# SMART INTERFACE BOX FOR SOLAR PANELS

**Final Report**

**December 12, 2019**

**ECE 445 (Fall 2019)**

Team #3

Dillon Vadgama (dvadga2)

Douglas J. Lee (dlee242)

Sachin Reddy (ssreddy2)

TA: Shaoyu Meng

**Abstract**

The Smart Interface Box for Solar Panels is a protection system that is designed specifically for the Electrical and Computer Engineering building's (ECEB) research solar panels. The primary goal of this design is to shut down the solar panel in the case of failure conditions such as overvoltage, overcurrent, or overheating are detected in order to prevent catastrophic failure or collateral damage to the building and/or solar panels. Additionally, this design contains a remote user interface to allow for system configuration and monitoring of solar panel parameters over a network connection. The device was successfully designed, tested, and implemented. At the time of writing this report, a prototype unit was installed on one of the research solar panels on the roof of ECEB.

# Contents

# 1 Introduction

## 1.1 Problem and Solution Overview

The Electrical and Computer Engineering building (ECEB) at the University of Illinois at Urbana-Champaign has 60 research solar panels located on its roof; however, because there are no protection interfaces between the solar panels and the load they are delivering power to, none of these panels are in use. Additionally, without such a protection system, these solar panels are at risk of suffering the same type of failure as the ones that caused seven rooftop fires to break out in Walmart stores in Beavercreak, Ohio.[1]

A smart interface box attached to each solar panel has the ability to disconnect the solar panel from the electrical load in the event certain failure conditions (over-current, over-voltage, and overheating) are met and allows users to remotely monitor system behavior/parameters helps to prevent a disaster like Walmart's. In addition, the ability to remotely configure the number of cells on the solar panel delivering power to the load is particularly useful to researchers. A visual aid of where the proposed systems fits in between a solar panel and the electrical load is shown in Figure 1.

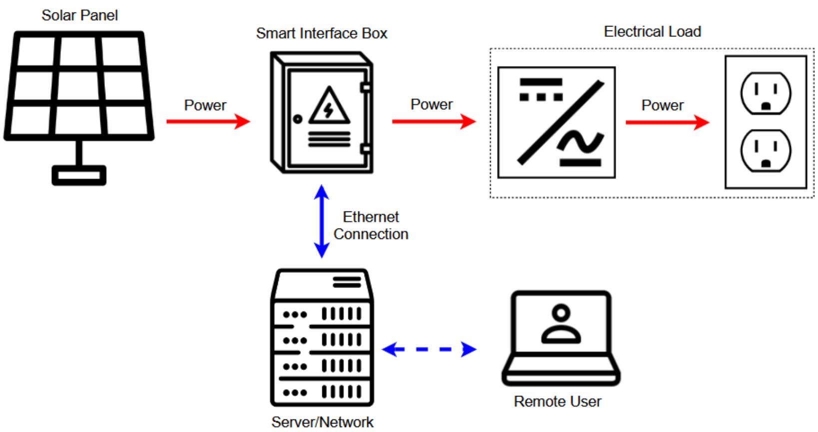



**Figure 1:** Visual aid

After constructing the proposed system, testing with the solar panels on the roof of ECEB proved to be largely successful. In general, very little design changes were made progressing from the proposal stage to the implementation stage.

## 1.2 High Level Requirements

- The output of the interface box will be configurable to be connected to either 32 solar cells, 64 solar cells, or 128 solar cells on the solar panel.
- The output of the interface box will be disconnected from the load in the event a solar panel failure condition occurs (over-voltage, over-current, or overheating), or the isolated 12 V supply powering the interface box is no longer connected.
- The end user will be able to remotely monitor system parameters (e.g. output configuration, output current, voltages on all solar panel partitions, output power, and temperatures) and will also be able to remotely configure various system parameters (e.g. set output configuration and set failure condition thresholds).

# 2 Design

## 2.1 Block Diagram

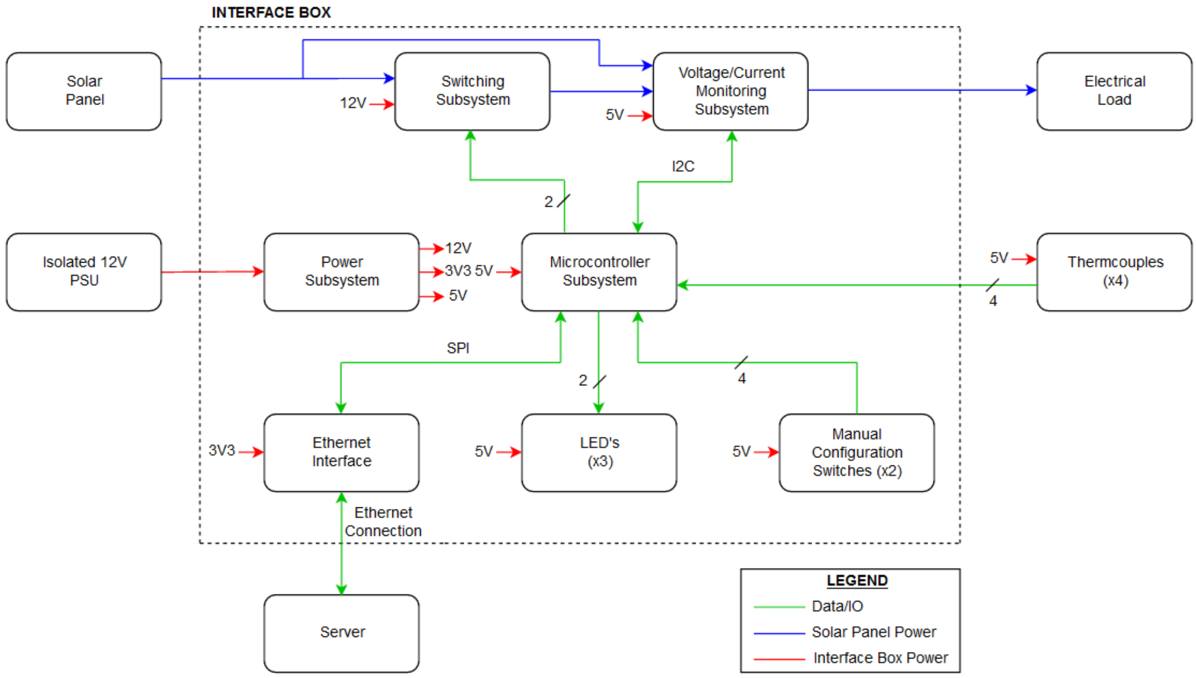



**Figure 2:** System Block Diagram

The interface box for the solar panel consists of eight important subsystems as shown in Figure 2: the Power Subsystem, Switching Subsystem, Voltage/Current Monitoring Subsystem, Thermocouples, Ethernet Interface, LED's, Manual Configuration Switches, and Micro-controller/Processing Subsystem.

The Power Subsystem is responsible for generating from a 12V power source all of the voltage rails necessary for the interface box to operate as expected as well as toggling power to the interface box in general. These voltages include 5V and 3.3V.

The Switching Subsystem connects the appropriate solar panel cell combination (32 cells, 64 cells, or 128 cells) to the output of the interface box and is controlled by the Microcontroller Subsystem.

The Voltage/Current Monitoring Subsystem is responsible for measuring the voltages of all possible output configurations and the current delivered to the load, as well as reporting that information over an I2C data bus.

Thermocouples that are mounted directly on the back of the solar panel report temperature data back to the Microcontroller Subsystem.

The Ethernet Interface is necessary for ensuring that the interface box is able to connect to a network/server and, additionally, allowing a remote user to monitor and configure the interface box. When an Ethernet connection is not available, users can control the output configuration and toggle the output via the Manual Configuration Switches.

LED's aid in manual configuration. They are used to indicate interface box power, whether the output is enabled, and whether the Ethernet interface is being used.

The Manual Configuration Switches are used to allow a user to physically control the switching subsystem and connect/disconnect the output of the solar panel from the electrical load.

Lastly, the Microcontroller Subsystem acts as the central processing unit of the system and is responsible for carrying out commands sent through the Ethernet Interface by the user (or the Manual Configuration Switches if Ethernet is not being used), monitoring for failure conditions that occur, setting the appropriate output configuration, and setting the LED's to their appropriate states.

## 2.2 Mechanical Design

Because the interface box is installed outside with the solar panels, an important requirement is that the enclosure is weather-proof. To achieve this goal, an IP-67 rated enclosure was chosen to house the electronics.

There are four primary connections from the electronics to the outside world: temperature sensors, 12V supply, Ethernet cable, and solar panel connections. The enclosure connected to the solar panel via MC4 connectors as shown in Figure 13 in Appendix C, while all of the other cables are secured with cable glands for a waterproof connection as shown in Figure 12 in Appendix C. All holes drilled in the enclosure have O-ring seals.

## 2.3 Solar Panel Structure

Before delving into each subsystem contained in the interface box, it is important to discuss the inner workings and specifications of the solar panel that this system was designed for. Table 1 shows important specifications that were used throughout the design process.[2]

As a part of the requirements, the interface box needs to be able to control the output configuration such that the load is connected to one of three different solar cell combinations: 32 cells, 64 cells, and 128 cells. From the structure of the solar panel,

| | |
|---|---|
| Current at Max Power | 5.83 A |
| Voltage at Max Power | 72.9 V |
| Short Circuit Current | 6.21 A |
| Open Circuit Voltage | 85.6 V |

**Table 1:** Important Solar Panel Specifications

it was determined that there were three partitions of cells wired in series on the panel: two 32 cells partitions, and one 64 cell partition. For the purposes of the design, this information was used to create a voltage source model of the panel as shown in Figure 3. $V_{32}$ and $V_{64}$ are the voltages across the 32 cell and 64 cell partitions, respectively. Using the open circuit voltage in the specifications above and assuming that each cell generates the same voltage, the expected voltages of the sources can be calculated: $V_{32} = 21.4$ V and $V_{64} = 42.8$ V.

