ELECTRIC MOTORCYCLE BATTERY SYSTEM FINAL REPORT

Power System Management Team

Messiah College
Department of Engineering
Collaboratory – Transportation Group

Matt Ayre
Amy Bowie
Brice Farrell
Paul Gustafson

12 May 2009

Abstract

The Power System Management team created a system to monitor and support a rechargeable Lithium-Ion battery pack and drive an in-hub brushless motor in a Kawasaki motorcycle. Using a microprocessor and additional circuitry, the system monitors the voltage of each battery module (one module consisting of ten cells), observes the total current of the entire system, and displays pertinent information to the motorcycle's driver. The custom microprocessor-based system incorporates an original algorithm to control the drive motor, and is a main component in an upgrade to an in-hub brushless DC motor and a higher voltage power source. Developing a lightweight Kevlar-based honeycomb battery cradle (with an aluminum housing) and incorporating the new batteries into the motorcycle frame will increase the efficiency and handling of the motorcycle. The Power System Management Team consists of Matt Ayre, Amy Bowie, Brice Farrell, and Paul Gustafson. The Collaboratory's Transportation Group was the project's sponsor, and Dr. Pratt served as the team's advisor.

Table of Contents

1.0 Introduction

The following section will provide a complete description of the project and solution.

1.1 Description and Purpose/Benefit

For the past few years, the Collaboratory's Transportation Group has been supporting the idea of developing projects that advance the concept of a solar-powered commuter vehicle. Even though this technology already exists, it is still impractical for everyday use; the Transportation Group strove to find a way to make use of the existing methods while creating a more practical vehicle. While current work is focused on converting a conventional Kawasaki motorcycle into such a machine, the ultimate goal is to be able to apply the Group's work to create a small solar-powered car.

At present, previous Transportation Group teams have successfully converted the motorcycle over to one that uses lead-acid batteries. Although these batteries create a functional system, they are very heavy and do not provide very much range between charges. In order to improve the bike, Black & Decker was kind enough to donate over three hundred lithium-ion batteries that are not only much smaller and lighter, but have the capacity to significantly extend the bike's range. Therefore, the Power System Management Team needed to incorporate these new batteries into the motorcycle's frame, lowering its center of gravity to no more than two feet above the wheel base and making the bike easier to handle. Before actually being used on the bike, each of these batteries needed to be tested for Amp hour and voltage capacity so that the batteries could be matched to create balanced modules.

In addition to re-designing the layout of the motorcycle, the team needed to design and build a system to monitor each module of lithium-ion cells (with one module consisting of ten cells connected in parallel). The system was to consist of a microprocessor that controls its surrounding circuitry to measure the voltage of each module, in addition to the voltage and current of the battery pack as a whole. This battery-monitoring system needed to be able to interface with the microcontroller for

the bike's motor. The team's project was to be designed as adaptable enough to accommodate two different power requirements—the prototype will be used with the bike's existing drive train, but the design is easily adaptable for future use with a brushless DC motor. All measurements made by the processor needed to be sorted and sent to an LCD that informs the driver of the status of the lowest and highest voltage battery modules, the total pack voltage, and the total pack current.

Another aspect of the motorcycle performance that will be improved is the motor controller. The motor controller that is currently in use is not able to be used for a higher voltage system, but more importantly it will only drive brushed motors so the switch to a brushless motor requires a new controller as well. By creating a custom-designed motor controller that is able to control a brushless DC motor, the team will be able to tune the performance to match the specific requirements of the project and develop an application-specific solution. The motor controller will interface with the battery monitoring system to provide both rider and system safety. Since close cooperation is required in this aspect, the microprocessor will be shared between these two applications. Having complete control of the motor in this way opens up the possibility of implementing an algorithm for regenerative braking to further increase the overall efficiency of the system.

In order to effectively store and protect the lithium ion cells, a new battery box needed to be designed and constructed. The battery box was to utilize the most efficient arrangement of cells in order to maximize the small amount of available space. In other words, this introduces essentially two engineering problems in one: how to hold 300 Lithium Cells in place, and how to protect these cells from the elements and other damage. Its ultimate goal was to be as light as possible so as not to add unnecessary weight to the bike, while at the same time providing enough structure and support to shield the batteries while the motorcycle is in motion. Including all structure, the entire power system was to weigh less than 75 pounds.

Throughout the process, all the team's work was well-documented and presented in such a way that future Transportation Group members can benefit from the work, and use it toward prospective projects in the next few years.

As previously stated, the Transportation Group's ultimate long-term goal is to create an efficient and practical solar-powered commuter vehicle. In other words, the Group is trying to transform an ordinary motorcycle into a more environmentally-friendly and economically-justified mode of transportation. The Power System Management Team's specific project ties directly into these broader objectives, as the integration of these lithium-ion batteries takes the motorcycle multiple steps closer to being a feasible commuter vehicle. With an enhanced driver interface and lighter frame, not to mention significantly extended mileage, the new system will allow the driver to travel further distances (with an eventual goal of over 50 miles) without the fear of not having enough power to last through the return trip. These factors will yield many more opportunities for the electric motorcycle to be utilized, which will result in a reduced cost to the driver (in comparison to sometimes being forced into driving a gasoline-powered vehicle). Any improvements made on this motorcycle design will bring the Transportation Group closer to reaching its overall goals.

1.2 Literature Review

The following is a summary of our research for this project, including different types of rechargeable batteries available for use in electric vehicles (those currently available on the market) and the relevant types of possible microprocessors. After completing this exploration, we are reassured that the choice to use lithium-ion batteries is advantageous and the best choice for our current design, and decided on a specific HCS12 processor from Freescale Semiconductor. This section also highlights our different options and final selection of programming software, as well as all the information we have gathered about regenerative braking up to this point (please refer to Section 8.7, for source information):

BATTERY TYPES:

Lead-Acid

These batteries are very reliable, inexpensive, and have been in use for a number of years. Unfortunately, they are very heavy, and only have a specific energy of 50 Watt-hours per kilogram (Wh/kg). They are also characterized by a long life, as long as they are never brought below about 80% of their charge capacity. If frequently discharged below this point, they will quickly lose charge capacity. Since we want to be able to use the full battery power (not to mention reducing the weight of the bike), lead-acids would not be a good choice for our project.

Nickel-Cadmium (NiCd)

These batteries are very similar to the lead-acids, have cell voltages between 1.2 and 1.3 Volts, and specific energies of 30 to 50 Wh/kg. They also are characterized by a flat discharge voltage, a long life, and excellent reliability. Unfortunately, they are very expensive and only work well at low temperatures. There is a great deal of toxicity in the Cadmium, and various trials with these batteries inside electric vehicle applications have concluded that they have insufficient power in such situations. Therefore, they are not acceptable for use in our project.

Nickel-Metal-Hydride (NiMH)

These batteries are the successor to Nickel-Hydrogen batteries, and are already in use in some electric vehicles (the Chrysler Electric Epic Minivan, the Toyota Electric RAV-EV, and the HEV-Prius). They have the same discharge voltage and rate as the NiCd, but have a much greater capacity (60 to 80 Wh/kg, with a specific power as high as 250 W/kg). They have a longer life-cycle than lead-acid batteries, and are safe and fairly abuse-tolerant. Unfortunately, they are fairly expensive and have a higher self-discharge rate than NiCd. They are difficult to charge when hot, and have a rather low cell efficiency; therefore, they are most likely not the best choice for our application.

Lithium-Ion

Lithium-Ion batteries have a comparatively high voltage potential and a very low atomic mass, which creates an extremely light and powerful cell. They have high specific energies and specific powers, as well as high energy efficiency and excellent

performance at high temperatures. Lithium-Ions have a low self-discharge rate, are recyclable, and have been shown to be highly suitable for use in electric vehicle applications. They generally have a very high cost, but in our case, they have been donated to the school. Weighing all options, especially their light weight and high efficiency, Lithium-Ions seem to be the best choice for our project.

Lithium-Polymer

Lithium-Polymer batteries are very similar to Lithium-ion cells. However, these batteries make use of solid-state electrolytes and utilize the properties of Vanadium-oxide. They are very thin cells that have the added capability of being able to be formed into a battery of any size and shape to fit the available space. Their life-cycle and calendar life are excellent, however they are not applicable to our project for several reasons. Not only are they very expensive, but they can only operate optimally at temperatures between 80 and 100 degrees Celsius, a condition that would be very difficult to meet.

Zinc-Air

These batteries are still an emerging technology and are quite expensive, and are therefore not a good choice for this project. However, they do seem very promising—they have already been tested in electric vehicles (particularly in Mercedes-Benz postal vans), and have shown the ability to last between 300 and 600 kilometers between recharges. These cells are analogous to fuel cells, and their recharging time can be very rapid with suitable infrastructure. Apparently, the battery is even recharged from outside of the battery, a concept unlike any of the others currently on the market. This battery proves to act more like a fuel-cell than a true battery due to this method of 'recharge.' The addition of a new 'charge' manifests itself in the addition of a new 'cell,' or the addition of more air/zinc to a current 'cell.' This system does not use electricity to recharge but rather actual materials.

Sodium-Sulfur (NaS)

These cells have a very high nominal electromechanical potential (2.71 V), and tend to be lower cost than Lithium-Ion cells. However, they have many drawbacks that make them unappealing for use on the bike. Their cell operating temperature is over 300 degrees Celsius, which implies a need for adequate insulation and a thermal

controlling unit. The batteries also do not have an overcharge mechanism, and if cells are brought too high they can develop a very high internal resistance that will pull down the voltage of the entire module. There are also a great deal of safety concerns with these batteries, as the chemical reaction that occurs inside the cells can cause excessive heat and even explosion.

Sodium-Metal-Chloride

These batteries are very similar to the NaS, but have provisions for overcharge and over-discharge. Unfortunately, they also have very high operating temperatures that require thick insulation and a thermal controlling unit—properties that we do not want to deal with on this project.

**It became apparent early on that our choice of Lithium-Ion batteries for use in our project was not only adequate, but incorporated our very best option out of all current technologies on the market.

MICROPROCESSORS:

Maxim Microcontrollers

Maxim manufactures an extensive line of products in what they call "battery management," including items that are compatible with the Lithium-ion batteries that we are using in the bike. Please refer to the list of references at the end of this document for a listing of datasheets of the various models we considered. Our first instinct was to use one of these microcontrollers because they are directly designed to serve our purpose of monitoring battery voltage, but we decided that it would be more advantageous to use a processor that could instead handle all aspects of our project (including calculations and integration with the motor controller). We would also need to have numerous Maxim controllers present, as each microcontroller could only handle monitoring a few cells at once. Even with skillfully planned out circuitry and switches to reduce the number of controllers needed, it would still require several units—we concluded that it would be a much more efficient design (not to mention a less expensive route) to use a general microprocessor instead of a specific microcontroller.

PIC

We next considered using a PIC, or Programmable Intelligent Computer, from Microchip Technology. This option was proposed to the group by Steve Frank, as some of their products have been used by senior projects in the past. While they supply many different kinds of processors, none of them were really that ideal for our particular project. Most of them restricted our functionality, and they had a limited instruction set which would have taken extra effort to work around. Overall, these processors would have been sufficient, but really just a minimal solution or a last resort. Therefore, we decided to continue researching other types of processors in an attempt to find something that would be a better fit for our needs.

