Streets Ahead - Thunderstruck the Meter

Hieu Le | Jacob Clatterbuck | Kristian Johnson

12/10/2020

Capstone Design ECE 4440 / ECE4991

Signatures

J

acob Clatterbuck

Hieu Le

Kristian Johnson

Statement of work:

Jake:

The core job of team member Jacob Clatterbuck was to design and implement the software used in the workings of the DC panel meter. This includes the i2c communication between the ADC and MSP430 and the I2C communication with the 16x2 LCD, along with software filters and value conversions. This allowed the device to perform its primary task of displaying the voltage and current being drawn from the power supply to the user. He also assisted in the construction and assembly of the soldered PCB and metal enclosure.

Jacob Clatterbuck also was in charge of the creating software test plan and testing. Like the other members, he contributed to the creation of all major document submissions such as the proposal and this report. Another responsibility he was in charge of was the scribing of notes for the weekly meetings with the professors. Finally, he filmed and narrated the video demonstration of the final product.

Hieu:

Hieu's main responsibilities are circuit design and calculation. He came up with the overall design of the project and researched all of the components needed for the design. All of the components datasheets were reviewed thoroughly to make sure they meet the requirement. He used the software kiCad to design the schematic and PCB. He used Multisim to come up with different protection designs, simulate the protection circuits, and tested its maximum functionality.

Hieu also worked with Kristian to come up with a PCB test plan. He designed the PCB to have different isolations and easy to test. The test plan we came up with allowed us to safely test each section of the circuit safely without worrying about Ics getting destroyed because the previous Ics had malfunctions. Hieu also assisted during the actual testing and debugging process.

Kristian:

Kristian was responsible for the assembly of the project. This included the soldering of all of the components, as well as creating an enclosure to house the device. Although the enclosure was a purchased aluminum box, Kristian drilled holes and cut out areas such that the LCD, switch, and wire leads could access the PCB within the aluminum box. Kristian was also responsible for selecting many of the components and preparing the weekly DigiKey order forms. Kristian and Hieu worked together to create a test plan for the PCB. As Hieu mentioned above, our test plan allowed verification of functionality in each section of the PCB design. As for all of our team members, Kristian contributed to all of the writing assignments - such as the proposals, midterm design review, and final report.

Table of Contents

Contents

Capstone Design ECE 4440 / ECE4991 1
Signatures 1
Statement of work: 2
Table of Contents 4
Table of Figures 5
Abstract 6
Background 6
Design Constraints 7
External Standards 9
Tools Employed 11
Ethical, Social, and Economic Concerns 12
Intellectual Property Issues 14
Detailed Technical Description of Project 15
Project Timeline 27
Test Plan 30
Final Results 37
Costs 39
Future Work 42
References 43
Appendix 44

Table of Figures

Figure 1: Design Overview 17

Figure 2: 50V Product Overview 18

Figure 3: >50V Product Overview 19

Figure 4: Main circuit schematic 20

Figure 5: Protection circuit 21

Figure 6: 5V and 3.3V buck converter circuit 22

Figure 7: PCB layout 23

Figure 8: PCB filled 24

Figure 9: 3D view of PCB 25

Figure 10: Soldered PCB 26

Figure 11: Original Timeline 28

Figure 12: Final Timeline 29

Figure 13: ADC Test Plan 30

Figure 14: LCD test plan 31

Figure 15: Filter test plan 32

Figure 16: PCB Test Plan 33

Figure 17: Buck Converter Test Plan 35

Figure 18: MSP Test Plan 36

Figure 19: Spy Bi-Wire Test Plan 36

Figure 20: Final enclosed prototype 39

Figure 21: First Parts Order 40

Figure 22: Second Parts Order 41

Figure 23: Third Parts Order 41

Abstract

The goal of this project, Thunderstruck the Meter, is to create an inexpensive DC Panel Meter that combines both the functionality of a voltmeter and a current meter for use alongside an existing power supply. The maximum values that the meter is capable of measuring are 100V and 50A. To accomplish this, a voltage divider and current sensor IC will be used in the design. In addition, the product will also feature an MSP430 microcontroller and a 16bit Analog to Digital Converter (ADC), which can measure up to 187.5uV. This allows the voltage and current from the power supply to be shown on a 2x16 Liquid Crystal Display (LCD). The device will also be designed with reverse polarity protection, short circuit protection, and overcurrent protection.

Background

While there are existing DC Panel Meters on the market, they can be extremely expensive, ranging anywhere from one hundred dollars up to several hundred dollars. The available options either include unnecessary features or are poorly designed and provide low accuracy with no protection. Our project aims to stay in the middle, taking into consideration both an attractive price point as well as well thought out functionality. We want to design a DC panel meter that can read current and voltage down to 1mA and 10mV resolution, while still providing the necessary reverse polarity, short circuit, and overcurrent protection. The device will also have the option of using an external power supply if the main power supply is over 50V. Our final product aims to be small, compact, and accessible to any users with very little experience regarding electrical systems. Using any existing power supply already at hand, the user can turn it into a more useful power supply bench. Because the device has reverse polarity, short circuit, and overcurrent protection, it can also act as a test circuit for any power supply source below 100V.

Several of the lessons learned in previous coursework will come into play for this project. From ECE-3430 (Introduction to Embedded Systems), our group learned valuable programming skills and how to program the microcontroller. ECE-3430 will also be useful in learning how to program the LCD on the DC Panel Meter. The Fundamentals of Circuits classes, ECE 2630, 2660, and 3750 are relevant for the regulation of the power supply voltage to the microcontroller and display. Two of our members took an Electromagnetic Energy Conversion course (ECE-3250) last semester and learned how to correctly measure voltage, current, and power. One of the challenges for designing a DC Panel Meter relates to carefully choosing the correct IC components that satisfy the voltage, current, and power requirements. The knowledge and experience gained from courses like Microelectronic Integrated Circuit Fabrication (ECE-5150), Solid State Devices (ECE-3103), and Introduction to Materials Science (MSE-2090) will be very useful in selecting the correct components.

Constraints

Design Constraints

The first design constraint is about the current limiting protection. We use a P-channel MOSFET, a PNP BJT, and a shunt resistor. Depending on the shunt resistor value, the current flowing through the MOSFET will be set a certain limit. The BJT will act as a current sensor loop, when the current is at maximum setting, it will slowly shut down the gate hence limiting the current from flowing. This protection design relies heavily on the capability of the P-channel MOSFET. The MOSFET we used is the BSS84LT1G which has the drive voltage at 5V. This limits our device minimum voltage input to 5V. The power dissipation of the MOSFET is about 250mW which directly affects the amount of current that can flow through it. If the Voltage input is high the maximum current needs to be decreased to keep the device from heating up. We got around this problem by carefully calculating the total amount consumed by the device at maximum operation mode which is under 150mW. We then set the shunt resistor to allow a maximum of 200mW which has it well under 250mW power dissipation limit of the MOSFET.

The accuracy of the current measuring is constrained by the current sensor Ic, ACS758. The sensor has the sensitivity of 40mV/A. To solve this problem, we need an ADC that could measure accurately the current at .4u V solution which is about 16 bits. We chose the 16bits ADS1115. We need to balance between the speed and the accuracy of the 16 bits Delta Sigma (ΔΣ) ADC. The ADC we used is the ADS1115 has a programmable data rate at 8 SPS to 860 SPS. ΔΣ analog-to-digital converters (ADCs) are based on the principle of oversampling. The input signal of a ΔΣ ADC is sampled at a high frequency (modulator frequency) and subsequently filtered and decimated in the digital domain to yield a conversion result at the respective output data rate. The ratio between modulator frequency and output data rate is called oversampling ratio (OSR). By increasing the OSR, and thus reducing the output data rate, the noise performance of the ADC can be optimized. In other words, the input referred noise drops when reducing the output data rate because more samples of the internal modulator are averaged to yield one conversion result. Increasing the gain also reduces the input-referred noise, which is particularly useful when measuring low-level signals.

One of the other constraints were delays when ordering any parts through any vendor similar to the ordering process from Digikey. There was the delay for manufacturing each PCB that was ordered. These delays usually took one or two weeks and were extremely disruptive to the design process.

Economic and Cost Constraints

The main limitation of our budget was the strict $500 limit. We were also limited to a certain number of PCB board sendouts and therefore had to make sure that each submission was not wasteful and was either functional entirely or usable for our purposes. In the unique circumstances of a semester in quarantine, our ability to pick up packages was hindered and all components had to be ordered and delivered, increasing the costs through shipping prices. While our prototype used components in low numbers, mass production would allow us to access bulk prices from vendors.

External Standards

IPC-2221[1] is generally accepted in the electronic industry as a generic PCB design standard. However, when it comes to distances between the PC traces, in my view, the IPC-2221 table 6-1 stepwise limits are mostly baseless: the curve for spacing vs. voltage should be linear. Of course, it is not the only standard that defines the electrical clearance. For power conversion circuits IPC-9592 initial draft provided the following linear circuit board spacing recommendations: SPACING (mm) = 0.6 + Vpeak x .005. In general, a linear relationship makes more sense. However, these requirements were also too conservative and at low voltages were not even practically doable. The updated IPC-

9592B document left the above equation only for $V \geq 100V$. At other voltages the limits are as following: 0.13mm for $V<15V$, 0.25mm for $15V \leq V<30V$ and $0.1+Vpeak \times 0.01$ for $30V \leq V<100V$. Note that all IPC standards are voluntary rather than mandatory. Conversely, for the products covered by safety standards the creepage and clearance requirements of a respective UL/IEC standard are mandatory.

The 16x2 standard alphanumeric LCD displays[2] are extremely common and is a fast way to have your project show status messages. An LCD (Liquid Crystal Display) screen is an electronic display module and has a wide range of applications. A 16x2 LCD display is a very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in a 5x7 pixel matrix. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. This LCD has two registers, namely, Command and Data. Command register stores various commands given to the display. Data register stores data to be displayed. The process of controlling the display involves putting the data that form the image of what you want to display into the data registers, then putting instructions in the instruction register.

The embedded memory of MSP430 microcontrollers (MCUs) can be programmed using the on-chip JTAG interface. The MSP430 MCUs support 2-wire JTAG interface. The 2-wire JTAG interface is referred to as Spy-Bi-Wire (SBW)[3]. Using Spy-Bi-Wire a host can access the programmable memory (flash memory), the data memory (RAM), and in FRAM devices, the nonvolatile FRAM memory. The host MCU can access the memory of the target MSP430 MCU during the prototyping phase, final production, and in service (field software updates). This application report uses the SimpleLink™ MSP432P401R, CC3220, and CC2640R2F devices as the hosts for the SBW communication with the target MSP430 MCU. Both flash-based and FRAM-based MSP430 MCUs are used in this report to showcase the differences between the two device families, related to SBW communication. Software examples are provided for each of the SimpleLink devices mentioned. The software examples also make use of the SimpleLink Software Development Kit (SDK), making it easy to port to other SimpleLink devices.

Tools Employed

All of the materials needed for assembly and testing can be found within the ECE lab, aside from a few power tools used for constructing the enclosure The software tools that we utilized during the design of this project included: National Instruments MultiSim[4] for designing the schematic, National Instruments UltiBoard[5] for creating the PCB, Code Composer Studio[6] for programming the MSP430, and VirtualBench[7] for testing the PCB. The PCB assembly required a soldering iron, solder, a copper wick, and a flux pen. The enclosure assembly required the use of several tools including, but not limited to a power drill, a milling machine, a dremel, sandpaper, files, needle-nose pliers, and a hot glue gun. All members of our team have had some experience with all of the software tools that were utilized in the making of this project. It was Kristian's first time using a milling machine for cutting out sections in the aluminum enclosure - he hopes to apply this new knowledge to future projects. One of the biggest learning opportunities for our group was working together remotely. It was a bit of a challenge at times with each member in a different geographical location and attempting to work on the same project. Our communication skills improved in working with one another over Zoom calls and GroupMe messages. In addition, Kristian and Hieu first met Jake at the beginning of this semester - allowing each of us the opportunity to learn how to work together with team members that we may not know well.

Ethical, Social, and Economic Concerns

Environmental Impact

The design of our prototype does not impose any direct impact to the environment. However, the manufacturing of some of the components could potentially release harmful chemicals into the environment. This could result in negative effects to the workers, local wildlife, local bodies of water, and local communities. In addition, if there was a complication with the DC meter that resulted in the PCB heating up and catching fire, the PCB would generate dangerous by-products. Lastly, as with the majority of electrical designs, the disposal of the product is not very environmentally friendly.

Sustainability

Despite the lower cost, the DC Panel Meter's design still incorporates protection against short circuits, reverse polarity, and too much current - providing a safe device for the user. The device is designed to be paired with an existing power supply, eliminating the need for purchasing a new and more expensive power supply that has these filters built-in.

Health and Safety

Several industry standards are relevant for the development and design of this product. Foremost in our minds is safety standards as we will be dealing with power supplies capable of high voltage and amperage. OSHA standards for controlling electrical hazards[8] are particularly relevant here. These standards define ways to avoid unexpected equipment startup, how to tell when an electric shock is dangerous, and what level of protective equipment is appropriate. We will be using these guidelines to determine the material our enclosure is made of, making sure it's insulating. As the final product will be held within an enclosure, NEMA standards[9] will be used to express to the user the level of protection the enclosure offers from dust, water, and condensation. Our product's enclosure is of NEMA Type 1 as its primary goal is to prevent contact with live parts and provide general protection from dust and splashing water. As the device will be primarily used indoors, weather protection is unnecessary and it does not need to be watertight.