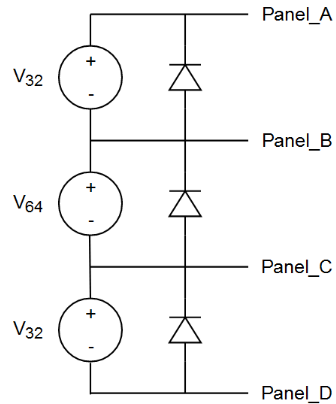



**Figure 3:** Solar Panel Equivalent Circuit

The possible output configurations chosen to satisfy the requirements are A-D (128 cells), B-C (64 cells), C-D (32 cells), or simply not connected at all. Table 2 summarizes the possible output configurations of the interface box and their expected open circuit voltages.

## 2.4 Switching Subsystem

After defining the four required output configurations of the interface box above, the implementation of the Switching Subsystem can be discussed. Page 2/5 of the schematic in the Appendix C shows the design of the Switching Subsystem. The primary switching components responsible for setting the output configuration are two double-pole double-

| Output Configuration | Number of Solar Cells | Expected $V_{oc}$ |
|---|---|---|
| A-D | 128 | 85.6 V |
| B-C | 64 | 42.8 V |
| C-D | 32 | 21.4 V |
| Not Connected (NC) | 0 | 0 V |

**Table 2:** Output configurations of the interface box and their expected open circuit voltages.

throw (DPDT) relays that are controlled by two I/O pins on the microcontroller. The choice of relays was determined by the maximum voltages and currents they would encounter in any mode of operation. The output of the Switching Subsystem (denoted by RELAY_OUT+ and INTERFACE_OUT-) will be connected to some combination of solar panel connections depending on which relays are actuated. For the purposes of ensuring that the solar panel is ground referenced when the output of the interface box is disconnected, PANEL_D is connected to GND.

## 2.5 Voltage/Current Monitoring Subsystem

Connecting directly to the output of the switching subsystem, the Voltage/Current Monitoring Subsystem is responsible for measuring the current delivered by the solar panel to the electrical load as well as the voltage across the three sources in Figure 3. The schematic of the Voltage/Current Monitoring Subsystem is shown in Page 3/5 of the schematic in the Appendix C. The subsystem has two outputs: one that communicates current/voltage data to the microcontroller via an I2C bus and another that passes power generated by the solar panel through the Switching Subsystem to the output of the interface box. The subsystem is powered by the 5V line from the Power Subsystem.

Because researchers are going to be using the solar panels, having more accurate measurements for voltages and currents will be beneficial given that costs are reasonable. Provided that the solar panels have an open circuit voltage of 85.6V and a short circuit current of 6.21A, it was determined that a 100mV and 100mA resolution on voltages and currents, respectively, would be suitable for this application; however, this was not a requirement given by the customer.

To achieve accurate voltage and current sensing and provide this information back to the microcontroller, a four channel 16-bit analog to digital converter (ADC) with an input range of 0-5V was utilized. This integrated circuit has an I2C interface which it can use to communicate the voltage at any of its channels to the microcontroller. Three of the four channels are used for voltage sensing and the fourth is for current sensing.

To sense voltages in a safe manner, the voltages present at each of the partitions of the solar panel need to be stepped down first. This was achieved using three sets of voltage dividers attached to PANEL_A, PANEL_B, and PANEL_C to ensure that the voltage at the input pins of the ADC were within the 0-5V range. The resistors of voltage dividers were chosen to be high impedance to minimize power dissipation in this subsystem. They were also chosen such that when the solar panel is open circuited, the output of the voltage dividers will be approximately 4.5V. Doing so allows for the detection of any rise in voltage and the opportunity to shut down the entire system if a failure condition occurs. Knowing the bit resolution of the ADC and the values of resistors in the voltage dividers, we can calculate the voltage resolution that can measured at the solar panel terminals for each ADC channel. Equation 1 shows the theoretical voltage resolution of the first channel after calibration. Equation 2 shows the theoretical voltage resolution of the second channel. Equation 3 shows the theoretical voltage resolution of the third channel. All of these are within 10mV accuracy.

$$\frac{5V}{2^{16}} \cdot \left(\frac{10k}{180k+10k}\right)^{-1} = 1.45mV \quad (1)$$

$$\frac{5V}{2^{16}} \cdot \left(\frac{10k}{135k+10k}\right)^{-1} = 1.12mV \quad (2)$$

$$\frac{5V}{2^{16}} \cdot \left(\frac{10k}{40k+10k}\right)^{-1} = 0.38mV \quad (3)$$

There were two primary topologies considered for the high-side current sensing portion of the design. The first used a low resistance shunt resistor in series with the primary current path. The second method is to use a hall effect current sensor. The latter was chosen because a shunt resistor would not be required, and they are more suitable for high-voltage current sensing such as in this application.

In the current sensor chosen for the design, the isolation is suitable up to 420 $V_{pk}$. The output voltage of this sensor (I_SENSE) rises linearly up to 5V as the sensed current rises from 0-20A with an overall sensitivity of 200 mV/A. Given these specifications we can calculate the current measurement resolution. This theoretical value after calibration is shown in Equation 4 and is within 10mA accuracy.

Any current flowing from the solar panel to the electrical load flows through the current sensor. The input current coming from the Switching Subsystem is given by

RELAY_OUT+ and the output of this subsystem is given by INTERFACE_OUT+. This is also the positive terminal of the power output of interface box. The negative terminal is given by INTERFACE_OUT- which was shown in the last section.

$$\frac{5V}{2^{16}} \cdot (200mV/A)^{-1} = 381.4\mu A \tag{4}$$

## 2.6 Thermocouples

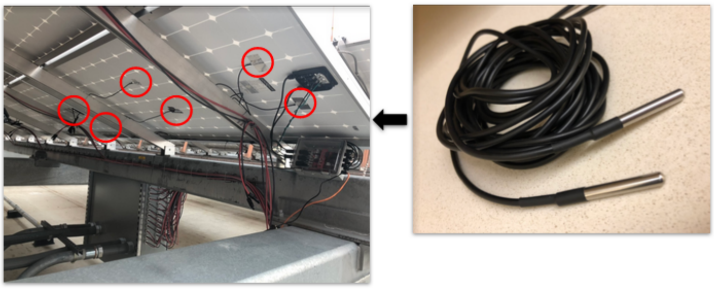


**Figure 4:** Temperature sensors mounted on solar panel

One of the most important sensors for safety are the thermocouples used to measure the temperature on various areas of the solar panel. Because these devices are external to the interface box, they are represented by headers on Page 3/5 on the schematic in Appendix C. This subsystem is powered by the 5V supply and only communicates with the Microcontroller Subsystem.

A critical issue that needed to be overcome in the design of the interface box was the amount of I/O pins available on the microcontroller of choice (ATmega328P). Due to the limited number of I/O pins available, a temperature that either used one digital I/O pin, I2C, or SPI was preferable. Ultimately, the DS18B20 temperature sensor was chosen because of its waterproof housing, and 1-Wire interface which only required one digital I/O pin to communicate with many sensors.

In regard to the temperature accuracy, these measurements are mostly used as a protective measure. If a general upward trend in the temperature of the panel is detected and it is increasing to knowingly damaging levels, the panel can be disconnected until it cools down enough for use again. Because of this, individual temperature measurements do not need to be extremely accurate. For this portion of the design, it was decided that an accuracy of ±5°F would be sufficient to detect any trends of overheating and shut off the system as necessary.

Keeping this in mind, the temperature sensor chosen was the DS18B20 which has a ±2.9°F temperature accuracy over a temperature range of 14°F-185°F [3], and this falls within the accuracy requirement. Even though this accuracy is only valid with a lower bound of 14°F, temperatures at this level are not damaging to the panel and do not pose a danger, so slight larger inaccuracies are tolerable at colder temperatures.

The chosen sensor is packaged in a waterproof housing at lengths of 3 meters with a 3-pin connector attached. In the schematic, all the data lines of the temperature sensor are tied together and are connected to the microcontroller through the T_SENSE net. A 4.7k pull up resistor was necessary as specified by the datasheet of the sensor. A total of six temperature sensors can be used in the system.

## 2.7 Microcontroller Subsystem

The Microcontroller Subsystem operates as the central processing unit in the interface box. This subsystem interfaces with every other subsystem and controls the general operation of the overall system. This subsystem is powered by the 5V voltage supply. Page 1/5 of the schematic in Appendix C shows the schematic of the Microcontroller Subsystem.

For the microcontroller itself, the ATmega328P was chosen because it is the same microcontroller used in an Arduino development board and contains all of the pins and interfaces needed to satisfy the requirements. Additionally, the Arduino programming language can used as well. This includes access to useful libraries used to communicate with other subsystems such as the Ethernet Interface and the temperature sensors.

Pin 15 of the microcontroller is used to communicate with the temperature sensors as discussed in the previous section. Pins 16-19 make up the SPI bus necessary for communicating with the Ethernet Interface. Pins 27 and 28 make up the I2C bus needed to communicate with the Voltage/Current Monitoring Subsystem. The device is being operated at a 16 MHz clock speed. An onboard tactile switch connected to pin 1 allows a user to reset the microcontroller if necessary.

On the left side of schematic, there is a 10-pin header used to connect the microcontroller to another board called the Manual Configuration Interface (discussed in the next section). Signals from this Manual Configuration Interface allow the Microcontroller Subsystem to detect when the user sets the Manual Configuration Switches and allows the subsystem to set the state of various status indicators. Another 6 pin header is connected to the microcontroller's serial data lines. This serves as a port used to flash firmware or monitor serial data depending on the needs of a developer.

