [33].

With the type and composition of the transformer selected, the next step is to choose the shape. The BMS will have a cluster of either 12 or 25 transformers, depending on the vehicle, which will be packaged together tightly. Therefore, significant shielding and minimal space requirements are critical to the design. The pot core shape excels in these characteristics, so it will be used [32].

Next, the size of the pot core is to be selected. A common approach to determine the power handling capability of a transformer core is its available core window area, effective cross-sectional area product (WaAc). The WaAc product is approximated by Equation 3.21, where $P_o$ is the output power in W, $C$ is the current capacity in cm$^2$/A, $e$

is the transformer efficiency, $B$ is the flux density in G, $f$ is the switching frequency in Hz, and $K$ is the winding factor [32].

$$WaAc = \frac{10^8 P_o C}{4eBfK} \tag{3.21}$$

The values that MAGNETICS recommends for a pot-core transformer with square-wave excitation are shown in Table 3.7.

Table 3.7: Recommended values for a pot-core transformer with square-wave excitation

| Variable | Value | Comment |
|---|---|---|
| C | 5.07E-03 | square wave excitation of pot core |
| e | 0.9 | transformer |
| K | 0.3 | pot core |

The maximum output power $P_o$ of the CEC is 160 W, and the switching frequency $f$ is 25 kHz. The flux level $B$ is set to 2000 G to maintain a maximum 100 mW/cm$^3$ core loss density. These values result in the WaAc product for the CEC to be 1.5. The MAGNETICS pot core 0_43622UG has a WaAc product of 1.53, and is chosen to be the pot core for the CEC. It has a two-section bobbin 00B362202 which will simplify the winding of the transformer, and a mounting clamp 00C362217 to secure the transformer.

The number of turns on the module side of the transformer $N_{bm}$ is set by Equation 3.22, where $L_m$ is the desired magnetizing inductance in mH, and $A_L$ is the inductance rating of the transformer core in mH/1000turns [32].

$$N_{bm} = 10^3 \sqrt{\frac{L_m}{A_L}} \tag{3.22}$$

The minimum $A_L$ value for the 0_43622UG pot core is 222 mH/1000turns, and is specified by the gap code A222. For the desired magnetizing inductance of 1.75 μH, $N_{bm}$ is set to 3. With the winding ratio previously set to 17, the number of turns on the bus side of the transformer $N_{hv}$ is set to 51. The bobbin has an average length of 7.4 cm per turn. The transformer is going to be mounted separately from the circuit board, and will add an estimated 10 cm to the length of each winding. Therefore, the module side winding will approximately be 32.2 cm in length and the bus-side winding will be approximately 387.4 cm in length.

With the number of turns specified, the size of the wire used for the transformer can be determined. Litz wire will be used to minimize the ac resistive loses that are caused by the high-frequency skin effect. The first number in the litz wire description is the number of strands, and the second number is the AWG of each strand. Strands of 36 AWG are recommended for switching frequencies of 20 kHz to 50 kHz [34]. To determine the maximum OD of the wire, the dimensions of the bobbin window are analyzed. The base of the bobbin window is 6.1 mm, and the height of the bobbin window is 5.86 mm. To maximize the coupling of the module side inductance to the bus side inductance, the module side windings are kept to one layer. Therefore, the outer diameter of the module side windings must be less than 2.03 mm. This requirement is met by 105/36 litz wire. The bus side windings require multiple layers to reach 51 turns. With 8 turns per layer and 7 layers total, the outer diameter of the bus side windings must be less than 0.76 mm. This requirement is met by 16/36 litz wire.

### 3.3.3 Control

To provide the gate drive signal to the electronic switches and short-circuit protection, a current-mode controller will be used. It is a bipolar totem-pole drive, and the use of a 4.7-Ω gate resistor and Schottky protection diodes will prevent ringing and damage to the drive circuitry [35]. At the start of a switching period, the electronic switch is turned on and the switch current ramps up. When it reaches a desired level it is turned off, and it remains off until the beginning of the next switching period. In a typical dc-to-dc converter, the desired peak current level is set by an error amplifier that measures the output voltage and compares it to a set output level. For the CEC, a variable, desired peak current signal will be sent to the current-mode controller from the microcontroller. The microcontroller will then monitor the average current, and adjust the desired peak current signal as needed to maintain the desired average current. The current-mode controller LT1243 is selected because it has a maximum duty cycle of 96%, a wide operating temperature range, and flexibility in the access to the error amplifier signals.

To set the peak current level, the current-mode controller provides access to an input feedback pin and an output compensation pin. There is an internal op-amp, and the feedback pin is wired to the negative terminal of the op-amp. The positive terminal of the op-amp is internally wired to 2.5 V, and the output of the op-amp is wired to the compensation pin. The peak current cutoff is directly proportional to the compensation pin voltage. When the voltage of the compensation pin is below 1.4 V, the output duty cycle is zero. Any voltage above 4.4 V is internally clamped. For a compensation

voltage $V_{comp}$ between 1.4 V and 4.4 V, and a shunt resistance $R_s$, the peak current is approximated by Equation 3.23.

$$\hat{i} = \frac{V_{comp} - 1.4}{3R_s} \quad 1.4 \le V_{comp} \le 4.4 \qquad 3.23$$

The maximum input voltage of the current sense pin on the current-mode controller is 1 V. Therefore, a 0.01-Ω, 7-W shunt resistor is placed in series with the MOSFET $S_1$ to generate a 0 to 0.85-V signal. This represents the peak current value of $S_1$. Similarly, a 0.2 Ω, 3-W shunt resistor is placed in series with the IGBT $S_2$ to generate a 0 to 1-V signal. This represents the peak current value of $S_2$.

To generate the desired peak current signal, the microcontroller will send isolated, serial-peripheral interface (SPI) signals to a digital-to-analog converter (DAC). The ACSL-6400 digital isolation chip will be used to isolate the SPI signals. It provides 1000 V of common-mode isolation, and is capable of high-speed data transmission at 15 MBd. The MCP4921 DAC will be used to generate an analog, peak-current control signal. It has 12-bit resolution, rail-to-rail output, and an SPI interface. The output of the DAC is 0 V at power-up and up to 5 V at full power.

An op-amp circuit will be used to convert the 0 V output of the DAC to 0.7 V at the $V_{comp}$ pin, and the 5 V output of the DAC to 4.4 V at the $V_{comp}$ pin. The set-point 0.7 V at the $V_{comp}$ when the DAC output is 0 V was established to accommodate op-amp offset voltages and resistor tolerances. The LT1097 chip will be used for the op-amp. It is a low cost, precision op-amp. A circuit schematic of the gate drive is shown in Figure 3.5.

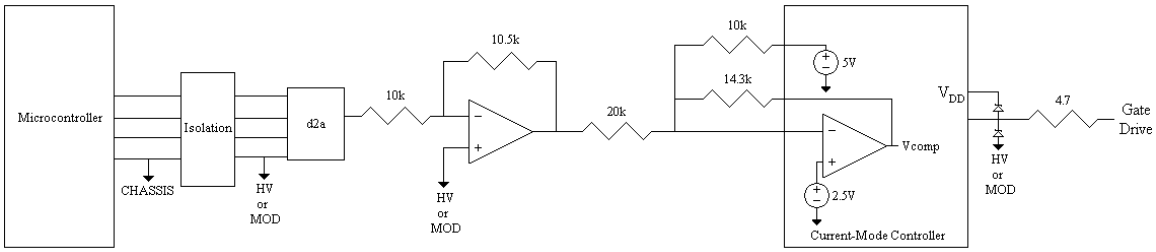



Figure 3.5: Gate drive schematic

The relationship between the peak current $\hat{i}_{S1}$ of the MOSFET $S_1$ to the DAC output voltage $V_{d2a}$ is approximated by Equation 3.24. The relationship between the peak current $\hat{i}_{S2}$ of the IGBT $S_2$ to $V_{d2a}$ is approximated by Equation 3.25.

$$\hat{i}_{S1} = 25.03V_{d2a} - 22.92 \quad 0.92 \le V_{d2a} \le 4.91 \tag{3.24}$$

$$\hat{i}_{S2} = 1.25V_{d2a} - 1.15 \quad 0.92 \le V_{d2a} \le 4.91 \tag{3.25}$$

With the gate drive circuits designed, the remaining control circuitry to be designed handles the feedback signals. The respective voltages of the battery module and high-voltage bus will be filtered with single-pole, low-pass, RC filters. They then will be isolated, and sent to the microcontroller. The voltage of the battery module will also be sent to the MASTER controller to allow it to monitor the SOC of that battery module. The average current of the MOSFET $S_1$ and freewheeling diode $D_1$ will be filtered with an active realization of a two-pole, low-pass, Butterworth filter. They then will be isolated and sent to the microcontroller.

The voltage signal of the battery module will be filtered with a single-pole, RC filter, with a cutoff frequency of 100 Hz. The maximum voltage to be measured is 16 V.

The analog-to-digital converter (ADC) of the MASTER controller can measure up to 10 V, and the ADC of the microcontroller can measure up to 5 V. Therefore, the passband gain of the RC filter will be set to 5/8. This signal will be isolated, and then sent to the MASTER controller. It also will be sent to a voltage divider with a gain of 1/2, and then sent to the microcontroller. The ISO124U chip will provide the analog isolation. It provides 1500 V of isolation, and is the most inexpensive, analog isolation chip available. The transfer function of the battery module voltage to the microcontroller ADC voltage $H_{Vbm}(s)$ is approximated by Equation 3.26. The circuit schematic of the module RC filter is shown in Figure 3.6, and the Bode magnitude and phase plots are shown in Figure 3.7.

$$H_{Vbm}(s) = \frac{0.3125}{0.00165s + 1} \qquad 3.26$$

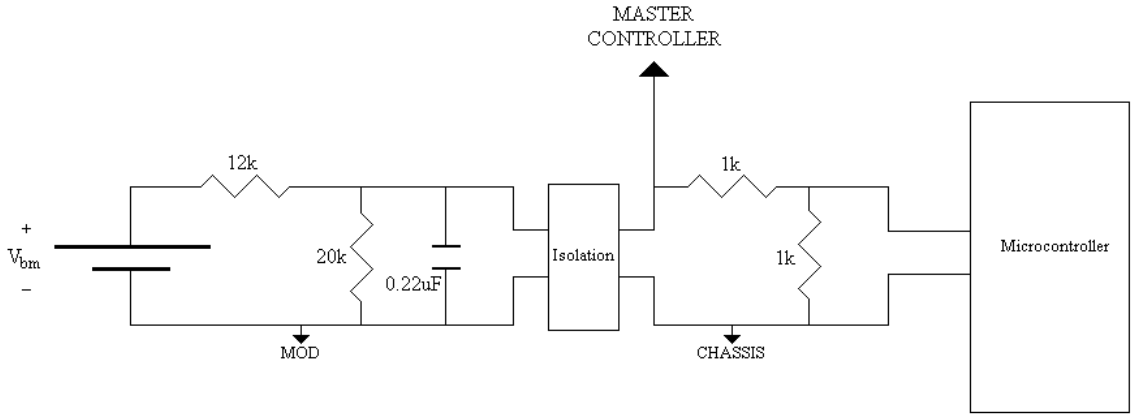



Figure 3.6: Single-pole, low-pass RC filter of the battery module voltage

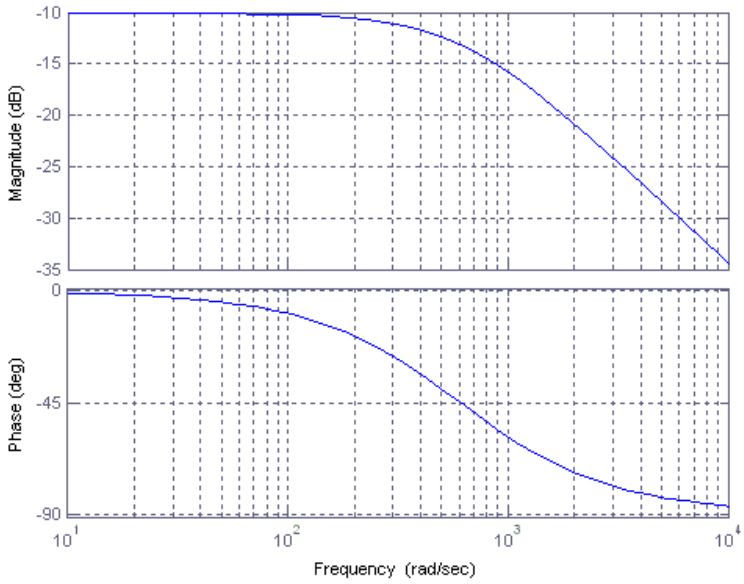



Figure 3.7: Bode magnitude and phase plots of the battery module voltage RC filter

The voltage signal of the high-voltage bus will be filtered with a single-pole, RC filter, with a cutoff frequency of 100 Hz. The maximum voltage to be measured is 400 V. The ADC of the microcontroller can measure up to 5 V. Therefore, the passband gain of the RC filter will be set to 1/80. This signal will be isolated, and then sent to the microcontroller. The ISO124U chip will provide the analog isolation. It provides 1500 V of isolation, and is the most inexpensive, analog isolation chip. The transfer function of the high-voltage bus voltage to the microcontroller ADC voltage $H_{Vhv}(S)$ is approximated by Equation 3.27. The circuit schematic of the bus voltage RC filter is shown in Figure 3.8, and the Bode magnitude and phase plots are shown in Figure 3.9.

$$H_{Vhv}(s) = \frac{0.0361}{0.001735s + 1} \qquad 3.27$$

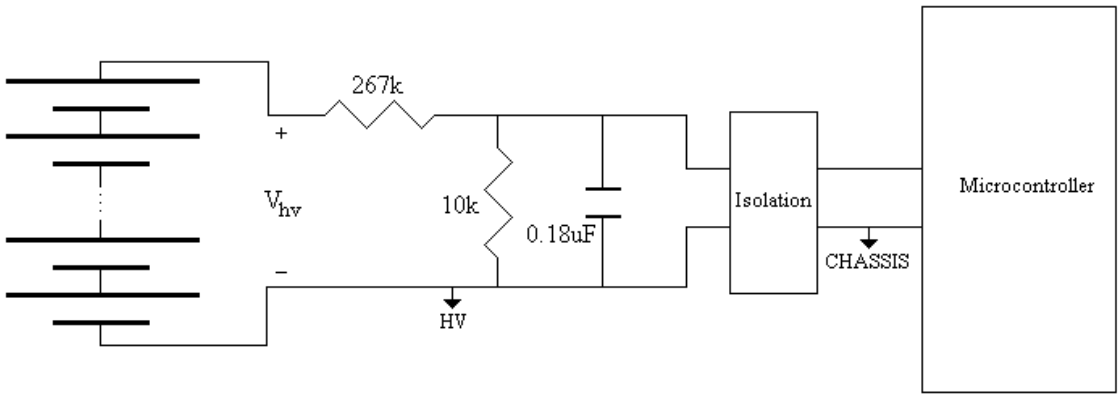



Figure 3.8: Single-pole, low-pass RC filter of the high-voltage bus voltage

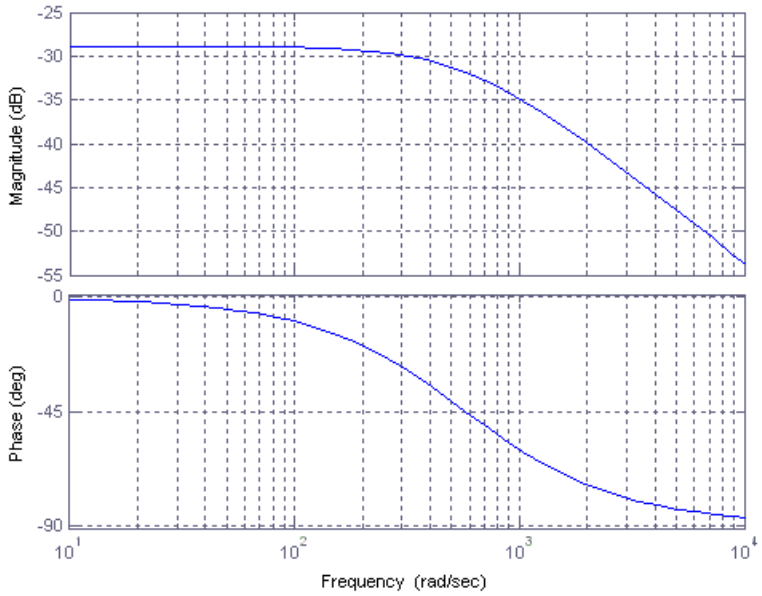



Figure 3.9: Bode magnitude and phase plots of the high-voltage bus voltage RC filter

The current signals of the MOSFET $S_1$ and freewheeling diode $D_1$ will be filtered by active realizations of a two-pole, low-pass, Butterworth filter. The cutoff frequencies will be 25 Hz, so that the 25-kHz switching frequency is three orders of

magnitude above the cutoff frequencies. The voltage representing the current signal of $S_1$ is a triangle wave that ranges from 0 to 0.85 V. The average value of this waveform ranges from 0 to 0.1 V. The voltage representing the current signal of $D_1$ is the inverse of the $S_1$ waveform. It will be passed through a unity-gain, inverting op-amp to compensate for this. The average voltage values will be amplified by the passband gain of the filters, isolated, and sent to the microcontroller. The passband gain is set to 31 to amplify significantly the average voltage signals. Ideally, a gain of 50 would be used to set the range of these signals equal to the range of the ADC on the microcontroller. However, this has been reduced to prevent saturation of the ADC that would be caused by the offset voltages that are inherent to practical op-amps.

The transfer function of the discharging-current voltage signal to the microcontroller ADC voltage $H_{DIS}(s)$ is approximated by Equation 3.28. The transfer function of the charging-current voltage signal to the microcontroller ADC voltage $H_{CH}(s)$ is approximated by Equation 3.29. The circuit schematics for the discharging-current signal active filter and the charging-current signal active filter are shown in Figure 3.10. Finally, the Bode magnitude and phase plots are shown in Figure 3.11 for the discharging-current active filter, and in Figure 3.12 for the charging-current active filter.

$$H_{DIS}(s) = \frac{31}{0.00003988s^2 + 0.00952s + 1} \tag{3.28}$$

$$H_{CH}(s) = \frac{-31}{0.00003988s^2 + 0.00952s + 1} \tag{3.29}$$

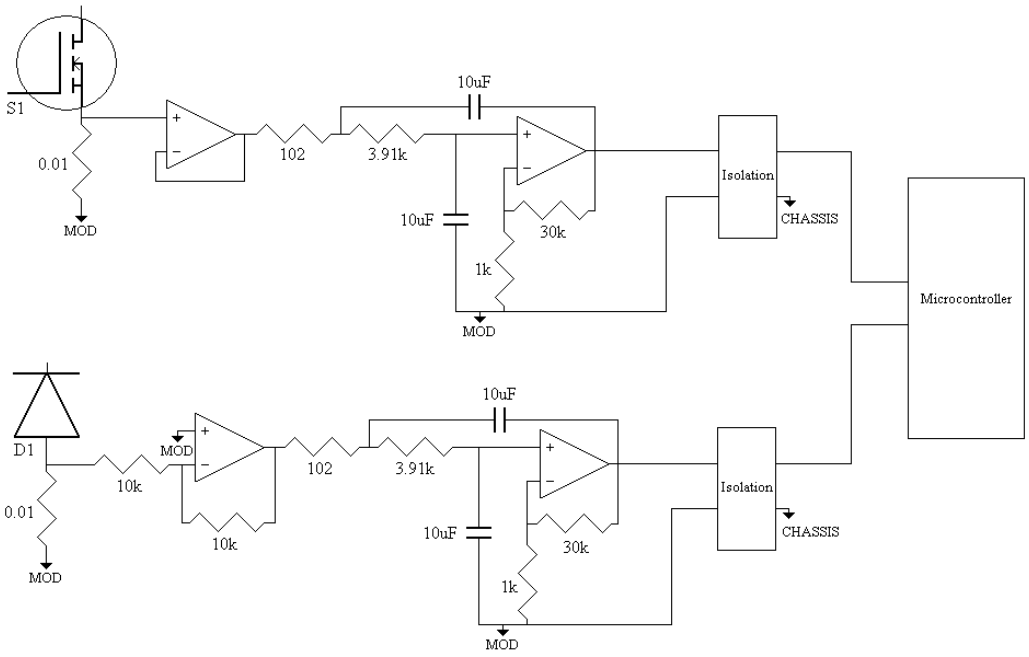



Figure 3.10: Circuit schematics of the current-signal active filters

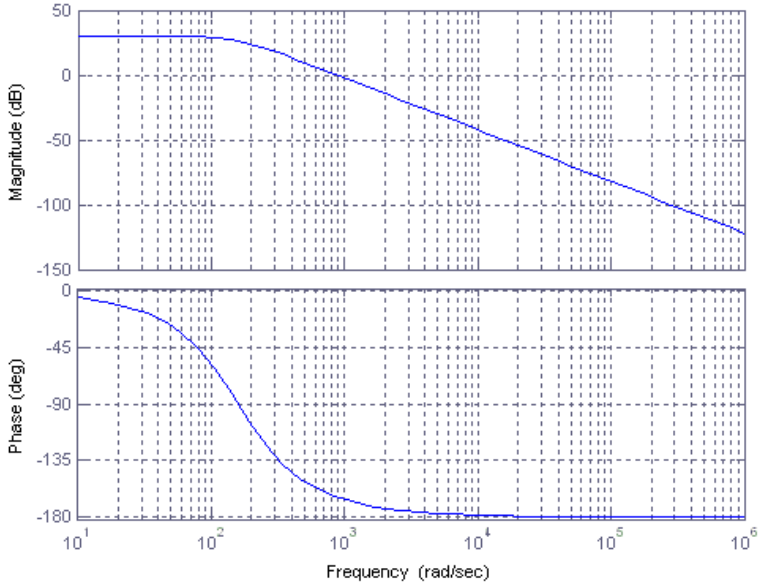



Figure 3.11: Bode magnitude and phase plots of the discharging-current active filter

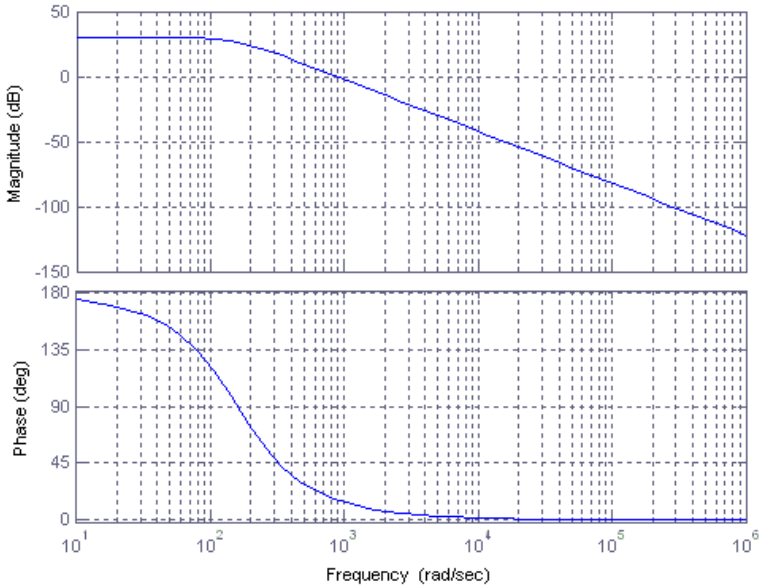



Figure 3.12: Bode magnitude and phase plots of the charging-current active filter

### 3.3.4 Communication

A multi-drop communication network is desired to reduce the number of communication lines between the master controller and the individual CECs. The system is to be placed within the same cavity as the battery pack, and must be able to endure the large EMI that is present. An RS485 communication network meets these requirements. The device SN65HVD23D is selected as the RS485 transceiver. It has a reduced unit-load to allow up to 256 devices per network, 5-V operation, and an extended common-mode operation for harsh operating conditions. The schematic diagram of the communication network is shown in Figure 3.13.

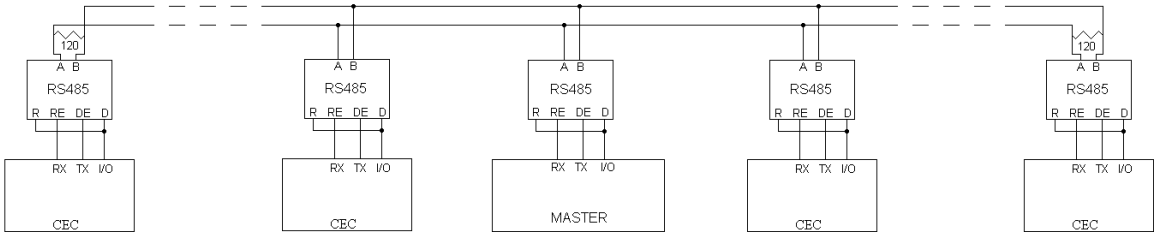



Figure 3.13: Schematic diagram of the RS485 communication network

The master controller is the central node of the system. To suppress reflections present in the RS485 communication channels, two 120-Ω resistors are placed across channels A and B at the end nodes. Communication is asynchronous, and the baud rate is 57600 b/s. Bytes are transmitted with one start bit, eight data bits, and one stop bit. Data bits are transmitted with least-significant bit first. Each message has a checksum, which is two bytes in length and is sent out in big-endian format (most significant byte is transmitted first).

All communication is initiated by the master controller. During a communication sequence, the master controller will send a message to a particular CEC. The CEC will then reply to that message, and act on it accordingly. The CEC can not initiate communication on its own, it can only reply to a master message. Each node on the communication link has a specific address. The address of the master controller is zero. The addresses of the CEC begin at one, and increase as needed.

The general message structure is shown in Table 3.8. Every message will start with two start bytes. The next byte provides the length of message, which includes the two start bytes and the two checksum bytes. Then the source address byte is provided, which indicates the node that is sending the message. The next byte is the destination

address, which indicates the node that the message is intended for. Then the command byte is provided, which indicates the specific message type. Following this are the data bytes, which vary in number depending on the message. Finally, the last two bytes of the message are the checksum. It is the sum of all of the bytes in the message. The checksum is transmitted with the most-significant byte first.

Table 3.8: General communication message structure

| | Start Byte 1 | Start Byte 2 | Length of Packet | Source Address | Destination Address | Command | Data Bytes | Check Sum |
|---|---|---|---|---|---|---|---|---|
| **Type** | Byte | Byte | Byte | Byte | Byte | Byte | Variable | 2 Bytes |
| **Value** | 0xAA | 0xAA | | | | | | |

The master controller has several messages it can transmit to the CEC. The flash LED request message is used to test communications between the master controller and the CEC. The CEC will flash the error (red) and equalization in progress (blue) LEDs, and send a reply message. The status request message is used to get the present status of the CEC. The CEC will reply with an error code (voltage or current levels out of range), the direction of the current (charge or discharge the battery module), the average magnitude of the current, and the duration of the present equalization sequence. The initiate equalization sequence message is used to instruct a CEC to transfer a specific magnitude of current, in a certain direction, and for a set duration of time. The CEC will reply with the direction of current, magnitude of current, and duration of the requested equalization sequence. The equalization completed request is used to acquire the statistics of the previously completed equalization sequence. The CEC will reply with an error code, the direction of the current, the average magnitude of current, and the duration

of the previous equalization sequence. If an error occurred during the equalization sequence, the sequence will stop at the instant of the error. The reply will contain the statistics of the equalization sequence that was completed before the error occurred. Finally, the stop equalization sequence message is used to instruct a CEC to stop a present equalization sequence prematurely. The CEC will reply with an acknowledgement of the stop command.