Standards relating to the design of the PCB are important for ensuring that it can handle the high current and voltage requirements as well as avoiding common mistakes such as acid traps. IPC standards[10] such as defining acceptable jobs, chip mounting, and trace angles will be implemented. Any acute angles in the traces will be avoided as the corners created by these angles provide an area for acid to accumulate and remain on the board. Chips mounted on the PCB will need to lie generally flat and not have one side unevenly spaced from the board. The soldering of components will be inspected for coverage and to ensure a stable electrical connection.

Manufacturability

The DC Panel Meter project is designed and assembled using widely available components purchased from retailers such as DigiKey[11], Mouser[12], and Newark[13].

DigiKey will be our primary source in order to reduce shipping and ordering costs. However, one of the disadvantages is the product availability. Digikey does offer a wide variety of quality products to choose from; however, there are many specialized products that are not offered.

The PCB is designed to allow for ease in soldering the components and debugging the circuitry. For example, multiple test points are present on the PCB to allow space for several probes to be attached. The most expensive components required for this project are the sensor chips, as well as the PCB, LCD, aluminum enclosure, and MSP430 Microcontroller. Development boards are also included within the prototype budget; however, these are not required for the final product. The final cost of the prototype will be well within the $500 budget - and the production cost will be much less than the prototype cost; allowing our design to provide an inexpensive alternative to the public when compared to the existing costly competition. The goal is to keep the cost as low as possible, while maintaining the highest quality possible within the budget. With an unlimited budget, our design would have the potential to use higher end parts and be more sustainable and environmentally friendly; however, with the set prototype budget of $500, we are limited from using more expensive cutting-edge technology for our design.

Ethical Issues

While this design choice does eliminate waste by allowing the user to utilize a preexisting power supply to create a lab bench - making the design more sustainable and environmentally sound - it does create an ethical issue in the form of removing business and jobs from manufacturing companies that are creating newer products with these built-in features.

Intellectual Property Issues

Our prototype design is similar to a digital multimeter, but with less features in order to have a more desirable price point. Our design connects to an existing power supply. The design of "Digital multimeters including a remote display"[14] has a feature allowing the head of the device to be detachable from the body of the device. It also has a digital screen that displays the measured parameters. Our design is able to measure voltages up to 100 Volts (relative to ground) similar to the "Voltage measurement"[15] patent from 2015. The design is also able to measure current up to 10 Amps similar to the "Current measurement"[16] patent from 2020. While our DC Panel Meter prototype is unique and patentable, the existing multimeters used by companies would own the market for these devices. Our target market would be individuals who already have an existing power supply lying around, but we don't think that would provide enough sales in order to allow for the manufacturing of our device.

Detailed Technical Description of Project

The project is an inexpensive DC panel meter that combines both the functionality of a voltmeter and a current meter for use with an existing power supply. The maximum values that the meter is capable of measuring are 100V and 50A. The panel can read current and voltage down to 1mA and 10mV resolution, respectively. The device is designed to take a very small amount of power from the power supply that it is measuring in order to power itself up to 50V. If the voltage of the power supply exceeds 50V, we also have an option of using an external power supply using a single pole double throw (SPDT). A buck converter is used to reduce the voltage from the supply to the 3.3V needed for the microcontroller and other powered elements. The device is also designed with reverse voltage polarity, short circuit, and overcurrent protection which increases both the testing safety and the lifespan of the product.

The device operates in two modes: under 50V power supply mode, and between 50V and 100V power supply mode. The user can switch between the two modes by using the SPDT switch. In the under 50V power supply mode, the device uses the power supply that it is measuring to power itself. The current goes through a PMOS diode current feedback loop which is used as an overcurrent protection. We use another PMOS diode to block out the current if voltage polarity is reversed. If nothing is shorted and the voltage polarity is correct, the current can flow through the 3.3V switching buck converter. The buck converter is rated as a 50V input such that any voltage below 50V is completely safe. The advantage of using the switching buck converter is that it has an amazing efficiency reaching up to 90%. The 3.3V buck converter is the main power source for all of the ICs on the circuit, including the MSP430, the LCD, the ADC, and the current sensor (as shown below in Figure 1). The device uses a voltage divider to sense the voltage supply, and uses a current sensor IC (ACS758) to sense the current. The voltage and current readings will go to a 16 bit Analog to Digital Converter (ADC) (ADS 1115) which has 65k steps. The ADC allows us to measure the signal as low as 187.5uV. The ADC sends the signal to the MSP430 to process using the I2C communication protocol. After processing the signals, the MSP430 will display the reading on a 2x16 LCD. The device will resemble the design in Figure 2 with the switch on <50V mode.

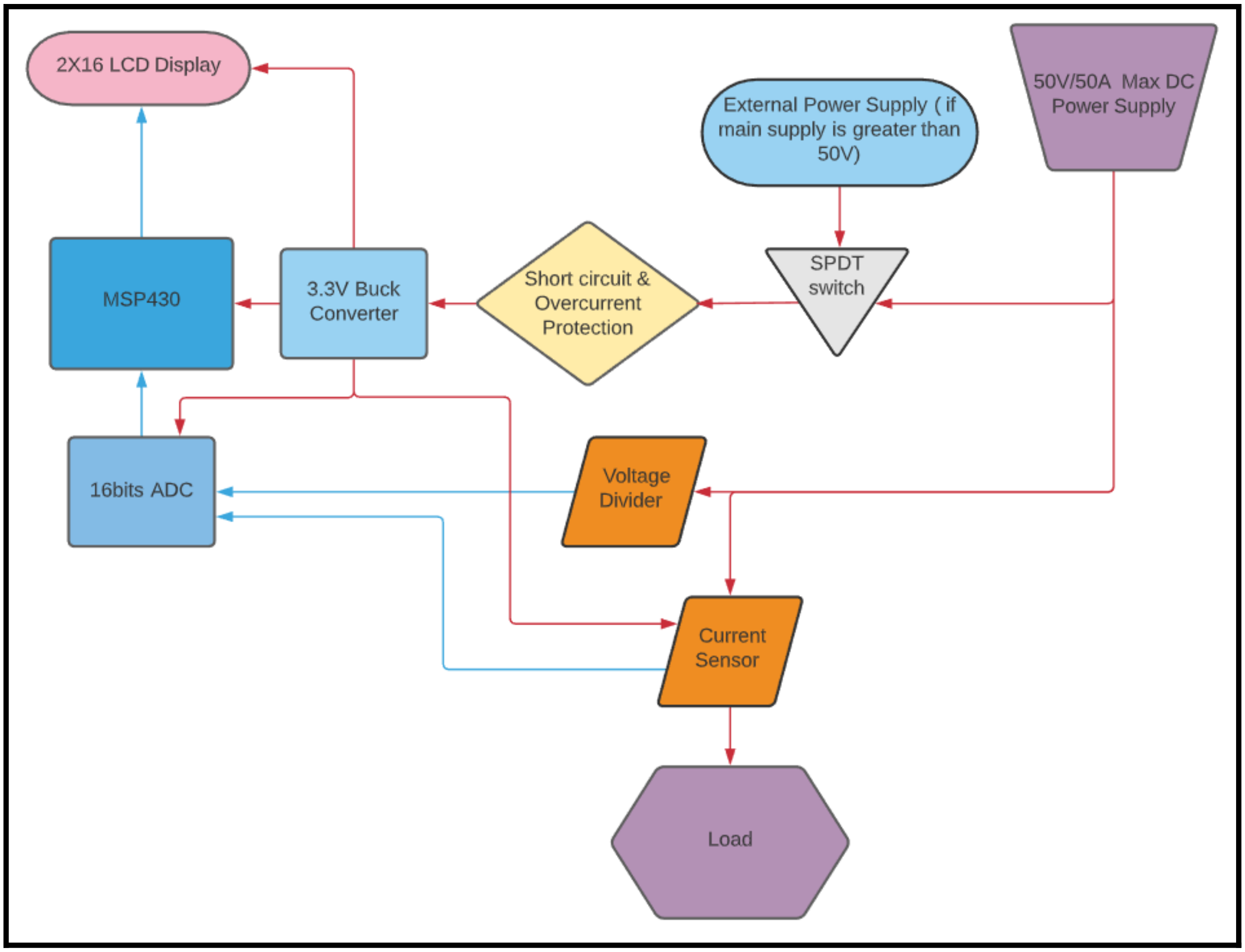

Figure 1: Design Overview

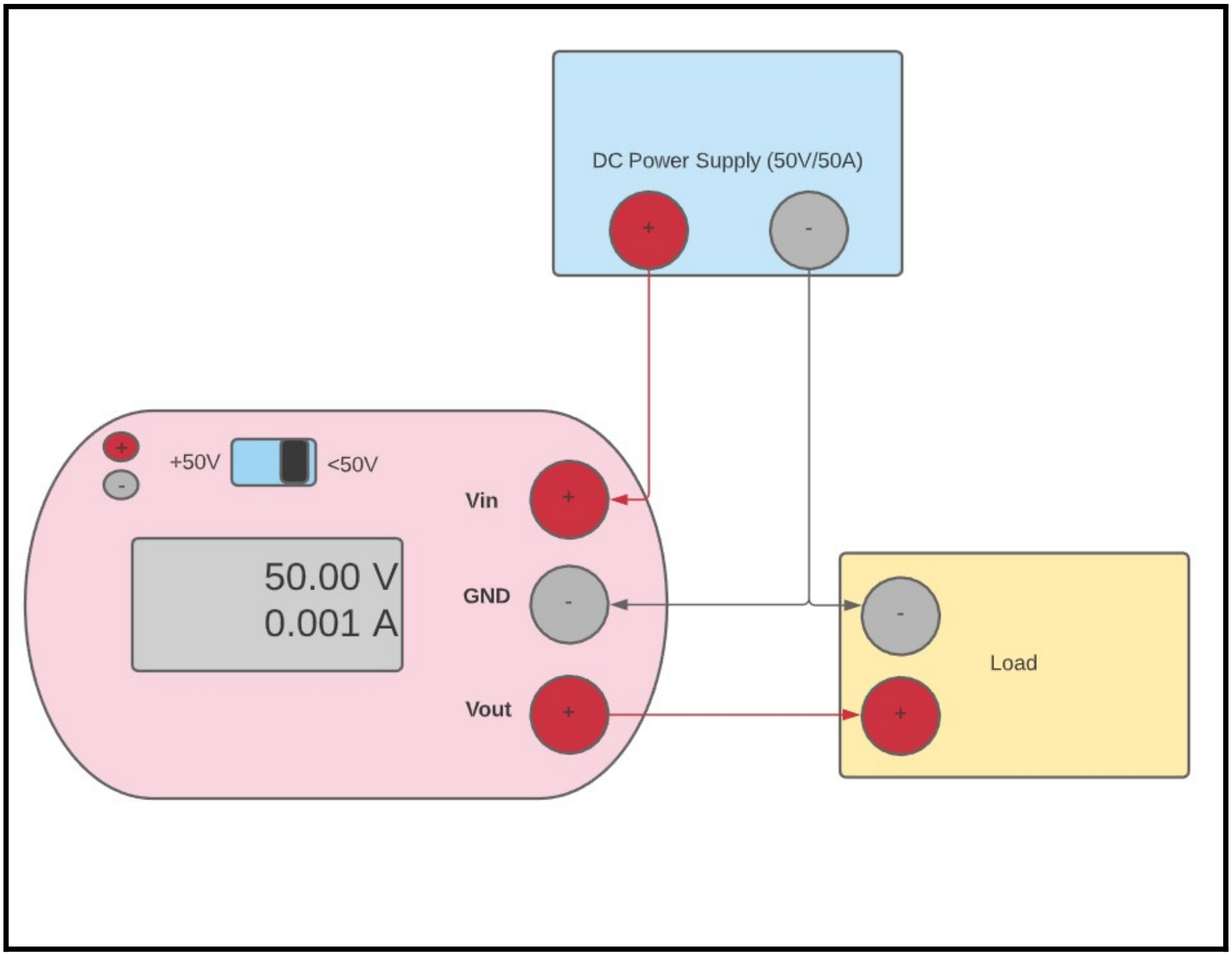

Figure 2: 50V Product Overview

In the under 50V power supply mode, the device needs to use an external power supply to supply its power. The reason for this is that the buck converter can only handle up to 50V. Anything over that will cause serious damage to the device. Any DC power supply source over 5V will suffice. The device operating in this mode is shown in Figure 3 below.