HCS12

These processors are the next generation of the 68HC11 modules that were previously used in the Microprocessor Applications course at Messiah. After a thorough analysis, the Engineering department has decided to adopt HCS12 development boards from Freescale Semiconductor for future use in this particular class. Processors within this family are available in 16-bit processing with the ability to be programmed through a USB connection instead of a serial port. They have ample A/D converters, and have varying numbers of I/O ports depending on the particular model. We decided that an HCS12 processor was the best option that we had available to us—not only had we used a similar processor in one of our courses, but we would have support from various members of the Engineering Department staff who are fairly familiar with the processor family. After narrowing down our choice to two specific HSC12 processors, the 9S12C128 and the 9S12DT256, we made a list of the characteristics of each in order to make the most informed decision:

9S12C128 – 128 KB flash EEPROM, 2KB RAM, physical size of 3.8 x 2.0 inches, 60 pin header I/O ports

9S12DT256 – 256 KB flash EEPROM, 12KB RAM, 112-pin LQFP package of I/O ports, 2 8-channel A/D converters with 10-bit resolution

Since the two processors both cost the same amount of money, and the second one not only has more memory but was said to be most similar to the processor that will be used in the Microprocessor Applications course, we chose the 9S12DT256. It has

more I/O ports and two A/D converters, which will be very useful considering how many voltages we will need to be measuring in a short amount of time.

PROGRAMMING SOFTWARE:

There were two main options suggested to us by our advisor—ASMIDE and CodeWarrior. While ASMIDE is free, "OK" (in the words of Dr. Pratt), and easy to use, it has no simulator which caused us to decide that CodeWarrior would be a better choice for our project. Not only does it have a simulator, something that will be essential for use in the programming process, but it supports all versions of HCS12 and accepts multiple programming languages. We were able to quickly download a free "evaluation" version over the internet (which simply means that it has a code limit, which fortunately was much larger than any code we would ever need to write for this project). Although CodeWarrior seems to be a bit more complex than ASMIDE at first glance, and will involve much more familiarization and getting used to, we hope that it will be more beneficial in the long run.

REGENERATIVE BRAKING:

Regenerative braking transforms the mechanical energy of the car's motion into chemical potential energy in the car's batteries. While conceptually simple, this is somewhat harder to implement due to the realities and nuances of batteries. Primarily, the voltage of the current produced by the motor has to be raised so that it can force its way into the batteries.

In a traditional PWM (pulse width modulation) controller, the control line triggers a MOSFET, allowing current to flow to the motor. When this current stops flowing, the current generated by the motor can flow through a freewheeler diode.

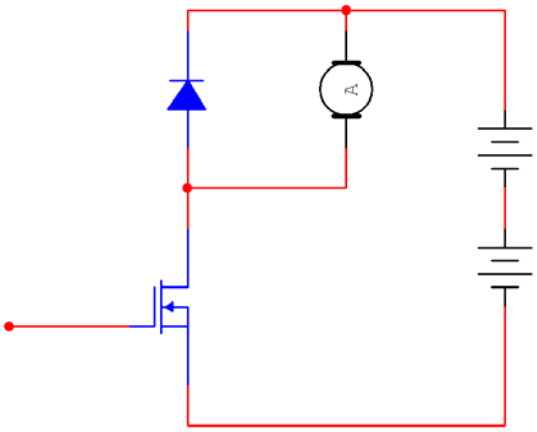


Figure1.2.1: DC Motor Control without Regenerative Braking Scheme

If this freewheeler diode is replaced with another MOSFET, the generated current can be controlled. By allowing it to build up in this motor-MOSFET-motor loop, and then shuffling the transistors and shunting it into the battery, regenerative braking can be achieved.

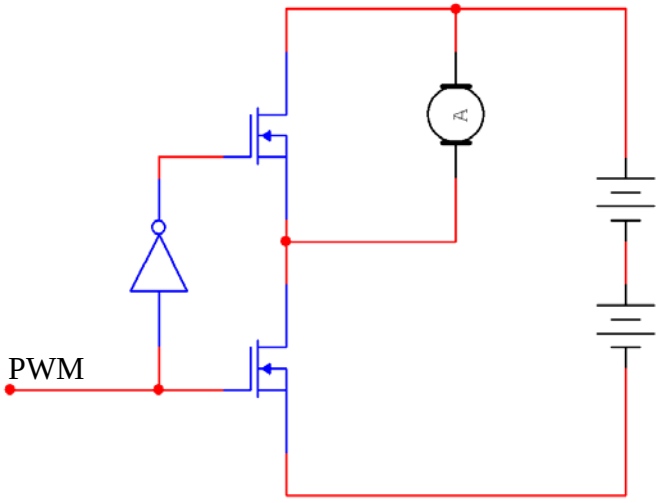


Figure 1.2.2: DC Motor Control with Regenerative Braking Scheme

While three-phase motors do differ from DC motors, the general idea is the same and the implementation is very similar. The main difference is that the motor produces

AC as measured between any two inputs. Early ideas tossed around by the team to simplify regenerative braking included using a transformer to raise the voltage. While we were initially very excited by this idea, we realized that it would be prohibitively heavy for implementation in a motorcycle. A lighter way of raising the voltage of an AC signal is to use a voltage-doubler, -tripler, or -multiplier.

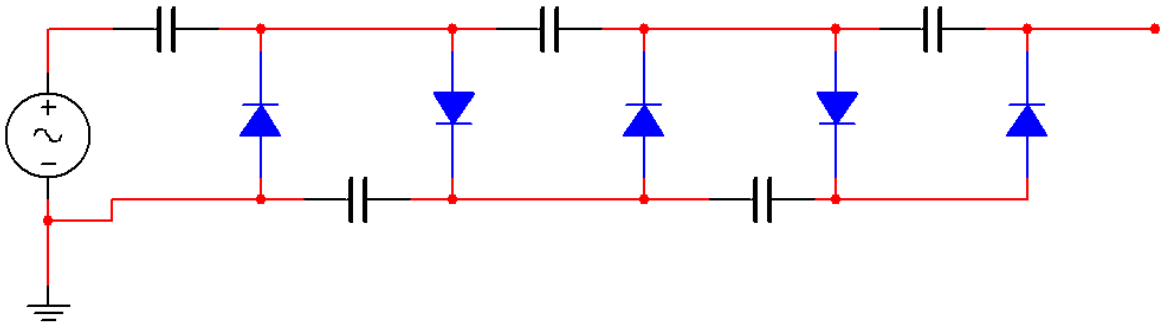


Figure 1.2.3: Voltage Multiplier

This uses a ladder of capacitors and diodes to raise the voltage. However, there were two problems with this. One: voltage multipliers are inefficient, and two: diodes and capacitors capable of handling the amount of current produced by a motorcycle braking from 65 mph are very expensive.

However, by treating the AC signals as a series of short-duration DC signals, the MOSFET-based regenerative solution can be applied to a three-phase motor. This also allows us to use the power electronics from the genesis project, a control board we know is capable of regenerative braking. As long as the chip is fast enough to switch the transistors faster than the phase of the motor is switching, current can be continually built up and dumped into the motor, smoothed by the in-line capacitors.

1.3 Solution

It is evident from our project description and original Gantt chart (Section 8.6) that we set incredibly lofty goals for ourselves to accomplish in just two semesters. While everything seemed completely feasible to us when we were planning things out in the fall, we very quickly realized that projects inevitably take at least twice as long as originally estimated. Since we had very little experience with project planning and

scheduling, we assumed that we could complete complicated tasks flawlessly the first time through, in less time than it would probably take a very experienced team of engineers.

Needless to say, we did not get as far as we had hoped on this project. We did learn a great deal about project management and scheduling along the way, but this still does not make up for the fact that everything we sought to accomplish was not finished. While we were very successful with completing the majority of the monitoring circuitry and mechanical construction, a lot is still left for future project groups. The power system still needs to be realized, the battery box needs numerous finishing touches, but most importantly, any degree of system integration has yet to take place. The batteries need to be matched and wired into packs, the box needs to be mounted into the bike, and all electrical systems need to be combined and integrated, not to mention overall testing. A more detailed explanation of these steps will be given in the future work section of this document (Section 7.0). The following paragraphs present the solutions that we *did* develop, as well as justification for our choices.

The final pack and frame design was the product of both the nature of the honeycomb pattern and the space available to house the pack in the motorcycle. The honeycomb pattern allows the frame to be shaped in many different configurations, but the hardest obstacle to overcome was the necessary groupings of the lithium cells. The lithium cells need to be grouped in modules of 10 cells, this meant that each panel of cells in the pack needed to contain some multiple of 10 modules. Another issue with the design was the very limited space available inside the motorcycle. The current brushed-DC motor in the bike occupies a large amount of space that limited both the size and shape of the pack design.

The frame to contain the pack was made out of TIG Welded 6061 Aluminum. It contains the cells while also adding support to the motorcycle frame. While the frame that was completed, it is not the correct size to accommodate all 300 cells. The

frame, as it is designed, will fit into the bike and provide the necessary support and containment. The front mounting system of the design is also built and functioning.

The electrical component of the project was actually twofold: the battery monitoring system and the brushless motor controller. Since one objective of the project was to implement the new battery pack with the current motor and controller, the battery monitoring system had a higher priority than developing the new controller. The circuitry for this consisted of an array of transistors and resistors which would allow individual measurement of the voltage at each of the 30 battery modules. Each module had a voltage divider with a common lower resistance for all of them so that the measurements taken would be consistent. A module was selected for measurement by the microprocessor through two 16 channel demultiplexers which activated a transistor in the middle of the voltage divider for that module, connecting the upper and lower resistors and creating a scaled voltage for the analog to digital converter on the microprocessor to read. As the program on the microprocessor ran, it would cycle through each module in the pack and keep track of the voltages as it went, subtracting the voltage measured previously to get the voltage of each module.

Controlling the new brushless DC motor that was being integrated into the hub of the rear wheel of the motorcycle by another group was the other aspect of the electrical side of the project. This required more work on hardware, since a controller algorithm had been developed by a group the previous year and we were planning on drawing from this for our design. However, they had implemented a low-power solution for the actual driving of the motor, which was insufficient for driving the bigger motor in the motorcycle. Instead of trying to find power MOSFETs big enough and designing the circuitry ourselves, we attempted to make use of the power board from the motor controller that was used in the Genesis solar car from 1997. The documentation for this controller could not be found, so testing and analysis had to start from the ground up, which caused several major delays and ultimately ended up preventing a successful implementation of this aspect of the project this semester. With more research on the operation of this board, a working model may still be possible.

As our team decided how we were going to implement each of these parts of the electrical system, we had several options from which to choose. Our first main decision was the microprocessor, which we chose over several alternatives for a few main reasons: it was small but still had adequate I/O ports for our needs, it was relatively inexpensive, and it was being used by a class so there were a couple of faculty members that would have experience with it in case we had questions or ran into problems. The battery monitoring circuitry was not as simple a choice. We went through several design iterations with different types of transistors before finally settling on the final working design. We could have also used analog switches or differential amplifiers or just used a separate analog to digital converter, but finding any of these with a high common-mode tolerance was very difficult and would have been more costly than our solution with the transistors and voltage dividers. The last main component was the power board for driving the motor. The power FETs that were used last year were one option, but we were given the opportunity to work with the motor controller from the Genesis '97 solar car, which had a commercial grade power board in it but did not have any documentation that could be found. The power board circuitry appeared to be relatively simple, so we decided to try to figure out how it worked and use that instead of attempting to design our own board with whatever large FETs we could find. Throughout the project our main driving force in choosing components for the system was functionality and ease of use, to produce a final product that works well and efficiently and is easy to operate and maintain.

2.0 Design Process

This section will summarize the processes that led to our design, including descriptions of analysis and experimental work.