A table of the MASTER and CEC messages is provided in the Appendix A.

### 3.3.5 Microcontroller

The microcontroller is to provide the supervisory control of the CEC. It must have four ADC input pins to measure the feedback control signals. In addition, an SPI is required to generate the gate drive control signals. Also, a universal, asynchronous receiver/transmitter (UART) is required to communicate on the RS485 link. Finally, it must accept an external oscillator, which will provide a clock signal with high precision. This is critical for the asynchronous communication.

The microcontroller will also control three LEDs. A green LED will flash every three seconds. This provides a visual signal that the microcontroller is running. A blue LED will turn on during an equalization sequence. Finally, a red LED will turn on when an error is detected. A voltage error is set when the module voltage is measured as below 7 V or above 17 V, or the bus voltage is measured as below 90 V or above 410 V. A current error is set when, during an equalization sequence, the equalization current is measured as above 15 A.

The Microchip part PIC18F2520 is selected for the microcontroller. It is an 8-bit microcontroller with 5-V operation, up to 40-MHz clock signal. It is packaged as a 28-pin, surface-mount part. The oscillator ECS-100-20-30B-TR is selected for the external oscillator. It is a 10-MHz oscillator in a surface-mount package.

### 3.3.6 Power

There are three different ground planes on the CEC. The microcontroller and RS485 transceiver are referenced to the chassis ground. The incoming 12-V power supply for the CEC is also referenced to the chassis ground. The IGBT gate drive and high-voltage bus voltage filter are referenced to the bus ground. The MOSFET gate drive, battery module voltage filter, and both the discharge and charge average-current filters are referenced to the module ground. The voltage ranges and maximum power requirements are shown in Table 3.9.

Table 3.9: Voltage and power requirements of the CEC

**Chassis Ground**

| | | Vmin (V) | Vmax (V) | Max Power (W) |
|---|---|---|---|---|
| **5V** | PIC18F2520 | 4.2 | 5.5 | 1.25 |
| | SN65HVD23 | 4.75 | 5.25 | 0.86 |
| | | | **Total Power:** | **2.01** |
| | | Vmin (V) | Vmax (V) | Max Power (W) |
| | ISO124 | +/-14.5 | +/-18 | 0.21 |
| **+/-15V** | ISO124 | +/-14.5 | +/-18 | 0.21 |
| | ISO124 | +/-14.5 | +/-18 | 0.21 |
| | ISO124 | +/-14.5 | +/-18 | 0.21 |
| | | | **Total Power:** | **0.84** |

**Bus Ground**

| | | Vmin (V) | Vmax (V) | Max Power (W) |
|---|---|---|---|---|
| **5V** | MCP4921 | 2.7 | 5.5 | 0.25 |
| | ACSL-6400-00TE | 3 | 5.5 | 0.75 |
| | | | **Total Power:** | **1** |
| | | Vmin (V) | Vmax (V) | Max Power (W) |
| | ISO124 | +/14.5 | +/-18 | 0.21 |
| **+/-15V** | LT1097 | +/-1.2 | +/-20 | 0.012 |
| | LT1243 | 8.4 | 25 | 0.15 |
| | | | **Total Power:** | **0.372** |

**Module Ground**

| | | Vmin (V) | Vmax (V) | Max Power (W) |
|---|---|---|---|---|
| | MCP4921 | 2.7 | 5.5 | 0.25 |
| **5V** | ACSL-6400-00TE | 3 | 5.5 | 0.75 |
| | LT1807 | 2.5 | 12.6 | 0.16 |
| | | | **Total Power:** | **1.16** |
| | | Vmin (V) | Vmax (V) | Max Power (W) |
| | ISO124 | +/-14.5 | +/-18 | 0.21 |
| | ISO124 | +/-14.5 | +/-18 | 0.21 |
| **+/-15V** | ISO124 | +/-14.5 | +/-18 | 0.21 |
| | LT1097 | +/-1.2 | +/-20 | 0.012 |
| | LT1243 | 8.4 | 25 | 0.21 |
| | LT1361 | +/-2.5 | +/-18 | 0.18 |
| | | | **Total Power:** | **1.032** |

The 12-V, chassis ground power rail can range from 8 V to 16 V. One on-board dc-to-dc converter will be specified to this input voltage range, and will power the remaining dc-to-dc converters. The maximum power consumption of the CEC is 6.4 W. The PTH08080WAH converter will input this 12-V power rail and provide a 5-V, chassis ground power rail. It can supply 10 W, is non-isolated, and operates with up to 93% efficiency. The 5-V power rail for the bus ground and module ground will be supplied from VWRBS2-D5-S5-SIP converters. Each can supply 2 W, provide 1500 V of isolation, and operate with up to 82% efficiency. The +/-15-V power rails for the bus ground, module ground, and chassis ground will be supplied from VWRAS2-D5-D15-SIP converters. Each can supply 2 W, provide 1500 V of isolation, and operate with up to 80% efficiency.

### 3.3.7 Software

The microcontroller provides the supervisory control for the CEC. It must communicate with the master controller, command the current-mode controllers, and monitor the battery module average current. The software is written with the MPLAB IDE software and the student-version C18 compiler. The main top level flowcharts will be discussed below. A copy of the software is provided in Appendix B.

Interrupts are utilized whenever applicable to reduce the number of software blocking functions. The ADC interrupt samples each analog feedback signal at 1 kHz. Fifty samples per signal are stored in memory to calculate moving-average values. The UART interrupt stores each incoming byte into a buffer, and sets a new-message-ready

flag when a complete message has been received. Finally, a timer overflow interrupt sets a main loop flag so that accurate timing of the main loop functions can be executed.

The top level flowchart of the main loop is shown in Figure 3.14. Every 100 ms the main loop acts to convert the ADC samples into voltage values and average current values. The current error value is calculated by subtracting the desired average current value from the measured average current value. The peak current value is then adjusted proportionally to the error. The top level adjust peak current value flowchart is shown in Figure 3.15. Every 1 ms the UART buffer is checked for a new message, and if a message has been received a reply is sent and appropriate actions are taken. Finally, the states of the LEDs are updated.

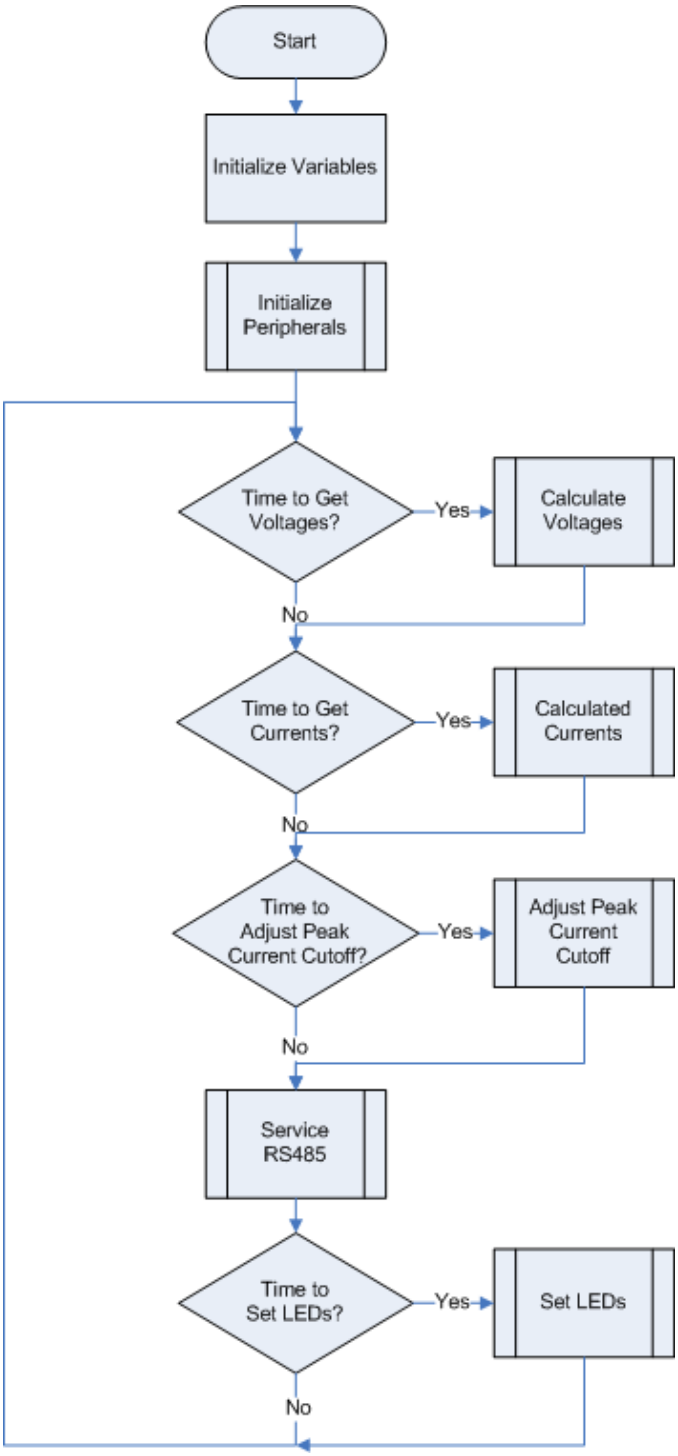



Figure 3.14: Main loop

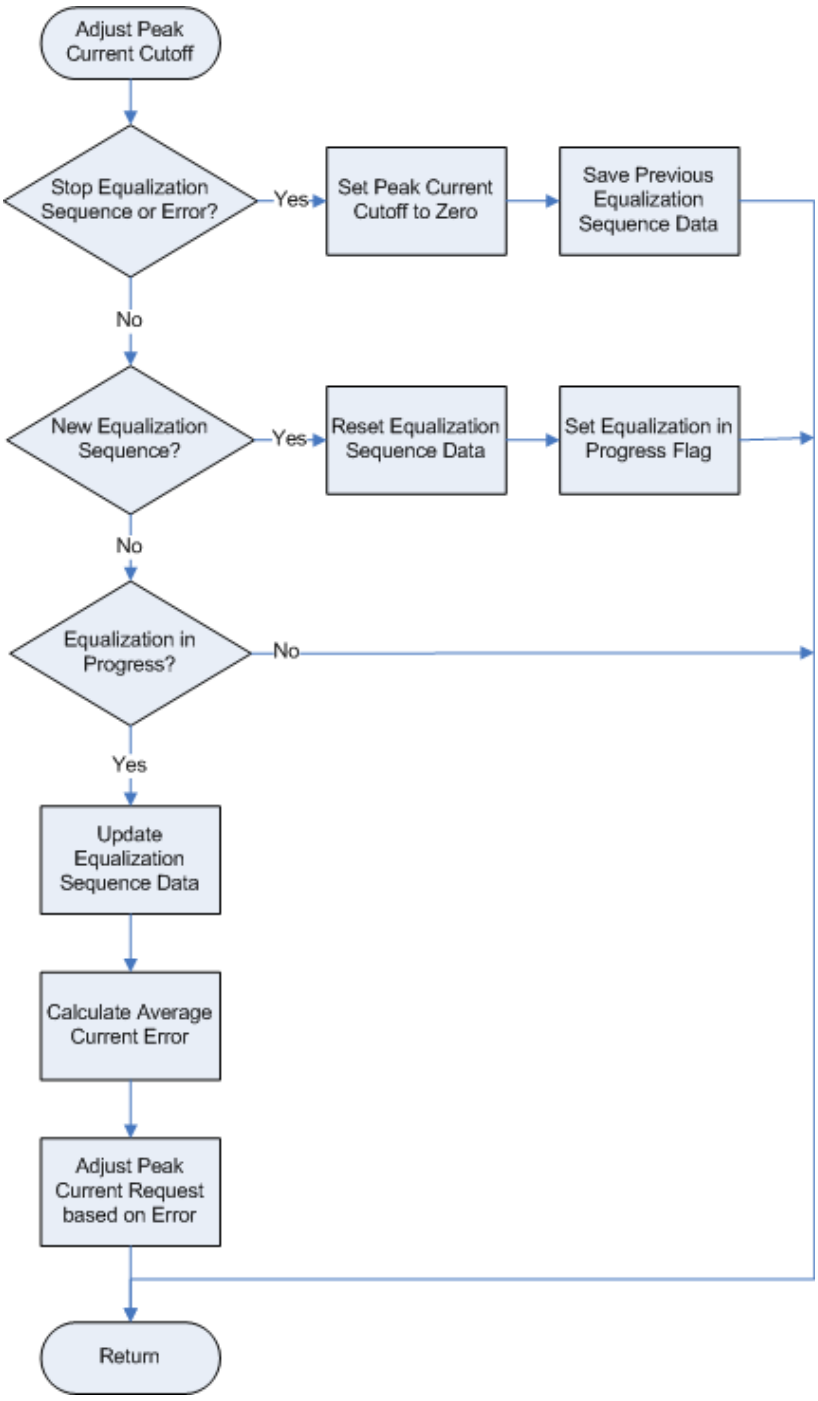



Figure 3.15: Adjust peak-current cutoff function

## 3.4 Implementation

To implement the CEC, a printed circuit board (PCB) was designed and fabricated. Then the on-board dc-to-dc converters were populated and tested over the input voltage range and isolation requirements. Next, the microcontroller was populated and tested by establishing communication with a development system. From there, the gate drive for the current-mode controllers were populated and tested. Next, the feedback signal filters were populated and tested. Finally, the inductor was gapped, wound, and connected.

### 3.4.1 Circuit Layout

The PCB fabrication was done by Advanced Circuits. The PCB was designed to meet their specifications for a $33 board. The key specifications for this special are shown in Table 3.10 [36]. The circuit layout was performed in the Easily Applicable Graphical Layout Editor (EAGLE). The completed schematic, board and parts list are shown in the Appendix C.

Table 3.10: Advanced Circuits printed circuit board specifications

- Two layers
- 1 oz copper plate
- Minimum 0.006" width traces
- Minimum 0.006" width space between copper
- Minimum 0.015" hole diameter
- Maximum 60 square inches area

### 3.4.2 On-Board DC-to-DC Converters

The first sub-circuit to be populated and tested was the on-board dc-to-dc converters. The dc-to-dc converters were populated, along with resistors across their outputs to set the minimum output load. The output voltages were tested for input voltages ranging from 8 V to 16 V. The results are shown in Table 3.11. All converters are generating voltages that are within their expected output ranges. Also, the voltage rails were driven to voltage differentials of 400 V to confirm isolation.

Table 3.11: Measured output voltages of the on-board dc-to-dc converters

| | | Input Voltage | | |
|---|---|---|---|---|
| | | 8 V | 12 V | 16 V |
| Board Voltage | 5 $V_{chas}$ | 5.00 | 4.99 | 4.99 |
| | 5 $V_{hv}$ | 5.05 | 5.04 | 5.05 |
| | 5 $V_{bm}$ | 5.05 | 5.05 | 5.05 |
| | 15 $V_{chas}$ | 15.00 | 14.99 | 15.00 |
| | -15 $V_{chas}$ | -15.02 | -15.02 | -15.02 |
| | 15 $V_{hv}$ | 14.95 | 14.93 | 14.95 |
| | -15 $V_{hv}$ | -14.95 | -14.96 | -14.95 |
| | 15 $V_{bm}$ | 15.02 | 15.01 | 15.01 |
| | -15 $V_{bm}$ | -15.03 | -15.03 | -15.04 |

### 3.4.3 Microcontroller Hello World

The next sub-circuit to be populated and tested was the microcontroller. Solder paste was used to attach the pins of the surface mount component to the board. The external oscillator, LEDs, and programming header were also populated. The In-Circuit-Debugger 2 (ICD2) from microchip was used to debug the software and program the microcontroller.

The timer0 overflow interrupt is programmed to set a main loop variable high every 1 ms. To confirm operation visually, the equalization in progress LED (blue LED) was programmed to toggle every iteration of the main loop. The image of the 'hello world' moment, the moment when the LED is switched on, is shown in Figure 3.16. Also, the period of the switching pin was verified with an oscilloscope. This confirmed that communication to the microcontroller was established and that the external oscillator was operating correctly.

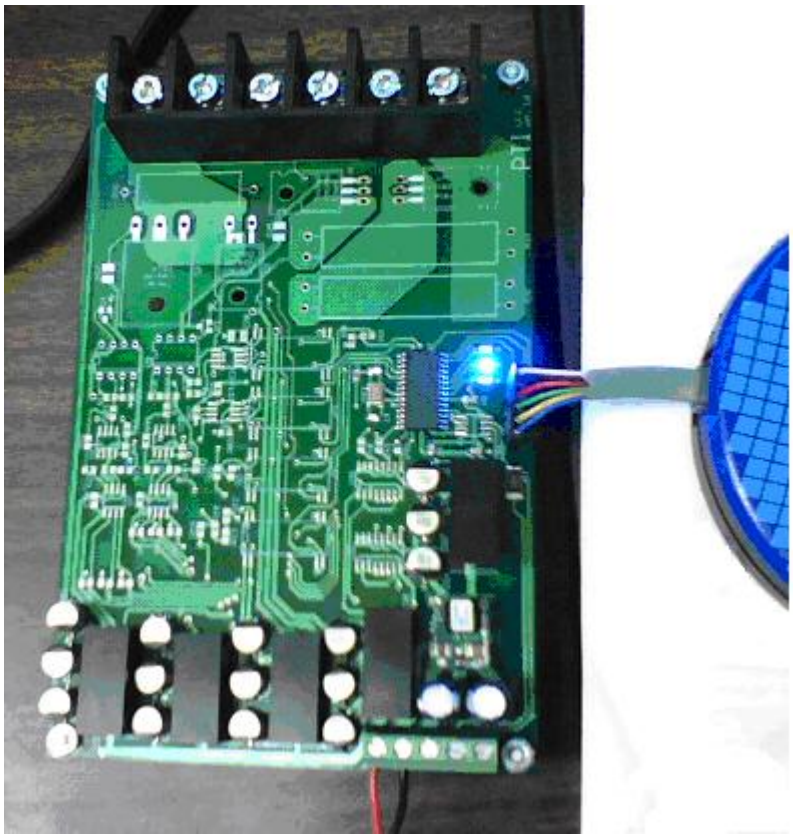


Figure 3.16: Microcontroller hello world

### 3.4.4 Communication Validation

The next sub-circuit to be populated and tested was the communication network. The UART was configured for asynchronous communication, 8-bit data bytes, and 57600 b/s. The byte 00001000 was then transmitted to verify the baud rate. The snapshot of the UART transmitting one bit is shown in Figure 3.17. The bit time was approximately 17 us, which indicates in an error of about 2%. This error is acceptable for asynchronous communication, and it verified that the baud rate was correct.

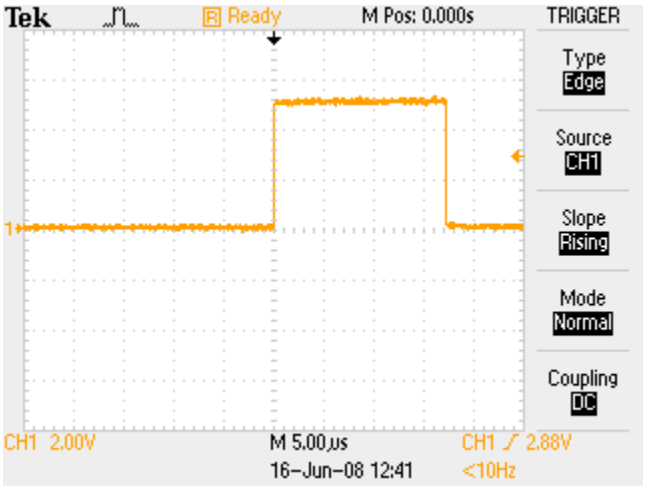



Figure 3.17: UART bit timing

With the baud rate verified, the RS485 transceiver was populated. An arbitrary packet was sent out, and the voltages on each RS485 channel were measured with respect to the chassis ground. A snapshot of the RS485 communication is shown in Figure 3.18, with the top signal showing the voltage of channel A, and the bottom signal showing the voltage of channel B.

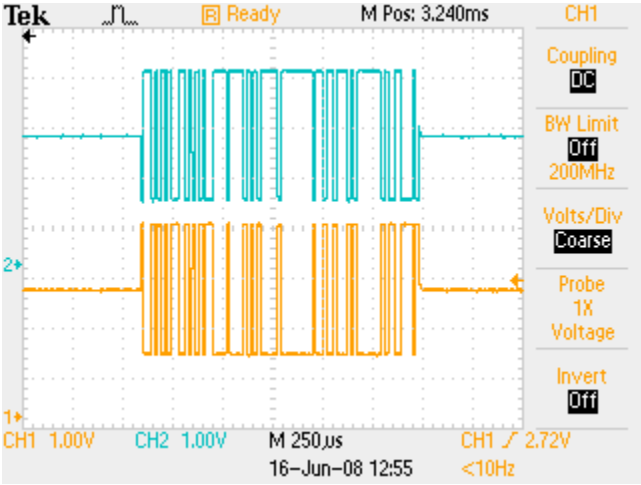



Figure 3.18: Transmission of a packet on the RS485 bus

The final test of the communication network was to have the CEC communicate with a computer. The program COMLAB is used to transmit a packet to the CEC, and to display the reply of the CEC. A universal serial bus (USB) to RS485 converter was used to connect the computer to the RS485 bus. A snapshot of a packet transmission and reply is shown in Figure 3.19. The dollar sign indicates the two following characters are hexadecimal. The top window displays the packet that was sent from the computer to the CEC. The bottom window displays the packet that was sent from the CEC to the computer, in response to the packet it received. This verified that the CEC was transmitting and receiving packets correctly.

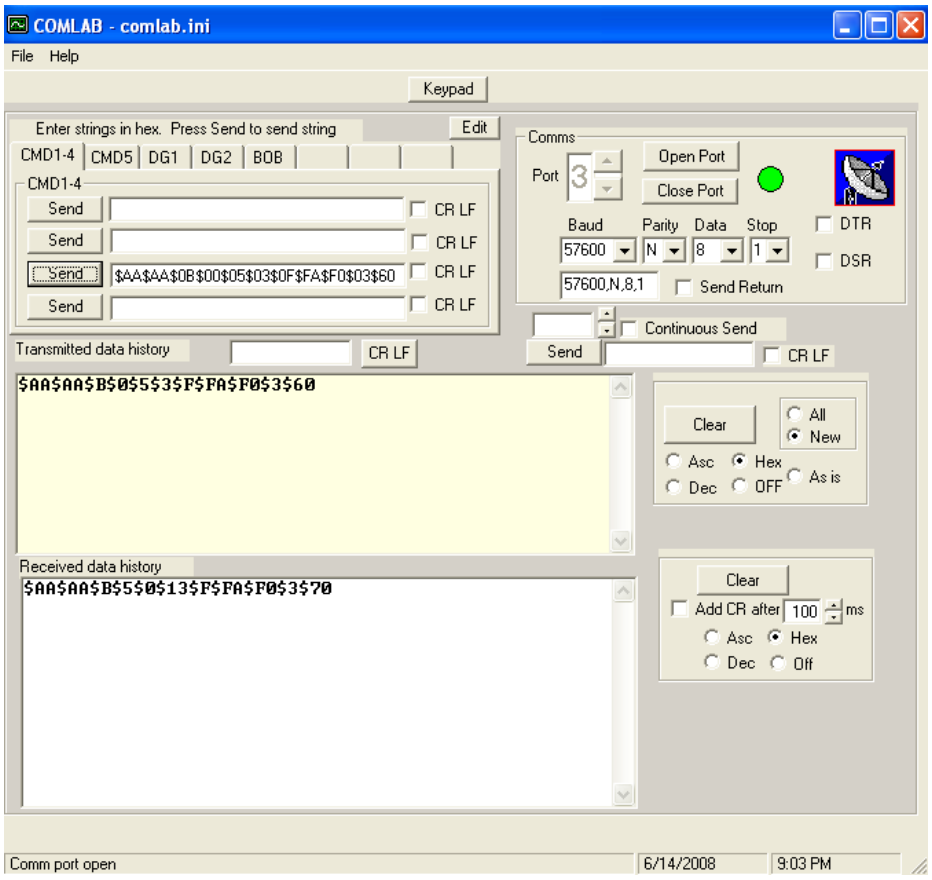



Figure 3.19: Computer to CEC packet transmission and reply

### 3.4.5 Gate Drive Validation

The next sub-circuit to be populated and tested was the gate drive circuitry. The serial out, clock, and chip select signals were passed from the microcontroller to the optical isolator, and then sent to the DAC. An unexpected characteristic of the optical isolator is that the output signals are inverted from the input signals. To compensate for this, the microcontroller SPI was setup to have an idle clock signal that is high, and the

packet to be transmitted and the chip select were inverted. A snapshot of a packet being received at the DAC is shown in Figure 3.20. The top signal (Ch 3) is the chip select (chip select is set low to accept communication), the middle signal (Ch 1) is the packet (0001 1111 1111 1111), and the bottom signal (Ch 2) is the clock.