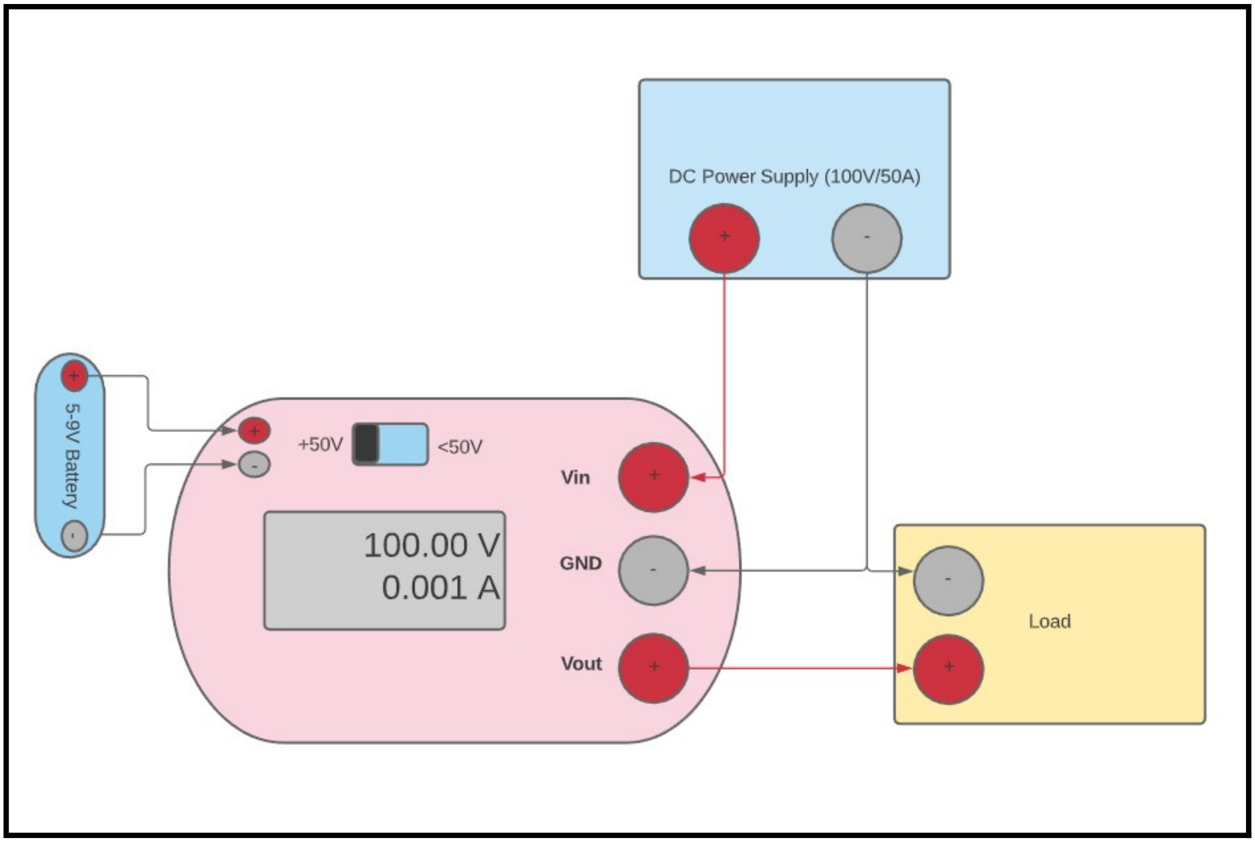

Figure 3: >50V Product Overview

One anticipated problem that comes to mind is having the product destroyed due to user error in not having the SPDT switch in the correct location. If more than 50V is applied, and the switch is set such that less than 50V should be applied, the device will likely be destroyed. This issue could be solved by installing an automatic switch that determines which setup should be used - depending on the amount of voltage being applied. However, an automatic switch increases the cost of the design, along with having the potential to choose the incorrect setup when the voltage is near the threshold of the two setups. We can solve this problem by adding an overvoltage protection circuit using the PNP transistors and a Zener diode. We simulated the combined circuit within MultiSim and confirmed the whole circuit reacted well together.

The figure below, Figure 4, is an overview of our schematic. We wanted to have a clear view of each section of the design to make it easy for non-expert view and easy to debug. It includes power supply block, voltage and current sensor blocks, filters, ADC, MSP430, and LCD block.

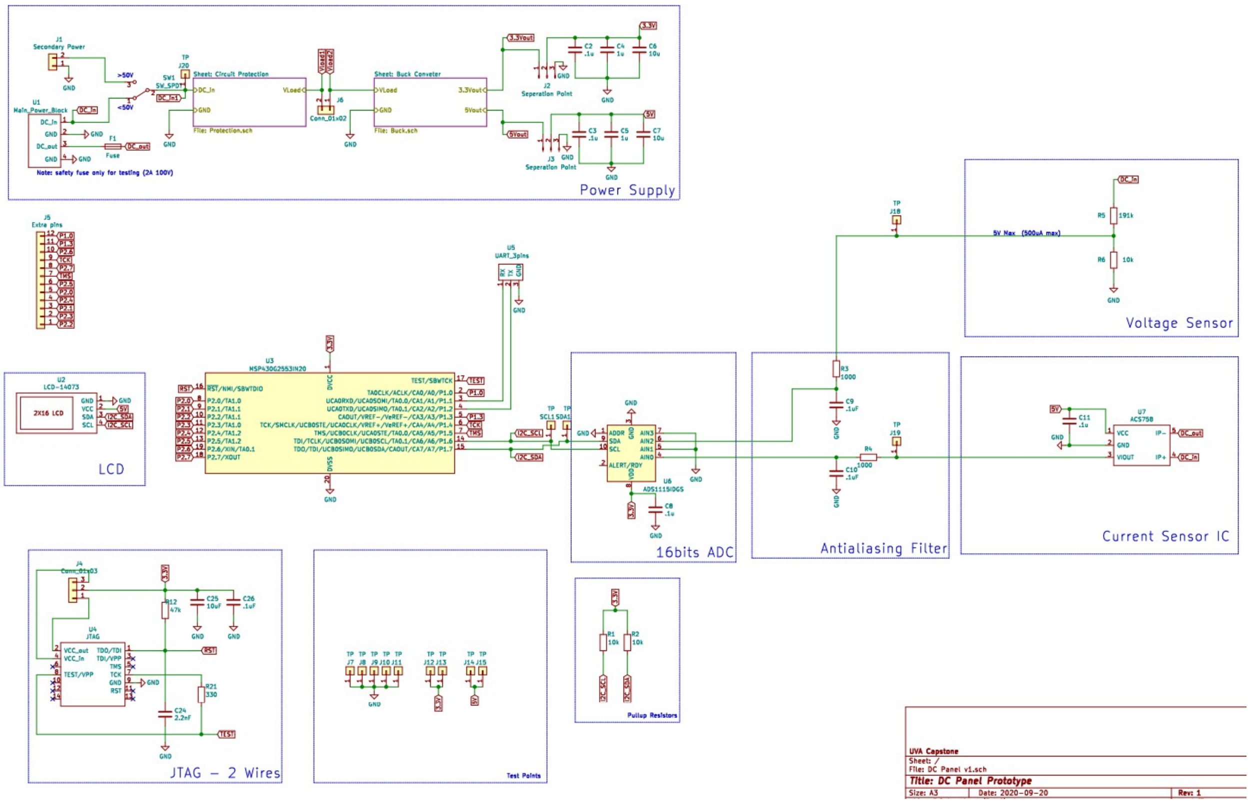

Figure 4: Main circuit schematic

The circuit below, Figure 5, is our circuit protection design. We think that it's important to add well thought out circuit protections. The circuit protection design includes reverse polarity protection, overvoltage protection, overcurrent, and short circuit protection. Using the advantage of the PNP BJT, if the user accidentally reverses the voltage input polarity, it will block all the current flowing since there is a positive voltage at the base. Left half of the circuit, we are using 2 PNP BJTs and adding a Zener Diode to make an overvoltage protection. Basically, when the supply voltage goes over 50V, the 50V Zener diode will be at reverse voltage breakdown and allows current to flow through hence causing a voltage drop at the base of the BJT hence blocking the current from flowing. To add a short circuit and current limiting protection, We use a P-channel MOSFET, a PNP BJT, and a shunt resistor. Depending on the shunt resistor value, the current flowing through the MOSFET will be set a certain limit. The BJT will act as a current sensor loop, when the current is at maximum setting, it will slowly shut down the gate hence limiting the current from flowing.

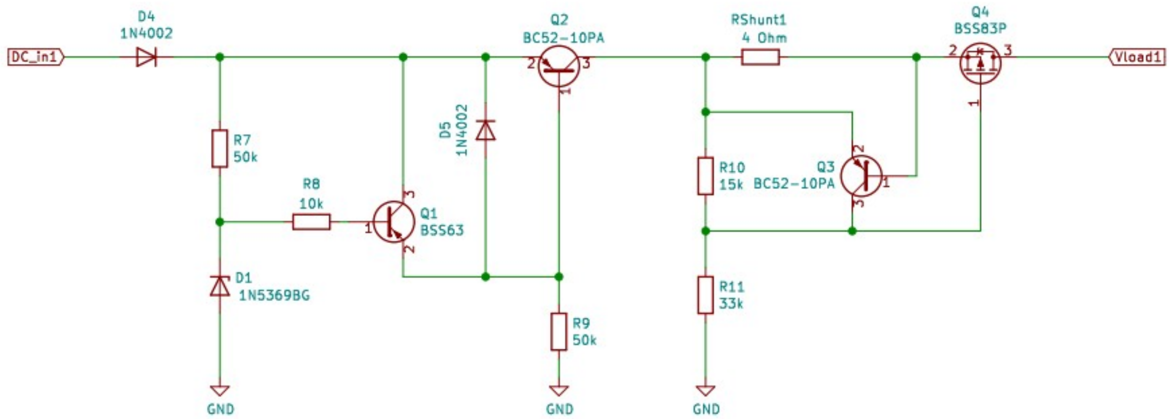

Figure 5: Protection circuit

The circuit below, Figure 6, is the buck converter circuit. The original plan is to have all the components that only use 3.3V and only have one buck converter to save cost. However, the accuracy of the current sensor, ACS758, is based on the voltage reference supply, it requires a steady 5V power supply. Though this is a setback for our plan of keeping the components cost low, however, there are other advantages of having two buck converters such as wider components availability and reducing the heat generated by only using one buck converter.

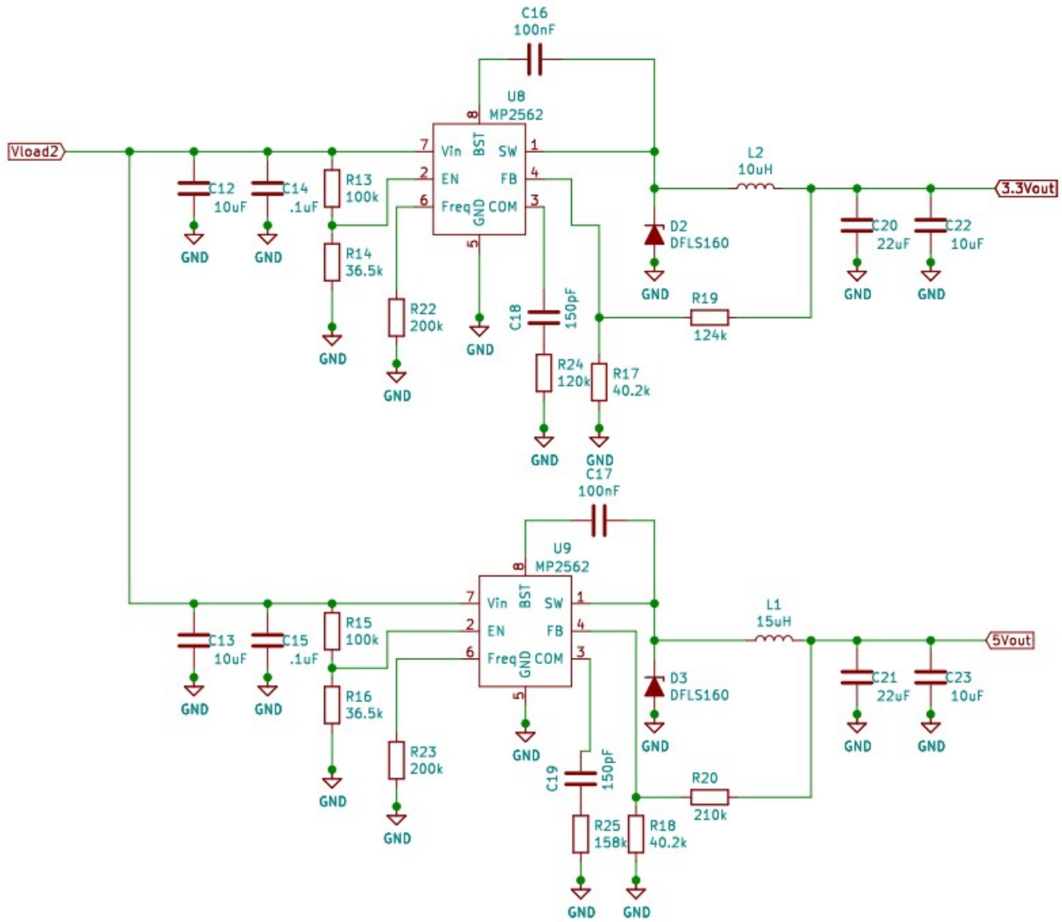

Figure 6: 5V and 3.3V buck converter circuit

The three figures below, Figure 7, 8 and 9, are our PCB layout, PCP layout with Ground filled, and 3D view of PCB respectively. One of the first things we did was paying close attention to the arrangement of all the components, to be more specific, components distance and order of the components. We followed as close to the arrangement of components in the schematic as possible. This way we can easily find the components associated with each other. It was a huge time saver when we needed to debug an IC and its supported components. We tried to keep the distance of each component at least about 2mm apart. This allowed us to easily solder in every single component with ease; and if we needed to desolder any, there was enough operating room to do so.