## 2.8 Manual Configuration Interface

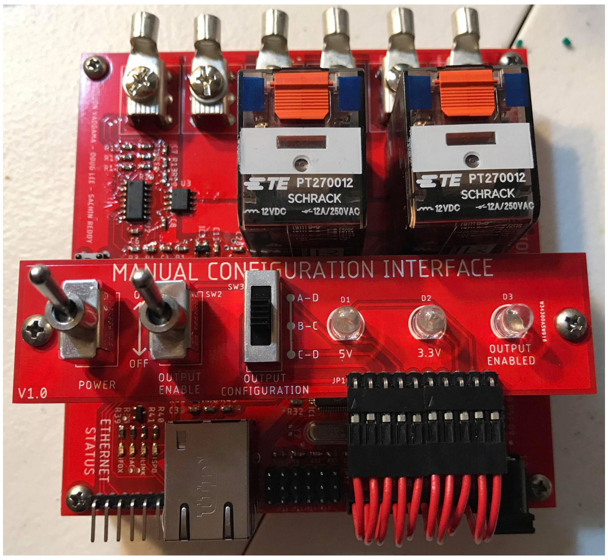



**Figure 5:** Manual Configuration Interface

The Manual Configuration Interface is a separate PCB mounted directly above the primary motherboard as shown in Figure 5. This PCB contains the components necessary for a user to control the interface box if they are physically close to where the interface box is installed on the solar panel. This is necessary in the case that the Ethernet connection to the box is disconnected or if users does not wish to use the associated graphical user interface.

The PCB itself has three switches that can be used to control the interface box. Referring to Figure 5, the left switch toggles the power supply being fed to the box. If this power supply is disconnected, the Switching Subsystem immediately goes into the disconnected state, and the solar panel cannot deliver power to the load. The rightmost switch is a three-position slide switch that can be used to set the output configuration of the interface box (AD, BC, or CD). After the user has set their desired output configuration, they can use the middle switch to toggle between the set output configuration and the disconnected state.

There are also several LED's placed in the system to indicate the current status. Near the Ethernet connector are four LED's to indicate Ethernet connection. On the Manual

Configuration Interface there are three LED's: two to indicate if the interface box is being properly powered, and another to indicate whether the output is enabled.

## 2.9 Ethernet Interface

The Ethernet Interface is responsible for facilitating communication between a server/PC and the micro-controller installed in the interface box. This is essential for allowing users to remotely control and monitor the interface box and the solar panel.

The interface was implemented using a WizNet W5500 IC [4]. This IC was chosen because it allows the system to be connected to the network, can easily connect to a microcontroller's SPI bus (which is available), and is the Ethernet controller used in the Arduino Ethernet Shield 2.[4] This means that any libraries already developed for this were accessible during the firmware implementation. The circuit of the Ethernet Interface can be found on Page 4/5 of the schematic in the Appendix C at the end of this document.

## 2.10 Power Subsystem

The power subsystem is responsible for generating all of the necessary supply rails for the interface box. In this system, 12V, 5V, and 3.3V are required for the various subsystems. The interface box as a whole is fed a 12V supply. As stated above, a switch in the Manual Configuration Interface can shut off this supply to the interface box. The overall system was estimated to draw less than 200mA when both relays of the Switching Subsystem are on.

The 3.3V and 5V voltage rails are generated using two LM1117 linear voltage regulators. The circuit for the Power Subsystem is shown in Page 5/5 of the schematic in Appendix C. Additionally, a series diode is present to prevent damage due to the reverse polarity at the 12V input. Each regulator also has an LED attached to indicate that their respective voltage rails are active.

# 3 Design Verification

## 3.1 Switching Subsystem

The Requirements and Verification table for this subsystem is shown in Appendix A. The relays are driven through a simple MOSFET control circuit. To avoid high currents being drawn through the 5V voltage rail, the 12 V source was chosen to actuate the relays; however, in this configuration, the microcontroller's digital I/O pins, connected through the RELAY_0 and RELAY_1 pins, can still control the state of the relays. When one of these I/O pins is HIGH, the N-channel MOSFET turns on allowing current to flow through the relay coil thereby actuating the device. Because relay coils are largely inductive, sudden changes in current can lead to voltage spikes that can potentially damage the MOSFET. To diminish this effect, a Schottky diode was placed in parallel to the relay coil such that it is reverse biased when the relay is on and forward biased immediately after turning off the MOSFET. The pull down resistor at the gate of the MOSFET ensures that even if the microcontroller turns off, the relays will remain off and the output disconnected. Table 3 summarizes how changes in RELAY_0 and RELAY_1 affect the output of the switching subsystem. Note, this encompasses all of the required output configurations of the interface box.

To verify the functionality of this subsystem, the microcontroller was set to toggle the RELAY_0 and RELAY_1 pins and a multi-meter was used to ensure that the correct connections were being made according to Table 3.

| RELAY_0 | RELAY_1 | RELAY_OUT+ | INTERFACE_OUT- |
|---|---|---|---|
| FLOATING | FLOATING | PANEL_B | NC |
| LOW | LOW | PANEL_B | NC |
| LOW | HIGH | PANEL_C | PANEL_D |
| HIGH | LOW | PANEL_B | PANEL_C |
| HIGH | HIGH | PANEL_A | PANEL_D |

**Table 3:** Relationship between the RELAY_0 and RELAY_1 inputs controlled by the microcontroller and the corresponding output configuration. LOW is 0V and HIGH is 5V.

To ensure that the voltage spikes generated by suddenly changing the current through the relay coil were not damaging to the MOSFET, the driver circuitry was simulated in LTspice. Figure 6 shows the circuit simulated. The relay coil was modeled as an inductor in series with a resistor. These values were chosen based on the datasheet of the relay. Figure 7 shows the drain voltage when the relay is turned off. This voltage spike is less than 13 V and is suitable for the MOSFET chosen.

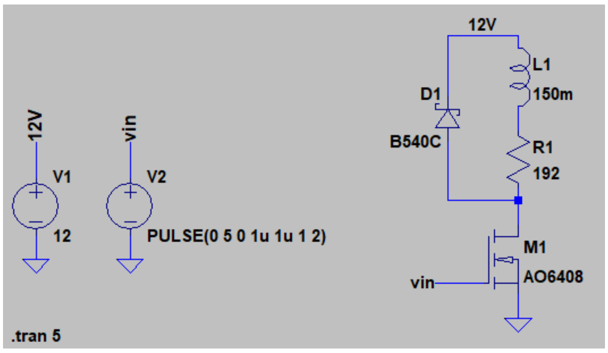



**Figure 6:** Simulation of relay driver circuit.

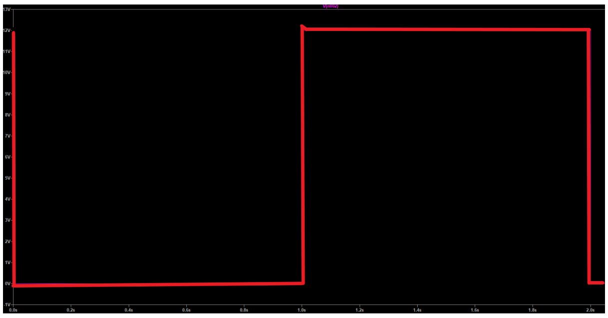


**Figure 7:** Driver circuit simulation results. Voltage spikes are less than 13 V which is suitable for the chosen MOSFET.

Although the simulation showed the feasibility of the design, a real-world test was carried out using the same components in the final version of the design. While turning on and off the relays, the drain-source voltage of the MOSFET was monitored with an oscilloscope as shown in Figure 8. Doing so showed that the maximum voltage across the MOSFET was less than 16V, which was suitable for use in the design.

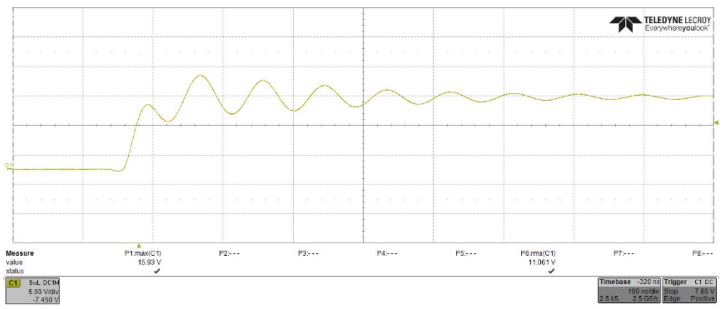



**Figure 8:** The drain to source voltage of the MOSFET was shown be less than 16V.

## 3.2 Voltage/Current Monitoring Subsystem

The Requirements and Verification table for this subsystem is shown in the Appendix A. The verification of the Voltage/Current Monitoring Subsystem was carried out in several tests. First, in order to ensure that both the I2C data bus and the ADC were functional, an address command from the microcontroller was sent on the bus using the address of the ADC. The IC was successfully able to acknowledge that it received the message, thereby showing that both the IC and the data bus were functional.

Second, three voltage sources were connected in series as shown in Figure 3 and connected to the solar panel connections of the interface box. Using firmware written for the microcontroller, the voltages read by the ADC were able to be captured and compared to a multimeter reading. Overall, the interface box is capable of reading voltages with an accuracy of $\pm 100mV$ prior to calibration. This accuracy would improve and move closer to the theoretical values discussed earlier with calibration completed in software. Again, the voltage measurement accuracy was not a requirement given by the customer.

Lastly, a constant current was sent through the Switching Subsystem to an electrical load. While doing so, the microcontroller was setup to read the current being delivered using the information from both the ADC and the hall effect current sensor. Current measurements were determined to be within $\pm 150mV$ prior to calibration. Again, a software calibration would bring accuracy levels higher.

## 3.3 Thermocouples

```
Locating devices...Found 1 devices.
Found device 0 with address: 282ABDE90A000074
Temperature for device: 0
Temp C: 23.25 Temp F: 73.85
Temperature for device: 0
```

**Figure 9:** Single DS18B20 Temperature Sensor Test Outputs

```
Locating devices...Found 2 devices.
Found device 0 with address: 282ABDE90A000074
Found device 1 with address: 28AA0DE852140189
Temperature for device: 0
Temp C: 23.69 Temp F: 74.64
Temperature for device: 1
Temp C: 22.31 Temp F: 72.16
```

**Figure 10:** Multiple DS18B20 Temperature Sensor Test Outputs

Figure 9 and Figure 10 display successful results of tests with the DS18B20 temperature sensors connected to an Arduino development board. In Figure 9, a single temperature sensor was tested. In Figure 10, to test the "One-Wire" configuration, two temperature sensors were connected to one digital I/O pin. In total, five sensors were attached to the Arduino and tested simultaneously. Each measurement was within the tolerance in the Design section.