2.1 Electrical Design

Overall design:

The batteries will be arranged into modules—one module consists of ten cells connected in parallel. Three hundred cells will be used overall, and with the current motor there will be two sets connected in parallel, each set consisting of fifteen

modules that are connected in series. When the new motor is installed in the future, it will require a higher voltage, and so the battery arrangement will be adapted; there will only be one continuous pack of thirty modules connected in series.

Monitoring circuitry:

In order to monitor each module's voltage, the voltage coming out of the pack must be stepped down to a level that the analog-to-digital converters on the processor can handle. Therefore, we have designed an interface consisting of a resistor voltage divider, transistors, and demultiplexers. All figures mentioned in this section can be found in the Appendix – Section 8.4.

A basic diagram of the circuit is visible in Figure 8.4.1. The voltage dividers will be at a set ratio that is known by the processor so that the chip can scale up the measurement it receives from the converter back into an accurate voltage reading. Essentially, each module is connected to a resistor, which in turn is connected to a transistor (which acts as a switch). All of the transistors in a set are connected to another resistor to create a voltage divider, and the line off the voltage divider is sent into the processor's A/D converter. A demultiplexer is connected to the gate of each transistor, and the processor sends a signal to the demultiplexer to determine which module is connected to the A/D converter (only one signal from a given set can be sent through at a time to each converter).

When the number assigned to a battery module is sent to the demultiplexer, the demultiplexer excites one of its control lines. By using this intermediary stage we have greatly increased the number of battery modules that we can address. Because we only want one battery module to be read at a time the demultiplexer is a perfect fit. This line that the demultiplexer pushes to high then activates the transistor for its battery module, effectively closing the switch and allowing the voltage to pass through to the analog-to-digital converter, as seen in Figure 8.4.2.

Originally, we had planned on using BJT's because we were more comfortable with them and knew more about them. Unfortunately, to turn the type of BJT that we were going to use on, the voltage at the emitter has to be about 0.7V greater than that of the base. Because the emitter would have been connected into the ATD converter of our processor, it would have been at a place that would have a constantly changing voltage, between 4.7 and 0V. Because of the changing nature of this voltage, it would have been incredibly difficult to maintain (or even know) whether you were maintaining the necessary voltage difference or not, implying that you would not really be able to control whether or not each transistor was on or off. To fix this problem, we decided to use FET's. This type of transistor is a bit easier to control—the polarity of the voltage between the gate and the source is all that matters for turning them on and off. As seen in the figure below, we could apply either a 5V or 0V signal from the demultiplexer, and immediately know whether or not we were turning the transistor on or off (5V will always create a positive difference, and 0V will always create a negative difference).

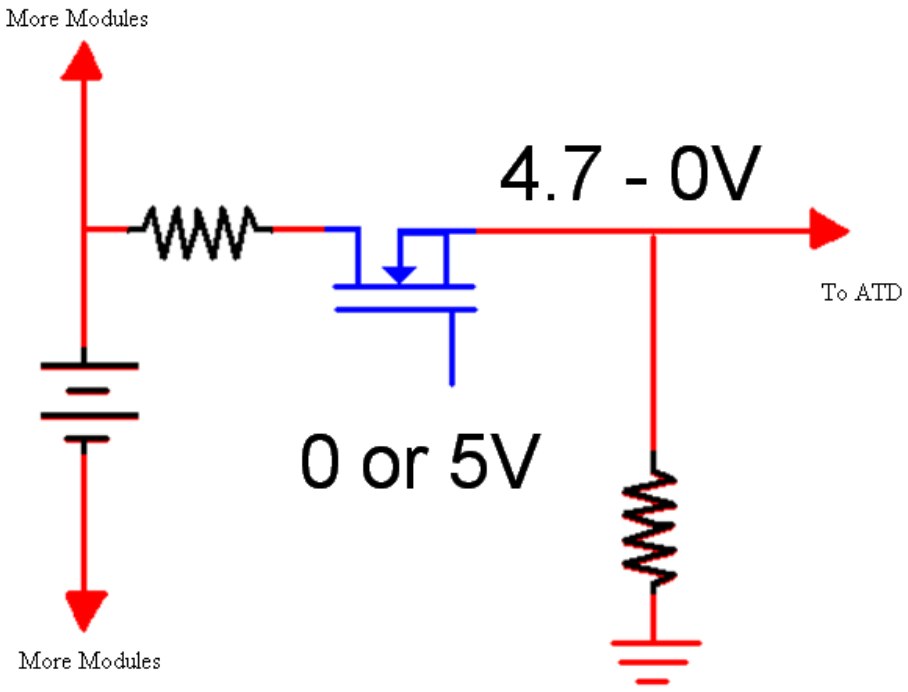


Figure 2.1.1 – Example of transistor circuitry

The LCD is a simple 2*24 character display with an embedded control chip. By placing the appropriate control bits – such as setting position – or character codes on its input lines, and then flashing an enable line, the controller chip responds displaying the data on the screen. Because of the control chip, the program doesn't need to worry about the pixels that form the letters, allowing it to run faster.

Power circuitry:

The design of our power circuitry was one chosen for expediency over optimal design. We had initially intended to use a scratch-built motor controller, including all the power transistors. However, an opportunity presented itself in the form of the 1997 motor controller used by the Genesis Solar Racing Team. After dissecting and reverse engineering the controller, we found that the driver board had the chip numbers burned off, and presented no avenue for the degree of control we desired. The power board was simpler to trace, and we discovered that it was built with high-quality high-amperage components, and controlled by six opto-isolated FET control circuits. We decided to interface with this board so that we could be sure of our ability to control the full-size motor being mounted in the motorcycle. As testing of this board proceeded, however, it became apparent that something was not operating as expected. We disassembled the board into increasingly smaller pieces, testing each part as we went, but everything worked as expected except for the FET pairs that controlled each phase. Individually the FETs worked fine, but when we attempted to actually activate any of the phases, either at motor control switching speed or statically, the pairs would function incorrectly and somehow short each phase between power and ground, which simply shunted current instead of routing it to the phase output. The only thing we were not able to try before running out of time for further testing was to remove the FETs from the board and test them individually instead of in pairs.

Motor Controller:

The motor controller design was probably the most flexible aspect of the whole project, since it was implemented primarily in software and had only a few requirements for the processor on which it would run. In order to run a brushless motor, the program needs to know where the motor is, requiring three inputs from the hall effect sensors on the motor, it needs to know what the operator is requesting, requiring two analog inputs from the throttle and brake in our case, and it must be able to activate the phases on the motor through the power circuitry, requiring three outputs. How these connections are implemented does not change the basic operation of the motor controller, just the interface, so we were able to write most of the motor controller code and test it with a dummy motor (just LEDs connected to the outputs) before having operational power circuitry. The program worked as expected for the limited testing functionality we were able to achieve, responding to a variable input which changed the pulse rate of the output, driving our imaginary motor faster or slower. Once basic functionality was established, focus shifted to the physical aspects in order to provide a real testing scenario, which would be much more informative and useful than the dry tests.

Processor:

The design work related to the processor was mainly code structure, since the processor itself was already functional on arrival and we only had to connect it to the various other components of the system. For the main program design, we started with drawing flowcharts (see Section 8.2) to direct the rest of the programming process. From these we wrote pseudocode based on the blocks of the flowcharts, turning some blocks or parts of blocks into small separate procedures to facilitate the actual code (see Section 8.3). As we tested the processor with the different parts of our design, we added real code lines to replace pseudocode for the aspects that were being tested. The analog to digital converter was one example of this, as we did not know what line we would actually use until we had worked out how the connections to the converter would actually be implemented. There are still a few pieces of pseudocode that need to be replaced as the last parts of the hardware are finalized.

Integration:

The overall design and integration of our electronics was driven primarily by the number of available data lines on our microprocessor. Because we needed at least 30 inputs (for all the module voltages) and at least 22 outputs (for the LCD and the demultiplexers control lines) our circuit design was driven by a desire to maximize dual-purpose lines while maintaining simple electronics. For example, we have 30 battery modules from which we need to read the voltage. After splitting the voltage down to a 0V-5V level, we use the above-mentioned demultiplexers and transistors to control which of these voltages is fed to the analog-to-digital converter. However, since we only need the demultiplexer input select lines during a brief period they can be overlapped with the LCD data lines – which are also only needed for a brief period. In order to keep from displaying voltage line numbers, or selecting battery pack "Warning: Low Voltage" we made sure that the multiplexers and the LCD had dedicated enable lines. Alternately, disabling both lines saves power. This overlap of data lines was repeated within the demultiplexers, with one line choosing which demultiplexer chip the four input data lines apply to. We also skipped one of the pins on the LCD screen, as it is only used for reading data back to the processor. These shortcuts allowed us to use only ten output lines, and one analog-to-digital converter, saving us room for future improvements.

2.2 Mechanical Design

The mechanical aspect of this project revolved solely around the packaging of the individual Lithium cells for the battery pack. This was accomplished using a Kevlar based honeycomb structure with aluminum housing. The housing also adds rigidity between the front and rear motor mounts on the motorcycle's frame.

In order to hold the cells in a rigid, lightweight, repeatable pattern, we were required to consider many different packaging schemes. These included egg crate designs, PVC cradles, simple stacked designs, and finally a honeycomb structure that would provide the support necessary but also a flexibility of overall design. The honeycomb structure allows the batteries to be secured in a close-packed design that nests the

batteries together without much wasted space. This tight pattern allowed us to package 300 cells in a relatively small area.

In order to house the Lithium Ion cells in a secure manner that also provides a close packed structure our design has gone through several iterations. The first thought was to create a grid of square holes that accept the lithium cells and hold them tight. This would arrange the cells in a lattice structure that would allow for easy access, but would not conserve space very well.

The other option we considered was simply packing the cells on top of one another; this would allow the cells to be as close-packed as possible. This would allow us to put the largest amount of cells in the smallest area. However, this is difficult to secure in structure that will support each battery and prevent it from moving axially in the box.

Our solution for the battery cradle is a honeycomb structure that is somewhere between the two aforementioned ideas. The hexagons allow us to pack the cells very close to one another, but still give us structure and strength. The hexagons are a little more difficult to manufacture than simple square grid, but the savings in space will be ideal.

The battery cradles themselves will be made of Kevlar, arranged in a honeycomb pattern. The Kevlar will be formed in a sandwich of bent sheet metal in order to create the half honeycomb pattern. The sheets that will be used for the battery surround will also be formed in this manner, only it will be pressed into flat sheets that will be glued to the aluminum frame.

The second feature of the battery pack is the aluminum frame in which the honeycomb is housed. The aluminum frame needed to provide structural rigidity to the motorcycle frame and also allow easy access to the lithium cells. This was simple enough to design, however placing the mounts and aligning the frame the correct way in the frame was a little more difficult. The space limitations of the interior of the

motorcycle required many iterations of the frame design as shown below in Figure 2.2.1. The final design is also shown below in Figure 2.2.2.