The flow of the circuit is started from the top left to the bottom left in clockwise motion. We designed it this way to visually separate and make it easy to interpret and debug. It is often overlooked how useful it is to design the PCB layout in the flow of either power supply or signal in/out, especially when prototyping a product. Like mentioned above, we followed the arrangement of components in the schematic and its flow. During the testing process, we were able to have a peace of mind on which section of the board was already tested and working, we could move on to the next section according to the flow of the schematic.

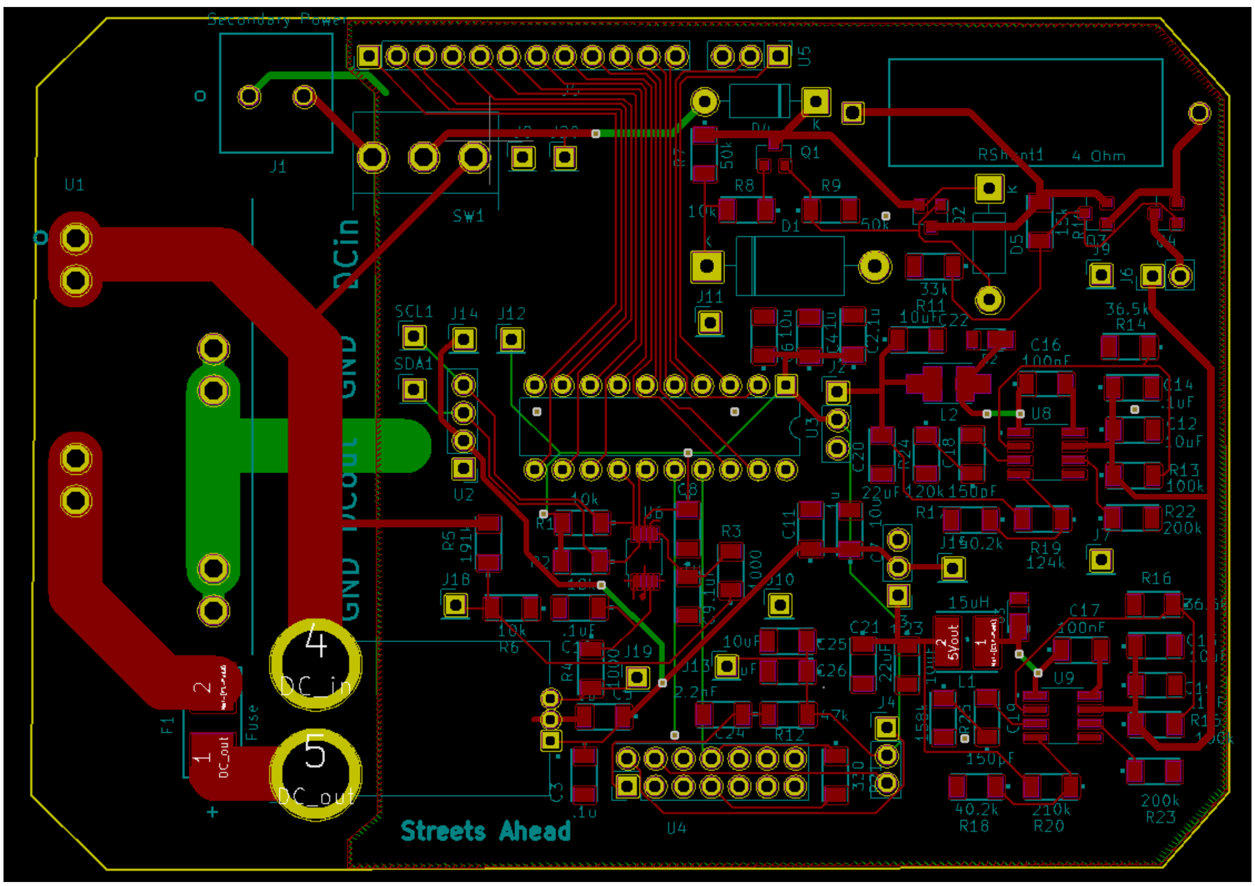

Figure 7: PCB layout

We intentionally left the ground copper, both top and bottom, on the left unfilled. This gave the device more isolation room where there was high voltage and current flow located. This is the necessary precaution because the device might need to measure the unregulated power supply and there is a chance it would have a voltage surge. The necessary distance can prevent any surged voltage from "jumping out" and destroy nearby components.

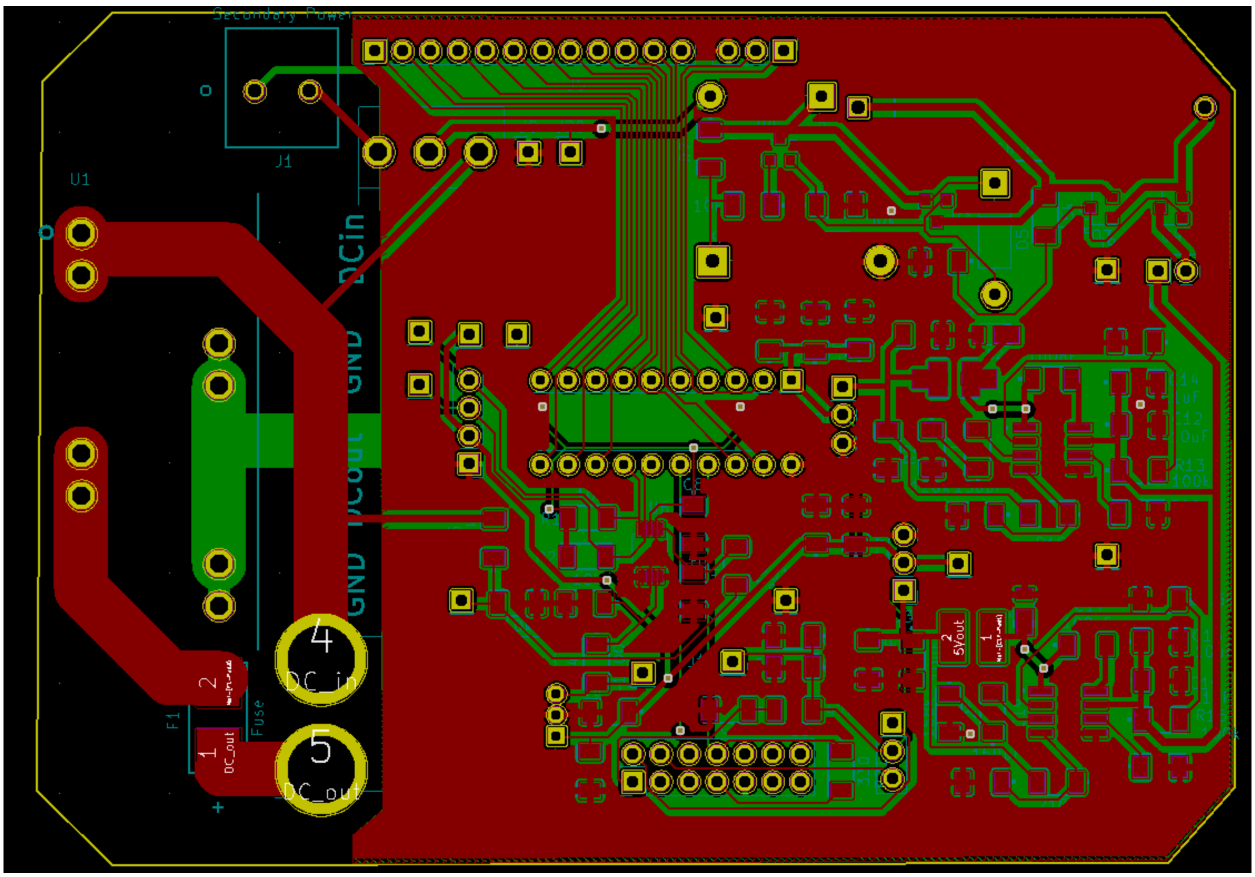

Figure 8: PCB filled

One of the safety precautions we needed to take into account is the current surge at the current measuring trace. The fuse was added for the prototype to prevent acceded current surge at the current measuring trace. The device needs to be able to measure at 100V 50A, that's a total of 5000 Watts can pass through the trace. Our solution for this is to increase the trace size and make it exposed copper. By doing so, the final product measuring trace can be filled with soldering lead to increase the amount of current passing by.