## 3.4 Microcontroller Subsystem

Verifying the Microcontroller Subsystem was completed by ensuring the microcontroller could be programmed and by successfully verifying the functionality of other subsystems. A USB-Serial converter was attached to the programming header on the primary PCB, and the microcontroller was loaded with a Blink program that could toggle an LED every second. Being able to run this program proved that the microcontroller could be programmed successfully. By successfully testing the other subsystems of the design with this microcontroller, all of the requirements of this subsystem were also met.

## 3.5 LED‘s

The LED’s were tested using the microcontroller loaded with a Blink program that toggled the LED’s every second. After it was verified that the LED’s were blinking every second, the LED’s were determined to be functional.

## 3.6 Manual Configuration Switches

The SPDT switches used in the Manual Configuration Interface by connecting the common terminal to a digital I/O pin of the microcontroller. The other two terminals were connected to 5V and GND, respectively. The microcontroller was setup to read where the output of the switches was HIGH or LOW. After ensuring that one position of the switch corresponded to the opposite state of the other position, the switches were determined to be functional.

## 3.7 Ethernet Interface

Testing the Ethernet Interface involved showing that the subsystem was able to receive data from the microcontroller through the SPI bus and then send that data to an external client over a network. To carry out the test, an Ethernet cable was connected from a router to the interface box. The microconroller was setup to read temperature data and send the data over the SPI bus to the Ethernet Interface. After ensuring that a remote client was able to access this data, the Ethernet Interface was proved to be functional.

Additional tests were done by including the temperature monitoring subsystem. Temperature data was retrieved by the microcontroller and sent to the Ethernet Interface, and finally to the client. This proved to work as well. After ordering the PCB's, no issues were experienced, and the Ethernet Interface was fully functional.

## 3.8 Power Subsystem

The Requirements and Verification table for this subsystem is shown in Appendix A. Verifying the Power Subsystem involved ensuring that the chosen voltage regulators worked correctly. The interface box is connected to an isolated 12V supply, while subsystems within the interface box use input voltages of 5V and 3.3V. Overall, the interface box draws approximately 0.150A during peak usage (i.e. when both relays of the Switching Subsystem are on). Verifying the 5V regulator under no load included connecting the 12V input and checking the output was between 4.5V and 5.5V. A programmable load was then attached to the output of the regulator that was set to 800mA. Another check was done to ensure the output voltage was between 4.5V and 5.5V.

A similar process was done to validate the 3.3V regulator was working as required. Under no load, 12V was attached to the input of the regulator and the output was measured to be between 3.0V and 3.6V. A programmable load was then attached to the output of the regulator that was set to 800mA. Another measurement was made to ensure the output voltage was between 3.0V and 3.6V.

# 4 Costs

## 4.1 Labor Costs

Our fixed development costs are estimated to be \$33.75 an hour assuming the average annual salary is \$65,000 per year, 8 hours a week for three people.

$$3 \cdot \left( \frac{\$33.75}{hr} \right) \cdot 10 weeks \cdot 2.5 = \$20,250 \tag{5}$$

## 4.2 Parts Costs

The itemized list of all the parts and materials required for fabricating this design is listed in Appendix B.

## 4.3 Total Cost

| | |
|---|---|
| Labor | \$20,250 |
| Parts | \$142.41 |
| **Grand Total** | \$20,392.41 |

**Table 4:** Grand Total

# 5 Ethics and Safety

The user of this monitoring device will be directly involved with the operation of the device; thus, it is important that we ensure a safe and reliable product. There must be safeguards in place to protect both the product and the user. For instance, if a user manually shuts off the device working to work on the solar panels, commands and controls from software do not override the box to turn it back on.

The one major component in the device that could be considered hazardous is the electrical circuitry when connected to the solar panel. The open circuit voltage of the solar panels is 85.6 VDC and sufficient actions should be taken to protect the user from coming into contact with these voltage levels. Since this is a device that is mounted in an outdoor environment, sufficient isolation was utilized to protect the internal circuitry and to prevent harm to the user.

In addition to weather-proofing the device, we must be able to allow for reliable circuitry to maintain the monitoring system notify the user if a failure condition has been met or close to being met. Notifying the user of this event and shutting down the operation will ensure that a hot-spot disaster will not occur. This then complies with the IEEE ethics code #1: 'hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment'[6].

As testing and user-feedback comes in, we must take all problems into consideration and make adjustments to the device with no hesitance. Since this device is meant to prevent a potentially dangerous situation from unfolding, all constructive criticism will be considered for future revisions of the design. This adheres to the IEEE ethics code #7: 'to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others'[6].

# 6 Conclusions

## 6.1 Accomplishments

Overall, the project was a success and we were able to mount our prototype board on one of the active solar panels on the roof of ECEB. With this, we were able to monitor incoming voltage, current and temperature data and test the failure conditions using our own developed graphical user interface shown in Figure 11 in Appendix C.

Professor Banerjee was very impressed by our efforts and has tentative plans to build 60 interface boxes for Spring 2020. There are also plans to iterate over the hardware and software based on the team's input.

## 6.2 Future Work

### 6.2.1 Hardware

There are several improvements that could be made to the hardware of this design in future iterations of the PCB. In order to have the design be more robust, operate at more stressful conditions, and last longer in the field, higher rated relays should be used in the switching subsystem. If more funding was allotted, the switching subsystem could replace the relays of the switching subsystem with a solid-state design instead. This would allow the power consumption of the box itself to drop by approximately 70%.

In the Microcontroller Subsystem, there are two updates that could be made to improve the overall system. One issue that was prominent toward the end of the course was that the memory usage of the ATmega328p was nearing 100%. Being near the limit of memory usage can have serious consequences including causing instability issues. By using a microcontroller with a large amount of static memory, this issue could be prevented. If the microcontroller were replaced altogether with a processor, multiple threads could be used to make the firmware more efficient.

Physically on the enclosure, a minor improvement that could be made is the polarity of the output MC4 connectors. On the roof of ECEB, each solar panel has a wire connection to the inside of the building. While testing on the roof, it was noticed that this connection had a reversed polarity relative to the interface box. Flipping the connectors on the box would mitigate this minor issue.

### 6.2.2 Firmware

As for the firmware, several features could be implemented and improved upon. The first and foremost being the optimization of the code and libraries. This would reduce the memory cost placed on the ATmega328p's on-chip memory.

Additionally, a power optimization feature could be implemented in order to reduce the power consumption of our device. The power optimization feature would include three states: Active, Idle, and Sleep. The active state would be checking the voltage, current, and temperature at a higher rate. The Idle state would allow preset timing intervals of the check function of the system parameters to cycle at a slower rate than the active state. Finally, the Sleep state would stop all system functions under certain conditions (e.g.During the night). This feature can be achieved through another additional feature of utilizing multi-threading and interrupt applications [5].

### 6.2.3 Software

As more protection systems are deployed, better visualizations of the grid can be added. This would include a separate frame to display all the active interfaces on the network. The current list of connections does not have a scrolling feature, so if more than 10 interfaces are on the network, it is impossible to access their data.

Also, having a user authentication method, so that only specific users can access the application should be added. This would increase the security of our design.

Currently, the application uses SQLite, which is a local database. Although being easy to interface with, it does not allow for data to persist between devices. For example, if one accessed the monitored data from one computer, then checked the monitored data from another, the monitored data will not continue from the first device.

In summary, there is a lot of room for improvement given the amount of data one will collect while running the application.

## CITATIONS AND REFERENCES


[1] D. Yaffe-bellany and I. Penn, "Walmart Says Tesla Solar Panels Set Fires Atop Stores," The New York Times, 20-Aug-2019. [Online].
Available: https://www.nytimes.com/2019/08/20/business/walmart-tesla-lawsuit-fires.html. [Accessed: 04-Oct-2019].

[2] SPR-425E-WHT-D Solar Panel from SunPower Posharp, Inc. [online].
Available at: http://www.posharp.com/photovoltaic/solarpanel.aspx?pid=32ceb0e4-27f3-47f6-a60a-935cfd768488 [Accessed Oct 1 2019]

[3] DS18B20 Programmable Resolution 1-Wire Digital Thermometer. Dallas Semiconductor [online], pp. 1-27.
Available at:https://cdn.sparkfun.com/datasheets/Sensors/Temp/DS18B20.pdf [Accessed Oct 3. 2019]

[4] Store.arduino.cc. (2019). Arduino Ethernet Shield 2. [online]
Available at:https://store.arduino.cc/usa/arduino-ethernet-shield-2 [Accessed Oct 3. 2019]

[5] Arduino - Interrupts [online]
https://www.tutorialspoint.com/arduino/arduino_interrupts.htm [Accessed Oct 2. 2019]

[6] "IEEE Code of Ethics." Institute of Electrical and Electronics Engineers. 2016. [online]. https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed Sept 6. 2019]

# Appendix A

## Appendix A: Requirements and Verification Tables

**Table 5:** Switching Subsystem

| Requirements | Verification |
|---|---|
| The Switching Subsystem will output all four configurations outlined in Table 2. | · Toggle RELAY_0 and RELAY_1 according to each row of Table 3.<br>· For each combination of the control line inputs, use a multimeter to ensure that both INTERFACE_OUT- and RELAY_OUT+ are connected to the appropriate nodes as defined by Table 3. |
| The default state of the Switching Subsystem output will be not connected. | · Connect three voltage sources in series and connect them to the four panel input connections.<br>· Ensure that when RELAY_0 and RELAY_1 are left floating that the output is 0 V. |
| Toggling the relays will not produce voltage spikes across the MOSFET greater than 30V. | · Monitor the voltage across the drain and source of the MOSFET with an oscilloscope.<br>· Ensure that when the control input is toggled that the voltage across the MOSFET does not exceed 30V. |