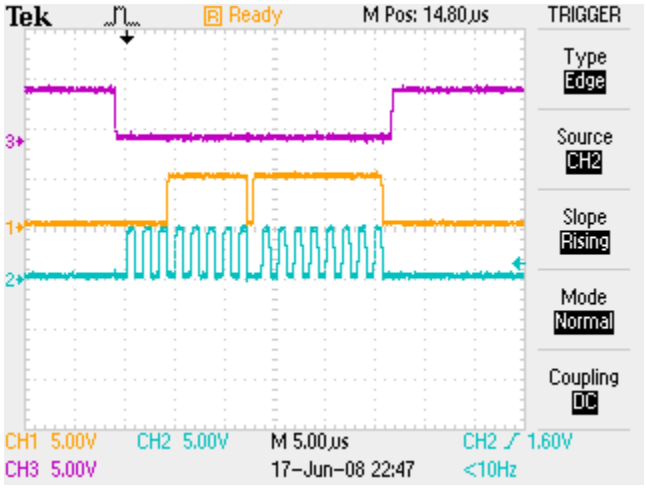
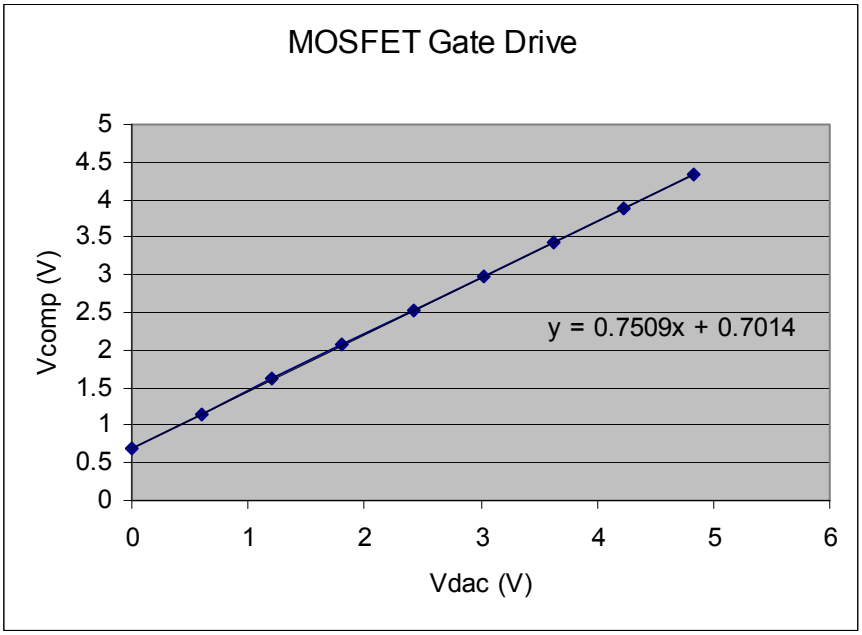



Figure 3.20: SPI communication to the DAC

With communication to the DAC established, the op-amp network that establishes a voltage at the compensation pin could be tested. When the DAC voltage is zero, the voltage at the compensation pin should be 0.7 V. When the DAC voltage is 5 V, the voltage at the compensation pin should be 4.4 V. The voltage at the compensation pin was measured as the voltage of the DAC was incremented from its minimum to maximum value. The results for both gate drives are shown in Figure 3.21.



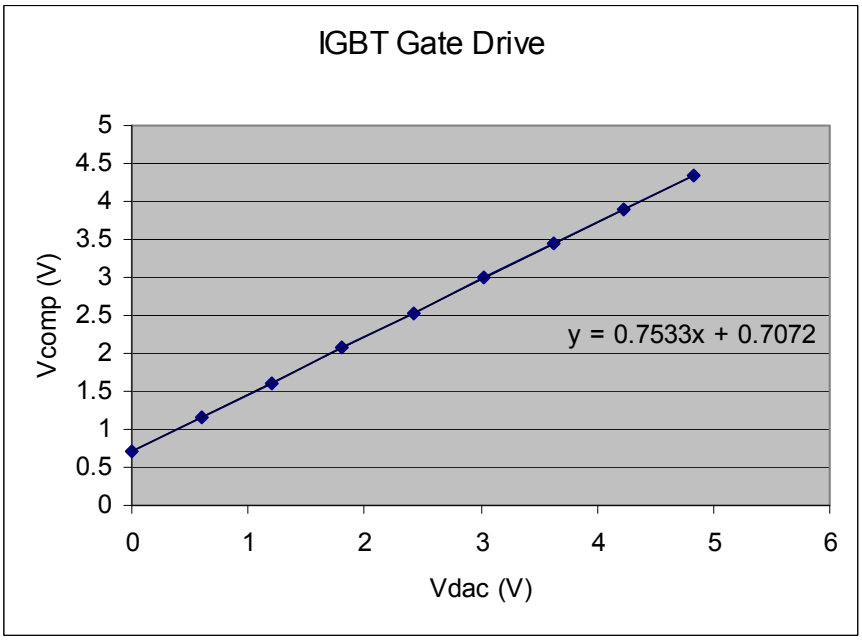




Figure 3.21: Relationship of compensation pin voltage to DAC voltage

### 3.4.6 Feedback Signal Filter Circuits Validation

The next sub-circuits to be populated and tested were the filters for the feedback signals. First, the RC circuits for the battery module voltage and high-voltage bus voltage were populated. Both circuits are first order, low-pass filters with cutoff frequencies of 100 Hz.

The battery module voltage filter has a passband gain of 0.625 V/V. Sinusoidal waves with frequencies of 10 Hz, 100 Hz, and 1 kHz were input to the filter, and the output signal was captured. Snapshots are shown in Figure 3.22. The output signal is the smaller sinusoid (Ch 1), and the input signal is the larger sinusoid (Ch 2). The 10-Hz input signal results in an output signal that is approximately in phase. The 100-Hz input signal results in an output signal that is approximately 45° out of phase. Finally, the 1-kHz input signal results in an output signal that is approximately 90° out of phase.

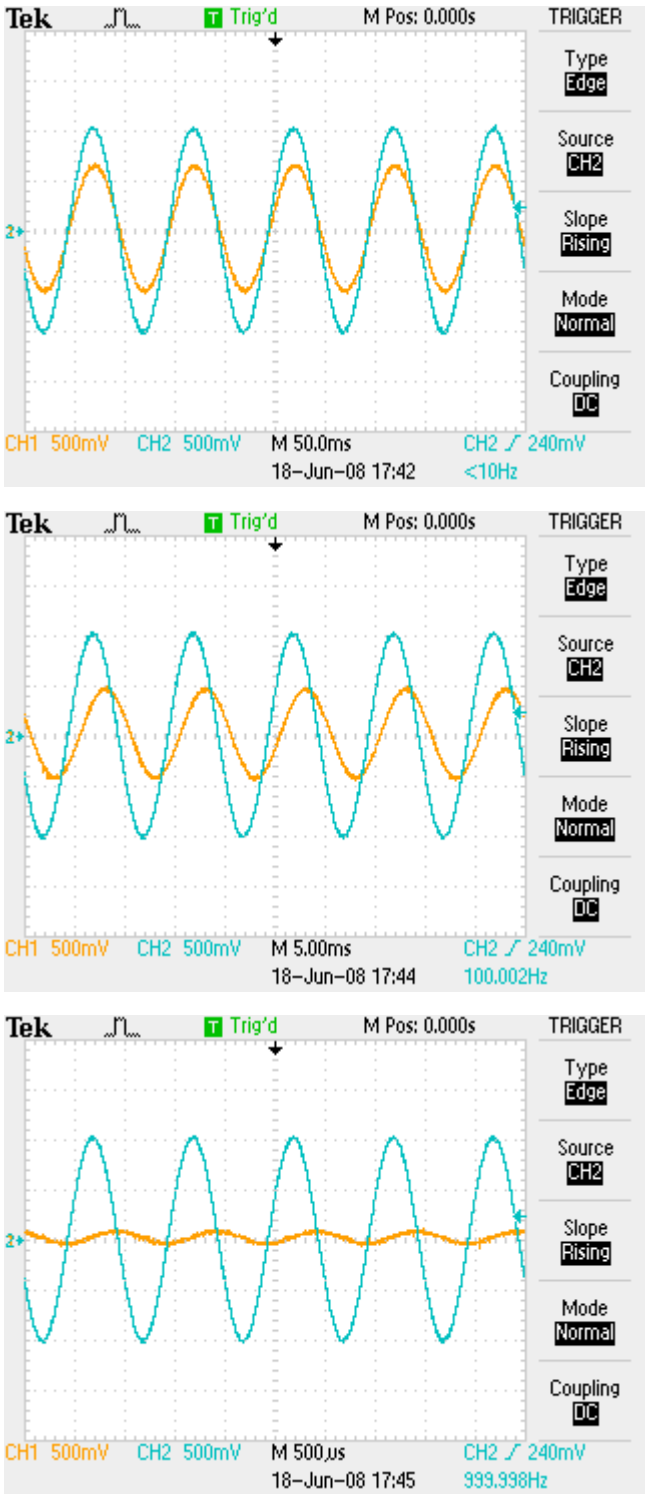



Figure 3.22: Response of the battery module voltage filter to input signals with 10 Hz, 100 Hz, and 1 kHz frequencies

The bus filter has a passband gain of 0.0125 V/V. Sinusoidal waves with frequencies of 10 Hz, 100 Hz, and 1 kHz were input to the filter, and the output signal was captured. Snapshots are shown in Figure 3.23. The output signal is the smaller sinusoid (Ch 1), and the input signal is the larger sinusoid (Ch 2). As before, the signal exhibits the expected phase shift of the first-order, low-pass filter.

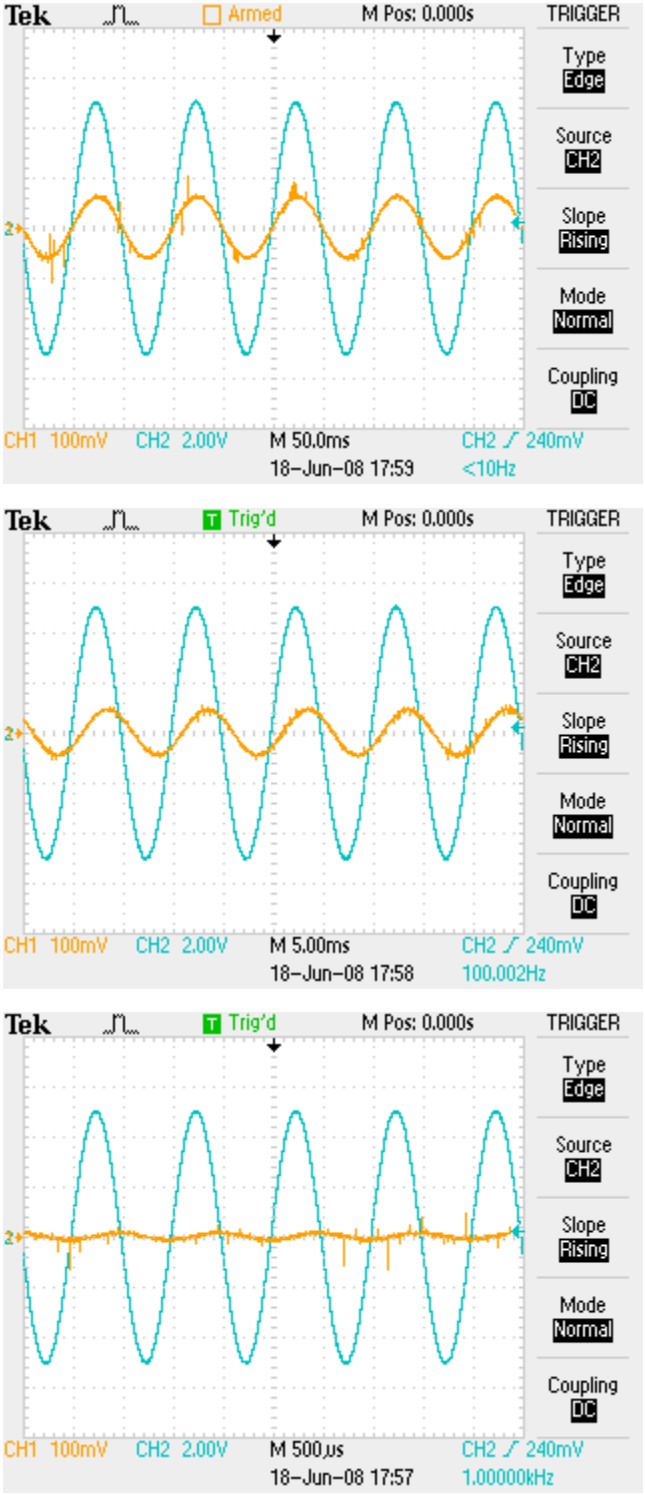



Figure 3.23: Response of the high-voltage bus voltage filter to input signals with 10 Hz, 100 Hz, and 1 kHz frequencies

The second set of filters to populate and test were the average charge and discharge filters. Both are active realizations of a second-order, low-pass Butterworth filter, with a cutoff frequency of 25 Hz. They have passband gains of 31 V/V. The charge current filter has a unity-gain, inverting op-amp placed before the filter. This results in the overall filter circuit having a passband gain of -31 V/V. The discharge current filter has a buffer op-amp placed before the filter. Therefore, it retains the passband gain of 31 V/V.

Sinusoidal signals with frequencies of 2.5 Hz, 25 Hz, and 250 Hz, and dc offsets of -50 mV, were input into the charging filter. The snapshots are shown in Figure 3.24. The output signal is the upper signal (Ch 2), and the input signal is the lower signal (Ch 1). The 2.5-Hz input signal results in an output signal that is approximately 180° out of phase. The 25-Hz input signal results in an output signal that is approximately 270° out of phase. Finally, the 250-Hz input signal results in an output signal that is approximately in phase with the input signal.

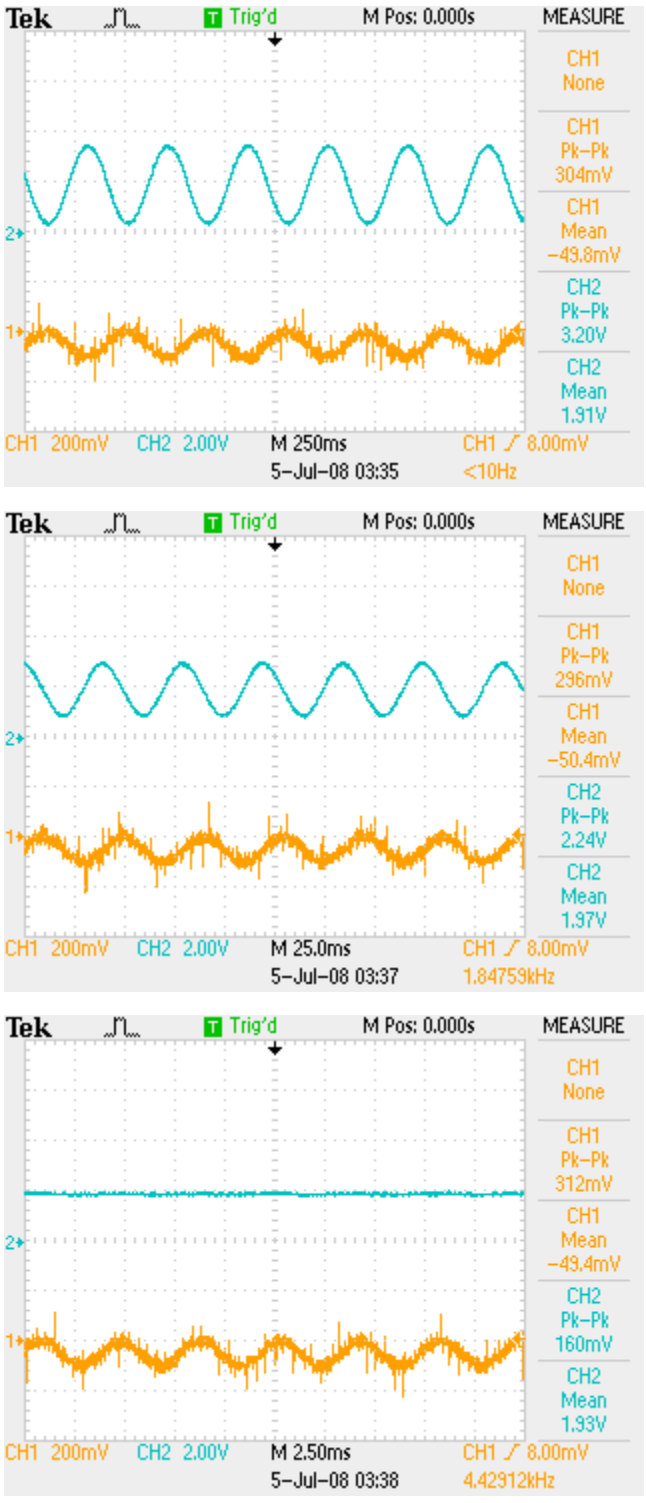



Figure 3.24: Response of the average charging current filter to input signals with 2.5 Hz, 25 Hz, and 250 Hz frequencies

Sinusoidal signals with frequencies of 2.5 Hz, 25 Hz, and 250 Hz, and dc offsets of 70 mV, were input into the discharging filter. The snapshots are shown in Figure 3.25. The output signal is the upper signal (Ch 2), and the input signal is the lower signal (Ch 1). The 2.5-Hz input signal results in an output signal that is approximately in phase with the input signal. The 25-Hz input signal results in an output signal that is approximately 90° out of phase of the input signal. Finally, the 250-Hz input signal results in an output signal that is approximately in 180° out of phase of the input signal.

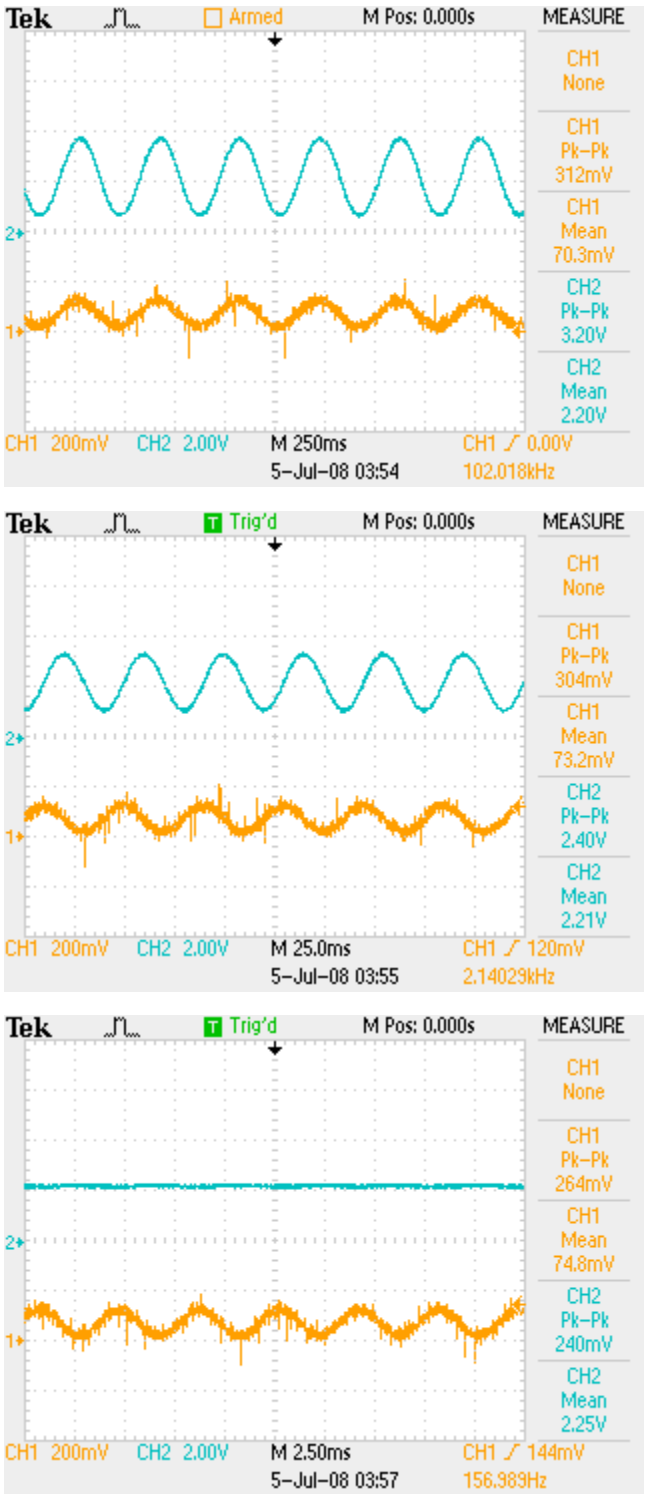



Figure 3.25: Response of the average discharging current filter to input signals with 2.5 Hz, 25 Hz, and 250 Hz frequencies

### 3.4.7 Building the Transformer

Transformer cores, when ordered in small quantities, are not available to be gapped from the distributor. Therefore, the gap required to achieve the 1.75 μH magnetizing inductance must be calculated and machined. Equation 3.30 is used to calculate the length of the gap $l_g$ in cm; $\mu_o$ is the permeability of air in H/m; $N$ is the number of turns on the battery module side; $A_e$ is the effective core area in $cm^2$; $L_m$ is the magnetizing inductance in μH; $D_{cp}$ is the diameter of the center post in cm [37].

$$l_g = \mu_o N^2 \frac{10^4 A_e (1 + \frac{l_g}{D_{cp}})^2}{L_m} \qquad 3.30$$

With $N = 3$, $\mu_o = 40^{-7}\pi$, $A_e = 2.02$, $L_m = 1.75$, and $D_{cp} = 1.59$, the length of the gap $l_g$ was calculated to be 0.158 cm (0.062 inches). To gap the center post of the core, a four-fluke, diamond-coated end mill was used. In addition, a water cooling system was used to cool the core continuously during machining. Finally, a mounting clamp was machined out of aluminum to secure the core. An image of the mounting clamp and core are shown in Figure 3.26.

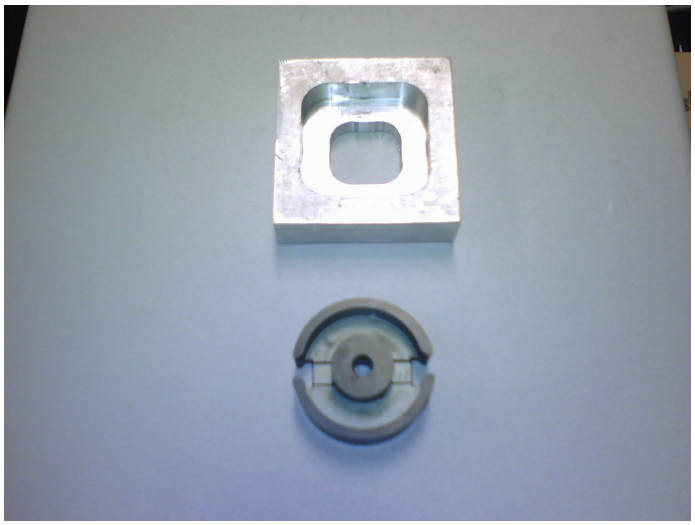


Figure 3.26: Mounting clamp used to machine the transformer core

The transformer core is very brittle, and care must be taken in securing and machining it. To prevent cracking the core, the clamps were only hand tightened, and one thousandth of an inch was removed during each pass. An image of the machining process is shown in Figure 3.27.

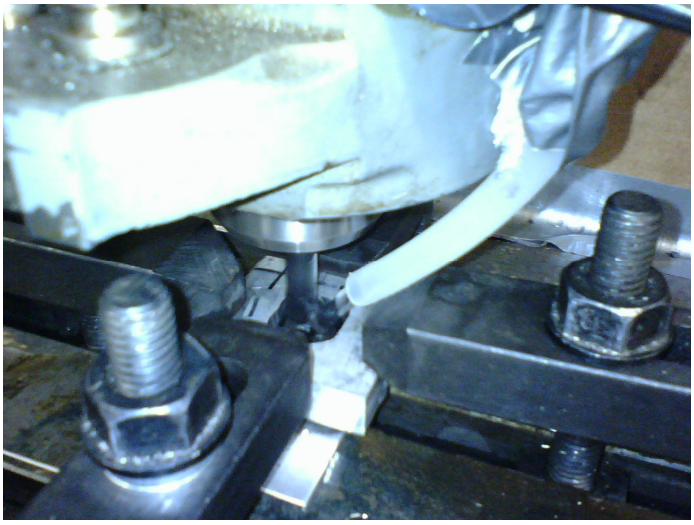


Figure 3.27: Machining the transformer core

Two transformer cores were gapped to different values, with the exact inductance values to be obtained through electrical testing. One core was gapped to 0.040 in, and the second core was gapped to 0.060 in. The accuracy of the mill is +/- 0.005 in.

With the cores gapped, the next step was to wind the bobbin. The litz wire identified during the design phase was not available, so alternate wires were identified: 120/38 single polyester, single nylon litz wire for the module-side winding and 60/44 single polyester, single nylon litz wire for the bus-side winding. The module side of the bobbin was wound with three turns, and the bus side of the bobbin was wound with 51 turns. A small amount of epoxy was used to hold the windings in place.

To solder the ring terminals to the wire, the litz wire ends first were tinned. A molten pool of solder was heated to 800 °F, and then the wire tip was dragged laterally across the solder pool for approximately 10 seconds [38]. This dissolves the polyester coating on each individual strand, and leaves a completely tinned end of the wire. An image of the tinned end of the wire and solder pool is shown in Figure 3.28. The tinned wire is then soldered into a ring terminal. An image of the wound bobbin is shown in Figure 3.29. Finally, an image of the completed transformer wired to the CEC is shown in Figure 3.30.

Figure 3.28: Tinning the litz wire

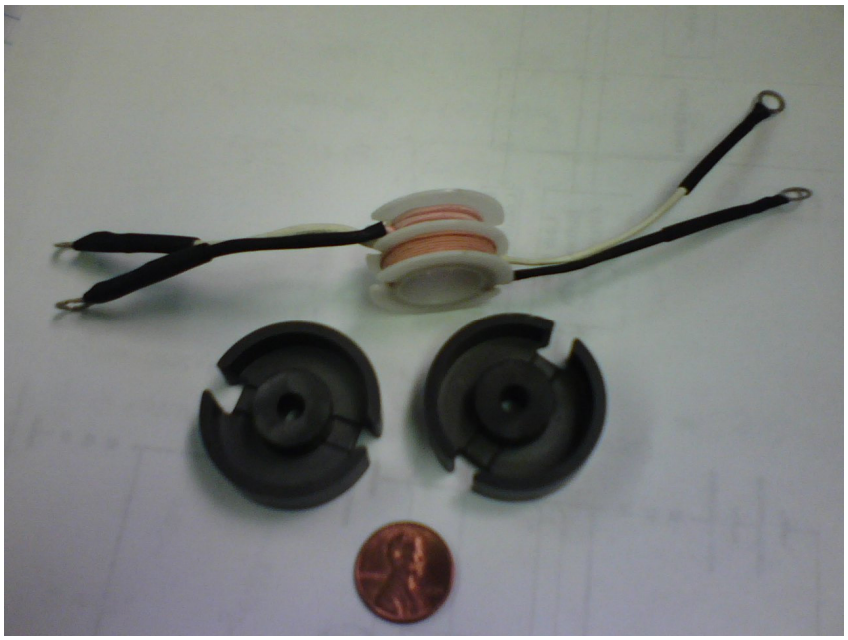


Figure 3.29:  Image of the wound bobbin and machined transformer cores

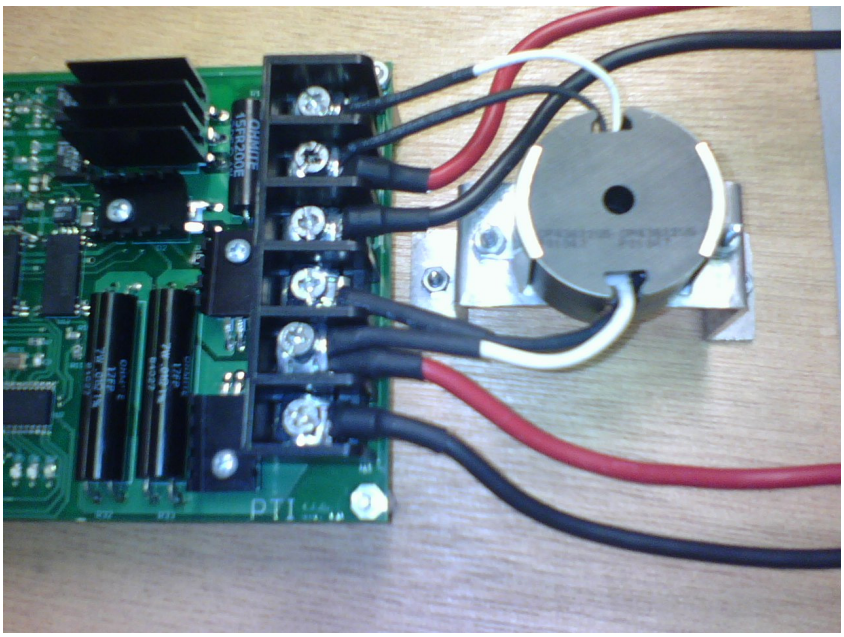



Figure 3.30: Completed transformer core wired to the CEC

# Chapter 4

# Experimental Results

Once the individual sub-circuits of the charge-equalization circuit (CEC) had been tested, the overall circuit was tested for its ability to transfer prescribed currents between a battery module and the high-voltage bus. First, an open-loop test was performed to test the power electronic components operation and calibrate the feedback signals. Then, proportional control was to be tested for its ability to accurately transfer a requested average current.