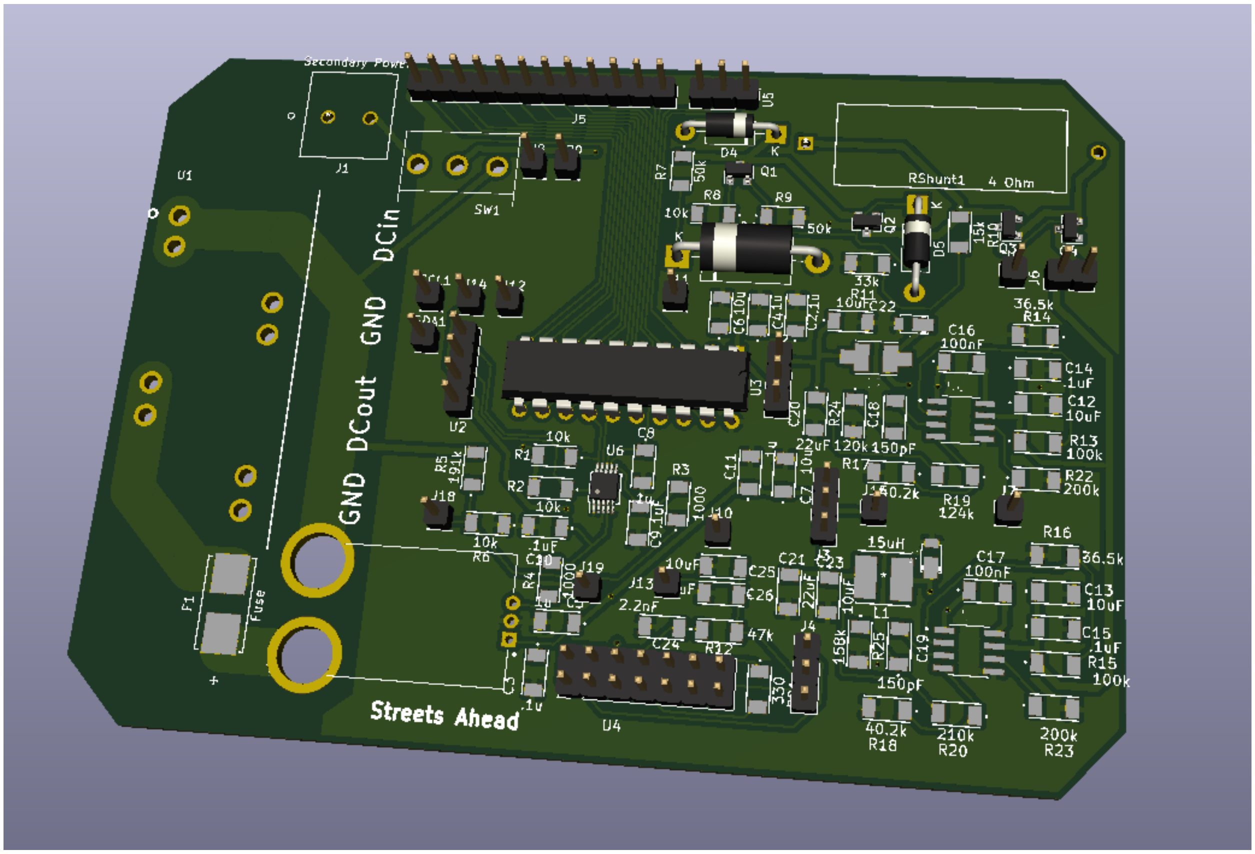

Figure 9: 3D view of PCB

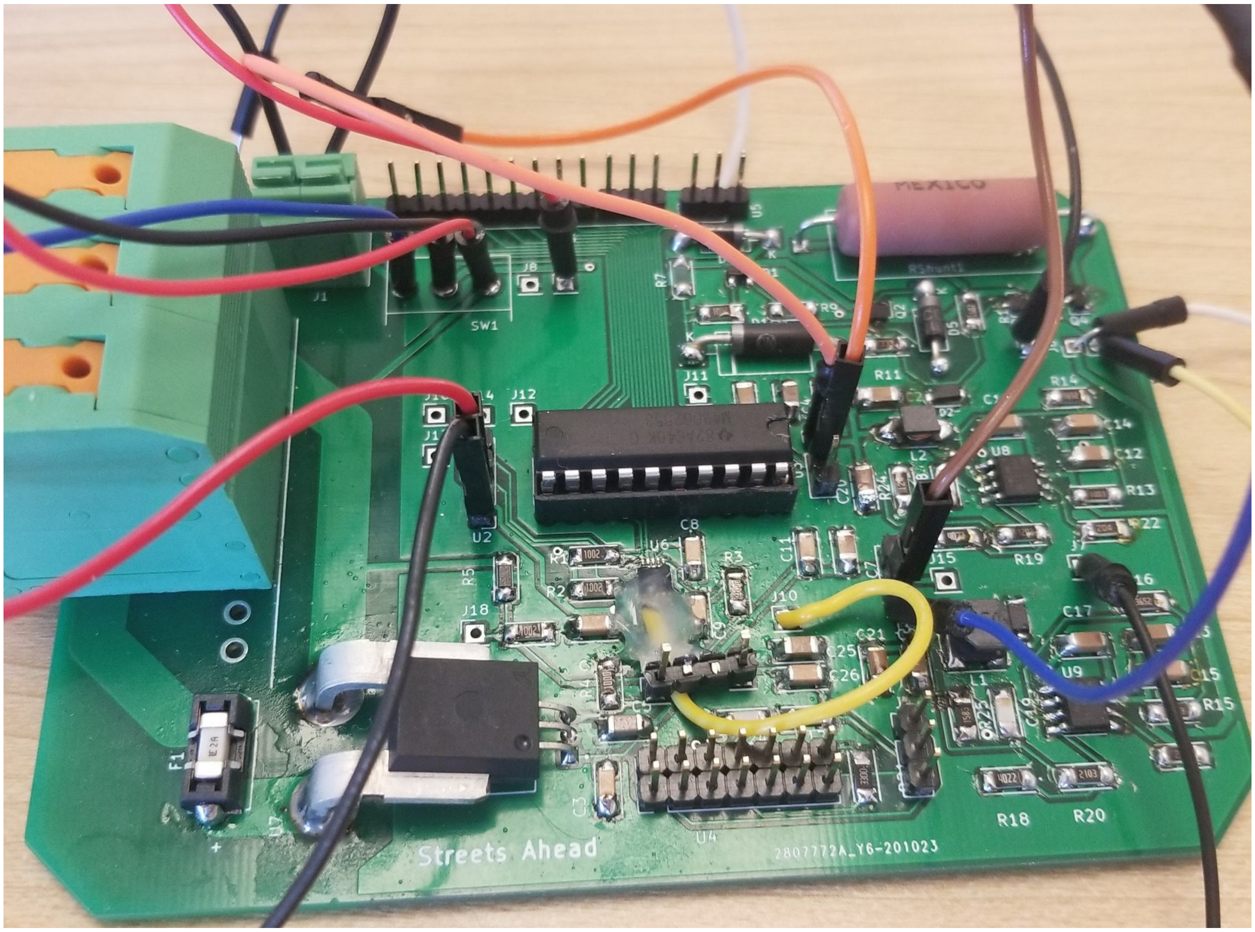

Figure 10: Soldered PCB

One of the setbacks we had was the address pin of the ADC needed to be grounded and the measuring pins of the current sensor needed to be widened. The address pin of the ADC needs to be grounded because it has multiple options for an I2C slave address based on its status. These problems were fixed in the updated PCB.

The project's software uses the MSP4302553 to control the I2C communication between the Analog to Digital Converter (ADC) and 16x2 LCD. I2C communication functions by sending and receiving data on one bus, SDA, while the master controls a clock bus, SCL, to tell the slave devices when to register a 1 or 0 on the data bus. Because the data line is connected to a pullup resistor, both the master and slave devices can control the data bit so that information can be both sent and received. Communication is initiated by a START condition consisting of pulling the data line low while the clock line is held high, and communication is ended with a STOP condition consisting of pulling the data line low while the clock is held high. Once a message has been started the MSP430 sends either the slave address of the LCD or ADC along with a read or write bit to declare what the processor wants to do with the following bits. Each byte of information is followed by an ACK to determine that the slave has properly received the data.

The voltage divider and current sensor both send a voltage on the range of 0-5V to an input on the ADC. In order to save power, single shot conversion is used to turn this voltage into a 16 bit, two's complement digital value. This value is sent back to the processor in two bytes with the most significant bits first. This number is then converted to a base 10 value by multiplying the binary number by the value the least significant bit represents, in the vase of the voltage, 187.5uV. These base 10 values are then added to an averaging filter to eliminate noise. These averaged values are then added to the char array to be sent to the show() method that takes each character and sends it to the LCD.

Project Timeline

Our original timeline did not vary too much from our final timeline. As shown in the Gantt chart figures below, more time was added for the PCB Design (minor changes in component footprints), and more time was added for the software, testing, assembly, and final report. The software timeline was extended due to the LCD's documentation and figuring out how to display the voltage and current values. Testing and assembly was delayed due to limited time with lab equipment and limited time for our team to meet in person. The final report was the last thing to do once the assembly and testing was completed.

Hieu is an Electrical Engineering major and his primary responsibility was to design the PCBs used in the DC Panel Meter. Hieu also assisted with calculations and testing. Jake is a Computer Engineering major and his primary focus was within the software portion, including programming the MSP430. Jake assisted with the physical assembly of the product as well as testing the software with the hardware. Kristian is an Electrical Engineering major and his primary focus was assembling the system, including soldering the PCBs and testing/debugging the PCBs. All members contributed to the final testing and assembly, along with the final report.

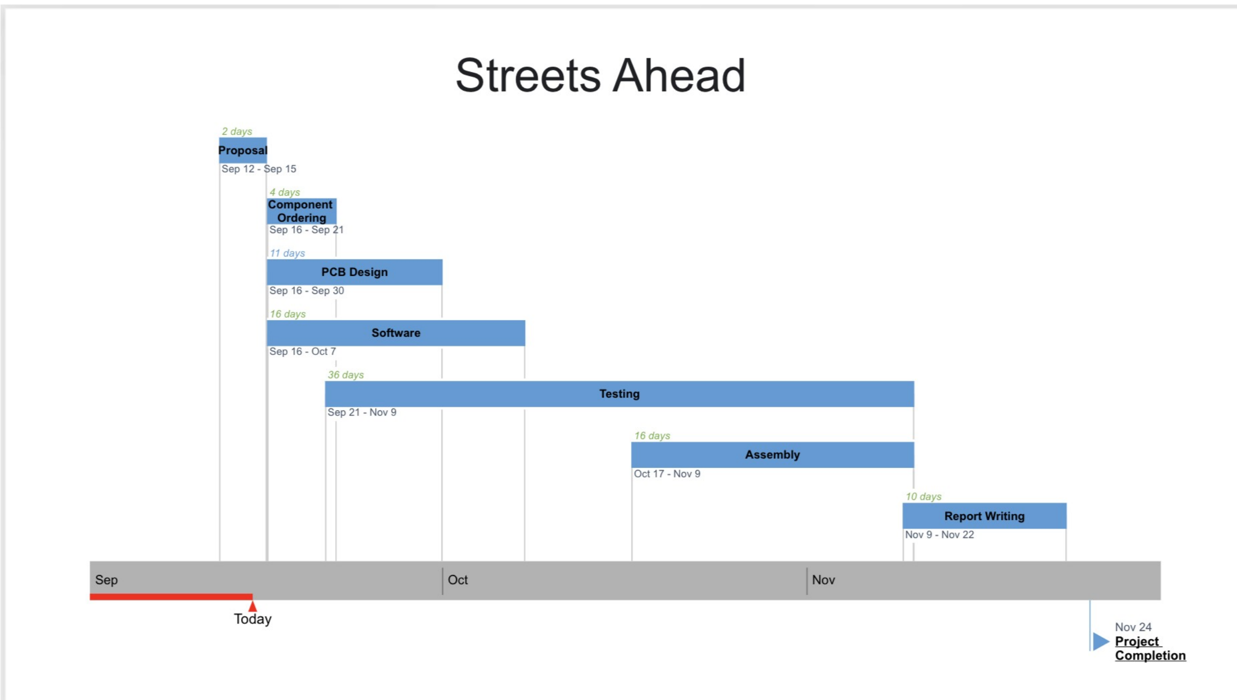

Figure 11: Original Timeline

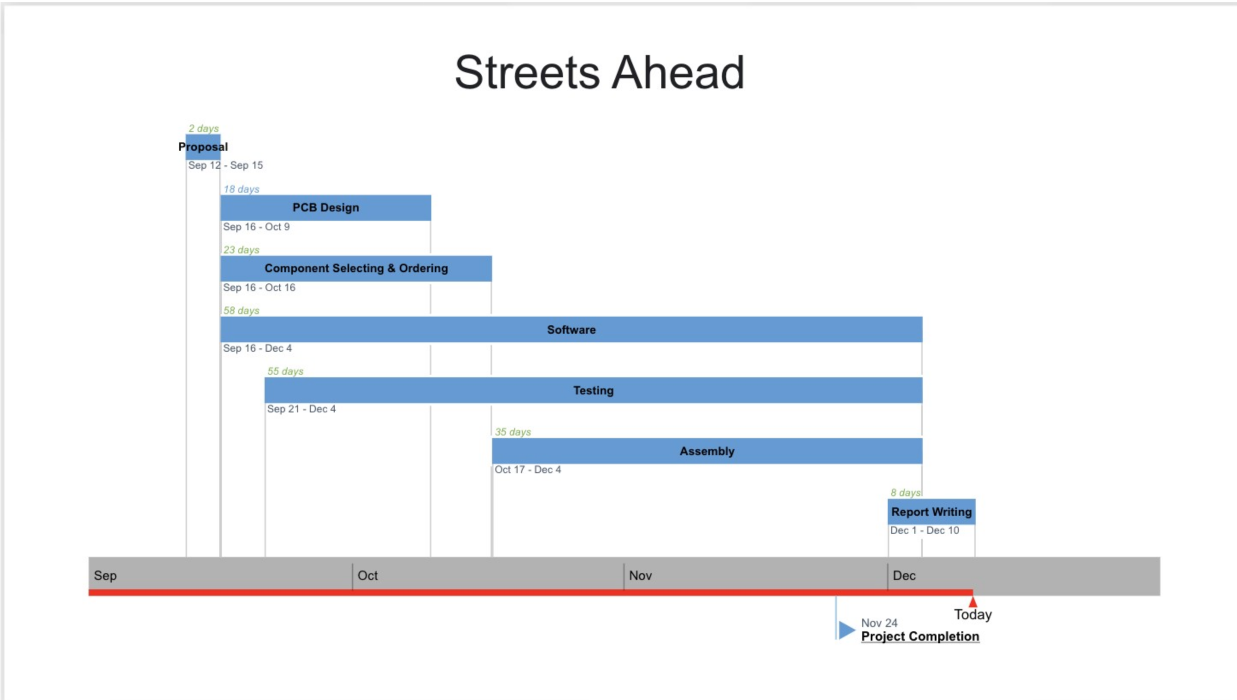

Figure 12: Final Timeline

Test Plan

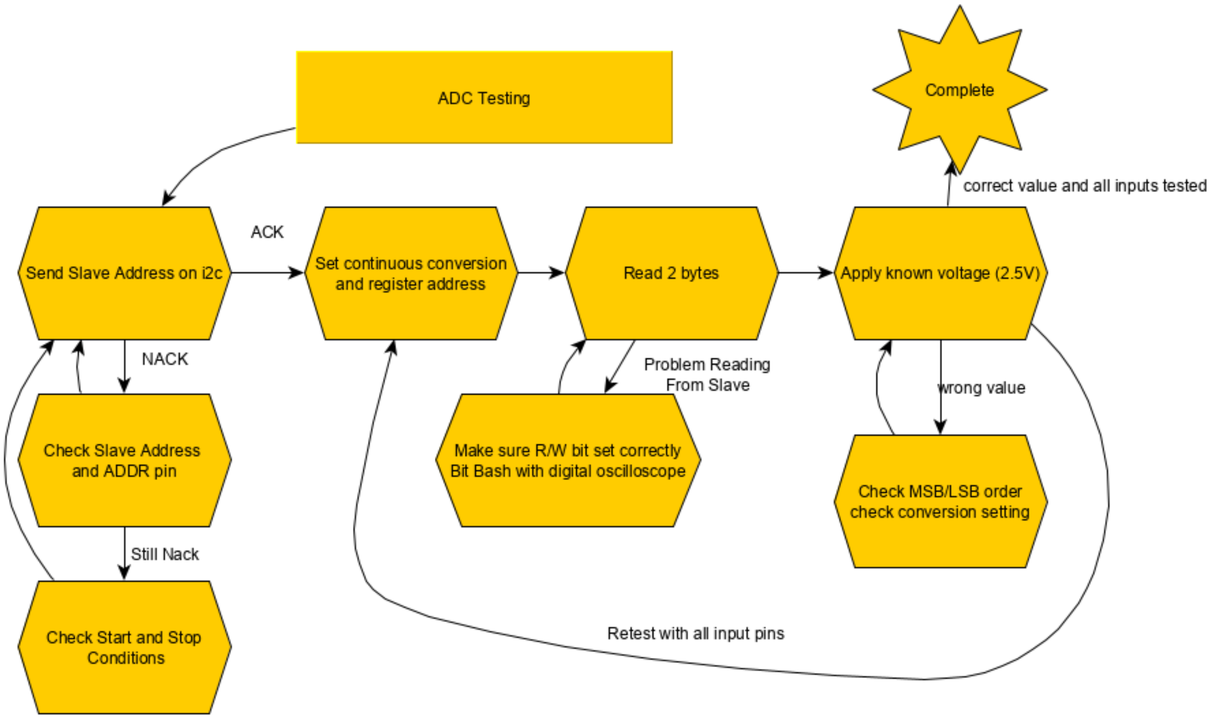

Figure 13: ADC Test Plan

Fortunately, the Analog to Digital Converter testing went smoothly thanks to any problems with the I2C system already having been discovered in the LCD testing. An ACK was successfully received from the device when its slave address was broadcast on the data line. The two bytes were able to be read and converted into the correct equivalent in terms of millivolts. The only change made was to later change the device to use single shot conversion as opposed to continuous to save power.