**Table 6:** Voltage/Current Monitoring Subsystem

| Requirements | Verification |
|---|---|
| The subsystem will accurately measure the voltages across the solar panel terminals to the resolutions determined in Equation 1, Equation 2, and Equation 3. This information will be sent over an I2C interface. | · Connect the Microcontroller Subsystem such that it can communicate with devices on the I2C bus.<br>· Connect three voltage sources to PANEL_A, PANEL_B, and PANEL_C.<br>· Run firmware to allow the microcontroller to read the voltages at Channel 1, Channel 2, and Channel 3 and extrapolate the voltages of the three sources.<br>· Verify measurements are within the resolutions determined in Equation 1, Equation 2, and Equation 3. |
| The Voltage/Current Monitoring subsystem will accurately measure the current delivered to the electrical load connected to the output of the interface box to the resolution determined in Equation 4. This information will be sent over an I2C interface. | · Connect the Microcontroller Subsystem such that it can communicate with devices on the I2C bus.<br>· Connect a current source across RELAY_OUT+ and INTERFACE_OUT+.<br>· Run firmware to allow the microcontroller to read the voltage at Channel 4 of the ADC and extrapolate the current value.<br>· Verify measurements are within the resolution determined in Equation 4. |

**Table 7:** Temperature Monitoring Subsystem

| Requirements | Verification |
|---|---|
| The Temperature Monitoring Subsystem will measure temperatures to an accuracy of ±5° F. | · Attach more than one temperature sensor to the Temperature Monitoring Subsystem.<br>· Connect the Microcontroller Subsystem such that T_SENSE is connected to a PWM enabled digital I/O pin.<br>· Run firmware on the microcontroller to read the data coming from each sensor using the 1-wire interface.<br>· Verify that the temperature given by each sensor is within ±5° F of the actual ambient temperature. |

**Table 8:** Microcontroller Subsystem

| Requirements | Verification |
|---|---|
| The microcontroller will have at least four digital inputs capable of distinguishing <1V as a logical LOW and >4V as a logical HIGH. | · Connect an SP3T switch to JP5, and an SPST switch to JP6.<br>· Run firmware to read the states of these switches.<br>· Verify the states change as each switch changes position. |
| The microcontroller will have at least four digital outputs capable of toggling between 0 and 5V at a rate greater than 10 Hz. | · Attach an oscilloscope to a digital output pin.<br>· Run firmware to toggle the output at 10 Hz.<br>· Verify the output alternates between 0 and 5V at 10 Hz.<br>· Repeat for each digital output. |
| The microcontroller will have at least one digital PWM output capable of communicating over a 1-Wire interface. | · Attach the Temperature Monitoring Subsystem.<br>· Run firmware to read temperature data through the PWM pin.<br>· Verify that the temperature data is accurate. |
| The microcontroller will be able to communicate over a standard I2C bus. | · Connect the Voltage/Current Monitoring system to the microcontroller's I2C lines.<br>· Run the i2cScanner sketch on the microcontroller.<br>· Verify the address received is 0x68. |
| The microcontroller will be able to communicate over a standard SPI bus. | · Attach a USB-SPI bridge to the ICSP header and connect it to a computer.<br>· Run firmware to read data on the SPI bus and echo it back.<br>· From a terminal, send data on the SPI bus and verify the data sent back is the same. |

**Table 9:** LEDs

| Requirements | Verification |
|---|---|
| The LED's can be powered from a 5V supply and/or toggled with the Microcontroller Subsystem | · Connect the LED to the micro-controller with a digital output pin.<br>· Run the 'Blink' example program and verify the LED blinks at 0.5 Hz. |

**Table 10:** Manual Configuration Switches

| Requirements | Verification |
|---|---|
| The voltage outputs of this subsystem must be readable by the Microcontroller Subsystem | · Connect an SP3T switch to JP5.<br>· Connect an SPST switch to JP6.<br>· Verify that in any combination of switch positions, the output voltages are either <1V or >4V. |
| Each combination of positions on the manual configuration switches must correspond to a unique state of switch outputs. | · Connect an SP3T switch to JP5.<br>· Connect an SPST switch to JP6.<br>· Measure the voltages on the four outputs of the subsystem.<br>· Verify that changing any switch position will change at most two outputs from HIGH to LOW or vice versa. |

**Table 11:** Ethernet Interface

| Requirements | Verification |
|---|---|
| The subsystem must be able to receive data from the microcontroller through SPI and send it over a connected network. Similarly, the subsystem must be able receive data over a network and send it to microcontroller through the SPI bus. | · Connect the Microcontroller Subsystem to the Ethernet Interface through the SPI bus.<br>· Connect the Ethernet Interface to a network using an Ethernet cable.<br>· Connect a PC to the network. On the microcontroller, run the UDPSendReceiveString sketch.<br>· From the remote PC, send data over the network to the Ethernet Interface.<br>· Verify that a response is received saying "acknowledged." |

**Table 12:** Power Subsystem

| Requirements | Verification |
|---|---|
| The subsystem will convert a 12V input to a 4.5-5.5V rail that remains within range with load currents of 0-800mA. | · Measure the voltage output of the 5V regulator under no load.<br>· Verify it is between 4.5-5.5V.<br>· Attach a programmable load to the output of the regulator that is set to 800 mA.<br>· Verify the output voltage is between 4.5-5.5V. |
| The subsystem will convert a 12V input to a 3.0-3.6V rail that remains within range with load currents of 0-800mA. | · Measure the voltage output of the 3.3V regulator under no load.<br>· Verify it is between 3.0-3.6V.<br>· Attach a programmable load to the output of the regulator that is set to 800 mA.<br>· Verify the output voltage is between 3.0-3.6V. |
| The subsystem does not generate any voltage rails when the 12V power switch is off. | · Attach an SPST switch to JP1.<br>· Make sure the switch is opened.<br>· Measure the voltages at the 5V and 3.3V output.<br>· Verify they are both 0V. |