## 4.1 Open-loop Peak-Current Test

The first test performed was the open-loop, peak-current test. A peak-current cutoff value was sent to the current-mode controller. The gate drive and clock signals of the current-mode controller were monitored, along with the voltage across the shunt-resistor. The MOSFET was tested first, followed by the IGBT.

The transformer that was gapped to 0.04 inches was used, leading to an expected inductance of 2.6 μH. A 10-cell, nickel-metal hydride (NiMH) battery module was used to generate the battery module voltage; the voltage was measured as 13.3 V. An AeroVironment ABC-150 power processing system was used to emulate a high-voltage bus. It was set to a voltage of 100 V with a current limit of 1 A. Pictures of the test setup are shown in Figure 4.1. The leftmost picture shows the overall setup. The desktop

PC controls remotely controls the ABC-150, which is shown in the background. The laptop is emulating the master controller, and it sends out commands on the RS485 bus to the CEC. Finally, the oscilloscope is measuring and recording data. The rightmost picture shows the CEC, high-voltage bus connections, and the NiMH battery module.

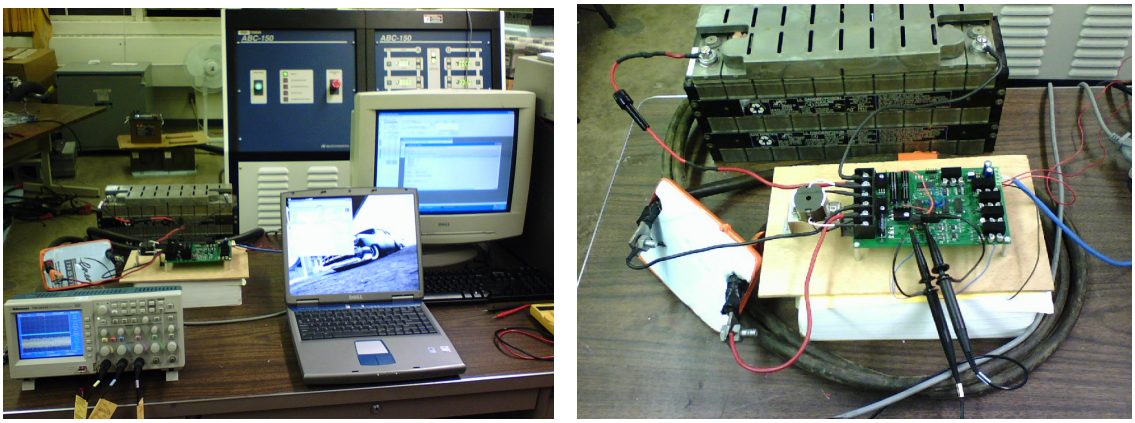


Figure 4.1: Peak-current test setup

### 4.1.1 MOSFET Peak-Current Test

To test the ability of the CEC to transfer a discharging current, the current-mode controller of the MOSFET was sent a peak-current request. The gate drive was monitored on Ch1 (top signal). The voltage across the shunt resistor was monitored on Ch2 (middle signal). Finally, the current mode controller clock signal was monitored on Ch3 (bottom signal). Idle and 20-A peak current snapshots are shown in Figure 4.2.

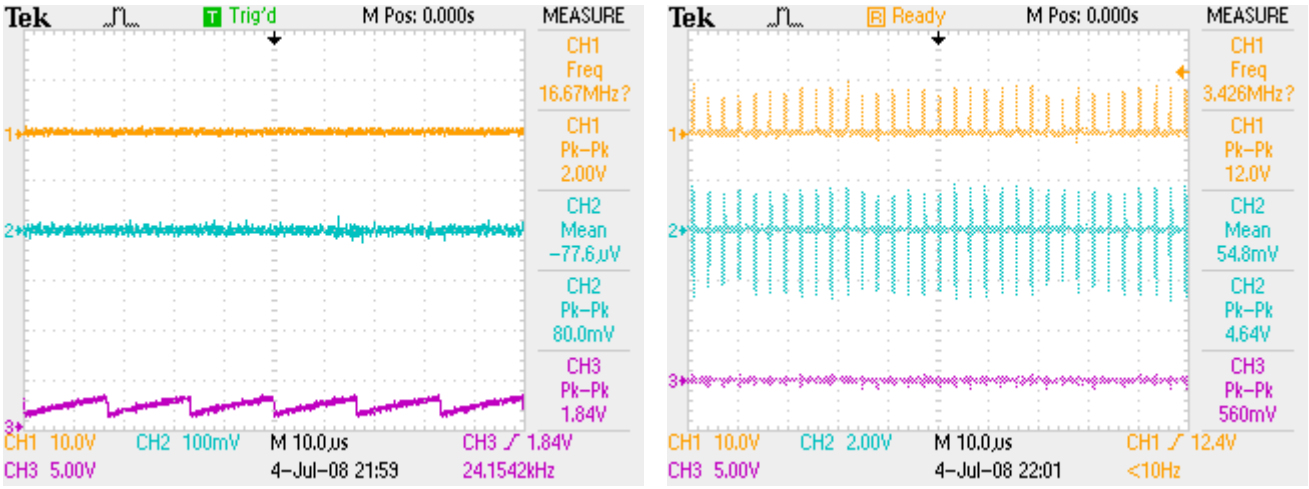



Figure 4.2: MOSFET idle snapshot and 20-A peak-current test snapshot

The current mode controller behaved erratically when the current request was made: the gate drive pulses occurred faster then the switching period, the voltage across the shunt resistor was not a triangular waveform, and the clock signal stopped. Also, when a 0-A peak current request was sent after the 20-A peak current request, the waveforms shown did not change from those of the 20-A request.

The problem appeared to be the result of a design flaw in the layout. More specifically, the ground pin of the shunt resistor is very close to the module ground connector, but the 15-V power and ground pins of the current-mode controller are at the end of long traces. These traces proved to be to long, with the voltage at ground pin of the current-mode controller differing significantly from the true module ground during switching pulses. As a work-around, an external jumper was wired from the ground pin of the current-mode controller to the module ground connector. The behavior of the

current-mode controller improved. The 25-µs window snapshots are shown in Figure 4.3.

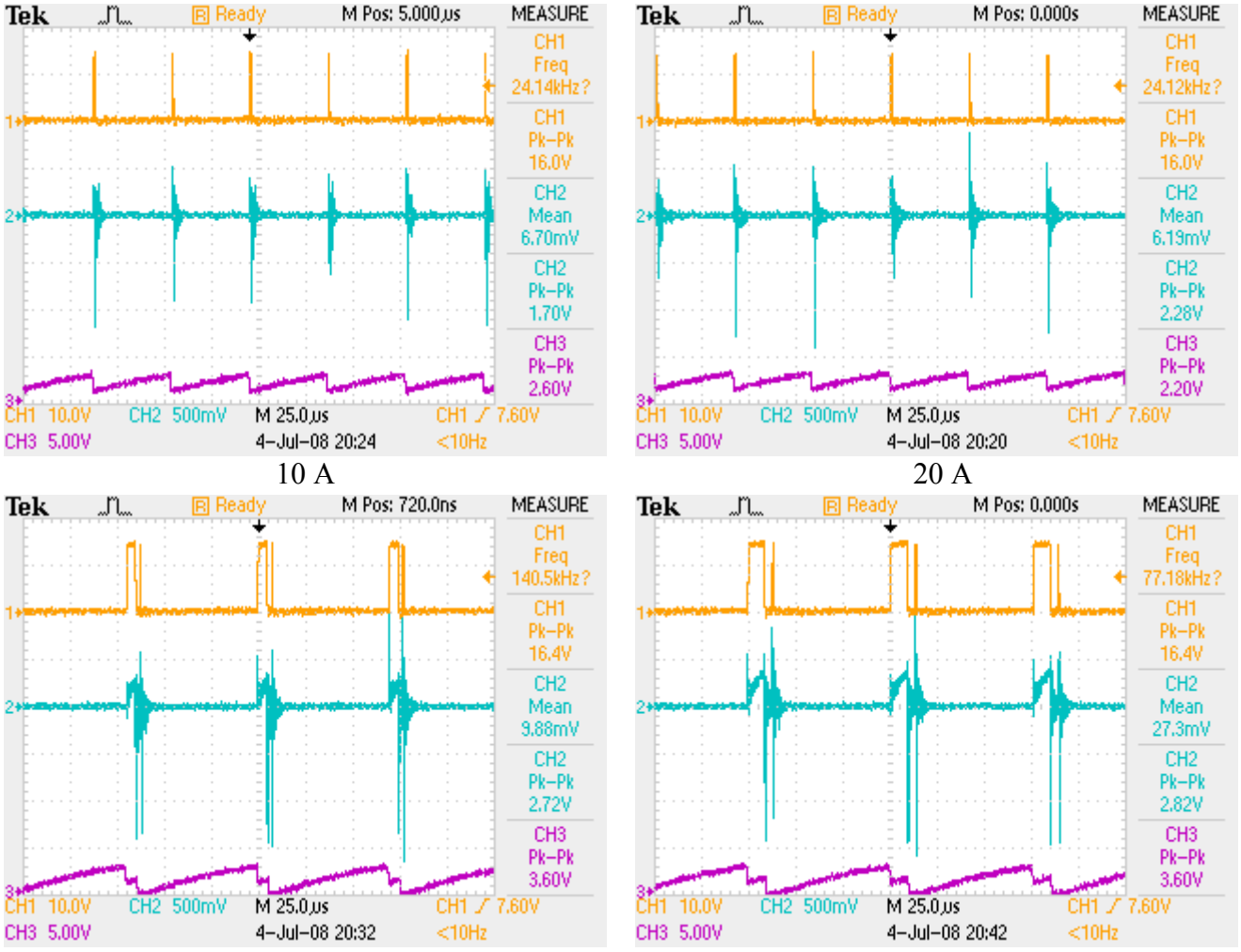



Figure 4.3: MOSFET peak-current test with jumper snapshots at a 25-µs window

Once the current-mode controller was working, the commanded value for the peak current was increased from 10 A to 40 A. The current-mode controller clock signal began to degrade at peak current requests of 30 A and greater. The controller also begins to skip some switching periods while producing two gate pulses instead of one for other switching periods. The erratic switching appeared to be closely related to the

disturbances in the clock signal. The oscillating voltages across the shunt resistor drives the current sense pin of the current-mode controller beyond its rated limits, which results in the degradation of the clock signal. To examine the signals more closely, the 2.5-µs window snapshots are shown in Figure 4.4.

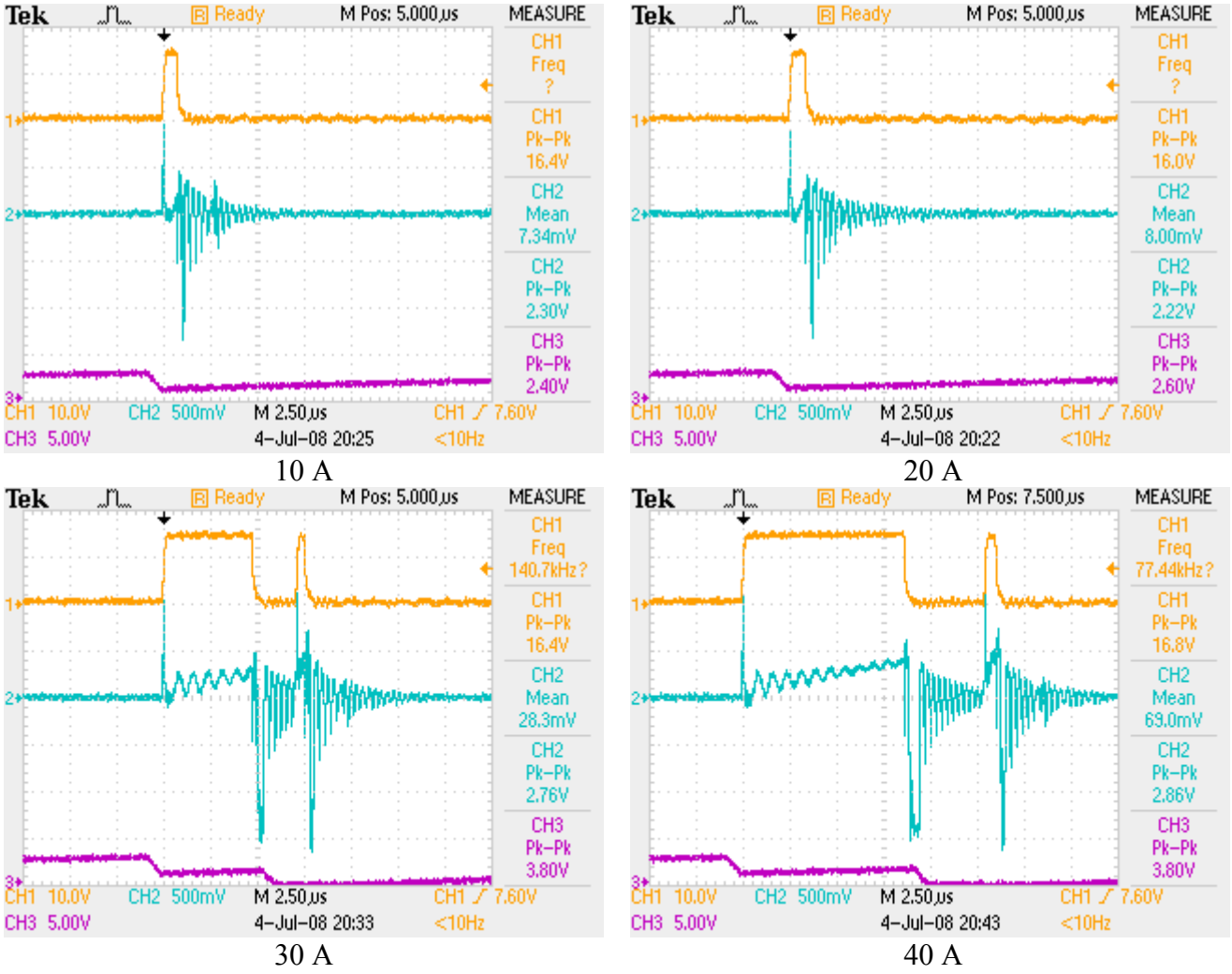
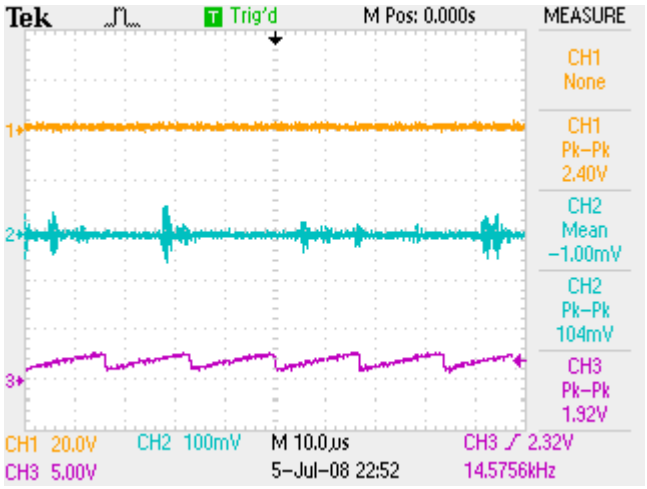



Figure 4.4: MOSFET peak-current test with jumper snapshots at a 2.5 µs window

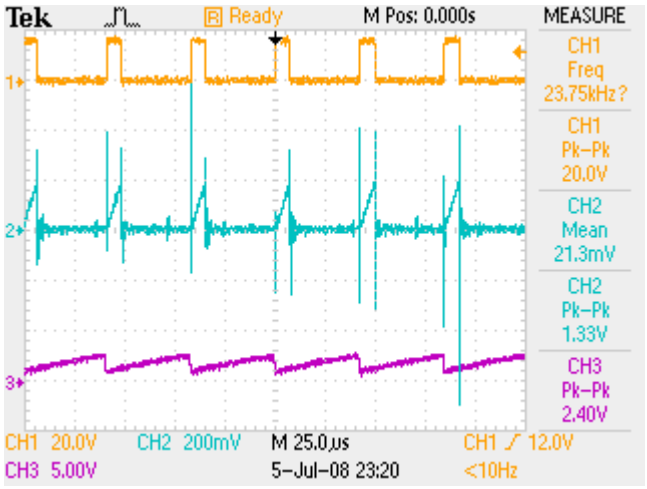
### 4.1.2 IGBT Peak-Current Test

To test the ability of the CEC to transfer charging current, the current-mode controller of the IGBT was sent a peak-current request. The gate drive was monitored on Ch1 (top signal). The voltage across the shunt resistor was monitored on Ch2 (middle signal). Finally, the current mode controller clock signal was monitored on Ch3 (bottom signal).

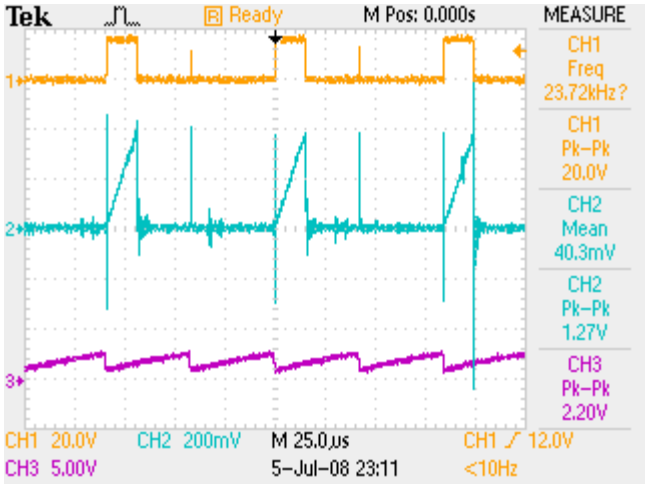
To begin, a 1-A peak-current request test was conducted. Without much time elapsing, the IGBT failed to a short circuit. The IGBT and current-mode controller were replaced, and the test continued. The 25-µs window snapshots are shown in Figure 4.5.



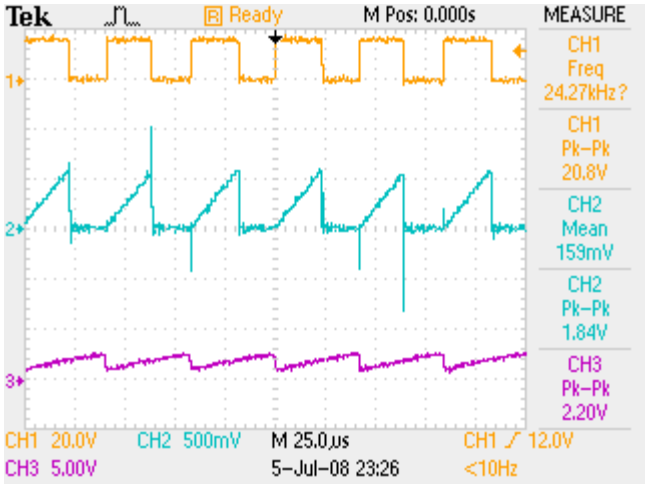

idle




1 A




2 A




3 A

Figure 4.5: IGBT peak-current test snapshots at a 25-µs window

The clock signal did not appear to be affected in the IGBT peak current tests, as it was with the MOSFET peak current tests. However, as before some switching periods were skipped. This can be seen in the 2-A peak test above. An external jumper was wired from the ground lead of the shunt resistor to the ground pin of the current-mode controller. The 25-µs window snapshots with the jumper are shown in Figure 4.6.

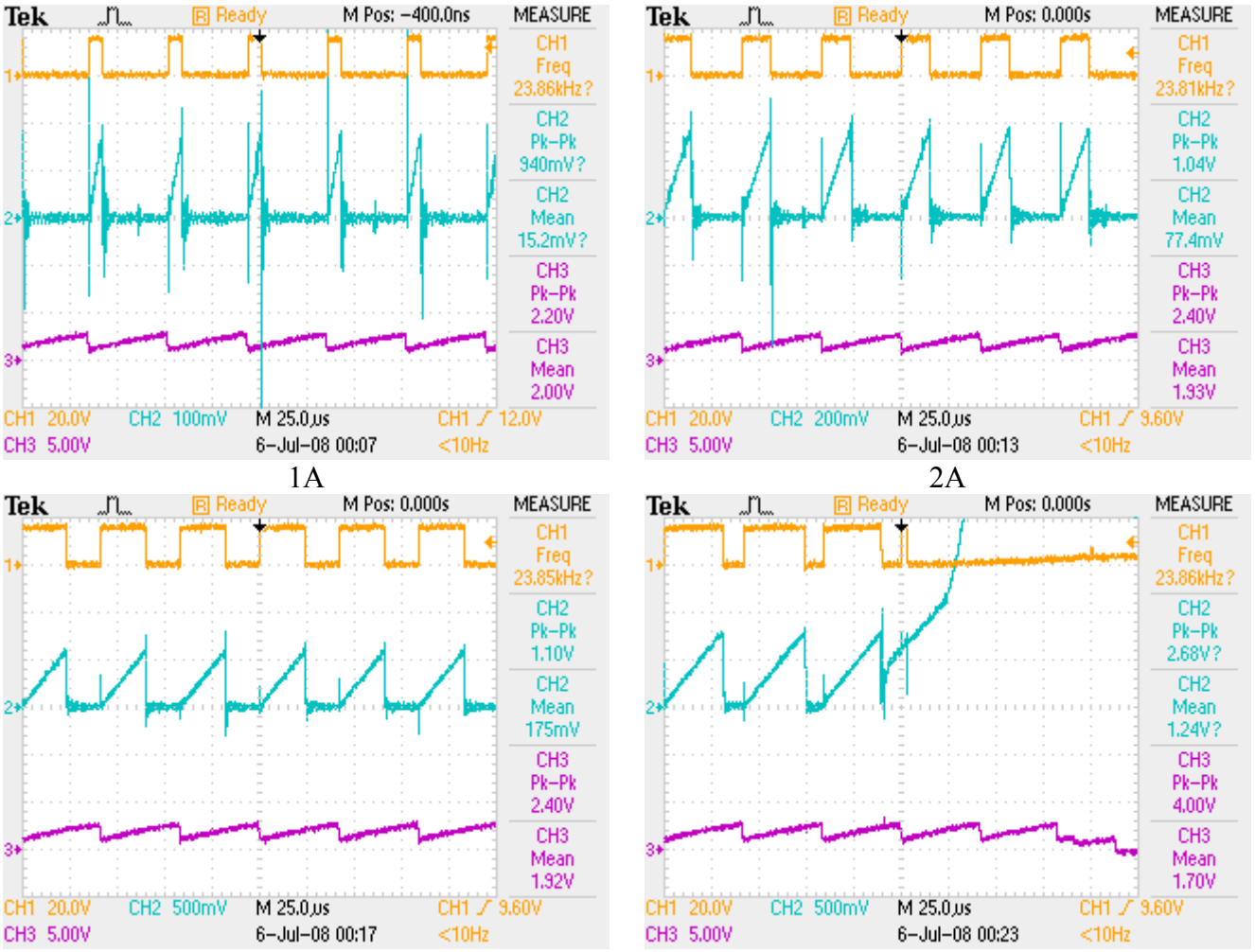



Figure 4.6: IGBT peak-current test with jumper snapshots at a 25-µs window

The 4-A peak-current test resulted in the IGBT failing to short circuit. Some of the on-board dc-to-dc converters also failed. These failures ended the peak-current testing. The 2.5-µs window snapshot of the 1-A peak-test is shown in Figure 4.7 and the 5-µs window snapshots of the 2-A and 3-A peak-tests are shown in Figure 4.8.

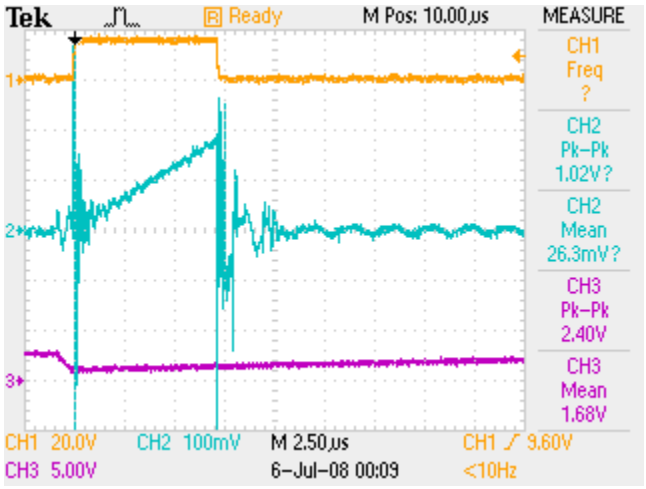
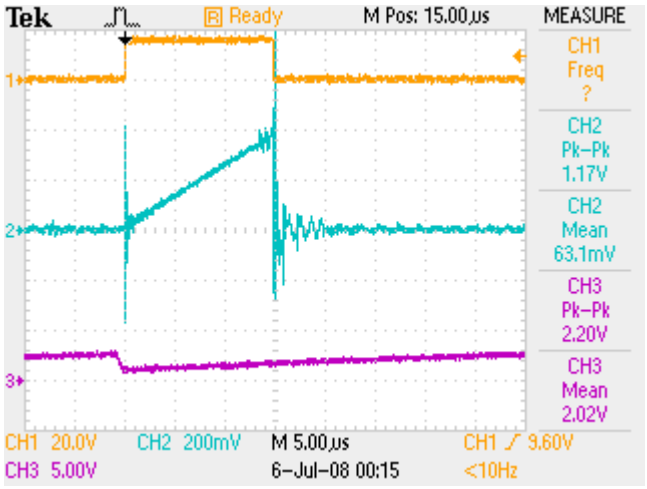
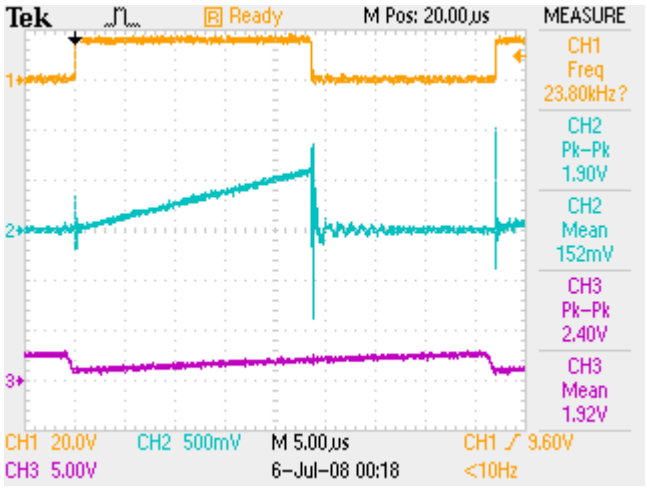



Figure 4.7: IGBT 1-A peak-current test with jumper snapshot at a 2.5-µs window




2 A




3 A

Figure 4.8: IGBT peak-current test with jumper snapshots at a 5-µs window

### 4.2 Proportional Control Test

With the problems that arose in the peak current tests, the proportional control test was not conducted. The waveforms generated in the MOSFET peak current test had too much ringing, coupled with the failures of the IGBT result in a need for redesign before the feedback control can be tested.