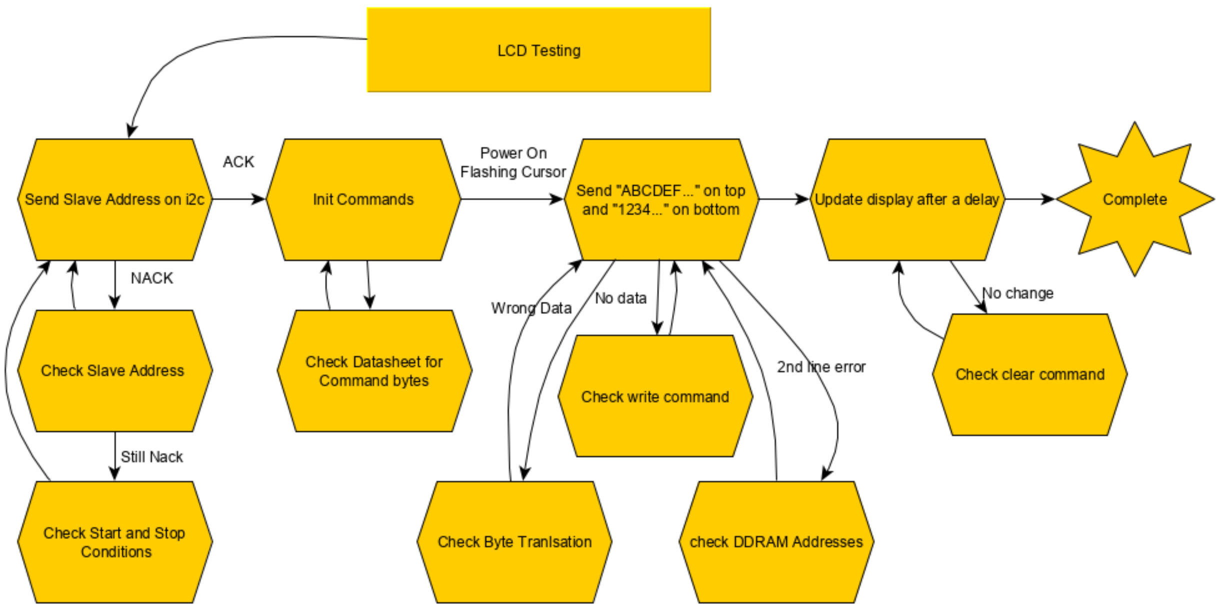

Figure 14: LCD test plan

Many obstacles were encountered due to the lacking Sparkfun datasheet. The init commands had to be found in the display driver's datasheet, but this lacked some of the commands that the sparkfun LCD had access to such as altering the color and brightness. Fortunately the slave address listed in the datasheet as the default was correct. In initial tests, blank characters were endemic when trying to display a line of "ABCDE...", this was eventually determined to be due to the sending of a data byte to let the device know character data was incoming was unnecessary, and was handled by the internal processor of the LCD.

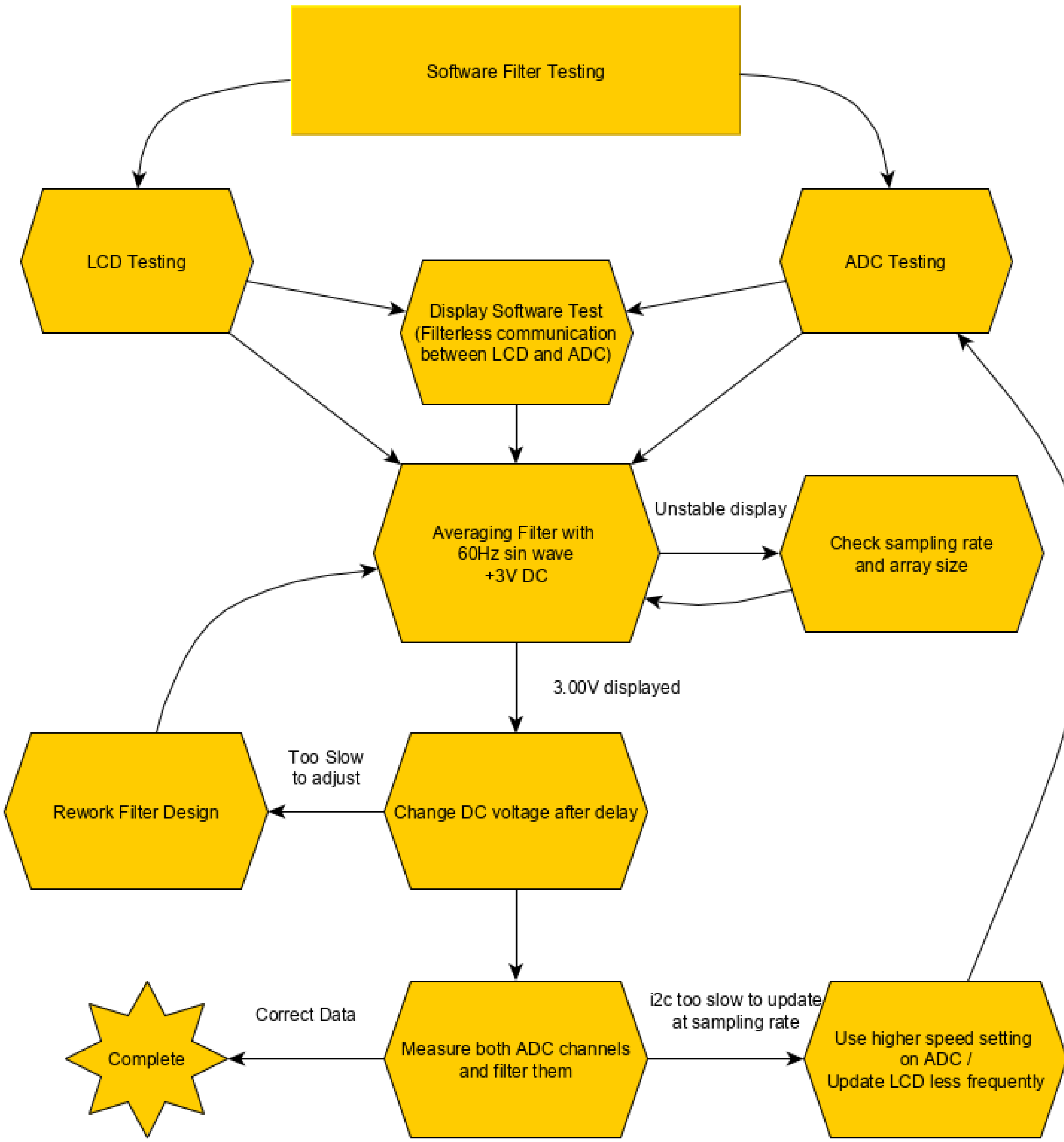

Figure 15: Filter test plan

In order to begin testing on the software filter, the ADC and LCD both needed to be functional and tested. A simple averaging filter was found to be sufficient to get the reading stable on the 10mV level, so no redesign was needed.

We began testing the hardware by checking for manufacturing issues - such as a ground connection where there should not be. We then checked to ensure that our schematic was implemented correctly. In order to test the Voltage Reverge Polarity Protection, the positive and negative wires were switched. If there was no voltage at the load when this change was made, then we moved on to the Overvoltage Protection test. For this test, we increased the voltage input to 50V-100V (with the switch set to less than 50V) and checked for a voltage at the load. If there was no voltage, we continued to the Current Limiting/Short Circuit Protection test. For this test, we shorted the load and ensured that there was no voltage reading at the load. If any of this criteria was not met, we checked our connections and components. Fortunately for us, everything in this portion of the test plan functioned as expected and we were able to continue to the Buck Converter testing.

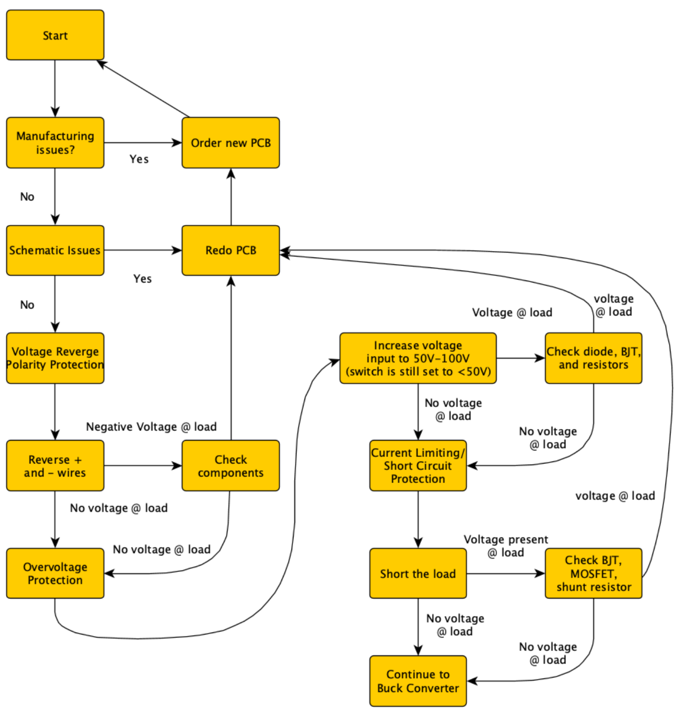

Figure 16: PCB Test Plan

In order to test the Buck Converter subsystem, we have to test both the 3.3 volt Buck converter and the 5 volt buck converter. We will test the 3.3 volt Buck Converter first. The voltage coming into the circuit is expected to be less than 50 volts. If the measured value does not match the expected value, we will ensure that the PCB connections are correct. Once the measured value is less than 50 volts, we will then test the voltage coming out of the 3.3 volt Buck Converter. If the measured V_out value is 0 volts, we need to replace the MP2456 component and start over testing the 3.3 volt Buck Converter. If the measured V_out value is 3.3 volts, we will then check to see if the value is fluctuating. If the 3.3 volts is fluctuating, we will ensure that the bypass capacitors are the correct values. If the 3.3 volts is not fluctuating, we can continue to test the 5 volt Buck Converter.

For the 5 volt Buck Converter, we will first test the voltage coming into the circuit. The expected V_in value is less than 50 volts. If the measured value does not match the expected value, we will ensure that the PCB connections are correct. Once the measured value is less than 50 volts, we will then test the voltage coming out of the 5 volt Buck Converter. If the measured V_out value is 0 volts, we need to replace the MP2456 component and start over testing the 5 volt Buck Converter. If the measured V_out value is 5 volts, we will then check to see if the value is fluctuating. If the 5 volts is fluctuating, we will ensure that the bypass capacitors are the correct values. If the 5 volts is not fluctuating, it can be concluded that the Buck Converter circuit subsystem is functioning properly and we can continue to test the MSP430 subsystem.

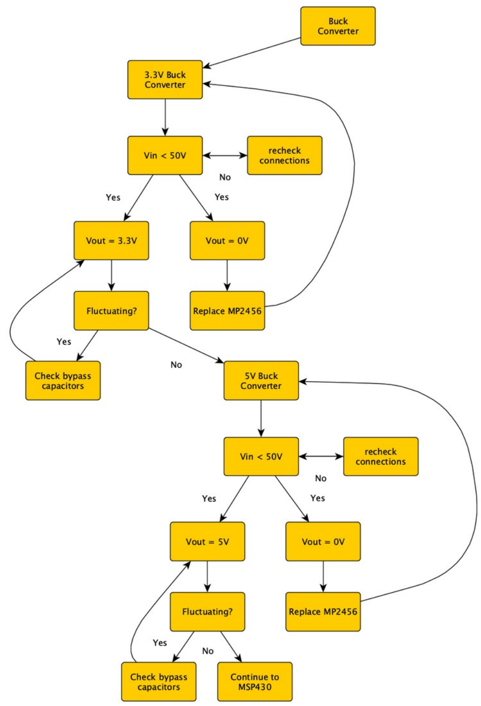

Figure 17: Buck Converter Test Plan

In order to test the MSP430 subsystem, we first will test the V_cc value. The expected value for V_cc is 3.3 volts. If the measured value does not equal the expected value, we will check the PCB connections. Once V_cc matches the expected value, we will check Pin 1.6 to see if there is a Pulse Width Modulation (PWM) signal. If there is no PWM signal present at Pin 1.6, the socket will need to be replaced and the MSP430 Test Plan will need to be restarted. If there is a PWM signal at Pin 1.6, it can be concluded that the MSP430 subsystem is functioning properly and we can continue to test the Logic Shifter subsystem.

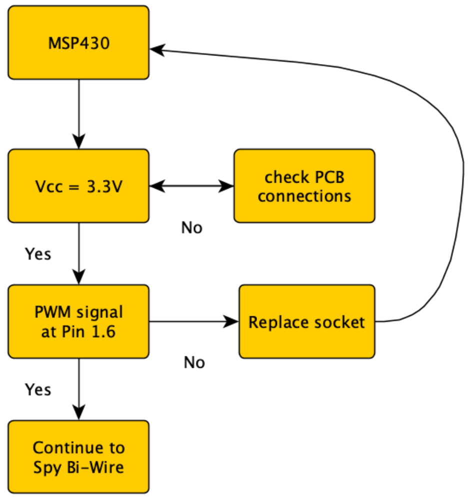

Figure 18: MSP Test Plan

In order to test the Spy Bi-Wire subsystem, we will test the voltage going into the circuit. The expected value for V_in is 3.3 volts. If the measured value does not equal the expected value, we will check the component values. Once V_in matches the expected value, it can be concluded that the Spy Bi-Wire subsystem is functioning properly and our testing of all of the subsystems is complete. A flowchart of the algorithm used to test the Spy Bi-Wire subsystem is shown in Figure 19: Spy Bi-Wire Test Plan.