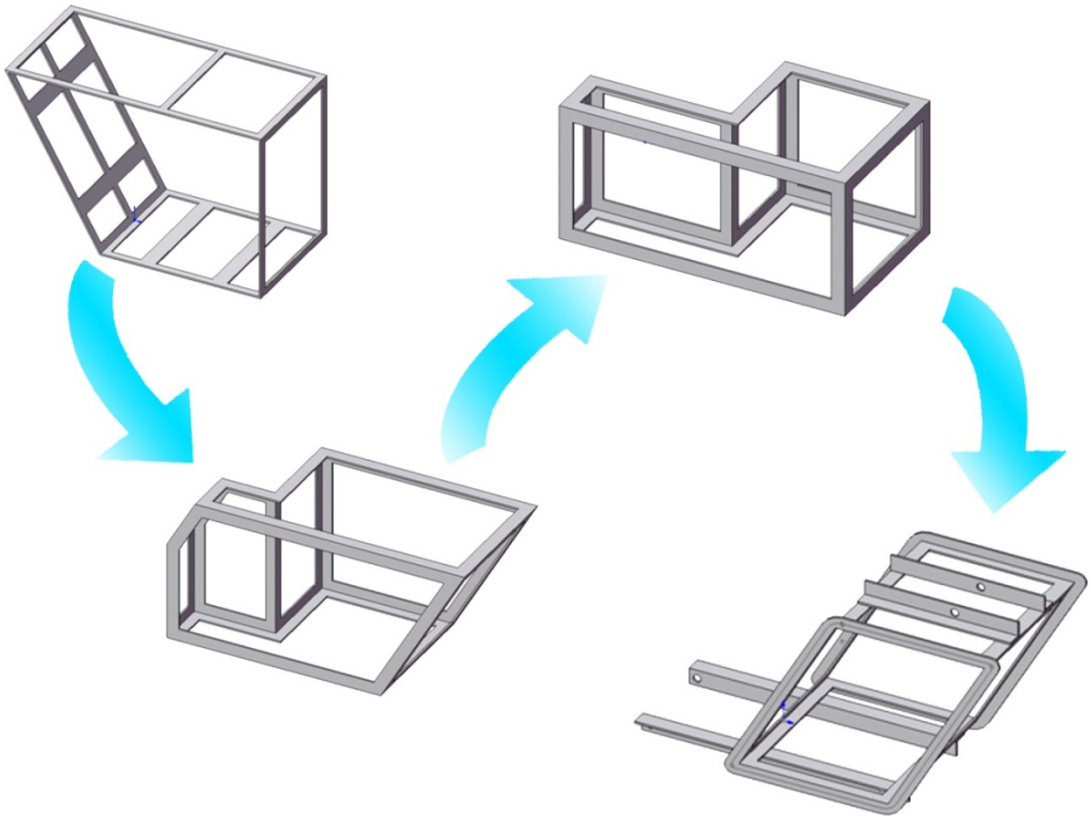


Figure 2.2.1 – Frame Progression

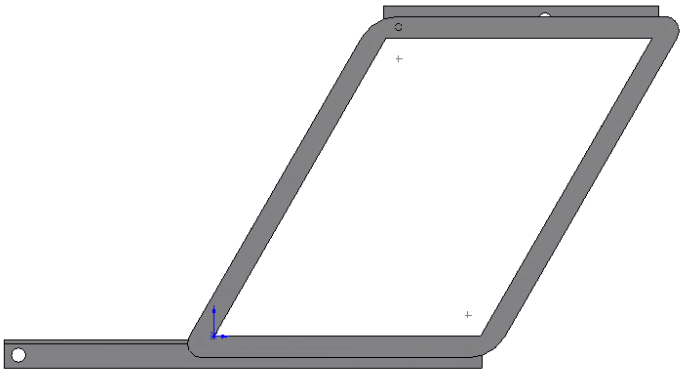


Figure 2.2.2 – Final Frame (Side)

3.0 Implementation

This section will summarize the construction and testing procedures and results involved in our project.

3.1 Construction

In regards to electrical construction, only the monitoring circuitry was actually seen though from start to finish this semester. The final prototype itself was built on two breadboards, while the LCD was temporarily attached to a third. It should be noted that minimal "construction" did take place involving the processor, however it only consisted of attaching the wires necessary to actually use it. A ribbon cable was connected to the I/O ports of the processor and run to a breadboard, where numerous wires were then sent over to their corresponding places on the LCD control chip and monitoring circuitry, as well as the logic indicators on a Proto-Board (for testing purposes only). The LCD itself was also connected to a 5V power supply in order to run its backlight.

The monitoring circuitry final design was described in Section 2.1, and explanatory figures and circuit diagrams can be found in the Appendix, Section 8.4. Thirty transistors were equally-spaced across one of the breadboards, and each corresponding first resistor of the voltage divider was connected to the source of the transistors. On the other side of those resistors was a labeled wire that will eventually run from a battery pack to the circuit. For now, the wires are hanging free, and were connected to power supplies set at reasonable values for testing when needed. At the gate of each transistor, a wire (labeled with the corresponding pack number) was run from that breadboard over to the second breadboard, where it was connected to the corresponding output line of one of the two demultiplexers (the second breadboard will be detailed in the next paragraph). Since all of the drains of all of the transistors are essentially connected together through the second resistor of the voltage divider, each drain was connected to a common line on the breadboard, which was then run over to the second breadboard and connected to that other resistor.

The second breadboard is not quite as busy as the first. It only contains the two demultiplexers and the second resistor of the voltage divider. As previously

mentioned, one side of the voltage divider resistor is essentially connected to all of the drains of all of the transistors—the other side is simply grounded. The two demultiplexers will eventually have their control lines run by the processor. However, since we did not quite get that far with our program, they are all currently connected to switches on a Proto-Board. This allowed us to quickly and easily select which "pack" was being read; we simply set the strobe of the demultiplexer that was in use to high, and used to switches to input the binary number corresponding to whatever pack we were currently testing. As previously implied, all the output lines of the demultiplexer were run over to the first breadboard, into the corresponding gate of the thirty transistors.

Although not entirely complete, this set-up did allow us to use the analog to digital converter (ATD) of the processor to display values on the LCD screen. The ATD was connected into the monitoring circuit right above the second resistor of the voltage divider (right where all the drains of all the transistors converge). Because the processor was connected to the LCD, our program did run enough to display whatever voltage was being measured at the time, allowing us to gather enough data to create the "look-up table" as described in Section 3.2.

It is important to note that both breadboards were connected to a common ground where necessary, as well as to a 5V power supply to provide power to the demultiplexer chips.

While the monitoring circuitry was technically the only electrical "construction" that happened this semester, the team has developed detailed ideas for further construction and plans for implementation. These can be found in Section 7.0, Future Work. Were it not for the various programming and hardware difficulties, as well as "strange" things happening with the microprocessor throughout the testing phase, many more of these tasks would have been accomplished.

The construction of the final battery pack was a very labor intensive process consisting of many unique and challenging obstacles. Notably, working with both Kevlar and aluminum in the construction of battle box was difficult.

The Kevlar honeycomb was assembled using the following process:

1. Prepare work surface with newsprint to protect from pre-impregnated epoxy.
2. Unroll pre-impregnated Kevlar and trace pattern with permanent marker.
3. Cut through both Kevlar and paper backing using diamond serrated scissors.
4. Cut oven safe plastic wrap to a size to cover both sides of the Kevlar.
5. Remove paper back from Kevlar and place onto oven wrap; fold over oven wrap to cover Kevlar.
6. Place one end onto the mold and secure the end with binder clips. Move your way down the mold securing as you go. Make sure that the Kevlar forms to the shape correctly. Secure using binder clips every few inches along the form.
7. Bake at 310ºF for one (1) hour. Remove from oven and allow to cool fully before removing forms.

(Take care to cut Kevlar carefully as not to bunch up the fibers, this tends to happen if cutting too fast.)

The aluminum frame designed to hold the cells was TIG welded by Matt Ayre. The TIG welding process proved to be difficult, but not impossible. This was assembled by welding the side pieces together first to form the two parallelograms, these were then welded together with the cross pieces. It would be recommended for the next box construction that the cross pieces be welded to the side pieces first. Then the top, bottom, and sides could be welded together much more easily.

3.2 Operation

Unfortunately, as mentioned in various other sections of this document, our project did not make it as far as we had hoped this year. Because of this fact, the majority of our specifications do not apply. Most of the specs were referencing a finished product—an actual working motorcycle with all parts integrated and installed. Since

we cannot really compare our actual work back to these specifications to test the success of our project, in this section we will discuss the actual operation of what we did accomplish, as well as their success relative to “new” goals and guidelines and/or how our progress would have led us or the next project group to meet our original specs.

Throughout the year while working on other parts of the project, battery testing was going on in the background. We wanted to test every cell that would go in the bike to be able to match them and create modules that would function optimally. However, testing each cell required discharging and charging each cell while measuring the voltage and current throughout the process, which was a very time consuming process. We were fortunate to have several underclassmen to help us with this part of the project, and we were able to collect discharge data for most of the batteries, and charge data for a smaller number of them. After looking at the results from this initial testing, it appears that the batteries are remarkably uniform. We have not done extensive comparisons to find out how similar they actually are, since the testing is incomplete, but the current results are very promising, and next year hopefully the testing can be completed and a detailed analysis performed, as mentioned in Section 7.0 on future work.

Fortunately, we were able to achieve fairly consistent positive results with the LCD. At the beginning of our testing phases, we spent a great deal of time just figuring out how the LCD worked, displaying simple messages just to make sure we knew how to control where characters were located, as well as housekeeping issues like clearing the screen and controlling how fast/often the screen would update. Eventually we were able to use the LCD to complete a testing plan involving the entire monitoring circuitry, so it successfully displayed the values seen by the ATD converter. Our program also effectively created the desired display layout that will eventually be used when the entire system is installed on the bike, as seen here:

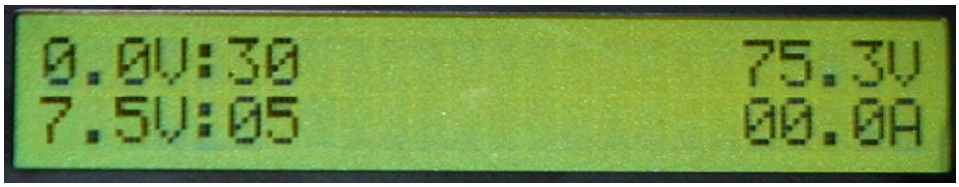


Figure 3.2.1 – LCD screen example

The left side of the display shows the lowest and highest pack voltages, as well as the actual pack numbers. The right side shows the total pack voltage, as well as a space for what eventually will be the total pack current. Overall, we are well on our way to meeting all specs involving the LCD. It already displays all pertinent information that will help the driver not only decide how much further he can go, but will help him figure out which packs are faulty or need to be charged (not to mention the two warning messages that were successfully displayed upon request regarding low voltage and high current). Because the LCD has a backlight that is constantly on as long as the LCD itself is receiving power, it will be easy to read in almost all driving conditions. As for the final spec involving LCD location, it will be up to the next project group to place the display in a location on the bike that will fulfill this requirement.

The monitoring circuitry itself was very successful in actual operation. All of the lines are functional and respond correctly—the correct line is selected when the corresponding demultiplexers receive certain input values. The microprocessor functions in regard to monitoring circuitry are all working, including control over the ATD converter used and taking values from the monitoring circuitry and sending them to the LCD. We were able to perform preliminary testing with actual microprocessor/demultiplexer integration; although the processor does not yet entirely control the demultiplexers, we were able to get the code to turn on and off one of the selection lines on the demultiplexer. This fact alone leads us to believe that with just a little bit more time, the code will be complete enough to totally and independently run the monitoring circuitry. When each line is selected, it draws less than .25mA of current, implying that the entire monitoring system (including the microprocessor) will draw much less than the original specification of 5W of power.

Since the processor is the central piece of the system, it was the first component to become functional, and will probably be the last part to be completed. Because there are so many things that use the processor, its degree of operability is complex. The code to run the monitoring circuitry is mostly developed, needing only a suitable test setup to check for full functionality. This will use a look-up table created from measurements of batteries compared to the output of the ATD and converted into a scale factor for each module that will allow fast calculation of the actual voltage at each module from the ATD reading. We tested it with an individual line to make sure that our design worked and that the interface between the processor and the other hardware would not cause problems. There have also been several tests with small component-specific programs to verify the operation of the LCD screen and the motor controller output. The LCD code has been added to the main program for certain events, but there are still a few parts of the display code to be implemented. The code for the motor control can be integrated into the main code block as soon as the hardware has been finished and the control code tested.