# Appendix B

## Appendix B: Parts Cost

| Description | Vendor | Vendor Part Number | Manufacturer Part Number | Quantity | Unit Price | Extended Price USD |
|---|---|---|---|---|---|---|
| CONN MAGJACK 1PORT 100 BASE-T | Digikey | 1278-1011-ND | RB1-125BAG1A | 1 | $2.55 | $2.55 |
| CRYSTAL 16.0000MHZ 18PF T/H | Digikey | 887-1244-ND | 9B-16.000MEEJ-B | 1 | $0.48 | $0.48 |
| TERM SCREW 6-32 6 PIN PCB | Digikey | 36-8197-ND | 8197 | 6 | $0.52 | $3.12 |
| RELAY SOCKET 8 POS THROUGH HOLE | Digikey | PB807-ND | 2.70E+221 | 2 | $2.16 | $4.32 |
| SWITCH TACTILE SPST-NO 0.05A 24V | Digikey | 450-1650-ND | 1825910-6 | 1 | $0.10 | $0.10 |
| CAP ALUM 47UF 20% 25V SMD | Digikey | 493-9423-1-ND | UCW1E470MCL1GS | 3 | $0.51 | $1.53 |
| IC BUF NON-INVERT 5.5V SC70-5 | Digikey | 296-11604-1-ND | SN74LVC1G125DCKR | 2 | $0.30 | $0.60 |
| MOSFET P-CH 20V 3.5A SOT23 | Digikey | 1727-5907-1-ND | PMV48XP,215 | 1 | $0.38 | $0.38 |
| FERRITE BEAD 30 OHM 0805 1LN | Digikey | MH2029-300YCT-ND | MH2029-300Y | 2 | $0.10 | $0.20 |
| IC ADC 16BIT SIGMA-DELTA 14SOIC | Digikey | MCP3428-E/SL-ND | MCP3428-E/SL | 1 | $3.56 | $3.56 |
| IC SUPERVISOR 1 CHANNEL SOT143 | Digikey | CAT811TTBI-GT3OSCT-ND | CAT811TTBI-GT3 | 1 | $0.43 | $0.43 |
| IC REG LIN POS ADJ 800MA SOT223 | Digikey | LM1117IMPX-ADJ/NOPBCT-ND | LM1117IMPX-ADJ/NOPB | 2 | $1.10 | $2.20 |
| SENSOR CURRENT HALL 20A DC | Digikey | 620-1645-1-ND | ACS723LLCTR-20AU-T | 1 | $5.27 | $5.27 |
| RES SMD 715 OHM 1% 1/10W 0603 | Digikey | 311-715HRCT-ND | RC0603FR-07715RL | 1 | $0.02 | $0.02 |
| RES SMD 470 OHM 1% 1/10W 0603 | Digikey | 311-470LDCT-ND | AC0603FR-07470RL | 4 | $0.03 | $0.11 |
| MOSFET N-CH 30V 1.4A SSOT3 | Digikey | NDS351ANCT-ND | NDS351AN | 2 | $0.40 | $0.80 |
| RES SMD 150K OHM 1% 1/10W 0603 | Digikey | YAG3567CT-ND | AC0603FR-07150KL | 1 | $0.03 | $0.03 |
| RES SMD 100K OHM 1% 1/10W 0603 | Digikey | 311-100KLDCT-ND | AC0603FR-07100KL | 1 | $0.03 | $0.03 |
| CAP CER 4.7UF 16V X5R 0603 | Digikey | 1276-1784-1-ND | CL10A475KO8NNNC | 1 | $0.26 | $0.26 |
| RES SMD 49.9 OHM 1% 1/10W 0603 | Digikey | YAG3611CT-ND | AC0603FR-0749R9L | 2 | $0.03 | $0.06 |
| RES SMD 4.7K OHM 1% 1/10W 0603 | Digikey | YAG3613CT-ND | AC0603FR-074K7L | 1 | $0.03 | $0.03 |
| RES SMD 390 OHM 1% 1/10W 0603 | Digikey | 311-390HRCT-ND | RC0603FR-07390RL | 1 | $0.02 | $0.02 |
| CRYSTAL 25.0000MHZ 18PF T/H | Digikey | 887-1253-ND | 9B-25.000MEEJ-B | 1 | $0.48 | $0.48 |
| DIODE SCHOTTKY 40V 3A SMA | Digikey | B340A-FDICT-ND | B340A-13-F | 3 | $0.42 | $1.26 |
| RES SMD 22 OHM 1% 1/10W 0603 | Digikey | YAG3581CT-ND | AC0603FR-0722RL | 1 | $0.03 | $0.03 |
| RES SMD 240 OHM 1% 1/10W 0603 | Digikey | YAG3582CT-ND | AC0603FR-07240RL | 2 | $0.03 | $0.06 |
| CAP CER 1000PF 25V X7R 0603 | Digikey | 311-3994-1-ND | CC0603JRX7R8BB102 | 1 | $0.06 | $0.06 |
| CAP CER 1UF 16V X7R 0603 | Digikey | 587-3305-1-ND | EMK107B7105KAHT | 4 | $0.11 | $0.44 |
| RES SMD 1K OHM 1% 1/10W 0603 | Digikey | 311-1KLDCT-ND | AC0603FR-071KL | 3 | $0.03 | $0.08 |
| RES SMD 1M OHM 1% 1/10W 0603 | Digikey | 311-1MLDCT-ND | AC0603FR-071ML | 2 | $0.03 | $0.06 |
| CAP CER 18PF 50V NPO 0603 | Digikey | 311-3889-1-ND | CC0603FRNPO9BN180 | 4 | $0.12 | $0.50 |
| RES SMD 12.4K OHM 1% 1/10W 0603 | Digikey | 311-12.4KHRCT-ND | RC0603FR-0712K4L | 1 | $0.02 | $0.02 |
| RES 133K OHM 1% 1/10W 0603 | Digikey | RMCF0603FT133KCT-ND | RMCF0603FT133K | 1 | $0.02 | $0.02 |
| CAP CER 10UF 25V X5R 0805 | Digikey | 587-2985-1-ND | TMK212BBJ106KG-T | 3 | $0.23 | $0.69 |
| RES 422K OHM 1% 1/10W 0603 | Digikey | RMCF0603FT422KCT-ND | RMCF0603FT422K | 1 | $0.02 | $0.02 |
| CAP CER 10000PF 25V X7R 0603 | Digikey | 311-3995-1-ND | CC0603JRX7R8BB103 | 8 | $0.07 | $0.58 |
| IC CTLR 3-1 ETH TCP/IP 48LQFP | Digikey | 1278-1021-ND | W5500 | 1 | $3.95 | $3.95 |
| LED GREEN CLEAR 0603 SMD | Digikey | 732-4980-1-ND | 150060VS75000 | 7 | $0.14 | $0.98 |
| RES 560K OHM 1% 1/10W 0603 | Digikey | RMCF0603FG560KCT-ND | RMCF0603FG560K | 1 | $0.02 | $0.02 |
| CAP CER 0.1UF 50V X7R 0603 | Digikey | 311-1344-1-ND | CC0603KRX7R9BB104 | 11 | $0.06 | $0.68 |
| RES SMD 10K OHM 1% 1/10W 0603 | Digikey | 311-10KLMCT-ND | AF0603FR-0710KL | 11 | $0.06 | $0.65 |
| SWITCH TOGGLE SPDT 5A 120V | Digikey | EG2355-ND | 100SP1T1B4M2QE | 2 | $2.90 | $5.80 |
| SWITCH SLIDE SP3T 200MA 30V | Digikey | CKN9553-ND | OS103012MU2QP1 | 1 | $0.73 | $0.73 |
| CONN PWR JACK 2X5.5MM SOLDER | Digikey | CP-037A-ND | PJ-037A | 1 | $0.58 | $0.58 |
| CONN PWR PLUG 2.1X5.5MM SOLDER | Digikey | EP501A-ND | EP501A | 1 | $1.45 | $1.45 |
| LED GREEN CLEAR T-1 3/4 T/H | Digikey | 754-1273-ND | WP7113SGC | 3 | $0.49 | $1.47 |
| CONN RING CIRC 10-14AWG M4 CRIMP | Digikey | 277-11157-ND | 3240083 | 6 | $0.21 | $1.28 |
| MACH SCREW PAN PHILLIPS M4X0.7 | Digikey | 335-1147-ND | RM4X6MM 2701 | 8 | $0.48 | $3.80 |
| HEX STANDOFF #4-40 ALUMINUM 1/2" | Digikey | 1772-1098-ND | 4505-440-AL | 4 | $0.47 | $1.88 |
| HEX STANDOFF #4-40 ALUMINUM 1" | Digikey | 36-1897-ND | 1897 | 4 | $0.75 | $3.00 |
| MACHINE SCREW PAN PHILLIPS 4-40 | Digikey | 36-9900-ND | 9900 | 2 | $0.10 | $0.20 |
| HEX NUT 0.184" STN STEEL 4-40 | Digikey | 36-7248-3-ND | 7248-3 | 4 | $0.15 | $0.60 |
| RELAY GEN PURPOSE DPDT 12A 12V | Digikey | PB908-ND | PT270012 | 2 | $7.73 | $15.46 |
| IC MCU 8BIT 32KB FLASH 28DIP | Digikey | ATMEGA328P-PU-ND | ATMEGA328P-PU | 1 | $2.14 | $2.14 |
| RELAY SOCKET 8 POS THROUGH HOLE | Digikey | PB807-ND | _27E220 | 2 | $2.16 | $4.32 |
| Weather Proof Enclosure | Amazon | - | - | 1 | $30.99 | $30.99 |
| Primary PCB | PCBWay | - | - | 1 | $4.90 | $4.90 |
| Manual Configuration PCB | PCBWay | - | - | 1 | $4.90 | $4.90 |
| Cable Glands | Amazon | - | - | 8 | $0.40 | $3.20 |
| MC4 Connectors | Amazon | - | - | 6 | $0.38 | $2.27 |
| DS18B20 Temperature Sensors | | | | 6 | $2.80 | $16.79 |
| **Total** | | | | | | $142.41 |