### 4.3 Observations and Recommended Changes

The magnetizing inductance was calculated from the slope of a MOSFET triangular waveform, and the error from the theoretical value is promising. Examining the 40-A waveform in Figure 4.4, the initial triangular waveform ramps up from 0 V to 400 mV, which corresponds to a peak current of 40 A. The width of the triangular waveform is approximately 8.5 μs. Therefore, the slope is approximately 4.71 MA/s. The magnetizing inductance is equal to the module voltage divided by the slope, and is approximately 2.8 μH. The value of the transformer magnetizing inductance is approximately 8% greater than expected.

The actual inductance value of the core is too large for the core to be used in the CEC. But its 8% error from the theoretical value does provide a good starting point. The theoretical value of the gap length is based on simplifications that the magnetic field lines travel in a straight line from one side of the core to the next. With the large gap that is required for the CEC core, this approximation leads to errors. To correct for the errors in

the length of gap calculation, an iterative process of removing additional amounts of the center post and retesting should be used.

The grounding issues with the discharging current-mode controller IC were significant. The IC was placed too far from the MOSFET and the current sense resistor, and it behaved erratically when a peak-current request was made. The pulse currents flowing through the trace to the ground pin of the IC created a significant voltage difference between that pin and true module ground. This resulted in spurious voltages on the current sense pin of the IC. A jumper wire placed from the ground pin of the IC to the battery module ground connector improved performance. Closer placement of the discharging current-mode controller IC to the MOSFET and the sense resistor along with the addition of a direct trace from the ground pin of the IC to the ground terminal of the sense resistor are recommended to reduce the voltage difference between the ground pin of IC and module ground.

Another problem with the discharging circuit is large current spikes at the switch turn-on, and the oscillation of the ramping current waveform. The large circuit traces and power electronic components create a substantial parasitic capacitance. With no snubber circuit designed, when the switch transitions to a short circuit, a large current spike is created as the capacitance discharges. The current-mode controller has a built in blanking time and correctly ignores the initial voltage spike that is created across the shunt resistor. However, the energy rings with the parasitic inductance as the current waveform ramps up. A current-damping snubber circuit is recommended to reduce this overshoot and ringing at switch turn-on.

In addition to the large spike and ringing at the switch turn-on, the MOSFET also has considerable ringing when the switch is turned-off. The large peak currents (theoretically up to 85.5 A) stores a considerable amount of energy in the leakage inductance. When the switch transitions to an open circuit, this leakage inductance causes a negative voltage spike and the energy again rings with the parasitic capacitance. The oscillating voltage across the sense resistor drives voltages on the current sense pin beyond the ratings of the device. The clock signal degrades, and the device begins to miss some switching periods while producing two gate pulses instead of one for other switching periods. A voltage-damping snubber circuit is recommended to reduce the overshoot and ringing at switch turn-off.

The charging current-mode controller was also placed with a long trace from its ground pin to the high-voltage bus ground connector. It was able to switch on the IGBT and create ramping currents, but some switching periods were skipped. A jumper placed from the current-mode controller ground pin to the high-voltage bus ground connector solved this problem. A more direct routing of the trace from the current-mode controller ground pin to the bus ground connector is recommended.

It also had voltage spikes on switching transitions, but the effects were much smaller than the MOSFET. The ringing in the ramping waveform at switch turn on was present but negligible. However, to reduce stress on the IGBT and the current-mode controller, a current-damping snubber circuit is recommended to reduce the overshoot and ringing at switch turn-on. As with the MOSFET, to reduce the stress on the IGBT, a voltage-damping snubber circuit is recommended to reduce the overshoot and ringing at switch turn-off.

In addition to the ringing at switch turn-on and turn-off, another problem with the charging circuit is that two IGBT failures occurred. Both resulted in the IGBT becoming a short circuit, and the second was also accompanied by the on-board dc-to-dc converters failing. It is not clear whether the IGBT or dc-to-dc converter failures were related. The IGBT was operating at a bus voltage of 100 V and peak-current levels of 4 A. Both values are well below the device limits. It is noted that the snapshot of the IGBT failure shows it fails during a transition to an open circuit. At this point, the voltage that must be blocked by the IGBT is equal to the high-voltage bus voltage plus the battery module voltage times the winding ratio. In this case, it theoretically was $100 + 13.3 \times 17 = 326$ V. It can be speculated that the additional voltage associated with the ringing of the leakage inductance and the IGBT/trace capacitance caused the device to fail. Implementation of the recommended snubber circuits and additional testing is recommended to eliminate the IGBT failures.

Finally, the heat dissipation of the electronic switches was underestimated. Tight packaging on the printed circuit board layout required use of small heat sinks. It is recommended that the spacing between the electronic switches and freewheeling diodes be increased to allow for larger heat sinks.

# Chapter 5

## Summary and Conclusions

A detailed comparison of Charge-Equalization Circuits (CECs) for automotive battery managements systems has been presented. Three different categories of CECs were discussed. Series CECs tend to be the simplest systems to implement and are suited for space- and cost-sensitive applications. Parallel CECs tend to have the most even redistribution of the recovered energy and are suited for applications that require a significant amount of energy to be redistributed. Finally, hybrid CECs offer a compromise in terms of complexity and performance.

It was determined that an even redistribution of the recovered energy, high power, a modular design, and the ability to equalize both higher- and lower-than-average SOC battery cells were characteristics of importance to the Penn State research vehicles. The Bidirectional, Flyback-Converter system was chosen as the CEC to implement. A complete circuit analysis was performed on the CEC based on the operating conditions of the research vehicles.

Once voltage, current, and power levels of the sub-circuits were calculated, the power electronic components were specified. A current-mode controller was selected to drive the electronic switches, and a proportional control strategy was implemented to set the peak-current cutoff value of the current-mode controller. With this type of control, the peak-current cutoff is adjusted automatically to reduce the error between the measured average value and the desired average value for the respective discharging or

charging current. The measured average values were obtained by filtering the triangular (dis)charging current signals with two-pole, low-pass Butterworth filters.

Once the power electronic components were selected and the control strategy was designed, an RS485 network was chosen to provide communications between the master controller and the individual CECs. A microcontroller was then selected with the necessary peripherals and pin connections. Finally, the on-board dc-to-dc converters used to power the control and communications integrated circuits were selected.

Once the pencil-and-paper design was completed, a printed circuit board layout was created and sent out for fabrication. The board was populated and tested one sub-circuit at a time: dc-to-dc converters, microcontroller and communication network, gate drive signal conditioning circuits, feedback signal conditioning circuits, and power electronic components. Finally, the transformer was gapped, wound, and connected to the board.

After all sub-circuits were populated and tested, the complete circuit was tested for its ability to transfer prescribed discharging and charging currents. First, an open-loop, peak-current test was conducted. Several problems arose during the testing of the both the discharging and charging current sub-circuits. The problems were severe enough that the proportional control strategy was not tested. Direct ground traces from the bus and module ground connectors to the current-mode controllers were recommended. Current and voltage snubber circuits were also recommended to reduce the overshoot and ringing that is present during the switch transitions. Finally, increased spacing between the power electronic components is recommended to allow for larger heatsinks.

Other possible extensions of this work include its integration into the battery management system, or a performance versus efficiency and cost effectiveness evaluation.

## Bibliography


1. "Converting Energy Units 101." Energy Information Administration. United States Department of Energy. 30 Apr 2008 <http://www.eia.doe.gov/basics/conversion_basics.html>.

2. "Weekly Retail Gasoline and Diesel Prices." Energy Information Administration. 28 Apr 2008. United States Department of Energy. 30 Apr 2008 <http://tonto.eia.doe.gov/dnav/pet/pet_pri_gnd_dcus_nus_w.htm>.

3. "Average Retail Price of Electricity to Ultimate Customers: Total by End-Use Sector." Energy Information Administration. 13 Mar. 2008. United States Department of Energy. 30 Apr. 2008 <http://www.eia.doe.gov/cneaf/electricity/epm/table5_3.html>.

4. Emadi, Ali. et al. "Topological Overview of Hybrid Electric and Fuel Cell Vehicular Power System Architectures and Configurations." IEEE Transactions on Vehicular Technology. v54.3 (2005): 763-770.

5 Chan, C.C. "The State of the Art of Electric and Hybrid Vehicles." Proceedings of the IEEE.v90.2 (2002): 247-275.

6. "Automobile Emissions: An Overview." United States Environmental Protection Agency. Apr. 1994. United States Environmental Protection Agency. 30 Apr 2008 < http://www.epa.gov/oms/consumer/05-autos.pdf>.

7. Pokharel, Sajal S., et al. "Emissions Reductions as a Result of Automobile Improvement." Environmental Science and Technology. v37.22 (2003): 5097-5101.

8. Bunkley, Nick. "California Sues Six Automakers over Global Warming." New York Times. 21 Sep. 2006: C2.

9. Chau, K. T., Y. S. Wong, and C. C. Chan. "An Overview of Energy Sources for Electric Vehicles." Energy Conversion and Management. v40.10 (1999): 1021-1039.

10. Chan, C. C., and K. T. Chau. Modern Electric Vehicle Technology. New York: Oxford, 2001.

11. "Prius 08 Specs." Toyota. 2008. Toyota. 1 May 2008 <http://www.toyota.com/prius-hybrid/specs.html>.

12. "Technical Specs – 2008 Model Year." Tesla Motors. 2008. Tesla Motors. 1 May 2008 <http://www.teslamotors.com/performance/tech_specs.php>.

13. Dhameja, Sandeep. Electric Vehicle Battery Systems. Boston: Newnes, 2002.

14. Dickinson, Blake E. and David H. Swan. "EV Battery Pack Life: Pack Degradation and Solutions." SAE Special Publications: Electric and Hybrid Vehicles – Implementation of Technology. SP-1105 (1995): 145-154.

15. Linden, David. Handbook of Batteries. New York: McGraw-Hill, 2002.

16. Bergvik, S. "Prolonged Useful Life and Reduced Maintenance of Lead-Acid Batteries by Means of Individual Cell Voltage Regulation." INTELEC Conference Proceedings. 1984: 63-66.

17. Bjork, D. "Maintenance of Batteries – New Trends in Batteries and Automatic Battery Charging." INTELEC Conference Proceedings. 1986: 355-360.

18. Pascual, Cesar and Philip T. Krein. "Switched Capacitor System for Automatic Series Battery Equalization." IEEE Applied Power Electronics Conference and Exposition. v2 (1997): 848-854.

19. Kobzev G. A. "Switched-Capacitor Systems for Battery Equalization." Modern Techniques and Technology, 2000. MTT 2000. Proceedings of the VI International Scientific and Practical Conference of Students, Post-Graduates and Young Scientists. 28 Feb. to 03 Mar, 2000: 57-59.

20. West, Sean and Philip T. Krein. "Equalization of Valve-Regulated Lead-Acid Batteries: Issues and Life Test Results." INTELEC Conference Proceedings. 2000: 439-446.

21. Kutkut, Nasser H. "A Modular Non Dissipative Current Diverter for EV Battery Charge Equalization." IEEE Applied Power Electronics Conference and Exposition. v2 (1998): 686-690.

22. Hopkins, D. C., C. R. Mosling and S. T. Hung. "The Use of Equalizing Converters for Serial Charging of Long Battery Strings." Applied Power Electronics Conference and Exposition. 10-15 Mar, 1991: 493-498.

23. Schmidt, H. and C. Siedle. "The Charge Equalizer – A New System to Extend Battery Lifetime in Photovoltaic Systems, UPS and Electric Vehicles." INTELEC Conference Proceedings. v2 (1993): 146-151.

24 Kutkut, Nasser H., Deepak M. Divan and Donald W. Novotny. "Charge Equalization for Series Connected Battery Strings." IEEE Industry Applications Society Annual Meeting. v2 (1994): 1008-1015.

25. Kutkut, N.H., D. M. Divan, and D. W. Novotny. “Charge Equalization for Series Connected Battery Strings.” IEEE Transactions on Industry Applications. v31.3 (1995): 562-568.

26. Zhao, Juan, Jiuchun Jiang, and Liyong Niu. “A Novel Charge Equalization Technique for Electric Vehicle Battery System.” The Fifth International Conference on Power Electronics and Drive Systems. v2 (2003): 853 – 857.

27. Hseih, Yao C., Chin S. Moo, and I. S. Tsai. “Balance Charging Circuit for Charge Equalization.” Proceedings of the Power Conversion Conference, 2002. v3. 2-5 April 2002: 1138-1143.

28. Hsieh, Yao C., Chin S. Moo, I S. Tsai, and Jung C. Cheng. “Dynamic Charge Equalization for Series-Connected Batteries.” 2002 International Conference on Industrial Technology. v1. 11-14 Dec 2002: 444-449.

29. Moo, Chin S., Yao C. Hsieh, and I S. Tsai. “Charge Equalization for Series-Connected Batteries.” IEEE Transactions on Aerospace and Electronic Systems. v39.2. Apr 2003: 704-710.

30. Moo, C. S., Y. C. Hsieh, I S. Tsai, and J. C. Cheng. “Dynamic Charge Equalization for Series-Connected Batteries.” IEEE Proceedings – Electric Power Applications. v150.5. 9 Sept 2003: 501-505.

31. Viera, J.C., M. Gonzalez, J. C. Anton, J. C. Campo, F. J. Ferrero, and M. Valledor. “NiMH vs NiCd Batteries under High Charging Rates.” INTELEC Conference Proceedings. Sept 2006: 1-6.

32. “Ferrite Cores Design Manual and Catalog.” Magnetics. 2006. Magnetics. 26 Jun 2008 <http://www.mag-inc.com/ferrites/fc601.asp>.

33. “Ferrite Core Material Selection Guide.” Magnetics. 2000. Magnetics. 26 Jun 2008 <http://www.mag-inc.com/pdf/fc-s1.pdf>.

34. “Round Litz Recommendations.” Litz-Wire. 2007. HM Wire International. 26 June 2008 <http://www.hmwire.com/pdf%20files/Round_Litz_Recommendations_2B72407.pdf>.

35. Balogh, Laszlo. “Design and Application Guide for High Speed MOSFET Gate Drive Circuits.” Texas Instruments. 2001. Texas Instruments. 10 Dec 2007 <http://focus.ti.com/lit/ml/slup169/slup169.pdf>.

36. “Full Spec 2-Layer Designs Only $33 Each!” Advanced Circuits. 2007. Advanced Circuits. 07 July 2008 <http://www.4pcb.com/index.php?load+content&page_id=130>.

37 “Section 5: Inductor and Flyback Transformer Design.” Texas Instruments. 2001. Texas Instruments. 12 Apr 2008 http://focus.ti.com/docs/training/catalog/events/event.jhtml?sku=SEM401014.

38. “Soldering of Litz Wire.” Litz-Wire.com. 2007. HM Wire International. 15 June 2008 <http://www.litz-wire.com/pdf%20files/Soldering_of_Litz_wire_231507.pdf>.

## Appendix A

## Communication Messages

The following abbreviations will be used in the communication messages:

- **Start Byte**: The byte is always 0xAA, and every message begins with two start bytes. These are used to reset the communication receiver and prepare it to accept the incoming message.
- **Length of Msg**: Length of Message. This byte identifies the total number of bytes in the message. It includes both start bytes and the checksum.
- **Source Addr**: Source Address. This byte identifies which node on the network is sending out the message. The address of the master controller is 0x00, and the addresses of the CEC begin at 0x01 and increment as needed.
- **Dest Addr**: Destination Address. This byte identifies which node on the network the message is intended for.
- **Cmd Num**: Command Number. This byte identifies the specific command of the message. Each master command and CEC reply has a unique command number.
- **Error Code**: This byte is used in CEC replies to tell the master controller of any voltage or current errors present.
- **Direction**: This byte indicates whether equalization current is charging or discharging the battery module.
- **Magnitude**: This byte indicates the magnitude of the equalization current.

- **Duration**: This byte indicates the amount of time to conduct the equalization sequence.
- **Check Sum**: These two bytes are the mathematical sum of the all of the bytes in the message, including the two start bytes. It is used to verify that the message was received correctly. The checksum is transmitted with the most significant byte first.

## Master Commands

**Status Request**

| | Start Byte 1 | Start Byte 2 | Length of Msg | Source Addr | Dest Addr | Cmd Num | Check Sum |
|---|---|---|---|---|---|---|---|
| **Type** | Byte | Byte | Byte | Byte | Byte | Byte | Two Bytes |
| **Value** | 0xAA | 0xAA | 0x08 | 0x00 | 0x01 to 0x19 | 0x01 | 0x0000 to 0xFFFF |

Ask CEC for the error code, and the direction, magnitude and duration of present equalization sequence.

**LED Flash Request**

| | Start Byte 1 | Start Byte 2 | Length of Msg | Source Addr | Dest Addr | Cmd Num | Check Sum |
|---|---|---|---|---|---|---|---|
| **Type** | Byte | Byte | Byte | Byte | Byte | Byte | Two Bytes |
| **Value** | 0xAA | 0xAA | 0x08 | 0x00 | 0x01 to 0x19 | 0x02 | 0x0000 to 0xFFFF |

Tell CEC to flash on-board LEDs and send reply. Used to test RS485 communication.

**Initiate Equalization Request**

| | Start Byte 1 | Start Byte 2 | Length of Msg | Source Addr | Dest Addr | Cmd Num | Requested Direction | Requested Magnitude | Requested Duration | Check Sum |
|---|---|---|---|---|---|---|---|---|---|---|
| **Type** | Byte | Byte | Byte | Byte | Byte | Byte | Byte | Byte | Byte | Two Bytes |
| **Value** | 0xAA | 0xAA | 0x0B | 0x00 | 0x01 to 0x19 | 0x03 | 0x0F or 0xFF | 0x00 to 0xFA | 0x00 to 0xF0 | 0x0000 to 0xFFFF |

Tell CEC to perform specific equalization sequence. This causes the CEC to delete any previous equalization sequence data.

**Requested Direction**

0x00 Charge Battery (Positive current into battery module)

0xFF Discharge Battery (Negative current into battery module)

**Requested Magnitude**

Range is 0x00 (zero amps) to 0xFA (10 amps)

Byte = Current Magnitude (amps) / 0.04

**Requested Duration**

Range is 0x00 (zero minutes) to 0xFF (256 minutes)

Byte = Duration (minutes)

**Equalization Completed Request**

| | Start Byte 1 | Start Byte 2 | Length of Msg | Source Addr | Dest Addr | Cmd Num | Check Sum |
|---|---|---|---|---|---|---|---|
| **Type** | Byte | Byte | Byte | Byte | Byte | Byte | Two Bytes |
| **Value** | 0xAA | 0xAA | 0x08 | 0x00 | 0x01 to 0x19 | 0x04 | 0x0000 to 0xFFFF |

Ask CEC for direction, magnitude, and duration of previous equalization sequence.

**Equalization Stop Request**

| | Start Byte 1 | Start Byte 2 | Length of Msg | Source Addr | Dest Addr | Cmd Num | Check Sum |
|---|---|---|---|---|---|---|---|
| **Type** | Byte | Byte | Byte | Byte | Byte | Byte | Two Bytes |
| **Value** | 0xAA | 0xAA | 0x08 | 0x00 | 0x01 to 0x19 | 0x05 | 0x0000 to 0xFFFF |

Tell CEC to stop present equalization sequence.

## CEC Replies

**Status Reply**

| | Start Byte 1 | Start Byte 2 | Length of Msg | Source Addr | Dest Addr | Cmd Num | Error Code | Present Direction | Present Magnitude | Present Duration | Check Sum |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Type** | Byte | Byte | Byte | Byte | Byte | Byte | Byte | Byte | Byte | Byte | Two Bytes |
| **Value** | 0xAA | 0xAA | 0x0C | 0x01 to 0x19 | 0x00 | 0x11 | 0x00, 0x0F, 0xF0, or 0xFF | 0x00 or 0xFF | 0x00 to 0xFA | 0x00 to 0xF0 | 0x0000 to 0xFFFF |

Contains data for the present equalization sequence.

**Error Code**

| | |
|---|---|
| 0x00 | No error |
| 0x0F | Voltage out of range error |
| 0xF0 | Average current out of range error |
| 0xFF | Both voltage and average current out of range |

**Present Direction**

| | |
|---|---|
| 0x00 | Charging battery module (Positive current into battery module) |
| 0xFF | Discharging battery module (Negative current into battery module) |

**Present Magnitude**

Range is 0x00 (zero amps) to 0xFA (10 amps)

Current Magnitude (amps) = 0.04 * Byte

**Present Duration**

Range is 0x00 (zero minutes) to 0xFF (256 minutes)

Equalization Duration (minutes) = Byte

**LED Flash Reply**

| | Start Byte 1 | Start Byte 2 | Length of Msg | Source Addr | Dest Addr | Cmd Num | Check Sum |
|---|---|---|---|---|---|---|---|
| **Type** | Byte | Byte | Byte | Byte | Byte | Byte | Two Bytes |
| **Value** | 0xAA | 0xAA | 0x08 | 0x01 to 0x19 | 0x00 | 0x12 | 0x0000 to 0xFFFF |

**Initiate Equalization Reply**

| | Start Byte 1 | Start Byte 2 | Length of Msg | Source Addr | Dest Addr | Cmd Num | Requested Direction | Requested Magnitude | Requested Duration | Check Sum |
|---|---|---|---|---|---|---|---|---|---|---|
| **Type** | Byte | Byte | Byte | Byte | Byte | Byte | Byte | Byte | Byte | Two Bytes |
| **Value** | 0xAA | 0xAA | 0x0B | 0x01 to 0x18 | 0x00 | 0x13 | 0x00 or 0xFF | 0x00 to 0xFA | 0x00 to 0xF0 | 0x0000 to 0xFFFF |

**Direction Requested**

0x00 Act to charge battery module (Positive current into battery module)

0xFF Act to discharge battery module(Negative current into battery module)

**Magnitude Requested**

Range is 0x00 (zero amps) to 0xFA (10 amps)

Current Magnitude (amps) = 0.04 * Byte

**Duration Requested**

Range is 0x00 (zero minutes) to 0xFF (256 minutes)

Equalization Duration (minutes) = Byte

**Equalization Completed Reply**

| | Start Byte 1 | Start Byte 2 | Length of Msg | Source Addr | Dest Addr | Cmd Num | Error Code | Previous Direction | Previous Magnitude | Previous Duration | Check Sum |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Type** | Byte | Byte | Byte | Byte | Byte | Byte | Byte | Byte | Byte | Byte | Two Bytes |
| **Value** | 0xAA | 0xAA | 0x0C | 0x01 to 0x18 | 0x00 | 0x14 | 0x00, 0x0F, 0xF0, or 0xFF | 0x00 or 0xFF | 0x00 to 0xFA | 0x00 to 0xF0 | 0x0000 to 0xFFFF |

**Error Code**

0x00 No error

0x0F Voltage out of range error

0xF0 Average current out of range error

0xFF Both voltage and average current out of range

**Previous Direction**

0x00 Charging battery module (Positive current into battery module)

0xFF Discharge battery module (Negative current into battery module)

**Previous Magnitude Completed**

Range is 0x00 (zero amps) to 0xFA (10 amps)

Current Magnitude (amps) = 0.04 * Byte

**Previous Duration Completed**

Range is 0x00 (zero minutes) to 0xFF (256 minutes)

Equalization Duration (minutes) = Byte

**Equalization Stop Reply**

| | Start Byte 1 | Start Byte 2 | Length of Msg | Source Addr | Dest Addr | Cmd Num | Check Sum |
|---|---|---|---|---|---|---|---|
| **Type** | Byte | Byte | Byte | Byte | Byte | Byte | Two Bytes |
| **Value** | 0xAA | 0xAA | 0x08 | 0x01 to 0x18 | 0x00 | 0x15 | 0x0000 to 0xFFFF |