The power board probably caused the most difficulty for operation. All of the individual components appear to be functioning correctly, but when we attempted to drive our test motor or even just activate one phase, the power MOSFETs misbehaved and caused the board to simply shunt current from power to ground instead of turning on the phases as we wanted. Since we did not have any documentation for the motor controller from which we had gotten the power board, we had to reverse engineer the board ourselves to determine how it worked. After identifying all the components on the board and still having trouble making it run correctly, we consulted several faculty members as well as getting the opinions of other students on the problem, but to no avail. We did receive some good advice which enabled us to narrow the scope of the problem to either the FETs themselves or the large buffer capacitors, but after further testing even our transistor expert was stumped. However, with a little more time for specific testing and perhaps a few fresh minds working on the problem, we believe that the board will be operational and should easily meet our specifications for operation.

Another test procedure that pertains more to the motor controller side of the project involved checking the Hall Effect sensors. This simple test was designed only to determine whether the sensors were analog or digital. We first hooked them up to 5 V from a power supply, and then spun the motor to which they were already connected. By looking at an oscilloscope across one of the sensors, we saw that they were obviously digital; they showed definite change between 5 V and 0 V. This confirmation contributed to further thought and planning involving the motor/micro-controller part of the power system.

While the mechanical aspects of the build were a little tougher to test because of the extent to which the project was finished, preliminary testing was done with finite element methods. The frame was shown to support the necessary loads of the battery pack in static loading conditions. Testing was also done with loads simulating the G forces associated with daily driving. The max G force the current design can experience is 4.5 Gs, as shown below in Figure 3.2.2.

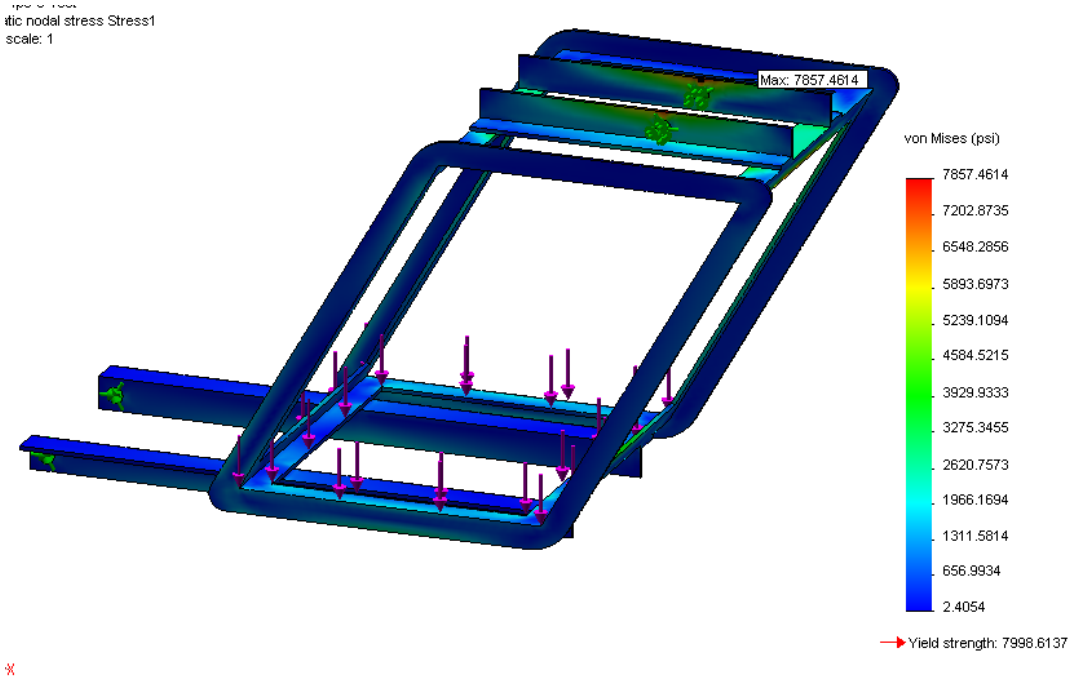


Figure 3.2.2 – Finite Element Model w/ 4.5G Load

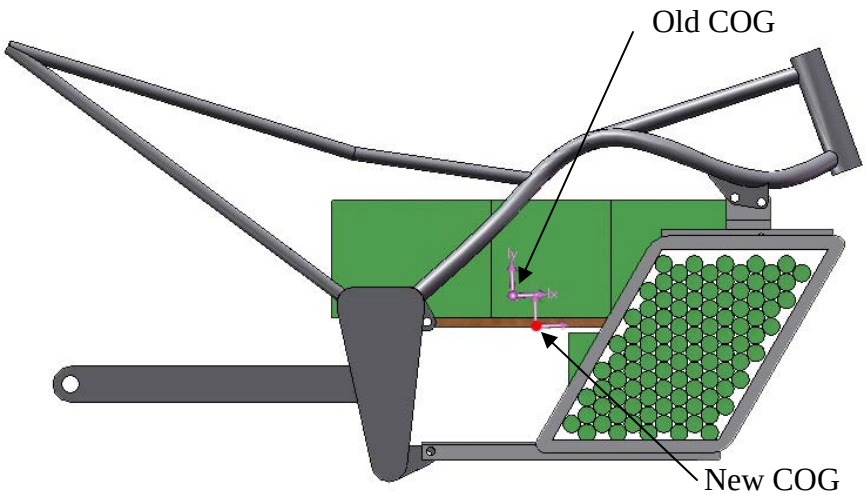


Figure 3.2.3 – Center Of Gravity Comparison

One of our objectives for the project was to lower the center of gravity to below 2 feet above the road surface. The results of our testing is shown above in Figure 3.2.3, indeed the new center of gravity is lower than the old. The exact numbers of the centroid are:

OLD

X = 6.97

Y = 11.98

NEW

X = 8.26

Y = 10.27

The centroid is lower than the original, but both of them are actually below the limit set by our objective. More importantly, however, is the reduction in total weight of the battery pack. The old battery pack weighed in at 128 pounds, while the new pack weighs only 52.5 pounds. Although this is well under our objective of 75 pounds, we still have the addition of the electronics components, but this would not likely weigh 25 pounds.

4.0 Project Management

This section discusses the work schedule of the project, including a comparison between predicted and actual completion times.

By comparing the two Gantt charts included (Sections 8.5 and 8.6), one can see the exact differences between our expected and our actual schedule. While showing the dates and completion times, these charts fail to explain the reasoning behind any of the delays.

Our project was most significantly set back by the delay in ordering the processor. After our initial request for the processor was submitted, there was confusion over who to bill, due to our team being split between Senior Project members and IPC students. In this confusion, the processor wasn't ordered until significantly later than intended, delaying all of the electrical phases of the project – those that depended on it. While we were able to start other portions of the project earlier, such as computer modeling of the frame, this delay set back all of the electrical portions of the project, most significantly the programming task. The programming itself also took about ten times as long as we had estimated; not only did it take us months to just become familiar with the microprocessor itself, but we encountered numerous quirks and "strange" things associated with it that we had to figure out before we could actually begin writing code for our own project.

Furthermore, some aspects of the project were added to the workload that we had not foreseen, or had lumped in with more general tasks – even when they deserved their own planning block to properly account for the time that they would consume. Regenerative braking research is a perfect example of this, as it went from being lumped in with battery research – a two week task – to its own two-month process. We were caught off guard by the dearth of information available on brushless regenerative braking, both online and in print. While designing the circuitry other delays came to the surface as well. The battery selection circuit, while simple at a glance, required a redesign to change from BJTs to FETs, stretching out that portion of our project.

The final factor contributing to the sliding of our schedule was that March was blocked out as a time without any tasks to complete. This resulted in less pressure during the previous tasks, as there was no pressing need to have them completed when they were scheduled to complete, as there was spare time later on. This relaxed attitude sabotaged the entire schedule.

There were several tasks on our original Gantt chart that we never even had the chance to begin. These mostly included overall testing of the entire power system and the bike itself, as well as various integration and installation procedures.

One aspect that did help to keep us on schedule, or at least closer than we would have otherwise been, was that our project was made up of almost three independent projects. This meant that electrical delays didn't slow down work on the battery frame, and that delays with the genesis controller funneled excess man-hours into the battery monitoring portion of the project.

5.0 Budget

This section presents a breakdown of the project's budget, including the expected cost of items received at no cost to our group.

We have been very fortunate throughout the course of this project to have incurred barely any cost at all. Not only did we receive a more than $6000 gift of lithium-ion batteries from Black & Decker, but we were able to obtain our microprocessor for free (by requesting the Student Demo edition). We were lucky enough to also obtain an LCD from the college, as it was one leftover from a previous senior project. Most of our electrical components were covered by the Engineering Department—our first few choices of transistors were all covered, as buying in bulk and keeping what we did not use was more appealing to the college than forcing us to pay shipping on certain components that totaled less than a dollar. We did, however, have to pay for our last order of transistors and demultiplexers, as well as the expedited shipping costs to have them arrive in time to test them before the final presentation. The

following table shows the total value of everything we used versus any costs we incurred. Of course, neither total truly reflects how much it would cost to build this project, as they only include the pieces we had time to tackle and finish. One would also need to consider the cost of the motorcycle itself, the motor controller, power components, assembly components, as well as testing and construction equipment.

Item	Qty.	Cost	Total Value	Project Total
Cells	300	$20.00	$6,000.00	Donation
Processor	1	$76.00	$76.00	Donation
LCD	1	$17.15	$17.15	Donation
Transistors	30	$1.08	$32.40	$32.40
Demultiplexer	2	$1.09	$2.18	$2.18
Misc Comps. & Wiring	1	$15.00	$15.00	$15.00
Kevlar	1	$100.00	$100.00	Donation
Epoxy	2	$6.50	$13.00	$13.00
Aluminum Angle	18	$0.64	$11.52	$11.52
Aluminum Bar Stock	24	$0.40	$9.60	$9.60
Misc Hardware	1	$25.56	$25.56	$25.56
			$6,302.41	**$109.26**

6.0 Conclusions

This section discusses the conclusions drawn from the entire project process, as well as what was learned along the way.

The mechanical aspect of our project was very insightful in the design challenges and different materials used. I learned that working with aluminum is difficult, but not impossible, while Kevlar is fairly easy but time consuming. The greatest lesson that our team learned was time management. Overall our time management of the project could have been much better, but I feel that both components of the project, electrical and mechanical, were still successful. The mechanical concepts of construction and design were proven with final designs and prototypes, and while it was not able to be

implemented, the work has been laid that will allow for manufacture of the battery box and frame.

The electrical portion of our project is most easily examined if we break it down into two different components – the battery monitoring component and the motor-driving component.

The battery-monitoring component was successful, both as a senior project and in implementation. The microprocessor selects an appropriate battery, reads the voltage, calculates basic statistics, and displays this information to the driver. The demultiplexers, LCD display, and microprocessor were all constructed, tested, and integrated successfully. In this portion of the project we learned to code in C, how to interface with LCD displays, and how to control analog-to-digital converters for practical applications.

The power circuitry component was less successful. We were unable to debug the power FETs used by the 1997 Genesis controller. However, we did learn to reverse engineer, and through inspection, testing, and research developed circuit diagrams for the current-switching portion of the motor controller. Furthermore, we learned how to decode the signals originating in hall effect sensors to determine which lines should be energized.