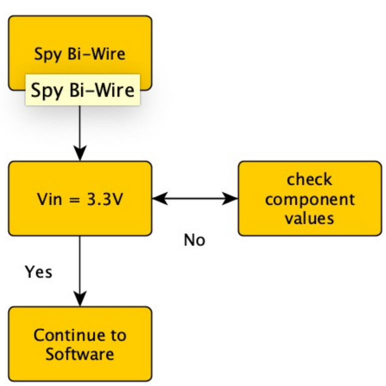

Figure 19: Spy Bi-Wire Test Plan

Final Results

Proposal Project Criteria	Project Completion Status
• Ability to measure both voltage and current down to 10mV and 1mA, respectively	**Achieved**
• Ability to safely measure up to 50V and 50A without any external power supply	**Achieved**
• Ability to safely measure up to 100V and 50A using an external power supply	**Achieved**
• Ability to display the correct voltage and current values on an LCD	**Achieved**
• Ability to offer reverse polarity, short circuit and overcurrent protection for the device	**Achieved**
• Ability to use a very small amount of power with high efficiency	**Achieved**
Resulting Letter Grade:	**A**

The figure below, Figure 20, is our final enclosed prototype. The maximum values that the meter is capable of measuring are 100V and 50A. The panel can read current and voltage down to 1mA and 10mV resolution, respectively. As expected, the device takes a very small amount of power from the power supply that it is measuring in order to power itself up to 50V. If the voltage of the power supply exceeds 50V, we also have an option of using an external power supply using a single pole double throw (SPDT). We did not have the high voltage supply which can go up to 100V and can provide 50A, however, we tested the device with Virtual Bench and it performed very well as expected. We also simulated the protections on Multisim and it performed as designed. We measure the voltage consumption of the DC panel to be around 100mW which meets our expectations.

All of the subsystems include: SPDT switch, protection circuit performance, Buck converter performance, the MSP430 and ADC communication, and other parts. All of the subsystems work flawlessly and the Project as a whole came together superbly in the end.

We tested the over-current protection functionality of the circuit, it automatically cut off when a short circuit was detected. We tested the reverse polarity protection by switching between positive input and negative input. The device shut off during the test and turned back on after the polarity was corrected.

Buck converters performance: All of the buck converters work without any drawbacks. They produced 3.3V and 5V continuously without any unaccounted fluctuation. We had a bit of concern about the 5V switching power regulator about how stable it is. We also had a backup plan about getting a reference voltage IC to replace it if it's not stable enough. The current sensor IC reliability was directly proportional to the stability of 5V voltage supply. After testing multiple scenarios, the 5V power supply from the switching regulator was able to perform flawlessly without any noticeable noise.

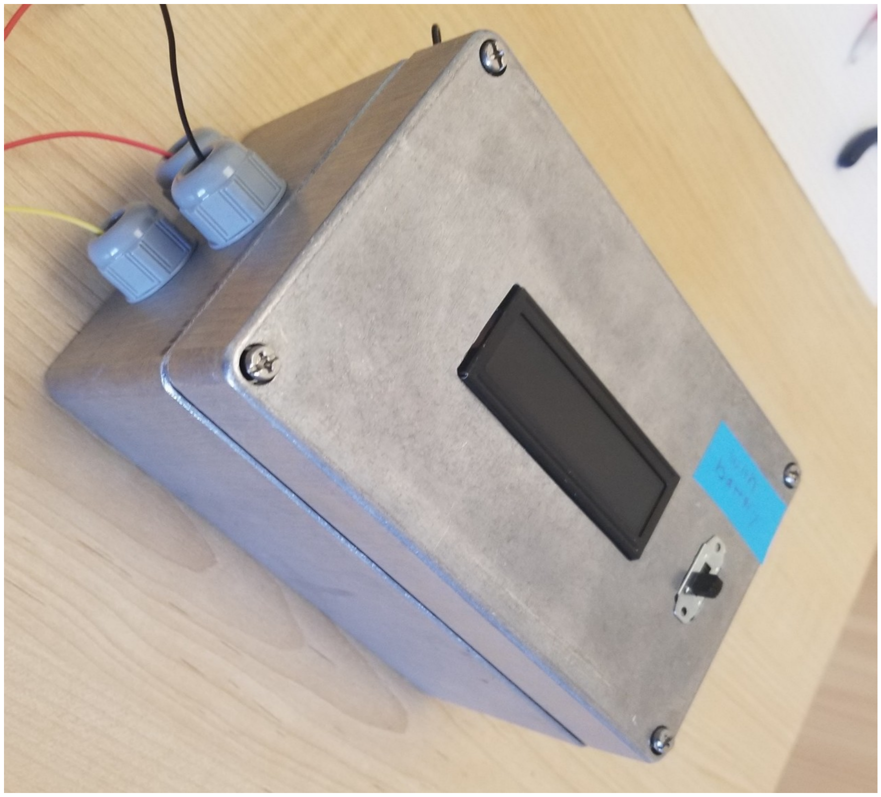

Figure 20: Final enclosed prototype

Costs

The three figures below show the components ordered during the three part sendouts. These numbers do not include the cost of the two PCBs that were ordered during the semester. We had a starting prototype budget of $500, and upon completion of the project, we were left with a surplus of $82.38. However, it should be noted that we ordered duplicate components in the event that the PCB needed to be redone, or if a component failed. With the amount of money that we spent, and the amount of parts that we ordered, we could have almost completed three prototypes (we would need another LCD, enclosure, and a few other components). If our prototype were to be manufactured and we were to purchase 10,000 times the components (not including the PCBs), the total cost would be $4,176,200. However, suppliers such as DigiKey offer discounts in bulk purchases, this would bring the cost down to $1,298,103.87 - nearly $3 million less. It is also important to note that if our prototype were to go into production, this cost would be much lower due to a different selection of some components. For example, the PCB cost would be far less than $33 per board, the aluminum box could be custom made for less money, the MSP430 could be interchanged with a less expensive MCU, and the LCD could be interchanged with a simpler display. Lastly, we purchased header boards for initial testing - these would not be required in manufacturing - resulting in a much lower cost for 10,000 units.

Component	Cost	Starting Budget	Remaining Budget
MSP430 Launchpad	$11.99	$500	$488.01

Figure 21: First Parts Order

Index	Quantity	Part Number	Manufacturer Part Number	Description	Customer Reference	Available	Backorder	Unit Price	Extended Price USD
1	33	445-5840-1-ND	CGA5L2X7R2A104K160AA	CAP CER 0.1UF 100V X7R 1206	STREETS AHEAD	33	0	0.208	6.86
2	6	587-1777-1-ND	HMK316B7105KL-T	CAP CER 1UF 100V X7R 1206	STREETS AHEAD	6	0	0.36	2.16
3	6	311-4392-1-ND	CC1206JRNPO0BN151	CAP CER 150PF 100V NPO 1206	STREETS AHEAD	6	0	0.28	1.68
4	3	478-1452-1-ND	12061A222JAT2A	CAP CER 2200PF 100V C0G/NP0 1206	STREETS AHEAD	3	0	0.39	1.17
5	6	445-8045-1-ND	C3216X5R1V226M160AC	CAP CER 22UF 35V X5R 1206	STREETS AHEAD	6	0	1.03	6.18
6	21	1276-6736-1-ND	CL31A106MBHNNNE	CAP CER 10UF 50V X5R 1206	STREETS AHEAD	21	0	0.2	4.2
7	6	CRS1206-FX-1001ELFCT-ND	CRS1206-FX-1001ELF	RES SMD 1K OHM 1% 1/2W 1206	STREETS AHEAD	6	0	0.42	2.52
8	6	PNM1206-50KBCT-ND	PNM1206E5002BST5	RES SMD 50K OHM 0.1% 0.4W 1206	STREETS AHEAD	6	0	1.54	9.24
9	3	RNCP1206FTD15K0CT-ND	RNCP1206FTD15K0	RES 15K OHM 1% 1/2W 1206	STREETS AHEAD	3	0	0.1	0.3
10	3	RHM33KICT-ND	ESR18EZPJ333	RES SMD 33K OHM 5% 1/2W 1206	STREETS AHEAD	3	0	0.14	0.42
11	3	RHM47KICT-ND	ESR18EZPJ473	RES SMD 47K OHM 5% 1/2W 1206	STREETS AHEAD	3	0	0.14	0.42
12	6	RHM100KAFCT-ND	ESR18EZPF1003	RES SMD 100K OHM 1% 1/2W 1206	STREETS AHEAD	6	0	0.13	0.78
13	6	A140235CT-ND	RQ73C2B36K5BTD	RES 36.5 KOHMS 0.1% 0.4W 1206	STREETS AHEAD	6	0	0.87	5.22
14	6	RNCP1206FTD40K2CT-ND	RNCP1206FTD40K2	RES 40.2K OHM 1% 1/2W 1206	STREETS AHEAD	6	0	0.1	0.6
15	3	A140968CT-ND	RQ73C2B124KBTD	RES 124 KOHMS 0.1% 0.4W 1206	STREETS AHEAD	3	0	0.87	2.61
16	3	A139762CT-ND	RQ73C2B210KBTD	RES 210 KOHMS 0.1% 0.4W 1206	STREETS AHEAD	3	0	0.87	2.61
17	6	RHM200KICT-ND	ESR18EZPJ204	RES SMD 200K OHM 5% 1/2W 1206	STREETS AHEAD	6	0	0.14	0.84
18	3	RHM120KICT-ND	ESR18EZPJ124	RES SMD 120K OHM 5% 1/2W 1206	STREETS AHEAD	3	0	0.14	0.42
19	3	A130532CT-ND	CRGP1206F330R	CRGP 1206 330R 1%	STREETS AHEAD	3	0	0.28	0.84
20	3	A139923CT-ND	RQ73C2B158KBTD	RES 158 KOHMS 0.1% 0.4W 1206	STREETS AHEAD	3	0	0.87	2.61
21	3	ALSR5F-4.0-ND	ALSR054R000FE12	RES 4 OHM 5W 1% AXIAL	STREETS AHEAD	3	0	1.71	5.13
22	3	CWI513-ND	GF-124-0013	SWITCH SLIDE SPDT 500MA 125V	STREETS AHEAD	3	0	0.86	2.58
23	3	SRN5040TA-150MCT-ND	SRN5040TA-150M	FIXED IND 15UH 1.8A 80 MOHM SMD	STREETS AHEAD	3	0	0.52	1.56
24	3	490-2519-1-ND	LQH43CN100K03L	FIXED IND 10UH 650MA 240 MOHM	STREETS AHEAD	3	0	0.43	1.29
25	3	BSS83PH6327XTSA1CT-ND	BSS83PH6327XTSA1	MOSFET P-CH 60V 0.33A SOT23	STREETS AHEAD	3	0	0.41	1.23
26	3	1N5369BGOS-ND	1N5369BG	DIODE ZENER 51V 5W AXIAL	STREETS AHEAD	3	0	0.48	1.44
27	6	DFLS160DICT-ND	DFLS160-7	DIODE SCHOTTKY 60V 1A POWERDI123	STREETS AHEAD	6	0	0.44	2.64
28	3	277-2518-ND	1792863	TERM BLK 2POS SIDE ENTRY 5MM PCB	STREETS AHEAD	3	0	0.34	1.02
29	3	2057-EB171A-03-H-ND	EB171A-03-H	EURO BLOCK, 3 POSITION	STREETS AHEAD	3	0	3.08	9.24
30	3	ALSR5F-4.0-ND	ALSR054R000FE12	RES 4 OHM 5W 1% AXIAL	STREETS AHEAD	3	0	1.71	5.13
31	3	DFLS160DICT-ND	DFLS160-7	DIODE SCHOTTKY 60V 1A POWERDI123	STREETS AHEAD	3	0	0.44	1.32
32	3	1175-2553-ND	245-20-1-03	CONN IC DIP SOCKET 20POS TIN	STREETS AHEAD	3	0	0.44	1.32
33	3	F6252CT-ND	0154002.DRL	FUSE BOARD MNT 2A 125VAC/VDC SMD	STREETS AHEAD	3	0	2.9	8.7
34	3	P191KBCCT-ND	ERA-8AEB1913V	RES SMD 191K OHM 0.1% 1/4W 1206	STREETS AHEAD	3	0	0.57	1.71
35	12	RTAN1206BKE10K0CT-ND	RTAN1206BKE10K0	RES 10K OHM 0.1% 2/5W 1206	STREETS AHEAD	12	0	0.525	6.3
36	2	296-28429-5-ND	MSP430G2553IN20	IC MCU 16BIT 16KB FLASH 20DIP	STREETS AHEAD	2	0	2.69	5.38
37	1	1568-1904-ND	LCD-14073	LCD MODULE 32 DIG 16 X 2	STREETS AHEAD	1	0	19.95	19.95
38	1	1528-1461-ND	1085	ADS1115 16BIT ADC 4CH PROG GAIN	Streets Ahead	1	0	14.95	14.95
39	1	1738-1102-ND	SEN0098	50A CURRENT SENSOR(AC/DC)	Streets Ahead	1	0	14.7	14.7
40	4	1589-1602-1-ND	MP2562DS-LF-Z	IC REG BUCK ADJUSTABLE 1A 8SOIC	Streets Ahead	4	0	2.86	11.44
41	1	620-1353-ND	ACS758LCB-050U-PFF-T	SENSOR CURRENT HALL 50A DC	Streets Ahead	1	0	7.18	7.18
42	1	296-24934-1-ND	ADS1115IDGST	IC ADC 16BIT SIGMA-DELTA 10VSSOP	Streets Ahead	1	0	6.1	6.1
43	6	1N4002-TPMSCT-ND	1N4002-TP	DIODE GEN PURP 100V 1A DO41	Streets Ahead	6	0	0.12	0.72
44	3	BSS63LT1GOSCT-ND	BSS63LT1G	TRANS PNP 100V 0.1A SOT-23	Streets Ahead	3	0	0.21	0.63
45	6	1727-1254-1-ND	BC52PA,115	TRANS PNP 60V 1A SOT1061	Streets Ahead	6	0	0.43	2.58
								Starting Budget:	500
								Total:	185.92
								Expenses:	11.99
								Remaining Budget:	302.09