# Appendix B

## Microcontroller Code

The microcontroller code was developed with the MPLAB IDE software and the student version of the C18 compiler. It consists of several .c files to organize the various sections of code, and their corresponding header files. The .c files include main.c, cec.c, a2d.c, interrupts.c, spi.c, timer0.c, and usart.c. The main.c file includes all initializations, and an infinite loop to continuously execute function calls. The cec.c file includes the function calls that are called from the main.c file and is specific to the charge equalization circuit (CEC). All other .c files setup and execute the peripherals of the microcontroller as required by the CEC. Each .c and corresponding header file is presented here.

```
////////////////////////////////////////////////////////
//
// Main.c
// CEC v1.0 - Main File
//
////////////////////////////////////////////////////////

#include "p18f2520.h"
#include "cec.h"
#include "interrupts.h"
#include "usart.h"
#include "timer0.h"
#include "a2d.h"
#include "spi.h"

#pragma config WDT = OFF       //Watchdog Timer off
#pragma config OSC = HSPLL   //Fosc = 10MHz*4(PLL) = 40MHz
#pragma config LVP = OFF       //Low Voltage Programming off

#pragma code

//Global variables
unsigned char RUN_MAIN_LOOP; //Timer0 variable indicating 1ms has passed, updated by timer0.c
unsigned char LED_PERIOD; //period to adjust Error and Equalization LEDs, updated by cec.c
unsigned char LED_COUNT; //counter to adjust Error and Equalization LEDs, updated by cec.c
unsigned int TIME_COUNT; //counter to keep track of ms passed, updated by cec.c
unsigned char MINUTES; //timer to keep track of minutes passed, updated by cec.c

void main(void)
{
	//Main loop variables
	unsigned char VOLTAGES_COUNT = 50;
	unsigned char VOLTAGES_PERIOD = 100; //period to calculate measured voltage
	unsigned int CURRENTS_COUNT = 0;
	unsigned int CURRENTS_PERIOD = 100; //period to calculate measured current
	unsigned char CMC_COUNT = 0;
	unsigned char CMC_PERIOD = 100; //period to set peak current request at CMC
	unsigned int HEARTBEAT_COUNT = 0;
	unsigned int HEARTBEAT_PERIOD = 500; //period to flash heartbeat LED (updated by CEC.c)

	//clear global variables
	RUN_MAIN_LOOP = 0;
	LED_COUNT = 0;
	LED_PERIOD = 10;
	TIME_COUNT = 0;
	MINUTES = 0;

	//Initialize peripherals
	Init_CEC();
	Init_Interrupts();
	Init_Usart();
	Init_Timer0();
	Init_A2D();
	Init_SPI();

	for(;;)
	{
		// Run_MAIN_LOOP set true every 1ms
		if (RUN_MAIN_LOOP)
		{
			RUN_MAIN_LOOP = 0;
			LATC ^= 0x10;		//get main loop period on RC4

			//system duration timer
			TIME_COUNT++;
			if (TIME_COUNT >= 60000)

			{
```

```
                    TIME_COUNT = 0;
                    MINUTES++;
                }

                //Check voltages
                VOLTAGES_COUNT++;
                if (VOLTAGES_COUNT >= VOLTAGES_PERIOD)
                {
                    VOLTAGES_COUNT = 0;
                    Get_Voltages();
                }

                //Check Currents
                CURRENTS_COUNT++;
                if (CURRENTS_COUNT >= CURRENTS_PERIOD)
                {
                    CURRENTS_COUNT = 0;
                    Get_Currents();
                }

                //Set Current Mode Controller peak current value
                CMC_COUNT++;
                if (CMC_COUNT >= CMC_PERIOD)
                {

                    CMC_COUNT = 0;
                    Set_Current_Request();
                }

                //Communication with MASTER
                Service_RS485();

                //LED Interface
                HEARTBEAT_COUNT++;
                if (HEARTBEAT_COUNT >= HEARTBEAT_PERIOD)
                {
                    HEARTBEAT_PERIOD = Update_Heartbeat_LED();
                    HEARTBEAT_COUNT = 0;
                }
                LED_COUNT++;
                if (LED_COUNT >= LED_PERIOD)
                {
                    LED_COUNT = 0;
                    LED_PERIOD = 10;  //reset period to adjust LEDs
                    Update_Equalization_LED();
                    Update_Error_LED();
                }
            }
        }
}
```

```
////////////////////////////////////////////////////////
//
// cec.h
//
////////////////////////////////////////////////////////

void Init_CEC(void);
void Get_Voltages(void);
void Get_Currents(void);
void Set_Current_Request(void);
void Service_RS485(void);
void Process_Message(void);
void Generate_TX_Checksum(void);
unsigned int Update_Heartbeat_LED(void);
void Update_Equalization_LED(void);
void Update_Error_LED(void);

////////////////////////////////////////////////////////
//
// cec.c
// CEC v1.0 - Charge Equalization Circuit Files
//
////////////////////////////////////////////////////////

#include "p18f2520.h"
#include "cec.h."
#include "usart.h"
#include "a2d.h"
#include "spi.h"

//cec.c global variables
const unsigned char MY_ADDRESS = 0x05; //Particular address of CEC (ranges 0x01 to 0x19)
//flags
unsigned char EQUALIZATION_IN_PROGRESS; //Equalization in progress flag
unsigned char VOLTAGE_ERROR; //Module or HV voltage measured out of range flag
unsigned char CURRENT_ERROR; //Discharge or Charge current measured out of range flag
unsigned char NEW_EQUALIZATION_SEQUENCE;  //start new equalization sequence flag
unsigned char STOP_EQUALIZATION_SEQUENCE; //premature stop of equalization sequence flag
unsigned char NEED_TO_TRANSMIT;       //packet ready to transmit flag
//equalization sequence variables, communication too
unsigned char PRESENT_ERROR; //error byte for transmission
unsigned char PREVIOUS_ERROR; //error byte for transmission
unsigned char REQUESTED_DIRECTION; //indicates MASTERs command for direction of current
unsigned char PRESENT_DIRECTION; //indicates current direction of current
unsigned char PREVIOUS_DIRECTION; //indicates previous direction of current
unsigned char REQUESTED_MAGNITUDE; //indicates MASTERs command for magnitude of current
unsigned char PRESENT_MAGNITUDE; //indicates average value of present magnitude of current
unsigned char PREVIOUS_MAGNITUDE; //indications average value of previous magnitude of current
unsigned char REQUESTED_DURATION; //indicates MASTERs command for duration of current
unsigned char PRESENT_DURATION; //indicates present duration of equalization in minutes
unsigned char PREVIOUS_DURATION; //indicates previous duration of equalization in minutes
float REQUESTED_MAGNITUDE_FLOAT; //variable to compare requested magnitude to measured magnitude
unsigned int PRESENT_MAGNITUDE_SUM; //variable to calculated average DC magnitude over duration
//measured values
float MODULE_VOLTAGE_MEASURED; //moving average, measured battery module voltage
float HV_BUS_VOLTAGE_MEASURED; //moving average, measured high-voltage bus voltage
float CHARGE_CURRENT_MEASURED; // moving average, measured average charge current
float DISCHARGE_CURRENT_MEASURED; // moving average, measured average discharge current

//main.c vairables
extern unsigned char LED_PERIOD; //Time between adjustments to Error and Equalization LEDs
extern unsigned char LED_COUNT; //Equalization and Error LEDs adjustment counter
extern unsigned int TIME_COUNT; //Duration of charge ms counter
extern unsigned char MINUTES; //Duration of equalization minutes counter (max 256)

//usart.c variables
extern unsigned char PROCESS_NEW_MESSAGE; //message in RX_DATA ready to be processed
```

```
extern unsigned char TX_BUFFER[15]; //Buffer to store bytes to transmit to Usart
extern unsigned char RX_DATA[15]; //Message stored here from buffer to be processed
extern unsigned int MODULE_VOLTAGE_SUM; //module voltage samples sum
extern unsigned int HV_BUS_VOLTAGE_SUM; //high voltage bus voltage samples sum
extern unsigned int DISCHARGE_CURRENT_SUM; //battery module discharge current samples sum
extern unsigned int CHARGE_CURRENT_SUM; //battery module charge current samples sum

//spi.c variables
extern unsigned char D2A_SELECT; //determines which switch to turn on (0 = IGBT), (1=MOSFET)
extern unsigned char D2A_FIRST_BYTE; //first byte to send to d2a
extern unsigned char D2A_SECOND_BYTE; //second byte to send to d2a


//Initialize ports A,B,C to digital outs
void Init_CEC(void)
{
	//Clear ports, set to output
	LATA = 0x00;
	TRISA = 0x00;
	LATB = 0x00;
	TRISB = 0x00;
	LATC = 0x00;
	TRISC = 0x00;

	//clear flags
	EQUALIZATION_IN_PROGRESS = 0;
	VOLTAGE_ERROR = 0;
	CURRENT_ERROR = 0;
	NEW_EQUALIZATION_SEQUENCE = 0;
	STOP_EQUALIZATION_SEQUENCE = 0;
	NEED_TO_TRANSMIT = 0;
	//clear equalization sequence variables
	PRESENT_ERROR = 0;
	PREVIOUS_ERROR = 0;
	REQUESTED_DIRECTION = 0;
	PRESENT_DIRECTION = 0;
	PREVIOUS_DIRECTION = 0;
	REQUESTED_MAGNITUDE = 0;
	REQUESTED_MAGNITUDE_FLOAT = 0;
	PRESENT_MAGNITUDE = 0;
	PRESENT_MAGNITUDE_SUM = 0;
	PREVIOUS_MAGNITUDE = 0;
	REQUESTED_DURATION = 0;
	PRESENT_MAGNITUDE = 0;
	PREVIOUS_DURATION = 0;
	//clear d2a value, set enable bit
	D2A_SELECT = 0;
	D2A_FIRST_BYTE = 0x10;
	D2A_SECOND_BYTE = 0x00;
	//clear measured values
	MODULE_VOLTAGE_MEASURED = 0;
	HV_BUS_VOLTAGE_MEASURED = 0;
	CHARGE_CURRENT_MEASURED = 0;
	DISCHARGE_CURRENT_MEASURED = 0;
}

//Get Voltages
void Get_Voltages(void)
{
//	IDEAL: MODULE_VOLTAGE = MODULE_VOLTAGE_SUM*3.125*5/50/1023;	//moving average voltage
	MODULE_VOLTAGE_MEASURED = (float)MODULE_VOLTAGE_SUM*0.000324; //calibrated with actual voltages measured
//	IDEAL: HV_BUS_VOLTAGE = HV_BUS_VOLTAGE_SUM*80*5/50/1023;		//moving average voltage
	HV_BUS_VOLTAGE_MEASURED = (float)HV_BUS_VOLTAGE_SUM*0.008372; //calibrated with actual voltages measured
	//check voltages within range
	if ((MODULE_VOLTAGE_MEASURED < 7) | (MODULE_VOLTAGE_MEASURED > 17) |
						(HV_BUS_VOLTAGE_MEASURED < 90) |  (HV_BUS_VOLTAGE_MEASURED > 410))
	{
		PRESENT_ERROR |= 0x0F;
```

```
		}
		else
		{
			PRESENT_ERROR &= ~0x0F;
		}
}

//Get Currents
void Get_Currents(void)
{
//		IDEAL: CHARGE_CURRENT = CHARGE_CURRENT_SUM*5/35.62/40/1023/0.01	//moving average current
		CHARGE_CURRENT_MEASURED = (float)CHARGE_CURRENT_SUM*0.00034304;
//		IDEAL: DISCHARGE_CURRENT = DISCHARGE_CURRENT_SUM*5/31.41/40/1023/0.01; //moving average current
		DISCHARGE_CURRENT_MEASURED = (float)DISCHARGE_CURRENT_SUM*0.00038902;
		//check currents not saturated (can't check if failure mode is measured zero amps)
		if ((CHARGE_CURRENT_MEASURED > 15) | (DISCHARGE_CURRENT_MEASURED > 15))
		{
			PRESENT_ERROR |= 0xF0;
		}
		else
		{
			PRESENT_ERROR &= ~0xF0;
		}
}

//Set Current Mode Controller
void Set_Current_Request(void)
{
		static unsigned char D2A_VALUE = 0;		//command sent to D2A for peak current cutoff
		static unsigned char LAST_MINUTE = 0; //last time average current data byte was updated
		static float MEASURED_CURRENT_ERROR = 0; //difference between requested and actual currents
		// check for stop command, OR voltage error, OR current error
		if ((STOP_EQUALIZATION_SEQUENCE) | (PRESENT_ERROR))
		{
			STOP_EQUALIZATION_SEQUENCE = 0;//clear the flag
			//check if in equalization sequence, if so store previous data
			if (EQUALIZATION_IN_PROGRESS)
			{
				EQUALIZATION_IN_PROGRESS = 0;	//clear the flag
				//store remaining data bytes
				PREVIOUS_ERROR = PRESENT_ERROR;		//save error code
				PREVIOUS_DIRECTION = PRESENT_DIRECTION; //save direction
				PREVIOUS_MAGNITUDE = PRESENT_MAGNITUDE;		//save average DC magnitude over past
duration
				PREVIOUS_DURATION = PRESENT_DURATION;  //save duration
			}
			//set request current peak cutoffs to zero
			D2A_SELECT = 0;	//set charging gate drive (S2) to 0% duty cycle
			D2A_FIRST_BYTE = 0x10;
			D2A_SECOND_BYTE = 0x00;
			Update_D2A();
			D2A_SELECT = 1;	//set discharging gate drive (S1) to 0% duty cycle
			Update_D2A();
		}
		//if no errors or stop request, check if new equalization sequence command
		else if (NEW_EQUALIZATION_SEQUENCE)
		{
			NEW_EQUALIZATION_SEQUENCE = 0; //clear the flag
			PRESENT_ERROR = 0; //voltage and currents in range
			PRESENT_DIRECTION = REQUESTED_DIRECTION; //direction to equalize (0x00 = PWM IGBT) or (0xFF = PWM
MOSFET)
			PRESENT_MAGNITUDE_SUM = 0; //reset magnitude sum to start averaging
			D2A_VALUE = 0; //clear command sent to D2A for peak current request
			MINUTES = 0;		//clear duration minutes counter   (updated automatically in main.c)
			LAST_MINUTE = 0; //last time present magnitude was updated
			TIME_COUNT = 0;	//clear duration milliseconds counter (updated automatically in main.c)
			REQUESTED_MAGNITUDE_FLOAT = (float)REQUESTED_MAGNITUDE*0.04; //used to calculate magnitude error
			EQUALIZATION_IN_PROGRESS = 1;	//set the flag
```

```
}
//check if equalization sequence in progress. if so, adjust peak current value from magnitude error
else if (EQUALIZATION_IN_PROGRESS)
{
	//update duration of equalization variable
	PRESENT_DURATION = MINUTES;
	//check if sequence has timed out
	if (PRESENT_DURATION >= REQUESTED_DURATION)
	{
		STOP_EQUALIZATION_SEQUENCE = 1; //set stop flag
	}
	//else update peak current request
	else
	{
		//check direction (1: PWM MOSFET (discharge), 0: PWM IGBT (charge))
		//update Present magnitude data byte, measured error
		if (PRESENT_DIRECTION)
		{
			//update average magnitude data
			if (MINUTES>LAST_MINUTE)
			{
				LAST_MINUTE = MINUTES;
				PRESENT_MAGNITUDE_SUM += (unsigned
					char)(DISCHARGE_CURRENT_MEASURED/0.04);
				PRESENT_MAGNITUDE = PRESENT_MAGNITUDE_SUM/LAST_MINUTE;
					//update magnitude average
			}
			//calculated average current error
			MEASURED_CURRENT_ERROR = REQUESTED_MAGNITUDE_FLOAT -
				DISCHARGE_CURRENT_MEASURED;
			//ensure IGBT is off
			D2A_SELECT = 0;
			D2A_FIRST_BYTE = 0x10;
			D2A_SECOND_BYTE = 0x00;
			Update_D2A();
			D2A_SELECT = 1;    //PWM switch 1 (MOSFET)
			//adjust peak current cutoff based on magnitude of error
			if (MEASURED_CURRENT_ERROR > 5)
			{
				D2A_VALUE = D2A_VALUE + 0x014D;   //increase by 10 Apk
			}
			else if (MEASURED_CURRENT_ERROR > 2)
			{
				D2A_VALUE = D2A_VALUE + 0x00A6;   //increase by 5 Apk
			}
			else if (MEASURED_CURRENT_ERROR > 0.2)
			{
				D2A_VALUE = D2A_VALUE  + 0x0021;   //increase by 1 Apk
			}
			else if (MEASURED_CURRENT_ERROR > -0.2)
			{
				//no adjustment to Apk
			}
			else if (MEASURED_CURRENT_ERROR > -2)
			{
				D2A_VALUE = D2A_VALUE - 0x0021;    //decrease by 1 Apk
			}
			else if (MEASURED_CURRENT_ERROR > -5)
			{
				D2A_VALUE = D2A_VALUE - 0x00A7;    //decrease by 5 Apk
			}
			else
			{
				D2A_VALUE = D2A_VALUE - 0x014D;    //decrease by 10 Apk
			}
		}
		else
		{
```

```
                                        //update average magnitude data
                                        if (MINUTES>LAST_MINUTE)
                                        {
                                                LAST_MINUTE = MINUTES;
                                                PRESENT_MAGNITUDE_SUM += (unsigned
                                                                        char)(CHARGE_CURRENT_MEASURED/0.04);
                                                PRESENT_MAGNITUDE = PRESENT_MAGNITUDE_SUM/LAST_MINUTE;

                                        }
                                        //calculated average current error
                                        MEASURED_CURRENT_ERROR = REQUESTED_MAGNITUDE_FLOAT -
                                                                                CHARGE_CURRENT_MEASURED;
                                        //ensure MOSFET is off
                                        D2A_SELECT = 1;
                                        D2A_FIRST_BYTE = 0x10;
                                        D2A_SECOND_BYTE = 0x00;
                                        Update_D2A();
                                        D2A_SELECT = 0;    //PWM switch 2 (IGBT)
                                        //adjust peak current cutoff based on magnitude of error
                                        if (MEASURED_CURRENT_ERROR > 5)
                                        {
                                                D2A_VALUE = D2A_VALUE + 0x0295;    //increase by 1 Apk
                                        }
                                        else if (MEASURED_CURRENT_ERROR > 2)
                                        {
                                                D2A_VALUE = D2A_VALUE + 0x00A5;   //increase by 0.25 Apk
                                        }
                                        else if (MEASURED_CURRENT_ERROR > 0.2)
                                        {
                                                D2A_VALUE = D2A_VALUE  + 0x0042;   //increase by 0.1 Apk
                                        }
                                        else if (MEASURED_CURRENT_ERROR > -0.2)
                                        {
                                                //no adjustment to Apk
                                        }
                                        else if (MEASURED_CURRENT_ERROR > -2)
                                        {
                                                D2A_VALUE = D2A_VALUE - 0x0042;    //decrease by 0.1 Apk
                                        }
                                        else if (MEASURED_CURRENT_ERROR > -5)
                                        {
                                                D2A_VALUE = D2A_VALUE - 0x00A5;    //decrease by 0.25 Apk
                                        }
                                        else
                                        {
                                        D2A_VALUE = D2A_VALUE - 0x0295;    //decrease by 1 Apk
                                        }
                                }
                                //check max value of D2A command
                                if (D2A_VALUE > 0x0FFF)
                                {
                                        D2A_VALUE = 0x0FFF;
                                }
                                D2A_VALUE += 0x1000; // enable bit
                                D2A_FIRST_BYTE = (unsigned char)((D2A_VALUE&0xFF00)>>8);
                                D2A_SECOND_BYTE = (unsigned char)(D2A_VALUE&0x00FF);
                                Update_D2A();
                        }
                }
}

//Communication with MASTER
void Service_RS485(void)
{
        // Time to Transmit?
        if (NEED_TO_TRANSMIT)
        {
                NEED_TO_TRANSMIT = 0;
```

```
                Usart_TX();
        }

        // Check if new message came in
        Validate_RX();

        // Act on any new messages that may have arrived
        Process_Message();
}

//Process new message from RS485 and Develop responses
void Process_Message(void)
{
        unsigned char ADDRESS; //packet addressed to
        unsigned char COMMAND; //packet command
        if (PROCESS_NEW_MESSAGE) //check if message received
        {
                PROCESS_NEW_MESSAGE = 0; //clear the flag
                ADDRESS = RX_DATA[4]; //grab address recipient
                COMMAND = RX_DATA[5]; //grab command
                if (ADDRESS == MY_ADDRESS) //check if addressed to this module
                {
                        TX_BUFFER[0] = 0xAA; //First start byte of transmission
                        TX_BUFFER[1] = 0xAA; //Second start byte of transmission
                        TX_BUFFER[3] = MY_ADDRESS; //Source Address of transmission
                        TX_BUFFER[4] = 0x00; //Destination Address of transmission (MASTER address)
                        switch (COMMAND)
                        {
                                case 0x01: //Status Request

                                        TX_BUFFER[2] = 0x0C; //Length byte
                                        TX_BUFFER[5] = 0x11; //Command
                                        TX_BUFFER[6] = PRESENT_ERROR; //Error Code
                                        TX_BUFFER[7] = PRESENT_DIRECTION; //Direction
                                        TX_BUFFER[8] = PRESENT_MAGNITUDE; //Current magnitude
                                        TX_BUFFER[9] = PRESENT_DURATION; //Duration completed
                                break;
                                case 0x02: //Flash LED Request
                                        LATB |= 0x06;           //Turn ON ERROR and EQUALIZATION LEDs
                                        LED_PERIOD = 200;//Set period to be on
                                        LED_COUNT = 0;     //Reset counter
                                        TX_BUFFER[2] = 0x08;            //Length byte
                                        TX_BUFFER[5] = 0x12;            //Command
                                break;
                                case 0x03: //Initiate Equalization Request
                                        REQUESTED_DIRECTION = RX_DATA[6]; //get direction requested
                                        REQUESTED_MAGNITUDE = RX_DATA[7]; //get current magnitude requested
                                        REQUESTED_DURATION = RX_DATA[8]; //get duration requested
                                        TX_BUFFER[2] = 0x0C; //Length byte
                                        TX_BUFFER[5] = 0x13; //Command
                                        TX_BUFFER[6] = PRESENT_ERROR; //Error code
                                        TX_BUFFER[7] = REQUESTED_DIRECTION; //echo direction requested