Another important lesson that we learned is that while it is often simpler to begin with preexisting components, rather than building from scratch, they might not work as expected. Not only did the transistors on the genesis chip puzzle us, but we also spent hours trying to debug an LCD screen that would only display the first eight characters, only to learn that it was a faulty controller chip. Overall, we also learned to plan realistically, and not plan to have everything done months early. This mistake let us feel as though we could delay everything indefinitely, as we had “all the time in the world” – which came back to be a problem for us in the last two weeks before our presentation.

7.0 Future Work

While we were able to accomplish a large percentage of our overall goals, there are still plenty of components to finalize and improvements to make. Given the opportunity to have a second chance at the project from the beginning, one of the main things that we would change is to start testing much earlier. We tried to do a lot before testing to see if it was operational or even feasible, and while most of our work was successful or at least on the right track the first time, we realized at the end that doing more testing right from the beginning would have been helpful in finding mistakes and changes that needed to be made sooner so that we would have had more time to remedy these situations, and maybe come up with even better ideas. Along the same lines, we also would have bought components earlier so that we could start building the parts of the design that we had while working through further designs. We attempted to have most of our design completed before starting construction, which wasted time when we had difficulty thinking which could have been used to put pieces together for initial testing. While trying to do everything at once is not a good idea, we could have benefitted from more concurrency.

Since we do not have the ability to go back and apply what we learned ourselves, we must pass the project on to others. To this end, we have several suggestions for what could be done next.

The motor controller software is mostly completed, but was not integrated due to a lack of testing on actual hardware. This code is spread out among a few files, mainly the BCP.mcp program that was our main depository of finished code, spintest.mcp which was what we used for testing the power board and the interaction with a variable resistor for throttle and brake input. We had several other files that were created throughout the year for specific testing of certain parts of the project, but the motor control code should be in these two files. We drew heavily on the work that the group before us had done, so that will be helpful in future coding as well.

The power board probably has the most potential for future accomplishment. The FETs should be taken off the board and tested individually to determine if they are all working correctly, and if they are, the weak link of the chain is most likely the buffer capacitors, which may have to be replaced, but should at least be individually tested to make sure they have not been compromised. If an alternate route is deemed the best route, then a whole new board will have to be designed and built, which will take research on FET operation and availability, plus overall brushless motor control to find out what is the purpose for the components on the board that we have, and whether we need them all or not.

Another avenue for extensive work is finishing the battery testing and cell matching for the modules. All of the data files from the testing need to be cleaned up to include only the relevant testing data, and then compared to determine the difference between them so that they can be assembled into 10-cell modules. The batteries will perform best if the modules consist of cells that have identical capacities, so that they will charge and discharge uniformly, so this task could have a major effect on the longevity of the battery pack.

In addition to simply displaying information, the team's goal could be to create an interactive display that allows the user to choose which information they would like to see. This display would also notify the driver of how many more miles and at what speed they can travel before the batteries need to be recharged.

On a more practical level, the circuitry designed needs to be implemented in a usable form. Currently, all of the circuitry is spread over multiple breadboards. To install this circuitry on the motorcycle a circuit board needs to be built. Ideally, in addition to containing the demultiplexers, the battery selection transistors and the voltage dividers, this board would also contain an 60 pin (2x30) header to allow the microprocessor to attach, and an additional cable-hookup up for the LCD screen, which would likely be mounted in a more distant position. Further cable hookups,

perhaps in db9 connectors such as the Genesis project used, would also be needed to attach the hall effect sensors, motor drive controllers, and throttle/brake inputs.

This would need to be contained in a box. The box for the current genesis controller is a possible choice, as it already has the connectors and is could be easily re-waterproofed. The board would then need to attach where the old control board was located, which would be simple if it were made of the right dimensions with holes in the four corners.

Another future expansion for the electric motorcycle controller would be the implementation of regenerative braking. Although we were able to develop an algorithm for regenerative braking, we were stymied in our effort to implement it, as we were unable to get the drive portion operational, which is a necessary first step. An implemented regenerative solution would expand the range of the vehicle, but would take a great deal of testing. Whereas the previous improvements could likely be completed by one person in a semester, we believe the regenerative implementation could take a year, or a multi-person team.

While the mechanical systems of the motorcycle were constructed enough to prove the feasibility of the design, much more construction will need to be done. As mentioned before the battery frame will need to be elongated by one inch to accommodate the extra height associated with the thick Kevlar. Additionally, more Kevlar honeycomb will need to be made and epoxied together. The panel that was constructed is perfectly acceptable and will serve as a pattern for the other two honeycomb panels. Kevlar will also need to be cured to cover the sides of the aluminum frame to waterproof the box. We would also like to see the additional design of a junction box to be mounted on the back of the battery box. The junction box would house a terminal block that would allow easily connection and disconnection of the cell panels without removing wires from the motor control unit.

8.0 Appendix

8.1 Original Specifications

The following is a list of the original design specifications we developed for this project. Although many of them were not realized, it is important to note what our original goals were:

- Physical Characteristics
  - Overall power system weight under 75 pounds
  - Center of gravity of the frame and power system no more than 2 feet above the wheel base
- Performance/Electrical Characteristics
  - Battery pack shall not limit the speed of the motorcycle up to 65 mph
  - Range will be greater than 50 miles under average driving conditions as defined by a specified test course
  - Voltage and current measurements shall be accurate to ±.01 V
  - There shall be over-current protection circuitry that will limit the total current to a maximum of 75 A
  - Surrounding circuitry and the microprocessor shall draw no more than 5 W from the total battery pack output
- Maintenance
  - Time between overhauls is no less than 6 months
  - Mean time between failure is no less than 3 years
- Cost
  - Total cost of project completion will be no more than $500.00, not to include $6000 battery donation from Black & Decker
- Environmental
  - Battery box is water resistant in light to moderate rain
  - All fasteners and structure associated with the power system will survive moderate road vibration analysis
- Aesthetics/Interfacing
  - Power system will display pertinent information in a manner and location that will not hinder the operation of the motorcycle
  - Display shall be visible in all lighting conditions experienced in dawn to dusk driving
  - Display switch will be located in a position so that the driver need not move hand further than six inches from the handlebar and need not move the throttle hand

8.2 Programming Flowcharts

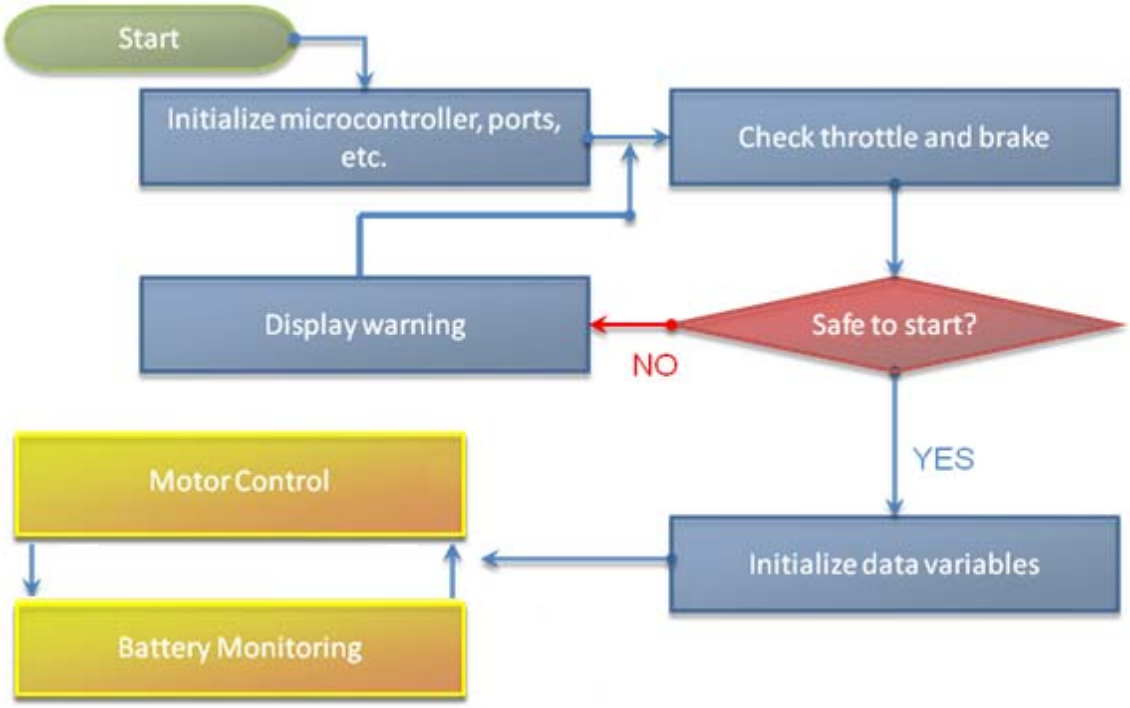


Figure 8.2.1: Flowchart for general program flow.

Figure 8.2.2: Flowchart for the Battery Monitoring component of the program.

Figure 8.3.3: Flowchart for the Motor Control component of the program.