Figure 22: Second Parts Order

Index	Quantity	Part Number	Manufacturer Part Number	Description	Customer Reference	Available	Backorder	Unit Price	Extended Price USD
1	2	296-24934-1-ND	ADS1115IDGST	IC ADC 16BIT SIGMA-DELTA 10VSSOP	Streets Ahead	2	0	6.1	12.2
2	2	620-1353-ND	ACS758LCB-050U-PFF-T	SENSOR CURRENT HALL 50A DC	Streets Ahead	2	0	7.18	14.36
3	3	RNCP1206FTD15K0CT-ND	RNCP1206FTD15K0	RES 15K OHM 1% 1/2W 1206	Streets Ahead	3	0	0.1	0.3
4	2	1589-1602-1-ND	MP2562DS-LF-Z	IC REG BUCK ADJUSTABLE 1A 8SOIC	Streets Ahead	2	0	2.86	5.72
5	6	478-1450-1-ND	12061A102JAT2A	CAP CER 1000PF 100V C0G/NP0 1206	Streets Ahead	6	0	0.28	1.68
6	6	RNCP1206FTD100RCT-ND	RNCP1206FTD100R	RES 100 OHM 1% 1/2W 1206	Streets Ahead	6	0	0.1	0.6
7	1	902-1300-ND	11016088	BOX ALUM UNPAINTED 6.3"LX3.94"W	Streets Ahead	1	0	46.43	46.43
8	10	WM11991-ND	936000343	POLYAMIDE CABLE GLAND M16X1,5	Streets Ahead	10	0	0.672	6.72
								Starting Budget:	500
								Order Total:	88.01
								Other total:	263.61
								Remaining:	148.38

Figure 23: Third Parts Order

Future Work

One of the greatest challenges on the software side of this project was dealing with the lacking datasheet for the Sparkfun LCD. While this would not be a problem if an 8 bus solution was used, as the HD44780 display driver datasheet is very clear for this use, the Sparkfun LCD uses its own processor for i2c commands, which were only available for Arduino use and not listed anywhere in the Sparkfun datasheets. If future students were to try and replicate this project and stick with an i2c method for driving the display, it is highly recommended that they go with an LCD that uses the ST7032 display driver, which is made for i2c communication only and has an in-depth datasheet for its use.

References

[1] “IPC-2221.pdf.” Accessed: Dec. 10, 2020. [Online]. Available: https://ee174-fall15-01.courses.soe.ucsc.edu/system/files/attachments/IPC-2221.pdf.

[2] “Interfacing 16x2 Alphanumeric LCD display with STM8 Microcontroller.” https://pic-microcontroller.com/interfacing-16x2-alphanumeric-lcd-display-with-stm8-microcontroller/ (accessed Dec. 10, 2020).

[3] “MSP430 Spy-Bi-Wire With SimpleLink MCUs,” p. 29, 2017.

[4] “What is Multisim™? - National Instruments.” https://www.ni.com/en-us/shop/electronic-test-instrumentation/application-software-for-electronic-test-and-instrumentation-category/what-is-multisim.html (accessed Sep. 22, 2019).

[5] “Ultiboard - National Instruments.” http://www.ni.com/en-us/shop/select/ultiboard (accessed Sep. 22, 2019).

[6] “CCSTUDIO Code Composer Studio (CCS) Integrated Development Environment (IDE) | TI.com.” http://www.ti.com/tool/CCSTUDIO (accessed Sep. 22, 2019).

[7] “What Is VirtualBench? - National Instruments.” http://www.ni.com/en-us/shop/electronic-test-instrumentation/virtualbench/what-is-virtualbench.html (accessed Sep. 22, 2019).

[8] “Controlling Electrical Hazards.” https://www.osha.gov/Publications/3075.html (accessed Sep. 15, 2020).

[9] “NEMA Ratings for Enclosures - NEMA Enclosure Ratings Chart (1,3, 3R, 4, 4X, 12) | Nema Enclosures.” https://www.nemaenclosures.com/enclosure-ratings/nema-rated-enclosures.html (accessed Sep. 08, 2020).

[10] “Electronics Assembly Standards IPC J-STD-001 and IPC-A-610 Updated to Cover More Advanced Technologies.” http://www.ipc.org/feature-article.aspx?aid=Electronics-Assembly-Standards-IPC-J-STD-001-and-IPC-A-610-Updated-to-Cover-More-Advanced-Technologies (accessed Dec. 16, 2019).

[11] “DigiKey Electronics - Electronic Components Distributor.” https://www.digikey.com/ (accessed Sep. 22, 2019).

[12] “Mouser Electronics - Electronic Components Distributor.” https://www.mouser.com/ (accessed Sep. 22, 2019).

[13] “Newark Electronics | Electronic Components Distributor.” https://www.newark.com/ (accessed Sep. 22, 2019).

[14] M. Marzynski and E. Eng, “Digital multimeters including a remote display,” US8803507B2, Aug. 12, 2014.

[15] S. A. A. DANESH and J. E. D. Hurwitz, “Voltage measurement,” US9151818B2, Oct. 06, 2015.

[16] S. J. M. WOOD, J. E. D. Hurwitz, and S. A. A. DANESH, “Current measurement,” US10527651B2, Jan. 07, 2020.

Appendix

Code:

```c
#include <msp430.h>
#include <stdio.h>

#define I2C_SDA BIT7   // Serial Data line
#define I2C_SCL BIT6   // Serial Clock line

/* Caveman delay function */
void delay( unsigned int n ) {
    volatile int i;

    for( ; n; n-- ) {
        for( i = 0; i < 50; i++ );
    }
}

void data_read(void ) {
    P1DIR &= ~I2C_SDA; // float to get ready to read
}

void data_high(void ) {
    P1DIR &= ~I2C_SDA; // float pin to go high
    //P1OUT |= I2C_SDA;
    //P1DIR |= I2C_SDA;
    delay( 5 );
}

void data_low(void ) {
    P1OUT &= ~I2C_SDA; // assert low
    P1DIR |= I2C_SDA;
    delay( 5 );
}

void clk_high(void) {
    P1DIR &= ~I2C_SCL;  // float pin to go high
    //P1OUT |= I2C_SCL;
    //P1DIR |= I2C_SCL;
    delay( 10 );
}

void clk_low(void) {
    P1OUT &= ~I2C_SCL;  // assert low
    P1DIR |= I2C_SCL;
    delay( 5 );
}

/* I2C communication starts when both the data and clock
 * lines go low, in that order. */
void I2C_Start(void) {
    clk_high();
    data_high();
    data_low();
    clk_low();
}

/* I2C communication stops with both the clock and data
 * lines going high, in that order. */
void I2C_Stop(void) {
    data_low();
    clk_low();
```

```
    clk_high();
    data_high();
}

/* Outputs 8-bit command or data via I2C lines. */
void I2C_out(unsigned char d) {
    int n;

    for( n = 0; n < 8; n++ ) {
        if( d & 0x80 ) {
            data_high();
        } else {
            data_low();
        }

        clk_high();
        clk_low();

        d <<= 1;        // Shift next bit into position.
    }

    data_read();        // Set data line to receive.
    clk_high();         // Clock goes high to wait for acknowledge.


    // Slave will pull data line low to acknowledge.
    /*while( P1IN & I2C_SDA ) {
        // Else toggle the clock line and check again
        clk_low();
        clk_high();
    }*/

    clk_low();
}
unsigned char I2C_read (void)
{
    unsigned char inputData, inBits;

    inputData = 0x00;
    /* 8 bits */
    for(inBits = 0; inBits < 8; inBits++)
    {
        inputData <<= 1;
        clk_high();
        inputData |= I2C_SDA;
        clk_low();
    }

   return inputData;
}


/* Initializes the LCD panel. */
void init_LCD(void) {
    I2C_Start();

    I2C_out( 0xE4 );    // Slave address of the LCD panel.
    //I2C_out( 0x00 );    // Control byte: all following bytes are commands.
```

```
    //I2C_out( 0x38 );    // 8-bit bus, 2-line display, normal instruction mode.
    //delay( 5 );
    //I2C_out( 0x0E );    // turn on display and cursor
    //delay( 5 );
    //I2C_out( 0x06 );    // Entry mode set to cursor-moves-right.
    //delay( 5 );

    I2C_Stop();
}

void init_ADC(void){
    I2C_Start();

    I2C_out( 0x90 );
    I2C_out( 0x01 );
    I2C_out( 0xC1 );
    I2C_out( 0x83 );


    I2C_Stop();
}

int read_ADC_1(void){
    I2C_Start();

    I2C_out( 0x90 );
    I2C_out( 0x00 );

    I2C_Stop();
    delay( 10 );

    I2C_Start();
    I2C_out( 0x91 );
    //read two bytes, MSB then LSB
    unsigned char msb = I2C_read();
    unsigned char lsb = I2C_read();
    I2C_Stop();

    //return (int)( lsb + (msb << 8u) );
    return 0;//test
}
int read_ADC_2(void){
    I2C_Start();

    I2C_out( 0x90 );
    I2C_out( 0x01 );

    I2C_Stop();
    delay( 10 );

    I2C_Start();
    I2C_out( 0x91 );
    //read two bytes, MSB then LSB
    unsigned char msb = I2C_read();
    unsigned char lsb = I2C_read();
    I2C_Stop();

    //return (int)( lsb + (msb << 8u) );
    return 0;
```

```
}

/* Sends the "clear display" command to the LCD. */
void clear_display(void) {
    I2C_Start();

    I2C_out( 0xE4 ); // Slave address of panel.
    delay(5);
    I2C_out( 0x00 ); // Control byte: all following bytes are commands.
    delay(5);
    I2C_out( 0x01 ); // Clear display.
    delay(5);
    I2C_out( 0x02 ); // Return Home

    I2C_Stop();
}

/* Writes a 20-char string to the RAM of the LCD. */
void show( unsigned char *text, float num) {
    int n;

    I2C_Start();

    I2C_out( 0xE4 ); // Slave address of panel.
    //I2C_out( 0x40 ); // Control byte: data bytes follow, data is RAM data.

    /*for( n = 0; n < 16; n++ ) {
        I2C_out( *text );
        text++;
    }*/
    for( n = 0; n < 10; n++ ) {
        I2C_out( *text );
        text++;
    }

    int intnum = (int)num;
    int decinum = (int)((int)(num*100.0) - intnum*100);

    for( n = 0; n < 2; n++ ) {
        int digit = intnum % 10;
        intnum = intnum/10;
        I2C_out(digit + 0x30);
        text++;
    }
    I2C_out(0x2E);//"."
    for( n = 0; n < 3; n++ ) {
        int digit2 = decinum % 10;
        I2C_out( *text );
        text++;
    }

    I2C_Stop();
}
void secondLine(){
    I2C_Start();
    I2C_out( 0xE4 );
    I2C_out( 0x00 );
    I2C_out( 0xC0 );
    I2C_Stop();
```

```
}

int main(void) {
    //int i;

    /* Stop the watchdog timer so it doesn't reset our chip */
    WDTCTL = WDTPW + WDTHOLD;

    init_LCD();
    float voltage = 0.0;
    float current = 0.0;
    int num = 0;
    while(1){
        //clear_display();
        delay(10);
        //secondLine();
        voltage = read_ADC_1() * .1875 * 20.1;
        current = (read_ADC_2()-2.5) * 7.284;
        //show( "Voltage = ", voltage);
        //show( "Current =       ", voltage);
        show( "Voltage = ", voltage);
        show( "Current = ", current);
    }
    __bis_SR_register( LPM3_bits );     /* go to sleep */


    /*init_LCD();
    while(1)
    {
        show( "Hello, world.   " );
        clear_display();
    }*/

    /*
    while(1)
    {
        I2C_Start();
        I2C_out(0x05);
        I2C_Stop();
        delay(20);
    }*/

    /*while(1)
    {
        clk_high();
        data_high();
        delay(5);
        clk_low();
        data_low();
    }*/

}
```