                                        TX_BUFFER[8] = REQUESTED_MAGNITUDE; //echo current magnitude requested
                                        TX_BUFFER[9] = REQUESTED_DURATION; //echo duration requested
                                        NEW_EQUALIZATION_SEQUENCE = 1; //set flag to start equalization sequence
                                break;
                                case 0x04: //Equalization Completed Request
                                        TX_BUFFER[2] = 0x0C; //Length Byte
                                        TX_BUFFER[5] = 0x14; //Command
                                        TX_BUFFER[6] = PRESENT_ERROR; //Error code
                                        TX_BUFFER[7] = PREVIOUS_DIRECTION; //previous cycle's direction
                                        TX_BUFFER[8] = PREVIOUS_MAGNITUDE; //previous cycle's current magnitude
                                        TX_BUFFER[9] = PREVIOUS_DURATION; //previous cycle's duration completed
                                break;
                                case 0x05: //Equalization Stop Request
                                        TX_BUFFER[2] = 0x08;            //Length Byte
                                        TX_BUFFER[5] = 0x15;            //Command
                                        STOP_EQUALIZATION_SEQUENCE = 1;//set stop flag
```

```
                                        Set_Current_Request(); //don't wait to stop, stop now
                                }
                                Generate_TX_Checksum();
                                NEED_TO_TRANSMIT = 1;
                        }
                }
}

//Generates checksum for transmission packet
void Generate_TX_Checksum(void)
{
        unsigned char i;
        unsigned char LENGTH = TX_BUFFER[2];//length of packet
        unsigned int TX_CHECKSUM =0;            //checksum to be calculated
        for (i=0;i<=LENGTH-3;i++)
        {
                TX_CHECKSUM += TX_BUFFER[i];      //sum up packet bytes
        }
        TX_BUFFER[LENGTH-2] = (unsigned char)(TX_CHECKSUM>>8);          //store MSByte
        TX_BUFFER[LENGTH-1] = (unsigned char)TX_CHECKSUM;   //store LSByte
}

//Heartbeat LED
unsigned int Update_Heartbeat_LED(void)
{
        static unsigned int THEARTBEAT = 3000;  //Period
        static unsigned char TDUTY = 100;           //Duty Cycle Period
        static unsigned char HEARTBEAT_STATUS = 1;      //Status Flag

        if (HEARTBEAT_STATUS)
        {
                LATB &= ~0x01;
                HEARTBEAT_STATUS = 0;
                return (THEARTBEAT - TDUTY);
        }
        else
        {
                LATB |= 0x01;
                HEARTBEAT_STATUS = 1;
                return TDUTY;
        }
}

//Equalization LED
void Update_Equalization_LED(void)
{
        if (EQUALIZATION_IN_PROGRESS)
        {
                LATB |= 0x02;
        }
        else
        {
                LATB &= ~0x02;
        }
}

//Error LED
void Update_Error_LED (void)
{
        if ((VOLTAGE_ERROR) | (CURRENT_ERROR))
        {
                LATB |= 0x04;
        }
        else
        {
                LATB &= ~0x04;
        }
}
```

```
//////////////////////////////////////////////////////////
//
// a2d.h
//
//////////////////////////////////////////////////////////

void Init_A2D(void);
void A2D_ISR(void);

//////////////////////////////////////////////////////////
//
// a2d.c
// CEC v1.0 - A2D to sample 4 different channels
//                  each channel sampled every 1mS
//////////////////////////////////////////////////////////

#include "p18f2520.h"
#include "a2d.h"

// Ts = 0.001 s for each signal

// 100 Hz low pass filter -> 5 periods of data
#pragma udata MODULE_VOLTAGE_SAMPLES
unsigned int MODULE_VOLTAGE_SAMPLES[50];   //module voltage vector (max 64, need bigger sum variable if more)
unsigned int MODULE_VOLTAGE_SUM;  //sum of module voltage samples

// 100 Hz low pass filter -> 5 periods of data
#pragma udata HV_BUS_VOLTAGE_SAMPLES
unsigned int HV_BUS_VOLTAGE_SAMPLES[50];    //high voltage bus voltage vector
unsigned int HV_BUS_VOLTAGE_SUM;   //sum of high voltage bus voltage samples

// 25 Hz low pass filter -> 1 period of data
#pragma udata DISCHARGE_CURRENT_SAMPLES
unsigned int DISCHARGE_CURRENT_SAMPLES[40];              //battery module discharge current vector
unsigned int DISCHARGE_CURRENT_SUM;             //sum of discharge current samples

// 25 Hz low pass filter -> 1 period of data
#pragma udata CHARGE_CURRENT_SAMPLES
unsigned int CHARGE_CURRENT_SAMPLES[40];    //battery module charge current vector
unsigned int CHARGE_CURRENT_SUM;  //sum of charge current samples

//Initialize A2D pins
void Init_A2D(void)
{
        unsigned char i;        //loop counter

        TRISA |= 0x0F;          //set pins RA0 to RA3 as input
        ADRESH = 0x00;          //clear A2D result high byte
        ADRESL = 0x00;          //clear A2D result low byte
        ADCON1 = 0x0C;          //set pins RA0 to RA3 as analog input
        ADCON0 = 0x00;          //Disable a2d, set channel to RA0
        ADCON2 = 0x84;          //Left-justified, 8 TAD acquisition time, Fosc/2
        ADCON0 |= 0x01;         //enable a2d (ADON)
        PIR1 &= ~0x40;          //clear a2d flag (ADIF)
        PIE1 |=     0x40;       //a2d interrupt enable (ADIE)
        CCPR2H = 0x09;          //Set CCP Special Event trigger to initiate an A2D
        CCPR2L = 0xC4;          //sample every T = 1/10e6*0x09C4 = 0.25mS
        CCP2CON |=              0x0B;       //enable CCP2 special event trigger -> triggers A2D conversion
        T1CON = 0x01;           //enable timer1, Tosc*4 = 0.4uS

        //clear variables
        MODULE_VOLTAGE_SUM = 0;
        HV_BUS_VOLTAGE_SUM = 0;
        CHARGE_CURRENT_SUM = 0;
        DISCHARGE_CURRENT_SUM = 0;
        for (i=0; i<50; i++)
        {
```

```
                MODULE_VOLTAGE_SAMPLES[i] = 0;
                HV_BUS_VOLTAGE_SAMPLES[i] = 0;
        }
        for (i=0; i<40; i++)
        {
                CHARGE_CURRENT_SAMPLES[i] = 0;
                DISCHARGE_CURRENT_SAMPLES[i] = 0;`
        }
}

//A2D interrupt routine
void A2D_ISR(void)
{
        unsigned int A2D_SAMPLE;      //variable grabs A2D result
        static unsigned char A2D_CHANNEL = 0; //channel pointer
        static unsigned char MODULE_VOLTAGE_COUNT = 0;          //module voltage sample counter
        static unsigned char HV_BUS_VOLTAGE_COUNT = 0;          //hv bus voltage sample counter
        static unsigned char CHARGE_CURRENT_COUNT = 0;          //charging current sample counter
        static unsigned char DISCHARGE_CURRENT_COUNT = 0;       //discharging current sample counter

        //grab sample
        A2D_SAMPLE = (int)(ADRESH)<<8;
        A2D_SAMPLE += (int)ADRESL;
        //Module voltage sample
        if (A2D_CHANNEL == 0)
        {
                MODULE_VOLTAGE_SUM = MODULE_VOLTAGE_SUM + A2D_SAMPLE -
                                MODULE_VOLTAGE_SAMPLES[MODULE_VOLTAGE_COUNT]; //moving average sum
                MODULE_VOLTAGE_SAMPLES[MODULE_VOLTAGE_COUNT++] = A2D_SAMPLE; //store sample in vector
                if (MODULE_VOLTAGE_COUNT >= 50)
                {
                        MODULE_VOLTAGE_COUNT = 0;
                }
                        //update moving average
                A2D_CHANNEL = 1; //change channel pointer
                ADCON0 |= 0x04; //setup A2D for channel 1
        }
        //Discharge current sample
        else if (A2D_CHANNEL == 1)
        {
                DISCHARGE_CURRENT_SUM = DISCHARGE_CURRENT_SUM + A2D_SAMPLE -
                                DISCHARGE_CURRENT_SAMPLES[CHARGE_CURRENT_COUNT]; //moving average sum
                DISCHARGE_CURRENT_SAMPLES[DISCHARGE_CURRENT_COUNT++] = A2D_SAMPLE; //store sample in
                                                                                                vector
                if (DISCHARGE_CURRENT_COUNT >= 40)
                {
                        DISCHARGE_CURRENT_COUNT = 0;
                }
                A2D_CHANNEL = 2; //change channel pointer
                ADCON0 &= ~0x04; //clear bit
                ADCON0 |= 0x08;     //setup A2D for channel 2

        }
        //Charge current sample
        else if (A2D_CHANNEL == 2)
        {
                CHARGE_CURRENT_SUM = CHARGE_CURRENT_SUM + A2D_SAMPLE -
                        CHARGE_CURRENT_SAMPLES[CHARGE_CURRENT_COUNT]; //moving average sum
                CHARGE_CURRENT_SAMPLES[CHARGE_CURRENT_COUNT++] = A2D_SAMPLE; //store sample in vector

                if (CHARGE_CURRENT_COUNT >=40)
                {
                        CHARGE_CURRENT_COUNT = 0; //reset counter
                }
                A2D_CHANNEL = 3; //change channel pointer
                ADCON0 |= 0x04; //setup A2D for channel 3
        }
        //High voltage bus voltage sample
```

```
        else
        {
                HV_BUS_VOLTAGE_SUM = HV_BUS_VOLTAGE_SUM + A2D_SAMPLE -
                                HV_BUS_VOLTAGE_SAMPLES[HV_BUS_VOLTAGE_COUNT]; //moving average sum vector
                A2D_CHANNEL = 0;  //change channel pointer
                ADCON0 &= ~0x0C; //setup for channel 0
                if (HV_BUS_VOLTAGE_COUNT >=50)
                {
                        HV_BUS_VOLTAGE_COUNT = 0;           //reset counter
                }
        }
}
```

```
////////////////////////////////////////////////////////
//
// interrupt.h
//
////////////////////////////////////////////////////////

void Init_Interrupts(void);
void InterruptVector (void);
void InterruptHandler (void);

////////////////////////////////////////////////////////
//
// interrupts.c
// CEC v1.0 - Interrupt handler
//
////////////////////////////////////////////////////////

#include "p18f2520.h"
#include "interrupts.h"
#include "timer0.h"
#include "usart.h"
#include "a2d.h"

// Initialize Interrupts
void Init_Interrupts(void)
{
	// Clear all flags, all interrupts low priority
	// and disable all interrupts
	INTCON = INTCON2 = INTCON3 = 0;
	PIR1 = PIR2 = 0;
	PIE1 = PIE2 = 0;
	IPR1 = IPR2 = 0;
	RCON = 0;

	// Configure & Enable Interrupts
	INTCON   |= 0x80; // Enable unmasked interrupts
	INTCON   |= 0x40; // Enable unmasked peripheral interrupts
}


// Interrupt vector
#pragma code InterruptVector = 0x08
void InterruptVector (void)
{
 _asm
   goto InterruptHandler //jump to interrupt routine
  _endasm
}

// Interrupt handler
#pragma code
#pragma interrupt InterruptHandler
void InterruptHandler (void)
{
	//Timer 0 Overflow Vector
	if (INTCON & 0x04)   //check for TMR0 overflow
	{
		INTCON &= ~0x04;  //clear interrupt flag
		Timer0_ISR();          //Routine to set Timer_Tick and register values
	}
	//Usart TX Interrupt Vector
	else if (PIR1 & 0x10) //check for Transmit ready flag
	{
		PIR1 &= ~0x10;        //clear interrupt flag
		TX_ISR(); //Routine to load transmit register with DATA to be sent
	}
	//Usart RX Interrupt Vector
	else if (PIR1 & 0x20) //check for Receive Interrupt Flag
```

```
	{
		PIR1 &= ~0x20;		//clear interrupt flag
		//check for overrun condition
		if (RCSTA & 0x02)
		{
			RCSTA &= ~0x10;	//disable continuous receive (CREN)
			RCSTA |= 0x01;		//enable continuous receive (CREN)
		}
		RX_ISR();//Routine to evaluate received byte
	}
	//A2D Interrupt Vector
	else if (PIR1 & 0x40)
	{
		PIR1 &= ~0x40;		//clear interrupt flag
		LATB ^= 0x10;
		A2D_ISR();		//Routine to grab a2d result and switch channel
	}
}
```

```
////////////////////////////////////////////////////////
//
// spi.h
//
////////////////////////////////////////////////////////

void Init_SPI(void);
void Update_D2A(void);


////////////////////////////////////////////////////////
//
// spi.c
// CEC v1.0 - Serial Peripheral Interface file.  Sets the
//                  peak current request of the CMC
////////////////////////////////////////////////////////

#include "p18f2520.h"
#include "spi.h"
#include "cec.h"

unsigned char D2A_SELECT;
unsigned char D2A_FIRST_BYTE;
unsigned char D2A_SECOND_BYTE;

//initialize SPI
void Init_SPI(void)
{
        LATC &= ~0x06;      //disable communication to both D2As (set high at D2As)
        SSPCON1 = 0x12;     //Master, F = Fosc/4 = 10MHz
        SSPSTAT = 0x40;     //Transmit occurs on rising edge
        SSPCON1 |= 0x20;    //SPI enable

        //set both peak current values to zero
        D2A_FIRST_BYTE = 0x00;
        D2A_SECOND_BYTE = 0x00;
        D2A_SELECT = 0;
        Update_D2A();
        D2A_SELECT = 0;
        Update_D2A();
}

//command peak current current cutoff
void Update_D2A(void)
{
        //turn on communication
        if (D2A_SELECT == 0)
        {
                LATC &= ~0x04;      //turn off D2A communication for MOSFET
                LATC |= 0x02;       //turn on D2A communication for IGBT
        }
        else
        {
                LATC |= 0x04;       //turn on D2A communication for MOSFET
                LATC &= ~0x02;      //turn off D2A communication for IGBT
        }
        SSPBUF = ~D2A_FIRST_BYTE;           //send first byte
        while ((volatile unsigned char)(SSPSTAT & 0x01) == 0)           //wait for transmission to end
        {
        }
        SSPBUF = ~D2A_SECOND_BYTE;          //send second byte
        while ((volatile unsigned char)(SSPSTAT & 0x01) == 0)           //wait for transmission to end
        {
        }
        //turn off communication to set results
        LATC &= ~0x04;      //turn off D2A communication for MOSFET
        LATC &= ~0x02;      //turn off D2A communication for IGBT
}
```

```
//////////////////////////////////////////////////////////
//
// timer0.h
//
//////////////////////////////////////////////////////////

void Init_Timer0(void);
void Timer0_ISR(void);

//////////////////////////////////////////////////////////
//
// timer0.c
// CEC v1.0 - Timer0 overflow sets RUN_MAIN_LOOP
//                  flag high every 1ms
//////////////////////////////////////////////////////////

#include "p18f2520.h"

extern unsigned char RUN_MAIN_LOOP;   //global main loop delay variable

void Init_Timer0(void)
{
        //Clear register of interest to Timer0
        T0CON = 0;  //clear timer0 control register

        //Setup Timer0 for overflow interrupt
        //Timer0 Step Time = Prescaler/Fosc = 1/10e6 = 1e-7
        INTCON |= 0x20;      //enable timer0 overflow interrupt
        T0CON |= 0x08;       //no prescaler
        T0CON |= 0x80;       //enable timer0

}

void Timer0_ISR(void)
{
        //Counter Steps = 1ms/1e-7 = 0x2710
        //Load timer counter = 0xFFFF - 0x2710 = 0xD8EF
        //Adjusted slightly so main loop runs at 1ms
        RUN_MAIN_LOOP = 1;           //set timer_tick flag for main loop
        TMR0H = 0xDB;
        TMR0L = 0x00;
}
```

```
////////////////////////////////////////////////////////
//
// usart.h
//
////////////////////////////////////////////////////////

void Init_Usart(void);
void Usart_TX(void);
void TX_ISR(void);
void RX_ISR(void);
void Validate_RX(void);

////////////////////////////////////////////////////////
//
// usart.c
// CEC v1.0 - Serial Communications File
//
////////////////////////////////////////////////////////

#include "p18f2520.h"
#include "usart.h"

//flags
unsigned char NEW_PACKET_STARTED; //first start byte detected flag
unsigned char GET_PACKET_LENGTH; //both start bytes detected, move to save length byte flag
unsigned char SAVE_PACKET_BYTES; //length byte saved, move to save remaining bytes flag
unsigned char NEW_MESSAGE_READY; //packet saved into buffer, ready to move to into data variable      flag
unsigned char PROCESS_NEW_MESSAGE; //message in RX_DATA ready to be processed

//data variables
unsigned char RX_BUFFER[15]; //Buffer to save received bytes from Usart
unsigned char TX_BUFFER[15]; //Buffer to store bytes to transmit to Usart
unsigned char RX_DATA[15];    //Message stored here from buffer to be processed
unsigned char RX_CNT; //Incoming packet counter
unsigned char TX_CNT; //Outgoing packet counter
unsigned char TX_PTR; //Pointer to indicate which byte to transmit

//Communications settings
//8Bit, Asynchrounous Mode, 57600bps
//1 start, 8 data, 1 stop bits
//transmits/receives Lsb first
void Init_Usart (void)
{
        //initialize global variables
        NEW_PACKET_STARTED = 0;
        NEW_MESSAGE_READY = 0;
        PROCESS_NEW_MESSAGE = 0;
        SAVE_PACKET_BYTES = 0;
        GET_PACKET_LENGTH = 0;

        //set up USART for communication
        TRISC |= 0xC0;      //Set RC7(RX) and RC6(TX) as input
                                        //Usart sets TX to output as needed
        TRISC &= ~0x01;     //RC0 (RS485 direction control) as output
        LATC &= ~0x01;      //Set RC0 low for receive
        TXSTA = 0x04;       // Hign Baud (BRGH)
        SPBRG = 42;         //57600bps = (Fosc/(16*(Baud+1))
        RCSTA = 0x80;       //Serial Port Enable (SPEN)
        PIE1 |= 0x20;       //Enable Usart Receiver Interrupts
        RCSTA |= 0x10;      //Enable Receiver (CREN)
}

//Usart1 TX Routine
//Load up variables and enable transmit interrupt
void Usart_TX(void)
{
        TX_CNT = TX_BUFFER[2];     // Packet length
        TX_PTR = 0;         //pointer starts at beginning of vector
```

```
	TXSTA |= 0x20;		//transmit enable (TXEN)
	LATC |= 0x01;		//RS485 direction to transmit (RC0)
	TXREG = TX_BUFFER[0]; // Write byte 0 into transmit register
	PIE1 |= 0x10;		//Enable TX Interrupt (TXIE)
}

//Service Usart 1 TX ISR
//Interrupt jumps to here
void TX_ISR(void)
{
	if (--TX_CNT == 0)   //run until last byte just shifted out of TXREG
	{
		while(((volatile unsigned char)(TXSTA)&0x02) == 0) {} //wait for last byte to get sent
		PIE1 &= ~0x10;		//Disable TX Interrupt
		TXSTA &= ~0x20; //Disable transmit function
		LATC &= ~0x01;		//RS485 direction to receive (RC0)
	}
	else //still have bytes to transmit
	{
		TX_PTR++;		//shift pointer to next byte in vector
		TXREG = TX_BUFFER[TX_PTR];		//load transmit register with next byte
	}
}

void RX_ISR(void)
{
	unsigned char INCOMING_BYTE;

	// ISR grabs data from incoming stream and puts in appropriate place
	INCOMING_BYTE = RCREG;

	//start bytes have been identified, check length of packet
	if (GET_PACKET_LENGTH)
	{
		GET_PACKET_LENGTH = 0;   //reset length byte flag
		if (INCOMING_BYTE <= 0x0F) //make sure packet will fit in buffer
		{
			RX_BUFFER[RX_CNT++] = INCOMING_BYTE; //store length of packet
			SAVE_PACKET_BYTES = 1;    //save remaining bytes flag
		}
	}
	//length byte stored, get remaing packet bytes
	else if (SAVE_PACKET_BYTES)
	{
		RX_BUFFER[RX_CNT++] = INCOMING_BYTE; // save byte
		if (RX_CNT >= RX_BUFFER[2])		//check if counter is equal to packet lenght byte
		{
			NEW_MESSAGE_READY = 1; //message saved, set message ready flag
			SAVE_PACKET_BYTES = 0;    //reset save bytes flag
		}
	}
	//no first start byte detected, check for start byte
	else if ((INCOMING_BYTE == 0xAA) && (!NEW_PACKET_STARTED)) // check if first start byte
	{
		NEW_PACKET_STARTED = 1;		//new packet detected, look for second start byte
		RX_BUFFER[0] = INCOMING_BYTE;		//store first start byte
	}
	//first byte detected, check for second byte
	else if ((INCOMING_BYTE == 0xAA) && (NEW_PACKET_STARTED))
	{
		NEW_PACKET_STARTED = 0;		//second start byte confirmed, clear flag
		GET_PACKET_LENGTH = 1;   //get packet length flag set
		RX_BUFFER[1] = INCOMING_BYTE;		//store second start byte
		RX_CNT = 2;		//byte saved pointer
	}
	//packet not valid, reset checks
	else
	{
```

```
                GET_PACKET_LENGTH = 0;
                SAVE_PACKET_BYTES = 0;
                NEW_PACKET_STARTED = 0;
        }
}

//Validate new message
void Validate_RX(void)
{
        unsigned char PACKET_LENGTH;            //packet length variable
        unsigned char i;        //loop variable
        unsigned int RX_CALCULATED_CHECKSUM, RX_PACKET_CHECKSUM;            //validate packet checksum
        if (NEW_MESSAGE_READY)
        {
                NEW_MESSAGE_READY = 0; //Clear the new message flag
                PACKET_LENGTH = RX_BUFFER[2];   //Get the length of the packet
                RX_CALCULATED_CHECKSUM = 0;     //clear calculated checksum
                RX_PACKET_CHECKSUM = 0;         //clear packet checksum
                //Copy RX_BUFFER into RX_DATA for data safety
                for(i=0;i<PACKET_LENGTH;i++)
                {
                        RX_DATA[i] = RX_BUFFER[i];
                }
                for(i=0;i<PACKET_LENGTH-2;i++)
                {
                        RX_CALCULATED_CHECKSUM += RX_DATA[i]; //sum up bytes in message (except check sum bytes)
                }
                //process new message if transmission was valid
                RX_PACKET_CHECKSUM = (int)RX_DATA[PACKET_LENGTH-2]<<8; //Arrange MSByte into upper 8bits
                RX_PACKET_CHECKSUM += (int)RX_DATA[PACKET_LENGTH-1];    //Add in LSByte
                if (RX_CALCULATED_CHECKSUM == RX_PACKET_CHECKSUM)       //compare calculated checksum to packet
                                                                                                        checksum
                {
                        PROCESS_NEW_MESSAGE = 1;            //Set process new message flag
                }
        }
}
```