8.3 Code

```
/*
Pin map, # means dedicated
a0# - text write enable
a1# - lcd enable
a2# - global demux enable
a3 - demux 1/2 select
b0\   \
b1 |   |- demux input lines
b2 |   |
b3 |  /
b4 |- LCD data lines
b5 |
b6 |
b7/

*/
#include <hidef.h>        /* common defines and macros */
#include <mc9s12dt256.h>  /* derivative information */
#include <stdio.h>

#define BATT_ATD   ATD0DR4                  //ATD0DR4 = measuring battery voltages
#define CURR_ATD   ATD0DR1                  //ATD0DR1 = measuring total current
#define HALL_EFFECTS (PORTA && 0b11100000)   //hall effect sensors coming in
                                                    upper 3 of port A???
#define THROTTLE   (PORTA && 0b00010000)     //throttle input line??
                                                     needs a different port
#define BRAKE      (PORTA && 0b00001000)     //brake input line??
                                                     needs a different port


const int ARRAY_SIZE = 30;
const float SHUNT_RESISTANCE = .03;          //ohms  ****NEED REAL VALUE****



//Declarations
void batteries(int moduleStepper, float totalCurrent, int highCFlag, float
moduleVoltages[],
           float rawVoltages[], float tempVoltage, int lowVFlag, int lowestIndex,
           int highestIndex);
void motor(float throttle, float brake, char hall_effects, char nextPhase);


/* ************************************************
                      Procedures
   ************************************************ */

//Flashes the LCD's enable
void flashEnable(){
    PORTA = 0x02;
    for(int i=0; i<800; i++);
    PORTA = 0x00;
}

//Flashes the LCD's enable for displaying text
void textEnable(){
    PORTA = 0b00000011;
    for(int i=0; i<800; i++);
    PORTA = 0x01;
}

//Converts a string into something the LCD can display
//TERMINATE WITH '~'!!!
void displayMessage(char s[]){
    int i=0;
    while(s[i] != '~'){
        PORTB = s[(i)];
        textEnable();
        i++;
```

```
    }
}

//Displays a single character on the LCD
void displayChar(char s){
        PORTB = s;
        textEnable();
}

//row 0 or 1; col 0->23 or 64-87 for row 1 (if 0 is used)
void setPosition(int row, int col){
    if (row>0)
        col+=64;
    col+=128;
    PORTB=col;
    flashEnable();
}

//Converts the A/D output into a voltage level between 0 and 5V
float analogToVoltage(word in){
    int anV = (int)in;
    float v = 5 * (float)anV / (256);
    return v;
}

//Converts a float into a character array
void numToChar(float in, char n[]){
    in = in * 1000;
    int next;
    for(int i = 1; i < 5; i++){
        next = in / 1000;
        in = (in - (next * 1000)) * 10;
        n[i] = (char) (next + 48);
    }
}

//converts the least significant place of an int to a character (0-9)
//eg: 53 returns "3"
char intToChar(int in){
    if (in<0)
        in=0-in;
    while (in>10)
        in-=10;
    return (char)(in+48);
}

//sets the demux to read from a certain battery
void selectBattery(int battery){
//0-15 on demux1, 16-29 on demux2
    if ((battery<0)||(battery>29))
        return;
    PORTB=0x00;
    PORTA=0x00;
    if (battery<16){
        PORTB=battery;
        PORTA=0b00001100;
    }else{
        PORTB=battery-16;
        PORTA=0b00000100;
    }
}

//Disables the demux
void disableDemux(){
    PORTA=0x00;
}
```

```
/* ************************************************
                        Main
   ************************************************ */

void main(void) {

    //Variable Definitions
    float rawVoltages[ARRAY_SIZE];       //Array for measured voltages
    float moduleVoltages[ARRAY_SIZE];    //Array for module voltages
    float tempVoltage = 0.0;             //Variable for subtraction
    int lowVFlagInit = 0;                //Initial low voltage flag
    int moduleStepper = 0;               //counter for "amy"
    float * totalPackV = &rawVoltages[ARRAY_SIZE-1];    //assigned address of
rawVoltages[29],
                                                        need *totalPackV to get the
value
    float totalCurrent = 0.0;
    int lowVFlag = 0;                //low voltage flag for while motor or batteries is
running
    int highCFlag = 0;                   //high current flag
    float throttle = 0.0;
    float brake = 0.0;
    //int hes1, hes2, hes3;   //hall effects are one variable
    //  hes1 = hes2 = hes3 = 0;
    char hall_effects = 0;
    char nextPhase = 0;
    int lowestIndex = 0;                 //index of the lowest V module
    int highestIndex = 0;                //index of the highest V module
    /*
        Variables we will need:
            phase outputs?
            addresses of I/O devices
                A/D (1 each for voltage read, current read, throttle, brake)
                latches (1 for each button)
                DeMUX control (6)
    */

    //global initialization of ports for I/O
    DDRB = 0xFF;
    DDRA = 0x1F;

    //LCD Initialization
    PORTA = 0x00;
    PORTB = 0b00110000;
    flashEnable();
    for(int i=0; i<800; i++);
    flashEnable();
    PORTB = 0b00111000;
    flashEnable();
    PORTB = 0x08;
    flashEnable();
    PORTB = 0x01;
    flashEnable();
    PORTB = 0x06;
    flashEnable();
    PORTB = 0b00001111;           //Display ON
    flashEnable();

    PORTB = 0b00000001;           //Return Home
    flashEnable();

    //Initializing arrays to 0.0
    for(int a = 0; a<ARRAY_SIZE; a++) {
        moduleVoltages[a] = 0.0;
        rawVoltages[a] = 0.0;
    }
```

```
    //Configuring A/D converter
    ATD0CTL2_ADPU=1;


    ATD0CTL3_S8C=1;
    ATD0CTL3_S4C=1;
    ATD0CTL3_S2C=1;
    ATD0CTL3_S1C=1;

    ATD0CTL4=0x85;

    ATD0CTL5_DJM=1;
    ATD0CTL5_DSGN=1;
    ATD0CTL5_MULT=1;
    ATD0CTL5_SCAN=1;

    do{
        //check status of throttle and brake
        throttle = THROTTLE;
        brake = BRAKE;
    }while(0/* throttle || !brake */);

    word rawV;

    for(int i=0; i<ARRAY_SIZE; i++){
        selectBattery(i);
        //Read A/D converter
        rawV = BATT_ATD;
        disableDemux();

        //Convert A/D value to analog voltage and store to rawVoltages
        rawVoltages[i] = analogToVoltage(rawV);

        moduleVoltages[i] = rawVoltages[i] - tempVoltage;
        tempVoltage = rawVoltages[i];
        if(moduleVoltages[i]<2.7) {
            lowVFlagInit=1;
        }
    }

    if(lowVFlagInit==1){
        //Display warning on LCD
            PORTB = 0b00000001;    //Return Home
            flashEnable();

            displayMessage("        WARNING!~");

            PORTB = 0b11000000;    //Set cursor to second line
            flashEnable();

            displayMessage("    VOLTAGE TOO LOW!~");

        for(;;);   //Do nothing
    }

    for(;;){       //always
        motor(throttle, brake, hall_effects, nextPhase);

        batteries(moduleStepper, totalCurrent, highCFlag, moduleVoltages,
            rawVoltages, tempVoltage, lowVFlag, lowestIndex, highestIndex);

    }

    EnableInterrupts;
}
```

```
/* ***********************************************
                        Batteries
   ********************************************** */

void batteries(int &moduleStepper, float &totalCurrent, int &highCFlag, float
moduleVoltages[],
               float rawVoltages[], float &tempVoltage, int &lowVFlag, int &lowestIndex,
               int &highestIndex)
{
    if(moduleStepper>=ARRAY_SIZE){
        moduleStepper=0;   //reset counter
        tempVoltage=0;     //reset subtractor

        //  Read A/D converter
        word rawC = CURR_ATD;
        //  Convert A/D value to analog current and store
        totalCurrent = analogToVoltage(rawC);

        //Test for total current too high
        if(totalCurrent>0/*too high*/)
            highCFlag=1;

        /*
        Use variable from motor procedure related to speed
        Make array of voltage vs. remaining capacity from discharge curves
        Use speed, current, and that array to calculate range

        Read each button/latch and change output values
        */

        //If current too high or voltage too low, send warning message to LCD
        if(highCFlag || lowVFlag){
            PORTB = 0b00000001;    //Return Home
            flashEnable();

            displayMessage("        WARNING!~");

            PORTB = 0b11000000;    //Set cursor to second line
            flashEnable();

            if (highCFlag)
                displayMessage("    CURRENT TOO HIGH~");
            if (lowVFlag)
                displayMessage("    VOLTAGE TOO LOW!~");
        }
        /*
        Update display
        */
    }else{
        selectBattery(moduleStepper);
        word rawV = BATT_ATD; //read ATD
        disableDemux();
        //  Convert A/D value to analog voltage and store to rawVoltages
        rawVoltages[moduleStepper] = analogToVoltage(rawV);

        moduleVoltages[moduleStepper] = rawVoltages[moduleStepper] - tempVoltage;
        tempVoltage = rawVoltages[moduleStepper];

        //  finding lowest and highest modules
        if(moduleVoltages[moduleStepper]<moduleVoltages[lowestIndex]){
            lowestIndex = moduleStepper;
        }else if(moduleVoltages[moduleStepper]>moduleVoltages[highestIndex]){
            highestIndex = moduleStepper;
        }

        if(moduleVoltages[moduleStepper]<2.7) {
            lowVFlag=1;
        }
        moduleStepper++;
    }
}
```

```
/* ***********************************************
                    Motor Control
   ********************************************** */

void motor(float &throttle, float &brake, char &hall_effects, char &nextPhase){

    throttle = THROTTLE;   //maybe use averaging algorithm from floattest.mcp
    brake = BRAKE;         //to keep this from being too jerky??
    hall_effects = HALL_EFFECTS>>5 && 0b00000111;

    //stolen from last year's project:
    if (hall_effects) { // some sensor is on
        switch (hall_effects) {
            case 0x01: nextPhase = 1; break;
            case 0x02: nextPhase = 2; break;
            case 0x04: nextPhase = 3; break;
            default: nextPhase = 0; break;
        }

        //need more here
    }
    /*
        *algorithm for control output*

        set power FET lines to correct value
        need to figure out delay based on throttle value and rest of program
    */
}
```

8.4 Monitoring Circuitry Diagrams

The following diagrams are included to help the reader understand the design of the monitoring circuitry a little more clearly. In each, only a few lines are depicted; however, one must understand that whatever is seen also exists for all thirty transistor/resistor/battery pack lines. Please see Section 2.1 for an explanation of these figures.