# Appendix C

## Appendix C: Large Figures and Circuit Schematics

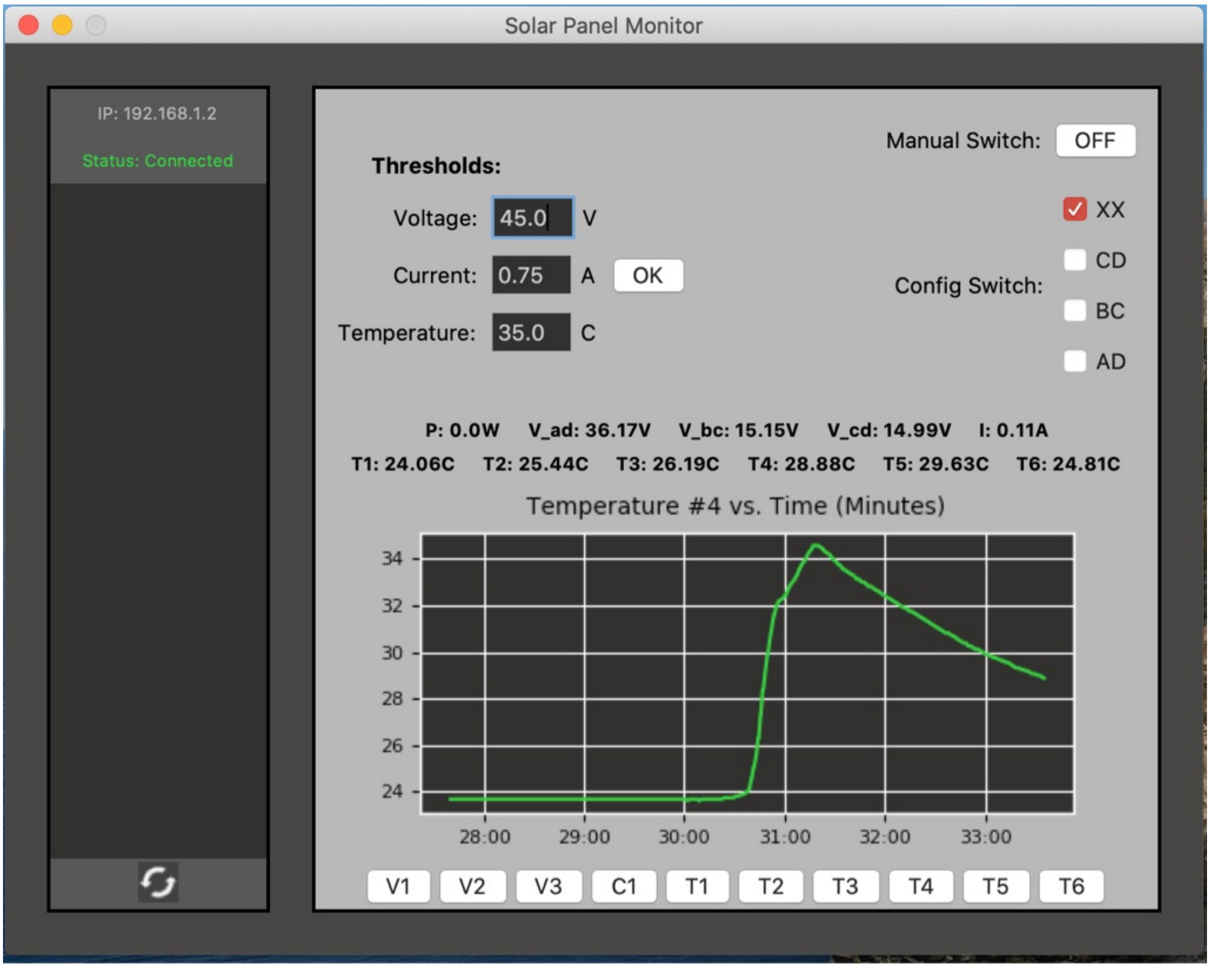
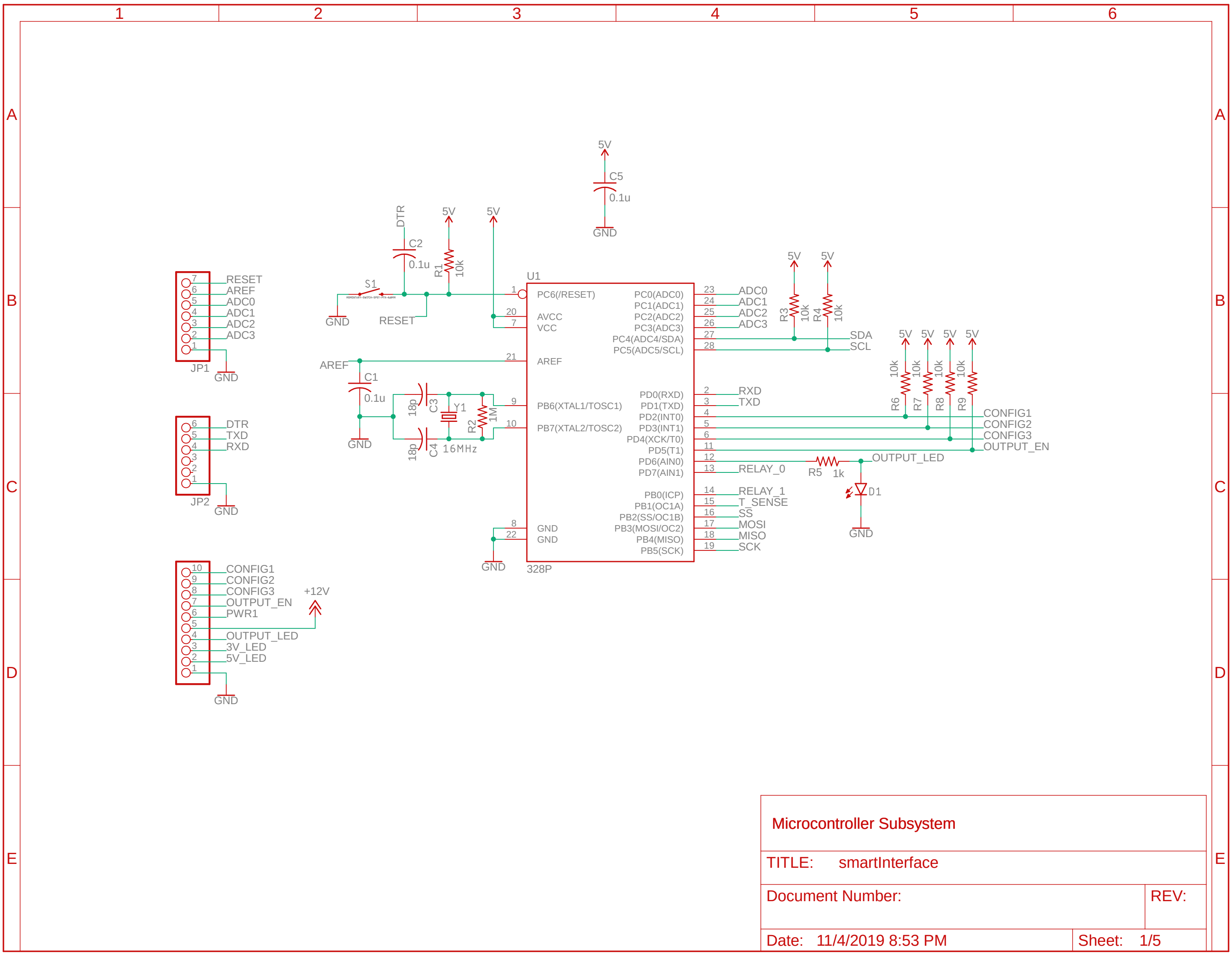
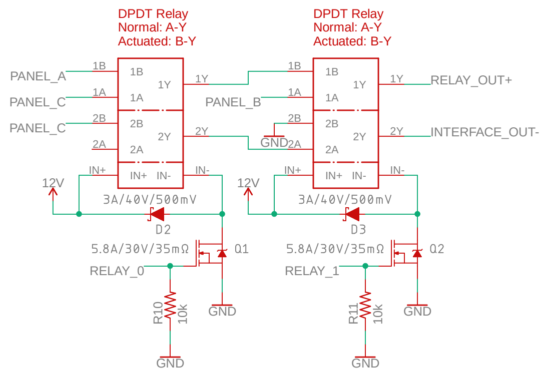



**Figure 11:** Final GUI

**Figure 12:** Weatherproof Enclosure for Interface Box: MC4 Connectors

**Figure 13:** Weatherproof Enclosure for Interface Box: Cable Glands


Microcontroller Subsystem
TITLE: smartInterface
Document Number:
REV:
Date: 11/4/2019 8:53 PM
Sheet: 1/5
U1
328P
JP1
JP2
RESET
AREF
ADC0
ADC1
ADC2
ADC3
DTR
TXD
RXD
CONFIG1
CONFIG2
CONFIG3
OUTPUT_EN
PWR1
OUTPUT_LED
3V_LED
5V_LED
+12V
GND
5V
C1
C2
C5
0.1u
C3
C4
18p
Y1
16MHz
R1
R2
1M
R3
R4
R5
1k
R6
R7
R8
R9
10k
D1
S1
SDA
SCL
RELAY_0
RELAY_1
T_SENSE
SS
MOSI
MISO
SCK
PC6(/RESET)
AVCC
VCC
AREF
PB6(XTAL1/TOSC1)
PB7(XTAL2/TOSC2)
PC0(ADC0)
PC1(ADC1)
PC2(ADC2)
PC3(ADC3)
PC4(ADC4/SDA)
PC5(ADC5/SCL)
PD0(RXD)
PD1(TXD)
PD2(INT0)
PD3(INT1)
PD4(XCK/T0)
PD5(T1)
PD6(AIN0)
PD7(AIN1)
PB0(ICP)
PB1(OC1A)
PB2(SS/OC1B)
PB3(MOSI/OC2)
PB4(MISO)
PB5(SCK)