# Appendix C

## Schematic, Printed Circuit Board, and Parts List

The circuit schematic and board layout were created with the EAGLE software by CadSoft Computer Inc., version 4.16r2, standard edition. A nonprofit license is currently available for $125. The schematic is separated onto different pages to allow the various components to be legible. Traces spanning multiple pages are labeled. The first page is the input and output headers schematic, and it is shown in Figure C.1. It illustrates the input and output connections of the board. The second page is the power schematic, and it is shown in Figure C.2. It shows the on-board dc-to-dc converters. The third page is the chassis ground schematic, and it is shown in Figure C.3. It illustrates the microcontroller and RS485 transceiver circuitry. The fourth page is the charge and discharge filters schematic, and it is shown in Figure C.4. It shows the active filter circuitry for both signals. The fifth page is the battery module voltage RC filter and MOSFET gate drive schematic, and it is shown in Figure C.5. It illustrates the battery module voltage RC filter and the MOSFET gate drive circuitry. The fifth and final page is the high-voltage bus voltage filter and IGBT gate drive schematic, and it is shown in Figure C.6. It shows the high-voltage bus voltage RC filter and the IGBT gate drive circuitry.

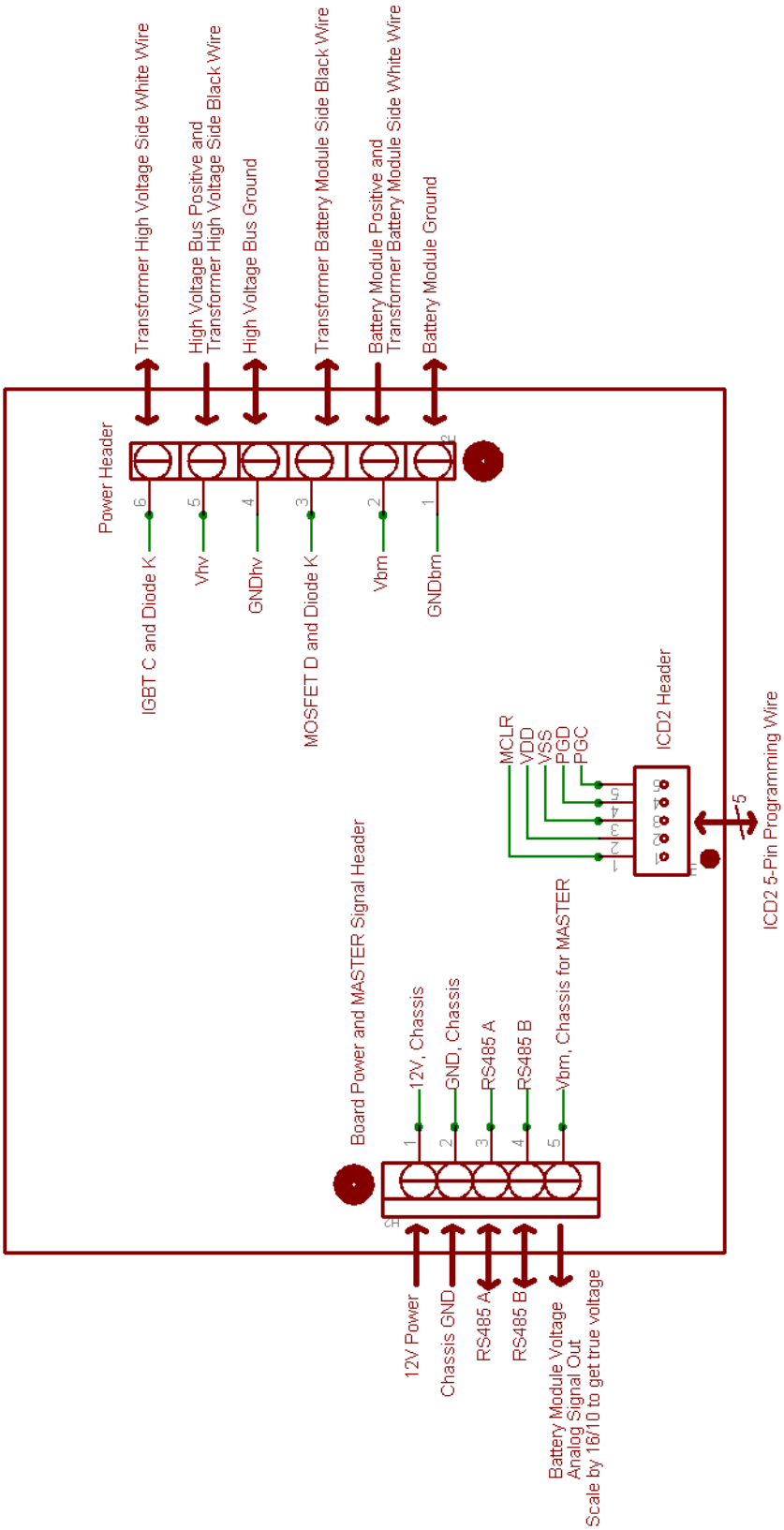



Figure C.1: Input and output headers schematic

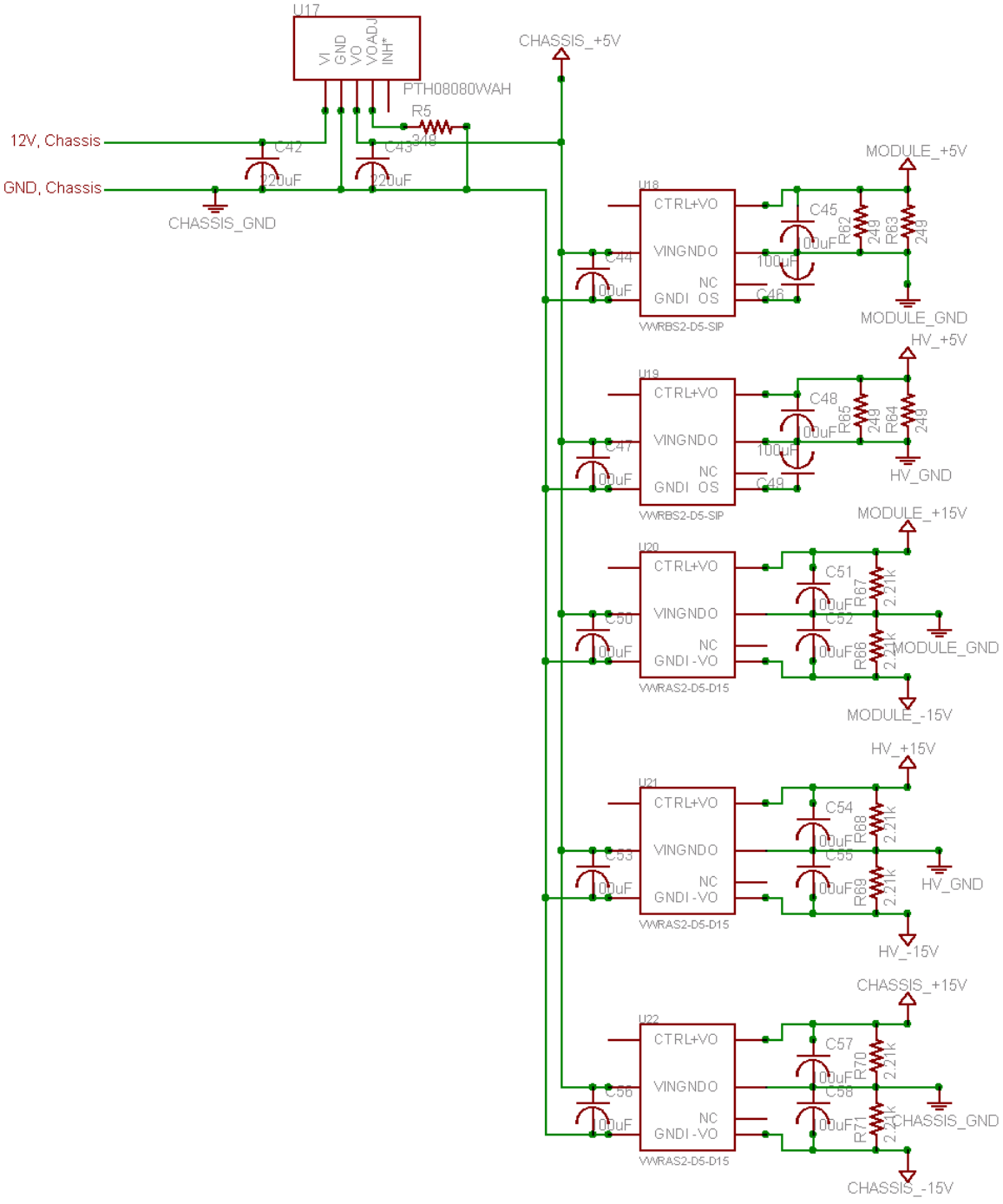



Figure C.2: Power schematic

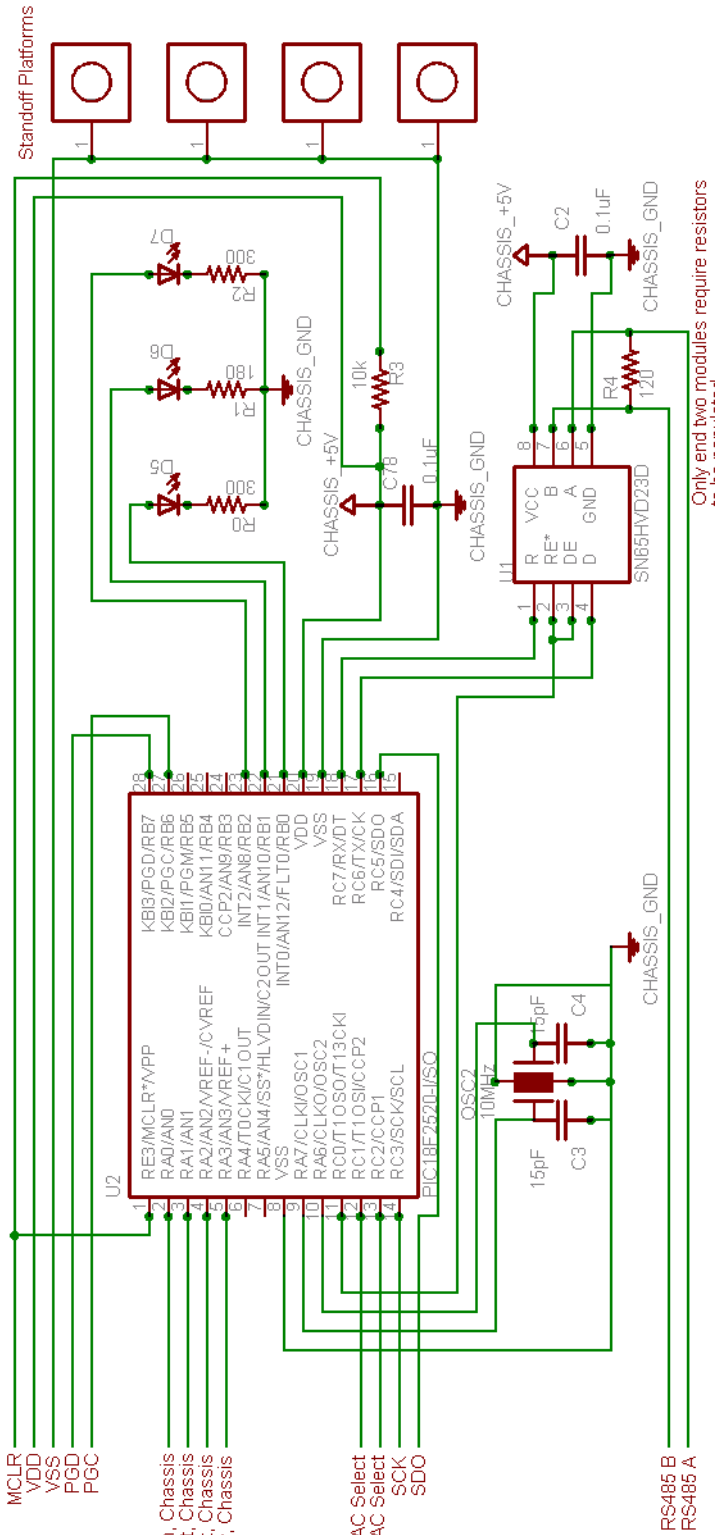



Figure C.3: Chassis ground schematic

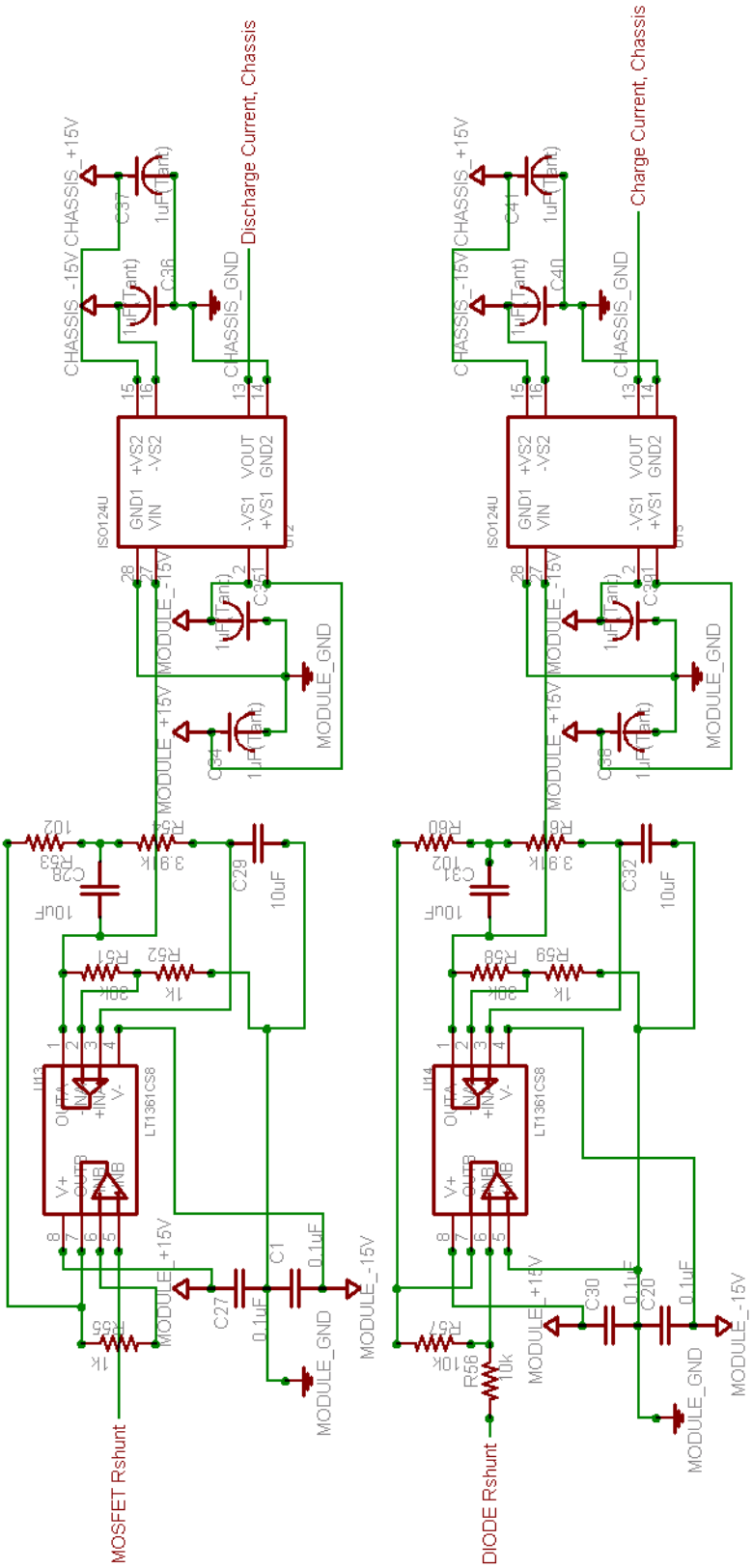



Figure C.4: Charge and discharge current filters

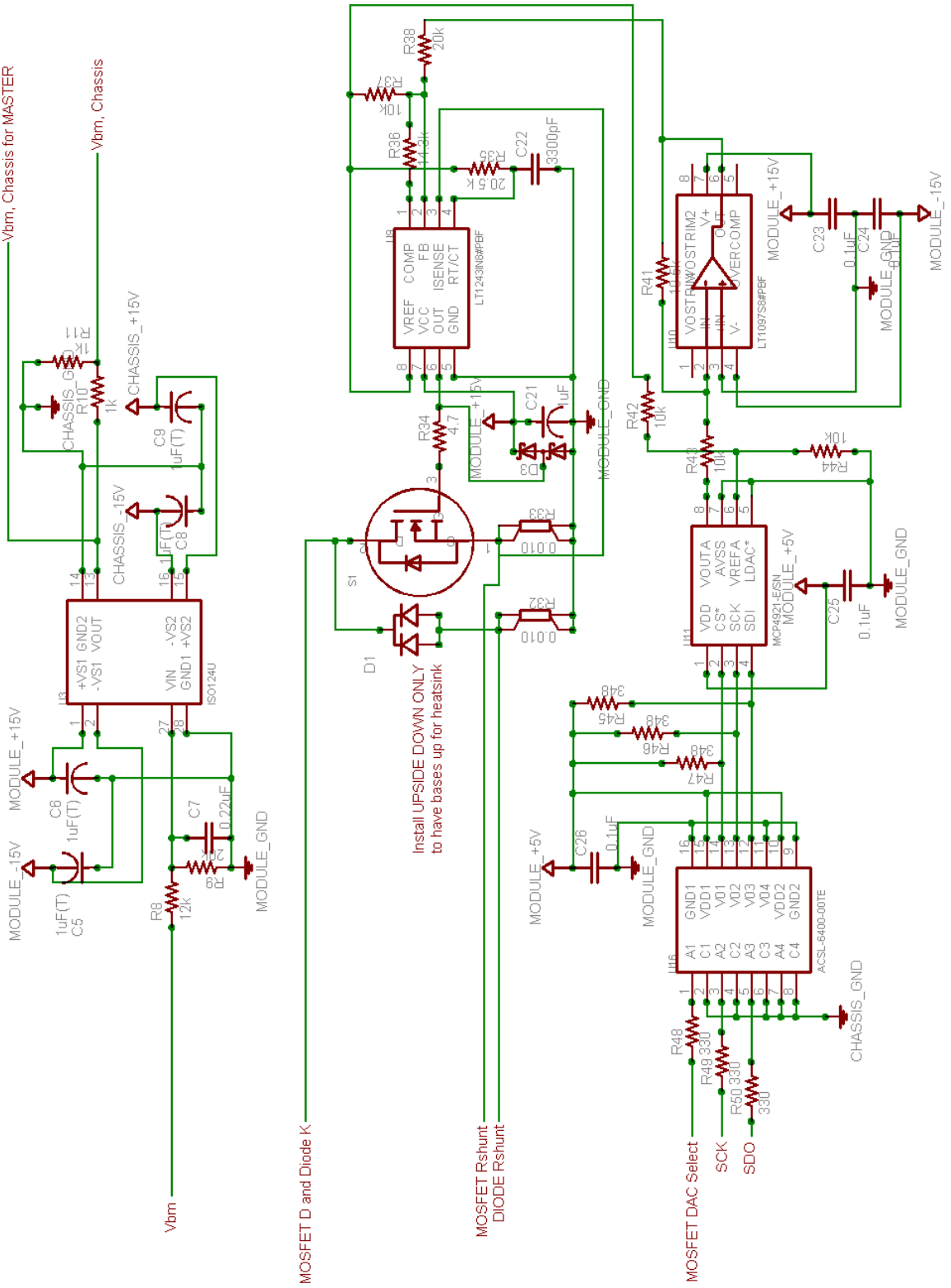



Figure C.5: Battery module voltage RC filter and MOSFET gate drive

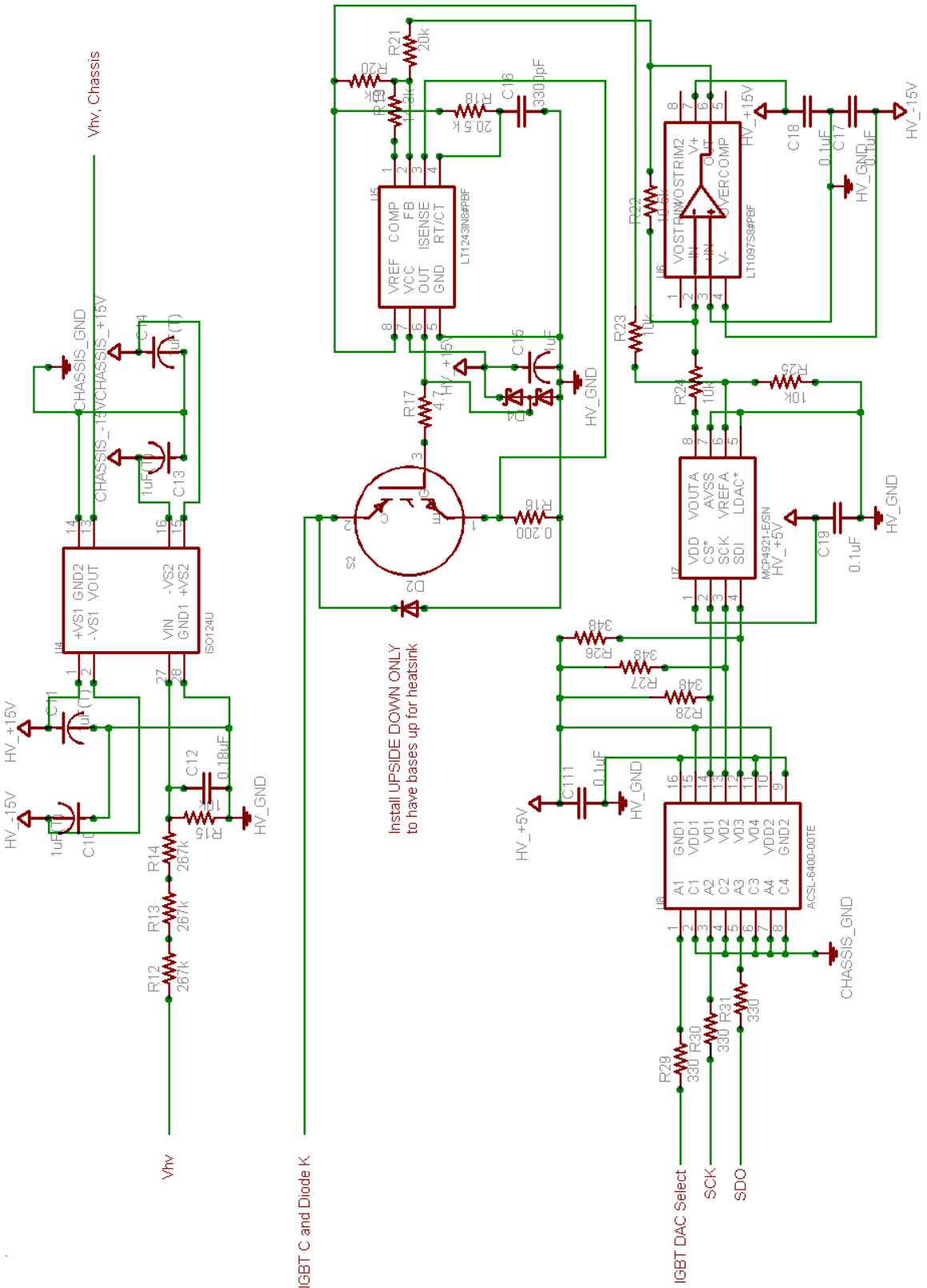



Figure C.6: High-Voltage bus voltage RC filter and IGBT gate drive

The first printed circuit board (PCB) image is the top layer, and is shown in Figure C.7. It is shown as if looking down at the top layer of the PCB. The left side of the board is where the dc-to-dc converters are located. The bottom, central area is where the microcontroller and RS485 transceiver are located. The central area is where the digital and analog isolation devices are located. The upper, central area is where the gate drive and filter circuitries are located. The bottom, right side of the board is where the module side power electronics are located. Finally, the top, right side of the board is where the bus side power electronics are located. The second PCB image is the bottom layer, and is shown in Figure C.8. It is shown as if looking up at the bottom layer of the PCB. This side of the board is mainly used to for traces, but the central area contains components for the filter circuitry.

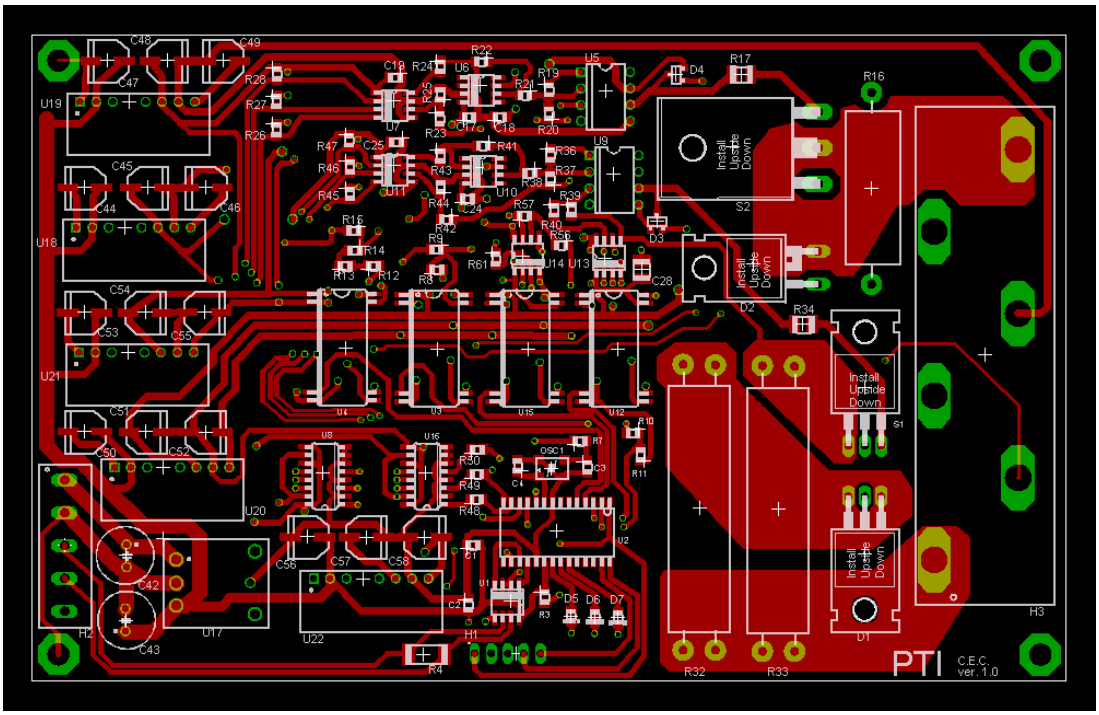



Figure C.7: Top of the PCB

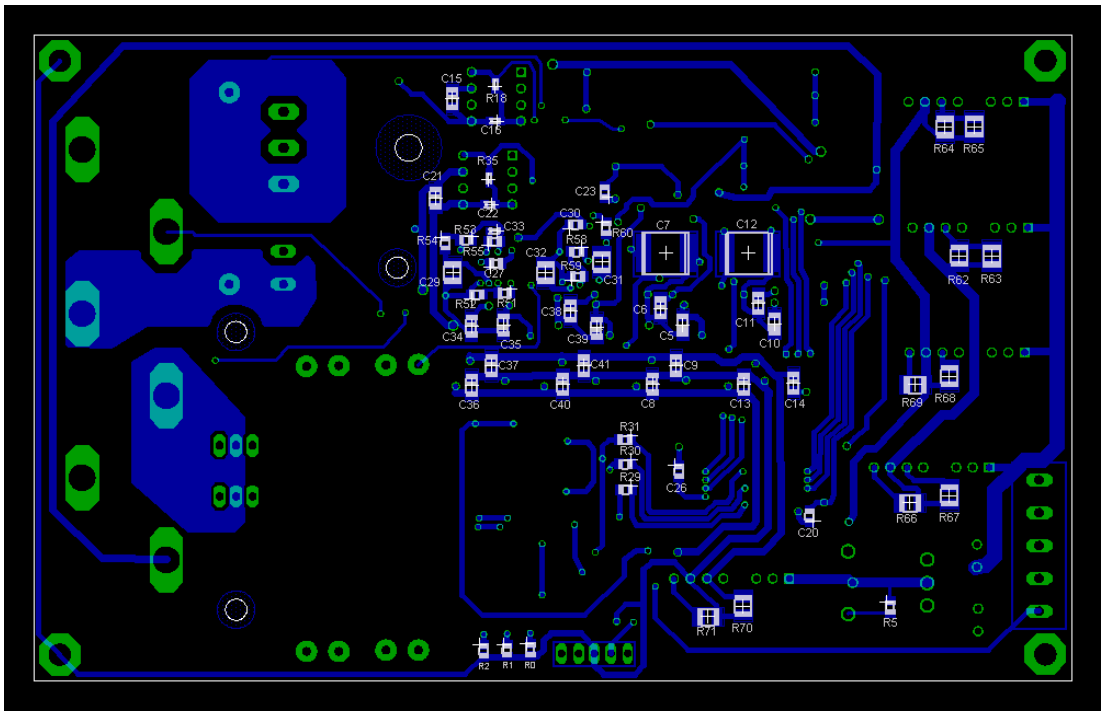



Figure C.8: Bottom of the PCB

The complete parts list for the PCB is provided next. The list includes a brief description, the board name, the manufacturer part number, the Digikey part number, and the total quantity required for one board. First, the capacitor list is shown in Table C.1. The diode list is shown in Table C.2. The resistor list is shown in Table C.3. The electronic switch list is shown in Table C.4. The integrated circuit list is shown in Table C.5. Finally, the header list is shown in Table C.6.

Table C.1: Capacitor list

| Description | Board Name | Manufacturer Part Number | Digikey Part Number | Number per board |
|---|---|---|---|---|
| 15pf | C3, C4 | C0805C150J5GACTU | 399-1111-1-ND | 3 |
| 3300pF, 5% | C16, C22 | C1608C0G1H332J | 445-1599-1-ND | 2 |
| 0.1uF,25V | C1, C2, C17, C18, C19, C20, C23, C24, C25, C26, C27, C59, C60 | ECJ-2VB1E104K | PCC1828CT-ND | 12 |
| 0.18uF, 2%, PPS | C12 | ECH-U1154GCV | PCF1508CT-ND | 1 |
| 0.22uF, 2%, PPS | C7 | ECH-U1224GCV | PCF1510CT-ND | 1 |
| 1uF, 35v, Tantalum | C5, C6, C8, C9, C10, C11, C13, C14, C15, C21, C34, C35, C36, C37, C38, C39, C40, C41 | 293D105X9035A2TE3 | 718-1116-1-ND | 18 |
| 10uF, 10% | C28, C29, C31, C32 | ECJ-4YB1A106K | PCC2170CT-ND | 4 |
| 100uF, 35v | C44, C45, C46, C47, C48, C49, C50, C51, C52, C53, C54, C55, C56, C57, C58 | EEE-1VA101XP | PCE3951CT-ND | 13 |
| 220uF, 35v, 720mAVrms | C42,C43 | ELXZ350ELL221MH15D | 565-1982-ND | 2 |

Table C.2: Diode list

| Description | Board Name | Manufacturer Part Number | Digikey Part Number | Number per board |
|---|---|---|---|---|
| 100v, 30A, INSTALL UPSIDE DOWN | D1 | 30CTQ100PBF | 30CTQ100PBF-ND | 1 |
| 1200v, 10A, INSTALL UPSIDE DOWN | D2 | 10ETF12 | 10ETF12-ND | 1 |
| 30V, Series, Shottkey | D3, D4 | BAT54S | BAT54SFSCT-ND | 2 |
| LED, Green, 35mcd, 2v | D5 | LTST-C171KGKT | 160-1426-1-ND | 1 |
| LED, Blue, 30mcd, 3.25v | D6 | LTST-C171TBKT | 160-1645-1-ND | 1 |
| LED, Red, 54mcd, 1.75v | D7 | LTST-C170KRKT | 160-1415-1-ND | 1 |

Table C.3: Resistor list

| Description | Board Name | Manufacturer Part Number | Digikey Part Number | Number per board |
|---|---|---|---|---|
| 0.01ohm, 7w, 1% | R32, R33 | 17FPR010E | 17FPR010E-ND | 2 |
| 0.2ohm, 5w, 1% | R16 | 15FR200E | 15FR200E-ND | 1 |
| 4.7ohm, 0.5W | R17, R34 | ERJ-P14J4R7U | P4.7ASCT-ND | 2 |
| 102ohm, 1% | R53, R60 | ERJ-6ENF1020V | P102CCT-ND | 2 |
| 120ohm, 1W | R4 | ERJ-1TYJ121U | PT120XCT-ND | 1 |
| 180ohm, 5% | R1 | ERJ-6GEYJ181V | P180ACT-ND | 1 |
| 249ohm, 0.25W | R62, R63, R64, R65 | MCR18EZHF2490 | RHM249FCT-ND | 4 |
| 300ohm | R0, R2 | ERJ-6GEYJ301V | P300ACT-ND | 2 |
| 330ohm | R29, R30, R31, R48, R49, R50 | ERJ-6GEYJ331V | P330ACT-ND | 6 |
| 348ohm | R5, R26, R27, R28, R45, R46, R47 | ERJ-6ENF3480V | P348CCT-ND | 7 |
| 1kohm, 0.5% | R10, R11, R52, R55, R59 | RR1220P-102-D | RR12P1.0KDCT-ND | 5 |
| 2.21kohm, 0.25W | R66, R67, R68, R69, R70, R71 | ERJ-8ENF2211V | P2.21KFCT-ND | 6 |
| 10kohm, 0.5% | R3, R15, R20, R23, R24, R25, R37, R42, R43, R44, R56, R57 | RR1220P-103-D | RR12P10.0KDCT-ND | 12 |
| 10.5kohm, 0.5% | R22, R41 | RR1220P-1052-D-M | RR12P10.5KDCT-ND | 2 |
| 12kohm, 0.5% | R8 | RR1220P-123-D | RR12P12.0KDCT-ND | 1 |
| 14.3kohm, 0.5% | R19, R36 | RR1220P-1432-D-M | RR12P14.3KDCT-ND | 2 |
| 20kohm, 0.5% | R9, R21, R38 | RR1220P-203-D | RR12P20.0KDCT-ND | 3 |
| 20.5kohm, 0.5% | R18, R35 | RR0816P-2052-D-31C | RR08P20.5KDCT-ND | 2 |
| 30kohm, 0.5% | R51, R58 | RR1220P-303-D | RR12P30.0KDCT-ND | 2 |
| 100kohm, 0.5% | R56, R57 | RR1220P-104-D | RR12P100KDCT-ND | 2 |
| 267kohm, 1% | R12, R13, R14 | ERJ-6ENF2673V | P267KCCT-ND | 3 |

Table C.4: Electronic switch list

| Description | Board Name | Manufacturer Part Number | Digikey Part Number | Number per board |
|---|---|---|---|---|
| MOSFET | S1 | IRFB4110PBF | IRFB4110PBF-ND | 1 |
| IGBT | S2 | IRG4PH40UPBF | IRG4PH40UPBF-ND | 1 |

Table C.5: Integrated circuit list

| Description | Board Name | Manufacturer Part Number | Digikey Part Number | Number per board |
|---|---|---|---|---|
| PWM Driver | U5, U9 | LT1243IN8#PBF | LT1243IN8#PBF-ND | 2 |
| DAC | U7, U11 | MCP4921-E/SN | MCP4921-E/SN-ND | 2 |
| Analog Isolation | U3, U4, U12, U15 | ISO124U | ISO124U-ND | 4 |
| Digital Isolation | U8, U16 | ACSL-6400-00TE | 516-1603-5-ND | 2 |
| OP-AMP | U13, U14 | LT1361CS8#PBF | LT1361CS8#PBF-ND | 2 |
| OP-AMP | U6, U10 | LT1097S8#PBF | LT1097S8#PBF-ND | 2 |
| Microcontroller | U2 | PIC18F2520-I/SO | PIC18F2520-I/SO-ND | 1 |
| Oscillator | OSC1 | ECS-100-20-30B-TR | XC1117CT-ND | 1 |
| RS485 | U1 | SN65HVD23D | 296-15230-5-ND | 1 |
| Voltage Regulator | U17 | PTH08080WAH | 296-20432-ND | 1 |
| | U18, U19 | VWRBS2-D5-S5-SIP | 102-1505-ND | 2 |
| | U20, U21, U22 | VWRAS2-D5-D15-SIP | 102-1524-ND | 3 |

Table C.6: Header list

| Description | Board Name | Manufacturer Part Number | Digikey Part Number | Number per board |
|---|---|---|---|---|
| ICD2 | H1 | SAM1029-50-ND | TSW-150-07-G-S | 0.1 |
| Board Signal and Power | H2 | 1729157 | 277-1250-ND | 1 |
| Power | H3 | 38660-7806 | WM5757-ND | 1 |