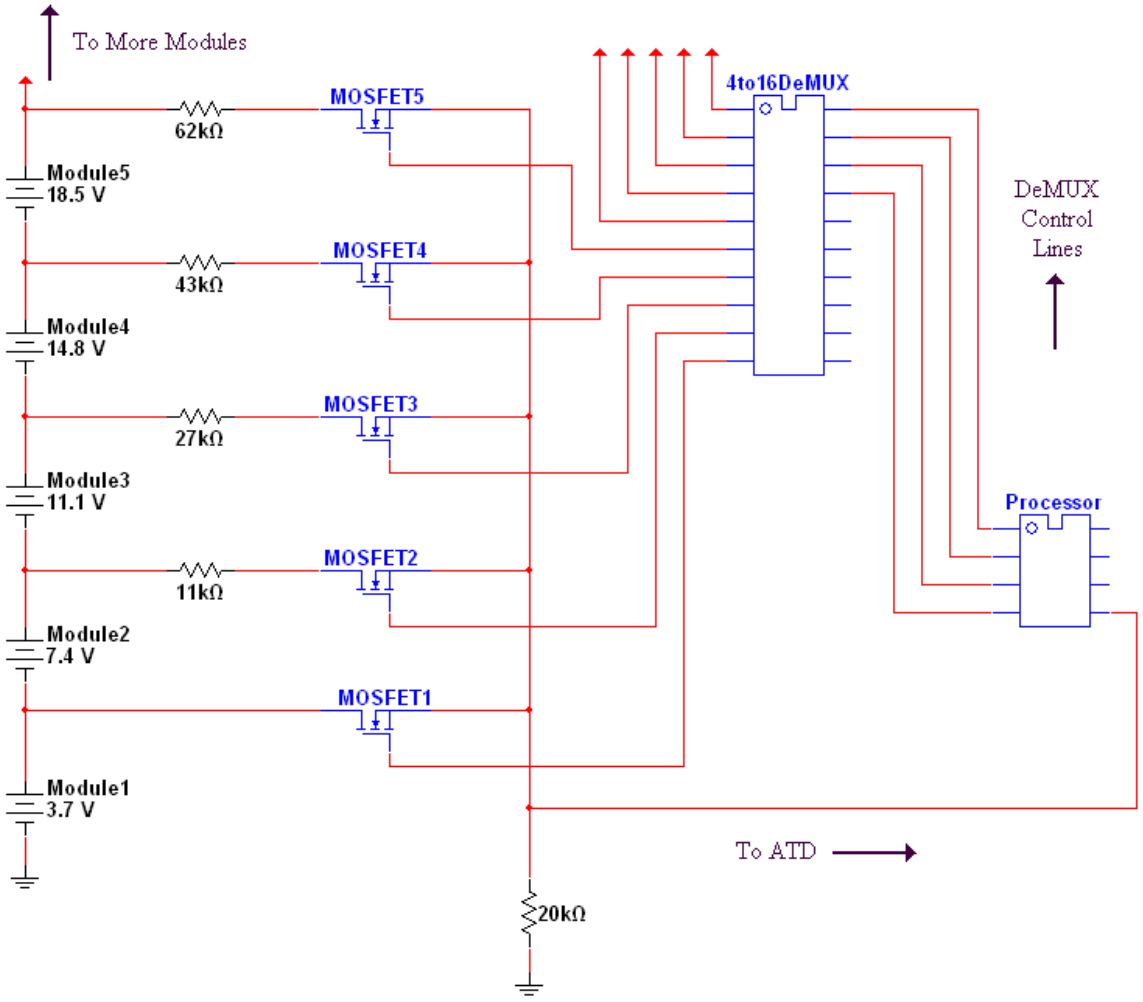


Figure 8.4.1 – Overall Monitoring Design

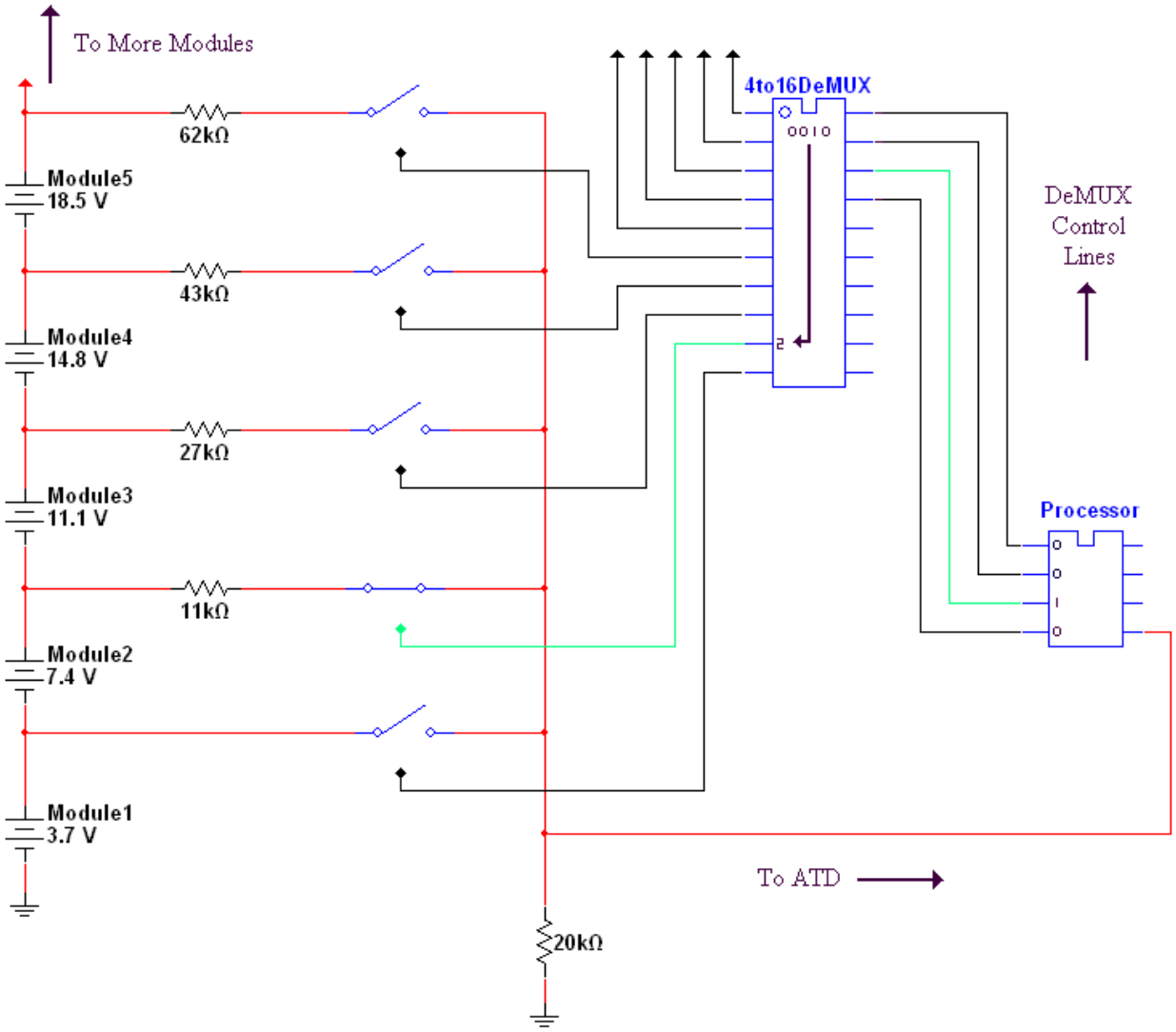
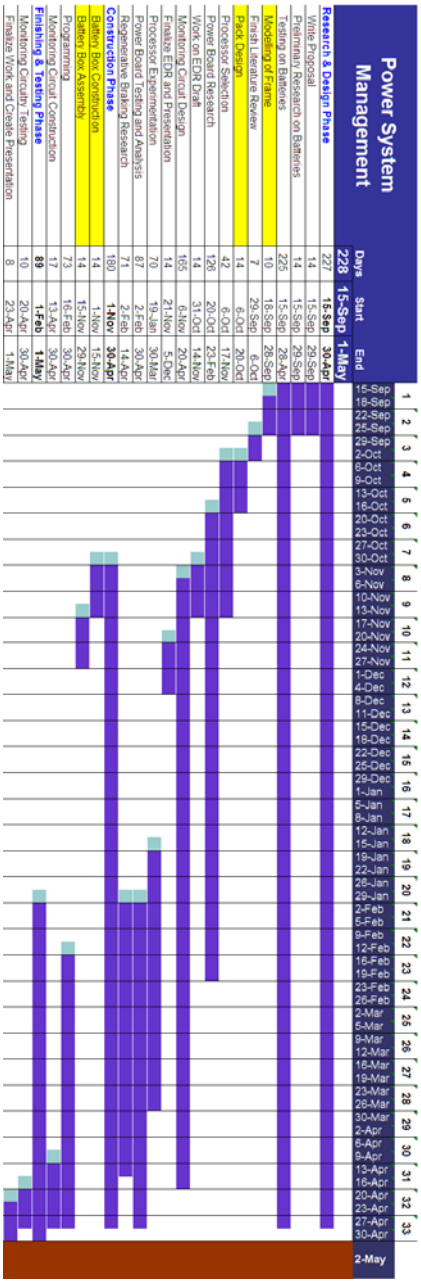


Figure 8.4.2 – Monitoring Explanation Using Switches

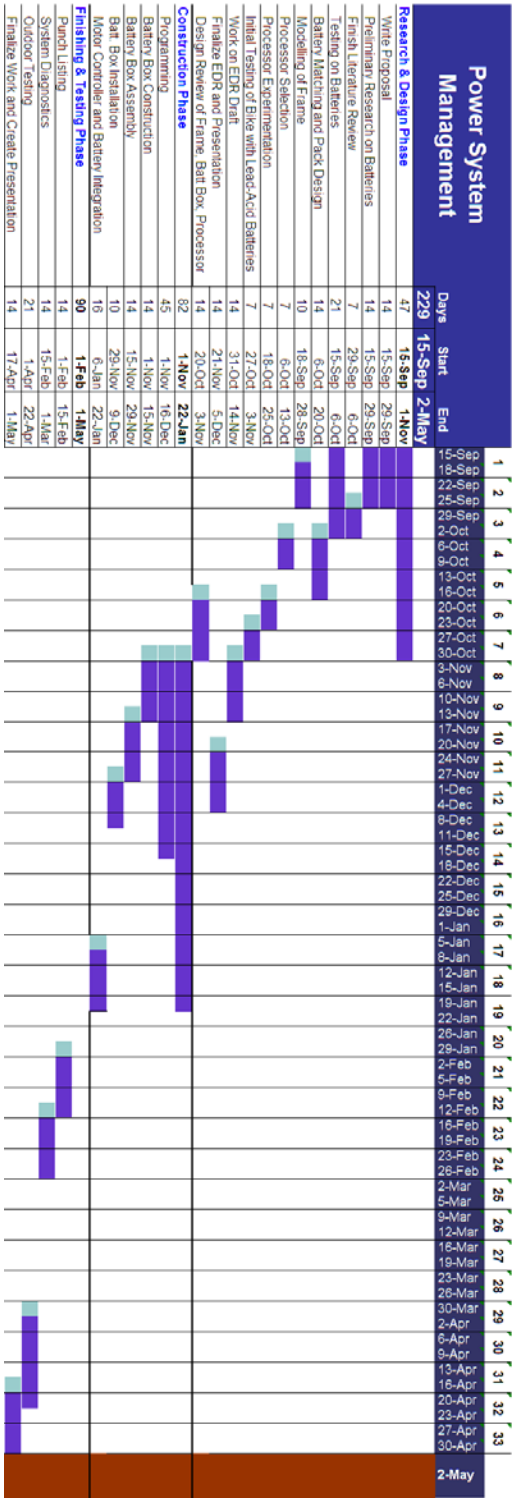
8.5 Gantt Chart

The following chart is a depiction of our actual task breakdown throughout the semester. Please refer to Section 4.0, Project Management, for a more detailed explanation of the reasons for the discrepancies between our actual Gantt chart, and the original one (show in Section 8.6).



8.6 Original Gantt Chart

The following chart is our original Gantt chart. We began planning this in the early Fall semester, and it was finalized by the beginning of November. As you can see, it differed drastically from the actual Gantt chart, show in the previous section.



8.7 Bibliography

The following is a listing of references that were used throughout the course of the project, followed by short annotations:

Battery Options and Electric Vehicles:

A123 Systems. Products and Applications. 8 Aug. 2008. 22 Sept. 2008. <http://www.a123systems.com/>. Battery data and general information.

Hussain, Iqbal. Electric and Hybrid Vehicles: Design Fundamentals. Washington, DC: CRC Press, 2003. Provided information regarding the different types of batteries currently on the market; used in the Literature Review.

Miller, John M. Propulsion Systems for Hybrid Vehicles. IEE Power and Energy Series. Stevenige: Institution of Electrical Engineers, 2004. Provided background information about the state of the art in our project area, as well as detailed pros and cons of the different types of batteries.

Maxim Microcontrollers (a possibility before deciding on the microprocessor):

www.maxim-ic.com/solutions/battery_management/index.mvp

-Battery Management Overview, Main Site

www.maxim-ic.com/solutions/battery_management/app_notes.mvp?pl_pk=9&go_btn.x=9&go_btn.y=15&go_btn=submit

-Application Notes for Power and Battery Management

www.maxim-ic.com/solutions/battery_management/parts.mvp/scpk/1224/pl_pk/0

-Data Sheets for Fuel Gauges

www.pdfserv.maxim-ic.com/en/ds/DS2746.pdf

-Data Sheet for a Low-Cost 2-Wire Battery Monitor

www.pdfserv.maxim-ic.com/en/ds/DS2756.pdf

-Data Sheet for a High-Accuracy Battery Fuel Gauge

www.pdfserv.maxim-ic.com/en/ds/DS2745.pdf

-Data Sheet for a Low-Cost I²C Battery Monitor

www.maxim-ic.com/appnotes.cfm/an_pk/189

-Application Notes for an Advanced Rechargeable Li-Battery Pack

Processor Research:

www.microchip.com

-Reference for the PIC we considered

www.freescale.com

-Reference for the processors from Freescale Semiconductor, as well as our source for downloading CodeWarrior

http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=HCS12CSLK&fsrch=1

-The 9S12C128 processor (the one we did not select)

http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=HCS12DT256SLK&parentCode=null&nodeId=0162469544

-The 9S12DT256 processor (our final choice)

www.ee.nmt.edu/~erives/308_08/Lecture20.pdf

-HCS12 analog to digital converter set-up explanation

Programming Help:

www.Freescale.com/CodeWarrior

-CodeWarrior software and support

www.cplusplus.com/reference

-General information, documentation, references, articles, source code, and help forums for C++

Datasheets and Part Lookups

www.mouser.com

-Ordering parts and part descriptions

www.alldatasheet.com

-Part lookups and descriptions

www.hantronix.com

-LCD information, selection, and ordering

Regenerative Braking:

http://archive.chipcenter.com/eexpert/dashby/dashby054.html

-Provided theoretical ideas for brushless regenerative braking designs