Switching Subsystem

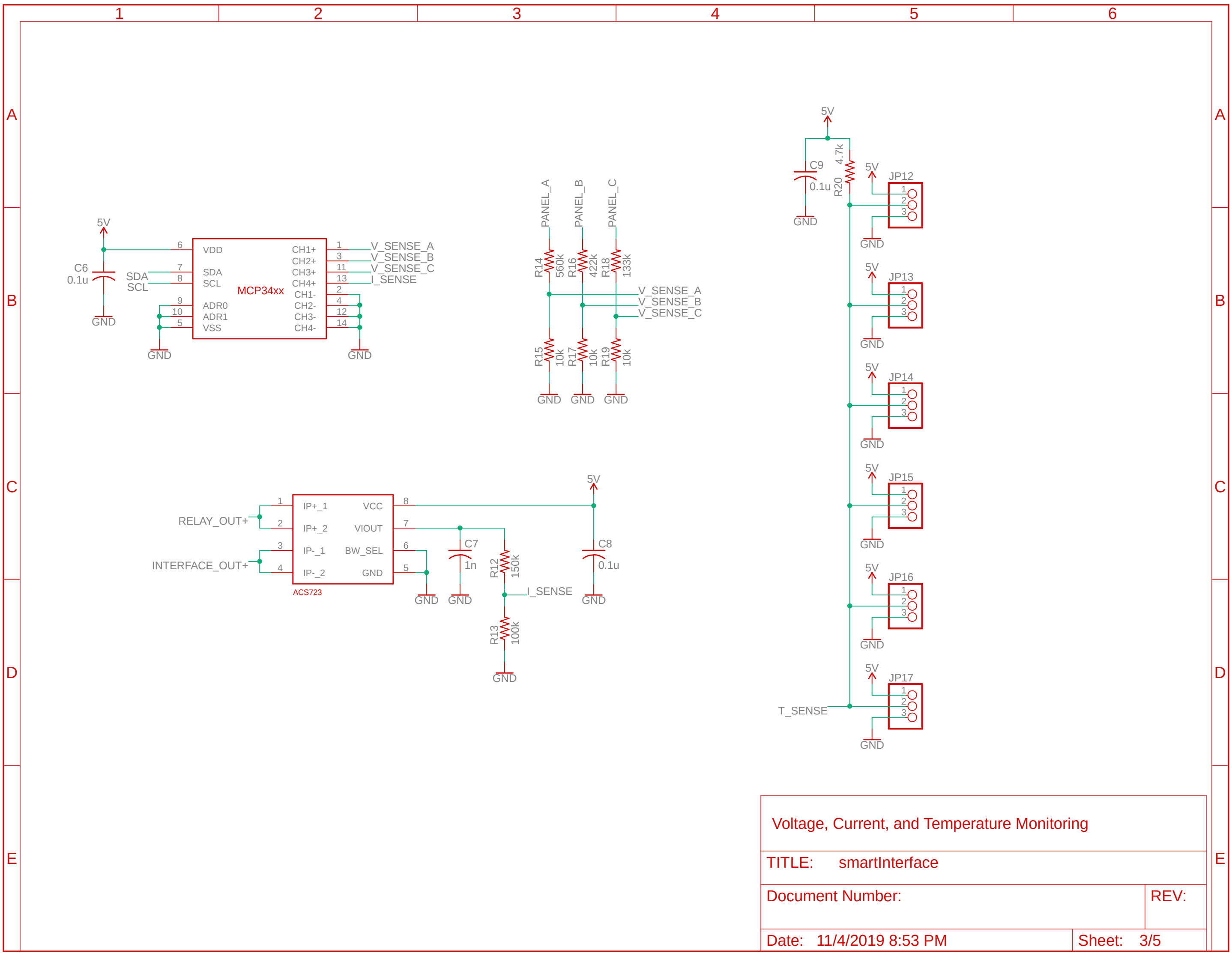
TITLE: smartInterface

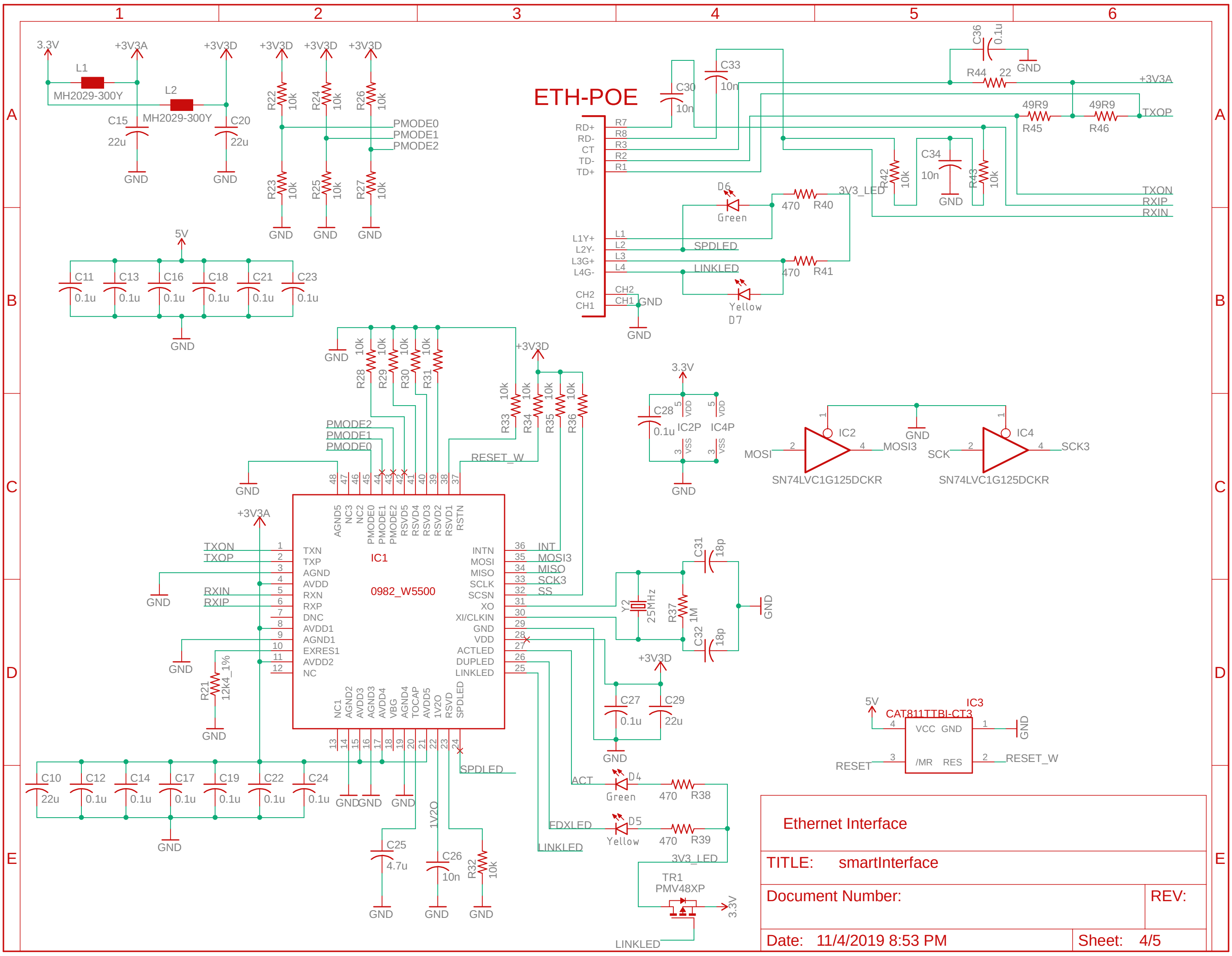
Document Number:

REV:

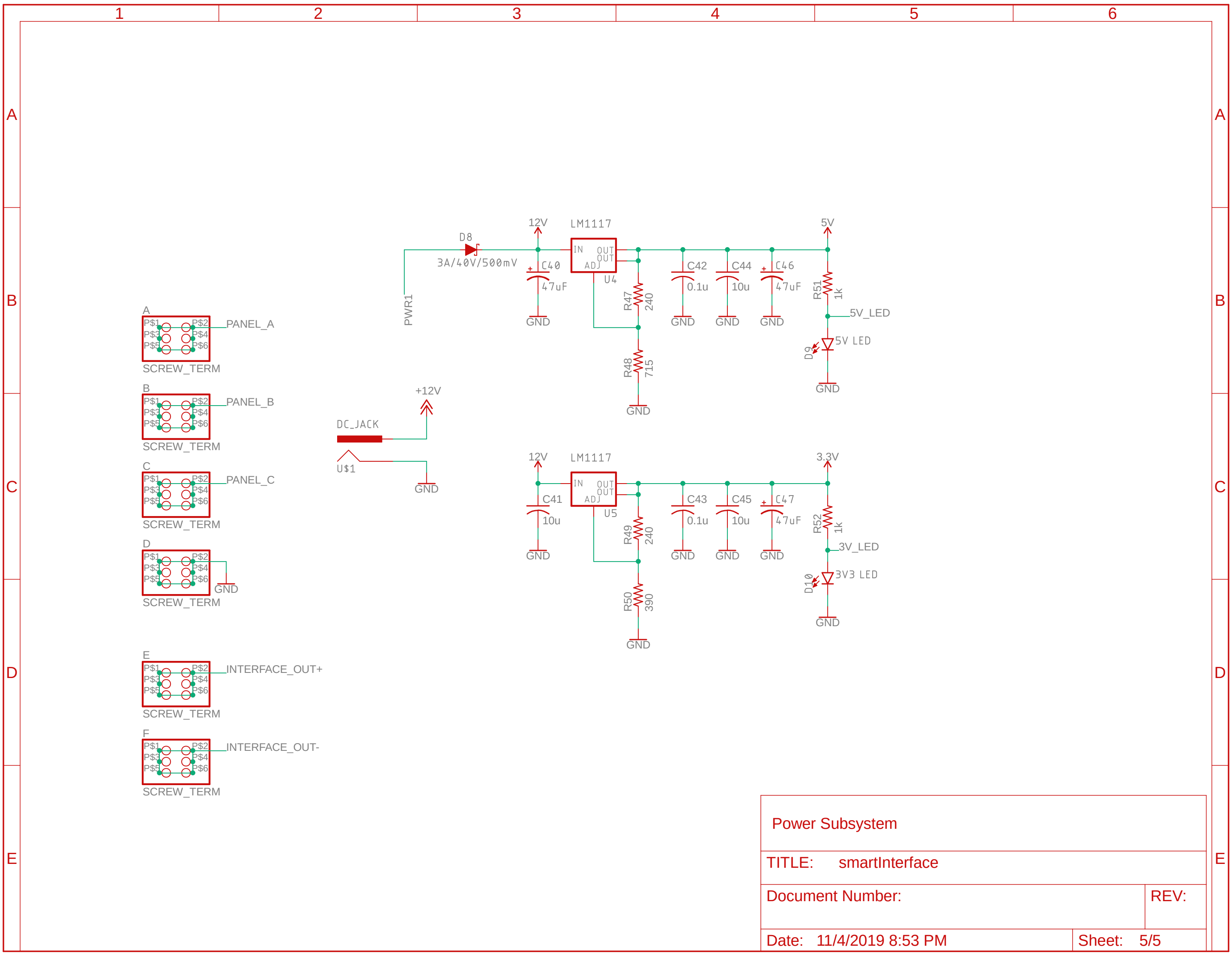
Date: 11/4/2019 8:53 PM

Sheet: 2/5


MCP34xx
VDD
SDA
SCL
ADR0
ADR1
VSS
CH1+
CH2+
CH3+
CH4+
CH1-
CH2-
CH3-
CH4-
V_SENSE_A
V_SENSE_B
V_SENSE_C
I_SENSE
5V
C6
0.1u
GND
PANEL_A
PANEL_B
PANEL_C
R14 560k
R16 422k
R18 133k
R15 10k
R17 10k
R19 10k
ACS723
IP+_1
IP+_2
IP-_1
IP-_2
VCC
VIOUT
BW_SEL
RELAY_OUT+
INTERFACE_OUT+
C7 1n
C8 0.1u
R12 150k
R13 100k
C9 0.1u
R20 4.7k
JP12
JP13
JP14
JP15
JP16
JP17
T_SENSE
Voltage, Current, and Temperature Monitoring
TITLE: smartInterface
Document Number:
REV:
Date: 11/4/2019 8:53 PM
Sheet: 3/5



ETH-POE
IC1
0982_W5500
SN74LVC1G125DCKR
CAT811TTBI-CT3
Ethernet Interface
TITLE: smartInterface
Document Number:
REV:
Date: 11/4/2019 8:53 PM
Sheet: 4/5


D8
3A/40V/500mV
PWR1
12V
LM1117
U4
C40
47uF
R47
240
R48
715
C42
0.1u
C44
10u
C46
47uF
5V
R51
1k
5V_LED
D9
5V LED
GND
DC_JACK
U$1
+12V
LM1117
U5
C41
10u
R49
240
R50
390
C43
0.1u
C45
10u
C47
47uF
3.3V
R52
1k
3V_LED
D10
3V3 LED
PANEL_A
PANEL_B
PANEL_C
INTERFACE_OUT+
INTERFACE_OUT-
SCREW_TERM
Power Subsystem
TITLE: smartInterface
Document Number:
REV:
Date: 11/4/2019 8:53 PM
Sheet: 5/5